

# Primerjava reševanja ILP v Matlab in Gurobi

David Rubin (david.rubin@student.um.si)

30. december 2020

## 1 Uvod

V tem poročilu si bomo ogledali kako se lahko pohitri reševanje ILP problemov z uvedbo distribuiranega procesiranja. Kot standarden čas reševanja bomo uporabili MATLAB oziroma njegovo funkcijo *intlinprog* iz *optimization toolbox-a*. Kot predlagano pohitritev, smo si v tem poročilu ogledali orodje Gurobi <sup>1</sup>. Orodje prav tako omogoča določevanje števila uporabljenih niti, tako da lahko orodje tudi nekoliko bolj pošteno primerjamo z *intlinprog*, ko nastavimo enojno nit. Za testne probleme smo si izbrali zbirko MIPLIB2017 [1]. Ker je zbirka precej obsežna, smo iz nje izluščili nekaj problemov, za katere domnevamo, da dovolj dobro predstavljajo testno domeno. Izbrane testne probleme smo predstavili v poglavju 3. Predvidevamo, da je bralec seznanjen s programskim okoljem MATLAB, zato bomo podali le kratek uvod v Gurobi.

## 2 Gurobi

Gurobi je komercialna programska oprema, ki omogoča optimiziranje problemov iz:

- lineranega programiranja (angl. *linear programming* - LP),
- mešanega celoštevilskega linearnega programiranja (angl. *mixed-integer linear programming* - MILP),
- kvadratičnega programiranja (angl. *quadratic programming* - QP),
- mešanega celoštevilskega kvadratičnega programiranja (angl. *mixed-integer quadratic programming* - MIQP),
- kvadratično omejenega programiranja (angl. *quadratically constrained programming* - QCP),
- mešano celoštevilskega kvadratično omejenega programiranja (angl. *Mixed-integer quadratically constrained programming* - MIQCP).

---

<sup>1</sup>Orodje Gurobi je dostopno na <https://www.gurobi.com/>. Za potrebe te naloge smo uporabili akademsko licenco

Pokriva torej veliko območje problemov, za zadnje štiri pa je vredno omeniti, da podpira tako konveksne kot tudi konkavne probleme. Za reševanje nabora problemov imajo implementirane metode Simplex in paralelno oviro s križanjem, sočasnostjo, in sejanjem (angl. *parallel barrier with crossover, concurrent, and sifting*) za bolj zahtevne pa uporabijo tudi deterministično paralelno ne-tradicionalno iskanje, heuristike, rezanja ravnin in pa razbijanje simetrij [3].

Algoritmi so zasnovani deterministično, torej bi več zagonov skozi isti model dalo enake rezultate. Zasnovani so tudi tako, da po privzetem uporabijo vsa jedra, ki so v nekem sistemu na voljo. Trdijo tudi, da njihovi QCP in MIQCP algoritmi nudijo dvakrat boljše učinkovitost kot njihov glavni tekmeči [2]. Slednje trditve ne moremo sami potrditi, ne moremo pa je niti zavreči, tako da prepuščamo bralcu, da sam oceni resničnost. Podjetje so ustanovili dr. Robert Bixby, ki je pred tem tudi ustanovil CPLEX in pa Dr. Zonghao Gu ter Dr. Edward Rothberg. Slednja sta pred ustanovitvijo sodelovala pri CPLEX kot glavna inženirja [4]. Kot zanimivost, ime *GuRoBi* je nastalo iz njihovih priimkov. Med njihove partnerje spadajo podjetja kot je recimo Toyota, Microsoft, Google, AirFrance in pa tudi SAP.

Gurobi podpira množico programskih jezikov in oblik programiranja. Nudijo objektno orientirane vmesnike (JAVA, C++, .NET in Python), matrično orientiranje (C, MATLAB in R), povezavo do standardnih modelnih jezikov (AIMMS, AMPL, MPL) in druge [3]. V kolikor bi orodje želeli uporabljati le v jeziku Python se lahko preskoči namestitev celotne knjižnice in uporabi `pip` ali **Anaconda** paket. Nudijo pa tudi možnost porazdeljenega reševanja problemov na gruči (*Gurobi Remote Services*), ki pa zahteva ločeno licenco.

## 2.1 Priprava okolja

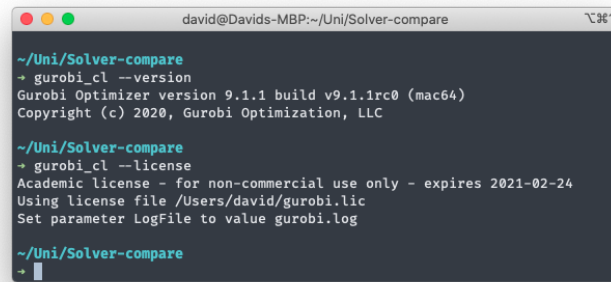
Za uporabnike je na voljo vodič za hitro namestitev za vse tri večje operacijske sisteme<sup>2</sup>. Posebno pozornost je potrebno posvetiti sekciji za pridobitev akademske licence. Zahtevana je uporaba epoštnega naslova univerze in imeti dostop do spleta v času aktivacije. Po prijavi nam je na voljo dostop do prenosa njihove knjižnice in CLI orodja<sup>3</sup>. Znotraj dokumenta navedejo, da se za akademski dostop poleg licenčnega ključa preverja tudi domensko ime iz lokacije dostopa. Pri naši aktivaciji se ni pojavila težava (ukaz `grbgetkey <koda>`), kljub temu, da smo med aktivacijo dostopali do njihovega strežnika zunaj omrežja univerze. V kolikor Gurobi med aktivacijo sporoči kakšno napako (primer *hostname ... not recognized as belonging to an academic domain*), pa je možna uporaba VPN, ki je na voljo vsem študentom (in zaposlenim) Univerze v Mariboru.

V kolikor je orodje nameščeno in je bila aktivacija uspešna, bi sedaj morali imeti delujoč *Gurobi optimizer* kot je prikazano na sliki 1. Ob težavah se je najbolje obrniti na njih uraden vodič glede na vaš operacijski sistem [5]. CLI orodje podpira 14 formatov za podajanje optimizacijskega modela in drugih pomožnih podatkov [6]. V nadaljevanju bomo uporabljali dva: LP in MPS format. LP format je namenjen za boljšo berljivost iz strani človeka, medtem ko je MPS format najstarejši in najpogostejše uporabljen za shranjevanje matematičnih programskih modelov. MPS format je veliko strožji kar se tiče zapisa (fiksni začetni stolpci za vrednosti

---

<sup>2</sup>Vodič je dostopen na <https://www.gurobi.com/documentation/quickstart.html>

<sup>3</sup>Programska oprema Gurobi se pridobi na naslovu <https://www.gurobi.com/downloads/gurobi-software/>



```
david@Davids-MBP:~/Uni/Solver-compare
~/Uni/Solver-compare
+ gurobi_cl --version
Gurobi Optimizer version 9.1.1 build v9.1.1rc0 (mac64)
Copyright (c) 2020, Gurobi Optimization, LLC

~/Uni/Solver-compare
+ gurobi_cl --license
Academic license - for non-commercial use only - expires 2021-02-24
Using license file /Users/david/gurobi.lic
Set parameter Logfile to value gurobi.log

~/Uni/Solver-compare
+ 
```

Slika 1: Primer izpisa ukazov za podatke o verziji in licenci po uspešni namestitvi *Gurobi optimizierja* na macOS.

ipd.), zato ga bomo le uporabili pri testnih primerih.

## 2.2 Reševanje ILP problemov z Gurobi optimizer

Za boljši vploged v orodje, si bomo v nadaljevanju ogledali enostaven primer optimizacijskega problema in kako se le ta reši s pomočjo Gurobi optimizierja. Primer je prilagojena verzija, katere original se nahaja tudi v vodiču za hiter zagon [5].

### 2.2.1 Opis problema

V času pisanja poročila je blizu konec koledarskega leta. *United States Mint* (ustanova, ki proizvaja ameriške kovance), ima še nekaj neporabljene zalogo materialov za proizvodnjo kovancev. Predpostavimo, da si želijo porabiti vso preteklo zalogo, preden se lotijo nabave materialov za serijo kovancev novo leto. Ustanova proizvaja šest različnih kovancev, sestava katerih pa je podana v tabeli 1. Ustanova sedaj želi proizvesti takšne tipe kovancev, da bo njihova skupna vrednost največja. Kakšna naj bo količina vsakega proizvedenega kovanca, če imamo podano zalogo vsakega materiala?

Tabela 1: Vsebnost materialov v ameriških kovancih glede na US Mint specifikacijo [7]

	<b>Cent</b>	<b>Nickel</b>	<b>Dime</b>	<b>Quarter</b>	<b>Half</b>	<b>Dollar</b>
Baker (Cu)	0.0625g	3.7500g	2.0791g	5.1977g	10.3954g	7.1685g
Nikelj (Ni)		1.2500g	0.1889g	0.4723g	0.9446g	0.1620g
Cink (Zn)	2.4375g					0.4860g
Mangan (Mn)						0.2835g
Skupaj	2.5g	5.0g	2.268g	5.670g	11.340g	8.1g

### 2.2.2 Priprava vhodnih datotek

Za rešitev problema potrebujemo izvesti 3 korake:

1. Določitev odločitvenih spremenljivk

2. Določitev odločitvene funkcije (v tem primeru linearne)

3. Določitev omejitev

Odločitvene spremenljivke so dokaj jasne: zanima nas količina vsakega kovanca, ki ga naj proizvedemo. Ker smo programerji in za lepšo nazornost jim bomo podali opisna imena, za matematike pa še vektorske spremenljivke. Spremenljivke torej poimenujemo *Cents*, *Nickels*, *Dimes*, *Quarters*, *Halves*, *Dollars* (oziroma  $x_1, x_2 \dots x_6$ ). Vpeljimo pa še imena spremenljivk za porabljen količino materialov *Cu*, *Ni*, *Zn*, *Mn* ( $y_1, y_2, y_3$  in  $y_4$ ). Te spremenljivke so določene in predstavljajo material, ki je na zalogi. Za zalogo bomo predpostavili 1000g bakra in po 50g vsakega ostalega materiala. Formalen matematični zapis modela (ki ga maksimiziramo) se glasi:

$$f(x) = 0.01x_1 + 0.05x_2 + 0.1x_3 + 0.25x_4 + 0.5x_5 + 1x_6 \quad (1)$$

pri čemer so  $x_n$  količine proizvedenih kovancev, koeficienti pred njimi pa predstavljajo vrednost posameznega kovanca. Omejitve so sledeče:

$$\begin{aligned} 0.0625x_1 + 3.7500x_2 + 2.0791x_3 + 5.1977x_4 + 10.3954x_5 + 7.1685x_6 &\leq 1000 \\ 1.2500x_2 + 0.1889x_3 + 0.4723x_4 + 0.9446x_5 + 0.1620x_6 &\leq 50 \\ 2.4375x_1 + 0.4860x_6 &\leq 50 \\ 0.2835x_6 &\leq 50 \\ x_1, x_2, x_3, x_4, x_5, x_6 &\in \mathbb{N} \end{aligned} \quad (2)$$

Skratka količina porabljenega materiala ne sme presežati zaloge, prav tako pa ne moremo proizvesti manj kot celoto kovanca (celoštevilska omejitev). Isto omejitev lahko zapišemo na programerski način (prej omenjene opisne spremenljivke) in LP format zapisa (glej kodo 1). Slednje nam predstavlja enega izmed možnih vhodov v Gurobi optimizer.

Koda 1: Primer LP formata za problem proizvodnje kovancev

```
Maximize
    .01 Pennies + .05 Nickels + .1 Dimes + .25 Quarters + .5 Halves + 1 Dollars
Subject To
    Copper: .0625 Pennies + 3.75 Nickels + 2.0791 Dimes + 5.1977 Quarters +
        ↪ 10.3954 Halves + 7.1685 Dollars - Cu = 0
    Nickel: 1.25 Nickels + .1889 Dimes + .4723 Quarters + .9446 Halves + .162
        ↪ Dollars - Ni = 0
    Zinc: 2.4375 Pennies + .486 Dollars - Zn = 0
    Manganese: .2835 Dollars - Mn = 0
Bounds
    Cu <= 1000
    Ni <= 50
    Zn <= 50
    Mn <= 50
Integers
    Pennies Nickels Dimes Quarters Halves Dollars
End
```

LP format ima nekaj posebnosti. V podanem primeru imamo štiri sekcije (*Maximize*, *Subject to*, *Bounds*, *Integers*), ki si morajo vedno slediti v tem vrstnem redu. Sekcij je lahko več in tudi

znotraj sekcij se lahko pojavi več scenarijev. Vse to je opisano v navodilih za uporabo<sup>4</sup>. Druga posebnost je zapis vrednosti (spremenljivk, konstant, operatorjev ...). Te morajo biti ločene s presledki oziroma novimi vrsticami. Tako je `+ 0.1x` napačen zapis. Pravilno je `+ 0.1 x`. Tretje splošno pravilo je, da se spremenljivke vedno pišejo na levi in konstante na desni (primer tega je v sekciji *Subject to*). Na koncu pa naj še omenimo, da je po privzetem spodnja meja enaka 0, razen če je eksplicitno zapisana. Zapis `Cu <= 1000` je v resnici `0 <= Cu <= 1000`. To pomeni, da Gurobi po privzetem vsako spremenljivko omeji na nenegativna števila.

### 2.2.3 Zagon in rešitve

Najenostavnejši način uporabe Gurobi, kadar imamo podan model v datoteki, je preko ukazne vrstice. Z ukazom `gurobi_cl <model.lp>` lahko poženemo orodje in se problem prične reševati. Kot rešitev imamo na standardnem izhodu podano samo vrednost cenične funkcije, če želimo videti še vrednosti posameznih spremenljivk, je potrebno dodati parameter `ResultFile=<izhod.sol>`. Celoten ukaz je torej:

```
gurobi_cl ResultFile=code/coins.sol code/coins.lp
```

Izpis ukaza je viden na sliki 2, izhodna datoteka z rešitvami pa v kodi 2.

```

~/Uni/Solver-compare
+ gurobi_cl ResultFile=code/coins.sol code/coins.lp
Academic license - for non-commercial use only - expires 2021-02-24
Using license file /Users/david/gurobi.lic
Set parameter LogFile to value gurobi.log

Gurobi Optimizer version 9.1.1 build v9.1.1rc0 (mac64)
Copyright (c) 2020, Gurobi Optimization, LLC

Read LP format model from file code/coins.lp
Reading time = 0.00 seconds
: 4 rows, 10 columns, 18 nonzeros
Thread count: 4 physical cores, 8 logical processors, using up to 8 threads
Optimize a model with 4 rows, 10 columns and 18 nonzeros
Model fingerprint: 0x90cc2da6
Variable types: 4 continuous, 6 integer (0 binary)
Coefficient statistics:
  Matrix range [6e-02, 1e+01]
  Objective range [1e-02, 1e+00]
  Bounds range [5e+01, 1e+03]
  RHS range [0e+00, 0e+00]
Found heuristic solution: objective -0.0000000
Presolve removed 1 rows and 6 columns
Presolve time: 0.00s
Presolved: 3 rows, 4 columns, 9 nonzeros
Variable types: 0 continuous, 4 integer (0 binary)

Root relaxation: objective 1.149294e+02, 2 iterations, 0.00 seconds

Nodes | Current Node | Objective Bounds | Work
Expl Unexpl | Obj Depth IntInf | Incumbent BestBd Gap | It/Node Time
-----+-----+-----+-----+-----+-----+-----+-----+-----+-----
0 0 114.92942 0 1 -0.00000 114.92942 - - 0s
H 0 0 114.8500000 114.92942 0.07% - 0s
H 0 0 114.9000000 114.92942 0.03% - 0s
0 0 114.92941 0 1 114.90000 114.92941 0.03% - 0s
0 1 114.92941 0 1 114.90000 114.92941 0.03% - 0s

Explored 94 nodes (94 simplex iterations) in 0.01 seconds
Thread count was 8 (of 8 available processors)

Solution count 3: 114.9 114.85 -0

Optimal solution found (tolerance 1.00e-04)
Best objective 1.149000000000e+02, best bound 1.149000000000e+02, gap 0.0000%

Wrote result file 'code/coins.sol'

```

Slika 2: Primer izpisa na standardni izhod pri reševanju problema s kovanci.

<sup>4</sup>Navodila za LP format v Gurobi [https://www.gurobi.com/documentation/9.1/refman/lp\\_format.html#format:LP](https://www.gurobi.com/documentation/9.1/refman/lp_format.html#format:LP).

Koda 2: Generirana datoteka z rešitvami za problem kovancev

```
# Objective value = 114.9
Pennies 0
Nickels 0
Dimes 4
Quarters 50
Halves 0
Dollars 102
Cu 999.3884
Ni 40.8946
Zn 4.9571999999999996e+01
Mn 2.8916999999999998e+01
```

Na standardnem izhodu ukaza lahko vidimo nekaj podatkov o problemu, vidimo da se je problem reševal v 8 nitih in da je potreboval 94 Simplex iteracij za doseženo rešitev \$114.9. V izhodni datoteki z rešitvami pa se nahajajo še količine posameznih kovancev: 4 *Dimes*, 50 *Quarters* in 102 *Dollars*.

Obstaja tudi *Gurobi Interactive Shell*, ki temelji na *Python Shell*. Potek ukazov je zelo podoben programiranju v Python, omogoča pa hitrejšo spreminjanje modela in zaganjanje v primerjavi z ročnim spreminjanjem vhodne datoteke in vnovičnega poganjanja ukazov. Bolj podroben opis je na voljo v uradni dokumentaciji ali v hitrem vodiču [5].

## 2.3 Uporaba v Python

Gurobi ima na voljo tudi vmensik za Python. Slednjega lahko namestimo na tri načine: preko **pip**, ročno ali pa uproabimo **Anaconda**. Ne glede na način, ki ga izberemo bo še vedno potrebna pridobitev in namestitev veljavne licence, kot smo opisali v enem izmed prejšnjih poglavjih. Za potrebe te naloge smo se odločili za namestitev v Python virtualno okolje in uporabo **pip**. Paket se imenuje **gurobipy** in v času pisanja naloge (še) ni bil dostopen v javnem PyPi strežniku. Zato se za namestitev uporabi njihov privaten strežnik in ukaz:

```
python3 -m pip install -i https://pypi.gurobi.com gurobipy
```

Ročna namestitev se izvede s pomočjo **setup.py**, ki se nahaja znotraj mape kamor smo namestili Gurobi med pridobivanjem licence. Pri obeh omenjenih načinih namestitve je potrebno paziti na kompatibilno verzijo Python (podprte so 2.7 in 3.6 - 3.9 in le 64-bitne različice). Zadnja možnost pa je uporaba **Anaconde**<sup>5</sup>

```
conda config --add channels https://conda.anaconda.org/gurobi
```

```
conda install gurobi
```

Namestitev se lahko preveri tudi s pomočjo **Jupyter notebook**, ki bi moral biti priložen temu poročilu (nahaja se v **code/Gurobi-demo.ipynb**). V kolikor si pa bralec želi igrati s tem Jupyter zvezkom, pa je na voljo tudi **requirements.txt**. V splošnem za zvezek potrebujete **numpy**, **scipy** in **gurobipy** (iz privatnega strežnika), vse drugega so odvisnosti teh paketov. Primer ukazov za macOS/Linux, ki pripravijo okolje in odprejo zvezek za urejanje:

```
python3 -m venv genv
```

```
source genv/bin/activate
```

```
python3 -m pip install -r requirements.txt
```

<sup>5</sup>Anaconda je distribucija Pythona dosegljiva na <https://www.anaconda.com/>

jupyter notebook code/Gurobi-demo.ipynb

Bralec, ki si ne želi urejati zvezek, si lahko tudi ogleda statičen zvezek na Github repozitoriju kjer se nahaja to poročilo<sup>6</sup>. Hiter povzetek uporabe Gurobi v Python skupaj z LP datotekami je viden v kodi 3.

Koda 3: Primer uporabe LP formatov in gurobipy

```
import gurobipy as gp

m = gp.read('coins.lp') # preberi model
m.optimize() # optimiziraj

# izpisi rezultate
for v in m.getVars():
    print(f'{v.varName}={v.x}')
print(f'vrednost: {m.objVal}')
```

## 2.4 Uporaba v Matlab

Prepiši iz Quickstart kako se uporabi v Matlab

## 3 Testna zbirka

Izbri in opisi probleme iz Miplib2017

## 4 Primerjava Gurobi in Matlab

Metodologija testiranja

### 4.1 Testno okolje

Opisi Davids-MBP

### 4.2 Performančni rezultati

Izrisi grafe za zagone in resene probleme

## 5 Zaključek

Povzemi kaj smo naredili, kaj smo ugotovili.

---

<sup>6</sup>Povezava do Jupyter zvezka <https://github.com/rubinda/gurobi-vs-matlab/blob/main/code/Gurobi-demo.ipynb>

# Priloge

## A Podrobni rezultati poskusov

Če je rezultatov v smislu tabel ali pa grafov v nalogi mnogo, predstavi v osnovnem besedilu samo glavne, podroben prikaz rezultatov pa lahko predstaviš v prilogi. V glavnem besedilu ne pozabi navesti, da so podrobni rezultati podani v prilogi.

## B Programska koda

Za domače naloge bo tipično potrebno kaj sprogramirati. Če ne bo od vas zahtevano, da kodo oddate posebej, to vključite v prilogo. Čisto za okus sem tu postavil nekaj kode, ki uporablja Orange (<http://www.biolab.si/orange>) in razvrščanje v skupine.

## Literatura

- [1] GLEIXNER, A., HENDEL, G., GAMRATH, G., ACHTERBERG, T., BASTUBBE, M., BERTHOLD, T., CHRISTOPHEL, P. M., JARCK, K., KOCH, T., LINDEROTH, J., LÜBBECKE, M., MITTELMANN, H. D., OZYURT, D., RALPHS, T. K., SALVAGNIN, D., AND SHINANO, Y. MIPLIB 2017: Data-Driven Compilation of the 6th Mixed-Integer Programming Library. *Mathematical Programming Computation* (2020 [obiskano 26. decembra, 2020]). <http://miplib.zib.de/>.
- [2] GUROBI OPTIMIZATION LLC. *Gurobi Corporate Brochure*, 2020 [obiskano 26. decembra, 2020]. <https://www.gurobi.com/pdfs/Gurobi-Corporate-Brochure.pdf>.
- [3] GUROBI OPTIMIZATION LLC. *Gurobi Optimizer*, 2020 [obiskano 26. decembra, 2020]. <https://www.gurobi.com/products/gurobi-optimizer/>.
- [4] GUROBI OPTIMIZATION LLC. *Our Team - Gurobi*, 2020 [obiskano 26. decembra, 2020]. <https://www.gurobi.com/company/our-team/>.
- [5] GUROBI OPTIMIZATION LLC. *Quick Start Guides*, 2020 [obiskano 26. decembra, 2020]. <https://www.gurobi.com/documentation/quickstart.html>.
- [6] GUROBI OPTIMIZATION LLC. *Model File Formats*, 2020 [obiskano 27. decembra, 2020]. [https://www.gurobi.com/documentation/9.1/refman/model\\_file\\_formats.html](https://www.gurobi.com/documentation/9.1/refman/model_file_formats.html).
- [7] U.S. MINT. *Coin Specifications*, 2019 [obiskano 27. decembra, 2020]. <https://www.usmint.gov/learn/coin-and-medal-programs/coin-specifications>.