UNIVERZA V LJUBLJANI
FAKULTETA ZA MATEMATIKO IN FIZIKO
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Računalništvo in matematika – 2. stopnja

Mitja Rozman

RAZPOREJANJE PRILAGODLJIVIH PONUDB ENERGIJE Z MEŠANIM CELOŠTEVILSKIM LINEARNIM PROGRAMIRANJEM

Magistrsko delo

Mentor: doc. dr. Alen Orbanić

Somentorica: dr. Tea Tušar

Ljubljana, 2019

Kazalo

Program dela

V magistrskem delu predstavite in formalno definirajte problem razporejanja prilagodljivih ponudb energije izhajajoč iz [7] in [17]. Predstavite mešano celoštevilsko linearno programiranje in naredite model problema razporejanja v obliki mešanega celoštevilskega linearnega programa. Pri tem si pomagajte z literaturo [9], [15] in [19]. Dodatno implementirajte model mešanega celoštevilskega linearnega programa za problem razporejanja s pomočjo Googlove knjižnice OR-Tools za matematično optimizacijo. Tako dobljeno programsko rešitev eksperimentalno ovrednotite.

Osnovna literatura

[7] B. Filipič, K. Gantar in T. Tušar, *GOFLEX-Scheduling: Specifications of optimization algorithms to schedule flexible offers for electricity production and consumption*, teh. por. IJS-DP 12400, Institut "Jožef Stefan", Ljubljana, 2018

[9] J. Grasselli in A. Vadnal, *Linearna algebra. Linearno programiranje*, Matematika-fizika: zbirka univerzitetnih učbenikov in monografij **4**, DMFA – založništvo, 2003

[15] J. E. Mitchell, *Branch-and-cut algorithms for combinatorial optimization problems*, v: Handbook of Applied Optimization, Oxford University Press, Oxford, 2000

[17] L. Šikšnys in dr., *D2.1 Automatic trading platform requirement & interface specification*, 2017, [ogled 10. 9. 2018], dostopno na `https://goflex-project.eu/Deliverables.html`

[19] R. J. Vanderbei, *Linear Programming: Foundations and Extensions*, Springer, New York, 2008

Podpis mentorja: Podpis somentorice:

Razporejanje prilagodljivih ponudb energije z mešanim celoštevilskim linearnim programiranjem

Povzetek

V magistrskem delu formalno definiramo problem razporejanja prilagodljivih ponudb električne energije. To so ponudbe proizvajalcev ali porabnikov električne energije, ki jih lahko distributer električnega omrežja prilagaja svojim potrebam. V delu naredimo model problema razporejanja v obliki mešanega celoštevilskega linearnega programa. Model implementiramo z uporabo Googlove knjižnice OR-Tools za matematično optimizacijo. Tako dobljeno programsko rešitev nato z uporabo obstoječih reševalnikov eksperimentalno ovrednotimo na iz prakse izhajajočih namišljenih scenarijih.

Scheduling flexible energy offers using mixed-integer linear programming

Abstract

In this master thesis we formaly define the problem of scheduling flexible energy offers. Flexible energy offers are offers from producers and consumers of elecrical energy to be scheduled by the Distribution System Operator. In the thesis we make a model of the scheduling problem in the form of mixed integer linear program. We implement the model using the Google Library OR-Tools for mathematical optimization. We experimentally evaluate this solution on simulated industry scenarios with the use of existing mixed integer linear program solvers.

Math. Subj. Class. (2010): 90B35, 90B80, 90C05, 90C06, 90C09, 90C11, 90C90
Ključne besede: mešano celoštevilsko linearno programiranje, matematično modeliranje, problem razporejanja

Keywords: mixed integer linear programming, mathematical modeling, scheduling problem

1 Uvod

Raba energije je ena od pomembnejših gonilnih sil človeškega napredka. Ljudje smo že v prazgodovini uporabljali ogenj za ogrevanje in obdelavo kovin. Kasneje smo se naučili učinkovito uporabljati energijo rek in vetra. V moderni dobi smo začeli širše koristiti tudi neobnovljivo energijo goriv in jedrsko energijo iz razcepa atomov.

Pri moderni rabi energije ima pomembno vlogo električna energija, energija, ki je direktno iz okolja še ne znamo pridobivati, znamo pa jo pretvoriti iz ostalih energij. Glavna prednost električne energije je, da je enostavna za prenos in uporabo. Kot primer si zamislimo, kako raba električne energije napram gorivom poenostavi javno razsvetljavo. Zato je električna energija prevladujoča vrsta energije v človeških aktivnostih. Vendar pa ima tudi veliko slabost, večjih količin električne energije se namreč ne more enostavno skladiščiti za kasnejšo uporabo.

Cena električne energije je zato zelo spremenljiva. Kadar jo je na voljo veliko, je poceni, kadar pa jo primanjkuje, se podraži. Ljudje v vsakdanjem življenju to na primer izkusijo med vikendi, ko je energija zaradi zmanjšanega delovanja industrije cenejša. Na nemškem električnem trgu je zaradi velike količine energije iz obnovljivih virov (predvsem vetrne in sončne) v določenih trenutkih prišlo celo do negativne cene [12]. Razlog za to so veliki stroški zaustavitve in zagona električnih obratov ter skladiščenja energije v električnem omrežju. Proizvodnja obnovljivih virov energije je namreč odvisna od vremenskih razmer in zato bolj nepredvidljiva od proizvodnje tradicionalnih virov energije (fosilna goriva, jedrska energija, ipd.). Zato operater distribucijskega omrežja pri veliki količini obnovljivih virov energije težko napove kakšna bo proizvodnja energije. Problemi nastanejo tudi v času pomanjkanja, ko cena električne energije naraste. Zato bi bilo dobrodošlo razporejanje električne energije za zagotavljanje nemotene dobave ter stabiliziranje stroškov in posledično cene energije. Smiselno je vzpostaviti sistem, v katerem bi imeli možnost razporejanja električne energije in bi lahko primanjkljaj le-te v nekem trenutku izničili z viškom, ki se pojavi enkrat drugič.

S tem izzivom se ukvarja evropski projekt GOFLEX [17] iz programa Obzorje 2020, katerega cilj je stroškovno učinkovita uporaba prilagodljivih virov električne energije na regionalni ravni. Institut "Jožef Stefan" sodeluje s partnerjem na projektu, podjetjem INEA (informatizacija, energetika, avtomatizacija), pri razvoju sistema za učinkovito razporejanje prilagodljivih ponudb energije. V okviru tega sodelovanja smo definirali problem razporejanja prilagodljivih ponudb energije kot problem mešanega celoštevilskega linearnega programiranja, zanj razvili model ter vzpostavili sistem, ki problem reši z uporabo obstoječih reševalnikov, ter delovanje sistema ovrednotili na tri načine, (1) na testnih scenarijih s poznano optimalno rešitvijo, (2) z analizo vpliva določenih parametrov problema na učinkovito delovanje sistema ter (3) s primerjavo uporabljenega reševalnika z drugim. Vse to je opisano v pričujočem magistrskem delu.

1.1 Pregled vsebine

V prvem delu uvoda smo navedli motivacijo magistrskega dela in ga umestili v konkreten projekt. Sedaj naredimo še pregled vsebine magistrskega dela. V poglavju 2 podrobneje opišemo problem razporejanja prilagodljivih ponudb energije. V poglavju 3 podamo matematično definicijo problema razporejanja, ki služi kot osnova za izdelavo matematičnega modela problema razporejanja. Problem razporejanja podamo kot model v obliki mešanega celoštevilskega linearnega programa. V poglavju 4 predstavimo teoretično ozadje mešanega celoštevilskega linearnega programiranja. Podamo matematično osnovo simpleksne metode, kot tudi nekatere podrobnosti algoritma razveji in odreži, ki se uporablja v izbranem reševalniku za mešano celoštevilsko linearno programiranje. V poglavju 5 navedemo nekaj zanimivih splošnih prijemov za izdelavo modelov zapisanih v obliki mešanega celoštevilskega linearnega programa. V poglavju 6 podamo glavni rezultat magistrskega dela, to je zapis problema razporejanja prilagodljivih ponudb energije v obliki mešanega celoštevilskega linearnega programa. V poglavju 7 orišemo implementacijo modela. Predstavimo razpoložljiva orodja, podamo pa tudi nekaj vpogleda v dejansko implementacijo končnega sistema, ki uporablja prosto dostopen reševalnik COIN-OR Branch and Cut [8]. V poglavju 8 predstavimo testne scenarije s poznano optimalno rešitvijo in eksperimentalno ovrednotimo rezultate izbranega reševalnika COIN-OR Branch and Cut. Nato analiziramo vpliv parametrov problema razporejanja na učinkovitost izbranega reševalnika. Naredimo tudi primerjavo izbranega reševalnika z reševalnikom Gurobi [10]. V zadnjem poglavju 9 navedemo še zaključke, kjer povzamemo in ovrednotimo rezultate magistrskega dela.

2 Razporejanje prilagodljivih ponudb energije

Oglejmo si problem razporejanja prilagodljivih ponudb energije. V nadaljevanju ga bomo najprej umestili v projekt GOFLEX. Nato si bomo pogledali definicijo vhodnih podatkov, ki jih za konkretno instanco problema podamo kot scenarij razporejanja.

2.1 Ideja projekta GOFLEX

Projekt GOFLEX [17] razvija informacijsko-tehnološke infrastrukture, ki bodo operaterjem distribucijskih omrežij omogočile učinkovito upravljanje z razpršeno porabo in proizvodnjo električne energije. Ideja projekta je, da končni uporabniki omrežja, tako proizvajalci kot porabniki, ponudijo svojo proizvodnjo ali porabo v obliki prilagodljivih ponudb, ki jih operaterji distribucijskega omrežja lahko izkoristijo za izravnavo odstopanj v omrežju. Te ponudbe so prilagodljive v času uporabe in/ali količini energije. V zameno za prilagodljivost uporabniki dobijo denarne ugodnosti.

Stanje energije v električnem omrežju je odvisno od količine energije, ki jo omrežje prejme od proizvajalcev in količine energije, ki jo omrežje odda porabnikom. Razlika med obema je energijsko odstopanje, ki ga mora operater omrežja izravnati bodisi s prodajo odvečne ali zakupom manjkajoče energije. V vsakem primeru energijsko odstopanje za operaterja predstavlja strošek in dodatno delo. Problem je tudi v tem, da je energijsko odstopanje težko vnaprej napovedati, saj sta tako proizvodnja kot poraba energije odvisni od številnih dejavnikov.

Planiranje oziroma napoved proizvodnje tradicionalnih generatorjev in obratov kot so termoelektrarne, hidroelektrarne in nuklearne elektrarne, ne povzroča težav. Vendar se je v zadnjem času povečal delež proizvodnje iz obnovljivih virov energije, kar predstavlja problem stabilnosti elektro-energetskega sistema, saj je poleg porabe potrebno napovedovati tudi proizvodnjo (predvsem obnovljivih virov energije), ki je odvisna od vremenskih razmer. Zato je uravnavanje porabe s proizvodnjo oteženo. Tudi za proizvodnjo imamo samo napoved, ki se lahko razlikuje od dejanskega stanja. Za napoved porabe se običajno upošteva:

- obdobje leta, kar vpliva predvsem na energijo za ogrevanje;
- delovni dnevi, kar je pomembno zaradi industrije in ostalih poslovnih dejavnosti;
- obdobje dneva, ker je poraba odvisna od ustaljenih ciklov človeškega dnevnega delovanja;
- druge dejavnike, kot na primer širjenje mesta, ki doprinese večjo porabo.

Pri klasičnih elektro-energetskih sistemih se izračuna razlika med napovedano proizvodnjo in porabo energije, to je napovedano energijsko odstopanje, ki predstavlja določen strošek. V primeru negativnega napovedanega energijskega odstopanja mora operater omrežja manjkajočo energijo dokupiti, v primeru pozitivnega napovedanega energijskega odstopanja pa mora odvečno energijo prodati, skladiščiti ali porabiti. Po končanem trgovanju mora biti energijsko odstopanje izničeno. Iz slike 1 je razvidno, kako proizvodnja in poraba energije privedeta do energijskega odstopanja. Oglejmo si, kako bi takšno stanje lahko izboljšali s sistemom, v katerem lahko s pomočjo prilagodljivih ponudb zmanjšamo energijsko odstopanje. V tem primeru ločimo napovedano neprilagodljivo energijo proizvodnje in porabe ter prilagodljive ponudbe proizvodnje in porabe. Na začetku so prilagodljive ponudbe razporejene po privzetem urniku izvajanja (po privzetem razporedu). Skupaj z neprilagodljivo energijo proizvodnje in porabe tvorijo celotno napoved za energijsko odstopanje. Naša naloga je, da napovedano energijsko odstopanje čim bolj zmanjšamo. Zato uporabimo časovno in energijsko prilagodljivost ponudb, kot je prikazano na sliki 2 in s tem zmanjšamo napovedano energijsko odstopanje, kot je prikazano na sliki 3. Tako smo spremenili urnik izvajanja prilagodljive ponudbe. Na slikah sta prilagodljivi ponudbi prikazani kot pravokotnika. Pri prvi prilagodljivi ponudbi (levi ponudbi s slik 2 in 3) koristimo časovno prilagodljivost, pri drugi (desni ponudbi s slik 2 in 3) pa energijsko prilagodljivost.

V praksi je lahko veliko virov prilagodljivih ponudb (glej [5]). Naštejmo jih le nekaj. Primeri prilagodljivih ponudb proizvajalcev energije:

- kogeneracijske enote,
- plinski agregati,
- dizelski generatorji.

Primeri prilagodljivih ponudb porabnikov energije:

- toplotne črpalke, grelci, hladilniki;
- livarske peči, talilnice, sušilnice;
- električna vozila.

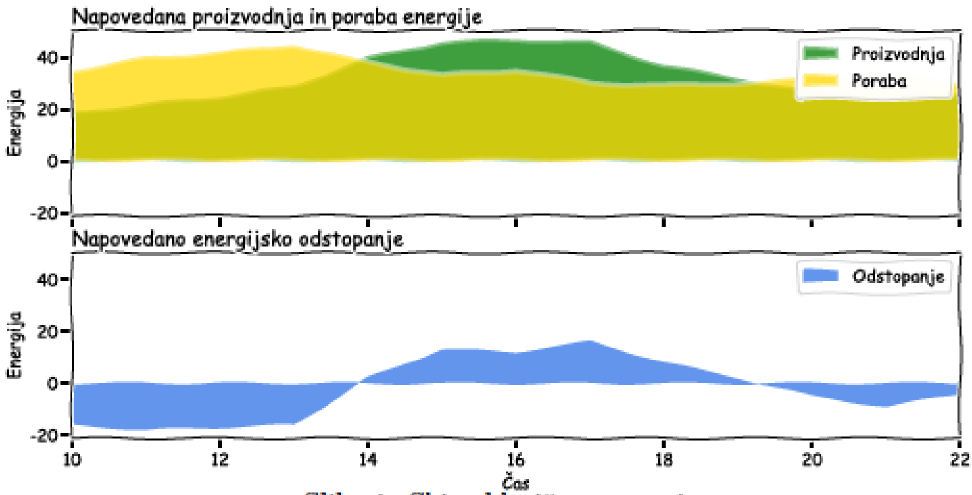
Poleg naštetih virov prilagodljivih ponudb poudarimo še baterije, ki lahko služijo kot proizvajalci in porabniki električne energije.

Prilagodljive ponudbe želimo razporediti tako, da bo napovedano energijsko odstopanje čim manjše oziroma natančneje, da bodo stroški celotnega scenarija, tako uporabe prilagodljivih ponudb, kot tudi stroškov za izničenje napovedanega odstopanja, čim manjši. Za optimalno izravnavanje odstopanj potrebujemo algoritem razporejanja prilagodljivih ponudb, ki za vsak podan scenarij razporejanja prilagodljivih ponudb minimizira stroške operaterja pri izravnavi napovedanega energijskega odstopanja. Ta algoritem za scenarij razporejanja vrne razpored prilagodljivih ponudb, končno napovedano energijsko odstopanje in predvidene stroške scenarija.

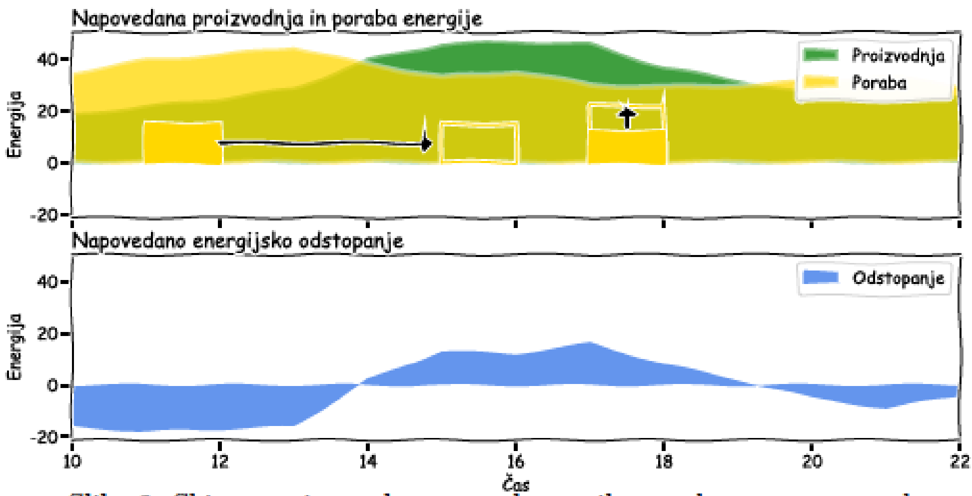
Zaradi velikega števila prilagodljivih ponudb je včasih dobro, da se jih združi v večje enote. Tudi te so definirane kot prilagodljive ponudbe. Prednost tega je, da se zmanjša število prilagodljivih ponudb ter posledično olajša delo algoritma za razporejanje prilagodljivih ponudb. Zaradi združevanja lahko predpostavimo, da bomo v enem scenariju razporejanja razporejali največ do nekaj sto prilagodljivih ponudb. Več o postopku združevanja prilagodljivih ponudb in projektu GOFLEX je na voljo v spletni dokumentaciji [17] ter prispevkih [5] in [6]. V nadaljevanju si poglejmo scenarij razporejanja in njegov formalen opis za potrebe algoritmičnih prijemov.

2.2 Scenarij razporejanja

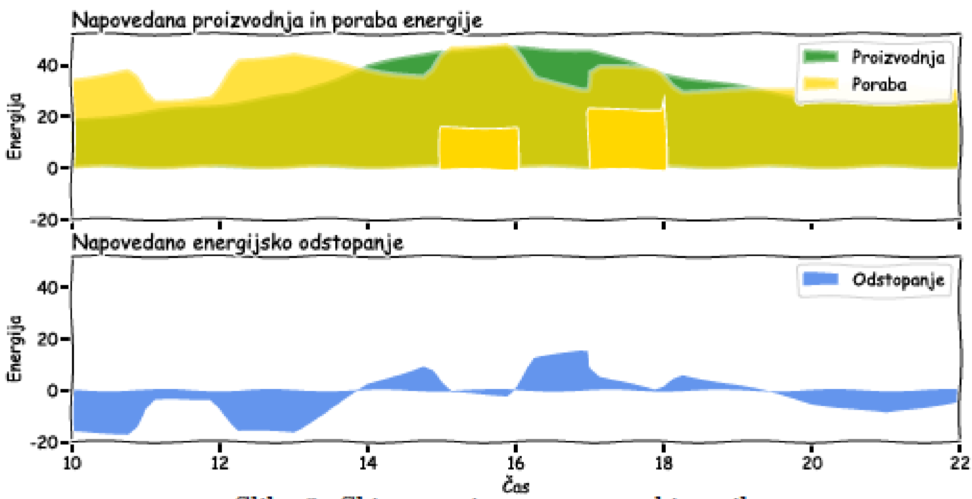
Scenarij razporejanja vsebuje podatke, ki opisujejo posamezen problem razporejanja prilagodljivih ponudb, to je podatke o časovnih intervalih, energijskem odstopanju ter prilagodljivih ponudbah. To so vhodni podatki za algoritem razporejanja prilagodljivih ponudb, ki vrne končni razpored prilagodljivih ponudb in predvidene stroške. Pri uvedbi novih pojmov so pogosto podani tudi angleški prevodi, ki služijo



Slika 1: Skica klasičnega stanja



Slika 2: Skica stanja pred spremembo urnika z nakazano spremembo



Slika 3: Skica stanja po spremembi urnika

kot pojasnilo pozneje uporabljenih oznak in simbolov.

2.2.1 Diskretizacija časa in časovni intervali

Problem razporejanja prilagodljivih ponudb je glede na čas podan kot diskreten problem. Podana sta *začetni čas* (angl. start time) in *končni čas* (angl. end time) izvajanja scenarija razporejanja ter *dolžina časovnega intervala* (angl. interval duration), ki je privzeto 15 minut. Implicitno je s tem podano tudi število časovnih intervalov scenarija, saj je število teh enako razliki med končnim in začetnim časom deljenim z dolžino časovnega intervala. Razdelitev na časovne intervale je pomembna, saj nam omogoča enostavno podajanje vhodnih podatkov in reševanje problema.

2.2.2 Energijsko odstopanje

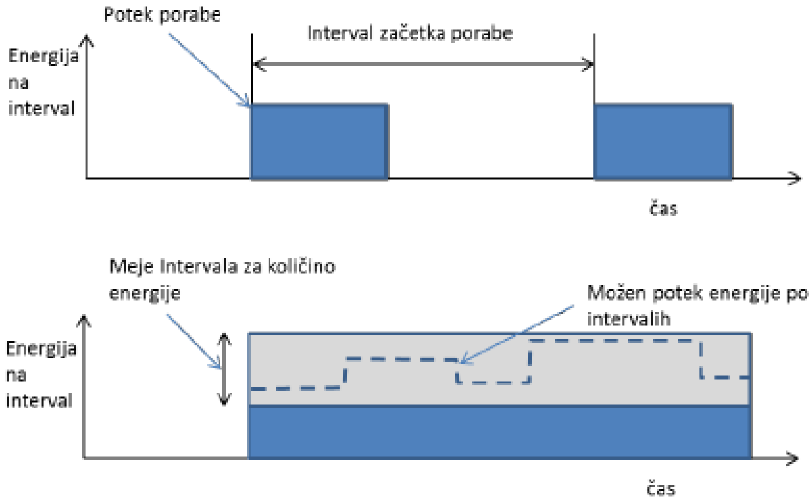
Energija odstopanja (angl. imbalance energy) je izračunana iz razlike napovedi proizvodnje energije (angl. energy production) in porabe energije (angl. energy consumption). *Začetna energija odstopanja* je napovedana energija odstopanja za trenutni scenarij pri privzetem urniku razporeda. *Končna energija odstopanja* pa je napovedana energija odstopanja pri urniku razporeda, ki ga vrne algoritem razporejanja.

Energija odstopanja je definirana na časovnih intervalih, poleg nje sta za vsak časovni interval podani tudi ceni pozitivne in negativne energije odstopanja. Ceni sta podani na enoto energije. Energijsko *odstopanje* (angl. imbalance) je pojem, ki zaobjema tako podatke o energiji odstopanja, kot tudi o pripadajočima cenama. Strošek energijskega odstopanja računamo iz energije odstopanja in njej pripadajočima cenama.

V scenariju je po intervalih podana začetna energija odstopanja ter pripadajoči ceni za negativno in pozitivno energijo odstopanja. Skupaj tvorijo začetno energijsko odstopanje zapisano z vektorji energije in cen. Napovedan strošek energijskega odstopanja je enak skalarnemu produktu komponent končne energije odstopanja s cenami odvisnimi od predznaka komponent končne energije odstopanja.

2.2.3 Prilagodljive ponudbe

Prilagodljiva ponudba (angl. flex-offer) je ponudba proizvajalca ali porabnika električne energije, s katero nudi svojo časovno in/ali energijsko prilagodljivost operaterju distribucijskega omrežja, to je sestavljalcu urnika izvajanja prilagodljivih ponudb. *Časovna prilagodljivost* (angl. time flexibility) pomeni, da se dana prilagodljiva ponudba lahko začne ob različnih časih. Natančneje to pomeni, da se prilagodljiva ponudba lahko začne v enem od časovnih intervalov znotraj množice možnih začetnih časovnih intervalov prilagodljive ponudbe. *Energijska prilagodljivost* (angl. energy flexibility) pomeni, da se prilagodljivo ponudbo na intervalih njenega delovanja lahko energijsko uravnava. Natančneje to pomeni, da lahko na določenih intervalih prilagodljive ponudbe nastavimo energijo znotraj vnaprej določenih omejitev. Na sliki 4 je ponazorjena časovna in energijska prilagodljivost prilagodljive ponudbe. Na vrhnjem grafu je prikazana časovna prilagodljivost, kjer lahko prilagajamo začetek uporabe prilagodljive ponudbe. Na spodnjem grafu je



Slika 4: Oblika spremenljive ponudbe za prilagajanje: časovno prilagajanje (zgoraj) in prilagajanje energije (spodaj), avtor: dr. Gregor Černe, vir [6].

prikazana energijska prilagodljivost, kjer za dane časovne intervale znotraj prilagodljive ponudbe lahko spreminjamo potek energije po intervalih. Pri danem primeru sta spodnja in zgornja meja energije enaki za vse intervale, v splošnem pa to ne drži nujno. Spodnja in zgornja meja energije sta namreč za vsak interval znotraj območja delovanja prilagodljive ponudbe lahko drugačni. Dodatno ima prilagodljiva ponudba lahko hkrati časovno in energijsko prilagodljivost.

Prilagodljiva ponudba lahko vsebuje tudi druge omejitve, ki opisujejo dodatne zahteve glede časovnega poteka pretoka energije ponudnika prilagodljive ponudbe. V nadaljevanju si poglejmo vse parametre prilagodljive ponudbe, ki jih kasneje uporabimo v formalni definiciji problema razporejanja. Prilagodljivo ponudbo opišemo z naslednjimi parametri in omejitvami:

1. Časovna prilagodljivost je podana z dvema parametroma, in sicer prvim možnim začetnim časom in zadnjim možnim začetnim časom. Vsi časovni intervali med tema časoma so možni intervali začetka prilagodljive ponudbe.

2. *Dolžina* (angl. length) je podana kot število intervalov, ki jih prilagodljiva ponudba zavzame.

3. Energijska prilagodljivost je podana za vse intervale znotraj območja delovanja prilagodljive ponudbe. Teh intervalov je toliko, kot je dolžina prilagodljive ponudbe. Za vsakega od teh relativnih časovnih intervalov sta podani najmanjša in največja dovoljena energija za ta relativni časovni interval. Če prilagodljiva ponudba ni energijsko prilagodljiva, potem sta najmanjša in največja dovoljena energija enaki.

4. *Skupna energija* (angl. total energy) je enaka seštevku energij po vseh intervalih prilagodljive ponudbe. Omejitev skupne energije je podana z najmanjšo in največjo dovoljeno skupno energijo. Omejitev ni obvezno podana.

5. *Kumulativna energija* (angl. cumulative energy) je definirana za vsak časovni interval prilagodljive ponudbe in je enaka vsoti energij intervalov prilagodljive ponudbe do danega časovnega intervala. Uporabljamo jo pri opisu prilagodljivih ponudb, katerih vir so baterije. Omejitev kumulativne energije je podana z najmanjšo in največjo dovoljeno kumulativno energijo. V praksi omejitev kumulativne energije preprečuje, da bi se baterija prekomerno izpraznila ali napolnila v času izvajanja urnika prilagodljive ponudbe. Velja omeniti, da je kumulativna energija zadnjega intervala prilagodljive ponudbe enaka skupni energiji prilagodljive ponudbe. Omejitev ni obvezno podana.

6. *Obvezni razpored* (angl. obligatory assignment) je parameter, ki pove ali je potrebno prilagodljivo ponudbo nujno uporabiti. Za prilagodljivo ponudbo imamo dve možnosti, razpored je lahko obvezen ali pa neobvezen.

7. *Privzeti urnik izvajanja* (angl. default schedule) je lahko podan, kadar moramo prilagodljivo ponudbo obvezno uporabiti. Privzeti urnik izvajanja je začetni urnik izvajanja prilagodljive ponudbe, katerega energija je že všteta v začetno energijsko odstopanje. Podan je z začetnim časom in intervali energije, ki si sledijo po vrsti od začetnega intervala. Časovne intervale prilagodljive ponudbe, ki se prekrivajo s privzetim urnikom izvajanja, obravnavamo drugače kot ostale intervale prilagodljive ponudbe, saj moramo za izračun stroška teh intervalov upoštevati spremembo energije glede na privzeti urnik izvajanja. Če je energija intervala enaka energiji sovpadajočega intervala privzetega urnika izvajanja, je strošek na tem intervalu enak nič. Več podrobnosti o posebni obravnavi cene intervalov privzetega urnika izvajanja je na voljo v razdelku 3.2.2. Množico vseh privzetih urnikov izvajanja prilagodljivih ponudb določenega scenarija poimenujmo *privzeti razpored*.

8. *Cena energije* (angl. energy price) je podana za vsak interval znotraj možnega delovanja prilagodljive ponudbe.

Optimizacijski algoritem za podan scenarij vrne *končni razpored* prilagodljivih ponudb. Urniku izvajanja posameznih prilagodljivih ponudb rečemo *izhodni urnik izvajanja*. Pomeni izbor intervalov in izbor energij na teh intervalih. Izbranim intervalom prilagodljive ponudbe rečemo tudi *intervali uporabe* prilagodljive ponudbe.

Prvotne specifikacije scenarija obravnavanega v projektu GOFLEX so zapisane v [7]. Tekom projekta so se specifikacije prilagodljivih ponudb spreminjale in dopolnjevale. Končno obliko specifikacij prilagodljivih ponudb formalno definiramo v naslednjem poglavju.

3 Matematična formulacija problema razporejanja

Razporejanje prilagodljivih ponudb energije je optimizacijski problem, pri katerem poskušamo čimbolj zmanjšati realne stroške z učinkovitim razporejanjem prilagodljivih ponudb. Za uspešno reševanje realnega problema je ključna določitev dobre matematične formulacije, ki služi kot osnova za izgradnjo modela, s katerim bomo problem reševali. Uspešno reševanje problema je mišljeno v smislu doseganja optimalnih ali dovolj dobrih rešitev v čim krajšem času.

V tem poglavju znova podamo scenarij razporejanja, tokrat v formalnem matematičnem zapisu. Nato sledi opis optimizacijskega kriterija, kjer definiramo izračun stroškov. Na koncu formalno zapišemo še omejitve, ki jih moramo upoštevati pri modeliranju.

3.1 Scenarij razporejanja

Podan imamo scenarij s časovnimi intervali, začetnim energijskim odstopanjem in prilagodljivimi ponudbami.

3.1.1 Diskretizacija časa in časovni intervali

Podano imamo množico časovnih intervalov $\mathcal{N}$, katere moč je enaka n. To je število časovnih intervalov na katerih je definiran scenarij. Množico definiramo kot $\mathcal{N} = \{0, 1, \ldots, n-1\}$, torej časovne intervale predstavimo kar z ničlo in naravnimi števili. Navadno en časovni interval traja 15 minut, tako da se scenarij v realnem času odvija $15n$ minut.

3.1.2 Prilagodljive ponudbe

Scenarij razporejanja vsebuje množico prilagodljivih ponudb $\mathcal{F}$. Za prilagodljivo ponudbo $f \in \mathcal{F}$ vpeljimo naslednje oznake:

1. Množico možnih začetnih časovnih intervalov prilagodljive ponudbe f označimo z $\mathcal{I}_f$, posamezen začetni časovni interval iz te množice pa z i.

2. Dolžino prilagodljive ponudbe f označimo z l_f. Množico relativnih časovnih intervalov delovanja prilagodljive ponudbe podamo z

$$\mathcal{L}_f = \{0, 1, \ldots, l_f - 1\},$$

kjer posamezen relativni časovni interval delovanja prilagodljive ponudbe označimo z j.

3. Vse možne časovne intervale delovanja prilagodljive ponudbe f podamo s $\mathcal{T}_f$, le-ti so enaki

$$\mathcal{T}_f = \{i + j \mid i \in \mathcal{I}_f, j \in \mathcal{L}_f\}.$$

Ker je $\mathcal{T}_f \subseteq \mathcal{N}$, posamezen časovni interval prilagodljive ponudbe navadno označimo s t. Pri začetku prilagodljive ponudbe ob času $i \in \mathcal{I}_f$ je $\{i + j \mid j \in \mathcal{L}_f\}$ množica intervalov delovanja prilagodljive ponudbe.

4. Za vsak možen časovni interval prilagodljive ponudbe $t \in \mathcal{T}_f$ energijo prilagodljive ponudbe označimo z $E_{f,t}$.

5. Za vsak relativni časovni interval prilagodljive ponudbe $j \in \mathcal{L}_f$ imamo podano najmanjšo $E_{f,j}^{\min}$ in največjo $E_{f,j}^{\max}$ dovoljeno energijo na tem intervalu.

6. Skupno energijo prilagodljive ponudbe f, ki je enaka vsoti energij po intervalih izvajanja, označimo z

$$E_f = \sum_{t \in \mathcal{T}_f} E_{f,t}.$$

   Najmanjšo in največjo dovoljeno skupno energijo označimo z $E_f^{\mathrm{T}\min}$ in $E_f^{\mathrm{T}\max}$.

7. V primeru, ko je vir prilagodljive ponudbe baterija, njeno najmanjšo in največjo dovoljeno kumulativno energijo označimo z $E_f^{\mathrm{C}\min}$ in $E_f^{\mathrm{C}\max}$. Začetno stanje energije v bateriji označimo z E_f^{Cst}.

8. Za informacijo o tem, ali je razpored obvezen, ne bomo vpeljali posebne oznake.

9. Energijo privzetega urnika izvajanja D prilagodljive ponudbe f v časovnem intervalu $t \in \mathcal{T}_f$ označimo z $E_{f,t}^D$. Kadar prilagodljiva ponudba nima privzetega urnika izvajanja oziroma za časovni interval $t \in \mathcal{T}_f$ privzeti urnik izvajanja ni definiran, energijo $E_{f,t}^D$ nastavimo na 0.

10. Za vsak možen časovni interval prilagodljive ponudbe $t \in \mathcal{T}_f$ spremembo energije prilagodljive ponudbe od privzetega urnika izvajanja zapišemo z $\widetilde{E}_{f,t}$. Zanjo velja zveza

$$\widetilde{E}_{f,t} = E_{f,t} - E_{f,t}^D.$$

11. Za vsak možen časovni interval prilagodljive ponudbe $t \in \mathcal{T}_f$ imamo podano ceno spremembe električne energije od privzetega razporeda. Cena je odvisna od predznaka spremembe energije prilagodljive ponudbe glede na privzeti razpored

$$p_{f,t} = \begin{cases} p_{f,t}^+, & \widetilde{E}_{f,t} \geq 0, \\ p_{f,t}^-, & \widetilde{E}_{f,t} < 0. \end{cases}$$

    S $p_{f,t}^+$ in $p_{f,t}^-$ označimo ceno za pozitivno in negativno spremembo električne energije na časovnem intervalu $t \in \mathcal{T}_f$ glede na privzeti urnik izvajanja prilagodljive ponudbe f.

12. Za vsak časovni interval $t \in \mathcal{T}_f$ lahko že pred razporejanjem prilagodljivih ponudb izračunamo strošek privzetega urnika izvajanja $c_{f,t}^D$. Strošek privzetega urnika izvajanja je enak

$$c_{f,t}^D = \begin{cases} E_{f,t}^D p_{f,t}^-, & E_{f,t}^D \geq 0, \\ E_{f,t}^D p_{f,t}^+, & E_{f,t}^D < 0, \end{cases}$$

kar izpeljemo iz stroška spremembe električne energije od privzetega urnika izvajanja za $E_{f,t} = 0$. Na časovnih intervalih $t \in \mathcal{T}_f$, kjer privzeti urnik izvajanja ni definiran, je strošek privzetega urnika izvajanja enak 0. Strošek privzetega urnika izvajanja je konstanta in ni odvisen od razporeditve prilagodljive ponudbe.

3.1.3 Energijsko odstopanje

Energijsko odstopanje predstavlja odvečno ali manjkajočo energijo, ki je posledica razlike med napovedjo proizvodnje in porabe. V primeru, ko se poraba in proizvodnja ujemata, energijskega odstopanja ni, oziroma je enako nič. Energijsko odstopanje je določeno z vektorjem energij in pripadajočih cen za pozitivno in negativno energijo. Cene so podane na enoto energije.

Energijo odstopanja zapišemo kot $\boldsymbol{E}^{\mathrm{I}} = (E_0^{\mathrm{I}}, E_1^{\mathrm{I}}, \ldots, E_{n-1}^{\mathrm{I}})$, kjer je n število časovnih intervalov scenarija, oznaka I pa označuje, da gre za energijsko odstopanje. Na časovnem intervalu $t \in \mathcal{N}$ je energija odstopanja enaka E_t^{I}. Energijsko odstopanje pred razporejanjem prilagodljivih ponudb imenujemo začetno energijsko odstopanje. Začetno energijsko odstopanje zapišemo kot $\boldsymbol{E}^{\mathrm{Ist}} = (E_0^{\mathrm{Ist}}, E_1^{\mathrm{Ist}}, \ldots, E_{n-1}^{\mathrm{Ist}})$, kjer za potrebe razločevanja uporabimo posebno oznako Ist (st - kot *start* v angleščini).

Začetna energija odstopanja je v scenariju podana. V praksi jo določa vsota energije napovedanega energijskega odstopanja brez uporabe prilagodljivih ponudb $E_t^{\mathrm{I}^*}$ in energije privzetega razporeda $\sum_{f\in\mathcal{F}} E_{f,t}^{D}$. Za vsak časovni interval t je določena z naslednjo zvezo

$$E_t^{\mathrm{Ist}} = E_t^{\mathrm{I}^*} + \sum_{f\in\mathcal{F}} E_{f,t}^{D}.$$

Energijo napovedanega energijskega odstopanja brez uporabe prilagodljivih ponudb $E_t^{\mathrm{I}^*}$ imenujemo tudi *izhodiščno odstopanje*, kar zaznamujemo z *. Energija privzetega razporeda $\sum_{f\in\mathcal{F}} E_{f,t}^{D}$ pa je vsota energij privzetih urnikov izvajanja prilagodljivih ponudb.

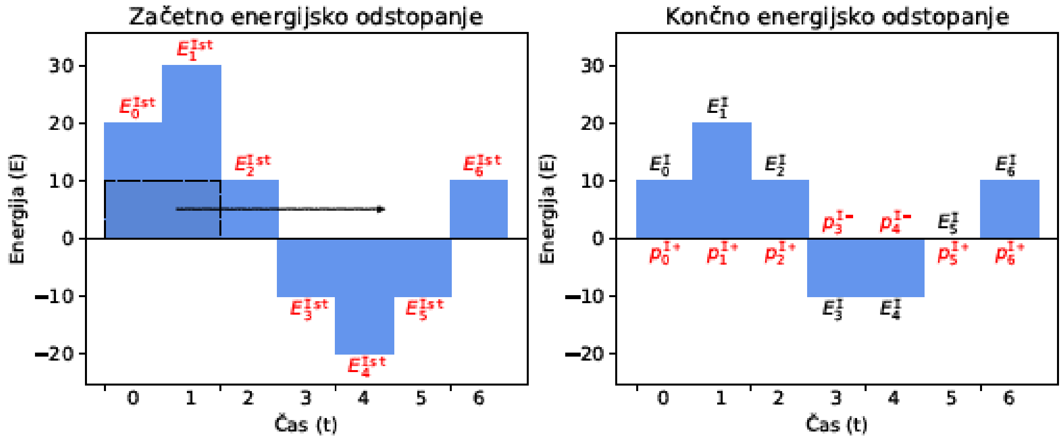
Končna energija odstopanja je preostala energija odstopanja po razporeditvi prilagodljivih ponudb. Energija odstopanja E_t^{I} je enaka vsoti začetne energije odstopanja E_t^{Ist} in spremembi energij prilagodljivih ponudb glede na njihov privzeti razpored $\sum_{f\in\mathcal{F}} \widetilde{E}_{f,t}$. Energijo odstopanja za vsak časovni interval $t \in \mathcal{N}$ zapišemo z naslednjo zvezo

$$E_t^{\mathrm{I}} = E_t^{\mathrm{Ist}} + \sum_{f\in\mathcal{F}} \widetilde{E}_{f,t}.$$

Za formalno definicijo posameznih energij $E_{f,t}^{D}$ in $\widetilde{E}_{f,t}$ glej razdelek 3.1.2.

Cena energijskega odstopanja je podana za vsak časovni interval $t \in \mathcal{N}$ in se razlikuje glede na predznak energijskega odstopanja. Podana je kot

$$p_t^{\mathrm{I}} = \begin{cases} p_t^{\mathrm{I}+}, & E_t^{\mathrm{I}} \geq 0, \\ p_t^{\mathrm{I}-}, & E_t^{\mathrm{I}} < 0, \end{cases}$$



Slika 5: Energijsko odstopanje pred in po razporejanju ene prilagodljive ponudbe

kjer je p_t^{I+} cena pozitivnega energijskega odstopanja in p_t^{I-} cena negativnega energijskega odstopanja. Pozitivna energija odstopanja predstavlja odvečno energijo, ki jo mora operater omrežja prodati. Ponavadi operater s prodajo služi. Kadar operater pri prodaji služi je cena energije odstopanja negativna, saj za operaterja predstavlja dobiček (negativen strošek). Strošek energijskega odstopanja definiramo v razdelku 3.2.1.

Primer 3.1. Sedaj si oglejmo sliko 5, ki prikazuje začetno in končno energijsko odstopanje za hipotetičen scenarij s sedmimi časovnimi intervali. Na sliki 5 so z rdečim tekstom zapisani podatki energijskega odstopanja iz scenarija. S črtkano črto je na levem grafu ponazorjen privzeti urnik razporeda neke prilagodljive ponudbe, ki je časovno prilagodljiva. S puščico je nakazan njen premik na kasnejši čas. Končno energijsko odstopanje, ki ga s tem premikom dosežemo, je prikazano na desni sliki. Strošek končnega energijskega odstopanja pa se izračuna s pomočjo cene, ki je različna glede na predznak njegove energije.

3.2 Optimizacijski kriterij in namenska funkcija

V grobem je naš namen, da z razporejanjem prilagodljivih ponudb zmanjšamo začetno energijo odstopanja, kar se po navadi odraža v manjših stroških operaterja omrežja. Zato po časovnih intervalih scenarija razporejamo prilagodljive ponudbe in variiramo njihovo električno energijo, tako da kar se da zmanjšamo energijsko odstopanje. Ker pa imajo tako energijsko odstopanje kot tudi prilagodljive ponudbe za različne intervale različne cene, optimalna rešitev zmanjšanja stroškov ni vedno tista, ki minimizira energijo odstopanja. Optimalna rešitev je tista, kjer so pri razporejanju prilagodljivih ponudb upoštevane tudi cene, ki so velikega pomena za minimiziranje stroška. Dodatno moramo upoštevati, da je operater stroške prilagodljivih ponudb po privzetem urniku izvajanja že vračunal in ga zanimajo le stroški energijskega odstopanja in stroški spremembe urnikov od privzetega razporeda.

Pri danem scenariju želimo razporediti prilagodljive ponudbe tako, da maksimiziramo dobiček operaterja distribucijskega omrežja. Torej moramo minimizirati operaterjev strošek, ki nastane pri izvedbi urnika za dan scenarij. Po definiciji problema razporejanja je *operaterjev strošek* enak vsoti stroška energijskega odstopanja c_I in stroška spremembe urnika razporeda $\widetilde{c}_{\mathcal{F}}$, to je

$$\widetilde{c} = c_I + \widetilde{c}_{\mathcal{F}}.$$

Za potrebe modeliranja pa dodatno uvedemo izhodni strošek. *Izhodni strošek* (angl. cost) je enak vsoti stroška energijskega odstopanja c_I in stroška uporabe prilagodljivih ponudb $c_{\mathcal{F}}$ to je

$$c = c_I + c_{\mathcal{F}}.$$

Zaradi posebnega statusa privzetega razporeda je izhodni strošek smiselno razpisati na

$$c = c_I + c_{\mathcal{F}}^D + \widetilde{c}_{\mathcal{F}},$$

kjer je c_I strošek energijskega odstopanja, $c_{\mathcal{F}}^D$ strošek uporabe prilagodljivih ponudb po privzetem razporedu in $\widetilde{c}_{\mathcal{F}}$ sprememba stroška prilagodljivih ponudb od stroška privzetega razporeda.

Strošek privzetega razporeda $c_{\mathcal{F}}^D$ je glede na scenarij konstanten, zato lahko operaterjev strošek $\widetilde{c}$ vedno izračunamo preko izhodnega stroška c upoštevajoč naslednjo zvezo

$$\widetilde{c} = c - c_{\mathcal{F}}^D.$$

Obravnava spremembe stroškov prilagodljivih ponudb od privzetega razporeda $\widetilde{c}_{\mathcal{F}}$ je zahtevnejša kot obravnava stroškov uporabe prilagodljivih ponudb $c_{\mathcal{F}}$, zato za potrebe modeliranja izberemo izhodni strošek c za namensko funkcijo, ki jo želimo minimizirati. V nadaljevanju sledijo še definicije izračuna stroška energijskega odstopanja c_I in stroška spremembe urnika razporeda $\widetilde{c}_{\mathcal{F}}$ ter ponazoritev stroškov na primeru namišljenega scenarija.

3.2.1 Strošek energijskega odstopanja

Najprej si poglejmo strošek energijskega odstopanja c_I. Končno energijo odstopanja zapišemo z vektorjem $\boldsymbol{E}^{\mathrm{I}} = (E_0^{\mathrm{I}}, E_1^{\mathrm{I}}, \ldots, E_{n-1}^{\mathrm{I}})$, kjer je za časovni interval $t \in \mathcal{N}$ energija odstopanja enaka E_t^{I}. Cena odstopanja na intervalu t je odvisna od predznaka energije odstopanja in je podana kot

$$p_t^{\mathrm{I}} = \begin{cases} p_t^{\mathrm{I}+}, & E_t^{\mathrm{I}} \geq 0, \\ p_t^{\mathrm{I}-}, & E_t^{\mathrm{I}} < 0, \end{cases}$$

kjer je $p_t^{\mathrm{I}+}$ cena v primeru pozitivne energije odstopanja (prodaja energije) in $p_t^{\mathrm{I}-}$ cena v primeru negativne energije odstopanja (dokup energije). Strošek končnega energijskega odstopanja je enak

$$c_I = \sum_{t \in \mathcal{N}} c_{I,t} = \sum_{t \in \mathcal{N}} E_t^{\mathrm{I}} p_t^{\mathrm{I}}.$$

To je pričakovan strošek operaterja za izničenje po razporejanju nastalega napovedanega energijskega odstopanja. Negativen strošek pomeni dobiček.

3.2.2 Strošek prilagodljivih ponudb

Strošek prilagodljivih ponudb $c_{\mathcal{F}}$ izračunamo kot vsoto stroška privzetega razporeda $c_{\mathcal{F}}^D$ in stroška spremembe prilagodljivih ponudb od privzetega razporeda $\widetilde{c}_{\mathcal{F}}$, to je

$$c_{\mathcal{F}} = c_{\mathcal{F}}^D + \widetilde{c}_{\mathcal{F}}.$$

Strošek privzetega razporeda $c_{\mathcal{F}}^D$ je znan vnaprej. Enak je vsoti stroškov privzetih urnikov izvajanja za posamezno prilagodljivo ponudbo, to je

$$c_{\mathcal{F}}^D = \sum_{f\in\mathcal{F}} c_f^D = \sum_{f\in\mathcal{F}} \sum_{t\in\mathcal{T}_f} c_{f,t}^D.$$

Strošek posameznega privzetega urnika izvajanja c_f^D je enak vsoti stroškov privzetega urnika izvajanja po časovnih intervalih.

Strošek spremembe od privzetega razporeda $\widetilde{c}_{\mathcal{F}}$ izračunamo kot vsoto stroškov sprememb privzetih urnikov izvajanja prilagodljivih ponudb

$$\widetilde{c}_{\mathcal{F}} = \sum_{f\in\mathcal{F}} \widetilde{c}_f = \sum_{f\in\mathcal{F}} \sum_{t\in\mathcal{T}_f} \widetilde{c}_{f,t}.$$

Strošek spremembe urnika izvajanja $\widetilde{c}_f$ posamezne prilagodljive ponudbe je enak vsoti stroškov spremembe po časovnih intervalih.

Strošek spremembe urnika prilagodljive ponudbe f na danem časovnem intervalu t je enak produktu razlike energije in cene razlike

$$\widetilde{c}_{f,t} = \widetilde{E}_{f,t} p_{f,t}.$$

Strošek $c_{f,t}$ posamezne prilagodljive ponudbe f na časovnem intervalu t je enak vsoti stroška privzetega urnika izvajanja $c_{f,t}^D$ in stroška spremembe urnika prilagodljive ponudbe $\widetilde{c}_{f,t}$ V primeru, da prilagodljive ponudbe v končnem urniku ne uporabimo, je njen strošek enak 0.

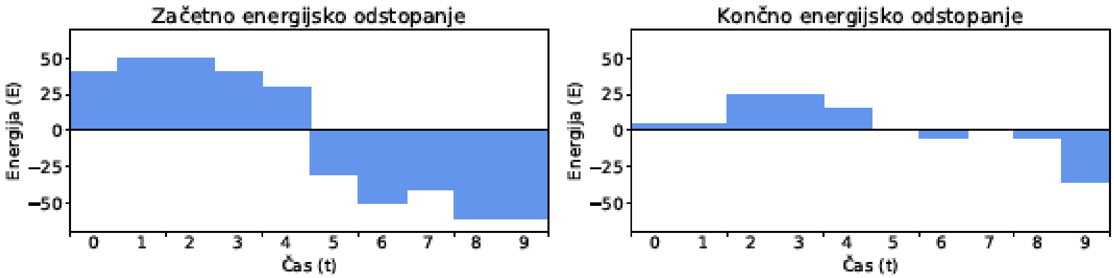
Prilagodljive ponudbe s privzetim urnikom izvajanja so obvezno vključene v končni razpored. Kadar se za prilagodljivo ponudbo f končni urnik izvajanja ujema s privzetim urnikom izvajanja, je njen strošek enak 0.

Opomba 3.2. Strošek privzetega urnika izvajanja izračunamo preko definicije, da je strošek prilagodljive ponudbe na intervalih, kjer ni uporabljena, enak 0.

Opomba 3.3. Poudariti velja, da je lahko strošek spremembe urnika izvajanja neničelen tudi na intervalih, kjer prilagodljiva ponudba ni uporabljena. Glej primer v razdelku 3.2.3 in opombo 3.6.

3.2.3 Ponazoritev stroškov na primeru namišljenega scenarija

V nadaljevanju si oglejmo dva primera, v katerih bomo obravnavali nek namišljen scenarij. V prvem primeru bomo orisali scenarij in nakazali možnost minimiziranja stroškov. V drugem primeru pa bomo prikazali izhodiščno energijsko odstopanje ter pripadajoč razpored brez prilagodljivih ponudb, ki sicer pogosto ni možen kot dejanski razpored. V povezavi s primerom navedemo tudi opombo, v kateri je narejena primerjava med energijo in spremembo energije ter stroški in spremembo stroškov.



Slika 6: Energijsko odstopanje prej (levo) in potem (desno)

Primer 3.4. Na sliki 6 je prikaz energijskega odstopanja pred spremembo razporeda in po njej. Na sliki 7 so pripadajoči razporedi prilagodljivih ponudb, katerih vir so po vrsti generator, hladilnik in baterija. Poudarimo, da je v začetno energijsko odstopanje že všteta energija privzetega razporeda. Torej energija privzetih urnikov izvajanja za generator in hladilnik. Cena prodaje odvečne energije je $p_t^{I+} = -27$, cena nakupa pa $p_t^{I-} = -32$. Za prvo prilagodljivo ponudbo (generator) so cene nakupa za operaterja omrežja v prvi polovici enake 30 v drugi pa 25. To simbolno zapišemo kot

$$\forall t \in \mathcal{T}_{f_1}, \qquad p^+_{f_1,t} = \begin{cases} 30, & t \in \{0, \dots, 4\}, \\ 25, & \text{sicer.} \end{cases}$$

Za drugo prilagodljivo ponudbo (hladilnik) so cene prodaje za operaterja enake 30, kar simbolno zapišemo kot

$$\forall t \in \mathcal{T}_{f_2}, \qquad p^-_{f_2,t} = 30.$$

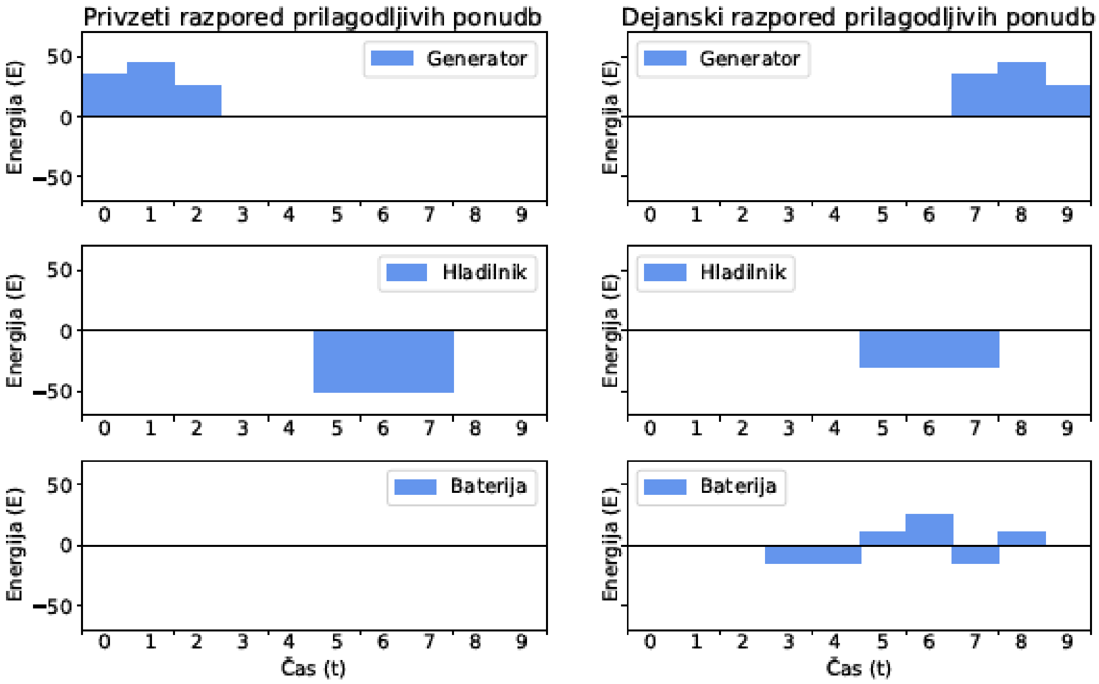
Za tretjo prilagodljivo ponudbo (baterijo) so cene tako prodaje kot nakupa za operaterja enake 30. To simbolno zapišemo kot

$$\forall t \in \mathcal{T}_{f_3}, \qquad p^+_{f_3,t} = p^-_{f_3,t} = 30.$$

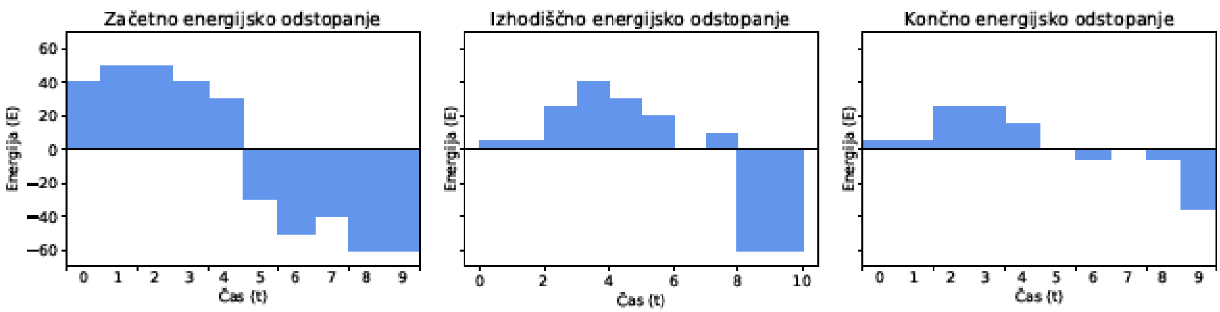
Zaradi privzetega urnika izvajanja prvih dveh prilagodljivih ponudb moramo dodatno navesti še ceni $p^-_{f_1,t} = p^+_{f_1,t} - 2$ ter $p^+_{f_2,t} = p^-_{f_2,t} + 2$.

Opazimo, da se strošek scenarija s spremembo razporeda zmanjša. Namreč, če zaporedno posamič spremenimo urnik prilagodljivih ponudb za generator, hladilnik in baterijo, pri vsaki spremembi posebej dobimo stroškovno ugodnejši razpored. Izboljšava izhaja iz (1) zmanjšanja energijskega odstopanja ter (2) premika prilagodljivih ponudb in energijskega odstopanja na območja ugodnejše cene.

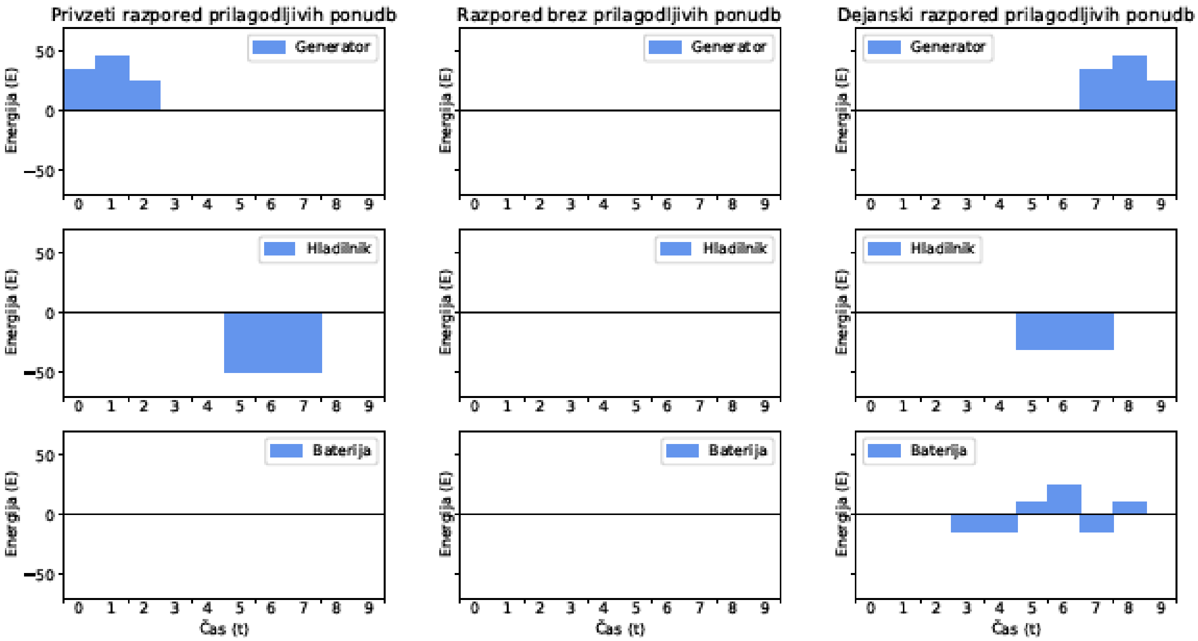
Primer 3.5. Nadaljujmo primer 3.4, kjer poleg začetnega in končnega stanja prikažemo še izhodiščno stanje, to je stanje brez uporabe prilagodljivih ponudb (glej sliko 8). Dodatno za vsakega od stanj prikažemo še pripadajočo razporeditev prilagodljivih ponudb (glej sliko 9), kjer pri izhodiščnem stanju prilagodljivih ponudb ni. Treba je opozoriti, da razporeditev izhodiščnega stanja (stanja brez prilagodljivih ponudb) sicer ni možna, saj je potrebno prilagodljive ponudbe z obveznim razporedom obvezno uporabiti.



Slika 7: Prilagodljive ponudbe prej (levo) in potem (desno)



Slika 8: Energijsko odstopanje (začetno, izhodiščno in končno)



Slika 9: Razpored prilagodljivih ponudb (privzeti, brez prilagodljivih ponudb in dejanski)

Opomba 3.6. Na sliki 9 s prikazom prve prilagodljive ponudbe f_1 si velja ogledati energijo razlike od privzetega razporeda in dejansko energijo

$$\begin{aligned}\widetilde{\boldsymbol{E}}_{\boldsymbol{f_1}} &= (\widetilde{E}_{f_1,0}, \widetilde{E}_{f_1,1}, \ldots, \widetilde{E}_{f_1,9}) = (-35, -45, -25, 0, 0, 0, 35, 45, 25),\\ \boldsymbol{E}_{\boldsymbol{f_1}} &= (E_{f_1,0}, E_{f_1,1}, \ldots, E_{f_1,9}) = (0, 0, 0, 0, 0, 0, 35, 45, 25).\end{aligned}$$

Opazimo, da je sprememba energije glede na privzeti urnik izvajanja neničelna tudi na časovnih intervalih, kjer prilagodljiva ponudba ni uporabljena. Posledično so tudi stroški po intervalih enaki

$$\begin{aligned}\widetilde{\boldsymbol{c}}_{\boldsymbol{f_1}} &= (\widetilde{c}_{f_1,0}, \widetilde{c}_{f_1,1}, \ldots, \widetilde{c}_{f_1,9})\\ &= (-35 \cdot 28, -45 \cdot 28, -25 \cdot 28, 0, 0, 0, 35 \cdot 25, 45 \cdot 25, 25 \cdot 25),\\ \boldsymbol{c}_{\boldsymbol{f_1}} &= (c_{f_1,0}, c_{f_1,1}, \ldots, c_{f_1,9})\\ &= (0, 0, 0, 0, 0, 0, 35 \cdot 25, 45 \cdot 25, 25 \cdot 25).\end{aligned}$$

3.3 Omejitve prilagodljive ponudbe

Pri razporejanju prilagodljivih ponudb moramo biti pozorni na naslednje osnovne omejitve posamezne prilagodljive ponudbe f:

1. Prilagodljive ponudbe z obveznim razporedom morajo biti obvezno uporabljene. V primeru podanega privzetega urnika izvajanja mora biti prilagodljiva ponudba obvezno uporabljena.

2. Upoštevati je potrebno nespreminjajoče število intervalov uporabe prilagodljive ponudbe, ki je enako dolžini prilagodljive ponudbe l_f.

3. Prilagodljiva ponudba se lahko začne le na določenih časovnih intervalih. Množica teh intervalov je $\mathcal{I}_f$ in ne sme biti v protislovju z dolžino prilagodljive ponudbe l_f in dolžino energijskega odstopanja n, torej $\mathcal{T}_f \subseteq \mathcal{N}$. Za upoštevanje tega pogoja je poskrbljeno že v predprocesiranju; ponudbe, ki ga ne upoštevajo, se avtomatsko zavrne.

Dodatno imamo še kompleksnejše omejitve v zvezi z energijo prilagodljive ponudbe. Oglejmo si jih v nadaljevanju.

3.3.1 Omejitev energije na relativnih intervalih

Omejitev energije na izbranem relativnem intervalu prilagodljive ponudbe je neodvisna od postavitve začetka prilagodljive ponudbe. Posamezna omejitev je torej vezana glede na relativni časovni interval znotraj prilagodljive ponudbe. Če se prilagodljiva ponudba f začne v času $i \in \mathcal{I}_f$, je energija $E_{f,i+j}$ za interval $j \in \mathcal{L}_f$ znotraj te postavitve omejena z

$$E_{f,j}^{\min} \leq E_{f,i+j} \leq E_{f,j}^{\max},$$

za ostale časovne intervale $t \notin \{i+j \mid j \in \mathcal{L}_f\}$ pa je energija prilagodljive ponudbe enaka nič, to je

$$E_{f,t} = 0,$$

kar se razume kot da prilagodljiva ponudba na teh intervalih ni uporabljena.

Primer 3.7. Za lažje razumevanje omejitev energije na relativnih intervalih si poglejmo prilagodljivo ponudbo f v hipotetičnem scenariju na 60 časovnih intervalih. Množica možnih začetnih intervalov prilagodljive ponudbe f naj bo $\mathcal{I}_f = \{10, 11, 12, 13, 14, 15\}$, njena dolžina pa tri intervale, torej $\mathcal{L}_f = \{0, 1, 2\}$. Omejitve energije po relativnih intervalih naj bodo

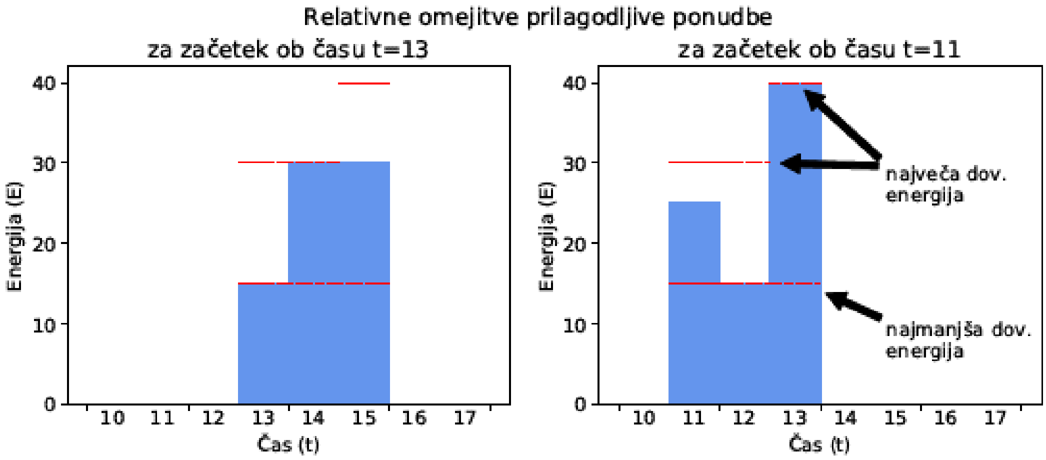
$$\begin{aligned}
&E_{f,0}^{\min} = E_{f,1}^{\min} = E_{f,2}^{\min} = 15, \\
&E_{f,0}^{\max} = E_{f,1}^{\max} = 30, \\
&E_{f,2}^{\max} = 40.
\end{aligned}$$

Ostalih podatkov zaenkrat ne navajajmo, ker za ta primer niso relavantni. Oglejmo si sliko 10, kjer so za to prilagodljivo ponudbo prikazane omejitve energije po relativnih časovnih intervalih. Najmanjše in največje dovoljene vrednosti za energijo posameznih relativnih časovnih intervalov so označene z rdečo črtkano črto in dodatno na desnem grafu s puščicami. Na levem grafu se prilagodljiva ponudba začne ob času $t = 13$, zato so njene energije za intervale, kjer prilagodljiva ponudba ni bila uporabljena, sledeče

$$E_{f,10} = E_{f,11} = E_{f,12} = E_{f,16} = E_{f,17} = 0.$$

Za intervale, kjer je bila uporabljena, pa veljajo omejitve

$$\begin{aligned}
E_{f,0}^{\min} \leq E_{f,13} \leq E_{f,0}^{\max}, \\
E_{f,1}^{\min} \leq E_{f,14} \leq E_{f,1}^{\max}, \\
E_{f,2}^{\min} \leq E_{f,15} \leq E_{f,2}^{\max}.
\end{aligned}$$



Slika 10: Omejitve energije po relativnih intervalih so vezane na začetek prilagodljive ponudbe

Na desnem grafu, kjer se prilagodljiva ponudba začne ob času $t = 11$, spet veljajo podobne omejitve. Za intervale, kjer je prilagodljiva ponudba uporabljena, veljajo naslednje neenačbe

$$
\begin{aligned}
15 &\leq E_{f,11} \leq 30, \\
15 &\leq E_{f,12} \leq 30, \\
15 &\leq E_{f,13} \leq 40,
\end{aligned}
$$

na ostalih intervalih pa je energija prilagodljive ponudbe enaka nič. Na desnem grafu prikazana prilagodljiva ponudba ima na časovnem intervalu $t = 12$ najmanjšo dovoljeno energijo, na časovnem intervalu $t = 13$ pa doseže največjo dovoljeno energijo za ta časovni interval.

3.3.2 Omejitev skupne energije

Včasih imamo za prilagodljivo ponudbo podano tudi omejitev skupne energije. To pomeni, da mora biti vsota energije po vseh intervalih prilagodljive ponudbe znotraj nekih mej. Omejitev velja le v primeru, ko je prilagodljiva ponudba uporabljena.

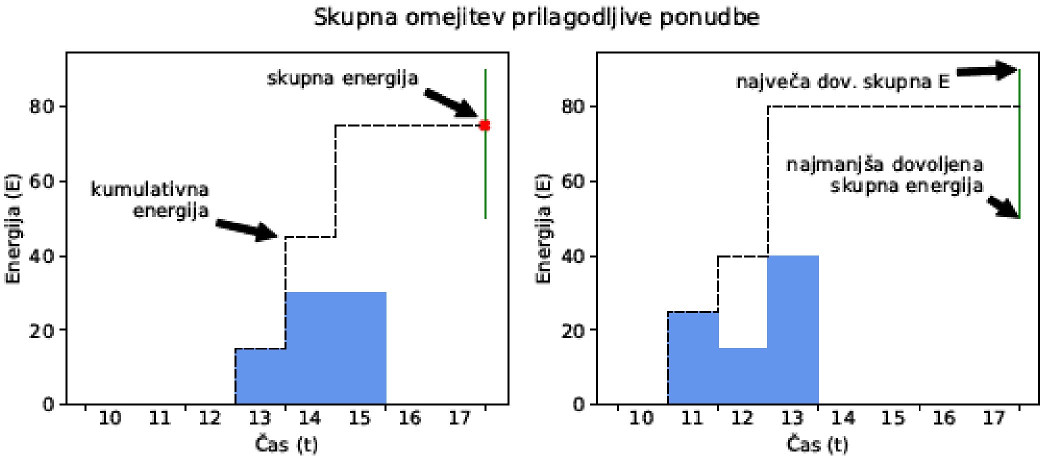
Če se prilagodljiva ponudba f začne v času $i \in \mathcal{I}_f$, lahko skupno energijo prilagodljive ponudbe zapišemo kot

$$E_f = \sum_{j \in \mathcal{L}_f} E_{f,i+j}.$$

Če pri tem upoštevamo, da je energija prilagodljive ponudbe za ostale časovne intervale enaka 0, pridemo do enostavnejšega zapisa

$$E_f = \sum_{t \in \mathcal{T}_f} E_{f,t},$$

kjer t teče po vseh možnih časovnih intervalih prilagodljive ponudbe.



Slika 11: Omejitev skupne energije

Če ima prilagodljiva ponudba f omejitev na skupno energijo, to zapišemo kot

$$E_f^{\mathrm{Tmin}} \leq E_f \leq E_f^{\mathrm{Tmax}},$$

kjer je E_f^{Tmin} najmanjša dovoljena skupna energija in E_f^{Tmax} največja dovoljena skupna energija.

Omejitev skupne energije prilagodljive ponudbe si oglejmo na primeru 3.8.

Primer 3.8. Nadaljujmo primer 3.7, kjer imamo že definiran scenarij z eno prilagodljivo ponudbo. Definiciji te prilagodljive ponudbe dodajmo še omejitev za skupno energijo. Naj bo skupna energija E_f večja od $E_f^{\mathrm{Tmin}} = 50$ in manjša od $E_f^{\mathrm{Tmax}} = 90$, kar zapišemo z omejitvijo

$$50 \leq \sum_{t \in \mathcal{T}_f} E_{f,t} \leq 90.$$

Sedaj si oglejmo sliko 11, kjer je prikazana omejitev skupne energije. S črtkano črto je označena trenutna kumulativna energija, le ta je na zadnjem intervalu enaka skupni energiji, kar je na levem grafu označeno z rdečim križcem. Na obeh grafih je prikazan ujemajoč razpored prilagodljivih ponudb kot pri sliki 10 primera 3.7. Na levem grafu sta s puščicami označeni kumulativna energija in skupna energija. Na desnem grafu pa spodnja in zgornja meja za skupno energijo. Zelena daljica v obeh primerih predstavlja ustrezno območje za skupno energijo.

3.3.3 Omejitev kumulativne energije

Za prilagodljive ponudbe, katerih vir so baterije, moramo upoštevati še dodatno omejitev, ki preprečuje, da bi baterijo preveč izpraznili ali napolnili. To je omejitev kumulativne energije, ki zahteva, da je v vsakem časovnem intervalu vsota do sedaj porabljene energije znotraj določenih mej.

Če je vir prilagodljive ponudbe f baterija, ki se začne v času $i \in \mathcal{I}_f$, moramo za vsak čas $j_0 \in \mathcal{L}_f$ upoštevati naslednji neenačbi

$$E_f^{\mathrm{Cmin}} \leq E_f^{\mathrm{Cst}} + \sum_{j \leq j_0} E_{f,i+j} \leq E_f^{\mathrm{Cmax}},$$

kjer je E_f^{Cmin} najmanjša dovoljena kumulativna energija, E_f^{Cmax} največja dovoljena kumulativna energija in E_f^{Cst} začetna energija baterije. Vsota energij

$$\sum_{j \le j_0} E_{f,i+j}$$

je v času scenarija prejeta energija baterije do intervala j_0. Navadno je E_f^{Cmin} enaka nič in E_f^{Cmax} enaka največji kapaciteti baterije.

Če upoštevamo, da je za čase t zunaj postavitve prilagodljive ponudbe f energija enaka nič, to je $E_{f,t} = 0$, lahko neenačbe omejitve kumulative energije poenostavimo na

$$\forall t_0 \in \mathcal{T}_f, \qquad E_f^{\mathrm{Cmin}} \le E_f^{\mathrm{Cst}} + \sum_{t \le t_0} E_{f,t} \le E_f^{\mathrm{Cmax}},$$

kjer t in t_0 tečeta po vseh možnih časovnih intervalih baterije. Tak zapis ima veliko prednosti predvsem v izgradnji modela (glej razdelek 6.2.5).

Sedaj si oglejmo primer 3.9 s prilagodljivo ponudbo, katere vir je baterija. V takem primeru so lahko energije prilagodljive ponudbe tako pozitivne kot negativne, zato omejitev skupne energije ne zadošča in za ponazoritev realnega problema potrebujemo uporabo kumulativne omejitve.

Primer 3.9. Vzemimo hipotetičen scenarij, ki se odvija na 90 časovnih intervalih. Na tem scenariju definirajmo prilagodljivo ponudbo f, katere vir je baterija. Prilagodljiva ponudba naj ne bo časovno prilagodljiva. Začne naj se s časovnim intervalom $t = 11$, njena dolžina pa naj bo $l_f = 7$. Začetna energija baterije naj bo $E_f^{\mathrm{Cst}} = 0$.

Omejitev energije na relativnih intervalih naj bo konstantna

$$\forall t \in \mathcal{T}_f, \qquad -20 \le E_{f,t} \le 20.$$

Dodatno naj ima prilagodljiva ponudba f še omejitev skupne energije in sicer

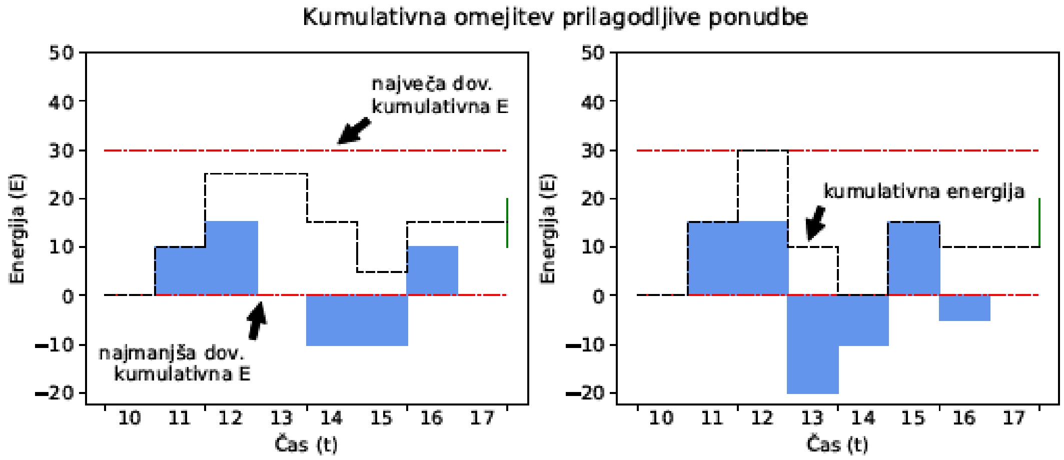
$$10 \le \sum_{t \in \mathcal{T}_f} E_{f,t} \le 20,$$

ter omejitev kumulativne energije

$$\forall t_0 \in \mathcal{T}_f, \qquad 0 \le \sum_{t \le t_0} E_{f,t} \le 30.$$

Praktični razlogi za omejitve bi bili lahko naslednji:

1. Baterija se v 15 minutah lahko napolni ali izprazni za največ 20 kW.
2. Na koncu jo želimo imeti napolnjeno nekje med 10 kW in 20 kW.
3. Ko je baterija prazna se je ne da več prazniti, polna pa je pri 30 kW.



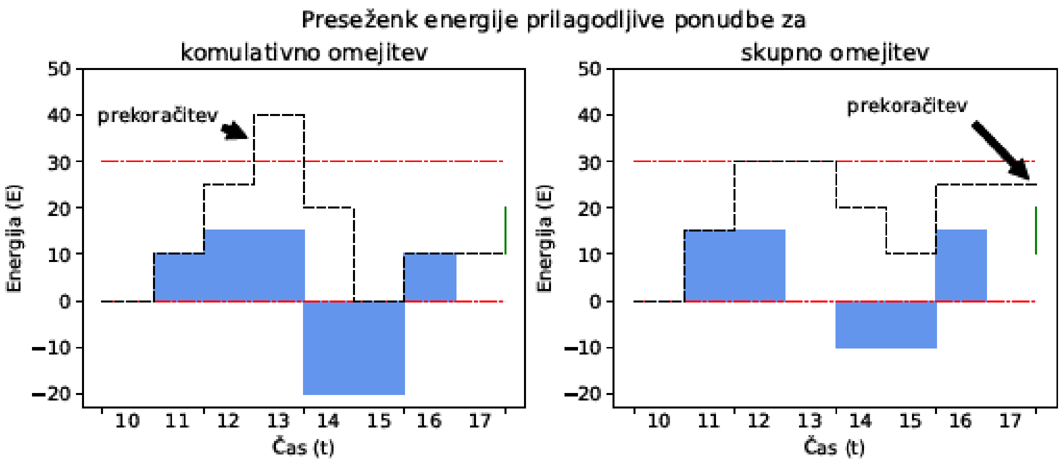
Slika 12: Omejitev kumulativne energije

Sedaj si oglejmo sliko 12, kjer sta prikazani dve možni postavitvi te prilagodljive ponudbe, označeni pa sta tudi omejitvi skupne in kumulativne energije. Na levem grafu sta s puščicami dodatno označeni najmanjša dovoljena in največja dovoljena kumulativna energija. Opazimo tudi, da je na levem grafu energija prilagodljive ponudbe na tretjem ($t = 13$) in sedmem ($t = 17$) relativnem intervalu enaka nič, čeprav je prilagodljiva ponudba na teh intervalih uporabljena. Na desnem grafu je s puščico označena kumulativna energija pri časovnem intervalu $t = 13$, kjer je enaka 10. Dodatno lahko opazimo, da pri desnem grafu v času $t = 12$ baterija doseže največjo dovoljeno kumulativno energijo in se zato v naslednjem intervalu ne more zopet polniti. Podobno je v času $t = 14$ baterija prazna, zato se ne more več prazniti. Zanimiva pa je še opazka, da se skupna energija prilagodljive ponudbe na koncu ujema s spodnjo mejo za skupno energijo, ki je enaka 10. Stanja, ko nas nekateri pogoji zelo omejujejo so v praksi lahko pogosta, saj se na ta način kar najbolje izkoristi vire energije in njihovo prilagodljivost.

3.3.4 Neupoštevanje omejitev

Sedaj si poglejmo še primer dveh razporeditev prilagodljivih ponudb, ki ne upoštevata omejitev. Naš algoritem takih izhodnih urnikov prilagodljivih ponudb ne vrača, saj upošteva omejitve, koristijo pa lahko kot prikaz, kaj gre lahko narobe, če ne bi upoštevali omejitev, oziroma bi delali z napačnimi podatki.

Primer 3.10. Prevzemimo enako prilagodljivo ponudbo, kot jo imamo v primeru 3.9. Na sliki 13 si oglejmo primera hipotetične razporeditve, pri katerih pride do prekoračenja omejitev prilagodljive ponudbe. Na levi sliki je presežena kumulativna omejitev baterije, kar bi v praksi lahko povzročilo okvaro in materialno škodo na bateriji. Na desni sliki je presežena skupna omejitev, kar bi v praksi lahko pomenilo neoptimalno delovanje sistema, direktna posledica bi bila tudi, da se baterija v prihodnosti lahko napolni za manj kot bi želeli.



Slika 13: Presežene omejitve

4 Mešano celoštevilsko linearno programiranje

Mešano celoštevilsko linearno programiranje je postopek, ki se uporablja za reševanje nekaterih optimizacijskih nalog. Pri tem je pomembno, da je optimizacijski problem v obliki, ki dopušča zapis namenske funkcije in omejitev v linearni obliki. Pogosto iz samega problema ni takoj razvidno, če se ga da modelirati na ta način in izgradnja modela zahteva globlji premislek.

Oglejmo si primer zapisa optimizacijskega problema v obliki mešanega celoštevilskega linearnega programa. Samo obliko zapisa bomo definirali kasneje tekom tega poglavja.

Primer 4.1. Hotel podira luksuzni bazen. Uprava hotela ponudi svojim zaposlenim brezplačen odkup umetnega okrasnega peska, oljnih bakel in okrasnih palm. Receptor Janez si že dlje časa želi podoben bazen na svojem vikendu. Pozanima se o cenah nakupa posameznih surovin in najame prevoz. Oljne bakle stanejo 30 evrov, so težke 2 kilograma in zasedejo 30 kubičnih decimetrov volumna pri prevozniku. Okrasne palme stanejo 70 evrov, so težke 3 kilograme in imajo 80 kubičnih decimetrov prostornine. Umetni pesek stane 10 evrov na liter in ima gostoto $1.7\,\mathrm{kg/dm^3}$. Omejitve prevoznika so, da ima lahko tovor maso največ 55 kilogramov in volumen največ 550 kubičnih decimetrov. Kaj naj Janez vzame, da bo prihranil čim več denarja pri gradnji svojega bazena?

Problem lahko zapišemo v obliki mešanega celoštevilskega linearnega programa

$$
\begin{aligned}
\text{maksimiziraj:}\quad & z = 30x_1 + 70x_2 + 10x_3,\\
\text{pri pogojih:}\quad & m = 2x_1 + 3x_2 + 1.7x_3 \leq 55\\
& v = 30x_1 + 80x_2 + x_3 \leq 550\\
& x_1, x_2, x_3 \geq 0\\
& x_1, x_2 \in \mathbb{Z}.
\end{aligned}
$$

Problem, zapisan v obliki mešanega celoštevilskega programiranja, ima tri spremenljivke. Spremenljivka x_1 je število oljnih bakel, spremenljivka x_2 je število okrasnih palm, spremenljivka x_3 pa je količina peska podana v litrih. V programu imamo tudi dve omejitvi, oziroma pogoja, in sicer enega za maso in enega za volumen tovora. Podan imamo tudi optimizacijski kriterij, ki je maksimizacija funkcionala z, ki predstavlja vrednost izbranih surovin.

Rešitev mešanega celoštevilskega linearnega programa je $x_1 = 7$, $x_2 = 4$ in $x_3 = 17.059$. Torej je optimalna Janezova izbira 7 bakel, 4 palme in 17.059 litrov okrasnega peska. Vrednost blaga je v tem primeru največja in enaka 660.59 evrov.

V nadaljevanju si bomo najprej ogledali linearno programiranje, katerega metode so ključne za dobro razumevanje mešanega celoštevilskega linearnega programiranja. Linearni program je tesno povezan z mešanim celoštevilskim linearnim programom, razlika je le, da so vse spremenljivke linearnih programov realne in niso omejene s pogojem celoštevilskosti. Nato si bomo ogledali standardno obliko mešanega celoštevilskega linearnega programa. Sledi še teoretična umestitev in opredelitev lastnosti mešanega celoštevilskega linearnega programa. Na koncu poglavja pa navedemo še algoritme, ki omogočajo učinkovito reševanje problemov, zapisanih v obliki mešanega celoštevilskega linearnega programa.

4.1 Linearni program

V splošnem je *linearni program* (angl. linear program) ali skrajšano LP naslednje oblike

$$\begin{aligned}
\text{optimiziraj:} \quad f(\boldsymbol{x}) = \quad & c_1x_1 + c_2x_2 + \cdots + c_sx_s, \\
\text{pri pogojih:} \quad & a_{1,1}x_1 + a_{1,2}x_2 + \cdots + a_{1,s}x_s \gtreqless b_1 \\
& a_{2,1}x_1 + a_{2,2}x_2 + \cdots + a_{2,s}x_s \gtreqless b_2 \\
& \qquad \cdots \\
& a_{m,1}x_1 + a_{m,2}x_2 + \cdots + a_{m,s}x_s \gtreqless b_m,
\end{aligned}$$

kjer so z $x_i, i = 1, \ldots, s$, označene realne spremenljivke, ki so predmet linearnega programiranja, in z $a_{i,j}, b_j, c_i, i = 1, \ldots, s, j = 1, \ldots, m$, realne konstante. Simboli $\gtreqless$ pa za vsak pogoj posebej predstavljajo enega izmed znakov $\geq$, $=$ ali $\leq$, zato so pogoji enačbe ali neenačbe. Funkcija f je namenska funkcija, katere optimum želimo poiskati. Včasih namesto f uporabimo fleksibilnejši zapis s funkcionalom z, pri katerem je odvisnost od spremenljivk implicitna.

Linearni program je matematična formulacija, pri kateri imamo podano namensko funkcijo $f(\boldsymbol{x})$ in pogoje oziroma omejitve, ki jim morajo spremenljivke $x_1, \ldots, x_s$ zadostiti. Znotraj teh omejitev iščemo take vrednosti spremenljivk, pri katerih bo imela namenska funkcija minimalno oziroma maksimalno vrednost. Beseda linearni ponazarja, da v namenski funkciji kot tudi v omejitvah spremenljivke $x_1, \ldots, x_s$ nastopajo linearno. Namenska funkcija je linearna kombinacija spremenljivk $x_1, \ldots, x_s$. Omejitve so podane kot linearne enačbe ali neenačbe istih spremenljivk. Omejitve določajo podprostor prostora $\mathbb{R}^s$, na katerem iščemo ekstremno vrednost namenske funkcije. Spremenljivke $x_1, \ldots, x_s$ so realne, v nasprotnem primeru gre za eno od linearnem programu sorodnih formulacij, ki jih bomo srečali v razdelku 4.3.

Definicija 4.2. Namenska funkcija linearnih programov $f(\boldsymbol{x}) = z$ je lahko oblike

$$z = k + c_1x_1 + c_2x_2 + \cdots + c_sx_s,$$

kjer od $\boldsymbol{x}$ neodvisen člen $k \in \mathbb{R}$ imenujemo ***številska vrednost*** funkcionala z. Številska vrednost je vrednost funkcionala z v točki $\boldsymbol{x} = \boldsymbol{0}$.

Definicija 4.2 je uporabna pri pretvorbah linearnih programov in še posebej za obrazložitev metode simpleksov. Oglejmo si konkreten primer zapisa linearnega programa za namišljen optimizacijski problem.

Primer 4.3. Recimo, da imamo problem pri katerem želimo minimizirati izraz $3x_1 - 2.2x_2 + 7x_3$ pri omejitvah, da so $3x_1 + 4x_3 \geq 14$, $x_2 = \frac{2x_1 - 8}{5}$ in x_2 je manjši od 5.3. To bi v obliki linearnega programa zapisali kot

$$\begin{aligned}
\text{minimiziraj:} \quad f(\boldsymbol{x}) = \quad & 3x_1 - 2.2x_2 + 7x_3, \\
\text{pri pogojih:} \quad & 3x_1 + 4x_3 \geq 14 \\
& 2x_1 - 5x_2 = 8 \\
& x_2 \leq 5.3.
\end{aligned}$$

Z uporabo notacije linearne algebre lahko zapis linearnega programa napišemo tudi v bolj kompaktni abstraktni obliki

$$\begin{aligned} \text{optimiziraj:} \quad & \boldsymbol{c}^T \boldsymbol{x}, \\ \text{pri pogojih:} \quad & \boldsymbol{A}\boldsymbol{x} \gtreqless \boldsymbol{b}, \end{aligned}$$

kjer je $\boldsymbol{c} \in \mathbb{R}^s$, $\boldsymbol{A} \in \mathbb{R}^{m \times s}$ in $\boldsymbol{b} \in \mathbb{R}^m$ ter spremenljivke $\boldsymbol{x} \in \mathbb{R}^s$. V taki obliki je potrebno simbol $\gtreqless$ definirati za vsako komponento posebej. Ponavadi zato uporabljamo poenoten zapis za vse komponente, kar je tudi smiselno za konkretne izračune. Kot primer uporabe poenotenega zapisa glej recimo standardno obliko v razdelku 4.1.2, dualni program v razdelku 4.1.3 in zapis z enakostjo pri uporabi simpleksne metode v razdelku 4.4.1. Za doseganje poenotenega zapisa je potrebno uporabiti določene pretvorbe, ki model preoblikujejo na način, da se z njegovo rešitvijo še vedno reši prvotni problem. V nadaljevanju si poglejmo nekaj standardnih prijemov za pretvorbo linearnih programov, ki nam med drugim služijo tudi za pretvorbo v standardno obliko.

4.1.1 Pretvorbe linearnih programov

Pri pretvorbi linearnih programov v različne oblike si lahko pomagamo z naslednjimi uporabnimi prijemi:

1. **Vpeljava nove spremenljivke.** Namesto spremenljivke x lahko v modelu naredimo linearno substitucijo $y = k_1 x + k_2$, kjer sta $k_1, k_2 \in \mathbb{R}$ in $k_1 \neq 0$.

2. **Uvedba ekvivalentne namenske funkcije.** Uvedemo lahko novo namensko funkcijo $g(\boldsymbol{x}) = k_1 f(\boldsymbol{x}) + k_2$, kjer sta $k_1, k_2 \in \mathbb{R}$ in $k_1 > 0$. Ta je ekvivalentna v smislu, da doseže optimum ravno takrat kot prvotna.

3. **Pretvorba med iskanjem minimuma in iskanjem maksimuma.** Če je naša optimizacijska naloga maksimizirati $f(\boldsymbol{x})$ lahko dosežemo isto, če minimiziramo $-f(\boldsymbol{x})$. Podobno lahko pretvorimo tudi iskanje minimuma v iskanje maksimuma.

4. **Pretvorba iz neenakosti v enakost.** Če imamo pogoj z neenakostjo

   $$a_1 x_1 + \cdots + a_s x_s \leq b,$$

   ga lahko z uvedbo dodatne spremenljivke $x_{s+1} \geq 0$ zapišemo v obliki

   $$a_1 x_1 + \cdots + a_s x_s + x_{s+1} = b.$$

   Podobno lahko storimo tudi za neenakost z $\geq$, le da spremenljivko x_{s+1} odštejemo namesto prištejemo.

5. **Pretvorba enakosti v neenakost.** Če imamo enačbo

   $$a_1 x_1 + \cdots + a_s x_s = b,$$

   jo lahko pretvorimo v dve neenačbi

   $$b \leq a_1 x_1 + \cdots + a_s x_s \leq b.$$

6. **Pretvorba enakosti v ostale pogoje.** Če imamo enačbo

$$a_1x_1 + \cdots + a_sx_s = b,$$

lahko eno izmed spremenljivk izrazimo z ostalimi. Izrazimo lahko recimo i-to spremenljivko kot

$$x_i = \frac{b}{a_i} - \frac{a_1}{a_i}x_1 - \cdots - \frac{a_{i-1}}{a_i}x_{i-1} - \frac{a_{i+1}}{a_i}x_{i+1} \cdots - \frac{a_s}{a_i}x_s$$

in naredimo ustrezno substitucijo v pogojih in namenski funkciji, kar nam za ena zmanjša število spremenljivk in pogojev.

7. **Menjava neenakosti.** Če imamo podano neenačbo

$$a_1x_1 + \cdots + a_sx_s \geq b,$$

z uporabo znaka za večje ali enako, jo lahko pretvorimo v neenačbo

$$-a_1x_1 - \cdots - a_sx_s \leq -b,$$

v kateri nastopa manjše ali enako. Tako dobimo nove koeficiente $a_i' = -a_i$ in $b' = -b$. Podobno storimo tudi za pretvorbo v obratno smer.

8. **Pretvorba realne spremenljivke v spremenljivki večji od nič.** Naj bo $x \in \mathbb{R}$, potem ga lahko zapišemo kot $x = x^+ - x^-$, kjer sta $x^+, x^- \geq 0$.

Primer 4.4. Oglejmo si primer uporabe zapisanih prijemov, na njih se sklicujmo z označbo v oklepaju. Recimo, da želimo linearni program

$$\begin{array}{lrcl}
\text{maksimiziraj:} & 4x_1 - 3x_2 - 8x_3 + 5x_4 + 20, & & \\
\text{pri pogojih:} & 2x_1 - 6x_2 \qquad\quad + 10x_4 & \leq & 300 \\
 & 3x_1 \qquad\qquad\qquad + 2x_4 & \leq & 80 \\
 & 9x_2 + 4x_3 - x_4 & \geq & 250 \\
 & 4x_1 + x_2 - 2x_3 + 12x_4 & \geq & 120
\end{array}$$

zapisati v obliki, kjer je povsod uporabljena neenakost manjše ali enako ter iščemo minimum namenske funkcije v kateri ni konstantnega člena in so vsi koeficienti spremenljivk enaki ena. To naredimo kot sledi v nadaljevanju.

Pretvorimo maksimiziranje v minimiziranje (prijem 3) in zamenjamo neenakost (prijem 7), kjer je potrebno

$$\begin{array}{lrcl}
\text{minimiziraj:} & -4x_1 + 3x_2 + 8x_3 - 5x_4 - 20, & & \\
\text{pri pogojih:} & 2x_1 - 6x_2 \qquad\quad + 10x_4 & \leq & 300 \\
 & 3x_1 \qquad\qquad\qquad + 2x_4 & \leq & 80 \\
 & -9x_2 - 4x_3 + x_4 & \leq & -250 \\
 & -4x_1 - x_2 + 2x_3 - 12x_4 & \leq & -120.
\end{array}$$

Vpeljemo nove spremenljivke (prijem 1)

$$y_1 = -4x_1, \quad y_2 = 3x_2, \quad y_3 = 8x_3, \quad y_4 = -5x_4$$

ter se znebimo konstante -20 v namenski funkciji (prijem 2)

$$\begin{array}{lrcrcrcrcr}
\text{minimiziraj:} & y_1 & + & y_2 & + & y_3 & + & y_4, & & \\
\text{pri pogojih:} & -\frac{1}{2}y_1 & - & 2y_2 & & & - & 2y_4 & \leq & 300 \\
 & -\frac{3}{4}y_1 & & & & & - & \frac{2}{5}y_4 & \leq & 80 \\
 & & - & 3y_2 & - & \frac{1}{2}y_3 & - & \frac{1}{5}y_4 & \leq & -250 \\
 & y_1 & - & \frac{1}{3}y_2 & + & \frac{1}{4}y_3 & + & \frac{12}{5}y_4 & \leq & -120.
\end{array}$$

To so popolnoma nesporni matematični prijemi, ki ne posegajo v definicijo problema. Prvotno rešitev lahko trivialno izračunamo iz tako preoblikovanega modela in pripadajoče rešitve. Z dodatnim znanjem o problemu lahko tudi dopolnimo modele, kar nam v nadaljevanju omogoča lažje reševanje. Tako je recimo smiselno, da vsako spremenljivko x_i, za katero imamo dodatne informacije, omejimo. Na primer, če vemo, da je $k_i^{\min}$ najmanjša možna vrednost za spremenljivko x_i, jo omejimo navzdol z $k_i^{\min} \leq x_i$. Ekvivalentno jo lahko pod podobnimi predpostavkami omejimo tudi navzgor z $x_i \leq k_i^{\max}$. Na tak način izdelamo model, ki ga z algoritmi kasneje lažje rešujemo.

4.1.2 Standardna oblika

Za lažjo obravnavo linearne programe pogosto zapišemo v standardni obliki

$$\begin{array}{rl}
\text{minimiziraj:} & \boldsymbol{c}^T\boldsymbol{x}, \\
\text{pri pogojih:} & \boldsymbol{A}\boldsymbol{x} \leq \boldsymbol{b} \\
 & \boldsymbol{x} \geq \boldsymbol{0},
\end{array}$$

kjer je $\boldsymbol{c} \in \mathbb{R}^s$, $\boldsymbol{A} \in \mathbb{R}^{m\times s}$ in $\boldsymbol{b} \in \mathbb{R}^m$. Poglavitna lastnost standardne oblike je, da so spremenljivke x nenegativne in je v ostalih pogojih uporabljena poenotena vrsta neenakosti ali enakosti. Obstaja namreč več standardnih oblik, različni avtorji navajajo različne (glej [3, 9, 11, 15, 19]). Bistvo zapisa linearnih programov s standardno obliko je lažji opis teorije in algoritmov. Kot bomo videli v razdelku 4.4.1 pri opisu metode simpleksov, je včasih smiselno zapis modela razširiti z dodatnimi nenegativnimi spremenljivkami. Tako dobimo model, ki poleg neenačb za pogoje nenegativnosti spremenljivk vsebuje le enačbe.

Oglejmo si primer pretvorbe nekega linearnega programa v standardno obliko. Pri tem si bomo pomagali z razdelkom 4.1.1, kjer smo opisali možne prijeme za pretvorbo.

Primer 4.5. V standardno obliko pretvorimo linearni program iz primera 4.3

$$\begin{array}{rrl}
\text{minimiziraj:} & z = & 3x_1 - 2.2x_2 + 7x_3, \\
\text{pri pogojih:} & & 3x_1 + 4x_3 \geq 14 \\
 & & 2x_1 - 5x_2 = 8 \\
 & & x_2 \leq 5.3.
\end{array}$$

Namenska funkcija se minimizira, kar je ustrezno. Popravimo še pogoje. Najprej odstranimo pogoj enakosti, iz pogoja $2x_1 - 5x_2 = 8$ izrazimo x_1, in naredimo substitucijo spremenljivke $x_1 = 2.5x_2 + 4$ v ostalih pogojih (prijem 6). Tako dobimo

$$\begin{aligned} \text{minimiziraj:} \quad z = \quad & 12 + 5.3x_2 + 7x_3, \\ \text{pri pogojih:} \quad & 7.5x_2 + 4x_3 \geq 2 \\ & x_2 \leq 5.3. \end{aligned}$$

Ker želimo, da so vse spremenljivke večje ali enake nič, uvedemo $x_4 = 5.3 - x_2$ (prijem 1) in dobimo

$$\begin{aligned} \text{minimiziraj:} \quad z = \quad & 40.09 + 7x_3 - 5.3x_4, \\ \text{pri pogojih:} \quad & 4x_3 - 7.5x_4 \geq -37.75 \\ & x_4 \geq 0. \end{aligned}$$

Uvedemo novo namensko funkcijo brez konstantnega člena $y = z - 40.09$ (prijem 2). Preuredimo pogoj $4x_3 - 7.5x_4 \geq -37.75$ v $-4x_3 + 7.5x_4 \leq 37.75$ (prijem 7) in se znebimo spremenljivke x_3, ki je na vse strani neomejena (prijem 8). Zapišemo jo kot $x_3 = x_5 - x_6$, kjer sta $x_5 \geq 0$ in $x_6 \geq 0$. Tako dobimo linearni program zapisan v standardni obliki

$$\begin{aligned} \text{minimiziraj:} \quad y = \quad & -5.3x_4 + 7x_5 - 7x_6, \\ \text{pri pogojih:} \quad & 7.5x_4 - 4x_5 + 4x_6 \leq 37.75 \\ & x_4, x_5, x_6 \geq 0. \end{aligned}$$

Ko dobimo rešitev tako zapisanega linearnega programa, iz vrednosti novih spremenljivk izračunamo prvotne spremenljivke linearnega programa. Vrednost funkcionala z izračunamo preko funkcionala y po enačbi $z = y + 40.09$. Funkcional z je tako od x-ov odvisna spremenljivka, s številsko vrednostjo 40.09.

4.1.3 Dualnost linearnih programov

Vsakemu linearnemu programu pripada ustrezni dualni linearni progam. Za linearne programe zapisane v standardni obliki (glej 4.1.2) je enak

$$\begin{aligned} \text{maksimiziraj:} \quad & \boldsymbol{b}^T \boldsymbol{y}, \\ \text{pri pogojih:} \quad & \boldsymbol{A}^T \boldsymbol{y} \leq \boldsymbol{c} \\ & \boldsymbol{y} \leq \boldsymbol{0}, \end{aligned}$$

kjer so $\boldsymbol{b}$, $\boldsymbol{A}$ in $\boldsymbol{c}$ iz standardne oblike, vektor spremenljivk $\boldsymbol{y}$ pa je rešitev, ki jo iščemo. Poznavanje pojma dualni linearni program nam pomaga pri boljšem razumevanju vseh aspektov linearnega programiranja, predvsem ponuja mnogo uporabnih idej za izdelavo algoritmov, ki rešujejo linearni program. Znan je tudi izrek o močni dualnosti, ki pravi, da sta vrednosti optimalnih rešitev linearnega programa in njegovega duala enaki

$$\boldsymbol{c}^T \boldsymbol{x} = \boldsymbol{b}^T \boldsymbol{y}.$$

Za dokaz izreka glej [19].

Oglejmo si še primer pretvorbe linearnega programa v dualni program.

Primer 4.6. Določimo dualni program za linearni program, kjer so pogoji podani z večje ali enako

$$\begin{aligned} \text{minimiziraj:} \quad & \boldsymbol{c}^T\boldsymbol{x}, \\ \text{pri pogojih:} \quad & \boldsymbol{A}\boldsymbol{x} \geq \boldsymbol{b} \\ & \boldsymbol{x} \geq \boldsymbol{0}. \end{aligned}$$

Najprej ga pretvorimo v standardno obliko

$$\begin{aligned} \text{minimiziraj:} \quad & \boldsymbol{c}^T\boldsymbol{x}, \\ \text{pri pogojih:} \quad & -\boldsymbol{A}\boldsymbol{x} \leq -\boldsymbol{b} \\ & \boldsymbol{x} \geq \boldsymbol{0}, \end{aligned}$$

in zapišemo njen dualni program

$$\begin{aligned} \text{maksimiziraj:} \quad & -\boldsymbol{b}^T\boldsymbol{y}', \\ \text{pri pogojih:} \quad & -\boldsymbol{A}^T\boldsymbol{y}' \leq \boldsymbol{c} \\ & \boldsymbol{y}' \leq \boldsymbol{0}. \end{aligned}$$

Uvedemo še nov vektor spremenljivk $\boldsymbol{y} = -\boldsymbol{y}'$, da dobimo preglednejši zapis

$$\begin{aligned} \text{maksimiziraj:} \quad & \boldsymbol{b}^T\boldsymbol{y}, \\ \text{pri pogojih:} \quad & \boldsymbol{A}^T\boldsymbol{y} \leq \boldsymbol{c} \\ & \boldsymbol{y} \geq \boldsymbol{0}. \end{aligned}$$

Za določitev dualnega programa sicer ne potrebujemo predhodne pretvorbe v standardno obliko. Pretvorbo lahko naredimo na pamet ali s pomočjo tabel (glej [3], tabela 6.1 na strani 264). Za več informacij o dualnem programu glej [9] in [19] ter primer 4.19 v razdelku z algoritmi 4.4.1.

4.2 Mešani celoštevilski linearni program

Mešani celoštevilski linearni program (angl. mixed integer linear program) ali skrajšano MILP je podoben linearnemu programu, le da za nekatere spremenljivke zahteva, da so celoštevilske namesto realne. Podobno kot pri linearnem programiranju lahko tudi za mešano celoštevilsko linearno programiranje zapišemo mešani celoštevilski linearni program v standardni obliki:

$$\begin{aligned} \text{minimiziraj:} \quad & \boldsymbol{c}^T\boldsymbol{x}, \\ \text{pri pogojih:} \quad & \boldsymbol{A}\boldsymbol{x} \leq \boldsymbol{b} \\ & \boldsymbol{x} \geq \boldsymbol{0} \\ & \forall i \leq p,\ x_i \in \mathbb{Z}, \end{aligned}$$

kjer je prvih p spremenljivk celoštevilskih. Primer konkretnega mešanega celoštevilskega linearnega programa smo si že ogledali na začetku poglavja 4. Uvedba standardne oblike nam omogoča enostavnejši zapis algoritmov v nadaljevanju, še posebej simpleksne metode kot del podprograma metod za reševanje mešanih celoštevilskih linearnih programov.

Včasih v mešanem celoštevilskem linearnem programu nastopajo celoštevilske spremenljivke, ki so s pogoji navzdol omejene z 0 in navzgor omejene z 1. Takim spremenljivkam rečemo binarne spremenljivke. Zaradi posebnega statusa jih pogosto posebej obravnavamo in označimo. Ponavadi predstavljajo indikator uporabe ali pojavitve nekega dogodka, izdelka oziroma objekta (glej recimo primer 4.15).

4.2.1 Povezava z linearnim programom

Razlika med linearnim programom (LP) in mešanim celoštevilskim linearnim programom (MILP) je, da ima MILP dodatne omejitve za celoštevilskost prvih p spremenljivk. Zaradi tega je z njim mogoče opisati bistveno več problemov kot z LP, saj je v veliko praktičnih problemih omejitev celoštevilskosti smiselna (glej primer 4.1).

Vseeno včasih odpravimo zahteve celoštevilskosti MILP in rešimo pripadajoč LP. Tako dobljena rešitev nam vrne neko informacijo o naravi problema s pomočjo katere lažje začrtamo smernice za reševanje MILP.

Definicija 4.7. Naj bo P_1 mešani celoštevilski linearni program zapisan v standardni obliki, pri katerem za prvih p spremenljivk velja omejitev celoštevilskosti, ostale pa so realne. Naj bo P_2 program, kjer za prvih p spremenljivk programa P_1 odpravimo omejitve celoštevilskosti, namensko funkcijo in ostale pogoje pa prepišemo. Program P_2 je tedaj *linearna relaksacija* programa P_1.

4.3 Teoretična umestitev in lastnosti

Mešano celoštevilsko linearno programiranje je reševanje optimizacijskega problema podanega z matematičnim modelom v obliki mešanega celoštevilskega linearnega programa. Pri izdelavi matematičnega modela za optimizacijski problem skušamo kar najbolje opisati realno stanje. Obstaja veliko optimizacijskih metod reševanja, s katerimi rešujemo optimizacijske probleme (glej recimo [11]). Mešano celoštevilsko linearno programiranje sodi v družino metod reševanj sorodnih linearnemu programiranju. Navedimo le nekaj sorodnih metod kot reference:

- linearno programiranje (LP; angl. linear programming),
- celoštevilsko linearno programiranje (ILP; angl. integer linear programming) – vse spremenljivke so celoštevilske, včasih pa se izraz uporablja tudi kot sinonim za MILP,
- binarno celoštevilsko programiranje (BIP; angl. binary integer programming) – podobno kot ILP, le da so spremenljivke binarne,
- mešano celoštevilsko linearno programiranje (MILP; angl. mixed integer linear programming),
- kvadratično programiranje (QP; angl. quadratic programming) – funkcional oziroma namenska funkcija je lahko v kvadratni odvisnosti od svojih spremenljivk, omejitve so linerne,
- nelinearno programiranje (NLP; angl. nonlinear programming) – tako namenska funkcija kot omejitve so lahko nelinearne, model optimizacijskega problema pa je podan v linearnemu programu sorodni obliki.

Za metode je značilno, da zahtevajo natančen opis in je samo podajanje problema v tej obliki dokaj togo. V nadaljevanju dela bomo obravnavali le MILP in LP.

4.3.1 Dopustne rešitve

Nabor vrednosti spremenljivk x_i oziroma vrednost vektorja $\boldsymbol{x}$ imenujemo rešitev optimizacijskega problema. Rešitve optimizacijskih problemov delimo na:

1. Nedopustne ali napačne, to so rešitve, ki ne upoštevajo vseh omejitev.

2. Dopustne ali sprejemljive, to so rešitve, ki zadoščajo omejitvam problema.

Definicija 4.8. Za minimizacijski problem z namensko funkcijo f je rešitev $\boldsymbol{x}$ *boljša* od $\boldsymbol{y}$, če je $f(\boldsymbol{x}) \leq f(\boldsymbol{y})$. Rešitvi, ki je boljša od vseh ostalih, rečemo *najboljša* rešitev.

Pojma sta definirana za splošne rešitve, brez upoštevanja dopustnosti. Tako je lahko neka nedopustna rešitev boljša od dopustne, seveda pa nas za končni rezultat zanimajo le dopustne rešitve.

Definicija 4.9. Najboljšo izmed vseh dopustnih rešitev imenujemo *optimalna rešitev.*

Najboljših (in optimalnih) rešitev je lahko več.

Sedaj si oglejmo poseben odnos med rešitvami MILP in rešitvami relaksacije MILP na LP (glej razdelek 4.2.1). Vsaka dopustna rešitev MILP je tudi dopustna rešitev pripadajoče relaksacije na LP. Optimalna rešitev MILP ni nujno optimalna rešitev relaksacije na LP. Optimalna rešitev LP je optimalna rešitev MILP, če so upoštevane omejitve MILP za celoštevilskost spremenljivk.

Zgornji odnos lahko opazujemo tudi v večji splošnosti. Naj bo P_1 optimizacijski problem, ki mu dodamo dodatne pogoje, da iz njega dobimo problem P_2. Optimalna rešitev problema P_2 ne bo boljša od optimalne rešitve problema P_1. To je enostavna posledica dejstva, da so vse dopustne rešitve problema P_2 tudi dopustne rešitve P_1. To dejstvo bo pomembno predvsem v nadaljevanju za razumevanje delovanja algoritma razveji in obreži v razdelku 4.4.2.

Definicija 4.10. Naj bo P minimizacijski problem z namensko funkcijo f. Naj bo $k \in \mathbb{R}$ in naj za vsako dopustno rešitev $\boldsymbol{x}$ problema P velja $k \leq f(\boldsymbol{x})$. Potem je k spodnja meja za optimalno rešitev, oziroma zgornja meja za kakovost optimalne rešitve.

Definicija 4.11. Naj bo P minimizacijski problem z namensko funkcijo f. Naj bo $\boldsymbol{x}$ sprejemljiva rešitev problema P. Potem je $k = f(\boldsymbol{x})$ spodnja meja za optimalno rešitev, oziroma zgornja meja za kakovost optimalne rešitve.

Vrednost namenske funkcije pri optimalni rešitvi je med zgornjo in spodnjo mejo za kakovost rešitve.

4.3.2 Časovna zahtevnost

Definicija 4.12. Problemi razreda NP so problemi, ki bi jih nedeterministični Turningov stroj lahko rešil v polinomskem času glede na velikost vhodnih podatkov problema.

NP problemi so problemi, katerih rešitev lahko preverimo v polinomskem času glede na velikost vhodnih podatkov.

Definicija 4.13. Polinomska prevedba problema P_1 na problem P_2 je postopek, kjer v polinomskem času glede na velikost vhodnih podatkov preoblikujemo problem P_1 v problem P_2, pri katerem lahko v polinomskem času glede na velikost vhodnih podatkov pridobimo rešitev problema P_1 iz rešitve problema P_2.

V definiciji 4.13 je zahtevnost problema P_2 vsaj tolikšna kot zahtevnost problema P_1. Kot enostaven primer je možno problem P_1 iskanja najmanjšega elementa množice M prevesti na problem P_2 urejanja elementov množice M po velikosti. Iz rešitve P_2 v tem primeru trivialno dobimo rešitev P_1, ki je prvi element urejenih elementov.

Definicija 4.14. Problem P_2 iz razreda NP, je NP-poln, če lahko nanj s polinomsko prevedbo prevedemo poljuben problem P_1 iz razreda NP.

Literatura o temi NP-polnih problemov je dokaj obsežna, z njo se ukvarjajo predvsem računalniški teoretiki in matematiki. Več o temi si lahko preberete v [16]. Mešano celoštevilsko linearno programiranje sodi med NP-polne probleme. Nanj se z enostavnimi polinomskimi prevedbami prevede mnogo drugih NP-polnih problemov, s čimer pokažemo, da je njegova zahtevnost vsaj takšna kot zahtevnost le teh. Znan NP-poln problem je recimo problem nahrbtnika, ki ga zlahka zapišemo kot mešani celoštevilski linearni program. Oglejmo si primer prevedbe problema nahrbtnika na mešani celoštevilski linearni program.

Primer 4.15. Recimo, da imamo nahrbtnik, v katerega želimo spraviti čim več predmetov. V nahrbtniku lahko nesemo največ m kilogramov vsebine, sicer se nahrbtnik raztrga. Naša naloga je, da ugotovimo, katere predmete se nam splača vzeti.

Predmete označimo z indeksi $1, 2, \ldots, n$. Njihove mase izražene v kilogramih označimo z $w_1, w_2, \ldots, w_n$ in njihove vrednosti z $c_1, c_2, \ldots, c_n$. Indikator ali smo predmet i vzeli pa označimo z x_i, le ta je enak 1 če smo predmet vzeli, sicer je 0. Naša naloga je rešiti naslednji celoštevilski linearni program:

$$\begin{aligned} \text{maksimiziraj:} \quad & c_1x_1 + c_2x_2 + \cdots + c_nx_n, \\ \text{pri pogojih:} \quad & w_1x_1 + w_2x_2 + \cdots + w_nx_n \leq 10 \\ & x_1, x_2, \ldots, x_n \in \{0, 1\}, \end{aligned}$$

s čimer rešimo tudi problem nahrbtnika.

Časovna zahtevnost mešanega celoštevilskega linearnega programiranja je vsaj tolikšna kot časovna zahtevnost linearnega programiranja, saj je njegova posplošitev (vse LP se lahko zapiše kot MILP). Izkazalo pa se je, da je tudi linearno programiranje, ki se v praksi sicer že dolgotrajno uspešno uporablja, v teoriji časovno zahteven problem. Dokazano je, da za linearno programiranje obstaja algoritem s polinomskim številom operacij glede na velikost vhodnih podatkov (šibka polinomska zahtevnost, glej [14]). Vendar pa zaradi odvisnosti velikosti v algoritmu uporabljenih števil od vhodnih podatkov izvajanje algoritma ni stroge polinomske časovne zahtevnosti. Slednje sledi iz dejstva, da so tudi zahtevnosti številskih operacij odvisne od velikosti števil.

Eden od odprtih problemov 21. stoletja je tudi problem linearnega programiranja, ki se glasi: "Poišči strogi polinomski algoritem, ki za dano matriko A in vektor b odloči ali obstaja tak x, da velja $Ax > b$." Problem spada med odprte Smalove probleme (glej [18]).

4.4 Algoritmi

Za reševanje celoštevilskih linearnih programov obstaja veliko algoritmov in prijemov (glej [15] in [19]). V grobem jih delimo na algoritme za reševanje linearnih programov in algoritme za reševanje mešanih celoštevilskih linearnih programov, ki v svojih pomožnih metodah uporabljajo algoritme za reševanje linearnih programov. Za reševanje linearnih programov poznamo dva bistveno različna pristopa. Algoritmi za reševanje mešanih celoštevilskih linearnih programov pa se v glavnem razlikujejo le po količini uporabljenih idej znotraj algoritma. Najboljši algoritmi namreč po potrebi uporabljajo vse obstoječe principe.

Najbolj znani metodi za reševanje linearnih programov:

1. *Metoda simpleksov* (angl. simplex method) je najbolj uveljavljena in preizkušena v praksi ter primerna predvsem za probleme zmernih velikosti. Poleg standardne metode simpleksov uporabljamo še dualno metodo simpleksov in dvofazno metodo simpleksov.

2. *Metoda notranje točke* (angl. interior point method) je pogosta alternativa za simpleksno metodo pri problemih z velikim številom spremenljivk, za razliko od nje ima namreč zagotovljeno polinomsko število operacij glede na velikost problema.

Algoritmi za reševanje linearnih programov se uporabljajo kot pomožne metode pri mešanem celoštevilskem linearnem programiranju. Sedaj naštejmo nekaj algoritmov in metod, ki se lahko uporabljajo za mešano celoštevilsko linearno programiranje:

1. Pri *reševanju relaksacije na LP* (angl. LP relaxation solving) umaknemo pogoj celoštevilskosti ter rešujemo problem z metodami za reševanje linearnih programov. Včasih tako dobljena končna rešitev zadošča prvotnim pogojem celoštevilskosti in dobimo rešitev prvotnega programa.

2. *Hevristične metode* (angl. hevristic methods) so pogosto uporabljene za iskanje neke dopustne začetne rešitve. To so ponavadi na neki ideji razviti algoritmi, ki nas praviloma usmerijo proti optimalni rešitvi.

3. *Metoda obrezovanja ravnin* (angl. cutting plane method) je uporabljena v kombinaciji z reševanjem relaksacije na LP. Uvede dodatne pogoje, ki zmanjšajo prostor rešitev relaksacije na LP, pri tem pa ohranijo prvotni prostor rešitev nespremenjen (odrežemo nekatere neceloštevilske rešitve relaksacije).

4. *Razveji in omeji* (angl. branch and bound) razdeli problem na dva podproblema, ponavadi z delitvijo neke spremenljivke na dve možnosti $x_i \leq k$ ter $x_i \geq k+1$, kjer je $k \in \mathbb{Z}$. Delitev je lahko tudi splošnejša. V nadaljevanju so dopustne rešitve nekega podproblema vodilo pri reševanju drugega. Kot primer, če smo v prvem podproblemu s pomočjo hevrističnih metod dobili rešitev, ki je boljša od rešitve linearne relaksacije drugega podproblema, lahko drug podproblem brez škode zavržemo, saj je optimalna rešitev zagotovo v prvem podproblemu.

5. *Razveji in obreži* (angl. branch and cut) je nadgradnja algoritma razveji in omeji, kjer se za potrebe boljšega približka relaksacije na LP uporablja tudi metoda obrezovanja ravnin. Ta algoritem združuje vse zgoraj navedene algoritme in principe.

6. *Razveji in oceni* (angl. branch and price) je nadgradnja algoritma razveji in omeji, ki se uporablja za ogromne probleme. Na začetku se pri algoritmu uporabljajo le nekateri stolpci matrike A, ki se tekom algoritma dodajajo. Če se v algoritmu uporablja še metoda obrezovanja ravnin, ga imenujemo *razveji, oceni in obreži* (angl. branch price and cut).

V nadaljevanju si bomo podrobneje ogledali metodo simpleksov in algoritem razveji in obreži, ki sta tudi standardno v uporabi pri naprednih reševalnikih, ki instance MILP praviloma uspešno rešujejo. V samem delu se z implementacijo teh algoritmov ne ukvarjamo, je pa njihovo razumevanje pomembno za dober pregled področja, saj so bistvenega pomena za reševanje problema. Tekom dela bomo pokazali, da je implementacija kakovostnih reševalnikov že dokaj rešen problem, zato se največje pridobitve lahko dobijo z izdelavo dobrega in učinkovitega modela realnega problema, ki ga z uporabo reševalnikov nato rešujemo. Več o reševalnikih je na voljo v poglavju 7 o implementaciji modela in poglavju 8.3 s primerjavo reševalnikov.

4.4.1 Metoda simpleksov

Metodo simpleksov uporabljamo za reševanje linearnih programov zapisanih v standardni obliki

$$\begin{aligned} \text{minimiziraj:} \quad & \boldsymbol{c}^T\boldsymbol{x}, \\ \text{pri pogojih:} \quad & \boldsymbol{A}\boldsymbol{x} \leq \boldsymbol{b} \\ & \boldsymbol{x} \geq \boldsymbol{0}, \end{aligned}$$

kjer je $\boldsymbol{c} \in \mathbb{R}^s$, $\boldsymbol{A} \in \mathbb{R}^{m\times s}$ in $\boldsymbol{b} \in \mathbb{R}^m$. Kot rešitev linearnega programa dobimo vektor $\boldsymbol{x} \in \mathbb{R}^s$, ki zadošča vsem pogojem in minimizira namensko funkcijo $f(\boldsymbol{x}) = \boldsymbol{c}^T\boldsymbol{x}$.

V nadaljevanju si bomo pogledali metodo simpleksov za linearni program zapisan v standardni obliki, pri katerem obstaja natanko ena optimalna rešitev. Metodo simpleksov lahko opišemo z naslednjim postopkom:

1. **Uvedemo dodatne spremenljivke.** Problem najprej zapišemo v taki obliki, da so razen pogojev o nenegativnosti vsi pogoji zapisani v obliki enačb. To dosežemo z uvedbo dodatnih spremenljivk glej razdelek 4.1.1 točko 4. Za

vsako neenačbo uvedemo eno dodatno spremenljivko. Tako za i-to neenačbo uvedemo pozitivno spremenljivko x_{s+i} kot

$$x_{s+i} = b_i - a_{i,1}x_1 - \cdots - a_{i,s}x_s,$$

ker velja $x_{s+1} \geq 0$, je to ravno ekvivalentno prvotnemu i-temu pogoju. Na ta način dobimo m dodatnih spremenljivk. Vsega skupaj imamo tako $s + m = n$ spremenljivk, torej $\boldsymbol{x} \in \mathbb{R}^n$.

Dodatno uvedemo tudi funkcional z za namensko funkcijo

$$z = k + c_1x_1 + c_2x_2 + \cdots + c_sx_s,$$

kjer je $k = 0$ števílska vrednost funkcionala z. Zapis v obliki funkcionala je potreben, saj bomo v nadaljevanju večkrat isti funkcional zapisali z različnimi spremenljivkami.

2. **Poiščemo neko dopustno rešitev.** Poiščemo neko dopustno rešitev $\boldsymbol{x}^S \in \mathbb{R}^n$. S pomočjo dopustne rešitve in prijemov za pretvorbe linearnih programov lahko model preoblikujemo v standardni zapis z enakostjo (glej nadaljevanje). Prijemi, ki jih pri tem lahko uporabimo, so zapisani v razdelku 4.1.1, kako to storimo pa je razvidno tudi iz primera 4.18.

3. **Model preoblikujemo v standardni zapis z enakostjo.** Model preoblikujemo, tako da ga lahko zapišemo v obliki

$$\begin{aligned} \text{minimiziraj:} \quad & z = k + \boldsymbol{c}^T\boldsymbol{x}, \\ \text{pri pogojih:} \quad & \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b} \\ & \boldsymbol{x} \geq \boldsymbol{0}, \end{aligned}$$

kjer so $k \in \mathbb{R}$, $\boldsymbol{c} \in \mathbb{R}^n$, $\boldsymbol{A} \in \mathbb{R}^{m\times n}$, $\boldsymbol{b} \in \mathbb{R}^m$ ter za njih velja, da so oblike

$$\begin{aligned} &\boldsymbol{c} = (c_1, \ldots, c_s, 0, \ldots, 0), \\ &\boldsymbol{A} = [\boldsymbol{A}' \quad \boldsymbol{I_m}], \qquad \text{kjer sta } \boldsymbol{A}' \in \mathbb{R}^{m\times s}, \boldsymbol{I_m} \in \mathbb{R}^{m\times m}, \\ &\boldsymbol{b} \geq \boldsymbol{0}. \end{aligned}$$

To pomeni, da je vektor $\boldsymbol{c}$ neničelen le na prvih s komponentah in spremenljivke $x_{s+1}, \ldots, x_n$, ne doprinesejo k vrednosti funkcionala z. Tedaj je $\boldsymbol{x} = (0, 0, \ldots, 0, b_{s+1}, \ldots, b_n)$ dopustna rešitev z vrednostjo k.

Za model zapisan v tej obliki spremenljivke $x_1, \ldots, x_s$ imenujemo nebazne, spremenljivke $x_{s+1}, \ldots, x_n$ pa bazne. Razlog za tako poimenovanje je sistem enačb omejitev, ki ga imenujemo slovar in je za zgornji zapis naslednji

$$\begin{aligned} x_{s+1} &= b_1 - a_{1,1}x_1 - a_{1,2}x_2 - \cdots - a_{1,s}x_s, \\ x_{s+2} &= b_2 - a_{2,1}x_1 - a_{2,2}x_2 - \cdots - a_{2,s}x_s, \\ &\cdots \\ x_n &= b_m - a_{m,1}x_1 - a_{m,2}x_2 - \cdots - a_{m,s}x_s. \end{aligned}$$

Poudariti je potrebno, da eksplicitno preindeksiranje (pri preoblikovanju modela) ni potrebno in smo ga naredili zato, da se na indekse baznih in nebaznih spremenljivk lažje sklicujemo v nadaljnjem opisu. Bistveno pa je, da je model zapisan v obliki, kjer lahko vse nebazne spremenljivke nastavimo na 0 in dobimo neko dopustno rešitev. Tedaj so bazne spremenljivke x_i enake b_i in zato pozitivne.

4. **Izberemo vstopajočo spremenljivko.** Izberemo nebazno spremenljivko z negativnim koeficientom v funkcionalu z. Negativnim zato, ker želimo namensko funkcijo minimizirati. Ponavadi izberemo tako, s katero bomo lahko najbolj zmanjšali številsko vrednost (trenutno dosežen minimum) funkcionala z. Imenujemo jo vstopajoča spremenljivka. Glej tudi opombo 4.16.

5. **Izberemo izstopajočo spremenljivko.** Pogoji nenegativnosti baznih spremenljivk omejujejo vrednosti vstopajoče spremenljivke. Izberemo tisto bazno spremenljivko, katere pripadajoči pogoj je najbolj omejujoč. Izbrano spremenljivko imenujemo izstopajoča spremenljivka. Glej tudi opombo 4.16.

6. **Zamenjamo spremenljivki in posodobimo enačbe.** Izrazimo vstopajočo spremenljivko x_i s pomočjo enačbe izstopajoče spremenljivke x_j. Iz enačbe

$$x_j = b_j - a_{j,1}x_1 - \cdots - a_{j,i}x_i - \cdots - a_{j,s}x_s$$

izrazimo spremenljivko x_i kot

$$x_i = \frac{b_j}{a_{j,i}} - \frac{a_{j,1}}{a_{j,i}}x_1 - \cdots - \frac{a_{j,i-1}}{a_{j,i}}x_{i-1} - \frac{a_{j,i+1}}{a_{j,i}}x_{i+1} \cdots - \frac{a_{j,s}}{a_{j,i}}x_s - \frac{1}{a_{j,i}}x_j$$

in naredimo substitucijo v ostalih enačbah. Vstopajoča spremenljivka x_i postane bazna spremenljivka, izstopajoča spremenljivka x_j postane nebazna spremenljivka.

7. **Ponavljamo postopek dokler ne najdemo rešitve.** Dokler je v funkcionalu z še kakšna spremenljivka z negativnim koeficientom, ponavljamo točke 3 do 6 (glej opombo 4.17). Ko imajo vse spremenljivke v funkcionalu pozitivni koeficient, zaključimo s postopkom – našli smo optimalno rešitev. Neničelnost spremenjivk v funkcionalu z bi v tem primeru namreč samo še povečala vrednost funkcionala. Torej so vse spremenljivke, ki nastopajo v funkcionalu z, enake 0. To so ravno nebazne spremenljivke. Sedaj odčitamo bazne spremenljivke, ki so hkrati tudi rešitev prvotnega linearnega programa. Minimum funkcionala z pa je enak k (številski vrednosti funkcionala z).

Opomba 4.16. Izbira vstopajoče in izstopajoče spremenljivke ni natančno definirana. Različni algoritmi in postopki izbirajo spremenljivke po različnih kriterijih. Pri ročnem reševanju za vstopajočo spremenljivko ponavadi izberemo tisto z najbolj negativnim koeficientom v funkcionalu z. Pri reševanju z računalnikom pa se lahko že na tem koraku upošteva omejitve, ki za dano spremenljivko veljajo, in izbere tisto, pri kateri bo največji napredek (glej tudi opombo 4.17).

Zanimivo je, da bi pri ročnem reševanju z uvedbo nove spremenljivke $x_i' = kx_i$ lahko spremenili potek reševanja. Premislek prepustimo bralcu.

Opomba 4.17. Pri neprevidni izbiri vstopnih in izstopnih spremenljivk ni zagotovila, da se bo podani postopek kadar koli končal (lahko pride do neskončnega cikličnega ponavljanja postopka). Končno število korakov algoritma pa na primer lahko zagotovimo z uporabo Blandovega pravila (glej [4]).

Z metodo simpleksov rešimo naslednji primer prirejen po nalogi iz učbenika [9].

Primer 4.18. Tri vrste vitaminov lahko kupimo v štirih različnih praških. Gram prvega praška ima 4 enote prvega in 2 enoti drugega vitamina; gram drugega praška ima 1 enoto prvega, 5 enot drugega in 2 enoti tretjega vitamina; gram tretjega praška ima 2 enoti prvega in 4 enote tretjega vitamina; gram četrtega praška ima 2 enoti prvega, 4 enote drugega in 1 enoto tretjega vitamina. Gram prvega praška stane 2, drugega 6, tretjega 8 in četrtega 5 denarnih enot. Kupiti moramo najmanj 80 enot prvega, 40 enot drugega in 120 enot tretjega vitamina. Koliko gramov praška vsakega tipa naj kupimo, da zadostimo zahtevi po vitaminih in da so stroški nakupa najmanjši?

Nastavimo model

$$\begin{aligned}
&\text{minimiziraj: } f(\boldsymbol{x}) = 2x_1 + 6x_2 + 8x_3 + 5x_4,\\
&\text{pri pogojih: } \quad v_1 = 4x_1 + x_2 + 2x_3 + 2x_4 \geq 80\\
&\qquad\qquad\qquad v_2 = 2x_1 + 5x_2 \qquad\; + 4x_4 \geq 40\\
&\qquad\qquad\qquad v_3 = \qquad\; 2x_2 + 4x_3 + x_4 \geq 120\\
&\qquad\qquad\qquad \boldsymbol{x} \geq \boldsymbol{0}.
\end{aligned}$$

Eksplicitno navedimo standardno obliko za ta linearni program

$$\begin{aligned}
&\text{minimiziraj: } f(\boldsymbol{x}) = 2x_1 + 6x_2 + 8x_3 + 5x_4,\\
&\text{pri pogojih: } \quad -4x_1 - x_2 - 2x_3 - 2x_4 \leq -80\\
&\qquad\qquad\qquad -2x_1 - 5x_2 \qquad\; - 4x_4 \leq -40\\
&\qquad\qquad\qquad\qquad\quad - 2x_2 - 4x_3 - x_4 \leq -120\\
&\qquad\qquad\qquad \boldsymbol{x} \geq \boldsymbol{0}.
\end{aligned}$$

Uvedemo dodatne spremenljivke:

$$\begin{aligned}
x_5 &= -80 + 4x_1 + x_2 + 2x_3 + 2x_4,\\
x_6 &= -40 + 2x_1 + 5x_2 \qquad\; + 4x_4,\\
x_7 &= -120 \qquad\quad + 2x_2 + 4x_3 + x_4,
\end{aligned}$$

kjer so $x_1, x_2, \ldots x_7 \geq 0$. Poiščemo neko ustrezno začetno rešitev

$$x_2 = 80, x_1 = 0, x_3 = 0, x_4 = 0, x_5 = 0, x_6 = 360, x_7 = 40.$$

Preoblikujemo model v obliko, kjer je zadoščeno vsem pogojem, če so nebazne spremenljivke enake nič

$$\begin{aligned}
x_2 &= 80 - 4x_1 - 2x_3 - 2x_4 + x_5,\\
x_6 &= 360 - 18x_1 - 10x_3 - 6x_4 + 5x_5,\\
x_7 &= 40 - 8x_1 \qquad\quad - 3x_4 + 2x_5.
\end{aligned}$$

Za nov funkcional namenske funkcije dobimo

$$z = 480 - 22x_1 - 4x_3 - 7x_4 + 6x_5,$$

iz česar je razvidno, da bo minimum namenske funkcije največ 480. Računamo naprej. Izberemo tako spremenljivko funkcionala z, ki nam lahko čim več doprinese k minimiziranju funkcionala. Za izstopno spremenljivko izberemo x_1, ki ima absolutno največji negativni koeficient.

Poiščemo najbolj omejujočo omejitev za spremenljivko x_1. Omejitve so

$$\begin{aligned}
x_2 &= 80 - 4x_1 \geq 0 \implies x_1 \leq 20,\\
x_6 &= 360 - 18x_1 \geq 0 \implies x_1 \leq 20,\\
x_7 &= 40 - 8x_1 \geq 0 \implies x_1 \leq 5.
\end{aligned}$$

Najbolj omejujoča je omejitev $x_1 \leq 5$, zato izberemo x_7 za izstopno spremenljivko. Ustrezno preoblikujemo model. Dobimo nove pogoje

$$\begin{aligned}
x_1 &= 5 - \tfrac{3}{8}x_4 + \tfrac{1}{4}x_5 - \tfrac{1}{8}x_7,\\
x_2 &= 60 - 2x_3 - \tfrac{5}{2}x_4 - \tfrac{1}{2}x_7,\\
x_6 &= 270 - 10x_3 + \tfrac{3}{4}x_4 - \tfrac{1}{2}x_5 + \tfrac{9}{4}x_7,
\end{aligned}$$

in funkcional namenske funkcije

$$z = 370 - 4x_3 + \frac{5}{4}x_4 + \frac{1}{2}x_5 + \frac{11}{4}x_7.$$

Funkcional nam pove, da bo minimum namenske funkcije največ 370, vidimo pa tudi, da se vrednost lahko izboljša le na spremenljivki x_3, ki ima edina negativen koeficient. Zato za vstopno spremenljivko izberemo spremenljivko x_3. Oglejmo si omejitve, ki jih za to spremenljivko imamo,

$$\begin{aligned}
x_2 &= 60 - 2x_3 \geq 0 \implies x_3 \leq 30,\\
x_6 &= 270 - 10x_3 \geq 0 \implies x_3 \leq 27.
\end{aligned}$$

Ti sta le dve, saj spremenljivka x_3 ne nastopa v enačbi bazne spremenljivke x_1. Od teh dveh omejitev je najbolj omejujoča omejitev $x_3 \leq 27$, zato za izstopno spremenljivko izberemo x_6. Ustrezno preoblikujemo model in dobimo nove pogoje

$$\begin{aligned}
x_1 &= 5 - \tfrac{3}{8}x_4 + \tfrac{1}{4}x_5 - \tfrac{1}{8}x_7,\\
x_2 &= 6 - \tfrac{53}{2}x_4 - \tfrac{1}{10}x_5 - \tfrac{1}{5}x_6 - \tfrac{19}{20}x_7,\\
x_3 &= 27 + \tfrac{3}{40}x_4 - \tfrac{1}{20}x_5 - \tfrac{1}{10}x_6 - \tfrac{9}{40}x_7,
\end{aligned}$$

in funkcional namenske funkcije

$$z = 262 + \frac{19}{20}x_4 + \frac{7}{10}x_5 + \frac{2}{5}x_6 + \frac{73}{20}x_7.$$

Ker velja $x_1, x_2, \ldots x_7 \geq 0$, lahko doseže dan funkcional najmanj 262, saj je zaradi pozitivnih koeficientov pred spremenljivkami x_4, x_5, x_6, x_7 funkcional najmanjši, če nastavimo vrednosti teh spremenljivk na 0.

Tako za optimalno rešitev dobimo $x_1 = 5$, $x_2 = 6$ in $x_3 = 27$ s pripadajočim minimumom 262.

V danem primeru smo morali poiskati neko začetno rešitev, da smo lahko začeli z izvajanjem ponavljajočega postopka algoritma. Da dobimo neko začetno dopustno rešitev, se lahko poslužujemo več različnih prijemov. Eden od prijemov je, da rešimo razširjeni linearni program kot je razloženo v [19]. Optimalna rešitev razširjenega linearnega programa je neka začetna dopustna rešitev za naš program. Včasih pa se izkaže, da je lažje, če namesto prvotnega linearnega programa rešimo njegov dualni linearni program. Z vrednostjo funkcionala optimalne rešitve dualnega programa že poznamo optimalno vrednost prvotnega linearnega programa. Velja namreč, da sta optimalni vrednosti namenske funkcije obeh programov enaki, če le ti obstajata. Če pa je dualni program rešen s pomočjo simpleksne metode, pa vemo še nekaj več. Iz končnega zapisa funkcionala lahko namreč preberemo rešitev $\boldsymbol{x}$ prvotnega linearnega programa. To bomo navedli brez dokaza, oglejmo si le primer, ki pokaže, kako to naredimo. Več o uporabi dualnega programa za reševanje linearnega programa si lahko preberete v [9] in [19].

Primer 4.19. Dva tipa vitaminov lahko dobimo v dveh vrstah praškov. Gram prvega praška ima 1 enoto prvega in 4 enote drugega vitamina; gram drugega praška ima 5 enot prvega in 1 enoto drugega vitamina. Gram prvega praška stane 24 drugi pa 25 denarnih enot. Koliko gramov praška vsake vrste naj kupimo, da dobimo najmanj 17 enot prvega in najmanj 11 enot drugega vitamina in bodo stroški nakupa najmanjši?

Linearni program

$$\begin{aligned} &\text{minimiziraj: } f(\boldsymbol{x}) = 24x_1 + 25x_2,\\ &\text{pri pogojih: } \quad v_1 = x_1 + 5x_2 \geq 17\\ &\qquad\qquad\qquad v_2 = 4x_1 + x_2 \geq 11\\ &\qquad\qquad\qquad \boldsymbol{x} \geq \boldsymbol{0}, \end{aligned}$$

zapišimo v dualni obliki

$$\begin{aligned} &\text{maksimiziraj: } z = 17v_1 + 11v_2,\\ &\text{pri pogojih: } \quad v_1 + 4v_2 \leq 24\\ &\qquad\qquad\qquad 5v_1 + v_2 \leq 25\\ &\qquad\qquad\qquad \boldsymbol{v} \geq \boldsymbol{0}. \end{aligned}$$

Dobljen linearni program brez pretvorbe v standardno obliko rešujmo po metodi simpleksov in upoštevamo, da namensko funkcijo maksimiziramo. Ker je rešitev $v_1 = 0$, $v_2 = 0$ že dopustna, preoblikujemo model v standardni zapis z enakostjo. Dobimo sistem enačb

$$\begin{aligned} v_3 &= 24 - v_1 - 4v_2,\\ v_4 &= 25 - 5v_1 - v_2, \end{aligned}$$

kjer so $v_1, v_2, v_3, v_4 \geq 0$. Izberemo spremenljivko z največjim (ker maksimiziramo) koeficientom v funkcionalu z, to je spremenljivka v_1. Sedaj poiščemo pogoj, ki je za spremenljivko v_1 najbolj omejujoč

$$\begin{aligned} v_3 &= 24 - v_1 \geq 0 \implies v_1 \leq 24,\\ v_4 &= 25 - 5v_1 \geq 0 \implies v_1 \leq 5, \end{aligned}$$

torej pogoj spremenljivke v_4. Izberemo v_1 kot vstopno in v_4 kot izstopno spremenljivko. Dobimo nov sistem enačb

$$\begin{aligned} v_1 &= 5 - \tfrac{1}{5}v_2 - \tfrac{1}{5}v_4, \\ v_3 &= 19 - \tfrac{19}{5}v_2 + \tfrac{1}{5}v_4, \end{aligned}$$

funkcional z pa je enak

$$z = 85 + \frac{38}{5}v_2 - \frac{17}{5}v_4.$$

Za vstopno spremenljivko izberemo v_2, saj je edina pozitivna spremenljivka, ki nastopa v funkcionalu. Oglejmo si, kakšni so pogoji, ki jo omejujejo

$$\begin{aligned} v_1 &= 5 - \tfrac{1}{5}v_2 \geq 0 \implies v_2 \leq 25, \\ v_3 &= 19 - \tfrac{19}{5}v_2 \geq 0 \implies v_2 \leq 5. \end{aligned}$$

Pogoj pri spremenljivki v_3 je najbolj omejujoč, zato jo izberemo kot izstopno spremenljivko. Naredimo substitucijo spremenljivk in dobimo enačbe

$$\begin{aligned} v_1 &= 4 + \tfrac{1}{19}v_3 - \tfrac{18}{95}v_4, \\ v_2 &= 5 - \tfrac{5}{19}v_3 + \tfrac{1}{19}v_4, \end{aligned}$$

ter namensko funkcijo

$$z = 123 - 2v_3 - 3v_4,$$

iz česar lahko preberemo optimalno vrednost prvotne namenske funkcije 123 in rešitev prvotnega linearnega programa $x_1 = 2$ in $x_2 = 3$.

4.4.2 Algoritem razveji in obreži

Algoritem razveji in obreži uporabljamo za reševanje celoštevilskih linearnih programov zapisanih v standardni obliki

$$\begin{aligned} \text{minimiziraj:} \quad & \boldsymbol{c}^T \boldsymbol{x}, \\ \text{pri pogojih:} \quad & \boldsymbol{A}\boldsymbol{x} \leq \boldsymbol{b} \\ & \boldsymbol{x} \geq \boldsymbol{0} \\ & \forall i \leq p,\ x_i \in \mathbb{Z}, \end{aligned}$$

kjer je prvih p spremenljivk celoštevilskih in velja da so $\boldsymbol{c} \in \mathbb{R}^s$, $\boldsymbol{A} \in \mathbb{R}^{m \times s}$ in $\boldsymbol{b} \in \mathbb{R}^m$. V algoritmu razveji in obreži uporabljamo naslednje uveljavljene tehnike za reševanje mešanega celoštevilskega linearnega programa:

1. **Relaksacija programa.** Za mešani celoštevilski linearni program odstranimo omejitev celoštevilskosti in dobimo linearni program. Rešimo linearni program in dobimo spodnjo mejo za rešitev (zgornjo mejo za kakovost rešitve, glej definicijo 4.10) mešanega celoštevilskega linearnega programa. Za reševanje linearnega programa uporabimo metodo simpleksov oziroma metodo notranje točke za večje probleme. Glej tudi 4.2.1 in 4.3.1.

2. **Hevristike za dopustno rešitev.** S pomočjo hevrističnih metod iščemo dopustne rešitve. Dopustna rešitev je pomembna, saj nam poda zgornjo mejo (spodnjo mejo za kakovost, glej definicijo 4.11) končne rešitve. Ključna je pri učinkovitem preiskovanju prostora rešitev.

3. **Delitev problema.** Celoštevilski linearni program lahko razdelimo na dva dela. Najenostavnejše so delitve tipa $x_i \geq k+1$ in $x_i \leq k$, kjer problem razdelimo glede na to ali je celoštevilska spremenljivka x_i večja ali manjša od $k \in \mathbb{Z}$. Pri tem sicer nobeden od podproblemov ni manjši, saj se število spremenljivk ohrani, število omejitev pa celo poveča. Vendar pa v posebnem primeru ko je x_i omejen le na dve vrednosti taka omejitev zmanjša število spremenljivk za eno. Tudi v ostalih primerih je to dobra tehnika, saj rešitev nekega podproblema vpliva na postopek reševanja pri ostalih podproblemih. Vsaka dopustna rešitev je namreč zgornja meja (spodnja meja kakovosti) za rešitev prvotnega problema, zato v nekaterih primerih ni potrebno nadaljevati z reševanjem podproblemov, saj najboljša rešitev podproblema ne more preseči do sedaj dobljene. Oglejmo si tehniki delitve na podprobleme in omejevanje glede na podprobleme.

   (a) Delitev na podprobleme je razdelitev problema glede na dve omejitvi, ki ju lahko zapišemo kot $\boldsymbol{a}^T\boldsymbol{x} \leq k$ in $\boldsymbol{a}^T\boldsymbol{x} > k$. Ponavadi se delitev naredi le za celoštevilske spremenljivke, tako razdelimo problem glede na

   $$a_1x_1 + \cdots + a_px_p \leq k \text{ in } a_1x_1 + \cdots + a_px_p > k.$$

   V posebnem primeru, ko so koeficienti a_i ter k pri celoštevilski delitvi celoštevilski, dobi dana delitev še prav posebno uporabno vrednost, saj jo lahko zapišemo kot

   $$a_1x_1 + \cdots + a_px_p \leq k \text{ in } a_1x_1 + \cdots + a_px_p \geq k+1,$$

   saj v tem primeru izraz $a_1x_1 + \cdots + a_px_p$ lahko zavzame le celoštevilske vrednosti. V posebnem primeru, ko je le eden izmed a_i neničelen, dobimo najpogosteje uporabljeno delitev, to je delitev glede na eno celoštevilsko spremenljivko.

   Za dan linearni program P z delitvijo na podprobleme dobimo linearna programa P_1 in P_2, kjer je v P_1 dodana omejitev oblike $\boldsymbol{a}^T\boldsymbol{x} \leq k$, v P_2 pa je dodana omejitev oblike $\boldsymbol{a}^T\boldsymbol{x} > k$.

   (b) Omejevanje glede na podprobleme uporabljamo za zavračanje reševanja nekaterih podproblemov, za katere vemo, da bo njihova optimalna rešitev neoptimalna rešitev prvotnega problema. Za vsak podproblem lahko ponavadi izračunamo rešitev relaksacije in neko dopustno rešitev izračunano s pomočjo hevristike. Naj bo P celoštevilski linearni program, P_1 in P_2 pa njegovi delitvi na podprograma. Naj bo $\boldsymbol{x}_1^R$ rešitev relaksacije P_1 in $\boldsymbol{x}_2^S$ dopustna rešitev P_2. Če je $\boldsymbol{x}_2^S$ boljša od rešitve $\boldsymbol{x}_1^R$, zaključimo z reševanjem linearnega programa P_1, saj bo imel zagotovo slabšo rešitev kot linearni program P_2.

4. **Obrezovanje prostora linearne relaksacije.** Velikokrat lahko z večdimenzionalnimi ravninami odrežemo del prostora dopustnih rešitev tako, da zmanjšamo prostor linearne relaksacije, ne pa tudi prostora prvotnega problema, zato ostane končna rešitev z vpeljavo odrezka večdimenzionalne ravnine (torej nekega dodatnega pogoja) nespremenjena. Vpeljava dodatnih pogojev pa pogosto pomaga pri omejevanju kakovosti rešitev podproblemov celoštevilskega linearnega programiranja. To lahko v kombinaciji s dopustno rešitvijo nekega drugega podproblema pripelje do tega, da lahko opustimo del računanja. Ugotovimo namreč, da za končno rešitev določen podproblem ni pomemben.

V grobem idejo algoritma razveji in obreži za mešani celoštevilski linearni program P opišemo z naslednjim postopkom:

1. Poišči rešitev relaksacije P na linearni program. Če je rešitev relaksacije tudi dopustna rešitev za mešani celoštevilski linearni program P, zaključi in vrni rešitev.

2. Sicer s pomočjo hevrističnih metod poišči kakšno dopustno rešitev, ki služi za smernico pri nadaljnjem reševanju.

3. Preoblikuj program v enega (obrezovanje; glej točko 4 zgoraj) ali dva podprograma (delitev; glej točko 3 zgoraj), katerih najboljša optimalna rešitev ustreza optimalni rešitvi P. Nadaljuj z reševanjem podprogramov.

4. Vrni najboljšo rešitev podprogramov.

V nadaljevanju si oglejmo primer konkretnega algoritma zgrajenega na tej ideji.

Primer 4.20. Podan naj bo mešani celoštevilski linearni program zapisan v standardni obliki. Algoritem razveji in obreži zapišemo s psevdokodo v sintaksi jezika Python kot sledi.

```python
def branch_and_cut(initial_program):
    """Branch and cut algorithm for solving MILP"""

    best_feasible_solution = Solution(None, float("inf"))
    task_list = TaskList()

    initial_relaxed_solution = simplex(initial_program.relax())
    task_list.smart_insert((initial_program, initial_relaxed_solution))

    while len(task_list) > 0:
        program, relaxed_solution = task_list.pop(0)

        if best_feasible_solution.value < relaxed_solution.value:
            continue

        if program.is_feasible(relaxed_solution):
            best_feasible_solution = relaxed_solution
```

```
            continue

        feasible_solution = heuristic_solve(program)
        if feasible_solution.value < best_feasible_solution.value:
            best_feasible_solution = feasible_solution

        constraint = heuristic_split(program)
        program1, program2 = program.split_on_constraint(constraint)

        if program1.is_maybe_feasible():
            relaxed_solution1 = simplex(program1.relax())
            task_list.smart_insert((program1, relaxed_solution1))

        if program2.is_maybe_feasible():
            relaxed_solution2 = simplex(program2.relax())
            task_list.smart_insert((program2, relaxed_solution2))

    return best_feasible_solution
```

V nadaljevanju opišimo razrede, metode in funkcije, uporabljene v tem algoritmu, ter po korakih obrazložimo posamezne dele zapisane psevdokode. Znotraj algoritma uporabljamo naslednje nestandardne pomožne konstrukte:

1. Razred `TaskList`, ki predstavlja seznam še ne razrešenih programov in pripadajočih rešitev relaksacije. Vrstni red programov v seznamu določa hevristika, ki jih razvrsti od najbolj do najmanj primernega kandidata za reševanje. Razred omogoča vstavljanje elementov v seznam z metodo `smart_insert` ter vračanje elementov iz seznama z metodo `pop`.

2. Razred `Solution`, ki vsebuje podatke o dani rešitvi. Objekt razreda konstruiramo tako, da podamo podatke o vrednostih spremenljivk in vrednost, ki jo mešani celoštevilski linearni program doseže za tako nastavljene spremenljivke. Neobstoječo rešitev določimo kot

   ```
   Solution(None, float("inf")).
   ```

3. Funkcijo `simplex`, ki kot vhod sprejme linearni program in vrne njegovo optimalno rešitev.

4. Funkcijo `heuristic_solve`, ki kot vhod sprejme mešani celoštevilski linearni program in vrne neko dopustno rešitev, oziroma

   ```
   Solution(None, float("inf")),
   ```

   če dopustne rešitve ne najde.

5. Funkcijo `heuristic_split`, ki kot vhod sprejme mešani celoštevilski linearni program in vrne omejitev, po kateri naj se linearni program razdeli na dva dela. Omejitev je izbrana po neki hevristiki.

6. Metodo `smart_insert`, ki jo kličemo nad seznamom nalog z argumentom dvojice mešanega celoštevilskega linearnega programa in vrednosti rešitve njegove relaksacije. Metoda vstavi dvojico na ustrezno mesto v seznamu. Mesto je določeno z neko hevristiko (več o tem v nadaljevanju).

7. Metodo `pop`, ki jo kličemo nad seznamom nalog. Kot argument ji podamo indeks izbranega elementa. Metoda izbran element vrne in pobriše iz seznama. Če želimo dostop do najbolj primernega kandidata za nadaljnje reševanje, jo kličemo kot `pop(0)`, s tem klicem se ta element vrne in hkrati izbriše iz seznama.

8. Metodo `relax`, ki jo kličemo nad mešanim celoštevilskim linearnim programom. Zanj vrne relaksacijo na linearni program.

Sedaj obrazložimo še posamezne dele psevdokode algoritma.

```
def branch_and_cut(initial_program):
    """Branch and cut algorithm for solving MILP"""
```

Definicija algoritma, ki kot vhod sprejme v standardni obliki zapisan mešani celoštevilski linearni program in vrne njegovo optimalno rešitev. Iščemo minimum.

```
    best_feasible_solution = Solution(None, float("inf"))
    task_list = TaskList()
```

Inicializacija trenutne najboljše rešitve in podatkovne strukture, ki hrani še neobdelane programe. V podatkovni strukturi hranimo po neki hevristiki urejene mešane celoštevilske linearne programe s pripadajočimi rešitvami relaksacij (le te so ponavadi osnova za hevristiko). Trenutna najboljša rešitev je na začetku neveljavna, njena vrednost je neskončna. Tekom delovanja algoritma jo posodobimo vsakič, ko naletimo na boljšo rešitev.

```
    initial_relaxed_solution = simplex(initial_program.relax())
    task_list.smart_insert((initial_program, initial_relaxed_solution))
```

Izračun relaksacije prvotnega mešanega celoštevilskega linearnega programa s pomočjo simpleks metode. Vstavitev prvotnega programa in rešitve pripadajoče relaksacije v podatkovno strukturo še ne obdelanih programov.

```
    while len(task_list) > 0:
        program, relaxed_solution = task_list.pop(0)
```

Zanka, ki se ponavlja, dokler je za določitev optimalne rešitve potrebno rešiti kakšen program. Na vsakem koraku se iz podatkovne strukture še ne razrešenih programov vzame naslednji program z rešitvijo pripadajoče relaksacije. Ker je vrstni red programov v podatkovni strukturi določen z neko hevristiko, je to program, za katerega se predvideva, da bo algoritem usmeril tako, da bo delovanje hitrejše napram alternativnim izbiram programa (glej opombo 4.21).

Opomba 4.21. Velikokrat je hevristika izbire linearnih programov določena glede na vrednost rešitve relaksacije. Ni pa nujno, v pošte se na primer lahko vzame tudi število razrešenih spremenljivk in število omejitev, ki jih program ima. Lahko se izbere tudi program, za katerega se predvideva, da bo imel dobro hevristično rešitev in bodo zaradi njega hitreje razrešeni ostali. Možnosti za izbiro hevristike je veliko.

Na kratko povzemimo idejo zanke od tu naprej. V nadaljevanju zanke se pregleda ali je rešitev programa lahko optimalna, če ni, se reševanje trenutnega programa zaključi. V vrsto se ne dodaja podprogramov, za katere se ve, da ne vsebujejo optimalne rešitve prvotnega podprograma. Nato se pregleda ali je že rešitev relaksacije programa ustrezna, če je ustrezna je trenutni program uspešno rešen in njegovo reševanje se zaključi. Sicer se poskuša najti čim boljšo hevristično rešitev, ki lahko pripomore k hitrejšemu pregledovanju programov. Program se nato razdeli na dva podprograma. Če je možno, da sta rešljiva, se ju doda v podatkovno strukturo še ne razrešenih programov. Tu se postopek ponovi za naslednjo izbiro programa. Sledi še postopni opis ukazov v zanki.

```
if best_feasible_solution.value < relaxed_solution.value:
    continue
```

Če je do sedaj dosežena dopustna rešitev boljša od rešitve relaksacije trenutnega programa, potem se trenutni program zavrže, saj ne pripomore k boljši rešitvi (glej razdelek 4.3.1). Nadaljuje se z reševanjem naslednjega programa.

```
if program.is_feasible(relaxed_solution):
    best_feasible_solution = relaxed_solution
    continue
```

Na tej točki se ve, da je rešitev relaksacije trenutnega programa boljša od do sedaj najdene dopustne rešitve. Če rešitev trenutne relaksacije ustreza pogojem celoštevilskosti, je to najboljša do sedaj dosežena dopustna rešitev. Hkrati je to tudi optimalna rešitev trenutnega programa, zato se nadaljuje z reševanjem naslednjega.

```
feasible_solution = heuristic_solve(program)
if feasible_solution.value < best_feasible_solution.value:
    best_feasible_solution = feasible_solution
```

S pomočjo hevrističnih metod se najde čim boljša dopustna rešitev. Če je vrednost rešitve hevristične metode boljša od vrednosti trenutne najboljše dopustne rešitve, je to nova najboljša dopustna rešitev. Vendar se v tem primeru reševanje programa ne prekine, možno je namreč, da se pri nadaljnji obravnavi tega programa najde še boljše rešitve.

```
constraint = heuristic_split(program)
program1, program2 = program.split_on_constraint(constraint)
```

Določi se omejitev za razdelitev trenutnega mešanega celoštevilskega linearnega programa. Le ta se nato razdeli na podprograma upoštevajoč izbrano omejitev. Podprograma sta kopiji prvotnega programa, le da se vsakemu doda še dodatni pogoj. Dodatna pogoja sta komplementarna, podprograma skupaj predstavljata prvotni program. Velja, da je boljša izmed optimalnih rešitev obeh podprogramov ravno rešitev trenutnega mešanega celoštevilskega linearnega programa.

```
        if program1.is_maybe_feasible():
            relaxed_solution1 = simplex(program1.relax())
            task_list.smart_insert((program1, relaxed_solution1))
```

Če je možno, da je prvi podprogram rešljiv, sledi izračun njegove relaksacije in pripadajoče rešitve s pomočjo metode simpleksov. Podprogram s pripadajočo rešitvijo relaksacije se vstavi v podatkovno strukturo še ne obdelanih programov. Vstavljanje se naredi upoštevajoč hevristiko strukture, klasično so na vrhu programi z boljšo rešitvijo relaksacije.

```
        if program2.is_maybe_feasible():
            relaxed_solution2 = simplex(program2.relax())
            task_list.smart_insert((program2, relaxed_solution2))
```

Enak postopek kot za prvi podprogram se izvede tudi za drugi podprogram.

```
    return best_feasible_solution
```

Ko so vsi podprogrami pregledani in je podatkovna struktura še ne razrešenih programov prazna, se vrne najboljša dopustna rešitev najdena tekom algoritma. To je optimalna rešitev danega mešanega celoštevilskega linearnega programa.

Za primere in uporabo tehnik algoritma razveji in obreži glej tudi [13] in [15].

5 Splošni prijemi za izdelavo modela

V tem poglavju je zbranih nekaj dognanj, ki pripomorejo k izdelavi modela v obliki mešanega celoštevilskega linearnega programa. Za več uporabnih prijemov za izdelavo modela glej [11].

5.1 Izbira dogodkov

Z binarnimi spremenljivkami lahko označujemo pojavitev nekega dogodka. Tako je lahko α_A indikator ali se je zgodil dogodek A, kar definiramo kot

$$\alpha_A = \begin{cases} 1, & \text{če se je dogodek } A \text{ zgodil,} \\ 0, & \text{sicer.} \end{cases}$$

Označimo n dogodkov z $A_1, \ldots A_n$. Naj bo $m < n$. Potem lahko pojavitev dogodkov zapišemo na naslednji način:

1. Izmed podanih n dogodkov se zgodi največ m dogodkov
$$\sum_{i \in \{1,\ldots,n\}} \alpha_{A_i} \leq m.$$
2. Izmed podanih n dogodkov se zgodi m dogodkov
$$\sum_{i \in \{1,\ldots,n\}} \alpha_{A_i} = m.$$
3. Izmed podanih n dogodkov se zgodi vsaj m dogodkov
$$\sum_{i \in \{1,\ldots,n\}} \alpha_{A_i} \geq m.$$

Za primer uporabe glej primer 4.15 ter razdelka 6.1.1 in 6.2.6.

Pravilo, da je dogodek A predpogoj za dogodek B, v jeziku mešanega celoštevilskega linearnega programiranja zapišemo kot

$$\alpha_B \leq \alpha_A.$$

Pravilo, kjer so dogodki $A_1, \ldots A_n$ predpogoj za B, za demonstracijo zapišimo na dva načina

1. Prvič kot več enostavnih omejitev
$$\forall i \in \{1, \ldots, n\}, \qquad \alpha_B \leq \alpha_{A_i}.$$
2. Drugič z eno kompleksnejšo omejitvijo
$$n\alpha_B \leq \sum_{i \in \{1,\ldots,n\}} \alpha_{A_i}.$$

Obstaja sicer še veliko drugih možnosti za zapis tega pravila, ki so predvsem kombinacije teh dveh principov.

5.2 Omejitev spremenljivke na intervale

Naj bo $a_1 \leq b_1 < a_2 \leq b_2 < \cdots < a_n \leq b_n$ in

$$x \in [a_1, b_1] \cup [a_2, b_2] \cup [a_3, b_3] \cup \cdots \cup [a_{n-1}, b_{n-1}] \cup [a_n, b_n],$$

torej x element teh disjunktnih intervalov. Naj bo α_i indikator ali je x na i-tem od teh intervalov, kar zapišemo kot

$$\alpha_i = \begin{cases} 1, & \text{če je } x \in [a_i, b_i], \\ 0, & \text{sicer.} \end{cases}$$

Omejitev za x v jeziku mešanega celoštevilskega linearnega programiranja zapišemo kot

$$\begin{gathered} a_1\alpha_1 + a_2\alpha_2 + \cdots + a_n\alpha_n \leq x \leq b_1\alpha_1 + \cdots + b_{n-1}\alpha_{n-1} + a_n\alpha_n, \\ \alpha_1 + \alpha_2 + \cdots + \alpha_n = 1, \end{gathered}$$

oziroma s kompaktno notacijo

$$\begin{gathered} \sum_{i \in \{1,\ldots n\}} a_i\alpha_i \leq x \leq \sum_{i \in \{1,\ldots n\}} b_i\alpha_i, \\ \sum_{i \in \{1,\ldots n\}} b_i\alpha_i = 1. \end{gathered}$$

Poudariti je potrebno, da je možno $a_i = b_i$, pri čemer je i-ti zaprti interval kar število. Velikokrat je smiselno tudi da sta $a_1 = -M_1$, kjer je $-M_1$ najmanjše predstavljivo število v računalniku in $b_n = M_2$, kje je M_2 največje predstavljivo število v računalniku. Oglejmo si še uporabo tega principa na konkretnih primerih.

Primer 5.1. Naj bo x enak 0 ali navzdol omejen z 20 ter navzgor z 30. Matematično to zapišemo kot

$$x \in \{0\} \cup [20, 30].$$

V jeziku mešanega celoštevilskega linearnega programiranja lahko to prevedemo na naslednji pogoj

$$20\alpha_2 \leq x \leq 30\alpha_2,$$

kjer je

$$\alpha_2 = \begin{cases} 1, & \text{če je } x \in [20, 30], \\ 0, & \text{če je } x \text{ enak nič.} \end{cases}$$

Primer 5.2. Naj bo $x < -50$ ali $x > 100$. Potem lahko zapišemo x v jeziku mešanega celoštevilskega linearnega programiranja kot

$$\begin{gathered} -M_1\alpha_1 + 100\alpha_2 \leq x \leq -50\alpha_1 + M_2\alpha_2, \\ \alpha_1 + \alpha_2 = 1. \end{gathered}$$

In v tem primeru velja

$$\alpha_1 = \begin{cases} 1, & \text{če je } x < -50, \\ 0, & \text{sicer} \end{cases} \qquad \text{in} \qquad \alpha_2 = \begin{cases} 1, & \text{če je } x > 100, \\ 0, & \text{sicer.} \end{cases}$$

Primer 5.3. Naj bo $x \in [20, 50] \wedge y \in \mathbb{R}$ ali pa $x \in [40, 70] \wedge y \in [100, 200]$. Potem pogoja za x in y v jeziku mešanega celoštevilskega linearnega programiranja zapišemo kot

$$\begin{aligned} 20\alpha_1 + 40\alpha_2 \leq x \leq 50\alpha_1 + 70\alpha_2, \\ -M_1\alpha_1 + 100\alpha_2 \leq y \leq M_2\alpha_1 + 200\alpha_2, \\ \alpha_1 + \alpha_2 = 1. \end{aligned}$$

5.3 Definiranje namenske funkcije

Oglejmo si še nekaj nestandardnih prijemov za formulacijo namenske funkcije:

1. Kako lahko celoštevilski dogodki doprinesejo k vrednosti namenske funkcije smo videli že pri problemu nahrbtnika v primeru 4.15. Sedaj si oglejmo še, kako bi namenski funkciji prišteli dodatek, če so izbrani vsi dogodki iz neke množice. Naj bo $\mathcal{A} = \{A_1, \ldots A_n\}$ množica dogodkov. Indikator, da se zgodijo vsi dogodki iz množice $\mathcal{A}$, označimo z $\alpha_{\mathcal{A}}$. V jeziku mešanega celoštevilskega linearnega programiranja ga smiselno definiramo z naslednjima neenakostima

$$n\alpha_{\mathcal{A}} \leq \sum_{i \in \{1,\ldots,n\}} \alpha_{A_i} \qquad \text{in} \qquad \sum_{i \in \{1,\ldots,n\}} \alpha_{A_i} \leq n - 1 + \alpha_{\mathcal{A}}.$$

   Namenski funkciji pa lahko tedaj prištejemo dodatek v primeru izbire vseh dogodkov iz $\mathcal{A}$. Kar v jeziku mešanega celoštevilskega programiranja zapišemo kot

$$z = \cdots + p_{\mathcal{A}}\alpha_{\mathcal{A}},$$

   kjer je $p_{\mathcal{A}}$ vrednost dodatka za izbiro vseh dogodkov iz množice $\mathcal{A}$.

2. Oglejmo si še, kako bi lahko v namenski funkciji simulirali kvadratno odvisnost funkcionala z od neke spremenljivke. Naj bo funkcija, katere minimum želimo poiskati,

$$z = \boldsymbol{c}^T\boldsymbol{x} + y^2,$$

   pri omejitvah pa naj y nastopa linearno ter naj dodatno velja $y \in [0, 20]$. Potem lahko dan problem pretvorimo v linearni program, ki je do neke mere približek prvotnega problema. Prepišemo vse omejitve in dodamo pogoje

$$\begin{aligned} \forall i \in \{1, \ldots, 20\}, \qquad 0 \leq y_i \leq 1, \\ y = \sum_{i \in \{1,\ldots,20\}} y_i. \end{aligned}$$

   Za približek funkcionala z s kvadratno odvisnostjo od spremenljivke y pa uvedemo namensko funkcijo

$$z = \boldsymbol{c}^T\boldsymbol{x} + \sum_{i \in \{1,\ldots,20\}} (2i - 1)y_i.$$

Z rešitvijo danega linearnega programa dobimo približek za rešitev prvotnega problema. Kadar je y celo število, se vrednost namenske funkcije namreč povsem ujema, saj velja $\sum_{i=1}^{n}(2i-1)=n^2$. Možnost razčlembe $y=\sum_{i\in\{1,\ldots,20\}} y_i$ je sicer poseben primer, pri katerem je ključnega pomena, da je namenska funkcija glede na y naraščajoča in se minimizira. Tako je zagotovljeno, da je lahko $y_{i+1}>0$, samo če je $y_i=0$. Sicer bi bilo potrebno uvesti indikatorje, ki bi zagotovili, da so y_i neničelni v pravilnem vrstnem redu. Zavedati se je potrebno, da je linearna aproksimacija za odvedljive funkcije na krajših odsekih dokaj dober približek dejanski funkciji.

3. Naj bo $x\in\mathbb{R}$ in naj bo cena za $x\geq 0$ enaka p_x^+, cena za $x<0$ pa enaka p_x^-. Strošek za x tako lahko zapišemo kot

$$c_x=\begin{cases} xp_x^+, & x\geq 0,\\ xp_x^-, & x<0.\end{cases}$$

Sedaj pišimo $x=x^+-x^-$, kjer sta $x^+,x^-\geq 0$. Strošek za x pa je enak

$$c_x=x^+p_x^+-x^-p_x^-.$$

Ker je delitev na x^+ (pozitivni) in x^- (negativni) del spremenljivke x dobro definirana le, če je največ ena od spremenljivk neničelna, je potrebno definirati $p_x^+>p_x^->0$. V tem primeru se pri minimiziranju namenske funkcije

$$z=\cdots+x^+p_x^+-x^-p_x^-+\cdots$$

avtomatično ena od spremenljivk x^+ ali x^- nastavi na nič.

4. Naj bo $a_1\leq b_1<a_2\leq b_2<\cdots<a_n\leq b_n$ in

$$x\in[a_1,b_1]\cup[a_2,b_2]\cup[a_3,b_3]\cup\cdots\cup[a_{n-1},b_{n-1}]\cup[a_n,b_n].$$

Dodatno naj bo cena za $x\in[a_i,b_i]$ enaka p_i. Spremenljivko x razdelimo na

$$x=\sum_{i\in\{1,\ldots n\}} x_i$$

in naj velja

$$\forall i\in\{1,\ldots n\},\qquad a_i\alpha_i\leq x_i\leq b_i\alpha_i\quad\wedge\quad \alpha_i\in\{0,1\}$$

ter

$$\sum_{i\in\{1,\ldots n\}}\alpha_i=1.$$

Potem lahko strošek spremenljivke x v namenski funkciji zapišemo kot

$$z=\cdots+\sum_{i\in\{1,\ldots,n\}} x_ip_i+\cdots,$$

s čimer smo dosegli, da ima x za vsako pojavitev na intervalu različno ceno. S tem smo naredili posplošitev točke 3.

S primeri smo pokazali, da je kljub togi formulaciji linearno programiranje močno orodje, ki se ga lahko z dodajanjem spremenljivk, računske zahtevnosti in zahtevnosti za razumevanje uporabi za mnogo več problemov, kot je razvidno na prvi pogled.

6 Matematični model problema razporejanja

Problem razporejanja smo formalno že definirali v poglavju 3. Navedli smo tudi nekaj uporabnih prijemov za izdelavo modela (glej poglavje 5). V tem poglavju naredimo še nakaj predpriprave, nato pa v celoti opišemo in zapišemo problem razporejanja prilagodljivih ponudb kot model v obliki mešanega celošteviskega linearnega programa. Optimizacijski kriterij bo minimiziranje namenske funkcije, to je izhodnih stroškov. Pogoji izhajajo iz omejitev prilagodljivih ponudb in upoštevanja definicije končnega energijskega odstopanja.

6.1 Predpriprava za izdelavo modela

Vpeljimo nekaj novih oznak za energijsko odstopanje in prilagodljivo ponudbo f, ki jih bomo uporabljali tekom tega poglavja.

6.1.1 Definicija dogodkov

Z $\alpha_{f,i}$ označimo ali se je prilagodljiva ponudba f začela v časovnem intervalu i. Za časovni interval iz možnih začetnih časov $i \in \mathcal{I}_f$ velja $\alpha_{f,i} = 1$, če se je prilagodljiva ponudba f začela v časovnem intervalu i, in $\alpha_{f,i} = 0$, če se prilagodljiva ponudba ni začela v tem časovnem intervalu (ni bila uporabljena ali pa se je začela na drugem intervalu). S formulo

$$\forall i \in \mathcal{I}_f, \qquad \alpha_{f,i} = \begin{cases} 1, & f \text{ se začne v času } i, \\ 0, & \text{sicer.} \end{cases}$$

Zaradi lažjega zapisa enačb za $i \notin \mathcal{I}_f$ dodatno definiramo $\alpha_{f,i} = 0$ in se zavedamo, da tega člena v modelu končnega programa ne potrebujemo in ga lahko izpustimo. Tak zapis nam pride prav zaradi enostavnejšega zapisa vsot v matematičnem modelu.

Iz definicije $\alpha_{f,i}$ lahko izpeljemo naslednje indikatorje:

1. Za začetek prilagodljive ponudbe f v času i

$$\alpha_{f,i} = \begin{cases} 1, & f \text{ se začne v času } i, \\ 0, & \text{sicer.} \end{cases}$$

2. Za uporabo prilagodljive ponudbe f v času t

$$\sum_{j \in \mathcal{L}_f} \alpha_{f,t-j} = \begin{cases} 1, & f \text{ uporabljena v času } t, \\ 0, & \text{sicer.} \end{cases}$$

3. Za uporabo prilagodljive ponudbe f kadarkoli

$$\sum_{i \in \mathcal{I}_f} \alpha_{f,i} = \begin{cases} 1, & f \text{ uporabljena,} \\ 0, & \text{sicer.} \end{cases}$$

Opomba 6.1. Pri drugi točki je lahko indeks v členu $\alpha_{f,t-j}$ izven možnih začetnih časov, to je $t - j \notin \mathcal{I}_f$, tedaj ta člen v celoti izpustimo iz vsote.

Ostalih dogodkov pri izdelavi modela ne bomo potrebovali. Za problem prilagodljivih ponudb, kot smo ga definirali v poglavju 3, ostali dogodki niso pomembni. Z dodatnimi dogodki bi sicer med drugim lahko obravnavali tudi različne dolžine prilagodljivih ponudb. V končnem mešanem celoštevilskem linearnem programu bodo indikatorji $\alpha_{f,i}$ za začetek prilagodljivih ponudb f v času t edine celoštevilske spremenljivke in še te binarne.

6.1.2 Obravnava energije odstopanja

Za časovni interval $t \in \mathcal{N}$ velja, da je energija končnega energijskega odstopanja enaka izhodiščnemu odstopanju z dodanim doprinosom energije uporabljenih prilagodljivih ponudb. To lahko tudi izpeljemo, oziroma z enačbo zapišemo kot

$$E_t^{\mathrm{I}} = E_t^{\mathrm{Ist}} + \sum_{f \in \mathcal{F}} \widetilde{E}_{f,t} = E_t^{\mathrm{I}^*} + \sum_{f \in \mathcal{F}} E_{f,t}^{D} + \sum_{f \in \mathcal{F}} \widetilde{E}_{f,t} = E_t^{\mathrm{I}^*} + \sum_{f \in \mathcal{F}} E_{f,t}.$$

S tem smo iz definicije končnega energijskega odstopanja odstranili člen spremembe energije prilagodljive ponudbe glede na privzeti razpored in ga nadomestili z energijo prilagodljive ponudbe.

Cena energijskega odstopanja je odvisna od predznaka energije odstopanja, zato je smiselno, da ločimo energijo odstopanja posameznega časovnega intervala t glede na pozitiven in negativen predznak. Uvedemo spremenljivki, za kateri želimo da velja

$$E_t^{\mathrm{I}+} = \begin{cases} E_t^{\mathrm{I}}, & E_t^{\mathrm{I}} \geq 0, \\ 0, & E_t^{\mathrm{I}} < 0 \end{cases} \qquad \text{in} \qquad E_t^{\mathrm{I}-} = \begin{cases} 0, & E_t^{\mathrm{I}} \geq 0, \\ E_t^{\mathrm{I}}, & E_t^{\mathrm{I}} < 0, \end{cases}$$

kjer je za energijo $E_t^{\mathrm{I}+}$ cena $p_t^{\mathrm{I}+}$ in za energijo $E_t^{\mathrm{I}-}$ cena $p_t^{\mathrm{I}-}$. To po principu iz točke 3 razdelka 5.3 dosežemo z uvedbo

$$E_t^{\mathrm{I}} = E_t^{\mathrm{I}+} + E_t^{\mathrm{I}-},$$

kjer je $E_t^{\mathrm{I}+}$ pozitivna komponenta energije odstopanja in $E_t^{\mathrm{I}-}$ negativna komponenta energije odstopanja. V tem primeru je pomembno, da je največ ena izmed spremenljivk $E_t^{\mathrm{I}+}$ in $E_t^{\mathrm{I}-}$ neničelna. To lahko zagotovimo tako, da pri optimizacijskem kriteriju spremenljivki nastopata tako, da ni smiselno, da bi bili obe neničelni. Zato mora biti namenska funkcija oblike

$$z = \ldots + p_t^{\mathrm{I}+} E_t^{\mathrm{I}+} + p_t^{\mathrm{I}-} E_t^{\mathrm{I}-} + \ldots,$$

kjer velja

$$p_t^{\mathrm{I}+} > p_t^{\mathrm{I}-}.$$

V praksi to pomeni, da je cena dokupa manjkajoče energije $p_t^{\mathrm{I}-}$ za operaterja dražja od cene $p_t^{\mathrm{I}+}$, po kateri lahko odvečno energijo prodaja. Pogoj je v realnem svetu zelo smiseln, sicer ima lahko operater na časovnem intervalu t neomejen zaslužek. V primeru da pogoj $p_t^{I+} > p_t^{I-}$ ni izpolnjen, algoritem pri uporabi enakega modela ne konvergira, saj bi bila vrednost namenske funkcije pri teh pogojih navzdol neomejena. Za možnost uporabe splošnejših cen ločenih tudi na več intervalov glej razdelek 5.3 točko 4.

6.1.3 Obravnava energije prilagodljivih ponudb

Strošek prilagodljive ponudbe je enak

$$c_{f,t} = \widetilde{c}_{f,t} + c_{f,t}^D,$$

kjer je $c_{f,t}^D$ strošek privzetega urnika razporeda in $\widetilde{c}_{f,t}$ strošek spremembe od le tega. Posebej velja poudariti, da je $c_{f,t}$ različen od nič le na intervalih, kjer smo prilagodljivo ponudbo uporabili. Torej lahko zaradi večje preglednosti strošek prilagodljive ponudbe f na časovnem intervalu t zapišemo tudi kot

$$c_{f,t} = \begin{cases} \widetilde{c}_{f,t} + c_{f,t}^D, & f \text{ uporabljena v času } t, \\ 0, & \text{sicer.} \end{cases}$$

Podobno lahko zapišemo tudi energijo prilagodljive ponudbe f na časovnem intervalu t kot

$$E_{f,t} = \begin{cases} \widetilde{E}_{f,t} + E_{f,t}^D, & f \text{ uporabljena v času } t, \\ 0, & \text{sicer.} \end{cases}$$

Za zapis stroška v kontekstu mešanega celoštevilskega linearnega programa je potrebno ločiti energijo na pozitivno in negativno odstopanje na intervalih uporabe. Definirajmo pozitivno in negativno spremembo prilagodljive ponudbe f na časovnem intervalu t, kjer je bila prilagodljiva ponudba uporabljena. Definirajmo

$$E_{f,t}^+ = \begin{cases} \widetilde{E}_{f,t}, & \widetilde{E}_{f,t} \geq 0 \text{ in } f \text{ uporabljena v času } t, \\ 0, & \text{sicer} \end{cases}$$

in

$$E_{f,t}^- = \begin{cases} \widetilde{E}_{f,t}, & \widetilde{E}_{f,t} \leq 0 \text{ in } f \text{ uporabljena v času } t, \\ 0, & \text{sicer.} \end{cases}$$

S tem dobimo ravno razdelitev spremembe energije prilagodljive ponudbe f v času t v primeru njene uporabe

$$E_{f,t}^+ + E_{f,t}^- = \begin{cases} \widetilde{E}_{f,t}, & f \text{ uporabljena v času } t, \\ 0, & \text{sicer.} \end{cases}$$

Velja torej

$$E_{f,t} = \begin{cases} E_{f,t}^+ + E_{f,t}^- + E_{f,t}^D, & f \text{ uporabljena v času } t, \\ 0, & \text{sicer.} \end{cases}$$

To lahko zapišemo v obliki primerni za mešani celoštevilski linearni program kot

$$E_{f,t} = E^+_{f,t} + E^-_{f,t} + E^D_{f,t} \sum_{j \in \mathcal{L}_f} \alpha_{f,t-j},$$

kjer je

$$E^D_{f,t} \sum_{j \in \mathcal{L}_f} \alpha_{f,t-j} = \begin{cases} E^D_{f,t}, & f \text{ uporabljena v času } t, \\ 0, & \text{sicer.} \end{cases}$$

Opomba 6.2. Dodatno moramo biti zopet pozorni na izraz

$$E^+_{f,t} + E^-_{f,t} = \begin{cases} \widetilde{E}_{f,t}, & \text{če je bila } f \text{ uporabljena v času } t, \\ 0, & \text{sicer,} \end{cases}$$

kjer je največ eden izmed členov $E^+_{f,t}$ in $E^-_{f,t}$ neničelen. Princip, kako to dosežemo, smo že srečali v 6.1.2. Namenska funkcija mora biti oblike

$$z = \ldots + p^+_{f,t} E^+_{f,t} + p^-_{f,t} E^-_{f,t} \ldots$$

in veljati mora

$$p^+_{f,t} > p^-_{f,t},$$

kar zagotavlja, da je nakupna cena energije za distributerja na istem intervalu za posamezno prilagodljivo ponudbo dražja od prodajne. Slednje zahteva določitev obeh cen tudi kadar to pomensko ni smiselno, torej kadar se lahko energijo le porablja ali le proizvaja in ni privzetega razporeda. Za ostale prolagodljive ponube pa bi to zahtevo lahko odpravili ter izločili pripadajoče člene iz enačb modela.

6.2 Pretvorba definicije problema v matematičen model

V tem razdelku bomo naredili pretvorbo matematične definicije problema v matematičen model mešanega celoštevilskega linearnega programa. V nadaljnjem bomo pogosto uporabljali oznako za energijo

$$E_{f,t} = E^+_{f,t} + E^-_{f,t} + E^D_{f,t} \sum_{j \in \mathcal{L}_f} \alpha_{f,t-j},$$

ki služi samo kot okrajšava za zapis celotnega izraza. Sicer bi jo lahko vpeljali tudi kot svojo spremenljivko, vendar tega v modelu nismo naredili. Za prilagodljivo ponudbo f in čas izven možnih intervalov uporabe $t \notin \mathcal{T}_f$, člene $E_{f,t}$ pri implementaciji izpustimo.

6.2.1 Namenska funkcija

Operaterjev strošek $\widetilde{c}$ brez težav izračunamo iz izhodnega stroška c (glej razdelek 3.2). Ker se stroška razlikujeta za konstanto, lahko v modelu uporabimo izhodni strošek, ki je zaradi vključitve privzetih urnikov izvajanja enostavnejši za modeliranje. V mešanem celoštevilskem linearnem programu za namensko funkcijo uporabimo izhodni strošek izvajanja scenarija

$$c = c_I + c_{\mathcal{F}},$$

kjer je c_I strošek energijskega odstopanja in $c_{\mathcal{F}}$ strošek uporabe prilagodljivih ponudb. Strošek uporabe energijskega odstopanja je enak vsoti stroškov za vsak interval posebej

$$c_I = \sum_{t \in \mathcal{N}} c_{I,t} = \sum_{t \in \mathcal{N}} E_t^{\mathrm{I}} p_t^{\mathrm{I}} = \sum_{t \in \mathcal{N}} \left(E_t^{\mathrm{I}+} p_t^{\mathrm{I}+} + E_t^{\mathrm{I}-} p_t^{\mathrm{I}-} \right).$$

Strošek uporabe prilagodljivih ponudb pa definiramo kot vsoto stroškov uporabe vseh prilagodljivih ponudb

$$c_{\mathcal{F}} = \sum_{f \in \mathcal{F}} c_f = \sum_{f \in \mathcal{F}} \sum_{t \in \mathcal{T}_f} c_{f,t}.$$

Strošek c_f posamezne prilagodljive ponudbe f je enak vsoti stroškov prilagodljive ponudbe po vseh časovnih intervalih. Strošek prilagodljive ponudbe f v časovnem intervalu t pa je enak

$$c_{f,t} = \begin{cases} E_{f,t}^{+} p_{f,t}^{+} + E_{f,t}^{-} p_{f,t}^{-} + c_{f,t}^{D}, & \text{če je } f \text{ na intervalu } t \text{ uporabljena}, \\ 0, & \text{sicer}, \end{cases}$$

kjer je $c_{f,t}^{D}$ strošek privzetega urnika izvajanja, $E_{f,t}^{+} p_{f,t}^{+}$ strošek v primeru pozitivne razlike energije od privzetega urnika izvajanja in $E_{f,t}^{-} p_{f,t}^{-}$ strošek v primeru negativne razlike energije od privzetega urnika izvajanja. V jeziku mešanega celoštevilskega linearnega programiranja strošek prilagodljive ponudbe f na časovnem intervalu t zapišemo kot

$$c_{f,t} = E_{f,t}^{+} p_{f,t}^{+} + E_{f,t}^{-} p_{f,t}^{-} + c_{f,t}^{D} \sum_{j \in \mathcal{L}_f} \alpha_{f,t-j}.$$

V enačbi je $\sum_{j \in \mathcal{L}_f} \alpha_{f,t-j}$ indikator ali je bila prilagodljiva ponudba f v času t uporabljena, posledično je

$$c_{f,t}^{D} \sum_{j \in \mathcal{L}_f} \alpha_{f,t-j} = \begin{cases} c_{f,t}^{D}, & \text{če je } f \text{ na intervalu } t \text{ uporabljena}, \\ 0, & \text{sicer}. \end{cases}$$

Opomba 6.3. Glavna prednost stroška uporabe prilagodljive ponudbe c_f pred stroškom spremembe urnika izvajanja $\widetilde{c}_f$ je, da je strošek uporabe prilagodljive ponudbe neničelen le na intervalih dejanskega izvajanja prilagodljive ponudbe, medtem ko je strošek spremembe urnika izvajanja $\widetilde{c}_f$ lahko prisoten tako na območju privzetega urnika izvajanja kot tudi na območju dejanskega izvajanja.

Namenska funkcija je enaka

$$c = \sum_{t \in \mathcal{N}} \left(E_t^{\mathrm{I}+} p_t^{\mathrm{I}+} + E_t^{\mathrm{I}-} p_t^{\mathrm{I}-} + \sum_{f \in \mathcal{F}} \underbrace{\left(E_{f,t}^{+} p_{f,t}^{+} + E_{f,t}^{-} p_{f,t}^{-} + c_{f,t}^{D} \sum_{j \in \mathcal{L}_f} \alpha_{f,t-j} \right)}_{c_{f,t}} \right),$$

kjer v implementaciji izpustimo tiste sumande $c_{f,t}$, kjer indeks t ni možen časovni interval postavitve prilagodljive ponudbe, to je $t \notin \mathcal{T}_f$.

6.2.2 Omejitev energijskega odstopanja

Oglejmo si omejitev, s katero zahtevamo, da je končno energijsko odstopanje ustrezno nastavljeno. Za vsak časovni interval t je končna energija odstopanja enaka vsoti začetne energije odstopanja in spremembi energij prilagodljivih ponudb od privzetega razporeda

$$E_t^{\mathrm{I}} = E_t^{\mathrm{Ist}} + \sum_{f \in \mathcal{F}} \widetilde{E}_{f,t}.$$

Če v zgornji enačbi členu E_t^{Ist} odštejemo energijo privzetega razporeda in členu $\sum_{f \in \mathcal{F}} \widetilde{E}_{f,t}$ prištejemo energijo privzetega razporeda, dobimo

$$E_t^{\mathrm{I}} = \Big(E_t^{\mathrm{Ist}} - \sum_{f \in \mathcal{F}} E_{f,t}^{D}\Big) + \Big(\sum_{f \in \mathcal{F}} E_{f,t}^{D} + \sum_{f \in \mathcal{F}} \widetilde{E}_{f,t}\Big),$$

kar je ravno izhodiščna energija odstopanja in vsota energij prilagodljivih ponudb

$$E_t^{\mathrm{I}} = E_t^{\mathrm{I}^*} + \sum_{f \in \mathcal{F}} E_{f,t}.$$

Ko na pozitivno in negativno komponento razpišemo še končno energijo odstopanja, dobimo enačbo

$$E_t^{\mathrm{I}+} + E_t^{\mathrm{I}-} = E_t^{\mathrm{I}^*} + \sum_{f \in \mathcal{F}} E_{f,t},$$

ki zagotavlja, da je zadoščeno definiciji končnega energijskega odstopanja. To je tudi edina omejitev, ki povezuje energije več prilagodljivih ponudb v istih enačbah. Za prilagodljivo ponudbo f in čas izven možnih intervalov uporabe $t \notin \mathcal{T}_f$ člene $E_{f,t}$ v implementaciji izpustimo.

6.2.3 Omejitev energije na relativnih intervalih

Omejitev prilagodljive ponudbe na relativnih intervalih zagotavlja, da je energija prilagodljive ponudbe na posameznem relativnem intervalu znotraj prilagodljive ponudbe med najmanjšo in največjo dovoljeno energijo.

Naj bo prilagodljiva ponudba f v časovnem intervalu t uporabljena in naj bo njen začetek izvajanja v času $i \leq t$. Tedaj je $j = t - i$ j-ti relativni interval prilagodljive ponudbe. Tako za energijo prilagodljive ponudbe v časovnem intervalu t velja enačba

$$E_{f,j}^{\min} \leq E_{f,t} \leq E_{f,j}^{\max},$$

kjer je $E_{f,j}^{\min}$ najmanjša dovoljena energija j-tega relativnega časovnega intervala prilagodljive ponudbe in $E_{f,j}^{\max}$ največja dovoljena energija j-tega relativnega časovnega intervala prilagodljive ponudbe. Obe sta za posamezen relativni časovni interval določeni glede na začetni čas izvajanja prilagodljive ponudbe (glej razdelek 3.3.1), medtem ko je energija prilagodljive ponudbe $E_{f,t}$ določena na časovnem intervalu t danega scenarija.

V primeru, da se prilagodljiva ponudba f na danem časovnem intervalu t ne izvaja, velja enačba

$$E_{f,t} = 0,$$

kar lahko zapišemo z neenačbama kot

$$0 \leq E_{f,t} \leq 0.$$

Uporabimo princip iz primera 5.1 v razdelku 5.2, da omejimo energijo za oba primera (prijem ali neuporaba). Za prilagodljivo ponudbo f in časovni interval t uporabimo indikator $\sum_{j\in\mathcal{L}_f} \alpha_{f,t-j}$, s katerim določimo ali je bila prilagodljiva ponudba f v danem časovnem intervalu t uporabljena. Dobimo naslednji pogoj

$$\sum_{j\in\mathcal{L}_f} \alpha_{f,t-j} E^{\min}_{f,j} \leq E_{f,t} \leq \sum_{j\in\mathcal{L}_f} \alpha_{f,t-j} E^{\max}_{f,j}.$$

V zgornji enačbi je s $t-j$ obeležen začetek prilagodljive ponudbe. Velja omeniti, da je indeks $t-j$ pri $\alpha_{f,t-j}$ lahko izven območja možnih začetkov prilagodljive ponudbe, tedaj pripadajoč člen pri implementaciji modela izpustimo.

6.2.4 Omejitev skupne energije

Omejitev skupne energije določa meje za skupno energijo prilagodljive ponudbe v podanem scenariju. Z njo je zagotovljeno, da prilagodljiva ponudba na danem scenariju ne prejme ali odda preveč energije. Vsota energij prilagodljive ponudbe po intervalih je navzdol omejena z najmanjšo dovoljeno skupno energijo in navzgor z največjo dovoljeno skupno energijo.

V primeru uporabe prilagodljive ponudbe f nam to poda neenačbi

$$E^{\mathrm{T}\min}_f \leq \sum_{t\in\mathcal{T}_f} E_{f,t} \leq E^{\mathrm{T}\max}_f,$$

kjer je $\sum_{t\in\mathcal{T}_f} E_{f,t}$ vsota energij prilagodljive ponudbe po vseh časovnih intervalih, $E^{\mathrm{T}\min}_f$ spodnja meja za skupno energijo in $E^{\mathrm{T}\max}_f$ zgornja meja za skupno energijo. V primeru, da prilagodljiva ponudba ni uporabljena, mora biti skupna energija enaka 0, kar z enačbo zapišemo kot

$$E_f = \sum_{t\in\mathcal{T}_f} E_{f,t} = 0,$$

oziroma z neenačbama kot

$$0 \leq \sum_{t\in\mathcal{T}_f} E_{f,t} \leq 0.$$

Za združitev pogojev za primer uporabe in neuporabe prilagodljive ponudbe f moramo z ustreznim indikatorjem pomnožiti člena $E^{\mathrm{T}\min}_f$ in $E^{\mathrm{T}\max}_f$ prvotne neenačbe (glej primer 5.1). Uporabimo indikator $\sum_{i\in\mathcal{I}_f} \alpha_{f,i}$ s katerim preverimo ali je bila prilagodljiva ponudba v danem scenariju kadarkoli uporabljena. Indikator je enak 1 v primeru uporabe prilagodljive ponudbe, sicer pa je enak 0. Tako lahko zapišemo omejitev skupne energije prilagodljive ponudbe f z neenačbama

$$E_f^{\mathrm{Tmin}} \sum_{i \in \mathcal{I}_f} \alpha_{f,i} \leq \sum_{t \in \mathcal{T}_f} E_{f,t} \leq E_f^{\mathrm{Tmax}} \sum_{i \in \mathcal{I}_f} \alpha_{f,i}.$$

Na tem mestu velja poudariti, da je za izpolnjivost omejitev danega modela ključnega pomena, da se v primeru neuporabe zgornja in spodnja meja nastavita na nič. Sicer to, da se energije vseh intervalov nastavijo na nič, dosežemo že z omejitvijo energije na relativnih intervalih (glej razdelek 6.2.3), vendar je lahko problem tudi, da sta E_f^{Tmin} in E_f^{Tmax} istega predznaka, s čimer bi v primeru neuporabe prilagodljive ponudbe dobili neveljavni pogoj

$$E_f^{\mathrm{Tmin}} \leq 0 \leq E_f^{\mathrm{Tmax}}.$$

Zato je zares pomembno, da tudi pri tej omejitvi vpeljemo indikator uporabe.

6.2.5 Omejitev kumulativne energije

Za prilagodljive ponudbe, katerih vir so baterije, moramo upoštevati še omejitev kumulativne energije. Omejitev zagotavlja upoštevanje kapacitivnosti baterije in preprečuje prekomerno napolnitev ali izpraznitev. Trenutna energija je vsota energij po vseh časovnih intervalih do izbranega časovnega intervala. Kumulativna omejitev določa, da mora biti trenutna energija prilagodljive ponudbe za vsak izbran časovni interval $t^C \in \mathcal{T}_f$ znotraj določenih mej.

Za uporabljeno prilagodljivo ponudbo f tako za vsak časovni interval t^C velja, da je vsota začetne energije in vseh energij intervalov prilagodljive ponudbe do tega časovnega intervala večja od spodnje in manjša od zgornje meje kapacitete baterij, kar z neenačbo zapišemo kot

$$E_f^{\mathrm{Cmin}} \leq E_f^{\mathrm{Cst}} + \sum_{t \leq t^C} E_{f,t} \leq E_f^{\mathrm{Cmax}},$$

kjer je E_f^{Cst} začetna energija baterije, $\sum_{t \leq t^C} E_{f,t}$ vsota energij do danega trenutka in E_f^{Cmin} minimalna ter E_f^{Cmax} maksimalna meja.

Enačba velja le, če je prilagodljiva ponudba uporabljena, sicer mora biti vsota energij prilagodljive ponudbe po intervalih do danega trenutka enaka 0, kar z neenačbami zapišemo kot

$$0 \leq \sum_{t \leq t^C} E_{f,t} \leq 0.$$

Pri zgornjem pogoju začetna energija ne nastopa v enačbi, saj se le ta pri neuporabi baterije ne spremeni. Uporabimo enak prijem kot za omejitev skupne energije v razdelku 6.2.4. Uvedemo indikator $\sum_{i \in \mathcal{I}_f} \alpha_{f,i}$, s katerim preverimo uporabo prilagodljive ponudbe v danem scenariju. Za prilagodljivo ponudbo f, za katero še ne vemo ali bo uporabljena, tako zapišemo omejitev kumulativne energije razdeljeno na pogoj za spodnjo in zgornjo mejo, kot bo zapisano v nadaljevanju. Za spodnjo mejo veljajo neenačbe

$$\forall t^C \in \mathcal{T}_f, \qquad \left(E_f^{\mathrm{Cmin}} - E_f^{\mathrm{Cst}}\right) \sum_{i \in \mathcal{I}_f} \alpha_{f,i} \leq \sum_{t \leq t^C} E_{f,t}$$

in podobno za zgornjo mejo

$$\forall t^C \in \mathcal{T}_f, \qquad \sum_{t \leq t^C} E_{f,t} \leq \left(E_f^{\mathrm{Cmax}} - E_f^{\mathrm{Cst}}\right) \sum_{i \in \mathcal{I}_f} \alpha_{f,i}.$$

Omejitev smo razdelili na pogoja za spodnjo in zgornjo mejo zaradi preglednosti.

6.2.6 Omejitev uporabe

Vsaka prilagodljiva ponudba se lahko uporabi največ enkrat. To pomeni, da se vsaka prilagodljiva ponudba lahko začne največ enkrat. Začetek prilagodljive ponudbe f v časovnem intervalu i obeležimo z $\alpha_{f,i} = 1$. Če je bila prilagodljiva ponudba f v danem scenariju uporabljena natanko enkrat, je vsota $\sum_{i \in \mathcal{I}_f} \alpha_{f,i}$ enaka 1, saj se je v tem primeru prilagodljiva ponudba tudi natančno enkrat začela. V primeru neuporabe, je ta vsota enaka 0.

Prilagodljiva ponudba se lahko začne največ enkrat, kar z neenačbo zapišemo kot

$$\sum_{i \in \mathcal{I}_f} \alpha_{f,i} \leq 1.$$

V primeru, ko je prilagodljivo ponudbo f obvezno uporabiti, pa mora biti vsaj eden od $\alpha_{f,i}$ različen od 0 in mora biti zadoščeno enakosti

$$\sum_{i \in \mathcal{I}_f} \alpha_{f,i} = 1.$$

S tema pogojema zagotovimo smiselnost ostalih enačb. Prilagodljiva ponudba se lahko uporabi največ enkrat, v primeru obveznega razporeda pa natanko enkrat.

6.3 Zgoščen zapis matematičnega modela

Minimiziraj namensko funkcijo

$$c = \sum_{t \in \mathcal{N}} \left(E_t^{\mathrm{I}+} p_t^{\mathrm{I}+} + E_t^{\mathrm{I}-} p_t^{\mathrm{I}-} + \sum_{f \in \mathcal{F}} \left(E_{f,t}^{+} p_{f,t}^{+} + E_{f,t}^{-} p_{f,t}^{-} + c_{f,t}^{D} \sum_{j \in \mathcal{L}_f} \alpha_{f,t-j} \right) \right),$$

pri sledečih pogojih.
Omejitev energijskega odstopanja

$$\forall t \in \mathcal{N}, \qquad E_t^{\mathrm{I}+} + E_t^{\mathrm{I}-} = E_t^{\mathrm{I}*} + \sum_{f \in \mathcal{F}} E_{f,t}.$$

Omejitve prilagodljivih ponudb:

1. Omejitev energije prilagodljive ponudbe f na relativnih intervalih

$$\forall t \in \mathcal{T}_f, \qquad \sum_{j \in \mathcal{L}_f} \alpha_{f,t-j} E_{f,j}^{\min} \leq E_{f,t} \leq \sum_{j \in \mathcal{L}_f} \alpha_{f,t-j} E_{f,j}^{\max}.$$

2. Omejitev skupne energije prilagodljive ponudbe f

$$E_f^{\mathrm{Tmin}} \sum_{i \in \mathcal{I}_f} \alpha_{f,i} \leq \sum_{t \in \mathcal{T}_f} E_{f,t} \leq E_f^{\mathrm{Tmax}} \sum_{i \in \mathcal{I}_f} \alpha_{f,i}.$$

3. Omejitev kumulativne energije v primeru, da je f baterija,

$$\forall t^C \in \mathcal{T}_f, \qquad \left(E_f^{\mathrm{Cmin}} - E_f^{\mathrm{Cst}} \right) \sum_{i \in \mathcal{I}_f} \alpha_{f,i} \leq \sum_{t \leq t^C} E_{f,t} \leq \left(E_f^{\mathrm{Cmax}} - E_f^{\mathrm{Cst}} \right) \sum_{i \in \mathcal{I}_f} \alpha_{f,i}.$$

4. Omejitev uporabe prilagodljive ponudbe f, ločeno na primera
   - če f ni nujno uporabiti

$$0 \leq \sum_{i \in \mathcal{I}_f} \alpha_{f,i} \leq 1,$$

   - če mora biti f obvezno uporabljena

$$\sum_{i \in \mathcal{I}_f} \alpha_{f,i} = 1.$$

Uporabljeni simboli:

1. Okrajšava za energijo prilagodljive ponudbe f v času t je

$$E_{f,t} = E_{f,t}^{+} + E_{f,t}^{-} + E_{f,t}^{D} \sum_{j \in \mathcal{L}_f} \alpha_{f,t-j}.$$

2. Realne spremenljivke $E_t^{\mathrm{I}+} \geq 0, E_t^{\mathrm{I}-} \leq 0$ in $E_{f,t}^{+} \geq 0, E_{f,t}^{-} \leq 0$.
3. Binarne spremenljivke $\alpha_{f,i} \in \{0,1\}$.
4. Konstante $E_t^{\mathrm{I}*}$, $p_t^{\mathrm{I}+} > p_t^{\mathrm{I}-}$ in $E_{f,j}^{\min}, E_{f,j}^{\max}$, $E_f^{\mathrm{Tmin}}, E_f^{\mathrm{Tmax}}$, $E_f^{\mathrm{Cst}}, E_f^{\mathrm{Cmin}}, E_f^{\mathrm{Cmax}}$, $E_{f,t}^{D}$, $c_{f,t}^{D}$, $p_{f,t}^{+} > p_{f,t}^{-}$.

Opomba 6.4. Omejitvi za skupno in kumulativno energijo za posamezno prilagodljivo ponudbo nista nujno podani.

7 Implementacija modela

Za reševanje realnih problemov je pomembna dobra implementacija matematičnega modela. V našem primeru je bila implementacija bistveno olajšana zaradi uporabe Googlove knjižnice OR-Tools za matematično optimizacijo [1]. Poleg implementacije modela je bilo potrebno implementirati še strežniško komunikacijo, predpripravo podatkov, ustrezne podatkovne strukture in orodja za analizo sistema. Celotna implementacija presega okvire magistrskega dela, zato se bomo v nadaljevanju omejili le na oris implementacije matematičnega modela.

Za reševanje mešanih celoštevilskih linearnih programov obstajajo različne programske rešitve. V grobem je reševanje možno v okviru programskih paketov ali pa s pomočjo knjižnice v enem izmed popularnih programskih jezikov. Nivoji prijaznosti uporabniškega vmesnika se pri tem bistveno razlikujejo. Tako so za enostavnejše primere na voljo rešitve z grafičnim uporabniškim vmesnikom ali konzolo. Za zahtevnejše primere pa navadno uporabimo nek programski jezik in zanj pripravljen vmesnik do implementacije reševalnika v obliki knjižnice. Mnogi reševalniki omogočajo tudi podajanje problemov v tekstovnih oblikah.

V nadaljevanju si bomo ogledali paket Google OR-Tools ter navedli nekaj tehničnih podatkov o reševalnikih COIN-OR Branch and Cut in Gurobi. Navedli bomo tudi primer uporabe Google OR-Tools ter si ogledali enega izmed standardnih formatov za zapis modela.

7.1 Reševalnik COIN-OR Branch and Cut

Reševalnik COIN-OR Branch and Cut (CBC) je prostodostopni odprtno-kodni reševalnik za mešano celoštevilsko linearno programiranje. Napisan je v programskem jeziku `C++` in je na voljo kot samostojni program ali kot programska knjižnica. Reševalnik med drugim implementira tudi determinističen algoritem razveji in obreži (glej 4.4 in 4.4.2). V magistrskem delu smo uporabljali CBC verzijo 2.9.9 kot knjižnjico, ki jo kliče Google OR-tools. Za več informacij glej [8].

7.2 Reševalnik Gurobi

Komercialni reševalnik Gurobi je eden vodilnih reševalnikov za mešano celoštevilsko linearno programiranje. Program je plačljiv, možna pa je tudi brezplačna uporaba z akademsko licenco ali časovno omejenim testnim obdobjem. Na voljo je kot samostojni program ali z vmesniki v različnih programskih jezikih. V magistrskem delu smo uporabljali Gurobi verzijo 8.01. Za več informacij glej [10].

7.3 Paket Google OR-Tools

Paket oziroma knjižnica Google OR-Tools omogoča implementacijo modela v različnih programskih jezikih in uporabo različnih reševalnikov. Model smo implementirali v jeziku Java in uporabili prosto dostopen reševalnik CBC. V magistrskem delu smo uporabljali Google OR-Tools verzijo 6.7.2. Za več informacij glej [1].

S pomočjo Google OR-Tools lahko probleme izvozimo tudi v tekstovnem formatu na primer v CPLEX LP formatu [2].

Primer 7.1. Oglejmo si kako mešani celoštevilski linearni program iz primera 4.1 zapišemo v CPLEX LP formatu.

```
\ Generated by MPModelProtoExporter
\   Name              : ProblemPrevoza
\   Format            : Free
\   Constraints       : 2
\   Variables         : 3
\     Binary          : 0
\     Integer         : 2
\     Continuous      : 1
Maximize
 Obj: +30 bakle +70 palme +10 pesek
Subject to
 OmejitevMase: +2 bakle +3 palme +1.7 pesek  <= 55
 OmejitevVolumna: +30 bakle +80 palme +1 pesek  <= 550
Bounds
 0 <= bakle <= inf
 0 <= palme <= inf
 0 <= pesek
Generals
 bakle
 palme
End
```

Omejitvi oziroma pogoja sta 2, eden za maso in eden za volumen. Število spremenljivk je 3, od tega 2 celoštevilski in 1 realna. Optimalna vrednost namenske funkcije je 660.588235294. Optimalno vrednost dosežemo z izbiro 7 bakel, 4 palm in 17.0588235294 litrov peska.

8 Eksperimentalna analiza

Za ovrednotenje delovanja sistema, algoritma in matematičnega modela smo izvedli sistematične teste. Obravnavanje delovanja sistema smo v magistrskem delu izpustili. Ogledali pa si bomo kako smo ovrednotili matematični model in algoritem. Najprej smo izvedli teste pravilnosti, s katerimi smo preverili, da algoritem res naredi pravilen matematični model in ga z reševalnikom CBC tudi pravilno reši. S tem smo testirali tako lastno implementacijo modela kot tudi pravilnost delovanja zunanjih reševalnikov. Obravnavali smo tudi vpliv vhodnih podatkov na hitrost delovanja algoritma. Na koncu smo testirali še različne reševalnike. Na zahtevnejših primerih smo primerjali hitrost delovanja prosto dostopnega reševalnika CBC s komercialnim reševalnikom Gurobi.

8.1 Testi pravilnosti

Osnovni testni scenariji so scenariji z že vnaprej znanimi optimalnimi rešitvami. Z njihovo pomočjo smo preverili pravilnost delovanja algoritma in naredili osnovno poizvedbo o praktični časovni zahtevnosti problema. V nadaljevanju sledi najprej opis osnovnih testnih scenarijev, nato testno okolje, v katerem smo eksperimente izvajali, in nazadnje pregled rezultatov.

8.1.1 Definicija scenarijev

Pridobili smo 9 skupin ročno izdelanih testnih scenarijev, za katere poznamo optimalno vrednost (ročno izračunana). Ti scenariji imajo tudi praktično vrednost, se pravi ponazarjajo realne probleme.

1. **Energijsko prilagodljive prilagodljive ponudbe.**

   Pri teh scenarijih je podana prilagodljiva ponudba porabnika z daljšim časovnim trajanjem. Podanih je tudi več prilagodljivih ponudb proizvajalcev. Cena energije za porabnika je nižja od najdražje proizvodnje in višja od najcenejše proizvodnje. Cilj je, da se uporabi le cenovno ugodne prilagodljive ponudbe.

2. **Prilagodljive ponudbe s privzetim urnikom izvajanja (ena proizvodnja in mnogo porabnikov).**

   Pri teh scenarijih je podana prilagodljiva ponudba z daljšim časovnim trajanjem, katere skupna energija mora biti na koncu enaka nič ($E_f^{\mathrm{T}\min} = E_f^{\mathrm{T}\max} = 0$). Podanih je tudi več prilagodljivih ponudb porabnikov, katerih vir je električni avtomobil. Cilj je, da se energija potrebna za napolnitev električnih avtomobilov pridobi iz daljše prilagodljive ponudbe, le ta pa manjkajočo energijo pridobi na intervalih z ugodnejšo ceno.

3. **Prilagodljive ponudbe za zamik energije (ena proizvodnja in mnogo porabnikov).**

   Za to skupino scenarijev velja enako kot za prejšnjo, le da namesto avtomobilov nastopajo termični zalogovniki, ki se lahko tako polnijo kot praznijo, njihova skupna energija pa je enaka nič.

4. **Polnjenje in praznjenje baterij.**

   Pri teh scenarijih se testira uporaba baterij, predvsem se preverja, da se jih pred potrebo dovolj napolni in se jih kasneje primerno uporabi, pri tem pa je njihova kapaciteta omejena (podana je omejitev kumulativne energije).

5. **Časovno prilagodljive prilagodljive ponudbe.**

   Pri teh scenarijih je podana prilagodljiva ponudba z daljšim časovnim trajanjem, katere skupna energija mora biti na koncu enaka nič ($E_f^{\mathrm{T\,min}} = E_f^{\mathrm{T\,max}} = 0$). Podanih je tudi več prilagodljivih ponudb porabnikov, ki so časovno fleksibilne in energijsko nefleksibilne.

6. **Mešanica različnih prilagodljivih ponudb za premik in porabo energije se kombinira z eno daljšo ponudbo za proizvodnjo (prilagodljiva ponudba podana od operaterja omrežja ponuja porabo s proizvodnjo na začetku in koncu scenarija).**

   Pri teh scenarijih je podana prilagodljiva ponudba z daljšim časovnim trajanjem, ki na začetku in koncu deluje kot proizvajalec, na sredini kot porabnik. Podanih je več krajših prilagodljivih ponudb različnih tipov, s katerimi pokrijemo proizvodnjo in porabo daljše prilagodljive ponudbe.

7. **Mešanica različnih ponudb za proizvodnjo in premih energije se kombinira s ponudbo za porabo (prilagodljiva ponudba podana od operaterja omrežja ponuja proizvodnjo s porabo na začetku in koncu scenarija).**

   Pri teh scenarijih je podana prilagodljiva ponudba z daljšim časovnim trajanjem, ki na začetku in koncu deluje kot porabnik, na sredini kot proizvajalec. Podanih je več krajših prilagodljivih ponudb različnih tipov, s katerimi pokrijemo proizvodnjo in porabo daljše prilagodljive ponudbe.

8. **Časovno progresivno razporejanje prilagodljivih ponudb.**

   Podanih imamo več prilagodljivih ponudb porabnikov in proizvajalcev, lahko so različnih tipov. Pri teh scenarijih ni predvidena daljša prilagodljiva ponudba porabnikov ali proizvajalcev, ki bi pokrila energijska odstopanja.

9. **Prilagodljive ponudbe za proizvodnjo, porabo in zamik z različnimi cenami.**

   Podanih imamo več prilagodljivih ponudb porabnikov in proizvajalcev, lahko so različnih tipov in imajo različne cene. Dodatno imamo podano še neničelno začetno energijsko odstopanje.

8.1.2 Testno okolje

Vsak testni scenarij smo pognali s krajšo časovno omejitvijo reševalnika CBC (5 in 60 sekund). Za še ne potrjeno optimalno rešene scenarije smo scenarije dodatno pognali še s časovno omejitvijo enega dne (86400 sekund). Testi so bili izvedeni na računalniku v virtualnem okolju z 2 GB pomnilnika s frekvenco 1600 MHz in frekvenco procesorja 2.30 GHz.

8.1.3 Pregled rezultatov

Rezultati so zbrani v tabeli 1 in se pri 60 sekundah ujemajo s predvidenimi (ročno izračunanimi) rešitvami.

skupina scenarija	interna oznaka scenarija	število p.p.	rešitev po 5 s	rešitev po 60 s	zaznana opt. po 5 s	zaznana opt. po 60 s
1	1;1;-15;-15;1;0;0	9	−29.925	−29.925	da	da
1	1;1;-15;-15;0.3;0;0	9	16.325	16.325	da	da
1	1;4;-10;-10;1;0;0	33	−52.238	−52.238	da	da
1	1;10;-20;-20;1;0;0	81	−34.463	−34.463	da	da
2	2;1;0;0;0;0;0	9	20.213	20.213	da	da
2	2;1;4;0;0;0;0	9	15.45	15.45	da	da
2	2;4;0;0;0;0;0	33	13.35	13.35	da	da
2	2;10;0;0;0;0;0	81	0.575	0.575	da	da
3	3;1;0;0;0;0;0	9	9.638	9.638	da	da
3	3;10;0;0;0;0;0	81	−2.475	−2.475	da	da
4	4;1;0;0;0;0;0	9	2.46	2.46	da	da
4	4;10;0;0;0;0;0	81	−17.55	−17.55	da	da
5	5;1;1;1;1;0:0	9	−3.025	−3.025	da	da
5	5;8;1;1;1;0:0	65	−11.95	−11.95	da	da
5	5;16;1;1;1;0:0	129	−12.6	−12.6	da	da
6	6;1;0;0;0;0;0	9	−7.228	−7.228	da	da
6	6;10;0;0;0;0;0	81	−69.35	−69.35	da	da
7	7;1;0;0;0;0;0	9	−8.083	−8.083	da	da
7	7;10;0;0;0;0;0	81	−78.549	−78.549	da	da
8	10;2;2;1;0.3;20;0	24	−19	−14	da	da
8	10;2;2;1;0.35;20;0	24	−14	−19	da	da
8	10;2;2;1;0.35;35;0	39	−19.625	−19.625	da	da
8	10.1;10;20;4;0.5;20;0	50	−252.95	−252.95	še ne	še ne
8	10.1;10;20;4;0.5;50;0	80	−410.1	−410.1	še ne	še ne
8	10.1;10;20;4;0.5;100;0	130	−411.15	−411.15	da	da
8	10.2;10;40;1;0.5;500;-100	550	/	−571.665	nerešeno	da
8	11;2;2;5;5;0;0	14	−13.65	−13.65	da	da
8	11;10;20;50;50;0;0	130	−118.56	−118.56	še ne	še ne
8	12;5;5;0.15;20;30;0	60	−56.438	−56.438	da	da
9	20;10;10;10;10;10;50	50	539.4	539.4	da	da
9	20;20;50;50;50;50;-100	220	982.888	982.888	da	da
9	20;20;50;50;50;50;100	220	841.738	841.738	da	da
9	21;20;200;200;200;200;100	820	/	51.185	nerešeno	da

Tabela 1: Pravilnostni testi

Pri večini scenarijev reševalnik najde in potrdi optimalno rešitev po 5 sekundah, pri dveh pa v tem času še ne najde dopustne rešitve. To sta problema z 550 in 820 prilagodljivimi ponudbami. Izkaže se, da so najzahtevnejši problemi iz skupine scenarijev časovno progresivnih razporejanj prilagodljivih ponudb. Pri teh testih tudi po 60 sekundah algoritem za nekatere scenarije ne potrdi optimalne rešitve, čeprav jo izračuna. Splošna ugotovitev je, da je model, implementacija algoritma in delovanje reševalnika pravilno, saj se rešitve v tabeli ujemajo z uradnimi rešitvami za te scenarije. Dodatno opazimo, da je za večino scenarijev problem enostaven, saj algoritem že po 5 sekundah izračuna ustrezno rešitev.

8.2 Analiza vplivov parametrov problema na konvergenco algoritma

V tem razdelku predstavimo rezultate eksperimentov, s katerimi smo preverjali učinkovitost algoritma na zahtevnejših problemih. Predvsem se osredotočimo na vpliv različnih parametrov problema na hitrost konvergence algoritma. Vsi testi izhajajo iz centralnega scenarija, ki mu variiramo različne parametre, katerih vpliv preverjamo. Ogledali si bomo kako na časovno konvergenco algoritma vplivajo dolžina scenarija, število prilagodljivih ponudb in velikost časovne prilagodljivosti prilagodljivih ponudb.

V nadaljevanju si najprej oglejmo definicije scenarijev nato pa rezultate eksperimentov za vsak parameter posebej.

8.2.1 Definicija scenarijev

Testni scenariji so definirani z naslednjimi variabilnimi parametri:

1. Parametrom P_n za *osnovno dolžino scenarija*. Dejanska dolžina scenarija n je odvisna tudi od ostalih parametrov in jo izračunamo.
2. Parametrom P_m za *število prilagodljivih ponudb porabnikov istega tipa*.
3. Parametrom P_t za *velikost dodatne časovne prilagodljivosti prilagodljivih ponudb porabnikov*. Če ima prilagodljiva ponudba f dodatno časovno prilagodljivost P_t, jo lahko postavimo na $P_t + 1$ intervalov.

Testni scenariji imajo naslednje konstantne parametre:

1. Energija začetnega energijskega odstopanja je za vse intervale enaka 0. Ceni energijskega odstopanja pa sta po vseh časovnih intervalih t enaki $p_t^{\mathrm{I}+} = -0.30$ in $p_t^{\mathrm{I}-} = -0.45$.
2. Število neugodnih prilagodljivih ponudb proizvajalcev je 10.
3. Število ugodnih prilagodljivih ponudb proizvajalcev je 20.
4. V scenariju nastopajo 4 različni tipi prilagodljivih ponudb porabnikov:
   - prilagodljive ponudbe s časovno prilagodljivostjo brez energijske prilagodljivosti,
   - prilagodljive ponudbe s časovno in energijsko prilagodljivostjo,
   - električna vozila,
   - baterije.

V nadaljevanju si oglejmo izpeljane parametre, vsi so celoštevilski, kar v enačbah ni posebej poudarjeno. Testni scenariji imajo naslednje izpeljane parametre:

1. Parameter *dolžina mešanja* P_{MIX}, ki se uporablja za dolžino območja razvrščanja prilagodljivih ponudb različnih tipov. Izračunamo ga kot

$$P_{\mathrm{MIX}} = \frac{P_n}{P_m + 5}.$$

Parameter omogoča dodaten časovni zamik prilagodljivih ponudb. Z njim je zagotovljeno, da (na daljših scenarijih z malo prilagodljivimi ponudbami) prilagodljive ponudbe nimajo enakih začetnih intervalov postavitve. S tem se ognemo prekomernim prekrivanjem prilagodljivih ponudb.

2. Parameter *dodatnih intervalov scenarija* $P_{n_{\mathrm{ADD}}}$ se uporablja za podaljšanje dolžine scenarija tako, da so vse prilagodljive ponudbe znotraj scenarija. Izračunamo ga kot

$$P_{n_{\mathrm{ADD}}} = \max\{6, 3 + P_t + P_{\mathrm{MIX}}\},$$

kjer je 6 dolžina ugodnih prilagodljivih ponudb proizvajalcev ter 3 dolžina, P_t časovna prilagodljivost in P_{MIX} dolžina mešanja prilagodljivih ponudb porabnikov.

3. Dolžina scenarija je enaka

$$n = P_n + P_{n_{\mathrm{ADD}}}.$$

4. Število prilagodljivih ponudb je

$$m = 30 + 4P_m,$$

saj imamo v scenariju 30 prilagodljivih ponudb proizvajalcev in 4 različne tipe prilagodljivih ponudb porabnikov, kjer je P_m število prilagodljivih ponudb istega tipa.

Oglejmo si še definicije prilagodljivih ponudb zopet brez poudarjanja celoštevilskosti, ki velja za časovne intervale. Definicije prilagodljivih ponudb:

1. Osnovna proizvodnja oziroma prilagodljive ponudbe neugodnih proizvajalcev:

   (a) Dolžina prilagodljivih ponudb f je enaka dolžini scenarija $l_f = n$.

   (b) Prilagodljiva ponudba se začne na prvem intervalu scenarija

   $$\mathcal{I}_f = \{0\}.$$

   (c) Za vsak časovni interval $j \in \mathcal{T}_f$ so omejitve energije enake

   $$E^{\min}_{f,j} = 0,$$
   $$E^{\max}_{f,j} = 5.$$

   (d) Cena proizvodnje za prilagodljivo ponudbo f, je definirana preko njene zaporedne številke $f_i \in \{0, \ldots, 9\}$ in je za vsak časovni interval t enaka

   $$p^+_{f,t} = 0.075 + 0.03 \cdot f_i.$$

   Cena porabe je $p^-_{f,t} = p^+_{f,t} - 0.025.$

(e) Prilagodljive ponudbe niso časovno fleksibilne in nimajo privzetega urnika izvajanja. Ni omejitev skupne in kumulativne energije.

2. Cenejša proizvodnja oziroma prilagodljive ponudbe ugodnih proizvajalcev:

   (a) Dolžina prilagodljivih ponudb f je enaka $l_f = 6$.

   (b) Prilagodljiva ponudba f se začne na časovnem intervalu izračunanem iz njene zaporedne številke $f_i \in \{0, \ldots, 19\}$ in dolžine scenarija P_n, začetek je podan z

$$\mathcal{I}_f = \left\{ \frac{f_i \cdot P_n}{20} \right\}.$$

   (c) Za vsak časovni interval $j \in \mathcal{T}_f$ so omejitve energije enake

$$E^{\min}_{f,j} = 0,$$
$$E^{\max}_{f,j} = 5.$$

   (d) Cena proizvodnje prilagodljive ponudbe f je za vse časovne intervale t enaka

$$p^{+}_{f,t} = 0.025.$$

   Cena porabe je $p^{-}_{f,t} = 0$.

   (e) Prilagodljive ponudbe nimajo privzetega urnika izvajanja. Ni omejitve skupne in kumulativne energije.

3. Prilagodljive ponudbe s časovno prilagodljivostjo in brez energijske prilagodljivosti:

   (a) Dolžina prilagodljivih ponudb f je enaka $l_f = 3$.

   (b) Množica možnih začetkov prilagodljive ponudbe f se izračuna iz njene zaporedne številke $f_i \in \{0, \ldots, P_m\}$ in dolžine scenarija P_t, začetki so podani z

$$\mathcal{I}_f = \left\{ \frac{f_i \cdot P_n}{P_m} + k \mid k \in \{0, \ldots, P_t\} \right\}.$$

   (c) Za vsak časovni interval $j \in \mathcal{T}_f$ so omejitve energije enake

$$E^{\min}_{f,j} = E^{\max}_{f,j} = -10.$$

   (d) Cena porabe prilagodljive ponudbe f je za vse časovne intervale t enaka

$$p^{-}_{f,t} = 5 - \frac{f_i}{200},$$

   kjer je f_i zaporedna številka prilagodljive ponudbe f. Cena proizvodnje je $p^{+}_{f,t} = p^{-}_{f,t}$.

(e) Prilagodljive ponudbe niso energijsko prilagodljive in nimajo privzetega urnika izvajanja. Ni omejitve skupne in kumulativne energije.

4. Prilagodljive ponudbe tako s časovno kot z energijsko prilagodljivostjo:

   (a) Dolžina prilagodljivih ponudb f je enaka $l_f = 3$.

   (b) Množica možnih začetkov prilagodljive ponudbe f se izračuna iz njene zaporedne številke $f_i \in \{0, \ldots, P_m\}$ in dolžine scenarija P_n, začetki so podani z

$$\mathcal{I}_f = \left\{ \frac{f_i \cdot P_n}{P_m} + \frac{P_{\text{MIX}}}{7} + k \mid k \in \{0, \ldots, P_t\} \right\}.$$

   (c) Za vsak časovni interval $j \in \mathcal{T}_f$ so omejitve energije enake

$$E_{f,j}^{\min} = -2.5,$$
$$E_{f,j}^{\max} = 0.$$

   (d) Omejitev skupne energije je enaka

$$E_f^{\text{Tmin}} = E_f^{\text{Tmax}} = -3.$$

   (e) Cena porabe prilagodljive ponudbe f je za vse časovne intervale t enaka

$$p_{f,t}^{-} = 5 - \frac{f_i}{200} - 0.006 \cdot (f_i \bmod 3),$$

   kjer je f_i zaporedna številka prilagodljive ponudbe f in (f_i mod 3) ostanek pri celoštevilskem deljenju f_i s 3. Cena proizvodnje je $p_{f,t}^{+} = p_{f,t}^{-} + 0.012$.

   (f) Prilagodljive ponudbe nimajo privzetega urnika izvajanja. Ni omejitve kumulativne energije.

5. Prilagodljive ponudbe, katerih vir je električni avtomobil:

   (a) Dolžina prilagodljivih ponudb f je enaka $l_f = 3$.

   (b) Množica možnih začetkov prilagodljive ponudbe f se izračuna iz njene zaporedne številke $f_i \in \{0, \ldots, P_m\}$ in dolžine scenarija P_n, začetki so podani z

$$\mathcal{I}_f = \left\{ \frac{f_i \cdot P_n}{P_m} + \frac{2P_{\text{MIX}}}{7} + k \mid k \in \{0, \ldots, P_t\} \right\}.$$

   (c) Za vsak časovni interval $j \in \mathcal{T}_f$ so omejitve energije enake

$$E_{f,j}^{\min} = -2.5,$$
$$E_{f,j}^{\max} = -1.5.$$

(d) Omejitev skupne energije je enaka

$$E_f^{\mathrm{T\,min}} = E_f^{\mathrm{T\,max}} = -6.$$

(e) Cena porabe prilagodljive ponudbe f je za vse časovne intervale t enaka

$$p_{f,t}^{-} = 5 - \frac{f_i}{200} - 0.007 \cdot (f_i \bmod 3),$$

kjer je f_i zaporedna številka prilagodljive ponudbe f in (f_i mod 3) ostanek pri celoštevilskem deljenju f_i s 3. Cena proizvodnje je $p_{f,t}^{+} = p_{f,t}^{-} + 0.013$.

(f) Prilagodljive ponudbe nimajo privzetega urnika izvajanja. Ni omejitve kumulativne energije.

6. Prilagodljive ponudbe, katerih vir so baterije:

(a) Dolžina prilagodljivih ponudb f je enaka $l_f = 3$.

(b) Množica možnih začetkov prilagodljive ponudbe f se izračuna iz njene zaporedne številke $f_i \in \{0, \ldots, P_m\}$ in dolžine scenarija P_n, začetki so podani z

$$\mathcal{I}_f = \left\{ \frac{f_i \cdot P_n}{P_m} + \frac{3P_{\mathrm{MIX}}}{7} + k \mid k \in \{0, \ldots, P_t\} \right\}.$$

(c) Za vsak časovni interval $j \in \mathcal{T}_f$ so omejitve energije enake

$$E_{f,j}^{\min} = -1.5,$$
$$E_{f,j}^{\max} = 2.5.$$

(d) Omejitev skupne energije je enaka

$$E_f^{\mathrm{T\,min}} = -10,$$
$$E_f^{\mathrm{T\,max}} = 10.$$

(e) Omejitev kumulativne energije je enaka

$$E_f^{\mathrm{C\,min}} = -10,$$
$$E_f^{\mathrm{C\,max}} = 10.$$

(f) Cena porabe prilagodljive ponudbe f je za vse časovne intervale t enaka

$$p_{f,t}^{-} = 5 - \frac{f_i}{200} - 0.008 \cdot (f_i \bmod 3),$$

kjer je f_i zaporedna številka prilagodljive ponudbe f in (f_i mod 3) ostanek pri celoštevilskem deljenju f_i s 3. Cena proizvodnje pa je enaka

$$p_{f,t}^{+} = p_{f,t}^{-} + 0.014.$$

(g) Prilagodljive ponudbe nimajo privzetega urnika izvajanja.

Scenarij je v celoti določen z variabilnimi parametri P_n, P_m in P_t. Za testiranje vpliva parametrov smo si izbrali centralni scenarij, ki ga variiramo po izbranih parametrih. Parametre centralnega scenarija smo definirali kot sledi:

1. Osnovno dolžino scenarija smo nastavili na 40 časovnih intervalov, to je

$$P_n = 40.$$

2. Število prilagodljivih ponudb porabnikov smo nastavili na 30, to pomeni, da je v scenariju 150 prilagodljivih ponudb. Torej velja

$$P_m = 30.$$

3. Dodatno časovno prilagodljivost prilagodljivih ponudb smo nastavili na 16. To pomeni, da je za posamezno časovno fleksibilno prilagodljivo ponudbo 17 možnih začetkov. Parameter je nastavljen kot

$$P_t = 16.$$

Centralni scenarij smo izbrali tako, da nam variiranje parametrov pokaže določene lastnosti konvergence in da iz njega izhajajoči scenariji niso prekomerno enostavni za reševanje.

8.2.2 Testno okolje

Vsak testni scenarij smo pognali s krajšo časovno omejitvijo reševalnika CBC (5, 10, 15, 20, 25, 30, 35, 40, 45, 50, 55, 60, 120, 180, 240, 300, 360, 420, 480, 540, 600 sekund). Za še ne potrjeno optimalno rešene scenarije smo scenarije dodatno pognali še s časovno omejitvijo enega dne (86400 sekund). Eksperimenti so bili izvedeni na računalniku v virtualnem okolju s 3 GB pomnilnika s frekvenco 1333 MHz in frekvenco procesorja 2.67 GHz.

8.2.3 Vpliv števila časovnih intervalov scenarija

Pri tem eksperimentu preverjamo vpliv števila časovnih intervalov scenarija n na konvergenco algoritma. Ker moramo v scenarijih zagotoviti, da so vse prilagodljive ponudbe znotraj območja razporejanja, scenariju dodamo še $P_{n_{\mathrm{ADD}}}$ dodatnih intervalov

$$n = P_n + P_{n_{\mathrm{ADD}}}.$$

V eksperimentu spreminjamo dolžinski parameter P_n, kar se odraža v različnih dolžinah scenarija. Preostala parametra fiksiramo na

$$P_m = 30,$$
$$P_t = 16.$$

P_n	n
20	39
30	49
40	60
50	70
60	80

Tabela 2: Dolžina scenarija n v odvisnosti od osnovne dolžine scenarija P_n

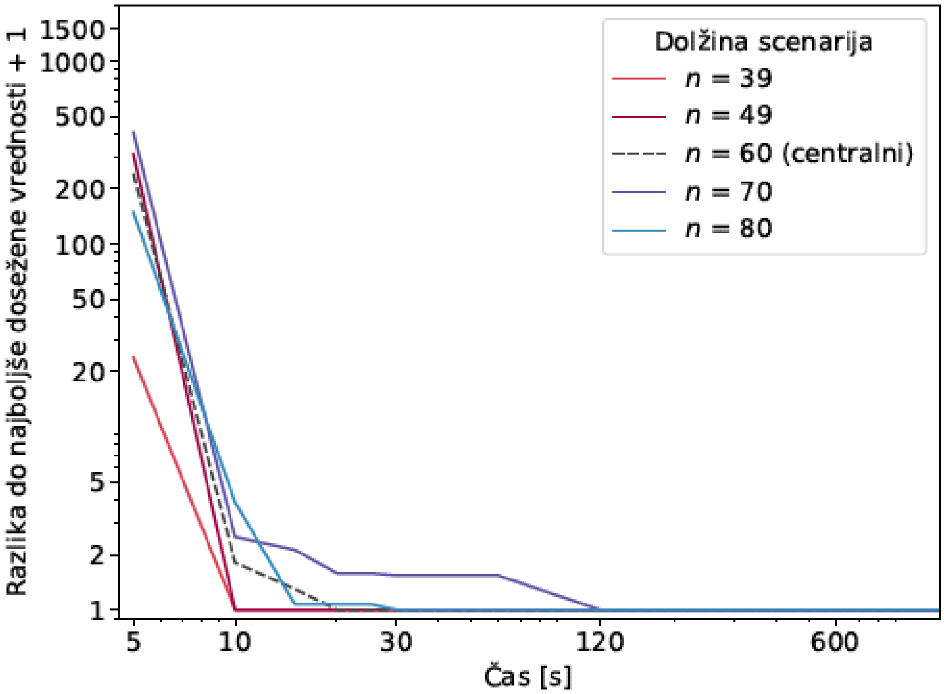
Dolžinski parameter od centralne vrednosti, $P_n = 40$, spreminjamo za 10, tako da zavzame vrednosti podane v tabeli 2.

Rezultati eksperimentov so prikazani v tabeli 3 in na sliki 14. V tabeli 3 so vrednosti namenske funkcije za pridobljeno rešitev v danem časovnem okvirju. Nespreminjajoče vrednosti so zaradi večje preglednosti nadomeščene z navpično črto |. V tabeli 4 so dodatno podana še stanja zaznavanja optimalnosti rešitve s strani algoritma. Na sliki 14 so prikazana odstopanja do najboljše dosežene rešitve, prišteta pa je še enka, da lahko za ordinato uporabimo logaritemsko skalo. Ker se rešitev po preteklem času 60 sekund ne spreminja bistveno, uporabimo tudi za absciso logaritemsko skalo.

	Dolžina scenarija n				
Čas [s]	39	49	60	70	80
5	-6306.77	-6027.02	-6098.01	-5934.84	-6192.46
10	-6329.63	-6336.04	-6338.57	-6340.43	-6337.42
15	|	|	-6339.08	-6340.81	-6340.21
20	|	|	-6339.38	-6341.35	|
25	|	|	|	|	|
30	|	|	|	-6341.39	-6340.28
35	|	|	|	|	|
40	|	|	|	|	|
45	|	|	|	|	|
50	|	|	|	|	|
55	|	|	|	|	|
60	|	|	|	|	|
120	|	|	|	-6341.93	|
180	|	|	|	|	|
240	|	|	|	|	|
300	|	|	|	|	|
360	|	|	|	|	|
420	|	|	|	|	|
480	|	|	|	|	|
540	|	|	|	|	|
600	|	|	|	|	|
86400	|	-6336.08	|	|	|

Tabela 3: Konvergenca rešitve pri različnih dolžinah scenarija

Opazimo, da so rezultati pri petih sekundah daleč od optimalnih vendar se hitro izboljšajo. Pri desetih sekundah reševalnik CBC že najde optimalno rešitev enega



Slika 14: Vpliv dolžine scenarija

	Dolžina scenarija n				
Čas [s]	39	49	60	70	80
60	**da**	ne	ne	ne	ne
600	**da**	ne	ne	**da**	ne
86400		ne	ne		ne

Tabela 4: Zaznana optimalnost pri analizi dolžine scenarija

problema in drastično izboljša rešitve ostalih problemov. Naknadno so izboljšave mnogo manjše, relativno gledano skoraj zanemarljive. Tudi po 600 sekundah reševalnik CBC za tri izmed petih scenarijev ni potrdil optimalnosti rešitve, kar se ne spremeni niti po enem dnevu. Rezultat pokaže, da v splošnem povečanje dolžine scenarija oteži reševanje problema. To sovpada z dejstvom, da ima za daljšo dolžino scenarija model v obliki mešanega celoštevilskega linearnega programa več spremenljivk.

8.2.4 Vpliv števila prilagodljivih ponudb

Pri tem eksperimentu variiramo število prilagodljivih ponudb m preko številskega parametra P_m, upoštevajoč enačbo

$$m = 4P_m + 30.$$

Ostala parametra scenarija pa ne spreminjamo in sta enaka kot v centralnem scenariju, to je

$$P_n = 40,$$
$$P_t = 16.$$

Številski parameter od centralne vrednosti, $P_m = 30$, spreminjamo za 10 kot je prikazano v tabeli 5.

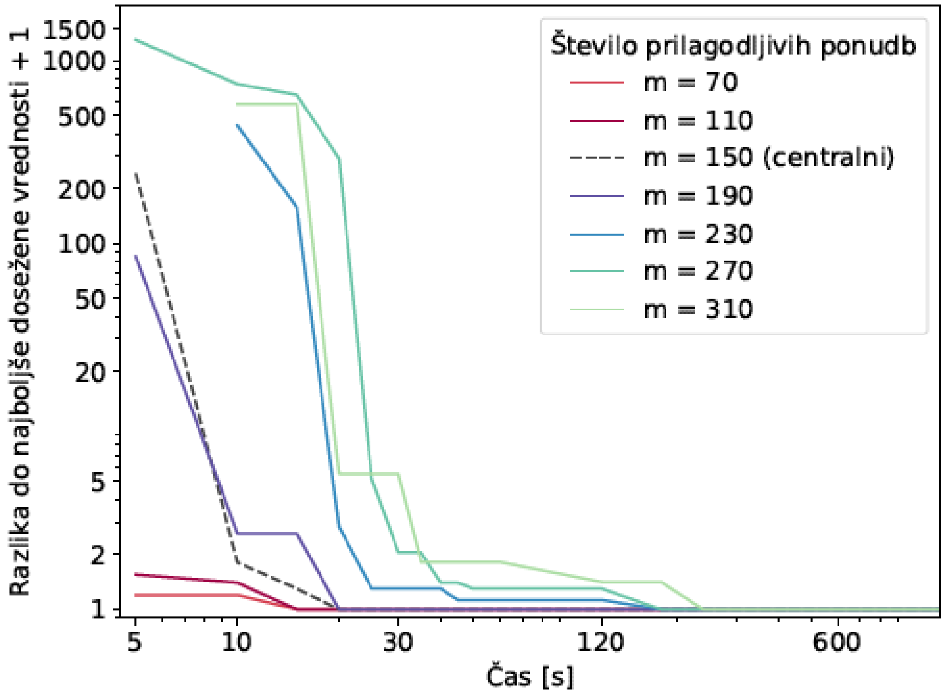
P_m	m
10	70
20	110
30	150
40	190
50	230
60	270
70	310

Tabela 5: Število prilagodljivih ponudb m v odvisnosti od števila P_m

Rezultati eksperimenta so podani v tabeli 6 in na sliki 15. Reševalnik za štiri manjše probleme hitro najde rešitev, medtem ko za večje tri probleme potrebuje več časa. Izkaže se, da potrebuje približno 30 sekund, da najde rešitve, ki so relativno blizu optimalnim. Opazimo lahko tudi očitno korelacijo med številom prilagodljivih ponudb in hitrostjo konvergence rešitve. Več prilagodljivih ponudb naredi problem zahtevnejši za reševanje. To je v skladu z dejstvom, da ima scenarij z več prilagodljivimi ponudbami v modelu mešanega celoštevilskega linearnega programa več spremenljivk in omejitev. V tabeli 7 so podana stanja zaznavanja optimalnosti rešitve s strani algoritma. Poudarimo, da je reševalnik CBC v petih od sedmih primerov v času 600 sekund potrdil optimalnost rešitev.

	Število prilagodljivih ponudb m						
Čas [s]	70	110	150	190	230	270	310
5	-2153.22	-4268.70	-6098.01	-8284.01	/	-10996.49	/
10	|	-4268.85	-6338.57	-8366.53	-9909.64	-11550.72	-13620.78
15	-2153.42	-4269.25	-6339.08	|	-10194.58	-11644.21	|
20	|	|	-6339.38	-8368.13	-10349.65	-12003.4	-14194.42
25	|	|	|	|	-10351.18	-12290.34	|
30	|	|	|	|	|	-12293.49	|
35	|	|	|	|	|	|	-14198.12
40	|	|	|	|	|	-12294.14	|
45	|	|	|	|	-10351.36	|	|
50	|	|	|	|	|	-12294.24	|
55	|	|	|	|	|	|	|
60	|	|	|	|	|	|	|
120	|	|	|	|	|	|	-14198.53
180	|	|	|	|	-10351.48	-12294.54	|
240	|	|	|	|	|	|	-14198.94
300	|	|	|	|	|	|	|
360	|	|	|	|	|	|	|
420	|	|	|	|	|	|	|
480	|	|	|	|	|	|	|
540	|	|	|	|	|	|	|
600	|	|	|	|	|	|	|
86400	|	-4269.65	|	|	|	|	|

Tabela 6: Konvergenca rešitve pri različnem številu prilagodljivih ponudb



Slika 15: Vpliv števila prilagodljivih ponudb

Čas [s]	70	110	150	190	230	270	310
	Število prilagodljivih ponudb						
60	**da**	ne	ne	**da**	ne	ne	ne
600	**da**	ne	ne	**da**	**da**	**da**	**da**
86400		**da**	ne				

Tabela 7: Zaznana optimalnost pri analizi števila prilagodljivih ponudb

8.2.5 Vpliv dolžine časovne prilagodljivosti prilagodljivih ponudb

V tem eksperimentu preverjamo vpliv dolžine časovne prilagodljivosti prilagodljivih ponudb $|\mathcal{I}_f|$ na hitrost konvergence rešitev. Dolžino časovne prilagodljivosti variiramo preko parametra P_t, velja namreč

$$|\mathcal{I}_f| = P_t + 1.$$

Preostala parametra scenarija sta enaka kot v centralnem scenariju, torej

$$P_n = 40,$$
$$P_m = 30.$$

Potrebno je poudariti, da se pri tem eksperimentu spreminja tudi dolžina scenarija n, ki je posredno odvisna tudi od dodatne časovne prilagodljivosti P_t. Dodatna

P_t	$\lvert\mathcal{I}_f\rvert$	n
0	1	46
8	9	52
16	17	60
24	25	68
32	32	76

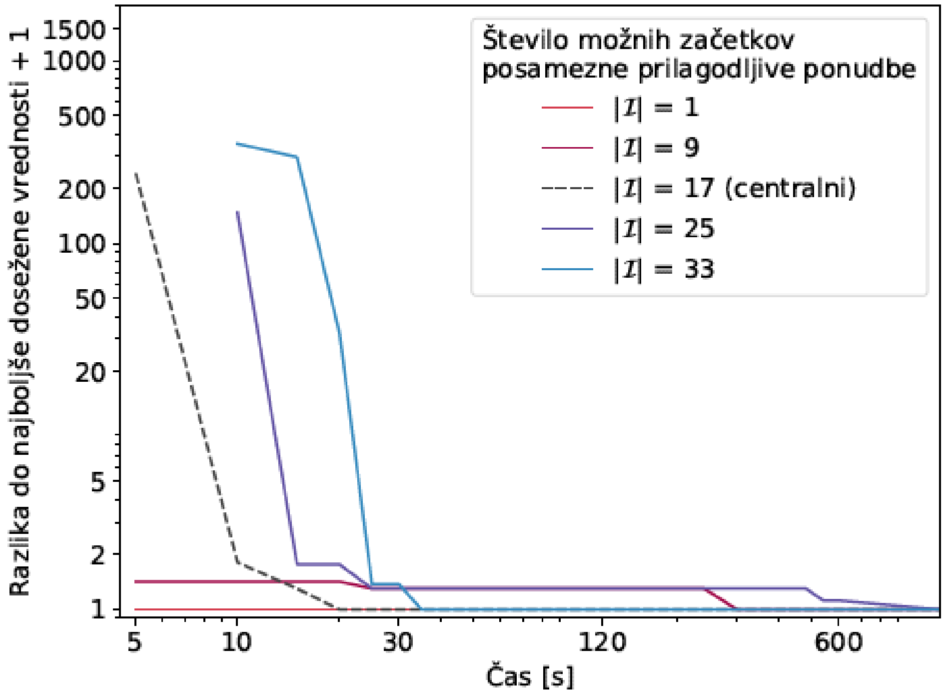
Tabela 8: Časovna prilagodljivost $|\mathcal{I}_f|$ in dolžina scenarija n v odvisnosti od dodatne časovne prilagodljivosti

časovna prilagodljivost je variirana za 8. Za ta eksperiment je v tabeli 8 prikazana odvisnost $|\mathcal{I}_f|$ in n od dodatne časovne prilagodljivosti P_t.

Rezultati eksperimenta so podani v tabeli 9 in prikazani na sliki 16. V tabeli 10 so dodatno podana še stanja zaznavanja optimalnosti rešitve s strani algoritma. Po pričakovanjih je scenarij brez časovne prilagodljivosti mnogo lažje rešljiv. Večje časovne prilagodljivosti pa naredijo reševanje scenarija zahtevnejše, kar se odraža v tem, da reševalnik po 5 sekundah ne najde še nobene dopustne rešitve. Podobno kot pri prejšnjem eksperimentu, je tudi pri tem za relativno sprejemljivo rešitev potrebno okoli 30 sekund.

<table>
<tr><th></th><th colspan="5">Časovna prilagodljivost $|\mathcal{I}_f|$</th></tr>
<tr><th>Čas [s]</th><th>1</th><th>9</th><th>17</th><th>25</th><th>33</th></tr>
<tr><td>5</td><td>-6223.56</td><td>-6325.16</td><td>-6098.01</td><td>/</td><td>/</td></tr>
<tr><td>10</td><td>|</td><td>|</td><td>-6338.57</td><td>-6195.29</td><td>-5994.31</td></tr>
<tr><td>15</td><td>|</td><td>|</td><td>-6339.08</td><td>-6341.62</td><td>-6048.08</td></tr>
<tr><td>20</td><td>|</td><td>|</td><td>-6339.38</td><td>|</td><td>-6312.58</td></tr>
<tr><td>25</td><td>|</td><td>-6325.28</td><td>|</td><td>-6342.08</td><td>-6344.11</td></tr>
<tr><td>30</td><td>|</td><td>|</td><td>|</td><td>|</td><td>|</td></tr>
<tr><td>35</td><td>|</td><td>|</td><td>|</td><td>|</td><td>-6344.48</td></tr>
<tr><td>40</td><td>|</td><td>|</td><td>|</td><td>|</td><td>|</td></tr>
<tr><td>45</td><td>|</td><td>|</td><td>|</td><td>|</td><td>|</td></tr>
<tr><td>50</td><td>|</td><td>|</td><td>|</td><td>|</td><td>|</td></tr>
<tr><td>55</td><td>|</td><td>|</td><td>|</td><td>|</td><td>|</td></tr>
<tr><td>60</td><td>|</td><td>|</td><td>|</td><td>|</td><td>|</td></tr>
<tr><td>120</td><td>|</td><td>|</td><td>|</td><td>|</td><td>|</td></tr>
<tr><td>180</td><td>|</td><td>|</td><td>|</td><td>|</td><td>|</td></tr>
<tr><td>240</td><td>|</td><td>|</td><td>|</td><td>|</td><td>|</td></tr>
<tr><td>300</td><td>|</td><td>-6325.58</td><td>|</td><td>|</td><td>|</td></tr>
<tr><td>360</td><td>|</td><td>|</td><td>|</td><td>|</td><td>|</td></tr>
<tr><td>420</td><td>|</td><td>|</td><td>|</td><td>|</td><td>|</td></tr>
<tr><td>480</td><td>|</td><td>|</td><td>|</td><td>|</td><td>|</td></tr>
<tr><td>540</td><td>|</td><td>|</td><td>|</td><td>-6342.26</td><td>|</td></tr>
<tr><td>600</td><td>|</td><td>|</td><td>|</td><td>|</td><td>|</td></tr>
<tr><td>86400</td><td>|</td><td>|</td><td>|</td><td>-6342.65</td><td>|</td></tr>
</table>

Tabela 9: Konvergenca rešitve pri različni časovni prilagodljivosti



Slika 16: Vpliv časovne prilagodljivosti prilagodljivih ponudb

	Časovna prilagodljivost				
Čas [s]	1	9	17	25	33
60	**da**	ne	ne	ne	ne
600	**da**	ne	ne	ne	**da**
86400		ne	ne	ne	

Tabela 10: Zaznana optimalnost pri analizi časovne prilagodljivosti

8.2.6 Diskusija rezultatov

Na splošno opazimo, da za dane scenarije hitro dobimo dober približek optimalni rešitvi, za točno rešitev pa je časovna zahtevnost eksponentna glede na velikost vhodnih podatkov. Opazimo, da pri teh scenarijih z večanjem velikosti scenarija (oziroma parametrov) dobimo večjo časovno zahtevnost. Opozorimo, da v splošnem to ni nujno res. Oglejmo si naslednje tri hipotetične protiprimere oziroma razmisleke:

1. Recimo, da imamo scenarij s pozitivnim energijskim odstopanjem, ki ga lahko izničimo z več krajšimi prilagodljivimi ponudbami. Če dodamo večjo zelo ugodno prilagodljivo ponudbo, lahko le ta trivialno pokrije vse odstopanje, kar se izračuna že z relaksacijo mešanega celoštevilskega linearnega programa. V tem primeru smo dodali prilagodljivo ponudbo in zmanjšali časovno zahtevnost.

2. Možno bi bilo tudi, da bi s povečanjem časovne prilagodljivosti neke prilagodljive ponudbe naredili njen razpored enostavnejši.

3. Daljšanje scenarija sicer povečuje problem, vendar s tem redči gostoto prilagodljivih ponudb, kar bi se lahko odražalo v manjši interakciji prilagodljivih ponudb in posledično manjši časovni zahtevnosti.

S temi primeri pokažemo, da lahko v teoriji tudi z večanjem števila spremenljivk dobimo krajši čas trajanja izvajanja algoritma.

8.3 Primerjava reševalnikov

V tem razdelku eksperimentalno ovrednotimo izbrani reševalnik CBC s primerjavo z komercialnim reševalnikom Gurobi.

8.3.1 Izbira testnih scenarijev

Za testne scenarije smo izbrali reprezentativni vzorec scenarijev iz razdelka 8.2. Izbrali smo tako lažje kot zahtevnejše scenarije in sicer scenarije s parametri:

- $P_n = 40; P_m = 30; P_t = 16$,
- $P_n = 40; P_m = 30; P_t = 32$,
- $P_n = 40; P_m = 70; P_t = 16$,
- $P_n = 30; P_m = 30; P_t = 16$,
- $P_n = 50; P_m = 30; P_t = 16$,

kjer je P_n osnovna dolžina scenarija, P_m število prilagodljivih ponudb in P_t velikost dodane časovne prilagodljivosti.

8.3.2 Testno okolje

Vsak testni scenarij smo pognali s krajšimi časovnimi omejitvami reševalnika CBC in Gurobi (5, 10, 15, 20, 25, 30, 35, 40, 45, 50, 55, 60 sekund). Ker v tem času reševalnik Gurobi potrdi optimalnost rešitve smo za daljše čase (120, 180, 240, 300, 360, 420, 480, 540, 600, 86400 sekund) pognali le reševalnik CBC. Testi so bili izvedeni na računalniku v virtualnem okolju z 2 GB pomnilnika s frekvenco 1600 MHz in frekvenco procesorja 2.30 GHz.

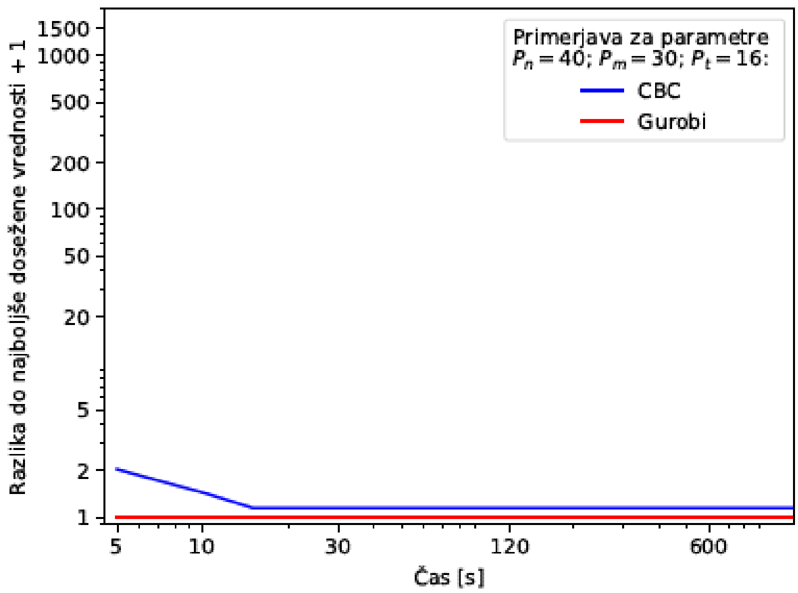
8.3.3 Pregled rezultatov

Reševalnika smo ovrednotili na petih scenarijih. Pri vseh scenarijih reševalnik Gurobi po 60 sekundah potrdi optimalnost rešitve, medtem ko reševalnik CBC tudi po enem dnevu (86400 sekundah) ne potrdi optimalne rešitve za vse scenarije, ampak le pri treh. Oglejmo si slike 17, 18 in 19, na katerih so prikazani scenariji, kjer Gurobi konvergira hitreje in tudi doseže boljšo izmed obeh optimalnih rešitev (glej opombo 8.1). Na teh scenarijih je razlika rešitve reševalnikov po 60 sekundah manjša od 1. Oglejmo si še sliko 20, ki je eden redkih primerov, kjer reševalnik CBC konvergira hitreje. Slika 21 prikazuje scenarij, kjer reševalnik Gurobi konvergira hitreje, vendar reševalnik CBC doseže boljšo izmed obeh optimalnih rešitev (glej opombo 8.1). Na teh dveh scenarijih je razlika rešitve reševalnikov po 60 sekundah manjša od 1.

Iz rezultatov je razvidno, da je reševalnik Gurobi boljši, saj tudi na zahtevnejših scenarijih najde optimalno rešitev po 15 sekundah. Razvidno je tudi, da obstajajo scenariji, na katerih je reševalnik CBC malenkost hitrejši.

Opomba 8.1. Odstopanja optimalnih rešitev za manj kot 1 lahko pripišemo numerični napaki, oziroma toleranci reševalnika do razlike od vrednosti namenske funkcije za optimalno rešitev.

Čas [s]	CBC	Gurobi
5	-6338.48	-6339.53
10	-6339.08	|
15	-6339.38	|
20	|	|
25	|	|
30	|	|
35	|	|
40	|	|
45	|	|
50	|	|
55	|	|
60	|	|
120	|	|
180	|	|
240	|	|
300	|	|
360	|	|
420	|	|
480	|	|
540	|	|
600	|	|
86400	|	|

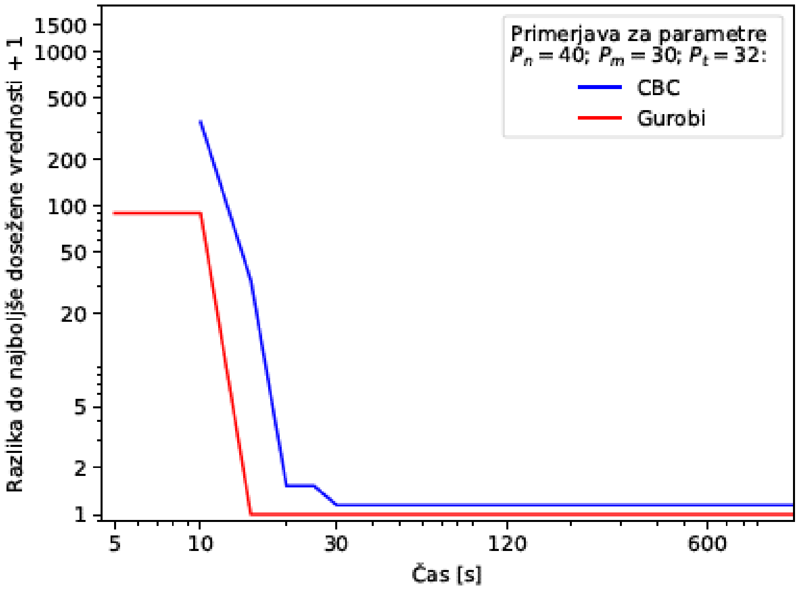


a) Absolutni rezultati

b) Relativni rezultati

Slika 17: Primerjava reševalnikov CBC in Gurobi za parametre $P_n = 40; P_m = 30; P_t = 16$

Čas [s]	CBC	Gurobi
5	/	-6255.27
10	-5994.31	|
15	-6312.58	-6344.63
20	-6344.11	|
25	|	|
30	-6344.48	|
35	|	|
40	|	|
45	|	|
50	|	|
55	|	|
60	|	|
120	|	|
180	|	|
240	|	|
300	|	|
360	|	|
420	|	|
480	|	|
540	|	|
600	|	|
86400	|	|

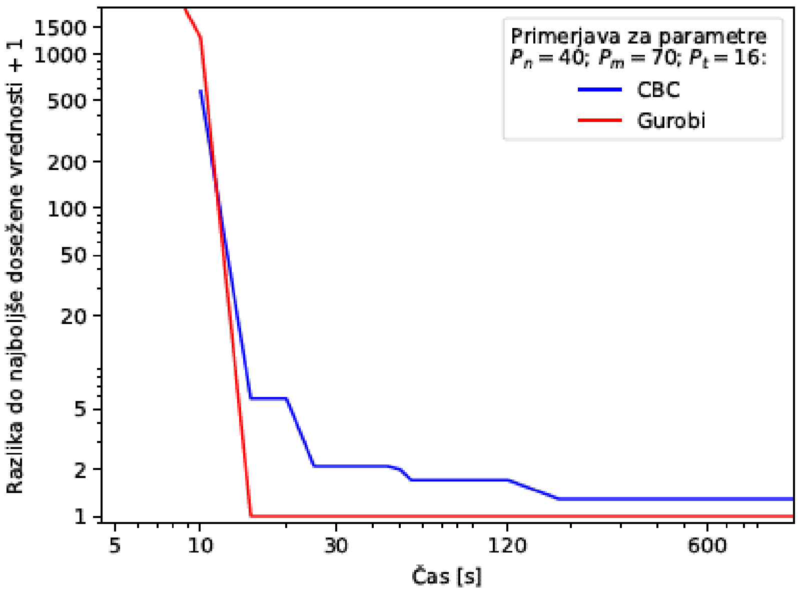


a) Absolutni rezultati

b) Relativni rezultati

Slika 18: Primerjava reševalnikov CBC in Gurobi za parametre $P_n = 40; P_m = 30; P_t = 32$

Čas [s]	CBC	Gurobi
5	/	0
10	-13620.78	-12921.39
15	-14194.42	-14199.24
20	|	|
25	-14198.12	|
30	|	|
35	|	|
40	|	|
45	|	|
50	-14198.23	|
55	-14198.53	|
60	|	|
120	|	|
180	-14198.94	|
240	|	|
300	|	|
360	|	|
420	|	|
480	|	|
540	|	|
600	|	|
86400	|	|

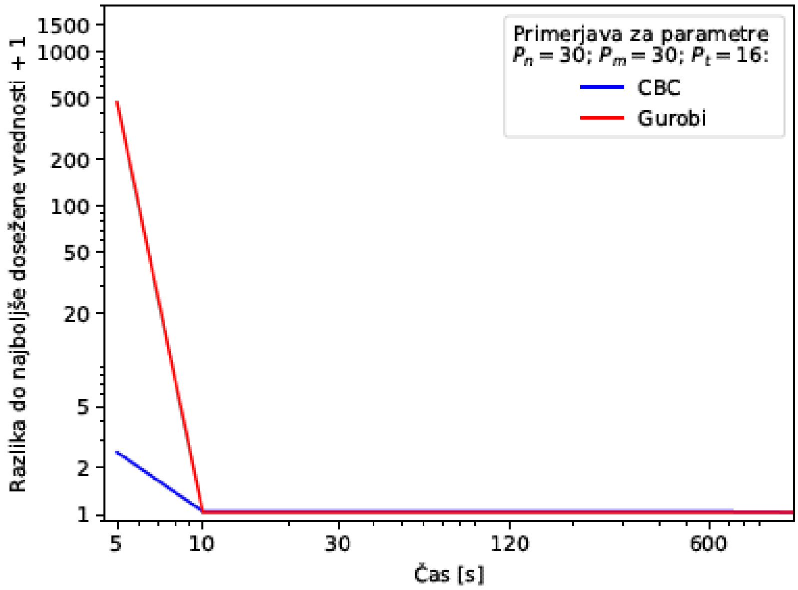


a) Absolutni rezultati

b) Relativni rezultati

Slika 19: Primerjava reševalnikov CBC in Gurobi za parametre $P_n = 40; P_m = 70; P_t = 16$

Čas [s]	CBC	Gurobi
5	-6334.55	-5862.79
10	-6336.04	-6336.05
15	|	|
20	|	|
25	|	|
30	|	|
35	|	|
40	|	|
45	|	|
50	|	|
55	|	|
60	|	|
120	|	|
180	|	|
240	|	|
300	|	|
360	|	|
420	|	|
480	|	|
540	|	|
600	|	|
86400	-6336.08	|

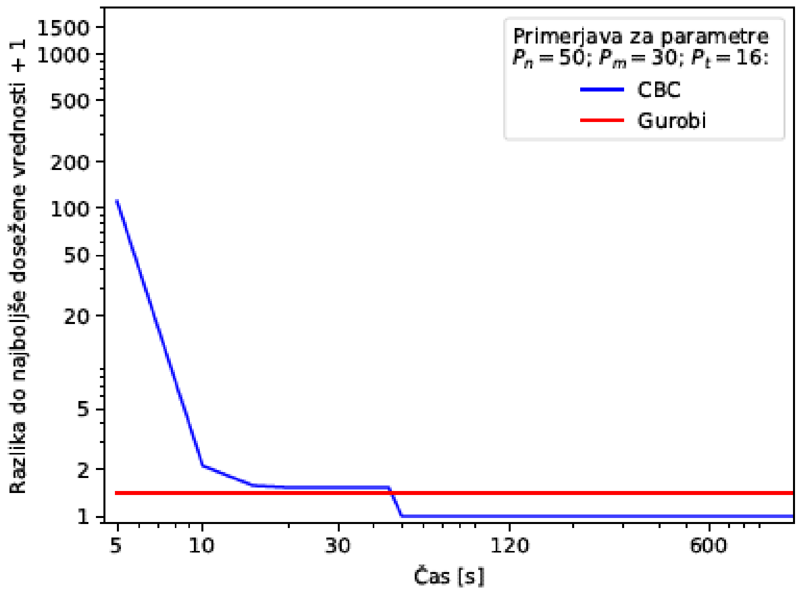


a) Absolutni rezultati

b) Relativni rezultati

Slika 20: Primerjava reševalnikov CBC in Gurobi za parametre $P_n = 30; P_m = 30; P_t = 16$

Čas [s]	CBC	Gurobi
5	-6231.64	-6341.53
10	-6340.81	|
15	-6341.35	|
20	-6341.39	|
25	|	|
30	|	|
35	|	|
40	|	|
45	|	|
50	-6341.93	|
55	|	|
60	|	|
120	|	|
180	|	|
240	|	|
300	|	|
360	|	|
420	|	|
480	|	|
540	|	|
600	|	|
86400	|	|



a) Absolutni rezultati

b) Relativni rezultati

Slika 21: Primerjava reševalnikov CBC in Gurobi za parametre $P_n = 50; P_m = 30; P_t = 16$

9 Zaključki

V magistrskem delu smo formalno definirali problem razporejanja prilagodljivih ponudb električne energije. Za problem smo naredili model v obliki mešanega celoštevilskega linearnega programa. Pri tem smo se opirali predvsem na poznavanje osnovnih logičnih simbolov, analize in algebre. Za samo izdelavo modela pa je pomemben predvsem navdih in prilagajanje formalne definicije potrebam modela. Velikokrat namreč obstaja več različnih dobro definiranih formulacij, ki dobro opišejo določen problem, poskušamo pa poiskati čim boljšo. V delu smo navedli tudi osnovno teorijo mešanega celoštevilskega linearnega programiranja. Navedli smo uporabne prijeme, ki smo jih dognali tekom magistrskega dela. Ti lahko služijo kot vodilo raziskovalcem, ki se srečujejo s podobnimi problemi in jih tudi poskušajo reševati z uporabo mešanega celoštevilskega linearnega programiranja.

V eksperimentalnem delu smo preverili pravilno implementacijo in pravilnost modela problema razporejanja prilagodljivih ponudb v obliki mešanega celoštevilskega linearnega programa. Na problemu analiza parametrov pokaže, da večanje dimenzij problema po pričakovanjih prinese večjo časovno zahtevnost. Pri večanju dimenzij problema namreč narašča velikost problema in še posebej število spremenljivk in omejitev mešanega celoštevilskega linearnega programa. Primerjava reševalnikov je pokazala, da je za ta problem reševalnik Gurobi malenkost boljši. Vendar pa oba reševalnika dosegata za praktične namene odlične rezultate.

Zaključimo lahko, da je mešano celoštevilsko linearno programiranje uporabno za reševanje takšne vrste problemov razporejanja. Za njihovo reševanje obstajajo kakovostni reševalniki. Če je le možno, se optimizacijski problem splača prevesti v obliko (mešanega celoštevilskega) linearnega programa, kjer se lahko z uporabo obstoječih reševalnikov doseže dobre rezultate.

Literatura

[1] *Google's OR-Tools*, [ogled 10. 9. 2018], dostopno na `https://developers.google.com/optimization/`.

[2] *LP file format*, 2019, [ogled 28. 5. 2019], dostopno na `http://lpsolve.sourceforge.net/5.5/lp-format.htm`.

[3] M. S. Bazaraa, J. J. Jarvis in H. D. Sherali, *Linear Programming and Network Flows*, John Wiley & Sons, 2011.

[4] R. G. Bland, *New finite pivoting rules for the simplex method*, Mathematics of operations Research **2** (1977) 103–107.

[5] G. Černe in Z. Marinšek, *Mirabel – prilagajanje odjema električne energije z zaprtimi pogodbami*, v: PIES' 12: Posvetovanje o informatiki v energetiki Slovenije, 2012.

[6] G. Černe, Z. Marinšek in M. Bizjak, *Zahteve pri uporabi prilagajanja odjema in razpršene proizvodnje*, v: 33. Kotnikovi dnevi, 2012.

[7] B. Filipič, K. Gantar in T. Tušar, *GOFLEX-Scheduling: Specifications of optimization algorithms to schedule flexible offers for electricity production and consumption*, teh. por. IJS-DP 12400, Institut "Jožef Stefan", Ljubljana, 2018.

[8] J. Forrest, *COIN-OR branch and cut*, 2019, [ogled 28. 5. 2019], dostopno na `https://github.com/coin-or/Cbc`.

[9] J. Grasselli in A. Vadnal, *Linearna algebra. Linearno programiranje*, Matematika-fizika: zbirka univerzitetnih učbenikov in monografij **4**, DMFA – založništvo, 2003.

[10] Gurobi Optimization, LLC, *Gurobi Optimizer Reference Manual*, [ogled 28. 5. 2019], dostopno na `https://www.gurobi.com`.

[11] F. Hillier in G. Lieberman, *Introduction to Operations Research*, McGraw-Hill Higher Education, Boston, 2010.

[12] H. Höfling in dr., *Negative prices on the electricity wholesale market and impacts of § 24 EEG*, 2015, [ogled 10. 9. 2018], dostopno na `https://www.erneuerbare-energien.de/EE/Redaktion/DE/Downloads/discussion-paper-negative-prices-long.pdf`.

[13] T. Janša, *Spletna aplikacija za linearno programiranje*, diplomsko delo, Fakulteta za matematiko in fiziko, Univerza v Ljubljani, 2012.

[14] N. Karmarkar, *A new polynomial-time algorithm for linear programming*, Combinatorica **4** (1984) 373–395.

[15] J. E. Mitchell, *Branch-and-cut algorithms for combinatorial optimization problems*, v: Handbook of Applied Optimization, Oxford University Press, Oxford, 2000.

[16] M. Petkovšek, *NP-polni problemi*, diplomsko delo, Univerza v Ljubljani, 1978.

[17] L. Šikšnys in dr., *D2.1 Automatic trading platform requirement & interface specification*, 2017, [ogled 10. 9. 2018], dostopno na `https://goflex-project.eu/Deliverables.html`.

[18] S. Smale, *Mathematical problems for the next century*, Mathematical Intelligencer **20** (1998) 7–15.

[19] R. J. Vanderbei, *Linear Programming: Foundations and Extensions*, Springer, New York, 2008.