

# **Škoda Auto Vysoká škola o.p.s.**

Studijní program: N0413A050001 Ekonomika a management

Studijní obor/specializace: Řízení mezinárodních dodavatelských řetězců

## **Využití optimalizačního systému AMPL při řízení výroby a dopravy Diplomová práce**

**Bc. Tereza JINDRÁKOVÁ**

Vedoucí práce: doc. Ing. Jan Fábry, Ph.D.

Děkuji doc. Ing. Janu Fábrymu, Ph.D. za odborné vedení závěrečné práce, poskytování rad a informačních podkladů. Ráda bych poděkovala studentu Ondřeji Šafkovi za poskytování rad a za pomoc při navrhování jednotlivých modelů v AMPL na základě jeho zkušeností. Dále bych ráda poděkovala mé mamince za podporu při studiu a mému příteli za to, že to se mnou zvládl a velmi mě podporoval.

## Obsah

Úvod .....	5
1 Optimalizace a lineární programování .....	6
2 Vybrané optimalizační problémy z oblasti výroby a dopravy.....	7
2.1 Řezná úloha .....	8
2.2 Dopravní problém.....	10
2.3 Okružní a rozvozní úlohy .....	14
3 Software pro matematické modelování.....	19
3.1 MPL for Windows .....	20
3.2 AMPL .....	22
3.3 Porovnání systémů MPL a AMPL .....	24
4 Aplikace systému AMPL .....	26
4.1 Řezná úloha .....	26
4.2 Kontejnerový dopravní problém .....	34
4.3 Rozvozní problém .....	39
Závěr .....	44
Seznam literatury .....	45
Seznam obrázků a tabulek .....	48
Seznam příloh .....	50

## Seznam použitých zkratk a symbolů

AIMMS	Advanced Integrated Multidimensional Modeling Software
AMPL	A Mathematical Programming Language
DP	Dopravní problém
IP	Celočíselné programování
MINOS	Modular In-core Nonlinear Optimization System
MPL	Mathematical Programming Language for Windows
MPSX	Mathematical Programming System Extended
NLP	Nelineární programování
LP	Lineární programování
ORÚ	Okružní a rozvozní úlohy
OÚ	Okružní úloha
ŘÚ	Řezná úloha
SDVRP	Split Delivery Vehicle Routing Problem
SNOPT	Sparse Nonlinear Optimization
TSP	Problém obchodního cestujícího
VRP	Problém s trasováním vozidla
QP	Kvadratické programování

## Úvod

Optimalizace systému se týká procesu maximalizace nebo minimalizace určitých cílů nebo opatření výkonu v rámci systému. Zahrnuje nalezení nejlepšího souboru rozhodnutí nebo akcí, které optimalizují celkový výkon, efektivitu nebo efektivitu komplexního systému.

Optimalizaci systému lze použít v různých oblastech, včetně průmyslu, obchodu, zemědělství, dopravy aj. Optimalizace je významnou součástí operačního výzkumu, jehož oblast obvykle zahrnuje použití matematického modelování a optimalizačních technik k nalezení optimálního řešení.

Techniky optimalizace systému lze použít ke zlepšení účinnosti, snížení nákladů, optimalizaci alokace zdrojů, zlepšení rozhodování a řešení různých složitých problémů v rámci systému. Jako příklady lze uvést mj. optimalizaci dodavatelského řetězce, plánování výroby, optimalizace dopravní sítě a umístění zařízení, řízení v oblasti logistiky a kvality apod.

Cílem této práce je popsat systém AMPL jako podpory modelování a optimalizace, včetně jeho praktického využití. K tomuto účelu budou naformulovány matematické modely vybraných úloh z oblasti výroby a řízení dopravy. Tyto modely jsou následně převedeny do prostředí systému AMPL, je zvýrazněna spolupráce tohoto systému s MS Excel.

Ve Škoda Auto Vysoké škole byla pro řešení úloh matematického programování doposud používána plná verze systému MPL for Windows. Vzhledem k tomu, že je systém dostupný již pouze v omezené licenci, program AMPL je vhodnou alternativou nejen pro výuku, ale především pro zpracování reálných úloh v rámci závěrečných prací.

# 1 Optimalizace a lineární programování

Lineární programování je matematická optimalizační technika používaná k maximalizaci nebo minimalizaci lineární funkce, která podléhá množině lineárních omezení. Je široce používán v různých oblastech, včetně operačního výzkumu, ekonomiky, inženýrství a managementu.

V lineárním programování je cílem najít hodnoty rozhodovacích proměnných, které optimalizují účelovou funkci při splnění omezujících podmínek. Účelová funkce je lineární výraz představující veličinu, která má být maximalizována nebo minimalizována. Rozhodovací proměnné představují neznámé v problému, které lze upravit tak, aby bylo dosaženo požadovaného výsledku (Fábry, 2019). Omezující podmínky jsou lineární nerovnosti nebo rovnice, které omezují možnou oblast řešení. Definují omezení nebo požadavky problému.

Optimalizace a lineární programování jsou matematické metody používané k nalezení nejlepšího možného řešení v rámci daných omezení. Lineární programování (LP) je konkrétní typ optimalizačního problému, ve kterém se snažíme minimalizovat nebo maximalizovat lineární účelovou funkci za přítomnosti lineárních omezení (Vanderbei, 1998).

Optimalizace může zahrnovat i další druhy problémů, které nejsou omezeny na lineární funkce nebo omezení. Může zahrnovat nelineární účelovou funkci a omezující podmínky ve formě nelineárních nerovnic nebo rovnic, diskrétní proměnné atd. Pro řešení optimalizačních problémů existuje mnoho metod, včetně lineárního programování, kvadratického programování, metody nejmenších čtverců, evolučních algoritmů, gradientního algoritmu a dalších. Vybraná metoda závisí na konkrétních vlastnostech a požadavcích problému.

Lineární programování lze použít k řešení široké škály optimalizačních problémů v různých oblastech. Lze jej použít např. k určení optimální alokace omezených zdrojů, jako je práce, kapitál a materiály, za účelem maximalizace produkce nebo minimalizace nákladů (Yevgen, 2019).

## **2 Vybrané optimalizační problémy z oblasti výroby a dopravy**

Optimalizační problémy v oblasti výroby a dopravy hrají klíčovou roli při maximalizaci efektivity, minimalizaci nákladů a optimalizaci procesů. V rámci této kapitoly se zaměříme na klíčové úlohy spojené s výrobním plánováním, což představuje kritický prvek v efektivní správě výrobních procesů (Fábry, 2019; Boucherie, Braaksma a Tijms 2022). V následujícím textu budou podrobně rozebrány některé z hlavních úloh v oblasti výrobního plánování, jejichž porozumění je nezbytné pro dosažení optimálního chodu výroby a dopravy.

### **Rozvrh výroby**

Rozvrh výroby patří mezi klíčové úkoly v oblasti výrobního plánování. Jeho cílem je efektivně naplánovat výrobní operace v daném časovém horizontu. Zabývá se otázkou, jak optimálně uspořádat jednotlivé výrobní zakázky tak, aby byla splněna časová a kapacitní omezení. Tato část se dále zabývá metodami a algoritmy pro tvorbu optimálních výrobních rozvrhů.

### **Plánování kapacit**

Plánování kapacit je další důležitou částí výrobního plánování. Zkoumá, jak efektivně plánovat využití výrobních kapacit a zajišťovat, aby byly dostatečné pro pokrytí poptávky. Důraz je kladen na identifikaci a řešení přetížení výrobních zdrojů, což může být klíčovým faktorem ovlivňujícím efektivitu výrobních procesů.

### **Skladování a řízení zásob**

Dalším důležitým aspektem výrobního plánování je skladování a řízení zásob. Tato část se zabývá otázkami optimalizace úrovně zásob surovin a hotových výrobků. Kromě toho zkoumá strategie minimalizace nákladů spojených se skladováním a řízením zásob.

### **Plánování materiálových potřeb**

Plánování materiálových potřeb je klíčovým prvkem v oblasti výroby. Tato část se zaměřuje na otázky plánování nákupu a dodávek surovin tak, aby byly k dispozici ve správném čase. Cílem je minimalizovat zásoby surovin a současně zajistit nepřetržitý výrobní proces.

## 2.1 Řezná úloha

Tato kapitola se věnuje jedné z klíčových optimalizačních úloh v oblasti výrobního plánování, a to řezné úloze. Řezná úloha (dále také ŘÚ) představuje komplexní problém, který vzniká ve výrobních prostředích a má významný vliv na efektivnost a náklady výrobního procesu. V této části práce budeme podrobněji zkoumat různé aspekty řezné úlohy a analyzovat strategie a metody, které lze použít pro její efektivní řešení. Řezná úloha spočívá v rozdělení omezeného materiálu (např. tyče, prkna, role materiálu apod.) na menší části tak, aby byly splněny specifické požadavky na rozměry těchto částí. Cílem je minimalizovat ztráty materiálu či optimalizovat využití zdrojů (Fábry, 2024).

### Matematický model řezné úlohy

#### Parametry

$m$  počet typů kratších dílů

$n$  počet řezných schémat

$b_i$  požadovaný počet dílů  $i$ -tého typu ( $i = 1, 2, \dots, m$ )

$a_{ij}$  počet dílů  $i$ -tého typu získaných podle  $j$ -tého řezného schématu

( $i = 1, 2, \dots, m; j = 1, 2, \dots, n$ )

#### Proměnné

$x_j$  počet kusů původního materiálu rozděleného podle  $j$ -tého řezného schématu

#### Účelová funkce

$$z = \sum_{j=1}^n x_j \rightarrow \min \quad (1)$$

#### Omezující podmínky

$$\sum_{j=1}^n a_{ij} x_j = b_i \quad i = 1, 2, \dots, m \quad (2)$$

$$x_j \in \mathbb{Z}_+ \quad j = 1, 2, \dots, n \quad (3)$$



Účelová funkce (1) představuje celkový počet rozřezaných kusů původního materiálu. Soustava rovnic (2) zajišťuje získání požadovaných počtů kusů kratších dílů. Podmínky (3) jsou podmínkami celočíselnosti pro všechny proměnné.

### **Klasifikace ŘÚ**

Jednodimenzionální (dále také 1D) ŘÚ se zaměřuje na rozdělení materiálu pouze ve směru jedné dimenze. Využívá se především při řezání materiálu na jednorozměrné části, například při výrobě trubek nebo tyčí. Příkladem pro 1D řezání může být například v oblasti stavebnictví, kde je potřeba řezat dlouhé trámy nebo roury na menší kusy s cílem splnit určité délky nebo požadavky na jednotlivé části konstrukce (Dyckhoff a Ute, 1992).

2D ŘÚ představuje rozdělení materiálu ve dvou rozměrech. Typickým příkladem je řezání plechů na menší části s minimálním odpadem. Další příklad 2D řezání může být v kontextu průmyslové výroby textilu nebo papíru, kde je třeba optimalizovat řezání materiálu na menší kusy s cílem minimalizovat odpad a zároveň splnit požadavky na určité tvary či rozměry (Dyckhoff a Ute, 1992).

3D ŘÚ v této úloze se zaměřujeme na řezání materiálu ve všech třech rozměrech, což je obvykle relevantní pro materiály s třetím rozměrem, jako jsou válce, válec, nebo další třírozměrné objekty. Konkrétním příkladem třírozměrného řezání může být řezání dřeva nebo kovu na různě tvarované části s cílem minimalizovat množství odpadu. Algoritmy pro řešení těchto problémů mohou využívat metody lineárního programování, heuristiky, nebo další optimalizační techniky (Taha, 2007).

Dalším možným příkladem pro 3D úlohu je například výroba součástek pro letecký průmysl, kdy výrobce potřebuje vyrábět komplexní díly ze speciálního materiálu, například hliníkové slitiny. Tyto díly mohou mít složité tvary a musí být vyřezány z velkých bloků materiálu (Dyckhoff a Ute, 1992)

Posledním rozdělením ŘÚ je nestabilní úloha, která zohledňuje nestabilitu materiálu, který může mít různou kvalitu nebo rozložení vlastností. Tato varianta řešení je často využívána v odvětvích s proměnlivými materiálovými vlastnostmi.

## **Optimalizační metody pro ŘÚ**

Algoritmy pro hromadné řezání zahrnují různé heuristiky a metaheuristiky pro hromadné řezání materiálu, což je často výzva v masové výrobě (Glover a Kochenberger, 2006) (Groner a Bischof, 2014). Další metodou je dynamické programování (Bersekas, 2012), které se zejména používá při 1D, 2D a 3D řezné úloze k nalezení optimálních strategií rozdělení materiálu. Mezi další optimalizační metody pro řeznou úlohy patří evoluční algoritmy, které jsou vhodné pro komplexní řezné úlohy, kde je třeba brát v úvahu mnoho proměnných a omezení.

### **Aplikace ŘÚ ve výrobním plánování**

První možností je aplikace redukce materiálu. Analyzuje se, jak efektivně řešit řeznou úlohu s cílem minimalizovat odpad materiálu. Další možností aplikace je časová efektivita, kde se studuje vliv řešení řezné úlohy na celkový čas výrobního procesu. Poslední aplikací, kterou bych zde chtěla zmínit, je integrace s ostatními výrobními úlohami. Představuje strategie pro úspěšnou integraci řezné úlohy do komplexního rámce výrobního plánování (Dyckhoff a Finke, 1992). Příkladem řezných úloh může být například řezání profilů (trubek) na menší kusy s cílem minimalizovat plýtvání a maximalizovat efektivitu výroby.

## **2.2 Dopravní problém**

Tato podkapitola se věnuje popisu dopravního problému jako klíčové optimalizační úlohy v oblasti logistiky. Dopravní problém (dále také DP) je charakterizován distribucí zboží mezi různými zdroji a cíli tak, aby byly minimalizovány přepravní náklady nebo čas. Tato kapitola podrobně rozebírá různé aspekty dopravního problému, představuje jeho teoretický rámec a zkoumá praktické aplikace v logistických systémech (Barnhart a Laporte, 2006). Dopravní problém je optimalizační úloha, která se zabývá plánováním a správou distribuce zboží z jednoho nebo více zdrojů na jedno nebo více míst s cílem minimalizovat celkové přepravní náklady. Tato úloha je relevantní pro širokou škálu včetně výroby, distribuce a dopravy.

## **Klasifikace dopravního problému**

Klasický DP zahrnuje distribuci zboží mezi jedním zdrojem a jedním cílem s cílem minimalizovat přepravní náklady. Mezi další rozdělení patří vícezdrojový dopravní problém, který rozšiřuje klasický problém na situaci s více zdroji, kde je třeba efektivně koordinovat distribuci zboží. Problém, který se zaměřuje na distribuci zboží z více zdrojů do více cílů za účelem minimalizace celkových nákladů se nazývá víceúčelový dopravní problém (Pavlov, 2020). Na rozdíl od klasického dopravního problému, kde se hledá optimální distribuce zboží, se víceúčelový dopravní problém zaměřuje na nalezení nejlevnějšího způsobu, jak uspokojit danou poptávku. Příkladem pro víceúčelový dopravní problém může být například: dopravní společnost, která chce minimalizovat náklady na přepravu zboží mezi svými sklady a zákazníky. Dalším příkladem může být například výrobní firma, která chce optimalizovat přepravu surovin a hotových výrobků mezi svými závody (Barnhart a Laporte, 2006).

## **Aplikace dopravního problému v logistických systémech**

Dopravní problém je klíčová optimalizační úloha v oblasti logistiky. Zabývá se plánováním a správou distribuce zboží z jednoho nebo více zdrojů na jedno nebo více míst s cílem minimalizovat celkové přepravní náklady. DP má široké spektrum aplikací v logistických systémech, od optimalizace distribučních sítí, až po rozvrhování výroby.

## **Optimalizace distribuční sítě**

DP se používá k analýze efektivity a nákladů distribuční sítě. Model DP umožňuje posoudit různé konfigurace sítě, jako je počet skladů, jejich umístění a kapacity, a vybrat tu nejefektivnější variantu s nejnižšími přepravními náklady (Chen, 2016).

Příkladem použití DP může být úloha, v níž velká maloobchodní společnost s rozsáhlou distribuční sítí skladů a prodejen chce optimalizovat distribuci zboží tak, aby minimalizovala celkové přepravní náklady. Společnost může využít model DP k analýze různých konfigurací sítě, včetně:

- Počet skladů: Model DP může posoudit, zda je optimální počet skladů zachovat, zvýšit nebo snížit.

- Umístění skladů: Model DP může analyzovat různé varianty umístění skladů a vybrat ty, které minimalizují přepravní vzdálenosti a náklady.

Kapacity skladů: Model DP může posoudit optimální kapacity skladů s ohledem na objem zboží a požadovanou úroveň zásob. Mezi další možnosti patří rozvrhování výroby, kdy se DP používá k optimalizaci tras a časů přepravy surovin a hotových výrobků v rámci výrobního procesu. Model DP umožňuje koordinovat výrobu a logistiku a minimalizovat celkové náklady na výrobu a distribuci (Sobottka, 2017)

Jako příklad uveďme výrobní závod s více výrobními linkami a sklady surovin a hotových výrobků, který chce optimalizovat rozvržení výroby a logistiky. Společnost může využít model DP k:

- Určení optimální posloupnosti výroby na jednotlivých linkách s ohledem na dostupnost surovin a požadavky na hotové výrobky.
- Plánování tras a časů přepravy surovin a hotových výrobků mezi výrobními linkami a sklady.
- Synchronizaci výroby a logistiky tak, aby se minimalizovaly prostoje a náklady na skladování.

Dopravní problém se používá i v dalších oblastech logistiky jako je například: plánování tras, řízení zásob nebo návrh logistických sítí včetně umístění skladů. V následujícím textu bude uveden matematický model pro kontejnerový dopravní problém (Boucherie, Braaksma a Timjs, 2022).

Matematický model pro kontejnerový dopravní problém v obecném tvaru uvádí Fábry,(2024).

### Parametry

$m$  počet dodavatelů

$n$  počet odběratelů

$c_{ij}$  náklady na přepravu jednoho kontejneru mezi  $i$ -tým dodavatelem a  $j$ -tým odběratelem ( $i = 1, 2, \dots, m; j = 1, 2, \dots, n$ )

$K$  kapacita kontejneru ve stejných jednotkách jako jsou kapacity a požadavky

$a_i$  kapacita  $i$ -tého dodavatele ( $i = 1, 2, \dots, m$ )

$b_j$  požadavky  $j$ -tého odběratele ( $j = 1, 2, \dots, n$ )

### Proměnné

$x_{ij}$  objem týdenní přepravy od  $i$ -tého dodavatele k  $j$ -tému odběrateli

$y_{ij}$  počet kontejnerů, které budou použity k přepravě od  $i$ -tého dodavatele  
k  $j$ -tému odběrateli

### Účelová funkce

$$z = \sum_{i=1}^m \sum_{j=1}^n c_{ij} y_{ij} \rightarrow \min \quad (4)$$

### Omezující podmínky

$$\sum_{j=1}^n x_{ij} \leq a_i \quad i = 1, 2, \dots, m \quad (5)$$

$$\sum_{i=1}^m x_{ij} = b_j \quad j = 1, 2, \dots, n \quad (6)$$

$$x_{ij} \leq K y_{ij} \quad i = 1, 2, \dots, m; j = 1, 2, \dots, n \quad (7)$$

$$x_{ij} \geq 0 \quad i = 1, 2, \dots, m; j = 1, 2, \dots, n \quad (8)$$

$$y_{ij} \geq 0, \text{ celé } i = 1, 2, \dots, m; j = 1, 2, \dots, n \quad (9)$$

Účelová funkce (4), má za cíl minimalizaci celkových přepravních nákladů. Omezující podmínka (5) stanovuje, že nesmíme překročit kapacitu dodavatelů. Dále pak podmínka (6) definuje, že musí být splněny veškeré požadavky odběratelů. Pokud v (7) bude hodnota proměnné  $x_{ij}$  kladná, pak to znamená, že počet použitých kontejnerů  $y_{ij}$  musí být takový, aby se takový daný objem dal přepravit. Nerovnice (7) by byla splněna i pro větší počet kontejnerů, než je minimum. Nicméně cílem úlohy optimalizace přepravy je minimalizovat náklady. Proto se bude v praxi používat nejnižší možný počet kontejnerů, který stále splňuje nerovnici a umožňuje přepravu daného objemu. Podmínka (7) definuje, že když se mezi danou dvojicí dodavatele a odběratele nebude používat žádný kontejner, tedy

$y_{ij} = 0$ , pak hodnota proměnné  $x_{ij}$  bude také nulová, což logicky znamená, že mezi nimi nemůže docházet k žádné přepravě. Omezující podmínky (8) zajišťují nezápornost proměnných  $x_{ij}$ . Nutné jsou také podmínky celočíselnosti a nezápornosti (9) pro proměnné  $y_{ij}$ .

## 2.3 Okružní a rozvozní úlohy

Okružní a rozvozní úlohy (dále také ORÚ) představují důležitou oblast optimalizace v logistice. Zaměřují se na nalezení optimálního způsobu, jak uspokojit danou poptávku po zboží s ohledem na specifické vlastnosti dané úlohy, jako je počet destinací, dostupnost vozidel a časové limity. Řešení ORÚ umožňuje optimalizovat logistické procesy a minimalizovat náklady na dopravu (Mertl, 2007) (Gendreau, Laporte a Séguin, 1996).

### 2.3.1 Okružní úloha

Okružní úloha (dále také OÚ) se zabývá nalezením optimální trasy pro jedno nebo více vozidel, která navštíví danou množinu destinací a vrátí se zpět do výchozího bodu. Cílem je minimalizovat celkovou délku trasy (tras), a tím i náklady na dopravu. Úloha obchodního cestujícího, v anglickém jazyce Traveling Salesman Problem (dále také TSP) je kombinatorický problém v oblasti optimalizace (Gutin a Punnen, 2006) (Applegate, 2006). Cílem této úlohy je najít nejkratší možnou cestu, která projde všemi místy (uzly) a vrátí se zpět do výchozího místa. Každé místo musí být navštíveno právě jednou, a celková délka cesty má být minimalizována. Při hledání přípustného řešení TSP lze využít řadu tzv. heuristických metod, které byly zmíněny v souvislosti s hledáním přípustného řešení dopravního problému a přiřazovacího problému (Fábry, 2019). Příkladem okružní úlohy může být toto zadání: rozvoz pizzy z pizzerie do 20 domů. Cílem je najít optimální trasu pro rozvozce pizzy, která minimalizuje celkovou délku trasy nebo čas doručení. V následujícím textu bude uveden matematický model obchodního cestujícího (Gendreau, Hertz a Laporte, 1994) (Cook, 2011).

#### Parametry

$n$  počet míst (výchozí místo a  $n - 1$  zákazníků)

$c_{ij}$  vzdálenosti mezi místy  $i$  a  $j$  ( $i, j = 1, 2, \dots, n$ )

## Proměnné

$$x_{ij} = \begin{cases} 1, & \text{jestliže vozidlo jede přímo z uzlu } i \text{ do uzlu } j \quad (i, j = 1, 2, \dots, n) \\ 0, & \text{jinak} \end{cases}$$

$u_i$  = umělá proměnná v omezujících podmínkách zabraňujících vytváření parciálních cyklů ( $i = 1, 2, \dots, n$ ).

## Model

$$z = \sum_{i=1}^n \sum_{j=1}^n c_{ij} x_{ij} \rightarrow \min, \quad (10)$$

$$\sum_{j=1}^n x_{ij} = 1 \quad i = 1, 2, \dots, n \quad (11)$$

$$\sum_{i=1}^n x_{ij} = 1 \quad j = 1, 2, \dots, n \quad (12)$$

$$u_i + 1 - (n - 1)(1 - x_{ij}) \leq u_j \quad i = 1, 2, \dots, n; j = 2, 3, \dots, n \quad (13)$$

$$x_{ij} \in \{0, 1\} \quad i = 1, 2, \dots, n; j = 1, 2, \dots, n \quad (14)$$

$$u_i \in R_+ \quad i = 1, 2, \dots, m \quad (15)$$

V předchozím obecném matematickém modelu je definována účelová funkce (10), která má za cíl najít nejkratší možnou cestu, která prochází všemi danými místy a vrací se zpět do výchozího bodu. Soustavy rovnic (11) a (12) zajišťují, aby každé místo v okruhu bylo navštíveno pouze jednou. Jinými slovy, nedochází k duplicitním zastávkám na žádné z pozic v okruhu. Nerovnice (13) s nezápornými proměnnými  $u_i$ , nazývané smyčkové podmínky, zabraňují vzniku parciálních cyklů v rámci řešení úlohy obchodního cestujícího. Podmínky (14) zajišťují binárnost proměnných  $x_{ij}$ , podmínky (15) jsou podmínkami nezápornosti proměnných  $u_i$ . Pokud znázorníme úlohu obchodního cestujícího pomocí úplného grafu, pak optimální řešení bude odpovídat Hamiltonovu cyklu s nejnižším součtem ohodnocení vybraných hran.

### 2.3.2 Rozvozní úloha

Rozvozní úloha v zahraniční literatuře uváděná jako Vehicle Routing Problem (dále také VRP), se zabývá nalezením optimálního způsobu, jak distribuovat zboží z jednoho zdroje do více destinací. Cílem je minimalizovat celkové náklady na dopravu, a to s ohledem na kapacitu vozidel (maximální množství zboží, které může vozidlo převézt), čas (z hlediska dodací lhůty pro jednotlivé destinace) a také omezení/překážky na cestě (jednosměrné ulice, uzávěry, nehody), (Gendreau, Laporte a Séguin, 1996).

VRP se vyskytuje v mnoha oblastech logistiky, kterými jsou dodávky/rozvoz potravin, e-commerce, což je doručování zásilek z e-shopů k zákazníkům a také doručování balíků kurýrní službou. Existuje mnoho algoritmů pro řešení VRP, které ale závisí na vlastnostech dané úlohy, kterými je počet destinací nebo typ vozidel.

Příkladem pro VRP může být distribuce zboží z centrálního skladu do 50 prodejen. Cílem je najít optimální způsob, jak rozložit zboží do vozidel tak, aby se minimalizovaly celkové náklady na dopravu a dodržely se dodací lhůty. V následujícím textu bude uveden matematický model pro rozvozní úlohu (Mertl, 2007) (Fábry, 2014).

#### Parametry

$V$  kapacita vozidla

$n$  počet míst (výchozí místo a  $n - 1$  zákazníků, tj. zakázek)

$q_i$  požadavek  $i$ -tého zákazníka ( $i = 2, 3, \dots, n$ )

$c_{ij}$  vzdálenost mezi místy  $i$  a  $j$  ( $i, j = 1, 2, \dots, n$ )

#### Předpoklady

$$\sum_{i=2}^n q_i > V, \quad (16)$$

$$q_i \leq V \quad (i = 2, 3, \dots, n) \quad (17)$$

Podle předpokladu (16) nelze upokojit požadavky zákazníků v rámci jedné trasy (jednoho okruhu), proto výsledné řešení bude obsahovat více tras. V případě, že by nebyl splněn předpoklad (17), musel by být požadavek zákazníka rozdělen do více



tras. K tomuto účelu se používají modely pro řešení rozvozní úlohy s dělenou dodávkou (Mertl, 2007; Fábry, 2014).

### Proměnné

$$x_{ij} = \begin{cases} 1, & \text{jestliže vozidlo jede přímo z uzlu } i \text{ do uzlu } j \\ 0, & \text{jinak} \end{cases} \quad (i, j = 1, 2, \dots, n)$$

$$u_i = \text{umělá proměnná zavedená pro bilanci nákladu vozidla} \quad (i = 1, 2, \dots, n)$$

### Model

$$z = \sum_{i=1}^n \sum_{j=1}^n c_{ij} x_{ij} \rightarrow \min \quad (18)$$

$$\sum_{j=1}^n x_{ij} = 1 \quad i = 2, 3, \dots, n \quad (19)$$

$$\sum_{i=1}^n x_{ij} = 1 \quad j = 2, 3, \dots, n \quad (20)$$

$$u_i + q_i - V(1 - x_{ij}) \leq u_j \quad i = 1, 2, \dots, n \quad j = 2, 3, \dots, n \quad (21)$$

$$u_i \leq V \quad i = 2, 3, \dots, n \quad (22)$$

$$u_1 = 0; \quad (23)$$

$$x_{ij} \in \{0, 1\} \quad i = 1, 2, \dots, n \quad j = 1, 2, \dots, n \quad (24)$$

$$u_i \in \mathbb{R}_+ \quad i = 2, 3, \dots, n \quad (25)$$

Účelová funkce (18) má za cíl minimalizovat celkovou vzdálenost ujetou vozidlem za podmínek (19) a (20) definujících, že každé místo (kromě místa výchozího) bude navštíveno právě jednou. Nerovnosti (21) zamezují vytváření cyklů neobsahujících výchozí místo a zároveň zohledňují zvyšování nákladu na vozidle. Podle podmínky (22) se nepovoluje překročení kapacity vozidla  $V$  a podmínka (23) zaručuje, že vozidlo bude při každém výjezdu na trasu prázdné. Podmínky (24) zajišťují binárnost proměnných  $x_{ij}$ , podmínky (25) jsou podmínkami nezápornosti proměnných  $u_i$ . Tento model není pro rozvozní úlohu, ale pro úlohu svozu.

## **Klasifikace okružních a rozvozních úloh**

Okružní a rozvozní úlohy (dále také ORÚ) se dále klasifikují na základě specifických vlastností a omezení, které je nutné při jejich řešení zohlednit. Tato klasifikace umožňuje lépe identifikovat typ ORÚ a vybrat vhodný algoritmus pro její řešení (Bersekas, 2012).

### **Okružní úlohy s časovými omezeními**

Tato úloha zahrnuje časová omezení pro návštěvu každé destinace. Cílem je minimalizovat celkovou délku trasy a zároveň dodržet časová okna pro doručení (Applegate, 2006).

### **Rozvozní úlohy s omezením kapacity**

Tato úloha řeší problém omezení kapacity vozidel při distribuci zboží. Cílem je minimalizovat počet vozidel a celkové náklady na dopravu s ohledem na maximální kapacitu každého vozidla (Laporte, 1992).

### **Okružní úloha s více depy**

Tato úloha zahrnuje více dep, ze kterých vozidla vyjíždějí a do kterých se vrací (Applegate, 2006).

### **Úloha s dělenou dodávkou**

Úloha s dělenou dodávkou (také známá jako SDVRP= Split Delivery Vehicle Routing Problem) je logistický problém, který řeší rozvoz zboží z centrálního skladu k mnoha zákazníkům. V případě, že je objednávka pro některé zákazníky vyšší, než je kapacita vozidla, tedy je porušen předpoklad (14), je nutné tuto objednávku rozdělit na více dílčích dodávek, které budou rozvezeny v rámci více tras (Mertl, 2007). Především v dynamických úlohách hraje možnost rozdělení požadavku do několika tras významnou roli a může dojít k výraznému snížení přepravních nákladů, resp. zkrácení celkové vzdálenosti ujeté všemi vozidly (Fábry, 2014).

### 3 Software pro matematické modelování

*Modelovací systémy a řešitelé* optimalizačních problémů jsou klíčovými nástroji v oblasti matematického programování. *Modelovací systémy* slouží k formulaci matematických modelů, zatímco *řešitelé* se specializují na nalezení optimálních nebo přijatelných řešení těchto modelů. Systémy lze rozdělit na uzavřené a otevřené. Uzavřené systémy mají svého vlastního řešitele a nelze k řešení využít žádný jiný nástroj. Otevřené systémy umožňují uživateli výběr z několika externích řešitelů (Fábry, 2019). Mezi otevřené systémy patří například AMPL či MPL. K uzavřeným systémům, které mají své vlastní řešitele patří, např. Lingo a Xpress MP.

*Řešitel* je software, který slouží k nalezení optimálního řešení matematických optimalizačních úloh. Tyto úlohy zahrnují nalezení minima nebo maxima funkce za daných omezení. Řešitelé používají různé algoritmy k prohledávání prostoru řešení a nalezení nejlepšího možného řešení. Existuje mnoho typů řešitelů, které se liší v závislosti na typu optimalizační úlohy, kterou řeší, a na algoritmech, které používají (Janáček, 2003) (Vajda, 2009). K nejpoužívanějším řešitelům, které jsou využity v otevřených systémech patří například CPLEX, Gurobi a Neos. Tyto řešitele podporují širokou škálu optimalizačních metod a integraci s AMPL a dalšími modelovacími jazyky.

Volba řešitele závisí na typu a velikosti optimalizační úlohy, kterou chce analytik řešit, na dostupných zdrojích a na jeho preferencích. Typem úlohy se především myslí její zařazení mezi úlohy lineárního či nelineárního programování, či mezi celočíselné, binární či smíšeně-celočíselné úlohy. Velikost úlohy je většinou dána počtem proměnných a omezujících podmínek. Důležitá je také dostupnost řešitele, pokud jde o licenci, cenu, platformu aj., dále také podpora a komunita (online zdroje, fóra, uživatelská podpora), a v neposlední řadě také typ algoritmů, který řešitel používá (Papadimitriou a Steiglitz, 1998).

### **3.1 MPL for Windows**

Maximal Software je vývojářem softwaru MPL (Mathematical Programming Language), pokročilého modelovacího systému, který umožňuje vývojářům modelů formulovat komplexní optimalizační modely jasným, stručným a efektivním způsobem. Modely vyvinuté v MPL pak lze řešit pomocí široké škály komerčních optimalizátorů dostupných na trhu (MPL, 2002).

#### **Popis systému MPL for Windows**

Modelovací systém MPL for Windows (dále jen MPL), také známý jako matematický programovací jazykový modelovací systém, je softwarový nástroj určený pro matematické programování a optimalizaci. Poskytuje modelovací jazyk na vysoké úrovni, který umožňuje uživatelům snadno formulovat modely a řešit optimalizační problémy. Modelovací systém MPL je široce používán v různých oblastech, včetně operačního výzkumu, managementu a inženýrství. Systém MPL poskytuje modelovací jazyk, který uživatelům umožňuje vyjádřit optimalizační problémy stručným a přirozeným způsobem. Tento jazyk obvykle obsahuje konstrukce pro definování rozhodovacích proměnných, objektivních funkcí a omezení (MPL, 2002).

MPL podporuje různé typy problémů matematického programování, včetně lineárního programování, celočíselného programování, kvadratického programování a nelineárního programování. Tento otevřený software je navržen pro práci s různými optimalizačními řešiteli. Uživatelé mohou formulovat své optimalizační problémy pomocí MPL a poté vybrat vhodného řešitele k nalezení optimálního řešení. Mezi nejpoužívanější řešitele kompatibilní s MPL, patří CPLEX a Gurobi. MPL umožňuje uživatelům snadno vkládat data do svých modelů. To může zahrnovat určení koeficientů, mezí a dalších parametrů. Systém také usnadňuje extrakci výsledků pro další analýzu. Jako každý softwarový systém je i MPL obvykle dodáván s dokumentací, která vysvětluje syntaxe a vlastnosti modelovacího jazyka. Uživatelé mohou vyhledat pokyny v dokumentaci a některé systémy také nabízejí služby uživatelské podpory. MPL se běžně používá v akademickém prostředí pro výuku optimalizace a matematického programování. Jeho intuitivní syntaxe jej zpřístupňuje studentům, kteří se učí optimalizační techniky.

## **Zápis matematického modelu v MPL**

MPL umožňuje definovat matematické modely optimalizačních úloh v deklarativním jazyce (Fábry, 2019). Model se skládá z proměnných, kterými mohou být buď reálná čísla, celá čísla, binární čísla, nebo množiny. Definuje funkci, která má být optimalizována (minimalizována nebo maximalizována). Definuje množinu omezení, které musí proměnné splňovat.

### *TITLE*

Klíčové slovo pro definici názvu modelu.

### *VARIABLES*

Jedná se o deklaraci rozhodovacích proměnných, v případě definice binárních proměnných se použije BINARY VARIABLES, u celočíselných proměnných INTEGER VARIABLES.

### *MODEL*

Za tímto slovem bude následovat definice modelu.

### *MAX*

Používá se pro maximalizační úlohy, pro minimalizační se používá MIN. Za tímto následuje matematický zápis účelové funkce.

### *SUBJECT TO*

Sekce vyhrazena pro omezující podmínky.

### *END*

Toto slovo je posledním slovem celého zápisu.

Tento model lze řešit pomocí libovolného numerického řešitele kompatibilního s MPL, například CPLEX nebo Gurobi. MPL umožňuje formulovat modely v intuitivním a snadno čitelném jazyce. Je kompatibilní s širokou škálou numerických řešitelů. Umožňuje uživatelům definovat vlastní funkce a typy dat. MPL je zastaralý jazyk a v současnosti se již běžně nepoužívá. Software MPL nemá tolik funkcí jako moderní modelovací jazyky, jako je například AMPL (MPL, 2002).

## 3.2 AMPL

Program AMPL, což je zkratka pro „A Mathematical Programming Language“, je vysokoúrovňový matematický modelovací jazyk používaný pro formulování a řešení složitých optimalizačních problémů. AMPL poskytuje flexibilní a intuitivní způsob, jak popsat problémy matematického programování, včetně lineárního programování, celočíselného programování, nelineárního programování a dalších optimalizačních technik.

### Popis systému AMPL

Pomocí programu AMPL lze vyjádřit optimalizační problémy v přirozené algebraické syntaxi, která se velmi podobá matematické formulaci problému. To uživatelům usnadňuje převod skutečných problémů do matematických modelů. Systém AMPL také umožňuje začlenit data z různých zdrojů, definovat rozhodovací proměnné, nastavit omezení a specifikovat účelovou funkci.

Jednou z klíčových výhod použití programu AMPL je jeho schopnost propojení s různými řešiteli, stejně jako u MPL. AMPL funguje jako modelovací jazyk a poskytuje pohodlné rozhraní pro připojení k optimalizačním řešitelům, jako jsou CPLEX, Gurobi a mnoho dalších. To umožňuje využít sílu nejmodernějších optimalizačních algoritmů k efektivnímu řešení formulovaných problémů.

Chceme-li použít AMPL, zapíšeme svůj optimalizační problém do souboru AMPL s příponou ".mod". Poté lze použít překladač AMPL nebo integrované vývojové prostředí (IDE) k načtení souboru modelu a vyvolání řešitelů k vyřešení problému. AMPL poskytuje bohatou sadu příkazů a funkcí pro interakci s modely, daty a výsledky. AMPL je široce používán v akademické sféře a průmyslu pro řešení různých problémů s optimalizací, včetně řízení dodavatelského řetězce, logistiky, financí, energetických systémů a dalších. Jeho flexibilita a kompatibilita s různými řešiteli z něj činí oblíbenou volbu mezi odborníky na optimalizaci a výzkumníky (AMPL, 2003).

## **Matematický model**

AMPL umožňuje definovat matematické modely optimalizačních úloh v deklarativním jazyce (Antoy a Hanus, 2005). Model obsahuje proměnné, kterými mohou být buď reálná čísla, celá čísla, binární čísla, nebo množiny. Účelová funkce je buď minimalizační, či maximalizační. Dále je třeba definovat množinu omezení, která musí proměnné splňovat (AMPL, 2003).

### *SET*

Klíčové slovo pro definici názvu modelu.

### *PARAM*

Toto klíčové slovo deklaruje parametry, kterými mohou být skalární hodnoty, nebo hodnoty v indexované množině.

### *VAR*

Zde je uveden seznam proměnných.

### *MAXIMIZE*

Používá se pro maximalizační úlohy, pro minimalizační se používá MINIMIZE. Za tímto následuje matematický zápis účelové funkce.

### *SUBJECT TO*

Sekce vyhrazena pro omezující podmínky.

Tento model lze řešit pomocí libovolného numerického řešitele kompatibilního s AMPL, například CPLEX nebo Gurobi. AMPL je vhodný pro řešení široké škály optimalizačních úloh, včetně úloh lineárního programování, nelineárního programování, celočíselného programování a také smíšeně-celočíselného programování. Stejně jako u ostatních modelovacích systémů je použití AMPL podmíněno získáním potřebné licence. Pro akademické účely firma nabízí časově omezenou bezplatnou licenci, která nesmí být použita ke komerčním účelům. Tento software se zpočátku může zdát obtížnější než MPL (AMPL, 2003).

### 3.3 Porovnání systémů MPL a AMPL

V této podkapitole budou porovnány dva populární modelovací jazyky pro optimalizační úlohy MPL (MPL, 2002) a AMPL (AMPL, 2003). Zaměříme se na jejich syntaxi, integraci s řešiteli, podporu platforem. Systémy MPL (Mathematical Programming Language) a AMPL (A Mathematical Programming Language) jsou oba nástroje pro modelování a řešení optimalizačních problémů. I když mají podobné názvy, jsou vyvíjeny různými týmy a jsou rozdílné. Následně jsou zde porovnány (viz Tab. 1) vlastnosti jednotlivých programovacích systémů tak, aby bylo možné rozlišit který z těchto dvou softwarů je například jednodušší, dostupnější nebo s jakými řešiteli spolupracuje.

Tab. 1 Porovnání MPL a AMPL

<b>Vlastnost</b>	<b>MPL for Windows</b>	<b>AMPL</b>
<b>Syntaxe</b>	Jednoduchá, intuitivní	Komplexní, formální
<b>Integrace s řešiteli</b>	CPLEX, Gurobi, CoinMP, Sulum, Mosek, Lindo, LPSolve, Ipop, Conopt, Knitro	CPLEX, Gurobi, Baron, Bonmin, CBC, Conopt, Couenne, FilMINT, FILTER, Gecode, IPOPT, JaCoP, Knitr, Lancelot, LGO, LINDO Global, LOQO, MINLP, MINOS, MINTO, MOSEK, NSIPS, OOQP, PATH, RAPOSa, SCIP, SNOPT, Xpress
<b>Podpora platform</b>	Windows	Windows, macOS, Linux
<b>Vhodnost</b>	Začátečníci, menší úlohy	Pokročilí uživatelé, velké a komplexní úlohy

#### Syntaxe

AMPL má komplexnější syntaxi, která je bližší formální matematické notaci. To umožňuje modelovat komplexní optimalizační úlohy s velkou přesností. Naproti tomu MPL for Windows má jednodušší a intuitivnější syntaxi, která je bližší běžnému jazyku. Díky tomu je snadnější se jí naučit a používat i pro začátečníky.

#### Integrace s řešiteli

MPL nemá vlastního řešitele (jedná se o omezenou studentskou verzi), ale v omezené licenci jsou nainstalovány řešitelé CPLEX a Gurobi. Studentská verze



MPL je dostupná zdarma pro studenty a pedagogy. Tato verze umožňuje spouštět modely MPL z příkazového řádku, přístup k datům a parametrům modelů MPL a také použití základních funkcí MPL. V případě AMPL studenti mají k dispozici řadu možností, jak začít s tímto softwarem, včetně bezplatných vzdělávacích materiálů o optimalizaci výuky. AMPL umožňuje integraci s řadou řešitelů optimalizačních úloh, čímž umožňuje uživatelům vybrat si nástroj nejlépe vyhovující jejich potřebám. Mezi nejčastěji využívané řešitele patří CPLEX, Gurobi, COIN-OR CBC nebo KNITRO.

### **Podpora platformem a cenová dostupnost**

AMPL je dostupný pro platformy Windows, macOS a Linux. MPL for Windows je dostupný pouze pro platformu Windows. MPL for Windows je komerční software, ale jeho cena je nižší než u systému AMPL, který je také komerční software, ale naopak s vyšší cenou, která může být pro některé uživatele nedostupná. Pro oba tyto softwary je možné vyzkoušet trial třicetidenní plné verze.

### **Vhodnost a závěrečné zhodnocení**

MPL je vhodný pro začátečníky a pro modelování menších a jednodušších optimalizačních úloh. AMPL je vhodný pro pokročilé uživatele a pro modelování velkých a komplexních optimalizačních úloh. Volba mezi AMPL a MPL závisí na specifických potřebách a preferencích uživatele. Pokud je uživatelem pokročilý programátor, který potřebuje modelovat komplexní optimalizační úlohy, pak je AMPL dobrou volbou. Pokud je uživatelem začátečník, který potřebuje jednoduchý nástroj pro modelování menších optimalizačních úloh, pak je MPL dostačující volbou.

## 4 Aplikace systému AMPL

V této kapitole budou prezentovány modely vybraných úloh z oblasti řízení výroby a logistiky v systému AMPL. Pro každou úlohu bude popsáno zadání, matematický model v jazyce AMPL a popis jednotlivých parametrů, proměnných, účelové funkce a omezujících podmínek. Jednotlivé úlohy jsou zároveň porovnány se systémem MPL for Windows, pro ověření správnosti zápisu a výsledku.

### 4.1 Řezná úloha

Firma se zabývá výrobou dřevěného stojanu na notebook a dřevěné krabičky na šperky. Za 20 dní proběhne výstava, na které se budou výrobky prodávat. Cena stojanu na notebook bude 2699 Kč a cena za krabičku na šperky bude 3087 Kč. Při výrobě se kromě jiného používají prkna o délce 30 cm, prkna dlouhá 25 cm a vruty, jichž je k dispozici 4000 kusů. Firma má k dispozici 500 prken dlouhých 1,1 m a 150 prken o délce 1,4 m. Výrobce pracuje denně 8 hodin. Spotřeba materiálu a časové nároky na výrobu jsou definovány v Tab. 2. Cílem je naplánovat výrobu tak, aby celkové tržby z prodeje výrobků byly maximální. V Tab. 3 jsou obsažena řezná schémata dle různé délky kratších dílů. Existuje celkem 9 možností, jak získat požadovaná prkna.

**Tab. 2 Spotřeba materiálu a časové nároky na výrobu**

	Stojan	Šperkovnice
Prkna 30 cm	2	1
Prkna 25 cm	4	1
Vruty	14	8
Čas (v min)	60	30

**Tab. 3 Tabulka řezných schémat**

Číslo schématu	Prkno o délce 1,1m				Prkno o délce 1,4m				
	1	2	3	4	5	6	7	8	9
Prkno o délce 30cm	3	2	1	0	4	3	2	1	0
Prkno o délce 25cm	0	2	3	4	0	2	3	4	5
Odpad (v cm)	20	0	5	10	20	0	5	10	15

## Proměnné

$x_i$  = počet dlouhých prken 1,1 m a 1,4 m rozřezaných dle schématu ( $i = 1, 2, \dots, 9$ )

$y_1$  = počet vyrobených stojanů na notebook

$y_2$  = počet vyrobených krabiček na šperky

## Účelová funkce

$$z = 450y_1 + 300y_2 \rightarrow \max \quad (26)$$

## Omezující podmínky

$$14y_1 + 8y_2 \leq 4000 \quad (27)$$

$$y_1 + 0,5y_2 \leq 160 \quad (28)$$

$$x_1 + x_2 + x_3 + x_4 \leq 500 \quad (29)$$

$$x_5 + x_6 + x_7 + x_8 + x_9 \leq 150 \quad (30)$$

$$3x_1 + 2x_2 + x_3 + 4x_5 + 3x_6 + 2x_7 + x_8 \geq y_1 + 2y_2 \quad (31)$$

$$2x_2 + 3x_3 + 4x_4 + 2x_6 + 3x_7 + 4x_8 + 5x_9 \geq y_1 + 4y_2 \quad (32)$$

$$x_1, x_2, \dots, x_9 \geq 0, \text{ celé} \quad (33)$$

$$y_1, y_2 \geq 0, \text{ celé} \quad (34)$$

Účelová funkce (26) představuje maximalizaci tržeb z prodeje výrobků, které byly vyrobeny z nařezaného materiálu. Omezení (27) vyjadřuje množství vrutů, které máme k dispozici, a zároveň potřebné množství vrutů na každý výrobek. Druhé omezení (28) definuje časový rámec, který nesmí být překročen a zároveň znázorňuje čas pro výrobu jednotlivých výrobků. Třetím omezením (29) je množství prken velikosti 1,1m, které máme k dispozici. Čtvrtým omezením (30) je množství prken velikosti 1,4m, které máme k dispozici. Nerovnice (31) se týká prken dlouhých 30 cm a nerovnice (32) se týká prken dlouhých 25 cm. Řezáním jich musíme získat alespoň tolik, kolik jich bude použito při výrobě stojanu na notebook a šperkovnici.

Podmínky (33) a (34) jsou podmínkami nezápornosti a celočíselnosti pro všechny proměnné.

### Zápis modelu řezné úlohy do AMPL

V rámci praktické části byl definován model pro řeznou úlohu, který vychází ze zadání výše. Tento model byl vyzkoušen v optimalizačním systému AMPL a byly k němu vytvořeny všechny potřebné soubory, které naleznete (viz Obr. 1) níže. Pro vizualizaci v programu AMPL je tento model součástí přílohy 1.

```
1 #Set;
2 set PRKNA;
3 set DIL;
4 set SCHEMA;
5 set VYROBEK;
6
7 #Parametry
8 param VrutyDisp          >= 0;
9 param CasDisp           >= 0;
10 param Pocet {PRKNA}    >= 0;
11 param Odpad {SCHEMA}   >= 0;
12 param Vrut {VYROBEK}   >= 0;
13 param Cas {VYROBEK}    >= 0;
14 param c {VYROBEK}      >= 0;
15 param a {DIL,SCHEMA}   >= 0;
16 param Pozadavek {DIL,VYROBEK} >= 0;
17 param d {PRKNA,SCHEMA} >= 0;
18
19 #Proměnné
20 var x {SCHEMA} integer >= 0;
21 var y {VYROBEK} integer >= 0;
22
23 #Účelová funkce
24 maximize Celkem:
25     sum {j in VYROBEK} c[j]*y[j];
26
27 #Omezující proměnné
28 subject to Omezeni1 {i in DIL}:
29     sum {j in SCHEMA} a[i,j]*x[j] >= sum {j in VYROBEK} Pozadavek[i,j]*y[j];
30
31 subject to Omezeni2:
32     sum {j in VYROBEK} Vrut[j]*y[j] <= VrutyDisp;
33
34 subject to Omezeni3:
35     sum {j in VYROBEK} Cas[j]*y[j] <= CasDisp;
36
37 subject to Omezeni4 {i in PRKNA}:
38     sum {j in SCHEMA} d[i,j]*x[j] <= Pocet[i];
```

Obr. 1 Model řezné úlohy v AMPL

Nejprve se do modelu zapíše množina **PRKNA**, která obsahuje dva typy prken a sice 1,1 m a 1,4m. Další množinou je **DIL**. Tato množina obsahuje dva typy dílů, které se používají k výrobě, tedy prkna o délce 30 cm a 25 cm. Dále zapíšeme množinu **SCHEMA**, v níž jsou schémata pro výrobu dílů z prken. Pro typ dílu, budeme dále používat index  $i$ , pro pořadové číslo schématu index  $j$ . Poslední

množinou je množina **VYROBEK**, obsahující dva typy výrobků, a sice krabičku na šperky a stojan na notebook.

Dále popíšeme všechny parametry modelu. U každého parametru je v závorce uveden příslušný název množiny. Parametr *VrutyDisp* definuje počet vrutů k dispozici v dílně. Dalším parametrem je *CasDisp*, který definuje maximální možný čas na dokončení obou výrobků, v tomto případě se jedná o 160 h (20 pracovních dní). Dalším parametrem je *Pocet*, který představuje disponibilní počet prken dlouhých 1,1m a 1,4 m, která jsou součástí množiny **PRKNA**. Dále zapíšeme parametr *Odpad*, tento parametr je definován pro jednotlivá schémata, tedy pro prvky množiny **SCHEMA**. Jedná se o vzniklý odpad, který získáme po rozřezání jednotlivých délek prken. Tyto hodnoty uvádíme i přesto, že jsou pro řešení úlohy irelevantní. Pokračujeme s parametrem *Vrut*, jehož hodnoty jsou počtem potřebných vrutů pro výrobu jednotlivých typů výrobků. Proto je tento parametr definován pro množinu **VYROBEK**. Podobným parametrem je *Cas*, který stanovuje čas potřebný pro výrobu jednotlivých výrobků. Parametr *c* představuje cenu výrobků. Parametr *a* je zadán pro množiny **DIL** a **PRKNA**. Jeho hodnoty tvoří matici koeficientů na levých stranách omezujících podmínek (29) a (30). Parametr *Pozadavek* uvádí, kolik prken typu 30 cm a 25 cm je potřeba k výrobě jednotlivých typů výrobků. Tento parametr je definovaný pro množiny **DIL** a **VYROBEK**. Posledním parametrem je parametr *d*, zavedený pro množiny **PRKNA** a **SCHEMA**. Podobně jako u parametru *a* jeho hodnoty tvoří matici koeficientů na levých stranách omezujících podmínek, tentokrát nerovnic (31) a (32).

Pokračujeme definováním proměnných. Podobně jako u parametru také u proměnných je vždy v závorce uveden název množiny, které se proměnná týká. Označení  $x$  používáme pro proměnné  $x_j$ ,  $y$  pak pro proměnné  $y_j$ .

Ze zadání účelová funkce (26) vyplývá, že chceme maximalizovat celkové tržby. Proto se používá klíčové slovo **maximize**. Ve složených závorkách u funkce **sum** je uveden index a název příslušné množiny. V tomto případě se jedná o množinu výrobků, pro které je použit index  $j$  v parametru  $c[j]$  a proměnné  $y[j]$ .

Nakonec je zapotřebí definovat omezující podmínky. Každá podmínka je uvozena slovním spojením **subject to**. Pokud se jedná o soustavu omezujících podmínek, je za názvem omezení ve složených závorkách uveden název množiny a index pro její

prvky. Následuje samotné omezení, tedy jeho levá a pravá strana. Ve většině případů je opět použita funkce **sum** podobně jako v účelové funkci. Zápis je analogický právě zápisu vysvětleném při definici účelové funkce.

Pokud se jedná pouze o jednu omezující podmínku, např. **Omezení2** (viz Obr. 1), pak samozřejmě za názvem omezení specifikace množiny a příslušného indexu ve složených závorkách chybí.

Po sestavení samotného modelu následuje vytvoření datového souboru dat. (viz Obr. 2), který bude spolupracovat s MS Excel. Jako první vytvoříme tabulku zdroje v MS Excel Obr.3, která obsahuje maximální disponibilní čas, tedy 160 h, a dále maximální množství vrutů k dispozici, v tomto případě se jedná o 4 000 ks. Tato tabulka musí být zanesena do AMPL pomocí **table** „název tabulky“, dále je uveden doplněk **amplxl**, který musí být použit z důvodu správné kooperace s MS Excel, a nakonec se uvede list, na kterém je tabulka umístěna. Další tabulkou je tabulka *reznaschemata* Obr.4, ve které jsou zapsána jednotlivá schémata, v tomto případě 9 schémat společně s vytvořeným odpadem vzniklým při řezání jednotlivých prken. Pokračujeme tabulkou prkna Obr.5, tato tabulka obsahuje počet prken, která jsou potřeba k výrobě jednotlivých typů výrobků. Tabulka *vyrobky* udává spotřebu vrutů, času a hodnoty cen pro jednotlivé typy výrobků (viz Obr. 6). Je nezbytné také definovat *dily* (viz Obr.7), tato tabulka uvádí typy prken, která jsou potřebná pro výrobu krabičky na šperky a stojanu na notebook. Matice *paramPozadavek* v tomto případě definuje počet prken 30 cm a 25 cm (viz Obr. 8) potřebných pro výrobu jednotlivých výrobků. Hodnoty *parama* tvoří matici (viz Obr. 9) koeficientů na levých stranách omezujících podmínek (29) a (30). Hodnoty *paramd* viz (Obr. 10), tvoří matici koeficientů na levých stranách omezujících podmínek (31) a (32). Nakonec pomocí klíčových slov **read table** načteme všechny tabulky ze souboru MS Excel.

```

1 data;
2
3
4 table zdroje IN "amplx1" "cut.xlsx":
5     [], CasDisp, VrutyDisp;
6
7 table reznaschemata IN "amplx1" "cut.xlsx":
8     SCHEMA <- [schemata], Odpad;
9
10 table prkna IN "amplx1" "cut.xlsx":
11     PRKNA <- [prkna], Pocet;
12
13 table vyrobky IN "amplx1" "cut.xlsx":
14     VYROBEK <- [vyrobky], Vrut, Cas, c;
15
16 table dily IN "amplx1" "cut.xlsx":
17     DIL <- [dily];
18
19 table paramPozadavek IN "amplx1" "2D" "cut.xlsx":
20     [DIL, VYROBEK], Pozadavek;
21
22 table parama IN "amplx1" "2D" "cut.xlsx":
23     [DIL, SCHEMA], a;
24
25 table paramd IN "amplx1" "2D" "cut.xlsx":
26     [PRKNA, SCHEMA], d;
27
28 read table zdroje;
29 read table reznaschemata;
30 read table prkna;
31 read table vyrobky;
32 read table dily;
33 read table paramPozadavek;
34 read table parama;
35 read table paramd;

```

Obr. 2 Datový soubor v AMPL

E	F
<b>ZDROJE</b>	
CasDisp	VrutyDisp
160	4000

Obr. 3 Zdroje

G	H
<b>SCHEMATA</b>	
schemata	Odpad
S1	20
S2	0
S3	5
S4	10
S5	20
S6	0
S7	5
S8	10
S9	15

Obr. 4 Řezná schémata

	A	B
1		
2	<b>VSTUPNI MATERIAL</b>	
3	prkna	Pocet
4	p110	500
5	p140	150

Obr. 5 Délka a počet prken k dispozici

12	<b>VYROBKY</b>			
13	vyrobky	Vrut	Cas	c
14	stojan	14	1	450
15	krabicka	8	0,5	300

Obr. 6 Spotřeba vrutů, času a ceny jednotlivých výrobků

7	<b>DILY</b>
8	dily
9	Dil25
10	Dil30

Obr. 7 Jednotlivé délky prken potřebných pro výrobu obou výrobků

<b>KUSOVNIK</b>		
DIL	stojan	krabicka
Dil30	1	1
Dil25	1	4

Obr. 8 Potřebná prkna 30 cm a 25 cm pro jednotlivé výrobky

PARAMETR D	S1	S2	S3	S4	S5	S6	S7	S8	S9
PRKNA									
p110		1	1	1	1	0	0	0	0
p140		0	0	0	0	1	1	1	1

Obr. 9 Hodnoty parametru d u proměnných týkajících se původních prken

PARAMETRA	S1	S2	S3	S4	S5	S6	S7	S8	S9
DIL									
Dil30		3	2	1	0	4	3	2	1
Dil25		0	2	3	4	0	2	3	4

Obr. 10 Hodnoty parametru a vyplývající z tabulky řezných schémat

Pro spuštění optimalizace je zapotřebí vytvořit poslední soubor s příponou run, v němž se musí postupně načíst jednotlivé soubory, a navíc umožněn výstup v souboru MS Excel (viz Obr. 11). Pro výstup v MS Excel je nutné definovat buňky, do kterých se výsledek promítne. Jednotlivé výstupy a výsledky v AMPL a v MS Excel jsou dostupné na Obr. 12 a Obr. 13.



```

1 reset;
2
3 model cut.mod;
4
5 load amplx1.dll;
6
7 include cut.dat;
8
9 option solver gurobi;
10
11 solve;
12
13 display x, y;
14
15 display Celkem;
16
17 table vystup OUT "amplx1" "cut.xlsx":
18     [schema], x, y;
19
20 table vysledek OUT "amplx1" "cut.xlsx":
21     [], Celkem;
22
23 write table vystup;
24 write table vysledek;

```

Obr. 11 Spuštění optimalizace v AMPL

```

:      x      y      :=
S1      0      .
S2      0      .
S3      0      .
S4      500    .
S5      80     .
S6      0      .
S7      0      .
S8      0      .
S9      70     .
krabicka .    320
stojan  .      0
;

Celkem = 96000

```

Obr. 12 Výsledek v AMPL

Z výsledku na Obr. 12 je zřejmé, kolik bude vyrobeno krabiček a kolik stojanů na notebook. Dále je zde uvedena i maximální hodnota tržeb. Pro porovnání je na Obr. 13 uveden stejný výsledek exportovaný do souboru MS Excel.

19	Celkem		schema	x	y
20	96000		S1	0	Missing
21			S2	0	Missing
22			S3	0	Missing
23			S4	500	Missing
24			S5	80	Missing
25			S6	0	Missing
26			S7	0	Missing
27			S8	0	Missing
28			S9	70	Missing
29			stojan	-	0
30			krabicka	-	320

Obr. 13 Výsledek v MS Excel

## 4.2 Kontejnerový dopravní problém

Přepravní společnost účtuje ceny za pronájem jedné cisterny mezi jednotlivými dodavateli a odběrateli. K přepravě lze použít cisterny o kapacitě 20 hl. V Tab. 4 jsou definovány náklady na přepravu společně s kapacitou dodavatelů a požadavků odběratelů. Cílem je jednak určit, kolik litrů benzínu se bude přepravovat mezi jednotlivými místy, ale také stanovit, kolik cisteren bude na tuto přepravu použito, aby byly celkové přepravní náklady minimální.

*Tab. 4 Náklady na přepravu, kapacity dodavatelů a požadavky odběratelů*

Dodavatel /Odběratel	Mladá Boleslav	Příbram	Hradec Králové	Brno	Kapacita (hl)
Praha	100	60	120	210	110
Olomouc	230	330	150	80	160
Jihlava	180	170	120	90	180
Poptávka (hl)	80	120	60	130	

### Zápis modelu kontejnerového dopravního problému v AMPL

V rámci teoretické části byl definován model pro kontejnerový dopravní problém. Tento model nyní aplikujeme na uvedené příklady v optimalizačním systému AMPL. Všechny potřebné soubory budou vysvětleny níže. Jejich zápis jsou tyto zápisy také součástí přílohy 2.

První se zapíše do souboru s modelem (viz Obr. 14) množina **DODAVATEL**. Tato množina zobrazuje tři dodavatele, kteří budou plnit požadavek čtyř odběratelů definovaných množinou **ODBĚRATEL**.

```

1
2 set DODAVATEL;
3 set ODBERATEL;
4
5
6 param Naklady {DODAVATEL, ODBERATEL} >=0;
7 param Kapacita {DODAVATEL} >=0;
8 param Pozadavek {ODBERATEL} >=0;
9 param K >=0;
10
11
12 var x {DODAVATEL, ODBERATEL} >= 0;
13 var y {DODAVATEL, ODBERATEL} integer >= 0;
14
15
16 minimize Celkem:
17     sum {i in DODAVATEL, j in ODBERATEL} Naklady[i,j]*y[i,j];
18
19
20 subject to Omezeni1 {i in DODAVATEL}:
21     sum {j in ODBERATEL} x[i,j] <= Kapacita[i];
22
23 subject to Omezeni2 {j in ODBERATEL}:
24     sum {i in DODAVATEL} x[i,j] = Pozadavek[j];
25
26 subject to Omezeni3 {i in DODAVATEL, j in ODBERATEL}:
27     x[i,j] <= K*y[i,j];
28

```

Obr. 14 Kontejnerový dopravní problém – zápis modelu

Dále opět popíšeme všechny parametry modelu. U každého parametru je v závorce uveden příslušný název množin, kterých se daný parametr týká. Prvním parametrem v modelu jsou *Naklady* na přepravu jednoho kontejneru mezi dodavatelem a odběratelem. Proto je tento parametr definován pro množiny **DODAVATEL** a **ODBERATEL**. Může zohledňovat vzdálenost, typ dopravy nebo palivové náklady. Další parametr *Kapacita* představuje kapacitu dodavatelů (v hl). Ta může být ovlivněna dostupností zboží, výrobní kapacitou a velikostí skladu. Pokračujeme ke stanovení požadavku odběratelů (v hl) popsáných parametrem *Pozadavek*. Kapacita *K* definuje kapacitu jednoho kontejneru, která je v tomto případě 20 hl.

Pokračujeme definicí proměnných. Podobně jakou u parametrů, také u proměnných je vždy v závorce uveden název množiny, které se proměnná týká. Proměnná *x* definuje objem přepravy (v hl). Proměnná *y* udává počet použitých kontejnerů.

Ze zadání účelové funkce (4) vyplývá, že chceme minimalizovat náklady na přepravu. Proto je použito klíčové slovo **minimize**. Ve složených závorkách u funkce **sum** je uveden index a název příslušných množin. V tomto případě se

jedná o množiny **DODAVATEL** a **ODBERATEL**, pro které je použit index  $i$  a  $j$  v parametru  $Naklady[i, j]$  a proměnných  $x[i, j]$  a  $y[i, j]$ .

Následně je nutné definovat omezující podmínky. Celý postup byl podrobně vysvětlen u řešení rezné úlohy, proto jednotlivé podmínky nebudeme dále rozebírat.

Dále vytvoříme datový soubor (viz Obr. 15), který bude spolupracovat se souborem MS Excel. V MS Excel je potřeba vytvořit jednotlivé tabulky, se kterými bude AMPL pracovat. Začínáme tabulkou *Kapacita*, který definuje kapacitu kontejneru k přepravě. Dále pokračujeme s tabulkou *Dodavatel*, kde jsou definováni jednotliví dodavatelé a jejich dostupná kapacita. Pokračujeme tabulkou *Odberatel*, v níž jsou definováni jednotliví odběratelé a jejich požadavky. Matice *Naklady* obsahuje náklady spojené s přepravou, které se na základě definované účelové funkce snažíme snížit tak, aby celkové přepravní náklady byly minimální.

```
1 data;
2
3 table Kapacita IN "amplx1" "kontejner.xlsx":
4     [], K;
5
6 table Dodavatel IN "amplx1" "kontejner.xlsx":
7     DODAVATEL <- [Dodavatel], Kapacita;
8
9 table Odberatel IN "amplx1" "kontejner.xlsx":
10    ODBERATEL <- [Odberatel], Pozadavek;
11
12 table Naklady IN "amplx1" "2D" "kontejner.xlsx":
13    [DODAVATEL, ODBERATEL], Naklady;
14
15 read table Kapacita;
16 read table Dodavatel;
17 read table Odberatel;
18 read table Naklady;
19
```

Obr. 15 Kontejnerový dopravní problém – definice dat

Nakonec je nutné vytvořit soubor pro spuštění optimalizace (viz. Obr. 16). Ze zápisu je zřejmé, že výsledky budou opět zobrazeny jak v AMPL, tak budou exportovány do souboru MS Excel do tabulek *vystup* a *vysledek*.

```

1 reset;
2
3 load "amplx1.dll";
4
5 option solver gurobi;
6
7 model kontejner.mod;
8 data kontejner.dat;
9
10 solve;
11
12 display x, y;
13 display Celkem;
14
15 table vystup OUT "amplx1" "kontejner.xlsx":
16     [DODAVATEL, ODBERATEL], x, y;
17
18 table vysledek OUT "amplx1" "kontejner.xlsx":
19     [], Celkem;
20
21 write table vystup;
22 write table vysledek;

```

Obr. 16 Kontejnerový dopravní problém – soubor pro spuštění optimalizace

Na Obr. 17 jsou uvedena vstupní data v souboru MS Excel. V oblasti A3:E6 se nachází náklady na přepravu jednoho kontejneru mezi  $i$  –tým dodavatelem a  $j$  –tým odběratelem. V oblasti A10:B13 jsou definováni jednotliví dodavatelé a jejich kapacity. V oblasti A16:B20 se nachází jednotliví odběratelé společně s jejich požadavky. V buňce B24 je definována kapacita jednoho kontejneru, tedy 20 hl.

	A	B	C	D	E
1					
2					
3	DODAVATEL	Mlada Boleslav	Pribram	Hradec Kralove	Brno
4	Praha	100	60	120	210
5	Olomouc	230	330	150	80
6	Jihlava	180	170	120	90
7					
8					
9					
10	Dodavatel	Kapacita			
11	Praha	110			
12	Olomouc	160			
13	Jihlava	180			
14					
15					
16	Odberatel	Pozadavek			
17	Mlada Boleslav	80			
18	Pribram	120			
19	Hradec Kralove	60			
20	Brno	130			
21					
22					
23					
24	K				
25	20				

Obr. 17 Vstupní data v MS Excel

Na Obr. 18 je uveden výstup z optimalizace v souboru MS Excel. V oblasti A1:D13 jsou objemy přepravy v hl a počty použitých kontejnerů. V buňce F6 se nachází minimální celkové přepravní náklady.

	A	B	C	D	E	F
1	DODAVATEL	ODBERATEL	x	y		
2	Praha	Mlada Boleslav	0	0		
3	Praha	Pribram	100	5		
4	Praha	Hradec Králove	0	0		
5	Praha	Brno	0	0		Celkem
6	Olomouc	Mlada Boleslav	0	0		2110
7	Olomouc	Pribram	0	0		
8	Olomouc	Hradec Králove	0	0		
9	Olomouc	Brno	130	7		
10	Jihlava	Mlada Boleslav	80	4		
11	Jihlava	Pribram	20	1		
12	Jihlava	Hradec Králove	60	3		
13	Jihlava	Brno	0	0		

Obr. 18 Výstup v MS Excel

Pro úplnost ještě uvedme výstup přímo v AMPL (viz. Obr. 19).

```

ampl: Gurobi 11.0.0: Gurobi 11.0.0: optimal solution; objective 2110
17 simplex iteration(s)
1 branching node(s)
:
:
Jihlava Brno                x    y    :=
Jihlava 'Hradec Králové'    60    3
Jihlava 'Mladá Boleslav'    80    4
Jihlava Příbram             20    1
Olomouc Brno                130   7
Olomouc 'Hradec Králové'    0     0
Olomouc 'Mladá Boleslav'    0     0
Olomouc Příbram             0     0
Praha Brno                  0     0
Praha 'Hradec Králové'     0     0
Praha 'Mladá Boleslav'     0     0
Praha Příbram              100   5
;

Celkem = 2110

```

Obr. 19 Výstup v AMPL

### 4.3 Rozvozní problém

Obchodní zástupce pivovaru uzavřel se všemi restauracemi výhodné smlouvy. V tabulce jsou uvedeny počty sudů, které budou restaurace pravidelně odebírat. Pro rozvoz bude použito vozidlo s kapacitou 60 sudů. Cílem je uspokojit všechny požadavky s minimální délkou všech tras. Požadavky společně s délkou trasy jsou uvedeny v Tab. 5.

**Tab. 5 Požadavky s délkou trasy**

	Požadavek
Velvary	0
Kralupy	8
Libčice	10
Slany	18
Zlonice	12
Vrany	11
Briza	10
Veltrusy	6

#### Zápis modelu rozvozní úlohy v AMPL

K řešení úlohy byl použit matematický model (18) – (25). Opět stručně představíme všechny potřebné soubory, jejichž celý zápis je uveden v příloze 3.

Úplně na začátku je potřeba v souboru s modelem (viz Obr. 20) definovat množinu **OBEC**, která zahrnuje všechny obce včetně depa.

Dále popíšeme jednotlivé parametry tohoto modelu. Prvním je *Pozadavek*, který definuje požadované množství sudů v restauracích. Druhým parametrem je *Vzdalenost*, ten určuje vzdálenost mezi obcemi  $i$  a  $j$ . Parametr *Kapacita* definuje kapacitu vozidla.

Pokračujeme definicí proměnných. Proměnná, které je v AMPL označena  $x$ , je binární proměnná  $x_{ij}$ , která má hodnotu 1, pokud vozidlo jede z obce  $i$  do obce  $j$ , jinak nabývá hodnoty 0. Parametr  $u$  je pomocná proměnná, jejíž zavedení vyplývá z omezujících podmínek (22) a (23).

```

1 set OBEC ordered;
2
3 param Pozadavek {i in OBEC} >= 0;
4 param Vzdalenost {i in OBEC, j in OBEC} >= 0;
5 param Kapacita;
6
7 var x {i in OBEC, j in OBEC} binary;
8 var u {i in OBEC};
9
10 minimize CelkovaVzdalenost:
11     sum {i in OBEC, j in OBEC} Vzdalenost[i,j]*x[i,j]
12     ;
13
14 subject to odjezdjednou {i in OBEC: i != first(OBEC)}:
15     sum {j in OBEC} x[i,j] = 1;
16
17 subject to prijezdjednou {j in OBEC: j != first(OBEC)}:
18     sum {i in OBEC} x[i,j] = 1;
19
20 subject to cykly {i in OBEC, j in OBEC: j!= first(OBEC)}:
21     u[i] + Pozadavek[j] - Kapacita*(1-x[i,j]) <= u[j];
22
23 subject to kapacita {i in OBEC: i != first(OBEC)}:
24     u[i]<= Kapacita;
25
26 subject to prvnimisto {i in OBEC: i = first(OBEC)}:
27     u[i] = 0;
28

```

Obr. 20 Rozvozní úloha – model v AMPL

Pokračujeme v definování účelové funkce (18), která v tomto modelu má za cíl minimalizovat celkovou vzdálenost ujetou vozidlem, a jednotlivých omezujících podmínek.

Dále je zapotřebí vytvořit datový soubor, který bude spolupracovat s MS Excel (viz Obr. 21). Tabulka *obec* obsahuje jednotlivé obce a jejich požadavky (viz Obr. 22). Tabulka *capacity* definuje kapacitu vozidla, v tomto případě 60 sudů (viz Obr. 23). Poslední je tabulka *vzdálenosti*, která obsahuje vzdálenosti mezi jednotlivými obcemi (viz Obr. 24).

```

1 data;
2
3
4 table obec IN "amplxl" "rozvozni.xlsx":
5     OBEC <- [obec], Pozadavek;
6
7 table capacity IN "amplxl" "rozvozni.xlsx":
8     [], Kapacita;
9
10 table vzdalenosti IN "amplxl" "2D" "rozvozni.xlsx":
11     [OBEC, OBEC1], Vzdalenost;
12
13 read table capacity;
14 read table obec;
15 read table vzdalenosti;
16

```

Obr. 21 Data pro rozvozní úlohu



obec	Pozadavek
Velvary	0
Kralupy	8
Libcice	10
Slany	18
Zlonice	12
Vrany	11
Briza	10
Veltrusy	6

Obr. 22 Jednotlivé obce v MS Excel

Kapacita	
60	

Obr. 23 Kapacita vozidla

	A	B	C	D	E	F	G	H	I	J
1										
2										
3		RESTAURACE	Velvary	Kralupy	Libcice	Slany	Zlonice	Vrany	Briza	Veltrusy
4		Velvary	0	8	14	13	10	17	12	9
5		Kralupy	8	0	6	16	18	25	17	4
6		Libcice	14	6	0	22	24	31	23	10
7		Slany	13	16	22	0	7	14	20	20
8		Zlonice	10	18	24	7	0	7	13	19
9		Vrany	17	25	31	14	7	0	15	26
10		Briza	12	17	23	20	13	15	0	13
11		Veltrusy	9	4	10	20	19	26	13	0

Obr. 24 Vzdálenosti mezi jednotlivými obcemi

Nakonec pro správné načtení tabulek ukončíme datový soubor se souborem příkazů **read table** ke správnému postupnému načtení všech tabulek vytvořených v MS Excel.

```

1 reset;
2
3 model rozvozni.mod;
4
5 load amplxl.dll;
6
7 include rozvozni.dat;
8
9 option solver cplex;
10
11 solve;
12
13 display x, u, CelkovaVzdalenost;
14
15 table uzly OUT "amplxl" "2D" "rozvozni.xlsx":
16     [OBEC, OBEC1], x;
17
18 table vysledek OUT "amplxl" "rozvozni.xlsx":
19     [], CelkovaVzdalenost;
20
21 table umelaPromenna OUT "amplxl" "rozvozni.xlsx":
22     [OBEC], u;
23
24 write table uzly;
25 write table vysledek;
26 write table umelaPromenna;

```

Obr. 25 Spuštění optimalizace rozvozní úlohy

Poslední částí je soubor s příponou run, díky kterému spustíme optimalizaci. Obsahuje načtení souborů s modelem a vstupními daty. Jak lze vidět na Obr. 25, jako řešitel byl použit CPLEX. Výsledek optimalizace, uvedený přímo v AMPL, je k nahlédnutí na Obr. 26. Pro ověření shody byl přidán ještě výsledek v MS Excel pro porovnání výstupu (viz Obr. 27).

```
x [*,*]
:
Velvary Kralupy Libcice Slany Zlonice Vrany Briza Veltrusy :=
Velvary 0 0 1 0 0 0 1 0
Kralupy 0 0 0 0 0 0 0 1
Libcice 0 1 0 0 0 0 0 0
Slany 1 0 0 0 0 0 0 0
Zlonice 0 0 0 1 0 0 0 0
Vrany 0 0 0 0 1 0 0 0
Briza 0 0 0 0 0 1 0 0
Veltrusy 1 0 0 0 0 0 0 0
;

u [*] :=
Velvary 0
Kralupy 18
Libcice 10
Slany 60
Zlonice 33
Vrany 21
Briza 10
Veltrusy 24
;

CelkovaVzdalenost = 87
```

Obr. 26 Výstup výsledků v AMPL

	A	B	C	D	E	F	G	H	I
1	OBEC	Velvary	Kralupy	Libcice	Slany	Zlonice	Vrany	Briza	Veltrusy
2	Velvary	0	0	1	0	0	0	1	0
3	Kralupy	0	0	0	0	0	0	0	1
4	Libcice	0	1	0	0	0	0	0	0
5	Slany	1	0	0	0	0	0	0	0
6	Zlonice	0	0	0	1	0	0	0	0
7	Vrany	0	0	0	0	1	0	0	0
8	Briza	0	0	0	0	0	1	0	0
9	Veltrusy	1	0	0	0	0	0	0	0
10									
11									
12									
13	CelkovaVzdalenost								
14	87								
15									
16	OBEC	u							
17	Velvary	0							
18	Kralupy	18							
19	Libcice	10							
20	Slany	60							
21	Zlonice	33							
22	Vrany	21							
23	Briza	10							
24	Veltrusy	24							

Obr. 27 Výstup výsledků v MS Excel

Cílem bylo uspokojit všechny požadavky s minimální délkou všech tras. Celková minimální vzdálenost byla tedy 87, a jak je viditelné z výstupu v AMPL, tak je postupně vidět, odkud kam se pojedje, aby trasa byla minimální a zároveň byly splněny všechny požadavky.

## Závěr

Práce byla zaměřena na využití systému AMPL pro modelování a optimalizaci výrobních a dopravních problémů. Teoretická část práce shrnuje poznatky z oblasti optimalizace a lineárního programování, byly naformulovány obecné matematické modely řezné úlohy, kontejnerového dopravního problému a rozvozního problému. V rámci práce byl představen systém AMPL, který je alternativou systému MPL for Windows, dosud používaného ve výuce na Škoda Auto Vysoké škole. Tyto dva systémy byly vzájemně porovnány.

Praktická část práce demonstruje implementaci vybraných modelů výrobního plánování a dopravy v systému AMPL. Pro jednotlivé modely naformulované v teoretické části byly vytvořeny příslušné soubory v systému AMPL. Byla využita pro uživatele příjemná možnost vstupu dat i výstupu výsledku do MS Excel. V příloze práce lze nalézt detailní zápis všech modelů v programu AMPL.

Cílem práce bylo ukázat možnost využití systému AMPL, který je na rozdíl od systému MPL for Windows nabízený v plné verzi, tzn. i pro řešení rozsáhlých úloh. I z tohoto důvodu tato práce představuje cenný příspěvek v oblasti modelování a optimalizace logistických procesů. Práce otevírá cestu k dalšímu výzkumu a implementaci pokročilých optimalizačních technik v praxi.

## Seznam literatury

AMPL. *A Modeling Language for Mathematical Programming* [Online]. 2003 [cit. 2024-04-24]. Dostupné z: <https://ampl.com>

ANTOY, Sergio; HANUS, Michael. Declarative programming with function patterns. In: *International Symposium on Logic-Based Program Synthesis and Transformation*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005. p. 6-22.

APPLEGATE, David L. *The traveling salesman problem: a computational study*. Princeton university press, 2006.

BARNHART, Cynthia a Gilbert LAPORTE. *Handbooks in Operations Research and Management Science*. 14th. North Holland, 2006. ISBN 0444513469.

BERTSEKAS, Dimitri. *Abstract dynamic programming*. Athena Scientific, 2022.

BERTSEKAS, Dimitri. *Dynamic programming and optimal control: Volume I*. Athena scientific, 2012.

BOUCHERIE, Richard J.; BRAAKSMA, Aleida; TIJMS, Henk. *Operations Research: Introduction to Models and Methods*. 2022.

COOK, William J., et al. *The traveling salesman problem: a computational study*. Princeton university press, 2011.

DYCKHOFF, Harald; FINKE, Ute. *Cutting and packing in production and distribution: A typology and bibliography*. Springer Science & Business Media, 1992.

FÁBRY, Jan. *Operační výzkum II: Prezentace* [Online]. Mladá Boleslav, 2024 [cit. 2024-04-24]. Dostupné z: <https://www.janfabry.cz/OV2-prezentace.pdf>

FÁBRY, Jan. *Operační výzkum pro prezenční a kombinovanou formu studia*. 1. vyd. ŠAVŠ o.p.s., 2019. 164 s. ISBN 978-80-87042-84-7.

FÁBRY, Jan. Okružní a rozvozní úlohy. *Habilitační práce. VŠE-FIS, Praha*, 2014.

FOURER, Robert; GAY, David M.; KERNIGHAN, Brian W. *AMPL: A mathematical programming language*. Murray Hill, NJ: AT & T Bell Laboratories, 1987.

GENDREAU, Michel; LAPORTE, Gilbert; SÉGUIN, René. Stochastic vehicle routing. *European journal of operational research*, 1996, 88.1: 3-12.

GENDREAU, Michel; HERTZ, Alain; LAPORTE, Gilbert. A tabu search heuristic for the vehicle routing problem. *Management science*, 1994, 40.10: 1276-1290.

GLOVER, Fred W.; KOCHENBERGER, Gary A. (ed.). *Handbook of metaheuristics*. Springer Science & Business Media, 2006.

GRONER, Rudolf; GRONER, Marina; BISCHOF, Walter F. *Methods of heuristics*. Routledge, 2014.

GUTIN, Gregory; PUNNEN, Abraham P. (ed.). *The traveling salesman problem and its variations*. Springer Science & Business Media, 2006.

CHEN, Y. X. Integrated optimization model for production planning and scheduling with logistics constraints. *International journal of simulation modelling*, 2016, 15.4: 711-720.

JANÁČEK, Jaroslav. *Matematické programování*. Žilinská univerzita, 2003.

LAPORTE, Gilbert. The vehicle routing problem: An overview of exact and approximate algorithms. *European journal of operational research*, 1992, 59.3: 345-358.

MERTL, Jakub. Rozvozní úloha s dělenou dodávkou – heuristické metody [online]. Praha, 2007 [cit. 2024-03-03]. Dostupné z: <https://theses.cz/id/c3b6do/>

MPL. *Mathematical Programming Language* [Online]. 2002 [cit. 2024-04-24]. Dostupné z: <https://www.maximalsoftware.com/mpl>

PAPADIMITRIOU, Christos H.; STEIGLITZ, Kenneth. *Combinatorial optimization: algorithms and complexity*. Courier Corporation, 1998.

PAVLOV, Alexander A. Models and algorithms of multipurpose linear programming. *Journal of Automation and Information Sciences*, 2020, 52.11.

SOBOTTKA, Thomas, et al. A case study for simulation and optimization based planning of production and logistics systems. In: *2017 Winter Simulation Conference (WSC)*. IEEE, 2017. p. 3495-3506.

TAHA, Hamdy A. *Operations research an introduction*. 2007.

VAJDA, Steven. *Mathematical programming*. Courier Corporation, 2009.

VANDERBEI, Robert J. Linear programming: foundations and extensions. *Journal of the Operational Research Society*, 1998, 49.1: 94-94.

YEVGEN, Ievseiev. *Optimalizace výroby pomocí modelu lineárního programování*. 2019. Bachelor's Thesis. České vysoké učení technické v Praze. Výpočetní a informační centrum.

## Seznam obrázků a tabulek

### Seznam obrázků

Obr. 1 Model řezné úlohy v AMPL.....	28
Obr. 2 Datový soubor v AMPL.....	31
Obr. 3 Zdroje .....	31
Obr. 4 Řezná schémata .....	31
Obr. 5 Délka a počet prken k dispozici .....	32
Obr. 6 Spotřeba vrutů, času a ceny jednotlivých výrobků.....	32
Obr. 7 Jednotlivé délky prken potřebných pro výrobu obou výrobků .....	32
Obr. 8 Potřebná prkna 30 cm a 25 cm pro jednotlivé výrobky .....	32
Obr. 9 Hodnoty parametru $d$ u proměnných týkajících se původních prken .....	32
Obr. 10 Hodnoty parametru a vyplývající z tabulky řezných schémat .....	32
Obr. 11 Spuštění optimalizace v AMPL.....	33
Obr. 12 Výsledek v AMPL .....	33
Obr. 13 Výsledek v MS Excel.....	33
Obr. 14 Kontejnerový dopravní problém – zápis modelu.....	35
Obr. 15 Kontejnerový dopravní problém – definice dat .....	36
Obr. 16 Kontejnerový dopravní problém – soubor pro spuštění optimalizace .....	37
Obr. 17 Vstupní data v MS Excel.....	37
Obr. 18 Výstup v MS Excel.....	38
Obr. 19 Výstup v AMPL.....	38
Obr. 20 Rozvozní úloha – model v AMPL .....	40
Obr. 21 Data pro rozvozní úlohu .....	40
Obr. 22 Jednotlivé obce v MS Excel.....	41
Obr. 23 Kapacita vozidla .....	41
Obr. 24 Vzdálenosti mezi jednotlivými obcemi .....	41



Obr. 25 Spuštění optimalizace rozvozní úlohy .....	41
Obr. 26 Výstup výsledků v AMPL.....	42
Obr. 27 Výstup výsledků v MS Excel.....	42

### **Seznam tabulek**

Tab. 1 Porovnání MPL a AMPL.....	24
Tab. 2 Spotřeba materiálu a časové nároky na výrobu .....	26
Tab. 3 Tabulka řezných schémat .....	26
Tab. 4 Náklady na přepravu, kapacity dodavatelů a požadavky odběratelů.....	34
Tab. 5 Požadavky s délkou trasy.....	39

## Seznam příloh

Příloha 1 Řezná úloha řešení v AMPL .....	51
Příloha 2 Kontejnerový dopravní problém řešení v AMPL .....	54
Příloha 3 Rozvozní úloha řešení v AMPL .....	56

## Příloha 1 Řezná úloha řešení v AMPL

cut.mod

SET

set PRKNA;

set DIL;

set SCHEMA;

set VYROBEK;

PARAMETERS

param VrutyDisp  $\geq 0$ ;

param CasDisp  $\geq 0$ ;

param Pocet {PRKNA}  $\geq 0$ ;

param Odpad {SCHEMA}  $\geq 0$ ;

param Vrut {VYROBEK}  $\geq 0$ ;

param Cas {VYROBEK}  $\geq 0$ ;

param c {VYROBEK}  $\geq 0$ ;

param a {DIL,SCHEMA}  $\geq 0$ ;

param Pozadavek {DIL,VYROBEK}  $\geq 0$ ;

param d {PRKNA,SCHEMA}  $\geq 0$ ;

VARIABLES

var x {SCHEMA} integer  $\geq 0$ ;

var y {VYROBEK} integer  $\geq 0$ ;

MAXIMIZE

maximize Celkem:

sum {j in VYROBEK} c[j]\*y[j];

SUBJECT TO

subject to Omezeni1 {i in DIL}:

sum {j in SCHEMA} a[i,j]\*x[j]  $\geq$  sum {j in VYROBEK} Pozadavek[i,j]\*y[j];

subject to Omezeni2:

sum {j in VYROBEK} Vrut[j]\*y[j]  $\leq$  VrutyDisp;

subject to Omezeni3:

sum {j in VYROBEK} Cas[j]\*y[j]  $\leq$  CasDisp;

```
subject to Omezeni4 {i in PRKNA}:  
  sum {j in SCHEMA} d[i,j]*x[j] <= Pocet[i];
```

**cut.dat**

**data;**

**table** zdroje **IN** "amplxl" "cut.xlsx":

[], CasDisp, VrutyDisp;

**table** reznaschemata **IN** "amplxl" "cut.xlsx":

SCHEMA <- [schemata], Odpad;

**table** prkna **IN** "amplxl" "cut.xlsx":

PRKNA <- [prkna], Pocet;

**table** vyrobky **IN** "amplxl" "cut.xlsx":

VYROBEK <- [vyrobky], Vrut, Cas, c;

**table** dily **IN** "amplxl" "cut.xlsx":

DIL <- [dily];

**table** paramPozadavek **IN** "amplxl" "2D" "cut.xlsx":

[DIL, VYROBEK], Pozadavek;

**table** parama **IN** "amplxl" "2D" "cut.xlsx":

[DIL, SCHEMA], a;

**table** paramd **IN** "amplxl" "2D" "cut.xlsx":

[PRKNA, SCHEMA], d;

**read table** zdroje;

**read table** reznaschemata;

**read table** prkna;

**read table** vyrobky;

**read table** dily;

**read table** paramPozadavek;

**read table** parama;

**read table** paramd;

**cut.run**

**reset;**

**model** cut.mod;

**load** amplxl.dll;

**include** cut.dat;

**option** solver gurobi;

**solve;**

**display** x, y;

**display** Celkem;

**table** vystup **OUT** "amplxl" "cut.xlsx":

[schema], x, y;

**table** vysledek **OUT** "amplxl" "cut.xlsx":

[], Celkem;

**write table** vystup;

**write table** vysledek;

## Příloha 2 Kontejnerový dopravní problém řešení v AMPL

### kontejner.mod

*SET*

**set** DODAVATEL;

**set** ODBERATEL;

*PARAMETERS*

**param** Naklady {DODAVATEL, ODBERATEL}  $\geq 0$ ;

**param** Kapacita {DODAVATEL}  $\geq 0$ ;

**param** Pozadavek {ODBERATEL}  $\geq 0$ ;

**param** K  $\geq 0$ ;

*VARIABLES*

**var** x {DODAVATEL, ODBERATEL}  $\geq 0$ ;

**var** y {DODAVATEL, ODBERATEL} **integer**  $\geq 0$ ;

*MINIMIZE*

**minimize** Celkem:

**sum** {i **in** DODAVATEL, j **in** ODBERATEL} Naklady [i,j]\*y[i,j];

*SUBJECT TO (Constrains)*

**subject to** Omezeni1 {i **in** DODAVATEL}:

**sum** {j **in** ODBERATEL} x[i,j]  $\leq$  Kapacita[i];

**subject to** Omezeni2 {j **in** ODBERATEL}:

**sum** {i **in** DODAVATEL} x[i,j] = Pozadavek[j];

**subject to** Omezeni3 {i **in** DODAVATEL, j **in** ODBERATEL}:

x[i,j]  $\leq$  K\*y[i,j];

## **kontejner.dat**

**data;**

**table** Kapacita **IN** "amplxl" "kontejner.xlsx":

[], K;

**table** Dodavatel **IN** "amplxl" "kontejner.xlsx":

DODAVATEL <- [Dodavatel], Kapacita;

**table** Odberatel **IN** "amplxl" "kontejner.xlsx":

ODBERATEL <- [Odberatel], Pozadavek;

**table** Naklady **IN** "amplxl" "2D" "kontejner.xlsx":

[DODAVATEL, ODBERATEL], Naklady;

**read table** Kapacita;

**read table** Dodavatel;

**read table** Odberatel;

**read table** Naklady;

## **kontejner.run**

**reset;**

**load** "amplxl.dll";

**option** solver gurobi;

**model** kontejner.mod;

**data** kontejner.dat;

**solve;**

**display** x, y;

**display** Celkem;

**table** vystup **OUT** "amplxl" "kontejner.xlsx":

[DODAVATEL, ODBERATEL], x, y;

**table** vysledek **OUT** "amplxl" "kontejner.xlsx":

[], Celkem;

**write table** vystup;

**write table** vysledek;

### Příloha 3 Rozvozní úloha řešení v AMPL

rozvozni.mod

SET

**set** OBEC **ordered**;

PARAMETERS

**param** Pozadavek {i **in** OBEC}  $\geq 0$ ;

**param** Vzdalenost {i **in** OBEC, j **in** OBEC}  $\geq 0$ ;

**param** Kapacita;

VARIABLES

**var** x {i **in** OBEC, j **in** OBEC} **binary**;

**var** u {i **in** OBEC};

MINIMIZE

**minimize** CelkovaVzdalenost:

**sum**{i **in** OBEC, j **in** OBEC} Vzdalenost[i,j]\*x[i,j];

SUBJECT TO

**subject to** odjezdjednou {i **in** OBEC: i  $\neq$  first(OBEC)}:

**sum**{j **in** OBEC} x[i,j] = 1;

**subject to** prijezdjednou {j **in** OBEC: j  $\neq$  first(OBEC)}:

**sum**{i **in** OBEC} x[i,j] = 1;

**subject to** cykly {i **in** OBEC, j **in** OBEC: j  $\neq$  first(OBEC)}:

$u[i] + \text{Pozadavek}[j] - \text{Kapacita} * (1 - x[i,j]) \leq u[j]$ ;

**subject to** kapacita {i **in** OBEC: i  $\neq$  first(OBEC)}:

$u[i] \leq \text{Kapacita}$ ;

**subject to** prvnimisto {i **in** OBEC: i = first(OBEC)}:

$u[i] = 0$ ;



## rozvozni.dat

**data;**

**table** obec **IN** "amplxl" "rozvozni.xlsx":

OBEC <- [obec], Pozadavek;

**table** capacity **IN** "amplxl" "rozvozni.xlsx":

[], Kapacita;

**table** vzdalenosti **IN** "amplxl" "2D" "rozvozni.xlsx":

[OBEC, OBEC1], Vzdalenost;

**read table** capacity;

**read table** restaurace;

**read table** vzdalenosti;

## rozvozni.run

**reset;**

**model** rozvozni.mod;

**load** amplxl.dll;

**include** rozvozni.dat;

**option** solver **cpex**;

**solve;**

**display** x, u, CelkovaVzdalenost;

**table** uzly **OUT** "amplxl" "2D" "rozvozni.xlsx":

[OBEC, OBEC1], x;

**table** vysledek **OUT** "amplxl" "rozvozni.xlsx":

[], CelkovaVzdalenost;

**table** umelaPromenna **OUT** "amplxl" "rozvozni.xlsx":

[OBEC], u;

**write table** uzly;

**write table** vysledek;

**write table** umelaPromenna;

## ANOTAČNÍ ZÁZNAM

<b>AUTOR</b>	Bc. Tereza Jindráková		
<b>STUDIJNÍ PROGRAM/OBOR/SPECIALIZACE</b>	Řízení mezinárodních dodavatelských řetězců		
<b>NÁZEV PRÁCE</b>	Využití optimalizačního systému AMPL při řízení výroby a dopravy		
<b>VEDOUCÍ PRÁCE</b>	doc. Ing. Jan Fábry, Ph.D.		
<b>KATEDRA</b>	KRVLK – Katedra řízení výroby, logistiky a kvality	<b>ROK ODEVZDÁNÍ</b>	2024
<b>POČET STRAN</b>	59		
<b>POČET OBRÁZKŮ</b>	27		
<b>POČET TABULEK</b>	5		
<b>POČET PŘÍLOH</b>	3		
<b>STRUČNÝ POPIS</b>	<p>Ve Škoda Auto Vysoké škole byla pro řešení úloh matematického programování doposud používána plná verze systému MPL for Windows. Vzhledem k tomu, že je systém dostupný již pouze v omezené licenci, program AMPL je vhodnou alternativou nejen pro výuku, ale především pro zpracování reálných úloh v rámci závěrečných prací.</p> <p>Cílem této práce je popsat systém AMPL jako podpory modelování a optimalizace, včetně jeho praktického využití. K tomuto účelu byly naformulovány matematické modely vybraných úloh z oblasti výroby a řízení dopravy.</p> <p>Pro jednotlivé modely naformulované v teoretické části byly vytvořeny příslušné soubory v systému AMPL. Byla využita pro uživatele příjemná možnost vstupu dat i výstupu výsledku do MS Excel. V příloze práce lze nalézt detailní zápis všech modelů v programu AMPL.</p>		
<b>KLÍČOVÁ SLOVA</b>	Optimalizace, lineární programování, řešitelé, systém AMPL, systém MPL, řízení dopravy, matematické modely, úlohy výrobního plánování		

## ANNOTATION

<b>AUTHOR</b>	<b>Bc. Tereza Jindráková</b>		
<b>FIELD</b>	<b>International Supply Chain Management</b>		
<b>THESIS TITLE</b>	<b>Use of the AMPL optimization system in production and transport management</b>		
<b>SUPERVISOR</b>	<b>doc. Ing. Jan Fábry, Ph.D.</b>		
<b>DEPARTMENT</b>	<b>KRVLK – Department of Production, Logistics and Quality Management</b>	<b>YEAR</b>	<b>2024</b>
<b>NUMBER OF PAGES</b>	<b>59</b>		
<b>NUMBER OF PICTURES</b>	<b>27</b>		
<b>NUMBER OF TABLES</b>	<b>5</b>		
<b>NUMBER OF APPENDICES</b>	<b>3</b>		
<b>SUMMARY</b>	<p>At Škoda Auto University, the full version of the MPL for Windows system has been used to solve mathematical programming tasks. Given that the system is only available as a limited license, the AMPL program is a suitable alternative not only for teaching, but especially for processing real tasks as part of final theses.</p> <p>The aim of this work is to describe the AMPL system as a support for modeling and optimization, including its practical use. For this purpose, mathematical models of selected tasks from the field of production and transport management were formulated.</p> <p>For the individual models formulated in the theoretical part, appropriate files were created in the AMPL system. The user-friendly possibility of entering data and outputting results to MS Excel was used. A detailed description of all models in the AMPL program can be found in the appendix of the thesis.</p>		
<b>KEY WORDS</b>	<b>Optimization, linear programming, solvers, AMPL system, MPL system, logistics control, mathematical models, production planning problems</b>		