

Aproximácia maximálnej nezávislej množiny vrcholov pomocou genetického algoritmu

(case study pre predmet „Evolučné algoritmy“)

Čestmír Hýbl

študent 4. ročníka Informatiky
na Matematicko-fyzikálnej fakulte Univerzity Komenského v Bratislave

Február 2000

(e-mail: cestmir@pobox.sk)

Obsah

Úvod.....	2
1. Stručne z teórie grafov.....	2
2. Formalizácia problému.....	3
3. Špecializácia GA pre MISP.....	3
3.1 Neformálne.....	3
3.2 Reprezentácia a kódovanie.....	3
3.3 Účelová funkcia.....	3
3.4 Parametre GA.....	4
4. Experimenty a výsledky.....	4
4.1 Metodika.....	4
4.1.1 Stochastické grafy.....	4
4.1.2 „Regulárne“ grafy.....	5
4.1.3 „Regulárne“ grafy a Hammingova bariéra.....	5
4.2 Experimenty.....	5
4.2.1 Experiment 1: Exemplárne riešenie.....	5
4.2.2 Experiment 2: Overenie výsledkov z [1].....	6
4.2.3 Experiment 3: Výpočet sily rankovaním.....	6
4.2.4 Experiment 4: Výpočet sily rankovaním, elitizmus.....	7
4.2.5 Experiment 5: Výpočet sily rankovaním, elitizmus, uniformný crossover.....	7
4.2.6 Experiment 6: Hľadanie riešenia na grafoch s počtom vrcholov 200.....	7
4.2.7 Experiment 7: Štatistika – závislosť riešenia od parametra P_{mutation}	8
4.2.8 Experiment 8: Štatistika – závislosť riešenia od parametra μ (veľkosť populácie).....	8
4.2.9 Experiment 9: Štatistika – závislosť riešenia od parametra $P_{\text{crossover}}$	8
4.2.10 Experiment 10: Štatistika – závislosť riešenia od parametra $P_{\text{mutation_inversion}}$	8
4.2.11 Experiment 11: Štatistika – závislosť riešenia od parametra ElitismCoefficient.....	9
Záver.....	9
Použitá literatúra.....	9
Príloha A - Tabuľky, grafy.....	10
Príloha B - Genetický algoritmus.....	15
Príloha C - Simulátor.....	17

ÚVOD

Evolučné algoritmy a špeciálne genetické algoritmy (GA) ponúkajú silný abstraktný základ pre riešenie optimalizačných problémov, ich nasadenie je elegantné nakoľko na rozdiel od iných heuristik zväčša vyžaduje len veľmi málo doménovo špecifickej informácie, pričom riešenie možno „ladiť“ v širokom rozsahu (voľbou kódovania, voľbou funkcie sily, voľbou parametrov GA).

V tejto práci sa zaoberáme aplikáciou genetického algoritmu pri aproximačnom riešení grafového optimalizačného problému – hľadania maximálnej nezávislej množiny vrcholov (ďalej len „MISP“ – maximal independent set problem). Hneď na úvod poznamenajme, že MISP patrí do triedy NP-úplných problémov, teda preň neexistuje optimálny algoritmus pracujúci v polynomiálnom čase. Hľadanie rýchlych, aproximačných algoritmov (heuristik) preto má zmysel.

Riešenie, popísané v tejto práci je veľmi podobné riešeniu prezentovanému v [1], nakoľko nám uvedená práca, žiaľ, bola predložená spolu so zadaním úlohy ako „doporučená literatúra“ (značne to zúži priestor na vlastné tvorivé uvažovanie o riešení...). Pri návrhu nášho riešenia sme sa najprv snažili vyhnúť „bezbrehému kopírovaniu“ uvedenej práce, čo však bolo len ťažké „dodržať“ nakoľko sa jedná o „najminimalistickejšie“ možné riešenie problému pomocou GA – ktoré keď raz človek niekde vidí, už ho ťažko napadne čokoľvek iné. V tomto prípade je však otázka, či by informatika so základnou znalosťou teórie grafov, grafových algoritmov a genetických algoritmov vôbec mohlo napadnúť nejaké iné riešenie.

Nakoniec sme preto zvolili iný prístup – na našu prácu možno nazerať ako na nezávislé overenie výsledkov z [1].

U čitateľa budeme predpokladať základné znalosti z teórie genetických algoritmov (Príloha A, viac v [2]) a teórie grafov (Kapitola 1, viac napr. v [4]). Model nástroja na výpočet genetického algoritmu (ďalej len „simulátor“) je popísaný v Prílohe C.

1. STRUČNE Z TEÓRIE GRAFOV

Graf je usporiadaná dvojica $G = (V, E)$, kde V je množina vrcholov, $E \subseteq V \times V$ je množina hrán medzi vrcholmi.

Množinu vrcholov grafu G označujeme $V(G)$, množinu hrán $E(G)$. Počet vrcholov konečného grafu (inak by sme formálne museli hovoriť o mohutnosti množiny vrcholov) nazývame stupeň grafu a označujeme $|G| = |V(G)|$. Počet hrán označujeme $\|G\| (=|E(G)|)$. V texte obvykle používame premennú n pre stupeň grafu a m pre počet hrán.

Vrcholy zvykneme označovať malými písmenami $u, v, x, y, z \in V(G)$. Hranu $(u, v) \in E(G)$ medzi vrcholmi u a v skrátene označujeme uv (alebo vu).

Podgraf grafu $G = (V, E)$ je graf $G' = (V', E')$, pre ktorý platí $V' \subseteq V$, $E' \subseteq E$. Reláciu označujeme $G' \subseteq G$.

Podgraf G' nazývame indukovaný graf, ak $\forall uv \in E: (u, v \in V') \Rightarrow (uv \in E')$ (teda ak obsahuje všetky hrany z G , ktoré mali oba konce vo V'). Označujeme ho $G[V']$ (čítame: podgraf indukovaný množinou vrcholov V').

Podmnožina U množiny vrcholov V sa nazýva nezávislá množina vrcholov ak $\forall u, v \in U: uv \notin E$ (žiadne dva vrcholy z nej nie sú v grafe spojené hranou). Nezávislá množina vrcholov sa nazýva maximálna, ak má spomedzi všetkých nezávislých množín grafu najväčšiu mohutnosť. Mohutnosť niektorej z maximálnych nezávislých množín grafu nazývame stabilita grafu a označujeme $\alpha(G)$.

Matica susednosti grafu $A = [a_{ij}]_{n \times n}$ je definovaná:

$$a_{ij} := \begin{cases} 1 & \text{ak } v_i v_j \in E(G) \\ 0 & \text{inak} \end{cases}$$

Vrcholové pokrytie grafu G je množina vrcholov V' taká, že $\forall uv \in E(G): u \in V' \vee v \in V'$ (množina vrcholov, "pokrývajúca" všetky hrany grafu). Komplement grafu $G = (V, E)$ je graf $G^C = (V, [V]^2 - E)$. Kompletný graf alebo klika je graf v ktorom platí $E = [V]^2$.

Veta 1:

Pre ľubovoľný graf $G = (V, E)$ a $U \subseteq V$ sú nasledovné výroky ekvivalentné:

- U je maximálna nezávislá množina G
- U je maximálna klika v G^C
- $V - U$ je minimálne vrcholové pokrytie G

Uvedená veta nám hovorí, že pokiaľ získame heuristiku na riešenie jedného z uvedených problémov, môžeme ňou riešiť všetky (MISP, hľadanie maximálnej kliky, hľadanie minimálneho vrcholového pokrytia).

2. FORMALIZÁCIA PROBLÉMU

Našou úlohou je zostaviť algoritmus, podľa nasledovnej špecifikácie:

Vstup:

Vhodne zakódovaný konečný graf G .

V našom prípade bude graf reprezentovaný maticou susednosti $A = [a_{ij}]_{n \times n}$.

Výstup:

Niektorá z nezávislých množín vrcholov G , v optimálnom prípade maximálna

3. ŠPECIALIZÁCIA GA PRE MISP

3.1 Neformálne

Aby sme mohli nejaký optimalizačný problém riešiť pomocou GA, treba vyriešiť najmä dva kľúčové podproblémy:

- zvoliť, čo (z hľadiska riešeného problému) budú reprezentovať jedince v populácii GA a ako „to“ bude kódované v chromozóme
- zvoliť „dobrú“ účelovú funkciu a efektívne ju algoritmicke implementovať

Potom treba vhodne nastaviť parametre GA tak, aby hľadal čo možno najoptimálnejšie riešenia.

3.2 Reprezentácia a kódovanie

V našom prípade bude chromozóm reprezentovať priamo jedno potenciálne riešenie – množinu vrcholov $U \subseteq V = \{v_1, v_2, \dots, v_n\}$. Bude to binárny vektor $\beta = [b_i]_n$ s vlastnosťou:

$$b_i := 1 \Leftrightarrow (v_i \in U)$$

Množinu vrcholov, charakterizovanú chromozómom budeme označovať U_β .

V populácii sa teda budú môcť nachádzať aj chromozómy, ktoré nereprezentujú platné riešenie (nimi charakterizovaná množina nebude nezávislá). Také chromozómy však budeme, ako uvedieme ďalej, účelovou funkciou silno penalizovať.

3.3 Účelová funkcia

Začneme s nasledovnou jednoduchou a intuitívnou funkciou:

$$f(\beta) := -|U_\beta| + n \|G[U_\beta]\| \quad (1)$$

Neformálne: -(počet vrcholov v množine charakterizovanej chromozómom) + n^* (počet hrán medzi vrcholmi v množine).

Podgraf je indukovaný nezávislou množinou vrcholov \Leftrightarrow neobsahuje žiadne hrany.

Pre korektné riešenie teda platí:

$$\|G[U_\beta]\| = 0 \quad (2)$$

Funkcia teda odmieňa (pripomíname, že GA hodnotu účelovej funkcie minimalizuje) nárast mohutnosti množiny a penalizuje chromozómy, ktoré nereprezentujú platné riešenie. V lineárnom usporiadaní „≤“ definovanom funkciou sú všetky neplatné chromozómy umiestnené až za všetkými platnými pričom chromozómy/riešenia v oboch skupinách sú funkciou ďalej zoradené podľa ich (ne)kvality. Platné chromozómy teda budú umelou evolúciou silno preferované a bude medzi nimi prebiehať súťaž smerom k zvyšovaniu kvality riešenia.

Funkciu ďalej rozvedieme na aritmetický výraz nad reprezentáciami vstupného grafu a množiny U_β :

$$f(\beta) := -\left(\sum_{i=1}^n b_i\right) + n\left(\sum_{i=1}^n \sum_{j=i}^n b_i \cdot a_{ij}\right)$$

3.4 Parametre GA

Pre úvodné experimenty sme použili štandardný koncept GA ako je uvedený v Pseudokóde 1 a pre porovnanie experimentálnych výsledkov sme na úvod prevzali parametre GA, použité v [1] aj napriek tomu, že v našom prípade sa neukázali ako optimálne.

Súhrn úvodného nastavenia operátorov a parametrov GA:

- O_{mutation} : inverzia bitov chromozómu nezávisle „po jednom“. Pravdepodobnosť inverzie (každého) jedného bitu P_{mutation} .
- $O_{\text{crossover}}$: 2-bodové kríženie (2 rodičia \rightarrow 2 potomkovia)
- $O_{\text{selection}}$: výber 2 jedincov Goldbergovou ruletou – kvázinahodne, proporcionálne k hodnote sily F
- $O_{\text{reproduction}}$: $O_{\text{crossover}}$ ($\alpha_{\text{parent1}}, \alpha_{\text{parent2}}, \alpha_{\text{child1}}, \alpha_{\text{child2}}$); O_{mutation} (α_{child1}); O_{mutation} (α_{child2})
- $O_{\text{replacement}}$: Podľa Pseudokódu 1. Dcérska populácia je vytváraná nanovo z rodičovskej napĺňaním jedincami (proporcionálne k sile) zvolenými operátorom $O_{\text{selection}}$ reprodukovanými s pravdepodobnosťou $P_{\text{reproductin}}$ operátorom $O_{\text{reproductin}}$. Jedince ktoré neboli zvolené pre reprodukciu sú do dcérskej populácie vkladané bez modifikácie. Proces končí, keď mohutnosť dcérskej populácie dosiahne mohutnosť rodičovskej.
- Výpočet F : dynamickým lineárnym škálovaním intervalu $[f\text{-min}, f\text{-max}]$ hodnôt účelovej funkcie v aktuálnej populácii.
- $P_{\text{reproductin}}$: 0.6
- P_{mutation} : $1/n$ (n = šírka chromozómu = $|G|$)
- Veľkosť populácie μ : 50
- Podmienka ukončenia: opäť pre porovnanie s [1] - definovaním maximálneho počtu volaní účelovej funkcie $t = 2 \cdot 10^4$ pre $n = 100$ resp. $t = 4 \cdot 10^4$ pre $n = 200$ čo pri veľkosti populácie $\mu = 50$ znamená 400 resp. 800 generácií (epoch) algoritmu.
Táto podmienka vyplýva z pozorovania, že po ňou stanovenom počte epoch je už pri daných parametroch GA ďalšie zlepšovanie kvality riešenia veľmi nepravdepodobné.

Nad'alej budeme uvádzať už len odchýlky od uvedených hodnôt.

4. EXPERIMENTY A VÝSLEDKY

4.1 Metodika

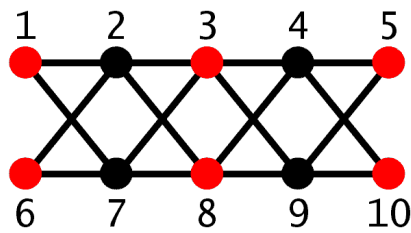
Efektívnosť riešenia problému sme testovali rovnako ako v [1] na vhodne zvolených triedach grafov:

4.1.1 Stochastické grafy

Náhodne generované grafy s danou hustotou hrán h (pravdepodobnosť existencie hrany medzi ľubovoľnými dvoma vrcholmi) a vnútenou nezávislou množinou vrcholov veľkosti k (medzi žiadnymi dvoma z prvých k vrcholov grafu nie je hrana). Budeme ich označovať „MISP-S-n-h-k“, kde h je hustota grafu v %.

4.1.2 „Regulárne“ grafy

Grafy so štruktúrou podľa Obr. 2. Ďalej ich budeme označovať „MISP-R-n“, kde n bude počet vrcholov grafu.



Obr. 2: Graf MISP-R-10

Grafy s počtom vrcholov n ; $n \bmod 4 = 2$ majú vzhľadom na MISP jediné globálne maximum – na obrázku je to červeno vyznačená množina vrcholov $V = \{1, 3, 5, 6, 8, 10\}$. Zvyšné vrcholy tvoria lokálne maximum – na obrázku množina $V' = \{2, 4, 7, 9\}$.

Grafy, kde $n \bmod 4 = 0$ majú dve rovnocenné, disjunktné globálne maximá.

4.1.3 „Regulárne“ grafy a Hammingova bariéra

Prečo sú tieto grafy aj z hľadiska zvolenej reprezentácie zaujímavé ukážeme na zobrazenom MISP-R-10. Množiny V a V' sú reprezentované chromozómami $\alpha_V = 1010110101$ a $\alpha_{V'} = 0101001010$, ktorých Hammingova vzdialenosť je 10, hodnoty účelovej funkcie sú $f(\alpha_V) = -6$, $f(\alpha_{V'}) = -4$ pričom neexistuje gradovaný prechod medzi uvedenými maximami a pri štandardnom operátore mutácie (invertovanie „po bitoch“) je pravdepodobnosť prekonania uvedenej Hammingovej bariéry veľmi malá.

Podotýkame, že problém prekonávania Hammingovej bariéry je tradičným problémom pri aplikácii GA. Je viacero možností, ako sa s ním vysporiadať:

- **neriešiť problém explicitne.** Pokiaľ v GA funguje dostatočný vnútorný paralelizmus, funkcia sily nie je príliš elitárska – v populácii GA sa teda dá očakávať veľká pestrosť v genotypoch jedincov – nebude sa teda často stávať, že populáciu rýchlo naplnia jedince s podobným genotypom, všetky ležiace v lokálnom maxime z ktorého by sa kvôli Hammingovej bariére nedostali.

- **zmena kódovania** – reprezentácie jedinca v chromozóme. Napr. v prípade, že chromozóm (úsek chromozómu) reprezentuje číslo v binárnej sústave, ktoré je parametrom účelovej funkcie, je vhodné používať reprezentáciu v Grayovom kóde. Odstráni sa tým Hammingova bariéra pri preklápaní do vyšších binárnych rádov (napr. susedné celé čísla 7 – 0111 a 8 – 1000 majú Hammingovu vzdialenosť 4). V Grayovom kóde majú všetky susedné čísla Hammingovu vzdialenosť 1.

- **preformulovanie účelovej funkcie** tak, aby jedince s blízkou hodnotou účelovej funkcie nemali veľkú Hammingovu vzdialenosť. To je však možné len zriedkakedy.

- **rozšírenie operátora mutácie o vhodnú operáciu**, prekonávajúcu Hammingovu bariéru. V prípade našich „regulárnych“ grafov je takouto operáciou inverzia celého (úseku) chromozómu. Ukázalo sa však, že jeho prínos je minimálny, a pre hľadanie MISP na stochastických grafoch je jeho zavedenie až kontraproduktívne.

4.2 Experimenty

4.2.1 Experiment 1: Exemplárne riešenie

Parametre: - 3.4

- výpočet F: rank

Vstup: - MISP-S-100-40-45 (stochastický, 100 vrcholov, hustota 40%, vnútená nezávislá mn. mohutnosti 45)

Výsledky: - Grafy 1,2

Ako prvý krok sme demonštrovali riešiteľnosť problému pomocou GA. Algoritmus v skúmanom prípade našiel v priebehu 400 generácií nezávislú množinu mohutnosti 40 vrcholov.

Graf 1 ukazuje priebeh 1-epochového a totálneho minima počas behu evolúcie. Na detailnom grafe je vidno charakteristický skokovitý priebeh evolúcie. Keďže v algoritme nie je zavedený elitizmus, nie je monotónny

(teda sa môže stať, že algoritmus mutáciou/križením „objaví“ jedinca, ktorý je dovedy najlepším minimom, avšak pri výbere do nasledujúcej generácie jedinec nie je stochastickým operátorom selekcie zvolený – jeho „kvalitný“ genotyp zaniká).

V grafe 2 sú zobrazené priebehy zložiek *pravdepodobnostného vektora* w (j-ta zložka predstavuje pravdepodobnosť, že j-ty bit chromozómu v populácii je jednotka). V populácii $P = \{\alpha_1, \alpha_2, \dots, \alpha_\mu\}$ rátame hodnoty zložiek vektora nasledovne:

$$w_j = \frac{\sum_{i=1}^{\mu} \alpha_i^{(j)}}{\mu}$$

V grafe je zároveň hrubou čiarou zobrazený priebeh *koefficientu usporiadania* chromozómov $\chi(w)$:

$$\chi(w) = \frac{4}{n} \sum_{i=1}^n (w_j - 0.5)^2$$

Koefficient nadobúda hodnoty z intervalu $[0, 1]$, pričom pre náhodne vygenerované chromozómy sa jeho hodnota blíži k nule a naopak pre úplne usporiadané chromozómy (na každej pozícii majú buď všetky nulu alebo všetky jednotku) je jeho hodnota 1.

Z uvedených priebehov sa dá usudzovať o konvergentnosti a nepriamo aj o kvalite parametrov, funkcie sily, kódovania v GA. Ak totiž algoritmus konverguje príliš rýchlo a k veľkému usporiadaniu, je veľká pravdepodobnosť, že sa zastaví v neoptimálnom lokálnom minime. Naopak – dlhodobo príliš nízke hodnoty koefficientu usporiadania (< 0.5) môžu naznačovať nevhodné kódovanie (Hammingove bariéry), nevhodný tvar účelovej funkcie, príliš veľkú pravdepodobnosť mutácie apod., čoho dôsledkom je opäť nízka kvalita riešení.

4.2.2 Experiment 2: Overenie výsledkov z [1]

Parametre: - 3.4

Vstup: - MISP-R-102, MISP-S-100-10-45, MISP-S-100-20-45, MISP-S-100-30-45, MISP-S-100-40-45, MISP-S-100-50-45

Výsledky: - Tab. 1

Pre porovnanie sme zopakovali experiment za podmienok uvedených v [1]. Išlo o hľadanie optima v 100 nezávislých behoch algoritmu so štandardnými parametrami ako sme ich uviedli v odstavci 3.4. Grafy použité na testovanie boli: „regulárny graf“ MISP-R-102 s globálnym optimom mohutnosti 52 a 5 stochastických grafov s počtom vrcholov 100, vnútenou nezávislou množinou vrcholov mohutnosti 45 a hustotami od 0.1 do 0.5.

Výsledky sú podstatne horšie ako výsledky prezentované autormi v [1]. Nimi získané priemery globálne najlepšej hodnoty účelovej funkcie (f_{avg}) boli pre jednotlivé testovacie grafy: 44.94, 37.39, 37.25, 42.38, 44.20, 44.72). Vzhľadom na nie celkom jasné definovanie použitých parametrov GA v [1] (výpočet sily, výmena generácií) je teda veľmi pravdepodobné, že autori použili odlišné parametre ako sme na základe textu ich práce predpokladali.

4.2.3 Experiment 3: Výpočet sily rankovaním

Parametre: - 3.4

- výpočet F: rank

Vstup: - MISP-R-102, MISP-S-100-10-45, MISP-S-100-20-45, MISP-S-100-30-45, MISP-S-100-40-45, MISP-S-100-50-45

Výsledky: - Tab. 2

Prvý krok v snahe zlepšiť kvalitu riešení bolo nahradenie výpočtu F dynamickým linearizovaním metódou rankovania jedincov, nakoľko aj z riešenia iných problémov pomocou GA sme mali s touto metódou dobré skúsenosti. Táto metóda ohodnotí jedince silou výhradne podľa poradia v usporiadaní daným účelovou funkciou – degraduje teda účelovú funkciu z funkcie „ohodnocujúcej jedince“ na funkciu „usporiadávajúcu jedince“. Rankovanie teda ohodnocuje jedince rovnomerne bez ohľadu na rovnomernosť rozdelenia hodnôt účelovej funkcie.

Výsledky potvrdzujú očakávanie – priemerná veľkosť nezávislej množiny vzrástla pri všetkých grafoch o vyše 10 vrcholov.

4.2.4 Experiment 4: Výpočet sily rankovaním, elitizmus

- Parametre: - 3.4
- výpočet F: rank
- koeficient elitizmu: $1/\mu$ (1 najlepší jedinec nútene prenášaný do ďalšej generácie)
- Vstup: - MISP-R-102, MISP-S-100-10-45, MISP-S-100-20-45, MISP-S-100-30-45, MISP-S-100-40-45, MISP-S-100-50-45
- Výsledky: - Tab. 3

Elitizmus zabezpečí monotónnosť GA (monotónnosť aktuálneho minima vzhľadom na epochy) explicitným prenosom (aspoň 1) najlepšieho jedinca do dcérskej populácie. To môže byť na jednej strane prínosom – algoritmus stochasticky neopúšťa nájdené minimá (bez elitizmu sa stáva, že algoritmus mutáciou/križením „objaví“ jedinca, ktorý je lepším minimom, avšak pri výbere do nasledujúcej generácie jedinec nie je stochastickým operátorom selekcie zvolený – jeho „kvalitný“ genotyp zaniká). Na druhej strane však elitizmus znižuje pestrosť genotypu sepropagovaním kódu konkrétnych jedincov.

Z výsledkov vidno, že priemerná hodnota globálneho minima sa príliš nezmenila, zmenila sa však distribúcia hodnôt miním – zvýšil sa počet „lepších“ miním. V prípade „regulárneho“ grafu je väčšina hodnôt miním zoskupená do dvoch tried – nezávislých množín s mohutnosťou 44 a 46 vrcholov. Pri stochastických grafoch elitizmus výrazne pomohol častejšiemu nachádzaniu nezávislej množiny s veľkosťou vnútornej množiny.

4.2.5 Experiment 5: Výpočet sily rankovaním, elitizmus, uniformný crossover

- Parametre: - 3.4
- výpočet F: rank
- koeficient elitizmu: $1/\mu$ (1 najlepší jedinec nútene prenášaný do ďalšej generácie)
- $O_{\text{crossover}}$: uniformné križenie $2 \rightarrow 2$ (“po bitoch” s pravdepodobnosťou výmeny bitov 0.5)
- Vstup: - MISP-R-102, MISP-S-100-10-45, MISP-S-100-20-45, MISP-S-100-30-45, MISP-S-100-40-45, MISP-S-100-50-45
- Výsledky: - Tab. 4

Uniformný crossover sa svojou povahou podobá veľkej mutácii s tým rozdielom, že zhodné alely križených chromozómov ponechá nezmenené. V literatúre sa zväčša spomína ako vhodná metóda na preklopenie lokálnych miním. V našom prípade priniesol isté zvýšenie kvality nachádzaných riešení pri stochastických grafoch, pri „regulárnom“ grafe jeho vplyv nie je preukazný.

4.2.6 Experiment 6: Hľadanie riešenia na grafoch s počtom vrcholov 200

- Parametre: - 3.4
- výpočet F: rank
- Vstup: - MISP-R-202, MISP-S-200-10-90, MISP-S-200-20-90, MISP-S-200-30-90, MISP-S-200-40-90, MISP-S-200-50-90
- Štatistické: - počet opakovaní pre 1 vstup: 100
- Výsledky: - Avg Min $f(x)$: -83.26, -63.79, -71.61, -77.81, -78.71, -79.12
- Min Min $f(x)$: -92, -79, -86, -85, -86, -86

Tento experiment sme zaradili len pre letmé overenie závislosti relatívnej kvality riešenia od počtu vrcholov grafu. Nájdené riešenia sú podľa očakávania – približne dvojnásobnej mohutnosti oproti riešeniam na grafoch s počtom vrcholov 100.

4.2.7 Experiment 7: Štatistika – závislosť riešenia od parametra P_{mutation}

- Parametre: - 3.4
- výpočet F: rank
- podmienka ukončenia: 400 generácií bez zlepšenia globálneho minima
- Štatistické: - množina hodnôt parametra: {0, 0.0005, 0.001, 0.005, 0.01, 0.02, 0.03, 0.04, 0.05, 0.1, 0.2, 0.5}
- počet opakovaní pre 1 hodnotu parametra: 100
- Vstup: - MISP-R-102
- Výsledky: - Grafy 3,4

Z výsledkov vidno malú citlivosť GA-MISP na hodnotu pravdepodobnosti mutácie v okolí štandardných hodnôt ($1/n$, n – šírka chromozómu). Algoritmus je úspešný aj pri najmensej testovanej hodnote – 0.0005. Interval najoptimálnejších hodnôt bol v tomto prípade [0.001, 0.005]. Pri hodnotách nad 0.02 úspešnosť algoritmu prudko klesá.

4.2.8 Experiment 8: Štatistika – závislosť riešenia od parametra μ (veľkosť populácie)

- Parametre: - 3.4
- výpočet F: rank
- podmienka ukončenia: 400 generácií bez zlepšenia globálneho minima
- Štatistické: - množina hodnôt parametra: {6, 10, 20, 30, 50, 70, 100, 200, 500}
- počet opakovaní pre 1 hodnotu parametra: 100
- Vstup: - MISP-R-102
- Výsledky: - Grafy 5,6

Úspešnosť GA s veľkosťou populácie rastie podľa očakávania. Nárast je silno nelineárny a od veľkosti 200 jedincov už úspešnosť viac nerastie. Treba podotknúť, že s veľkosťou μ rastie výpočtový čas a pamäťová náročnosť GA priamo úmerne.

4.2.9 Experiment 9: Štatistika – závislosť riešenia od parametra $P_{\text{crossover}}$

- Parametre: - 3.4
- výpočet F: rank
- $O_{\text{reproduction}}$: mutácia sa aplikuje na všetky jedince, vybrané do nasledovnej generácie. Zmena $P_{\text{crossover}}$ sa potom realizuje zmenou $P_{\text{reproduction}}$.
- Štatistické: - podmienka ukončenia: 400 generácií bez zlepšenia globálneho minima
- množina hodnôt parametra: {0, 0.05, 0.1, 0.2, 0.4, 0.6, 1}
- počet opakovaní pre 1 hodnotu parametra: 100
- Vstup: - MISP-R-102
- Výsledky: - Grafy 7,8

Úspešnosť GA mieme rastie s veľkosťou hodnoty pravdepodobnosti kríženia, bez vplyvu na počet generácií, potrebných pre nájdenie stabilného minima. Riešenia nájdené s optimálnym krížením sú oproti behu bez kríženia v najlepšom prípade lepšie o 5 vrcholov.

Vplyv kríženia je teda preukazný avšak slabý.

4.2.10 Experiment 10: Štatistika – závislosť riešenia od parametra $P_{\text{mutation_inversion}}$

- Parametre: - 3.4
- výpočet F: rank
- O_{mutation} : navyše k štandardnej mutácii „po bitoch“ sa s pravdepodobnosťou $P_{\text{mutation_inversion}}$ aplikuje 2-bodová inverzia náhodne zvoleného úseku chromozómu
- Štatistické: - podmienka ukončenia: 400 generácií bez zlepšenia globálneho minima
- množina hodnôt parametra: {0, 0.01, 0.05, 0.1, 0.2, 0.4, 0.5}
- počet opakovaní pre 1 hodnotu parametra: 100
- Vstup: - MISP-R-102
- Výsledky: - Grafy 9,10

K tomuto experimentu nás viedla myšlienka prekonávania Hammingovej bariéry pri prechode k doplnkovej nezávislej množine vrcholov „regulárnych“ grafov MISP-R-n (pozri odstavce 4.1.3) pomocou inverzie (úseku) chromozómu.

Metóda sa však ukázala ako kontraproduktívna – úspešnosť GA nielen že nezlepšila, naopak zhoršila ju.

4.2.11 Experiment 11: Štatistika – závislosť riešenia od parametra ElitismCoefficient

- Parametre:
- 3.4
 - výpočet F: rank
 - podmienka ukončenia: 400 generácií bez zlepšenia globálneho minima
 - $O_{\text{replacement}}$: nová generácia sa najprv naplní (ElitismCoefficient * μ) najlepšími jedincami z predošlej a potom sa v naplňaní pokračuje štandardne
- Štatistické:
- množina hodnôt parametra: {0, 0.01, 0.05, 0.1, 0.2, 0.4, 0.6}
 - počet opakovaní pre 1 hodnotu parametra: 100
- Vstup:
- MISP-R-102
- Výsledky:
- Grafy 11, 12

Analýzou prínosov a negatív elitizmu sme sa zaoberali v popise experimentu 4 (odstavce 4.2.4).

Týmto experimentom len ukazujeme, že v našom prípade nemá veľký význam prenášať do dcérskej populácie viac ako 1 chromozóm, pretože to nemá preukazný vplyv na kvalitu riešenia. Zväčšovanie parametra má pozitívny vplyv na počet generácií, potrebných pre nájdenie stabilného riešenia – najväčšie zlepšenie však rovnako nastáva pri prechode od behu bez elitizmu k behu s prenosom 1 chromozómu.

ZÁVER

Výsledky experimentov ukazujú dobrú aproximatívnu riešiteľnosť NP-úplného problému hľadania maximálnej nezávislej množiny vrcholov grafu pomocou genetického algoritmu. V práci ponúkame aj vhodné nastavenie parametrov GA pre čo najoptimálnejšie riešenie problému.

Práca môže zároveň poslúžiť aj ako inšpirácia pre aplikáciu genetických algoritmov pri riešení iných grafových problémov. Priamo, bezo zmeny možno popísanú heuristiku GA-MISP použiť na aproximatívne riešenie problému hľadania maximálnej kliky a problému hľadania minimálneho vrcholového pokrytia grafu (Veta 1).

POUŽITÁ LITERATÚRA

- [1] Thomas Bäck, Sami Khuri, *An Evolutionary Heuristic for the Maximum Independent Set Problem*, University of Dortmund, Germany (http://ls11-www.informatik.uni-dortmund.de/people/baeck/ea_application.html)
- [2] Vladimír Kvasnička, *Evolučné algoritmy*, texty prednášok na MFF UK, Bratislava, 1999
- [3] Martin Pelikán, *Genetic Algorithms, Marginal Distribution Recombination, Schemata and Quantitative Genetics*, diplomová práca, Bratislava, 1997
- [4] ???, *Graph Theory*, texty prednášok, 1996

Príloha A - TABUĽKY, GRAFY

misp-r-102		misp-s-100-10-45		misp-s-100-20-45		misp-s-100-30-45		misp-s-100-40-45		misp-s-100-50-45	
f(x)	N	f(x)	N	f(x)	N	f(x)	N	f(x)	N	f(x)	N
-52	---	-45	---	-45	---	-45	---	-45	---	-45	---
-37	3	-30	2	-33	1	-30	2	-32	1	-32	2
-36	1	-29	5	-30	2	-29	5	-31	2	-31	3
-35	3	-28	3	-29	2	-28	2	-30	4	-30	1
-34	6	-27	13	-28	5	-27	13	-29	9	-29	9
-33	11	-26	11	-27	1	-26	14	-28	6	-28	9
-32	19	-25	22	-26	11	-25	8	-27	11	-27	17
-31	22	-24	16	-25	15	-24	7	-26	14	-26	19
-30	13	-23	12	-24	15	-23	14	-25	10	-25	14
-29	15	-22	5	-23	9	-22	7	-24	15	-24	11
< 29	7	-21	8	-22	14	-21	2	-23	8	-23	6
		-20	3	-21	10	< -21	26	-22	5	-22	2
				< -21	15			-21	5	-21	2
								< -21	10	< -21	5
$f_{avg} = -31.28$		$f_{avg} = -24.74$		$f_{avg} = -23.45$		$f_{avg} = -23.3$		$f_{avg} = -24.96$		$f_{avg} = -25.9$	

Tab. 1: (Experiment 2: Overenie výsledkov z [1]) Distribúcia nájdených miním účelovej funkcie v jednotlivých behoch a priemerná hodnota pre jednotlivé grafy a 100 behov GA.

misp-r-102		misp-s-100-10-45		misp-s-100-20-45		misp-s-100-30-45		misp-s-100-40-45		misp-s-100-50-45	
f(x)	N	f(x)	N	f(x)	N	f(x)	N	f(x)	N	f(x)	N
-52	---	-45	---	-45	---	-45	---	-45	---	-45	1
-50	1	-44	1	-42	9	-44	1	-44	1	-44	2
-49	0	-41	3	-41	9	-43	7	-43	7	-43	9
-48	3	-40	2	-40	11	-42	16	-42	23	-42	17
-47	2	-39	1	-39	5	-41	19	-41	16	-41	22
-46	8	-38	6	-38	14	-40	16	-40	25	-40	19
-45	19	-37	8	-37	2	-39	11	-39	9	-39	15
-44	14	-36	9	-36	5	-38	7	-38	5	-38	6
-43	20	-35	11	-35	5	-37	4	-37	6	-37	3
-42	14	-34	11	-34	7	-36	3	-36	3	-36	4
-41	10	-33	7	-33	7	-35	3	-35	1	-35	1
-40	5	-32	14	-32	5	-34	2	-34	1	< -35	1
-39	2	-31	12	-31	3	-33	5	-33	1		
-38	0	< -31	15	< -31	18	< -33	6	< -33	2		
-37	2										
$f_{avg} = -43.37$		$f_{avg} = -33.85$		$f_{avg} = -35.38$		$f_{avg} = -38.76$		$f_{avg} = -40.02$		$f_{avg} = -40.3$	

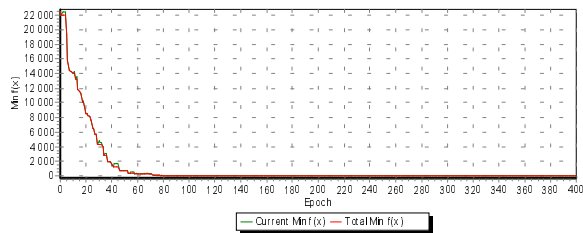
Tab. 2: (Experiment 3: Výpočet sily rankovaním) Distribúcia nájdených miním účelovej funkcie v jednotlivých behoch a priemerná hodnota pre jednotlivé grafy a 100 behov GA.

misp-r-102		misp-s-100-10-45		misp-s-100-20-45		misp-s-100-30-45		misp-s-100-40-45		misp-s-100-50-45	
f(x)	N	f(x)	N	f(x)	N	f(x)	N	f(x)	N	f(x)	N
-52	---	-45	---	-45	12	-45	48	-45	86	-45	96
-50	2	-42	1	-44	1	-44	0	-44	1	-44	0
-49	0	-41	1	-43	3	-43	1	-43	0	-43	0
-48	8	-40	3	-42	1	-42	3	-42	0	-42	0
-47	0	-39	3	-41	3	-41	1	-41	0	-41	0
-46	35	-38	9	-40	4	-40	0	-40	1	-40	0
-45	0	-37	5	-39	11	-39	2	-39	0	-39	0
-44	44	-36	11	-38	4	-38	1	-38	0	-38	1
-43	0	-35	10	-37	3	-37	4	-37	0	< -38	3
-42	8	-34	9	-36	2	-36	2	-36	2		
-41	0	-33	12	-35	6	-35	3	-35	0		
-40	3	-32	7	-34	4	-34	3	-34	0		
		-31	9	-33	3	-33	2	-33	0		
		< -31	20	< -33	43	< -33	30	< -33	10		
$f_{avg} = -44.86$		$f_{avg} = -33.76$		$f_{avg} = -33.99$		$f_{avg} = -37.03$		$f_{avg} = -42.42$		$f_{avg} = -44.3$	

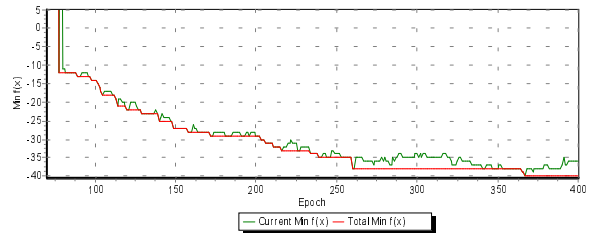
Tab. 3: (Experiment 4: Výpočet sily rankovaním, elitizmus) Distribúcia nájdených minim účelovej funkcie v jednotlivých behoch a priemerná hodnota pre jednotlivé grafy a 100 behov GA.

misp-r-102		misp-s-100-10-45		misp-s-100-20-45		misp-s-100-30-45		misp-s-100-40-45		misp-s-100-50-45	
f(x)	N	f(x)	N	f(x)	N	f(x)	N	f(x)	N	f(x)	N
-52	---	-45	2	-45	29	-45	85	-45	91	-45	99
-50	2	-44	0	-44	1	-44	2	-44	2	-44	0
-49	0	-43	3	-43	1	-43	0	-43	0	-43	0
-48	9	-42	2	-42	7	-42	0	-42	0	-42	0
-47	0	-41	3	-41	3	-41	1	-41	0	-41	0
-46	35	-40	2	-40	2	-40	0	-40	0	-40	0
-45	0	-39	6	-39	7	-39	2	-39	1	-39	0
-44	38	-38	6	-38	2	-38	2	-38	0	-38	0
-43	0	-37	11	-37	4	-37	0	-37	0	-37	0
-42	15	-36	13	-36	5	-36	2	-36	0	-36	0
-41	0	-35	7	-35	2	-35	1	-35	1	-35	0
-40	1	-34	6	-34	5	-34	1	-34	2	-34	0
		-33	10	-33	6	-33	1	-33	1	-33	0
		< -33	29	< -33	26	< -33	3	< -33	2	< -33	1
$f_{avg} = -44.84$		$f_{avg} = -35.09$		$f_{avg} = -36.87$		$f_{avg} = -43.45$		$f_{avg} = -44.07$		$f_{avg} = -44.7$	

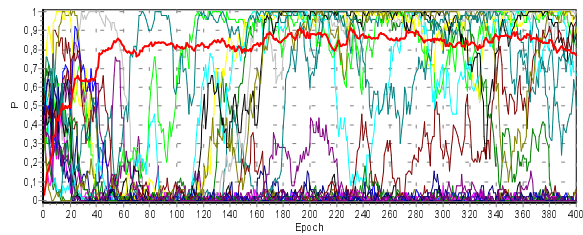
Tab. 4: (Experiment 5: Výpočet sily rankovaním, elitizmus, uniformný crossover) Distribúcia nájdených minim účelovej funkcie v jednotlivých behoch a priemerná hodnota pre jednotlivé grafy a 100 behov GA.



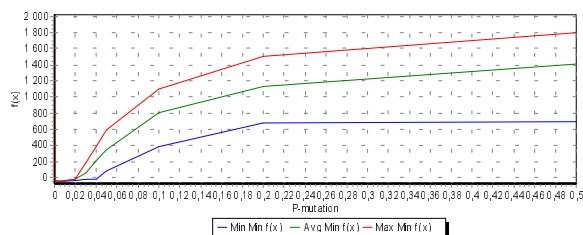
Graf 1a: Exemplárne riešenie – priebeh minima počas behu GA



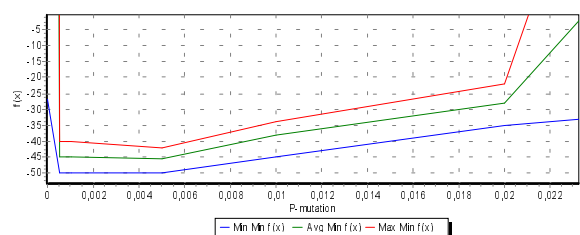
Graf 1b: Detail evolúcie korektnej nezávislej množiny.



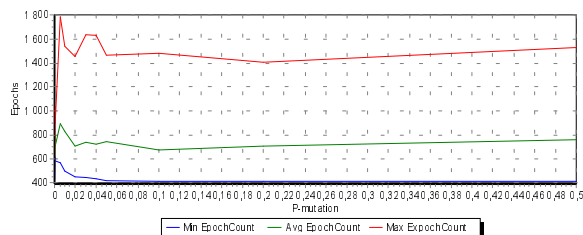
Graf 2: Exemplárne riešenie – priebeh zmien pravdepodobnostného vektora pre 20 náhodne zvolených bitov chromozómu.
Hrubá červená krivka je priebeh koeficientu usporiadosti chromozómov (cez všetky bity).



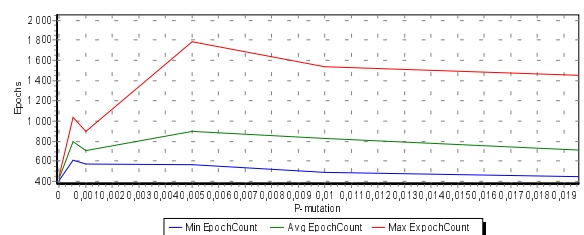
Graf 3a: Experiment 7: Štatistika – závislosť riešenia od parametra Pmutation
Závislosť Min Min $f(x)$, Avg Min $f(x)$, Max Min $f(x)$ za 100 behov GA pre každú hodnotu parametra štatistiky.



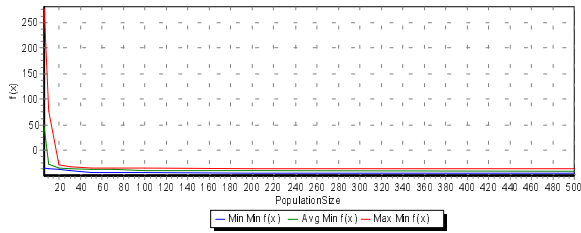
Graf 3b: Detail závislosti z rozsahu hodnôt parametra, vedúcich k nachádzaniu najoptimálnejších riešení.



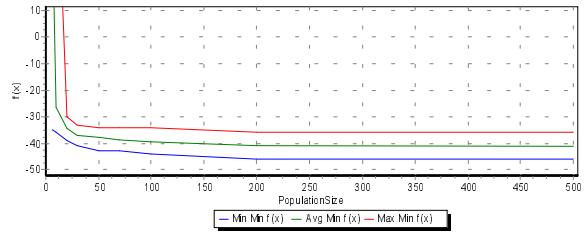
Graf 4a: Experiment 7: Štatistika – závislosť riešenia od parametra Pmutation
Závislosť počtu generácií, potrebných k nájdeniu stabilného minima od hodnoty parametra štatistiky.



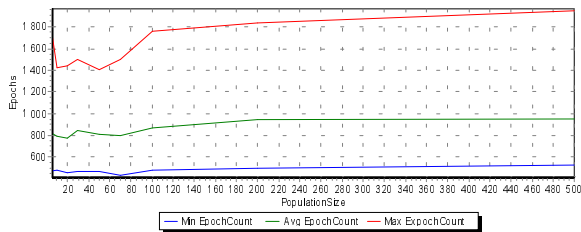
Graf 4b: Detail závislosti z rozsahu hodnôt parametra, vedúcich k nachádzaniu najoptimálnejších riešení.



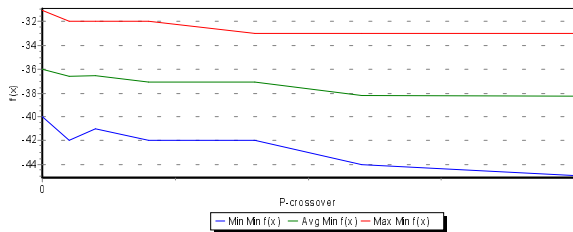
Graf 5a: Experiment 8: Štatistika – závislosť riešenia od parametra μ (veľkosť populácie)
Závislosť Min Min $f(x)$, Avg Min $f(x)$, Max Min $f(x)$ za 100 behov GA pre každú hodnotu parametra štatistiky.



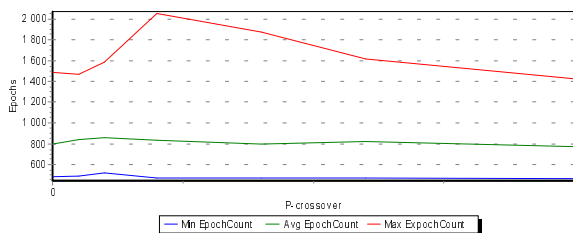
Graf 5b: Detail závislosti z rozsahu hodnôt parametra, vedúcich k nachádzaniu najoptimálnejších riešení



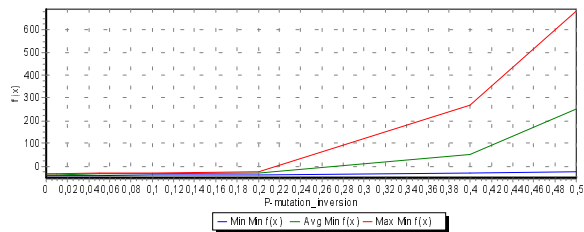
Graf 6: Experiment 8: Štatistika – závislosť riešenia od parametra μ (veľkosť populácie)
Závislosť počtu generácií, potrebných k nájdeniu stabilného minima od hodnoty parametra štatistiky.



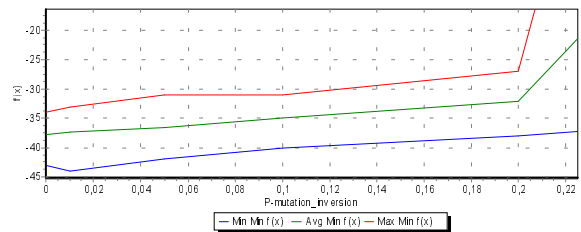
Graf 7: Experiment 9: Štatistika – závislosť riešenia od parametra Perrossover
Závislosť Min Min $f(x)$, Avg Min $f(x)$, Max Min $f(x)$ za 100 behov GA pre každú hodnotu parametra štatistiky.



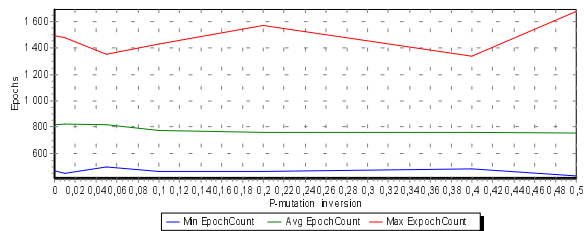
Graf 8: Experiment 9: Štatistika – závislosť riešenia od parametra Perrossover
Závislosť počtu generácií, potrebných k nájdeniu stabilného minima od hodnoty parametra štatistiky.



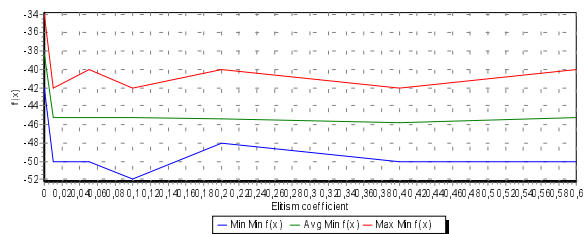
Graf 9a: Experiment 10: Štatistika – závislosť riešenia od parametra P.mutation_inversion
 Závislosť Min Min $f(x)$, Avg Min $f(x)$, Max Min $f(x)$ za 100 behov GA pre každú hodnotu parametra štatistiky.



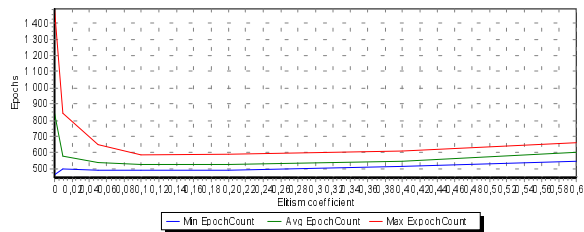
Graf 9b: Detail závislosti z rozsahu hodnôt parametra, vedúcich k nachádzaniu najoptimálnejších riešení



Graf 10: Experiment 10: Štatistika – závislosť riešenia od parametra P.mutation_inversion
 Závislosť počtu generácií, potrebných k nájdeniu stabilného minima od hodnoty parametra štatistiky.



Graf 11: Experiment 11: Štatistika – závislosť riešenia od parametra ElitismCoefficient
 Závislosť Min Min $f(x)$, Avg Min $f(x)$, Max Min $f(x)$ za 100 behov GA pre každú hodnotu parametra štatistiky.



Graf 12: Experiment 11: Štatistika – závislosť riešenia od parametra ElitismCoefficient
 Závislosť počtu generácií, potrebných k nájdeniu stabilného minima od hodnoty parametra štatistiky.

Príloha B - GENETICKÝ ALGORITMUS

Ospravedlňujeme sa prípadným čitateľom ktorí sa neorientujú v teórii GA za nespracovanie tejto pasáže. Záujemcom o problematiku odporúčame napr. [2].

Na tomto mieste vkladáme len veľmi stručný a zjednodušujúci popis GA pre čitateľa, ktorý sa s pojmom stretáva „po prvý raz“:

Genetický algoritmus je stochastický evolučný algoritmus, založený na metafore Darwinovskej evolúcie. Algoritmus si udržuje v kolekcii populáciu binárnych reťazcov – chromozómov, ktoré sú genotypom jedincov populácie. Nad chromozómami sú zavedené operácie mutácie a kríženia. Kvalitu jedinca – chromozómu ohodnocuje funkcia sily (fitness function, značí sa zväčša $F(x)$), ktorej funkčná hodnota je v najjednoduchšom prípade fenotypom jedinca. Funkcia zobrazuje kód chromozómu na reálne číslo – silu jedinca.

Algoritmus „simuluje“ prírodnú evolúciu – z aktuálnej populácie (inicializačná je vygenerovaná náhodne) vyberá „kvalitných“ jedincov – chromozómy (tie s veľkou hodnotou funkcie sily), reprodukuje ich (kríženie, mutácia) a vkladá do novej populácie.

Pokiaľ chromozómy „niečo“ vhodne reprezentujú, uvedený proces vedie k evolúcii kvalitných jedincov v zmysle hodnôt funkcie sily. To „niečo“ môže byť prakticky čokoľvek, zopár typických príkladov: argument triviálnej numerickej funkcie; reprezentácia mechanickej konštrukcie; kód zložitého agenta, žijúceho a konajúceho v nejakom prostredí.

Kľúčová vec, ktorá určuje predmet evolúcie je teda funkcia sily. Cez ňu sa do algoritmu „vkladá pointa“ evolúcie.

V prípade nášho prvého príkladu – ak funkcia sily bude priamo vracat' hodnoty numerickej funkcie nad reálnymi číslami a chromozómy budú reprezentovat' jej argument (reálne číslo zakódované do binárneho reťazca), GA bude hľadat' argument, pri ktorom nadobúda funkcia globálne maximum.

Pokiaľ by chromozómy reprezentovali genotypy agentov, funkcia sily by bola zložitejšia – pred každým krokom k dcérskej generácii GA by sa spúšťala simulácia života agentov v umelom prostredí a funkcia sily by ohodnocovala úspešnosť jednotlivých agentov pri ich umelom živote. Predmetom evolúcie teda bude genotyp čo najúspešnejšieho agenta (v danom prostredí).

Formálne takto postavený GA rieši optimalizačný problém: ($\alpha \in \{0,1\}^k$ je chromozóm šírky k)

$$\alpha_{opt} = \arg \max_{\alpha \in \{0,1\}^k} F(\alpha)$$

Priestor chromozómov $\{0,1\}^k$ je hyperkocka s 2^k vrcholmi, takže slepé prehľadávanie priestoru je možné len pre veľmi malé k . GA ponúka efektívny a abstraktný (nezávislý od konkrétneho problému) prístup k prehľadávaniu uvedeného priestoru.

V praxi sa pri riešení numerickej a iných úloh pomocou GA vkladá ešte jedna úroveň ohodnocovania pred funkciu sily – účelová funkcia (značí sa zväčša $f(x)$), do nej sa potom sústreďuje doménovo závislé ohodnocovanie a funkcia sily už len vhodne transformuje jej hodnotu spĺňajúc vzťah:

$$\forall \alpha_1, \alpha_2 \in \{0,1\}^k : f(\alpha_1) \leq f(\alpha_2) \Rightarrow F(\alpha_1) \geq F(\alpha_2)$$

GA teda maximalizuje $F(x)$ a tým minimalizuje $f(x)$. Optimalizačný problém sa teda dá preformulovat' na hľadanie:

$$\alpha_{opt} = \arg \min_{\alpha \in \{0,1\}^k} f(\alpha)$$

Popis najvýznamnejších operátorov a parametrov GA:

- O_{mutation} : stochastický operátor mutácie. Realizuje mutáciu chromozómu: $\alpha' = O_{\text{mutation}}(\alpha)$.
- $O_{\text{crossover}}$: stochastický operátor kríženia. Realizuje kríženie chromozómov pri pohlavnej reprodukcii: $(\alpha', \beta') = O_{\text{crossover}}(\alpha, \beta)$.
- $O_{\text{selection}}$: stochastický operátor selekcie jedincov. Realizuje výber 2 jedincov z populácie – kvázináhodne, proporcionálne k hodnote ich sily F .
- $O_{\text{reproduction}}$: operátor reprodukcie jedincov. Štandardne: $O_{\text{crossover}}(\alpha_{\text{parent1}}, \alpha_{\text{parent2}}, \alpha_{\text{child1}}, \alpha_{\text{child2}})$; $O_{\text{mutation}}(\alpha_{\text{child1}})$; $O_{\text{mutation}}(\alpha_{\text{child2}})$
- $O_{\text{replacement}}$: operátor nahrádzania starých jedincov v populácii novými (reprodukovanými). Bežný prístup je nasledovný: dcérska populácia je vytváraná nanovo z rodičovskej naplnením jedincami (proporcionálne k sile) zvolenými operátorom $O_{\text{selection}}$ reprodukovanými s pravdepodobnosťou $P_{\text{reproduction}}$ operátorom $O_{\text{reproduction}}$. Jedince ktoré neboli zvolené pre reprodukciu sú do dcérskej populácie vkladané bez modifikácie. Proces končí, keď mohutnosť dcérskej populácie dosiahne mohutnosť rodičovskej.
- Výpočet F : metóda transformácie hodnôt doménovo závislej účelovej funkcie na hodnoty funkcie sily.
- $P_{\text{reproduction}}$: pravdepodobnosť reprodukcie zvoleného jedinca. Štandardne sa používa hodnota 0.6.
- P_{mutation} : pravdepodobnosť mutácie bitu chromozómu. Štandardne sa používa hodnota $1/n$ (n = šírka chromozómu)

```

procedure GA(input: PopulationSize, P-reproduction, P-mutation; output: BestChromosome);
begin
  StopConditionSatisfied := false;
  EpochCount := 0;
  PCurrent.GeneratePopulationRandomely(PopulationSize);
  repeat
    EpochCount := EpochCount + 1;
    for each Individual of CurrentPopulation do
      Individual.EvaluateFitness();
    CurrentPopulation.RenormalizeFitnesses();
    NewPopulation := [];
    while NewPopulation.ChromosomeCount < CurrentPopulation.ChromosomeCount do
      begin
        Individual1 := CurrentPopulation.GetIndividualByRouletteWheel();
        Individual2 := CurrentPopulation.GetIndividualByRouletteWheel();
        if Random < P-reproduction then
          begin
            CrossOver(Individual1, Individual2, NewIndividual1, NewIndividual2);
            if Random < P-mutation then
              NewIndividual1.Mutation();
            if Random < P-mutation then
              NewIndividual2.Mutation();
          end
        else
          begin
            NewIndividual1 := Individual1;
            NewIndividual2 := Individual2;
          end;
        NewPopulation.Add(Individual1);
        NewPopulation.Add(Individual2);
      end;
    CurrentPopulation := NewPopulation;
    EvaluateStopCondition();
  until StopConditionSatisfied;
  BestChromosome := PCurrent.GetBestChromosome();
end;

```

Pseudokód 1: Štandardná implementácia genetického algoritmu.

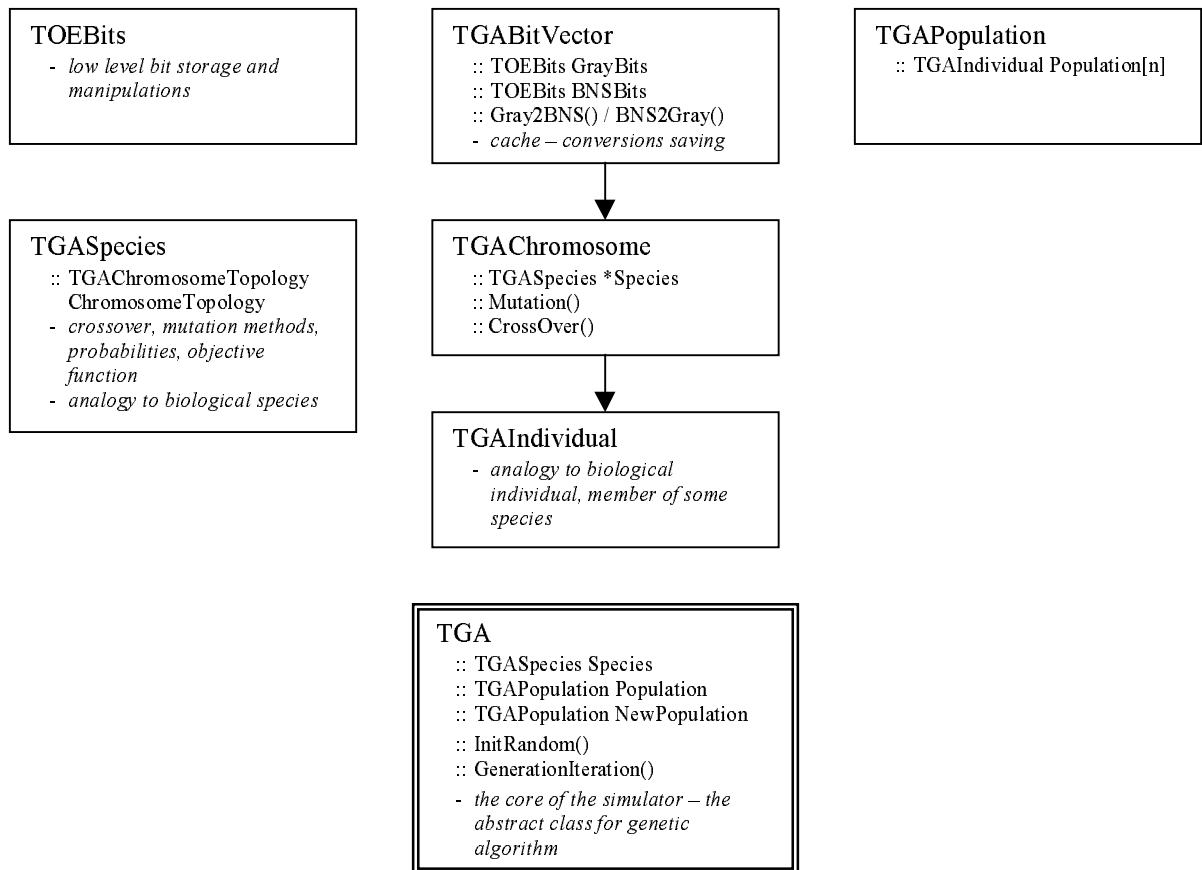
Príloha C - SIMULÁTOR

Na účely testov sme navrhli jednoduchý systém na výpočet genetických algoritmov. Systém je plne objektovo orientovaný a jeho hierarchia tried je vybudovaná od abstraktných "základných kameňov" až po implementáciu konkrétnych algoritmov (Obr. 1). Systém sme implementovali v jazyku Object Pascal s použitím grafickej knižnice Borland VCL.

Ako nutný základ systém umožňuje výpočet podľa zaužívaného konceptu genetického algoritmu (Pseudokód 1), ako ho nájdeme napr. v [2].

Popis vlastností simulátora:

- voľne definovateľná topológia chromozómu (rozdelenie na podreťazce bitov – každý sa môže interpretovať zvlášť – viacrozmerná účelová funkcia nad reťazcami)
- mutácie – viacero metód kvôli prekonávaniu Hammingovej bariéry (*s možnosťou kombinovať metódy v rámci jedného behu GA*):
 - štandardné invertovanie náhodne zvolených bitov vektora komponentu chromozómu v binárnej číselnej sústave (ďalej len BNS - Binary Number System)
 - invertovanie náhodne zvolených bitov vektora v Grayovom kóde (pričom účelová funkcia dostáva na vstup vždy BNS vektor – túto mutáciu si teda možno predstaviť ako konverziu z BNS do Grayovho kódu, mutáciu a následnú konverziu späť)
 - inverzia náhodne zvoleného úseku BNS vektora
 - posun náhodne zvoleného úseku BNS vektora o 1 bit doľava/doprava
 - ľubovoľná proprietárna metóda potomka triedy TGA
- kríženie (nad BNS vektorom, Grayovým vektorom alebo oboma s pravdepodobnosťou výberu reprezentácie 1/2):
 - uniformné
 - 1-bodové alebo 2-bodové, $2 \rightarrow 2$ (z dvoch jedincov vzniknú operáciou kríženia dva jedince)
- výpočet sily (fitness):
 - dynamické afinné škálovanie intervalu [f-min, f-max] minima/maxima účelovej funkcie nad aktuálnou populáciou na interval (0, 1] (podľa [2] (4.11c))
 - rankovaním jedincov podľa hodnoty účelovej funkcie
 - ľubovoľná proprietárna metóda potomka triedy TGA
- tvorba novej generácie:
 - ako v Pseudokóde 1 – renormalizáciou hodnôt sily a následným výberom Goldbergovou ruletou s pravdepodobnosťou výberu jedinca úmernou sile
 - navyše je možné *označiť si podmnožinu jedincov v populácii a vykonať nad nimi generačný cyklus nezávisle od ostatných* (počnúc rátaním hodnôt sily). Túto operáciu možno pri tvorbe novej populácie opakovať. Takto je možné širšie experimentovať s GA (elitizmus, propagácia jedincov, „objavujúcich“ nové minimá...)
- grafické užívateľské rozhranie (GUI) s tvorbou dôležitých on-line grafov:
 - priebeh účelovej funkcie
 - X-Y rozloženie populácie na „povrchu“ účelovej funkcie
 - distribúcia populácie - počet jedincov na podinterval argumentu účelovej funkcie
(Na účelovú funkciu sa pri hore uvedených grafoch nazerá ako na funkciu jednej premennej – celého čísla v binárnej sústave. Je zrejmé, že pokiaľ funkcia nemá takýto charakter - a takých je pri úlohách riešených GA väčšina - majú uvedené grafy len malú výpovednú hodnotu typu „vidno, že nám to hľadá nejaké minimá“...)
 - distribúcia populácie - počet jedincov na podinterval funkčných hodnôt účelovej funkcie
 - priebeh pravdepodobnostného vektora w (pravdepodobnosť nastavenia bitu chromozómu podľa generácií pre každý bit) a koeficientu usporiadania $\chi(w)$
 - aktuálne/totálne $\text{Min } f(x)$ podľa epoch (existencia, rýchlosť a priebeh konvergencie)Všetku on-line aktivitu je možné pre urýchlenie výpočtu zakázať.
- implementácia rôznych podmienok ukončenia
- štatistický mód (viacnásobné spúšťanie algoritmu s rovnakými parametrami so zaznamenávaním údajov do relačnej databázy pre štatistické vyhodnotenia). Zobrazovanie distribúcie hodnôt a grafov zo získaných priebehov. Dôležité pre základné posúdenie preukaznosti úprav pri optimalizačných experimentoch s GA a tvorbe priebehov efektívnosti behu GA podľa zvoleného parametra.



Obr. – schematická hierarchia tried simulátora