VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

ÚSTAV SOUDNÍHO INŽENÝRSTVÍ

INSTITUTE OF FORENSIC ENGINEERING

ODBOR INŽENÝRSTVÍ RIZIK

DEPARTMENT OF RISK ENGINEERING

OPTIMALIZAČNÍ MODELOVÁNÍ RIZIK V GAMSU

OPTIMIZATION RISK MODELLING IN GAMS

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE Bc. Vladislav Kutílek

AUTHOR

VEDOUCÍ PRÁCE RNDr. Pavel Popela, Ph.D.

SUPERVISOR

BRNO 2021

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ ÚSTAV SOUDNÍHO INŽENÝRSTVÍ

Zadání diplomové práce

Student:	**Bc. Vladislav Kutílek**
Studijní program:	Řízení rizik technických a ekonomických systémů
Studijní obor:	Řízení rizik ekonomických systémů
Vedoucí práce:	**RNDr. Pavel Popela, Ph.D.**
Akademický rok:	2020/21
Ústav:	Odbor inženýrství rizik

Ředitel ústavu Vám v souladu se zákonem č.111/1998 o vysokých školách a se Studijním a zkušebním řádem VUT v Brně určuje následující téma diplomové práce:

Optimalizační modelování rizik v GAMSu

Stručná charakteristika problematiky úkolu:

Student si prohloubí své znalosti optimalizačních úloh. Zaměří se na modelování správného rozhodování s ohledem na minimalizaci zvolených rizik v úlohách s náhodnými parametry. Doplní si a rozšíří znalosti modelovacích nástrojů, zejména jazyka GAMS. Důraz bude kladen zejména na modelování vybraných přístupů k problematice optimalizace rizik v GAMSu. Při modelování a řešení úloh s testovacími a případně reálnými daty se student zaměří na efektivní postupy řešení a verifikaci získaných výsledků.

Cíle diplomové práce:

1. Studium problematiky optimalizačních úloh.
2. Studium problematiky modelování rizik v optimalizaci.
3. Studium modelovacího nástroje GAMS.
4. Výběr modelů rizik, diskuse jejich vlastností a řešení
5. Efektivní softwarová implementace vybraných modelů.
6. Testovací výpočty, případná aplikace na reálná data vybraného problému.

Seznam doporučené literatury:

GAMS Modelling Language Manuals, GAMS, 2020.

Klapka, J. a kol. Metody operačního výzkumu, Brno 2000

Kall, P., Wallace, S. W.: Stochastic Programming, Wiley, 1994.

Birge,J., Louveaux F.: Introduction to Stochastic Programming. 2nd edition, Springer Verlag, 2011.

Bazaraa M., Jarvis R. Linear Programming and Network Flows, Wiley and Sons, 1991

Termín odevzdání diplomové práce je stanoven časovým plánem akademického roku 2020/21

V Brně, dne

L. S.

Ing. Jana Victoria Martincová, Ph.D. vedoucí odboru	prof. Ing. Karel Pospíšil, Ph.D., LL.M. ředitel

Abstrakt

Diplomová práce se zabývá možnostmi využití optimalizačního modelovacího programového systému GAMS v řízení rizik. Důraz je podle zadání kladen na postupné přiblížení programu pro zájemce o jeho využití v oblasti aplikací rizikového inženýrství. První část práce obsahuje poznatky potřebné k pochopení, co je program GAMS a k čemu se využívá. Další část práce shrnuje návod, jak program stáhnout, nainstalovat, aktivovat a jak vypadá uživatelské rozhraní programu. Díky matematickému programování bude následně vysvětleno na projektu, který se věnoval distribuci plicních ventilátorů, jakých základních přístupů lze využít při modelování rizik v programu GAMS nejprve na modelu deterministickém. Následují složitější modely stochastického programování, jako jsou wait-and-see, do kterých je zahrnut parametr pravděpodobnosti a modely here-and-now, kde například pracujeme s jednotlivými poptávkovými scénáři a verifikujeme, zda řešení bude vhodné i pro ostatní scénáře nebo počítáme náklady pro nejvyšší poptávky. Model two-stage patří také do here-and-now přístupu, ale je podstatně složitější svým rozsahem a daty a zahrnuje parametry dodatečné ceny pro přidané nebo odebrané kusy ventilátorů z objednávky.

Abstract

The diploma thesis deals with the possibilities of using the optimization modelling software system GAMS in risk management. According to the assignment, emphasis is placed on a detailed approach to the program for those, who are interested in its use in the field of risk engineering applications. The first part of the thesis contains the knowledge to understand what the GAMS program is and what it is used for. The next part of the work provides instructions on how to download, install, activate the program and what the user interface of the program looks like. Thanks to mathematical programming, it will be explained on a project on the distribution of lung ventilators, what basic approaches may be used in risk modelling in the GAMS program on a deterministic model. The following are more complex wait-and-see models, which contains the probability parameters and here-and-now models, where we work with demand scenarios and verify whether if they meets the requirements of other scenarios or calculate costs for the highest demands. The two-stage model is also one of the here-and-now models, but it is significantly more complex in its size and range of input data, it includes additional price parameters for added or removed pieces of lung ventilators from the order.

Klíčová slova

GAMS, optimalizace, modelování, optimalizační modelování, riziko, optimalizační modelování rizik, stochastické programování, here-and-now, wait-and-see, two-stage, přístup

Keywords

GAMS, optimization, modelling, optimization modelling, risk, optimization risk modelling, stochastic programming, here-and-now, wait-and-see, two-stage, approach

Bibliografická citace

KUTÍLEK, Vladislav. *Optimalizační modelování rizik v GAMSu.* Brno, 2021. Dostupné také z: https://www.vutbr.cz/studenti/zav-prace/detail/129943. Diplomová práce. Vysoké učení technické v Brně, Ústav soudního inženýrství, Odbor inženýrství rizik. Vedoucí práce Pavel Popela.

Čestné prohlášení

Prohlašuji, že předložená diplomová práce je původní a zpracoval jsem ji samostatně. Prohlašuji, že citace použitých pramenů je úplná, že jsem ve své práci neporušil autorská práva (ve smyslu zákona č. 121/2000 Sb., o právu autorském a o právech souvisejících s právem autorským, v platném znění).

V Brně dne 14. června 2021

………………………………

podpis studenta

Poděkování

Rád bych tímto poděkoval panu RNDr. Pavel Popela, Ph.D. za vedení mé diplomové práce a jeho cenné rady a připomínky, které mi během zpracovávání této práce poskytnul.

OBSAH

ÚVOD

V této diplomové práci se podívám na program GAMS jako na nástroj optimalizačního modelování rizik. Celým názvem jde o General Algebraic Modelling Systém neboli program pro matematické modelování. V této práci mě budou zajímat úlohy lineárního programování využitelné v řízení rizik, které budou muset také zahrnovat náhodné parametry. Tyto modely stochastického programování budou vysvětlovány na aktuálním příkladu inspirovaném událostmi posledního roku týkajícím se přepravy plicních ventilátorů z krajů, kde je jich dostatek, do krajů, kde je více nakažených a je v nich nedostatečná kapacita těchto lékařských přístrojů.

V první části práce bude uvedena teorie potřebná k pochopení, co je to vlastně program GAMS a k čemu je možné ho využít. Krátce je zmíněna lineární optimaliazce, na které budeme dál stavět a využívat ji při modelování mého konkrétního dopravního problému. Na jednoduchém příkladu je rozebrána konstrukce modelu i s vysvětlivkami co jaká hodnota znamená. Dále jsou zde vysvětleny jednotlivé přístupy k modelování.

V další části práce bude návod k programu GAMS, kde program stáhnout, jak GAMS nainstalovat, jak si založit licenci a provést aktivaci programu. Popíšu zde možnosti knihovny modelů obsažené v programu GAMS a také některé modely pro ukázku. Poté v této práci budu řešit svůj konkrétní projekt na distribuci plicních ventilátorů, takže obecně dopravní problém. Začínám zpracováním deterministického modelu, který je výchozí a podle mého nejlehčí na zpracování. V dalším zpracování budou modely wait-and-see a here-and-now, kde se dostaneme ke složitějším modelům. Poté následují „velká“ data a zpracování jejich modelů s popisem a návodem.

1 CÍLE A METODIKA PRÁCE

Cílem práce je představit program GAMS širšímu okruhu studentů a lidí, ať už jsou z praxe nebo se k programu dostali v rámci nějakého určitého projektu. K tomu je nutné zahrnout i popis programu, uvést v čem vůbec spočívají jeho funkce a zejména klást důraz na to, jaký může být jeho přínos pro řešitele různých úloh matematického programování. Začínající uživatel jistě také uvítá základní informace počínaje návodem na stažení programu, přes pokyny pro jeho instalaci až po jeho využití např. pro studijní účely uvedením řady možností optimalizace rozhodování v podmínkách existujících rizik. Klíčové tedy je popsání různých modelů, jejich odlišností a způsobů řešení daných problémů včetně představení různých modelových scénářů optimalizačního modelování rizik a vysvětlení jejich podstaty na příkladech.

1.1 Dílčí cíle:

- Výčet teorie pro lepší porozumění, co je to program GAMS a k čemu se využívá.
- Návod pro instalaci, aktivaci a základní nastavení a pohyb v programu.
- Uvedení několika různých modelů, jejich představení, odlišnosti a kde je najít v knihovně modelů programu GAMS.
- Uvedení několika druhů modelů pro optimalizační modelování rizik využívané v našem oboru.
- Detailní představení každého typu modelu s názorným příkladem z praxe.
- Závěr a zhodnocení přínosů.

2 TEORETICKÁ VÝCHODISKA PRÁCE

2.1 Motivace

V oblasti optimalizace bylo značného pokroku dosaženo v 50. a 60. letech minulého století s vynalezením algoritmů a počítačových kódů pro řešení velkých matematických programovacích problémů. Počet aplikací využívajících tyto nástroje byl v 70. letech menší, než se tehdy očekávalo, protože postupy řešení tvořily jen malou část celkových požadavků souvisejících s modelováním. Velkou část času potřebného k vývoji modelu totiž zahrnovala příprava a transformace vstupních dat a počítačové generování zpráv pro uživatele zahrnujících výsledky. Každý model požadoval mnoho hodin analýzy a programování, aby se uspořádala data a napsaly programy, které by data transformovaly do formy vyžadované tvůrci a programátory algoritmů matematického programování. Dále bylo obtížné detekovat a eliminovat chyby, protože programy, které prováděly datové operace, byly obvykle zcela srozumitelné pouze programátorovi, který je napsal. Analytici odpovědní za projekt a zahrnuté matematické modely většinou nedokázali program tak efektivně obsluhovat. (3 GAMS tutorial, str. 13)

GAMS byl vytvořený pro zlepšení této situace s cílem:

- Poskytnutí jazyka na vysoké úrovni pro kompaktní reprezentaci velkých a složitých modelů.
- Umožnění jednoduchých a bezpečných změn ve specifikaci modelu.
- Zvýšení čitelnosti programu zahrnutím zápisu matematických (algebraických) vztahů.
- Zavedení popisů modelů, které jsou nezávislé na algoritmech řešení.

(3 GAMS tutorial, str. 13)

2.2 Základní vlastnosti programu GAMS

Návrh programu GAMS zahrnoval myšlenky čerpané z teorie relačních databází a matematického programování a pokoušel se spojit tyto myšlenky tak, aby vyhovovaly potřebám tvůrců strategických modelů. (3 GAMS tutorial, str. 13) Teorie relačních databází poskytuje strukturovaný rámec pro práci s daty a jejich transformace. Matematické programování poskytuje způsob popisu optimalizačního (rozhodovacího) problému a různé metody jeho řešení. (5 Klapka, 6 Stochastic programming, 7 Birge, 8 Bazaraa) Při navrhování systému byly použity následující principy:

(3 GAMS tutorial, str. 13)

- Všechny existující algoritmické metody by měly být k dispozici beze změny reprezentace modelu uživatele. Zavádění nových metod nebo nových implementací stávajících metod by mělo být možné bez nutnosti změn ve stávajících modelech. (3 GAMS tutorial, str. 13)
- Problém s optimalizací by měl být vyjádřitelný nezávisle na datech která používá. Toto oddělení logiky a dat umožňuje zvětšit velikost problému, aniž by došlo ke zvýšení složitosti reprezentace. (3 GAMS tutorial, str. 13)
- Použití modelu relačních dat vyžaduje automatizaci alokace počítačových prostředků. To znamená, že lze sestavit velké a složité modely, aniž by se uživatel musel starat o detaily jako jsou například velikost polí. (3 GAMS tutorial, str. 14)

2.3 Dokumentace

Reprezentace modelu v programu GAMS je ve formě, kterou mohou snadno číst lidé i počítače. To znamená, že samotný program GAMS je dokumentací modelu a samostatný popis vyžadovaný v minulosti (který byl složitý na údržbu a měl problémy s aktuálností) už není třeba. Design GAMSu navíc zahrnuje následující funkce, které konkrétně řeší potřeby dokumentace uživatele: (3 GAMS tutorial, str. 14)

- Reprezentace modelu GAMS je stručná a plně využívá eleganci matematického vyjádření. (3 GAMS tutorial, str. 14)
- Všechny transformace dat jsou specifikovány stručně a algebraicky. To znamená, že všechna data mohou být zadána v jejich nejelementárnější podobě a že všechny transformace provedené při konstrukci modelu a jsou k dispozici pro kontrolu.
  (3 GAMS tutorial, str. 14)

- Vysvětlující text může být součástí definice všech symbolů a je reprodukován kdykoli jsou zobrazeny související hodnoty. (3 GAMS tutorial, str. 14)
- Všechny informace potřebné k pochopení modelu jsou v jednom dokumentu.

Samozřejmě je zapotřebí určité disciplíny, aby bylo možné plně využít těchto designových funkcí, ale cílem je zajistit, aby byly modely přístupnější, srozumitelnější, ověřitelnější, a tím i důvěryhodnější. (3 GAMS tutorial, str. 14)

2.4 Přenositelnost

Systém GAMS je navržen tak, aby modely bylo možné řešit na různých typech počítačů beze změn. Model vyvinutý na malém osobním počítači lze později vyřešit například na superpočítači. (3, str. 14) Jedna osoba může vyvinout model, který později použijí ostatní, kteří mohou být vzdálení fyzicky od původního vývojáře. Na rozdíl od předchozích přístupů je třeba přesunout pouze jeden dokument – zápis modelu v GAMSu. Obsahuje všechna data a specifikace potřebné k vyřešení modelu. (3 GAMS tutorial, str. 14)

2.5 Uživatelské rozhraní

Problémy s přenositelností mají také důsledky pro uživatelské rozhraní. Základní systém GAMS je souborově orientovaný a existuje speciální editor nebo grafické vstupní a výstupní nástroje. Spíše než zatěžovat uživatele nutností naučit se další sadu editačních příkazů, nabízí GAMS otevřenou architekturu, ve které může každý uživatel používat svůj vybraný editor textových procesorů. Toto základní uživatelské rozhraní usnadňuje integraci GAMSu s řadou stávajících i budoucích uživatelských prostředí. (3 GAMS tutorial, str. 14) Tvůrci GAMSu ale dále rozvíjejí uživatelské rozhraní. (3 GAMS tutorial)

2.6 Knihovna modelů

Když architekti začínají navrhovat novou budovu, rozvíjejí novou strukturu pomocí nápadů a technik, které byly testovány v předchozích strukturách. Totéž platí i v jiných oblastech: designové prvky z předchozích projektů slouží jako zdroje nápadů pro nový vývoj. (3 GAMS tutorial, str. 14)

Od ranných fází vývoje GAMSu je nashromážděno plno modelů pro použití v knihovně modelových příkladů. Mnoho z nich vychází ze standardních učebnic a lze je použít při výuce a úvodu do formulace modelů optimalizačních problémů nebo k ilustraci úvodních poznatků k používání programu GAMS. Další modely byly použity při analýze politického a ekonomického strategického rozhodování a jsou rovněž zajímavé pro data, která používají. Všechny neustále doplňované modely v knihovně jsou dále popisovány v dalších zdrojích jako jsou odborné články, knihy, manuály a www stránky věnované optimalizaci. Kolekce modelů je součástí všech systémů GAMS spolu s databází, která uživatelům pomáhá najít příklady, které pokrývají země nebo témata, která je zajímají.

(3 GAMS tutorial, str. 14)

2.7 Modelování v programu GAMS

Richard E. Rosenthal sestavil podrobný příklad, jak lze GAMS využít pro formulaci, řešení a analyzování malého a jednoduchého optimalizačního problému. Tento příklad je rychlým, ale kompletním přehledem programu a jeho vlastností.

(3 GAMS tutorial, str. 17)

Příkladem je řešení dopravního problému lineárním programováním, který sloužil jako testovací úloha svým způsobem „laboratorní myš“ při vývoji optimalizačních technologií. Je to i vhodná volba pro demonstraci možností jazyků algebraického modelování jako je GAMS, protože dopravní problém, nehledě na to, jak je velký, má jednoduchou a přehlednou algebraickou strukturu. Uvidíme, že téměř všechna tvrzení ve výstupním souboru GAMS, která se chystáme představit, mohou zůstat nezměněná i v případě, pokud by se řešil i mnohem větší dopravní problém. (3 GAMS tutorial, str. 17)

Ve známém dopravním problému uvažujeme zásoby pro několik závodů a požadavky na několika trzích pro jednu komoditu a známe jednotkové náklady na přepravu komodity ze závodů na trhy. Ekonomickou otázkou je: Jak velké zásilky by měly být mezi každým závodem a každým trhem, aby se minimalizovaly celkové náklady na dopravu?

(3 GAMS tutorial, str. 17)

Algebraická reprezentace tohoto problému může být následující:

Indexy:

- i = závody (podniky)
- j = trhy (3 GAMS tutorial, str. 17)

Známé údaje:

- a_i = zásobování komoditou ze závodu i
- b_i = poptávka po komoditě na trhu j
- c_{ij} = náklady na jednotku přepravovaného nákladu mezi závodem i a trhem j

(3 GAMS tutorial, str. 17)

Rozhodovací proměnné:

- x_{ij} = množství komodity, které je nutné přepravit ze závodu i na trh j
  kde $x_{ij} \geq 0$, pro všechna i, j (3 GAMS tutorial, str. 17)

Podmínky:

- Dodržení limitu dodávky na závod i: $\sum j\ x_{ij} \leq aj$ pro všechna i
- Uspokojení poptávky na trhu j: $\sum i\ x_{ij} \geq bj$ pro všechna j
- Účelová funkce: Minimalizace $\sum_i \sum_j c_{ij} x_{ij}$ výsledek v peněžních jednotkách

(3 GAMS tutorial, str. 17)

Tento jednoduchý příklad ilustruje základní modelovací postupy, které jsou považované za dobré v obecném pojetí a jsou standardní v GAMSovém prostředí. Nejprve jsou tedy všechny entity modelu identifikovány a seskupeny podle typu. Za druhé je pořadí entit zvoleno tak, že před definováním není odkazováno na žádný dosud nedefinovaný symbol. Za třetí jsou jednotky všech daných entit specifikovány a za čtvrté, jednotky jsou vybrány v takovém měřítku, aby číselné hodnoty, které mají být optimalizovány, měly relativně malé řády.

Nyní se podíváme na to, jak vypadá zápis dopravního problému do GAMSu v praxi. V tabulce máme výchozí hodnoty pro dopravní problém, se kterými budeme dále pracovat a předvádět na nich zápis optimalizačního modelu.(3 GAMS tutorial, str. 17-18)

Podniky	Dopravní vzdálenost na trh (1000km)			Dodávky
	New York	Chciago	Topeka	
Seattle	2.5	1.7	1.8	350
San Diego	2.5	1.8	1.4	600
Poptávka	325	300	275	

Tabulka 1 Vstupní data pro příklad dopravního problému (3, str. 18)

Jak je již známo podle tabulky, uvažujeme u úvodního dopravního problému 2 dodávající podniky a 3 trhy, kam je nutné zboží dopravit. Dopravní vzdálenosti jsou v tisících kilometrů a náklady na dopravu 90 Kč na jednotku dodávky a tisíc kilometrů. Zápis modelu v programu GAMS bude následující: (3 GAMS tutorial, str. 18)

```
Set
   i 'dodavatelske podniky' / seattle,  san-diego /
   j 'poptavajici trhy'        / new-york, chicago, topeka /;

Parameter
   a(i) 'kapacita podniku i v kusech'
        / seattle    350
          san-diego  600 /

   b(j) 'poptavka natrzich j v kusech'
        / new-york   325
          chicago    300
          topeka     275 /;

Table d(i,j) 'vzdalenost v tisicich km'
              new-york  chicago  topeka
   seattle         2.5      1.7     1.8
   san-diego       2.5      1.8     1.4;

Scalar f 'naklady prepravy v korunach na kus a ujetych 1000km' / 90 /;

Parameter c(i,j) 'prepravni naklady v tisicich korun na kus';
c(i,j) = f*d(i,j)/1000;

Variable
   x(i,j) 'prepravovane mnozstvi v kusech'
   z      'celkove naklady prepravy v tisicich korun';

Positive Variable x;

Equation
   cost      'definuje ucelovou funkci'
   supply(i) 'limit poctu kusu v podnicich i'
   demand(j) 'uspokojeni poptavky na trzich j';

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 transport / all /;

solve transport using lp minimizing z;

display x.l, x.m ;
```

Obr. 1 Vzorový model transport (9 GAMS)

Po vyřešení tohoto modelu stisknutím tlačítka F9, by se měl vytvořit textový soubor .lst obsahující výsledky tohoto optimalizačního modelu, tento soubor rovněž ihned uvidíte v pravé části obrazovky. Ve výsledcích uvidíte dvě důležité sady hodnot, jako první počty přepravovaných kusů mezi městy a druhá zobrazuje navýšení nákladů na přepravu zboží v tisících korun. Celková cena přepravy po provedení optimalizace je 165 675 Kč.

(3 GAMS tutorial, str. 19)

	New York	Chicago	Topeka
Seattle	50	300	
San Diego	275		275

Tabulka 2 Přepravované množství v kusech (GAMS)

	Chicago	Topeka
Seattle		0.036
San Diego	0.009	

Tabulka 3 Cena za jeden přepravovaný kus v tisísích korun (GAMS)

K výše zobrazeným tabulkám se lze dostat tak, že po vyřešení modelu stlačením klávesy F9 se v levém navigačním menu pod tímto gamsovým souborem (u mě uložený jako trnsport.gms) objeví soubor trnsport.lst nebo ve výsledcích na pravé straně obrazovky kliknete na modrý odkaz do souboru lst pod názvem: „Reading solution for model transport" v tomto souboru po dojetí až dolů na konec souboru jsou uvedeny právě hodnoty z těchto dvou tabulek. (9 GAMS) V základní optimalizaci se neuvažuje přeprava mezi Seattlem a Topekou a San Diegem a Chicagem, avšak pokud byste chtěli nebo bylo nutné přepravovat i do těchto trhů, tato tabulka nám udává navýšení nákladů přepravy na jeden kus zboží (v tisících korun). To znamená, že jeden kus zboží přepravovaný mezi Seattlem a Topekou bude znamenat navýšení celkových nákladů přepravy o 36 Kč a každý jeden kus zboží navíc přepravovaný mezi San Diegem a Chicagem znamená navýšení nákladů o 9 Kč. (3 GAMS tutorial, str. 19)

2.8 Přístupy využívané při optimalizačním modelování

2.8.1 Deterministický (výchozí) model

Základní deterministický optimalizační model (tzv. úlohu matematického programování) uvažujeme dále ve tvaru

$$\min\{f(\boldsymbol{x})|\boldsymbol{x} \in C\},$$

kde f označuje účelovou funkci. $\boldsymbol{x}$ je vektor rozhodovacích proměnných, C je množina přípustných řešení, který je obvykle určena omezeními ve tvaru rovnic a nerovnic a mezemi pro proměnné. (5 Klapka, 6 Stochastic programming) Pokud jsou výrazy v účelové funkci a omezeních lineární, hovoříme o optimalizační úloze lineárního programování. Speciální úlohou lineárního programování pak je výše uvedená dopravní úloha ve 2.7.

2.8.2 Model stochastické optimalizace

Pokud jsou některé parametry náhodné, zapisujeme to obecně ve tvaru tzv. původní úlohy stochastického programování (6 Stochastic programming, 7 Birge)

$$\min\{f(\boldsymbol{x},\boldsymbol{\xi})|\boldsymbol{x} \in C(\boldsymbol{\xi})\},$$

kde nový symbol ξ značí náhodný vektor viz. Matematika IV (10 Karpíšek). Přístupy stochastického programování jsou dále podstatné pro modelování rozhodování v podmínkách uvažování rizika v GAMSu. Uvedená úloha je syntakticky korektně zapsaná, ale sémantika je ze zápisu zatím nejasná, musíme rozhodnout, kdy rozhodnutí přijímáme. Podle toho dále uvádíme dva přístupy.

2.8.3 Přístup wait-and-see (WS)

Při wait-and-see (WS) přístupu rozhodnutí následuje (nebo může následovat) až po pozorování realizace náhodného vektoru. Rozhodnutí $\boldsymbol{x}$ tedy závisí na $\boldsymbol{\xi}$ a to zapisujme jako $\boldsymbol{x}(\boldsymbol{\xi})$ a celý model (tzv. deterministický přepis) zapíšeme

$$\min\{f(\boldsymbol{x}(\boldsymbol{\xi}),\boldsymbol{\xi})|\boldsymbol{x}(\boldsymbol{\xi}) \in C(\boldsymbol{\xi})\}$$

a v případě konečného počtu realizací náhodného vektoru řešení získáme opakovaným řešením deterministické úlohy pro jednotlivé realizace.

2.8.4 Přístup here-and-now (HN)

Při here-and-now (HN) přístupu rozhodnutí musí být přijato předtím, než známe realizaci náhodného vektoru. Na rozdíl od WS přístupu u HN přístupu je vektor rozhodnutí $\boldsymbol{x}$ stejný pro všechny budoucí realizace $\boldsymbol{\xi}$. Tento způsob rozhodování v praxi je častější, protože většinou neznáme budoucí poptávky, kapacity, dopravní ceny, návratnosti investic aj.

(6 Stochastic programming) V případě HN musíme ještě vyjasnit, jak budeme přistupovat k budoucí náhodě, proto uvádíme některé vybrané deterministické přepisy, které přiblížíme na uvedené dopravní úloze (viz 2.7), ve které budeme uvažovat náhodné poptávky.

2.8.5 Vybrané HN deterministické přepisy

Here-and-now Expected value (HN EV) přepis se vyznačuje tím, že pro náhodné poptávky vypočteme vážený průměr poptávky všech scénářů pro dané město a pro tyto hodnoty model vyřešíme. Tento postup obecně zapíšeme

$$\min\{f(\boldsymbol{x}, E\boldsymbol{\xi}) | \boldsymbol{x} \in C(E\boldsymbol{\xi})\},$$

kde $\boldsymbol{E}\xi$ značí střední hodnotu náhodného vektoru ξ.

Dále je vhodné verifikovat, zda rozhodnutí o dopravě pokryje poptávky všech scénářů v každém městě, pokud ano, model má pro každý scénář přípustné řešení, pokud ne, model nemá přípustné řešení a náklady by se daly označit za jdoucí do plus nekonečna.

Here-and-now minmax (HN MM) model bere tu nejvyšší možnou poptávku napříč všemi scénáři pro každé město. S těmito hodnotami se poté model vyřeší, model řešení má, ale je jasné, že náklady budou vysoké díky nejvyšší hodnotě pro poptávku každého města. Obecně tento přístup můžeme zapsat jako

$$\min \max\{f(\boldsymbol{x}, \boldsymbol{\xi}) | \boldsymbol{x} \in C(\boldsymbol{\xi})\ a.s.\},$$

kde max značí maximalizaci přes všechny možné realizace ξ, minimalizuje se vzhledem k x a zkratka $a.\ s.$ znamená, že x musí být přípustné skoro jistě, tedy s pravděpodobností 1, a tedy pro konečný počet realizací musí být přípustné pro všechny realizace (scénáře).

Here-and-now two stage (HN TS). Jedná se již o komplexnější model, do výpočtu se zahrnují všechny scénáře s jejich hodnotami poptávek a jejich pravděpodobnostmi a navíc k HN rozhodnutí o dopravě má každý scénář zavedeny poplatky za kus navíc přidaný do přepravy pro daný scénář a poplatky za odřeknutí kusu z přepravy v každém scénáři. Tímto přístupem a jeho modelováním v GAMSu se budeme dále podrobněji zabývat.

2.9 Simplexová metoda

Je iterativní způsob řešení problémů lineárního programování (optimalizace) objevený matematikem Georgem Dantzigem. (5 Klapka, 8 Bazaraa) Simplexový algoritmus postupuje od základního řešení, v každém svém kroku řešení pozmění takovým způsobem, aby hodnota účelové funkce byla vyšší než v kroku předchozím. Algoritmus se zastaví, pokud již nelze zlepšit řešení, tzv. pokud již je optimální. (5 Klapka) Simplexová metoda je v GAMSu implementována v několika řešičích např. v řešiči CPLEX. (9 GAMS)

3 NÁVOD K PROGRAMU A MODELOVÁNÍ

3.1 Instalace programu a přidání licence

V prvé řadě, pokud si chcete nainstalovat program GAMS je nutné otevřít si stránky programu: www.gams.com

Zde si můžete povšimnout nabídky stránky, první položkou jsou Products, dále Documentation a třetí položka nás zajímá nejvíce, a to je položka Download. Klikněte si na položku Download, po načtení stránky zde uvidíte 3 modrá pole s nápisem Download. Vyberte si podle svého systému, zda používáte Windows, což bude i pravděpodobně Váš případ nebo další na výběr jsou platformy Linux a Mac OS. Dobré je, že nemusíte sami vědět, zda máte 64-bitový nebo 32-bitový operační systém, protože instalace obsahuje soubory pro oba typy operačního systému.

3.1.1 Instalace programu GAMS

Po stažení si spusťte instalaci programu, kde Vás uvítá úvodní okno instalace. Doporučuji si v okénku dole zaškrtnout možnost „Use advanced installation mode“ a pokračovat klepnutím na tlačítko „Next“. V dalším okně volíte cestu, kam se GAMS nainstaluje. Standardní cesta je na disk C:\GAMS\XX, kde XX budou nahrazena aktuální dostupnou verzí programu. Doporučuji zde nechat klasickou předvolenou cestu, pokud nepotřebujete program dát na jiný disk například z důvodu nedostatečného místa na disku C: nebo pokud nemáte například jednotku D: disk SSD, který je rychlý a pro program bude určitě lepší. Další stránka instalace nabízí výběr, zda umístit zástupce do nabídky Start v systému, tuto položku bych doporučil nechat jak je a přesunout se na další stranu. V dalším kroku se Vás program ptá, zda chcete defaultně spouštět GAMS soubory v GAMS Studiu nebo v GAMS IDE. GAMS IDE je prostředí více podobné klasickému příkazovému řádku, ale může mít různé problémy s kompatibilitou, jelikož je zastaralé oproti GAMS Studiu, které je modernější, mělo by mít větší kompatibilitu s různými systémy a programy a navíc je v grafickém rozhraní. Pro školní účely jsme používali ale GAMS IDE, takže pro začátek bych doporučil zvolit si GAMS IDE a poté časem můžete zjistit, které prostředí je více vyhovující a více funkční. Samozřejmě soubory se i po tomto výběru dají spouštět v obou rozhraních, na což upozorňuje i text pod výběrem při instalaci.

Přesunem na další stranu se dostáváme k výběru, zda chcete nainstalovat program pro všechny uživatele počítače, což je volba, kterou je třeba řešit jen v případě, že PC používá více lidí na více účtech a také více lidí bude potřebovat pracovat v tomto programu. Jestliže je na počítači jeden účet, tuhle volbu není nutné řešit. Volbu „Add GAMS directory to PATH environment variable“ je lepší zaškrtnout pro budoucí usnadnění práce, jelikož v programu nebudete muset psát celou cestu k souboru, ale pouze od složky s GAMSem dále. Nakonec samozřejmě volby vytvořit zástupce pro GAMS IDE a Studio doporučuji ponechat zaškrtnuté, pokud si program nechcete spouštět ze složky, ze Startu či jiným způsobem. Na další stránce se instalace ptá, zda máte licenci k programu, vyberte volbu „No license“ a pokračujte dále. Zde už je jen shrnutí navolených možností a tlačítko „Install“, kterým program nainstalujete a tímto máte úspěšně program přidaný v počítači a připravený ke spuštění.

3.1.2 Aktivace a reaktivace programu licencí

Nyní jděte znovu na stránky programu GAMS, vyberte záložku Download a sjeďte o kousek dolů pod samotná stažení, kde najdete nadpis „Request a Free Demo License“ a zde vyplňte všechny požadované údaje, což jsou:

- Křestní jméno
- Příjmení
- Email
- Instituce/Organizace – zde vyplňte například VUT
- Země
- Nakonec captcha kód

Po vyplnění správného captcha kódu se tento řádek změní na „Submit“ a tím pak potvrdíte registraci. Po potvrzení na Váš email dorazí zpráva s odkazem pro potvrzení registrace, na tento link klepněte a registraci dokončíte. Přijde druhý email, ve kterém už je licence, kterou představuje 6 řádků s údaji. Tyto údaje zkopírujte a spusťte program GAMS Studio nebo GAMS IDE.

Ještě než se program spustí, rozpozná, že máte zkopírovanou licenci ve schránce a vypíše zprávu, že rozpoznal licenci a zda chcete vytvořit licenční soubor umístěný v dokumentech ve složce GAMS. Kliknout na tlačítko „Yes“ a tímto se vytvoří licenční soubor a můžete začít používat program. Pokud si chcete zkontrolovat licenci, je to možné po kliknutí na „Help“ v horní části programu a výběru možnosti GAMS Licensing.

Pokud by jste například v program udělali i diplomovou práci, tak jako já, nejspíš Vám nebude poskytovaná roční licence stačit a vyprší Vám její platnost, nemusíte se však ničeho bát, na stejný email, který jste registrovali při první licenci můžete provést i druhou registraci, postup je naprosto stejný jako při vytváření první licence. Tento odstavec zde uvádím proto, protože mě samotnému licence vypršela před dopsáním diplomové práce a zjišťoval jsem tedy jakým způsobem se provádí reaktivace GAMSu.

3.2 Knihovna modelů

V GAMSU můžete najít také knihovnu modelů, pokud jste v programu začátečník a nevíte, jak napsat kostru programu nebo jen zkrátka nechcete psát celou kostru ručně kvůli riziku způsobení nějaké chyby v zápisu či z jiných důvodů. V knihovně najdete například modely pro transportní problémy, problém namíchání nové slitiny z dostupných materiálů, rozšíření výroby podniku o další produkty, problém skladování zásob a spousta dalších modelů, kde je potřeba minimalizovat, maximalizovat nebo řešit matematické úlohy. Vybrané základní modely vhodné pro začínající uživatele budou uvedeny níže.

3.2.1 Model Transport

Tento model lze v programu v záložce Model library explorer lze najít pod názvem „trnsport“. Je velice rozšířený model, který jsem uvedl i v předchozí kapitole 2.7 a inspiroval mne také v předmětu Matematické modely rozhodování pro zpracování části semestrální práce. Cílem tohoto modelu je minimalizace nákladů na dopravu zboží. Nyní zde popíšu model modifikovaný pro náhodnou poptávku. Začátek modelu je jeho nadefinování, což znamená uvedení zákazníků, dodavatelů a vybraného datového scénáře, jaký se v úloze bude nejprve uplatňovat. Poté v dalším kroku následuje definice hodnot pro dodavatele i poptávající, vše musí být pečlivě a přehledně uvedeno a zapsáno se správnými parametry. Pak je zapsána tabulka vzdáleností dodavatelů a poptávajících, kde vzdálenost je uváděna v kilometrech. Pod hodnotou Scalar se ukrývá cena rozvozu, tzn. kolik korun poptávající zaplatí dodavateli za cestu při rozvozu zboží. Parametr c už nám počítá cenu násobenou ujetými kilometry. Nyní nadefinujeme proměnné a rovnice, které se při výpočtu minimalizace nákladů budou řešit.

```
Set
   i 'canning plants' / seattle,  san-diego /
   j 'markets'        / new-york, chicago, topeka /;

Parameter
   a(i) 'capacity of plant i in cases'
        / seattle    350
          san-diego  600 /

   b(j) 'demand at market j in cases'
        / new-york   325
          chicago    300
          topeka     275 /;

Table d(i,j) 'distance in thousands of miles'
              new-york  chicago  topeka
   seattle         2.5      1.7     1.8
   san-diego       2.5      1.8     1.4;

Scalar f 'freight in dollars per case per thousand miles' / 90 /;

Parameter c(i,j) 'transport cost in thousands of dollars per case';
c(i,j) = f*d(i,j)/1000;

Variable
   x(i,j) 'shipment quantities in cases'
   z      'total transportation costs in thousands of dollars';

Positive Variable x;

Equation
   cost      'define objective function'
   supply(i) 'observe supply limit at plant i'
   demand(j) 'satisfy demand at market j';

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 transport / all /;

solve transport using lp minimizing z;

display x.l, x.m;
```

Obr.: 2 Model Transport (9 GAMS)

Nyní si tento model rozebereme do detailu od začátku jeho sestavování, aby bylo jasné, proč je vše použito a proč je to použito zrovna takovým způsobem, jaký vidíte na obrázku výše. Ještě bych zde rád uvedl, že to, jak vidíte model i prostorově rozvržený je dobré dodržovat kvůli přehlednosti při případném ladění chyb v pozdějších fázích sestavování modelu a v případě tabulky **Table** je nutnost dodržet přesně takové sestavení, jelikož hodnoty v ní obsažené musí být umístěny přesně pod (v tomto případě) městy ke kterým jsou vztaženy a ve správných řádcích opět k městům, ke kterým jsou vztaženy.

```
Set
   i 'canning plants' / seattle,  san-diego /
   j 'markets'        / new-york, chicago, topeka /;

Parameter
   a(i) 'capacity of plant i in cases'
        / seattle    350
          san-diego  600 /

   b(j) 'demand at market j in cases'
        / new-york   325
          chicago    300
          topeka     275 /;
```

Obr. 3 Nastavení Setů a parametrů (9 GAMS)

V nastavení **`Set`** identifikujeme entity vstupující do optimalizačního modelu. Konkrétně v tomto případě jsme nastavili, že **i** jsou podniky dodávající zboží a **j** jsou poptávající trhy, které chtějí zboží nakoupit. V **`Parameter`** nastavujeme **a(i)**, jaké množství zboží jsou dodavatelé schopni dodat a **b(j)** označuje poptávané množství zboží pro každý trh.

```
Table d(i,j) 'distance in thousands of miles'
              new-york  chicago  topeka
   seattle         2.5      1.7     1.8
   san-diego       2.5      1.8     1.4;

Scalar f 'freight in dollars per case per thousand miles' / 90 /;

Parameter c(i,j) 'transport cost in thousands of dollars per case';
c(i,j) = f*d(i,j)/1000;
```

Obr. 4 Tabulka, cena přepravy a parametr c (9 GAMS)

V **`Table`** zapisujeme do tabulky vzdálenosti dodavatelů a trhů v tisících mil. Jak již bylo výše zmíněno, tak zmíním i zde, protože se jedná o velice důležitou věc a to, že vzdálenosti musí být přesně pod trhem a vedle dodavatele, ke kterému se vztahují.

`Scalar` f nám udává, že cena přepravy za jeden kus zboží (bednu) přepravenou na tisíc mil je 90 dolarů. **`Parameter` c** představuje součin ceny přepravy a vzdáleností z tabulky v podílu k 1000 mil.

```
Variable
   x(i,j) 'shipment quantities in cases'
   z      'total transportation costs in thousands of dollars';

Positive Variable x;
```

Obr. 5 Proměnná a kladná proměnná (9 GAMS)

`Variable` x(i,j) je proměnná a vyjadřuje množství přepravovaných beden v dodávce zboží. Proměnná **z** ponese hodnotu celkových nákladů přepravy na ujetých tisíc mil.

```
Equation
   cost      'define objective function'
   supply(i) 'observe supply limit at plant i'
   demand(j) 'satisfy demand at market j';

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);
```

Obr. 6 Pravidla rovnosti modelu (9 GAMS)

Cost zde reprezentuje účelovou funkci celkových nákladů přepravy. Je nadefinována jako **z** rovnající se sumě součinů parametru **c** a proměnné **x**. **Supply** zde reprezentuje omezení počtem volných kusů dodavatelů a v definici rovnice níže je zanesena rovnost, že **x** (počet přepravovaných kusů) musí být rovno nebo menší počtu volných kusů zboží na skladě **a** uvedených v tabulce. **Demand** značí poptávku po zboží na trzích a níže je uvedena rovnost, že **x** musí být rovno nebo větší než je uvedená poptávka **b** po zboží na trzích v tabulce.

```
Model transport / all /;

solve transport using lp minimizing z;

display x.l, x.m;
```

Obr. 7 Model, řešitel a výpis výsledků (9 GAMS)

Model transport /all/ znamená, že jsme modelu dali název transport a do řešení zahrnujeme všechna omezení, abychom nemuseli jednotlivě vypisovat /cost, supply, demand/. **Solve** příkazem zadáme programu instrukce, aby vyřešil model **transport** minimalizací **z** za použití lineárního programování. **Display x.l** vypíše v lst souboru tabulku s přepravovanými počty kusů zboží a **x.m** vypíše tabulku s cenou přepravy jednoho kusu zboží mezi dodavatelem a trhem v tisících dolarů.

3.2.2 Model Blend

Tenhle model lze v programu v záložce Model library explorer lze najít pod názvem „blend". Tento model se zabývá problémem slitin. Bere v potaz materiály dostupné na trhu a ty pak kombinuje ve snaze utvořit požadovanou slitinu s co nejmenšími náklady výroby. Problém se řeší v tomto případě ve dvou modelech. Model první bez přihlédnutí k materiálové vyváženosti a model druhý s materiálovou vyvážeností.

```
Set
   alloy 'products on the market' / a*i /
   elem  'required elements'       / lead, zinc, tin /;

Table compdat(*,alloy) 'composition data (pct and price)'
          a     b     c     d     e     f     g     h     i
   lead   10    10    40    60    30    30    30    50    20
   zinc   10    30    50    30    30    40    20    40    30
   tin    80    60    10    10    40    30    50    10    50
   price  4.1   4.3   5.8   6.0   7.6   7.5   7.3   6.9   7.3;

Parameter
   rb(elem)   'required blend' / lead 30, zinc 30, tin 40 /
   ce(alloy) 'composition error (pct-100)';

ce(alloy) = sum(elem, compdat(elem,alloy)) - 100;
display ce;

Variable
   v(alloy) 'purchase of alloy (pounds)'
   phi      'total cost';

Positive Variable v;

Equation
   pc(elem) 'purchase constraint'
   mb       'material balance'
   ac       'accounting: total cost';

pc(elem).. sum(alloy, compdat(elem,alloy)*v(alloy)) =e= rb(elem);

mb..       sum(alloy, v(alloy)) =e= 1;

ac..       phi =e= sum(alloy, compdat("price",alloy)*v(alloy));

Model
   b1 'problem without mb' / pc,     ac /
   b2 'problem with mb'    / pc, mb, ac /;

Parameter report(alloy,*) 'comparison of model 1 and 2';

solve b1 minimizing phi using lp;
report(alloy,"blend-1") = v.l(alloy);

solve b2 minimizing phi using lp;
report(alloy,"blend-2") = v.l(alloy);

display report;
```

Obr.: 8 Model Blend (9 GAMS)

S tímto modelem nemám větší osobní zkušenost, ale shrneme si tu o čem model je kousek po kousku. **`Set`** udává, že v tomto modelu budeme mít **alloy** (slitinu) a **elem** (elementy), neboli materiály potřebné pro výrobu slitiny. V **`Table`** jsou uvedeny slitiny pro slévání **a** až **i**, podíl elementů obsažený ve slitině a cena za slitinu. **`Parameter`** **rb** udává požadovaný počet elementů ve slitině a **ce** udává, že chemické složení slitiny má dohromady ze všech tří elementů činit 100%. Pod příkazem **`Variable`** máme proměnnou **v** udávající nákup slitiny v librách (váhová jednotka) a **phi**, která zde zastupuje celkové náklady, dále je uvedeno, že **v** musí být kladná proměnná. **`Equation`** (rovnosti) jsou uvedeny tři, konkrétně **pc(elem)** pro omezení nákupu surovin **mb** pro nastaveni materiálové vyváženosti a **ac** jsou celkové náklady, které se rovnají součinu ceny slitiny a počtu nakoupených slitin. V příkazu **`Model`** máme nastaveno řešení dvou modelů a to **b1** pro řešení modelu bez materiálového vyvážení a **b2** pro řešení modelu s materiálovým vyvážením. Ve výsledku vyšly oba modely stejně, protože shodně používali 0,6 jednotek slitiny **b** a 0,4 jednotek slitiny **d**. Celkové náklady jsou 4.98 Kč nebo jiných jednotek podle toho, co si stanovíte jako cenu.

```
----     61 PARAMETER report  comparison of model 1 and 2

      blend-1     blend-2

b       0.600       0.600
d       0.400       0.400
```

Obr. 9 Výsledek porovnání použitých slitin v modelu 1 a 2 (9 GAMS)

3.2.3 Model Production mix

Tenhle model lze v programu v záložce Model library explorer lze najít pod názvem „prodmix“. V tomto modelu jde o snahu maximalizovat profit, v tomto případě firmy vyrábějící nábytek rozšířením sortimentu o několik nových druhů stolů lišících se v truhlářském zpracování a dokončovacími pracemi na desce stolu. Do výpočtu vstupují proměnné mix(desk), což znamená kolik kusů konkrétního typu stolů se bude vyrábět. Druhá proměnná je zde profit vypočítávající maximalizovaný celkový zisk za prodané výrobky.

```
Set
   desk / d1*d4 /
   shop / carpentry, finishing /;

Table labor(shop,desk) 'labor requirements (man-hours)'
                d1     d2     d3     d4
   carpentry     4      9      7     10
   finishing     1      1      3     40;

Parameter
   caplim(shop) 'capacity (man hours)' / carpentry 6000, finishing 4000 /
   price(desk)  'per unit sold ($)'    / d1 12, d2 20, d3 18, d4 40     /;

Variable
   mix(desk) 'mix of desks produced (number of desks)'
   profit    'total profit                       ($)';

Positive Variable mix;

Equation
   cap(shop) 'capacity constraint (man-hours)'
   ap        'accounting: total profit    ($)';

cap(shop).. sum(desk, labor(shop,desk)*mix(desk)) =l= caplim(shop);

ap..        profit =e= sum(desk, price(desk)*mix(desk));

Model pmp 'product mix problem' / all /;

solve pmp maximizing profit using lp;
```

Obr.: 10 Model Production mix (9 GAMS)

V **`Set`** máme nastaveno **desk** identifikující jednotlivé typy stolů a **shop** zastávající truhlářskou a dokončovací práci na stolech. **`Table`** udává časovou náročnost truhlářské a dokončovací práce pro jednotlivé typy stolů. **`Parameter`** **caplim(shop)** limituje využitelný časový fond pro truhlařinu na 6000 hodin a dokončovacích prací na 4000 hodin, **price(desk)** znamená cenu inkasovanou za jeden prodaný stůl v dolarech. **`Variable`** **mix(desk)** znamená proměnnou v počtu produkovaných kusů u různých typů stolů a **profit** je potom celkový zisk z prodeje vyrobených stolů. Kladnou proměnnou je zde **mix**. **`Equation`** (omezující podmínky) jsou **cap(shop)** je kapacitní omezení, které musí být menší nebo rovno **caplim(shop)** a **ap** zastupující celková zisk rovný součinu ceně stolů a vyrobenému množství stolů. V **`Solver`** uvádíme, že u modelu **pmp** (production mix problem) chceme maximalizovat zisk použitím optimalizace lineárního programování.

```
---- VAR mix  mix of desks produced (number of desks)

          LOWER          LEVEL          UPPER         MARGINAL

d1          .          1333.3333        +INF            .
d2          .              .            +INF          -6.6667
d3          .              .            +INF          -3.3333
d4          .            66.6667        +INF            .

                              LOWER          LEVEL          UPPER          MARGINAL

---- VAR profit               -INF        18666.6667        +INF              .

  profit  total profit                         ($)
```

Obr. 11 Výsledky modelu produktového mixu (9 GAMS)

3.2.4 Model Warehouse

Tento model lze v programu v záložce Model library explorer najít pod názvem „whouse“. Problém řešený tímto modelem se týká skladu a jeho omezené velikosti pro skladování zboží. Kdy nakoupit zboží a naopak, kdy prodávat zásoby tak, aby byly minimalizovány celkové náklady na skladování. Hlavní úlohu v tomto problému hraje čas, který je nadefinován jako první veličina. Další důležité parametry jsou prodejní cena jednotky zboží a počáteční stav zásob ve skladu. Dále máme uvedenou cenu za skladování jednotky zboží a maximální velikost skladu (kolik se dá uskladnit jednotek zboží). Proměnnými v tomto modelu jsou skladové zásoby v daném čase, počet prodaných jednotek v určitém čase, kusy nakoupených jednotek v daném čase a celková cena. Podmínky rovnosti jsou nastaveny stavem zásob a vyúčtováním celkových nákladů za pohyb a uložení zboží ve skladu. Počet jednotek na skladu v daném čase **sb(t)** je roven zásobám na skladě v minulém období s přičtením nakoupených jednotek, odečtením prodaných jednotek a přičtením počátečních zásob. Účelová funkce nákladů na skladování zboží **at** je definována jako suma součinu ceny a nakoupeného zboží s odečtením prodaných jednotek zboží a přičtením součinu nákladů na skladování zboží a jednotek skladovaného zboží. Řešitele je v tomto modelu nastavený na minimalizaci nákladů skladování za použití lineárního programování.

```
Set t 'time in quarters' / q-1*q-4 /;

Parameter
   price(t)  'selling price ($ per unit)' / q-1 10, q-2 12, q-3 8, q-4 9 /
   istock(t) 'initial stock      (units)' / q-1 50 /;

Scalar
   storecost 'storage cost  ($ per quarter per unit)' /   1 /
   storecap  'stocking capacity of warehouse (units)' / 100 /;

Variable
   stock(t)  'stock stored at time t (units)'
   sell(t)   'stock sold at time t   (units)'
   buy(t)    'stock bought at time t (units)'
   cost      'total cost                 ($)';

Positive Variable stock, sell, buy;

Equation
   sb(t) 'stock balance at time t (units)'
   at    'accounting: total cost      ($)';

sb(t).. stock(t) =e= stock(t-1) + buy(t) - sell(t)  + istock(t);

at..    cost =e= sum(t, price(t)*(buy(t) - sell(t)) + storecost*stock(t));

stock.up(t) = storecap;

Model swp 'simple warehouse problem' / all /;

solve swp minimizing cost using lp;
```

Obr.: 12 Model Warehouse (9 GAMS)

```
---- VAR stock  stock stored at time t (units)

          LOWER          LEVEL          UPPER         MARGINAL

q-1         .          100.0000       100.0000        -1.0000
q-2         .             .           100.0000         5.0000
q-3         .             .           100.0000          EPS
q-4         .             .           100.0000        10.0000

---- VAR sell  stock sold at time t   (units)

          LOWER          LEVEL          UPPER         MARGINAL

q-1         .             .             +INF            EPS
q-2         .          100.0000         +INF             .
q-3         .             .             +INF             .
q-4         .             .             +INF             .

---- VAR buy  stock bought at time t (units)

          LOWER          LEVEL          UPPER         MARGINAL

q-1         .           50.0000         +INF             .
q-2         .             .             +INF            EPS
q-3         .             .             +INF            EPS
q-4         .             .             +INF            EPS

                         LOWER          LEVEL          UPPER         MARGINAL

---- VAR cost             -INF         -600.0000        +INF             .

  cost  total cost                 ($)
```

Obr. 13 Výsledky modelu skladových zásob (9 GAMS)

Při optimalizaci nám vyšel výsledek, že na začátku bylo ve skladu 100 jednotek zboží, poté se všech 100 jednotek zboží prodalo a jediné co zbylo, bylo zboží pevně zadané jako **istock** (počáteční množství skladu) o velikosti 50 jednotek. Celkové náklady na skladování však máme na hodnotě -600 Kč, což je pro nás výsledek dobrý, jelikož za skladování jsme nic nezaplatili a ještě jsme na skladu 600 Kč vydělali.

3.3 Jak začít zpracování semestrální práce

V další části práce se zaměřuji na rady, jak se systematicky věnovat prohlubování znalostí o GAMSu tak, aby to byly užitečné pro zájemce zaměřeného na rizikové inženýrství. Po konzultaci s vedoucím své práce jsem se rozhodl zúročit své několikaměsíční nabyté zkušenosti z týmové spolupráce na řešení reálného problému pomocí optimalizačního modelování a tyto dosud nesdílené zkušenosti zde původním způsobem prezentovat.

3.3.1 Téma

Pokud chcete rozvíjet znalosti některého programovacího nástroje, v našem případě GAMSu, nejvhodnější je věnovat se konkrétní optimalizační úloze. V první řadě je tedy nutné si zvolit téma a identifikovat, jak se úloha bude řešit, tj. zda půjde o problém se sklady, rozmanitostí výrobního programu nebo o dopravní problém. O dopravě se zmiňuji z jednoho prostého důvodu a to proto, že tuhle úlohu jsem řešil ve svém případě. Projekt v semestru sice řeší skupina studentů, ale v případě naší skupiny byl GAMS čistě mojí záležitostí. Pro zpracování byl společně vybrán v té době aktuální problém, a to nedostatek plicních ventilátorů vzniklý kvůli situaci způsobené virem Covid-19. Řešila se doprava ventilátorů z nemocnic s „přebytečnými" kapacitami do nemocnic s nedostatečnými kapacitami těchto přístrojů. Data jsou použita reálná z webu Ministerstva zdravotnictví a je s nimi pracováno po celou dobu zpracovávání práce. Poté v rozšířených modelech na „velká data" jsou hodnoty doplněné už o fiktivní počty, ale v reálně možném rozsahu, žádné přehnané hodnoty. Uvedené vzdálenosti jsou zjištěné pomocí Google maps a tudíž jsou reálné. Následně jsme vytvářeli společně matematický model a jednotlivě i jeho implementaci v Řešiteli Excelu. Než přejdu k dalšímu kroku zpracování, kterým bude vytvoření prvního modelu v GAMSu, rád bych zde uvedl některé zajímavé, dosud nesdílené vlastní zkušenosti s formulací a analýzou zvoleného problému, které nebyly předmětem hodnocení zmiňované týmové práce.

Pro začátek práce je dobré identifikovat, jakého charakteru řešená optimalizační úloha bude, zda budete řešit dopravní problém, problém skladu, a také zda se bude jednat o úlohu minimalizační nebo maximalizační či jiného typu. Po prozkoumání GAMSu a získání přehledu o tom, jakým modelem by se Vaše práce dala řešit je důležité mít data pečlivě setříděna, k utřídění dat a získání lepšího přehledu o tom, co by se vlastně mělo realizovat mi osobně dost pomohlo zpracovávání práce v Řešiteli Excelu, tam si přehledně uspořádáte vstupní data, proměnné a definujete účelovou funkci a díky těmto věcem i ve většině případů lze lépe pochopit, jak má potom práce probíhat v programu GAMS, co bude ve vztazích, navazovat na sebe a co je náš cíl výstupu v GAMSu.

3.3.2 Začínáme v GAMSu

Nejprve si vytvoříme na počítači pracovní složku, kde budeme náš model v GAMSu tvořit. Potom spustíme GAMS IDE podobně jako jiný program ve Windows s využitím ikony program na ploše počítače. Po spuštění GAMS IDE vidíme základní obrazovku. Při prvním spuštění GAMSu musíme nastavit řešitele. V menu File zvolit položku Options a v otevřeném formuláři pak záložku Solvers, kde vybereme řádek řešitele CPLEX a zaklikneme pole ve sloupci LP (lineární programování). Tím jsme zvolili optimalizační postup pro naši úlohu. Dále pro naši úlohu vytvoříme projekt v menu File a položce Project a dílčí položce New Project, kde zapíšeme podle své volby název projektu v námi dříve vytvořené složce, kde se vytvoří soubor s příponou gpr (odvozenou od slov GAMS project). Pak si vytvoříme pracovní soubor v menu File a položce New. Zvolíme název souboru, který se vytvoří ve složce projektového souboru, kam se dále implicitně umisťují všechny naše soubory související s projektem. Pak již můžeme využít zkušenosti ze čtení kódů zpracovaných jinými autory a diskutovanými v předchozích kapitolách a můžeme začít psaní toho našeho zápisu modelu v GAMSu. Při zahájení práce s GAMSem bych ale v první řadě všem doporučil psát bez diakritiky, třeba GAMS IDE zobrazí i háčky a čárky, ale pokud budete dělat na více zařízeních a chtít větší kompatibilitu, tak doporučím používat GAMS Studio a tam se písmena s diakritikou zobrazují tímhle symbolem: „�“, a proto by vše napsané s diakritikou bylo nepoužitelné. Dále si proto pro zpracovávání témata zvolím GAMS Studio. Po zapnutí GAMS Studia je práce s ním jednodušší a intuitivnější, není totiž nutné nastavovat vše, co jsem výše popsal v rámci GAMS IDE rozhraní. Řešitel je nastavený defaultně a program GAMS

Studio sám rozezná, jakého řešitele pro úlohu použít. Také zde odpadá nutnost vytvářet pracovní složku pro projekt v GAMSu a nutnost tuto složku mít implementovanou do GAMSu, nicméně Vám doporučím i přes to si nějakou složku pro GAMSy vytvořit. Pokud děláte jeden model a ten umístíte do složky jiného programu nebo k jiným věcem nesouvisejícím s GAMSem, tak se v nich ještě asi vyznáte, ale po vytvoření většího množství modelů, například 20, již začnete mít s orientací v souborech nejspíše problém. Berte v úvahu i fakt, že po vyřešení modelu klávesou F9 nebo zelenou šipkou v GAMS Studiu se k modelu vytvoří ještě lst soubor s výsledky a v souborech nastane chaos. Na samém začátku, když ještě nemám žádný model sestavený, ve studiu zmáčknu tlačítko F6 a jelikož vybrané téma je dopravní úloha, tak napíšu „trnsport“, ve vyhledávači modelů mi zůstane jediný model s číslem 001 a v popisu je uvedeno „A Transportation Problem“, na tento si kliknu dvakrát a naskočí kostra modelu v podobě, kterou je třeba upravit a poté v pozdějších fázích i rozšířit podle potřeb a stupňované náročnosti modelů. Základní jednoduchý model je deterministický.

3.4 Deterministický model

Pokud se zabýváme v rizikovém inženýrství rozhodovacími modely s náhodnými parametry, tak vyjděme z toho, že základní model každého problému bude deterministický. Jsou zde stanoveny jen zjištěné nebo určené hodnoty, bez náhodnosti, tudíž zatím pracujeme za bezrizikových podmínek. V mých modelech neuvádím požadavek na celočíselné řešení, protože z literatury viz. Klapka plyne, že to není nutné, protože pro celočíselné poptávky je to zaručené. (5)

```
Sets
       i   dodavajici kraje   / jihomoravsky, pardubicky, olomoucky, vysocina /
       j   poptavajici kraje    / praha, stredocesky, jihocesky, plzensky /;
Parameters a(i) kapacity volnych ventilatoru v dodavatelskych krajich
         /   jihomoravsky              75
             pardubicky                12
             olomoucky                 27
             vysocina                  20
                                           /
       b(j)  celkova poptavka poptavajicich kraju
         /    praha          40
              stredocesky    25
              jihocesky      0
              plzensky       10            / ;
Table d(i,j)  vzdalenost v km
                           praha         stredocesky        jihocesky       plzensky
      jihomoravsky          205             227                217             291
      pardubicky            126             119                211             220
      olomoucky             280             243                296             370
      vysocina              163             190                136             254;
Scalar f   cena za km prepravy na jeden kus  /1/ ;
Parameter c(i,j)   cena prepravy na 1 km ;
          c(i,j) = f * d(i,j) ;
  variable z1; positive variable x(i,j);
  Equations
       cost         ucelova funkce
       supply(i)    omezeni poctem volnych kusu dodavajicich kraju
       demand(j)    uspokojeni poptavky kraju j ;
cost ..         z1   =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))   =e=   b(j) ;
Model doprava /all/ ;
Solve doprava using lp minimizing z1 ;
Display x.l, z1.l;
```

Obr.: 14 deterministický model (9 GAMS)

Tohle je můj první model, který jsem tvořil sám a musím říct, že v začátcích je nejistota ze samostatného tvoření obrovská, takže mě vnitřní pocit nutil neustále koukat do již existujících modelů, jestli je vše tak, jak by být mělo nebo jestli jsem se někdy v kódu odchýlil a udělal chybu. Chce to nebát se a vytvářet si svůj vlastní kód a ten do sebe skládat jako příběh, jelikož každý řešený problém je unikátní a proto jsem se stejně dostal do bodu, kdy jsem musel přestat nahlížet do jiných prací nebo modelů v modelové knihovně GAMSu, zamyslet se nad vlastní problematikou a té se snažit porozumět a implementovat toto porozumění do zdrojového kódu. Zde již nebudu popisovat jednotlivé segmenty zdrojového kódu, jelikož jeho stavba je stejná, jako u modelu transport probíraného výše v kapitole 3.2.1. Kód budu dále rozebírat v kapitolách následujících se složitějšími modely. Všechna data pro pochopení modelu vyčtete z obrázku Obr. 14 deterministický model, jelikož veškerá data vstupující do modelu musí být zahrnuta ve zdrojovém kódu. Nyní se podíváme na výsledky modelu po spuštění optimalizace.

```
----     34 VARIABLE x.L

                   praha  stredoces~    plzensky

jihomoravsky      20.000      13.000      10.000
pardubicky                    12.000
vysocina          20.000


----     34 VARIABLE z1.L                  =    14649.000
```

Obr. 15 Výsledky deterministického modelu (9 GAMS)

V tabulce na Obr. 15 Výsledky deterministického modelu vidíme počty kusů plicních ventilátorů přepravovaných mezi městy a níže je uvedena celková cena přepravy po proběhnutí optimalizace, která činí 14 649 Kč.

3.5 Modely wait-and-see

Tato část navazuje na předchozí kapitolu 2.8.3, která stručně uvedla přístup WS. S ohledem na to, že začínajícímu uživateli chci ukázat, jak komplikovanější úlohy stochastického programování vyřešíme pomocí přechozích deterministických modelů, zvolíme možnost jejich opakovaného použití po drobné úpravě.

3.5.1 Model WS 1

Tento model tedy vychází z deterministického modelu. Zkratka WS1 značí, že jde o model wait-and-see pro první scénář. Poptávka je v tomto modelu jiná a také se model stává o něco složitějším přidáním náhodnosti a tímto krokem pro nás začíná být úloha důležitější, protože náhodnost je spojená s rizikem, proto modelujeme problémy úloh v GAMSu, jelikož je zde možné naprogramovat úlohu na optimalizaci a vzít v potaz riziko, které působí vždy ve všech odvětvích a je charakteristické pro dynamicky se vyvíjející problémy světa.

Obr.: 16 Model Wait and see scénář 1 (9 GAMS)

```
Sets
       i    dodavajici kraje   / jihomoravsky, pardubicky, olomoucky, vysocina /
       j    poptavajici kraje          / praha, stredocesky, jihocesky, plzensky /
       s    scenar                     / s1 /           ;
Parameters a(i) kapacity volnych ventilatoru v dodavatelskych krajich
         /   jihomoravsky              75
             pardubicky                12
             olomoucky                 27
             vysocina                  20
                                           /
       b(j)  celkova poptavka poptavajicich kraju
         /    praha          40
              stredocesky    25
              jihocesky      10
              plzensky       15            / ;
Parameter p(s) / s1    0.4 / ;
Table d(i,j)  vzdalenost v km
                            praha         stredocesky      jihocesky      plzensky
      jihomoravsky           205             227              217            291
      pardubicky             126             119              211            220
      olomoucky              280             243              296            370
      vysocina               163             190              136            254;
Scalar f  cena za km prepravy na jeden kus  /1/ ;
Parameter c(i,j)  cena prepravy na 1 km ; c(i,j) = f * d(i,j) ;
variable z1; positive variable x(i,j);
Equations
       cost        ucelova funkce
       supply(i)   omezeni poctem volnych kusu dodavajicich kraju
       demand(j)   uspokojeni poptavky kraju j ;
cost ..        z1  =e=  sum((i,j), c(i,j)*x(i,j)) * sum (s,p(s)) ;
supply(i) ..   sum(j, x(i,j))  =l=  a(i) ;
demand(j) ..   sum(i, x(i,j))  =e=  b(j) ;
Model transport2 /all/ ;
Solve transport2 using lp minimizing z1 ;
Display x.l, z1.l ;
```

Vidíte, že při programování modelu se dá dobře vycházet ze základního modelu, přidáváme scénář **s** do **`Sets`**, **`parameter` p(s)** s hodnotou 0,4, což je moje zvolená pravděpodobnost pro tento scénář. Ke konci v zápisu rovnice je nutné tento parametr zahrnout do výpočtu celkových nákladů, což lze buď provést pomocí přiřazovacího příkazu (viz. model trnsport a výpočet dopravní ceny) nebo jej lze zahrnout do účelové funkce, jak je uvedeno. Pokud by se v tomto případě několik scénářů shodovalo, můžeme pak jim odpovídající pravděpodobnosti sečíst, což je zde zahrnuto.

```
----      35 VARIABLE x.L

                   praha  stredoces~   jihocesky    plzensky

jihomoravsky      30.000      13.000                  15.000
pardubicky                    12.000
vysocina          10.000                  10.000


----      35 VARIABLE z1.L                   =      7153.600
```

Obr. 17 Výsledky modelu WS1 (9 GAMS)

S touto hodnotou celkové náklady scénáře vycházejí na 7153,6 Kč. Tento výpočet odpovídá způsobu vyhodnocení rizika v rizikovém inženýrství, kde vyhodnocujeme riziko pomocí součinu pravděpodobnosti a dopadu.

3.5.2 Model WS 2

Tento model vychází z modelu wait-and-see pro první scénář. V tomto modelu je změněná poptávka po zboží na vyšší hodnotu než v prvním případě, a také pravděpodobnost nastání tohoto scénáře je vyšší a to s hodnotou 0,6, protože logicky spíše očekáváme, že reálnější bude vyšší poptávka po vytížených přístrojích.

```
Sets
      i    dodavajici kraje   / jihomoravsky, pardubicky, olomoucky, vysocina /
      j    poptavajici kraje          / praha, stredocesky, jihocesky, plzensky /
      s    scenar                     / s2 /            ;
 Parameters a(i) kapacity volnych ventilatory v dodavatelskych krajich
       /   jihomoravsky               75
           pardubicky                 12
           olomoucky                  27
           vysocina                   20
                                          /
           b(j)  celkova poptavka poptavajicich kraju
       /    praha          60
            stredocesky    35
            jihocesky      5
            plzensky       10            / ;
Parameter p(s) / s2    0.6 / ;
Table d(i,j)  vzd�lenost v km
                          praha         stredocesky      jihocesky      plzensky
      jihomoravsky         205            227              217            291
      pardubicky           126            119              211            220
      olomoucky            280            243              296            370
      vysocina             163            190              136            254;

Scalar f  cena za km prepravy na jeden kus  /1/ ;
Parameter c(i,j)  cena prepravy na 1 km ; c(i,j) = f * d(i,j) ;
variable zl; positive variable x(i,j);
Equations
       cost        ucelova funkce
       supply(i)   omezeni poctem volnych kusu dodavajicich kraju
       demand(j)   uspokojeni poptavky kraju j ;
cost ..        zl  =e=  sum((i,j), c(i,j)*x(i,j)) * sum (s,p(s)) ;
supply(i) ..   sum(j, x(i,j))  =l=  a(i) ;
demand(j) ..   sum(i, x(i,j))  =e=  b(j) ;
Model transport2 /all/ ;
Solve transport2 using lp minimizing zl ;
Display x.l, zl.l ;
```

Obr. 18 Model WS2 (9 GAMS)

V modelu se podívejte do popisku tabulky **`Table`**, kde zůstalo nepřepsané dlouhé „á“ ve slově vzdálenost a zobrazuje se nám ikonka otazníku, zmiňovaná výše v kapitole 3.3.2. S konstrukcí tohoto modelu již nebyl takový problém právě díky tomu, že konstrukce modelu zůstala stejná a mění se jen vstupní data.

```
----     36 VARIABLE x.L

                  praha  stredoces~   jihocesky    plzensky

jihomoravsky     45.000      20.000                  10.000
pardubicky                   12.000
olomoucky                     3.000
vysocina         15.000                   5.000


----     36 VARIABLE zl.L                  =     13174.200
```

Obr. 19 Výsledky modelu WS2 (9 GAMS)

Na obrázku vidíme počty přepravovaných kusů plicních ventilátorů a při prvním pohledu lze říci, že hodnoty jsou vyšší než u modelu WS1, což se dále promítá i na celkových nákladech přepravy, které jsou v tomto modelu téměř dvojnásobné a to 13 174,2 Kč.

3.5.3 Očekávané budoucí náklady ze scénářů modelů WS

Pokud bych počítal optimální hodnotu účelové funkce pro scénáře WS, dosadím do zdrojového kódu modelu jen hodnotu poptávky bez pravděpodobností a vzorec pro optimální hodnotu WS by byl:

$$z_{min}^{WS1} \text{ a } z_{min}^{WS2}$$

V dalším kroku vynásobíme optimální hodnotu účelové funkce pravděpodobností scénáře. V mých modelech v GAMSu je už rovnou hodnota vynásobena s pravděpodobností, pro tento vztah platí vzorec:

$$z_{min}^{WS1} * p_1 \text{ a } z_{min}^{WS2} * p_2$$

Nyní už se dostávám k jádru věci, kvůli které jsem výše zmiňoval dané vzorce a kterou chci v této kapitole spočítat a to jsou očekávané budoucí (průměrné) náklady přepravy.

$$z^{WS} = E\left[z_{min}^{WS(xi)}\right] = z_{min}^{WS1} * p_1 + z_{min}^{WS2} * p_2$$

Do rovnice dosadíme výsledky ze scénářů 1 a 2 a výpočet by měl vypadat následovně.

$$z^{WS} = E\left[z_{min}^{WS(xi)}\right] = 7153{,}6 + 13174{,}2$$

$$z^{WS} = 20327{,}8\ Kč$$

Očekávané budoucí náklady vycházející ze scénářů modelu WS mi ve výsledku vyšly s v celkové hodnotě 20 327,8 Kč.

3.6 Modely here-and-now

Jedná se o modely, kdy potřebujeme nyní udělat rozhodnutí a nemůžeme čekat déle na další posouzení scénářů. V této kapitole budeme navazovat na úvod do problematiky here-and-now modelů viz kapitola 3.3.3. Do výpočtů zde vstupují oba scénáře naráz. Scénáře se porovnávají v různých vztazích. HN_IS popisuje, zda poptávka zvolená jako výchozí pokryje poptávané množství i pro poptávku sekundární. HN_EV zprůměruje poptávku obou scénářů a pak zpětně porovnává, jestli by bylo řešení pro zprůměrované hodnoty obou scénářů schopno oběma scénářům vyhovět , pokud průměrná poptávka nestačí zásobovat byť jen jeden kraj, tak se scénář průměrné poptávky zamítá. HN_MM je model, který opět spojuje oba poptávkové scénáře. Tentokrát však pro každý kraj zvlášť určí z obou scénářů nejvyšší poptávku a pro tu pak optimalizuje model, takže logicky jde o model s nejvyšším počtem přepravovaných kusů ventilátorů a s nejvyššími celkovými náklady přepravy.

3.6.1 Model HN_IS

V tomto kroku ověříme, zda námi vybraná výchozí poptávka je schopna zároveň splnit požadavky druhé poptávky pro všechny odběratelské kraje. Pokud není výchozí poptávka schopna splnit i druhou poptávku zároveň model nám vyjde s hodnotou infeasible, neboli nemá řešení v GAMSu. Jak lze vidět na zdrojovém kódu níže pro model HN_IS, vyskytuje se nám zde několik nových parametrů, hodnot a proměnných. V **`Parameters`** máme nyní uvedeny oba scénáře. Do **`Parameter`** zahrnuje rovněž pravděpodobnosti obou scénářů. Dále pod **`Scalar`** přidáváme Parameter **bb(j).** Pod zobrazovací příkaz **`Display`** jsem ještě umístil **`Parameter`** **zIS**, **xIS(i,j)** a v posledních 4 řádcích modelů je zapsána verifikace pro výchozí (vyšší) scénář 2.

```
Sets    i   dodavatelske kraje   / jihomoravsky, pardubicky, olomoucky, vysocina /
        j   poptavajici kraje          / praha, stredocesky, jihocesky, plzensky /
        s   scenare            / s1, s2 /   ;
Parameters
        a(i)  volne kapacity v dodavajicich krajich
          /   jihomoravsky            75
              pardubicky              12
              olomoucky               27
              vysocina                20
                                         /
        b(j,s)  poptavka v krajich s prictenym scenarem
          /    praha.s1          40
               stredocesky.s1    25
               jihocesky.s1      10
               plzensky.s1       15

               praha.s2          60
               stredocesky.s2    35
               jihocesky.s2      5
               plzensky.s2       10               /  ;
Parameter p(s)
    /  s1   0.4, s2   0.6/ ;
Table d(i,j)  vzdalenost v km
                          praha        stredocesky      jihocesky      plzensky
      jihomoravsky         205           227              217            291
      pardubicky           126           119              211            220
      olomoucky            280           243              296            370
      vysocina             163           190              136            254 ;
Scalar f  cena za kus na km prepravy  /1/ ;
Parameter bb(j) ;
Parameter c(i,j)  cena prepravy v korunach na jeden kilometr ;
            c(i,j) = f * d(i,j) ;
variable z;
positive variable x(i,j);
Equations
        cost          ucelova funkce
        supply(i)     omezeni poctem volnych kusu dodavajicich kraju
        demand(j,s)   uspokojeni poptavky kraju j    ;
cost ..          z  =e=  sum((i,j), c(i,j)*x(i,j)  * sum(s,p(s)));
supply(i) ..    sum(j, x(i,j))  =l=  a(i) ;
demand(j,s) ..   sum(i, x(i,j)) =e=  b(j,s) ;
Model doprava /all/ ;
Solve doprava using lp minimizing z ;
display z.L, x.L, doprava.modelstat ;

parameter zIS,xIS(i,j);

zIS = z.L; xIS(i,j) = x.L(i,j); x.LO(i,j) = xIS(i,j); x.UP(i,j) = xIS(i,j);
bb(j) = b(j,"s2") ;
solve doprava minimizing z using LP ;
display doprava.modelstat, z.L,x.L;
```

Obr. 20 Model HN_IS (9 GAMS)

```
--- LP status (3): infeasible.
--- Cplex Time: 0.00sec (det. 0.00 ticks)


Model has been proven infeasible
```

Obr. 21 Výsledek modelu HN_IS (9 GAMS)

Jak lze vidět, optimalizace modelu nám ve výsledku nevypsala žádné náklady přepravy, ale jen informaci, že model nemá řešení.

```
parameter zIS,xIS(i,j);

zIS = z.L; xIS(i,j) = x.L(i,j); x.LO(i,j) = xIS(i,j); x.UP(i,j) = xIS(i,j);
bb(j) = b(j,"s2") ;
solve doprava minimizing z using LP ;
display doprava.modelstat, z.L,x.L;
```

Obr. 22 Nové řádky v modelu HN_IS (9 GAMS)

Největší problém mi při zpracování modelu dělalo pochopit, proč se vlastně musely přidávat tyto 4 řádky na konec zdrojového kódu, proč jsem nemohl spustit optimalizaci s kódem bez těchto nových věcí, ono by to samozřejmě šlo, ale model by ukazoval jen obrovskou celkovou cenu, a to by nebylo správné řešení tohoto modelu. Tyto nové přírůstky do zdrojového kódu zajišťují, že dojde k verifikaci scénáře 2 pro ostatní scénáře, v tomto případě pro scénář 1, a protože scénář 2 není schopen splnit veškerou poptávku scénáře 1, model nemá žádné přípustné řešení.

3.6.2 Model HN_EV

Tento model HN_EV značí, že velikost poptávky máme v očekávaných hodnotách, kterých dosáhneme součtem součinu poptávky scénáře 1 s pravděpodobností scénáře 1 a součinu poptávky scénáře 2 s pravděpodobností scénáře 2. Po vyřešení HN_EV úlohy výpočtem ověřujeme, zda EV řešení (tj. řešení pro očekávanou hodnotu poptávky) bude schopno splnit požadavky poptávky jak scénáře 1, tak scénáře 2. Po spuštění optimalizace vychází hodnota 20 299 Kč, což je zdánlivě správná (konečná) hodnota nákladů, ale dále se ve výsledcích píše, že pro ověření EV hodnoty na scénář 1 i 2 „model has been proven infeasible“ což znamená, že model nemá přípustné řešení, a tedy nelze nalézt optimální řešení. Tato situace je způsobená tím, že očekávaná hodnota poptávky nevyhovuje poptávce 1 v Jihočeském a Plzeňském kraji a poptávce 2 v Praze a Středočeském kraji.

```
Sets     i   dodavatelske kraje   / jihomoravsky, pardubicky, olomoucky, vysocina /
         j   poptavajici kraje          / praha, stredocesky, jihocesky, plzensky /
         s   scenare            / s1, s2 /     ;
Parameters
         a(i)  volne kapacity v dodavajicich krajich
           /   jihomoravsky              75
               pardubicky                12
               olomoucky                 27
               vysocina                  20
                                             /
         b(j,s)  poptavka v krajich s prictenem scenarem
           /     praha.s1          40
                 stredocesky.s1    25
                 jihocesky.s1      10
                 plzensky.s1       15

                 praha.s2          60
                 stredocesky.s2    35
                 jihocesky.s2      5
                 plzensky.s2       10                /  ;
Parameter p(s)
    /  s1    0.4,  s2    0.6/ ;
Table d(i,j)  vzdalenost v km
                            praha         stredocesky       jihocesky      plzensky
      jihomoravsky           205             227              217            291
      pardubicky             126             119              211            220
      olomoucky              280             243              296            370
      vysocina               163             190              136            254 ;
Scalar f  cena za kus na km prepravy  /1/ ;
Parameter bb(j);
bb(j) = sum(s, p(s) * b(j,s));
Parameter c(i,j)  cena prepravy v korunach na jeden kilometr ;
            c(i,j) = f * d(i,j)  ;
variable z;
positive variable x(i,j);
Equations
       cost          ucelova funkce
       supply(i)     omezeni poctem volnych kusu dodavajicich kraju
       demand(j,s)  uspokojeni poptavky kraju j    ;
cost ..          z  =e=  sum((i,j), c(i,j)*x(i,j));
supply(i) ..     sum(j, x(i,j))  =l=  a(i) ;
demand(j,s) ..    sum(i, x(i,j)) =e=  bb(j) ;
Model doprava /all/ ;
Solve doprava using lp minimizing z ;
display z.L, x.L, doprava.modelstat;

parameter zEV,xEV(i,j);

zEV = z.L; xEV(i,j) = x.L(i,j); x.LO(i,j) = xEV(i,j); x.UP(i,j) = xEV(i,j);
bb(j) = b(j,"s1");
solve doprava minimizing z using LP;
display doprava.modelstat, z.L,x.L;

bb(j) = b(j,"s2");
solve doprava minimizing z using LP;
display doprava.modelstat, z.L,x.L;
```

Obr.: 23 Model EV (9 GAMS)

Nyní se podíváme na to, co se v modelu muselo změnit, aby z HN_IS vznikl model HN_EV. Pro lepší přehlednost jsem nové věci zakroužkoval červenou barvou, protože **`Parameter`** **bb(j)** je schovaný uprostřed zdrojového kódu. Díky rovnici, že se parametr **bb(j)** = součinu pravděpodobností **p(s)** scénářů a hodnot požadovaných kusů přepravy **b(j,s).** Na konci kódu pak máme nadefinované zEV, přes které budeme ověřovat platnost výpočtu pro scénáře 1 a 2. Jak si můžete všimnout, prvně ověřuji platnost modelu pro scénář 1, kde mám nadefinované **bb(j) = b(j,“s1“)** a níže potom pro scénář 2, kde je to realizováno zápisem **bb(j) = b(j,“s2“).**

3.6.3 Model HN_MM

Jedná se o další z řady here-and-now modelů, tentokrát se však optimalizace bude týkat nákladů přepravy, pokud vezmeme v potaz vždy nejvyšší hodnotu poptávky pro daný kraj napříč oběma scénáři. Proto název HN_MM, kde MM zkracuje zápis min max, čili že hledáme nejnižší dopravní náklady pro nejhorší (nejdražší) poptávky. Hodnoty poptávky 2 budeme brát pro Prahu a Středočeský kraj a hodnotu poptávky 1 pro Jihočeský a Plzeňský kraj, protože v těchto krajích jsou tyto dané poptávky nejvyšší. Toho se v GAMSu dosáhne tím, že chceme, aby řešení splňovalo omezující podmínky pro všechny scénáře. Protože náhodnost není zahrnuta do dopravních cen v naší úloze, nemusím zde řešit implementaci min max přístupu v účelové funkci.

```
Sets
       i   dodavatelske kraje   / jihomoravsky, pardubicky, olomoucky, vysocina /
       j   poptavajici kraje          / praha, stredocesky, jihocesky, plzensky /
       s   scenare            / s1, s2 /     ;
Parameters
       a(i)  volne kapacity v dodavajicich krajich
         /   jihomoravsky               75
             pardubicky                 12
             olomoucky                  27
             vysocina                   20
                                            /
       b(j,s)  poptavka v krajich s prictenym scenarem
         /    praha.s1          40
              stredocesky.s1    25
              jihocesky.s1      10
              plzensky.s1       15

              praha.s2          60
              stredocesky.s2    35
              jihocesky.s2      5
              plzensky.s2       10               /  ;
Parameter p(s)
    /  s1   0.4, s2   0.6/ ;
Table d(i,j)  vzdalenost v km
                         praha         stredocesky      jihocesky      plzensky
      jihomoravsky        205            227              217            291
      pardubicky          126            119              211            220
      olomoucky           280            243              296            370
      vysocina            163            190              136            254 ;
Scalar f  cena za kus na km prepravy  /1/ ;
Parameter c(i,j) cena prepravy v korunach na jeden kilometr ;c(i,j) = f * d(i,j);
variable z2;positive variable x(i,j);
Equations
cost        ucelova funkce
supply(i)   omezeni poctem volnych kusu dodavajicich kraju
demand(j,s) uspokojeni poptavky kraju j   ;
  cost ..          z2  =e=  sum((i,j), c(i,j)*x(i,j)) ;
  supply(i) ..     sum(j, x(i,j))   =l=  a(i) ;
  demand(j,s) ..    sum(i, x(i,j))   =G=  b(j,s) ;
Model transport2 /all/ ;
Solve transport2 using lp minimizing z2 ;
display z2.L, x.L;
```

Obr.: 24 Model HN_MM (9 GAMS)

Jak lze vidět z obrázku, docílit toho, aby model počítal vždy tu nejvyšší poptávku mezi všemi scénáři, je vcelku jednoduchá úprava. Stačí v omezujících podmínkách k poptávce uvést, že počet kusů **x(i,j)** má být větší nebo roven hodnotám v parametru **b(j,s)**, kde máme uvedeny poptávky obou scénářů, při optimalizaci si model automaticky načítá pro každé město tu nejvyšší poptávku skrze všemi scénáři.

```
----     42 VARIABLE z2.L                  =    24462.000

----     42 VARIABLE x.L

                  praha  stredoces~   jihocesky    plzensky

jihomoravsky     50.000      10.000                  15.000
pardubicky                   12.000
olomoucky                    13.000
vysocina         10.000                  10.000
```

Obr. 25 Výsledky modelu HN_MM (9 GAMS)

Z obrázku výše vidíme, že hodnota účelové funkce **z2** vychází 24 462 Kč, což jsou celkové náklady na přepravu tohoto modelu. V tabulce pod celkovými náklady je znázorněný počet přepravovaných kusů mezi dodavateli a odběrateli.

3.6.4 Model HN_TS

Jedná se o dvojstupňový model, HN_TS znamená zkratky pro slova here-and-now a two stage. Změna v tomto modelu oproti předchozím modelům tkví v jeho větší míře reality a použitelnosti do praxe. Zde nám do modelu vstupují hodnoty poptávek z obou scénářů včetně jejich pravděpodobností a navíc nově každý ze scénářů má své náklady přepravy na kus dodatečné jednotky plicního ventilátoru. Do pole pro definici výpočtu účelové funkce je proto nutné zahrnout i tyto dodatečné náklady, které máme v modelu nadefinovány viz. obrázek modelu.

```
Sets    i   dodavatelske kraje   / jihomoravsky, pardubicky, olomoucky, vysocina /
        j   poptavajici kraje          / praha, stredocesky, jihocesky, plzensky /
        s   scenare            / s1, s2 /    ;
Parameters
        a(i)  volne kapacity v dodavajicich krajich
          /   jihomoravsky              75
              pardubicky                12
              olomoucky                 27
              vysocina                  20
                                            /
        b(j,s)  poptavka v krajich pro oba scenare
          /    praha.s1          40
               stredocesky.s1    25
               jihocesky.s1      10
               plzensky.s1       15

               praha.s2          60
               stredocesky.s2    35
               jihocesky.s2      5
               plzensky.s2       10               /  ;
Parameter p(s)
    /  s1   0.4, s2   0.6/ ;
Parameter qplus(j) /praha 300, stredocesky 350, jihocesky 350, plzensky 1000/;
Parameter qminus(j) /praha 10, stredocesky 10, jihocesky 10, plzensky 10 /;
Table d(i,j)  vzd�lenost v km
                          praha        stredocesky      jihocesky      plzensky
      jihomoravsky         205            227              217            291
      pardubicky           126            119              211            220
      olomoucky            280            243              296            370
      vysocina             163            190              136            254 ;
Scalar f  cena za kus na km prepravy  /1/ ;
Parameter c(i,j)  cena prepravy v korunach na jeden kilometr ;
          c(i,j) = f * d(i,j) ;
variable z2;
positive variable x(i,j);
positive variable yplus(j,s), yminus(j,s);
Equations
       cost         ucelova funkce
       supply(i)    omezeni poctem volnych kusu dodavajicich kraju
       demand(j,s)  uspokojeni poptavky kraju j   ;
cost ..         z2  =e=  sum((i,j), c(i,j)*x(i,j)) + sum(s,p(s) *
(sum(j, qplus(j)*yplus(j,s)) + sum(j, qminus(j)*yminus(j,s)))) ;
supply(i) ..   sum(j, x(i,j))  =l=  a(i) ;
demand(j,s) ..   sum(i, x(i,j)) + yplus(j,s) - yminus(j,s) =e=  b(j,s) ;
Model transport2 /all/ ;
Solve transport2 using lp minimizing z2 ;
display z2.L, x.L, yplus.L, yminus.L;
```

Obr.: 26 Model HN_TS (9 GAMS)

```
Parameter p(s)
    /  s1   0.4, s2   0.6/ ;
Parameter qplus(j) /praha 300, stredocesky 350, jihocesky 350, plzensky 1000/;
Parameter qminus(j) /praha 10, stredocesky 10, jihocesky 10, plzensky 10 /;
```

Obr. 27 Nové parametry v modelu HN_TS (9 GAMS)

Pod parametrem označujícím pravděpodobnosti scénářů modelu vidíme dva nové parametry pod příkazem **`Parameter`** a těmi jsou **qplus(j)** označující peněžní hodnotu penalizace za dodatečný kus převáženého přístroje a **qminus(j)** označuje peněžní hodnotu penalizace za odřeknutý kus přístroje z přepravy.

```
variable z2;
positive variable x(i,j);
positive variable yplus(j,s), yminus(j,s);
Equations
       cost         ucelova funkce
       supply(i)    omezeni poctem volnych kusu dodavajicich kraju
       demand(j,s)  uspokojeni poptavky kraju j    ;
cost ..        z2   =e=   sum((i,j), c(i,j)*x(i,j)) + sum(s,p(s) *
(sum(j, qplus(j)*yplus(j,s)) + sum(j, qminus(j)*yminus(j,s)))) ;
supply(i) ..   sum(j, x(i,j))   =l=  a(i) ;
demand(j,s) ..   sum(i, x(i,j)) + yplus(j,s) - yminus(j,s) =e=  b(j,s) ;
Model transport2 /all/ ;
Solve transport2 using lp minimizing z2 ;
display z2.L, x.L, yplus.L, yminus.L;
```

Obr. 28 Nově definovaná účelová funkce pro model HN_TS (9 GAMS)

Do **`positive variable`** jsme zahrnuli i nové parametry **yplus(j,s)** a **yminus(j,s)** a tyto nám dále vstupují společně s parametry **qplus(j)** a **qminus(j)** i do rovnice pro výpočet hodnoty celkových nákladů modelu. Do rovnice pro poptávku nám vstupují nově proměnné **yplus(j,s)** a **yminus(j,s)**.

```
----     47 VARIABLE z2.L                  =    23424.000

----     47 VARIABLE x.L

                 praha  stredoces~   jihocesky    plzensky

jihomoravsky    25.000      13.000                  15.000
pardubicky                  12.000
vysocina        15.000                   5.000


----     47 VARIABLE yplus.L

                  s1          s2

praha                     20.000
stredocesky               10.000
jihocesky        5.000


----     47 VARIABLE yminus.L

                  s2

plzensky         5.000
```

Obr. 29 Výsledky modelu HN_TS (9 GAMS)

Ve výsledcích jako první vidíme celkové náklady přepravy, které jsou 23 424 Kč. Pod náklady máme tabulku pro **x.L**, ve které je uveden počet přepravovaných kusů přístrojů a pod tabulkou níže jsou uvedeny tabulky **yplus.L**, která sděluje počet přiobjednaných kusů přístroje a **yminus.L**, která naopak udává počet odřeknutých kusů přístroje oproti objednávce.

3.7 Deterministický model pro velká data

U všech modelů pro velká data před zkratkou modelu uvádím ještě zkratku VS, což je můj popis, jak se v modelech vyznat a je to zkratka pro slova velký soubor. Podstata tohoto modelu zůstává stejná jako u výše popisovaného deterministického modelu, s tím rozdílem, že byly rozšířeny počty dodavatelů i odběratelů na dvojnásobek. Tenhle krok byl učiněn pro zvětšení role rizika, které vstupuje do modelů, a také pro zvýšení reálnosti projektu, jelikož v životě by se optimalizační úloha optimalizovala pro velký soubor dodavatelů a poptávajících.

Níže je vložen opět snímek obrazovky z programu GAMS Studio, kde je vidět napsaný zdrojový kód pro optimalizaci modelu na velkém souboru. Všimnou si můžete, že kód je prakticky totožný s prvním deterministickým modelem, pouze zde vidíme podstatně více údajů a možnost zápisu tabulky, který předtím nebyl potřebný. Pokud se totiž do tabulky nevejdou všechny údaje, tak aby byly viditelné, tzn. překračovaly by až za hranu stránky v programu, je možné do tabulky zapsat jen pár sloupců a potom pod tabulku napsat tzv. pokračovací znak **„+“.** Pod tento znak je opět nutné zapsat všechny řádky představující dodavatelské okresy a do sloupců už jen zbytek poptávajících okresů, které se nevešly do první tabulky. Další změněnou pak bude jen počet iterací dříve zvolené simplexové metody (v řešiteli CPLEX) lineárního programování, protože s dvojnásobným počtem dodavatelů a odběratelů vzroste i náročnost modelu na jeho výpočet, ale to v zápisu nijak nepoznáme kromě už mnou popsaných změn výše.

```
Sets i   dodavatelske okresy / ceskebudejovice, ceskykrumlov, jindrichuvhradec,
            pisek, prachatice, strakonice, tabor, blansko, breclav, hodonin /
     j   poptavajici okresy / benesov, beroun, kladno, kolin, kutnahora, melnik,
                              mladaboleslav, nymburk, pribram, rakovnik /  ;
Parameters a(i)  volne kapacity v dodavajicich okresech
         /   ceskebudejovice         54
             ceskykrumlov            45
             jindrichuvhradec        36
             pisek                   31
             prachatice              41
             strakonice              58
             tabor                   46
             blansko                 47
             breclav                 29
             hodonin                 50
                                           /
         b(j)  poptavka v okresech
         /    benesov       44
              beroun        42
              kladno        30
              kolin         28
              kutnahora     39
              melnik        35
              mladaboleslav 53
              nymburk       14
              pribram       37
              rakovnik      13  /  ;
Table d(i,j)  vzdalenost v km
                         benesov   beroun   kladno   kolin   kutnahora   melnik
      ceskebudejovice      100      176      159      162       159       200
      ceskykrumlov         127      174      168      160       145       212
      jindrichuvhradec     99       158      172      145       125       206
      pisek                90       186      148      156       189       226
      prachatice           146      190      186      176       159       243
      strakonice           130      192      182      145       145       216
      tabor                112      182      179      125       168       218
      blansko              200      260      252      212       231       356
      breclav              243      312      300      268       276       396
      hodonin              250      318      312      280       290       412
+
                              mladaboleslav   nymburk   pribram   rakovnik
      ceskebudejovice              226          212       100       212
      ceskykrumlov                 246          200       121       268
      jindrichuvhradec             238          218       112       248
      pisek                        249          238       118       218
      prachatice                   242          242       134       236
      strakonice                   236          226       126       246
      tabor                        212          216       145       186
      blansko                      268          300       265       312
      breclav                      312          328       315       336
      hodonin                      418          365       326       363 ;
Scalar f  cena za kus na km prepravy  /1/ ;
Parameter c(i,j)  cena prepravy v korunach na jeden kilometr ; c(i,j) = f * d(i,j) ;
variable z;positive variable x(i,j);
Equations cost        ucelova funkce
          supply(i)   omezeni poctem volnych kusu dodavajicich okresu
          demand(j) uspokojeni poptavky okresu j    ;
  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)) =e=  b(j) ;
Model transport2 /all/ ;
Solve transport2 using lp minimizing z ;
display z.L, x.L ;
```

Obr.: 30 Deterministický model pro velká data (9 GAMS)

```
----     63 VARIABLE z.L                  =    54955.000

----     63 VARIABLE x.L
```

	benesov	beroun	kladno	kolin	kutnahora	melnik	mladabole~	nymburk	pribram	rakovnik
ceskebudejovice	13.000					4.000			37.000	
ceskykrumlov			30.000			1.000		14.000		
jindrichuvhradec		25.000			11.000					
pisek	31.000									
prachatice		17.000					24.000			
strakonice					28.000	30.000				
tabor				28.000			5.000			13.000
blansko							24.000			

Obr. 31 Výsledky deterministického modelu pro velká data (9 GAMS)

Celkové náklady tohoto modelu vychází na 54 955 Kč, což při porovnání se základním deterministickým modelem je skoro čtyřnásobná hodnota, která je ale zapříčiněná vyšším množstvím dodavatelů a odběratelů a samozřejmě i vzdálenostmi mezi nimi, které je třeba při přepravě urazit. Na první pohled člověka asi zarazí rozsah kódu, když jsem začínal poprvé zpracovávat velká data, tak jsem měl strach, jak to vše půjde vložit do GAMSu, jak to bude vypadat a jestli bude program schopný pojmout tolik dat. Žádné z těchto věcí se však bát nemusíte, ve finále je to vlastně jednoduché. Nyní k tabulce, která na první pohled působí zvláštním dojmem díky rozdělení na dva úseky a spojení znakem **„+“,** i ta funguje přesně tak, jak má a není potřeba si o ni dělat starost. U velkých dat je kladen velký důraz na pečlivost při zadávání dat, musíte být důslední, jinak se snadno stane, že se překliknete, napíšete špatnou hodnotu na špatné místo a ihned bude výsledek po optimalizaci vycházet jinak a třeba budete dlouho hledat, kde se stala chyba, než ji v tomto rozsáhlém kódu objevíte.

3.8 Modely wait-and-see pro velká data

Základem zde je opět deterministický model pro velká data a v rámci modelů wait-and-see došlo k rozšíření poptávkových scénářů ze 2 na 5. Tímto krokem se také zvyšuje riziko modelů, protože každý model má jinou hodnotu poptávek a pravděpodobnost, se kterou může nastat.

3.8.1 Model VS_WS 1

```
Sets i   dodavatelske okresy /ceskebudejovice, ceskykrumlov, jindrichuvhradec,
            pisek, prachatice, strakonice, tabor, blansko, breclav, hodonin /
     j   poptavajici okresy /benesov, beroun, kladno, kolin, kutnahora, melnik,
                             mladaboleslav, nymburk, pribram, rakovnik /
      s   scenare            / s1 /    ;
Table d(i,j)  vzdalenost v km
                           benesov   beroun   kladno   kolin   kutnahora   melnik
      ceskebudejovice        100      176      159      162      159        200
      ceskykrumlov           127      174      168      160      145        212
      jindrichuvhradec       99       158      172      145      125        206
      pisek                  90       186      148      156      189        226
      prachatice             146      190      186      176      159        243
      strakonice             130      192      182      145      145        216
      tabor                  112      182      179      125      168        218
      blansko                200      260      252      212      231        356
      breclav                243      312      300      268      276        396
      hodonin                250      318      312      280      290        412
+
                                 mladaboleslav   nymburk   pribram   rakovnik
      ceskebudejovice                226            212       100       212
      ceskykrumlov                   246            200       121       268
      jindrichuvhradec               238            218       112       248
      pisek                          249            238       118       218
      prachatice                     242            242       134       236
      strakonice                     236            226       126       246
      tabor                          212            216       145       186
      blansko                        268            300       265       312
      breclav                        312            328       315       336
      hodonin                        418            365       326       363 ;
Parameter p(s) / s1   0.15 / ;
Parameters    a(i)  volne kapacity v dodavajicich okresech
         /    ceskebudejovice        54
              ceskykrumlov           45
              jindrichuvhradec       36
              pisek                  31
              prachatice             41
              strakonice             58
              tabor                  46
              blansko                47
              breclav                29
              hodonin                50
                                           /
       b(j,s)  poptavka v okresech s danym scenarem
         /     benesov.s1        53
               beroun.s1         52
               kladno.s1         35
               kolin.s1          32
               kutnahora.s1      42
               melnik.s1         37
               mladaboleslav.s1  60
               nymburk.s1        18
               pribram.s1        42
               rakovnik.s1       15 /  ;
Scalar f  cena za kus na km prepravy  /1/ ;
Parameter c(i,j) cena prepravy v korunach na jeden kilometr; c(i,j) = f * d(i,j);
variable z; positive variable x(i,j);
Equations cost        ucelova funkce
       supply(i)   omezeni poctem volnych kusu dodavajicich okresu
       demand(j,s) uspokojeni poptavky okresu j   ;
cost ..          z  =e=  sum((i,j), c(i,j)*x(i,j)) * sum(s, p(s)) ;
supply(i) ..    sum(j, x(i,j))  =l=  a(i) ;
demand(j,s) ..   sum(i, x(i,j)) =g=  b(j,s) ;
Model vswaitandseel /all/ ;
Solve vswaitandseel using lp minimizing z ;
display z.L, x.L;
```

Obr. 32 Model VS_WS1 (9 GAMS)

Když se u tohoto modelu podíváme na poptávku, která je vyšší než u modelu deterministického, tak by se na první pohled dalo říci, že náklady přepravy budou vyšší, ale kvůli tomu, že nám do modelu opět vstupuje i pravděpodobnost, jsou náklady o dost nižší, konkrétně 10 096,65 Kč jak můžete vidět níže na obrázku s výsledky optimalizace.

```
----     65 VARIABLE z.L                   =     10096.650

----     65 VARIABLE x.L
```

	benesov	beroun	kladno	kolin	kutnahora	melnik	mladabole~	nymburk	pribram	rakovnik
ceskebudejovice	12.000								42.000	
ceskykrumlov		6.000	21.000					18.000		
jindrichuvhradec	10.000	5.000			21.000					
pisek	31.000									
prachatice		41.000								
strakonice					21.000	37.000				
tabor				31.000						15.000
blansko			14.000	1.000			32.000			
breclav							28.000			

Obr. 33 Výsledky modelu VS_WS1 (9 GAMS)

3.8.2 Model VS_WS2

Nyní zde bude zveřejněn obrázek pouze s částí kódu, která se oproti základnímu modelu pro tuto kapitolu, kterým je VS_WS1, změnila.

```
Parameter p(s)  /  s2   0.25 / ;
Parameters a(i)  volne kapacity v dodavajicich okresech
        /   ceskebudejovice         54
            ceskykrumlov            45
            jindrichuvhradec        36
            pisek                   31
            prachatice              41
            strakonice              58
            tabor                   46
            blansko                 47
            breclav                 29
            hodonin                 50
                                         /
      b(j,s)  poptavka v okresech s danym scenarem
        /    benesov.s2        50
             beroun.s2         49
             kladno.s2         32
             kolin.s2          28
             kutnahora.s2      39
             melnik.s2         35
             mladaboleslav.s2  63
             nymburk.s2        14
             pribram.s2        45
             rakovnik.s2       26 /  ;
```

Obr. 34 Změněná část zdrojového kódu v modelu VS_WS2 (9 GAMS)

Můžete vidět, že změna proběhla v příkazu **`Parameter` p(s)** na pravděpodobnost scénáře 2 a v **`Parameters` b(j,s)** na hodnoty poptávky scénáře 2. **`Parameters` a(i)** hodnota zůstává stejná, protože počty volných (nevyužitých) přístrojů se v našem případě nemění.

```
----     65 VARIABLE z.L                   =    16533.000

----     65 VARIABLE x.L
```

	benesov	beroun	kladno	kolin	kutnahora	melnik	mladabole~	nymburk	pribram	rakovnik
ceskebudejovice	9.000								45.000	
ceskykrumlov			31.000					14.000		
jindrichuvhradec	10.000	9.000			17.000					
pisek	31.000									
prachatice		40.000	1.000							
strakonice				1.000	22.000	35.000				
tabor				20.000						26.000
blansko				7.000			40.000			
breclav							23.000			

Obr. 35 Výsledky modelu VS_WS2 (9 GAMS)

3.8.3 Modely VS_WS3, VS_WS4, VS_WS5

V této kapitole budou uvedeny změny v modelech zmíněných v názvu kapitoly. Protože se změny pořád opakují, rozhodl jsem se zde uvést jen důležité (měnící se) části kódu, zatímco zbytek zdrojového kódu zůstává stejný jako v kapitole 3.8.1.

```
  Parameter p(s)/  s3   0.1 / ;
Parameters a(i)  volne kapacity v dodavajicich okresech
         /    ceskebudejovice        54
              ceskykrumlov           45
              jindrichuvhradec       36
              pisek                  31
              prachatice             41
              strakonice             58
              tabor                  46
              blansko                47
              breclav                29
              hodonin                50
                                          /
       b(j,s)  poptavka v okresech s danym scenarem
         /     benesov.s3        49
               beroun.s3         48
               kladno.s3         38
               kolin.s3          30
               kutnahora.s3      40
               melnik.s3         38
               mladaboleslav.s3  60
               nymburk.s3        16
               pribram.s3        40
               rakovnik.s3       18 /  ;
```

Obr. 36 Změněná část zdrojového kódu v modelu VS_WS3 (9 GAMS)

```
Parameter p(s) /  s4   0.3 / ;
  Parameters a(i)  volny kapacity v dodavajicich okresech
         /    ceskebudejovice         54
              ceskykrumlov            45
              jindrichuvhradec        36
              pisek                   31
              prachatice              41
              strakonice              58
              tabor                   46
              blansko                 47
              breclav                 29
              hodonin                 50
                                            /
       b(j,s)  poptavka v okresech s danym scenarem
         /     benesov.s4       54
               beroun.s4        49
               kladno.s4        35
               kolin.s4         43
               kutnahora.s4     39
               melnik.s4        35
               mladaboleslav.s4 53
               nymburk.s4       14
               pribram.s4       37
               rakovnik.s4      13 /  ;
```

Obr. 37 Změněná část zdrojového kódu v modelu VS_WS4 (9 GAMS)

```
Parameter p(s)  /  s5   0.2 / ;
Parameters a(i)  volne kapacity v dodavajicich okresech
         /    ceskebudejovice         54
              ceskykrumlov            45
              jindrichuvhradec        36
              pisek                   31
              prachatice              41
              strakonice              58
              tabor                   46
              blansko                 47
              breclav                 29
              hodonin                 50
                                            /
       b(j,s)  poptavka v okresech s danym scenarem
         /     benesov.s5       64
               beroun.s5        42
               kladno.s5        30
               kolin.s5         43
               kutnahora.s5     39
               melnik.s5        35
               mladaboleslav.s5 60
               nymburk.s5       20
               pribram.s5       37
               rakovnik.s5      13 /  ;
```

Obr. 38 Změněná část zdrojového kódu v modelu VS_WS5 (9 GAMS)

3.8.4 Výsledky modelů VS_WS3, VS_WS4 a VS_WS5

Tato kapitola bude zahrnuje výsledky modelů zmíněných v nadpisu. Pro každý ze 3 modelů tu bude výřez z lst souboru, který obsahuje výsledek účelové funkce na řádku **`VARIABLE` z.L**, což představuje náklady na přepravu přístrojů v tomto scénáři a níže v tabulce pod **`VARIABLE` x.L** můžeme vidět počty kusů přepravovaných přístrojů od dodavatelů k odběratelům.

---- 65 VARIABLE z.L = 6512.800

---- 65 VARIABLE x.L

	benesov	beroun	kladno	kolin	kutnahora	melnik	mladabole~	nymburk	pribram	rakovnik
ceskebudejovice	14.000								40.000	
ceskykrumlov			29.000					16.000		
jindrichuvhradec	4.000	12.000			20.000					
pisek	31.000									
prachatice		36.000	5.000							
strakonice					20.000	38.000				
tabor				28.000						18.000
blansko			4.000	2.000			41.000			
breclav							19.000			

Obr. 39 Výsledky modelu VS_WS3 (9 GAMS)

---- 65 VARIABLE z.L = 18799.800

---- 65 VARIABLE x.L

	benesov	beroun	kladno	kolin	kutnahora	melnik	mladabole~	nymburk	pribram	rakovnik
ceskebudejovice	17.000								37.000	
ceskykrumlov			31.000					14.000		
jindrichuvhradec	6.000	12.000			18.000					
pisek	31.000									
prachatice		37.000	4.000							
strakonice				2.000	21.000	35.000				
tabor				33.000						13.000
blansko				8.000			39.000			
breclav							14.000			

Obr. 40 Výsledky modelu VS_WS4 (9 GAMS)

---- 65 VARIABLE z.L = 13125.200

---- 65 VARIABLE x.L

	benesov	beroun	kladno	kolin	kutnahora	melnik	mladabole~	nymburk	pribram	rakovnik
ceskebudejovice	17.000								37.000	
ceskykrumlov			25.000					20.000		
jindrichuvhradec	16.000	4.000			16.000					
pisek	31.000									
prachatice		38.000	3.000							
strakonice					23.000	35.000				
tabor				33.000						13.000
blansko			2.000	10.000			35.000			
breclav							25.000			

Obr. 41 Výsledky modelu VS_WS5 (9 GAMS)

3.8.5 Očekávané budoucí náklady ze scénářů modelů VS_WS

Podobně jako u modelů WS pro malá data i z modelů velkých dat nakonec chci zjistit očekávané budoucí náklady přepravy. U velkých dat nesmím zapomenout, že tu je modelů 5, takže celkový výpočet bude o něco málo složitější tím, že bude delší. Pokud bych počítal optimální hodnotu účelové funkce pro scénáře WS, dosadím do zdrojového kódu modelu jen hodnotu poptávky bez pravděpodobností a vzorec pro optimální hodnotu WS by byl:

$$z_{min}^{WS1}\ , z_{min}^{WS2}, z_{min}^{WS3}, z_{min}^{WS4}, z_{min}^{WS5}$$

V dalším kroku vynásobíme optimální hodnotu účelové funkce pravděpodobností scénáře. V mých modelech v GAMSu je tento krok zahrnut do výpočtu rovnou, pro tento vztah platí vzorec:

$$z_{min}^{WS1} * p_1\ , z_{min}^{WS2} * p_2, z_{min}^{WS3} * p_3, z_{min}^{WS4} * p_4, z_{min}^{WS5} * p_5$$

Nyní už se dostávám k jádru věci, kvůli které jsem výše zmiňoval dané vzorce a kterou chci v této kapitole spočítat a to jsou očekávané budoucí (průměrné) náklady přepravy.

$$z^{WS} = E\left[z_{min}^{WS(xi)}\right] = z_{min}^{WS1} * p_1 + z_{min}^{WS2} * p_2 + z_{min}^{WS3} * p_3 + z_{min}^{WS4} * p_4 + z_{min}^{WS5} * p_5$$

Do rovnice dosadíme výsledky ze scénářů 1 a 2 a výpočet by měl vypadat následovně.

$$z^{WS} = E\left[z_{min}^{WS(xi)}\right] = 10096{,}65 + 16533 + 6512{,}8 + 18799{,}8 + 13125{,}2$$

$$z^{WS} = 65\ 067{,}45\ Kč$$

Očekávané budoucí náklady vycházející ze scénářů modelu WS mi ve výsledku vyšly s v celkové hodnotě 65 067,45 Kč.

3.9 Modely here-and-now pro velká data

Do těchto modelů nám vstupují scénáře z předchozího kroku, z modelu wait-and-see a s těmito se dále pracuje v přístupu here-and-now. Na obrázku níže bude vložen zdrojový kód pro modely VS_HN, který bude u všech modelů stejný, jedná se o počáteční nadefinování modelu a vložení poptávkových scénářů s pravděpodobnostmi.

```
Sets i   dodavatelske okresy /ceskebudejovice, ceskykrumlov, jindrichuvhradec,
            pisek, prachatice, strakonice, tabor, blansko, breclav, hodonin /
     j   poptavajici okresy /benesov, beroun, kladno, kolin, kutnahora, melnik,
                             mladaboleslav, nymburk, pribram, rakovnik /
       s   scenare           / s1, s2, s3, s4, s5 /     ;
Parameter p(s)/s1 0.15, s2 0.25, s3 0.1, s4 0.3, s5 0.2/;
Parameters  a(i)  volne kapacity v dodavajicich okresech
         /    ceskebudejovice         54
              ceskykrumlov            45
              jindrichuvhradec        36
              pisek                   31
              prachatice              41
              strakonice              58
              tabor                   46
              blansko                 47
              breclav                 29
              hodonin                 50
                                           /
       b(j,s)  poptavka v okresech s prictenim scenaru
         /     benesov.s1        53
               beroun.s1         52
               kladno.s1         35
               kolin.s1          32
               kutnahora.s1      42
               melnik.s1         37
               mladaboleslav.s1  60
               nymburk.s1        18
               pribram.s1        42
               rakovnik.s1       15

               benesov.s2        50
               beroun.s2         49
               kladno.s2         32
               kolin.s2          28
               kutnahora.s2      39
               melnik.s2         35
               mladaboleslav.s2  63
               nymburk.s2        14
               pribram.s2        45
               rakovnik.s2       26

               benesov.s3        49
               beroun.s3         48
               kladno.s3         38
               kolin.s3          30
               kutnahora.s3      40
               melnik.s3         38
               mladaboleslav.s3  60
               nymburk.s3        16
               pribram.s3        40
               rakovnik.s3       18

               benesov.s4        54
               beroun.s4         49
               kladno.s4         35
               kolin.s4          43
               kutnahora.s4      39
               melnik.s4         35
               mladaboleslav.s4  53
               nymburk.s4        14
               pribram.s4        37
               rakovnik.s4       13

               benesov.s5        64
               beroun.s5         42
               kladno.s5         30
               kolin.s5          43
               kutnahora.s5      39
               melnik.s5         35
               mladaboleslav.s5  60
               nymburk.s5        20
               pribram.s5        37
               rakovnik.s5       13             /  ;
```

Obr. 42 Vstupní společná data pro modely VS_HN (9 GAMS)

3.9.1 Model VS_HN_IS

Tento model ověřuje výchozí poptávku (vybrána z 5 scénářů) zda je schopna splnit poptávkové hodnoty ostatních scénářů, tzv. když si hned musíme vybrat jednu poptávku, zda bude tato schopna splnit i ostatní poptávky.

```
Table d(i,j)  vzdalenost v km
                      benesov   beroun   kladno   kolin   kutnahora   melnik
      ceskebudejovice   100      176      159      162      159        200
      ceskykrumlov      127      174      168      160      145        212
      jindrichuvhradec  99       158      172      145      125        206
      pisek             90       186      148      156      189        226
      prachatice        146      190      186      176      159        243
      strakonice        130      192      182      145      145        216
      tabor             112      182      179      125      168        218
      blansko           200      260      252      212      231        356
      breclav           243      312      300      268      276        396
      hodonin           250      318      312      280      290        412
+
                              mladaboleslav   nymburk   pribram   rakovnik
      ceskebudejovice              226          212       100       212
      ceskykrumlov                 246          200       121       268
      jindrichuvhradec             238          218       112       248
      pisek                        249          238       118       218
      prachatice                   242          242       134       236
      strakonice                   236          226       126       246
      tabor                        212          216       145       186
      blansko                      268          300       265       312
      breclav                      312          328       315       336
      hodonin                      418          365       326       363 ;
Scalar f  cena za kus na km prepravy  /1/ ;
parameter bb(j);
bb(j)= b(j,"s4");
Parameter c(i,j)  cena prepravy v korunach na jeden kilometr;c(i,j) = f * d(i,j);
variable z; positive variable x(i,j);
Equations cost        ucelova funkce
       supply(i)   omezeni poctem volnych kusu dodavajicich okresu
       demand(j,s) uspokojeni poptavky okresu j   ;
cost ..          z  =e=  sum((i,j), c(i,j)*x(i,j)) ;
supply(i) ..     sum(j, x(i,j))   =l=  a(i) ;
demand(j,s) ..    sum(i, x(i,j)) =e=  bb(j) ;
Model vshnis /all/ ;
Solve vshnis using lp minimizing z ;
display z.L, x.L, vshnis.modelstat;

parameter zIS,xIS(i,j);

zIS = z.L; xIS(i,j) = x.L(i,j); x.LO(i,j) = xIS(i,j); x.UP(i,j) = xIS(i,j);
bb(j) = b(j,"s1");
solve vshnis minimizing z using LP;
display vshnis.modelstat, z.L,x.L;

bb(j) = b(j,"s2");
solve vshnis minimizing z using LP;
display vshnis.modelstat, z.L,x.L;

bb(j) = b(j,"s3");
solve vshnis minimizing z using LP;
display vshnis.modelstat, z.L,x.L;

bb(j) = b(j,"s5");
solve vshnis minimizing z using LP;
display vshnis.modelstat, z.L,x.L;
```

Obr. 43 Zdrojový kód modelu VS_HN_IS (9 GAMS)

Věc, na kterou chci upozornit u tohoto modelu je **`Parameter` bb(j)**, který jsem v tomto modelu musel nadefinovat na poptávku scénáře 4, jelikož tu jsem zvolil jako výchozí. Níže mám tuto poptávku zahrnutou do omezující podmínky **demand(j,s)**. Model nám prvně vyřeší optimalizaci pro scénář 4 a následně máme nadefinované parametry **`Parameter` zIS** a **xIS(i,j)**, s jejichž pomocí ověřuji platnost výchozí poptávky pro ostatní poptávkové scénáře.

```
----    111 VARIABLE z.L                  =    62666.000

----    111 VARIABLE x.L

                     benesov      beroun      kladno       kolin   kutnahora      melnik  mladabole~     nymburk     pribram   rakovnik

ceskebudejovice       17.000                                                                                          37.000
ceskykrumlov                                  31.000                                                      14.000
jindrichuvhradec       6.000      12.000                                18.000
pisek                 31.000
prachatice                        37.000       4.000
strakonice                                                 2.000      21.000      35.000
tabor                                                     33.000                                                                  13.000
blansko                                                    8.000                               39.000
breclav                                                                                        14.000
```

Obr. 44 Výsledek modelu VS_HN_IS pro scénář 4 (9 GAMS)

V prvním výsledku vidíme, že pro scénář 4 vyšly náklady přepravy na 62 666Kč a pod náklady v tabulce jsou přepravované kusy mezi dodavateli a odběrateli.

```
Iteration log . . .
Iteration:      1    Infeasibility =

--- LP status (3): infeasible.
--- Cplex Time: 0.00sec (det. 0.04 ticks)


Model has been proven infeasible
```

Obr. 45 Výsledek modelu po ověřování platnosti scénáře 4 pro ostatní scénáře (9 GAMS)

Vidíme zde zprávu, že model nemá přípustné řešení a pro nás to znamená, že při ověřování platnosti scénáře 4 bylo programem zjištěno, že tato poptávka pro ostatní scénáře nebude dostatečná.

3.9.2 Model VS_HN_EV

V tomto modelu bereme všechny poptávky pro všechny scénáře a zprůměrujeme je. Dostaneme průměrnou hodnotu poptávky v každém okresu. Dále spočítáme optimální řešení pro tyto zprůměrované scénáře.

```
Scalar f  cena za kus na km prepravy  /1/ ;
parameter bb(j);
bb(j) = sum(s, p(s) * b(j,s));
Parameter c(i,j)  cena prepravy v korunach na jeden kilometr;c(i,j) = f * d(i,j);
variable z; positive variable x(i,j); positive variable yp(j,s), ym(j,s);
Equations cost         ucelova funkce
       supply(i)   omezeni poctem volnych kusu dodavajicich okresu
       demand(j,s) uspokojeni poptavky okresu j   ;
cost ..          z  =e=  sum((i,j), c(i,j)*x(i,j)) ;
supply(i) ..   sum(j, x(i,j))  =l=  a(i) ;
demand(j,s) ..   sum(i, x(i,j)) =e=  bb(j) ;
Model vshnev /all/ ;
Solve vshnev using lp minimizing z ;
display z.L, x.L, vshnev.modelstat;

parameter zEV,xEV(i,j);

zEV = z.L; xEV(i,j) = x.L(i,j); x.LO(i,j) = xEV(i,j); x.UP(i,j) = xEV(i,j);
bb(j) = b(j,"s1");
solve vshnev minimizing z using LP;
display vshnev.modelstat, z.L,x.L;

bb(j) = b(j,"s2");
solve vshnev minimizing z using LP;
display vshnev.modelstat, z.L,x.L;

bb(j) = b(j,"s3");
solve vshnev minimizing z using LP;
display vshnev.modelstat, z.L,x.L;

bb(j) = b(j,"s4");
solve vshnev minimizing z using LP;
display vshnev.modelstat, z.L,x.L;

bb(j) = b(j,"s5");
solve vshnev minimizing z using LP;
display vshnev.modelstat, z.L,x.L;
```

Obr. 46 Zdrojový kód pro model VS_HN_EV (9 GAMS)

Na obrázku zdrojového kódu si prvně všimněme, že je zde změna pro **parameter bb(j)**, pro který zde máme uvedenou rovnice na zprůměrování poptávky, konkrétně tím způsobem, že pravděpodobnostmi scénářů násobíme odpovídající hodnoty poptávek. Dále v omezující podmínce **demand(j,s)** máme uveden **Parameter bb(j)**, který v sobě již obsahuje hodnoty průměrných poptávek. Pod **parameter zEV a xEV(i,j)** nám přibyl kód pro jednu verifikaci navíc oproti modelu VS_HN_IS a to z důvodu, že v tomto modelu ověřujeme platnost průměrné poptávky pro všechny scénáře.

```
----     111 VARIABLE z.L                   =    65056.300

----     111 VARIABLE x.L

                    benesov      beroun      kladno       kolin   kutnahora      melnik  mladabole~     nymburk     pribram   rakovnik

ceskebudejovice      13.950                                                                                        40.050
ceskykrumlov                                 29.000                                                    16.000
jindrichuvhradec      9.400       9.450                              17.150
pisek                31.000
prachatice                       38.500       2.500
strakonice                                                           22.400      35.600
tabor                                                    28.950                                                                17.050
blansko                                       2.050       7.350                                37.600
breclav                                                                                        21.050
```

Obr. 47 Optimální řešení modelu VS_HN_EV (9 GAMS)

Z obrázku vyčteme, že optimální řešení, jedná se o řádek **VARIABLE z.L**, pro průměrnou poptávku má hodnotu nákladů, 65 056,3 Kč a pod řádkem **VARIABLE x.L** je umístěna tabulka počtu přepravovaných kusů.

```
--- LP status (3): infeasible.
--- Cplex Time: 0.00sec (det. 0.04 ticks)


Model has been proven infeasible
```

Obr. 48 Výsledek modelu po ověření scénáře průměrné poptávky pro ostatní scénáře (9 GAMS)

Zde mi program opět vypsal, že model nemá přípustné řešení. Je to způsobeno nedostatečným počtem kusů dodávky pro pokrytí ostatních scénářů.

3.9.3 Model VS_HN_MM

Tento model bere tu nejvyšší poptávku ze všech scénářů pro daný okres a s tou počítá dále v optimalizaci. V tomto modelu počítáme skoro jako v jednoduchém deterministickém, pointa je ve změně omezení pro poptávku. Na zdrojovém kódu níže si změnu ukážeme a vysvětlím na ni, proč je tak zásadní pro tento model.

```
Table d(i,j)  vzdalenost v km
                           benesov   beroun   kladno   kolin   kutnahora   melnik
      ceskebudejovice        100       176      159      162      159        200
      ceskykrumlov           127       174      168      160      145        212
      jindrichuvhradec       99        158      172      145      125        206
      pisek                  90        186      148      156      189        226
      prachatice             146       190      186      176      159        243
      strakonice             130       192      182      145      145        216
      tabor                  112       182      179      125      168        218
      blansko                200       260      252      212      231        356
      breclav                243       312      300      268      276        396
      hodonin                250       318      312      280      290        412
+
                                   mladaboleslav   nymburk   pribram   rakovnik
      ceskebudejovice                   226           212       100       212
      ceskykrumlov                      246           200       121       268
      jindrichuvhradec                  238           218       112       248
      pisek                             249           238       118       218
      prachatice                        242           242       134       236
      strakonice                        236           226       126       246
      tabor                             212           216       145       186
      blansko                           268           300       265       312
      breclav                           312           328       315       336
      hodonin                           418           365       326       363 ;
Scalar f  cena za kus na km prepravy  /1/ ;
Parameter c(i,j)  cena prepravy v korunach na jeden kilometr;c(i,j) = f * d(i,j);
variable z2; positive variable x(i,j);
Equations
       cost         ucelova funkce
       supply(i)    omezeni poctem volnych kusu dodavajicich okresu
       demand(j,s)  uspokojeni poptavky okresu j    ;
cost ..        z2  =e=  sum((i,j), c(i,j)*x(i,j))  ;
supply(i) ..      sum(j, x(i,j))  =l=  a(i) ;
demand(j,s) ..     sum(i, x(i,j)) =g=  b(j,s) ;
Model vshnmm /all/ ;
Solve vshnmm using lp minimizing z2 ;
display z2.L, x.L;
```

Obr. 49 Zdrojový kód pro model VS_HN_MM (9 GAMS)

V omezující podmínce **demand(j,s)** nastává změna rovnosti, resp. nerovnosti na **„=g="** větší nebo rovno, toto nastavení programu umožňuje vzít nejvyšší poptávku napříč všemi poptávkovými scénáři pro dané město. V případě ponechání rovnosti **„=e="** by model označil optimalizaci za nemající řešení (infeasible).

```
----    109 VARIABLE z2.L                  =    80360.000

----    109 VARIABLE x.L

                    benesov      beroun      kladno       kolin   kutnahora      melnik  mladabole~     nymburk     pribram   rakovnik

ceskebudejovice                                                                   9.000                              45.000
ceskykrumlov                                 25.000                                                   20.000
jindrichuvhradec                 13.000                               23.000
pisek                20.000                  11.000
prachatice                       39.000       2.000
strakonice                                               10.000       19.000     29.000
tabor                                                    20.000                                                                  26.000
blansko                                                  13.000                              34.000
breclav                                                                                      29.000
hodonin              44.000
```

Obr. 50 Výsledky modelu VS_HN_MM (9 GAMS)

Z výsledků je patrné, že optimální řešení modelu je velmi drahé kvůli faktu, že bere pro každé město tu nejvyšší poptávku. Zde jen poradím, abyste se nezalekli vysokých nákladů po optimalizaci, protože v tomto modelu je zkrátka musíte očekávat. Celkové náklady přepravy vychází na 80 360 Kč a pod nimi vidíte tabulku s počty přepravovaných přístrojů.

3.9.4 Model VS_HN_TS

Jde o dvojstupňový model pro velká data, kde můžeme splnění poptávky ve scénářích zajistit tím, že penalizujeme odchylky způsobené tím, že volíme HN rozhodnutí pro dopravu (viz. proměnné x), které musí být pro všechny scénáře stejné. Tyto odchylky budou možné za dodatečný poplatek, a to jak v případě přidání kusů, tak i při odřeknutí zboží. Každý scénář s sebou nese poplatky za dodatečný kus přidaný do dopravy a za odřeknutý kus.

```
Table d(i,j)  vzdalenost v km
                        benesov   beroun   kladno   kolin   kutnahora   melnik
      ceskebudejovice     100      176      159      162      159        200
      ceskykrumlov        127      174      168      160      145        212
      jindrichuvhradec    99       158      172      145      125        206
      pisek               90       186      148      156      189        226
      prachatice          146      190      186      176      159        243
      strakonice          130      192      182      145      145        216
      tabor               112      182      179      125      168        218
      blansko             200      260      252      212      231        356
      breclav             243      312      300      268      276        396
      hodonin             250      318      312      280      290        412
+
                                mladaboleslav   nymburk   pribram   rakovnik
      ceskebudejovice               226           212       100       212
      ceskykrumlov                  246           200       121       268
      jindrichuvhradec              238           218       112       248
      pisek                         249           238       118       218
      prachatice                    242           242       134       236
      strakonice                    236           226       126       246
      tabor                         212           216       145       186
      blansko                       268           300       265       312
      breclav                       312           328       315       336
      hodonin                       418           365       326       363 ;
Parameter qp(j)
    /         benesov       300
              beroun        350
              kladno        350
              kolin         1000
              kutnahora     300
              melnik        350
              mladaboleslav 350
              nymburk       1000
              pribram       350
              rakovnik      1000    / ;
Parameter qm(j)
    /         benesov       2
              beroun        10
              kladno        10
              kolin         2
              kutnahora     10
              melnik        10
              mladaboleslav 10
              nymburk       1
              pribram       1
              rakovnik      10     / ;
Scalar f  cena za kus na km prepravy  /1/ ;
Parameter c(i,j)  cena prepravy v korunach na jeden kilometr;c(i,j) = f * d(i,j);
variable z2; positive variable x(i,j); positive variable yp(j,s), ym(j,s);
Equations cost         ucelova funkce
       supply(i)    omezeni poctem volnych kusu dodavajicich okresu
       demand(j,s)  uspokojeni poptavky okresu j    ;
cost ..        z2  =e=  sum((i,j), c(i,j)*x(i,j)) + sum(s, p(s) *
(sum(j, qp(j)*yp(j,s)) + sum(j, qm(j)*ym(j,s)))) ;
supply(i) ..    sum(j, x(i,j))   =l=  a(i) ;
demand(j,s) ..    sum(i, x(i,j)) + yp(j,s)  - ym(j,s)  =e=  b(j,s) ;
Model vshnts /all/ ;
Solve vshnts using lp minimizing z2 ;
display z2.L, x.L, yp.L, ym.L;
```

Obr. 51 Zdrojový kód pro model VS_HN_TS (9 GAMS)

Jak lze vidět podle červeně orámovaných částí kódu, máme tu hned několik nových věcí oproti předchozím kódům. První je **`Parameter`** **qp(j)** a **qm(j)**, které nám udávají penalizační cenu za jeden přidaný kus do objednávky a cenu penalizace za odebraný kus z objednávky. Dále jsem nadefinoval **`positive variable`** **yp(j,s)** a **ym(j,s)**, která bude udávat počty přidaných kusů do objednávky a počty odebraných kusů z dodávky. Jak vidíte dále, s novými parametry a proměnnými pracujeme v omezujících podmínkách v účelové funkci **cost** a v omezující podmínce **demand(j,s)**. Když chceme vypsat přidané a odebrané kusy z objednávky ve výsledcích modelu, musíme nakonec do příkazu **`display`** zahrnout i nové proměnné **yp.L** a **ym.L**.

```
----    132 VARIABLE z2.L                =    71676.500

----    132 VARIABLE x.L
```

	benesov	beroun	kladno	kolin	kutnahora	melnik	mladabole~	nymburk	pribram	rakovnik
ceskebudejovice	17.000								37.000	
ceskykrumlov			27.000					18.000		
jindrichuvhradec	2.000	4.000			30.000					
pisek	31.000									
prachatice		38.000	3.000							
strakonice				14.000	9.000	35.000				
tabor				28.000						18.000
blansko				1.000			46.000			
breclav							7.000			

```
----    132 VARIABLE yp.L
```

	s1	s2	s3	s4	s5
benesov	3.000			4.000	14.000
beroun	10.000	7.000	6.000	7.000	
kladno	5.000	2.000	8.000	5.000	
kutnahora	3.000		1.000		
melnik	2.000		3.000		
mladaboleslav	7.000	10.000	7.000		7.000
nymburk					2.000
pribram	5.000	8.000	3.000		
rakovnik		8.000			

```
----    132 VARIABLE ym.L
```

	s1	s2	s3	s4	s5
benesov			1.000		
kolin	11.000	15.000	13.000		
nymburk		4.000	2.000	4.000	
rakovnik	3.000			5.000	5.000

Obr. 52 Výsledky modelu VS_HN_TS (9 GAMS)

Ve výsledcích po řešení modelu vidíme jako první celkové náklady pro daný model s hodnotou 71 676,5 Kč. Následuje tabulka přepravovaných kusů mezi dodavateli a odběrateli. Třetí položka je tabulka přidaných kusů do objednávky za penalizační ceny a poslední tabulka reprezentuje odřeknuté kusy z objednávky za penalizační ceny.

3.10 Ukázka modelu s využitím cyklu a výpis výsledků do txt souboru

```
Sets
       i   dodavajici kraje   / jihomoravsky, pardubicky, olomoucky, vysocina /
       j   poptavajici kraje          / praha, stredocesky, jihocesky, plzensky /
       s   scenar                     / s1, s2 /             ;
Parameters a(i) kapacity volnych ventilatoru v dodavatelskych krajich
         /   jihomoravsky               75
             pardubicky                 12
             olomoucky                  27
             vysocina                   20
                                            /
       b(j,s)  celkova poptavka poptavajicich kraju
         /    praha.s1         40
              stredocesky.s1   25
              jihocesky.s1     10
              plzensky.s1      15

              praha.s2         60
              stredocesky.s2   35
              jihocesky.s2     5
              plzensky.s2      10             / ;
Parameter p(s) / s1    0.4, s2   0.6 /, zWS ;
Table d(i,j)  vzdalenost v km
                          praha         stredocesky       jihocesky       plzensky
      jihomoravsky         205              227              217            291
      pardubicky           126              119              211            220
      olomoucky            280              243              296            370
      vysocina             163              190              136            254;
Scalar f  cena za km prepravy na jeden kus  /1/ ;
Scalar EzWS;
Parameter c(i,j)  cena prepravy na 1 km, bb(j) ; c(i,j) = f * d(i,j) ;
variable z; positive variable x(i,j);
Equations
       cost          ucelova funkce
       supply(i)     omezeni poctem volnych kusu dodavajicich kraju
       demand(j,s)     uspokojeni poptavky kraju j ;
cost ..          z  =e=  sum((i,j), c(i,j)*x(i,j)) * sum (s,p(s)) ;
supply(i) ..    sum(j, x(i,j))   =l=  a(i)  ;
demand(j,s) ..    sum(i, x(i,j))   =e=  b(j,s);
Model doprava /cost, supply, demand/ ;
file out / "vysledekWS.txt" /;
put out;
Solve doprava using lp minimizing z ;
display z.L, x.L, supply.L, demand.L;
* WS model
equations costWS, supplyWS(i), demandWS(j);
costWS.. z =e=  sum((i,j), c(i,j)*x(i,j));
supplyWS(i)..     sum(j, x(i,j))   =l=  a(i);
demandWS(j).. sum(i, x(i,j))   =e=  bb(j);
model dopravaWS / costWS, supplyWS, demandWS /;
put "WS model:" /;
put " zWS";
put /;
loop(s, bb(j) = b(j,s);
solve dopravaWS minimizing z using LP;
display z.L, x.L, supply.L, demand.L;
zWS(s) = z.L;
put dopravaWS.modelstat:3:0, z.L:10:2;
put /;
); put /;
*Očekávané budoucí náklady WS
EzWS = sum(s, p(s) * zWS(s));
put "E z^WS = ", EzWS:10:3 /;
```

Obr. 53 Model WS s využitím příkazu put a loop (9 GAMS)

Na zdrojovém kódu, který zahrnuje oba poptávkové scénáře modelu wait-and-see nyní můžete vidět několik modifikací. První se objevuje v Parameter, přidáním parametru zWS. Poté pod příkazem Scalar jsme si nadefinovali EzWS pro hodnotu budoucích nákladů modelu WS. Příkazem file out jsem nadefinoval modelu ať při řešení modelu vytvoří soubor s názvem „vysledekWS.txt“. V části WS model jsem stanovil 3 omezující podmínky pro výpočet nákladů wait-and-see modelu bez pravděpodobnosti nadefinoval jsem model dopravaWS, aby počítal podle těchto 3 omezení a pak jsem zadal vypsání textu „WS model“ do výsledkového souboru za textem odsazení o řádek. Poté se vypíše „zWS“ a následně řeším model pro výpočet obou scénářů modelu a vypsání řešení pomocí příkazu put dopravaWS.modelstat:3:0, z.L:10:2. Nyní si jednotlivé položky rozebereme, dopravaWS.modelstat:3:0 znamená, že se nám do výsledkového souboru vypíšou v tomto konkrétním případě číslice reprezentující jednotlivé výsledky scénářů bez desetinných míst, odsazené o 3 znaky doprava a z.L:10:2 již znamená vypsání konkrétních výsledků scénářů odsazených o 10 znaků doprava a se dvěma desetinnými místy. Pod těmito příkazy je nadefinován výpočet EzWS, což je očekávaná hodnota budoucích nákladů modelu WS a pod výpočtem jeho vypsání do výsledkového souboru. Níže pro ilustraci přikládám výřez z výsledkového souboru pro ilustraci.

```
WS model:
 zWS
  1  17884.00
  1  21957.00

E z^WS =  20327.800
```

Obr. 54 Výsledky modelu ve výsledkovém souboru txt (9 GAMS)

4 ZÁVĚR

Začátkem práce se z teorie dozvídáme základní informace o programu GAMS, například k čemu program slouží, jaké jsou jeho základní vlastnosti, o přenositelnosti modelů, která je v případě GAMSu výborná, jelikož jedna osoba může vyvinout model a pak daný model může použít libovolný počet osob od sebe jakkoli daleko fyzicky vzdálených. Dále je uveden jednoduchý příklad dopravního problému, na kterém je vidět zápis optimalizačního modelu, a také výstup modelu v podobě vyřešené optimalizace i s jejím popisem. V teoretické části je ještě obsažena kapitola o přístupech využívaných při optimalizačním modelování. Jedná se o přístup deterministický, wait-and-see a here-and-now.

V praktické části práce se přesuneme prvně k tomu, jak program stáhnout a nainstalovat, jak si založit licenci a tou následně aktivovat program. Dále zde bude představena knihovna modelů a popsáno, jak ji lze v programu nalézt a k čemu Vám může být dobrá. V podkapitole knihovny modelů jsou uvedeny 4 základní předdefinované modely pro řešení dopravního problému, výrobu slitin, rozšíření sortimentu firmy, skladu a jeho velikosti pro skladování zboží. V další části práce se již zabývám zpracováním vlastního tématu týkajícího se distribuce plicních ventilátorů od méně zasažených oblastí České republiky (dodavatelů) Covidem-19 do oblastí rizikovějších s vyšším počtem nakažených (odběratelé). Nyní už přecházíme k samotnému modelování, kde začínám deterministickým modelem, na kterém vysvětluji konstrukci modelu a co jeho jednotlivé úseky znamenají. Dále jsou uvedeny modely wait-and-see a popsáno, co se v nich změnilo (přidalo) oproti modelu deterministickému. Poté následují modely here-and-now a jejich různé modifikace, které se liší od deterministického modelu i modelů wait-and-see. Pokračováním v dokumentu dále narazíme na modely pro „velká“ data, které se hlavně liší velikostí a množstvím vstupních dat do modelů. Při zpracovávání velkých dat musíme být obezřetní, jelikož větším rozsahem vzniká větší místo pro chyby. Účelem práce je usnadnit vstup a poskytnout náhled zájemcům do problematiky optimalizačního modelování rizik v programu GAMS a pomoci jim s orientací a zvládnutím předmětu vyučovaného na Ústavu soudního inženýrství VUT v Brně nazvaného Matematické modely rozhodování.

SEZNAM POUŽITÝCH ZDROJŮ

(1) GAMS - Download. *GAMS* [online]. Německo: GAMS, 2020 [cit. 2021-02-11]. Dostupné z: https://www.gams.com/download/

(2) The GAMS Model Library. *GAMS* [online]. Německo: GAMS, 2020 [cit. 2021-02-11]. Dostupné z: https://www.gams.com/latest/gamslib_ml/libhtml/index.html#gamslib

(3) GAMS tutorial. *United Nations* [online]. United Nations: United Nations, 2008 [cit. 2021-5-24]. Dostupné z: https://www.un.org/en/development/desa/policy/mdg_workshops/eclac_training_mdgs/brookeetal2008_gamsusersguide.pdf?fbclid=IwAR1hB5Ytr3IgND3K4jyFGH4tVOSWAJQhpKdE4Y78nq_IepjF1kAwH-lcEwY

(4) Algoritmy. *Algoritmy* [online]. Česká republika: algoritmy, 2021 [cit. 2021-6-2]. Dostupné z: https://www.algoritmy.net/article/1416/Simplexova-metoda

(5) KLAPKA, Jindřich, Jiří DVOŘÁK a Pavel POPELA. *Metody operačního výzkumu*. 2. Brno: Vysoké učení technické, 2000. ISBN 80-214-0817-0.

(6) *Stochastic programming* [online]. 2. Švýcarsko: University of Zurich, 2003 [cit. 2021-6-10]. Dostupné z: http://web.ba.ntu.edu.tw/stochastic/10.1.1.111.631.pdf. Odborná publikace. University of Zurich.

(7) BIRGE, John R. a Francois LOUVEAUX. *Introduction to Stochastic Programming*. 2. New York: Springer, 2011. ISBN 978-1-4614-0236-7.

(8) BAZARAA, Mokhtar S., John J. JARVIS a Hanif D. SHERALI. *Linear Programming and Network Flows*. 4. New Jersey: Wiley, 2010. ISBN 978-0-470-46272-0.

(9) GAMS program

(10) KARPÍŠEK, Zdeněk. *Matematika IV*. 4., přeprac. vyd. Brno: Akademické nakladatelství CERM, 2014. ISBN 978-80-214-4858-2.

SEZNAM OBRÁZKŮ

SEZNAM TABULEK