



ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE

FAKULTA DOPRAVNÍ

Bc. Martin Bil

**AUTOMATIZACE PROCESU TVORBY FEASIBILITY
VE VYBRANÉ SPOLEČNOSTI**

Diplomová práce

2020



K617..... Ústav logistiky a managementu dopravy

ZADÁNÍ DIPLOMOVÉ PRÁCE
(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení studenta (včetně titulů):

Bc. Martin Bil

Kód studijního programu a studijní obor studenta:

N 3710 – LA – Logistika a řízení dopravních procesů

Název tématu (česky): **Automatizace procesu tvorby feasibility ve vybrané společnosti**

Název tématu (anglicky): Automation of the feasibility processes In the selected company

Zásady pro vypracování

Při zpracování diplomové práce se řiďte následujícími pokyny:

- Definice problémové situace v dané společnosti – současný stav, technologie, informační tok
- Návrh automatizace procesu tvorby feasibility v prostředí MS Excel
- Popis zvoleného způsobu řešení, teoretický popis části makra a jeho analýza
- Praktická ukázka využití makra
- Zhodnocení navrženého řešení



- Rozsah grafických prací: podle pokynů vedoucího diplomové práce
- Rozsah průvodní zprávy: minimálně 55 stran textu (včetně obrázků, grafů a tabulek, které jsou součástí průvodní zprávy)
- Seznam odborné literatury: Pernica, P. Logistika pro 21. století. Radix, 2005
Svoboda, V., Latýn, P. Logistika. Vydavatelství ČVUT, 2003

Vedoucí diplomové práce: **doc. Ing. Tomáš Horák, Ph.D.**

Datum zadání diplomové práce: **29. června 2019**
(datum prvního zadání této práce, které musí být nejpozději 10 měsíců před datem prvního předpokládaného odevzdání této práce vyplývajícího ze standardní doby studia)

Datum odevzdání diplomové práce: **18. května 2020**
a) datum prvního předpokládaného odevzdání práce vyplývající ze standardní doby studia a z doporučeného časového plánu studia
b) v případě odkladu odevzdání práce následující datum odevzdání práce vyplývající z doporučeného časového plánu studia

doc. Ing. Tomáš Horák, Ph.D.

vedoucí

Ústavu logistiky a managementu dopravy



doc. Ing. Pavel Hrubeš, Ph.D.

děkan fakulty

Potvrzuji převzetí zadání diplomové práce.

Bc. Martin Bil
jméno a podpis studenta

V Praze dne 29. června 2019

Poděkování

Rád bych poděkoval všem, kteří mi poskytli podklady a svůj čas k vypracování této práce. Zvláště děkuji oddělení logistiky společnosti „XY“ za poskytnuté téma a neúnavnou spolupráci při řešení dílčích problémů. Dále patří velké poděkování panu doc. Ing. Tomášovi Horákovi, Ph.D. za pravidelnou konzultaci a odborné vedení mé diplomové práce od samého začátku.

Prohlášení

Předkládám k posouzení a obhajobě diplomovou práci, kterou jsem zpracoval na závěr studia ČVUT v Praze Fakultě dopravní.

Čestně prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací.

Nemám závažný důvod proti užívání tohoto školního díla ve smyslu § 60 Zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon).

V Praze dne 22. července 2020



podpis

ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE

Fakulta dopravní

AUTOMATIZACE PROCESU TVORBY FEASIBILITY VE VYBRANÉ SPOLEČNOSTI

Diplomová práce

Bc. Martin Bil

ABSTRAKT

Předmětem diplomové práce „Automatizace procesu tvorby feasibility ve vybrané společnosti“ je analyzovat a vyřešit jeden z problémů, který vzniká při plánování tras nákladních automobilů. Práce se věnuje používaným technologiím ve společnosti „XY“ a následnému popisu vytvořeného excelového makra, které automatizuje tvorbu feasibility. Naprogramovaná aplikace je v praktické části testována na několika příkladech.

ABSTRACT

The subject of the diploma thesis "Automation of the process of creating feasibility in a selected company" is to analyze and solve one of the problems that arise in the planning of truck routes. The work focuses on the technologies used in the company "XY" and the subsequent description of the created Excel macro, which automates the creation of feasibility. The programmed application is tested on several examples in the practical part.

KLÍČOVÁ SLOVA

Feasibilita, optimalizace, automatizace feasibility, makro, MS Excel, metaheuristika

KEYWORDS

Feasibility, optimization, feasibility automation, macro, MS Excel, metaheuristics

Obsah

Seznam zkratk	6
1 Úvod	7
2 Řešený problém	8
2.1 O společnosti	8
2.2 Pracovní náplň společnosti „XY“	8
2.2.1 Proces feasibility	11
2.3 Definice problému	12
3 Technologie	14
3.1 Outsourcing	14
3.2 Cross-docking	15
3.3 Just in Time (JIT)	16
3.4 Kanban	17
3.5 Heijunka	19
3.6 Milkrun	19
4 Optimalizační metody	21
4.1 Exaktní přístup	23
4.2 Heuristický přístup	24
4.2.1 Metaheuristický přístup	24
5 Způsob řešení	26
5.1 Vývojové prostředí	26
5.1.1 Visual Basic for Applications (VBA)	27
5.2 Získání dat	27
5.2.1 Skript	30
5.3 Postup optimalizace	33
5.3.1 Řešitel (Solver)	33
5.3.2 Model	33
5.3.3 Omezující podmínky	34
5.3.4 Nastavení evolučního algoritmu	35
5.4 Implementace	37
5.5 Struktura makra	40

5.6	Analýza zvoleného řešení	41
6	Praktické využití makra	43
6.1	Kontrola optimality současných linek	45
6.1.1	Příklad č.1	45
6.1.2	Příklad č.2	48
6.1.3	Příklad č.3	50
6.1.4	Shrnutí.....	51
6.2	Generování trasy pomocí genetického algoritmu	52
6.3	Porovnání způsobů řešení různých algoritmů	54
6.3.1	Metoda hrubé síly (brute-force).....	56
6.3.2	Metoda větví a mezí (branch and bound).....	57
6.3.3	Evoluční algoritmus (Excel Solver)	58
6.3.4	Metoda nejbližšího souseda (nearest neighbor).....	58
6.3.5	Hladový algoritmus (greedy search)	59
6.3.6	Shrnutí.....	61
7	Závěr	63
	Použitá literatura	64
	Seznam obrázků	68
	Seznam grafů	69
	Seznam tabulek	70
	Seznam příloh.....	71

Seznam zkratek

ČR – Česká republika

TSP – Traveling salesman problem

VRP – Vehicle routing problem

GA – Genetický algoritmus

M&G – Software PTV Map and Guide

VBA – Visual Basic for Applications

VB – Visual Basic

FTS – Freight tracking system

MS Excel – Microsoft Excel

NP – Nedeterministicky polynomiální

JIS – Just in sequence

JIT – Just in time

FIFO – First in - First out

PS – Planning system

FTS - Freight tracking system

TSP – Traveling salesman problem

VRP – Vehicle routing problem

1 Úvod

V rámci diplomové práce „automatizace procesu výroby feasibility ve vybrané společnosti“ jsem řešil problém mezinárodní firmy, která bude v celé práci pojmenovávána jako společnost „XY“.

„XY“ zajišťuje významnému světovému automobilovému výrobcí logistické služby v podobně přepravy součástek mezi dodavateli a výrobními halami v pravidelných intervalech. Jedním z procesů, který při řízení vzniká je tzv. feasibility, která mimo jiné zajišťuje, že pořadí dodavatelů na trase okružních linek je z hlediska určitých faktorů optimální.

Celý problém byl původně z velké části řešen manuálně, což zabralo spoustu času a znamenalo snížení produktivity práce. Hlavním úkolem diplomové práce bylo navrhnout a naprogramovat systém (makro) v prostředí MS Excel, který by proces automatizoval a zároveň vygeneroval výsledný formulář.

Práce obsahuje popis problému a technologií, které se běžně ve společnosti „XY“ používají pro efektivní fungování logistického řetězce. Dále jsem se zaměřil na teoretický základ optimalizační metody, jenž je v makru používána. Samotný program je rozdělen do tří částí a jeho funkčnost popsána SWOT analýzou.

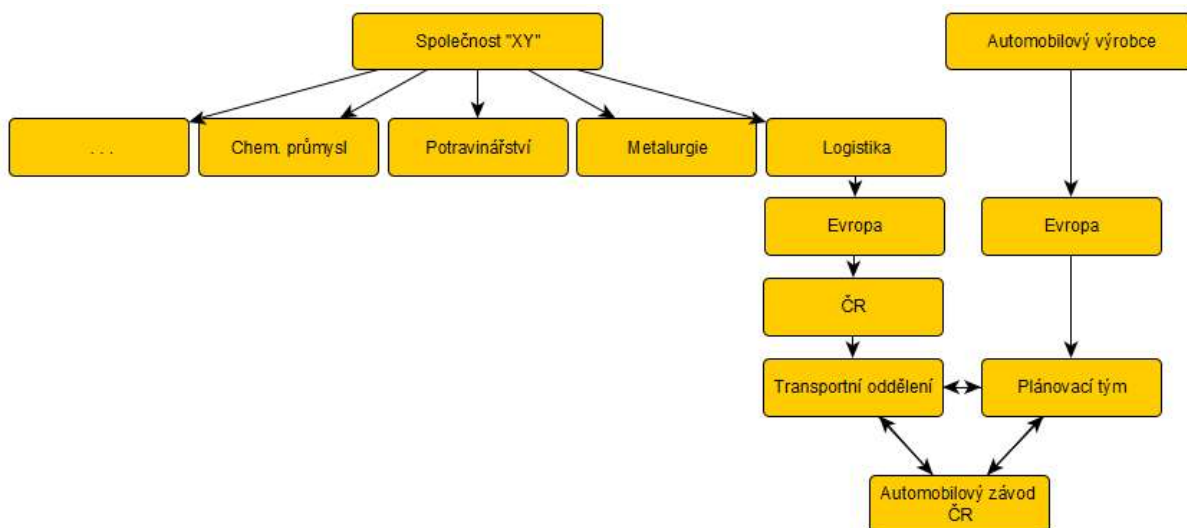
Praktická část zahrnuje následné testování a ukázkou způsobu, jak makro v praxi využívat. Nejdříve se pokusím zjistit, zdali jsou současné náhodně vybrané linky vedeny po optimální trase. Vzhledem k tomu, že optimalizace může být prováděna pomocí několika různých algoritmů, porovnáám nejznámější metody a vyberu jednu, která se jeví v rámci zadaného problému nejlépe. Zvolený algoritmus nakonec podrobím detailnější analýze pro možnost využití jako generátor optimální trasy v rámci tvorby feasibility.

2 Řešený problém

2.1 O společnosti

Firma „XY“ je zahraniční společnost s tržbami v jednotkách bilionů korun, která své podnikatelské portfolio směřuje do několika odvětví jako metalurgie, chemický průmysl, energetika nebo logistika. Ve světě existuje více než 100 poboček, které se liší různým zastoupením služeb. V rámci divize logistiky společnost poskytuje služby jedné z největších automobilek na světě. Hlavním úkolem divize operující v ČR je zajistit logistické služby nedalekému automobilovému závodu.

Transportní oddělení pobočky firmy „XY“ zajišťuje přepravu dílů a součástí mezi dodavateli těchto komponentů a konkrétním výrobním závodem automobilky. Samotná společnost „XY“ pouze organizuje přepravu a napomáhá automobilce při plánování. Nevlastní a nezaměstnává žádná nákladní vozidla nebo řidiče, ale smluvně najímá dopravce pro své potřeby. Struktura je zaznamenána na obrázku 1.



2.2 Pracovní náplň společnosti „XY“

Automobilka disponuje plánovacím týmem pro celou Evropu. Ten má za úkol zajistit nepřetržitý chod materiálu mezi továrnami a dodavateli. Jednou z činností je vytváření okružních tras pro nákladní vozidla s jedním či více nakládkových míst tak, aby jednotlivé trasy byly z hlediska územního, časového, objemového a z pohledu pravidelnosti optimální. Trasa je během vytváření plánovacím týmem automobilky na základě těchto faktorů připomínkována firmou „XY“, která s týmem spolupracuje a snaží se po celou dobu napomáhat optimalizaci. Dochází

ke kontrole proveditelnosti (feasibility) trasy, po které je linka uvedena do provozu skrz společnost „XY“. Všeobecnou náplň práce firmy „XY“ lze shrnout v následujících bodech:

- Kontrola plánu vytvořeného plánovacím týmem automobilky
 - Časové hledisko – trasa musí brát v potaz Evropskou dohodu o práci osádek vozidel v mezinárodní silniční dopravě AETR, (pravidla jsou popsány v tabulce č.1), zákazy jízd o svátcích či ostatních např. nevýrobních dnech továren
 - Struktura linek – vhodnost a správnost uspořádání jednotlivých uzlů na linkách zajišťující efektivní provoz logistického řetězce
- Zpracování plánu pro dopravce
 - Komunikace s dopravci a jejich zavedení do společného systému pro sdílení informací a tracking
 - Zpracování fakturačních podkladů automobilky a dopravců
- Vypracování feasibility
 - Dokument, který obsahuje konkrétní informace o lince jako body trasy, časy nakládek/vykládek, pracovní čas řidiče, počet řidičů a tahačů nebo mapu zobrazující průjezdnou trasu. Na jeho základě jsou poskytovány údaje dodavatelům.

Tabulka 1 – Pravidla AETR. [2]

Denní doba řízení	max. 9 hodin (možné zvýšení 2x týdně na 10 hod.) mezi dvěma odpočinky
Týdenní doba řízení	max. 56 hod (10 + 10 + 9 + 9 + 9 + 9)
Týdenní pracovní doba Nař. vlády č.589/2006 Sb. §5 odst. 1	48 hod., lze prodloužit na max. 60 hod, pokud za 26 po sobě jdoucích týdnů nebude překročena průměrná pracovní doba 48 hod. (včetně ostatních pracovních činností (bez přesčasů))
Celková doba řízení za období dvou po sobě jdoucích týdnů	Max. 90 hod
Přestávka v řízení	nejpozději po 4,5 hodinách řízení nejméně 45 minut. Lze rozdělit na trvalí nejméně 15 min
Běžná denní doba odpočinku	nejméně 11 hod v průběhu 24 hod. od skončení předchozí doby odpočinku
Rozdělení běžné denní doby odpočinku	při prodloužení na nejméně 12 hod lze rozdělit až do 3 úseků, přičemž jeden úsek musí být nejméně 8 souvislých hodin
Přerušení běžné denní doby odpočinku na trajektu	1x nejvýše na 2 hod. za podmínek uvedených v čl.9 nařízení. Řidič však musí mít během odpočinku dispozici lůžko nebo lehátko
Zkrácená denní doba odpočinku	možné zkrácení max. 3x týdně až na 9 hod. s tím, že dojde do konce týdne k odpovídajícímu vyrovnání
Denní doba odpočinku (nejméně 2 řidiči)	8 hodin za každé časové období 30 hod
Běžná týdenní doba odpočinku	nejméně 45 hod. včetně jedné denní doby odpočinku
Zkrácená týdenní doba odpočinku	nejméně 36 hod (je-li čerpána v místě obvyklého odstavení vozidla) nebo nejméně 24 hod. (mimo místo obvyklého odstavení vozidla), s vyrovnáním do konce 3 následujícího týdne
Začátek týdenního odpočinku	Po nejvýše šesti denních dobách řízení. (v případě mezinárodní osobní příležitostné dopravy po nejvýše 12denních dobách řízení)
Povinnost řidičů předložit záznamové listy ke kontrole	z běžného týdne a posledního dne týdne předcházejícího, během něhož řídili

2.2.1 Proces feasibility

Tvorbu feasibility lze rozdělit do několika kroků, jedná se zejména o komunikaci mezi dodavateli, dopravci, společností „XY“ a automobilkou.

1. Plánovací tým automobilky požádá o feasibility a zadá dodavatele na vybranou okružní linku
2. „XY“ připraví feasibility (formulář v excelu)
3. „XY“ pošle feasibility dopravci
4. Dopravce dá zpětnou vazbu „XY“
5. „XY“ implementuje zpětnou vazbu a znovu si vyžádá kontrolu dopravcem, dokud ji nepotvrdí
6. „XY“ pošle informace plánovacímu týmu automobilky
7. Plánovací tým automobilky dá zpětnou vazbu, implementuje ji do systému PS
8. „XY“ tento krok zkontroluje

2.2.1.1 Informační systémy

Pro výměnu a správu informací mezi dopravci, dodavateli, plánovacím týmem automobilky a společností „XY“ existuje několik systémů. Jejich propojení je zobrazeno na obr. 2.

PS (Planning system)

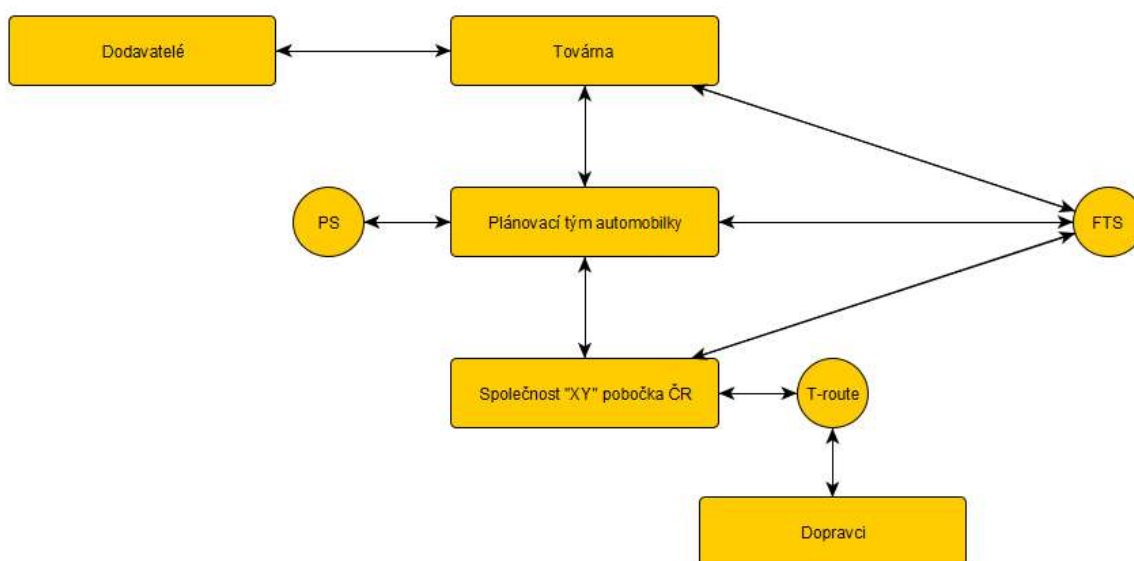
- Jedná se o systém automobilového výrobce, do kterého je po optimalizaci manuálně přepsána feasibility. PS propojí data s dalšími částmi logistického řetězce jako crossdock nebo výrobní závod.
- Dále lze sledovat jednotlivé dodávky (manifesty) zboží od dodavatelů a informovat kdy a na které lince se bude manifest nakládat. Kontroluje se rovnoměrnost nakládky v čase a vytížení návěsu nákladního vozidla.
- V prostředí programu lze zobrazit pohyb návěsu s manifesty v reálném čase a omezení typu zakazy jízd, nepracovní dny továren či dodavatelů.

FTS (Freight tracking system)

- Aplikace sloužící k řízení abnormalit (zpoždění, poškození nákladu atd.) při obsluze dodavatelů a v reálném čase udává pozici návěsů pomocí GPS. Do systémů má přístup „XY“, plánovací tým automobilky, závody nebo překladiště. Mimo výše zmíněné informace se v FTS zobrazuje aktuální status trasy (route), její identifikační číslo nebo konečný bod.

T-Route

- Systém společnosti „XY“ zobrazuje časový plán nakládek u dodavatelů nebo vykládek v crossdocku či továrně. Jsou zde uloženy kontakty na dodavatele, SPZ, otevírací doba atd., přičemž přístup do aplikace mají pouze dopravci.
- Do T-Route jsou průběžně vyplňovány reálné časy příjezdů/odjezdů, informace o návěsu, vozidle či řidiči. Tímto lze sledovat odchylky od plánu a pružně na ně reagovat.



Obrázek 2 – Tok informací mezi systémy v logistickém řetězci. [1]

2.3 Definice problému

V současné době je celý proces feasibility prováděn manuálně. Ačkoliv jsou použity podpůrné programy jako M&G pro kalkulaci trasy (mýto, podmínky AETR atd.) nebo internetové mapové aplikace, není proces automatický a snižuje produktivitu práce.

Největší problém tkví ve volbě vhodné trasy. Jednotliví dodavatelé na linkách jsou zvoleni plánovacím týmem automobilky, a tak společnosti „XY“ zbývá vytvořit linku a překontrolovat všechny náležitosti (feasibilitu). Trasa je na základně určitých parametrů odhadována ručně a nelze zaručit, že je optimální. Proto je snaha o automatizaci a optimalizaci vytvořením programu, který by navrhl nejlepší řešení na základě určitých podmínek.

Průměrně se v dodavatelském řetězci vyskytuje několik desítek linek, s tím, že jsou často měněny v závislosti na poptávce. Může dojít k úbytku či rozšíření počtu dodavatelů, zvýšení frekvence linky atd. Počet dodavatelů na jedné trase se liší od jednoho po více než 15. Je-li

počet dodavatelů nízký, lze trasu za určitých podmínek namodelovat ručně, při vyšším počtu už je to však obtížné a daleko efektivnější je použít optimalizační software.

Úkolem je tak navrhnout řešení, které by za minimálních nákladů zrychlilo proces feasibility a nabídlo optimální trasu mezi crossdockem a dodavateli.

3 Technologie

V rámci poskytování logistických služeb společnost společně s automobilkou využívá několik technologií. Jedná se o nástroje sloužící k zefektivnění výroby a maximálnímu snížení nákladů.

3.1 Outsourcing

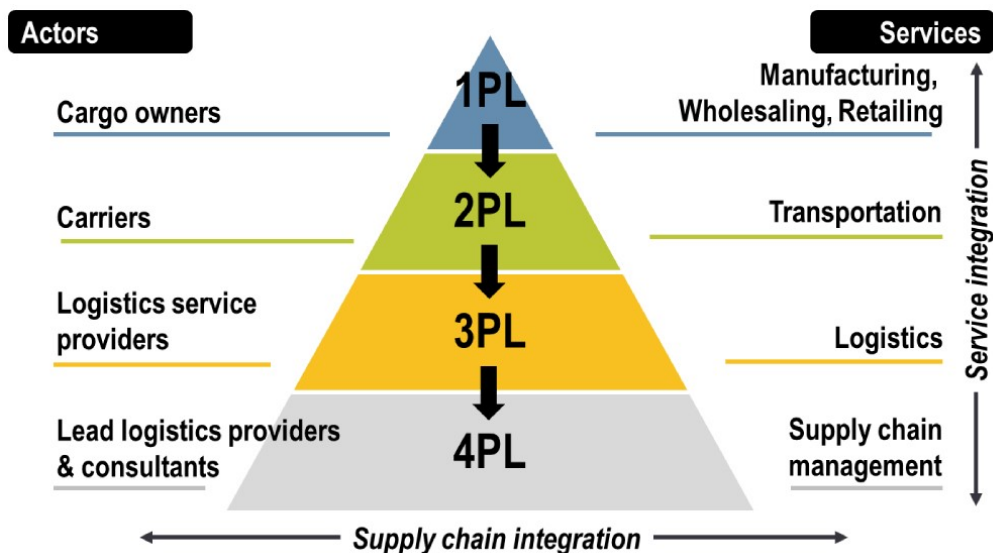
K outsourcingu dochází při smluvnímu svěření určité podnikové činnosti do rukou externí společnosti. Firmy zajišťující tuto službu jsou profesionály se zkušenostmi ve svém oboru, a tak dokážou značně ušetřit náklady svěřenému podniku. Smlouvou přechází zodpovědnost za danou činnost na outsourcingovou společnost, díky čemuž se outsourcovaná firma může soustředit na své důležité interní záležitosti. [3]

Důvody vedoucí k outsourcingu mohou být nízká frekvence určité činnosti, nedostatečné know-how, koncentrace na jiné úseky podniku, celkové zvýšení úrovně logistických služeb nebo pomoc s předvídatelností nákladů. [4]

Automobilka využívá outsourcingu společnosti „XY“ v poskytování poradenství a zajišťování logistických služeb. Rozsah pravomocí lze zařadit na úroveň 3PL.

V Logistice se setkáváme zejména s outsourcingem přepravy, skladování, ale i kompletního logistického řetězce. Úroveň rozsahu poskytování služeb lze rozdělit na hlavní tři stupně. [5]

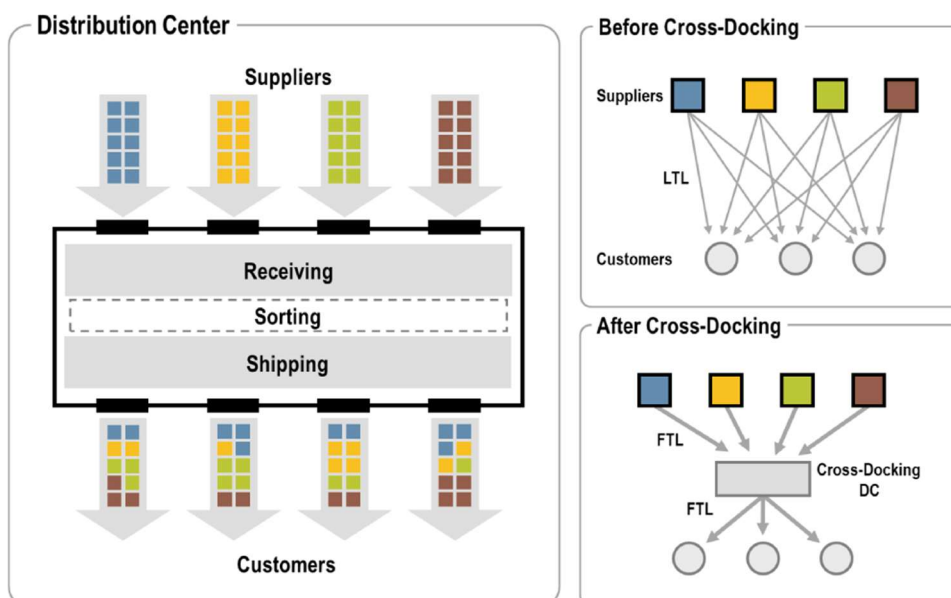
- **Second-party logistics (2PL)** – Společnost je propojená na úrovni zprostředkování jednotných specializovaných služeb jako doprava nebo skladování. Tato úroveň je nejtypičtější u malých firem.
- **Third-party logistics (3PL)** – Outsourcing v podobně převzetí kontroly části nebo celého dodavatelského řetězce. Firma (třetí strana) „*se zabývá jak přímým dopravováním zboží, tak organizací toku zboží, plánováním, péčí o zákazníka a podobnými administrativními záležitostmi.*“ [6] Často uplatňováno ve velkých společnostech např. automobilkách.
- **Fourth-party logistics (4PL)** – Kompletní převzetí kontroly logistického řetězce. Jediný partner zodpovídá za celý design optimalizace procesů a jeho výsledky.



Obrázek 3 – Úrovně outsourcingu. [7]

3.2 Cross-docking

Jedná se o technologii užívání konsolidačních „meziskladů“, která snižuje náklady a čas distribučního řetězce. Snahou je minimalizovat přebytečné skladování a manipulaci se zbožím. Společnost „XY“ disponuje několika sklady v Evropě, optimálně rozmístěnými v prostoru, do kterých proudí zásilky od dodavatelů ze všech směrů. Náklad je zde zdržen v řádech desítek hodin. „Zboží se ve skladu fyzicky nehromadí, ale pouze jím prochází. Tím je zajištěna rychlost,



Obrázek 4 – Ukázka implementace technologie Cross-docking. [8]

nákladová efektivita, umožněná minimální manipulací a skladováním.“ [9]

Úkolem crossdocku je jednotlivé příchozí zboží konsolidovat do ucelených zásilek, které jsou okamžitě vypraveny do konkrétního cíle (továrny). Odpadne tak velké množství neefektivních přeprav mezi dodavateli a jednotlivými výrobny, kde je produkt potřeba. Při budování těchto „meziskladů“ je kladen důraz na počet nákladních ramp, které zajišťují rychlý odbyt a také konstrukci budov, u kterých je snahou minimalizovat podpěrné sloupy pro vyšší užitnou plochu. Největší konsolidační sklady v České republice dosahují užité plochy v řádech desítek tisíc metrů čtverečních, počet nákladních ramp se pohybuje i okolo stovky. Technologie klade velký důraz na kvalitu infrastruktury a systém plánování. Může tak dojít např. k zahlcení docku způsobeným nepravidelnou dodávkou zboží jednoho z dodavatelů.

3.3 Just in Time (JIT)

Metoda, mnohdy nazývána spíše filozofií má kořeny v japonské automobilce Toyota. Typickými znaky je plánování a výroba na objednávku (tzv. push systém), eliminace zásob a skladování, které zvyšuje náklady, a naopak snižuje efektivitu výroby. Aby nedocházelo k hromadění či nedostatku materiálu pro výrobu, měly by dodávky dorazit „právě v čas“. V praxi však často dochází k fluktuaci výrobního řetězce, a tak většina firem udržuje minimální pojistné zásoby pro případ nahodilého výpadku. Jako podpůrný nástroj k dosažení JIT slouží např. technologie Kanban. [10]

Požadavky

- Kvalitní dopravní infrastruktura pro zamezení výpadků zásobování
- Úzce spolupracující a proškolený personál s vrcholným managementem
- Společná koordinace a dlouhodobá spolupráce s lokálními dodavateli
- Implementace IT systémů pro okamžitou kontrolu a reakci v celém logistickém řetězci
- Eliminace skladování
- Vysoká frekvence dodávek v malých dávkách
- Nízká heterogenita produktů a stabilní poptávka

Výhody

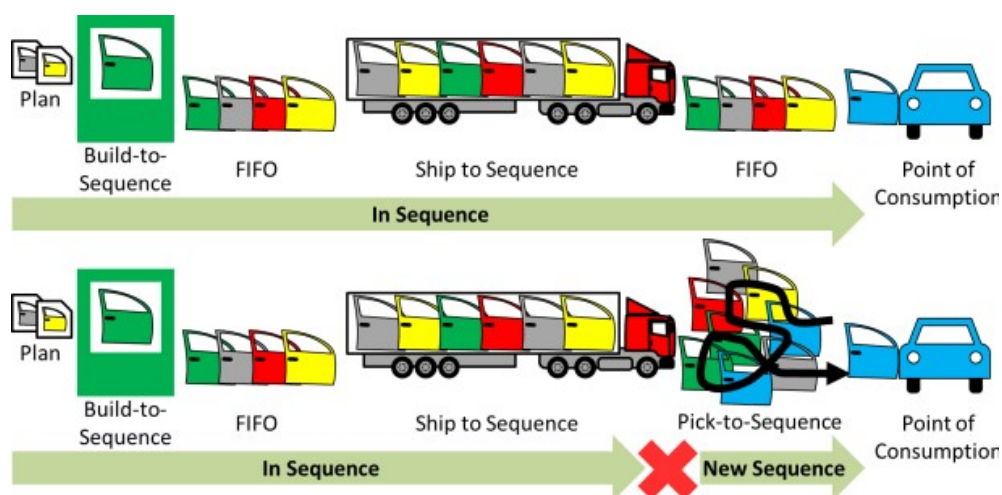
- Redukce skladování – snížení nákladů
- Zrychlení výrobního času
- Zvýšení produktivity práce a reakce na změnu poptávky
- Zvýhodnění oproti konkurenci
- Snížení množství odpadu

Nevýhody

- Riziko vzniklé krátkodobým plánováním výroby (výpadky zásobování, poptávky)
- Dodavatelé musí být v kontaktu se společností – nutnost pořízení IT systémů
- Budování dlouhodobých vztahů a strategie s dodavateli
- Zvýšení frekvence dodávek a tím vzniklých nákladů

Jednou z pokročilejších forem JIT je **Just in sequence (JIS)**. Zboží (materiál) v nákladních vozidlech je při cestě do továrny (nebo předešlém kroku ve výrobě) uloženo v pořadí, v jakém bude postupně přicházet na výrobní linku. Obvykle ke konsolidaci ve správném pořadí dochází v již zmíněném crossdocku.

Souběžně s JIS se uplatňuje také **FIFO princip** (First In – First Out), při kterém je do výroby použit materiál, jenž dorazil do továrny nejdříve. Klíčem ke správnému chodu výroby je předcházení přerušení stanovené sekvence produkce. [11]



Obrázek 5 – Uplatnění JIS a FIFO v praxi s vyobrazením rizika při přerušení sekvence. [12]

3.4 Kanban

Snahou tohoto systému je harmonizace průběhu výroby s materiálovým tokem. Výroba by měla být řízena objednávkou a tím redukováno neefektivní tvoření zásob. Vychází z metodiky „Štíhle výroby“ (Lean manufacturing), která má za úkol zjednodušit, zefektivnit a optimalizovat pracovní a výrobní podmínky pro dosažení maximálního zisku. Kanban byl vyvinut v polovině 20. století automobilkou Toyota a v překladu znamená „štítek“ nebo „karta“. [13]

Samotný Kanban je jedním z nástrojů k dosažení JIT. Jednotlivé součástky ve výrobě se shlukují ve formě „krabic“ s identifikačním štítkem (Kanbanem), který obsahuje dodavatele/zákazníka, číslo dílu, identifikační kód, popis nebo množství. Při výrobě jsou

jednotlivé kanbanové kontejnery spotřebovávány. Jakmile dojde k vyprázdnění, je vyslán signál společně s kanbanem k dodavateli (nebo doplňovacímu místu), který kontejner opět naplní a pošle zpět. V praxi tak často existuje rezerva několika kontejnerů, která je při poklesu pod určitou mez automaticky doplňována. Vzorec pro výpočet množství kanban karet má tvar:

$$N = \frac{D \cdot (T_w + T_p) \cdot (1 + \alpha)}{C} \quad [14]$$

N – Počet kanban karet

D – Odbyt za časovou jednotu

T_w – Čas čekání na dávku kanban (v desetínách dne)

T_p – Čas zpracování dávky dílců kanban (v desetínách dne)

C – Kapacita kanban nosiče

α – Bezpečnostní koeficient

Díky implementaci kanbanu dochází k vyšší obrátkovosti zásob, dostupnosti materiálu, zefektivnění výroby či lepšímu využití skladových prostor. Pokud je u jednoho dodavatele nakládáno více součástí (ve více přepravech), je ve společnosti „XY“ celkový náklad nazýván **manifest**. Zpravidla se jedná o jednu paletu materiálu. [13]

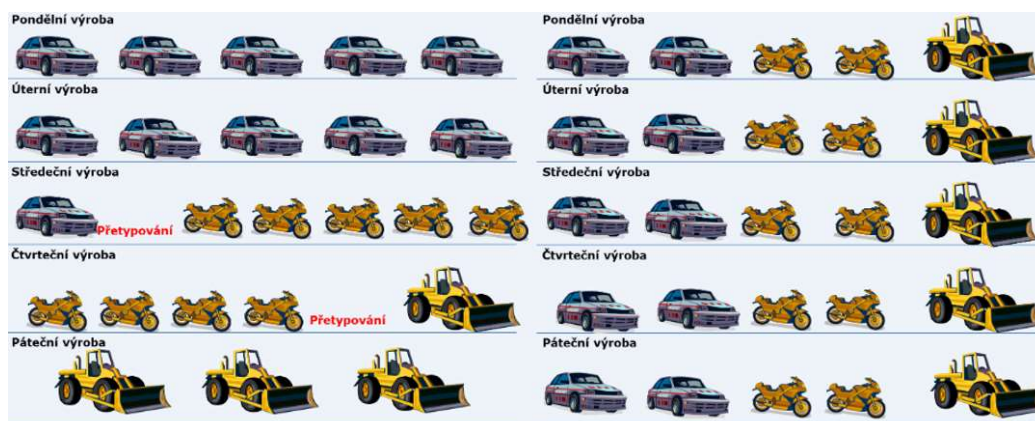


Obrázek 6 – Zásobníky kanban připraveny k převozu na výrobní linku. [15]

3.5 Heijunka

Úkolem heijunky je pokusit se navrhnout takový výrobní plán (produktový mix), „který nevyhovuje jen aktuálním potřebám, ale počítá i s dlouhodobějším předpokladem poptávky.“

[16]



Obrázek 7 – Původní plán v porovnání s plánem řídicím se heijunkou. [17]

Jedná se o jeden z mnoha nástrojů „štíhlé výroby“. V praxi do výroby vstupují výrobky dle požadavku zákazníka. S tím dochází k neefektivním prostojům při přenastavování strojů a dalších věcí způsobených nepředvídatelností poptávky. Proto je stanoven výrobní mix, který se bude pravidelně opakovat každý den. Jedná se o kompromis mezi flexibilitou, stabilitou a předvídatelností. [18]

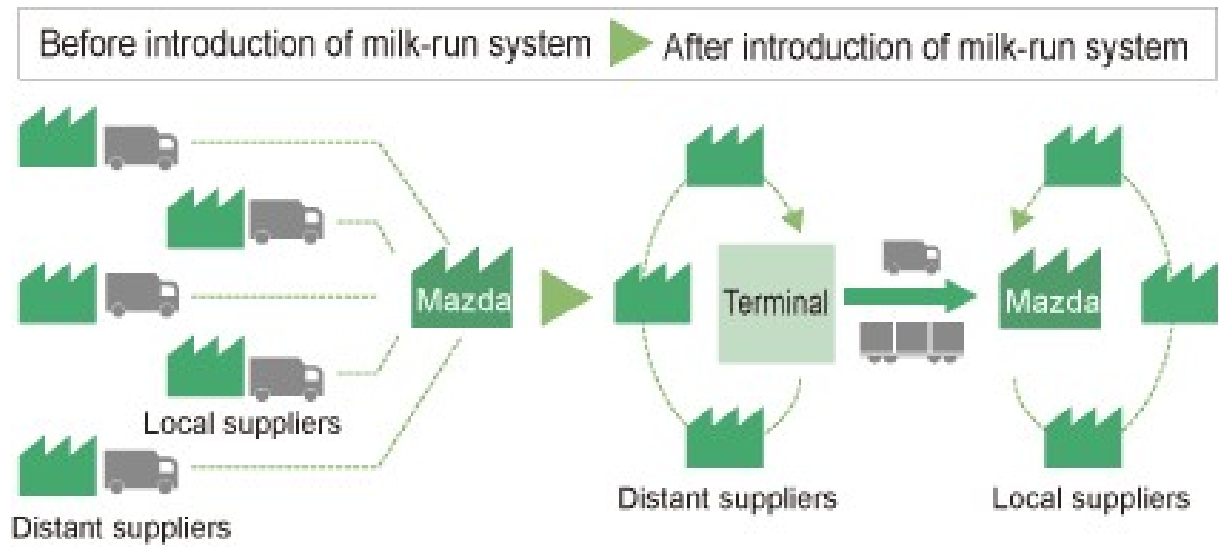
3.6 Milkrun

Počátky milkrunu spadají do Anglie, kde bylo potřeba pravidelně vyzvednout mléko z farem a na oplátku rozvést zpět prázdné sklenice. Místo toho, aby každý dodavatel vysílal kamion se zbožím jednomu zákazníkovi jsou všichni dodavatelé obslouženi jedním kamionem zákazníka (obr. 8). Tím se výrazně snižují náklady na přepravu. Milkrun se uplatňuje nejen v tzv. „parts“ logistice (logistice dílů a součástí k výrobě mezi dodavateli a továrnou), ale i ve výrobě. V továrnách tato technologie často autonomně zásobuje výrobní linku přepravkami kanban. [19]

Společnost „XY“ využívá milkrun pro svoz součástí od dodavatelů. Výchozím bodem pro okružní trasu (route) je crossdock, z kterého následně vyjíždí konsolidovaný tahač s návěsem do fabriky. Pro cestu z crossdocku do montovny se používá označení **main route**.

Dalším druhem linek jsou **speciály**, které vyjíždějí v mimořádných situacích. Jedná se často o dodávkové automobily, schopné okamžitě reagovat na výpadek materiálu. Jejich trasa je

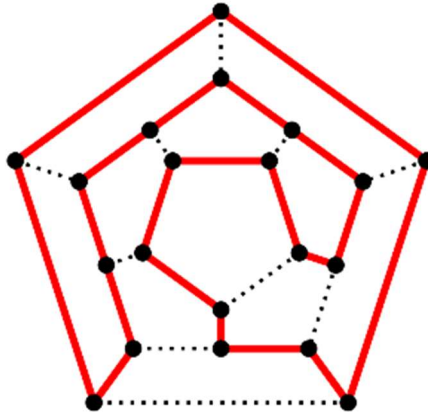
dopředu neplánovaná a obvykle se skládá z jedné návštěvy dodavatele, jehož součástky ve výrobě právě chybí.



Obrázek 8 – Společnost před a po zavedení technologie milkrun. [20]

4 Optimalizační metody

Celý problém lze interpretovat jako úlohu obchodního cestujícího (Traveling salesman problem) zkráceně TSP. Jedná se o optimalizační problém, jehož cílem je v ohodnoceném úplném grafu najít minimálně ohodnocenou hamiltonovskou kružnici (HK). Graf, který obsahuje HK se nazývá hamiltonovský. Problém obchodního cestujícího zmínil již roku 1759 Švýcarský matematik Leonhard Euler. [21] [22]



Obrázek 9 – Graf obsahující hamiltonovskou kružnici. [23]

Abychom poznali, zdali je graf hamiltonovský, musí platit alespoň jedna ze tří následujících podmínek:

Diracova podmínka

- Stupeň každého vrcholu (počet incidujících hran) je větší než $n/2$, kde n je počet vrcholů.

Oreho podmínka

- Je-li pro každou dvojici vrcholů v grafu $n \geq 3$, které nejsou spojeny hranou, součet jejich stupňů alespoň n (kde n je počet vrcholů).

Pósova podmínka

- V grafu $n \geq 3$, kde n je počet vrcholů, pro každé přirozené číslo $k < (n/2)$ je počet vrcholů, jejichž stupeň nepřevyšuje k , menší než k . [21]

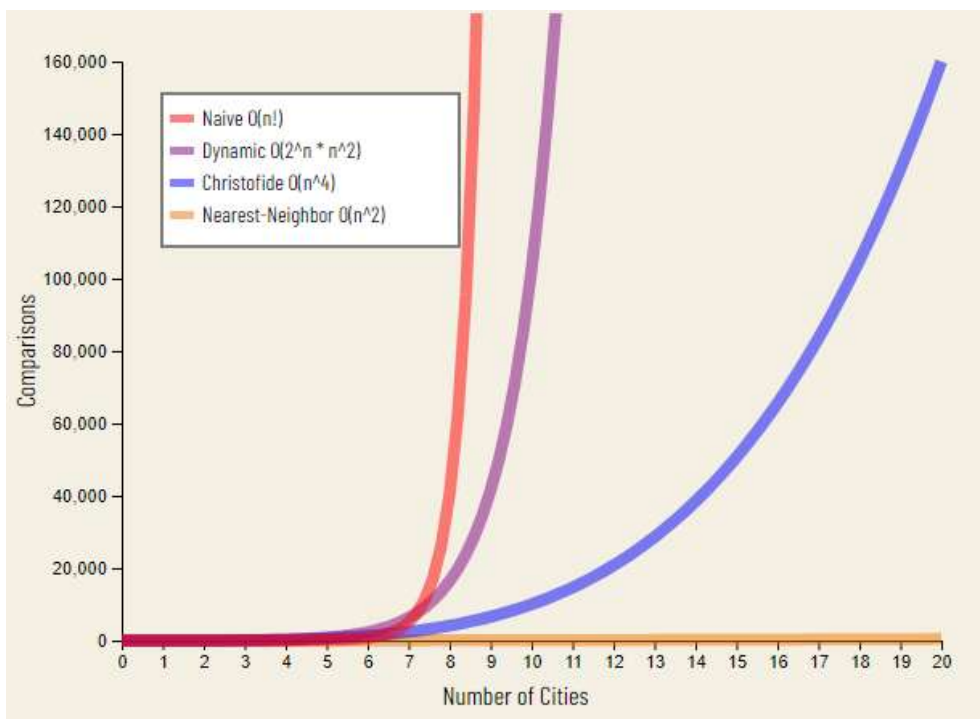
Za předpokladu, že silniční síť s jednotlivými vrcholy tvoří kompletní graf, jsou všechny tři podmínky splněny.

Úkolem je tedy navrhnout takovou okružní trasu, která obslouží všechny vrcholy z daného výchozího bodu právě jednou tak, aby suma minimalizačního kritéria (vzdálenost, čas nebo náklady) byla minimální.

Z TSP vychází mnoho složitějších optimalizačních úloh. Vícenásobný obchodní cestující (VRP) popisuje problém okružních jízd. Existuje množina obsluhovaných bodů včetně výchozího depa a množina vozidel. Vozidla jsou však omezena kapacitou a neexistuje vozidlo, schopné obsloužit všechny vrcholy grafu najednou. Zároveň však alespoň jedno vozidlo dokáže obsloužit kterýkoliv vrchol. Snahou je tak obsloužit vrcholy právě jedním vozidlem za předpokladu minimálních nákladů (vzdálenosti) a z toho plynoucího počtu obsluhujících vozidel. Další varianty VRP berou v potaz časová okna nakládky/vykládky v jednotlivých bodech, v jiných úlohách se vozidla nemusí vracet do počátečního depa, či mohou vykonat více než jeden okruh. [24]

Úloha obchodního cestujícího najde kromě původního cíle také uplatnění v širokém odvětví. Ve strojírenství se často používá pro minimalizaci pohybu zařízení pro vrtání děr do desek plošných spojů. S obdobným cílem se používá v astronomii, kde se snaží zjistit pořadí sledovaných objektů tak, aby se posunem a nastavením teleskopu strávilo co nejméně času. [25]

Optimalizační problém obchodního cestujícího lze řešit třemi způsoby – exaktně, heuristicky nebo metaheuristicky. TSP patří do NP (nedeterministicky polynomiální) – těžké třídy složitosti.



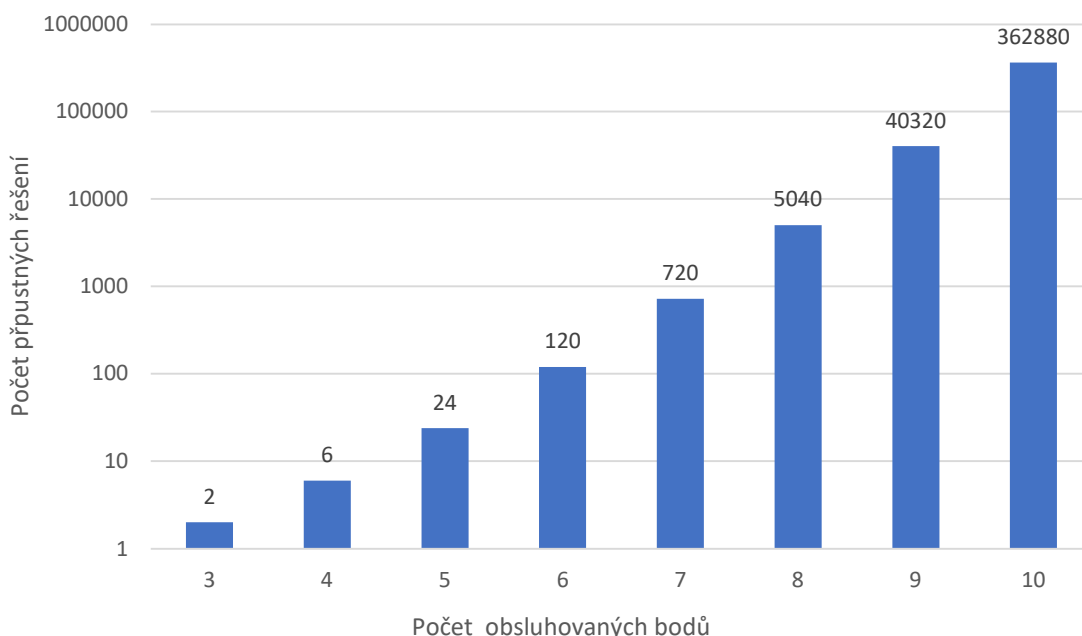
Obrázek 10 – Počet operací při hledání optima různými metodami v závislosti na počtu obsluhovaných měst [27]

Neexistuje tedy žádný polynomiální algoritmus, „který by našel libovolné řešení, které je nejhůře k -násobkem optimálního řešení.“ [26] Proto je důležité zvolit správnou metodu s přihlédnutím na velikost množiny vrcholů. V další části se pokusím přiblížit základní přístupy k řešení a popsat nejznámější používané algoritmy. [25] [26] [27]

4.1 Exaktní přístup

Exaktními metodami se dobereme požadovanému výsledku, tedy optima. Zaplatíme však daň v podobě času, který je často klíčový při řešení optimalizačních úloh. [28]

- **Metoda hrubé síly (brute-force)** počítá a porovnává všechny možné permutace tras, ze kterých vybere optimální řešení. Problémem je časová a paměťová náročnost celého algoritmu. Počet vygenerovaných tras je roven $(n - 1)!$, což faktoriálně zvyšuje množinu přípustných řešení.



Graf 1 – Závislost počtu přístupných řešení na počtu vrcholů grafů. Zdroj: vlastní zpracování

- **Metoda větví a mezí (branch and bound)** byla poprvé představena a popularizována roku 1963 v práci "An algorithm for the traveling salesman problem" Johna D. Littla a jeho dalších kolegů. Algoritmus, který tým popsal ve své práci je tak často nazýván právě po jednom z hlavních řešitelů. Množina řešení je přirovnávána ke stromu, jehož větve představují dílčí části prostoru řešení. Pro větve jsou stanoveny odhady spodní meze účelové funkce a jsou vyřazeny, pokud se v nich na základně odhadů nemůže objevit optimum. Tento způsob může několikanásobně urychlit výpočet, ale může

nastat situace, kdy algoritmus prozkoumá všechny řešení tak jako v případě metody hrubé síly. [28]

4.2 Heuristický přístup

Pomocí heuristiky je možno nalézt lokální optimum, které ovšem nemusí být globální. O kolik se nalezené řešení liší od globálního optima (jak je přesné) ovšem nevíme. Pomocí tohoto přístupu jsme schopni v relativně krátkém čase naleznout použitelné řešení, které má pro nás jistě větší přínos nežli žádné. Pokud pracujeme s množinou vrcholů v řádech stovek nebo tisíců, je takřka nemožné k řešení využít exaktní metody pro výpočet v reálném čase. Zároveň platí, že s přibývajícím časem pro výpočet se u některých heuristických algoritmů nalezené lokální optimum přibližuje optimu globálnímu. [27] [28]

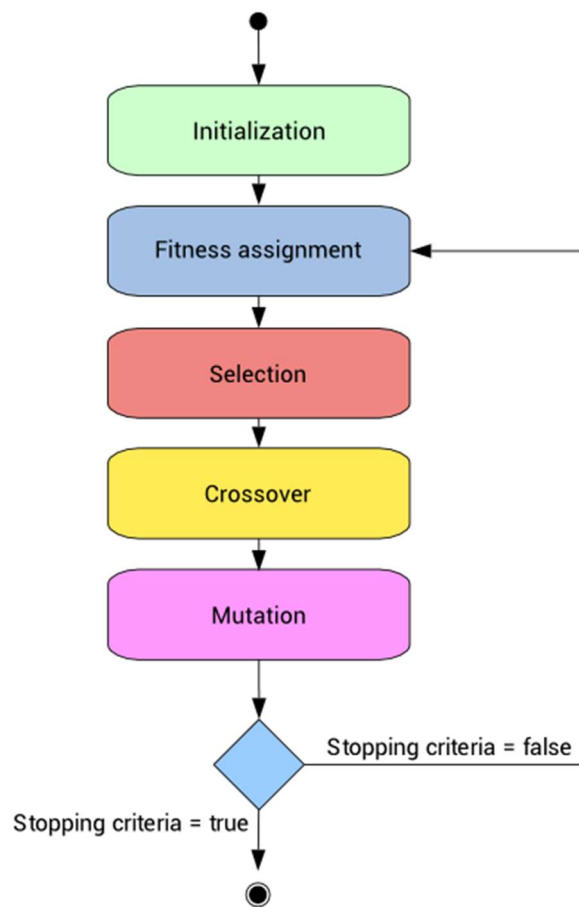
- **Metoda nejbližšího souseda (nearest neighbor)** – Algoritmus z výchozího bodu postupuje do dalšího na základě nejkratší vzdálenosti mezi nimi do doby, kdy „obslouží“ všechny vrcholy a vrátí se zpět do počátku.
- **Hladový algoritmus (greedy search)** nejprve seřadí všechny dvojice měst podle jejich vzdálenosti. Následně se začíná tvořit výsledná trasa od hrany s nejnižším ohodnocením tak, aby nedošlo k předčasnému „zacyklení“.
- **Christofidova metoda** je dalším způsobem, jak zjistit přípustné řešení. Algoritmus obsahuje několik kroků a garantuje, že řešení nebude horší než 1,5násobek globálního optima. Vychází ze zjištění minimální kostry grafu.

4.2.1 Metaheuristický přístup

Jedná se o stochastické heuristické metody, mezi které patří zejména genetické algoritmy, simulované žhání, horolezecký algoritmus nebo zakázané prohledávání. Tyto algoritmy nám nemusí zajistit globální optimum, ale v přijatelném čase vygenerují použitelné přípustné řešení. Často jsou odvozeny z jevů vyskytujících se v přírodě a drtivá většina v sobě obsahuje proces učení. [29] [30]

- **Genetické algoritmy** se snaží napodobit Darwinovu evoluční teorii přirozeného výběru vyskytující se v přírodě, kterou používají při řešení úloh. Na začátku máme populaci jedinců (chromozómů) popsaných bitovým řetězcem obsahujícím geny, ze které se vyberou kvalitní jedinci s vysokou schopností reprodukce (hodnotou fitness). Z vybraných jedinců se začne pomocí mutace a křížení tvořit další generace. Křížením se rozumí výměna části informace mezi dvěma jedinci. Mutace inverzně mění hodnoty genů s předem stanovenou pravděpodobností. Snahou vybírat jedince s požadovanou hodnotou fitness, by mělo postupně dojít k nalezení nejzdatnějšího jedince, který

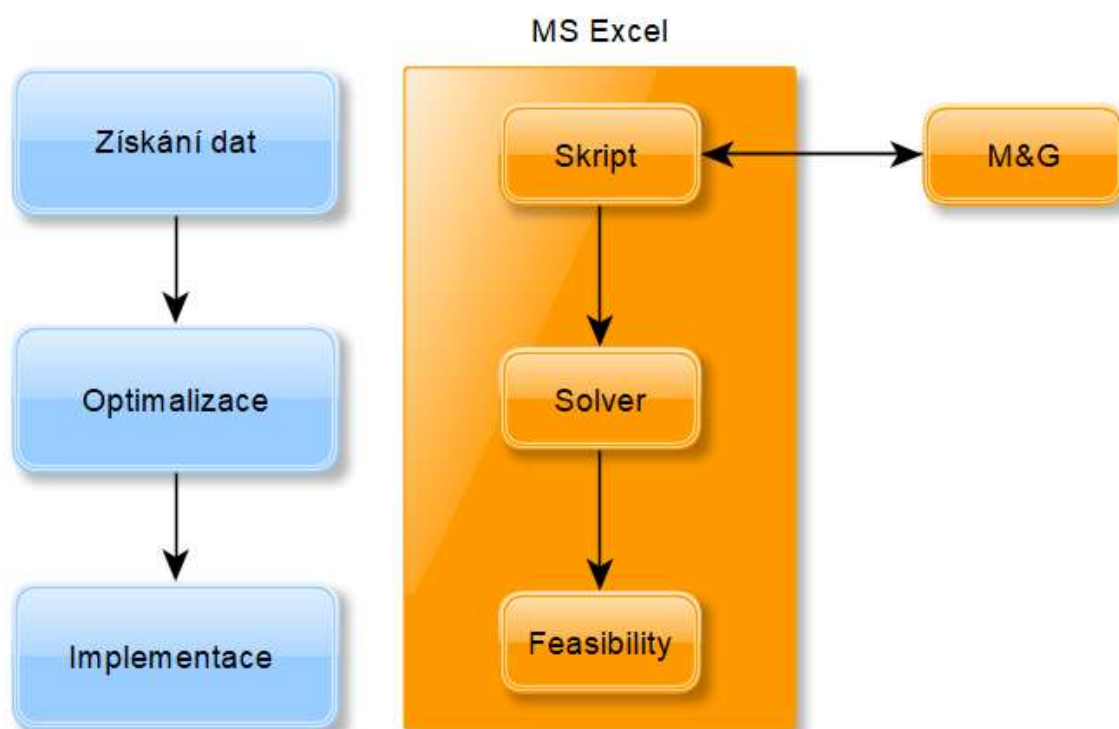
reprezentuje hledané optimum. Před výpočtem je důležité správné nastavení parametrů jako je rozsah počáteční populace, pravděpodobnost křížení a mutace.



Obrázek 11 – Vývojový diagram popisující genetický algoritmus.[31]

5 Způsob řešení

Cílem je zautomatizovat celý proces a vytvořit feasibility, která bude obsahovat optimální trasu s veškerými náležitostmi jako doba jízdy, počet ujetých km s mýtem, seznam dodavatelů a dalších informací dle aktuální potřeby. Tento úkol je možné vyřešit skrze MS Excel a vytvořit jednoduché makro (aplikaci), které by autonomně vykonalo dříve rutinní manuálně prováděné úkony. Požadovanou aplikaci (makro) lze z funkční logiky rozdělit na tři části. Nejdříve je potřeba získat zdrojová data pro opt. algoritmus, provést samotnou operaci optimalizace a na závěr data implementovat do výsledného formuláře (viz. obr. 12).



Obrázek 12 – Postupné kroky makra při tvorbě feasibility v MS Excel a vazby mezi jednotlivými nástroji. Zdroj: vlastní zpracování

5.1 Vývojové prostředí

Celý problém jsem se pokusil vyřešit skrze kancelářský program Microsoft Excel. Tento program je nejrozšířenější tabulkový procesor na světě, je vhodný pro tvorbu maker a aplikace, kterou mohou jednoduše kdekoliv používat všichni odpovědní zaměstnanci. Velkou výhodou Excelu je možnost přidání různých Add-onů (doplňků) a využití editoru Visual Basic for application (VBA) pro programování složitějších aplikací.

Pro tvorbu feasibility bude sloužit excelový sešit, na jehož prvním listě se nachází veškeré nastavení parametrů (adresy dodavatelů, nastavení opt. nástroje atd.). Po spuštění makra se

začnou ostatní listy plnit daty, na jejichž základě dojde k optimalizaci a tvorbě výsledného formuláře.

5.1.1 Visual Basic for Applications (VBA)

Jedná se o vývojové prostředí a programovací jazyk, který je příbuzný známějšímu Visual Basic (VB) od firmy Microsoft. VBA se používá pro programování v kancelářských programech Microsoft Office – zejména v Excelu nebo Wordu. Jeho pomocí se tvoří jednoduchá, ale i komplexnější makra. Nevýhodou je nemožnost spustit aplikaci (makro) jako samostatný (.exe) program. Vždy musí být spuštěn přes některý z nástrojů MS Office. Velkou výhodou je jeho jednoduchost, totožná syntaxe s hojně používaným Visual Basic a fakt, že je součástí kancelářských balíků MS Office. Stále se však nejedná o plnohodnotnou náhradu VB. Vzhledem k tomu, že vývojářem jazyku a prostředí je Microsoft, interoperabilita mezi systémy a programy cizích výrobců je bohužel špatná nebo často nulová. Pro zautomatizování rutinních a časově náročných činností je však i přes některé uvedené nedostatky výborným nástrojem. [32]

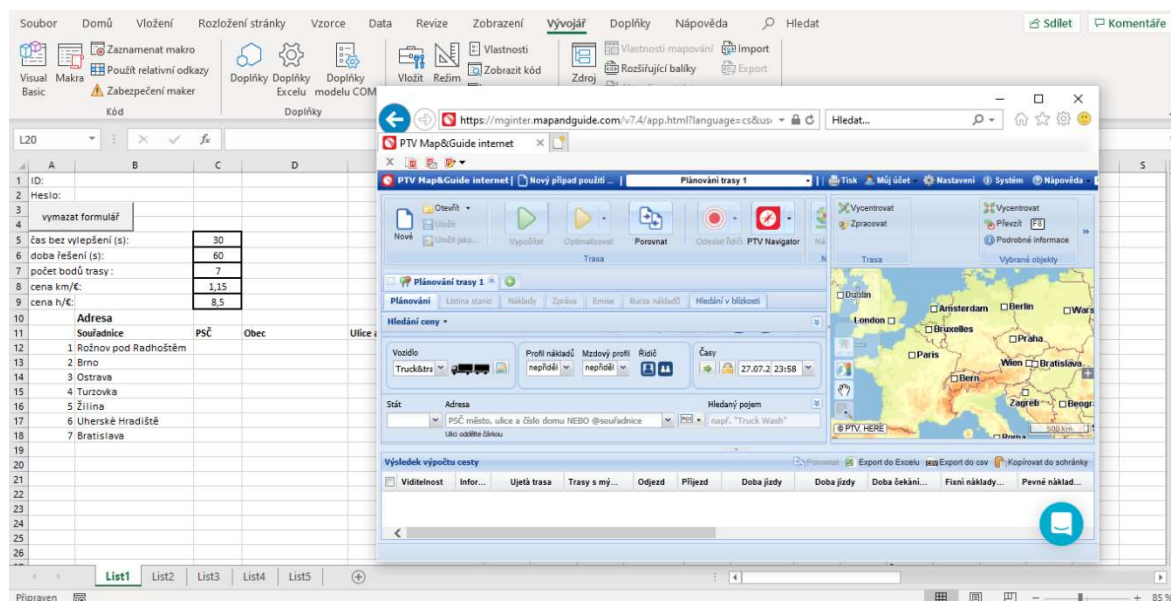
5.2 Získání dat

Jako vstupní data pro opt. algoritmus je potřeba získat nákladovou matici zobrazující náklady způsobené jízdou mezi všemi možnými dvojicemi bodů v matici. To znamená spočítat nejlevnější trasu mezi každými dvěma prvky matice. Tyto hodnoty však nikde nenajdeme, a tak je potřeba je vypočítat na základě „průměrných“ nákladů. Tímto ústupkem se vzdáváme optimálního řešení, nicméně je takřka nereálné s dostupným softwarem se k nejlevnější trase dopracovat. Proto budu vycházet z „nejrychlejší trasy“ kterou automaticky generuje drtivá většina on-line platform pro plánování tras a následně ji vyjádřím přepočtem na peněžní jednotky. Dá se předpokládat, že ve většině případů automaticky vygenerovaná „nejrychlejší trasa“ je ekonomicky výhodná. V praxi to znamená, že je ujetá vzdálenost vynásobena průměrnými náklady na jeden km a doba jízdy hodinovou mzdou řidiče. Odhad slouží pouze ke kvantifikaci veškerých nákladů a možnou optimalizaci. Výsledné vypočtené náklady jsou pouze orientační a liší se od reality.

Nejjednodušší metoda pro určení trasy se zdá být použití online map Google, Mapy.cz nebo jiné platformy pro plánování tras. Tyto zdarma dostupné aplikace ovšem nejsou tvořeny pro nákladní kamionovou dopravu, a tak z nich nezískáme informace o mýtě, sjízdnosti, zákazech, či jiných restrikcích. Z tohoto pohledu se jeví jako nejlepší užití specializovaného programu na nákladní dopravu PTV Map and Guide, který je využíván společností „XY“ a obsahuje plno pokročilých funkcí pro plánování tras. vyhledá nejrychlejší spojení mezi dvěma body s přihlédnutím na sjízdnost či hmotnostní nebo časové restrikce.

- **Map and Guide** je nástroj společnosti PTV Group pro plánování tras v nákladní dopravě. Dokáže spočítat náklady pro konkrétní trasu a typ (profil) vozidla. M&G na základě průjezdných bodů vymodeluje trasu a zohlední restriktce pro daný typ vozidla a vyčíslí dodatečné poplatky jako mýto nebo trajekt. Nevýhodou je fakt, že algoritmus aplikace nepočítá varianty s jiným než výchozím seřazením dodavatelů, a tak je potřeba počáteční pořadí externě optimalizovat. [33]

Než celý automatizmus započne je třeba v excelovém sešitě vyplnit jednotlivé adresy (souřadnice) dodavatelů, které bude trasa obsahovat a s tím spojené náležitosti jako počet zastávek nebo doba výpočtu algoritmu. Po zadání dodavatelů jsme schopni postupně vyplnit matici hodnotami z PTV Map and Guide. Tento úkol je však manuálně vysoce časově náročný. Při výpočtu s n městy je potřeba úkon, za předpokladu, že zpáteční trasa je totožná vykonat x krát, jestliže $x = \frac{n \cdot (n-1)}{2}$. To nám při 15 městech dává 105 manuálních výpočtů (kopírování měst, výsledků atd.). Není divu, že tento proces je potřeba zautomatizovat, a to pomocí skriptu, který lze naprogramovat v rámci excelového vývojářského prostředí Visual Basic.



Obrázek 13 – Prostředí MS Excel s automaticky spuštěným webovým prohlížečem pro interakci se skriptem po spuštění makra. Zdroj: vlastní zpracování

Cílem je exportovat každé dva názvy (adresy) bodů (nebo jejich souřadnice) z Excelu do internetové aplikace M&G, která mezi nimi vypočítá “nejrychlejší trasu” a následné informace o vzdálenosti, mýtě a času zpět importuje do Excelu. Pro tuto operaci jsem vytvořil makro, které používá ovládací prvek Microsoft Web Browser. Ten vytvoří skrz tabulkový editor prostředí internetového prohlížeče, přes které lze pomocí naprogramovaného kódu ovládat

jednotlivé prvky webu (obr. 13). Pro autonomní ovládání prohlížeče jsem využil zejména VBA objekt *GetElement*, díky kterému je možno odkazovat na formuláře (importovat/exportovat data), tlačítka nebo další prvky internetových stránek. Tímto způsobem se lze automaticky přihlásit do systému M&G, nastavit počáteční podmínky (např. druh vozidla) a následně importovat a exportovat data mezi excelelem a aplikací. Výsledným produktem skriptu jsou čtyři matice, které obsahují vzdálenost, čas, výši mýta a počet kilometrů ujetých s mýtem mezi všemi body trasy. [34]

Protože počet obsluhovaných měst se bude pokaždé lišit, import a export dat v programovacím prostředí funguje díky cyklům For, které provádí operaci dle potřeby. Před samotnou optimalizací musí být vedle adres (souřadnic) také vyplněn počet bodů, aby cyklus věděl, kolikrát bude jednotlivé kroky skriptu opakovat. Vzhledem k tomu, že počítáme s faktem, že zpáteční trasa (mezi jakýmkoliv body) je vždy totožná, jsou tyto hodnoty (pro opačný směr) pouze zkopírovány, nikoliv znova importovány pomocí skriptu, což ušetří polovinu času.

V dalším kroku je potřeba hodnoty upravit na společný základ – náklady vyjádřené v peněžních jednotkách. To se děje plně automaticky vynásobením průměrnými náklady na jeden km a hodinovou mzdou řidičů. Nezbyvá než všechny hodnoty matic sečíst a dostaneme se tak ke konečné tabulce obsahující náklady mezi jednotlivými body odpovídající nejrychlejší trase. Takto je matice připravena k optimalizaci. Výsledná tabulka pro opt. je zobrazena v tab. č. 2.

Tabulka 2 – Tabulka obsahující náklady spojené s jízdou mezi jednotlivými body trasy připravená k optimalizaci. Zdroj: vlastní zpracování

	Rožnov p. R.	Brno	Ostrava	Turzovka	Žilina	Uherské Hradiště	Bratislava
Rožnov pod Radhoštěm	0	219,72	115,65	64,38	107,94	159,86	342,49
Brno	219,72	0	252,74	266,70	306,27	108,32	191,80
Ostrava	115,65	252,74	0	172,65	216,21	190,70	423,73
Turzovka	64,38	266,70	172,65	0	79,56	221,27	318,04
Žilina	107,94	306,27	216,21	79,56	0	195,63	292,26
Uherské Hradiště	159,86	108,32	190,70	221,27	195,63	0	193,70
Bratislava	342,49	191,80	423,73	318,04	292,26	193,70	0

5.2.1 Skript

Stěžení část automatického importu a exportu dat z excelu do webové aplikace M&G tvoří dva vnořené cykly For. Nejprve jsou vypočítány všechny možné trasy vedoucí z prvního bodu v seznamu, poté z druhého a tak dále. Proces je v podstatě analogický pro všechny body až na fakt, že při změně výchozího bodu se už nepočítají trasy vedoucí do bodů, které byly už dříve vloženy jako výchozí. Vycházíme z předpokladu, že trasa má v obou směrech stejné parametry, a tak tyto úseky není nutné znova počítat.

V následujících řádcích se pokusím popsat klíčové části zdrojového kódu. Kompletní zdrojový kód, který zaručuje automatickou tvorbu formuláře se nachází v příloze 1.

```
.  
.br/>.
```

```
For i = 12 To (10 + n) `První cyklus pro vložení výchozího bodu, n = počet  
bodů trasy
```

```
_____Zadání výchozího bodu
```

```
doc.getElementById("mgi-addresssearchform-search2").Value =  
ThisWorkbook.Sheets("List1").Cells(i, 2).Value `Do adresového pole M&G se  
vloží adresa příslušného bodu  
doc.getElementById("ext-gen517").Click  
Application.SendKeys "~" `Simulace stisknutí klávesy enter pro vyhledání  
zadané adresy  
Application.Wait DateAdd("s", 3, Now) ` "Uspání" aplikace po dobu 3 s. pro  
vyhledání adresy  
Application.SendKeys "~" `Simulace stisknutí klávesy enter pro potvrzení  
adresy  
Application.Wait DateAdd("s", 1, Now)
```

```
_____Kopie informací o prvním bodě
```

```
Cells(i, 3).Value = doc.getElementsByClassName("x-grid3-cell-inner x-grid3-  
col-postCode x-unselectable")(0).innerText `Kopie PSČ z M&G do Excelu  
Cells(i, 4).Value = doc.getElementsByClassName("x-grid3-cell-inner x-grid3-  
col-city x-unselectable")(0).innerText `Kopie Města z M&G do Excelu  
Cells(i, 5).Value = doc.getElementsByClassName("x-grid3-cell-inner x-grid3-  
col-street x-unselectable")(0).innerText `Kopie Ulice a č.p. z M&G do  
Excelu
```

```
For j = 13 + y To 11 + n `Cyklus č.2 pro vložení cílového bodu
```

```
_____Zadání koncového bodu
```

```
doc.getElementById("mgi-addresssearchform-search2").Value=  
ThisWorkbook.Sheets("List1").Cells(j, 2).Value  
doc.getElementById("ext-gen517").Click  
Application.SendKeys "~"  
Application.Wait DateAdd("s", 3, Now)  
Application.SendKeys "~"
```



```
_____ Výpočet trasy
Application.Wait DateAdd("s", 1, Now)
Application.SendKeys "{F9}" `Simulace stisknutí tlačítka F9, které dá povel
výpočtu trasy mezi zvolenými body
```

```
_____ Uložení délky trasy z M&G do proměnné delkaTrasy
Application.Wait DateAdd("s", 4, Now)
delkaTrasy1 = doc.getElementsByClassName("x-grid3-cell-inner x-grid3-col-
totalDistance x-unselectable")(0).innerText `Kopíruje vnitřní text
formuláře obsahujícího počet km
u = Split(delkaTrasy1, " ") Rozdělí text na místě mezery a vybere z něj
pouze číslo počtu km
delkaTrasy = u(0)
```

```
If delkaTrasy = "0,00" Then `Podmínka, která zaokrouhlí délku tras kratších
než 10 metrů na 10 m, aby se ve vstupních maticích optimalizace neobjevila
nula
delkaTrasy = "0,01"
Else: delkaTrasy = u(0)
End If
```

```
_____ Export cestovní doby z M&G do proměnné CestovniCas
CestovniCas = doc.getElementsByClassName("x-grid3-cell-inner x-grid3-col-
totalTime x-unselectable")(0).innerText
u = Split(CestovniCas, " ")
CestovniCas = u(0)
```

```
If CestovniCas = "0:00" Then
CestovniCas = "0:01"
Else: CestovniCas = u(0)
End If
```

```
_____ Export nákladů (mýto) z M&G do proměnné Naklady
naklady1 = doc.getElementsByClassName("x-grid3-cell-inner x-grid3-col-
tollCosts x-unselectable")(0).innerText
u = Split(naklady1, " ")
Naklady = u(0)
```

```
If Naklady = "-" Then
Naklady = "0"
Else: Naklady = u(0)
End If
```

```
_____ Export počtu km pod mýtem z M&G do proměnné
Cesta_s_mytem
Cesta_s_mytem = doc.getElementsByClassName("x-grid3-cell-inner x-grid3-col-
tollDistance x-unselectable")(0).innerText
u = Split(Cesta_s_mytem, " ")
Cesta_s_mytem = u(0)
```

```
If Cesta_s_mytem = "-" Then
Cesta_s_mytem = "0"
```

```
Else: Cesta_s_mytem = u(0)
End If
```

```
_____Vytvoření excelových matic z výše získaných hodnot
Cells(Z, x).Value = delkaTrasy * 1 'Vytvoření "horní" poloviny matice
Cells(Z - y + 1 + (x - c), x - (x - c + 1 - y)).Value = delkaTrasy * 1
'Vytvoření "dolní" poloviny matice, násobení jednou z důvodu převodu
hodnoty na číslo
```

```
Cells(r, x).Value = CestovniCas
Cells(r - y + 1 + (x - c), x - (x - c + 1 - y)).Value = CestovniCas
```

```
Cells(S, x).Value = Naklady * 1
Cells(S - y + 1 + (x - c), x - (x - c + 1 - y)).Value = Naklady * 1
```

```
Cells(t, x).Value = Cesta_s_mytem * 1
Cells(t - y + 1 + (x - c), x - (x - c + 1 - y)).Value = Cesta_s_mytem * 1
```

```
_____Vymazání koncového bodu v M&G
doc.getElementsByClassName("ux-row-action-item fam-delete ")(1).Click'
Kliknutí na tlačítko, které smaže daný bod trasy
x = x + 1
```

```
Next j
```

```
_____Vymazání výchozího bodu v M&G
doc.getElementsByClassName("ux-row-action-item fam-delete ")(0).Click
```

```
x = c + M
y = y + 1
Z = Z + 1
S = S + 1
M = M + 1
r = r + 1
t = t + 1 'Proměnné x,y,Z,S,M,r nebo t slouží k práci se souřadnicemi
jednotlivých buněk v excelu, do (nebo z) kterých jsou postupně kopírovány
hodnoty
```

```
Next i
```

```
.
.
.
```

5.3 Postup optimalizace

Vzhledem k tomu, že jsem problém řešil v programu MS Excel tak, aby mohl být celý „program“ co nejjednodušeji používán, využil jsem nástroje Solver (Řešitel), který při vhodném uspořádání dat umožňuje optimalizovat i sofistikované úlohy.

5.3.1 Řešitel (Solver)

Jedná se o nástroj společnosti FrontlineSolvers, jehož základní verze je dostupná zdarma v MS Excelu. Placená verze nabízí celou škálu analytických, simulačních nebo optimalizačních nástrojů. Standartní varianta ovšem k řešení postačuje. Na výběr máme tři metody řešení – Evoluční algoritmus, Simplexovou nebo Gradientní metodu. Simplexová metoda sice nalezne optimální řešení, může být však aplikována pouze na lineární funkce a je omezena počtem proměnných. To můj model nespĺňuje, protože se v něm objevují excelové funkce typu INDEX. Proto jsem zvolil optimalizaci pomocí evolučního algoritmu. [35]

5.3.2 Model

Nyní je potřeba správně připravit data a nastavit parametry řešitele – určit účelovou funkci, proměnné a omezující podmínky. Postupoval jsem podle oficiální příručky „Microsoft Excel 2019 Data Analysis and Business Modeling“. Níže jsou popsány jednotlivé buňky. [35]


	A	B	C	D	E	F	G	H	I	J	K
2											
3	1	1									
4											
5											
6											
7		219,72	1	1457,49	1	2	3	4	5	6	7
8		252,74	2	1	0	219,72	115,65	64,38	107,94	159,86	342,49
9		172,65	3	2	219,72	0	252,74	266,70	306,27	108,32	191,80
10		79,56	4	3	115,65	252,74	0	172,65	216,21	190,70	423,73
11		195,63	5	4	64,38	266,70	172,65	0	79,56	221,27	318,04
12		193,70	6	5	107,94	306,27	216,21	79,56	0	195,63	292,26
13		342,49	7	6	159,86	108,32	190,70	221,27	195,63	0	193,70
14			1	7	342,49	191,80	423,73	318,04	292,26	193,70	0


Obrázek 14 – Tabulka v kompatibilní podobě určená k optimalizaci nástrojem Excel Solver.


Zdroj: vlastní zpracování


- (1) Buňky obsahují hodnotu, která je dána souřadnicemi ve vedlejším sloupci podle vzorce INDEX(Matice;Řádek;Sloupec). „Matice“ odpovídá matici „(6)“ obsahující jednotlivé vzdálenosti mezi vrcholy. „Řádek“ odkazuje na vedlejší buňku, ve které se nachází proměnná. „Sloupec“ odkazuje na buňku, která se nachází o jeden řádek níže než buňka předešlá („řádková“). Pro hodnotu 219,72 je vzorec roven

INDEX(matice;C7;C8), což po dosazení odpovídá INDEX(matice;1;2). Následující buňka 252,74 pak INDEX(matice;2;3). Při minimalizaci bude snaha nalézt buňky s co nejnižší možnou hodnotou, na základě měnících se proměnných.


 • (2) Proměnné, reprezentující jednotlivé vrcholy, zároveň tvoří souřadnice pro buňky „(1)“, které se mění na základě změny pořadí proměnných.

 • (3) Účelová funkce – funkce, která reprezentuje optimalizovanou veličinu, jedná se o sumu hodnot buněk „(1)“.

 1 • (4) Hodnota buňky zajišťuje návrat do počátečního bodu. Díky ní budou souřadnice posledního řádkového kritéria [x;1], což odpovídá návratové hodnotě z x (posledního bodu) do 1 (prvního).

 • (5) Obsluhované vrcholy.

 • (6) Matice obsahující náklady vytvořené jízdou mezi dílčími vrcholy.

 • (7) Bivalentní proměnná (BP) – nabývá dvou hodnot (0 nebo 1). Zajistí, že počátek HK bude v prvním vrcholu. Pokud bude proměnná odpovídající bodu na první pozici rovna 1, hodnota BP bude 1, pokud ne, hodnota bude 0. Tento mechanismus funguje na základě vzorce $KDYŽ(C7=A3;1;0)$. Podmínka bude aktivní (BP=1) za předpokladu, že C7 (první obsluhovaný vrchol) bude roven hodnotě buňky A3 (obsahuje hodnotu 1). V opačném případě bude BP=0.

5.3.3 Omezující podmínky

Dále už stačí v parametrech určit omezující podmínky optimalizace (obr. 15). V modelu použijí tři omezení:

- Protože má být každý vrchol obslužen právě jednou, nastavím, aby hodnota proměnných byla různá – každé proměnné bude náležet jiná hodnota.

$SCS7:SCS13 = \text{Vše_různé}$

- Aby HK začínala v prvním vrcholu, musí zmíněná bivalentní proměnná nabývat hodnoty 1, což zajistí druhá podmínka.

$SB\$3 = 1$

- Nakonec nastavíme podmínky nezápornosti „zakliknutím“ v parametrech řešitele.

Parametry Řešitele

Účejová funkce:

Hledat: Max Min Hodnota:

Proměnné modelu:

Omezující podmínky:

\$C\$7:\$C\$13 = Vše_různé
\$B\$3 = 1

Nastavit podmínky nezápornosti

Vyberte metodu řešení:

Metoda řešení
Simplexovou metodu zvolte pro lineární optimalizační problémy, Gradientní metodu pro hladké nelineární problémy a Evoluční algoritmus pro nehladké nelineární problémy.

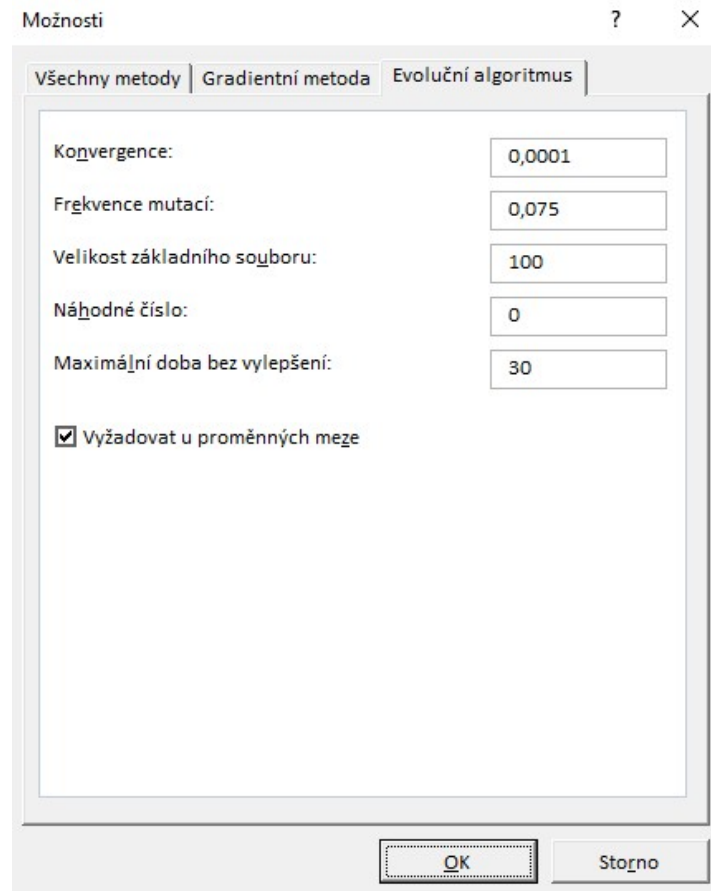
Obrázek 15 – Nastavení nástroje Excel Solver. Zdroj: vlastní zpracování

5.3.4 Nastavení evolučního algoritmu

V pokročilém nastavení lze dále upravovat několik parametrů před startem řešitele (obr. 16) [36]:

- **Konvergence** – evoluční algoritmus se zastaví, pokud relativní rozdíl hodnot fitness více než 99 % jedinců populace bude menší než daná tolerance.
- **Frekvence mutací** – hodnota mezi 0 a 1, čím větší bude, tím více bude následující populace různorodá a vyšší šance, že bude nalezeno lepší přípustné řešení. S tím ovšem roste doba výpočtu.
- **Velikost základního sboru** – počet jedinců v počáteční populaci. S rostoucím počtem populace se zvyšuje pravděpodobnost, že řešitel nalezne nebo se výrazně přiblíží optimu, to ovšem opět znamená nárůst času.
- **Náhodné číslo** – kladná celočíselná hodnota, na jejíž základě se budou generovat náhodná rozhodnutí evolučního algoritmu.

- **Max. doba bez vylepšení** – čas po jehož uplynutí se řešitel zastaví, nedejde-li k vylepšení hodnoty účelové funkce.



Obrázek 16 – Nastavení metody evoluční algoritmus v nástroji Excel Solver. Zdroj: vlastní zpracování

Jelikož je řešitel spuštěn v rámci makra automaticky, je potřeba všechny změny nastavení provádět v kódu ve Visual Basic. Lze však kterémukoliv jednotlivému nastavení naprogramovat buňku, kterou uživatel vyplní před startem kompletního makra. Z tohoto důvodu jsem na 1. listu umožnil nastavit čas řešení a maximální dobu bez vylepšení přípustného řešení, po které se algoritmus zastaví. Samotná syntaxe nastavení vypadá následovně:

```
n = ThisWorkbook.Sheets("List1").Cells(6, 3).Value
```

```
SolverReset
```

```
SolverOk SetCell:="$D$7", MaxMinVal:=2, ValueOf:=0, ByChange:="Proměnné", _  
Engine:=3, EngineDesc:="Evolutionary"
```

```
SolverAdd CellRef:="Proměnné", Relation:=6, FormulaText:="Vše_různé"  
SolverAdd CellRef:="$B$3", Relation:=2, FormulaText:="1"
```

```
SolverOptions MaxTime:=n, Iterations:=0, Precision:=0.000001, Convergence:=
0.0001, StepThru:=False, Scaling:=True, AssumeNonNeg:=True, Derivatives:=1
SolverOptions PopulationSize:=100, RandomSeed:=0, MutationRate:=0.075,
Multistart _:=False, RequireBounds:=True, MaxSubproblems:=0,
MaxIntegerSols:=0, _
IntTolerance:=0.1, SolveWithout:=False, MaxTimeNoImp:=30

SolverSolve
```

5.4 Implementace

Poslední část makra slouží k prezentaci výsledku předchozí optimalizace. Výstupem celého procesu by měl být částečně makrem vyplněný formulář (obr. 18), který se ve společnosti standartně používá a dále zpracovává.

Makro do standardizovaného formuláře nakopíruje vypočítanou optimální trasu – chronologicky seřazené body, adresy, počet kilometrů jízdy pod mýtem, dobu jízdy od zvoleného okamžiku nebo průměrné rychlosti a vzdálenosti mezi jednotlivými úseky.

Vygenerovaná trasa je současně vložena do programu M&G (obr. 17), kde se zobrazí přehledně na mapě. Protože při optimalizaci dochází ke „skládání“ jednotlivých úseků v kompaktní trasu, trasa v M&G (zadána se všemi body najednou) se často liší v řádu jednotek kilometrů. Dále program zobrazuje dodatečné informace, které při vkládání pouze dílčích tras (mezi dvěma body) logicky nemohl nabídnout. Jedná se zejména o informace plynoucí z dohody AETR, která definuje pravidla osádek vozidel mezinárodní silniční dopravy (povinné přestávky řidičů).

Uživateli nakonec zbývá doplnit několik informací, jako doba nakládky/vykládky a další věci, které vyplývají z letité zkušenosti s provozem společnosti a jsou tak do formulářů zadávány ručně.

Tabulka 3 – Vygenerovaná tabulka na prvním listu v Excelu s detailními informacemi pro jednotlivé úseky. Zdroj: vlastní zpracování

	Celkem	Rožnov	Turzovka	Žilina	Bratislava	Brno	Uh.Hradiště	Ostrava	Rožnov
Trasa		1	4	5	7	2	6	3	1
Vzdálenost (km)	733,53	-	45,07	53,98	203,66	134,2	78,46	135,69	82,47
Čas (h)	13:35	-	1:24	1:31	2:54	2:06	1:29	2:22	1:49
Mýto (€)	83,65	-	0,65	4,59	33,40	19,62	5,48	14,54	5,37
Mýto (km)	518,6	-	4,39	27,57	175,82	116,42	59,2	99,22	35,98
Náklady (€)	1042,67	-	64,38	79,56	292,26	191,80	108,32	190,70	115,65

V poslední řadě nabízí excel na 1. listu detailní informace jako rychlost, celkové náklady, vzdálenost, počet kilometrů pod mýtem, samotnou výši mýta a čas pro jednotlivé úseky (tab. 3).

The screenshot displays the PTV MapGuide software interface. The main window shows a map of Central Europe with a highlighted route. The route is divided into segments, each with a number and a name. The 'Seznam tras' (Route List) panel on the left provides detailed information for each segment, including the start and end points, distance, and toll amount.

Identifikační záložky	PSČ	Obec
1	726 01	Roznov pod Radhoštěm
2	023 54	Turovka
3	010 **	Zlín
4	6*****	Bratislava
5	8*****	Bрно
6	688 0*	Uherské Hradiště
7	7*****	Ostava
8	726 01	Roznov pod Radhoštěm

The interface also shows a summary of the route: 'Výsledek výpočtu ceny' (Calculation result) with a total distance of 730.47 km and a total toll of 616.29 km. The 'Hledání ceny' (Find price) section shows a total cost of 730.77 kg. The map shows the route starting in Roznov pod Radhoštěm, passing through Turovka, Zlín, Bratislava, Brno, Uherské Hradiště, Ostava, and returning to Roznov pod Radhoštěm.

Obrázek 17 – Vymodelovaná zpětně vložená trasa do M&G. Zdroj: vlastní zpracování

P02 2020 - LP proposal													
ROUTE	Supplier Code	Supplier Name	City	Address	Arrival Time	Departure Time	Transit time calculation	Transit Time Basic Diagram	TLMS input minutes	Driver Break	kms map point	avg. Speed	Collection time
K002		Rožnov pod Radhoštěm				10:00	1:40	1:24			45,07	32	
			Turzovka		11:24	11:24	1:52	1:31			53,98	36	
			Zlín		12:55	12:55	2:90	2:54			203,66	70	
			Bratislava		15:49	15:49	2:10	2:06			134,2	64	
			Brno		17:55	17:55	1:48	1:29			78,46	53	
			Uherské Hradiště		19:24	19:24	2:37	2:22			135,69	57	
		Ostrava			21:46	21:46	1:82	1:49			82,47	45	
		Rožnov pod Radhoštěm			23:35								
							13,58	13:35	0	0:00	733,53	51	0:00
unload/load													

1 loop feasibility	
Total Km	733,53
Total trans	13:35
Collection t	0:00

Resources for one loop	
Driver	
Tractor	
Km	
Tolled Km	518,6

break	0:00
total work	13:35

Obrázek 18 – Výsledná podoba vygenerovaného formuláře feasibility před ručním vyplněním některých náležitostí. Zdroj: vlastní zpracování

5.5 Struktura makra

Celá aplikace se skládá z několika menších maker. Ta slouží pro vygenerování matic potřebných k výpočtu, přepočítání nákladů nebo smazání starých dat. Aplikace *program* postupně automaticky spouští jednotlivá makra a její zdrojový kód je zobrazen níže.

```
Sub program()  
  
Call Skript  
Call seznam  
Call Kopie_vstupu  
Call SoucetNakladu  
Call Kopie_nakladu_do_Resitele  
Call PosunMinim  
Application.Wait DateAdd("s", 1, Now)  
Call resitel  
Call Výsledek1  
Call mesta_vysledky  
Call autofill_feasibility  
Call Optimalni_trasa_PTV  
Call Delete  
  
End Sub
```

- **Skript** – již zmiňovaná část, která generuje zdrojová data z M&G pro optimalizaci.
- **seznam** – vytvoří vzor čtyř matic (o správném počtu řádků/sloupců, s nadepsanými městy), které jsou následně vyplněny skriptem.
- **Kopie_vstupu** – dojde ke zkopírování dat ze čtyř matic do listu č.2, kde jsou hodnoty přepočítány na společný jmenovatel (náklady).
- **SoucetNakladu** – náklady ze čtyř matic jsou sečteny do jediné.
- **Kopie_nakladu_do_Resitele** – matice s náklady je zkopírována do oblasti, která je vstupem pro optimalizační software Řešitel.
- **PosunMinim** – upraví vzorce buněk v konkrétní oblasti tak, aby mohly být načteny Řešitelem.
- **resitel** – spustí se řešitel s počátečními přednastavenými hodnotami.
- **Výsledek1** – nakopíruje výsledné hodnoty do 1. listu.
- **mesta_vysledky** – další z maker sloužící k prezentaci výsledku, vloží optimálně seřazená města do 1.listu.
- **autofill_feasibility** – vyplnění formuláře (feasibility) daty.
- **Optimalni_trasa_PTV** – spuštění M&G a následné vložení optimální posloupnosti dodavatelů, které slouží k vizuální kontrole.
- **Delete** – vymaže starou feasibility s použitými maticemi, makro je připraveno pro optimalizaci další linky.

5.6 Analýza zvoleného řešení

V následující části se pokusím popsat jednotlivé silné a slabé stránky zvoleného řešení. Pro tento krok je ideální použít SWOT analýzu. Existuje přehršel způsobů, jakým lze problém vyřešit. Jak už bylo zmíněno, cílem je zautomatizovat proces tvorby formuláře (feasibility), který obsahuje optimální trasu se všemi náležitostmi.

Silné stránky

- Kompletní makro funguje autonomně a zvyšuje produktivitu práce.
- Při změně či implementaci nové linky lze nástroj použít pro prvotní vygenerování trasy a dále s ní pracovat.
- Aplikace je poměrně jednoduchá na obsluhu a s validními vstupními daty poskytuje relativně přesný výstup.
- Jednotlivé části lze použít odděleně – při předem známém optimálním pořadí může makro pouze vygenerovat formulář feasibility.

Slabé stránky

- Optimalizační algoritmus je vysoce závislý na kvalitně poskytnutých vstupních dat.
- M&G neposkytuje vždy validní data v daný okamžik – nepřesné informace o zákazech jízdy pro nákladní dopravu, teoretické průměrné rychlosti kamionů se od praxe často liší.
- Makro ve VBA nelze v průběhu zastavit, při ojedinělé chybě musí program začít znovu. Neexistuje tak možnost pozastavit makro při inicializaci aplikace M&G, nastavit si všechny požadované parametry (profil nákladního automobilu, čas) a následně makro opět spustit. Pro nastavení je tak vymezen fixní čas, po jehož překročení makro dále pokračuje.
- Výpočet pomocí metaheuristického evolučního algoritmu – nezaručuje nalezení optimálního řešení.
- Vkládání uzlů z Excelu do internetové aplikace M&G i přes automatizaci trvá delší dobu v závislosti na počtu uzlů (doba pro výpočet všech hodnot mezi dvěma body a kopírování dat zpět do Excelu trvá přibližně 10 s, při trase mezi 15 městy 17,5 minut).
- Vstupní data pro algoritmus vychází z nejrychlejší trasy mezi úseky vytvořené v M&G (náklady jsou vypočítané pro nejrychlejší trasu v rámci průměrných nákladů na kilometr/hodinu jízdy).

Příležitosti

- Vývojářské prostředí VBA umožňuje mnoho modifikací aktuální verze (změna podoby výsledného formuláře (feasibility) nebo kopírování dalších údajů z M&G jako průměrné náklady na km, odjezd/příjezd, spotřeba atd.)
- Zdrojový kód makra může být dále zdokonalován i mimo Excel a při spolupráci s programátory přetvořen v plnohodnotnou aplikaci.

Hrozby

- Makro je silně vázáno na data z M&G a můžou tak nastat problémy po aktualizacích (M&G).
- Při změně identifikačního názvu (např. ID) objektů na web. stránce (v M&G při rozsáhlé aktualizaci), na které odkazují položky z excelu (formulář pro vložení bodu „A“ a bodu „B“, tlačítko pro kalkulaci trasy atd.) je potřeba změnu manuálně opravit ve zdrojovém kódu.

6 Praktické využití makra

Následující příklady demonstrují využití celého makra ať už k účelům kontroly současných nebo generaci nových linek. Došlo k porovnání jednotlivých způsobů řešení TSP s aplikovaným genetickým algoritmem. Nejprve však detailně popíšu postup práce s makrem vytvářejícím feasibility.

1. Vložení přihlašovacích údajů

- Nejdříve je potřeba vyplnit přihlašovací údaje do internetové aplikace M&G. V excelu se jedná o buňky v prvním listu vedle položek ID a Heslo.

	A	B	C	D	E
1	ID:				
2	Heslo:				
3	vymazat formulář				
4					
5	čas bez vylepšení (s):		30		
6	doba řešení (s):		60		
7	počet bodů trasy :		7		
8	cena km/€:		1,15		
9	cena h/€:		8,5		
10	Adresa				
11	Souřadnice		PSČ	Obec	Ulice a číslo domu
12	1 Rožnov pod Radhoštěm				
13	2 Brno				
14	3 Ostrava				
15	4 Turzovka				
16	5 Žilina				
17	6 Uherské Hradiště				
18	7 Bratislava				

Obrázek 19 – Výchozí stav formuláře před individuálním nastavením, vyplněním dodavatelů na požadované trase a spuštěním makra. Zdroj: vlastní zpracování

2. Nastavení makra

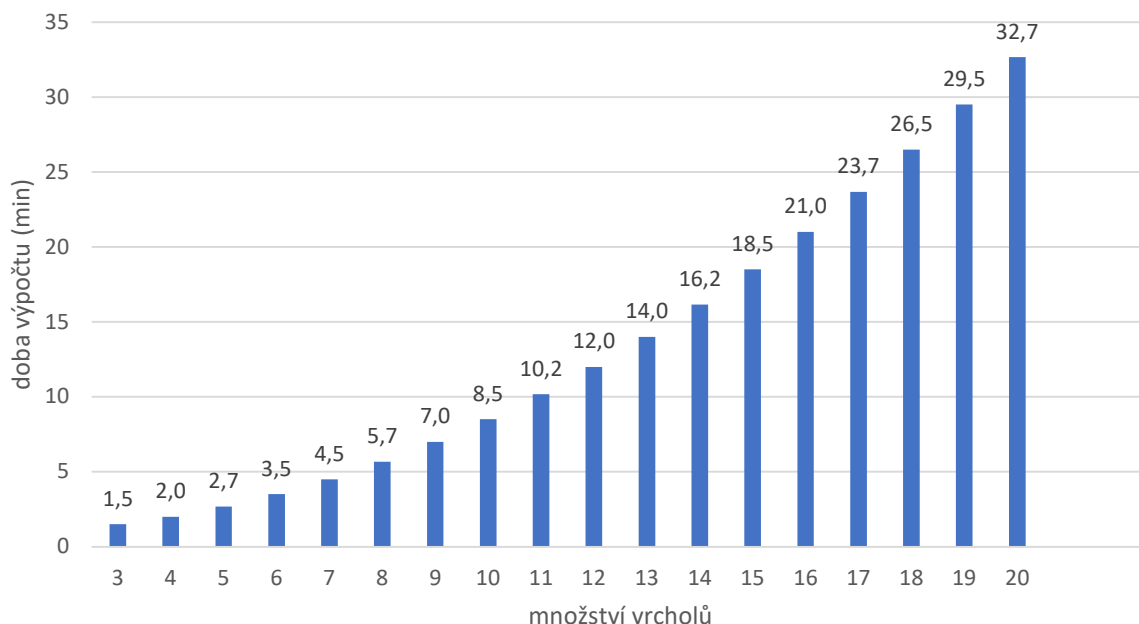
- V tomto kroku nakonfigurujeme všechna potřebná nastavení potřebná pro výpočet. Vyplníme parametry řešitele – čas bez vylepšení, požadovanou dobu řešení. Dále určíme počet bodů trasy včetně výchozího. Tímto krokem se automaticky vygeneruje číslovaný seznam pro vepsání adres nebo souřadnic jednotlivých měst v následujícím kroku. V neposlední řadě zadáme předem odhadnuté náklady na jednu hodinu a kilometr v měně, které budeme chtít vyčíslit výsledné náklady.

3. Vložení bodů trasy

- Po automatickém vygenerování číslovaného seznamu zadáme souřadnice nebo adresu jednotlivých míst trasy do polí sloupce „Souřadnice/Adresa“. Ostatní sloupce zůstávají volné. Vyplní se automaticky pokud jsou zadány pouze souřadnice. Body musí být ve formátu, který je podporován M&G. Nejprve se vepíše PSČ nebo název města, následně ulice a číslo domu oddělené čárkou. Skript obsah buňky zkopíruje a vyplní požadovaný formulář „Adresa“ v internetové aplikaci. Pokud místa zadáváme pomocí souřadnic, M&G před nimi vyžaduje vložit znak „@“.

4. Spuštění makra

- Jakmile je skript nastaven, adresy dodavatelů vloženy, nezbyvá než spustit kompletní makro. Na kartě „Vývojář“ otevřeme nabídku „Makra“ a vybereme makro s názvem „program“. Po kliknutí dojde ke spuštění. Makro automaticky otevře webový prohlížeč s adresou M&G a přihlásí se do systému. Protože makro nelze zastavit, je mu po přihlášení dána pauza cca 1 minutu, po kterou je možno nastavit profil vozidla a další nastavení dle potřeby. Po uplynutí stanoveného času se dává makro opět do pohybu. Doba výpočtu je závislá na počtu vrcholů viz. graf č. 2.



Graf 2 – Závislost doby výpočtu na množství vrcholů na trase. Zdroj: vlastní zpracování

5. Výsledek

- V listu č.4 je vyhotovena feasibility s optimálním seřazením vrcholů a částečně vyplněnými údaji. Souběžně se automaticky spustí M&G s kompletní vymodelovanou trasou zobrazenou v mapě. Zbývá dopsat jen několik informací jako přestávky řidičů, čas vykládky/nakládky atd. Detailnější údaje obsahuje první list excelového sešitu, kde je možno vyčíst data i pro jednotlivé úseky.

6.1 Kontrola optimality současných linek

V tomto kroku jsem se rozhodl prověřit makro několika reálnými (dříve používanými) trasami tak, aby bylo možné zjistit, zdali jsou optimální, případně upozornit na chyby, které při výpočtech mohou nastat. Jedná se zejména o trasy s menším počtem dodavatelů, a tak je pravděpodobné, že byly namodelovány správně. Pro přepočítání nákladů jsem u všech příkladů použil hodnoty 8,5 €/h a 1,15 €/km.

6.1.1 Příklad č.1

První příklad je okružní linka z výchozího bodu 1, na které se vyskytuje 6 dodavatelů (body 2-7). V M&G byl vybrán profil tahač s návěsem, pro který se při výpočtu zohledňují restrikce pro nákladní vozidlo. Spuštěním makra jsou vygenerovány následující tabulky.

Tabulka 4 – Vzdálenost (km). Zdroj: vlastní zpracování

	1	2	3	4	5	6	7
1	0	380,55	351,57	412,36	412,25	405,27	345,89
2	380,55	0	41,03	43,89	43,79	49,39	46,75
3	351,57	41,03	0	94,27	94,17	87,18	25,51
4	412,36	43,89	94,27	0	0,10	86,42	79,06
5	412,25	43,79	94,17	0,10	0	86,32	78,96
6	405,27	49,39	87,18	86,42	86,32	0	71,85
7	345,89	46,75	25,51	79,06	78,96	71,85	0

Tabulka 5 – Mýto (€). Zdroj: vlastní zpracování

	1	2	3	4	5	6	7
1	0	51,07	36,99	54,88	54,88	51,65	43,66
2	51,07	0	3,03	7,24	7,24	3,65	7,89
3	36,99	3,03	0	14,48	14,48	11,25	3,26
4	54,88	7,24	14,48	0	0,00	10,37	11,70
5	54,88	7,24	14,48	0,00	0	10,37	11,70
6	51,65	3,65	11,25	10,37	10,37	0	8,47
7	43,66	7,89	3,26	11,70	11,70	8,47	0

Tabulka 6 – Jízdní doba (h:m). Zdroj: vlastní zpracování

	1	2	3	4	5	6	7
1	0	6:03	6:06	6:34	6:33	6:50	5:43
2	6:03	0	1:03	0:45	0:45	1:17	0:42
3	6:06	1:03	0	1:24	1:24	1:40	0:26
4	6:34	0:45	1:24	0	0:01	1:52	1:13
5	6:33	0:45	1:24	0:01	0	1:51	1:13
6	6:50	1:17	1:40	1:52	1:51	0	1:29
7	5:43	0:42	0:26	1:13	1:13	1:29	0

Tabulka 7 – Počet kilometrů pod mýtem (km). Zdroj: vlastní zpracování

	1	2	3	4	5	6	7
1	0	327,93	299,23	347,97	347,97	339,72	288,9
2	327,93	0	15,99	38,15	38,15	23,18	41,57
3	299,23	15,99	0	76,23	76,23	67,98	17,15
4	347,97	38,15	76,23	0	0	58,59	61,61
5	347,97	38,15	76,23	0	0	58,59	61,61
6	339,72	23,18	67,98	58,59	58,59	0	67,96
7	288,9	41,57	17,15	61,61	61,61	67,96	0

Po naplnění tabulek daty pomocí skriptu jsou všechny hodnoty přepočítány na stejný základ (náklady v peněžních jednotkách, tab. 8) a sečteny, aby bylo možné je optimalizovat. Poté se spustí nástroj řešitel, který se snaží nalézt optimální řešení.

Tabulka 8 – Celkové náklady mezi jednotlivými body trasy (€). Zdroj: vlastní zpracování

	1	2	3	4	5	6	7
1	0	540,13	493,15	584,91	584,64	575,79	490,03
2	540,13	0	59,14	64,09	63,97	71,36	67,60
3	493,15	59,14	0	134,79	134,68	125,67	36,28
4	584,91	64,09	134,79	0	0,26	125,62	112,96
5	584,64	63,97	134,68	0,26	0	125,36	112,85
6	575,79	71,36	125,67	125,62	125,36	0	103,71
7	490,03	67,60	36,28	112,96	112,85	103,71	0

Jakmile nedochází předem stanovenou dobu ke zlepšení současné hodnoty účelové funkce, řešitel se zastaví a interpretuje výsledky do tabulky (tab. 9).

Tabulka 9 – Souhrnné informace o jednotlivých úsecích. Zdroj: vlastní zpracování

Trasa (Z – Do)		1	3	2	4	5	6	7	1
Vzdálenost (km)	940,65	0	351,57	41,03	43,89	0,1	86,32	71,85	345,89
Čas (h)	16:58	0	6:06	1:03	0:45	0:01	1:51	1:29	5:43
Mýto (€)	109,76	0	36,99	3,03	7,24	0	10,37	8,47	43,66
Mýto (km)	768,82	0	299,23	15,99	38,15	0	58,59	67,96	288,9
Celkové náklady (€)	1335,72	0	493,15	59,14	64,09	0,26	125,36	103,71	490,03

V tuto chvíli je potřeba naměřená data („Makro“) srovnat s hodnotami reálné trasy („Původní“) používané společností „XY“. Pro lepší přehled jsou data nakopírována z původní feasibility do tabulek 10 a 11.

Tabulka 10 – Porovnání původní a nově vymodelované trasy.

Makro	1	3	2	4	5	6	7	1
Původní	1	3	2	5	4	6	7	1

Tabulka 11 – Porovnání parametrů původní a nově vymodelované trasy. Zdroj: vlastní zpracování

	Makro	Původní
Vzdálenost (km)	941	848
Čas (h)	16:58	14:53
Mýto (€)	109,76	108,59
Mýto (km)	769	764
Celkové náklady (€)	1336	1210

Obě dvě trasy jsou obslouženy téměř totožně, liší se pouze zaměněním dvou bodů 4 a 5. Jedná se o dodavatele vzdálené několik set metrů od sebe, proto je tento detail zanedbatelný. Ačkoliv jsou místa na trase seřazeny v podstatě totožně, významný rozdíl se objevuje v počtu kilometrů.

Chyba vznikla již v počátku, při získávání dat mezi jednotlivými body trasy. V M&G byl zvolen profil, který zohledňuje restriktce pro nákladní vozidla. Program bohužel chybně zamezil průjezd silnicí, kterou se v realitě kamion běžně pohybuje. Tím se znehodnotily vstupní data a výpočet je tak poměrně nepřesný. Na obrázku 20 lze vidět rozdílné vedení tras na základě nepřesných hodnot.



Obrázek 20 – Srovnání původní a nově vymodelované trasy makrem v příkladě č. 1. Zdroj: vlastní zpracování

6.1.2 Příklad č.2

Další trasa je obsluhována dodávkovým automobilem, jehož profil byl v M&G vybrán pro výpočet. Mýto se na tento typ auta nevztahuje. Postupně je třeba obslužit deset vrcholů.

Tabulka 12 – Celkové náklady mezi jednotlivými body trasy (€). Zdroj: vlastní zpracování

	1	2	3	4	5	6	7	8	9	10	11
1	0	257,9	369,7	390,7	414,3	280,2	289,3	284,3	234,4	248,0	371,0
2	257,9	0	120,7	208,8	232,4	155,4	40,6	37,9	108,9	15,8	171,8
3	369,7	120,7	0	121,0	144,7	251,3	93,8	95,5	204,5	135,2	84,0
4	390,7	208,8	121,0	0	27,7	204,8	166,1	183,5	158,0	213,6	43,1
5	414,3	232,4	144,7	27,7	0	228,1	189,5	207,0	181,3	236,9	66,6
6	280,2	155,4	251,3	204,8	228,1	0	117,7	178,4	47,0	158,7	183,9
7	289,3	40,6	93,8	166,1	189,5	117,7	0	51,0	71,2	50,4	144,8
8	284,3	37,9	95,5	183,5	207,0	178,4	51,0	0	131,9	49,7	146,6
9	234,4	108,9	204,5	158,0	181,3	47,0	71,2	131,9	0	112,2	138,2
10	248,0	15,8	135,2	213,6	236,9	158,7	50,4	49,7	112,2	0	191,4
11	371,0	171,8	84,0	43,1	66,6	183,9	144,8	146,6	138,2	191,4	0

Obě linky jsou téměř totožné až na drobný detail prohození kompletního pořadí, bodu 9 a 6 resp. 7 a 8.

Tabulka 13 – Porovnání původní a nově vymodelované trasy. Zdroj: vlastní zpracování

Makro	1	6	9	5	4	11	3	7	8	2	10	1
Původní	1	10	2	7	8	3	11	4	5	6	9	1

Po optimalizaci došlo k ušetření tří kilometrů a několik minut, což lze nejspíše přičíst k nepřesnosti M&G, protože při zadávání optimalizované posloupnosti dodavatelů do programu najednou se výpočet chová jinak než při sčítání jednotlivě vypočítaných úseků v excelu. Může docházet např. k tranzitu měst po jiných částech svých městských okruhů a s tím spojeným nárůstem nebo úbytkem vzdáleností. Celkově můžeme konstatovat, že používaná trasa na obr. 21 je optimální.

Tabulka 14 – Porovnání parametrů původní a nově vymodelované trasy. Zdroj: vlastní zpracování

	Makro	Původní
Vzdálenost (km)	883	886
Čas (h)	11:09	11:21
Mýto (€)	0	0
Mýto (km)	0	0
Celkové náklady (€)	1110	1115



Obrázek 21 – Makrem vymodelovaná trasa v příkladě č. 2. Zdroj: vlastní zpracování

6.1.3 Příklad č.3

Okruh o pěti dodavatelích je obsluhován tahačem s návěsem. Trasa je vedena Českou republikou a Německem.

Tabulka 15 – Celkové náklady mezi jednotlivými body trasy (€). Zdroj: vlastní zpracování

	1	2	3	4	5	6
1	0	91,85	92,57	150,84	243,96	427,93
2	91,85	0	0,58	64,94	158,06	373,26
3	92,57	0,58	0	65,65	158,77	373,97
4	150,84	64,94	65,65	0	113,56	391,28
5	243,96	158,06	158,77	113,56	0	326,75
6	427,93	373,26	373,97	391,28	326,75	0

Tabulka 16 – Porovnání původní a nově vymodelované trasy. Zdroj: vlastní zpracování

Makro	1	2	3	4	5	6	1
Původní	1	2	3	4	5	6	1

Tabulka 17 – Porovnání parametrů původní a nově vymodelované trasy. Zdroj: vlastní zpracování

	Makro	Původní
Vzdálenost (km)	719	737
Čas (h)	11:48	12:27
Mýto (€)	100	98
Mýto (km)	542	538
Celkové náklady (€)	1026	1052

Na téhle trase při stejném pořadí zastávek software M&G namodeloval okruh o 18 kilometrů kratší a 30 min. rychlejší. Jedná se ovšem o teoretický výpočet a v realitě se parametry mohou lišit. Je patrné, že M&G kolem bodu 4 opět zvolil lehce rozdílnou trasu z důvodu neaktuálních informací o restrikcích, a tak musí být výsledky brány s rezervou a být vždy překontrolovány.



Obrázek 22 – Srovnání původní (vlevo) a nově (vpravo) vymodelované trasy makrem. Zdroj: vlastní zpracování

6.1.4 Shrnutí

Jak bylo z výsledků patrné, celý proces optimalizace je silně závislý na datech z programu M&G. Ačkoliv dochází k pravidelným aktualizacím a vylepšení, je v podstatě nemožné mít vždy po ruce aktuální a validní data ať už z M&G nebo jiného softwaru. Může nastat problém s chybným označením zakázaných silnic pro kamiony nebo krátkodobými mimořádnostmi omezujícími provoz. Každá trasa musí být manuálně prověřena kompetentní osobou a až poté

zařazena do ostrého provozu. Kontrola současných linek z hlediska optimality tak není plně automatizována.

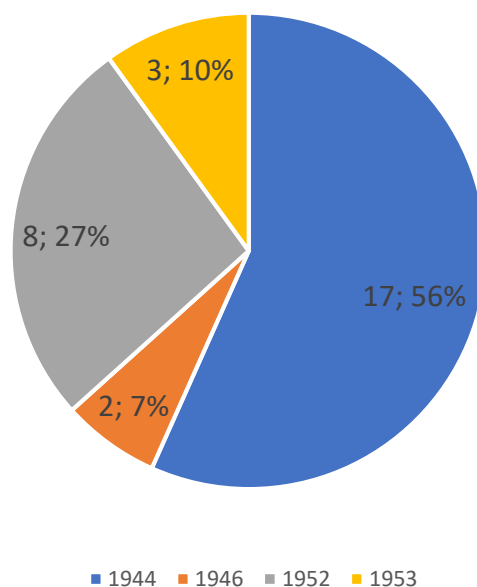
6.2 Generování trasy pomocí genetického algoritmu

Makro může sloužit zejména pro prvotní vygenerování nové trasy ze seznamu dodavatelů. Automaticky vytvoří okruh, který pak bude dodatečně zkontrolován. K demonstraci této skutečnosti jsem vybral náhodnou trasu s vysokým počtem měst, aby byla metoda evolučního algoritmu důkladně prověřena. Zisk dat pomocí skriptu trval více než 30 minut. Po několika přepočtech došlo k vytvoření následující nákladové matice.

Tabulka 18 – Celkové náklady mezi jednotlivými body trasy (€). Zdroj: vlastní zpracování

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	0	220	116	64	108	160	342	167	299	420	122	170	148	160	217	136	287	61	262	117
2	220	0	254	267	307	109	192	332	90	212	125	199	144	163	363	225	85	189	145	281
3	116	254	0	173	216	191	425	114	330	451	153	290	179	191	122	98	318	137	289	75
4	64	267	173	0	80	221	318	232	344	470	208	145	213	246	282	201	299	126	237	181
5	108	307	216	80	0	196	292	270	378	445	246	120	211	284	319	238	273	164	211	219
6	160	109	191	221	196	0	194	270	185	257	114	94	37	151	302	221	86	81	93	220
7	342	192	425	318	292	194	0	497	256	134	291	181	225	329	529	391	115	270	138	447
8	167	332	114	232	270	270	497	0	409	530	232	338	258	270	79	221	396	197	368	71
9	299	90	330	344	378	185	256	409	0	137	201	275	220	239	439	301	110	265	214	357
10	420	212	451	470	445	257	134	530	137	0	325	345	344	363	563	425	174	388	247	481
11	122	125	153	208	246	114	291	232	201	325	0	211	101	48	263	115	188	145	249	181
12	170	199	290	145	120	94	181	338	275	345	211	0	109	249	399	319	171	153	109	286
13	148	144	179	213	211	37	225	258	220	344	101	109	0	139	289	208	123	44	130	207
14	160	163	191	246	284	151	329	270	239	363	48	249	139	0	301	163	227	184	287	219
15	217	363	122	282	319	302	529	79	439	563	263	399	289	301	0	208	427	247	399	131
16	136	225	98	201	238	221	391	221	301	425	115	319	208	163	208	0	290	165	350	171
17	287	85	318	299	273	86	115	396	110	174	188	171	123	227	427	290	0	167	73	345
18	61	189	137	126	164	81	270	197	265	388	145	153	44	184	247	165	167	0	174	146
19	262	145	289	237	211	93	138	368	214	247	249	109	130	287	399	350	73	174	0	328
20	117	281	75	181	219	220	447	71	357	481	181	286	207	219	131	171	345	146	328	0

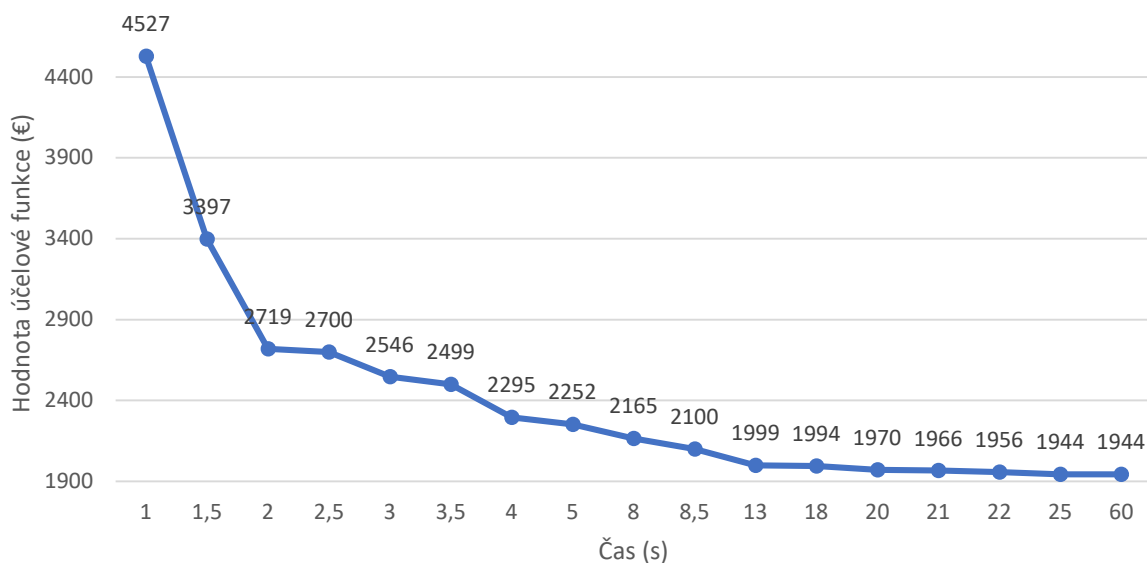
Pro lepší představu o nalezení přípustného řešení byl algoritmus spuštěn 30krát. V následujícím grafu č. 3 jsou vyobrazeny četnosti konečných hodnot účelových funkcí, ke kterým Excel Solver během výpočtu došel.



Graf 3 – Četnost finálních hodnot účelové funkce v testu. Zdroj: vlastní zpracování

Excel Solver našel nejlepší přípustné řešení v hodnotě 1944 po 25 sekundách. S přibývajícím časem nedošlo ani po 35 s k jeho zlepšení, a tak byla optimalizace automaticky ukončena (graf č. 4). K číslu 1944 došel řešitel v 56 % případů (17krát). Maximální odchylka od nejlepšího suboptima je pouze 9 € (0,46 %), což dělá ze všech výsledků poměrně kvalitní výstup pro modelování linky. Vypočítaná trasa s nejnižšími náklady začínající a končící bodem č. 1 je zobrazena na obr. 23 a vypadá následovně:

1 – 20 – 8 – 15 – 3 – 16 – 11 – 14 – 13 – 18 – 6 – 2 – 9 – 10 – 7 – 17 – 19 – 12 – 5 – 4 – 1



Graf 4 – Postup hledání optima evolučního algoritmu. Zdroj: vlastní zpracování



Obrázek 23 – Vizualizace optimalizované trasy. Zdroj: vlastní zpracování

6.3 Porovnání způsobů řešení různých algoritmů

Pro optimalizaci byl zvolen evoluční algoritmus, jenž je součástí nástroje Excel Solver. Úloha obchodního cestujícího lze ovšem řešit několika způsoby. Abychom mohli jednotlivé metody porovnat, rozhodl jsem se na stejnou množinu vrcholů se shodným počátečním bodem aplikovat jednotlivé algoritmy. Skriptem, který je součástí makra jsem vygeneroval matici (tab. č. 19), která obsahuje náklady vzniklé jízdou mezi jednotlivými městy tahačem s návěsem při dodržování restrikcí pro nákladní vozidla.

Náhodně bylo vybráno 15 měst ležících poblíž východní části České republiky (obr. 24). Depo, sloužící jako počáteční a koncový bod se nachází v Rožnově pod Radhoštěm. Na trase se nalézají několik polských měst jako Bielsko-Biala, Katowice nebo slovenských – Žilina, Bratislava, Turzovka a Trenčín. Jeden dodavatel se nachází v rakouské Vídni.

Tabulka 19 – Celkové náklady mezi jednotlivými body trasy (€). Zdroj: vlastní zpracování

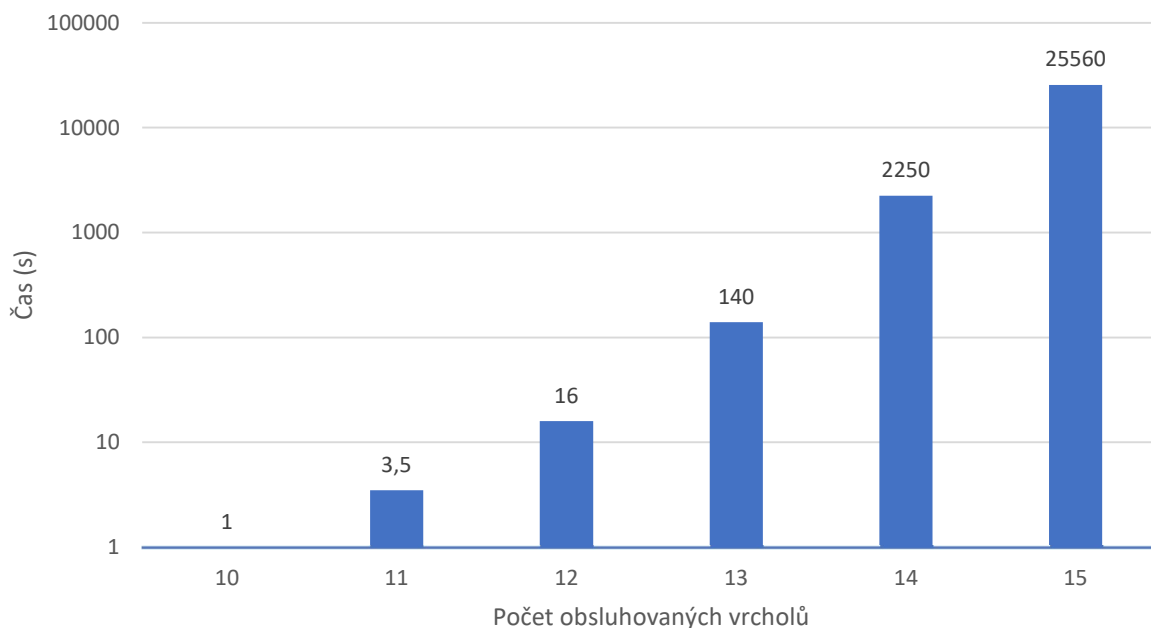
	Rožnov p. R.	Brno	Ostrava	Turzovka	Žilina	Uherské H.	Bratislava	Bielsko-Biala	Znojmo	Vídeň	Olomouc	Trenčín	Zlín	Mohelnice	Katowice
Rožnov pod Radhoštěm	0	220	116	64	108	160	342	167	299	420	122	170	148	160	217
Brno	220	0	254	267	307	109	192	332	90	212	125	199	144	163	363
Ostrava	116	254	0	173	216	191	425	114	330	451	153	290	180	191	122
Turzovka	64	267	173	0	80	221	318	232	344	470	208	145	213	246	282
Žilina	108	307	216	80	0	196	292	270	378	445	246	120	211	284	319
Uherské Hradiště	160	109	191	221	196	0	194	270	185	257	114	94	37	151	302
Bratislava	342	192	425	318	292	194	0	497	256	134	291	181	225	329	529
Bielsko-Biala	167	332	114	232	270	270	497	0	409	530	232	338	258	270	79
Znojmo	299	90	330	344	378	185	256	409	0	137	201	275	220	239	439
Vídeň	420	212	451	470	445	257	134	530	137	0	325	345	344	363	563
Olomouc	122	125	153	208	246	114	291	232	201	325	0	211	101	48	263
Trenčín	170	199	290	145	120	94	181	338	275	345	211	0	109	249	399
Zlín	148	144	180	213	211	37	225	258	220	344	101	109	0	139	289
Mohelnice	160	163	191	246	284	151	329	270	239	363	48	249	139	0	301
Katowice	217	363	122	282	319	302	529	79	439	563	263	399	289	301	0



Obrázek 24 – Vyobrazení obsluhovaných uzlů. Zdroj: vlastní zpracování

6.3.1 Metoda hrubé síly (brute-force)

Brute-force algoritmus prohledává celý prostor možných řešení. Je tudíž vysoce náročný čas. K výpočtu byl použit již existující volně šiřitelný C++ zdrojový kód algoritmu, který byl jen drobně upraven pro vložení požadovaného množství dat. Rozhodl jsem se také vypočítat úlohu po postupném odebrání několika vrcholů, abych demonstroval extrémní nárůst nebo pokles doby řešení. [27] [37]



Graf 5 – Celková doba výpočtu optima pomocí metody hrubé síly. Zdroj: vlastní zpracování

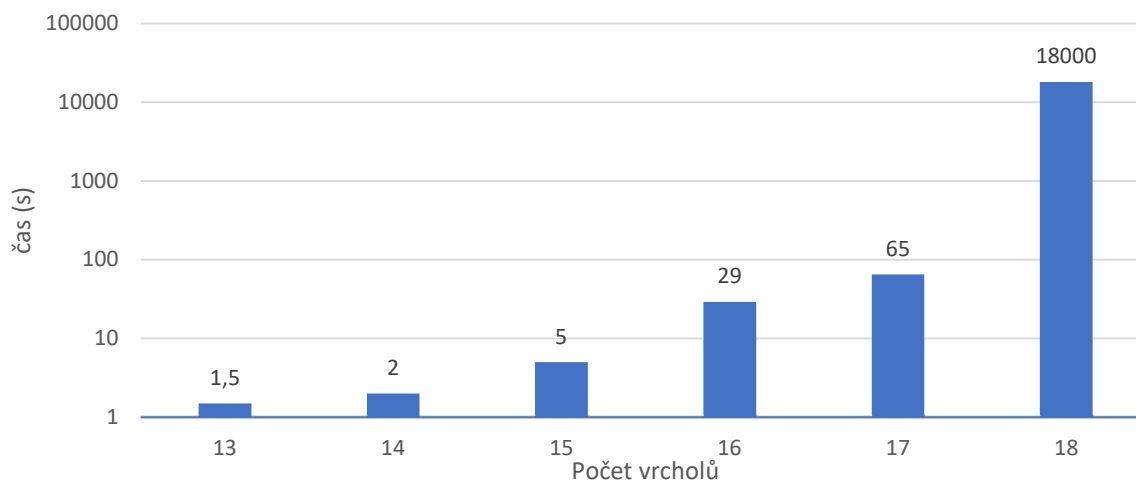
Metoda našla optimální řešení v hodnotě 1660, což představuje náklady pro trasu Rožnov pod Radhoštěm (1) – Bielsko-Biala (8) – Katowice (15) – Ostrava (3) – Mohelnice (14) – Olomouc (11) – Zlín (13) – Uherské Hradiště (6) – Brno (2) – Znojmo (9) – Vídeň (10) – Bratislava (7) – Trenčín (12) – Žilina (5) – Turzovka (4) – Rožnov pod Radhoštěm (1).



Obrázek 25 – Vizualizace trasy optimalizované metodou hrubé síly.
Zdroj: vlastní zpracování

6.3.2 Metoda větví a mezí (branch and bound)

Další z exaktních metod se dopracovala k optimálnímu řešení pomocí upraveného volně šířitelného algoritmu za velice krátkou dobu – 2 s. Nepochybně došlo k obrovskému zrychlení

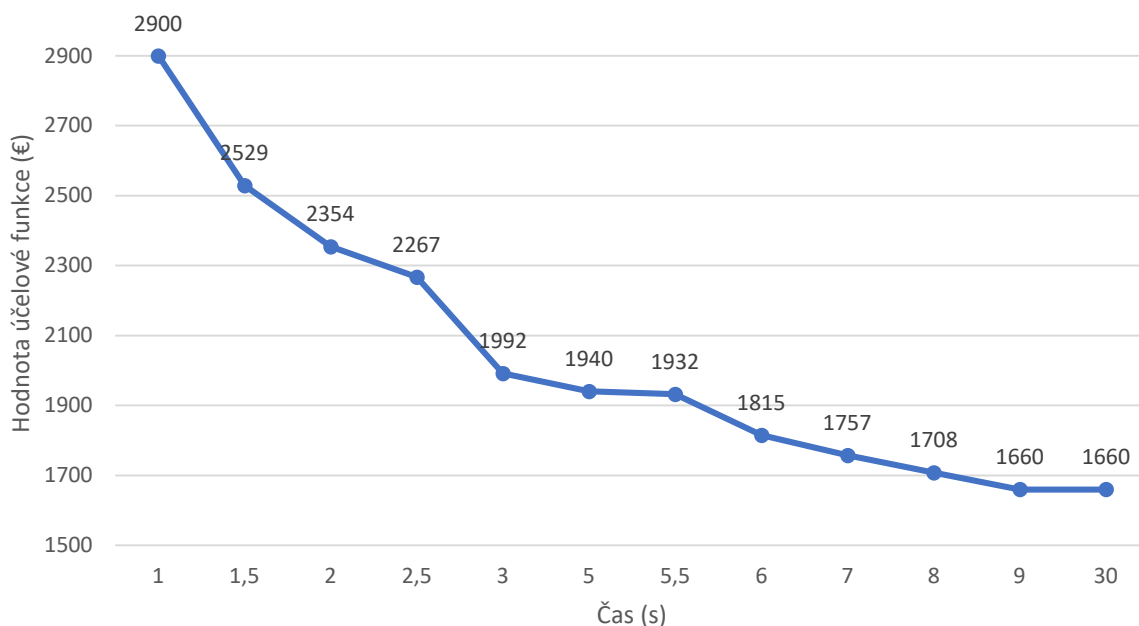


Graf 6 – Průměrná doba výpočtu optima pomocí metody větví a mezí. Zdroj: vlastní zpracování

oproti předchozí metodě hrubé síly. Pro lepší představu o času nutném k nalezení optima jsem provedl několik testů s jiným počtem uzlů (graf č. 6). [28] [38]

6.3.3 Evoluční algoritmus (Excel Solver)

Po aplikaci evolučního algoritmu nástroje Excel Solver jsem došel ke stejnému výsledku, jako při použití metody hrubé síly nebo větví a mezí. Vzhledem k poměrně nízkému počtu bodů na trase je použití evolučního algoritmu dostačující a s vysokou pravděpodobností vede k nalezení optima. K nalezení optima stačilo algoritmu 9 sekund.



Graf 7 – Postup hledání optima evolučního algoritmu. Zdroj: vlastní zpracování

6.3.4 Metoda nejbližšího souseda (nearest neighbor)

Výchozím vrcholem bylo zvolen Rožnov pod Radhoštěm, ze kterého se postupuje do dalších měst na základě nejkratší vzdálenosti. Jedná se o jednu z nejrychlejších metod je zjištění přípustného řešení. Rychlost je však kompenzována poměrně velkou nepřesností, což demonstruje výsledná hodnota účelové funkce – 1903. Algoritmus byl pro svoji triviálnost řešen manuálně.

Byla vymodelována trasa (obr. 26) v následujícím pořadí:

Rožnov pod Radhoštěm (1) – Turzovka (4) – Žilina (5) – Trenčín (12) – Uherské Hradiště (6) – Zlín (13) – Olomouc (11) – Mohelnice (14) – Brno (2) – Znojmo (9) – Vídeň (10) – Bratislava (7) – Ostrava (3) – Bielsko-Biala (8) – Katowice (15) – Rožnov pod Radhoštěm (1).



Obrázek 26 – Vizualizace trasy optimalizované metodou nejbližšího souseda. Zdroj: vlastní zpracování

6.3.5 Hladový algoritmus (greedy search)

Při výpočtu pomocí tohoto algoritmu byly dvojice měst seřazeny na základě výše nákladů (vzniklých jízdou mezi dvojicemi měst) od nejnižších po nejvyšší. Následně byla tvořena trasa postupným vkládáním hran s nejnižším ohodnocením tak, aby nedošlo k předčasnému uzavření kružnice (zacyklení). Jako první byl vybrán nejméně ohodnocený úsek mezi Uherským hradištěm a Zlínem.

Vkládány byly hrany v tomto pořadí:

Uherské Hradiště (6) – Zlín (13) – **37**

Mohelnice (14) – Olomouc (11) – **48**

Rožnov pod Radhoštěm (1) – Turzovka (4) – **64**

Katowice (15) – Bielsko-Biala (8) – **79**

Turzovka (4) – Žilina (5) – **80**

Znojmo (9) – Brno (2) – **90**

Uherské Hradiště (6) – Trenčín (12) – **94**

Olomouc (11) – Zlín (13) – **101**

Ostrava (3) – Bielsko-Biala (8) – **114**

Rožnov pod Radhoštěm (1) – Ostrava (3) – **116**

Žilina (5) – Trenčín (12) – **120**

Ostrava (3) – Katowice (15) – **122**

Bratislava (7) – Vídeň (10) – **134**

Brno (2) – Bratislava (7) – **192**

Mohelnice (14) – Vídeň (10) – **363**

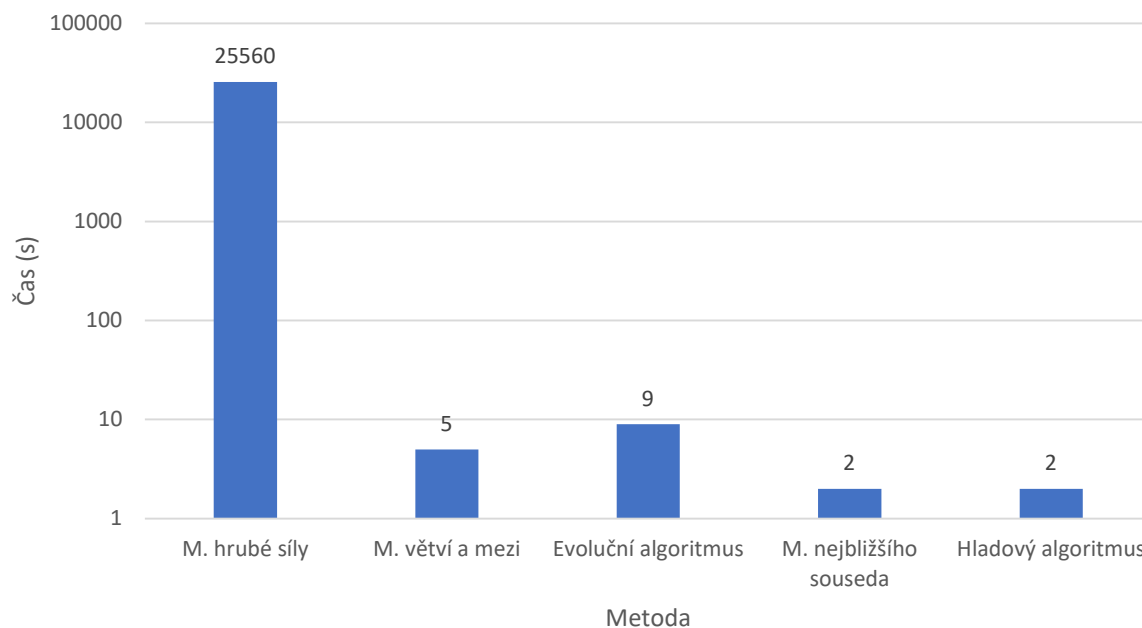
Výsledkem je Hamiltonová kružnice s ohodnocením 1754 s pořadím zastávek Rožnov pod Radhoštěm (1) – Turzovka (4) – Žilina (5) – Trenčín (12) – Uherské Hradiště (6) – Zlín (13) – Olomouc (11) – Mohelnice (14) – Vídeň (10) – Bratislava (7) – Brno (2) – Znojmo (9) – Bielsko-Biala (8) – Katowice (15) – Ostrava (3) – Rožnov pod Radhoštěm (1).



Obrázek 27 – Vizualizace trasy optimalizované metodou hladového algoritmu. Zdroj: vlastní zpracování

6.3.6 Shrnutí

Byly otestovány dva exaktní, dva heuristické a jeden metaheuristický přístup k nalezení přípustného nebo optimálního řešení úlohy. Metoda hrubé síly se jeví efektivní pro trasy s max. 13 obsluhovanými body (včetně počátečního). S vyšším počtem rapidně vzrůstá doba řešení a metoda se stává pro okamžité přepočty linek neefektivní.

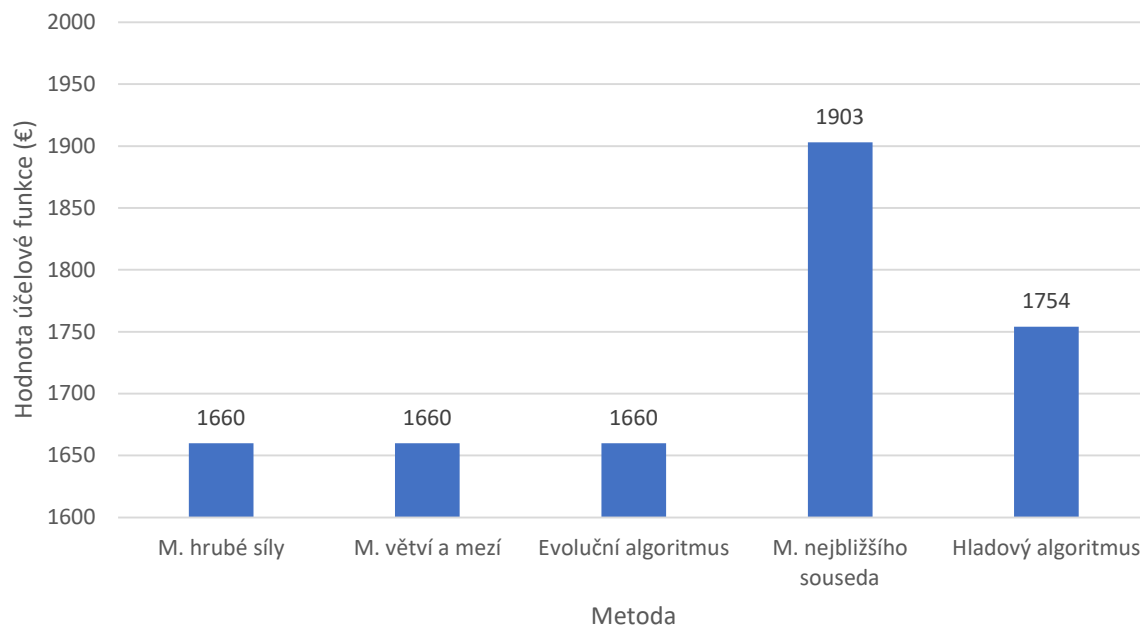


Graf 8 – Doba výpočtu příkladu s 15 vrcholy pomocí různých optimalizačních metod. Zdroj: vlastní zpracování

Jako lepší způsob z testu vyplynula metoda větví a mezí. K nalezení optima v grafu čítajícím 17 vrcholů postačila přibližně minuta. S 18 vrcholy ovšem skokově narostl čas na 5 hodin, a tak pro větší počet dodavatelů je algoritmus nevhodný.

Obě heuristické metody byly testovány manuálně, ale v případě naprogramování fungují nejrychleji z použitých metod i pro rozsáhlý počet vrcholů. Vzhledem k tomu, že běžné trasy nákladních automobilů čítají kolem 10 dodavatelů, jsou heuristiky díky svým značným nepřesnostem neefektivní. V řešeném příkladu se jednalo o odchylku 303 resp. 94 eur, což už je notné navýšení nákladů.

Ačkoliv evoluční algoritmus negarantuje optimální řešení, i při počtu přes 20 vrcholů jej (nebo suboptimum s nízkou odchylkou) nachází s vysokou pravděpodobností v relativně krátkém čase. Díky dispozici nástroje Solver v MS Excel je metoda pro řešení vysoce vhodná.



Graf 9 – Výsledná hodnota účelové funkce při výpočtu různými metodami. Zdroj: vlastní zpracování

7 Závěr

Předmětem diplomové práce bylo zautomatizovat proces tvorby feasibility, elektronického formuláře, jehož data se dále importují do plánovacích a koordinačních systémů. Feasibilita obsahuje optimální uspořádání dodavatelů okružní linky, která začíná a končí v konsolidačním centru. Mimo to obsahuje informace o času nakládky/vykládky, dodavatelích, ujetých kilometrech či počtu km pod mýtem. V minulosti proces probíhal manuálně a pořadí dodavatelů bylo určováno intuitivně, protože používaný software M&G neumí zohlednit různé kombinace obsluhovaných míst při hledání trasy.

Pomocí světově nejrozšířenějšího kancelářského nástroje MS Excel jsem se pokusil vytvořit makro, které by formulář automaticky vytvořilo a vygenerovalo (potvrdilo) optimální trasu mezi dodavateli.

Aplikace napsaná v programovacím jazyce Visual Basic for Applications (VBA) se skládá ze tří částí. Nejprve pomocí skriptu pro optimalizaci získá data, provede samotnou optimalizaci a vygeneruje výsledný formulář.

Při získávání dat se potvrdil problém s jejich nekvalitou. Ačkoliv je M&G profesionální placený software s pravidelnými aktualizacemi, může nastat situace, kdy vede trasu silnicí se zákazy pro kamiony a naopak. Tento fakt se potvrdil v jednom z příkladů. Dalším ústupkem od optimálního výsledku je fakt, že zdrojová data nevychází z nejlevnějších ale z nejrychlejších tras mezi jednotlivými úseky.

V praktické části jsem makro otestoval a popsal problémy, které mohou nastat. Nejprve byly zkontrolovány tři současné trasy, jsou-li optimální. Potvrdily se výše zmíněné obavy a ukázalo se, že ne vždy je pro kontrolu současných linek makro vhodným nástrojem. Oproti tomu lze makro využít ke generování nově vzniklých tras, které by měly být dodatečně zkontrolovány. Nakonec byly porovnány známé exaktní, heuristické a metaheuristické metody optimalizace TSP. Potvrdilo se, že nejlepší metoda pro daný problém (metoda větví a mezí) může být nahrazena GA, protože vede ve většině případů k optimálním výsledkům a je součástí excelového nástroje Solver (řešitel). Celkově je tak makro použitelné v reálném provozu, avšak musí se dbát na jeho limity a zpětně výsledky překontrolovat.

Během práce jsem si osvojil mnoho nástrojů a pokročilých funkcí MS Excel, naučil se psát složitější makra v programovacím jazyce VBA a dostal se k mnoha zajímavým zdrojům jako např. elektronické verzi originálu práce Johna D. Littla ("An algorithm for the traveling salesman problem"). Mimo jiné jsem nabyl zkušenosti ze spolupráce s nadnárodní společností, která působí na poli logistiky již desítky let.

Použitá literatura

- [1] SPOLEČNOST "XY", Logistický tým. *Konzultace*. Praha, 2020.
- [2] Tabulka AETR. In: *Centrum-preprav.cz* [online]. [cit. 2020-07-27]. Dostupné z: <http://www.centrum-preprav.cz/clanky-prilohy/aetr-vs-561.pdf>
- [3] PERNICA, Petr. *Logistika pro 21. století: (Supply chain management)*. Praha: Radix, 2005. ISBN 80-860-3159-4.
- [4] BAZALA, Jaroslav. *Aktuality / události Outsourcing – kdy, jak a proč* [online]. 12. 5. 2015 [cit. 2020-08-03]. Dostupné z: <https://www.logistickaakademie.cz/blog/aktuality/outsourcing-kdy-jak-a-proc>
- [5] JEŽEK, Vladimír. *Outsourcing služeb v logistice a skladování - způsob snížení firemních nákladů* [online]. In: . 20. 6. 2012 [cit. 2020-08-03]. Dostupné z: <https://byznys.ihned.cz/finance-rizeni-nakladu/c1-56228250-outsourcing-sluzeb-v-logistice-a-skladovani>
- [6] DASHÖFER, Verlag, TVRDOŇ, Leo, ed. *Logistické služby* [online]. In: . 17.9.2015 [cit. 2020-08-03]. Dostupné z: https://www.dlprofi.cz/log/onb/33/logisticke-sluzby-uniqueidmRRWSbk196FNf8-jVUh4EpW525SCOlV77sfcG4y5kw8/?uri_view_type=5
- [7] The Geography of Transport Systems. *Layers to Logistics Services* [online]. In: . [cit. 2020-08-03]. Dostupné z: https://transportgeography.org/wp-content/uploads/layers_logistics_services-e1567878789239.png93615804851000&source=images&cd=vfe&ved=0CAIQjRxqFw oTCJilhdPnqeoCFQAAAAAdAAAAABAD
- [8] The Geography of Transport Systems. *Cross-Docking Distribution Center* [online]. In: . 16. 11. 2018 [cit. 2020-08-03]. Dostupné z: https://transportgeography.org/wp-content/uploads/cross_docking.png
- [9] KOTORA, Bohumír. *Cross-docking zrychluje toky zboží* [online]. In: . [cit. 2020-08-03]. Dostupné z: <https://logistika.ihned.cz/c1-66342560-cross-docking-zrychluje-toky-zbozi>
- [10] *JUST IN TIME PREREQUISITES* [online]. In: . 11.11.2011 [cit. 2020-08-03]. Dostupné z: <https://corporateideasleadership-brando.blogspot.com/2011/11/just-in-time-prerequisites.html>
- [11] ROSE, Prof. Dr. Christoph a Pavel ONDRA. *Just in Sequence (3) – Co se může pokazit?* [online]. In: . 22.8.2018 [cit. 2020-08-03]. Dostupné z: <https://www.prumysloveinzenyrstvi.cz/just-in-sequence-3-co-se-muze-pokazit/>

- [12] ROSE, Prof. Dr. Christoph. *Just in Sequence Keep and Break Sequence* [online]. In: . 29.7.2017 [cit. 2020-08-03]. Dostupné z: <https://i2.wp.com/www.allaboutlean.com/wp-content/uploads/2019/07/JIS-Keep-and-break-sequence.png?resize=1024%2C502&ssl=1>
- [13] *Kanbanový Systém a kontrola Tahem* [online]. In: . [cit. 2020-08-03]. Dostupné z: <https://www.kanban-system.com/cs/kanbanovy-system-a-kontrola-tahem/>
- [14] VAVRUŠKA, Jan. *LOGISTIKA: KANBAN – dílenské řízení výroby* [online]. Liberec, 2012 [cit. 2020-08-03]. Dostupné z: <http://www.ksa.tul.cz/getFile/id:3809>. Technická univerzita v Liberci.
- [15] metalcom.cz. *Obrázek* [online]. In: . [cit. 2020-08-03]. Dostupné z: <https://www.metalcom.cz/sites/default/files/kanban/kanban2.jpg>
- [16] Plantune. *Heijunka* [online]. In: . [cit. 2020-08-03]. Dostupné z: <http://www.plantune.cz/slovník/heijunka/>
- [17] Svět produktivity a Materiály Productive system, s.r.o. *Heijunka* [online]. In: . [cit. 2020-08-03]. Dostupné z: <https://www.svetproduktivity.cz/media/Base/1210/Image1868.png>
- [18] Svět produktivity a Materiály Productive system, s.r.o. *Heijunka* [online]. In: . [cit. 2020-08-03]. Dostupné z: <https://www.svetproduktivity.cz/slovník/Heijunka.htm>
- [19] JANOTTA, David. *Milk Run – základní logistiky* [online]. In: . 3.3.2017 [cit. 2020-08-03]. Dostupné z: <https://www.autocont.cz/forum/Blogy/AC-Industry/Brezen-2017/Milk-Run-%E2%80%93-zakladni-logistiky>
- [20] *Milkrun* [online]. In: . [cit. 2020-08-03]. Dostupné z: http://1.bp.blogspot.com/-E3RkInCjmbY/TlyvGsJRhwI/AAAAAAAAABPY/sZ4iqjXLC_Y/s1600/Milk+run+logistics-4.jpg
- [21] *Eulerovské a hamiltonovské grafy* [online]. [cit. 2020-08-03]. Dostupné z: https://is.mendelu.cz/eknihovna/opory/zobraz_cast.pl?cast=23498. Mendelova univerzita v Brně.
- [22] FOLTÝNEK, Tomáš a Jana DANNHOFEROVÁ. *Teorie grafů*. Brno: Mendelova univerzita v Brně, 2011. ISBN 978-80-7375-500-3.
- [23] SOMMER, Christoph. *Hamiltonian Path through a Dodecahedron* [online]. In: . [cit. 2020-08-03].

- [24] TOTH, Paolo a Daniele VIGO. *The Vehicle Routing Problem: Monographs on Discrete Mathematics and Applications* [online]. Philadelphia: Society for Industrial and Applied Mathematics, 2001 [cit. 2020-08-05]. ISBN 0-89871-579-2.
- [25] LITTLE, John D.C., Katta G. MURPHY, Dura W. SWEENEY a Caroline KAREL. *An algorithm for the traveling salesman problem* [online]. Massachusetts Institute of Technology, 1963 [cit. 2020-08-03]. Dostupné z: <https://dspace.mit.edu/bitstream/handle/1721.1/46828/algorithmfortrav00litt.pdf>
- [26] HANZÁLEK, Zdeněk a Marie DEMLOVÁ. *Problém obchodního cestujícího* [online]. In: . [cit. 2020-08-03]. Dostupné z: <https://www.algoritmy.net/article/5407/Obchodni-cestujici>
- [27] MATHEW, Roy, Divya CHERUKUPALLI, Kevin PUSICH a Kevin ZHAO. *Traveling Salesman Algorithms: From Naive to Christofide* [online]. In: . [cit. 2020-08-03]. Dostupné z: <https://cse442-17f.github.io/Traveling-Salesman-Algorithms/>
- [28] VOLEK, Josef a Bohdan LINDA. *Teorie grafů - aplikace v dopravě a veřejné správě*. Pardubice: Univerzita Pardubice, 2012. ISBN 978-80-7395-225-9.
- [29] ŠEDA, Miloš. *Teorie grafů*. VUT FSI v Brně, listopad 2003. Dostupné také z: http://www.uai.fme.vutbr.cz/~mseda/TG03_MS.pdf
- [30] TVRDÍK, Josef. *Evoluční algoritmy* [online]. Ostravská Univerzita, únor 2010 [cit. 2020-08-03]. Dostupné z: https://www1.osu.cz/~tvrdik/wp-content/uploads/XEVALG_10.pdf
- [31] GOMEZ, Fernando, Alberto QUESADA a Roberto LOPEZ. *Genetic algorithms for feature selection* [online]. In: . [cit. 2020-08-04]. Dostupné z: https://www.neuraldesigner.com/images/genetic_algorithm.png
- [32] LASÁK, Pavel. *Co je VBA* [online]. 2012 [cit. 2020-08-04]. Dostupné z: <https://office.lasakovi.com/excel/vba/co-je-VBA/>
- [33] *PTV Map&Guide* [online]. [cit. 2020-08-04]. Dostupné z: <https://www.ptvgroup.com/en/solutions/products/ptv-mapandguide/>
- [34] *Common VBA Methods & Properties used in web automation* [online]. [cit. 2020-08-04]. Dostupné z: <http://automatetheweb.net/common-vba-methods-properties-web-automation/>
- [35] WINSTON, Wayne. *Microsoft Excel 2019 Data Analysis and Business Modeling*. 6. 2019. ISBN 978-1509305889.

- [36] *EXCEL SOLVER - CHANGE OPTIONS FOR EVOLUTIONARY SOLVING METHOD* [online]. [cit. 2020-08-04]. Dostupné z: <https://www.solver.com/excel-solver-change-options-evolutionary-solving-method>
- [37] CHAUDHARY, Nishant. *Traveling Salesman Problem (TSP) Implementation* [online]. 11.11.2017 [cit. 2020-08-04]. Dostupné z: <https://www.geeksforgeeks.org/traveling-salesman-problem-tsp-implementation/>
- [38] RAI, Anurag. *Traveling Salesman Problem using Branch And Bound* [online]. 13.10.2016 [cit. 2020-08-04]. Dostupné z: <https://www.geeksforgeeks.org/traveling-salesman-problem-using-branch-and-bound-2/>

Seznam obrázků

Obrázek 1 - Struktura společnosti a její spolupráce s automobilkou.	8
Obrázek 2 – Tok informací mezi systémy v logistickém řetězci.	12
Obrázek 3 – Úrovně outsourcingu.	15
Obrázek 4 – Ukázka implementace technologie Cross-docking.	15
Obrázek 5 – Uplatnění JIS a FIFO v praxi s vyobrazením rizika při přerušení sekvence.	17
Obrázek 6 – Zásobníky kanban připraveny k převozu na výrobní linku.	18
Obrázek 7 – Původní plán v porovnání s plánem řídicím se heijunkou.	19
Obrázek 8 – Společnost před a po zavedení technologie milkrun.	20
Obrázek 9 – Graf obsahující hamiltonovskou kružnici.	21
Obrázek 10 – Počet operací při hledání optima různými metodami v závislosti na počtu obsluhovaných měst	22
Obrázek 11 – Vývojový diagram popisující genetický algoritmus.	25
Obrázek 12 – Postupné kroky makra při tvorbě feasibility v MS Excel a vazby mezi jednotlivými nástroji.	26
Obrázek 13 – Prostřední MS Excel s automaticky spuštěným webovým prohlížečem pro interakci se skriptem po spuštění makra.	28
Obrázek 14 – Tabulka v kompatibilní podobě určená k optimalizaci nástrojem Excel Solver.	33
Obrázek 15 – Nastavení nástroje Excel Solver.	35
Obrázek 16 – Nastavení metody evoluční algoritmus v nástroji Excel Solver.	36
Obrázek 17 – Vymodelovaná zpětně vložená trasa do M&G.	38
Obrázek 18 – Výsledná podoba vygenerovaného formuláře feasibility před ručním vyplněním některých náležitostí.	39
Obrázek 19 – Výchozí stav formuláře před individuálním nastavením, vyplněním dodavatelů na požadované trase a spuštěním makra.	43
Obrázek 20 – Srovnání původní a nově vymodelované trasy makrem v příkladě č. 1.	48
Obrázek 21 – Makrem vymodelovaná trasa v příkladě č. 2.	50
Obrázek 22 – Srovnání původní a nově vymodelované trasy makrem.	51
Obrázek 23 – Vizualizace optimalizované trasy.	54
Obrázek 24 – Vyobrazení obsluhovaných uzlů.	55
Obrázek 25 – Vizualizace trasy optimalizované metodou hrubé síly.	57
Obrázek 26 – Vizualizace trasy optimalizované metodou nejbližšího souseda.	59
Obrázek 27 – Vizualizace trasy optimalizované metodou hladového algoritmu.	60

Seznam grafů

Graf 1 – Závislost počtu přípustných řešení na počtu vrcholů grafů	23
Graf 2 – Závislost doby výpočtu na množství vrcholů na trase.....	44
Graf 3 – Četnost finálních hodnot účelové funkce v testu.....	53
Graf 4 – Postup hledání optima evolučního algoritmu.....	53
Graf 5 - Celková doba výpočtu optima pomocí metody hrubé síly.....	56
Graf 6 – Průměrná doba výpočtu optima pomocí metody větví a mezí.	57
Graf 7 – Postup hledání optima evolučního algoritmu.....	58
Graf 8 – Doba výpočtu příkladu s 15 vrcholy pomocí různých optimalizačních metod.....	61
Graf 9 – Výsledná hodnota účelové funkce při výpočtu různými metodami	62

Seznam tabulek

Tabulka 1 – Pravidla AETR.....	10
Tabulka 2 – Tabulka obsahující náklady spojené s jízdou mezi jednotlivými body trasy připravená k optimalizaci.	29
Tabulka 3 – Vygenerovaná tabulka na prvním listu v Excelu s detailními informacemi pro jednotlivé úseky.	37
Tabulka 4 – Vzdálenost (km).	45
Tabulka 5 – Mýto (€).....	45
Tabulka 6 – Jízdní doba (h:m).	46
Tabulka 7 – Počet kilometrů pod mýtem (km).	46
Tabulka 8 – Celkové náklady mezi jednotlivými body trasy (€).....	46
Tabulka 9 – Souhrnné informace o jednotlivých úsecích.....	47
Tabulka 10 – Porovnání původní a nově vymodelované trasy.	47
Tabulka 11 – Porovnání parametrů původní a nově vymodelované trasy.	47
Tabulka 12 – Celkové náklady mezi jednotlivými body trasy (€).....	48
Tabulka 13 – Porovnání původní a nově vymodelované trasy.	49
Tabulka 14 – Porovnání parametrů původní a nově vymodelované trasy.	49
Tabulka 15 – Celkové náklady mezi jednotlivými body trasy (€).....	50
Tabulka 16 – Porovnání původní a nově vymodelované trasy.	51
Tabulka 17 – Porovnání parametrů původní a nově vymodelované trasy.	51
Tabulka 18 – Celkové náklady mezi jednotlivými body trasy (€).....	52
Tabulka 19 – Celkové náklady mezi jednotlivými body trasy (€).....	55

Seznam příloh

Příloha 1

Příloha 1 – Zdrojový kód

```
Sub seznam()  
Sheets("List1").Select  
n = Sheets("List1").Range("C7").Value  
  
Cells(20, 7) = "Vzdálenost"  
Cells(50, 7) = "Mýto"  
Cells(80, 7) = "Jízdní doba"  
Cells(110, 7) = "Mýto/Km"  
  
For i = 1 To n  
  
Cells(i + 11, 1) = (i)  
  
Cells(21, i + 8).Value = Cells(11 + i, 4)  
Cells(21 + i, 8).Value = Cells(11 + i, 4)  
Cells(21 + i, 8 + i) = 0  
  
Cells(51, i + 8).Value = Cells(11 + i, 4)  
Cells(51 + i, 8).Value = Cells(11 + i, 4)  
Cells(51 + i, 8 + i) = 0  
  
Cells(81, i + 8).Value = Cells(11 + i, 4)  
Cells(81 + i, 8).Value = Cells(11 + i, 4)  
Cells(81 + i, 8 + i) = 0  
  
Cells(111, i + 8).Value = Cells(11 + i, 4)  
Cells(111 + i, 8).Value = Cells(11 + i, 4)  
Cells(111 + i, 8 + i) = 0  
  
Next  
  
End Sub  
  
Sub Skript()  
  
Dim IE As Object  
Dim doc As HTMLDocument  
Dim u As Variant  
Dim n As Long  
Dim i As Long  
Dim k As Long  
Dim x As Long  
Dim y As Long  
Dim Naklady As String  
Dim delkaTrasy As String  
Dim CestovniCas As String  
  
n = Sheets("List1").Range("C7").Value  
x = 10  
y = 0  
Z = 22  
c = 10  
S = 52  
M = 1  
r = 82  
t = 112  
  
Set IE = CreateObject("InternetExplorer.Application")  
IE.Visible = True
```

```

IE.navigate "https://mginter.mapandguide.com/v7.3/?language=cs"

Do While IE.Busy
Application.Wait DateAdd("s", 1, Now)
Loop

Set doc = IE.document

'-----přihlášení

doc.getElementById("login").Value = ThisWorkbook.Sheets("List1").Cells(1,
2).Value
doc.getElementById("password").Value =
ThisWorkbook.Sheets("List1").Cells(2, 2).Value
doc.getElementById("loginuserbtn").Click

Application.Wait DateAdd("s", 60, Now)
'-----setup před spuštěním - defaultní druh vozidla

doc.getElementById("ext-gen378").Click
Application.Wait DateAdd("s", 1, Now)
Application.SendKeys "{DOWN}"
Application.SendKeys "{DOWN}"
Application.SendKeys "{DOWN}"
Application.SendKeys "{DOWN}"
Application.SendKeys "{DOWN}"
Application.SendKeys "{DOWN}"
Application.SendKeys "{DOWN}"
Application.SendKeys "~"
Application.Wait DateAdd("s", 1, Now)

'-----počítadlo
  For i = 12 To (10 + n)
'-----vložení prvního místa

doc.getElementById("mgi-addresssearchform-search2").Value =
ThisWorkbook.Sheets("List1").Cells(i, 2).Value

doc.getElementById("ext-gen517").Click

Application.SendKeys "~"

Application.Wait DateAdd("s", 3, Now)

Application.SendKeys "~"

Application.Wait DateAdd("s", 1, Now)

Cells(i, 3).Value = doc.getElementsByClassName("x-grid3-cell-inner x-grid3-
col-postCode x-unselectable")(0).innerText 'vložení PSČ atd.
Cells(i, 4).Value = doc.getElementsByClassName("x-grid3-cell-inner x-grid3-
col-city x-unselectable")(0).innerText
Cells(i, 5).Value = doc.getElementsByClassName("x-grid3-cell-inner x-grid3-
col-street x-unselectable")(0).innerText

'-----počítadlo
  For j = 13 + y To 11 + n
'-----vložení druhého místa

```

```

doc.getElementById("mgi-addresssearchform-search2").Value =
ThisWorkbook.Sheets("List1").Cells(j, 2).Value

doc.getElementById("ext-gen517").Click

Application.SendKeys "~"

Application.Wait DateAdd("s", 3, Now)

Application.SendKeys "~"

'-----vypočítej trasu
Application.Wait DateAdd("s", 1, Now)

Application.SendKeys "{F9}"

'-----vystup dat
Application.Wait DateAdd("s", 4, Now)

delkaTrasy1 = doc.getElementsByClassName("x-grid3-cell-inner x-grid3-col-
totalDistance x-unselectable")(0).innerText
u = Split(delkaTrasy1, " ")
delkaTrasy = u(0)

If delkaTrasy = "0,00" Then
delkaTrasy = "0,01"
Else: delkaTrasy = u(0)
End If

CestovniCas = doc.getElementsByClassName("x-grid3-cell-inner x-grid3-col-
totalTime x-unselectable")(0).innerText
u = Split(CestovniCas, " ")
CestovniCas = u(0)

If CestovniCas = "0:00" Then
CestovniCas = "0:01"
Else: CestovniCas = u(0)
End If

naklady1 = doc.getElementsByClassName("x-grid3-cell-inner x-grid3-col-
tollCosts x-unselectable")(0).innerText
u = Split(naklady1, " ")
Naklady = u(0)

If Naklady = "-" Then
Naklady = "0"
Else: Naklady = u(0)
End If

Cesta_s_mytem = doc.getElementsByClassName("x-grid3-cell-inner x-grid3-col-
tollDistance x-unselectable")(0).innerText
u = Split(Cesta_s_mytem, " ")
Cesta_s_mytem = u(0)

If Cesta_s_mytem = "-" Then
Cesta_s_mytem = "0"
Else: Cesta_s_mytem = u(0)
End If

```

```

Cells(Z, x).Value = delkaTrasy * 1 'horni polovina matice
Cells(Z - y + 1 + (x - c), x - (x - c + 1 - y)).Value = delkaTrasy * 1
'dolni polovina matice

Cells(r, x).Value = CestovniCas
Cells(r - y + 1 + (x - c), x - (x - c + 1 - y)).Value = CestovniCas

Cells(S, x).Value = Naklady * 1
Cells(S - y + 1 + (x - c), x - (x - c + 1 - y)).Value = Naklady * 1

Cells(t, x).Value = Cesta_s_mytem * 1
Cells(t - y + 1 + (x - c), x - (x - c + 1 - y)).Value = Cesta_s_mytem * 1

doc.getElementsByClassName("ux-row-action-item fam-delete ")(1).Click
'vymaže druhé místo
    x = x + 1

    Next j

```

```

doc.getElementsByClassName("ux-row-action-item fam-delete ")(0).Click
'vymaže první místo

```

```

x = c + M
y = y + 1
Z = Z + 1
S = S + 1
M = M + 1
r = r + 1
t = t + 1
Next i

```

```

doc.getElementById("ext-gen517").Click

```

```

Application.SendKeys "~"

```

```

Application.Wait DateAdd("s", 3, Now)

```

```

Application.SendKeys "~"

```

```

Application.Wait DateAdd("s", 1, Now)

```

```

Cells(i, 3).Value = doc.getElementsByClassName("x-grid3-cell-inner x-grid3-
col-postCode x-unselectable")(0).innerText 'vložení PSČ atd.
Cells(i, 4).Value = doc.getElementsByClassName("x-grid3-cell-inner x-grid3-
col-city x-unselectable")(0).innerText
Cells(i, 5).Value = doc.getElementsByClassName("x-grid3-cell-inner x-grid3-
col-street x-unselectable")(0).innerText

```

```

End Sub

```

```

Sub Kopie_vstupu()
Dim n As Long

```

```

Sheets("List1").Select
n = Range("C7").Value
nakladKM = Range("C8").Value
nakladH = Range("C9").Value

```

```

Sheets("List1").Select
Range(Cells(20, 8), (Cells(22 + n, 9 + n))).Select

```

```

Selection.Copy
Sheets("List2").Select
Range("D6").Select
ActiveSheet.Paste
Selection.NumberFormat = "0.000"

```

```

Sheets("List1").Select
Range(Cells(50, 8), (Cells(52 + n, 9 + n))).Select
Selection.Copy
Sheets("List2").Select
Range("E40").Select
ActiveSheet.Paste

```

```

Sheets("List1").Select
Range(Cells(80, 8), (Cells(82 + n, 9 + n))).Select
Selection.Copy
Sheets("List2").Select
Range("E76").Select
ActiveSheet.Paste
Selection.NumberFormat = "0.000"

```

```

For i = 1 To n
Cells(6 + i, 3) = (i)
Cells(7 + n, 3) = (1)

```

```

Cells(40 + i, 4) = (i)
Cells(41 + n, 4) = (1)

```

```

Cells(76 + i, 4) = (i)
Cells(77 + n, 4) = (1)

```

Next

'----- pronásobení náklady na km a h

```

i = 0
j = 0
For i = 0 To n - 1
For j = 0 To n - 1

Cells(8 + i, 5 + j).Value = Cells(8 + i, 5 + j) * nakladKM

```

Next
Next

```

i = 0
j = 0
For i = 0 To n - 1
For j = 0 To n - 1

Cells(78 + i, 6 + j).Value = Cells(78 + i, 6 + j) * 24 * nakladH

```

Next
Next

```

i = 0
j = 0
For i = 0 To n - 1

```

```

    For j = 0 To n - 1

Cells(42 + i, 6 + j).Value = Cells(42 + i, 6 + j) * 1

    Next
    Next

End Sub

Sub SoucetNakladu()

Dim i As Variant
Dim j As Variant

n = ThisWorkbook.Sheets("List1").Cells(7, 3).Value
i = 0
j = 0

For j = 0 To n - 1
For i = 0 To n - 1
ThisWorkbook.Sheets("List3").Cells(10 + i, 5 + j).Value =
(ThisWorkbook.Sheets("List2").Cells(8 + i, 5 + j).Value +
ThisWorkbook.Sheets("List2").Cells(42 + i, 6 + j).Value +
ThisWorkbook.Sheets("List2").Cells(78 + i, 6 + j).Value)

Next
Next

End Sub

Sub Kopie_nakladu_do_Resitele()

n = ThisWorkbook.Sheets("List1").Cells(7, 3).Value
i = 0
j = 0

For j = 0 To n - 1
For i = 0 To n - 1
ThisWorkbook.Sheets("List2").Cells(8 + i, 5 + j).Value =
ThisWorkbook.Sheets("List3").Cells(10 + i, 5 + j).Value

Next
Next

End Sub

Sub PosunMinim()
Application.Wait DateAdd("s", 1, Now)
Sheets("List2").Select

Sheets("List2").Range("B7").Formula = "=INDEX(matice,C7,C8)"

Range("D7").Select
Selection.Formula = "=SUM(Minima)"

n = Range("A1").Value
Range("B7").Select
Selection.AutoFill Destination:=Range("B7:B" & n + 6), Type:=xlFillDefault
Application.Wait DateAdd("s", 1, Now)

```

```

End Sub

Sub resitel()

n = ThisWorkbook.Sheets("List1").Cells(6, 3).Value

SolverReset

SolverOk SetCell:="$D$7", MaxMinVal:=2, ValueOf:=0, ByChange:="Proměnné", _
Engine:=3, EngineDesc:"Evolutionary"

SolverAdd CellRef:="Proměnné", Relation:=6, FormulaText:"Vše_různé"
SolverAdd CellRef:="$B$3", Relation:=2, FormulaText:"1"

SolverOptions MaxTime:=n, Iterations:=0, Precision:=0.000001, Convergence:=
0.0001, StepThru:=False, Scaling:=True, AssumeNonNeg:=True, Derivatives:=1
SolverOptions PopulationSize:=100, RandomSeed:=0, MutationRate:=0.075,
Multistart
:=False, RequireBounds:=True, MaxSubproblems:=0, MaxIntegerSols:=0, _
IntTolerance:=0.1, SolveWithout:=False, MaxTimeNoImp:=30

SolverSolve

End Sub

Sub Výsledekl()

Sheets("List2").Select
n = Range("A1").Value
Range("C7:C" & n + 7).Select
Selection.Copy
Sheets("List1").Select
Range("J6").Select
Selection.PasteSpecial Paste:=xlPasteAll, Operation:=xlNone,
SkipBlanks:=
False, Transpose:=True
Sheets("List2").Select
Range("D7").Select
Application.CutCopyMode = False
Selection.Copy
Sheets("List1").Select
Range("I11").Select
Selection.PasteSpecial Paste:=xlPasteValues, Operation:=xlNone,
SkipBlanks
:=False, Transpose:=False
End Sub

Sub mesta_vysledky()

Sheets("List1").Select
n = Sheets("List1").Range("C7").Value
Range("serazena_mesta_souradnice").Formula =
"=VLOOKUP(R[2]C,R12C1:R67C2,2,FALSE)" 'souradnice
Range("serazena_mesta_km").Formula =
"=VLOOKUP(R[1]C,R12C1:R67C4,4,FALSE)" 'nazvy bodu
Range("serazena_mesta_jednotlive_km").Formula =
"=(index(matice_km,j6,k6)*1)" 'počet km mezi body

```



```

Range("serazena_mesta_cas_jednotlive").Formula =
"=(index(matice_cas,j6,k6)*1)" 'cas mezi body
Range("serazena_mesta_myto_jednotlive").Formula =
"=(index(matice_myto,j6,k6)*1)" 'cas mezi body
Range("serazena_mesta_mytokm").Formula =
"=(index(matice_mytokm,j6,k6)*1)" 'myto/km

For i = 0 To n - 1
Sheets("List1").Cells(11, 11 + i) = Sheets("List2").Cells(7 + i, 2)
Next

End Sub

Sub autofill_feasibility()
n = Sheets("List1").Range("C7").Value

Sheets("List4").Select
Cells.Select
Range("S8").Activate
Selection.Delete Shift:=xlUp
Sheets("List5").Select
Range("A1:Y21").Select
Selection.Copy
Sheets("List4").Select
Range("A1").Select
ActiveSheet.Paste

For i = 0 To n - 3
Sheets("List4").Range("C10:Q10").Select
Selection.EntireRow.Insert , CopyOrigin:=xlFormatFromLeftOrAbove
Next

For i = 0 To n
Sheets("List4").Cells(8 + i, 5).Value = Sheets("List1").Cells(5, 10 +
i).Value
Next

For i = 0 To n - 1
Sheets("List4").Cells(8 + i, 15).Value = Sheets("List1").Cells(7, 11 +
i).Value * 1
Next

For i = 0 To n - 1
Sheets("List4").Cells(8 + i, 11).FormulaR1C1 = "=RC[4]/RC[5]"
Next

For i = 0 To n - 1
Sheets("List4").Cells(8 + i, 12).FormulaR1C1 = "=RC[-1]/24"
Next

For i = 0 To n - 1
Sheets("List4").Cells(9 + i, 9).FormulaR1C1 = "=R[-1]C[1]+R[-1]C[3]+R[-
1]C[5]"
Next

For i = 0 To n - 2
Sheets("List4").Cells(9 + i, 10).FormulaR1C1 = "=R[0]C[-1]+R[0]C[7]"
Next

```

```

    For i = 0 To n - 1
        Sheets("List4").Cells(8 + i, 16).Value = (Sheets("List1").Cells(7, 11 +
i).Value * 1) / ((Sheets("List1").Cells(8, 11 + i).Value * 24))
    Next

```

```

    Sheets("List4").Cells(16 + n - 2, 10).Value = Cells(11 + n - 2,
15).Value

```

```

    Sheets("List4").Cells(18 + n - 1, 14).Value = Sheets("List1").Cells(10,
9).Value

```

```

    Sheets("List4").Cells(17 + n - 2, 10).FormulaR1C1 = "=R[-6]C[2]"

```

```

    Sheets("List4").Cells(18 + n - 2, 10).FormulaR1C1 = "=R[-7]C[7]"

```

```

    Sheets("List4").Cells(20 + n - 2, 10).FormulaR1C1 = "=R[-9]C[4]"

```

```

End Sub

```

```

Sub Optimalni_trasa_PTV()

```

```

Dim IE As Object

```

```

Dim doc As HTMLDocument

```

```

Dim i As Single

```

```

n = Sheets("List1").Range("C7").Value

```

```

Set IE = CreateObject("InternetExplorer.Application")

```

```

IE.Visible = True

```

```

IE.navigate "https://mginter.mapandguide.com/v7.3/?language=cs"

```

```

Do While IE.Busy

```

```

Application.Wait DateAdd("s", 1, Now)

```

```

Loop

```

```

Set doc = IE.document

```

```

doc.getElementById("login").Value = ThisWorkbook.Sheets("List1").Cells(1,
2).Value

```

```

doc.getElementById("password").Value =
ThisWorkbook.Sheets("List1").Cells(2, 2).Value

```

```

doc.getElementById("loginuserbtn").Click
Application.Wait DateAdd("s", 15, Now)

```

```

doc.getElementById("ext-gen378").Click

```

```

Application.Wait DateAdd("s", 1, Now)

```

```

Application.SendKeys "{DOWN}"

```

```

Application.SendKeys "{DOWN}"

```

```

Application.SendKeys "~"

```

```

Application.Wait DateAdd("s", 1, Now)

```

```

For i = 0 To n

```

```

doc.getElementById("mgi-addresssearchform-search2").Value =
ThisWorkbook.Sheets("List1").Cells(4, i + 10).Value
doc.getElementById("ext-gen517").Click
Application.SendKeys "~"
Application.Wait DateAdd("s", 1, Now)
Application.SendKeys "~"
Next

```

```

Application.Wait DateAdd("s", 1, Now)
Application.SendKeys "{F9}"

```

End Sub

Sub Restart_List4()

```

    Sheets("List4").Select
    Cells.Select
    Range("S8").Activate
    Selection.Delete Shift:=xlUp
    Sheets("List5").Select
    Range("A1:Y21").Select
    Selection.Copy
    Sheets("List4").Select
    Range("A1").Select
    ActiveSheet.Paste

```

End Sub

Sub Delete()

```

Sheets("List2").Range("matice_delete").ClearContents
Sheets("List3").Range("List3_delete").ClearContents
Sheets("List1").Range("matice_mytokm_delete").ClearContents
Sheets("List1").Range("matice_cas_delete").ClearContents
Sheets("List1").Range("matice_myto_delete").ClearContents
Sheets("List1").Range("matice_km_delete").ClearContents
Sheets("List2").Range("matice2List2").ClearContents
Sheets("List2").Range("matice3List2").ClearContents
Sheets("List1").Range("serazena_mesta_delete").ClearContents
Sheets("List1").Range("adresy_delete").ClearContents
Sheets("List2").Range("minima_delete").ClearContents
Sheets("List2").Range("Proměnné_delete").ClearContents

```

Call Restart_List4

End Sub