VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA STROJNÍHO INŽENÝRSTVÍ
FACULTY OF MECHANICAL ENGINEERING

ÚSTAV MATEMATIKY
INSTITUTE OF MATHEMATICS

DOPRAVNÍ MODELY A JEJICH APLIKACE
LOGISTIC PROBLEMS AND THEIR APPLICATIONS

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE Petr Žlebek
AUTHOR

VEDOUCÍ PRÁCE Ing. Martin Pavlas, Ph.D.
SUPERVISOR

BRNO 2018

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ FAKULTA STROJNÍHO INŽENÝRSTVÍ

Zadání bakalářské práce

Ústav:	Ústav matematiky
Student:	**Petr Žlebek**
Studijní program:	Aplikované vědy v inženýrství
Studijní obor:	Matematické inženýrství
Vedoucí práce:	**Ing. Martin Pavlas, Ph.D.**
Akademický rok:	2017/18

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 bakalářské práce:

Dopravní modely a jejich aplikace

Stručná charakteristika problematiky úkolu:

V rámci svozových úloh, lokační a alokačních problému hraje často významnou roli přeprava. V mnoha případech je cena uvažována jako konstanta bez ohledu na přepravované množství. Řešení projektu Centrum kompetence pro energetické využití odpadů (WtECC, řešitel: Ústav procesního inženýrství VUT v Brně) ukázalo, že jednoduchý model dostatečně nepopisuje realitu. Navíc, málo kdy jsou do úlohy zahrnuty dopravní bariéry na hranách (omezení spojené s reálným stavem dopravní infrastruktury – nosnost mostů, výška tunelů atd. Bakalářská práce se bude zabývat analýzou významu vybraných aspektů dopravních úloh. Nosným bodem bude vývoj matematických modelů a jejich implementace do výpočtových nástrojů. Modely budou aplikovány na reálná data v rámci ČR.

Cíle bakalářské práce:

1. Seznámení se z problematikou nákladní dopravy.
2. Osvojení si matematického programování v dopravních úlohách.
3. Vytvoření vlastních optimalizačních modelů v předmětné oblasti.
4. Prezentace výsledků práce a jejich diskuze.

Seznam doporučené literatury:

WILLIAMS, H. P. Model building in mathematical programming. 5th ed. Hoboken, N.J.: Wiley, 2013. ISBN 978-1-118-44333-0.

PATRIKSSON, M. The traffic assignment problem: models and methods. New York: Dover Publications, 2015. ISBN 978-048-6787-909.

UDDIN, MD. F. A Network Model: Minimum Cost Network Flow Problem (MCNFP) Mathematical Analysis of Minimum Cost Network Flow Problem. Neue Ausg. Saarbrücken: LAP LAMBERT Academic Publishing, 2012. ISBN 9783659104077.

GHIANI, G., G. LAPORTE and R. MUSMANNO. Introduction to logistics systems management. 2nd edition. ISBN 978-1-119-94338-9.

GREGOR, J., M. PAVLAS and R. ŠOMPLÁK. Transportation Cost as an Integral Part of Supply Chain Optimization in the Field of Waste Management. In Proceedings of the 19th International Conference on Process Integration, Modelling and Optimisation for Energy Saving and Pollution Reduction (PRES 2016). CHEMICAL ENGINEERING TRANSACTIONS, 2016. ISBN: 978-88-95608-47- 1.

NASH, S. et al. Linear and nonlinear programming. McGraw-Hill, 1995.

Termín odevzdání bakalářské práce je stanoven časovým plánem akademického roku 2017/18

V Brně, dne

L. S.

prof. RNDr. Josef Šlapal, CSc.
ředitel ústavu

doc. Ing. Jaroslav Katolický, Ph.D.
děkan fakulty

Abstrakt
Práce se zabývá využitím optimalizace v oblasti dopravních modelů. V první části práce jsou shrnuty teoretické poznatky z oblasti teorie grafů, matematického programování, dopravních úloh a genetického algoritmu. Tyto informace jsou následně aplikovány ve druhé části práce. V té je důkladně popsán dopravní model, který se zabývá svozem odpadu. K implementaci modelu bylo využito prostředí Matlab. Na vytvořený výpočtový aparát jsou pak aplikována reálná data pro území České republiky. V závěrečné části práce jsou shrnuty a okomentovány výsledky modelu.

Summary
The thesis deals with transportation problems using optimization methods. The first part summarizes theoretical facts about the graph theory, mathematical programming, transportation problems and genetic algorithm. Those informations are used in the second part of the thesis where the specific transportation model dealing with waste collection management is described. The model is implemented in Matlab environment. Real data from the area of Czech republic are then applied to the model. Finally the results are throughly discussed.

Klíčová slova
Optimalizace, genetický algoritmus, dopravní úloha, teorie grafů

Keywords
Optimization, genetic algorithm, transportation problem, graph theory

ŽLEBEK, P.*Dopravní modely a jejich aplikace*. Brno: Vysoké učení technické v Brně, Fakulta strojního inženýrství, 2018. 35 s. Vedoucí Ing. Martin Pavlas, Ph.D.

Prohlášení

Prohlašuji, že jsem práci napsal s pomocí pouze citovaných zdrojů zmíněných v seznamu literatury a sebe samého.

V Novém Jičíně, dne 22. 5. 2018.

Petr Žlebek

Poděkování
Chtěl bych tímto poděkovat vedoucímu mé bakalářské práce panu Ing. Martinu Pavlasovi, Ph.D. a panu Ing. Radovanu Šomplákovi, Ph.D. za cenné rady, které mi byly při psaní poskytnuty.

Děkuji také své rodině a blízkým za podporu během dlouhých třech let studia.

Petr Žlebek

Obsah

1. Úvod a motivace

1.1. Úvod

Tato práce nese název *Dopravní modely a jejich aplikace*. Jejím cílem je seznámit se s problematikou matematického programování v oblasti dopravních úloh a na základě těchto dovedností vytvořit vlastní optimalizační model.

Úvodní část se věnuje teoretickým poznatkům z oblastí teorie grafů a matematického programování. Jsou zde nastíněny také problémy dopravních úloh a v závěrečné podkapitole je popsána analogie biologického přirozeného výběru s hledáním vhodného řešení matematické úlohy.

Další část práce je věnována popisu dopravního modelu. Ten se zabývá svozem odpadu po hranách mezi uzly tak, aby to bylo co nejekonomičtější. Model je nejprve pro snadné pochopení vysvětlen na malých a konkrétních datech. Následuje pasáž, ve které je popsán matematicky s využitím teoretických poznatků z první části práce. V poslední sekci praktické části práce je výpočtový model aplikován na reálný problém. Řešení je pak prezentováno nejen číselně, ale i o dost názornější mapou.

V závěrečné diskusi jsou shrnuty dosažené cíle a je zde také zhodnoceno řešení konkrétní úlohy.

1.2. Motivace

Česká republika, coby členská země Evropské unie, je zavázána k tomu, že bude v nejbližších letech na skládkách ukládat čím dál tím méně komunálního odpadu. [11]

Odpadové hospodářství v naší zemi je sice na dobré cestě, ale vždy je zde prostor ke zlepšení. Čím méně odpadu bude uloženo na skládkách, tím „zelenější“ bude budoucnost naší planety.

Vhodnou alternativou ke skládkování může být například zpracování odpadu ve spalovnách. Spalovna je zařízení, ve kterém je odpad přeměněn na tepelnou a elektrickou energii.

Je nutno podotknout, že vybudování a následný provoz takového zařízení je obrovskou investicí, a proto je třeba vzít v potaz všechny faktory ovlivňující její provoz. A právě tímto problémem se zabývá model, který je praktickou částí této práce.

2. Teoretická část

2.1. Teorie grafů

Zdrojem teoretických poznatků popsaných v následujících dvou podkapitolách jsou [6] a [3]. Zmíněné definice a popisy jsou důležité pro další pochopení problematiky dopravních úloh.

2.1.1. Základní definice

Definice 1 (Orientovaného grafu)**.** *Orientovaný graf* je trojice $G = (V, E, \epsilon)$, kde V je neprázdná konečná množina, jejíž prvky nazýváme *vrcholy*, E je konečná množina, jejíž prvky nazýváme *orientovanými hranami* a $\epsilon : E \to V^2$ je zobrazení, které nazýváme *vztahem incidence*. Množina hran E může být prázdná.

Definice 2 (Bipartitního grafu)**.** Graf $G = (V, E, \epsilon)$ je *bipartitní*, pokud platí $V = V_1 \cup V_2, V_1 \cap V_2 = \emptyset$ a $\forall e = \{u, v\}, e \in E : u \in V_1 \wedge v \in V_2$. Platí-li navíc $E = V_1 \times V_2$ (tedy v grafu existují všechny hrany s touto vlastností), pak se graf nazývá *úplný bipartitní*.

Definice 3 (Počátečního a koncového vrcholu hrany)**.** Zobrazení $\epsilon : E \to V^2$ přiřazuje každé hraně $e \in E$ uspořádanou dvojici vrcholů (x, y), kde $x, y \in V$. První z nich, x, nazýváme *počátečním vrcholem hrany* a značíme jej $Pv(e)$. Druhý z nich, y, nazýváme *koncovým vrcholem hrany* a značíme jej $Kv(e)$.

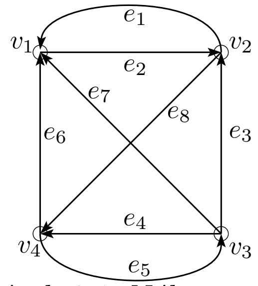
Definice 4 (Smyčky)**.** Jestliže $Pv(e) = Kv(e)$, pak hranu e nazýváme (orientovanou) *smyčkou*.

Definice 5 (Orientovaného sledu)**.** Posloupnost vrcholů a hran $v_0, e_1, v_1, e_2, v_2, \ldots, e_k, v_k$ nazýváme *orientovaným sledem*, jestliže pro každou hranu e_i z této posloupnosti platí $Pv(e_i) = v_{i-1}$ a $Kv(e_i) = v_i$.

2.1.2. Druhy popisu grafu

Nákres

Nákres je jednoduchým popisem grafu, který se však pro rozsáhlejší úlohy stává nepraktickým. Vhodný je zejména ke snadné ilustraci řešeného problému (viz obrázek 2.1).



Obrázek 2.1: Nákres grafu.

Seznamy vrcholů a hran

Množina vrcholů je popsána výčtem prvků, množina hran je popsána seznamem trojic tvořených jménem hrany a jejím počátečním a koncovým vrcholem. V podstatě se jedná o úplný popis grafu podle definice. Pokud nám nezáleží na jménech hran, můžeme je vypustit a hrany popisovat pouze dvojicí vrcholů.

Příklad: Graf z obrázku 2.1 lze seznamem popsat takto:

$$\begin{array}{ll} \text{Vrcholy:} & v_1, v_2, v_3, v_4. \\ \text{Hrany:} & (e_1, v_2, v_1), (e_1, v_1, v_2), \\ & (e_3, v_3, v_2), (e_4, v_3, v_4), \\ & (e_5, v_4, v_3), (e_6, v_4, v_1), \\ & (e_7, v_3, v_1), (e_8, v_2, v_4). \end{array}$$

Matice incidence

Definice 6 (Matice incidence)**.** Nechť G je orientovaný graf bez smyček. Zvolíme-li (libovolně, ale pevně) nejen pořadí vrcholů $v_1, \ldots, v_n \in V$, ale také pořadí hran $e_1, \ldots, e_m \in E$, pak můžeme grafu G přiřadit *matici incidence* A_G typu (m, n) pro kterou platí

$$a_{i,j} = \begin{cases} 1 & \text{jestliže } v_i \text{ je počátečním vrcholem hrany } e_j, \\ -1 & \text{jestliže } v_i \text{ je koncovým vrcholem hrany } e_j, \\ 0 & \text{v ostatních případech.} \end{cases} \tag{2.1}$$

Příklad: Grafu z obrázku 2.1 odpovídá následující matice incidence:

$$A_G = \begin{pmatrix} -1 & 1 & 0 & 0 & 0 & -1 & -1 & 0 \\ 1 & -1 & -1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & -1 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 & 1 & 1 & 0 & -1 \end{pmatrix}$$

Využití matice incidence je matematicky velice elegantní řešení popisu orientovaného grafu. Je ale také velmi neúsporné, protože se v každém sloupci nacházejí pouze dva nenulové prvky. I přes to je právě matice incidence k popisu grafu v dopravních úlohách využívána nejčastěji.

2.1.3. Graf dopravní sítě

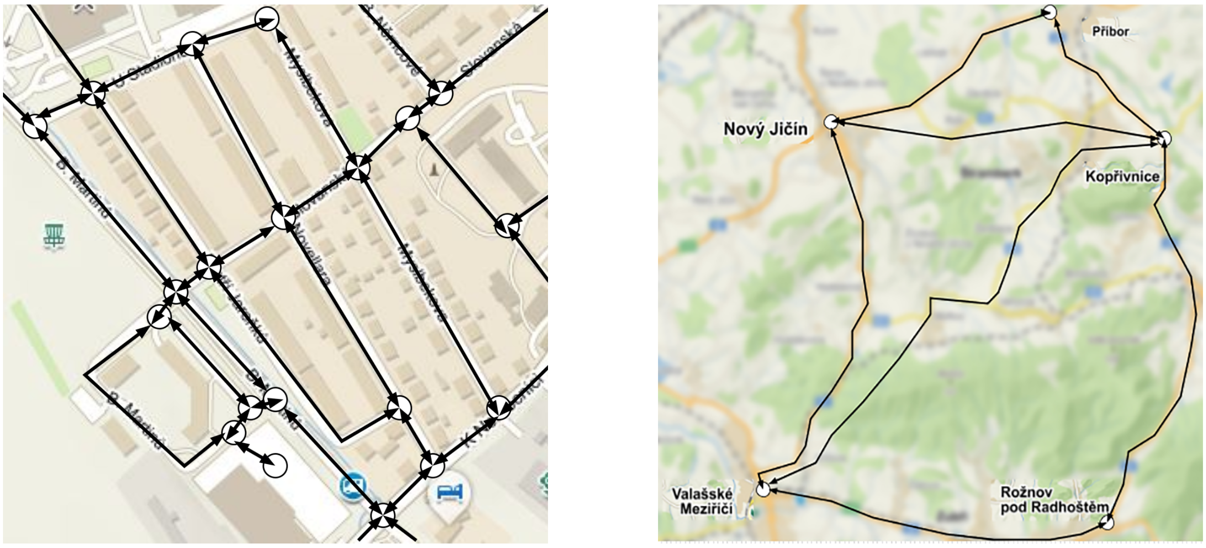
Teorie grafů se dá aplikovat také při řešení dopravních úloh. Grafy dopravních sítí mohou být různě detailní.

Na úrovni křižovatek

Graf dopravní sítě na obrázku 2.2(a) je takový graf, ve kterém množinu vrcholů V tvoří všechny křižovatky ve vybrané oblasti a množinu hran E tvoří všechny cesty mezi sousedními křižovatkami. Tento typ grafu se využívá například při řešení problému svozu popelnic.

Na úrovni obcí

Graf dopravní sítě na obrázku 2.2(b) je graf, ve kterém množinu vrcholů V tvoří všechny obce ve vybrané oblasti a množinu hran E tvoří všechny cesty spojující sousední obce.



(a) Křižovatky. (b) Obce.

Obrázek 2.2: Srovnání výše zmíněných grafů dopravní sítě.

2.2. Matematické programování

2.2.1. Obecné informace

Matematické programování (nebo také optimalizace) má v oblastech technické praxe nepřeberné využití. Krom dopravních úloh, kterým je věnována celá následující podkapitola, se optimalizace využívá také v ekonomice či při výrobních procesech. V následujících podkapitolách budou shrnuty základní poznatky potřebné pro další pochopení problematiky dopravních úloh. Prameny těchto informací jsou [8] a [1].

Definice 7 (Úlohy matematického programování)**.** Nechť $S \subset \mathbb{R}^n$ a $f : \mathbb{R}^n \to \mathbb{R}$. Pak řekneme, že *úloha matematického programování* má následující tvar:

$$\min_{x}\{f(\mathbf{x}), \mathbf{x} \subset S\}. \tag{2.2}$$

Poznámka. Funkci f nazýváme *účelovou funkcí* a vektor $\mathbf{x}$ je vektor *rozhodovacích proměnných*.

Poznámka. V souvislosti s matematickým programováním často zaznívá slovo *model*. *Modelem* nazýváme zjednodušený popis reálného problému. Snahou je model konstruovat tak, aby se co nejvíce podobal realitě a aby byl snadněji řešitelný než původní problém.

2.2.2. Lineární programování

Definice 8 (Úlohy lineárního programování)**.** Řekneme, že úloha matematického programování je *lineární*, jestliže jsou její účelová funkce a omezující podmínky lineární. Úloha je tedy tvaru:

$$\begin{aligned} \min \ & \mathbf{c}^T\mathbf{x} \\ \text{s. t. } & \mathbf{A}\mathbf{x} \leq \mathbf{b} \\ & \mathbf{x} \geq \mathbf{0}, \end{aligned} \tag{2.3}$$

kde vektor $\mathbf{c}^T \in \mathbb{R}^n$ je vektor *vah (cen)*. Vektor $\mathbf{x} \in \mathbb{R}^n$ je vektor *rozhodovacích proměnných*. Matice $\mathbf{A}$ je typu $m \times n$. Vektor $\mathbf{b} \in \mathbb{R}^m$ je *vektor poptávek a kapacit*. Skalární součin $\mathbf{c}^T\mathbf{x}$ je minimalizovaná účelová funkce f.

Metoda řešení úloh lineárního programování - simplexová metoda

K řešení úloh lineárního programování se nejčastěji využívá tzv. *simplexová metoda*, kterou v roce 1947 světu odhalil americký matematik George Dantzig. [2]

Simplexová metoda vychází z principu existence optimálního řešení lineární úlohy matematického programování, který říká, že jestliže množina přípustných řešení úlohy má globální extrém, pak existuje optimální řešení. K důkazu tohoto tvrzení a následnému vytvoření algoritmu simplexové metody se využívají poznatky z lineární algebry, konvexní analýzy a teorie polyedrických množin.

Simplexová metoda byla popsána již v mnoha pracích zabývajících se lineárním programováním. Mou příručkou k porozumění této metodě byl [7].

2.2.3. Celočíselné lineární programování

Definice 9 (Úlohy celočíselného programování)**.** Řekneme, že úloha matematického programování je *celočíselná*, jestliže alespoň jedna rozhodovací proměnná nabírá pouze celočíselných hodnot.

Poznámka. Speciální případem celočíselné rozhodovací proměnné je binární a nabírá pouze hodnot 0 a 1.

Typickou úlohou celočíselného lineárního programování je tzv. *Travelling Salesman Problem (Problém obchodního cestujícího)*, který je podrobně popsán například zde [1].

2.3. Dopravní úloha

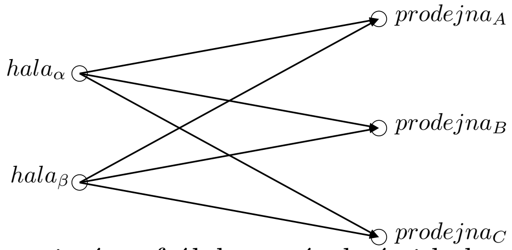
2.3.1. Tok sítí s minimalizací nákladů

Poprvé tento typ úlohy formuloval Frank L. Hitchcock v roce 1941 ve svém článku [5].

Princip se dá snadno vysvětlit na následujícím konkrétním problému:

Fiktivní firma *Cukroušek s.r.o.* vyrábí moučkový cukr. Firma vlastní dvě výrobní haly ($hala_\alpha, hala_\beta$). Moučkový cukr v nich vyprodukovaný je dodáván do třech různých prodejen ($prodejna_A, prodejna_B, prodejna_C$) (viz obrázek 2.3). Haly mají různé týdenní výrobní kapacity a v prodejnách je po moučkovém cukru různá týdenní poptávka. Vyrobený cukr se z hal do prodejen převáží jednou za týden. Vzdálenosti mezi jednotlivými halami a prodejnami jsou různé. Cena přepravy roste přímo úměrně s ujetou vzdáleností.

Úkolem pro vedení firmy je určit, kolik kilogramů moučkového cukru je nutno přepravit z jednotlivých hal do konkrétních prodejen tak, aby nebyly překročeny týdenní výrobní kapacity všech hal, aby byly uspokojeny týdenní poptávky všech prodejen a aby firma na přepravě co nejvíce ušetřila.



Obrázek 2.3: Bipartitní graf úlohy s výrobními halami a prodejnami.

Tato úloha by se dala rozšířit na rozsáhlejší problém s velkým počtem podmínek.

2.3.2. Standardní tvar

Standardní tvar dopravní úlohy vychází ze vztahů (2.2) a (2.3) uvedených v předchozí podkapitole. Pro názornost je vhodné přeznačit množinu vrcholů grafu V a množinu hran grafu E na tradičnější označení indexových množin. Množinu všech vrcholů (např. měst, obcí či obcí s rozšířenou působností) označme I a množinu všech hran (cest mezi jednotlivými vrcholy) označme J.

$$\begin{aligned} I &\equiv V, \\ J &\equiv E. \end{aligned} \tag{2.4}$$

V základním tvaru dopravní úlohy se vyskytuje několik symbolů, jejichž význam je uveden v tabulce 2.1. Úloha je formulována takto:

$$\begin{aligned} \min \quad & \sum_{j \in J} c_j x_j \\ \text{s. t.} \quad & x_j \geq 0, \qquad \forall j \in J, \\ & \sum_{j \in J} a_{i,j} x_j \leq b_i, \quad \forall i \in I. \end{aligned} \tag{2.5}$$

Tabulka 2.1

označení	typ	význam
c_j	vstupní parametr	ohodnocení hrany j (cena či vzdálenost)
x_j	rozhodovací proměnná	množství komodity přepravené po hraně j
$a_{i,j}$	vstupní parametr	incidence mezi vrcholem i a hranou j
b_i	vstupní parametr	kapacitní omezení vrcholu i

2.4. Genetický algoritmus

2.4.1. Základní informace

Genetický algoritmus patří mezi tzv. *heuristické postupy.* Heuristika se používá v případě, kdy tradiční (exaktní) způsoby řešení selhávají, nebo jsou časově příliš náročné. Heuristické algoritmy jsou do značné míry náhodné, a proto často naleznou pouze přibližné řešení.

Genetický algoritmus je jeden z tzv. *evolučních algoritmů*, které při hledání správného řešení využívají principy evoluční biologie [4]. V následující podkapitole bude na konkrétním smyšleném biologickém problému popsán tzv. *binární genetický algoritmus.* Tento postup je pak na základě zřejmé analogie aplikován v praktické části této práce.

2.4.2. Reálná aplikace

Mějme populaci fiktivního zvířecího druhu čítající šest jedinců.

$$n = 6. \tag{2.6}$$

Fiktivní zvířecí druh nerozlišuje pohlaví a je schopen rozmnožovat se dvěma způsoby. Prvním způsobem je stejně jako u lidí *heterogamie*, při které nový jedinec vzniká ze dvou rodičů. Druhým způsobem je už hůře představitelná *parentogeneze*, při níž dochází ke vzniku nového jedince pouze z jednoho rodiče. Pro účel popisu binárního genetického algoritmu budeme způsoby rozmnožování nazývat *křížení* a *mutace*.

Řekněme, že zvířecí druh má osm důležitých vlastností, které ovlivňují to, jak dobrým běžcem je daný jedinec. Pro názornost předpokládejme, že jedinec c_i může daný znak v_j buď *„mít“* (značíme $a_{i,j} = 1$), nebo *„nemít“* (značíme $a_{i,j} = 0$). Každý jedinec „má“ právě čtyři vlastnosti.

$$\begin{aligned} |J| &= 8, \\ p_v &= \sum_{j \in J} a_{i,j} = 4, \forall i \in I. \end{aligned} \tag{2.7}$$

Přehled uvažovaných vlastností je v tabulce 2.2.

Tabulka 2.2

označení	vlastnost
v_1	„je silný“
v_2	„je štíhlý“
v_3	„je vysoký“
v_4	„je hbitý“
v_5	„je vytrvalý“
v_6	„je inteligentní“
v_7	„je ohebný“
v_8	„je odhodlaný“

Na základě těchto znaků je ohodnocen číselnou hodnotou, které říkáme *kvalita* a budeme ji značit z. Čím vyšší je kvalita daného jedince, tím lepším je běžcem. Kvalita jedince se spočte podle následujícího vztahu:

$$z = 2v_1 + 4v_2 + 3v_3 + 6v_4 + 5v_5 + 2v_6 + 4v_7 + v_8. \tag{2.8}$$

Náhodně vytvořená populace šesti jedinců, jejich vlastnosti a jim příslušné kvality jsou uvedeny v tabulce 2.3.

Tabulka 2.3

$a_{i,j}$	v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_8	z
c_1	0	1	1	1	0	0	1	0	17
c_2	0	1	1	0	1	0	0	1	13
c_3	0	0	1	1	0	1	1	0	15
c_4	1	0	0	1	0	1	1	0	14
c_5	1	1	1	0	0	0	0	1	10
c_6	1	1	0	0	0	1	0	1	9

Naším úkolem je „vyšlechtit“ jedince s takovou kombinací rysů, které z něj udělají nejlepšího běžce.

2.4.3. Procedury binárního genetického algoritmu

Je zřejmé, že tento ilustrační příklad by se dal poměrně rychle vyřešit porovnáním kvality všech možných kombinací vlastností, ale pro uvedení do problematiky na úlohu aplikujeme binární genetický algoritmus. Pro další pokračování je nutné znát pět procedur, ze kterých je binární genetický algoritmus složený.

Vytvoření populace

Původní populace je vytvořena zcela náhodně. Jediným parametrem populace je její velikost, kterou značíme písmenem n. Pro další využití je populace už při svém vytvoření řazena podle kvality od jedince s nejvyšší kvalitou.

Způsob „zrození“ nové generace je pak závislý na hodnotách dvou vstupních parametrů. Jedná se o pravděpodobnosti s jakými dojde ke vzniku nového jedince křížením, nebo v opačném případě mutací. Součet těchto pravděpodobností je roven 1, protože u fiktivního zvířecího druhu, o kterém pojednává výše uvedený příklad, k jinému typu množení dojít nemůže. Nová generace je stejně velká jako původní populace.

Po tom, co se vytvoří n nových jedinců (tedy celá nová generace) jsou porovnáni všichni z nové i původní populace a je z nich vybráno pouze n unikátních a nejkvalitnějších. Z nich je vyhodnocen doposud nejkvalitnější nalezený jedinec, kterého považujeme za prozatímní správné řešení. Tato hodnota je důležitá pro ověřování podmínky dostatečné přesnosti. Není-li splněna, vytváří se nová populace.

Poznámka. Kvalita původní populace může mít rozhodující vliv na rychlost i přesnost řešení. Proto existují tzv. *hybridní genetické algoritmy*, ve kterých se původní populace vytváří jiným než zcela náhodným způsobem. To ovšem není případ našeho příkladu.

Ověření dostatečné přesnosti

Tato procedura závisí na potřebách konkrétního problému. Není nutné, aby se v algoritmu vyskytovala a celý proces řešení může skončit po předem daném počtu vytvořených generací.

U tohoto příkladu algoritmus skončí v případě, kdy se doposud nejkvalitnější nalezený jedinec nezmění v k po sobě vytvořených generacích. Parametr k pracovně označíme *stupeň přesnosti řešení* a volíme jej libovolně v závislosti na přesnosti řešení, kterou požadujeme.

Výběr jedince k množení

K množení je vybrán náhodný jedinec s pravděpodobností odpovídající jeho pořadí podle kvality. To znamená, že k množení bude častěji vybrán v pořadí druhý než pátý jedinec. Pořadí daného jedince označme písmenem i. Pravděpodobnost výběru spočteme podle následujícího vztahu:

$$p(c_i) = \frac{n+1-i}{\sum\limits_{i=1}^{n} i} \tag{2.9}$$

Poznámka. Dalším možným způsobem výběru jedince k množení je tzv. *ruleta.* V tomto případě není bráno v potaz pořadí jedinců seřazených podle kvality, ale je uvažována přímo číselnou hodnotu kvality. Po důkladném zvážení byla pro účely úlohy zvolena výše uvedená metoda.

Křížení dvou jedinců

Při křížení dvou různých jedinců vzniká nový jedinec tak, že v situaci, kdy je hodnota dané vlastnosti stejná u obou rodičů, zůstává stejná i u potomka. Nicméně, pokud je hodnota u rodičů rozdílná, pak se náhodně vybere, po kterém z rodičů potomek vlastnost zdědí.

Vzhledem k charakteru uvedené úlohy je počet rozdílů mezi rodiči vždy sudý. Je zbytečné, aby se spolu křížili jedinci, kteří se liší pouze ve dvou rysech, protože jejich spojením vždy vznikne identická kopie jednoho z nich. Tímto předcházíme možnému zacyklení výpočtu. Algoritmus tedy náhodně vybere přesně polovinu z rozdílných znaků rodičů a přidělí jim hodnotu 1. Zbývající polovině přidělí 0.

Mutace jedince

Při mutaci vzniká z jednoho rodiče nový potomek tak, že se náhodně prohodí dvě jeho vlastnosti s rozdílnou hodnotou. To znamená, že potomek „má“ jednu vlastnost, kterou jeho rodič „nemá“ a obráceně.

Celkové schéma

Jednotlivé procedury binárního genetického algoritmu jsou poskládány tak, jak je uvedeno v algoritmu 1.

```
vytvoření původní populace;
seřazení jedinců podle kvality;
while řešení není dostatečně přesné do
    for ( i = 0; i ≤N; i++ ) {
        křížení nebo mutace;
        výpočet kvality nového jedince;
        rozhodnutí o zařazení jedince do populace;
    }
    ověření dostatečné přesnosti;
end
```

Algoritmus 1: Pseudokód binárního genetického algoritmu

2.4.4. Řešení problému

Z rovnice (2.6) pro výpočet kvality jedince je zřejmé, že nejlepší běžec bude mít vlastnosti v_2, v_4, v_5 a v_7 (tedy štíhlost, hbitost, vytrvalost a ohebnost), protože právě u nich stojí nejvyšší koeficienty. Kvalita takového jedince bude rovna 19. Při řešení rozsáhlejších úloh, u kterých bude výpočet kvality daleko složitější, to už samozřejmě tak snadno nepůjde.

Jedná se pouze o ilustrativní příklad, a tak zde nebude uveden zdrojový kód řešení této úlohy. Ten bude rozebrán v praktické části práce na dopravní úloze. Je však nasnadě zde poukázat na to, že využitím binárního genetického algoritmu tak, jak byl popsán výše, byl oproti úplné enumeraci opravdu urychlen výpočet správného řešení.

Celkový počet různých jedinců N, kteří ze zadaných vlastností mohou vzniknout se vypočítá následujícím kombinatorickým vztahem. Vycházíme ze vztahu (2.7).

$$N = \binom{|J|}{p_v} = \binom{8}{4} = 70. \tag{2.10}$$

Parametr k označující stupeň přesnosti řešení byl zvolen $k = 3$.

Tabulka 2.4 ukazuje, kolik bylo u příslušného pokusu nutno vytvořit generací (a tedy kolikrát proběhl výpočet kvality) k ukončení algoritmu.

Tabulka 2.4

číslo pokusu	počet generací	počet výpočtů
1	8	48
2	7	42
3	8	48
4	6	36
5	7	42

Z tabulky je patrné, že počet výpočtů při využití binárního genetického algoritmu byl vždy nižší než 70, jak by tomu bylo při úplné enumeraci řešení. Za úspěch se dá považovat také to, že genetický algoritmus vždy našel optimální řešení a je zřejmě správně podmíněný.

3. Dopravní model

Vytvoření dopravního modelu, který se zabývá svozem odpadu, je praktickou částí této práce. Jsou zde aplikovány poznatky z kapitoly 2.

3.1. Definice úlohy

V této kapitole bude úloha vysvětlena nejprve slovně, pak velmi zjednodušeně interpretována na konkrétním příkladě a následně popsána matematicky.

3.1.1. Slovní znění

Územní oblast je dle potřeby rozdělena na určitý počet podoblastí. Každá podoblast má známý *počet obyvatel* a také *průměrnou roční produkci odpadu jedním obyvatelem.* V každé podoblasti je možnost vybudovat zařízení, které umí odpad zpracovat (jinak než uložením na skládku). V každé podoblasti navíc může, ale nemusí, být určeno minimální procentuální množství odpadu, který bude zpracován v jednom z nově vybudovaných zařízení. Tento parametr budeme označovat jako tzv. *parciální požadavek.*

Každé potenciální zařízení má danou *maximální roční kapacitu* (tj. hmotnost odpadu, který v něm může být za jeden rok zpracován) a také *cenu za zpracování jedné tuny odpadu.*

Sousední podoblasti jsou propojeny cestami. Jednotlivé cesty mají různou *délku* a mohou se také lišit *cenou za ujetý kilometr* (např. nutnost jízdy po dálnici či vyšší spotřeba pohonných hmot).

V celé oblasti je pak určeno minimální procentuální množství odpadu, který bude zpracován v některých z potenciálních zařízení. Tento parametr budeme označovat jako *celkový požadavek.* Dále je s přihlédnutím k počtu podoblastí a požadovanému zpracovanému množství odpadu určeno, kolik zařízení je nutno v celé oblasti vybudovat. Tato hodnota je značena jako *počet postavených zařízení.*

Úkolem je nalézt takové rozložení nově postavených zařízení, při kterém jsou minimalizovány náklady na přepravu a zpracování odpadu a zároveň jsou splněny všechny podmínky: v každé podoblasti je vyhověno příslušnému parciálnímu požadavku, v celé oblasti je splněn celkový požadavek a není překročena kapacita žádného z potenciálních zařízení.

3.1.2. Zjednodušená interpretace

V této podkapitole bude problém demonstrován na malých a konkrétních datech fiktivním příkladem, jehož vstupní data jsou smyšlená, nemají žádný reálný význam a slouží pouze ke snadnému pochopení principu modelu.

Mějme tři sousední okresní města Nový Jičín, Vsetín a Přerov. Do každého z nich je svážen veškerý odpad vyprodukovaný na území příslušného okresu. Bylo rozhodnuto, že v právě jednom z těchto měst bude vybudováno zařízení zpracovávající odpad. Požadavkem je, aby v novém zařízení bylo zpracováno alespoň 67% z celkové produkce odpadu na území všech tří okresů dohromady. Parciálním požadavkem okresu Vsetín je zpracovat alespoň 50% a okres Přerov žádá alespoň 15%. Úkolem pro vedení smyšleného projektu je rozhodnout, ve kterém z okresních měst bude nejekonomičtější zařízení postavit.

Parametry úlohy

Předpoklady úlohy jsou následující:

- Kapacita každého potenciálního zařízení bezpečně pokryje celkové minimální požadované množství.
- Cena za použití dané cesty je uvažována oboustranně a skládá se z nákladů na pohonné hmoty, pracovní sílu a dalších faktorů ovlivňujících cenu přepravy. Tímto faktorem může být například jízda po zpoplatněné silnici. V tomto příkladě uvažujeme, že do (resp. z) Přerova se z (resp. do) obou dalších okresních měst dostaneme pouze po dálnici.

Vstupní parametry úlohy jsou uvedeny v tabulkách 3.1 až 3.4.

Tabulka 3.1: Vlastnosti okresních měst a parciální požadavky na zpracované množství.

okresní město	počet obyvatel $[ob.]$	produkce $[t \cdot r^{-1} \cdot ob.^{-1}]$	požadavek $[\%]$	minimální množství $[t \cdot r^{-1}]$
Nový Jičín	150000	$0,50$	0	0
Vsetín	140000	$0,40$	50	28000
Přerov	130000	$0,45$	15	8775

Tabulka 3.2: Celkový požadavek na zpracované množství.

celková produkce $[t \cdot r^{-1}]$	požadavek $[\%]$	minimální množství $[t \cdot r^{-1}]$
189500	67	126965

Tabulka 3.3: Vlastnosti potenciálních zařízení v daných městech.

okresní město	kapacita $[t \cdot r^{-1}]$	cena za zpracování $[kc \cdot t^{-1}]$
Nový Jičín	150000	17
Vsetín	200000	20
Přerov	175000	18

Poznámka. Údaje v tabulkách jsou smyšlené a slouží pouze ke snadné ilustraci principu dopravní úlohy.

Tabulka 3.4: Vlastnosti všech cest spojujících jednotlivá okresní města.

cesta	délka $[km]$	cena za použití $[kc \cdot km^{-1}]$
Nový Jičín - Vsetín	36	5
Vsetín - Přerov	53	6
Přerov - Nový Jičín	57	6

Řešení

Úloha s těmito parametry byla algoritmem velmi rychle vyřešena. Nejekonomičtějším řešením je postavit zařízení v Novém Jičíně. Z okresu Přerov je do Nového Jičína převezeno pouze takové množství odpadu, aby byl splněn parciální požadavek Přerova. Z Nového Jičína je do zařízení odvezen veškerý odpad vyprodukovaný na území tohoto okresu. Ze Vsetína do Nového Jičína je pak odvezeno takové množství odpadu, aby byl v součtu splněn celkový požadavek (viz tabulka 3.5).

Tabulka 3.5: Řešení

cesta	množství převezeného odpadu $[t \cdot r^{-1}]$
Vsetín - Nový Jičín	43190
Přerov - Nový Jičín	8775
Nový Jičín - Zařízení Nový Jičín	126965

3.2. Matematická interpretace

V následujících podkapitolách budou symbolicky popsány všechny množiny, vstupní parametry a proměnné, které se v modelu vyskytují. Pomocí těchto symbolů bude následně popsán tvar úlohy lineárního programování vycházející ze standardního tvaru dopravní úlohy (rovnice 2.5).

3.2.1. Množiny

Pro množiny, které jsou součástí úlohy, platí následující vztahy. Množiny jsou popsány v tabulce 3.6.

$$\begin{aligned} M \cap Z &= \emptyset, \\ M \cup Z &= I. \end{aligned} \tag{3.1}$$

$$\begin{aligned} V \cap S &= \emptyset, \\ V \cup S &= J. \end{aligned} \tag{3.2}$$

Tabulka 3.6

označení	název	poznámka
I	množina uzlů	
M	množina obcí	$M \subset I$
Z	množina potenciálních zařízení	$Z \subset I$
J	množina hran	
V	množina hran propojující dvě obce	$C \subset J$
S	množina hran propojující obec a jí příslušné potenciální zařízení	$S \subset J$

3.2.2. Vstupní parametry

V tabulce 3.7 jsou uvedeny všechny vstupní parametry úlohy včetně popisu a jednotky.

Tabulka 3.7

označení	typ	název	jednotka
pop_m	vektor $\|M\| \times 1$	Populace podoblasti	$[ob.]$
$prod_m$	vektor $\|M\| \times 1$	Produkce odpadu podoblasti	$[t \cdot r^{-1} \cdot ob.^{-1}]$
dem_m	vektor $\|M\| \times 1$	Parciální požadavek podoblasti	$[\%]$
cap_z	vektor $\|Z\| \times 1$	Kapacita zařízení	$[t \cdot r^{-1}]$
d_v	vektor $1 \times \|V\|$	Vzdálenost hrany	$[km]$
w_v	vektor $1 \times \|V\|$	Cena za přepravu po hraně	$[kc \cdot km^{-1} \cdot t^{-1}]$
w_s	vektor $1 \times \|Z\|$	Cena za zpracování v zařízení	$[kc \cdot t^{-1}]$
A_{ij}	matice $\|I\| \times \|J\|$	Incidenční matice	$[-]$
C_{zk}	matice $\|Z\| \times \|K\|$	Kombinační matice řešení	$[-]$
dem	skalár	Celkový požadavek oblasti	$[\%]$
$pocet$	skalár	Počet vybudovaných zařízení	$[-]$

Kombinační matice řešení C je matice $|Z| \times |K|$, kde $|Z|$ odpovídá počtu podoblastí a $|K|$ je počet řešení, které připadají při výpočtu v úvahu. Množina K je indexová množina. Pro sloupce matice C pak platí následující vztah:

$$\sum_{z \in Z} C_{zk} = pocet, \forall k \in K. \tag{3.3}$$

Dokud nebude řečeno jinak, uvažujme, že sloupce matice C jsou všechny možné kombinace, které mohou nastat.

3.2.3. Proměnné

V tabulce 3.8. jsou uvedeny veškeré proměnné, které do úlohy vstupují včetně popisu a jednotky.

Tabulka 3.8

označení	typ	název	jednotka
x_j	vektor $1 \times \lvert J \rvert$	Množství odpadu převezené po hraně j	$[t \cdot r^{-1}]$
y_k	vektor $1 \times \lvert K \rvert$	Hodnota účelové funkce pro danou kombinaci k	$[kc \cdot r^{-1}]$

3.2.4. Úloha

Následující vztahy jsou úpravou standardního tvaru dopravní úlohy (rovnice 2.5).

$$
\begin{aligned}
& \forall k \in K : \\
\min \quad & y_k = \sum_{v \in V} d_v \cdot w_v \cdot x_v + \sum_{s \in S} w_s \cdot x_s \\
\text{s. t.} \quad & x_j \geq 0, && \forall j \in J, \\
& \sum_{j \in J} -a_{mj} x_j \leq -dem_m, && \forall m \in M, \\
& \sum_{j \in J} a_{mj} x_j \leq prod_m \cdot pop_m, && \forall m \in M, \\
& \sum_{s \in S} a_{zs} x_s \leq cap_z \cdot C_{zk}, && \forall z \in Z, \\
& dem \cdot \sum_{m \in M} prod_m \leq \sum_{z \in Z} \sum_{s \in S} a_{zs} \cdot x_s.
\end{aligned}
\tag{3.4}
$$

$$
\min_{k \in K} y_k. \tag{3.5}
$$

Úloha je řešena jako cyklus o $|K|$ iteracích. Každá iterace se liší pouze hodnotami sloupcového vektoru C_k. Z vypočtených řešení je pak vybráno to minimální.

3.3. Struktura modelu

Celý model je naprogramován v prostředí Matlab [10]. Všechny zdrojové kódy a vstupní soubory jsou uloženy v příloze této práce.

3.3.1. Vstupní data

Vstupem modelu je tabulka vytvořená v prostředí Microsoft Excel [9]. Obrázek 3.1 ukazuje vstupní data úlohy popsané v kapitole 3.1.2.

	Novy_Jici	Vsetin -	Novy_Jici	Prerov -	Vsetin -	Prerov - Vsetin		pocet_ob	produkce	pozadova	kapacita	cena_za_spalenou_tunu
Novy_Jici	1	-1	1	-1	0	0		150000	0,5	0	150000	17
Vsetin	-1	1	0	0	1	-1		140000	0,4	50	200000	20
Prerov	0	0	-1	1	-1	1		130000	0,45	15	175000	18
								[ob.]	[t/ob./r]	[%]	[t/rok]	[kč/t]
delka_hr	36	36	57	57	53	53	[km]		pozad	67	[%]	
cena_za_	5	5	6	6	6	6	[kč/km/t]		pocet	1	[-]	

Obrázek 3.1: Vstupní data ilustrační úlohy.

3.3.2. Využití vhodného algoritmu v závislosti na časové náročnosti

Způsob řešení tak, jak je popsán v kapitole 3.2.4 je sice správný způsob řešení, ale pro úlohy rozsáhlejšího charakteru se stává z časového hlediska nevhodným. Časovou náročnost výpočtu reprezentuje tabulka 3.9.

Tabulka 3.9

počet uzlů	počet operací	doba výpočtu
3	3	2s
6	15	3s
15	3003	30s
21	116280	623s
40	18 643 560	odhadem 26 hodin

Odhadovaný čas doby výpočtu pro situaci v posledním řádku tabulky byl vypočten na znalosti průměrné doby výpočtu jedné operace.

Model je opatřen rozhodovacím větvením, které v závislosti na rozsáhlosti vstupních dat úlohy rozhodne, zda bude problém řešený tradiční metodou řešení, nebo zda bude k řešení využit genetický algoritmus.

Uživatel zvolí hraniční počet operací. Pokud vstupní data tento počet operací překonají, na řadu přichází genetický algoritmus.

Tradiční metody řešení

Problém je řešen exaktně úplnou enumerací. Kombinační matice C obsahuje všechny možné kombinace postavených a nepostavených zařízení. Pro počet prvků množiny K platí následující vztah.

$$|K| = \binom{|Z|}{pocet}. \tag{3.6}$$

Takto získané řešení je s jistotou správné řešení minimalizační úlohy.

Využití genetického algoritmu

Na problém je aplikován genetický algoritmus v takové formě, v jaké byl popsán v teoretické části této práce (viz kapitola 2.4).

Hovoří se zde o tzv. *jedincích.* Jedinci jsou v tomto modelu reprezentováni sloupci matice C. Každý sloupec C_k odpovídá jednomu jedinci. V případě zjednodušené interpretace modelu popsané v kapitole 3.1.2 by matice C byla následujícího tvaru.

$$C = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

Poznámka. V případě popsaném v kapitole 3.1.2 byla úloha vyřešena deterministicky, protože množina K měla pouze 3 prvky. Její matice C je zde uvedena pouze z důvodu snadné ilustrace.

U řešení získaného pomocí genetického algoritmu není obecně zaručeno, že bude správné, protože do algoritmu vstupuje jistý prvek náhody. Nicméně, po důkladném testování a po provedení mnoha porovnávacích výsledků je možné tvrdit, že algoritmus je správně podmíněn a řešení, s jeho pomocí vypočtené za mnohem kratší čas, je opravdu řešením minimalizační úlohy.

3.3.3. Výstup

Výstupem modelu je textový soubor. Obrázek 3.2 ukazuje, jak vypadá výstupní textový soubor v případě zjednodušené interpretace modelu popsané v kapitole 3.1.2.

```
Hodnota ucelove funkce:  12933655.0 kc/rok

  zarizeni                                           stav
Zarizeni_Novy_Jicin                               otevreno
Zarizeni_Vsetin                                   neotevreno
Zarizeni_Prerov                                   neotevreno

     hrana                                      mnozstvi [t/rok]
Vsetin - Novy Jicin                                  43190
Prerov - Novy_Jicin                                  8775
Novy_Jicin-Zarizeni_Novy_Jicin                       126965
```

Obrázek 3.2: Výstupní data ilustrační úlohy.

3.4. Aplikace modelu na reálných datech

3.4.1. Zadání problému

Uvažovanou oblastí je celé území České republiky. Podoblastmi jsou v tomto konkrétním příkladě tzv. obce s rozšířenou působností (dále ORP), kterých je na území ČR 206.

Cílem úlohy je určit optimální rozmístění postavených zařízení tak, aby v nich bylo zpracováno 100% vyprodukovaného odpadu na celém území ČR. Počet postavených zařízení je zvolen s přihlédnutím k celkové produkci a průměrné kapacitě potencionálních zařízení.

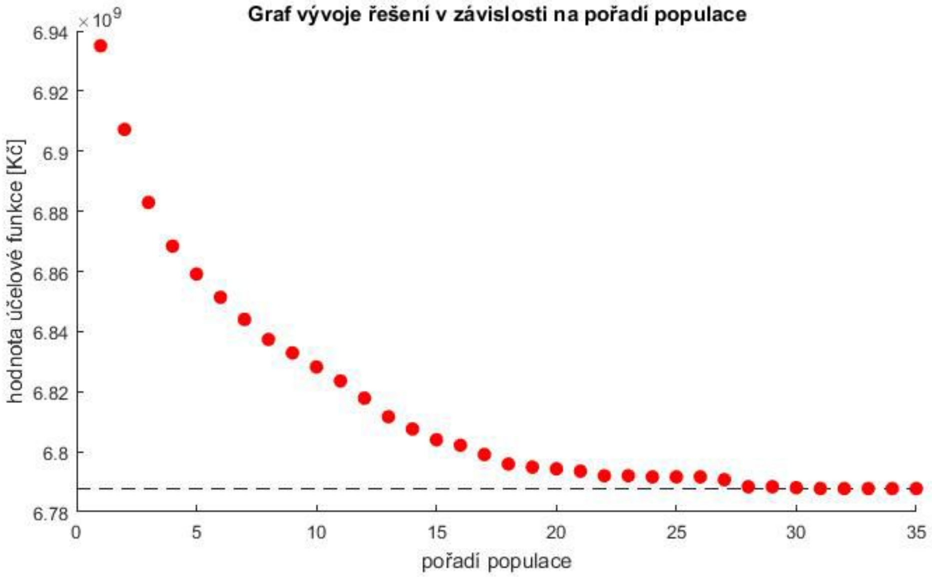
Veškeré vstupní parametry jsou přiloženy v souboru s názvem *data.xlsx*. Seznam ORP, jejich ročních produkcí odpadu, cest mezi jednotlivými ORP, jejich vzdáleností a příslušná incidenční matice a další data byla pro účely této práce poskytnuta Ústavem procesního inženýrství FSI VUT v Brně.

Cena za přepravu byla stanovena pro všechny hrany stejně na $4kc \cdot km^{-1} \cdot t^{-1}$. Kapacita všech potencionálních zařízení je $150kt \cdot r^{-1}$. S přihlédnutím k těmto parametrům bylo stanoveno, že je nutné vybudovat 22 zařízení.

3.4.2. Řešení

K řešení byl využit genetický algoritmus popsaný v kapitole 2.4. Úloha obsahuje celkem 1889 reálných a 206 binárních proměnných. Hledání řešení trvalo necelé 3 hodiny.

Graf na obrázku 3.3 ukazuje, jak se měnilo optimální řešení v závislosti na pořadí populace vytvořené genetickým algoritmem.



Obrázek 3.3: Závislost hodnoty účelové funkce na pořadí populace.

Z grafu je patrné, že ze začátku výpočtu se hodnota účelové funkce nejlepšího jedince měnila poměrně razantně. Až od 30. do 35. vytvořené populace zůstalo nejlepší dosavadní řešení nezměněno, algoritmus se s touto hodnotou účelové funkce spokojil a celý průběh byl ukončen.

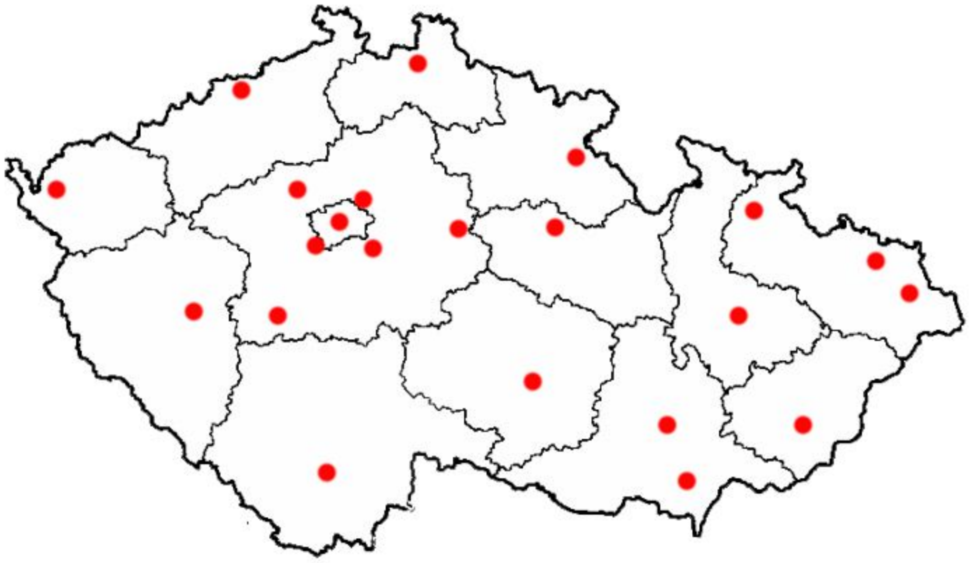
Výstup modelu ve formě textového souboru (viz kapitola 3.3.3) je v příloze této práce pod názvem *vystup_reseni.txt*. Optimální rozmístění postavených zařízení dle modelu je znázorněno na obrázku 3.4.

3.4.3. Vylepšení řešení

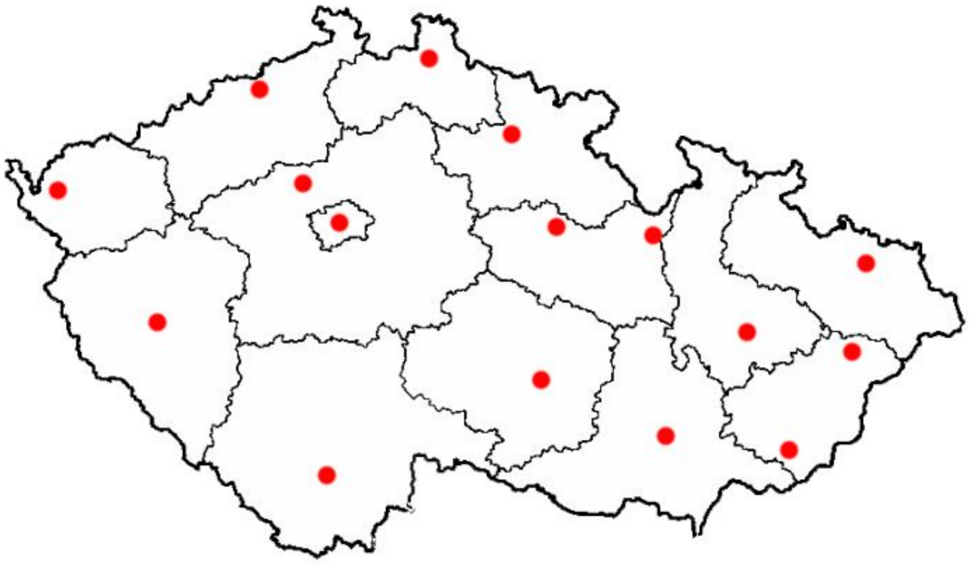
Při pohledu na obrázek 3.4 je zřejmé, že v okolí největších měst (Praha, Brno a Ostrava) je vybudováno více zařízení na relativně malém prostoru, což se jeví jako poměrně neekonomické. Uživatelským zásahem tedy pozměníme vstupní data úlohy tak, aby situace dopadla jinak.

Konkrétně je třeba zvýšit kapacity potencionálních zařízení v Praze, Brně a Ostravě. Je nutno podotknout, že kapacita potenciálního zařízení může být i podstatně větší než uvažovaných $150kt \cdot r^{-1}$. Zvýšíme tedy kapacitu zařízení v Praze pětkrát a v Brně a Ostravě dvakrát. Celý algoritmus s pozměněnými parametry byl spuštěn znovu. Takto modifikovaná vstupní data jsou přiloženy v souboru s názvem *data_upravene.xlsx*

Optimální rozmístění postavených zařízení se změněnými vstupními parametry je znázorněno na obrázku 3.5.



Obrázek 3.4: Mapa ČR s rozmístěním postavených zařízení.



Obrázek 3.5: Mapa ČR s rozmístěním postavených zařízení při změně vstupních dat.

4. Závěr

Tato práce byla zaměřena na problematiku dopravních úloh. V teoretické části práce byly popsány základní pojmy teorie grafů a také její aplikovatelnost v dopravních úlohách. Byly zde také vysvětleny základy matematického programování. Na smyšleném příkladě byla popsána typická dopravní úloha a v neposlední řadě byl také osvětlen princip genetického algoritmu včetně konkrétní aplikace.

Praktickou částí práce bylo vytvoření funkčního dopravního modelu. To sestávalo jak z naprogramování výpočtového aparátu v Matlabu, tak z jeho použití na konkrétním příkladě optimalizace svozu odpadu na území ČR.

Vzhledem k časové náročnosti výpočtu při rozsáhlejších vstupních datech se součástí modelu stal také genetický algoritmus. Rozsáhlou svozovou úlohu, která je popsána v kapitole 3.4 výpočtový aparát vyřešil za necelé 3 hodiny, což je oproti exaktnímu řešení (úplné enumeraci) odhadem o 30 milionů let rychlejší.

Řešení, které je reprezentováno obrázkem 3.4 v kapitole 3.4.2 bylo už na pohled neefektivní, a tak byly její okrajové podmínky vylepšeny tak, jak je popsáno v kapitole 3.4.3. Vylepšené řešení je pak ukázáno na obrázku 3.5.

Optimalizační modely a výpočtové aparáty obecně jsou rozhodně vynikající způsob, jak zefektivnit své počínání v různých oblastech. Nicméně, ne vždy je výstup modelu opravdu reálně použitelný, a proto je třeba, aby poslední a rozhodující slovo měl vždy člověk. Ten na základě výsledků vyhodnotí, jak vlastně řešení problému bude vypadat.

Literatura

[1] BAZARAA, M.S.: *Linear programming and network flows.* New York: Wiley, 1977. 565 s. ISBN 0-471-06015-1.

[2] DANTZIG, G.B.: *Application of the simplex method to a transportation problem, in Activity Analysis of Production and Allocation (T.C. Koopmans, ed.).* New York: Wiley, 1951. p. 359-373.

[3] DEMEL, J.: *Grafy a jejich aplikace.* Praha: Academia, 2002. 257 s. ISBN: 80-200-0990-6.

[4] HAUPT, R.L., HAUPT, S.E.: *Practical genetic algorithms.* Hoboken: John Wiley & Sons, 2004. 253 s. ISBN 0-471-45565-2.

[5] HITCHCOCK, F.L.: The Distribution of a Product from Several Sources to Numerous Localities *Supplied studies in mathematics*, vol. 20, April 1941, p.224-230.

[6] NEŠETŘIL, J.: *Teorie grafů.* Praha: SNTL, 1979. 316 s.

[7] POPELA, P.: *Optimization I.* Brno, February 2017. 122s.

[8] WILLIAMS, H.P.: *Model building in mathematical programming.* New York: Wiley, 1999. 354 s. ISBN 0-471-99788-9.

[9] *Microsoft excel documentation* [online]. [cit. 12-05-2018]. Dostupné z `https://msdn.microsoft.com/en-us/library/office/fp179694.aspx`

[10] *Matlab documentation* [online]. [cit. 11-04-2018]. Dostupné z `https://www.mathworks.com/help/matlab/`

[11] *Waste - Environment* [online]. [cit. 10-04-2018]. Dostupné z `http://ec.europa.eu/environment/waste/index.htm`

5. Seznam použitých zkratek a symbolů

s. t.	Z anglického *subject to*, česky *vzhledem k*
I	Množina uzlů
M	Množina obcí
Z	Množina potenciálních zařízení
V	Množina hran
J	Množina hran propojující dvě obce
S	Množina hran propojující obec a jí příslušné zařízení
K	Množina všech řešení
pop_m	Populace podoblasti [$ob.$]
$prod_m$	Produkce odpadu podoblasti [$t \cdot r^{-1} \cdot ob.^{-1}$]
dem_m	Parciální požadavek podoblasti [%]
cap_z	Kapacita zařízení [$t \cdot r^{-1}$]
d_v	Vzdálenost hrany [km]
w_v	Cena za přepravu po hraně [$kc \cdot km^{-1} \cdot t^{-1}$]
w_s	Cena za zpracování v zařízení [$kc \cdot t^{-1}$]
$A_{i,j}$	Incidenční matice
$C_{z,k}$	Kombinační matice řešení
dem	Celkový požadavek oblasti [%]
$pocet$	Počet vybudovaných zařízení
x_j	Množství odpadu převezeného po hraně [$t \cdot r{-}1$]
y_k	Hodnota účelové funkce [$kc \cdot r{-}1$]

6. Seznam příloh

Všechny přílohy jsou uloženy v jednom *zip* archivu.

Hlavní adresář archivu obsahuje dva podadresáře.

- *Model - vstup a výstup*
- *Model - program*

Adresář: Model - vstup a výstup

V tomto podadresáři se nacházejí soubory týkající se kapitoly 3.4 - „Aplikace modelu na reálných datech".

- *data.xlsx* (Vstupní soubor úlohy popsané v 3.4.1)
- *data_upravene.xlsx* (Upravený vstupní soubor popsaný v 3.4.3)
- *vystup_reseni.txt* (Výstup řešení úlohy popsané v 3.4.1.)

Adresář: Model - program

V tomto podadresáři se nacházejí soubory matlabovského typu, které dohromady tvoří celý výpočtový aparát a také jiné alternativní vstupní soubory (excelovské tabulky).

Tyto soubory zde budou zmíněny pouze výčtem, jejich význam je buďto zřejmý už z jejich pojmenování, nebo je „zapoznámkován" na prvních řádcích příslušného kódu. Soubory jsou pro přehlednost rozděleny do dalších podadresářů.

- *Model.m*
- *Combination_matrix.m*
- *Try_every_option.m*
- *Count_probability.m*
- *crossover.m*
- *Find_quality.m*
- *Generate_population.m*
- *Genetic_algorithm.m*
- *Choose_individual.m*
- *mutate.m*
- *reproduce.m*
- *Reproducted_population.m*

- *Find_best_solution.m*
- *Operation_count.m*
- *Solve_problem.m*
- *load_built.m*
- *load_caps.m*
- *load_edges.m*
- *load_incidence.m*
- *load_knots.m*
- *load_required.m*
- *load_weights.m*
- *data_giganticke.xlsx*
- *data_giganticke_modifikovane.xlsx*
- *data_ilustracni.xlsx*
- *data_male.xlsx*
- *data_stredni.xlsx*