

Tóth Zsolt

Döntési problémák és a GAMS



Lektorálta: dr. Szűts István

Nyugat-Magyarországi Egyetem
Közgazdaságtudományi Kar
Gazdasági Informatika Intézet

Sopron 2003.

Döntési problémák és a GAMS

TARTALOMJEGYZÉK

1. BEVEZETÉS	3
2. SZÁLLÍTÁSI PROBLÉMA	6
3. A GAMS MODELLEK ÁLTALÁNOS SZERKEZETE	9
4. SETS	10
5. ADATOK	11
5.1. ADATBEVITEL LISTÁVAL	11
5.2. ADATBEVITEL TÁBLÁZATTAL	13
5.3. ADATBEVITEL KÖZVETLEN ÉRTÉKADÁSSAL	13
6. VÁLTOZÓK	14
7. EGYENLETEK	15
7.1. AZ EGYENLETEK DEKLARÁCIÓJA.....	15
7.2. A GAMS ÖSSZEGZŐ ÉS SZORZÓ KIFEJEZÉSEI	16
7.3. EGYENLET DEFINÍCIÓ	17
8. CÉLFÜGGVÉNY	18
9. A MODEL ÉS A SOLVE UTASÍTÁS	18
10. A DISPLAY UTASÍTÁS	19
11. AZ ".LO, .L, .UP, .M" ADATBÁZIS	20
12. A VÁLTOZÓK HATÁRAINAK ÉS KEZDŐÉRTÉKEINEK MEGADÁSA	20
13. A GAMS OUTPUT	21
14. OPERÁCIÓKUTATÁSI PÉLDÁK	21
14. 1. MINIMUMFELADAT MEGOLDÁSA	21
14. 2. KÖLTSÉGMINIMALIZÁSI FELADAT MEGOLDÁSA	22
14. 3. HASZNOSSÁGI FELADAT MEGOLDÁSA	23
14. 4. HOZZÁRENDELÉSI FELADAT MEGOLDÁSA.....	23
14. 5. EGÉSZÉRTÉKŰ PROGRAMOZÁSI FELADAT INPUT FÁJLJA	24
14. 6. EGY KLASSZIKUS MUNKAERŐGAZDÁLKODÁSI MODELL	25
14. 7. EGY OPTIMÁLIS NÖVEKEDÉSI MODELL MEGOLDÁSA	26
14. 8. PORTFOLIÓ-ANALÍZIS	29
FELHASZNÁLT IRODALOM	31
MELLÉKLET	32

Döntési problémák és a GAMS

1. BEVEZETÉS

A GAMS az 1980-as évek vége óta a nyugat-európai és észak-amerikai gazdasági jellegű felsőoktatási intézményekben döntési problémák modellezésére széleskörűen használt, magasszintű, matematikai modellező nyelv.

A nemzetközi szakirodalomban általános nézet szerint a gazdasági felsőoktatásban is elterjedt, hagyományosan strukturált (Fortran, Pascal, C, Basic stb.) és objektumorientált (Visual Basic, C++, Java, Delphi stb.) programnyelvek mellett érdemes a magasszintű, matematikai modellező nyelvek oktatására nagyobb hangsúlyt helyezni, mivel ez utóbbiak az önálló, közgazdasági modellezést már viszonylag rövid tanulás után jelentősen megkönnyítik. A GAMS nyelv alapjainak megismertetése tehát nem tűnik haszontalannak, különösen igaz ez az informatikai tárgyakat magasabb óraszámban hallgató diákokra, hiszen a döntéstámogatás minden gazdasági informatikai tantárgyprogramban kiemelt szerepet játszik.

Mielőtt rátérnénk a GAMS nyelv sajátosságaira, érdemes röviden tisztázni azt, hogy a mit értünk szakértői rendszer és döntéstámogató rendszer alatt.

A *szakértői rendszerek* olyan intelligens rendszerek, amelyek egy adott területen képesek emberi szakértőket kiváltani vagy részben helyettesíteni, s szakértői döntéseket támogatni illetve automatizálni. Felhasználásuk igen sokrétű, többek között sikeresen alkalmazhatók az alábbi területeken:

- diagnózisok készítése – betegségek felismerése
- hibakeresés – mechanikai rendszerek hibás alkatrészeinek megtalálása
- adatok interpretálása – kémiai molekulák szerkezetének meghatározása
- előrejelzések készítése – projekttervezés
- oktatás és tréning - adótanácsadás

A szakértői rendszerek általában az alábbi komponenseket tartalmazzák:

- tudásbázis, azaz HA ... AKKOR típusú szabályok összessége,
- adatbázis, amely egy adott terület ismereteit, tényeit tartalmazza,
- következtető modul (*inference engine*), amely tények és szabályok felhasználásával következtetéseket képes levonni,
- magyarázó modul, amely megmagyarázza, hogy a szakértői rendszer hogyan jutott egy bizonyos következtetésre, azaz milyen következtetéssorozat vezetett az eredményhez,
- tudásszervező modul, amely új információkat képes gyűjteni az emberi szakértőktől, illetve külső adatbázisokból,
- felhasználói illesztés, amely az ember – gép kommunikációt végzi.

A *döntéstámogató rendszerek* (DTR) olyan interaktív rendszerek, amelyek mind az egyéni, mind a csoportos döntéshozatalt támogatják, amikor a probléma csak részben vagy egyáltalán nem strukturált.

Döntési problémák és a GAMS

A döntéstámogató rendszerek legfontosabb feladatai az alábbiak:

- modellalkotás,
- érzékenységvizsgálat,
- célérték keresése,
- kockázatelemzés,
- grafikus elemzés.

A döntéstámogató rendszerek legfontosabb jellemzői:

- strukturálatlan vagy kevésbé strukturált problémák megoldására alkalmasak,
- kifinomult elemző és modellező eszközöket használnak,
- könnyen adaptálhatók, rugalmasak.

A GAMS (General Algebraic Modeling System) programnyelvet a Világbank megbízásából Alexander Meeraus és munkatársai fejlesztették ki az 1980-as évek végén. A döntéstámogatáson belül elsősorban klasszikus operációkutatási feladatokra használják szerte a világon, de a különböző, ma már a Világhálón is hozzáférhető modellkönyvtárakban (pl. <http://www.gams.com/modlib>) számos, a magyar viszonyok között is adaptálható, jórészt a Világbank által készített vagy készített modell egyszerűsített változatát találjuk elsősorban az alábbi témakörökben:

- **makrogazdaságtan:** optimális növekedés, beruházási modellek, ágazati modellek
- **mikrogazdaságtan:** kínálati modellek, árdiszkrimináció, piaci helyzetek
- **menedzsment:** just-in-time szállítás, munkaerő-gazdálkodás, szervezetek kialakítása, szervezeti munka hatékonysága
- **mezőgazdaság és agrárközgazdaságtan:** a fejlődő világ mezőgazdasági és élelmezési problémái, általános mezőgazdasági modellek, öntözési és talajfelhasználási problémák
- **általános egyensúly:** a fejlődő (pl. India, Kamerun), az utóbbi évtizedekben gyors gazdasági fejlődési pályára állt (pl. a Koreai Köztársaság, Kína) és a különböző okokból az elmúlt időszakban gazdasági szerkezetváltásra kényszerült fejlett országok (pl. Finnország) egyensúlyi problémái, adózási kérdések, a versenyt korlátozó piaci helyzetek, globális egyensúlytalanság, a gazdasági integrációk által generált egyensúlyi problémák (pl. Nagy Britannia)
- **ökonometria:** regresszió, maximum likelihood becslés
- **gazdasági fejlődés:** strukturális problémák, szűk keresztmetszetek, új erőforrások, termelési tényezők helyettesítése
- **energiagazdaságtan:** nemzetközi földgáz- és kőolaj-kereskedelem, megújuló energiaforrások, a fejlett, de sok energiát fogyasztó ipari országok problémái (pl. Koreai Köztársaság, Egyesült Államok), az OPEC optimális árképzése, az energiatermelés és felhasználás környezetszennyező hatásai
- **pénzügy:** portfólió-elemzés, pénzügyi tervezés, kockázatkezelés
- **nemzetközi kereskedelem:** rézkereskedelem, alumíniumkereskedelem, szénhidrogének kereskedelme, észak-déli kereskedelem és tőkeáramlás

Döntési problémák és a GAMS

- **műszaki tudományok:** vegyészet, csővezetékek és szállítási útvonalak tervezése, alkatrészszállítás, vízi utak biztonságos hasznosítása, járművek szárnytervezése, optimális felületek kialakítása, árvízvédelem, szivattyú hálózatok tervezése, hulladékanyag-minimalizálás
- **erdőgazdálkodás:** fakitermelés és erdőtelepítés, az erdőgazdálkodás környezeti terhei, a trópusi és az északi erdők megóvása
- **sztochasztikus programozás:** optimális beruházás, repülőjáratok tervezése
- **statisztikai, matematikai alkalmazások**

A fent említett példák is mutatják, hogy a GAMS meglehetősen széles körben használható programnyelv, azonban az összetettebb modellek kialakítása elég bonyolult, a legtöbb modell leegyszerűsített változatának bemutatása is meghaladná az egyetemi oktatás kereteit, kezdetnek azonban néhány egyszerűbb modell áttekintése is elegendő. Természetesen a GAMS programnyelv - a világbanki preferenciák ellenére - teljesen semleges fejlesztőeszköz, mindegy, hogy - a közgazdasági modellezésnél maradván - milyen közgazdasági iskola gondolatain nyugvó modellt kívánunk a segítségével megfogalmazni.

A jegyzetben használt GAMS 20.7-es Student (leegyszerűsített, kevesebb modellt és változót használó) változata a washingtoni GAMS Development Corporation hivatalos oldaláról regisztráció után ingyenesen letölthető (<http://www.gams.com>), de a cég giesseni európai központjának a honlapján is elérhető. (<http://www.gams.de>) Az ingyenes változat is rendkívül hasznos fejlesztőeszköz, jómagam a Világhálón nagyon sok, egymástól rendkívül eltérő fejlettségű és kultúrájú ország egyetemén és kutatóintézeteiben megírt GAMS kódra találtam. (Pl. Svédország, Irán, Németország, Kazahsztán, Spanyolország, Törökország - meglehetősen sokszínű országcsoport.) A GAMS „ingyenessége” ellenére a magyar honlapokon 2003 elején a legismertebb keresőprogramok segítségével sem sikerült GAMS modellekre találnom, bár történtek arra utalások, hogy a GAMS nyelvet itt-ott egy-két órában oktatják.

A jegyzet első felében egy példaalkalmazáson keresztül rövid, de a teljességre törekvő áttekintést nyújtok a GAMS programnyelv legfontosabb jellegzetességeiről. A GAMS iránt mélyebben érdeklődő olvasók Richard E. Rosenthal *GAMS User's Guide* c. művében teljes leírást találnak, azonban a példaalkalmazás megértése után bárki képessé válhat a különböző, gyakrabban előforduló lineáris programozási feladatok gyors megoldására anélkül, hogy a meglehetősen sokrétű GAMS programnyelv használatát teljes egészében elsajátítaná.

A jegyzet második felében néhány jól ismert operációkutatási alkalmazáson kívül bonyolultabb közgazdasági modellek leegyszerűsített változatai közül mutatok be néhányat.

2. SZÁLLÍTÁSI PROBLÉMA

Példaalkalmazásunk a lineáris programozás Dantzig munkássága nyomán klasszikussá vált szállítási problémája, amely az optimalizációs technológia fejlődésében betöltött szerepe és a gyakori hivatkozások révén az operációkutatási feladatok "állatorvosi lovának" is tekinthető. A GAMS és a hozzá hasonló matematikai modellező nyelvek erősségeinek illusztrálására is kiválóan alkalmas, mivel a szállítási probléma, függetlenül az adott modell nagyságától, egyszerű, átlátható algebrai szerkezettel rendelkezik. A későbbiekben majd nyilvánvalóvá válik, hogy nagyobb és bonyolultabb szállítási probléma esetén, az egyszerűbb szállítási feladatra vonatkozó, a GAMS bemeneti fájljában megfogalmazott állításaink alig változnak.

A klasszikus szállítási problémánál megadjuk egy árucikk különböző gyárakban termelt és a helyi piacokon keresett mennyiségét, ezenkívül az egyes gyárak és piacok közötti szállítási egységkötségek is adottak. Hatékonysági kérdésünk a következő: Mennyit szállítsunk az egyes gyárak és piacok között úgy, hogy a szállítási összköltség minimális legyen?

A probléma matematikai modelljét az alábbi formában írhatjuk le:

Halmazok:

i = gyárak
 j = piacok

Exogén változók:

a_i = i gyár árukínálata (szállítóládában megadva)
 b_j = j piac árukereslete (szintén szállítóládában megadva)
 c_{ij} = egységnyi szállításra eső költség i gyár és j piac között (pénzegység/láda)

Döntési (endogén) változók:

x_{ij} = szállításra kerülő árumennyiség i gyár és j piac között (szállítóládában)
ahol $x_{ij} \geq 0$, minden i, j -re

Korlátok:

Kínálati korlát i gyárban: $\sum_j x_{ij} \leq a_i$, minden i -re
Kielégítendő kereslet j piacon: $\sum_i x_{ij} \leq b_j$, minden j -re

Célfüggvény:

$$\sum_i \sum_j c_{ij} x_{ij} \rightarrow \text{minimum}$$

A fenti matematikai modellen alapuló GAMS példánk megmutatja a GAMS modellek néhány olyan alapvető követelményét, amelyet a gyakorlat során érdemes mindig betartani:

1. A modell elemeinek meghatározása és csoportosítása típusok szerint történik.
2. Az elemeknél be kell tartanunk azt a szabályt, hogy semmire sem hivatkozhatunk anélkül, hogy azt korábban ne definiáltuk volna.
3. Az elemek összetevői kötöttek, tehát meghatározott rend szerint kell őket megadnunk.

Döntési problémák és a GAMS

4. A számadatokat, amelyeket a GAMS megoldó algoritmusai értelmeznek, lehetőleg kis abszolút nagyságrendben kell megadnunk. (Pl. ezer forintban.)

A modellezéssel foglalkozó szakemberek az egyes típuselemekre gyakran különböző elnevezéseket használnak. Például a modell ismert adatait, együtthatóit, paramétereit exogén változóknak, a döntési változókat endogén változóknak is nevezhetjük. A GAMS által alkalmazott terminológia a következő: A halmazokat **sets**, a paramétereiket **parameters**, a döntési változókat **variables**, a korlátokat és a célfüggvényeket pedig **equations** néven azonosítjuk.

A szállítási példa GAMS modellje külsőleg nagyon hasonlít a fenti matematikai modellre. Lényegében a matematikai modell kis átalakításával a fordító által értelmezhető kódhoz jutunk.

A szállítási problémára vonatkozó példánkban két konzervgyárból három helyi piacra szállítunk. A példa azonos Dantzig 1963-ban publikált, mára már klasszikussá vált szállítási feladatával, azonban a modellt kissé „magyarítottuk”. A modell exogén változóit a következő táblázat mutatja:

Gyárak	Szállítási távolság			Kínálat
	Piacok			
	Sopron	Szombathely	Zalaegerszeg	
Kecskemét	302	279	266	350
Szeged	308	354	342	600
Kereslet	325	300	275	

A szállítási távolságot kilométerben adtuk meg, a szállítási költségről pedig azt feltételezzük, hogy egységesen 90 forint /szállítóláda /kilométer. A probléma GAMS kódja a következő:

Sets

i kgyarak / Kmet, Szeged /
j piacok / Sopron, Szhely, Zeg /;

Parameters

a(i) i konzervgyar kapacitasa ladaban
/ Kmet 350
Szeged 600 /
b(j) kereslet j piacon ladaban
/ Sopron 325
Szhely 300
Zeg 275 /;

Table d(i,j) tavolsag kilometerben

	Sopron	Szhely	Zeg
Kmet	302	279	266
Szeged	388	354	342 ;

Scalar f kilometerre eso fuvardij ladankent forintban /90/;

Döntési problémák és a GAMS

Parameter $c(i,j)$ ladankenti szállítási költség forintban;

$$c(i,j) = f*d(i,j);$$

Variables

$x(i,j)$ szállított mennyiség ladaban
z forintban mert osszes szállítási költség ;

Positive variable x ;

Equations

cost celfuggveny

supply(i) i gyar kinalati korlatja

demand(j) j piac kielegitendo kereslete;

cost.. z =e= sum((i,j), c(i,j)*x(i,j)) ;

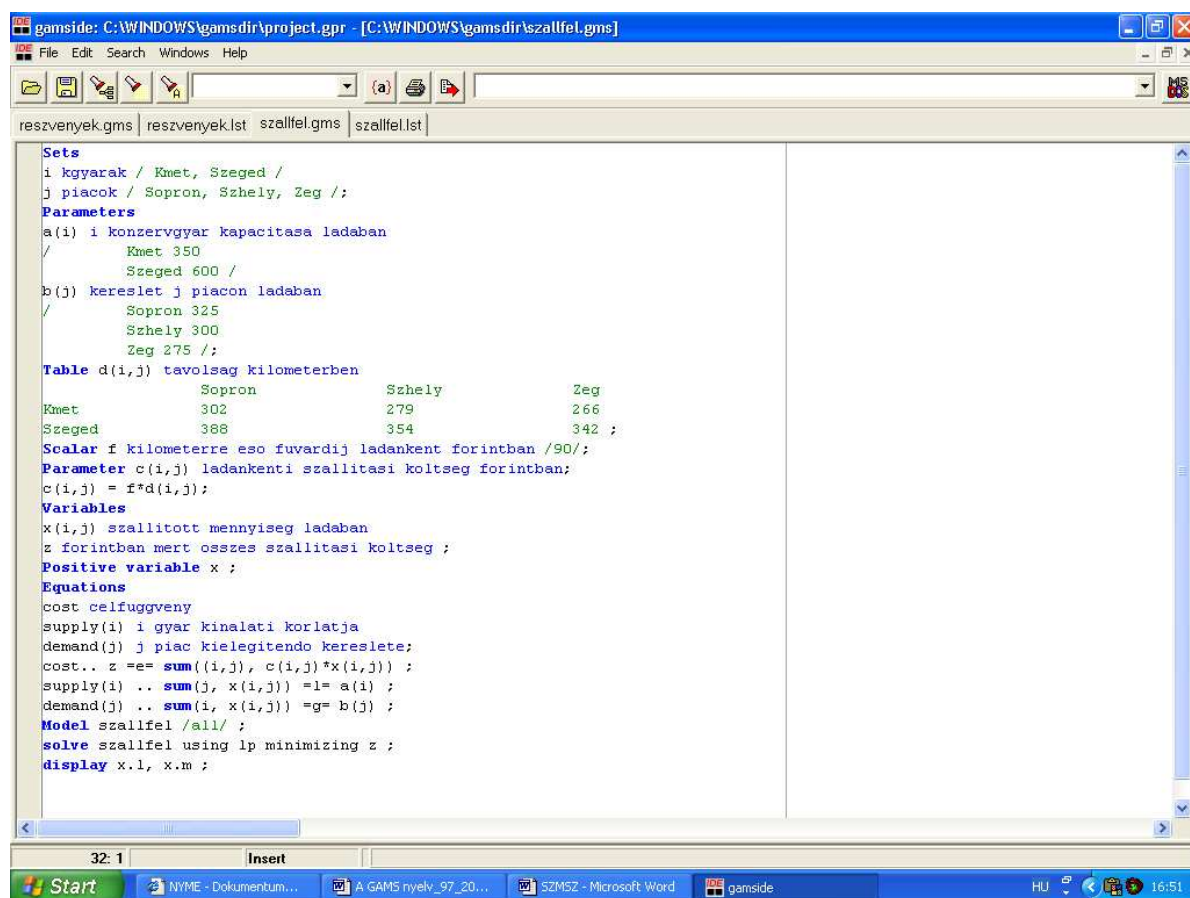
supply(i) .. sum(j, x(i,j)) =l= a(i) ;

demand(j) .. sum(i, x(i,j)) =g= b(j) ;

Model szallfel /all/ ;

solve szallfel using lp minimizing z ;

display x.l, x.m ;



```
gamside: C:\WINDOWS\gamsdir\project.gpr - [C:\WINDOWS\gamsdir\szallfel.gms]
File Edit Search Windows Help
reszvenyek.gms | reszvenyek.lst | szallfel.gms | szallfel.lst

Sets
i kgyarak / Kmet, Szeged /
j piacok / Sopron, Szhely, Zeg /;

Parameters
a(i) i konzervgyar kapacitasa ladaban
/
Kmet 350
Szeged 600 /
b(j) kereslet j piacon ladaban
/
Sopron 325
Szhely 300
Zeg 275 /;

Table d(i,j) tavolsag kilometerben
      Sopron      Szhely      Zeg
Kmet   302         279         266
Szeged 388         354         342 ;

Scalar f kilometerre eso fuvardij ladankent forintban /90/;
Parameter c(i,j) ladankenti szállítási költség forintban;
c(i,j) = f*d(i,j);

Variables
x(i,j) szállított mennyiség ladaban
z forintban mert osszes szállítási költség ;

Positive variable x ;

Equations
cost celfuggveny
supply(i) i gyar kinalati korlatja
demand(j) j piac kielegitendo kereslete;
cost.. z =e= sum((i,j), c(i,j)*x(i,j)) ;
supply(i) .. sum(j, x(i,j)) =l= a(i) ;
demand(j) .. sum(i, x(i,j)) =g= b(j) ;

Model szallfel /all/ ;
solve szallfel using lp minimizing z ;
display x.l, x.m ;
```

2.1 ábra A szállítási modell bemeneti fájlja GAMSIDE 20.7 Student felületen

Döntési problémák és a GAMS

3. A GAMS MODELLEK ÁLTALÁNOS SZERKEZETE

A jegyzet hátralévő részében, a GAMS alapvető komponenseit elemezzük, s minden elemnél visszautalunk a már megoldott szállítási példánkra.

Az input fájl részei	Az output fájl részei
<ul style="list-style-type: none">• <i>Sets</i> Deklaráció A tagok felsorolása, meghatározása• Adatok (<i>Parameters, Tables, Scalar</i>) Deklaráció Értékkadás• <i>Variables</i> Deklaráció Típus megadása• Korlátok és/vagy kezdőértékek megadása (nem kötelező)• <i>Equations</i> Deklaráció Definíció• <i>Model</i> és <i>Solve</i> utasítás• <i>Display</i> utasítás (nem kötelező)	<ul style="list-style-type: none">• Echo Print (visszajelzés)• Referencia térkép• Egyenletlisták• Státusz jelentések• Eredmények

A fentiekén kívül az input fájl több, a professzionális alkalmazásoknál fontos szerepet játszó, választható összetevőt is tartalmaz (hamis adatok ellenőrzése, az eredmények testreszabott megjelenítése, korábbi modellek mentése és helyreállítása, modelltöbbszörözés stb.), azonban a jegyzet terjedelmi korlátai csak az alapvető összetevők bemutatását teszik lehetővé.

Mielőtt az egyes alkotóelemeket részletesen bemutatnánk, néhány általános érvényű észrevételt tennénk:

1. A GAMS modell GAMS nyelvben megfogalmazott utasítások gyűjteménye. Az utasítások sorrendjénél egy szabályt mindig be kell tartani: a modell egyetlen elemére sem hivatkozhatunk az elem előzetes deklarációja nélkül!
2. A GAMS utasításokat tipográfiaiilag szabadon megfogalmazhatjuk. Egy utasítást több sorba, több utasítást egy sorba is írhatunk, azonban tapasztalataink szerint a standard angol szavak karakterkészletétől jobb nem eltérni, pl. az ékezetes betűk hibához vezethetnek.
3. A kezdő GAMS felhasználók jobban teszik, ha minden egyes utasítást pontos vesszővel zárnak le, mint ahogy az a példaalkalmazásban is látható.
4. A GAMS fordító nem tesz különbséget a kis-, és nagybetűs szavak között, bármelyik forma szabadon használható.
5. A megfelelő dokumentáció a matematikai modellek lényeges és hasznos eleme. A magyarázatokat, megjegyzéseket szerencsésebb a modellbe ágyazni, mintsem külön lejegyezni. A GAMS modellben alapvetően két lehetőség is van a magyarázatok beágyazására. A GAMS fordítója figyelmen kívül hagy minden "*" jellel kezdődő sort, pontosabban megjegyzésként kezeli azokat. A másik, talán

Döntési problémák és a GAMS

hasznosabbnak tűnő módszer a magyarázatok GAMS utasításokba történő beszúrása. A szállítási modell kisbetűs szavai a második módszert szemléltetik.

6. Ahogy az input fájl elemeinek felsorolásánál is látható, a GAMS elemek létrehozása két részből áll: deklarációból és definícióból (értékadásból, felsorolásból). A deklaráció itt létrehozást, elnevezést-megnevezést jelent, a definíció során pedig a létrehozott és névvel ellátott "objektumot" értékkel, formával látjuk el. Az *equations* elemtípusnál a deklarációt és a definíciót külön GAMS utasításokkal kell elvégezni. A többi elemtípusnál a deklarációt és a definíciót egy utasításban is elvégezhetjük.
7. Az egyes típuselemek, entitások elnevezéseinek betűvel kell kezdődniük, s maximum tíz karakterből (betűkből és számokból) állhatnak.

4. SETS

A GAMS modellek alapvető építőeleme a *Set*, amely egyértelműen megfeleltethető az algebrai modellek halmazáival. A set-et tömbként is fordíthatnánk, azonban a Parameters vagy a Tables utasítással is tömböket hozunk létre, ezért a fogalmi zavarok elkerülése érdekében maradjunk a „halmaz” kifejezésnél, ill. az angol eredetnél. Szállítási modellünk csak egy set utasítást tartalmaz:

```
Sets
  i kgyarak / Kmet, Szeged /
  j piacok / Sopron, Szhely, Zeg /;
```

Az utasítás eredménye nyilvánvaló. Deklaráltunk két halmazt, és *i* valamint *j* névvel láttuk el őket. Az *i* és a *j* halmaz elemeit is felsoroltuk, amelyek hagyományos matematikai jelöléssel a következők:

```
i = {Kecskemét, Szeged}
j = {Sopron, Szombathely, Zalaegerszeg}.
```

A halmaz elemeinek felsorolásánál figyeljük meg a GAMS utasítások és az általános matematikai formák közötti különbségeket! A GAMS "/" jeleket használ "{}" helyett, mivel a billentyűzetek egy része nem tartalmaz kapcsos zárójelet. A tagok nem állhatnak több szóból, kötőjelet kell használnunk az összefűzésükre.

A fenti sets utasításban szereplő kisbetűs szavakat a GAMS terminológia szerint *text*-nek nevezzük. A text megadása nem kötelező, magyarázatként szolgál, a modell kiegészítő része. A GAMS fordító a text részt nem fordítja le, csak elmenti.

Az egyes set elemtípusokat nem kötelező egyetlen sets-be sűríteni. A példaalkalmazás sets utasítását így is megfogalmazhatjuk:

```
Set   i kgyarak /Kmet, Szeged/;
Set   j piacok  /Sopron, Szhely, Zeg/;
```

Az üres helyek nagysága a kis-, és nagybetűkhöz hasonlóan szabadon megválasztható. A GAMS felhasználók, mint minden programozó, általában saját

Döntési problémák és a GAMS

stilisztikai szabályokat követnek. Pusztán az egyéni ízléstől függ az is, hogy több set-et, vagy egyetlen sets-et használunk, a GAMS nem tesz különbséget a két forma között.

A halmazelemek legfeljebb 10, a magyarázatul szolgáló részek legfeljebb 80 karakterből állhatnak. (Emiatt kénytelen voltam a magyar településnevek rövidítéseit használni.)

Abban az esetben, ha a halmaz elemei szekvenciát alkotnak, a tagok felsorolásánál a "*" jel hasznos segítséget jelenthet:

```
Set t ido periodus      /1991*2000/;  
Set g gepek            /gep1*gep24/;
```

Algebrai jelöléssel:

```
t = {1991, 1992, 1993, ..., 2000}  
g = {gep1, gep2, gep3, ..., gep24}.
```

Meg kell jegyezni, hogy a halmaz elemei karaktersorozatként (sztringként) tárolódnak, ezért pl. t elemei sem szám típusúak.

Az **alias** utasítás segítségével egy korábban már deklarált halmazt új névvel láthatunk el.

A példáink során statikus tömbként kezelt halmazokkal foglalkoztunk, tehát a halmaz elemeit (a tömb tagjait) közvetlenül a felhasználó adja meg, s az elemek az eljárás során már nem változnak. A GAMS azonban a dinamikus tömbök létrehozását is támogatja (lásd GAMS User's Guide).

5. ADATOK

GAMS-ban megfogalmazott szállítási modellünk az adatbevitel mindhárom alapvető formáját - listák, táblázatok, közvetlen értékadás - tartalmazza.

5.1. ADATBEVITEL LISTÁVAL

A listával történő adatbevitel a **Parameters** utasítással történik. Szállítási feladatunk Parameters utasítása a következő volt:

```
Parameters  
a(i) i konzervgyar kapacitasa ladaban  
    /      Kmet  350  
      Szeged 600 /  
  
b(j) kereslet j piacon ladaban  
    /      Sopron 325  
      Szhely  300  
      Zeg    275 /;
```

Döntési problémák és a GAMS

A fenti utasítás két paramétert (külső adatot, exogén változót) deklarált, a és b névvel illette őket, deklarálta a hozzájuk tartozó "domain"-eket (tartományokat). (A tartomány egy halmaz, amelyre paraméter, változó vagy egyenlet/egyenlőtlenség épül.) Az utasítás minden paraméternél magyarázó szöveget is tartalmaz, valamint i és j minden elemére megadja a(i) és b(j) értékeit. Teljesen elfogadott, hogy utasításainkat több utasításba tördeljük, ahogy az alábbiakban is látható:

```
Parameter a(i) i konzervgyar kapacitasa ladaban
/      Kmet 350
      Szeged 600 /;
```

```
Parameter b(j) kereslet j piacon ladaban
/      Sopron 325
      Szhely 300
      Zeg 275 /;
```

Amikor az adatokat listával visszük be, néhány alapvető szabályt érdemes betartani:

1. A tartományok elemeit és a hozzájuk tartozó paraméterek értékeit lényegében tetszés szerint lehet elrendezni. Egy szabályt azonban be kell tartani: a listát perjelek (slash) közé kell tenni és a listaelemeket vesszővel kell elválasztani vagy külön sorba kell írni.
2. A lista elemei, a név, a tartomány és a magyarázó szöveg közé nem kell pontosvesszőt tenni, hiszen az utasítás minden eleme szükséges a deklarációhoz. Ha csak listával akarnánk paramétereket létrehozni, a GAMS hibát jelezne.
3. A GAMS fordító rendelkezik egy rendkívül hasznos vonással, az ún. "domain checking"-gel (tartományellenőrzéssel), amely a tartomány minden eleménél megvizsgálja, hogy része-e a hivatkozott halmaznak. Ha például i halmaz elemei közé a Zeg városnevet helyesen vettük fel, de egy későbbi listában helytelenül Zegerszeg néven hivatkozunk rá, a fordító hibát jelez.
4. Minden paraméter alapértelmezett értéke nulla, ezért az értéklistában csak a nullától különböző értékeket kell feltüntetnünk.
5. A skalár tartomány nélküli paraméter. A **Scalar** utasítás csak egyetlen értéket tartalmaz, ahogy azt a szállítási példánkban is láthattuk:

```
Scalar f kilometerre eso fuvardij ladankent forintban /90/;
```

Ha a paraméter tartománya két vagy több dimenziós, az értékeket akkor is felvehetjük lista formátumban. Ez rendkívül hasznos hiányos (kevés nullától különböző értéket tartalmazó) tömb esetén.

Döntési problémák és a GAMS

5.2. ADATBEVITEL TÁBLÁZATTAL

Az optimalizációs problémáknál gyakran megfigyelhető, hogy a nagyobb, bonyolultabb modellek bemenő adatai viszonylag kis számtáblázatokból származnak, így a GAMS nyelv által lehetővé tett táblázatos adatbevitel megkönnyíti a munkánkat. Szállítási problémánk egy kétdimenziós táblázatot tartalmaz:

Table d(i,j) tavolsag kilometerben

	Sopron	Szhely	Zeg
Kmet	302	279	266
Szeged	388	354	342 ;

Az utasítás deklarálja d paramétert, meghatározza a hozzátartozó tartományt (domain) i és j Descartes-szorzataként. A d-hez tartozó értékeket is az utasításban adjuk meg. A táblázatban esetlegesen kihagyott helyeket a GAMS nullaként értelmezi.

A listával történő adatbevitelhez hasonlóan itt is működik a tartományellenőrzés (domain checking), tehát a GAMS fordítója megvizsgálja, hogy a táblázat sor-, és oszlopnevei tagjai-e a hivatkozott halmaznak/tartománynak. Az összetettebb és a kettőnél több dimenziót tartalmazó táblázatokról a felhasználói kézikönyvekben tájékozódhatunk.

5.3. ADATBEVITEL KÖZVETLEN ÉRTÉKADÁSSAL

A közvetlen értékadással történő adatbevitel alapvetően különbözik a listás és a táblázatos adatbeviteltől, mivel a paraméter deklarálása és az értékadás külön utasításokban történik. Szállítási modellünkben a következő közvetlen értékadás található:

Parameter c(i,j) ladankenti szallitasi koltseg forintban;

$$c(i,j) = f*d(i,j) ;$$

Fontos, hogy első sor végén is pontosvessző található, hiszen e nélkül a GAMS fordítója a két sort egy utasításként próbálná értelmezni. (A GAMS természetesen képtelen lenne értelmezni a kódot, és hibüzenetet küldene.)

Az első utasítás hatására megtörténik c deklarációja és a hozzá tartozó tartomány (domain) kijelölése. Az utasítás magyarázó szöveget is tartalmaz. A második utasítás értéket ad c(i,j)-nek. Természetesen ez az értékadás csak akkor lehetséges, ha f és d(i,j) értékadása már egy korábbi utasításban megtörtént.

A fenti közvetlen értékadás minden c(i,j) elemre vonatkozik. Ha egy konkrét c(i,j) elemnek akarunk értéket adni, akkor a tartományneveket idézőjelbe kell tennünk. Pl.:

$$c('Kmet','Sopron') = 302 ;$$

Ugyanannak a paraméternek természetesen többször is értéket adhatunk. Minden értékadás azonnal felülírja a korábbi értékadást. Ellenben a paramétereket csak

Döntési problémák és a GAMS

egyszer deklarálni, így elkerülhetjük, hogy két eltérő objektumot ugyanolyan névvel illessünk.

Az értékadás során a matematikai kifejezések és beépített függvények gazdag választékából válogathatunk. A programozáshoz egy keveset már értő felhasználóknak nem jelenthet gondot a GAMS használata, azonban a GAMS rendelkezik a más programnyelvekben ismeretlen, de a matematikai programozásban rendkívül hasznos jellemzőkkel. Például, $c(i,j)$ -nek minden (i,j) párra értéket adhatunk ciklusok használata nélkül.

Az alábbi néhány példa a GAMS függvényeinek és standard operátorainak a használatát szemlélteti. A használt paramétereket az értékadás előtt minden esetben deklarálni kell!

```
csquared = sqr(c);
e = m * csquared;
w = 1 / lamda;

eoq(i) = sqrt ( 2 * demand(i) * ordcost(i) / holdcost(i));
t(i) = min (p(i), q(i) / r(i), log (s(i)));
euclidean(i,j) = sqrt (sqr(xi(i) - xi(j) + sqr (x2(i) - x2(j)));
present(j) = future(j) * exp(-interest*time(j));
```

Az összegző és a szorzó operátorokról a segédlet későbbi részében még szó esik.

6. VÁLTOZÓK

A döntési változókat (endogén változókat) a GAMS-ban megfogalmazott modellekben **Variables** utasítással kell deklarálni. Minden változónak nevet kell adni, s ha szükséges, a változót értelmezési tartománnyal (domain) és magyarázattal is elláthatjuk. Szállítási modellünkben a Variables utasításra a következő példát találjuk:

```
Variables
x(i,j) szállított mennyiség ladaban
z forintban mert osszes szállitasi költség ;
```

A fenti utasítás a szállított mennyiséget minden (i,j) párra deklarálni. A z változó nem rendelkezik tartománnyal (domain-nel), mivel skalár mennyiség. Minden optimalizációs GAMS-modell rendelkezik egy z -hez hasonló változóval, amelynek a mennyiségét maximalizálni vagy minimalizálni kell.

Döntési problémák és a GAMS

A deklaráció során minden változónak meg kell határoznunk a típusát. Az alábbi típusokat választhatjuk:

A változó típusa	A változó értelmezési tartománya
Free	$-\infty \dots +\infty$
Positive	$0 \dots +\infty$
Negative	$-\infty \dots 0$
Binary	0 vagy 1
Integer	0,1,..., 100

A változót, amelynek értékét optimalizálni kívánjuk, free típusú skaláris változónak kell kiválasztanunk. Szállítási példánkban, z típusát meghagytuk az alapértelmezés szerinti free típusnál, de x (i,j) változó értelmezési tartományát nemnegatívra tettük.

Positive variable x ;

Jegyezzük meg, hogy x típusának megjelölésénél a tartományt (domain) már nem kell szerepeltetni, a tartomány minden eleme automatikusan felveszi a változó típusát.

A változók alsó és felső korlátairól, kezdőértékeiről a későbbiekben még szó esik.

7. EGYENLETEK

A GAMS és a hozzá hasonló matematikai modellező programnyelvek egyik fő erőssége a modellt alkotó egyenletek és egyenlőtlenségek algebrai struktúrájának rugalmas, szimultán módon történő létrehozásában rejlik.

7.1. AZ EGYENLETEK DEKLARÁCIÓJA

Az egyenleteket külön utasításokkal kell deklarálni, ill. definiálni. A deklaráció lényegében azonos a többi GAMS-elemnél megismertekkel. Az Equations utasításnév után következik az egyenlet neve, a tartomány (domain) és a magyarázó szöveg. Szállítási példánk a következő egyenlet deklaráció(ka)t tartalmazza:

Equations

cost celfuggveny
supply(i) i gyar kinalati korlatja
demand(j) j piac kielegitendo kereslete;

A GAMS-nyelvben az Equations utasítással nemcsak egyenleteket, hanem egyenlőtlenségeket is létrehozhatunk, tartományra (domainre) való hivatkozással vagy anélkül. A továbbiakban az egyenlet szó alatt egyenlőtlenségekre is gondolunk.

Döntési problémák és a GAMS

7.2. A GAMS ÖSSZEGZŐ ÉS SZORZÓ KIFEJEZÉSEI

Mielőtt megismerkednénk az egyenletek definíciójával, érdemes néhány szót ejteni a GAMS-nyelv összegző és szorzó kifejezéseiről. A GAMS-ot úgy tervezték, hogy használata a speciális szimbólumokat nem tartalmazó billentyűzetek használata mellett is kényelmes legyen (ennek igazából csak a nem latin betűs íráshoz tervezett billentyűzetnél van jelentősége.)

A GAMS összegző jelölései egyszerű és összetett kifejezéseknél egyaránt használhatók. Az összegzés alapformája egy két argumentumból álló operátor:

Sum(az összegzés indexe, összegzendő)

A két argumentumot minden esetben vesszővel kell elválasztani, akkor is, ha az első argumentum zárójelben van. A második argumentum bármilyen, összegzést tartalmazó matematikai kifejezés lehet.

Szállítási modellünk az alábbi egyszerű kifejezést tartalmazza:

Sum(j, x(i,j))

Ez ekvivalens a $\sum_j x_{ij}$ kifejezéssel.

Egy kicsit összetettebb példa:

Sum((i,j), c(i,j)*x(i,j))

Hagyományos matematikai jelöléssel: $\sum_i \sum_j c_{ij} x_{ij}$.

Utolsó példánk egymásba ágyazott összegző kifejezéseket szemléltet:

Sum(i, Sum(j, c(i,j)*x(i,j)))

A GAMS-nyelvből a szorzat pontosan ugyanolyan formátumban írható fel, csak Sum helyett Prod operátort kell használnunk. Például a

Prod(j, x(i, j))

ekvivalens a $\prod_j x_{ij}$ kifejezéssel.

Az összegző és a szorzó operátorok a paraméterek közvetlen értékadásánál is használhatók.

scalar totsupply a gyarak osszkinalata
totsupply = sum(i, b(i)) ;

Döntési problémák és a GAMS

7.3. EGYENLET DEFINÍCIÓ

Az egyenlet definíció a GAMS legösszetettebb utasítása. Az egyenlet definíció részei sorrendben a következők:

- Az egyenlet (egyenlőtlenség) neve
- A tartomány (domain)
- A tartományt korlátozó feltétel (nem kötelező)
- A "." jelölés
- Az egyenlet (egyenlőtlenség) baloldala
- Relációs operátor: =l=, =e=, =g=
- Az egyenlet (egyenlőtlenség) jobboldala

A szállítási példa három egyenlet definíciót tartalmaz:

```
cost.. z =e= sum((i,j), c(i,j)*x(i,j)) ;  
supply(i) .. sum(j, x(i,j)) =l= a(i) ;  
demand(j) .. sum(i, x(i,j)) =g= b(j) ;
```

Néhány dolgot érdemes megjegyeznünk:

- Az összetett egyenletek az egész tartományokra (domain-ekre) vonatkozó definíciók miatt viszonylag könnyen felírhatók. Például a fenti demand korlát a j tartomány minden elemére vonatkozik, ahogy ezt a szállítási feladathoz tartozó GAMS outputból származó részlet is megmutatja:

```
demand(Sopron)..  
x(Kmet, Sopron) + x(Szeged, Sopron) =g= 325 ;  
demand(Szhely)..  
x(Kmet, Szhely) + x(Szeged, Szhely) =g= 300 ;  
demand(Zeg)..  
x(Kmet, Zeg) + x(Szeged, Zeg) =g= 275 ;
```

A GAMS hatékony voltát mutatja, hogy egy ilyen meglehetősen kevés változót tartalmazó feladat keresleti korlátját ugyanúgy írjuk fel, mint egy többtízezer változós feladatot. Mindkét esetben elegendő, ha a felhasználó az algebrai modellt megfogalmazza, az egyes korlátokat a GAMS minden esetben hozzáilleszti a modellhez, ahogy azt a fenti output részletben is láthattuk. (Más optimalizációs programnyelvekben már a bemeneti oldalon a fenti, kimeneti oldalból kiemelt egyenleteket, egyenlőtlenségeket kell felírunk.)

- A relációs operátorok jelentése a következő:

```
=l=   kisebb vagy egyenlő  
=g=   nagyobb vagy egyenlő  
=e=   egyenlő
```

Döntési problémák és a GAMS

A "=" és "=e=" szimbólumokat nem szabad összekeverni. Az "=" jelet csak közvetlen értékadásra, a "=e=" szimbólumot csak egyenlet definiálására használjuk.

- A változókat az egyenletek bármelyik oldalán szerepeltethetjük.
- Az egyenlet definíció a GAMS bemeneti oldalán bárhol felírható, de az egyenletben szereplő változókat, paramétereket, s persze magát az egyenletet az egyenlet definiálása előtt deklarálnunk kell. Az egyenleteket nem kell a deklarációnál kialakított sorrendben definiálnunk, bár az áttekinthetőség emellett szól.

8. CÉLFÜGGVÉNY

A GAMS nem rendelkezik külön célfüggvényt szimbolizáló elemmel. A célfüggvény létrehozásához egy free típusú, skaláris változót kell deklarálnunk, a változót egy egyenlet definícióba kell beépítenünk, s ezzel lehetővé válik, hogy a változó betöltse a célfüggvény funkcióját.

9. A MODEL ÉS A SOLVE UTASÍTÁS

A *Model* szó alatt egy egyenletgyűjteményt értünk, tehát kijelöljük azt, hogy az optimalizációs eljárásnál mely egyenleteket kívánjuk felhasználni. A használandó modellt a deklaráció során, a *Model* utasításnevet követően névvel látjuk el, majd zárójelek (perjelek) között felsoroljuk, hogy a modellünknek mely egyenletek a részei. Ha minden korábban definiált egyenletet be kívánunk építeni, a zárójelek közé az *all* szót kell begépelnünk. Szállítási modellünkben egyetlen *model* utasítás szerepel:

```
Model szállfel /all/;
```

Az utasítás talán feleslegesnek tűnhet, de rendkívül hasznos azoknak a haladó szinten lévő GAMS felhasználóknak, akik egyetlen GAMS programban több különböző modellt alkotnak. Fenti utasításunknak ekvivalens a következővel:

```
Model szállfel /cost, supply, demand/;
```

A tartományok természetesen nem szerepelnek az egyenletek nevei után.

Miután a modellünket megalkotunk, nincs más hátra, mint hogy meghívjuk a szükségesnek ítélt solvert (megoldó eljárást). A meghívás a Solve utasítással történik, ahogy az a szállítási modellünkben is látható:

```
solve szállfel using lp minimizing z ;
```

A solve utasítás a következő formában írható fel:

1. Solve utasításnév
2. A megoldandó modell neve

Döntési problémák és a GAMS

3. A using kulcsszó
4. Az alkalmazandó megoldó (solver) eljárás megnevezése
 - lp - lineáris programozás
 - nlp - nemlineáris programozás
 - mip - kevert egészértékű programozás
 - minlp - kevert egészértékű nemlineáris programozás
 - stb. (számos egyéb megoldó eljárás alkalmazható)
5. A minimizing vagy maximizing kulcsszó
6. Az optimalizálni kívánt változó

10. A DISPLAY UTASÍTÁS

A .gms kiterjesztésű input fájl elkészítése után az állományt lefuttatjuk. A GAMS fordító létrehoz egy .lst kiterjesztésű output fájlt, amely tartalmazza a modellünk megoldását, ill. minden, a modellel kapcsolatos információt. A modell szintaktikai hibái, a hibákhoz tartozó számkódokkal egy külön fájlban jelennek meg. Az output fájl meglehetősen összetett adathalmazából a display utasítás segítségével érdemes néhány adatot kiemelni. Esetünkben a

Display x.l, x.m;

utasítás x(i,j) primál- (.l) és duálváltozóit (.m) jeleníti meg könnyen áttekinthető formában.

```
GAMS Rev 133 Copyright (C) 1987-2002 GAMS Development. All rights reserved
Licensee: GAMS Development Corporation, Washington, DC G871201:0000XX-XXX
Free Demo, 202-342-0180, sales@gams.com, www.gams.com DC9999

--- Starting compilation
--- SZALLFEL.GMS(32) 1 Mb
--- Starting execution
--- SZALLFEL.GMS(18) 2 Mb
--- Generating model szallfel
--- SZALLFEL.GMS(31) 2 Mb
--- 6 rows, 7 columns, and 19 non-zeros.
--- Executing BDMLP

BDMLP 1.3 Jun 14, 2002 WIN.00.NA 20.7 057.046.040.VIS

Reading data...
Work space allocated -- 0.03 Mb

Iter Sinf/Objective Status Num Freq
 1 9.00000000E+02 infeas 3 1
 7 2.66850000E+07 nopt 0

SOLVER STATUS: 1 NORMAL COMPLETION
MODEL STATUS : 1 OPTIMAL
OBJECTIVE VALUE 26685000.

--- Restarting execution
--- SZALLFEL.GMS(31) 0 Mb
--- Reading solution for model szallfel
--- SZALLFEL.GMS(32) 2 Mb
*** Status: Normal completion
```

2.1 ábra A szállítási modellhez kapcsolódó futási információk GAMSIDE 20.7 Student felületen

Döntési problémák és a GAMS

11. AZ ".LO, .L, .UP, .M" ADATBÁZIS

A GAMS tartalmaz egy kis adatbázist, ahol a GAMS minden egyes változóhoz és egyenlethez egy rekordot rendel hozzá. Minden rekord négy mezőt tartalmaz:

.lo	=	alsó határ
.l	=	primálváltozó
.up	=	felső határ
.m	=	duálváltozó

A fenti mezőkhöz a felhasználónak teljes olvasási/írási jogosultsága van. Ez elsősorban a professzionális alkalmazásoknál előnyös.

12. A VÁLTOZÓK HATÁRAINAK ÉS KEZDŐÉRTÉKEINEK MEGADÁSA

A változók alsó és felső korlátja a változó típusából (free, positive, negative, binary, integer) automatikusan következik, de ezeket a korlátokat a GAMS felhasználó bármikor felülírhatja. Néhány lehetséges kódrészlet:

```
x.up(i,j) = capacity(i,j) ;  
x.lo(i,j) = 10.0 ;  
x.up('Kmet','Sopron') = 1.2*capacity('Kmet','Sopron') ;
```

Természetesen azt feltételezzük, hogy az első és a harmadik példában lévő capacity(i,j) paramétert már korábban deklaráltuk és értékkel láttuk el. Ezeket az utasításokat a változó deklaráció után, de a Solve utasítás előtt kell elhelyeznünk. A jobboldalon minden matematikai kifejezés felhasználható.

Nemlineáris programozási feladatoknál nagyon fontos, hogy a modellező a lehető legszűkebb értelmezési tartományt állítsa be az alsó és/vagy a felső korlát segítségével, és megkönnyítse ezzel a megoldó eljárás (solver) munkáját. Nagyon hasznos, ha megadunk egy, a várható megoldástól nem túlságosan messze eső kezdőértéket az .l és vagy az .m mezők beállításával, amely az optimum keresésénél kiindulási alapul szolgálhat. Egy lehetséges kódrészlet:

```
quantity.l(i) = 0.5*eoq(i) ;
```

(Az alapértelmezett kezdőérték általában nulla. Ha a nulla nem része az alsó és felső korláttal beállított értelmezési tartományunknak, akkor az alapértelmezett kezdőérték a nullához legközelebb eső érték.)

Az .lo és .up mezők teljes egészében a GAMS felhasználók felügyelete alatt állnak. Ezzel szemben az .l és az .m mezők kezdeti értékét ugyan a felhasználó állítja be, azonban a mezőket az aktuális megoldó eljárás (solver) szabályozza, végső értékük a modell és a megoldó eljárás függvénye. (Felhívnánk a figyelmet arra, hogy a korábbi display utasításnál az .l és az .m mezők a primál és duál változók megjelenítését szolgálta!)

Döntési problémák és a GAMS

13. A GAMS OUTPUT

A GAMS output (kimeneti fájl) része nagyon sok elemből áll, és meglehetősen informatív. Az egyes összetevők részletes tárgyalását a terjedelmi korlátok miatt elhagyjuk, a különböző felhasználói kézikönyvek részletesen foglalkoznak a kimeneti fájl szerkezetével.

14. OPERÁCIÓKUTATÁSI PÉLDÁK

A GAMS nyelvet elsősorban lineáris programozási feladatok megoldására hozták létre, azonban mostanra a GAMS nyelv segítségével minden operációkutatási feladatot megoldhatunk, az alábbiakban ezek közül mutatok be néhányat.

14.1. MINIMUMFELADAT MEGOLDÁSA

Az operációkutatásból jól ismert minimumfeladat megoldására az alábbi GAMS program nyújt példát. A minimumfeladatnál, az előző szállítási feladathoz hasonlóan lp (lineáris programozás) solvert (megoldó eljárást) alkalmazunk. Talán nem felesleges felírni a lineáris programozási feladatok matematikai alakját.

$$\begin{array}{ll} \text{Min. vagy max.} & \mathbf{cx} \\ \text{Feltételek:} & \mathbf{Ax} \geq \mathbf{b} \\ & \mathbf{L} \leq \mathbf{x} \leq \mathbf{U} \end{array}$$

ahol \mathbf{x} változóvektor valós szám, \mathbf{cx} a célfüggvény, $\mathbf{Ax} \geq \mathbf{b}$ a feltételeket, \mathbf{L} és \mathbf{U} a változók alsó és felső határait jelöli. (L általában 0) A konkrét feladat matematikai modelljét - egyszerűsége miatt - a GAMS bemeneti fájl megjegyzései közé szúrtam be. Minimumfeladatunk megoldása tehát a következő:

```
*MINIMALIZALNI KELL AZ
*F(x)= 2*x1 - x2
*KORLATOZO FELTETEEK:
*x1-2x2-2=0
*x1+x2-2>=0
*2x1 + 3x2 -12<=0
*x1>0
*x2>0

VARIABLES X1, X2, F;

POSITIVE VARIABLES X1, X2;

EQUATIONS CELFGV, EGYENL1, NAGYOBB1, KISEBB1;

CELFV.. F =E= 2*X1 - X2;
EGYENL1.. X1-2*X2 =E= 2;
NAGYOBB1.. X1+X2 =G= 2;
KISEBB1.. 2*X1 + 3*X2 =L= 12;

MODEL OPTIM /ALL/;

SOLVE OPTIM USING LP MINIMIZING F;

DISPLAY F.L;
```

Döntési problémák és a GAMS

```
DISPLAY X1.L;  
DISPLAY X2.L;  
DISPLAY CELFGV.L;  
DISPLAY EGYENL1.L;  
DISPLAY KISEBB1.L;
```

Természetesen a maximumfeladatok, normálfeladatok, módosított normálfeladatok, általános feladatok megoldásánál a fenti kódrészlet kis módosítással felhasználható.

14.2. KÖLTSÉGMINIMALIZÁSI FELADAT MEGOLDÁSA

Következő példánk a mikrogazdaságtanból jól ismert Cope-Douglas termelési függvényt feltételező termelési költségminimalizálási nlp (nemlineáris) programozási példa.

A nemlineáris problémák matematikai alakja:

$$\begin{array}{ll} \text{Min.} & f(\mathbf{x}) \\ \text{Feltételek:} & \mathbf{g}(\mathbf{x}) \leq 0 \\ & \mathbf{L} \leq \mathbf{x} \leq \mathbf{U} \end{array}$$

ahol \mathbf{x} változóvektor valós szám, $f(\mathbf{x})$ a célfüggvény, $\mathbf{g}(\mathbf{x})$ a korlátokat, \mathbf{L} és \mathbf{U} a változók alsó és felső határait jelöli.

A feladat megoldásánál egyedül a hatványra emelésnél használt '**' jelölés jelenthet újdonságot. A konkrét feladat matematikai modelljét - egyszerűsége és a mikrogazdaságtanból való ismertsége miatt - a GAMS bemeneti fájl megjegyzései közé szúrtam be:

```
*K,L,C TERMELESI KTSG MINIMALIZALAS  
*MIN C= 4*K + 5*L  
*KORLAT 10*K(HATVANY 0.5) * L(HATVANY 0.5) >= 1000  
* K>=10 L>=10  
  
VARIABLES K, L, C;  
  
POSITIVE VARIABLES K,L;  
K.L=10;  
L.L=10;  
  
EQUATIONS  
OBJ KOLTSEGFUGGVENY  
P TERMELES MENNYISEGE;  
  
OBJ.. C =E= 4*K + 5*L;  
P.. 10*(K**0.5 * L**0.5) =G= 1000;  
  
MODEL MINOBJ /ALL/;  
  
SOLVE MINOBJ USING NLP MINIMIZING C;
```

14.3. HASZNOSSÁGI FELADAT MEGOLDÁSA

Kéttermékes hasznossági modellünk szintén a mikro gazdaságtanból átemelt nemlineáris programozási példa. A Solver munkájának könnyítése érdekében X és Y változókhoz alsó korlátokat (határokat) rendeltünk. Ez nemlineáris programozási feladatoknál gyakori. A hasznossági modell a standard mikroökonómiából ismert, így matematikai modelljét elhagytam.

```
VARIABLES
X
Y
U;
X.L=1;
Y.L=1;

EQUATIONS

OBJ HASZNOSSAG FUGGVENY
RP KOLTSEGVETESI KORLAT;

OBJ.. U =E= (X+2)*(Y+1);
RP.. 4*X + 6*Y =E= 130;

MODEL HASZNMAT /OBJ, RP/;

SOLVE HASZNMAT USING NLP MAXIMIZING U;
```

14.4. HOZZÁRENDELÉSI FELADAT MEGOLDÁSA

Hozzárendelési feladatunk a szállítási feladattal rokon lineáris programozási feladat. Általában hozzárendelési probléma címen foglalják össze azokat a szállítási feladatokat, amelyekben a feladóhelyek mindegyikén egyetlen termék elszállításáról van szó és a felvevőhelyek igénye is egyetlen termék.

Klasszikus példa erre a következő feladat (a hozzárendelési feladat legegyszerűbb formája): n számú feladat elvégzésére n munkás áll rendelkezésre úgy, hogy egy munkás csak egy feladatot végez el, és egy feladattal csak egy munkás foglalkozik. (A munkások és a feladatok kölcsönösen és egyértelműen egymáshoz lesznek rendelve.) Matematikai jelölésekkel:

c_{ij} = i-edik munkás j-edik feladatot mekkora c_{ij} költséggel végzi
 x_{ij} = i-edik munkás j-edik feladatot elvégzi v. sem

$$\sum_{i=1}^n \sum_{j=1}^n x_{ij} c_{ij} = z \rightarrow \min$$

$$\text{Felt. } \sum_{i=1}^n x_{ij} = 1$$

Döntési problémák és a GAMS

$$\sum_{j=1}^n x_{ij} = 1$$

$$x_{ij}^2 = x_{ij}$$

A GAMS bemeneti fájlhoz tartozó kód:

```
SETS
  GEP /GEP1, GEP2, GEP3, GEP4/
  MUNKAS /MUN1, MUN2, MUN3, MUN4/;

TABLE T(I,J) IDO
      UN1   MUN2   MUN3   MUN4
GEP1  14    5     8     7
GEP2   2    12    6     5
GEP3   7     8     3     9
GEP4   2     4     6    10;

VARIABLES
  X(I,J)
  Z;
POSITIVE VARIABLES X(I,J);

EQUATIONS

OBJ
KORL1 (i)
KORL2 (j);

OBJ.. Z =E= SUM((I,J), T(I,J)*X(I,J));
KORL1(i).. SUM(J, X(I,J)) =E= 1;
KORL2(j).. SUM(I, X(I,J)) =E= 1;

MODEL HOZZAR/ALL/;

SOLVE HOZZAR USING LP MINIMIZING Z;

DISPLAY X,L, Z,L;
```

14. 5. EGÉSZÉRTÉKŰ PROGRAMOZÁSI FELADAT INPUT FÁJLJA

Az egészértékű programozási feladat (MIP solver) matematikai modellje a következő:

A nemlineáris problémák matematikai alakja:

$$\begin{array}{ll} \text{Min.} & f(\mathbf{x}) \\ \text{Feltételek:} & \mathbf{g}(\mathbf{x}) \leq 0 \\ & \mathbf{L} \leq \mathbf{x} \leq \mathbf{U} \end{array}$$

ahol \mathbf{x} változóvektor valós szám, $f(\mathbf{x})$ a célfüggvény, $\mathbf{g}(\mathbf{x})$ a korlátokat, \mathbf{L} és \mathbf{U} a változók alsó és felső határait jelöli. A konkrét feladat matematikai modelljét - egyszerűsége miatt - a GAMS bemeneti fájl megjegyzései közé szúrtam be:

Döntési problémák és a GAMS

```

*EGESZERTEKU PROGRAMOZAS
*MAXIMALIZALNI Z= 4X1+X2
*KORLATOK:
*2X1+X2=<5
*2X1+3X2=5
*X1>=0
*X2>=0
*X1,X2 EGESZ

VARIABLES X1, X2, Z;

INTEGER VARIABLES X1, X2;

EQUATIONS OBJ, EGY1, EGY2;

OBJ.. Z =E= 4*X1 + X2;
EGY1.. 2*X1 + X2 =L= 5;
EGY2.. 2*X1 + 3*X2 =E= 5;

MODEL EGESZLP /ALL/;

SOLVE EGESZLP USING MIP MAXIMIZING Z;

```

14. 6. EGY KLASSZIKUS MUNKAERŐGAZDÁLKODÁSI MODELL

Az alábbi, munkaerő-gazdálkodási modell a klasszikus, Dantzig által publikált modell egyszerűsített változata. A matematikai modell az Equations rész egyenletei-egyenlőtlenségei alapján egyértelmű, külön magyarázatot nem igényel.

Set	i	idoszak (hetben)	/ 1*5 /
Scalars	term	termelekenyseg	(aruegysegben munkasonkent) / 8 /
	kap	oktato kapacitasa	(egy oktatora eso dolgozo) / 6 /
	ber	ber	(euroban dolgozonkent) / 100 /
Parameter	ka(i)	kezdo arukeszlet (aruegysegben)	/ 1 10 /
	kl(i)	alkalmazottak indulo letszama	/ 1 20 /
	vs(i)	vegkielegites (euroban)	/ 5 100 /
	ke(i)	kereslet (aruegysegben)	/ 1 100, 2 200, 3 300 4 400, 5 200 /

Positive Variables

t(i) termelesi szint i idoszakban (aruegysegben)
a(i) arukeszlet i idoszakban (raktar) (aruegysegben)
u(i) kielegitetlen kereslet i idoszakban (aruegysegben)
p(i) potencialis dolgozok szama
h(i) alkalmazásra kerulok szama
f(i) elbocsatottak szama

Variables

c osszkoltseg (euroban)

Equations

ae(i) aruppiaci egyensuly
me(i) munkapiaci egyensuly idoszaki osszehasonlitasban
met(i) munkapiaci egyensuly a termes figyelembevetelevel

Döntési problémák és a GAMS

```
cel    celfuggveny          (euroban);

ae(i).. a(i) =e= a(i-1) + t(i) - ke(i) - u(i-1) + u(i) + ka(i);

me(i).. p(i) =e= p(i-1) - f(i) + h(i) + kl(i);

met(i).. p(i) =g= t(i)/term + (1 + 1/kap)*h(i);

cel.. c =e= sum(i, 10*a(i) + 30*u(i) + (ber + vs(i))*p(i) );

Model munkaero munkaero kepzesi modell / all /;

Solve munkaero minimizing c using lp;
```

14. 7. EGY OPTIMÁLIS NÖVEKEDÉSI MODELL MEGOLDÁSA

Az alábbi, Chakravartytól származó - a Ramsey-modellhez hasonló - növekedési modell bemeneti fájlja a GAMS nyelv újabb eszközeivel, lehetőségeivel ismerteti meg az olvasót. Az új elemek bemutatása előtt az eddigieknél bonyolultabb közgazdasági modellt érdemes matematikai jelölésekkel is felírni. A megoldás során néhány esetben kénytelen voltam a jelölésektől eltérni. Az általam közölt kód a GAMS fordító szempontjából, a matematikai modell közgazdasági-matematikai szempontból tekinthető teljesnek. (Valójában a két modell ekvivalens egymással.) Az egyes jelöléseken túl a GAMS kód elemei is némileg eltérő jelentéssel bírnak. A modell algebrai jelölésekkel a következő:

Y_t = kibocsátás (jövedelem, output) t időszakban
 α = termelési függvény paraméter
 K_0 = tőkeállomány 0 időszakban
 K_t = tőkeállomány t időszakban
 β = tőkekitevő - tőkehatvány
 C_t = fogyasztás t időszakban
 $U(C_t)$ hasznosság t időszakban a fogyasztás függvényében
 τ = hasznossági függvény paramétere - rugalmasság
 J = teljesítmény - kibocsátás
 dis = diszkontfaktor
 ρ = (jóléti) diszkonttényező

Felt. $K_{t+1} = K_t + Y_t - C_t$

$$K_{t+1} = K_t + \alpha K_t^\beta - C_t$$

$$K_n \geq K^*$$

$$U(C_t) = \frac{C_t^{(1-\tau)}}{(1-\tau)}$$

$$J = \sum_{t=0}^n (dis)^t U(C_t)$$

Döntési problémák és a GAMS

$$\text{dis} = \frac{1}{1+\rho}$$

Max. J

A GAMS bemeneti fájlban a bevezető **\$Title** utasítása utáni, címként funkcionáló részt és az **\$Ontext** - **\$Offtext** között elhelyezkedő magyarázó részt a fordító figyelmen kívül hagyja. Az **\$Ontext** - **\$Offtext** között olyan, a megértést szolgáló szöveges részeket helyezhetünk el, amelyeknél a magyar nyelvnek megfelelő közép-európai kódolású betűkészletet, így ékezetes betűket is használhatunk!

A Sets-nél szereplő bázisidőszak és tárgyidőszak dinamikus tömbök definíciója több, eddig nem tárgyalt elemet tartalmaz:

- Az **eq** numerikus reláció operátor két számértéket hasonlít össze, egyezés esetén a visszatérési érték *True*, ellenkező esetben *False*. (Igaz-Hamis, 1-0)
- Alapértelmezés a szerint a *Set* segítségével definiált halmazok elemei számértékkel - még ha az elemek csupa számjegyből is állnak - nem rendelkeznek. Az **ord** operátor az egydimenziós, rendezett elemekből álló, statikus halmaz (tömb) elemeit számértékké alakítja át, mégpedig úgy, hogy visszaadja az elem relatív pozícióját. A példában $t=0$ esetén $\text{ord}(t)=1$, $t=1$ esetén $\text{ord}(t)=2$ stb. Más programnyelvektől eltérően a halmazelemek-tömbelemek sorszámozása nem 0-val, hanem 1-gyel kezdődik. A **card** operátor bármely Set utasítással létrehozott halmaz elemszámát adja vissza.
- A **Yes** kulcsszót (a **No** kulcsszóval együtt) a más programnyelvekből ismert dinamikus tömbök definiálására használhatjuk fel a

`set_név (domain_név | domain_elem) = yes | no ;`

szintaxis alapján. A **Yes** után szereplő **\$** jel segítségével a *Yes* kulcsszót logikai utasítássá alakítjuk, esetünkben pl. az első dinamikus tömbnél - itt most „halmaz” helyett a többi programozási nyelvvel történő összehasonlíthatóság érdekében a „tömb” kifejezést használom - $\text{ord}(t)$ akkor lesz $\text{tb}(t)$ dinamikus tömb tagja, ha $\text{ord}(t)=1$.

- A fentiek alapján $\text{tb}(t)=0$, $\text{tt}(t)=20$.

Növekedési modellünk GAMS bemeneti fájlja tehát a következő:

`$Title Optimalis novekedesi modell`

`$Ontext`

Egyszektoros nemlineáris növekedési modell

Kendrick, D, and Taylor, L, Numerical methods and Nonlinear Optimizing

Döntési problémák és a GAMS

models for Economic Planning. In Chenery, H B, Ed, Studies of Development Planning. Harvard University Press, 1971.

Chakravarty, S, Optimum Savings with a Finite Planning Horizon. International Economic Review 3 (1962), 338-355.

\$Offtext

```
Sets    t      idotav / 0*20 /
        tb(t)  bazisidoszak
        tt(t)  targyidoszak;
```

tb(t) = yes\$(ord(t)eq 1); tt(t) = yes\$(ord(t) eq card(t));

Display tb,tt;

```
Scalars    deltt erteckcsokkenesi rata      / .05 /
           beta  tokehatvany                / .75 /
           a     hatekonysag
           r     munkaeronovekedesi rata    / .025 /
           tau   rugalmassag                / .9 /
           z     technikai haladas         / .01 /
           rho   joleti diszkonttenyezo    / .03 /
           y0    kezdo jovedelem           / 4.275 /
           k0    kezdotoke                  / 15.0 /
```

```
Parameters dis(t) diszkontfaktor
           alfa(t) termelesi fuggveny ;
```

a = y0/k0**beta;
dis(t) = (1+rho)**(1-ord(t))/(1-tau);
alfa(t) = a*(1+r*(1-beta)+z)**(ord(t)-1);
Display a, dis, alfa ;

```
Variables  c(t)  fogyasztas
           y(t)  jovedelem kibocsatas
           k(t)  tokeallomany
           j     makroekonomiai kibocsatas novekedes
```

```
Equations  kb(t) tokemerleg
           yd(t) jovedelem definicio
           jd     makroekonomiai kibocsatas definicio ;
```

jd.. j =e= sum(t, dis(t-1)*c(t-1)**(1-tau));

yd(t).. y(t) =e= alfa(t)*k(t)**beta;

kb(t+1).. k(t+1) =e= y(t) - c(t) + (1-delt)*k(t);

y.l(t) = y0*(1.06)**(ord(t)-1); k.l(t) = (y.l(t)/alfa(t))**(1/beta);

c.l(t) = y.l(t) + (1-delt)*k.l(t) - k.l(t+1); display c.l, k.l, y.l;

k.lo(t) = 1; y.lo(t) = 1; c.lo(t) = 1; y.fx(tb) = y.l(tb); y.fx(tt) = y.l(tt);

Model novekedes / all /;

Solve novekedes maximizing j using nlp;

Döntési problémák és a GAMS

```
Parameter report solution summary ;
report(t,"k") = k.l(t); report(t,"y") = y.l(t); report(t,"c") = c.l(t);
report(t,"s-rate") = (y.l(t)-c.l(t))/y.l(t);
Display report;
```

14. 8. PORTFOLIÓ-ANALÍZIS

A portfólió analízis a GAMS-modellek egyik kedvelt alkalmazási területe, az alábbi modell erre nyújt egy rendkívül leegyszerűsített példát:

i = Részvények (A részvény, B, részvény, C részvény)

B = Befektetésre szánt összeg

R = Elvárt hozam

r_i = i részvény várható hozama

σ_{ij} = i és j részvény hozama közötti kovariancia

x_i = i részvénybe befektetésre kerülő összeg (e Ft)

$$\min \sum_{i=1}^3 \sum_{j=1}^3 x_i x_j \sigma_{ij}$$

$$\text{felt. } \sum_{i=1}^3 x_i \leq B,$$

$$\sum_{i=1}^3 r_i x_i \geq R,$$

$$x_i \geq 0$$

$$B = 1000.00, \quad R = 50.00$$

σ_{ij}	1. részvény	2. részvény	3. részvény
1. részvény	0.017087987	0.003298885	0.001224849
2. részvény	0.003298885	0.005900944	0.004488271
3. részvény	0.001224849	0.004488271	0.063000818
r_i	0.026	0.008	0.074

A feladat GAMS kódja:

```
sets i/1*3/
alias (i,j);
```

```
parameter rbar(i)
/
1    0.0260023
2    0.008100891
3    0.073774971
```

Döntési problémák és a GAMS

```
;/  
  
table sigma(i,j)  
      1          2          3  
1     0.017087987 0.003298885 0.001224849  
2     0.003298885 0.005900944 0.004488271  
3     0.001224849 0.004488271 0.063000818  
;  
  
scalar  B /1000.00/  
        R /50.00/  
  
positive variables x(i);  
free variable z;  
  
equations obj, c1, b1;  
  
obj.. z =e= sum(i, sum(j, sigma(i,j)*x(i)*x(j)));  
b1.. sum(i, x(i)) =l= B;  
c1.. sum(i, rbar(i)*x(i)) =g= R;  
  
model portfolio /all/;  
  
solve portfolio minimizing z using nlp;  
  
display x.l;
```

Döntési problémák és a GAMS

FELHASZNÁLT IRODALOM

Anthony Brooke, David Kendrick, Alexander Meeraus, Ramesh Raman: GAMS - A USER'S GUIDE, GAMS Development Corporation, 1998.

Csala Péter - Csetényi Arthur - Tarlós Béla: Informatika alapjai, ComputerBooks, Budapest , 2001.

Chakravarty, S, Optimum Savings with a Finite Planning Horizon. International Economic Review 3 (1962), 338-355.

Dantzig, G. B.: Linear Programming and Extensions. Princeton University Press, Princeton, New Jersey, 1963.

Kendrick, D, and Taylor, L, Numerical methods and Nonlinear Optimizing models for Economic Planning. In Chenery, H B, Ed, Studies of Development Planning. Harvard University Press, 1971.

Richard E. Rosenthal: A GAMS TUTORIAL, Naval Postgraduate School, Monterey, California USA 1998

Sántáné Tóth Edit: Tudásalapú technológia - szakértő rendszerek, Dunaújvárosi Főiskola, 2000.

Velencei Jolán: Döntéstámogatás, Budapesti Műszaki és Gazdaságtudományi Egyetem, Budapest 2002.

Döntési problémák és a GAMS

MELLÉKLET

A SZÁLLÍTÁSI FELADAT OUTPUT FÁJLJA

```
gamside: C:\WINDOWS\gamsdir\project.gpr - [C:\WINDOWS\gamsdir\szallfel.lst]
File Edit Search Windows Help
reszvenyek.gms | reszvenyek.lst | szallfel.gms | szallfel.lst
GAMS Rev 133 Windows NT/95/98 05/14/03 16:52:13 Page 1
General Algebraic Modeling System
Compilation

1 Sets
2 i kgyarak / Kmet, Szeged /
3 j piacok / Sopron, Szhely, Zeg ;;
4 Parameters
5 a(i) i konzervgyar kapacitasa ladaban
6 /
7   Kmet 350
8   Szeged 600 /
9 b(j) kereslet j piacon ladaban
10 /
11   Sopron 325
12   Szhely 300
13   Zeg 275 ;;
14 Table d(i,j) tavolsag kilometerben
15   Sopron      Szhely      Zeg
16   Kmet        302         279         266
17   Szeged      388         354         342 ;
18 Scalar f kilometerre eso fuvardij ladankent forintban /90/;
19 Parameter c(i,j) ladankenti szallitasi koltseg forintban;
20 c(i,j) = f*d(i,j);
21 Variables
22 x(i,j) szallitott mennyiseg ladaban
23 z forintban mert osszes szallitasi koltseg ;
24 Positive variable x ;
25 Equations
26 cost celfuggveny
27 supply(i) i gyar kinalati korlatja
28 demand(j) j piac kielegitendo kereslete;
29 cost.. z =e= sum((i,j), c(i,j)*x(i,j)) ;
30 supply(i) .. sum(j, x(i,j)) =l= a(i) ;
31 demand(j) .. sum(i, x(i,j)) =g= b(j) ;
```

```
gamside: C:\WINDOWS\gamsdir\project.gpr - [C:\WINDOWS\gamsdir\szallfel.lst]
File Edit Search Windows Help
reszvenyek.gms | reszvenyek.lst | szallfel.gms | szallfel.lst
29 demand(j) .. sum(i, x(i,j)) =g= b(j) ;
30 Model szallfel /all/ ;
31 solve szallfel using lp minimizing z ;
32 display x.l, x.m ;

COMPILATION TIME = 0.000 SECONDS 0.8 Mb WIN207-133
GAMS Rev 133 Windows NT/95/98 05/14/03 16:52:13 Page 2
General Algebraic Modeling System
Equation Listing SOLVE szallfel Using LP From line 31

---- cost =E= celfuggveny
cost.. - 27180*x(Kmet,Sopron) - 25110*x(Kmet,Szhely) - 23940*x(Kmet,Zeg)
- 34920*x(Szeged,Sopron) - 31860*x(Szeged,Szhely) - 30780*x(Szeged,Zeg)
+ z =E= 0 ; (LHS = 0)

---- supply =L= i gyar kinalati korlatja
supply(Kmet).. x(Kmet,Sopron) + x(Kmet,Szhely) + x(Kmet,Zeg) =L= 350 ;
(LHS = 0)
supply(Szeged).. x(Szeged,Sopron) + x(Szeged,Szhely) + x(Szeged,Zeg) =L= 600 ;
(LHS = 0)

---- demand =G= j piac kielegitendo kereslete
```

Döntési problémák és a GAMS

```

gamside: C:\WINDOWS\gamsdir\project.gpr - [C:\WINDOWS\gamsdir\szallfel.lst]
File Edit Search Windows Help
reszvenyek.gms | részvenyek.lst | szállfel.gms | szállfel.lst

demand(Sopron).. x(Kmet,Sopron) + x(Szeged,Sopron) =G= 325 ;
(LHS = 0, INFES = 325 ***)

demand(Szhely).. x(Kmet,Szhely) + x(Szeged,Szhely) =G= 300 ;
(LHS = 0, INFES = 300 ***)

demand(Zeg).. x(Kmet,Zeg) + x(Szeged,Zeg) =G= 275 ; (LHS = 0, INFES = 275 ***)

D GAMS Rev 133 Windows NT/95/98 05/14/03 16:52:13 Page 3
General Algebraic Modeling System
Column Listing SOLVE szállfel Using LP From line 31

---- x szállított mennyiség ladaban

x(Kmet,Sopron)
      (.LO, .L, .UP = 0, 0, +INF)
-27180 cost
      1 supply(Kmet)
      1 demand(Sopron)

x(Kmet,Szhely)
      (.LO, .L, .UP = 0, 0, +INF)
-25110 cost
      1 supply(Kmet)
      1 demand(Szhely)

x(Kmet,Zeg)
      (.LO, .L, .UP = 0, 0, +INF)
-23940 cost
      1 supply(Kmet)
  
```

```

gamside: C:\WINDOWS\gamsdir\project.gpr - [C:\WINDOWS\gamsdir\szallfel.lst]
File Edit Search Windows Help
reszvenyek.gms | részvenyek.lst | szállfel.gms | szállfel.lst

      1 supply(Kmet)
      1 demand(Zeg)

REMAINING 3 ENTRIES SKIPPED

---- z forintban mert osszes szállitasi költség

z
      (.LO, .L, .UP = -INF, 0, +INF)
      1 cost

D GAMS Rev 133 Windows NT/95/98 05/14/03 16:52:13 Page 4
General Algebraic Modeling System
Model Statistics SOLVE szállfel Using LP From line 31

MODEL STATISTICS

BLOCKS OF EQUATIONS      3      SINGLE EQUATIONS      6
BLOCKS OF VARIABLES      2      SINGLE VARIABLES      7
NON ZERO ELEMENTS      19

GENERATION TIME = 0.000 SECONDS 1.6 Mb WIN207-133

EXECUTION TIME = 0.000 SECONDS 1.6 Mb WIN207-133
D GAMS Rev 133 Windows NT/95/98 05/14/03 16:52:13 Page 5
General Algebraic Modeling System
Solution Report SOLVE szállfel Using LP From line 31
  
```

Döntési problémák és a GAMS

```

gamside: C:\WINDOWS\gamsdir\project.gpr - [C:\WINDOWS\gamsdir\szallfel.lst]
File Edit Search Windows Help
reszvenyek.gms | részvenyek.lst | szállfel.gms | szállfel.lst

S O L V E      S U M M A R Y

MODEL  szállfel      OBJECTIVE  z
TYPE   LP            DIRECTION  MINIMIZE
SOLVER  BDMLP        FROM LINE  31

**** SOLVER STATUS      1 NORMAL COMPLETION
**** MODEL STATUS      1 OPTIMAL
**** OBJECTIVE VALUE    26685000.0000

RESOURCE USAGE, LIMIT      0.000      1000.000
ITERATION COUNT, LIMIT     7          10000

BDMLP 1.3      Jun 14, 2002 WIN.00.NA 20.7 057.046.040.VIS

Originally developed by
A. Brooke, A. Drud, and A. Meeraus,
World Bank, Washington, D.C., U.S.A.

Work space allocated      --      0.03 Mb

EXIT -- OPTIMAL SOLUTION FOUND.

          LOWER      LEVEL      UPPER      MARGINAL
---- EQU cost          .          .          .          1.000

cost celfuggveny

---- EQU supply i gyar kinalati korlatja

          LOWER      LEVEL      UPPER      MARGINAL

```

```

gamside: C:\WINDOWS\gamsdir\project.gpr - [C:\WINDOWS\gamsdir\szallfel.lst]
File Edit Search Windows Help
reszvenyek.gms | részvenyek.lst | szállfel.gms | szállfel.lst

Kmet      -INF      350.000      350.000      -6840.000
Szeged     -INF      550.000      600.000      .

---- EQU demand j piac kielegitendo kereslete

          LOWER      LEVEL      UPPER      MARGINAL
Sopron     325.000      325.000      +INF      34020.000
Szehely    300.000      300.000      +INF      31860.000
Zeg        275.000      275.000      +INF      30780.000

---- VAR x szallitott mennyiseg ladaban

          LOWER      LEVEL      UPPER      MARGINAL
Kmet .Sopron      .          325.000      +INF      .
Kmet .Szehely     .          .          +INF      90.000
Kmet .Zeg         .          25.000      +INF      .
Szeged.Sopron    .          .          +INF      900.000
Szeged.Szehely   .          300.000      +INF      .
Szeged.Zeg       .          250.000      +INF      .

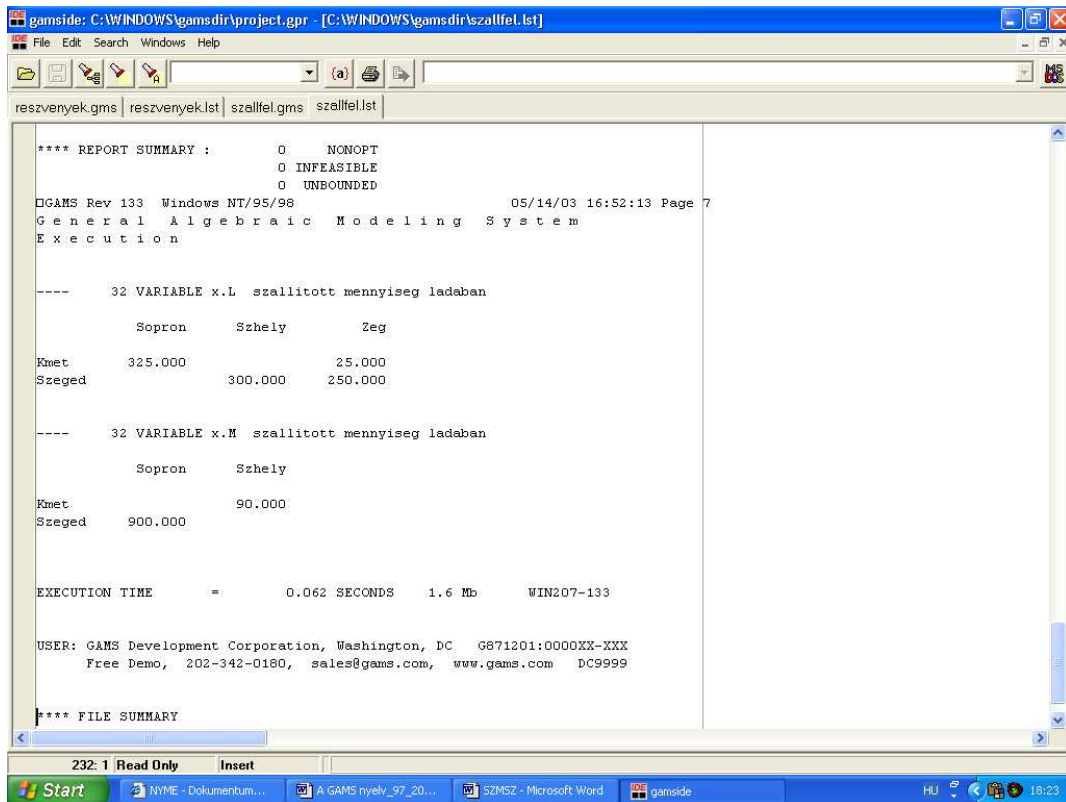
DGAMS Rev 133 Windows NT/95/98      05/14/03 16:52:13 Page 6
General Algebraic Modeling System
Solution Report      SOLVE szállfel Using LP From line 31

          LOWER      LEVEL      UPPER      MARGINAL
---- VAR z          -INF      2.6685E+7      +INF      .

z forintban mert osszes szallitasi koltseg

```

Döntési problémák és a GAMS



The screenshot shows the GAMS software interface. The title bar indicates the file path: C:\WINDOWS\gamsdir\project.gpr. The menu bar includes File, Edit, Search, Windows, and Help. The toolbar contains icons for file operations. The main window displays the following text:

```
**** REPORT SUMMARY :      0  NONOPT
                          0  INFEASIBLE
                          0  UNBOUNDED
GAMS Rev 133  Windows NT/95/98      05/14/03 16:52:13 Page 7
General Algebraic Modeling System
Execution

----  32 VARIABLE x.L  szállított mennyiség ladaban
      Sopron      Szhely      Zeg
Kmet      325.000      25.000
Szeged      300.000      250.000

----  32 VARIABLE x.M  szállított mennyiség ladaban
      Sopron      Szhely
Kmet      90.000
Szeged      900.000

EXECUTION TIME      =      0.062 SECONDS      1.6 Mb      WIN207-133

USER: GAMS Development Corporation, Washington, DC  G871201:0000XX-XXX
Free Demo, 202-342-0180, sales@gams.com, www.gams.com  DC9999

**** FILE SUMMARY
```

The status bar at the bottom shows the current file is '232: 1 Read Only' and the active window is 'gamside'. The taskbar includes the Start button and several open applications: NYME - Dokumentum..., A GAMS nyelv_97_20..., SZMSZ - Microsoft Word, and gamside. The system tray shows the date and time as 18:23.