

fejezet

Felhasznált technológiák

MILP

A szakdolgozatomban elkészített modellek MILP (Mixed Integer Linear Programming) vegyes egész értékű lineáris programozási modellek.

Egy modell általánosságban a következő részeket tartalmazza:

- paraméterek: állandók
- változók: az optimalizálási folyamat során kapnak értéket
- korlátozások: korlátozó feltételeket kifejező egyenlőtlenségek
- célfüggvény: a modell célja (minimalizálás/maximalizálás)

A vegyes egész értékű lineáris programozás egy elég széles körben használt módszer, ezért többféle szoftveres megoldó (solver) is a rendelkezésünkre áll.[1] Néhány példa a teljesség igénye nélkül:

- CPLEX
- GLPK
- Gurobi

Több nyelv áll rendelkezésre a MILP modellek leírására és vannak olyan eszközök, amik többfajta kiterjesztést is támogatnak. A GLPK (GNU Linear Programming Kit) modellező nyelve a GMPL (GNU MathProg Language), általánosan például a .mod és .dat kiterjesztéseket használja, míg például a CPLEX .lp kiterjesztésű fájljait a Gurobival is lehet futtatni.

A solverek közül azért választottam a GLPK-t, mert ingyenesen hozzáférhető és könnyű használni a modellalkotáshoz, a Gurobit pedig azért, mert hatékonyan lehet vele a komplexebb modelleket is tesztelni.

GMPL

Az általam készített modelleket az AMPL matematikai modellező nyelv szintaxisára épülő GMPL nyelven írtam.[2] A GMPL modelleknek két fő része van: a modell és az adatok. A modell részben lehet meghatározni a paramétereket és a változókat, amikkel a célfüggvényt szeretnénk optimalizálni, illetve itt adhatók meg a feltételek is. Az adat részben a paramétereknek kell értéket szolgáltatni. A modell részben a kimenet formázására is van lehetőség.

glpsol

A glpsol a GLPK csomag részeként egy önálló LP/MIP solver programként működik. Én a fejlesztés során tesztelésre, illetve a Parancssorból használtam arra, hogy .mod kiterjesztésű fájlból .lp kiterjesztésű fájlt hozzak létre, mivel a Gurobi azt már tudja kezelni. Az átalakítást a következő módon lehet megtenni:

```
glpsol --check --wlp pelda.lp --math pelda.mod
```

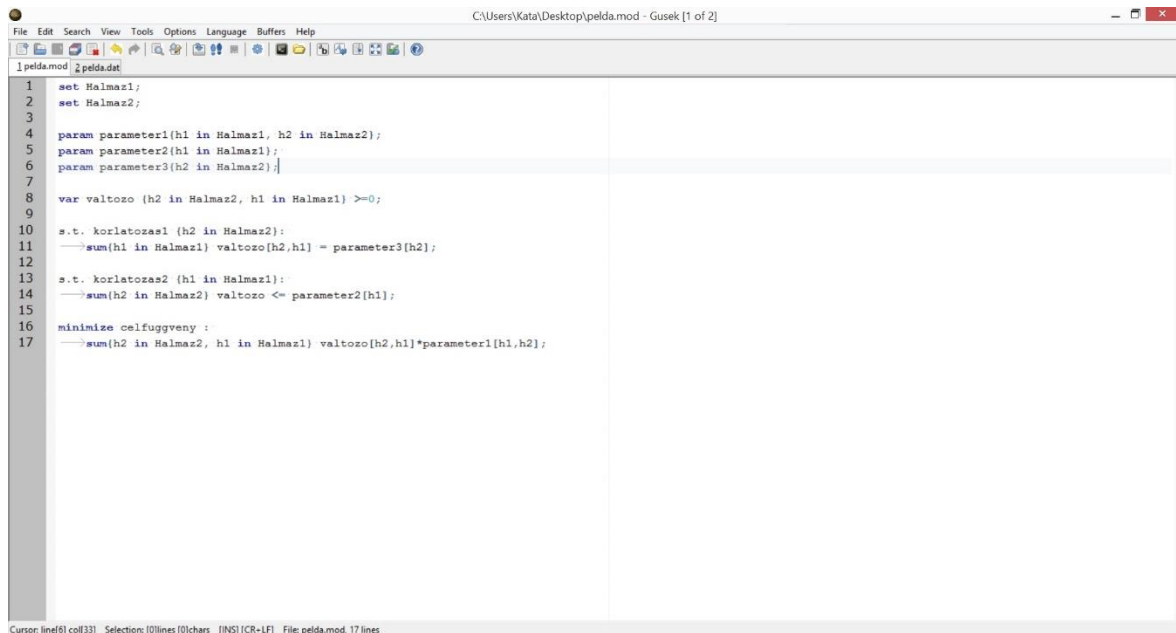
Gusek

A MILP modelljeim leírására a Gusek nevű nyílt forráskódú fejlesztői környezetet használtam. Ez a szoftver támogatja például a GLPK LP/MIP és a CPLEX LP modellek fejlesztését is.

Az alábbi kiterjesztésű fájlokkal dolgoztam a Gusek-ben:

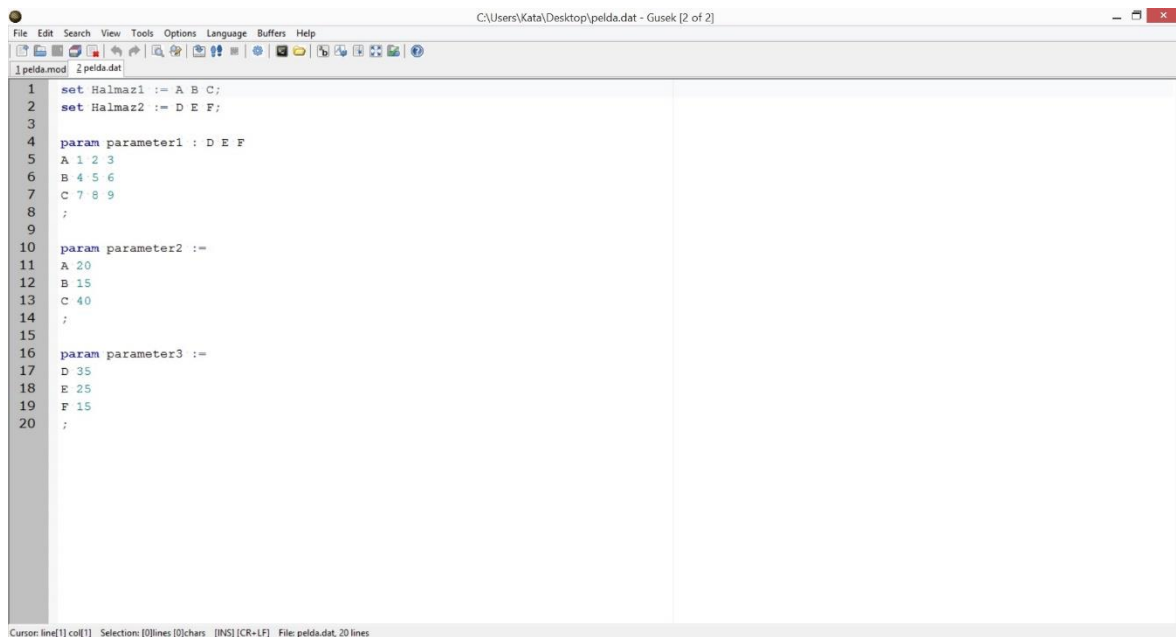
- .mod: GMPL modellfájl
- .dat: GMPL adatfájl
- .out: GLPK kimeneti fájl

Egyidejűleg több fájl is lehet nyitva a Gusek ablakban, amik esetenként csak a kiterjesztésükben különböznek. Az . és a .ábrán egy rövid példával szemléltetek a Gusek fejlesztői környezetben egy GMPL nyelven írt modellt.

The screenshot shows the Gurobi IDE interface with the file 'pelda.mod' open. The code defines two sets, Halmaz1 and Halmaz2, and three parameters: parameter1 (a function of h1 and h2), parameter2 (a function of h1), and parameter3 (a function of h2). It also defines a variable 'valtozo' and two constraints, 'korlatosas1' and 'korlatosas2', which involve sums over the sets. Finally, it defines an objective function 'celfuggveny' to be minimized, which is a sum over the sets.

```
1 set Halmaz1;
2 set Halmaz2;
3
4 param parameter1(h1 in Halmaz1, h2 in Halmaz2);
5 param parameter2(h1 in Halmaz1);
6 param parameter3(h2 in Halmaz2);
7
8 var valtozo (h2 in Halmaz2, h1 in Halmaz1) >=0;
9
10 s.t. korlatosas1 (h2 in Halmaz2):
11     sum(h1 in Halmaz1) valtozo[h2,h1] = parameter3[h2];
12
13 s.t. korlatosas2 (h1 in Halmaz1):
14     sum(h2 in Halmaz2) valtozo[h2,h1] <= parameter2[h1];
15
16 minimize celfuggveny :
17     sum(h2 in Halmaz2, h1 in Halmaz1) valtozo[h2,h1]*parameter1[h1,h2];
```

ábra: .mod fájl a Gurobi IDE-ben

The screenshot shows the Gurobi IDE interface with the file 'pelda.dat' open. The data file provides numerical values for the sets and parameters defined in the .mod file. Halmaz1 is defined as the set of letters A, B, and C, and Halmaz2 as D, E, and F. Parameter1 is a 3x3 matrix of values, parameter2 is a vector of values for A, B, and C, and parameter3 is a vector of values for D, E, and F.

```
1 set Halmaz1 := A B C;
2 set Halmaz2 := D E F;
3
4 param parameter1 : D E F
5 A 1 2 3
6 B 4 5 6
7 C 7 8 9
8 ;
9
10 param parameter2 :=
11 A 20
12 B 15
13 C 40
14 ;
15
16 param parameter3 :=
17 D 35
18 E 25
19 F 15
20 ;
```

.ábra: .dat fájl Gurobi IDE-ben

Gurobi

Főként a végleges modellem teszteléséhez használtam a Gurobi Optimization megoldót, mivel a Gurobi már nem tudott megbirkózni a megnövekedett adatokból fakadó nagyobb modellel. Ez a solver több problématípust is képes kezelni, például a lineáris programozási modelleken kívül a kvadratikus (nemlineáris) programozási modelleket is megoldja.

Többféle modellező és programnyelvet támogat, mint például az AMPL, MATLAB, C, C++, Python, Java, vagy a .NET.

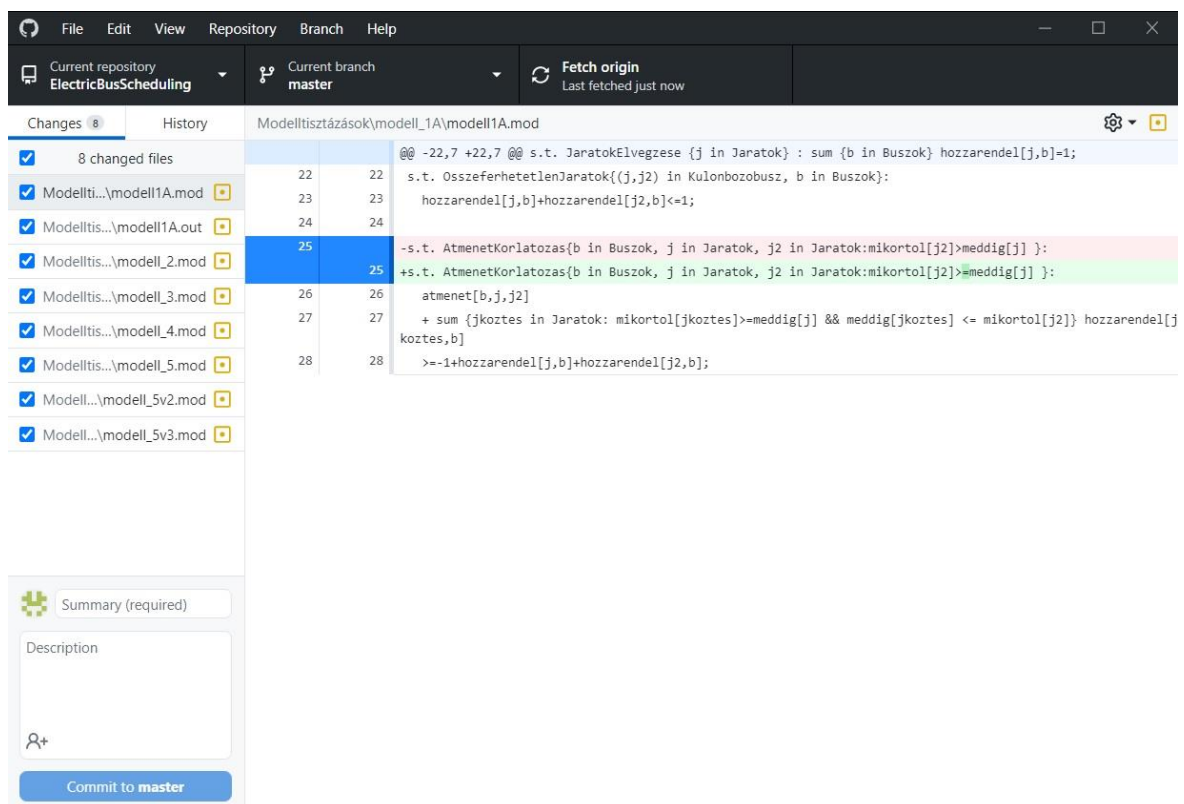
A modelljeim futtatásához a Gurobi parancssori interfészét használtam a `gurobi_cl` paranccsal. A Gusekben fejlesztett `.mod` kiterjesztésű modellfájlokat az előbbieken a `glpsol`-nál megismert módon átalakítottam `.lp` fájlokká, ezután pedig a `ResultFile` paraméter segítségével megadtam, hogy az `.lp` fájl milyen `.sol` kiterjesztésű fájlba mentse az eredményeket. Példa a modell futtatásra:

```
gurobi_cl ResultFile=pelda.sol pelda.lp
```

GitHub

A modellek fejlesztése során a kódokat a GitHub-on tároltam. Ez egy verziókövetési szolgáltatás, ami a Git nevű nyílt forráskódú elosztott verziókezelő rendszeren [3] alapul.

A GitHub Desktopon található helyi repositorym a .ábrán látható.



.ábra: GitHub Desktop

Hivatkozások:

[1]: Linderoth, J. T., & Lodi, A. (2010). MILP software. *Wiley encyclopedia of operations research and management science*.

[2]: <http://gusek.sourceforge.net/gmpl.pdf>

[3]: Loeliger, J., & McCullough, M. (2012). *Version Control with Git: Powerful tools and techniques for collaborative software development*. " O'Reilly Media, Inc."