ŘEŠENÍ PROBLÉMU LOKACE HUBŮ POMOCÍ GENETICKÉHO ALGORITMU

SOLVING THE SINGLE ALLOCATION HUB LOCATION PROBLEM USING GENETIC ALGORITHM

Miroslav Slivoně[1]

Anotace: Článek je zaměřuje na problém lokace hubů v případě, kdy huby nejsou kapacitně omezeny a obsluhované uzly jsou k hubům jednoznačně přiřazeny. Počet hubů je buď předem zadán, nebo může být předmětem optimalizace. Základní problém a jeho variace jsou v textu formulovány, poté je uveden genetický algoritmus k řešení problému. Tento algoritmus byl převeden do softwarové podoby a otestován na standardních datových souborech (CAB, AP). Ukazuje se, že genetické algoritmy jsou velice efektivním nástrojem k nalezení řešení i značně rozsáhlých instancí tohoto NP-těžkého problému.

Klíčová slova: problém lokace hubů, genetické algoritmy

Summary: This paper focuses on uncapacited single allocation hub location problem. The number of hubs can be either given or to be an object of optimization. The basic problem and its variations are formulated and then a GA-based algorithm is described in the text. This algorithm was tested on standard data sets (CAB, AP). The experiment verified the effectiveness of GAs for solving even extensive instances of this NP-hard problem.

Key words: hub location problem, genetic algorithms

1. ÚVOD

Distribuční systém je možné stručně definovat jako dopravní systém zabezpečující přepravu zboží z primárních zdrojů k zákazníkům. Typicky je v takovém systému realizována přeprava z jednoho nebo několika málo primárních zdrojů (např. výrobců zboží) přes případné terminály (distribuční sklady) k zákazníkům (jednotlivý prodejci zboží). Speciální, velice významnou třídu distribučních systémů tvoří takzvané systémy „od mnohých k mnohým“ („many to many“), ve kterých je řádově stejný počet primárních zdrojů (odesílatelů) jako zákazníků (příjemců).

Příkladem takových systémů jsou různé poštovní služby, přepravní společnosti (DHL, DPD, PPL aj.), nákladní železniční přeprava, intermodální přeprava kontejnerů, osobní letecká přeprava, ale i telekomunikační a informační sítě.

Pro všechny tyto systémy je typické, že z velkého množství primárních zdrojů je odesíláno relativně malé množství elementů (zásilek, cestujících), které se vyplatí slučovat do

[1] Ing. Miroslav Slivoně, Univerzita Pardubice, DFJP, Katedra technologie a řízení dopravy, Studentská 95, 532 10 Pardubice, Tel. +420 466 036 198, E-mail: Miroslav.Slivone@upce.cz

větších dávek v terminálech budovaných v blízkosti zdrojů, poté je ve větších dávkách přepravovat do terminálů v blízkosti cílové destinace, tam je opět rozpouštět a přepravovat na místo určení. Právě popsaná organizace přepravy se nazývá hub-and-spoke. Namísto přímé přepravy z výchozího uzlu do cílového uzlu je využíváno terminálů, tzv. hubů.

Výhoda organizace hub-and-spoke v koncentraci přepravních proudů do hubů, což umožňuje mezi huby přepravovat zásilky ekonomicky (vyšší objemy přeprav) a v přijatelných lhůtách (vyšší frekvence přeprav). Důsledkem zpracování elementů v hubech a možného prodloužení trasy dochází k prodloužení doručovacích lhůt oproti přímé přepravě, proto je vhodné organizovat prioritní přepravy a přepravy na relacích s vysokým objemem přepravního proudu jako přepravy přímé.

2. SPECIFIKACE A FORMULACE PROBLÉMU

2.1. Specifikace problému, verbální model

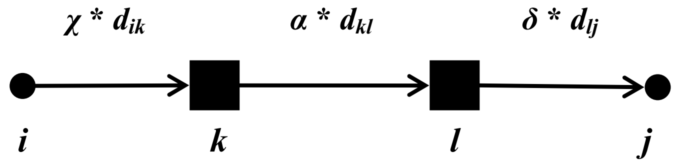
Problematika lokace hubů je poměrně široká, tento text je zaměřen na třídu problémů, ve kterých platí 3 níže uvedené předpoklady:

- *Přeprava mezi uzly je vždy realizována prostřednictvím 1 nebo 2 hubů*. Přímé přepravy mezi uzly nejsou povoleny. Přepravy realizované prostřednictvím více než dvou hubů rovněž nejsou povoleny, tj. v každém hubu budou tvořeny přímé relace všech ostatních hubů. V případě, že uzel odeslání a uzel určení leží v atrakčním obvodu stejného hubu, bude zásilka z prvního hubu přepravena do uzlu určení a přeprava tedy bude realizována prostřednictvím jediného hubu. V ostatních případech bude zásilka trasována přes dva huby.
- *Prostá alokace*. Obsluhované uzly jsou jednoznačně přiřazeny některému hubu – každý uzel patří do atrakčního obvodu právě jednoho hubu. Opačným případem je tzv. násobná alokace, kdy mohou být uzly zařazeny do atrakčních obvodů hned několika hubů současně. Výhodou násobné alokace je částečná eliminace „protisměrných“ přeprav a tudíž snížení hodnoty účelové funkce oproti alokaci prosté; nevýhodou je pak složitější organizace systému (svoz resp. rozvoz není pro konkrétní obsluhovaný uzel realizován prostřednictvím stále stejného hubu tak jako v případě prosté alokace, volba příslušné dvojice hubů záleží na konkrétní relaci výchozího uzlu i a cílového uzlu j). Problém s násobnou alokací je z hlediska výpočetní složitosti snazší a na jeho řešení existují algoritmy s polynomiální složitostí.
- *Kapacita hubů není omezená*. Každý hub může zpracovat neomezený počet zásilek.

Dopravní síť je modelována kompletním grafem G s množinou uzlů V a množinou ohodnocených hran H. Každá hrana je ohodnocena číslem d_{ij}, které reprezentuje vzdálenost uzlu i od uzlu j v reálné dopravní síti. Velikost přepravního proudu z uzlu i do uzlu j je označena jako b_{ij}.

Každá přeprava mezi uzlem i a uzlem j se skládá ze tří složek: přeprava z uzlu i do hubu k (svozní část), přeprava z hubu k do hubu l a posléze distribuce zásilky z hubu l do uzlu j

(distribuční část). Náklady na přepravu jednotkového množství toku z uzlu i do uzlu j prostřednictvím hubů k a l budou počítány dle vztahu $c_{ij} = \chi * d_{ik} + \alpha * d_{kl} + \delta * d_{lj}$.



zdroj: autor

Obr.1 – Schéma výpočtu jednotkových přepravních nákladů

Parametry χ, α, δ umožňující rozlišení nákladů na svoz, přepravu mezi huby a rozvoz. Parametry χ a δ bývají obvykle rovny 1 (v některých aplikacích se rozlišují náklady na svoz a rozvoz), volbou hodnoty parametru α lze reflektovat výši úspory přepravních nákladů vyplývající z koncentrace přeprav mezi huby (hodnota parametru α se v praktických úlohách obvykle pohybuje v rozmezí 0,6 až 0,7). Náklady na přepravu jednotkového množství c_{ij} mohou být pomocí odpovídajících hodnot parametrů χ, α, δ vyjádřeny v peněžních jednotkách při předpokladu lineárního růstu nákladů v závislosti na kilometrické vzdálenosti.

Množství přepravní práce vykonané při přepravě množství b_{ij} z uzlu i do uzlu j prostřednictvím hubů k a l se pak vypočítá jako $b_{ij} * c_{ij}$.

Cílem úlohy je rozhodnout o umístění jednotlivých hubů a o přiřazení (alokaci) obsluhovaných uzlů k těmto hubům tak, aby celková velikost vykonané přepravní práce byla minimální. Pokud jsou známy fixní náklady f_i na vybudování resp. provoz hubu za sledované období v uzlu i, je možné formulovat úlohu, která minimalizuje celkové náklady, přičemž počet hubů vystupuje v účelové funkci jako proměnná.

2.2. Matematická formulace problému

Vlastní rozhodnutí, zda je uzel i přiřazen k hubu k či nikoli, budou modelovat proměnné h_{ik}. Hodnota $h_{ik} = 1$ tedy znamená, že uzel i je přiřazen k hubu k, jinak bude hodnota $h_{ik} = 0$. Protože každý uzel k, který se stává hubem, musí být přiřazen sám sobě, proměnná $h_{kk} = 1$ zároveň indikuje, že uzel k je hub.

Úloha s pevným počtem hubů

Počet hubů je předem daný, označen jako p.

Minimalizovat $$\sum_i \sum_j b_{ij} \left(\sum_k \chi\, d_{ik}\, h_{ik} + \sum_k \sum_l \alpha\, d_{kl}\, h_{ik}\, h_{jl} + \sum_l \delta\, d_{jl}\, h_{jl} \right) + \sum_k h_{kk}\, f_k \tag{1}$$

Za podmínek: $$\sum_k h_{kk} = p \tag{2}$$

$$\sum_k h_{ik} = 1 \qquad pro \;\; i \in V \tag{3}$$

$$h_{ik} \le h_{kk} \qquad pro \;\; i, k \in V \tag{4}$$

$$h_{ik} \in \{0,1\} \qquad pro \;\; i, k \in V \tag{5}$$

Účelová funkce (1) vyjadřuje celkové náklady (součet přepravní práce vyjádřené v peněžních jednotkách a fixních nákladů). Podmínka (2) zajišťuje, že bude zvoleno právě p hubů, podmínka (3) garantuje, že každý uzel bude přiřazen právě jednomu hubu. Podmínka (4) zajišťuje, že veškeré zboží je přepravováno pouze prostřednictvím uzlů, ve kterých jsou zřízeny huby.

V případě neznámých nebo ekvivalentních fixních nákladů (např. nově budovaný systém) je možné složku účelové funkce obsahující fixní náklady vynechat. Tím nebudou zvýhodněny uzly, které již obsahují potřebnou infrastrukturu, pozemky jsou v nich levnější apod.

V anglické literatuře je tento typ problémů označován jako USApHMP (Uncapacitated Single Allocation p-Hub Median Problem). Jedná se o NP-těžkou úlohu, k jejímu řešení není znám algoritmus s polynomiální složitostí a je tedy nutné využívat některý z heuristických algoritmů.

Úloha s variabilním počtem hubů

Počet hubů vystupuje jako proměnná, fixní náklady jsou vždy známy. Formulace úlohy bude vypadat jako v předchozím případě s tím rozdílem, že ze souboru omezujících podmínek zmizí podmínka (2). V anglické literatuře bývá tento typ problémů označován jako USAHLP (Uncapacitated Single Allocation Hub Location Problem). I zde se jedná o NP-těžkou úlohu.

2.3. Přístupy používané k řešení formulovaného problému

Problém lokace hubů s prostou alokací bez kapacitního omezení poprvé formuloval O‘Kelly jako NP-těžký problém kvadratického celočíselného programování s nekonvexní účelovou funkcí [1]. Od té doby byla vyvinuta řada heuristických metoda založených na principu Branch-and-Bound, neuronových sítí, Tabu Search [2], simulovaného žíhání [3, 4] a v neposlední řadě genetických algoritmů [5, 6].

3. POPIS GENETICKÉHO ALGORITMU

Uvedený algoritmus vychází především z algoritmu GAHUB2, jehož autorem je J. Kratica a kolektiv [5], rozšiřuje jej o možnost použití pro úlohu s variabilním počtem hubů. Některé prvky algoritmu (strategie generování výchozí generace) jsou převzaty od autorů postupu popsaného v [6].

3.1. Základní schéma genetického algoritmu

Genetické algoritmy slouží k vyhledání suboptimálního řešení složitých kombinatorických úloh. Pracují na principu simulace procesu evoluce v přírodě, která spočívá v:

- zakódování řešení úlohy ve tvaru tzv. chromozomu (stavebními prvky chromozomu jsou geny) a přiřazení fitness hodnoty (reprezentuje míru kvality řešení, tj. hodnotu účelové funkce) každému chromozomu,

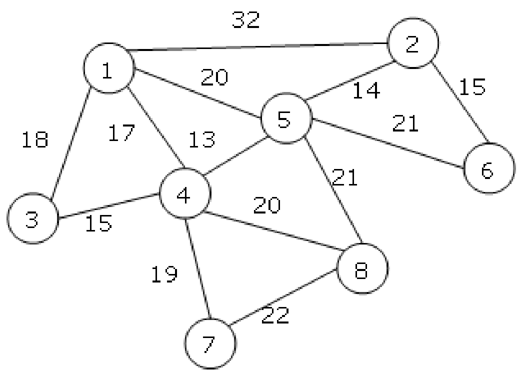
- vytvoření počáteční populace (tj. množiny chromozomů),
- výběru jedinců k reprodukci (na základě jejich fitness hodnoty),
- procesu reprodukce (pomocí operátorů křížení a mutace),
- vytvoření nové generace,
- opakování procesu simulované evoluce až do doby, dokud není dosaženo požadované hodnoty účelové funkce nebo předem definovaného počtu generací.

3.2. Zakódování úlohy

Zakódování řešení úlohy do tvaru chromozomu lze popsat následovně: délka chromozomu (tj. počet jeho genů) je rovna počtu uzlů v grafu *G*. Každý chromozom (odpovídající jednomu uzlu) se skládá ze dvou segmentů. První segment indikuje, zda je daný uzel hubem či nikoli (tj. odpovídá hodnotě proměnné h_{kk}), druhý segment vyjadřuje přiřazení daného uzlu ke konkrétnímu hubu. Obecně platí, že přiřazení uzlu k nejbližšímu hubu nemusí být vždy optimální (záleží na rozložení přepravních proudů v síti). Pro každý uzel je proto vytvořen seznam hubů uspořádaný vzestupně podle vzdálenosti jednotlivých hubů od daného uzlu. Pokud druhý segment genu nabývá hodnoty 0, je daný uzel přiřazen k němu nejbližšímu hubu, pokud nabývá hodnoty 1, je přiřazen ke druhému nejbližšímu hubu atd.

Výhodou použitého způsobu kódování pro vyjádření alokace uzlu k hubu je zachování přípustnosti řešení i po provedení operátorů křížení a mutace (tj. eliminace neplatných odkazů na uzly, které v nově získaném řešení dále nejsou huby).

Příklad: Chromozom 00 | 00 | 00 | 10 | 10 | 00 | 00 | 01 *odpovídá řešení, ve kterém jsou huby umístěny ve vrcholech 4 a 5. Vrcholy 1, 3, 4 a 7 jsou přiřazeny k hubu 4, vrcholy 2, 5, 6 a 8 jsou přiřazeny k hubu 5. Odpovídající model sítě je znázorněn na obrázku 2.*



zdroj: autor

Obr.2 – Modelová síť k uvedenému příkladu

3.3. Generování počáteční populace

Při vytváření výchozí populace o požadované velikosti (výchozí hodnota je nastavena na 150 jedinců, je možné volit jinou velikost) je použito následujících strategií.

Pro úlohu s pevným počtem hubů:

První segment každého genu nabývá hodnoty 1 s pravděpodobností p/n, kde p je požadovaný počet hubů a n je celkový počet uzlů v grafu G. Při generování hodnot prvních segmentů genů se může stát, že se skutečný počet hubů bude lišit od čísla p; v takovém případě je nutné počet hubů v daném chromozomu opravit (směr opravy řetězce je volen náhodně s pravděpodobností 0,5).

Druhý segment genu je uchován ve tvaru binárního čísla a jeho výchozí hodnota jsou samé nuly (délka binárního čísla je pro konkrétní instanci úlohy určena požadovaným počtem hubů). Generování hodnoty druhého segmentu genu je provedeno tak, že jeho poslední bit nabývá hodnoty 1 s pravděpodobností $1/n$, každý další bit směrem k začátku binárního čísla nabývá hodnoty 1 s pravděpodobností dvakrát menší (tj. $1/2n$, $1/4n$, …). Důvod je prostý: změna bitu na poslední pozici v binárním čísle vyvolá změnu hodnoty odpovídajícího decimálního čísla o ±1, změna na předposlední pozici o ±2, změna na další pozici o ±4 atd. Zvolená strategie by měla zaručovat, že velká většina uzlů bude přiřazena ke svému nejbližšímu hubu, menší počet uzlů k druhému nejbližšímu hubu, ještě méně uzlů k třetímu nejbližšímu hubu atd. Této strategie není používáno pro huby, které jsou vždy přiřazeny samy sobě.

Algoritmus umožňuje při vytváření počáteční populace zohlednit celkovou velikost přepravního toku v uzlech – v takovém případě je vytvořen seznam uzlů uspořádaný sestupně podle velikosti celkového přepravního toku v jednotlivých uzlech. Prvních 75% populace je vytvořeno tak, že se huby vybírají pouze z prvních 2/3 tohoto seznamu, pro zbývajících 25% jsou huby vybírány z celé množiny uzlů.

Pro úlohu s variabilním počtem hubů:

Počet hubů (odpovídající číslu p) je pro 75% populace generován v rozmezí $1 – n/4$, pro zbývajících 25% populace v rozmezí $n/4 – n/2$, kde n je celkový počet uzlů. Uvedená strategie je použita v [6]. S takto získanými počty hubů jsou jednotlivá řešení generována stejně jako v případě úlohy s pevným počtem hubů.

3.4. Proces selekce

Selekci, tedy výběru jedinců k reprodukci, vždy předchází výpočet hodnoty účelové funkce - fitness hodnoty pro každé řešení úlohy. Při vytváření nové generace je uplatňováno tzv. elitářství – to znamená, že definovaný počet (výchozí hodnota je 100) nejlepších řešení přímo přechází do nové generace. Smyslem elitářství je zachovat kvalitní genetický materiál.

Selekce jedinců, kteří se budou účastnit simulované reprodukce, je prováděna turnajovým způsobem - tj. z celé současné generace je vybrán určitý počet jedinců (tzv. velikost turnaje, kterou je možné zvolit), „vítězem“ turnaje se stává jedinec s nejnižší hodnotou fitness. Pomocí turnajových výběrů jsou sestaveny dvojice jedinců, kteří budou podrobeni křížení a mutaci. Počet těchto dvojic je určen jako rozdíl dané velikosti populace a počtu elitních jedinců zařazených přímo do nové generace.

3.5. Proces reprodukce

3.5.1. Křížení

Použitím operátoru křížení je vytvářena kombinace genetického materiálu dvou (kvalitních) jedinců s nadějí, že touto kombinací bude získáno řešení ještě lepší.

Pro úlohu s pevným počtem hubů:

Křížení dvou jedinců probíhá tak, že program prochází genetický kód obou křížených jedinců ve směru zprava doleva, přičemž hledá takovou pozici i, na které má první jedinec v prvním segmentu genu hodnotu 1 a druhý jedinec hodnotu 0. Pokud takovou pozici nalezne, vymění celé geny obou jedinců na této pozici. Souběžně s výše popsaným procesem program prochází genetický kód obou jedinců v opačném směru (zleva doprava), přičemž hledá takovou pozici j, na které má první jedinec v prvním segmentu genu hodnotu 0 a druhý jedinec hodnotu 1. Pokud takovou pozici nalezne, opět příslušné geny jedinců vymění. Oba procesy probíhají zároveň až do chvíle, kdy je $j \geq i$. Použitím operátoru křížení v popsaném tvaru je zaručeno, že oba nově vzniklí jedinci budou obsahovat právě p hubů. Výchozí pravděpodobnost křížení p_c je nastavena na 0,85 (tzn. s touto pravděpodobností do nové generace postupují jedinci nově vzniklí křížením, s pravděpodobností 0,15 postupují do nové generace oba původní jedinci).

Příklad, p=3 (pozice, na kterých došlo ke změně, jsou označeny tučně):

rodič1 10 | 00 | 00 | 10 | 10 | 00 | 00 | 01 *potomek1* 10 | **10** | **10** | **01** | **00** | 00 | 00 | 01
rodič2 01 | 10 | 10 | 01 | 00 | 10 | 01 | 00 *potomek2* 01 | **00** | **00** | **10** | **10** | 10 | 01 | 00

Pro úlohu s variabilním počtem hubů:

V tomto případě křížení probíhá tak, že se náhodně vygeneruje číslo r od 1 do n-1, kde n je počet uzlů. Za takto vygenerovanou pozicí r, se dva chromozomy „překříží“. Tento postup nezaručuje zachování stejného počtu hubů, což přirozeně ani není žádoucí.

Příklad, r=4 (pozice, na kterých došlo ke změně, jsou označeny tučně) :

rodič1 10 | 00 | 00 | 01 | 10 | 00 | 00 | 01 *potomek1* 10 | 00 | 00 | 01 | **00** | **10** | **01** | **00**
rodič2 01 | 10 | 10 | 01 | 00 | 10 | 01 | 00 *potomek2* 01 | 10 | 10 | 01 | **10** | **00** | **00** | **01**

3.5.2. Mutace

Účelem operátoru mutace je produkovat dosud neprozkoumaný nebo během nahrazování generací ztracený genetický materiál a zamezovat tak předčasné konvergenci úlohy k některému lokálnímu optimu. Oba segmenty každého genu jsou z tohoto důvodu s nějakou (velmi malou) pravděpodobností pozměněny. Pravděpodobnost mutace pro první segment každého genu je p_m^1/n, kde n je počet uzlů. Pravděpodobnost mutace pro druhý segment každého genu je p_m^2/n pro poslední bit, pro každý další bit směrem dopředu je pravděpodobnost mutace (z důvodu objasněného v kapitole 3.3) dvakrát nižší (tj. 0,05/n, 0,025/n,…). Kratica v algoritmu GAHUB2 používá hodnoty p_m^1 = 0,4 a p_m^2 = 0,1, které se osvědčily na základě provedení numerických testů.

Použití operátoru mutace může v případě řešení úlohy s pevným počtem hubů zapříčinit, že počet hubů bude různý od čísla *p*. Proto je třeba každý chromozom zkontrolovat a v případě potřeby doplnit / odstranit příslušný počet hubů.

3.6. Další aspekty algoritmu

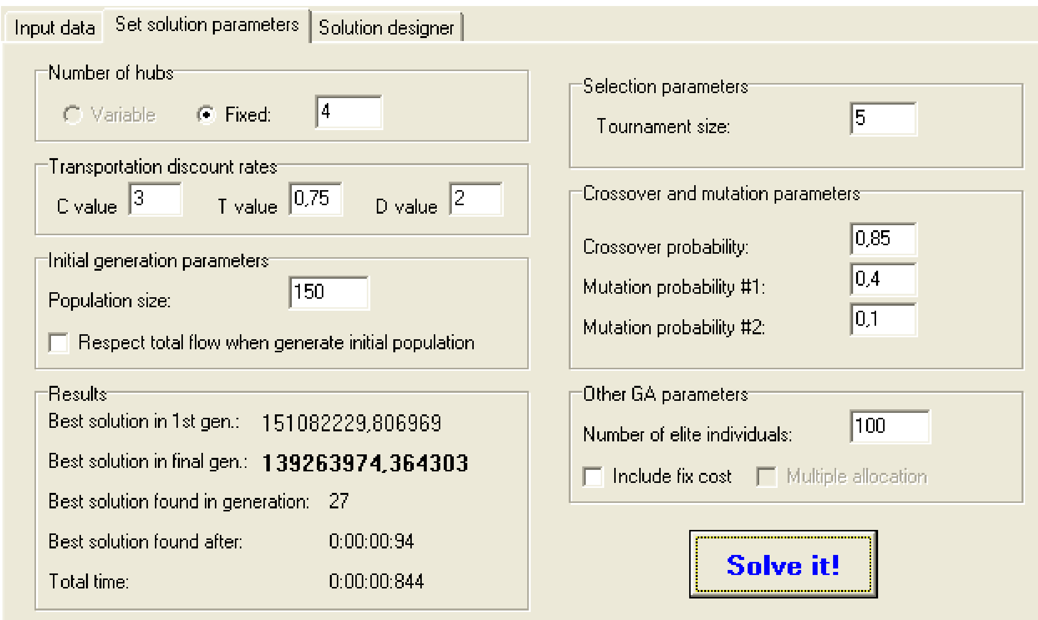
Při běhu genetického algoritmu se může stát, že geny velké většiny jedinců budou na některé pozici stejné. Takové geny se nazývají „zamrzlé" a jejich výskyt je nežádoucí, protože úloha pak může předčasně konvergovat k některému lokálnímu optimu. Proto je algoritmus doplněn o kontrolu výskytu těchto „zamrzlých" genů. Při jejich nalezení je pro tyto geny podstatně zvýšena pravděpodobnost mutace.

Počet generací lze libovolně volit, výchozí hodnota je nastavena na 500, což postačuje pro menší instance úlohy (do cca 50 uzlů), pro větší instance úlohy je vhodné počet generací navýšit.

4. SOFTWAROVÁ REALIZACE, TESTOVÁNÍ ALGORITMU

4.1. Program HubLoc

Prezentovaný algoritmus byl převeden do podoby počítačového programu HubLoc, který byl vytvořen ve vývojovém prostředí Borland Delphi 6. Program umožňuje variabilní zadávání všech parametrů genetického algoritmu popisovaných výše v textu.



zdroj: autor

Obr. 3 – HubLoc: zadávání parametrů genetického algoritmu

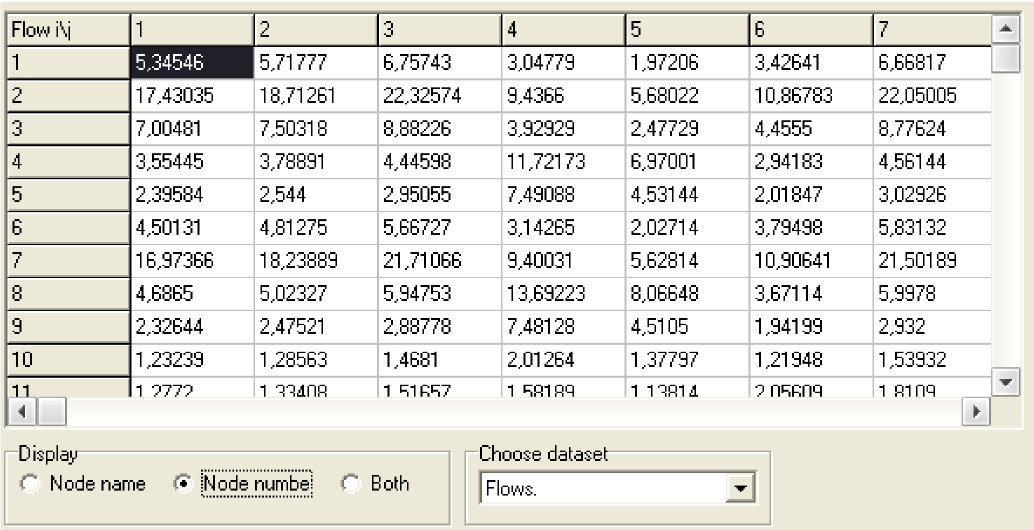
Vstupní data program načítá z textového souboru, který vychází ze souborového formátu CSV (data oddělená středníkem). Po ukončení běhu algoritmu program zobrazí hodnotu účelové funkce odpovídající nejlepšímu nalezenému řešení, číslo generace, ve které se toto řešení poprvé objevilo, a tomu odpovídající čas. Detailní výsledky včetně podoby řešení program exportuje do souboru CSV, který je možné prohlížet např. v MS Excel.

Kromě vlastní optimalizace program umožňuje návrh vlastního řešení a výpočet jemu odpovídající hodnoty účelové funkce.

4.2. Testování algoritmu – úloha s pevným počtem hubů

Pro testování algoritmu byly použity dva datové soubory. Soubor CAB (Civil Aeronautics Board) obsahující data o přepravních tocích v osobní letecké přepravě mezi 25 největšími městy v USA a soubor AP (Australian Post) obsahující data o velikosti poštovních přeprav mezi 200 městy v Austrálii. Oba soubory je možné získat na internetu [7] včetně optimálních (resp. nejlepších známých) řešení.

V textu jsou uvedeny výsledky z testování rychlosti výpočtu a kvality řešení prováděného na souboru AP. Na souboru CAB, jehož maximální velikost instance úlohy (25 uzlů) je poměrně malá, algoritmus během velice krátké doby celkem spolehlivě dosahoval optimálního řešení.



Flow i\j	1	2	3	4	5	6	7
1	5,34546	5,71777	6,75743	3,04779	1,97206	3,42641	6,66817
2	17,43035	18,71261	22,32574	9,4366	5,68022	10,86783	22,05005
3	7,00481	7,50318	8,88226	3,92929	2,47729	4,4555	8,77624
4	3,55445	3,78891	4,44598	11,72173	6,97001	2,94183	4,56144
5	2,39584	2,544	2,95055	7,49088	4,53144	2,01847	3,02926
6	4,50131	4,81275	5,66727	3,14265	2,02714	3,79498	5,83132
7	16,97366	18,23889	21,71066	9,40031	5,62814	10,90641	21,50189
8	4,6865	5,02327	5,94753	13,69223	8,06648	3,67114	5,9978
9	2,32644	2,47521	2,88778	7,48128	4,5105	1,94199	2,932
10	1,23239	1,28563	1,4681	2,01264	1,37797	1,21948	1,53932
11	1,2772	1,33408	1,51657	1,58189	1,13814	2,05609	1,8109

zdroj: autor

Obr. 4 – HubLoc: načítání AP vstupních dat

Soubor AP obsahuje kartézské souřadnice 200 měst a velikost přeprav mezi těmito uzly. A.T. Ernst a M. Krishnamoorthy jej doplnili o zdrojový text v jazyce C, který umožňuje generovat i menší instance souborů (na principu agregace daného počtu uzlů do clusterů, přepočítání souřadnic a velikosti přeprav), a o výsledky jejich výpočtů pro vybrané velikosti instancí úlohy a dané počty hubů. S využitím toho zdrojového textu byly vygenerovány instance úlohy s 10, 20, 25, 40, 50, 75, 100 a 200 uzly. Tyto soubory byly poté konvertovány do souborového formátu programu HubLoc, přičemž kartézské souřadnice uzlů byly přepočítány na Euklidovské vzdálenosti mezi všemi uzly navzájem a získanými vzdálenostmi byla naplněna distanční matice.

Tabulka 1 obsahuje výsledky řešení úlohy s pevným počtem hubů na vybraných instancích dat AP. Výsledky pro 10, 20, 25, 40 a 50 uzlů jsou porovnány s optimálními výsledky uvedenými v [7]. Parametry χ, α, δ byly nastaveny na hodnoty 3; 0,75 a 2.

Genetický algoritmus pracoval s následujícími parametry: Velikost populace 150 jedinců, 100 elitních jedinců, velikost turnaje 5, pravděpodobnost křížení 0,85, pravděpodobnost mutace 0,4/n resp. 0,1/n, počet generací byl nastaven na 500.

Výpočet byl spuštěn desetkrát, v tabulce je uveden vždy ten nejlepší výsledek, procentuální úspěšnost dosažení optima (tedy v kolika případech z 10 se výsledek shodoval s optimálním řešením), případná velikost odchylky od hodnoty optima a zaznamenány orientační časy výpočtu. Výpočty proběhly na PC s jednojádrovým procesorem Intel Pentium IV na frekvenci 3 GHz.

Tab. 1 – Výsledky pro instance souboru AP do 50 uzlů

Velikost instance n / p	Optimum uvedené v (7)	Nejlepší výsledek HubLoc	Úspěšnost nalezení optima [%]	Odchylka nejlepšího / nejhoršího dosaženého výsledku od optima [%]	Čas nejlepšího / celkový čas [s]
10 / 2	167 493,065	= opt	100	0 / 0	0,0 / 0,250
10 / 3	136 008,126	= opt	100	0 / 0	0,0 / 0,250
10 / 4	112 396,068	= opt	100	0 / 0	0,008 / 0,260
10 / 5	91 105, 371	= opt	100	0 / 0	0,00 / 0,281
20 / 2	172 816,690	= opt	100	0 / 0	0,008 / 0,531
20 / 3	151 533,084	= opt	100	0 / 0	0,016 / 0,537
20 / 4	135 624,884	= opt	100	0 / 0	0,015 / 0,570
20 / 5	123 130,095	= opt	33	0 / 0,2	0,031 / 0,578
25 / 2	175 541,978	= opt	90	0 / 0,2	0,029 / 0,652
25 / 3	155 256,323	= opt	40	0 / 0,2	0,031 / 0,703
25 / 4	139 197,169	= opt	20	0 / 0,3	0,015 / 0,719
25 / 5	123 574,289	= opt	60	0 / 0,07	0,031 / 0,760
40 / 2	177471,674	= opt	100	0 / 0	0,110 / 1,359
40 / 3	158 830,545	= opt	80	0 / 0,19	0,094 / 1,360
40 / 4	143 968,876	= opt	100	0 / 0	0,063 / 1,468
40 / 5	134 264,967	= opt	70	0 / 0,85	0,047 / 1,485
50 / 2	178 484,286	= opt	10	0 / 0,04	0,187 / 1,984
50 / 3	158 569,933	= opt	100	0 / 0	0,312 / 1,985
50 / 4	143 378,046	= opt	50	0 / 0,2	0,313 / 2,141
50 / 5	132 366,953	= opt*	5	0 / 1,02	0,375 / 2,124
* pro instanci 50 / 5 nebylo dosaženo optima pro prvních 10 opakování výpočtu, počet opakování byl proto zvýšen na 20					

zdroj: autor, [7]

Výsledky řešení úlohy pro větší počet uzlů jsou uvedeny v tabulce 2 a porovnány s výsledky uvedenými v [5]. Algoritmus byl pro každou instanci spuštěn tentokrát pouze pětkrát, tabulka obsahuje hodnotu nejlepšího takto získaného řešení. Parametry genetického algoritmu zůstaly stejné jako v předchozím případě. Počet generací byl ponechán na 500. Důsledkem zvýšení počtu generací by mohlo být nalezení lepších výsledků, ale také prodloužení doby výpočtu (zejména u instancí 200 / 10 a 200 / 20 je z časů výpočtů patrné, že nejlepší řešení bylo nalezeno v některé z posledních generací – zde by zvýšení jejich počtu zcela jistě mělo smysl).

Tab. 2 – Výsledky pro instance souboru AP do 200 uzlů

Velikost instance n / p	Nejlepší hodnota uvedená v (5) (počet generací = 5 000)	Nejlepší výsledek HubLoc (počet generací = 500)	Odchylka od nejlepší hodnoty uvedené v (5) [%]	Čas nejlepšího / celkový čas [s]
75 / 2	neuvedeno	180 118,912	-	0,406 / 4,109
75 / 3	neuvedeno	161 056,742	-	0,422 / 4,157
75 / 4	neuvedeno	145 734,205	-	0,250 / 4,344
75 / 5	neuvedeno	136 011,354	-	0,922 / 4,469
100 / 2	180 223,801	180 223,801	0	1,969 / 7,218
100 / 3	160 847,001	160 847,001	0	1,234 / 7,171
100 / 4	145 896,578	145 896,578	0	2,453 / 7,359
100 / 5	136 929,444	136 993,674	+ 0,05	4,219 / 7,375
100 / 10	106 469,566	106 671,277	+0,19	4,437 / 7,890
100 / 20	80 270,962	80 378, 209	+0,13	7,203 / 9,812
200 / 5	140 175,645	140 569,803	+0,28	16,359 / 27,437
200 / 10	110 147,657	110 351,405	+0,18	23,905 / 27,843
200 / 20	85 129,343	86 028,376	+1,06	31,734 / 32,237

zdroj: autor, [5]

4.3. Zhodnocení výsledků testování algoritmu

Pro menší instance velikosti úlohy do 50 uzlů algoritmus vždy nalezl optimální řešení. Do velikosti úlohy 20 uzlů a 4 huby dosahoval 100% úspěšnosti (tedy v 10 případech z 10), pro větší velikosti instance úlohy úspěšnost nalezení optima klesá. V těchto případech však lze těžit z rychlosti běhu algoritmu na soudobých počítačích a výpočet několikrát zopakovat. Například řešení úlohy s 50 uzly a 5 huby trvá při počtu 500 generací okolo 2 sekund – opakované provedení výpočtu při řešení praktické úlohy jistě nebude problém. I v případech, kdy algoritmus našel pouze některé suboptimální řešení, nebyla odchylka v kvalitě tohoto řešení v žádném z případů velká (cca do 1%).

Pro větší velikosti instance úlohy do 200 uzlů již nejsou známé hodnoty skutečně optimálního řešení, získané řešení je porovnáváno s nejlepšími výsledky, kterých se podařilo dosáhnout autorům v [5]. Vzhledem k faktu, že použité algoritmy jsou téměř identické, se dá očekávat podobnost výsledků, tudíž nebylo prováděno nějaké podrobnější srovnávání.

Od velikosti instance 100 uzlů a 5 hubů se během 5 opakování běhu algoritmu nepodařilo dosáhnou lepšího nebo stejně kvalitního řešení jako autorům, ale získané výsledky se jim velice blížily (odchylka nebyla větší než 1%). Je však třeba poznamenat, že autoři nastavili znatelně větší počet generací (5 000 místo 500, tím se znatelně prodloužila doba výpočtu) a prováděli výpočet vícekrát (dvacetkrát místo pětkrát).

Při porovnání výsledků s výsledky dosaženými autory v [3, 4] získaných také metaheuristikou – simulovaným žíháním (pro každou instanci byl výpočet opakován desetkrát), je pro velké instance úloh ve dvou případech dosaženo lepšího řešení, celkově se jedná o srovnatelné výsledky.

Tab. 3 – Porovnání výsledků s Ernstem a Krishnamoorthym [3, 4]

Velikost instance (n / p)	Simulované žíhání (3, 4)	Genetický algoritmus (HubLoc)
50 / 5	**132 366,953**	**132 366,953**
100 / 5	**136 929.44**	136 993,674
100 / 10	**106 469.57**	106 671,277
100 / 20	80 682.71	**80 378, 209**
200 / 5	**140 409.41**	140 569,803
200 / 10	111 088.33	**110 351,405**
200 / 20	**85 560.39**	86 028,376

zdroj: autor, [5]

5. ZÁVĚR

Genetické algoritmy mají obrovský potenciál při řešení úloh lokace hubů a v lokačních úlohách jako takových vůbec. Program HubLoc je založen na algoritmu GAHUB2 popsaným v [5]. Nezaměřuje se však jen na řešení úlohy s pevným počtem hubů, doplněna byla možnost řešit úlohu s variabilním počtem hubů a v současné době je program rozšiřován o možnost řešit úlohu s násobnou alokací uzlů k hubům.

Program HubLoc byl úspěšně použit také pro řešení úloh s reálnými daty, jmenovitě pro optimalizaci rozmístění vlakotvorných stanic [10] nebo pro optimalizaci rozmístění dep firmy DPD.

POUŽITÁ LITERATURA

[1] O'Kelly M., A quadratic integer program for the location of interacting Hub facilities. European Journal of Operational Research (1987); ISSN 0377-2217.

[2] J.G. Klincewicz, Avoiding local optima in the p-hub location problem using tabu search and grasp, Annals of Operations Research 40 (1992); ISSN: 0254-5330.

[3] Ernst, A.T., Krishnamoorthy, M. Efficient Algorithms for the Uncapacitated SingleAllocation p-hub Median Problem, Computers & Operations Research – Location Science 4 (1996), ISSN 0377-2217.

[4] Ernst A.T., Krishnamoorthy M., An exact solution approach based on shortest-paths for p-hub median problems, INFORMS Journal on Computing 10 (1998), ISSN 1091-9856.

[5] Kratica, J., Stanimirovic, Z., Tosic, D., Filipovic, V. Two Genetic Algorithms for Solving the Uncapacitated Single Allocation p-Hub Median Problem, European Journal of Operational Research 182 (2006), ISSN 0377-2217.

[6] Topcuoglu, H., Corut, F., Ermis, M., Yilmaz, G. Solving the Uncapacitated Hub Location Problem Using Genetic Algorithms, Computers & Operations Research 32 (2005), ISSN 0305-0548.

[7] J.E. Beasley, http://people.brunel.ac.uk/~mastjjb/jeb/info.html

[8] Alumur, S., Kara, B.Y. Network Hub Location Problems: the State of the Art, European Journal of Operational Research (přijato k otištění 2008), Elsevier, ISSN 0377-2217.
[9] Klincewicz, J.G. Enumeration and Search Procedures for a Hub Location Problem with Economies of Scale, Annals of Operations Research 110 (2002), Springer Netherlands, ISSN 1572-9338.
[10] Široký, J., Slivoně, M., Cempírek, V., Centra nákladní dopravy a jejich optimalizace na vybrané dopravní síti, Perner's Contacts 2 (2008), ISSN 1801-674X.

Článek vznikl za podpory Institucionálního výzkumu MSM 0021627505 „Teorie dopravních systémů" Univerzity Pardubice.

Recenzent: doc. Ing. Jaromír Široký, Ph.D.
Univerzita Pardubice, DFJP, Katedra technologie a řízení dopravy