

TSP – Analýza a optimalizace pomocí GA

Projekt do předmětu Soft Computing

22.10.2010

Autor: Jaroslav Bendl, xbendl00@stud.fit.vutbr.cz

Fakulta informačních technologií

Vysoké učení technické v Brně

Obsah

Obsah.....	2
1. Formulace zadání.....	3
2. Analýza řešeného problému	4
2.1 Reprezentace řešení a způsob výpočtu fitness	4
2.2 Genetické operátory	4
2.2.1 Operátor křížení s uspořádáním	5
2.2.2 Operátor křížení s částečným přiřazením.....	5
2.2.3 Operátor křížení s rekombinací hran.....	6
2.2.4 Cyklický operátor křížení.....	7
2.2.5 Operátor křížení s uspořádáním, varianta 2	7
2.2.6 Konzervativní operátor křížení.....	8
2.2.7 Operátor křížení s hlasováním.....	8
2.2.8 Operátor křížení se střídáním pozice.....	9
2.2.9 Mutace Swap	9
2.2.10 Mutace 2-OPT a 3-OPT.....	10
2.2.11 Nahrazující mutace.....	10
2.2.12 Vkládací mutace	10
2.2.13 Inverzní mutace	11
2.2.14 Mixující mutace.....	11
2.3 Selekce a obnova populace	11
3. Experimenty.....	12
3.1 Srovnání operátorů křížení podle kvality řešení	12
3.2 Srovnání operátorů mutace podle kvality řešení.....	14
3.3 Srovnání operátorů křížení podle počtu evaluací.....	17
3.4 Srovnání operátorů podle časové náročnosti.....	17
3.4 Srovnání operátorů podle časové náročnosti.....	18
4. Závěr.....	19
Reference.....	20
Příloha A: Manuál k programu	21
A.1 Instalace a požadavky programu	21
A.2 Formát konfiguračního souboru	21
A.3 Formát vstupního datového souboru	21
A.4 Ovládání programu.....	22

Kapitola 1

1) Formulace zadání

Cílem projektu je implementace známého problému obchodního cestujícího se zaměřením na experimenty s použitými genetickými operátory. Nedílnou součástí zadání je tedy porovnání a analýza zkoumaných operátorů z hlediska vhodnosti jejich použití pro tuto konkrétní úlohu.

Problém obchodního cestujícího byl poprvé matematicky formulován v roce 1930 a dodnes zůstává jedním z nejznámějších a nejvýznamnějších optimalizačních problému. Úloha spočívá v hledání optimální posloupnosti průchodu všemi zadanými městy tak, aby součet jednotlivých vzdáleností byl minimální. Protože jde o NP-těžký problém [1], nelze obecně nalézt deterministický algoritmus, s nímž by bylo možné dosáhnout řešení v polynomiálně omezeném čase. Úplný algoritmus, jenž by umožňoval nalezení optimálního řešení, totiž musí vyzkoušet všechny možné permutace zadaných měst, kterých je $n!$. Z toho je patrné, že takový postup bude i pro relativně malý počet měst zcela nevhodný. Řešením je proto použití metaheuristických metod, mezi které patří kromě genetických algoritmů také simulované žíhání, metoda tabu search či neuronové sítě. Při použití genetického algoritmu lze kvalitu výsledku významně ovlivnit typem použitých variačních operátorů, jejichž porovnání a analýza je hlavním tématem této práce.

Kapitola 2

Analýza řešeného problému

V této kapitole bude problém analyzován z hlediska použitého způsobu kódování, selekce a algoritmu ohodnocování kandidátních řešení. Hlavní pozornost pak bude věnována použitým genetickým operátorům

2.1 Reprezentace řešení a způsob výpočtu fitness

Řešení problému obchodního cestujícího je permutace navštívených měst, ve které se žádné město neopakuje. Tato omezující podmínka zapříčiňuje rozpad stavového prostoru na oblast s přípustnými a nepřípustnými řešeními a musí být proto použity techniky umožňující se s omezujícími podmínkami vypořádat. Řešení spočívá buď v použití pokutových funkcí, nebo opravných procedur a dekodérů, či použití aplikačně specifické reprezentace problému. Aplikačně specifická reprezentace problému je z těchto principů nejefektivnější, jelikož je založena na speciálním kódování jedinců, které spolu s upravenými variačními operátory nepřipustí vznik nesprávného řešení [5].

Binární reprezentace je pro zakódování řešení nevhodná z celé řady důvodů, mezi které patří zejména nepřírozený charakter binární reprezentace, omezená možnost kombinace vhodných stavebních bloků a s tím související značný destrukční vliv klasických genetických operátorů [1]. Z toho důvodu je mnohem vhodnější zvolit permutační zakódování, u kterého je řešením vektor délky rovné počtu měst. Permutační kódování pro problém obchodního cestujícího pak může využívat takzvanou sousedskou reprezentaci, ordinální reprezentaci či přirozenou reprezentaci cesty, která je pro tuto úlohu vybírána nejčastěji a je rovněž nejefektivnější. Chromozom ve tvaru (3, 5, 1, 4, 2) pak odpovídá průchodu měst v pořadí 3-5-1-4-2.

Účelová funkce je definována součtem vzdáleností dvojic sousedních měst výsledného řešení. Tímto způsobem vypočítanou hodnotu lze s výhodou použít pro ohodnocení konkrétního jedince. Formalizace tohoto problému je tedy následující [5]:

$$P_{opt} = \underset{P \in G_n}{\operatorname{argmin}} f(P) \quad , \text{ kde } P \text{ je permutace a } G_n \text{ je symetrická grupa permutací.}$$

2.2 Genetické operátory

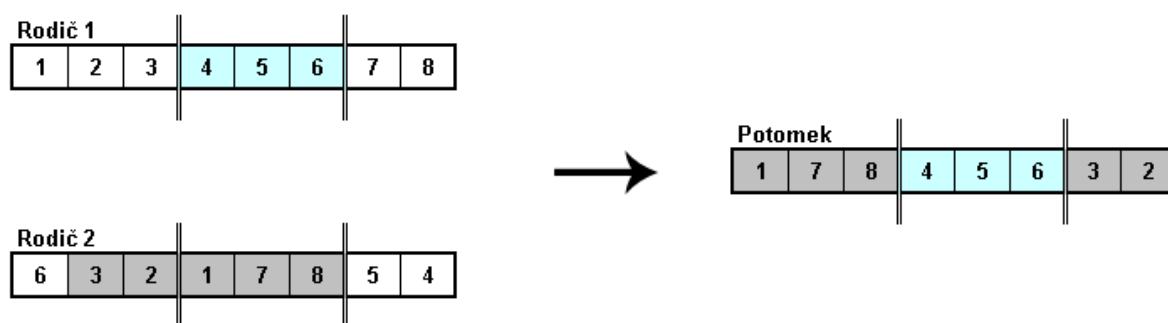
Genetické operátory pro problém obchodního cestujícího musí respektovat omezující podmínku spočívající ve skutečnosti, že žádné město se nemůže ve výsledné permutaci opakovat. V níže uvedených podkapitolách jsou prezentovány algoritmy křížení a mutace, které se s touto podmínkou dokáží vypořádat, a které jsou v rámci této práce studovány.

Křížení je ve výsledném programu prováděno s pevnou pravděpodobností 1,0. Pravděpodobnost mutace lze nastavit v intervalu $<0;1>$, přičemž se jedná o pravděpodobnost provedení jedné mutace na chromozom.

2.2.1 Operátor křížení s uspořádáním

Operátor křížení s uspořádáním (OX – order crossover) je aplikován na dvojici rodičů a vzniká jeden potomek. Princip lze popsat takto [3]:

- 1) Náhodně jsou vygenerovány dva body křížení.
- 2) Geny mezi body křížení prvního rodičovského chromozomu jsou zkopírovány na korespondující pozice v chromozomu potomka.
- 3) Postupně je od pozice druhého bodu křížení procházen chromozom druhého rodiče a pokud se aktuální gen dosud nenachází v chromozomu potomka, je k němu tento gen zkopírován. V opačném případě se tento gen přeskočí.

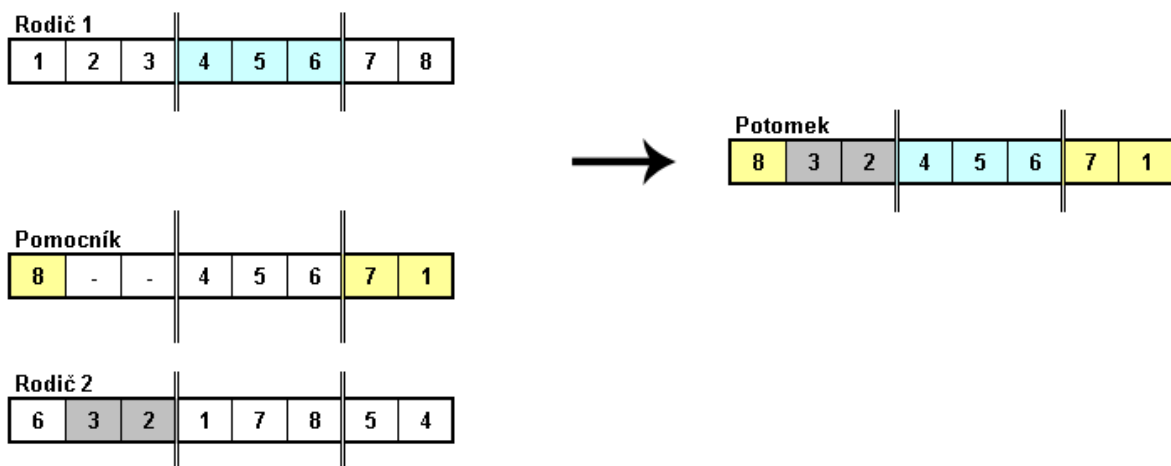


Obrázek č.1: Ilustrace vzniku potomka křížením typu OX.

2.2.2 Operátor křížení s částečným přiřazením

Operátor křížení s částečným uspořádáním (PMX – partial matched crossover) je aplikován na dvojici rodičů a vzniká jeden potomek. Princip lze popsat takto [1]:

- 1) Náhodně jsou vygenerovány dva body křížení.
- 2) Geny mezi body křížení prvního rodičovského chromozomu jsou zkopírovány na korespondující pozice v chromozomu potomka.
- 3) Postupně je procházen úsek prvního rodičovského chromozomu mezi body křížení.
 - a. Aktuální gen je spojen s protilehlým genem druhého rodičovského chromozomu a zároveň se stejnou hodnotou v druhém rodičovském chromozomu.
 - b. Pokud nemá protilehlý gen stejnou hodnotu jako aktuální, jsou oba geny zkopírovány do pomocného chromozomu potomka a jsou prohozeny jejich pozice.
- 4) Pomocný chromozom je mimo úsek vymezený body křížení zkopírován do potomka.
- 5) Zbývající volné pozice v chromozomu potomka jsou obsazeny odpovídajícími pozicemi z druhého rodičovského chromozomu.

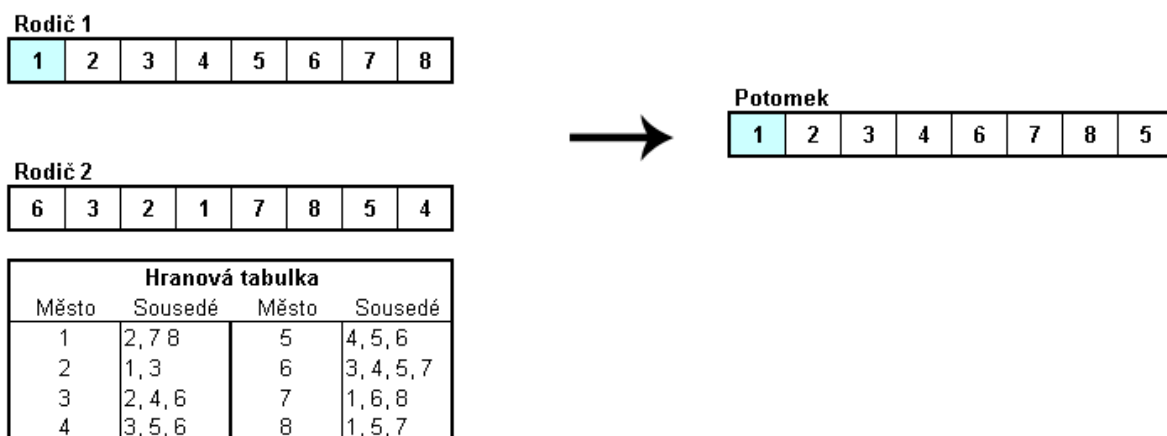


Obrázek č.2: Ilustrace vzniku potomka křížením typu PMX.

2.2.3 Operátor křížení s rekombinací hran

Operátor křížení s rekombinací hran (ERX – edge recombination crossover) je aplikován na dvojici rodičů a vzniká jeden potomek. Princip lze popsat takto [1]:

- 1) Nejprve je sestavena hranová tabulka, která obsahuje ke každému genu výčet jeho sousedů (berou se v potaz sousedé z obou rodičovských chromozomů).
- 2) Náhodně se vybere první gen jednoho z rodičovských chromozomů a ten se považuje za aktuální.
- 3) Dokud délka chromozomu není rovna počtu měst:
 - a. Aktuální gen se přidá do chromozomu potomka a odstraní se výskyt aktuálního genu v seznamu sousedů jednotlivých měst hranové tabulky.
 - b. Pokud nemá aktuální gen žádné sousední geny v hranové tabulce, je jako aktuální vybrán náhodný gen, který se prozatím nenachází v chromozomu potomka. V opačném případě je jako aktuální gen vybrán sousední gen aktuálního chromozomu, který má nejmenší počet vlastních sousedů. Je-li takových genů více, určí se náhodně jeden z nich.

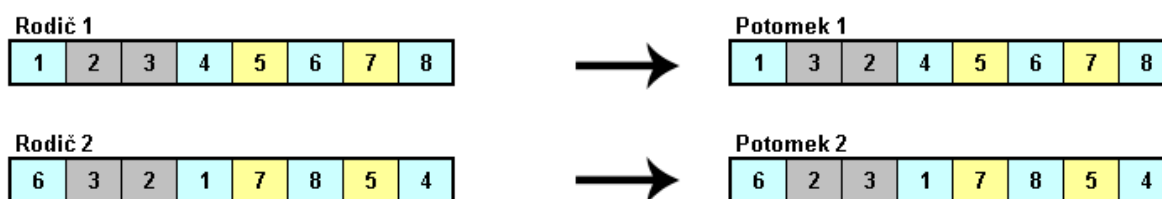


Obrázek č.3: Ilustrace vzniku potomka křížením typu ERX.

2.2.4 Cyklický operátor křížení

Cyklický operátor křížení (CX – cycle crossover) je aplikován na dvojici rodičů a vznikají dva potomci. Princip lze popsat takto [3]:

- 1) Jako aktuální gen se vezme první gen prvního chromozomu.
- 2) Aktuální gen je spojen s protilehlým genem druhého rodičovského chromozomu.
- 3) Na základě hodnoty protilehlého genu druhého rodičovského chromozomu se vybere gen v prvním rodičovském chromozomu.
- 4) Je-li gen vybraný v bodě 3) zároveň aktuálním genem, pak cyklus skončí. Poté se vybere první nezacyklený gen a pokračuje se bodem 2). Pokud žádný nezacyklený gen neexistuje, pokračuje se bodem 5).
- 5) Jako aktuální skupina se vezme skupina určená prvním genem obou chromozomů.
- 6) Všechny geny aktuální skupiny (geny se stejnou barvou) se zkopírují na korespondující pozice potomků bez aplikace křížení.
- 7) Jako aktuální skupina se vezme skupina určená dalším genem, který ještě nebyl zkopírován. Pokud žádný takový gen již neexistuje, algoritmus skončí.
- 8) Všechny geny aktuální skupiny (geny se stejnou barvou) se zkopírují na korespondující pozice potomků při aplikaci křížení.

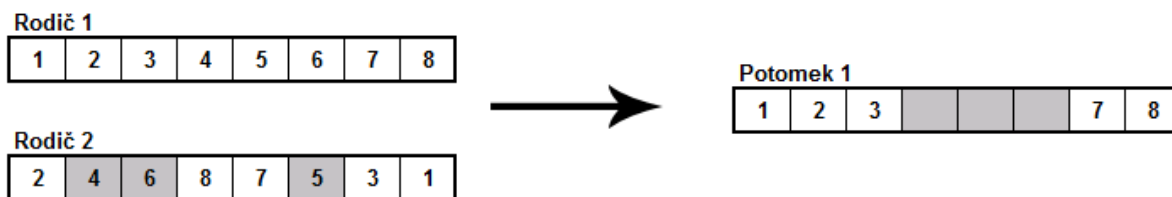


Obrázek č.4: Ilustrace vzniku potomka křížením typu CX.

2.2.5 Operátor křížení s uspořádáním, varianta 2

Operátor křížení s uspořádáním (OX2 – order crossover) je aplikován na dvojici rodičů a vzniká jeden potomek. Princip lze popsat takto [7]:

- 1) Chromozom prvního rodiče je zkopírován do chromozomu potomka.
- 2) Náhodně se určí počet modifikovaných pozic v chromozomu potomka.
- 3) Náhodně se určí konkrétní modifikované pozice a zjistí se konkrétní hodnoty genů na těchto pozicích v chromozomu druhého rodiče.
- 4) Hodnoty genů vybraných v bodě 3) se naleznou v chromozomu potomka a na daných pozicích se nastaví hodnoty genů na *null*.
- 5) Prochází se chromozom potomka a geny s hodnotou *null* jsou nastaveny na první hodnotu genu z chromozomu druhého rodiče (procházeno zleva doprava), která ještě není obsažena v chromozomu potomka.

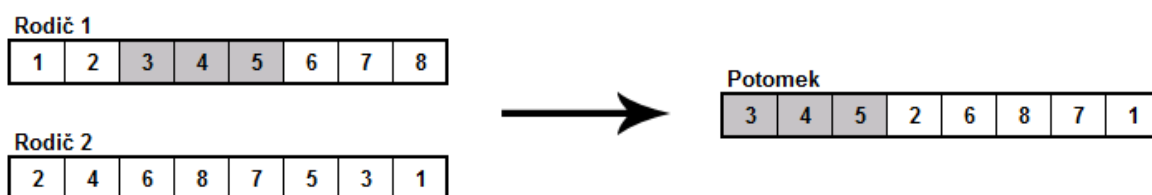


Obrázek č.5: Ilustrace vzniku potomka křížením typu OX2. Náhodně jsou určeny tři pozice v druhém rodiči (druhá, třetí, šestá), na kterých se nacházejí geny s hodnotami 4, 6 a 5. Tyto geny v chromozomu prvního rodiče zabírají čtvrtou, pátou a šestou pozici, proto je na těchto pozicích v potomkovi *null*. Potomek bude nakonec doplněn do tvaru „1 2 3 4 5 6 7 8“ (viz 5.bod v prezentovaném algoritmu).

2.2.6 Konzervativní operátor křížení

Konzervativní operátor křížení (MPX – maximal preservative crossover) je aplikován na dvojici rodičů a vzniká jeden potomek. Princip lze popsat takto [7]:

- 1) V chromozomu prvního rodiče je náhodně vybrána sekvence genů.
- 2) Zvolená sekvence genů je vložena na začátek chromozomu potomka.
- 3) Zbývající geny v chromozomu potomka jsou nastaveny podle genů druhého rodiče (ten se prochází zleva doprava), přičemž jsou vynechávány ty hodnoty, které již chromozom potomka obsahuje.

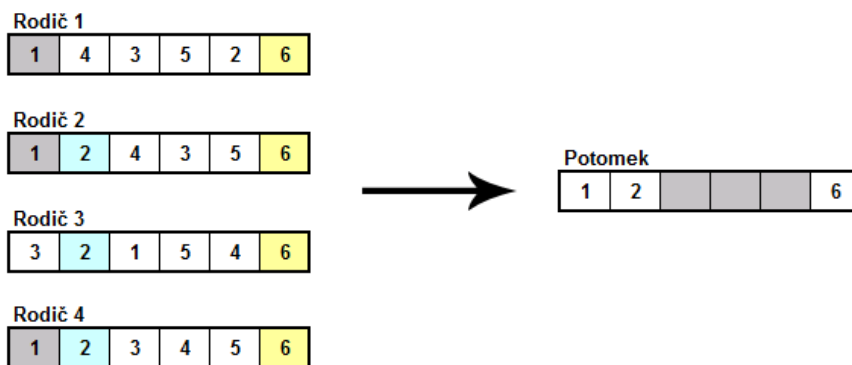


Obrázek č.6: Ilustrace vzniku potomka křížením MPX.

2.2.7 Operátor křížení s hlasováním

Operátor křížení s hlasováním (VR – Voting recombination crossover) je aplikován na množinu rodičů a vzniká jeden potomek. Princip lze popsat takto [7]:

- 1) Náhodně je vybráno k rodičů ($k > 2$), z kterých bude potomek sestaven.
- 2) Náhodně je určen práh *threshhold* (*threshhold* > 0).
- 3) Postupně se prochází přes jednotlivé pozice chromozomů. Pro každou pozici se zjistí hodnoty a jejich četnost v chromozomech rodičů. Má-li hodnota s nejvyšší četností tuto četnost větší než stanovený práh *threshhold*, pak se na danou pozici chromozomu potomka vloží gen s touto hodnotou.
- 4) Nevyplněné pozice v chromozomu potomka jsou doplněny náhodně generovanými hodnotami.

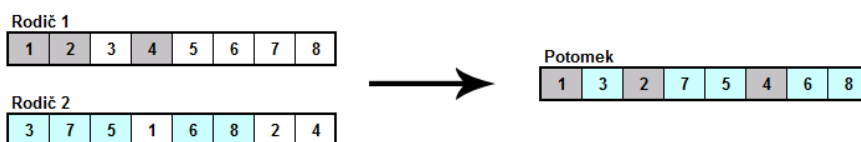


Obrázek č.7: Ilustrace vzniku potomka křížením VR pro počet rodičů $k=4$ a práh $threshold=3$. Zbylé pozice (v potomkovi podbarveny šedou barvou) budou určeny náhodně.

2.2.8 Operátor křížení se střídáním pozice

Operátor křížení se střídáním pozice (AP – alternativ-position crossover) je aplikován na dvojici rodičů a vzniká jeden potomek. Princip lze popsat takto [7]:

- 1) První gen prvního rodiče se vloží do chromozomu potomka.
- 2) První gen druhého rodiče (nemá-li stejnou hodnotu jako první gen prvního rodiče) se vloží do chromozomu potomka.
- 3) Dále se postupuje stejným způsobem jako v předcházejících dvou bodech, tj. střídají se ve vkládání genů oba chromozomy, přičemž je kontrolováno, zda-li se již uvažovaná hodnota v chromozomu potomka nenachází (pokud ano, gen s danou hodnotou není vložen).



Obrázek č.8: Ilustrace vzniku potomka křížením AP. Šedým podbarvením jsou vyznačeny pozice převzaté z chromozomu prvního rodiče, modrým podbarvením jsou vyznačeny pozice převzaté z chromozomu druhého rodiče.

2.2.9 Mutace Swap

Operátor mutace Swap spočívá v jednoduché záměně dvou náhodně vybraných genů

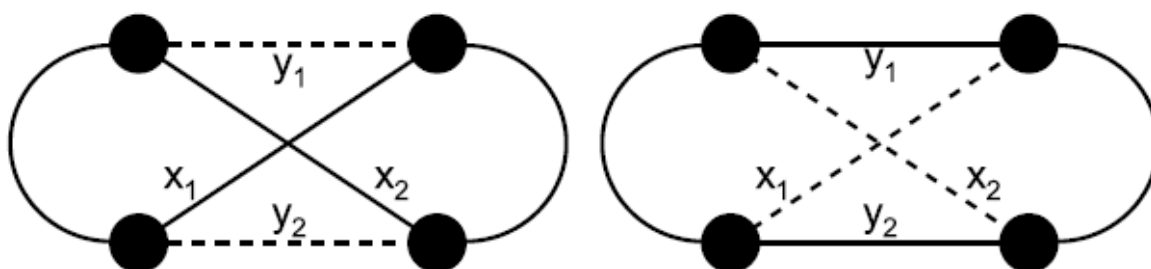


Obrázek č.9: Ilustrace Swap mutace.

2.2.10 Mutace 2-OPT a 3-OPT

Algoritmus 2-OPT patří mezi lokální heuristiky. Je nutné mu dodat vygenerovanou okružní jízdu (permutaci měst), kterou se posléze snaží vylepšit. Princip vylepšení spočívá v náhodném výběru dvou přímo nenavazujících hran a jejich následné výměně.

Tento algoritmus lze jednoduše zobecnit na k-OPT. Potom se vybírá nejvýše k-hran nebo právě k-hran.



Obrázek č.10: Ilustrace mutace 2-OPT (převzato z [6]).

2.2.11 Nahrazující mutace

Operátor nahrazující mutace (DM – displacement station) spočívá ve výběru náhodné cesty v daném jedinci, vyjmutí této cesty a umístění na jinou, náhodně zvolenou pozici.



Obrázek č.11: Ilustrace nahrazující mutace pro zvolenou cestu „3 4 5“ a zvolenou pozici vložení 5.

2.2.12 Vkládací mutace

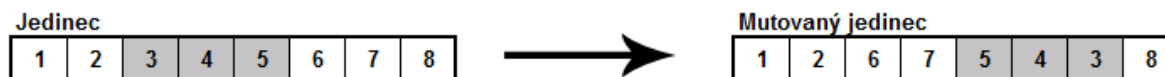
Operátor vkládací mutace (ISM – insertion mutation) spočívá ve výběru náhodného genu, vyjmutí tohoto genu a umístění na jinou, náhodně zvolenou pozici.



Obrázek č.12: Ilustrace vkládací mutace pro vybraný gen 5 a zvolenou pozici vložení 2.

2.2.13 Inverzní mutace

Operátor inverzní mutace (IVM – inversion mutation) spočívá ve výběru náhodné cesty v daném jedinci, vyjmutí této cesty, inverzi této cesty a umístěním na jinou, náhodně zvolenou pozici.



Obrázek č.13: Ilustrace inverzní mutace pro zvolenou cestu „3 4 5“ a zvolenou pozici vložení 5.

2.2.14 Mixující mutace

Operátor mixující mutace (SM – scramble mutation) spočívá ve výběru náhodné cesty v daném jedinci a libovolné záměně hodnot genů této sekvence.

2.3 Selektce a obnova populace

Selektce rozhoduje o výběrů jedinců vhodných k další reprodukci. Tento proces by měl napodobovat přirozený výběr, v němž má větší šanci výběru silnější jedinec. Mezi často používané selekční techniky patří lineární uspořádání, exponenciální uspořádání, proporcionální selektce a turnajová selektce. V rámci této práce se pro volbu jedinců vhodných pro proces reprodukce stará turnajová selektce.

Turnajová selektce se inspirovat procesy známými z přírody, kdy se jedinci určitého druhu musí vzájemně fyzicky utkat o možnost účasti v reprodukčním procesu. Podobně v této technice se náhodně určí k jedinců (počet jedinců k lze v programu nastavit), z nichž je vybrán jedinec s nejvyšším fitness.

Pro obnovu populace, která určuje jedince postupující do další generace, se používá kombinace technik částečné obnovy (steady-state) a elitismu. Elitismus spočívá v možnosti určit, kolik nejlépe ohodnocených potomků automaticky postupuje do další populace. Zbývající místa jsou poté doplněna jedinci vybranými pomocí turnaje ze sjednocení množiny potomků a množiny rodičů. Množina potomků má přitom vždy stejnou velikost jako množina rodičů (tj. aplikuje se křížení tak dlouho, dokud nezískáme dostatečný počet jedinců).

Kapitola 3

Experimenty

Při experimentování s navrženým modelem bylo využito instancí problému obchodního cestujícího získaných z databází *National TSP Collection*¹ a *VLSI TSP Collection*². V jednotlivých experimentech je analyzován zejména vliv výběru rekombinačních operátorů na kvalitu (cenu cesty) dosaženého řešení. Pozornost je rovněž věnována vzájemnému srovnání operátorů z hlediska jejich časové náročnosti.

3.1 Srovnání operátorů křížení podle kvality řešení

V tomto experimentu je zjišťován vliv výběru operátoru křížení na kvalitu (cenu cesty) dosaženého řešení. Výsledné křivky pro každý zkoumaný operátor vznikly aproximací z deseti nezávislých běhů. K použitým úlohám je známo jejich optimální řešení (v grafu je vyznačeno), díky čemuž lze objektivně posoudit kvalitu získaných výsledků.

Velikost populace v prováděných experimentech byla stanovena empiricky s ohledem na výpočetní možnosti strojů, na kterých experimentování probíhalo. Testování potvrdilo obecně známou skutečnost, že při určitém počtu jedinců již nemá valný efekt populaci zvětšovat, neboť se to na výsledku takřka neprojeví a dochází pouze k růstu časové náročnosti. Při zvolené velikosti populace byl experiment prováděn po takový počet generací, dokud ještě docházelo ke změnám výsledné ceny.

Rovněž velikost mutace byla určena na základě provedených experimentů. V rámci snahy o nalezení optimální hodnoty tohoto parametru bylo zjištěno, že příliš malá mutace nedokáže vnášet dostatečný počet změn k zachování rozmanitosti populace a dochází k předčasné konvergenci. Naopak příliš velká mutace má zcela opačný efekt a neustálé poškození významné části genotypu nevede k nalezení rozumného řešení. Jako typ mutace byl zvolen 2-OPT, jelikož vykazoval nejlepší výsledky pro analyzované mutační operátory.

¹ <http://www.tsp.gatech.edu/world/countries.html>

² <http://www.tsp.gatech.edu/vlsi/index.html>

Souhrnné údaje o experimentu č. 1

Název problému: **Quatar** (examples_country/02_katar_194.tsp)

Dimenze (počet měst): 194

Společné parametry evoluce:

Velikost populace: 500

Počet generací: 5 000

Typ mutace: 2-OPT

Pravděpodobnost mutace: 0,15

Počet účastníků turnajové selekce: 2

Elitismus: 1

Výsledky evoluce:

Cena nejlepšího řešení (AP): 10 562

Cena nejlepšího řešení (ERX): 10 242

Cena nejlepšího řešení (CX): 10 533

Cena nejlepšího řešení (MPX): 10 489

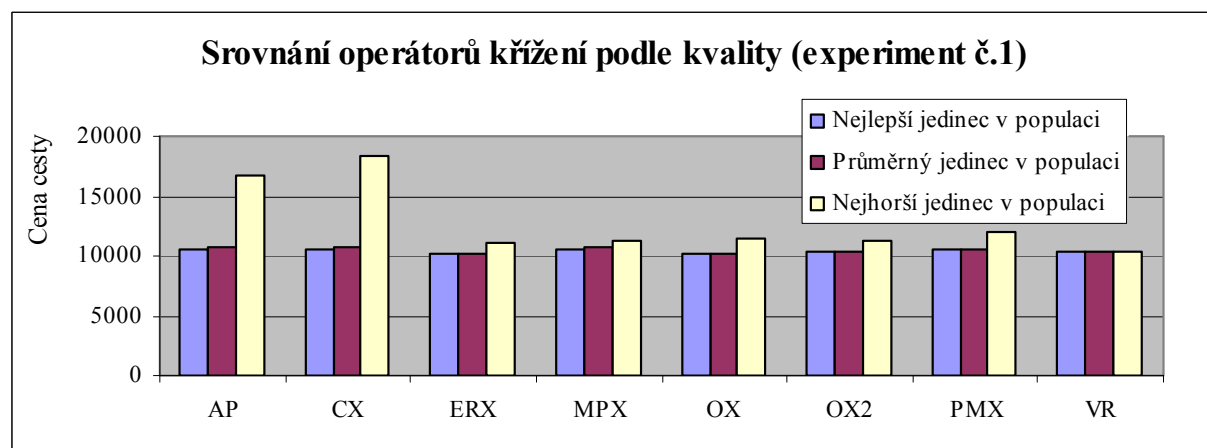
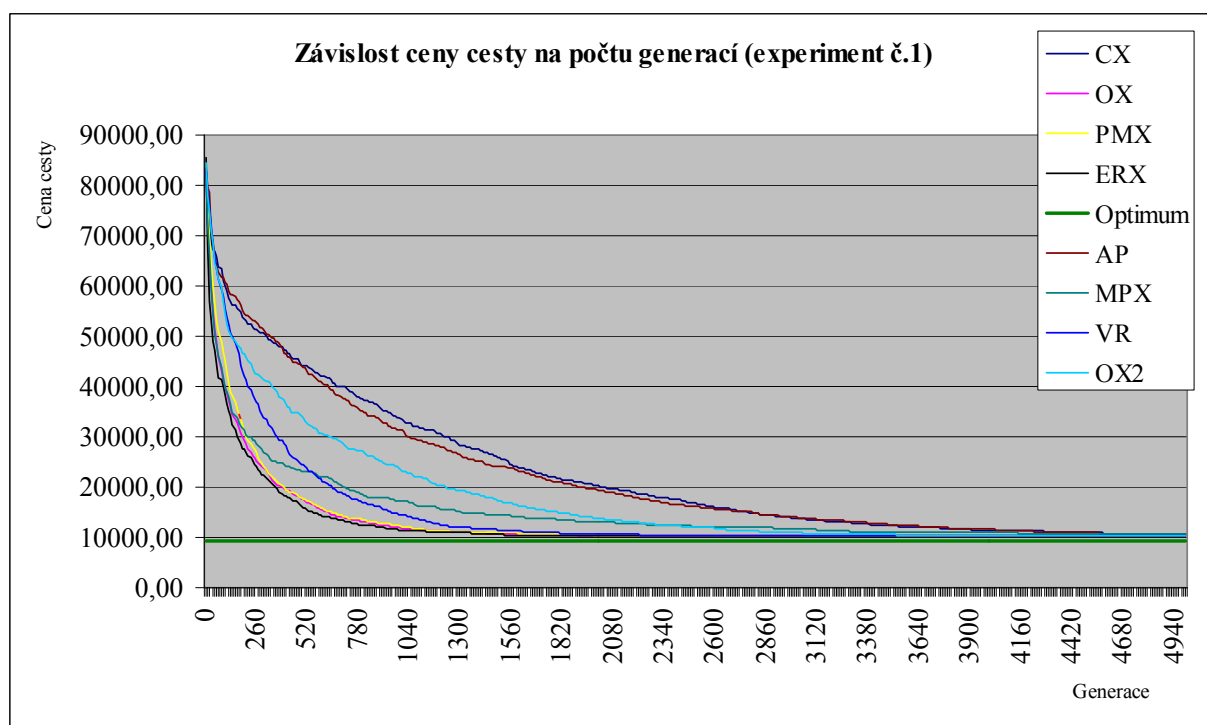
Cena nejlepšího řešení (OX): 10 249

Cena nejlepšího řešení (OX2): 10 365

Cena nejlepšího řešení (PMX): 10 474

Cena nejlepšího řešení (VR): 10 347

Cena optimálního řešení: 9 352



Souhrnné údaje o experimentu č. 2

Název problému: pbl395 (examples_vlsi/03_pbl_395.tsp)

Dimenze (počet měst): 395

Společné parametry evoluce:

Velikost populace: 75

Počet generací: 10 000

Typ mutace: 2-OPT

Pravděpodobnost mutace: 0,15

Počet účastníků turnajové selekce: 2

Elitismus: 1

Výsledky evoluce:

Cena nejlepšího řešení (AP): 2 360

Cena nejlepšího řešení (ERX): 1 857

Cena nejlepšího řešení (MPX): 3 320

Cena nejlepšího řešení (OX): 1 948

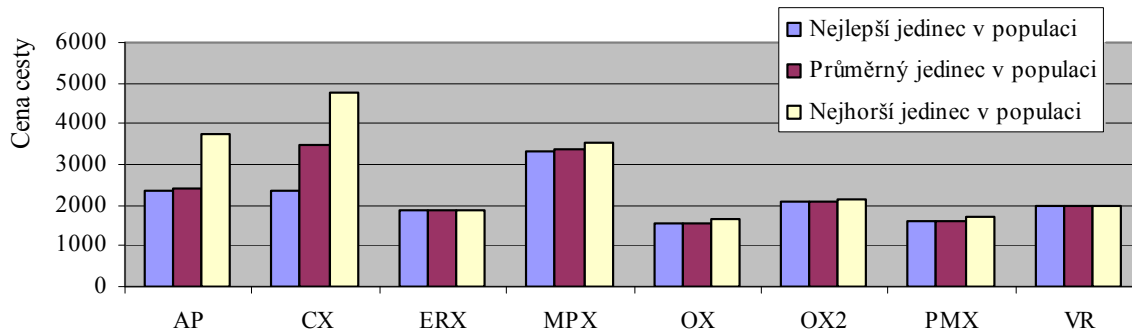
Cena nejlepšího řešení (OX2): 2 103

Cena nejlepšího řešení (PMX): 1 917

Cena nejlepšího řešení (VR): 1 970

Cena optimálního řešení: 1 281

Srovnání operátorů křížení podle kvality (experiment č.2)



Z uvedených výsledků a křivek grafů vyplývá, že rozdíly mezi cenou nejlepších nalezených řešení po dokončení evoluce nejsou u jednotlivých operátorů křížení při dané konfiguraci parametrů příliš markantní. Při testování s jinými konfiguracemi parametrů byly dosažené nejlepší výsledky jednotlivých operátorů křížení opět vzájemně velmi podobné.

3.2 Srovnání operátorů mutace podle kvality řešení

V tomto experimentu je zjišťován vliv výběru operátoru mutace na kvalitu (cenu cesty) dosaženého řešení. Výsledné křivky pro každý zkoumaný operátor vznikly aproximací z deseti nezávislých běhů. K použitým úlohám je známo jejich optimální řešení (v grafu je vyznačeno), díky čemuž lze objektivně posoudit kvalitu získaných výsledků.

Velikost populace v prováděných experimentech byla stanovena empiricky s ohledem na výpočetní možnosti strojů, na kterých experimentování probíhalo. Testování potvrdilo obecně známou skutečnost, že při určitém počtu jedinců již nemá valný efekt populaci zvětšovat, neboť se to na výsledku takřka neprojeví a dochází pouze k růstu časové náročnosti. Při zvolené velikosti populace byl experiment prováděn po takový počet generací, dokud ještě docházelo ke změnám výsledné ceny.

Rovněž velikost mutace byla určena na základě provedených experimentů. V rámci snahy o nalezení optimální hodnoty tohoto parametru bylo zjištěno, že příliš malá mutace nedokáže vnášet dostatečný počet změn k zachování rozmanitosti populace a dochází k předčasné konvergenci. Naopak příliš velká mutace má zcela opačný efekt a neustálé poškozování významné části genotypu nevede k nalezení rozumného řešení. Jako typ křížení byl zvolen OX, jelikož vykazoval nejlepší výsledky z analyzovaných operátorů.

Souhrnné údaje o experimentu č. 3

Název problému: Qatar (examples_country/02_katar_194.tsp)

Dimenze (počet měst): 194

Společné parametry evoluce:

Velikost populace: 500

Počet generací: 5 000

Typ křížení: OX

Počet účastníků turnajové selekce: 2

Pravděpodobnost mutace: 0,15

Elitismus: 1

Výsledky evoluce:

Cena nejlepšího řešení (2-OPT): 10 249

Cena nejlepšího řešení (3-OPT): 10 877

Cena nejlepšího řešení (DM): 11 308

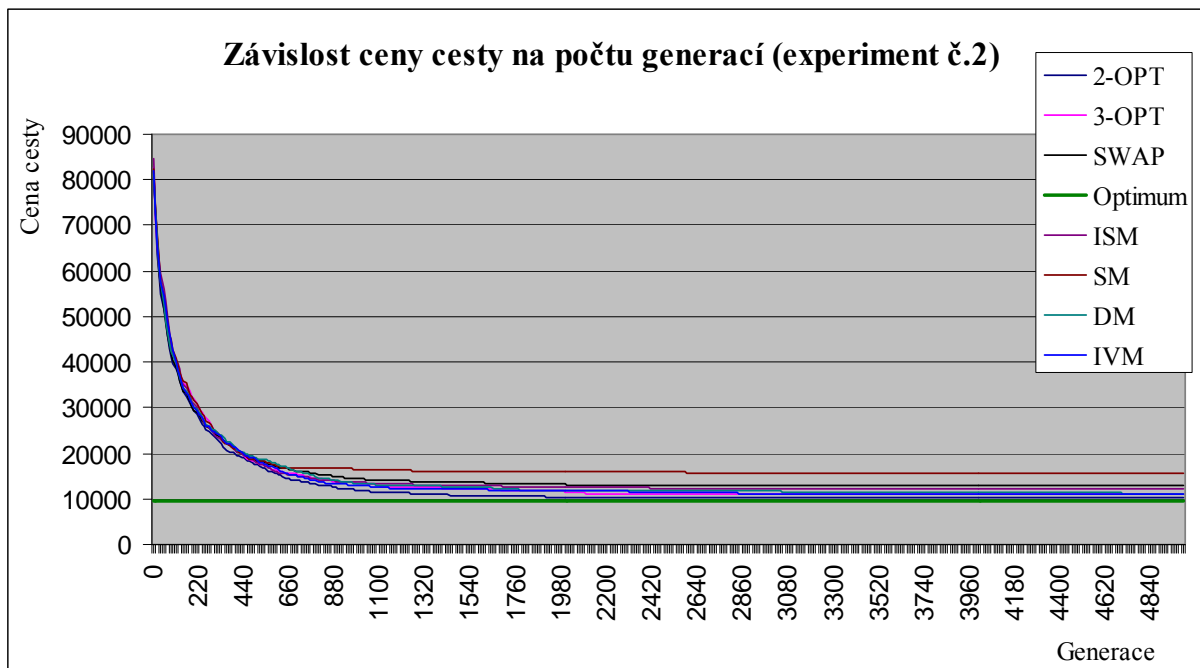
Cena nejlepšího řešení (ISM): 12 231

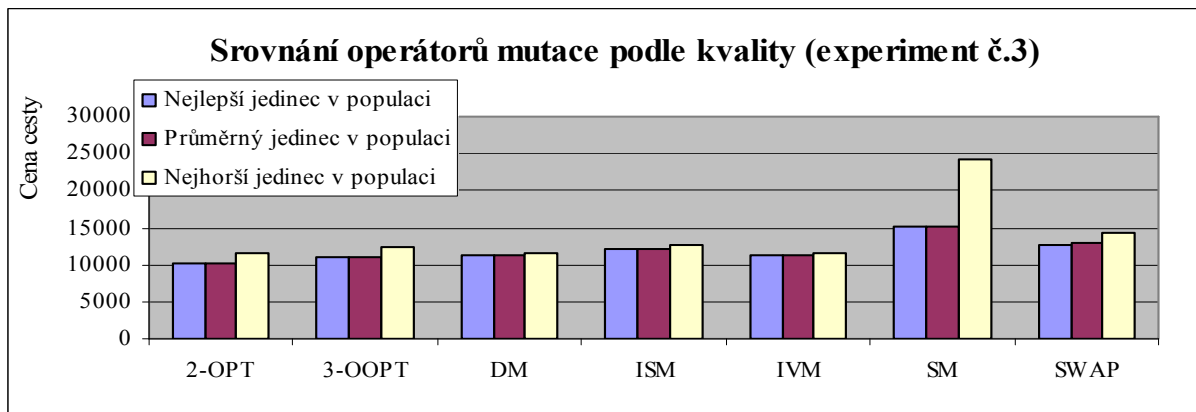
Cena nejlepšího řešení (IVM): 11 190

Cena nejlepšího řešení (SM): 15 029

Cena nejlepšího řešení (SWAP): 12 794

Cena optimálního řešení: 9 352





Souhrnné údaje o experimentu č. 4

Název problému: Uruguay (examples_country/03_uruguay_737.tsp)

Dimenze (počet měst): 737

Společné parametry evoluce:

Velikost populace: 500

Počet generací: 15 000

Typ křížení: OX

Počet účastníků turnajové selekce: 2

Pravděpodobnost mutace: 0,15

Elitismus: 1

Výsledky evoluce:

Cena nejlepšího řešení (2-OPT): 95 368

Cena nejlepšího řešení (3-OPT): 140 912

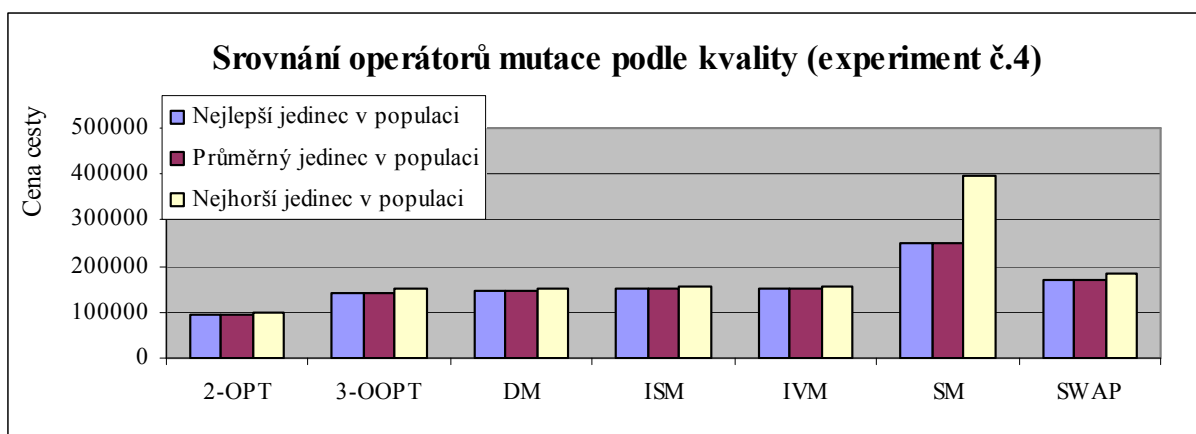
Cena nejlepšího řešení (DM): 144 553

Cena nejlepšího řešení (ISM): 151 769

Cena nejlepšího řešení (IVM): 151 311

Cena nejlepšího řešení (SM): 248 298

Cena nejlepšího řešení (SWAP): 171 736



3.3 Srovnání operátorů křížení podle počtu evaluací

V tomto experimentu je zjišťován vliv výběru operátoru křížení na celkový počet evaluací. Tato metrika je sledována na jednoduchém problému „Western Sahara“ s pouhými 29 městy, aby bylo zajištěno, že operátory vždy naleznou globální optimum a srovnání tak bylo objektivní. Naměřené hodnoty jsou aproximací z 10 nezávislých běhů.

Souhrnné údaje o experimentu č. 5

Název problému: Western Sahara (examples_country/01_western_sahara_29.tsp)

Dimenze (počet měst): 29

Společné parametry evoluce:

Velikost populace: 200

Cena optimálního řešení: 27 603

Typ mutace: 2-OPT

Pravděpodobnost mutace: 0,15

Počet účastníků turnajové selekce: 2

Elitismus: 1

Výsledky evoluce:

Počet evaluací (AP): 49 200

Počet evaluací (CX): 40 913

Počet evaluací (ERX): 11 973

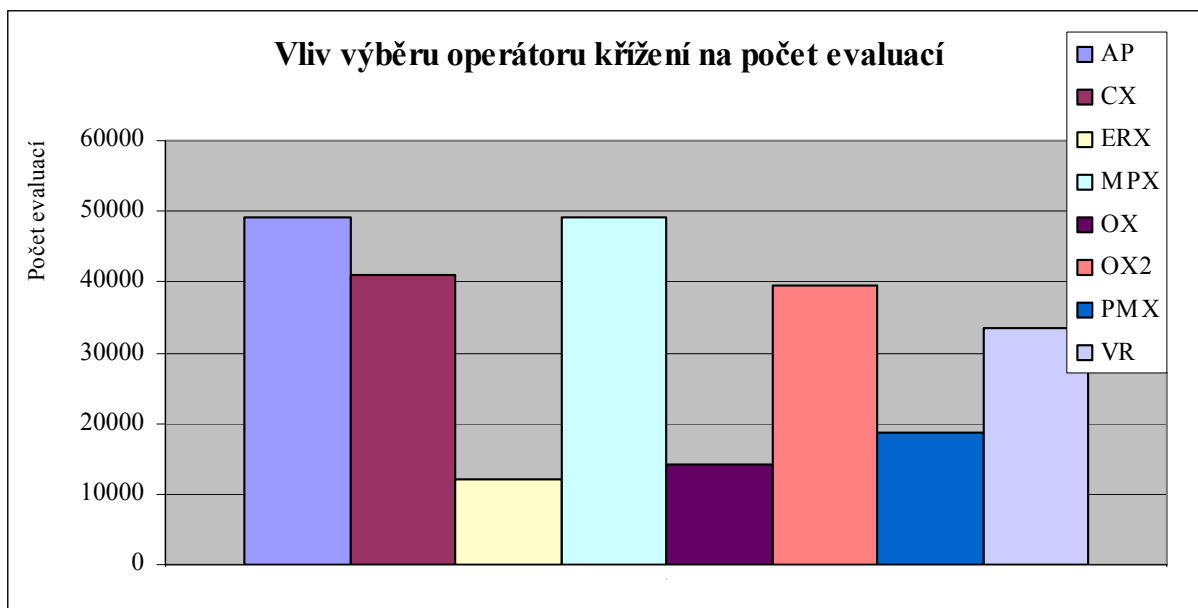
Počet evaluací (MPX): 49 000

Počet evaluací (OX): 14 113

Počet evaluací (OX2): 39 600

Počet evaluací (PMX): 18 785

Počet evaluací (VR): 33 600

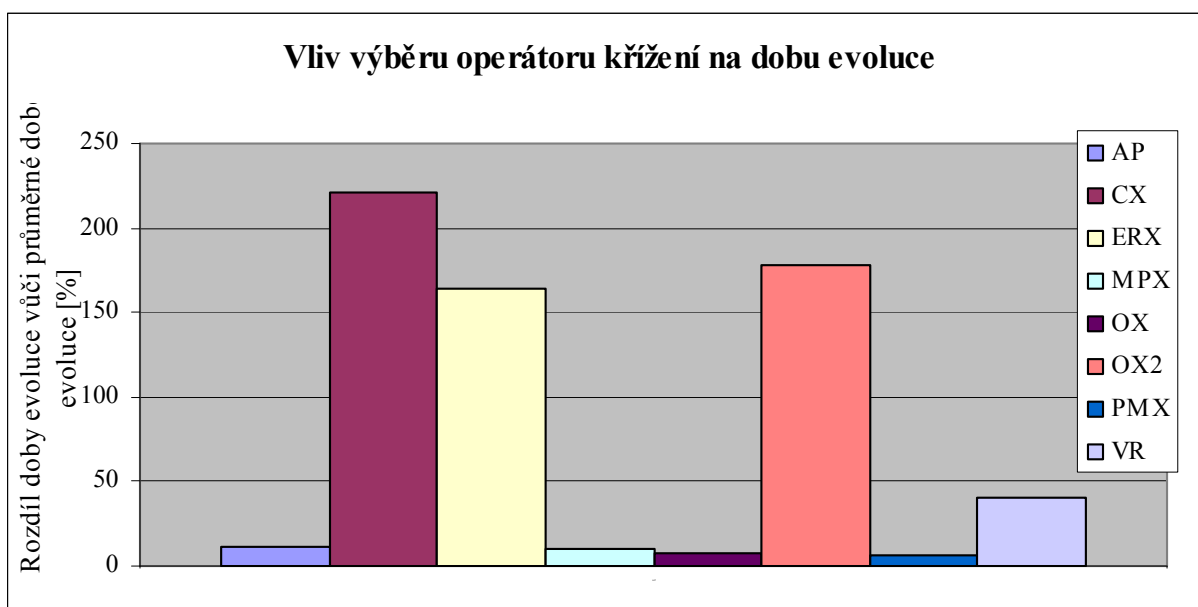


3.4 Srovnání operátorů podle časové náročnosti

V tomto experimentu jsou vzájemně porovnávány operátory křížení a mutace z hlediska jejich časové náročnosti. Naměřené hodnoty jsou ovlivněny více faktory, zejména však volbou řešené úlohy, způsobem implementace (viz Kapitola 2) a také použitým hardware (velikost cache, ...). Z toho důvodu získané výsledky nelze zobecňovat.

Na základě provedených měření bylo zjištěno, že časová náročnost se při použití různých typů mutace takřka nemění. Jedinou výjimkou jsou operátory SWAP a ISM, u kterých je doba evoluce kratší přibližně o 1%.

Větší rozdíly v časové náročnosti lze sledovat při změnách typu křížení. Zhruba srovnatelnou časovou náročnost mají operátory AP, MPX, PMX, OX, VR a poté operátory ERX, CX, OX2. Rozdíl v době evoluce mezi těmito dvěma skupinami je velmi markantní. To je způsobeno zejména skutečností, že při implementaci operátorů ERX, CX a OX2 je nutné používat další pomocné struktury pro každé uvažované město (hranová tabulka u operátoru ERX, tabulka sekcí u operátoru CX).



Kapitola 4

Závěr

V rámci této práce byl navržen a implementován program pro výpočet cesty obchodního cestujícího využívající ke své činnosti genetický algoritmus. Hlavní pozornost byla přitom věnována analýze vybraných typů rekombinačních operátorů, detailně popsanych v kapitole 2, z hlediska jejich vhodnosti pro řešení tohoto problému.

Experimentováním s navrženým modelem (viz Kapitola 3) bylo zjištěno, že volba operátoru křížení nemá významný vliv na kvalitu řešení (ve významu ceny cesty nejlepšího jedince), která je zhruba srovnatelná u všech zkoumaných typů. Doba evoluce (měřená v počtech časových jednotek, nikoliv v generacích) je však významně kratší u operátorů OX, PMX, AP, VR a MPX. Oproti tomu typ mutace má významný vliv na kvalitu řešení s tím, že nejúspěšnější mutační operátor 2-OPT vykazuje průměrně zhruba o 20% nižší cenu než mutační operátor SWAP a o 50-100% nižší cenu než nejhorší mutační operátor SM. Doba evoluce se však vlivem výběru operátorů mutace prakticky nemění.

Program by jistě bylo možné dále rozvíjet. Efektivitu algoritmů pro implementované operátory lze dále zlepšovat a rovněž experimentovat s jejich modifikacemi, případně po jistém počtu generací náhodně střídat.

Reference

- [1] Hynek, J.: Genetické algoritmy a genetické programování. Grada Publishing, a.s., 2000.
- [2] Kvasnička, V.; Pospíchal, J.; Tiňo, P.: Evolučné algoritmy. STU Bratislava, 2000. Kapitola 7.1, s. 92-95.
- [3] Rubicite Genetic Algorith Framework. [online], [cit. 2010-04-10]
URL <<http://www.tsp.gatech.edu/>>
- [4] Travelling Salesman Problem. [online], poslední modifikace 2010-03-10
[cit. 2010-04-10].
URL <<http://www.rubicite.com/genetic/tutorial/general1.php>>
- [5] Schwarz, J.; Sekanina, L.: Aplikované evoluční algoritmy: Studijní opora.
- [6] Weiner, V.: Řešení problému obchodního cestujícího pomocí PC. Fakulta dopravní, ČVUT v Praze, 2008.
- [7] Larranaga, P., Kuijpers, C; Murga, R; Inza, I.; Dizdarevic, S.: Genetic Algorithms for the Travelling Salesman Problem: A Review of Representations and Operators. Artificial Intelligence Review 13, 1999.
URL <<http://www.springerlink.com/content/qj3230n51tp22226/fulltext.pdf>>

Příloha A: Manuál k programu

A.1 Instalace a požadavky programu

Požadavky:

- Operační systém Linux.
- Překladač jazyka C++.
- Knihovna Qt verze 4 a vyšší.
- Utilita `make`.

Instalace:

Instalace programu proběhne po spuštění skriptu `install.sh`, který vytvoří ve stejném adresáři spustitelný soubor `tspl`. Poznámka: Tento skript je připraven pro instalaci knihovny Qt na serveru merlin. Pokud je program instalován v jiném prostředí, je potřeba změnit cestu ke knihovně Qt (nastavení proměnné `$QM` ve skriptu `install.sh`).

A.2 Formát konfiguračního souboru

Program lze spustit s parametrem určujícím jméno konfiguračního souboru. Konfigurační soubor udává cestu k souboru se vstupními daty, cestu k výstupnímu souboru k uložení výsledků a základní parametry běhu genetického algoritmu:

<code>PROBLEM_FILE</code>	Cesta k souboru s řešeným problémem.
<code>S_POPULATION</code>	Velikost populace.
<code>N_GENERATIONS</code>	Počet generací evolučního běhu.
<code>P_MUTATION</code>	Pravděpodobnost mutace.
<code>T_MUTATION</code>	Typ operátoru mutace.
<code>T_CROSSOVER</code>	Typ operátoru křížení.
<code>S_TOURNAMENT</code>	Počet jedinců, kteří se účastní turnajové selekce.
<code>S_ELITISM</code>	Počet jedinců, na které se aplikuje elitismus.
<code>LOG_INTERVAL</code>	Počet generací, po kterých bude ukládána statistika.

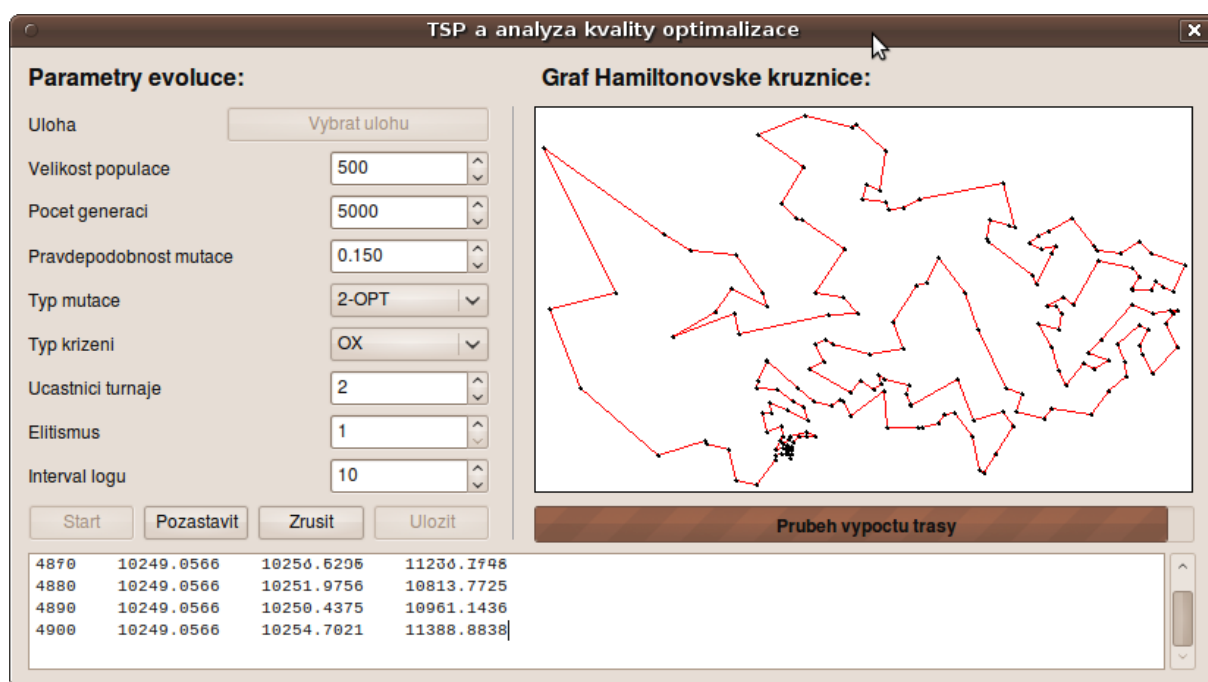
A.3 Formát vstupního datového souboru

Vstupní datové soubory se řídí konvencí knihovny TSPLib³, což je knihovna instancí problému obchodního cestujícího. Instance problému je v souboru popsána specifikační a datovou částí. Ve specifikační části je uveden název problému, dimenze problému (počet měst) a případné komentáře, zatímco v datové části jsou pak samotná data – na každém řádku se nachází aktuální číslo města a jeho pozice na mapě (souřadnice x, y).

³ <http://comopt.ifl.uni-heidelberg.de/software/TSPLIB95/>

A.4 Ovládání programu

V programu je možné jednak vybrat cestu k řešení úloze a dále pak nastavit parametry evolučního procesoru. Tyto parametry jsou totožné s parametry vysvětlenými v sekci A.2. Průběžné i celkové statistiky probíhajícího evolučního běhu jsou vypisovány do samostatného boxu s výsledky v dolní části okna. V pravé části okna lze sledovat grafickou reprezentaci nejlepšího nalezeného řešení aktuální generace (viz obrázek A.1).



Obrázek A.1: Ukázka aplikace (úloha Quatar).