Vysoká škola ekonomická v Praze
Fakulta informatiky a statistiky

3D problém obchodního cestujícího – heuristiky a metaheuristiky

AUTOREFERÁT DIPLOMOVÉ PRÁCE

Studijní program: Ekonometrie a operační výzkum

Autor: Bc. Vojtěch Vávra
Vedoucí práce: prof. Ing. Josef Jablonský, CSc.

Praha, květen 2022

Cíl práce

Práce se zaměřuje na aplikaci vybraných heuristik a metaheuristik k řešení problému obchodního cestujícího (TSP) v trojrozměrném prostoru. V rámci práce jsou teoreticky zahrnuty i další přístupy a snaží se o obsáhlý výčet těchto metod.

Dalším z cílů práce je zhodnotit efektivitu vybraných metod a zvolit nejvhodnější pro řešení 3D verzi TSP. Dále se v práci identifikují kvality a nedostatky jednotlivých metod.

Druhotným úkolem práce je otestovat prostředí jazyka R, zda je vhodný k případné výuce metod pro řešení úlohy obchodního cestujícího a dalších obdobných úloh pro Katedru ekonometrie na VŠE.

Nakonec práce slibuje popis vytvořených funkcí pro práci v jazyku R s 3D rozměrnými strukturami, heuristikami a metaheuristikami.

Použité metody

Metody aplikované v práci byly rozděleny do dvou skupin – heuristiky a metaheuristiky.

Heuristiky

Heuristiky jsou algoritmy, které se používají pro řešení daného problému. Ve většině případů je jejich výpočetní složitost nízká, což snižuje i výpočetní dobu, ale šance nalezení optimálního řešení je mizivá. Používají se tedy k nalezení přípustného řešení blízkého optimálnímu řešení, též označovanému jako suboptimální řešení.

Algoritmus nejbližšího souseda (Nearest neighbor, NN) se často zaměňuje s hladovým algoritmem, i když jeho hamižnost je významná a je zařazován do třídy hladových algoritmů. Jeho krása spočívá v jednoduchosti a počáteční efektivitě. Ta je ovšem ke konci algoritmu penalizována používáním delších a delších hran, které vedou k neefektivitě algoritmu. Smysl algoritmu spočívá v nalezení nejkratší hrany, kam se můžeme přesunout, a její následné použití k přesunu.

Hladový algoritmus (Greedy algorithm) se podobá algoritmu nejbližšího souseda. Hlavním rozdílem je, že nevybíráme pouze z hran, kam se můžeme přesunout z posledního navštíveného bodu, ale vždy vybíráme nejkratší přípustnou hranu na grafu. Zde se komplikuje algoritmus tím, že se musí kontrolovat, zda nedojde k vytvoření podcyklu, což by znemožnilo vytvoření Hamiltonova cyklu.

Metoda výhodnostních čísel (Clarke, Wright's savings method) spočívá ve výpočtu výhodnostní matice $S = s_{ij}$, která je složena z nejkratších hran mezi vrcholy v grafu. Vzorec pro výpočet matice S je dán vzorcem:

$$s_{ij} = d_{i1} + d_{1j} - d_{ij}, \quad i,j = 2,3,\ldots,n, \quad i \neq j. \tag{1}$$

Metoda minimální kostry (minimum spanning tree method, metoda MST) slouží pro symetrické úlohy a její složitost je $\mathcal{O}(n^2)$. Výhodou tohoto algoritmu je, že vytváří menší skupinky uzlů, které spojí nejkratší cestou. Problém ovšem nastává, když začne propojovat tyto skupinky. Kvůli tomu dochází k neefektivitě řešení. Tato metoda je ovšem díky pevnému základu vhodná pro následné úpravy pomocí algoritmů zlepšujících řešení. Součástí algoritmu je i úloha převodu Eulerova cyklu na Hamiltonův cyklus. To lze řešit Hierholzerovým algoritmem.

Metoda výměn (Lin Kerninghen method) je určena pro zlepšování již nalezeného řešení. Vylepšení řešení je dosaženo pomocí výměny 2 hran za jiné 2 hrany (2-opt). Lze provést i výměny 3 a 3 hran (3-opt). Pro výpočet tedy již předpokládáme uspořádaný Hamiltonův cyklus C popsaný jednotlivými uzly a množinou použitých hran E. Dnes je tato metoda považována za jednu z nejefektivnějších a specializuje se na ni Keld Helsgaun. Metoda končí, pokud již byly použity všechny dvojice hran a nebylo dosaženo lepšího řešení. Nebo lze také použít omezený počet výměn. V práci byla implementována verze 2-opt LK heuristiky.

Metaheuristiky

Metaheuristiky jsou přístupy, které se používají pro řešení obecných problémů. Po drobných úpravách je lze použít na dané problémy jako je například diskrétní TSP. Optimální řešení mohou nalézt, ale spíše naleznou řešení blížící se optimálnímu (suboptimální). V porovnání s exaktními metodami jsou metaheuristiky rychlejší, ale méně přesné.

V metodě prahové akceptace (Threshold acceptance, TA) se lepší hodnota účelové funkce hned neoznačí jako nejlepší, nýbrž pouze tehdy, pokud bude vylepšeno $(f(x') - f(x))$ o předem stanovenou hodnotu. Tuto hodnotu označujeme jako práh T (threshold). Pokud budeme tento práh postupně snižovat parametrem $r \in (0{,}1)$, pak zabráníme zacyklení a povolíme přechod do lokálního minima. Pokud však bude $r = 0$, jedná se o lokální hledání.

Metoda simulovaného žíhání oceli (simulated anneling method, SA, SIAM) rozšiřuje metodu lokálního hledání o stochastickou složku. Je založena na fyzikálních principech. Laicky řečeno „kováme, dokud je železo žhavé“. Zchladlá část oceli se již netvaruje a postupně celá vychladne. Pro metodu si potřebujeme stanovit hodnotu teploty $T > 0$, při které stále kováme. Dalším zmíněným prvkem je ochlazování dané $r \in (0{,}1)$. Určíme také dobu R, po kterou budeme ocel žíhat. Rozdíly v hodnotách účelových funkcí inspirovaného a inspirujícího řešení pak určuje, zda bude hned nově nalezené řešení použito nebo pokud bude horší, zda bude přesun do tohoto řešení možný. To určujeme pravděpodobností přechodu následovně:

$$P = \exp \frac{-(f(x') - f(x))}{T}. \tag{2}$$

Genetický algoritmus (Genetic algorithm, GA) je postaven na základních principech evoluce – přirozeného výběru. Přežijí silní a slabí se vytratí. Genetická informace jednotlivce je uchovávána v chromozomu. Jednotlivci tvoří populaci, která obsahuje celkem N jedinců. Ti jsou každou generací lepší a nestane se, že by slabší jedinec nahradil silnějšího jedince z populace. Platí však, že silnější jedinec nahradí slabšího v populaci. Nejlepší jedinec v populaci se označuje jako fitness. To, jak je který jedinec silný, se vypočítá pomocí účelové funkce, kterou pro případ TSP minimalizujeme. Populace prochází postupně generacemi (počet generací dán parametrem G). Noví potomci vznikají několika možnostmi:

- ▷ křížení matky a otce (případně i více jedinců),
- ▷ mutace (inverze),
- ▷ reprodukce.

Pro výběr otce a matky bylo v práci použito turnajové pravidlo. Otec s matkou jsou pak kříženi pomocí metody částečně zmapovaného křížení (Partially mapped crossover, PMX). Jakmile vznikne nové dítě, může být ještě zmutováno a k tomu bylo v práci použito prohození (Swap), náhodná (Scramble) mutace a inverze části genetického kódu. Nakonec jsou z populace vyřazeni nejslabší jedinci, aby byl zachován počet jedinců v populaci.

Optimalizace mravenčí kolonií (Ant colony optimization, ACO) vychází z myšlenky spolupráce v rámci roje. Mravenci (artificial ants = umělí mravenci, agenti) si při cestě za potravou předávají informace pomocí feromonu. Ten mravenci řekne, kudy lze dojít k potravě a které cesty již využili ostatní mravenci. Mravenec se pak sám rozhodne, jestli použije cestu, kterou už použil jiný mravenec, nebo třeba použije cesty od dvou předchůdců a zkombinuje je. Také se může rozhodnout najít úplně novou cestu. Vše se rozhoduje podle síly feromonu (čím více mravenců půjde jednou cestou, tím bude feromon silnější) a nastavením parametrů. Feromon však časem vyprchává, což dává příležitost mravenci nalézt nové neprozkoumané cesty. Díky tomu můžeme překonat lokální minima a nalézt lepší řešení, ne však nutně optimální řešení. Algoritmus se snaží simulovat chování mravenců a lze jej přizpůsobit pro větší problémy, protože ACO má 2 zásadní problémy. Za prvé má vysoké nároky na paměť ukládání hodnot feromonů. A za druhé přepočet, který uzel má být navštíven jako další, je výpočetně náročný.

Optimalizace včelí kolonií (Bee colony optimization, BCO) vychází opět z chování zvířat a opět půjde o způsob sběru potravy. Včela když objeví nový zdroj potravy, přiletí zpět k úlu a tzv. včelím tanečkem (waggle dance) v podobě osmičky sděluje úlu, kterým směrem a jak daleko se nová potrava nachází. Díky tomu se k potravě dostanou další včely. Potravu pro úlohu TSP představuje hodnota účelové funkce v podobě celkové uražené vzdálenosti. Čím je trasa kratší, tím dokáže včela přinést více nektaru. Včelí taneček může včela provést pouze tehdy, pokud její trasa byla lepší než v posledním řešení. Každá včela si tedy musí pamatovat hodnotu své nejlepší účelové funkce během výpočtu. Včela, která tančí, doporučuje svoji cestu ostatním včelkám a je nahrazena jinou včelu z úlu, aby hledala novou cestu místo ní. Při hledání cesty si včela vybere včelu θ, podle které se inspiruje v prohledávání své nové trasy.

Dosažené výsledky

Pořadí efektivity (výpočetní čas, hodnota účelové funkce) heuristik mezi sebou a metaheuristik mezi sebou uvádíme v tabulce 1. Pro metaheuristiky bylo provedeno několik výpočtů a z nich vybrány ty nejlepší výsledky. To je z důvodu náhody, která se objevuje napříč jejich aplikací.

Tabulka 1: Pořadí heuristik a metaheuristik podle času a hodnoty účelové funkce (Z).

Heuristiky			Metaheuristiky		
Metoda	Čas	Z	Metoda	Čas	Z
Nejbližší soused	2.	3.	TA	2.	2.
Hladový algoritmus	3.	4.	SIAM	3.	1.
Výhodnostní čísla	4.	2.	GA	1.	3.
Metoda MST	1.	5.	ACO	5.	4.
2-opt LK	5.	1.	BCO	4.	5.

Překvapivě dobrých výsledků dosáhla heuristika nejbližšího souseda. Nejlepších výsledků však dosáhla 2-opt LK heuristika s nejvyšší časovou náročností. Dosáhla tak dobrých řešení, že nakonec předčila i genetický algoritmus, který však pro menší úlohy dával lepší řešení.

Velmi rychlá a však špatných hodnot dosahující metoda minimální kostry by mohla být vylepšena použitím v teoretické části zmíněné Christofidovy metody, která by ovšem rapidně navýšila výpočetní čas.

Metoda výhodnostních čísel dopadla vždy lépe, co se týče hodnoty účelové funkce, než hladový algoritmus. Obě tyto metody jsou založeny na podobném principu. Nicméně metoda výhodnostních čísel je výpočetně náročnější a výpočet jí trvá déle.

Metaheuristiky ACO a BCO nedosahovaly dobrých výsledků. Pro metodu ACO by šlo rychlost řešení vylepšit vyšší koncentrací feromonu na hranách použitých některou z heuristik. Pro metodu BCO jsme již aplikovali zlepšování pomocí jednoho kroku 2-opt heuristiky a inicializací tancujících včel, jejichž cesty byly dány heuristikou nejbližšího souseda. Mohli bychom řešení vylepšit aplikací 3-opt LK heuristiky.

Nejlepší mezi metaheuristikami byly TA a SIAM, které dosahovaly velmi podobných výsledků. Metoda SIAM dosahuje nejlepších řešení. Pro tuto metodu bychom mohli dále implementovat 3-opt LK heuristiku a pro její zrychlení inicializovat řešení libovolnou rychlou heuristikou. Obdobné platí i pro TA metaheuristiku. Pro GA bychom mohli vytvořit další přístupy, podle kterých volit okolní řešení pro větší škálu řešení.

Závěrem uvádíme tabulku 2, kde vypisujeme pořadí všech aplikovaných metod podle výpočetního času a podle hodnot účelových funkcí.

Tabulka 2: Pořadí implementovaných metod podle času a hodnoty účelové funkce (Z).

Metoda	Čas	Z
Nejbližší soused	2.	7.
Hladový algoritmus	3.	9.
Výhodnostní čísla	4.	6.
Metoda MST	1.	10.
2-opt LK	5.	3.
TA	7.	2.
SIAM	8.	1.
GA	6.	4.
ACO	10.	5.
BCO	9.	8.

Vlastní přínos autora

Práce obsahuje kód k výpočtu vzdáleností, vykreslování cest na grafech ve 3D prostoru a pomocné funkce, které jsou použity pro aplikování jednotlivých metod. Aplikované metody pak jsou detailně vysvětleny i s přiloženými zápisy skriptů. Práce též obsahuje stroze zmíněné ladění parametrů metaheuristik. V neposlední řadě se v práci nachází několik tabulek a obrázků obsahující výsledky výpočetních experimentů. Práce se též jemně dotýká spolupráce sazebního prostření LaTeXa jazyka R pro usnadněný zápis tabulek.