

**České vysoké učení technické v Praze  
Fakulta dopravní**

**Czech Technical University in Prague  
Faculty of Transportation Sciences**

Ing. Denisa Mocková, Ph.D.

**Aplikace genetických algoritmů vybrané lokační úlohy**

**Application of Genetic Algorithms to Selected Location Tasks**

## **SUMMARY**

Most transportation tasks are tasks of a network nature. In the case of tasks requiring the use of a large data set there arises the problem that exact solutions are usually greatly time consuming. Therefore, in order to get a solution within an acceptable timeframe we have to find a way of solving tasks by means of different procedures. For application in practice, methods termed metaheuristics have been developed, which work on the principles of combinatorial optimisation and search for a solution within a limited space of all the admissible solutions. The advantage of metaheuristics is their polynomial time complexity, while their disadvantage is that there is no guarantee of finding the optimal solution. Thanks to their relative simplicity, the most frequently used and the most researched are genetic algorithms. The aim of this paper was to create an own design for the application of genetic algorithms and testing of genetic operators of a genetic algorithm for calculating a selected location task. The optimisation criterion was the selected minimisation of transportation work for its simplicity and possibility of calculation by way of an exact method, required for the comparison of metaheuristic results and the selection of an appropriate combination of genetic operators and parameters of the task. Since the criterion does not form part of the genetic algorithm's mechanism, in future it will be possible to change the criterion while keeping the genetic algorithm fully functional. For the purpose of testing, software has been developed for solving the given task by way of a brute-force x method and applied genetic algorithm with designed variant operators of selection, crossover and mutation and further parameters of the task. Using a smaller dataset, a test was run on all the designed variants of genetic operators and parameters that were compared with the optimal solution. The best two resultant combinations of parameters found for a different number of placed depots were then tested on a larger dataset. A comparison with the optimal solution was made only for those cases that could be calculated within an adequate time. The results of extensive tasks were gained exclusively via the genetic algorithm (optimal solutions would take a number of years for a computer to find) and were depicted in graphs. The outcome of the work is a functioning application for calculating selected discrete location tasks with the aid of a genetic algorithm. In this application the user is able to create a complete graph to be examined, to set genetic operators and their parameters and to run a calculation of genetic algorithms. At the same time the application is able to calculate a given task via an exact method (for scope-limited tasks) so that the user can compare the genetic algorithm results with the optimal solution to the task.

## SOUHRN

Většina problémů v dopravě představuje problémy, které mají síťový charakter. Při aplikaci na problémy, které vyžadují použití velkého množství dat se setkáváme s problémem, že exaktní řešení mají obvykle velké časové nároky. Proto, abychom dostali řešení v přijatelném čase, musíme najít způsoby řešení problémů jinými postupy. Pro aplikace v praxi byly vyvinuty metody, označované jako metaheuristiky, které pracují na principech kombinatorické optimalizace a hledají řešení na omezeném prostoru všech přípustných řešení. Výhodou metaheuristik je polynomiální časová náročnost, naopak nevýhodou je, že není zaručeno nalezení optimálního řešení. Nejčastěji používané a nejvíce prozkoumané jsou pro svou relativní jednoduchost genetické algoritmy.

Cílem předmětné práce byl vlastní návrh aplikace genetických algoritmů a testování genetických operátorů genetického algoritmu pro výpočet vybrané lokační úlohy. Kritériem optimalizace byla zvolena minimalizace dopravní práce pro svou jednoduchost a možnost výpočtu exaktní metodou, nutného pro porovnání s výsledky metaheuristického přístupu a volby vhodné kombinace genetických operátorů a parametrů úlohy. Vzhledem k tomu, že kritérium není součástí mechanismu genetického algoritmu, bude možné v budoucnu kritérium změnit, a přesto genetický algoritmus bude plně funkční. Pro testování byl vyvinut software pro řešení dané úlohy pomocí exaktní metody hrubé síly a aplikovaného genetického algoritmu s navrženými variantními operátory selekce, křížení a mutace a dalšími parametry úlohy. Pro menší soubor dat byly testovány všechny navržené varianty genetických operátorů a parametrů, které byly porovnány s optimálním řešením. Dvě nejlepší nalezené výsledné kombinace parametrů pro rozdílný počet umísťovaných dep byly dále testovány na větším souboru dat. S optimálním řešením byly porovnány jen případy, které byly možné vypočítat v přijatelném čase. Výsledky rozsáhlých úloh byly získány pouze genetickým algoritmem a byly graficky zobrazeny (optimální řešení by počítač hledal v řádu let).

Výstupem práce je funkční aplikace pro výpočet vybrané diskrétní lokační úlohy pomocí genetického algoritmu. V předmětné aplikaci bude moci uživatel vytvořit úplný graf, který chce zkoumat, nastavit genetické operátory a jejich parametry a spustit samotný výpočet genetického algoritmu. Zároveň je aplikace schopna vypočítat danou úlohu exaktní metodou (pro omezeně rozsáhlé úlohy), aby uživatel měl možnost porovnat výsledky genetického algoritmu s optimálním řešením úlohy.

**Klíčová slova:** lokační úloha, genetický algoritmus, genetické operátory, fitness funkce, selekce, křížení, mutace, elitářství

**Keywords:** location task, genetic algorithm, genetic operators, fitness function, selection, crossover, mutation, elitism

## **OBSAH**

<b>1. LOKAČNÍ ÚLOHY.....</b>	<b>6</b>
<b>1.1 Charakteristika vybrané dopravní lokační úlohy.....</b>	<b>7</b>
<b>2. VLASTNÍ NÁVRH GA VYBRANÉ LOKAČNÍ ÚLOHY... 7</b>	
<b>2.1 Kódování jedince, fitness funkce, elitářství, kontrola .....</b>	<b>7</b>
<b>2.2 Genetický operátor selekce .....</b>	<b>9</b>
<b>2.3 Genetický operátor křížení .....</b>	<b>10</b>
<b>2.4 Genetický operátor mutace.....</b>	<b>16</b>
<b>2.5 Vyvinutý software pro řešení vybrané lokační úlohy....</b>	<b>17</b>
<b>3. TESTOVÁNÍ A POSOUZENÍ KVALITY VÝSLEDKŮ... 17</b>	
<b>3.1 Porovnání výsledků použitých metod .....</b>	<b>18</b>
<b>4. ZÁVĚR .....</b>	<b>21</b>
<b>SEZNAM POUŽITÉ LITERATURY .....</b>	<b>24</b>
<b>CURRICULUM VITAE .....</b>	<b>25</b>

# 1. LOKAČNÍ ÚLOHY

Lokační úlohy řeší optimální rozmístění středisek obsluhy (dep) na síti vzhledem k požadavkům na obsluhu s předem známým počtem umístěvaných dep. Umístění je optimalizováno z hlediska optimalizačních kritérií, které závisí na charakteru úlohy a funkci zařízení. Kritéria mohou zahrnovat maximalizace zisku (rozdíl mezi příjmy plynoucími z poskytování služeb zákazníkům a náklady na výstavbu zařízení a jejich provozu), maximalizace objemu toků (maximální objem poskytovaných služeb), minimalizace délky toků (časové - minimalizace času potřebného k dosažení každého místa sítě, ve kterém může požadavek na obsluhu vzniknout, metrické či nákladové) při dosažení limitního objemu toků či splnění jiného limitu (doba cestování nesmí překročit jistý limit), maximalizace efektu pro zákazníka, minimalizace nákladů pro zákazníka, minimalizace dopravní práce. Při řešení lokačních úloh se uplatňuje především aparát matematické a lokační analýzy, matematické programování, optimalizace na grafech, heuristické metody a metaheuristiky. Značné potíže dosud působí rozsáhlé diskrétní lokační úlohy z důvodu kombinatorického charakteru úlohy [1].

Kombinací výše uvedených možností vzniká množství úloh, pro které je třeba aplikovat různé matematické modely a postupy řešení. Praktických postupů vedoucích k řešení lokačních úloh je velké množství. Mezi nejběžnější aplikace v dopravě patří:

- optimalizace umístění vozidel hasičské služby, rychlé záchranné služby,
- optimalizace umístění opraven automobilů,
- optimalizace umístění havarijních středisek pro plynárenskou, vodovodní, energetickou, telekomunikační síť,
- optimalizace umístění nových administrativních, komerčních či jiných objektů,
- optimalizace umístění skladů, pekáren,
- optimalizace umístění čistíren, sběren prádla,
- optimalizace umístění poštovních úřadů, bankomatů,
- optimalizace umístění čerpacích stanic pohonných hmot,
- optimalizace umístění strojů ve výrobní hale,
- optimalizace umístění skládek posypového materiálu pro zimní údržbu komunikací,
- optimalizace umístění pobočkové sítě pro servisní činnost,

- optimalizace umístění velkoobchodů,
- optimalizace umístění logistických center.

## 1.1 Charakteristika vybrané dopravní lokační úlohy

Rozsáhlá diskretní lokační úloha s deterministickým požadavkem na obsluhu. Běžné komunikační sítě, kde vrcholy představují křižovatky komunikací a hrany jednotlivé úseky komunikací. Neorientovaný graf je hranově ohodnocený, ohodnocení vyjadřuje délku hrany. Pojmem středisko je označen určitý uzel dopravní sítě, který má pro určitou část sítě, tzv. atrakční obvod, jisté zvláštní poslání. Dále jsou zadány buď požadavky ve vrcholech. Cílem je stanovit vrcholy sítě, do kterých mají být umístěna střediska obsluhy (depa) tak, aby byla minimalizována nákladová funkce. Byla zvolena nákladová funkce vyjadřující dopravní práci při obsluze vrcholů, a to z toho důvodu, abychom mohli danou úlohu řešit i exaktním výpočtem pro malý počet vrcholů sítě a srovnat oba výsledky. Poté aplikovat na rozsáhlé diskretní úloze s velkým počtem vrcholů sítě. Váhy vrcholů budou představovat počet obyvatel a tudíž důležitost vrcholu. Terminologie a základní pojmy vychází z ([2], [3]). Uvažujeme kompletní souvislý, ohodnocený, neorientovaný graf. Vybraná lokační úlohy bude řešena jednou z metaheuristik, aplikací genetických algoritmů (GA).

## 2. VLASTNÍ NÁVRH GA VYBRANÉ LOKAČNÍ ÚLOHY

Předmětná kapitola popisuje návrh genetického algoritmu, který bude přizpůsoben pro použití při výpočtu vybrané lokační úlohy. Z důvodu následného testování vybraných variant a parametrů návrhu řešení zde budou vysvětleny jednotlivé kroky genetického algoritmu v podobě vhodného kódování úlohy, volby kriteriální funkce, variantní návrhy jednotlivých genetických operátorů včetně parametricky zvolené pravděpodobnosti křížení a mutace a doplnění úlohy o stupeň elitismu. Některá řešení budou jen teoretická a nebudou vstupovat do následné softwarové aplikace vyvinuté pro testování kvality zvoleného návrhu, a to z důvodu dodržení přiměřeného rozsahu této práce ([1], [4], [5], [6], [7], [8], [9], [10]).

### 2.1 Kódování jedince, fitness funkce, elitářství, kontrola

#### Kódování jedince

Jedinec je vybraná podmnožina středisek obsluhy - dep ( $k$ -tice) z celkového počtu  $n$  uzlů. Reprezentace jedince je přirozené zobrazení podmnožiny dep,

tj. pole celých čísel o  $k$  prvcích, čísla v poli jsou v rozsahu  $1-n$ . Z hlediska implementace algoritmu mutace (a také křížení) je výhodnější zvolit reprezentaci, která se běžně používá pro implementaci množiny v programovacích jazycích. Jde o zobrazení charakteristické funkce množiny. Pro celkový počet  $n$  uzlů je jedinec reprezentován jako bitové pole o  $n$  prvcích. Patří-li uzel  $i$  do vybrané podmnožiny  $dep$ , má prvek (bit) v poli s indexem  $i$  hodnotu 1, jinak 0. U každého jedince bude v poli  $k$  jedniček. Máme celkem  $n = 8$  uzlů označených  $v1$  až  $v8$ . Nechť  $k = 4$ . Podmnožina  $dep$  ( $v1, v4, v5, v8$ ) bude reprezentována tak, jak ukazuje obrázek 1.

V1	V2	V3	V4	V5	V6	V7	V8
1	0	0	1	1	0	0	1

*Zdroj: autor*

Obrázek 1 Reprezentace jedince

### Fitness funkce

Kritériem kvality je dopravní práce podle vztahu (1) a (2), kterou budeme minimalizovat.

$$f(D_k) = \min_{D'_k} \{f(D'_k)\} \quad (1)$$

$$f(D'_k) = \sum_{v \in D'_k} \sum_{u \in A^+(v)} 2 \cdot d(u, v) \cdot w(u) \quad (2)$$

kde:  $D'_k$  - všechny  $k$  - prvkové podmnožiny  $V$ ,  $d(u, v)$  - vzdálenost mezi dvěma vrcholy,  $w(v)$  - váha vrcholu  $v$ , počet požadavků na obsluhu, důležitost vrcholu apod..

### Elitářství

Vedlejším parametrem algoritmu je elitářství, které může nabývat stupně jedna až  $(n-3)$ , kde  $n$  vyjadřuje počet jedinců v generaci. Uvedený rozsah je zvolen proto, aby při jakémkoliv nastavení mohl algoritmus běžet dále (při nastavení terciální selekce by algoritmus neměl již z čeho vybírat). Prakticky však budeme volit elitářství v rozmezí  $0-2$ .

### Kontrola jedinců v generaci jedinců



Kontrola platných jedinců probíhá průběžně v operaci křížení a v mutaci. Genetické operátory jsou navrženy tak, aby respektovaly platnost jedinců (stejný počet dep/počet jedniček v jedinci). Dále probíhá průběžná kontrola v každé iteraci a při každé operaci křížení a mutace, aby dva jedinci nebyli shodní s již přiřazených do nové generace. Naším požadavkem je, aby v nové generaci se předmětné duplicity nenacházely, proto označení jedinci jsou zahazeni a počítá se další množina jedinců, dokud se nová generace nenaplní unikátními jedinci.

## 2.2 Genetický operátor selekce

Pro úlohu byla variantně zvolena binární a terciální turnajová selekce a vážená ruleta.

- binární – z předchozí generace náhodně vybíráme dva jedince a z nich posouváme dál jedince s lepší hodnotou fitness funkce,
- terciální – náhodně vybíráme tři jedince a z nich nejlepší (s nejnižší hodnotou fitness funkce) pokračuje v algoritmu dále,
- vážená ruleta - z důvodu zvolení selekce váženou ruletou, musí být fitness funkce transformována, protože při výpočtu kritériální funkce jsou lepší jedinci ohodnoceni menší hodnotou. Algoritmy výběru pak předpokládají, že pravděpodobnost výběru  $i$ -tého jedince do další generace je úměrná hodnotě fitness  $f_i$  podle vztahu (3).

Lineární transformace podle vztahu uvedeného v [11]:

$$fitness = 1 + \frac{(s-1)(DP - avg)}{best - avg} \quad (3)$$

kde  $avg$  je průměrná hodnota dopravní práce (DP) v generaci,  $best$  je nejlepší hodnota DP v generaci. Parametr  $s$  (obvykle v rozsahu 1,2-2) řídí selekční tlak. Aby jedinci s hodnotou dopravní práce horší než průměr neměli zápornou hodnotu fitness, volí se podle [11] alternativně  $s$ :

$$s = 1 + \frac{best - avg}{best - worst} \quad (4)$$

Pro danou hodnotu  $s$  mají nejhorší jedinci hodnotu fitness nulovou. Při popsaném postupu transformace fitness se nejhorší jedinci v žádném případě nedostanou do další generace, což může být někdy nevýhoda (mutací špatného jedince se může stát dobrý).

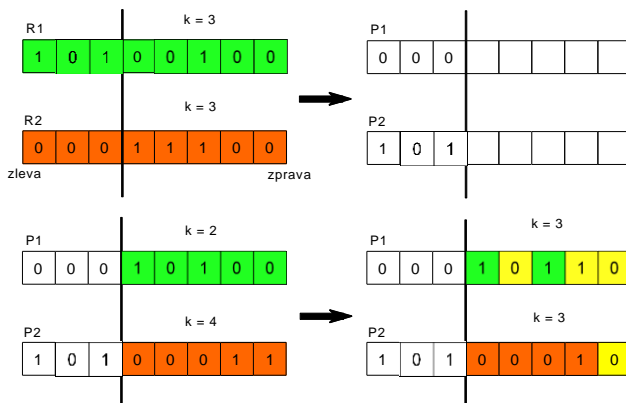
## 2.3 Genetický operátor křížení

Z předchozího kroku (selektce) nám vystupuje průběžně množina dvou jedinců až do naplnění generace, kteří přistupují dále ke křížení, které proběhne v závislosti na pravděpodobnosti křížení (označeno  $P_C$ ). Pokud ke křížení dojde, tak je možných 8 variant, které budou sledovány. Dalších devět variant křížení bude jen teoreticky naznačeno. Jednotlivé varianty křížení se liší podobou kopírování části genetické informace z rodiče na potomka a doplněním jedince na kompletního jedince. Kopírování probíhá buď napevno (pro celý průběh algoritmu) nebo opět náhodně určíme, zda u jedinců budeme kopírovat levou či pravou stranu pro zvýšení stupně volnosti genetického algoritmu. Potom levá či pravá strana R1 se zkopíruje do P2 a obdobně z R2 do P1.

V dalším kroku je potřeba určit, jak se provede doplnění zbývajících částí potomků. Opět záleží na způsobu zakódování jedince. Dané doplnění musí fungovat tak, aby se po zpracování tvořili platní jedinci.

Zvolenými způsoby tedy jsou:

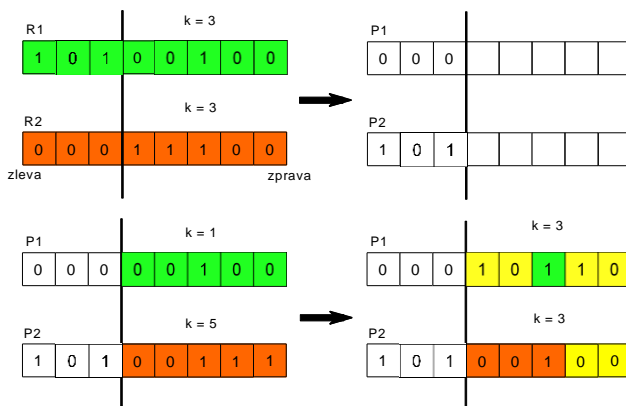
**Křížení jednobodové – Varianta 1 (LL)** – při daném křížení se náhodně vybere bod křížení. Budeme kopírovat napevno (pro celý průběh algoritmu) levou stranu (LS) rodičů (R1, R2) do potomků (P1, P2) – (LS R1 do P2 a LS R2 do P1) a doplnění na platného jedince bude probíhat zleva (doplnění P1 probíhá od bodu křížení vpravo rodičem R1 zleva). Doplnění na platného jedince probíhá na jednotlivých pozicích (genech) do doby, kdy jedničky dosáhnou celkového počtu  $k$  jedniček (počtu rozmístřovaných dep ( $k$ )). V opačném případě, kdy jsme doplněním na platného jedince nedosáhli požadovaného počtu jedniček v jedinci, náhodně zvolíme gen/y, kde bude/ou jednička/y doplněna/y, ale pouze v doplňovací části jedince, jak znázorňuje obrázek 2 .



Zdroj: autor

Obrázek 2 Princip jednobodového křížení – Varianta 1

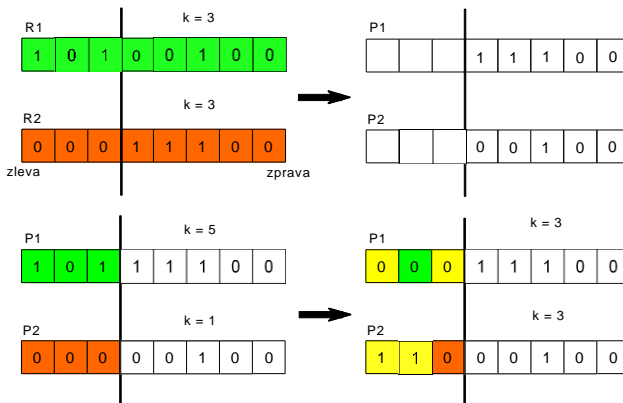
**Křížení jednobodové – Varianta 2 (LR)** – při křížení se náhodně vybere bod křížení. Budeme kopírovat napevno (pro celý průběh algoritmu) levou stranu (LS) rodičů (R1, R2) do potomků (P1, P2) – (LS R1 do P2 a LS R2 do P1) a doplnění na platného jedince bude probíhat zprava (doplnění P1 probíhá od bodu křížení vpravo rodičem R1 zprava). Doplnění na platného jedince probíhá na jednotlivých pozicích (genech) do doby, kdy jedničky dosáhnou celkového počtu  $k$  jedniček (počtu rozmístěovaných dep ( $k$ )). V opačném případě, kdy jsme doplněním na platného jedince nedosáhli požadovaného počtu jedniček v jedinci, náhodně zvolíme gen/y, kde bude/ou jednička/y doplněna/y, ale pouze v doplňovací části jedince, jak znázorňuje obrázek 3.



Zdroj: autor

Obrázek 3 Princip jednobodového křížení – Varianta 2

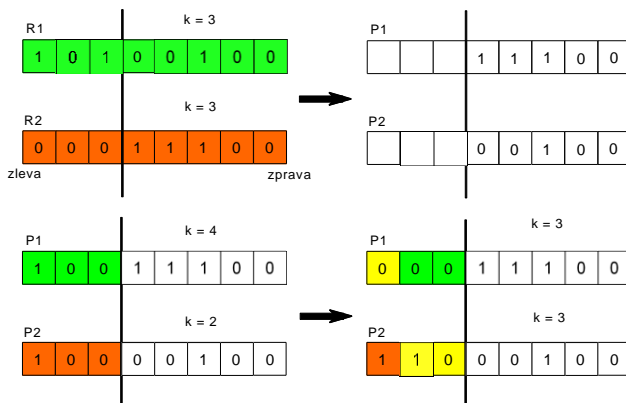
**Křížení jednobodové – Varianta 3 (RL)** – při křížení se náhodně vybere bod křížení. Budeme kopírovat napevno (pro celý průběh algoritmu) pravou stranu (PS) rodičů (R1, R2) do potomků (P1, P2) – (PS R1 do P2 a PS R2 do P1) a doplnění na platného jedince bude probíhat zleva (doplnění P1 probíhá od bodu křížení vlevo rodičem R1 zleva). Doplnění na platného jedince probíhá na jednotlivých pozicích (genech) do doby, kdy jedničky dosáhnou celkového počtu  $k$  jedniček (počtu rozmísťovaných dep ( $k$ )). V opačném případě, kdy jsme doplněním na platného jedince nedosáhli požadovaného počtu jedniček v jedinci, náhodně zvolíme gen/y, kde bude/ou jednička/y doplněna/y, ale pouze v doplňovací části jedince, jak znázorňuje obrázek 4.



*Zdroj: autor*

Obrázek 4 Princip jednobodového křížení – Varianta 3

**Křížení jednobodové – Varianta 4 (RR)** – při křížení se náhodně vybere bod křížení. Budeme kopírovat napevno (pro celý průběh algoritmu) pravou stranu (PS) rodičů (R1, R2) do potomků (P1, P2) – (PS R1 do P2 a PS R2 do P1) a doplnění na platného jedince bude probíhat zprava (doplnění P1 probíhá od bodu křížení vlevo rodičem R1 zprava). Doplnění na platného jedince probíhá na jednotlivých pozicích (genech) do doby, kdy jedničky dosáhnou celkového počtu  $k$  jedniček (počtu rozmísťovaných dep ( $k$ )). V opačném případě, kdy jsme doplněním na platného jedince nedosáhli požadovaného počtu jedniček v jedinci, náhodně zvolíme gen/y, kde bude/ou jednička/y doplněna/y, ale pouze v doplňovací části jedince, jak znázorňuje obrázek 5.

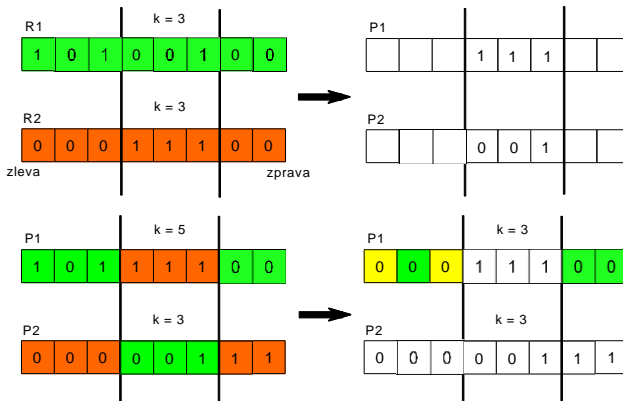


Zdroj: autor

Obrázek 5 Princip jednobodového křížení – Varianta 4

**Křížení jednobodové – Varianta 5 (N)** – při křížení se náhodně vybere bod křížení. Budeme kopírovat náhodně (pro celý průběh algoritmu) levou stranu (LS) nebo pravou stranu (PS) rodičů (R1, R2) do potomků (P1, P2) – (LS/PS R1 do P2 a LS/PS R2 do P1) a doplnění na platného jedince bude probíhat náhodně zleva/zprava (doplnění P1 probíhá od bodu křížení vpravo/vlevo rodičem R1 zleva/zprava). Doplnění na platného jedince probíhá na jednotlivých pozicích (genech) do doby, kdy jedničky dosáhnou celkového počtu  $k$  jedniček (počtu rozmíst'ovaných dep ( $k$ )). V opačném případě, kdy jsme doplněním na platného jedince nedosáhli požadovaného počtu jedniček v jedinci, náhodně zvolíme gen/y, kde bude/ou jednička/y doplněna/y, ale pouze v doplňovací části jedince.

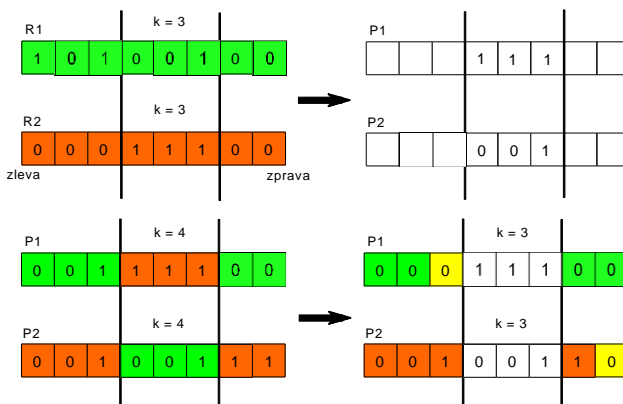
**Křížení dvoubodové – Varianta 6 (L)** – při křížení se náhodně vyberou dvě pozice, mezi kterými se jedinci zkříží, tj. všechny geny, které se nachází mezi předmětnými pozicemi, si potomci těchto rodičů prohodí. Doplnění na platného jedince bude probíhat zleva (doplnění P1 probíhá zleva od začátku jedince rodičem R1 zleva). Doplnění na platného jedince probíhá na jednotlivých pozicích (genech) do doby, kdy jedničky dosáhnou celkového počtu  $k$  jedniček (počtu rozmíst'ovaných dep ( $k$ )). V opačném případě, kdy jsme doplněním na platného jedince nedosáhli požadovaného počtu jedniček v jedinci, náhodně zvolíme gen/y, kde bude/ou jednička/y doplněna/y, ale pouze v doplňovací části jedince, jak znázorňuje obrázek 6.



Zdroj: autor

Obrázek 6 Princip dvoubodového křížení – Varianta 6

**Křížení dvoubodové – Varianta 7 (R)** – při křížení se náhodně vyberou dvě pozice, mezi kterými se jedinci zkříží, tj. všechny geny, které se nachází mezi předmětnými pozicemi, si potomci těchto rodičů prohodí. Doplnění na platného jedince bude probíhat zprava (doplnění P1 probíhá zleva od začátku jedince rodičem R1 zprava). Doplnění na platného jedince probíhá na jednotlivých pozicích (genech) do doby, kdy jedničky dosáhnou celkového počtu  $k$  jedniček (počtu rozmísťovaných dep ( $k$ )). V opačném případě, kdy jsme doplněním na platného jedince nedosáhli požadovaného počtu jedniček v jedinci, náhodně zvolíme gen/y, kde bude/ou jednička/y doplněna/y, ale pouze v doplňovací části jedince, jak znázorňuje obrázek 7.



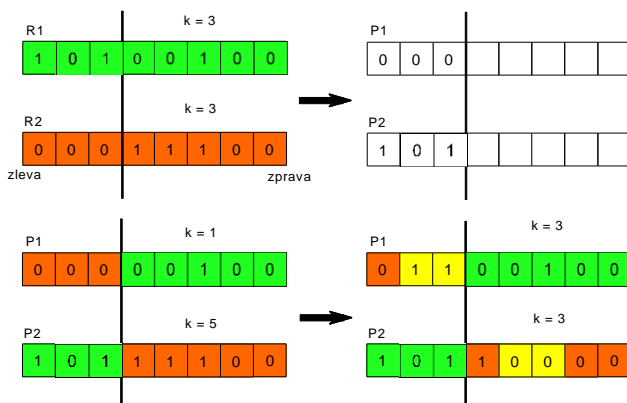
Zdroj: autor

Obrázek 7 Princip dvoubodového křížení – Varianta 7

**Křížení dvoubodové – Varianta 8 (N)** – při křížení se náhodně vyberou dvě pozice, mezi kterými se jedinci zkříží, tj. všechny geny, které se nachází mezi předmětnými pozicemi, si potomci těchto rodičů prohodí. Doplnění na platného jedince bude probíhat náhodně zleva/zprava (doplnění P1 probíhá zleva od začátku jedince rodičem R1 zleva/zprava). Doplnění na platného jedince probíhá na jednotlivých pozicích (genech) do doby, kdy jedničky dosáhnou celkového počtu  $k$  jedniček (počtu rozmísťovaných dep ( $k$ )). V opačném případě, kdy jsme doplněním na platného jedince nedosáhli požadovaného počtu jedniček v jedinci, náhodně zvolíme gen/y, kde bude/ou jednička/y doplněna/y, ale pouze v doplňovací části jedince.

Další možné varianty křížení, které nebudou dále testovány:

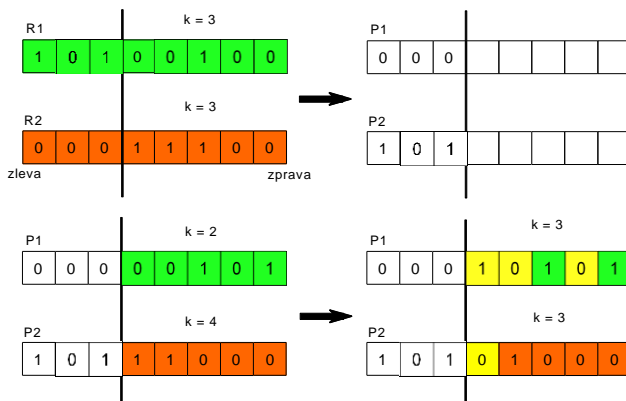
**Varianta 9 – 18** – jednobodové křížení, náhodně je vygenerován bod křížení. Tvorba potomka - část prvního rodiče nalevo od bodu křížení je beze změny nakopírována do potomka. Z druhého rodiče je pak postupně zprava od bodu křížení doplněn jedinec a tím i jedničky do celkového počtu  $k$  jedniček. Pokud celkový počet jedniček neodpovídá  $k$ , pokračujeme doplňováním jedniček zprava u druhého rodiče nalevo od bodu křížení, jak znázorňuje obrázek 8. Další modifikace vzniknou: kopírováním LS nebo PS nebo náhodně části genetické informace jednoho rodiče. Doplnění na platného jedince probíhá z druhé části druhého rodiče zprava nebo zleva nebo náhodně od bodu křížení vpravo/vlevo nebo vpravo/vlevo od nejbližšího genu k bodu křížení. Vznikne celkem 10 různých variant řešení.



Zdroj: autor

Obrázek 8 Varianta 9 - 18

**Varianta 19 – 24** - jednobodové křížení, náhodně je vygenerován bod křížení. Jde o modifikaci Varianty 1 až 5, kdy doplnění na platného jedince bude probíhat zleva nebo zprava nebo náhodně (doplnění P1 probíhá od nejvzdálenějšího genu k bodu křížení vpravo/vlevo rodičem R1 zleva/zprava). Vznikne celkem 6 různých variant řešení. Obrázek 9 znázorňuje modifikaci Varianty 1 (LL).



*Zdroj: autor*

Obrázek 9 Varianta 19 - 24

## 2.4 Genetický operátor mutace

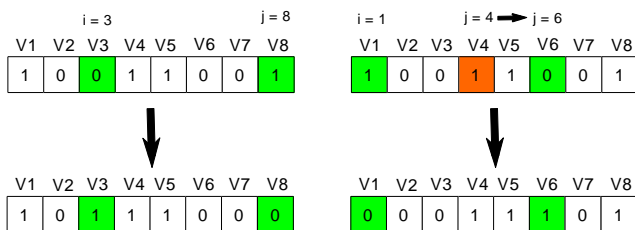
Dalším krokem v cyklu algoritmu je mutace, ke které přistupuje každý jedinec z množiny jednotlivě. Na základě hodnoty pravděpodobnosti mutace (označované  $P_M$ ) daný jedinec zmutuje. Následně byly zvoleny tyto dva typy mutace:

- **Prohození jednoho genu** – v jedinci se navzájem prohodí dva geny,

Z podmnožiny je náhodně vybrán uzel, který je nahrazen uzlem, který se v podmnožině nevyskytuje, a který je také vybrán náhodně. Prakticky je mutace realizována následovně: náhodně (s rovnoměrným rozložením) se vygeneruje index  $i$  do pole. Tím se určí alela (poloha genu), která bude měněna. Je-li gen (bit) s indexem  $i$  nastaven (uzel je v podmnožině), vynuluje se (uzel se vyjme z podmnožiny), je-li gen (bit) s indexem  $i$  nulový (uzel není v podmnožině), nastaví se (uzel se vloží do podmnožiny). Pak se vygeneruje náhodně druhý index  $j$ , určí se alela druhého genu. Pokud má bit s indexem  $j$  opačnou hodnotu, než měl původně bit s indexem  $i$ , invertuje se. Tím je záměna dvou uzlů dokončena. Pokud má bit s indexem  $j$  stejnou



hodnotu jako měl  $i$ -tý bit, hledá se lineárně (cyklicky) směrem k vyšším indexům bit s opačnou hodnotou než měl bit  $i$ , a ten se invertuje. Protože je  $l < k < n$ , předmětný bit se vždy najde. Princip mutace znázorňuje obrázek 10.



*Zdroj: autor*

Obrázek 10 Princip navržené mutace

- **Prohození dvou genů** - jedná se o stejný postup jako při prohození jednoho genu, ale daná operace se provede dvakrát za sebou.

## 2.5 Vyvinutý software pro řešení vybrané lokační úlohy

Navržené variantní řešení aplikace genetických algoritmů na vybranou lokační úlohu bylo nutné testovat pro zjištění kvality řešení. Byla vyvinuta aplikace s grafickým rozhraním pro prostředí MS Windows. Aplikace je nazvána „Location task“, je naprogramována v jazyce C# a umožňuje uživateli plně využít výše navržené řešení výpočtu lokační úlohy pomocí genetického algoritmu. Součástí aplikace je i výpočet exaktní metodou hrubé síly.

## 3. TESTOVÁNÍ A POSOUZENÍ KVALITY VÝSLEDKŮ

Vybraná lokační úloha byla řešena dvěma metodami, exaktně metodou hrubé síly a výše navrženým genetickým algoritmem s variantní volbou genetických operátorů a parametrů. Všechny navržené varianty a parametry byly testovány v softwarové aplikaci „Location task“ na menším datovém souboru (72 vrcholů v grafu) a pouze varianty a parametry nejlepších nalezených řešení byly testovány na větším datovém souboru (394 vrcholů v grafu). Exaktně byly řešeny všechny varianty u menšího souboru dat, protože u většího souboru dat je možné v reálném čase testovat jen vybrané kombinace vrcholů a umístěvaných dep.

**Menší soubor dat** – jedná se o okresní města České republiky. Počet vrcholů v tomto souboru je 72. Váhy vrcholů jsou stanoveny poměrově podle počtu obyvatel. Bylo vypočítáno optimální řešení exaktní metodou vybrané úlohy pro daný soubor dat s počtem umístěvaných dep  $k$  ( $k = 2$ ,  $k = 3$ ). Byla vypočítána optimální hodnota dopravní práce, umístění dep do vrcholů, atrakční obvody daných dep, doba výpočtu a grafické znázornění optimálního řešení. Navrženým genetickým algoritmem vzniklo 3072 kombinací parametrů (variantní počet umístěvaných dep, počet iterací, počet jedinců v generaci, elitářství ano/ne, 3 varianty selekce, 8 variant křížení, 2 varianty mutace, variantní pravděpodobnosti křížení i mutace). Byla nalezena nejlepší hodnota dopravní práce při konkrétní iteraci, umístění dep do vrcholů, atrakční obvody daných dep, doba výpočtu a grafické znázornění průběhu iterací. Dvě nejlepší nalezená řešení z přípustných 3072 řešení pro umístění dvou a tří dep na síti byla graficky znázorněna a dále testována na větším vzorku dat.

**Větší soubor dat** – v souboru je 394 vybraných obcí (definiční body obcí s pověřeným obecním úřadem). Bylo vypočítáno optimální řešení exaktní metodou vybrané úlohy pro daný soubor dat s počtem umístěvaných dep  $k$  ( $k = 2$ ,  $k = 3$ ). Pro větší hodnotu  $k$  nelze aplikovat exaktní výpočet z důvodu časové náročnosti. Byla vypočítána optimální hodnota dopravní práce, umístění dep do vrcholů, atrakční obvody daných dep, doba výpočtu a grafické znázornění optimálního řešení. Navrženým genetickým algoritmem vzniklo 20 kombinací parametrů (variantní počet umístěvaných dep  $k = 2$ ,  $k = 3$ ,  $k = 4$ ,  $k = 5$ ,  $k = 6$  a počet jedinců v generaci dvou nejlepších nalezených řešení z předchozího testování na menším souboru dat). Byla nalezena nejlepší hodnota dopravní práce při konkrétní iteraci, umístění dep do vrcholů, atrakční obvody daných dep, doba výpočtu a grafické znázornění průběhu iterací.

### 3.1 Porovnání výsledků použitých metod

#### Menší vzorek dat

Celkem bylo testováno 3072 kombinací parametrů, z nich 1536 kombinací parametrů pro  $k = 2$ , a to 768 kombinací parametrů při počtu 1 000 iterací a 768 při počtu 200 iterací a 1536 kombinací parametrů pro  $k = 3$ , a to 768 kombinací parametrů při počtu 1 000 iterací a 768 při počtu 200 iterací. Obě nejlepší nalezená řešení pro  $k = 2$  (testování 1) a  $k = 3$  (testování 2) jsou dále testována na větším vzorku dat.

## **Porovnání výsledků pro dvě depa**

Pro vzorek dat obsahující 72 vrcholů, genetický algoritmus při 1 000 iteracích nedosáhl vždy optimálního řešení. Bylo testováno celkem 768 kombinací parametrů, přičemž 81 kombinací (10,5%) nebylo rovno optimu. Optimální řešení má hodnotu dopravní práce 57 280 km. Odchylky řešení od optimální hodnoty dopravní práce byly v intervalu (57 280, 71 564), což vzhledem ke kritériální funkci je zanedbatelné a nalezené řešení se blíží k optimálnímu řešení. Při počtu 200 iterací bylo testováno celkem 768 kombinací, přičemž 301 kombinací (39%) nebylo rovno optimu. Z celkového počtu bylo 384 kombinací testováno bez elitismu (odchylka 173 kombinací (45%) a hodnota dopravní práce byla v intervalu (57 280, 67 466)), a s elitismem 1. stupně (odchylka 131 kombinací (34%)), kdy hodnota dopravní práce byla v intervalu (57 280, 71 564)). Nejčteněji se objevovalo 2. a 3. nejlepší řešení s hodnotou dopravní práce 57 404 (resp. 57 408), které je velmi blízké optimálnímu řešení. Z testovaných 1536 kombinací pro umístění dvou dep do sítě vzešlo 96 nejlepších řešení pro každou kombinaci parametrů. Průměrný čas na výpočet u nejlepších řešení se pohyboval okolo 1,27 sekund, což je oproti času potřebnému k exaktnímu výpočtu (2 sekundy) zanedbatelné. Větší rozdíly ve výpočetních časech se objeví u rozsáhlejších úloh. V případě shodnosti výsledků, je za nejlepší nalezené řešení považováno řešení, které v první hierarchii má nejnižší hodnotu dopravní práce, v druhé hierarchii má lepší výpočetní čas a ve třetí hierarchii konkrétní, co nejnižší iteraci. Z uvedených výsledků se dá odvodit, které operátory a parametry mají pozitivní či negativní vliv na výsledek výpočtu. Ze zkoumání výsledků vychází, že nejvíce výsledek ovlivňuje typ zvoleného křížení včetně pravděpodobnosti křížení a stupeň elitářství.

## **Porovnání výsledků pro tři depa**

Pro vzorek dat o velikosti 72 vrcholů, genetický algoritmus při 1 000 iteracích nedosáhl vždy optimálního řešení. Bylo testováno celkem 768 kombinací parametrů, přičemž 284 kombinací (37%) nebylo rovno optimu. Optimální řešení má hodnotu dopravní práce 45 052 km. Odchylky řešení od optimální hodnoty dopravní práce byly v intervalu (45 052, 54 076), což vzhledem ke kritériální funkci je zanedbatelné a nalezené řešení se blíží k optimálnímu řešení. Při počtu 200 iterací bylo testováno celkem 768 kombinací, přičemž 545 kombinací (71%) nebylo rovno optimu. Z celkového počtu bylo 384 kombinací testováno bez elitismu (odchylka 300 kombinací (78%), hodnota dopravní práce byla v intervalu (45 052,

54 650)) a s elitismem 1. stupně (odchylka 245 kombinací (64%), kdy hodnota dopravní práce byla v intervalu (45 052, 56 404)). Odchylky řešení od optimální hodnoty dopravní práce se pohybovaly v intervalu (45 052, 56 404), což vzhledem ke kritériální funkci je zanedbatelné a nalezené řešení se blíží k optimálnímu řešení. Z testovaných 1536 kombinací pro umístění tří dep na síti vzešlo 96 nejlepších řešení pro každou kombinaci parametrů. Průměrný čas na výpočet u nejlepších řešení se pohyboval okolo 4,01 sekund, což je oproti času potřebnému k exaktnímu výpočtu (10 sekund) zanedbatelné. Z uvedených výsledků se dá odvodit, které operátory a parametry mají pozitivní či negativní vliv na výsledek výpočtu. Ze zkoumání výsledků vychází, že nejvíce výsledek ovlivňuje typ zvoleného křížení, mutace a stupeň elitářství.

### Větší vzorek dat

Celkem bylo testováno 20 kombinací parametrů, přičemž 10 kombinací parametrů pro testování 1 a 10 kombinací pro testování 2 při počtu iterací vždy 10 000, počtu jedinců v generaci 25 a 50 s počtem umisťovaných dep  $k = 2$ ,  $k = 3$ ,  $k = 4$ ,  $k = 5$ ,  $k = 6$ . Z testovaných 20 kombinací vzešlo 10 nejlepších. Pro umístění dvou a tří dep máme k dispozici i exaktní výpočet. Výsledky aplikace genetických algoritmů byly ve všech čtyřech případech ve shodě s exaktním řešením. Pro umístění čtyř, pěti a šesti dep na síti nemáme výsledek exaktní metody k dispozici z důvodu velkých časových nároků výpočtu, jak ukazuje Tabulka 1.

Tabulka 1 Časová náročnost většího vzorku exaktní metodou v závislosti na počtu vrcholů a dep

Počet vrcholů v grafu	Počet dep	Počet kombinací	Časová náročnost [dny:h:m:s]
394	2	77 421	00:00:01:25
394	3	10 116 344	00:03:01:40
394	4	988 872 626	13:02:44:38
394	5	$7,713 \cdot 10^{10}$	2 roky 362 dní
394	6	$5,001 \cdot 10^{12}$	205 let

*Zdroj: autor*

Nicméně z výsledků aplikace genetických algoritmů a z grafického znázornění vyplývá potvrzení úvahy, že se při testování podařilo nalézt výsledek optimální nebo blízký se optimálnímu řešení úlohy.

Průměrný čas na výpočet všech 20 kombinací se pohyboval kolem 28 minut (pro  $k = 2$  – 20,54 minut, pro  $k = 3$  – 28,35 minut, pro  $k = 4$  – 28,41 minut, pro  $k = 5$  – 32,31 minut, pro  $k = 6$  – 36,25 minut), což je oproti času potřebnému k exaktnímu výpočtu (pro  $k = 2$  – 1,25 minut, pro  $k = 3$  – 3,01 hodin, pro  $k = 4$  – 13,02 dnů, pro  $k = 5$  – 2 roky 362 dní, pro  $k = 6$  – 205 let) velmi významná úspora času. U rozsáhlých úloh diskrétní lokace se jedná o možnost, jak dosáhnout suboptimálního řešení v rozumném čase.

#### 4. ZÁVĚR

Cílem bylo řešení problému umístění středisek obsluhy (dep) na síti. Aplikovat genetické algoritmy na vybranou lokační úlohu, navrhnout kódování problému, funkční genetické operátory a další parametry pro řešení lokační úlohy a následné otestování kvality těchto operátorů a parametrů. Kódování a genetické operátory bylo nutné přizpůsobit povaze zvolené úlohy. Genetická reprezentace má zásadní význam na úspěch genetického algoritmu. Před samotným návrhem algoritmu a jeho operátorů bylo zapotřebí zvolit správné kódování jedinců. Bylo zvoleno unikátní kódování pomocí zobrazení charakteristické funkce množiny, které ideálně vyhovuje danému požadavku. V jedinci jsou v jednotlivých genech nuly a jedničky, přičemž jedničky a jejich počet reprezentují počet umístěvaných dep na síti. Kromě umístění genetický algoritmus na základě minimalizace kritériální funkce vygeneruje i atrakční obvody jednotlivých dep, což je důležité pro následné grafické zobrazení úlohy.

Operátorem selekce byly zvoleny tři varianty pro následné testování. Turnajová varianta (z důvodu její rychlé implementace), a to ve dvou modifikacích – binární a terciální a vážená ruleta. U vážené rulety bylo nutné transformovat kritériální funkci, protože při výpočtu kritériální funkce byli lepší jedinci ohodnoceni menší hodnotou. Avšak v základní verzi byla selekce váženou ruletou koncipována na maximalizační kritérium. Při selekci nedochází ke kontrole shodných jedinců. A proto mohou postoupit do dalších kroků cyklu (iterace) algoritmu dva shodní jedinci. Výstupem selekce byla množina dvou jedinců, kteří dále postoupili ke křížení. Operátorem křížení bylo zvoleno osm způsobů křížení - jednobodové a dvoubodové, vždy ve čtyřech variantách. Celkem bylo využito osm operátorů křížení a dvě pravděpodobnosti křížení pro následné testování.

Operátorem mutace bylo zvoleno prohození genu, a to ve dvou variantách – prohození jednoho genu a dvou genů. Posledním krokem při iteračním cyklu genetického algoritmu byla kontrola jedinců před zařazením do nové generace, byla provedena průběžně v každé iteraci a při každé operaci křížení a mutace tak, aby dva jedinci nebyli shodní v již přiřazených do nové generace. Genetické operátory byly navrženy tak, aby respektovaly platnost jedinců (stejný počet dep - počet jedniček v jedinci).

Po navržení algoritmu a jeho operátorů byla vyvinuta aplikace pro testování. Aplikace byla vytvořena v grafickém rozhraní pro prostředí systému MS Windows. V této aplikaci proběhla všechna testování uvedená v předložené práci. Testování operátorů bylo prováděno na dvou vzorcích dat – malém (72 vrcholů – okresní města České republiky) a velkém (394 vrcholů – definiční body obcí s pověřeným obecním úřadem České republiky). Menší vzorek dat byl využit pro testování, kdy bylo možno výsledky testů porovnat s optimálním řešením úlohy. Celkem bylo testováno 3072 kombinací parametrů. Dvě nejlepší nalezená řešení pro umístění dvou a tří dep na síti byla dále testována na větším vzorku dat. Bylo nutné zvýšit počet iterací na 10 000, a to z důvodu velikosti množiny všech přípustných řešení a variantně počet jedinců v generaci odpovídající rozměru úlohy. Srovnání výsledků exaktní metody a metody používající definovaný metaheuristický přístup založený na genetickém algoritmu ukázalo dobrou shodu výsledků obou metod u malého souboru dat. Z uvedených výsledků bylo odvozeno, které operátory a parametry mají pozitivní či negativní vliv na výsledek výpočtu. Ze zkoumání výsledků je zřejmé, že nejvíce výsledek ovlivňuje typ zvoleného křížení včetně pravděpodobnosti křížení, mutace a stupeň elitářství. Nejvíce významným faktorem v daném případě byl stupeň elitářství. Pokud byl nastaven na elitářství 1. stupně a výše, došlo v algoritmu okamžitě k lepšímu výsledku za kratší počet iterací, protože díky elitářství jedinci nedegenerují, tj. alespoň jeden jedinec v generaci byl stále nejlepší. Výpočetní čas malého vzorku dat aplikace genetických algoritmů byl téměř shodný s výpočetním časem exaktní metody. Uvedená skutečnost však platila pouze v případě méně rozsáhlých diskrétních lokačních úloh. Z tohoto důvodu lze vyslovit předpoklad, že u velkých souborů dat, kde exaktní metoda byla obtížně aplikovatelná, ji nahradila docela dobře navržená metoda, která uzpůsobila metaheuristiku založenou na genetickém algoritmu. Diference výpočetních časů aplikace genetických algoritmů a exaktního řešení byly velmi výrazné. Proto byl při testování zvolen trochu jiný způsob ověření předpokladu kvality navržených operátorů a dalších parametrů, a to grafické zobrazení řešení na mapě ČR, dep a atrakčních obvodů. Nesmíme opomenout, že

v genetických algoritmech hraje velkou roli náhoda. Zároveň lze konstatovat, že výsledky testování velkého vzorku dat se od nejnižší nalezené hodnoty dopravní práce nijak významně nelišily. Z tohoto důvodu lze usuzovat, že výsledky se při testování přiblížily k optimálnímu řešení úlohy pro daný vzorek dat.

## SEZNAM POUŽITÉ LITERATURY

- [1] MOCKOVÁ, D. Alokačně-lokační úlohy a jejich řešení. In *LOGI 2007*. Pardubice: Institut Jana Pernera, 2007, s. 117 - 124. ISBN 80-86530-35-3.
- [2] VOLEK, J., LINDA B. *Teorie grafů - aplikace v dopravě a veřejné správě*. Pardubice: Univerzita Pardubice, 2012. ISBN 978-80-739-5225-9.
- [3] MOCKOVÁ, D. *Základy teorie dopravy - úlohy*. Praha: Vydavatelství ČVUT, 2007. ISBN 978-80-01-0379-1.
- [4] MOCKOVÁ, D. Allocation and location of transport logistics centres. *Acta Polytechnica, Czech Technical University in Prague*. 2010, vol. 50, no 1, pp. 30-34. ISSN 1210-2709.
- [5] MOCKOVÁ, D. Proposal of genetic operators for solving transportation problems. In *11th European Transport Congress*. Praha: Czech Technical University in Prague, 2013, s. 169-177, ISBN 978-80-01-05321-8.
- [6] MOCKOVÁ, D. Návrh genetických operátorů úloh diskretní optimalizace. In *Logistika-Ekonomika-Prax 2012 - recenzovaný zborník z mezinárodnej vedeckej konferencie*. Žilina: Výtvarná agentúra A1, Slovensko, 2012, s. 212-219. ISSN 1336-5878.
- [7] MOCKOVÁ, D. Heuristiky v lokačních úlohách. In *Úlohy diskretní optimalizace v dopravní praxi*. Pardubice: Univerzita Pardubice, 2008, s. 50-54. ISBN 978-80-7395-076-7.
- [8] MOCKOVÁ, D. Genetické algoritmy a lokační úlohy. In *Dopravní systémy 05*. Pardubice: Univerzita Pardubice, Dopravní fakulta Jana Pernera, 2005, s. 300 - 306. ISBN 80-7194-805-5.
- [9] MOCKOVÁ, D., FÁBERA, V. Aplikace genetických algoritmů v lokačně-alokačních úlohách, In *Věda o dopravě 2004*, Praha: Vydavatelství ČVUT, 2004, s. 241-245. ISBN 80-01-03047-4.
- [10] MAŘÍK, V., ŠTĚPÁNKOVÁ, O., LAŽANSKÝ, J. a kol. *Umělá inteligence (4)*, Praha: Academia, 475 s., 2001. ISBN 80-200-1044-0.
- [11] MAŘÍK, V., ŠTĚPÁNKOVÁ, O., LAŽANSKÝ, J. a kol. *Umělá inteligence (3)*, Praha: Academia, 328 s., 2001. ISBN 80-200-0472-6.



## **Ing. DENISA MOCKOVÁ, Ph.D.**

### **Vzdělání:**

- 2005 Ph.D. (doktor); ČVUT v Praze Fakulta dopravní, Praha (*obor „Technologie a management v dopravě a telekomunikacích“*)
- 1998 Ing. (dopravní inženýr); ČVUT V Praze Fakulta dopravní, Praha (*obor „Management a ekonomika dopravy a telekomunikací“*)

### **Zaměstnání:**

- 2010 → zástupce proděkana pro pedagogickou činnost, ČVUT v Praze Fakulta dopravní, Praha
- 2010 → manažer pro pedagogickou činnost ústavu, ČVUT v Praze Fakulta dopravní, Praha
- 2000 - 2006 tajemník katedry, odborný asistent, ČVUT v Praze Fakulta dopravní, Praha

### **Vědecká činnost:**

- odborná spolupráce na grantu FOCUS Proposal no. 261633, topic SEC-2010.6.3-2 „*Foresight Security Scenarios: Mapping Research to a Comprehensive Approach to Exogenous EU Roles*“, ČVUT v Praze Fakulta dopravní, Praha, 2011 - 2013
- odborná spolupráce na výzkumném záměru MSM 6840770043 „*Rozvoj metod návrhu a provozu dopravních sítí z hlediska jejich optimalizace*“, ČVUT v Praze Fakulta dopravní, Praha, 2007 - 2013
- odborná spolupráce na výzkumném záměru MSM 212600023 „*Rozvoj metod systémové analýzy, algoritmů a statistických metod pro dopravu a spoje*“, ČVUT v Praze Fakulta dopravní, Praha, 2002 – 2005
- odborná spolupráce na výzkumném záměru MSM 210000024 „*Automatické systémy v dopravě, diagnostika dopravních systémů a procesů*“, ČVUT v Praze Fakulta dopravní, Praha, 2003
- odborná spolupráce na výzkumném záměru MSM 212600025 „*Modely dopravy a řízení dopravních procesů v území*“, ČVUT v Praze Fakulta dopravní, Praha, 2003 – 2004

### **Pedagogická činnost:**

- přednášky z předmětů Kvantitativní metody v dopravě, Optimalizace na dopravních sítích, Teorie grafů a její aplikace v dopravě, Teorie dopravy, Základy teorie dopravy, ČVUT v Praze Fakulta dopravní

- cvičení z předmětů Kvantitativní metody v dopravě, Teorie grafů a její aplikace v dopravě, Základy teorie dopravy, ČVUT v Praze Fakulta dopravní
- vedení projektu Optimalizační úlohy na logistickém řetězci, ČVUT v Praze Fakulta dopravní
- školitel 5 studentů doktorského studia
- vedoucí bakalářských a diplomových prací (5 oceněných pochvalou děkana)

### **Jiné aktivity:**

- předsedkyně Akademického senátu ČVUT v Praze Fakulty dopravní, Praha; 2011 – 2013
- člen Rady celoživotního vzdělávání na ČVUT, Praha, od 2011
- člen komise pro výběr projektů v rámci FRVŠ a RP MŠMT později IRP, ČVUT v Praze Fakulty dopravní, Praha, od 2010
- člen komise pro státní závěrečné zkoušky bakalářského studia oboru „Management a ekonomika dopravy a telekomunikací“, ČVUT v Praze Fakulty dopravní, Praha, Děčín
- člen komise pro státní závěrečné zkoušky magisterského studia oboru „Management a ekonomika dopravy a telekomunikací“, ČVUT v Praze Fakulty dopravní, Praha
- místopředsedkyně komise pro státní závěrečné zkoušky magisterského studia oboru „Logistika, technologie a management dopravy“, ČVUT v Praze Fakulty dopravní, Praha
- člen komise/hlavní přijímací komise pro přijímací řízení ČVUT v Praze Fakulty dopravní, Praha, od 2002
- tajemník komise pro státní závěrečné zkoušky bakalářského a magisterského studia v oboru „Management a ekonomika dopravy a telekomunikací“, ČVUT v Praze Fakulty dopravní, Praha, od 2000 – 2005
- editor a organizační garant mezinárodní konference „Věda o dopravě“, Praha, 2004

### **Publikační činnost:**

- 1 článek v impaktovaném časopise
- 3 články v mezinárodním recenzovaném časopise
- 15 příspěvků ve sborníku na mezinárodních konferencích

- 10 původních článků v českém vědeckém a odborném časopise
- 1 původní příspěvek na české konferenci
- 7 citací ČR

Publikace (výběr, celkem 27)

- MOCKOVÁ, D., RYBIČKOVÁ, A. Application of Genetic Algorithms to Vehicle Routing Problem. *Neural Network World*. 2014, vol. 24, no. 1, p. 57-78. ISSN 1210-0552. DOI: 10.14311/NNW.2014.24.003.
- MOCKOVÁ, D. Methods for solving discrete optimisation problems. *Transactions on Transport Sciences. Ministry of Transport*. 2012, vol. 5, no. 2, pp. 71-82. ISSN 1802-971X. DOI: 10.2478/v10158-012-0009-0.
- MOCKOVÁ, D. Allocation and location of transport logistics centres. *Acta Polytechnica*, Czech Technical University in Prague. 2010, vol. 50, no 1, pp. 30-34. ISSN 1210-2709.
- MOCKOVÁ, D. Decision support systems for allocation of logistics centres. *Machines, Technologies, Material. Scientific-technical Union of Mechanical Engineering*. 2008, vol. 2, issue 6-7, pp. 25-26. ISSN 1313-0226.
- MOCKOVÁ, D. Proposal of genetic operators for solving transportation problems. In *11th European Transport Congress*. Praha: Czech Technical University in Prague, 2013, s. 169-177, ISBN 978-80-01-05321-8.
- MOCKOVÁ, D., BRODSKÝ, J. Vyvinutý software pro řešení vybrané lokační úlohy genetickými algoritmy. In *Úlohy diskrétní optimalizace v dopravní praxi 2013. SW nástroje na řešení úloh racionalizace. Objektivizace a optimalizace pokrytí území veřejnými obslužnými systémy*. Pardubice: Univerzita Pardubice, 2013, s. 106-113, ISBN 978-80-7395-744-5.
- MOCKOVÁ, D. Návrh genetických operátorů úloh diskrétní optimalizace. In *Logistika-Ekonomika-Prax 2012 - recenzovaný zborník z mezinárodnej vedeckej konferencie*. Žilina: Výtvarná agentúra A1, Slovensko, 2012, s. 212-219. ISSN 1336-5878.
- MOCKOVÁ, D. Matematický model pro lokační úlohu. In *Transport & Logistics*. Slovensko: Štrbské pleso, Vysoké Tatry, 2009. roč. 1, č. 6, s. 168-170. Vydáno jako 6. mimořádné číslo časopisu Transport & Logistics, ISSN 1451-107X.