Česká zemědělská univerzita v Praze

Provozně ekonomická fakulta

Katedra systémového inženýrství

Bakalářská práce

Optimalizace dopravních tras mezi firmou a jejími dodavateli a zákazníky

Petr Veselý

ČESKÁ ZEMĚDĚLSKÁ UNIVERZITA V PRAZE

Provozně ekonomická fakulta

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Petr Veselý

Systémové inženýrství

Název práce

Optimalizace dopravních tras mezi firmou a jejími dodavateli a zákazníky

Název anglicky

Optimization of transportation routes between a chosen company and its clients

---

Cíle práce

Cílem práce je optimalizace tras vozidla firmy rozvážejícího zboží do maloobchodů, za pomoci metod operačního výzkumu.

Metodika

Práce bude rozdělena na dvě části, teoretickou a praktickou. V teoretické části bude popis metod vybraných pro výpočet tras. V praktické části tyto metody budou použity pro výpočet optimálních tras a výsledek bude následně porovnán s trasami, které používá firma.

Doporučený rozsah práce

30-40 stran

Klíčová slova

Optimalizace tras, Operační výzkum, Okružní dopravní problém, Logistika

---

Doporučené zdroje informací

BROŽOVÁ, H. – HOUŠKA, M. – ČESKÁ ZEMĚDĚLSKÁ UNIVERZITA V PRAZE. PROVOZNĚ EKONOMICKÁ FAKULTA. *Základní metody operační analýzy.* Praha: Česká zemědělská univerzita v Praze, Provozně ekonomická fakulta ve vydavatelství Credit, 2002. ISBN 80-213-0951-2.

COOK, W. *Po stopách obchodního cestujícího : matematika na hranicích možností.* Praha: Dokořán, 2012. ISBN 978-80-7363-412-4.

KOSKOVÁ, I. – ČESKÁ ZEMĚDĚLSKÁ UNIVERZITA V PRAZE. KATEDRA OPERAČNÍ A SYSTÉMOVÉ ANALÝZY. *Distribuční úlohy I.* Praha: ČZU-PEF, 2004. ISBN 80-213-1156-8.

ŠUBRT, T. *Ekonomicko-matematické metody.* Plzeň: Vydavatelství a nakladatelství Aleš Čeněk, s.r.o., 2015. ISBN 978-80-7380-563-0.

Předběžný termín obhajoby

2021/22 LS – PEF

Vedoucí práce

RNDr. Petr Kučera, Ph.D.

Garantující pracoviště

Katedra systémového inženýrství

Elektronicky schváleno dne 24. 11. 2021

doc. Ing. Tomáš Šubrt, Ph.D.

Vedoucí katedry

Elektronicky schváleno dne 25. 11. 2021

Ing. Martin Pelikán, Ph.D.

Děkan

V Praze dne 19. 12. 2021

Čestné prohlášení

Prohlašuji, že svou bakalářskou práci " Optimalizace dopravních tras mezi firmou a jejími dodavateli a zákazníky" jsem vypracoval samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou citovány v práci a uvedeny v seznamu použitých zdrojů na konci práce. Jako autor uvedené bakalářské práce dále prohlašuji, že jsem v souvislosti s jejím vytvořením neporušil autorská práva třetích osob.

V Praze dne 15.03.2022 ______________________________

Poděkování

Rád bych touto cestou poděkoval svému vedoucímu panu RNDr. Petru Kučerovi, Ph.D. za konzultace, cenné rady a odborné vedení práce. Dále bych rád poděkoval panu majiteli společnosti J.G.S. TRADE s.r.o za jeho spolupráci a poskytnutí dat potřebných k vykonání mé bakalářské práce. V neposlední řadě děkuji svým blízkým a rodině za jejich podporu během mého studia.

Optimalizace dopravních tras mezi firmou a jejími dodavateli a zákazníky

Abstrakt

Bakalářská práce se zabývá optimalizací dopravních tras mezi firmou J.G.S TRADE s.r.o a maloobchody po celé České republice. Cílem práce je optimalizovat trasy vozidla pomocí metod operačního výzkumu. Cílem této optimalizace je minimalizovat délku tras a tím minimalizovat časovou a finanční náročnost těchto tras.

Práce je rozdělena na dvě části, teoretickou a praktickou. V teoretické části jsou popsány důležité pojmy a metody potřebné pro následné řešení problému, jako je Lineární programování, Jednookruhový okružní dopravní problém apod.

V části praktické je představena firma a důkladněji popsán řešený problém a vstupní data. Následně je popsán proces řešení problému vhodnými metodami. V závěru jsou shrnuty a interpretovány výsledky. Výsledky jsou také porovnány s trasami, které firma běžně používá.

Klíčová slova: Optimalizace tras, Operační výzkum, Okružní dopravní problém, Logistika

Optimization of transportation routes between a chosen company and its clients

Abstract

The bachelor's thesis focuses on optimalisation of logistics routes between the company J.G.S TRADE s.r.o and small businesses throughout the Czech Republic. The goal of the thesis is optimizing the vehicle routes using operations research methods. The objective is to minimize the route length, which in turn minimizes the time and resource requirements of these routes.

The theoretical part establishes important terms and describes the methods needed for solving the problem – linear programming, travelling salesman problem etc.

The empirical part introduces the company and describes the given problem in more depth, as well as provides the input data. Furthermore, the process of solving the problem using appropriate methods is described in detail. The conclusion reports and interprets the output of the calculations, which is also compared to the actual routes, which are in reality being used by the company on the daily basis.

Keywords: Optimization of transportation routes, Operation research, Traveling salesman problem, Logistics

Obsah

Seznam obrázků

Seznam tabulek

1 Úvod

Bakalářská práce se zabývá optimalizací okružních tras pro rozvoz zboží do několika konkrétních předem určených míst s ohledem na otevírací dobu míst a pracovní dobu řidiče rozvážejícího zboží. Cílem této optimalizace je ušetřit čas a najít ty nejrychlejší možné okruhy tras, které zároveň navštíví všechny místa během jejich otevírací doby.

Spousta společností k nalezení těchto tras nepoužívá žádné matematické metody. To znamená, že nepoužívají optimální trasy a tím zbytečně ztrácí čas a zvyšují si tím náklady nebo se jim nedaří navštívit jednotlivá místa ve správný čas.

Distribuční problémy se zabývají tvorbou a aproximací těchto okružních tras. Metody pro řešení těchto problému vycházejí z lineárního programování a jsou často označovány za problémy obchodního cestujícího.

Pro tuto práci byla vybrána společnost J.G.S. TRADE s.r.o. se sídlem v Praze, odkud zajišťuje svépomocí distribuci svého zboží do maloobchodů na území České republiky. K distribuci firma používá opakující se trasy, které budou předmětem optimalizace k ověření jejich optimality a následné porovnání s trasami, které firma k distribuci aktuálně používá.

Cíl práce a metodika

1.1 Cíl práce

Cílem práce je optimalizace tras vozidla firmy J.G.S. TRADE s.r.o. rozvážejícího zboží do maloobchodů na území České republiky za pomoci metod operačního výzkumu. Záměrem je nalezení těch nejrychlejších okružních tras s ohledem na otevírací doby zásobovaných maloobchodů a pracovní doby řidiče vozidla. Optimalizace pomocí rozšířené metody jednookruhového okružního dopravního problému proběhne na dvou opakujících se trasách, které vozidlo jezdí každé dva týdny. Pro výpočet je použit software OpenSolver.

1.2 Metodika

Práce je rozdělena na dvě části, teoretickou a praktickou. Teoretická část se zabývá studiem odborné literatury, které napomáhá k objasnění problematiky optimalizačních modelů a metod pro jejich výpočet. V praktické části tyto metody jsou použity pro výpočet optimálních tras a výsledek je následně porovnán s trasami, které používá firma.

2 Teoretická východiska

2.1 Operační výzkum

Na základě potřeb řízení operací v systémech vznikla vědecká disciplína nesoucí název operační výzkum (operační analýza). Operační výzkum řeší složité systémové problémy pomocí exaktních matematických metod a jiných systémových nástrojů.

Historicky se počátek operačního výzkumu datuje k roku 1937, kdy ve Velké Británii zkoumali systém správného nasazení radarů sloužících k protiletecké obraně. To vedlo ke vzniku první skupiny operačního výzkumu, která byla součástí Královského letectva (RAF). Po tom, co skončila druhá světová válka se začaly metody operačního výzkumu rozšiřovat do civilního průmyslu a následně do dalších oblastí lidského života. K dalšímu vývoji této vědecké disciplíny jednoznačně přispěl vývoj výpočetní techniky. Stejně tak přispěl operační výzkum rozvoji výpočetní techniky.

Hlavním cílem operační analýzy je poskytnutí exaktních systémových informací pro řešení problémů složitých systémů. Podstatnými rysy operačního výzkumu jsou:

1. Práce v týmu
2. Standardní postup řešení problémů
3. Systémový přístup
4. Matematické modely a modelování
5. Použití výpočetní techniky

Úkolem matematických modelů operačního výzkumu je poskytnout srozumitelný popis všech faktorů relevantních k dané situaci a umožnit studium všech podstatných vztahů mezi prvky systému, který je zkoumán. Podstatou těchto matematických modelů je, aby v jejich konečné úpravě představovaly matematické schéma (např. soustavou rovnic a funkcí) popisující zkoumaný systém (Brožová a Houška, 2002).

2.1.1 Význam modelování a operačního výzkumu

Modelování člověku umožňuje myšlenkově pochopit složité souvislosti, vztahy a vazby mezi systémy a následně na základě modelů provádět v systému racionální zásahy a řízení. Matematické modely poskytují srozumitelný a koncentrovaný popis podstatných faktorů dané situace. To umožní odhalit všechny podstatné vztahy mezi jednotlivými prvky zkoumaného systému a vyjádřit tyto vztahy kvantitativně.

Matematické modely operačního výzkumu umožňují zjištění potřebných informací, když není možné nebo je velmi obtížné vyvodit závěry přímo ze zkoumaného objektu. Urychlují rozhodovací procesy. Stručností a přehledností modelové formy také usnadňují a racionalizují rozhodovací procesy. Umožňují různé varianty řešení a na základě jejich analýzy výběr nejvhodnějšího řešení dle aktuální situace. Zabraňují ztrátám v reálných systémech (Brožová a Houška, 2002).

2.2 Optimalizační modely

Dle Brožové a Houšky (2002, s. 11) „Je to skupina velmi obecných modelů. Tyto modely slouží k nalezení nejlepšího řešení problémů, přitom možná řešení jsou prvky nějaké konečné či nekonečné množiny. Patří sem např. lineární, nelineární, dynamické a stochastické programování nebo vícekriteriální rozhodování“.

Rozhodovací problémy jsou spojeny s řadou předpokladů vymezujících reálná řešení. Každá omezující podmínka musí být při řešení plně respektována a výsledné řešení musí být nalezeno v rámci těchto omezujících podmínek (Šubrt a kolektiv, 2015).

2.3 Lineární programování

Lineární programování je technika určena pro optimalizaci lineární účelové funkce a má za cíl na základě omezujících podmínek a účelové funkce najít hodnoty proměnných tak aby hodnota účelové funkce nabyla požadovaného minimálního nebo maximálního extrému (Brožova a Houška, 2002).

O model lineárního programování se jedná v případě, kdy se pro formulaci optimalizačního modelu používá pro vyjádření kritéria lineární funkce a pro vyjádření omezujících podmínek lineární rovnice a nerovnice. Tyto modely mají nevýhodu v určité míře nepřesnosti, vycházející z předpokladu linearity zobrazovaných procesů a deterministického charakteru parametrů modelu. I přes tyto nevýhody jsou důležité pro podporu rozhodování. Díky jednoduchosti a široké použitelnosti je lineární programovaní jednou z nejrozšířenějších metod využívaných při rozhodování. Jednoduchost řešení převažuje nad nevýhodami vyplívajících z nepřesnosti matematického popisu. Tato nepřesnost vychází z popisu závislostí nelineárního charakteru mezi proměnnýma pomocí soustavy lineárních rovnic (Šubrt a kolektiv, 2015).

Mezi nejčastější aplikace lineárního programování patří například:

- Optimalizace výrobní struktury
  - Má za cíl najít optimální rozsahy výrobních procesů v rámci daných výrobních kapacit
- Alokační problémy
  - Problémy, které mají rozdělit zdroje na pořízení určitých objektů, jako může být například optimalizace formy reklamy nebo portfolia
- Směšovací problémy
  - Cílem je najít ideální množství složek ve směsích tak aby byly zajištěny výsledné vlastnosti směsí
- Problémy dělení materiálu
  - Volba způsobu dělení materiálu, tak aby byla zajištěna požadovaná množství jeho jednotlivých částí a zbylo co nejméně nevyužitého materiálu
- Distribuční problémy
  - Optimalizují distribuci zboží mezi dodavateli a odběrateli. Více v podkapitole Distribuční problémy. (Brožová a Houška, 2002)

2.3.1 Matematická formulace lineárního optimalizačního modelu

Model lineárního programování má 3 základní prvky, a to vektor proměnných, omezující podmínky (včetně podmínek nezápornosti) a kriteriální nebo účelovou funkci. (Brožová a Houška, 2002)

Proměnné v modelu LP reprezentují jednotlivé procesy zajímavé z hlediska hledaného rozhodnutí. Na první pohled nemusí být jasné jak správně a účelně proměnné formulovat. Správné vymezení cíle řešeného problému by mělo poskytnout odkazy na procesy, které je možné pozorovat jako proměnné. Při určení proměnné je důležité, aby proměnná měla také jednotku, ve které je vyjádřena.

Omezující podmínky slouží k vymezení přípustné kombinace identifikovaných procesů. Mohou být pouze ve formě lineárních rovnic nebo nerovnic. Na levé straně těchto podmínek se vyskytuje skalární součin hodnot proměnných a tzv. technicko-ekonomických koeficientů. Tento skalární součin vyjadřuje množství vyčerpaného zdroje vyjádřeného v jednotkách procesu. Na pravé straně omezujících podmínek se vyskytuje konstanta, která omezuje minimální požadavek nebo maximální kapacitu.

Účelová funkce vyjadřuje cíl vyřešení problému. Oceňuje kvalitu jednotlivé přípustné kombinace.

Poslední nedílnou součástí modelu LP jsou podmínky nezápornosti. Mají dvojí význam. První význam je interpretovatelnost nalezeného řešení. Přesto, že matematicky by mohla být v pořádku záporná hodnota proměnné, tak v praxi je problematické přidat záporné množství zdroje. Druhý význam je výpočetní. Bez podmínek nezápornosti by nebylo možné používat např. simplexovou metodu. (Šubrt a kolektiv, 2015)

Matematicky je model lineárního programování zapsán takto:

Najít extrém účelové(kriteriální) funkce

$$z(x) = cTx \text{ à } MIN/MAX \tag{3.1}$$

za omezujících podmínek

$$Ax \begin{matrix} \leq \\ = \\ \geq \end{matrix} b \tag{3.2}$$

a za podmínek nezápornosti

$$x \geq 0 \tag{3.3}$$

a kde platí:

$x = (x_1, \ldots, x_n)^T$ je vektorem proměnných,

$z(x) = c^T x = \sum_{j=1}^{n} c_j x_j$ je účelovou funkcí,

$c^T = (c_1, \ldots, c_n)$ je vektorem cen nebo sazeb proměnných,

$A = (a_{ij})$ je maticí technickoekonomických koeficientů,

$b = (b_1, \ldots, b_n)^T$ je vektorem pravých stran soustavy omezujících podmínek a

$x \geq 0$ jsou podmínky nezápornosti, $x_j \geq 0, j = 1, \ldots, n$. (Brožová a Houška, 2002)

2.3.2 Řešení úlohy lineárního programování

Když už máme stanovený model Lineárního programování, můžeme ho začít řešit. Jsou dvě hlavní metody, kterými se dá úloha vyřešit, a to graficky nebo Simplexovým algoritmem. Grafické řešení je vhodné pouze v případě, kdy má model pouze 2 proměnné, a proto v praxi nemá příliš velké využití, neboť takto jednoduchých modelů moc není. Z toho důvodu se k řešení používá primárně simplexový algoritmus.

2.3.3 Simplexový algoritmus

Postup řešení úlohy lineárního programování je obecně založen na numerickém řešení soustavy nerovnic a rovnic odpovídající omezujícím podmínkám tak, aby účelová funkce nabývala hodnoty požadovaného extrému, tedy minima nebo maxima.

Nejznámější univerzální metoda pro řešení úloh lineárního programování je Simplexový algoritmus (Simplexová metoda). Je to iterační metoda, která vychází z Jordanovy eliminační metody a je doplněna o dvě kritéria, které umožňují najít optimální řešení. (Brožová a Houška, 2002)

2.3.3.1 Historická fakta

Simplexový algoritmus přinesl světu George Dantzig se svými spolupracovníky v roce 1947. Jeho objev vyřešil dříve neřešitelné matematické problémy a dodnes pomáhá šetřit přírodní a jiné zdroje téměř ve všech odvětvích, které si můžeme představit. (Cook, 2012)

2.3.3.2 Jordanova eliminační metoda

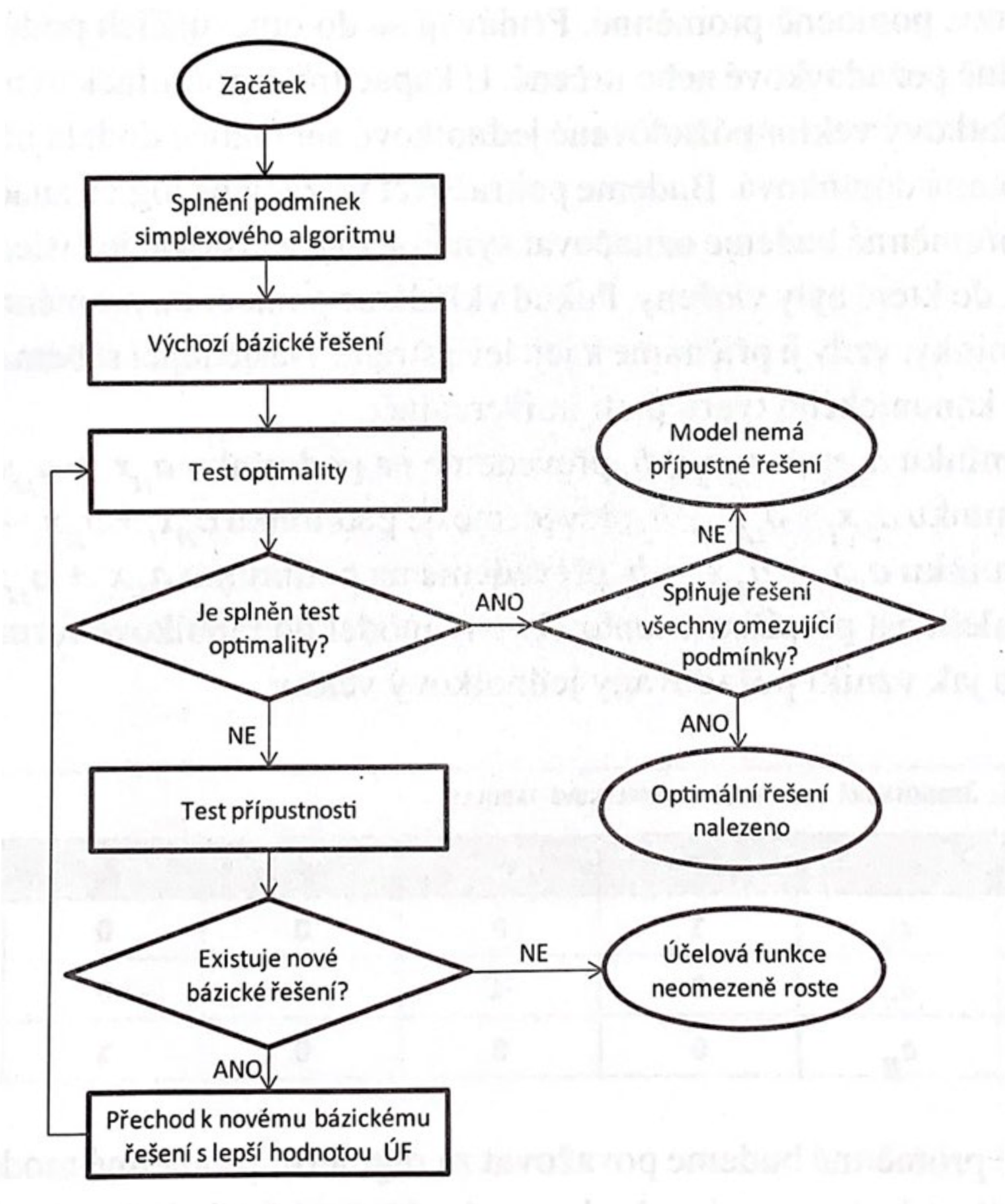
Jordanova eliminační metoda je obecná metoda řešení soustavy lineárních rovnic. Tato eliminační metoda vytvoří k původní soustavě rovnic ekvivalentní soustavu rovnic, jejíž matice soustavy je diagonální s jedničkami na diagonále. Cílem Jordanovy eliminační metody je úprava soustavy lineárních rovnic do kanonického tvaru.

Každý krok postupu eliminace začíná volbou pivota, neboli řídícího prvku pro další úpravy soustavy lineárních rovnic. Rovnice, která obsahuje řídící prvek se nazývá řídící rovnice a prvek, která má řídící prvek jako jeden z koeficientů, se nazývá řídící proměnná. Poté se řídící rovnice vydělí hodnotou řídícího prvku, tak aby jeho hodnota v upravené řídící rovnici byla rovna jedné. Dále se ostatní rovnice odečítáním a přičítáním vhodných násobků ostatních rovnic takovým způsobem, aby se ostatní koeficienty řídící proměnné rovnaly nule.

Pro řešení lineárních optimalizačních modelů jsou povoleny jen tyto eliminační úpravy soustavy rovnic:

1. Násobení řídící rovnice převrácenou hodnotou řídícího prvku.
2. Přičítání vhodného násobku řídící rovnice k upravované rovnici. (Brožová a Houška, 2002)

2.3.3.3 Řešení modelu lineárního programování pomocí simplexového algoritmu



Obrázek 1: Vývojový diagram simplexového algoritmu (Šubrt a kolektiv, 2015)

Ve vývojovém diagramu (Obrázek 1) je znázorněn postup aplikace simplexového algoritmu. Algoritmus předpokládá postupné aplikování znázorněných operací. Jednotlivé kroky algoritmu jsou detailněji popsány v následujícím textu. Pro lepší demonstraci postupu algoritmu budou jeho jednotlivé kroky vysvětleny na tomto modelu lineárního programování:

$$a_{11}x_1 + a_{12}x_2 \leq b_1$$
$$a_{21}x_1 + a_{22}x_2 \geq b_2$$
$$a_{31} + a_{32} = b_3$$
$$z = c_1x_1 + c_2x_2 \rightarrow MAX$$
$$x_{1,2} \geq 0$$

Splnění podmínek pro aplikaci simplexového algoritmu

Simplexovým algoritmem je možné řešit jen ty modely lineárního programování, jejichž matematický zápis splňuje tyto dvě podmínky:

1. ve vektoru pravých stran mohou být jen nezáporné hodnoty;
2. matice soustavy musí být v kanonickém tvaru.

Zajištění první podmínky je snadné. V případě, kdy je na pravé straně některé omezující podmínky záporná hodnota, vynásobíme ji hodnotou -1.

U druhé podmínky musí být zajištěno, aby matice soustavy byla v kanonickém tvaru. Pro splnění této podmínky musí obsahovat úplnou jednotkovou submatici. Situace, kdy je tato podmínka splněna po prosté formulaci modelu LP, je vzácná, proto je nutné celý model do kanonického tvaru převést. Nejprve se model převede do rovnicového tvaru tím, že se do omezujících podmínek ve tvaru nerovnic přidají doplňkové proměnné *d*. Pravidla pro převod podmínek do rovnicového tvaru jsou následující:

- kapacitní podmínka $a_{11}x_1 + a_{12}x_2 \leq b_1$ se převede na podmínku $a_{11}x_1 + a_{12}x_2 + d_1 = b_1$;
- požadavková podmínka $a_{21}x_1 + a_{22}x_2 \geq b_2$ se převede na podmínku $a_{21}x_1 + a_{22}x_2 - d_2 = b_2$;
- podmínka typu určení $a_{31} + a_{32} = b_3$ se ponechá beze změn.

Převod do rovnicového tvaru většinou není dostačující a matice soustavy, ani po této úpravě, neobsahuje úplnou jednotkovou submatici. Následuje tedy další krok a tím je převod do kanonického tvaru. Rovnicový tvar se rozšíří o pomocné proměnné *p*, za následujících pravidel:

- kapacitní podmínka $a_{11}x_1 + a_{12}x_2 + d_1 = b_1$ si zachová svůj rovnicový tvar
- požadavková podmínka $a_{21}x_1 + a_{22}x_2 - d_2 = b_2$ se převede na podmínku $a_{21}x_1 + a_{22}x_2 - d_2 + p_2 = b_2$
- podmínka typu určení $a_{31} + a_{32} = b_3$ se převede na podmínku $a_{31} + a_{32} + p_3 = b_3$

  (Šubrt a kolektiv, 2015)

Výchozí bazické řešení

Model LP, který můžeme považovat za bazické řešení je takový model, který obsahuje n proměnných a m omezujících podmínek, a ve kterém nejvýše m proměnných nabývá kladných hodnot. Přísluší jim jednotkové vektory v matici soustavy a nabývají hodnot odpovídající složky vektoru pravých stran. Všechny ostatní proměnné jsou považovány za nebázické a rovné nule. Optimální řešení modelu LP lze vybírat z konečného počtu bázických řešení úlohy.

Test optimality

Před každou další iterací je potřeba zjistit, zda aktuální bazické řešení lze zlepšit. Ke zjistění hodnoty testu optimality se použije tento vzorec: $z_j - c_j = c_B a_j - c_j$,

kde

c_B je vektor cen bázických proměnných;

a_j je vector matice soustavy pod testovanou proměnnou a

c_j cena testované proměnné.

U testu optimality se po zařazení nebazické proměnné

- v případě záporné hodnoty testu optimality se zvyšuje hodnota úcelové funkce;
- v případě kladné hodnoty testu optimality snižuje;
- v případě nulové hodnoty nezmění.

Řešení je optimální právě tehdy, kdy při maximalizaci nebude existovat žádná nebazická proměnná, jejíž zařazení zvýší hodnotu účelové funkce, a naopak u minimalizace, když neexistuje nebázická proměnná, která hodnotu účelové funkce sníží.

Jestliže řešení optimální není, je nutné vybrat nebázickou proměnnou, která v dalším kroku vstoupí do báze. Vždy se vybere proměnná, která má nejvyšší absolutní hodnotu testu optimality. Sloupec, ve kterém se proměnná vyskytuje se nazývá klíčový sloupec.

Test přípustnosti

Která proměnná vstupuje do báze určuje klíčový sloupec. Ale je třeba ještě určit, která z bázických proměnných bázi opustí. K tomu slouží test přípustnosti. Vztah pro výběr řádku vyřazované proměnné je formulován následujícím způsobem:

$$i: \min_{i=1} \frac{b_i}{a_{ij}} \forall a_{ij} > 0 \tag{3.4}$$

Vybere se tedy řádek s minimální kladnou hodnotou testu přípustnosti, tedy klíčový řádek a přejde se na nové bazické řešení.

Přechod k novému bazickému řešení

Po testu přístupnosti je známa proměnná, která vstupuje do báze vstupuje a proměnná, která opouští bázi. Sloupec vstupující proměnné se nazývá klíčový sloupec. Řádek opouštějící proměnné se nazývá klíčový řádek. Klíčový prvek se nachází na průsečíku klíčového sloupce a řádku. Přes tento prvek se provede jeden krok Jordanovy eliminační metody. Tím se zamění proměnné v bázi úlohy. Algoritmus pokračuje návratem k testu optimality nového řešení.

Model nemá žádné přípustné řešení

Tato situace nastává v momentě, kdy báze úlohy obsahuje alespoň jednu pomocnou proměnnou. V případě, že test optimality ukazuje, že dané řešení už nejde zlepšit, model nemá žádné přípustné řešení.

Hodnota účelové funkce může neomezeně růst

V této situaci existuje přípustné bazické řešení. Problémem je, že množina přípustných řešení není omezena ve směru růstu účelové funkce. Nakonec nejde nalézt další bazické řešení s lepší hodnotou účelové funkce, v důsledku neexistence kladné hodnoty v klíčovém sloupci. (Šubrt a kolektiv, 2015)

Model má právě jedno optimální řešení

V případě, že řešení splňuje všechny omezující podmínky, test optimality ukazuje, že už neexistuje způsob, jak zlepšit hodnotu účelové funkce a všechny nebazické proměnné zhoršují hodnotu účelové funkce, model má právě jedno optimální řešení.

Model má nekonečně mnoho optimálních řešení

Tato situace může nastat v případě, kdy je přímka účelové funkce rovnoběžná s některou z hran množiny přípustných řešení. U simplexového algoritmu, to nastane v momentě, kdy je splněn test optimality a zároveň existuje nebázická proměnná s hodnotou testu optimality rovnou nule. V takovém případě, změna báze nezmění hodnotu účelové funkce a model tak má nekonečně mnoho optimálních řešení.

2.4 Distribuční a dopravní modely

„Distribuční úlohy tvoří speciální skupinu úloh lineárního programování. Zařazujeme mezi ně problémy jednostupňové, dvoustupňové, přiřazovací, zobecněné, okružní, trasovací a mnoho dalších typů. Všechny tyto úlohy se dají vyjádřit pomocí lineárních modelů, jaké znáte z předchozí kapitoly. U některých z těchto úloh umožňují jejich specifické vlastnosti použít k řešení speciální metody, které jsou jednodušší než simplexová metoda. U jiných by naopak velikost modelu i při malé velikosti úlohy - malém počtu míst, mezi nimiž je třeba přepravu zajistit - vyžadovala výpočetní kapacitu, která neumožní efektivně nalézt jejich přesné teoretické optimum.“ (Šubrt a kolektiv, 2015)

2.4.1 Okružní dopravní problém

V praxi se okružní dopravní problémy vyskytují často v případech, kdy je potřeba rozvést materiál od jednoho nebo více dodavatelů k většímu množství spotřebitelů nebo v opačném případě od mnoha dodavatelů k jednomu spotřebiteli. V porovnání, kdyby byla realizována každá jednotlivá trasa od dodavatele ke spotřebiteli zvlášť, ušetří se okružním spojením náklady na jednotlivé výjezdy vozidel od stejného dodavatele nebo jízdy k jen jednomu spotřebiteli.

Z matematického hlediska okružní dopravní problém patří mezi tzv. NP-úplné problémy, pro které není žádný existující efektivní algoritmus, který by nalezl úplně přesné matematické optimum. To je způsobeno tím, že počet omezujících podmínek v matematickém modelu těchto úloh s rostoucím počtem míst roste exponenciálně. Tím stejně rychle roste i doba výpočtu jakoukoliv metodou. Proto existuje řada aproximačních metod, jejichž řešení se považuje za ekonomické optimum. Mezi tyto aproximační metody patří např. metoda nejbližšího souseda, Vogelova aproximační metoda a Mayerova metoda.

Dále se pak okružní dopravní problémy dělí na jednookruhové a víceokruhové okružní dopravní problémy dle počtu okruhů. (Šubrt a kolektiv, 2015)

2.4.1.1 **Jednookruhový okružní dopravní problém**

Jednookruhový okružní dopravní problém, také nazýván problémem obchodního cestujícího (TSP – Traveling salesman problem), je nejjednodušším typem okružních dopravních problémů. Přeprava materiálu mezi všemi obsluhovanými místy je v tomto případě realizována jedním okruhem. (Šubrt a kolektiv, 2015)

TSP má ve světe velmi široké využití. Jak z názvu vyplívá jeho hlavním využitím je výpočet okružních tras. To může sloužit k plánování tras např. služebních cest, doručování zásilek nebo dovolené na motorce jejíž cílem je navštívit 48 státu v USA. Ovšem plánování tras vozidel není jediným využitím. Pomocí TSP se mohou zaměřovat teleskopy, rentgeny a lasery. V továrnách na výrobu např. tištěných spojů se používá k přesnému ovládaní strojů, které dělají opakující činnosti jako je vrtání nebo pájení. (Cook, 2012)

Okružní dopravní problémy lze přehledně zobrazit pomocí grafů, jehož vrcholy jsou místa a hrany možná spojení jednotlivých míst. Je možné je rozdělit na základní dva typy, a to podle charakteru sítě, která spojuje místa. Na obrázku 3.2 je znázorněn okružní problém s úplnou sítí cest a problém s neúplnou sítí cest. V okružním problému s úplnou sítí cest existují přímá spojení mezi všemi dvojicemi míst. Naopak v tom bez úplné sítě cest, přímá spojení mezi všemi dvojicemi neexistuje. V případě neúplnosti cestní sítě, některá spojení nejsou či nesmí být realizována. (Brožová a Houška, 2002)

Obrázek 2: Okružní problém s úplnou nebo neúplnou cestní sítí (Brožová a Houška, 2002)

2.4.1.1.1 Obecná formulace TSP

Je dáno n míst (měst, uzlů) a sazba c_{ij} pro každou dvojici těchto měst (i, j) představující např. vzdálenost nebo spotřebu času či náklady pro spojení mezi místem i a místem j. Cílem úlohy je propojit všechna místa okružním spojením, tedy najít takovou posloupnost těchto míst tak aby se každé místo vyskytovalo v okruhu právě jednou s vyjímkou počátečního místa, které se znovu objeví na jejím konci, aby byl součet vazeb pro jednotlivá spojení v této posloupnosti minimální. (Šubrt a kolektiv, 2015)

2.4.1.1.2 Matematický model TSP

Cílem je nalezení minima lineární funkce

$$z = \sum_{i=1}^{n}\sum_{j=1}^{n} c_{ij}x_{ij} \rightarrow MIN\,, \tag{3.5}$$

za splnění podmínek

$$\sum_{j=1}^{n} x_{ij} = 1, \qquad i = 1,2,\dots,n, \tag{3.6}$$

$$\sum_{i=1}^{n} x_{ij} = 1, \qquad j = 1,2,\dots,n \tag{3.7}$$

$$u_i - u_j + nx_{ij} \leq n-1, i = 1,2,\dots,n;\ j = 1,2,\dots,n;\ i \neq j, \tag{3.8}$$

$$x_{ij} \in \{0;1\}, i = 1,2,\dots,n;\ j = 1,2,\dots,n. \tag{3.9}$$

(Šubrt a kolektiv, 2015)

V tomto matematickém modelu představuje *n* počet míst, které musí vozidlo projet a c_{ij} představuje vzdálenosti mezi místy i a j. Proměnná x_{ij} je bivalentní proměnná, která v případě kdy vozidlo jede do místa j z místa i nabývá hodnoty 1. V opačném případě nabývá hodnoty 0. Podmínky (3.6) a (3.7) zajišťují pouze jednu návštěvu místa. Soustava podmínek (3.8) je opatření proti vytváření parciálních cyklů. (Fábry, 2006)

2.4.1.2 **Jednookruhový okružní dopravní problém s časovými okny**

Tato úloha se často nazývá úloha obchodního cestujícího s časovými okny (TSPTW – Traveling salesman problem with time windows). Tato úloha stejně jako TSP, předpokládá znalost všech požadavku před zahájením optimalizace a samotnou realizací okružní trasy. Časové okno *i*-tého zákazníka je definováno intervalem mezi možného začátku obsluhy e_i a nejpozději přípustným začátkem obsluhy l_i. Okamžik, ve kterém začne obsluha *i*-tého zákazníka je označen a_i.

2.4.1.2.1 Matematická formulace modelu TSPTW s intervaly na obsluhu a intervaly čekání na obsluhu

Cílem je nalezení minima lineární funkce

$$z = \sum_{i=1}^{n}\sum_{j=1}^{n} t_{ij}x_{ij} + \sum_{i=2}^{n} S_i + \sum_{j=2}^{n} W_j \rightarrow MIN \tag{3.10}$$

za splnění podmínek

$$\sum_{j=1}^{n} x_{ij} = 1\,, \qquad i = 1,2,\dots,n, \tag{3.11}$$

$$\sum_{i=1}^{n} x_{ij} = 1, \qquad j = 1,2,\dots,n\,, \tag{3.12}$$

$$u_i - u_j + nx_{ij} \leq n - 1, \qquad i = 1,2,\dots,n;\ j = 1,2,\dots,n;\ i \neq j, \tag{3.13}$$

$$0 \leq v_{ij} \leq 2M\left(1 - x_{ij}\right), \qquad i = 1,2,\dots,n;\ j = 2,3,\dots,n;\ i \neq j\,, \tag{3.14}$$

$$a_i - a_j + Mx_{ij} + W_j + v_{ij} = M - S_i - t_{ij,} \qquad i = 1,2,\dots,n;\ j = 2,3,\dots,n;\ i \neq j, \tag{3.15}$$

$$x_{ij} \in \{0; 1\}, \qquad i = 1,2,\dots,n;\ j = 1,2,\dots,n, \tag{3.16}$$

$$a_1 = 0, \tag{3.17}$$

$$a_i \geq 0, i = 2,3,\dots,n, \tag{3.18}$$

$$e_i \leq a_i \leq l_i, \quad i = 2,3,\dots,n, \tag{3.19}$$

$$x_{ij} = 0, \quad i = 1,2,\dots,n;\ j = 1,2,\dots,n;\ i = j, \tag{3.20}$$

$$S_1 = 0, \tag{3.21}$$

$$W_1 = 0, \tag{3.22}$$

$$Si \geq 0, \quad i = 2,3, \dots, n, \tag{3.23}$$

$$Wj \geq 0, \quad j = 2,3, \dots, n, \tag{3.24}$$

a kde platí, že n je počet míst, které vozidlo musí projet, t_{ij} představuje dobu přejezdu mezi místy i a j. Hodnoty e_i a l_i představují nejdříve možný a nejpozději přípustný termín obsluhy zákazníka i, M představuje vysokou konstantu. Proměnná x_{ij} je bivalentní proměnná, která v případě kdy vozidlo jede do místa j z místa i nabývá hodnoty 1. V opačném případě nabývá hodnoty 0. Proměnná a_i udává moment, ve kterém proběhne návštěva místa i. Proměnná S_i představuje interval obsluhy a proměnná W_j interval čekání na obsluhu

Účelová funkce (3.10) a podmínky (3.11) - (3.13) mají stejný význam jako v modelu u jednookružního okružního dopravního problému. Podmínky (3.14) stanovují obsloužení míst uvnitř časového okna. Omezení (3.15) zajišťují, že časový interval mezi návšťevou místa j hned po místě i má minimálně hodnotu t_{ij}, s ohledem na interval čekání na obsluhu W_j a interval obsluhy S_i. V případě, že vozidlo nepojede od místa i k místu j, pak díky vysoké konstantě M je tato nerovnost splněna vždy. Rovnice (3.17) definuje nulový okamžik výjezdu z výchozího místa. Rovnice (3.21) definuje nulový interval obsluhy u výchozího místa a rovnice (3.22) nulový interval čekání na obsluhu u výchozího místa. Podmínky (3.23) – (3.24) zajišťují nezápornost proměnných S_i a W_j. *(Fábry, 2006)*

2.4.1.3 **Víceokruhový okružní dopravní problém**

Víceokruhový okružní dopravní problém se používá v případě, kdy kvůli kapacitním omezení je potřeba okružní přepravu rozdělit do více okruhů. Takovým omezením může být např. nedostačující kapacita vozidla. Mnohdy kapacita vozidla nepokrývá požadavky všech míst na množství materiálu, které je třeba svézt nebo rozvézt. Za předpokladu, že všechna vozidla mají kapacitu stejnou a zároveň menší než celkový objem požadavků, je nutné naplánovat pro každé vozidlo okruh, který začíná a končí v centrálním místě, jeho suma požadavků všech necentrálních míst není větší než kapacita vozidla a každé necentrální místo leží na jednom okruhu. Jedno z možných řešení tohoto problému je Mayerova metoda. (Šubrt a kolektiv, 2015)

2.5 Řešení úloh lineárního programování softwarem

Lineární programování je všeobecně časově náročné. V důsledku toho se pro výpočty větších modelů používá software.

Obvykle software pro optimalizaci lineárního programování vychází z konvenčních algoritmů jako je např. simplexová metoda. Tyto algoritmy se často kombinují s dalšími matematickými metodami pro rychlejší a efektivnější řešení problému.

2.5.1 OpenSolver

OpenSolver je open-source doplněk pro tabulkový procesor Microsoft Excel. Je určený pro řešení lineárních, nelineárních a celočíselných optimalizačních úloh. Vyvíjí a spravuje ho Andrew Mason na Aucklandské univerzitě a Iann Dunning. Tento doplněk rozšiřuje původní řešitel(solver), který je součástí Excelu, o několik více výkonných řešitelů, které nejsou tak omezeny jako tradiční solver. To posouvá limity původního řešitele a umožňuje počítat rozsáhlejší úlohy. Další výhodou oproti klasickému solveru je vizualizace modelu, která zvýrazňuje proměnné, účelovou funkci a všechny podmínky přímo uvnitř tabulkového procesoru. (OpenSolver for Excel, 2022)

3 Vlastní práce

3.1 Charakteristika subjektu

Společnost J.G.S. TRADE s.r.o. je na trhu přes 20 let a zabývá se prodejem a distribucí střelných zbraní, střeliva a dalšího se zbraněmi souvisejícího příslušenství. Sídlem společnosti jsou pražské Vršovice, kde se nachází i centrální sklad.

Zboží distribuují především maloobchodům po celé České republice. Vzhledem k povaze zboží a bezpečnosti, distribuci zajišťují vlastními dodávkami.

3.2 Specifikace problému

Společnost zásobuje pravidelně maloobchody na území České republiky. Všechny trasy se obvykle opakují každé dva týdny. Tato pravidelnost znamená především to, že má společnost určeno, jaké dny v měsíci musí rozvézt zboží do určitých měst. Z tohoto důvodu je důležité, aby jednotlivé trasy byly co nejlépe optimalizovány, vzhledem k jejich opakování může být totiž úspora na nákladech značná.

Cílem práce je tedy optimalizovat dopravní trasy mezi jejími odběrateli a výchozím místem, tedy centrálním skladem a následně výsledky porovnat s trasami, které společnost obvykle využívá a tím zjistit, zda tyto trasy jsou optimální. Výchozí místo zároveň slouží jako bod pro návrat vozidla. Vhodná metoda pro optimalizování těchto dopravních tras je Okružní dopravní problém rozšířený o časová okna, ke zohlednění otevíracích dob jednotlivých odběratelů, pracovní doby řidiče vozu a času na vyložení zboží.

3.3 Vstupní data

Jako vstupní data mi byly poskytnuty informace o dvou pracovních dnech. Tyto data obsahují informace o odběratelích, které společnost musela zavézt zbožím během těchto dvou dnů. Konkrétněji data o každém odběrateli obsahují adresu, kde se nachází a jeho otevírací dobu pro určení časových oken, ve kterých je možné zboží přivézt. Z důvodu anonymizace jsou adresy zúženy na města. Tyto data jsou zobrazena v tabulce č.1.

Vstupní data			
Den	Město	Otevírací doba	
		od	do
1	Písek	9:00	12:00
1	Prachatice	9:00	12:00
1	Strakonice	10:00	17:00
1	Příbram	10:00	18:00
2	Chrudim	8:30	13:00
2	Pardubice	9:00	18:00
2	Hradec Králové	10:00	19:00
2	Nový Bydžov	10:00	14:00
2	Nymburk	12:00	18:00

Tabulka 1: Vstupní data (Vlastní zpracování, 2022)

Dále vím, že řidičovi začíná 8,5 hodiny dlouhá pracovní doba v 7:00, což je jeden z podstatných aspektů určení časových oken. Průměrná doba na převzetí zboží odběratelem činí 20 minut.

3.4 Sestavení modelu TSPTW s intervalem na obsluhu a intervalem čekání na obsluhu

Pro následný výpočet modelu je nejdříve potřeba sestavit vhodný model lineárního programování. Součástí modelu je účelová funkce, proměnné a omezující podmínky. Model je sestaven na základě modelu z kapitoly 3.4.1.2.

Tento model je následně jednotlivě použit pro výpočet obou tras pomocí programu Microsoft Excel s doplňkem OpenSolver.

3.5 Příprava modelu pro OpenSolver

Před výpočtem je potřebné vytvoření matice vzdáleností (v kilometrech) a matice časů (v minutách). Data pro tyto matice jsem získal pomocí služby Google Maps. Pro každé spojení mezi místy jsem našel vzdálenost a časovou náročnost cesty mezi nimi, pro konkrétní den, kdy se koná distribuce zboží. Získaná data jsem následně zanesl do jednotlivých matic.

V tabulkách č.2 a č.3 jsou vyobrazeny matice vzdáleností a matice časů pro první trasu. Pro druhou trasu jsou zobrazeny v tabulkách č.4 a č.5.

Matice vzdáleností - Trasa 1					
$C=(c_{ij})$	**Praha**	**Písek**	**Prachatice**	**Strakonice**	**Příbram**
Praha	0	111	153	114	65
Písek	111	0	45	27	53
Prachatice	153	45	0	36	97
Strakonice	114	27	36	0	57
Příbram	65	53	97	57	0

Tabulka 2: Matice vzdáleností pro trasu 1 (Vlastní zpracování, 2022)

Matice časů - Trasa 1					
$T=(t_{ij})$	**Praha**	**Písek**	**Prachatice**	**Strakonice**	**Příbram**
Praha	0	100	150	110	70
Písek	100	0	50	26	55
Prachatice	150	50	0	45	100
Strakonice	110	26	45	0	60
Příbram	70	55	100	60	0

Tabulka 3: Matice časů pro model trasy 1 (Vlastní zpracování, 2022)

Matice vzdáleností - Trasa 2						
C=(cij)	Praha	Chrudim	Pardubice	Hradec Králove	Nový Bydžov	Nymburk
Praha	0	127	119	110	91	54
Chrudim	127	0	12	34	28	84
Pardubice	119	12	0	26	38	76
Hradec Králove	110	34	26	0	27	67
Nový Bydžov	91	28	38	27	0	39
Nymburk	54	84	76	67	39	0

Tabulka 4: Matice vzdáleností pro model trasy 2 (Vlastní zpracování, 2022)

Matice časů - Trasa 2						
T=(tij)	Praha	Chrudim	Pardubice	Hradec Králove	Nový Bydžov	Nymburk
Praha	0	110	105	100	75	50
Chrudim	110	0	24	45	60	80
Pardubice	105	24	0	35	55	70
Hradec Králove	100	45	35	0	40	65
Nový Bydžov	75	60	55	40	0	45
Nymburk	50	80	70	65	45	0

Tabulka 5: Matice časů pro model trasy 2 (Vlastní zpracování, 2022)

Dále je potřeba vytvoření tabulek pro všechny proměnné, které jsou potřebné. Konkrétně se jedná o vektory u, a_i, W_j, S_i, matici $X=(x_{ij})$ a pomocnou proměnnou M. Tabulka vektoru proměnných v_{ij} se vytvoří až v dalším kroku s podmínkami. Vektor S_i je předvyplněn dle předpokladu, že doba návštěvy řidiče na každém místě je 20 minut. Tyto tabulky jsou zobrazeny na obrázku č.3.

X=(xij)	Praha	Písek	Prachatice	Strakonice	Příbram
Praha	0	0	0	0	0
Písek	0	0	0	0	0
Prachatice	0	0	0	0	0
Strakonice	0	0	0	0	0
Příbram	0	0	0	0	0

u	
u1	0
u2	0
u3	0
u4	0
u5	0

ai	
a1	0
a2	0
a3	0
a4	0
a5	0

Wj	
W1	0
W2	0
W3	0
W4	0
W5	0

Si - Doba návštěvy	
S1	0
S2	20
S3	20
S4	20
S5	20

M	1000

Obrázek 3: Tabulky proměnných pro trasu 1 v prostředí programu Microsoft Excel (Vlastní zpracování, 2022)

Stejně tak jak se vytvořily tabulky pro proměnné se ještě musí vytvořit tabulky pro všechny omezující podmínky. Všechny podmínky jsou zobrazeny v Tabulce 4, až na podmínky pro časová okna. Podmínky pro časová okna jsou zobrazeny v Tabulce 5.

i	j	xij	tij	vij	ui-uj+nxij <= n-1			ai-aj+M*xij+Wj+vij = M-Si-tij			vij <= 2M(1-xij)		
1	2	1	100	0		<=	4	0	=	900		<=	0
1	3	0	150	0		<=	4	0	=	850		<=	2000
1	4	0	110	0		<=	4	0	=	890		<=	2000
1	5	0	70	0		<=	4	0	=	930		<=	2000
2	3	1	50	0		<=	4	0	=	930		<=	0
2	4	0	26	0		<=	4	0	=	954		<=	2000
2	5	0	55	0		<=	4	0	=	925		<=	2000
3	2	0	50	0		<=	4	0	=	930		<=	2000
3	4	1	45	0		<=	4	0	=	935		<=	0
3	5	0	100	0		<=	4	0	=	880		<=	2000
4	2	0	26	0		<=	4	0	=	954		<=	2000
4	3	0	45	0		<=	4	0	=	935		<=	2000
4	5	1	60	0		<=	4	0	=	920		<=	0
5	2	0	55	0		<=	4	0	=	925		<=	2000
5	3	0	100	0		<=	4	0	=	880		<=	2000
5	4	0	60	0		<=	4	0	=	920		<=	2000

Tabulka 6: Tabulka podmínek pro model první trasy (Vlastní zpracování, 2022)

i	j	xij	tij	vij	ui-uj+nxij <= n-1			ai-aj+M*xij+Wj+vij = M-Si-tij			vij <= 2M(1-xij)		
1	2	1	110	0	5	<=	5	890	=	890	0	<=	0
1	3	0	105	1049	-2	<=	5	895	=	895	1049	<=	2000
1	4	0	100	1109	-3	<=	5	900	=	900	1109	<=	2000
1	5	0	75	1194	-4	<=	5	925	=	925	1194	<=	2000
1	6	0	50	1284	-5	<=	5	950	=	950	1284	<=	2000
2	3	1	24	0	5	<=	5	956	=	956	0	<=	0
2	4	0	45	1034	-2	<=	5	935	=	935	1034	<=	2000
2	5	0	60	1079	-3	<=	5	920	=	920	1079	<=	2000
2	6	0	80	1124	-4	<=	5	900	=	900	1124	<=	2000
3	2	0	24	912	1	<=	5	956	=	956	912	<=	2000
3	4	1	35	0	5	<=	5	945	=	945	0	<=	0
3	5	0	55	1040	-2	<=	5	925	=	925	1040	<=	2000
3	6	0	70	1090	-3	<=	5	910	=	910	1090	<=	2000
4	2	0	45	836	2	<=	5	935	=	935	836	<=	2000
4	3	0	35	890	1	<=	5	945	=	945	890	<=	2000
4	5	1	40	0	5	<=	5	940	=	940	0	<=	0
4	6	0	65	1040	-2	<=	5	915	=	915	1040	<=	2000
5	2	0	60	761	3	<=	5	920	=	920	761	<=	2000
5	3	0	55	810	2	<=	5	925	=	925	810	<=	2000
5	4	0	40	880	1	<=	5	940	=	940	880	<=	2000
5	6	1	45	0	5	<=	5	935	=	935	0	<=	0
6	2	0	80	676	4	<=	5	900	=	900	676	<=	2000
6	3	0	70	730	3	<=	5	910	=	910	730	<=	2000
6	4	0	65	790	2	<=	5	915	=	915	790	<=	2000
6	5	0	45	870	1	<=	5	935	=	935	870	<=	2000

Tabulka 7: Tabulka podmínek pro model druhé trasy (Vlastní zpracování, 2022)

Data v tabulce č.5 byla vypočtena na základě otevírací doby maloobchodů (Tabulka 1) a pracovní doby řidiče. Pracovní doba stanovuje základní maximální hodnotu všem časovým oknům na 510 minut. Toto maximum je následně v některých případech sníženo otevírací dobou. Otevírací doba také v některých případech zvyšuje minimální hodnotu časových oken. Proměnná a_1 jako počáteční časové okno se vždy rovná nule.

Časová okna				
ei <= ai <= li				
ei		**ai**		**li**
		0	=	0
120	<=	0	<=	300
120	<=	0	<=	300
180	<=	0	<=	510
180	<=	0	<=	510

Tabulka 8: Tabulka podmínek pro časová okna u trasy 1 (Vlastní zpracování, 2022)

Časová okna				
ei <= ai <= li				
ei		**ai**		**li**
		0	=	0
90	<=	110	<=	360
120	<=	154	<=	510
180	<=	209	<=	510
180	<=	269	<=	420
300	<=	334	<=	510

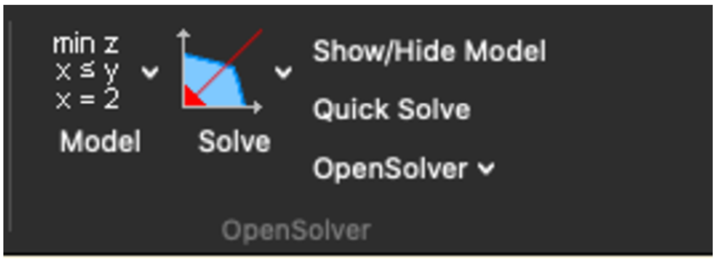
Tabulka 9: Tabulka podmínek pro časová okna u trasy 2 (Vlastní zpracování, 2022)

Poslední část modelu, co chybí k naplnění OpenSolver modelu je buňka, která obsahuje následující vzorec účelové funkce: $z = \sum_{i=1}^{n} \sum_{j=1}^{n} t_{ij} x_{ij} + \sum_{i=2}^{n} S_i + \sum_{j=2}^{n} W_j \rightarrow MIN$. V programu Microsoft Excel jde tohoto výpočtu docílit součtem skalárního součinu matice T a X, sumou vektoru S_i a sumou vektoru W_j. Pro skalární součin se využije funkce SUMPRODUCT a pro sumy funkce SUM. Konkrétní excelový vzorec v mé tabulce vypadá takto: **„=SUMPRODUCT(C18:G22;C11:G15)+SUM(L4:L8)+SUM(I29:I33)"**. Navíc se ještě vytvoří buňka s funkcí, která udělá skalární součin matice X a matice vzdáleností (C), díky které vypočítáme krom časové náročnosti i vzdálenost, kterou vozidlo ujede. Ačkoliv se úloha neoptimalizuje za účelem nejkratší, ale nejrychlejší cesty, je důležité zjistit i délku okružní trasy.

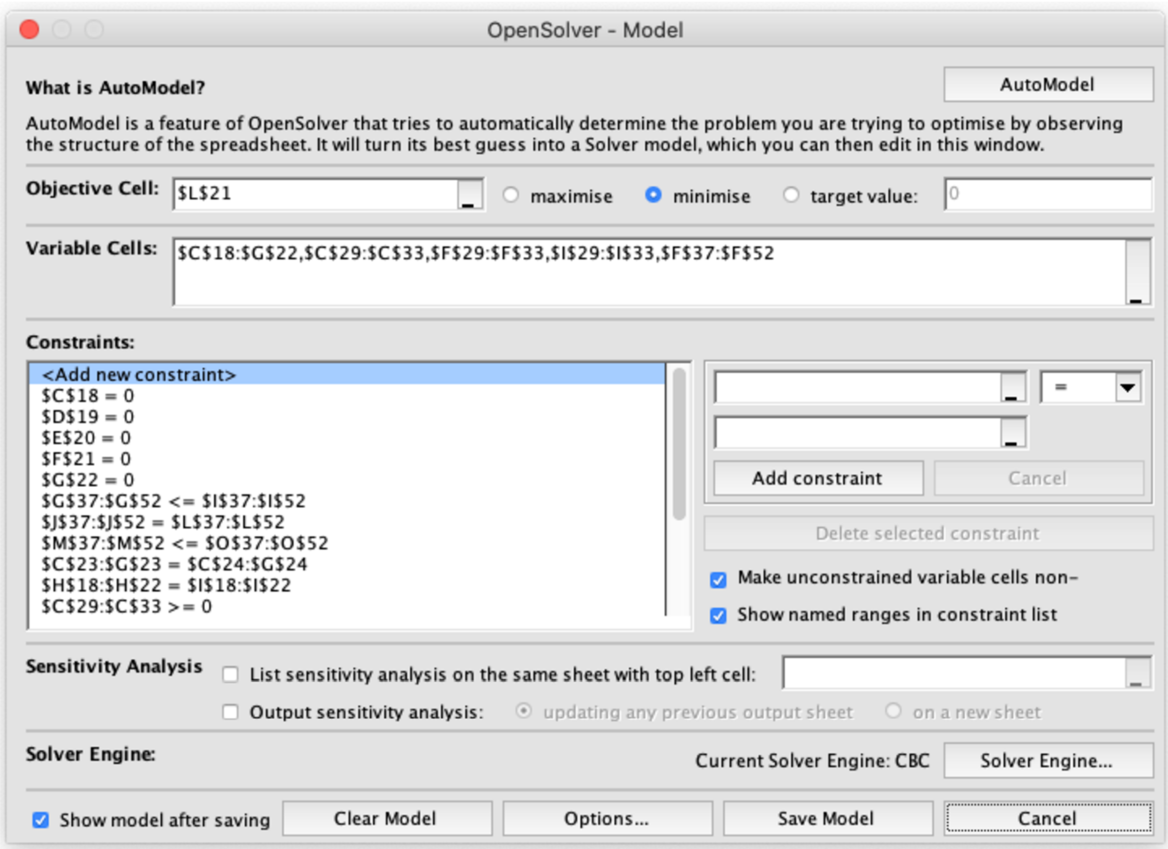
3.5.1 **Vložení matematického modelu do OpenSolveru**

Po vytvoření všech potřebných tabulek, tedy tabulek pro proměnné a podmínky, nastává vytvoření modelu v rozšíření OpenSolver. Po nainstalování rozšíření OpenSolver se uvnitř programu Microsoft Excel otevře záložka *data*. V této záložce má OpenSolver v pravém horním rohu svůj vlastní box, vzhled tohoto boxu je vidět na Obrázku 4. Jak lze vidět na

Obrázku 4, je tam tlačítko s názvem *Model*, na které když se klikne otevře se dialogové okno. Po otevření dialogového okna (Obrázek 4) se ručně zadají jednotlivé součásti modelu formou odkazů na buňky.



Obrázek 4: Box rozšíření OpenSolver uvnitř programu MS Excel (Vlastní zrpacování, 2022)



Obrázek 5: Vytvoření matematického modelu (Vlastní zpracování, 2022)

3.5.1.1 **Vložení účelové funkce**

Do řádku s nadpisem *Objective Cell* se zadá odkaz na buňku obsahující výpočet účelové funkce. Dále je v tomto řádku na výběr ze tří možností *maximise*, *minimise*, *target value*, tedy maximalizovat, minimalizovat a cílová hodnota. Účelová funkce tohoto modelu má za cíl minimalizovat, vybere se tedy *minimise*.

3.5.1.2 **Vložení proměnných**

Do následujícího řádku s nadpisem *Variable Cells*, se jednoduše vyberou odkazy na všechny buňky, které obsahují proměnné modelu, které má solver měnit. Nepatří mezi ně tedy vektor S_i a pomocná proměnná M, ale pouze vektory $\boldsymbol{u}$, a_i, W_j, a matice X.

3.5.1.3 **Vložení omezujících podmínek**

Zbývá ten nejnáročnější krok zadávání modelu a to řádek s nadpisem *Constraints*, do kterého se zadávají všechny podmínky. Tři pole na pravé straně tohoto řádku slouží k zapisování podmínek. První pole je pro zadání levé strany podmínky, druhé nabízí nabídku operátorů „<=“, „=“, „>=“, int (celočíselné) a bin (binární) a třetí pole slouží k zadaní pravé strany podmínky. Je dobré začít těmi méně komplikovanými, které nepotřebují žádnou speciální tabulku uvnitř excelu. Pro podmínku (3.16) patří do levé strany podmínky celá matice X, jako operátor se nastaví bin a pravá strana podmínky zůstane prázdna. Tím se docílí toho, že všechny proměnné v matici X mohou být buď 0 nebo 1. Pro podmínku (3.20) se vytvoří jednotlivé podmínky pro všechny x_{ij}, kde se $i=j$, tedy x_{11}, x_{22}, …, x_{55}. x_{11}, x_{22}, …, x_{66} v případě modelu druhé trasy. Na levé straně podmínky bude x_{ij}, operátorem bude rovná se a do pravé strany podmínky se jednoduše napíše 0. Ke splnění podmínky (3.11) se zadá podmínka s levou stranou obsahující jednotlivé sumy řádků rovné pravé straně o hodnotě 1. To stejné platí pro sloupce u podmínky (3.12). Dále buňkám, které obsahují proměnné typu v_{ij}, S_i, W_j a $\boldsymbol{u}$ se nastaví podmínka, tak aby vždy musely být větší nebo rovny 0. Ke splnění zbytku podmínek stačí vytvořit podmínky pro jednotlivé tabulky z tabulky proměnných (Tabulka 6 pro model trasy 1 a Tabulka 7 pro model trasy 2), kde na levé straně podmínek budou sloupce s levými stranami podmínek, adekvátních operátorů dle tabulek a na pravé straně podmínek sloupce pravých stran. To stejné se uděla i pro tabulky časových oken (Tabulka 8 pro model trasy 1 a Tabulka 9 pro model trasy 2) Na závěr, je dobré nezapomenout zašktnout pole *Make unconstrained variable cells non-negative*, díky čemuž se automaticky vytvoří podmínky nezápornosti.

3.5.1.4 **Uložení a kontrola modelu**

Poslední řádek *Solver Engine* se ponechá beze změny. Je zvolen solver CBC, který je určen pro výpočet lineárních modelů. V případě větších modelů by šel použít např. solver Gurobi.

K uložení vytvořeného modelu stačí kliknout na *Save Model*. Po uložení se dialog zavře a přes původní excel tabulku se zobrazí vizualizace modelu (Obrázek 6). To umožňuje vizuálně zkontrolovat zda jsou všechny podmínky správně přířazeny a nedošlo k chybě při zadávání dat. Na obrázkcích č.6 a č.8 je vidět, že k žádným chybám nedošlo a jsou zadány všechny podmínky správně. V takovém případě lze přistoupit k výpočtu. Výpočet se spustí tlačítkem *Solve* (Obrázek 4).

Matice vzdáleností - Trasa 1					
C=(cij)	**Praha**	**Písek**	**Prachatice**	**Strakonice**	**Příbram**
Praha	0	111	153	114	65
Písek	111	0	45	27	53
Prachatice	153	45	0	36	97
Strakonice	114	27	36	0	57
Příbram	65	53	97	57	0

Matice časů - Trasa 1					
T=(tij)	**Praha**	**Písek**	**Prachatice**	**Strakonice**	**Příbram**
Praha	0	100	150	110	70
Písek	100	0	50	26	55
Prachatice	150	50	0	45	100
Strakonice	110	26	45	0	60
Příbram	70	55	100	60	0

Si - Doba návštěvy	
S1	0
S2	20
S3	20
S4	20
S5	20

Časová okna				
ei <= ai <= li				
ei		**ai**		**li**
		0	=	= 0
120	<=	120	<=	300
120	<=	190	<=	300
180	<=	255	<=	510
180	<=	335	<=	510

xij	**Praha**	**Písek**	**Prachatice**	**Strakonice**	**Příbram**	Suma řádku	
Praha	= 0	1	0	0	0	1	1
Písek	0	= 0	1	0	0	1	1
Prachatice	0	0	= 0	1	0	1	= 1
Strakonice	0	0	0	= 0	1	1	1
Příbram	1	0	0	0	= 0	1	1
Suma sloupce	1	1	1	1	1		
	= 1	1	1	1	1		

Z(MIN)	min 425
Délka trasy	314

u	
u1	0
u2	1
u3	0≤ 2
u4	3
u5	4

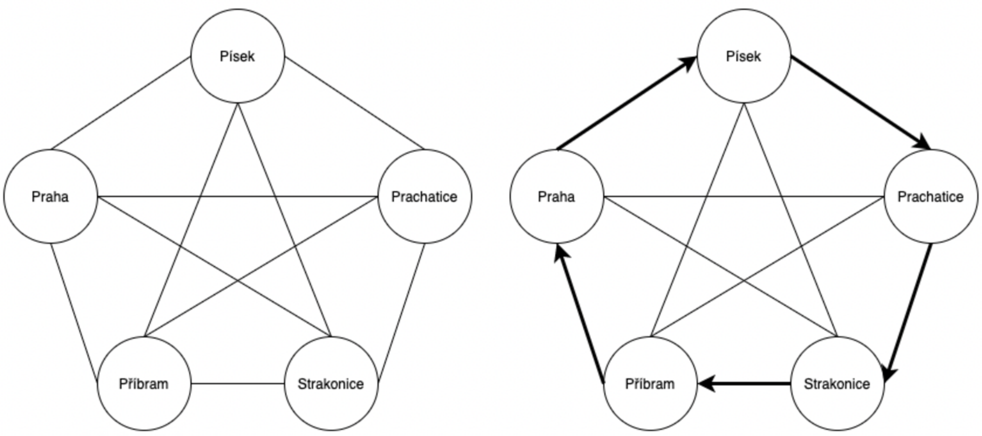
ai	
a1	0
a2	120
a3	0≤ 190
a4	255
a5	335

Wj	
W1	0
W2	20
W3	0≤ 0
W4	0
W5	0

n	5
n-1	4
M	1000
2M	2000

i	j	xij	tij	vij	ui-uj+nxij <= n-1			ai-aj+M*xij+Wj+vij = M-Si-tij			vij <= 2M(1-xij)		
1	2	1	100	0	4	<=	4	900	=	900	0	<=	0
1	3	0	150	1040	-2	<=	4	850	=	850	1040	<=	2000
1	4	0	110	1145	-3	<=	4	890	=	890	1145	<=	2000
1	5	0	70	1265	-4	<=	4	930	=	930	1265	<=	2000
2	3	1	50	0	4	<=	4	930	=	930	0	<=	0
2	4	0	26	1089	-2	<=	4	954	=	954	1089	<=	2000
2	5	0	55	1140	-3	<=	4	925	=	925	1140	<=	2000
3	2	0	50	840	1	<=	4	930	=	930	840	<=	2000
3	4	1	45	0≤ 0	4	<=	4	935	=	935	0	<=	0
3	5	0	100	1025	-2	<=	4	880	=	880	1025	<=	2000
4	2	0	26	799	2	<=	4	954	=	954	799	<=	2000
4	3	0	45	870	1	<=	4	935	=	935	870	<=	2000
4	5	1	60	0	4	<=	4	920	=	920	0	<=	0
5	2	0	55	690	3	<=	4	925	=	925	690	<=	2000
5	3	0	100	735	2	<=	4	880	=	880	735	<=	2000
5	4	0	60	840	1	<=	4	920	=	920	840	<=	2000

Obrázek 6: Finální model pro Trasu 1 s vypočtenými hodnotami (Vlastní zpracování, 2022)



Obrázek 7: Trasa 1 - Graf okružního problému a graf okružního problému s výslednou trasou (Vlastní zpracování, 2022)

Matice vzdáleností - Trasa 2						
C=(cij)	Praha	Chrudim	Pardubice	Hradec Králove	Nový Bydžov	Nymburk
Praha	0	127	119	110	91	54
Chrudim	127	0	12	34	28	84
Pardubice	119	12	0	26	38	76
Hradec Králove	110	34	26	0	27	67
Nový Bydžov	91	28	38	27	0	39
Nymburk	54	84	76	67	39	0

Si - Doba návštěvy	
S1	0
S2	20
S3	20
S4	20
S5	20
S6	20

Matice časů - Trasa 2						
T=(tij)	Praha	Chrudim	Pardubice	Hradec Králove	Nový Bydžov	Nymburk
Praha	0	110	105	100	75	50
Chrudim	110	0	24	45	60	80
Pardubice	105	24	0	35	55	70
Hradec Králove	100	45	35	0	40	65
Nový Bydžov	75	60	55	40	0	45
Nymburk	50	80	70	65	45	0

Časová okna				
ei <= ai <= li				
ei		ai		li
		0	=	0
90	<=	110	<=	360
120	<=	154	<=	510
180	<=	209	<=	510
180	<=	269	<=	420
300	<=	334	<=	510

X=(xij)	Praha	Chrudim	Pardubice	Hradec Králove	Nový Bydžov	Nymburk	Suma řádku	
Praha	0	1	0	0	0	0	1	1
Chrudim	0	0	1	0	0	0	1	1
Pardubice	0	0	0	1	0	0	1	1
Hradec Králove	0	0	0	0	1	0	1	1
Nový Bydžov	0	0	0	0	0	1	1	1
Nymburk	1	0	0	0	0	0	1	1
Suma sloupce	1	1	1	1	1	1		
	1	1	1	1	1	1		

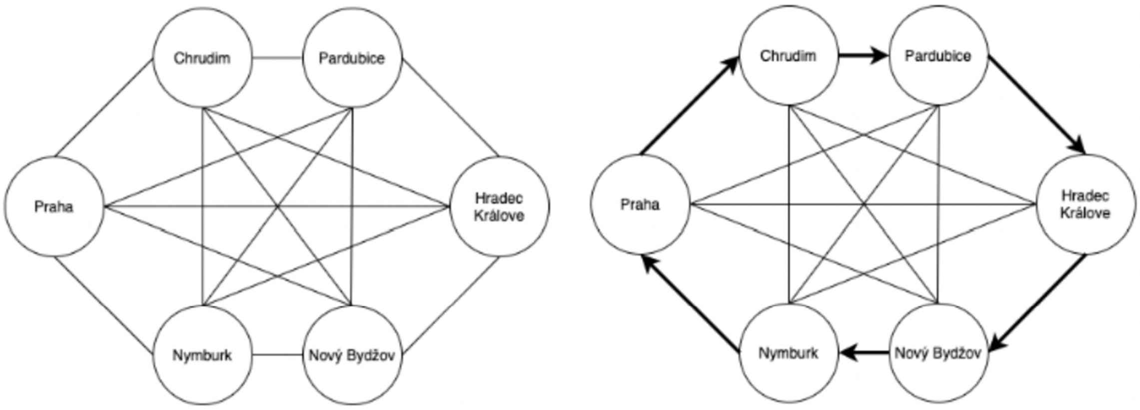
Z(min)	454
Délka Trasy	339

u		ai		Wj	
u1	0	a1	0	W1	0
u2	1	a2	110	W2	0
u3	2	a3	154	W3	0
u4	3	a4	209	W4	0
u5	4	a5	269	W5	0
u6	5	a6	334	W6	0

n	6
n-1	5
M	1000
2M	2000

i	j	xij	tij	vij	ui-uj+nxij <= n-1			ai-aj+M*xij+Wj+vij = M-Si-tij			vij <= 2M(1-xij)		
1	2	1	110	0	5	<=	5	890	=	890	0	<=	0
1	3	0	105	1049	-2	<=	5	895	=	895	1049	<=	2000
1	4	0	100	1109	-3	<=	5	900	=	900	1109	<=	2000
1	5	0	75	1194	-4	<=	5	925	=	925	1194	<=	2000
1	6	0	50	1284	-5	<=	5	950	=	950	1284	<=	2000
2	3	1	24	0	5	<=	5	956	=	956	0	<=	0
2	4	0	45	1034	-2	<=	5	935	=	935	1034	<=	2000
2	5	0	60	1079	-3	<=	5	920	=	920	1079	<=	2000
2	6	0	80	1124	-4	<=	5	900	=	900	1124	<=	2000
3	2	0	24	912	1	<=	5	956	=	956	912	<=	2000
3	4	1	35	0	5	<=	5	945	=	945	0	<=	0
3	5	0	55	1040	-2	<=	5	925	=	925	1040	<=	2000
3	6	0	70	1090	-3	<=	5	910	=	910	1090	<=	2000
4	2	0	45	836	2	<=	5	935	=	935	836	<=	2000
4	3	0	35	890	1	<=	5	945	=	945	890	<=	2000
4	5	1	40	0	5	<=	5	940	=	940	0	<=	0
4	6	0	65	1040	-2	<=	5	915	=	915	1040	<=	2000
5	2	0	60	761	3	<=	5	920	=	920	761	<=	2000
5	3	0	55	810	2	<=	5	925	=	925	810	<=	2000
5	4	0	40	880	1	<=	5	940	=	940	880	<=	2000
5	6	1	45	0	5	<=	5	935	=	935	0	<=	0
6	2	0	80	676	4	<=	5	900	=	900	676	<=	2000
6	3	0	70	730	3	<=	5	910	=	910	730	<=	2000
6	4	0	65	790	2	<=	5	915	=	915	790	<=	2000
6	5	0	45	870	1	<=	5	935	=	935	870	<=	2000

Obrázek 8: Finální model pro Trasu 2 s vypočtenými hodnotami (Vlastní zpracování, 2022)



Obrázek 9: Trasa 2 - graf okružního problému a graf okružního problému s výslednou trasou (Vlastní zpracování, 2022)

3.6 Interpretace Výsledků

Pro trasu 1 (Obrázek 6) dle pořadí míst ve vektoru *u*, je nejrychlejší okruh v následujícím pořadí míst **Praha** => **Písek** => **Prachatice** => **Strakonice** => **Příbram** => **Praha.** Trasa je vyobrazena grafem na obrázku 7. Dle účelové funkce nejrychlejší trasa trvá 425 minut, zbyde tedy řidiči značná časová rezerva 85 minut. Vzdálenost, kterou řidič vozidlem ujede je 314 km. Z proměnné W_2, lze odvodit, že řidič musel čekat v Písku na začátek otevírací doby 20 minut. Pro porovnání trasa, kterou použila společnost měla pořadí míst **Praha** => **Písek** => **Prachatice** => **Strakonice** => **Příbram** => **Praha**, výpočtem tedy bylo ověřeno, že společnost používá optimální trasu.

U trasy 2 byl algoritmem vybrán okruh s místy v pořadí **Praha** => **Chrudim** => **Pardubice** => **Hradec Králove** => **Nový Bydžov** => **Nymburk** => **Praha.** Pořadí navštívení jednotlivých míst je znázorněno na obrázku 9. Optimální okruh dle účelové funkce trvá 454 minut. Okruh je 339 km dlouhý. Řidiči zbývá rezerva 56 minut. Auto nikde nemuselo čekat. Trasa, kterou jelo vozidlo společnosti proběhla v pořadí míst **Praha** => **Chrudim** => **Pardubice** => **Hradec Králove** => **Nový Bydžov** => **Nymburk** => **Praha**. I v případě druhé trasy bylo tedy ověřeno, že firma používá optimální okružní trasu.

4 Závěr

Cílem práce bylo optimalizovat okružní trasy, které slouží k opakované distribuci zboží mezi centrálním skladem společnosti J.G.S. TRADE s.r.o. a maloobchody, které zásobují svým zbožím a porovnání výsledků s trasami které firma k rozvozu aktuálně používá.

Data poskytnuta k řešení problému obsahovaly místa a otevírací doby míst, které musí společnost zavézt zbožím během dvou jednotlivých dní a trasy, kterými realizovala distribuci. Pro první den byly místy Písek, Prachatice, Strakonice a Příbram a pro druhý den to byla Chrudim, Pardubice, Hradec Králove, Nový Bydžov a Nymburk. Pro oba dva dny byl stejný výchozí a konečný bod, a to Praha.

Na základě těchto dat byly nalezeny data potřebné pro výpočet, a to vzdálenosti a délka trvání cest mezi jednotlivými místy za pomoci služby Google Maps.

Data byla následně zanesena do optimalizačního modelu metody jednookruhového okružního dopravního problému s časovými okny, intervaly na obsluhu a intervaly čekání na obsluhu. Tato optimalizační metoda byla aplikována pomocí tabulkového procesoru Microsoft Excel s rozšířením OpenSolver.

Výsledkem optimalizace byla pro první den okružní trasa s následujícím pořadím míst: **Praha** => **Písek** => **Prachatice** => **Strakonice** => **Příbram** => **Praha**. Tato okružní trasa trvala 425 minut a byla dlouhá 314 kilometrů. Pořadí, ve kterém byly místa navštíveny vozem společnosti byla také: **Praha** => **Písek** => **Prachatice** => **Strakonice** => **Příbram** => **Praha.**

Pro druhý den vyšlo pořadí míst optimální okružní trasy takto: **Praha** => **Chrudim** => **Pardubice** => **Hradec Králove** => **Nový Bydžov** => **Nymburk** => **Praha**. Okružní trasa trvala 454 minut a dlouhá 339 kilometrů. V tomto případě byly místa vozidlem firmy opět projety optimální okružní trasou v pořadí míst: **Praha** => **Chrudim** => **Pardubice** => **Hradec Králove** => **Nový Bydžov** => **Nymburk** => **Praha.**

Obě dvě vypočtené optimální trasy vyšly s místy v úplně stejném pořadí, jako je doopravdy jelo vozidlo firmy. Z toho lze udělat závěr, že firma má optimalizaci okružních tras na dobré úrovni a jejich okružní trasy jsou optimální, tedy nejrychlejší.

5 Seznam použitých zdrojů

BROŽOVÁ, Helena a Milan HOUŠKA. *Základní metody operační analýzy*. Praha: Credit, 2002. ISBN 80-213-0951-2.

COOK, William. *Po stopách obchodního cestujícího: matematika na hranicích možností*. Praha: Argo, 2012. Zip (Argo: Dokořán). ISBN 978-80-7363-412-4.

ŠUBRT, Tomáš. *Ekonomicko-matematické metody*. 2. upravené vydání. Plzeň: Vydavatelství a nakladatelství Aleš Čeněk, 2015. ISBN 978-80-7380-563-0.

OpenSolver for Excel. About OpenSolver. OpenSolver: The Open Source Optimization Solver for Excel [online]. 2022 [cit. 2022-03-14]. Dostupné z: https://opensolver.org/

5 best linear programming software for Windows [2022 Guide]. *Windowsreport* [online]. in: windowsreport.com, 2021 [cit. 2022-03-15]. Dostupné z: https://windowsreport.com/

FÁBRY, J., 2006. Dynamické okružní a rozvozní úlohy: disertační práce. VŠE-FIS, Praha.