

---

# Evolučné algoritmy

Prvky a princípy

---

Marián Mach

Košice

2009

doc. Ing. Marián Mach, CSc.  
Katedra kybernetiky a umelej inteligencie  
Fakulta elektrotechniky a informatiky  
Technická univerzita v Košiciach  
Marian.Mach@tuke.sk

Táto práca vznikla aj za podpory VEGA grantu MŠ SR č. 1/4074/07 "Metódy anotovania, vyhľadávania, tvorby a sprístupňovania znalostí s využitím metadát pre sémantický popis znalostí".

Edícia vedeckých spisov Fakulty elektrotechniky a informatiky TU Košice.

Lektorovali: *prof. Ing. Vladimír Kvasnička, DrSc., FIIT STU Bratislava*  
*doc. RNDr. Jozef Uličný, CSc., PF UPJŠ Košice*  
*Ing. Rudolf Jakša, Ph.D., FEI TU Košice*

© Marián Mach, Košice 2009

Žiadna časť tejto publikácie nesmie byť reprodukováaná, zadaná do informačného systému alebo prenášaná v inej forme či inými prostriedkami bez predchádzajúceho písomného súhlasu autora.

Všetky práva vyhradené.

ISBN 978-80-8086-123-0

*Ak chceme, môžeme sa bezcieľne potulovať po kontinente náhod. Podobne ako okrídlené semienko bez koreňa, ktoré unáša jarný vánok.*

*Pri tom všetkom zároveň môžeme poprieť, že nejaké náhody vôbec existujú.*

*Haruki Murakami*

*Evolúcia je pre nedokonalé veci. Tie dokonalé sú večné a nemeniteľné.*

*anonym*



# Predslov

Evolučné algoritmy sú jedným z výsledkov snahy využiť princípy, mechanizmy a vzory vynájdene a overené prírodou. Vytvárajú pomerne rozsiahlu triedu pravdepodobnostných populačných prehľadávacích algoritmov, vhodných pre riešenie optimalizačných úloh – a to najmä v prípadoch, keď nie je nutné vždy nájsť optimálne riešenie ale postačujúcim bude aj kvalitné suboptimálne riešenie, svojou kvalitou blízke optimálnemu. Vďaka ich nenáročnosti a jednoduchšej použiteľnosti ich popularita stále rastie a stále častejšie si nachádzajú cestu aj do oblasti riešenia praktických úloh.

V rozpore s pozornosťou, ktorá je týmto algoritmom venovaná v cudzojazyčnej odbornej literatúre, však záujemca iba sporadicky natrafi na publikáciu v slovenskom jazyku, venovanú tejto problematike, či už úplne alebo čiastočne. Cieľom predkladanej knihy je vyplniť túto medzeru a podať slovenským čitateľom informácie o tejto zaujímavej triede algoritmov.

Samotné evolučné algoritmy môžu mať mnoho rozličných podôb reprezentujúcich viac či menej podobné algoritmy, navzájom príbuzné vďaka princípom, na ktorých sú založené, ako aj vďaka využívaniu rovnakých algoritmickejých prvkov. Prezentovaný text tieto varianty chápe iba ako inštancie jednej zovšeobecnenej podoby evolučného algoritmu a predstavuje štruktúru tejto zovšeobecnenej podoby a prvky, ktorými je možné túto štruktúru vyplniť. Takýmto spôsobom sa chápanie evolučného algoritmu redukuje na využívanie dvoch princípov:

- princípu preferencie určujúceho smer a oblasť prehľadávania,
- princípu rôznorodosti umožňujúceho meniť smer a oblasť prehľadávania.

Keďže tieto dva princípy pôsobia proti sebe a prílišný dôraz na jeden z nich potláča princíp druhý, návrh evolučného algoritmu je chápaný ako hľadanie rovnováhy medzi nimi v pomere, vhodnom pre aplikáciu algoritmu pre riešenie konkrétnych úloh.

Pred použitím evolučného algoritmu pre riešenie nejakého konkrétneho problému je potrebné daný algoritmus vhodne zostaviť a nastaviť. Toto zahŕňa sériu rozhodnutí, nevyhnutných pre voľbu<sup>1</sup>:

- plochy vhodnosti, ktorú je potrebné prehľadávať (spôsob reprezentácie a výpočtu vhodnosti určia priestor prehľadávania a jeho vzťah k priestoru kandidátov),
- realizácie jednotlivých blokov algoritmu (výber metód pre realizáciu jednotlivých blokov stanoví spôsob vytvárania selekčného tlaku a prehľadávania plochy vhodnosti),
- parametrov algoritmu (hodnoty parametrov zvolených metód ovplyvnia celkovú výkonnosť algoritmu na danej ploche vhodnosti).

Prezentovaný text sa snaží vybaviť čitateľa zásobou znalostí o možnostiach, ktoré má k dispozícii pri tomto rozhodovaní, ako aj o možných následkoch či súvislostiach, ktoré jeho rozhodnutia môžu priniesť.

Kniha je v zásade rozdelená do šiestich častí. Prvá časť zavádza základné pojmy a predstavuje všeobecnú štruktúru evolučného algoritmu. Tieto pojmy a princípy, ukotvené v štruktúre algoritmu, sú demonštrované na ilustračnom príklade. Druhá časť zavádza pojem plochy vhodnosti a venuje sa spôsobu, ako konkrétny problém, ktorý je potrebné riešiť, ovplyvňuje tvar tejto plochy vhodnosti. Tretia časť je venovaná možnostiam, ako je možné realizovať jednotlivé bloky všeobecnej štruktúry algoritmu. Poskytuje katalóg alternatívnych metód, ktorými je možné naplniť štruktúru. Štvrtá časť je zameraná na prípravu zostaveného algoritmu pre jeho použitie pre riešenie konkrétneho problému a na udržiavanie takého prostredia počas realizácie algoritmu, ktoré umožňuje algoritmu účinne pokračovať v prehľadávaní. Piata časť sa snaží poskytnúť predstavu o procesoch, prebiehajúcich vo vnútri algoritmu počas riešenia úloh, pričom poskytuje ako teoretický pohľad tak aj možnosti pre získanie empirického pohľadu. Šiesta časť predstavuje známe varianty evolučného algoritmu, vystupujúce pod svojím vlastným menom, pričom ich popisuje v pojmoch a prvkoch všeobecného evolučného algoritmu. Prílohy dopĺňajú hlavný text a prezentujú informácie, užitočné pre orientáciu v ňom.

Aby čitateľ nezískal iba vymenovanie alternatívnych prvkov, ktoré je možné vložiť do štruktúry evolučného algoritmu, ale aj nejaké vodítko pre

---

<sup>1</sup>Nedajte sa odradiť zopár odbornými termínmi, ale berte ich ako výzvu k zahĺbeniu sa do predloženého textu. Naopak, po jeho prečítaní sa vráťte na toto miesto a vnímajte ho ako zhrnutie všetkého, čo sa vám kniha snažila prezradiť.

výber svojich favoritov, kniha ponúka v niektorých častiach aj informácie, ktoré môžu uľahčiť túto voľbu. Či už majú tvar ilustračného použitia jednotlivých alternatív so skúmaním ich vplyvu na proces prehľadávania, priame porovnanie alternatív na základe empirických experimentov alebo teoretického rozboru ich vlastností, umožnia lepšie pochopenie problematiky a uľahčia orientáciu v pomerne veľkom množstve prezentovaných prvkov.

Kniha sa nesnaží byť vyčerpávacím textom z predkladanej oblasti evolučných algoritmov a v žiadnom prípade si nerobí nárok na úplnosť. Jej cieľom je priblížiť iba základnú problematiku evolučných algoritmov – na akých princípoch je založený evolučný algoritmus a z akých prvkov je možné zostaviť konkrétnu podobu algoritmu tak, aby tieto základné princípy boli naplnené. Nekladie si za cieľ predstaviť najnovšie trendy v oblasti, podať prehľad o rôznych aplikáciách evolučných algoritmov, ani poskytnúť návod ako aplikovať evolučný algoritmus na riešenie rôznych typov úloh.

Je vhodná ako pre tých, ktorí by chceli získať základný prehľad a hľadajú informácie všeobecnejšieho charakteru, tak aj pre tých, ktorí majú hlbší záujem o prezentovanú problematiku a radi sa dozvedia detaily. Ak sa jej podarilo podnieť záujem čitateľov o túto problematiku, tak potom splnila svoj cieľ.

Vďaka organizácii knihy nie je nutné jej sekvenčné čítanie. Všetci čitatelia by si však mali prečítať prvú časť pre oboznámenie sa so základnými pojmami a štruktúrou algoritmu. Výber ďalších častí závisí na cieľoch čitateľa:

- Záujemca o pochopenie prvkov algoritmu by mal pokračovať druhou a treťou časťou a prípadne druhou kapitolou štvrtej časti. Ak by mal záujem aj o teoretické pozadie činnosti algoritmu, tak by mal pozornosť venovať aj prvej kapitole piatej časti.
- Záujemca o praktické použitie algoritmu by nemal vynechať štvrtú časť a druhú kapitolu piatej časti.
- Záujemca o nejaký konkrétny variant algoritmu by si mal prečítať príslušný text v šiestej časti a následne vyhľadať popis prvkov, použitých v danom variante, v druhej a tretej časti.

Samozrejme, možných ciest cez predkladaný materiál je viac (vrátane priamej cesty cez všetky časti) a určite si každý dokáže nájsť tú svoju podľa svojich cieľov.

Kniha vznikla na Katedre kybernetiky a umelej inteligencie Fakulty elektrotechniky a informatiky Technickej univerzity v Košiciach. Pri jej písaní

autor vychádzal zo svojich dlhoročných skúseností, získaných počas riešenia viacerých výskumných projektov ako aj zabezpečovania výuky predmetu Evolučné algoritmy. Preto je čiastočne aj dielom mnohých študentov tohto predmetu, ktorí svojimi podnetmi prinútili autora premýšľať, ako vykladať danú problematiku čo najprístupnejším spôsobom.

Na tomto mieste by som chcel vyjadriť poďakovanie recenzentom za starostlivé prečítanie rukopisu, opravu mnohých formálnych aj vecných chýb a za cenné pripomienky a námety, ktoré obohatili obsah predkladaného textu.

Košice, október 2009

autor



# Obsah

<b>Predslov</b>	<b>i</b>
<b>1 Úvod</b>	<b>1</b>
<b>I Štruktúra evolučného algoritmu</b>	<b>9</b>
<b>2 Všeobecná štruktúra</b>	<b>11</b>
2.1 Ilustračná evolúcia . . . . .	16
<b>3 Priestor kandidátov riešení a priestor prehľadávania</b>	<b>21</b>
<b>II Plocha vhodnosti</b>	<b>27</b>
<b>4 Reprezentácia</b>	<b>29</b>
4.1 Štandardné kódovacie schémy . . . . .	31
4.2 Vybrané ukážky kódovania . . . . .	34
4.2.1 Reprezentácia reálneho čísla . . . . .	34
4.2.2 Reprezentácia binárnej/n-árnej hodnoty . . . . .	36
4.2.3 Reprezentácia poradia . . . . .	37
4.2.4 Reprezentácia stromových štruktúr . . . . .	38
4.3 Ilustrácia alternatívnych reprezentácií . . . . .	38
<b>5 Vhodnosť</b>	<b>45</b>
5.1 Aproximatívna vhodnosť . . . . .	47
5.2 Súťažná vhodnosť . . . . .	50
5.3 Vhodnosť založená na komplexnosti . . . . .	51
5.4 Kombinovanie vhodností . . . . .	52
5.5 Ilustrácia alternatívnych vhodností . . . . .	53

<b>III</b>	<b>Základné bloky algoritmu</b>	<b>61</b>
<b>6</b>	<b>Selekcia</b>	<b>63</b>
6.1	Vzorkovanie populácie . . . . .	66
6.1.1	Vzorkovanie s explicitnými pravdepodobnosťami . . . . .	68
6.1.1.1	Výber jedincov . . . . .	69
6.1.1.2	Určenie pravdepodobnosti selekcie . . . . .	71
6.1.2	Vzorkovanie s implicitnými pravdepodobnosťami . . . . .	74
6.1.2.1	Turnaje . . . . .	74
6.1.2.2	Orezanie . . . . .	77
6.1.2.3	Náhodný výber . . . . .	79
6.2	Premapovanie vhodnosti . . . . .	81
6.2.1	Premapovanie škálovaním . . . . .	82
6.2.2	Premapovanie zotriedením . . . . .	86
6.3	Porovnanie selekčných metód . . . . .	89
<b>7</b>	<b>Náhrada</b>	<b>99</b>
7.1	Spôsoby náhrady . . . . .	99
7.2	Výber jedincov a selekčný tlak . . . . .	102
<b>8</b>	<b>Genetické operátory</b>	<b>105</b>
8.1	Operátory pre binárne kódovanie . . . . .	107
8.2	Operátory pre mnohoznakové kódovanie . . . . .	111
8.3	Operátory pre reálne kódovanie . . . . .	111
8.4	Operátory pre permutačné kódovanie . . . . .	118
8.5	Porovnanie rekombinačných operátorov . . . . .	120
<b>IV</b>	<b>Pripraviť sa, pozor, štart</b>	<b>129</b>
<b>9</b>	<b>Nastavovanie parametrov</b>	<b>131</b>
9.1	Exogénne nastavovanie . . . . .	132
9.2	Adaptívne nastavovanie . . . . .	136
9.3	Samoadaptívne nastavovanie . . . . .	139
9.4	“No free lunch” teoréma . . . . .	143
<b>10</b>	<b>Udržiavanie rôznorodosti populácie</b>	<b>147</b>
10.1	Infúzia materiálu . . . . .	151
10.2	Udržiavanie subpopulácií . . . . .	154
10.3	Penalizácia . . . . .	156
10.4	Preferencia odlišnosti . . . . .	159

---

10.5 Porovnanie metód pre udržiavanie rôznorodosti . . . . .	160
<b>V Pohľad pod kapotu</b>	<b>169</b>
<b>11 Hypotéza stavebných blokov</b>	<b>171</b>
<b>12 Vizualizácia</b>	<b>179</b>
12.1 Vizualizácia populácie . . . . .	180
12.2 Vizualizácia jedincov . . . . .	181
12.3 Vizualizácia atribútov . . . . .	189
<b>VI Čo je čo</b>	<b>191</b>
<b>13 Varianty evolučného algoritmu</b>	<b>193</b>
13.1 Klasické prístupy . . . . .	193
13.1.1 Evolučná stratégia . . . . .	194
13.1.2 Evolučné programovanie . . . . .	197
13.1.3 Genetický algoritmus . . . . .	199
13.2 Novšie prístupy . . . . .	202
13.2.1 Genetické programovanie . . . . .	202
13.2.2 Šľachtiteľský algoritmus . . . . .	204
13.2.3 Diferenciálna evolúcia . . . . .	206
13.2.4 Samo-organizujúci sa migračný algoritmus . . . . .	208
13.2.5 Harmonické prehľadávanie . . . . .	209
13.2.6 Eugenická evolúcia . . . . .	211
<b>VII Prílohy</b>	<b>215</b>
Matematické symboly . . . . .	217
Slovensko – anglický slovník . . . . .	221
<b>Literatúra</b>	<b>227</b>
<b>Register</b>	<b>233</b>



# Kapitola 1

## Úvod

Evolučné algoritmy sú špeciálnym prípadom tzv. prehľadávacích algoritmov, ktoré reprezentujú pomerne všeobecný princíp zisťovania riešenia nejakého problému pomocou vyhľadania tohto riešenia. Podstatou tejto triedy algoritmov je pohyb v priestore prehľadávania. Keďže v prípade týchto algoritmov každý bod tohto priestoru zastupuje jedno kompletne riešenie (presnejšie povedané kandidáta na riešenie, pričom rôzni kandidáti v princípe predstavujú rôzne kvalitné riešenia – od optimálneho riešenia cez vyhovujúce až k neprijateľným), tak tento priestor býva označovaný aj ako priestor (kandidátov) riešení.

Prehľadávanie tohto priestoru v sebe typicky kombinuje generovanie resp. výber kandidátov na riešenie s testovaním týchto kandidátov. Pri riešení praktických úloh je potrebné preskúmať veľké množstvo kandidátov (typicky to sú desaťtisíce až milióny kandidátov) kým je nájdené uspokojivé riešenie. Samotný pohyb v priestore prehľadávania je špecifický pre ten-ktorý prehľadávací algoritmus.

Prehľadávacie algoritmy majú iba pomerne málo požiadaviek na riešené úlohy, čo im umožňuje riešiť širokú paletu úloh. V podstate sa od úlohy, aby ju bolo možné riešiť prehľadávacím algoritmom, vyžaduje iba to, aby jej riešenie spĺňalo dve podmienky. Podľa nich riešenie musí byť:

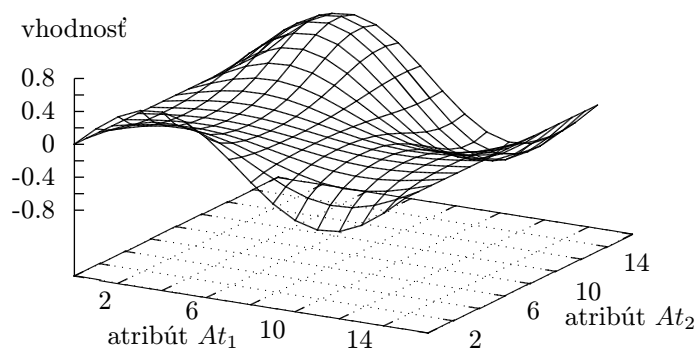
- rozložiteľné,
- ohodnotiteľné.

O hľadanom riešení sa predpokladá, že má tvar dekomponovateľný na zložky. Táto dekompozícia môže vyjadrovať prirodzené zloženie riešenia, ale tak isto aj nejakú účelovú transformáciu. Takéto riešenie sa potom dá vy-

jadriť ako súbor atribútov s ich hodnotami, pričom sa môže jednať o atribúty rôznych typov (napr. binárne, nominálne, reálne, atď.). Príkladmi sú napríklad konštrukcia technického zariadenia (atribúty reprezentujú súčasti zariadenia a ich spojenie), tvorba plánov a postupov (atribúty môžu vyjadrovať poradie jednotlivých aktivít), či hľadanie nejakej numerickej hodnoty (rozložiteľnej napríklad na bity pri binárnej reprezentácii tejto hodnoty).

Keďže kandidáti na riešenie môžu mať rôznu kvalitu, vrátane úplnej nevhodnosti, vyžaduje sa existencia nejakej miery, ktorá umožní jednotlivých kandidátov ohodnotiť – zavádza sa usporiadanie nad týmito kandidátmi a teda je možné týchto kandidátov vzájomne porovnať s následným rozhodnutím, ktorý z nich je lepším a ktorý zase horším riešením konkrétneho riešeného problému. V tejto úlohe sa používa numerická miera, ktorú evolučné algoritmy označujú ako vhodnosť. Čím má kandidát na riešenie vyššiu kvalitu, teda čím lepším riešením je, tým má väčšiu vhodnosť.

Pri splnení týchto dvoch požiadaviek, kladených na tvar hľadaného riešenia, všetci kandidáti na riešenie môžu vytvoriť  $(n + 1)$ -rozmerný priestor jednoduchým spôsobom:  $n$  osí reprezentuje  $n$  atribútov použitých pre popis riešenia (každému atribútu zodpovedá jedna súradnicová os, pričom možné hodnoty atribútu sú mapované na hodnoty na príslušnej súradnicovej osi) a  $(n + 1)$ -vá súradnicová os reprezentuje vhodnosť, označujúcu nakoľko dobrým riešením je nejaká konkrétna kombinácia hodnôt atribútov. Príklad takéhoto priestoru je na obr. 1.1, kde riešenie je predpokladané v tvare dvojice atribútov.

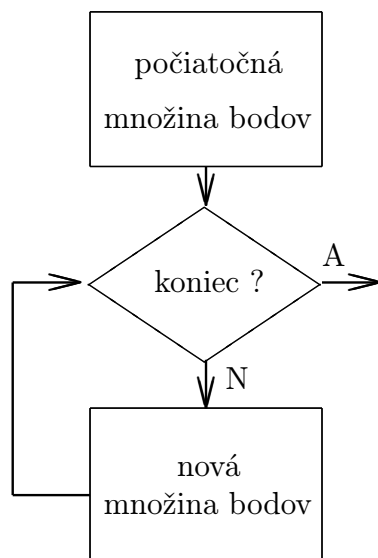


Obr. 1.1: Príklad plochy vhodnosti

Jednotlivé body v tomto priestore (každý z nich reprezentuje jedného kandidáta riešenia) vytvárajú hyperplochu označovanú ako plocha vhodnosti. Platí, že čím je väčšia hodnota vhodnosti, tým je daná kombinácia

hodnôt atribútov lepším riešením. Prehľadávací algoritmus vlastne hľadá polohu extrémnu plochy vhodnosti. Teda prehľadávanie priestoru kandidátov na riešenie (priestoru prehľadávania) je ekvivalentné pohybu<sup>1</sup> po ploche vhodnosti a hľadaniu kombinácie hodnôt atribútov s najväčšou vhodnosťou (optimálne riešenie) alebo s vhodnosťou veľmi blízko maximálnej vhodnosti (suboptimálne riešenie).

Prehľadávacie algoritmy pri pohybe v priestore prehľadávania skúmajú jednotlivých kandidátov na riešenie – body tohto priestoru. Základná štruktúra týchto algoritmov je pomerne jednoduchá, pozostáva iba z niekoľko málo krokov. Je znázornená na obr. 1.2.



Obr. 1.2: Základná štruktúra prehľadávacieho algoritmu

Algoritmus začína svoje prehľadávanie výberom a skúmaním nejakej množiny bodov reprezentujúcich kandidátov riešenia danej úlohy. Ak sa medzi nimi nenachádza hľadané riešenie, tak pokračuje vo vzorkovaní plochy vhodnosti a prechádza na výber a skúmanie ďalších bodov. Celý proces sa cyklicky opakuje. Takýmto spôsobom sa prehľadáva stále väčšia a väčšia časť priestoru prehľadávania, pričom vzorkovanie plochy vhodnosti poskytuje algoritmu stále presnejšiu predstavu o jej tvare.

V tejto štruktúre sú zahrnuté iba dve charakteristiky, v ktorých sa jed-

<sup>1</sup>Tento pohyb má charakter nespojitého vzorkovania plochy vhodnosti.

notlivé prehľadávacie algoritmy navzájom líšia: kardinalita (mohutnosť) množiny súčasne skúmaných bodov priestoru a spôsob vytvárania/výberu nových bodov.

Algoritmus môže súčasne pracovať nielen s jedným bodom ale s celou populáciou týchto bodov. Pri práci s viacerými bodmi naraz sú zvyčajne informácie, získané z vyhodnotenia viacerých bodov, navzájom kombinované. Podľa kardinality použitej populácie sa algoritmy delia do dvoch základných skupín:

- individuálne algoritmy – pracujú súčasne iba s jedným bodom,
- populačné algoritmy – používajú populáciu viacerých bodov, pričom prehľadávanie môže mať charakter niekoľkých nezávislých prehľadávaní súčasne alebo jedného kombinovaného prehľadávania.

Podobne aj podľa spôsobu vytvárania nových bodov je možné medzi rozličnými prehľadávacími algoritmi nájsť rozdiely. Extrémnymi (hraničnými) prípadmi prehľadávacích stratégií sú prístupy zamerané na

- skúmanie (priestoru prehľadávania),
- využívanie (známej informácie).

Prístup zameraný na skúmanie priestoru prehľadávania vyberá nové body nezávisle na predchádzajúcich (teda nevyužíva sa informácia, získaná vyhodnotením predchádzajúcich bodov - plocha vhodnosti je vzorkovaná bez ohľadu na získané informácie o štruktúre plochy vhodnosti). Snahou je preskúmať stále nové a nové oblasti priestoru prehľadávania. Počas prehľadávania nedochádza k sústredeniu pozornosti iba na nejakú špecifickú oblasť priestoru prehľadávania.

Naproti tomu prístup, zameraný na využívanie už známej informácie, sa snaží vybrať nové body na preskúmanie na základe informácie, získanej z vyhodnotenia predchádzajúcich bodov. Inak povedané, snaží sa čo najviac využiť všetku informáciu, ktorú doteraz nazhromaždil, a na jej základe sa sústreďuje na tie oblasti priestoru prehľadávania, ktoré považuje za najslubnejšie (informácia o štruktúre plochy vhodnosti je využívaná v maximálnej miere). Tieto najslubnejšie oblasti sa následne snaží hlbšie preskúmať.

Príkladom prvého extrémneho prístupu je napr. náhodné hľadanie. Rerezentantom druhého je zase horolezecký algoritmus. Mnoho prehľadávacích algoritmov však nepatrí ani do jednej extrémnej skupiny. Sú to tie algoritmy, ktoré sa snažia v nejakom pomere kombinovať skúmanie nových častí priestoru prehľadávania s využívaním doteraz nahromadenej informácie.



Evolučné algoritmy reprezentujú pomerne širokú triedu algoritmov. Ako už bolo spomenuté, sú špeciálnym prípadom prehľadávacích algoritmov. Sú to populačné algoritmy, pracujúce s populáciou kandidátov na riešenie (označovaných ako jedince), pričom kardinalita tejto populácie je typicky desiatky až stovky jedincov.

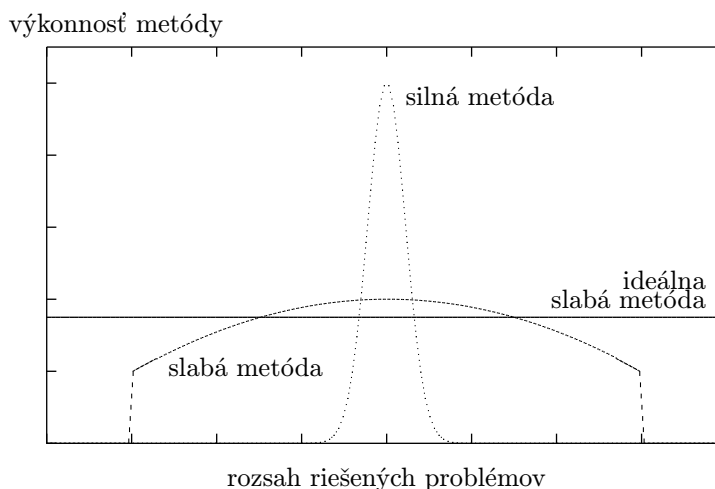
Výber nových jedincov má stochastický charakter, avšak hľadanie nie je úplne náhodné – pravdepodobnosti jednotlivých výberov sú cieľavedome ovplyvňované. Jedná sa o kombinovanie skúmania priestoru prehľadávania (reprezentované stochastickým výberom) a využívania dostupnej informácie (ovplyvňovanie pravdepodobností výberov tak, aby výber smeroval do určitých oblastí priestoru prehľadávania).

Pri prílišnom dôraze na faktor využívania by sa hľadanie sústreďovalo iba do nejakej úzko ohraničenej oblasti priestoru prehľadávania. Toto by malo za následok stratu variability jedincov – jedince uvažované v rámci jednej populácie by boli navzájom príliš podobné. Pri vytváraní novej populácie jedincov by boli potom tieto jedince vyberané iba z danej oblasti, čo by spôsobovalo algoritmu ťažkosti pri nutnosti preniesť pozornosť z jednej oblasti priestoru prehľadávania na oblasť inú. Na druhej strane, prílišné zdôraznenie faktoru skúmania by viedlo k veľkej rôznorodosti jedincov, ktorá by mohla byť prekážkou pri nutnosti detailnejšieho preskúmania nejakej oblasti priestoru prehľadávania. Návrh evolučného algoritmu je teda vlastne procesom hľadania vhodného kompromisu medzi snahou sústrediť hľadanie na nejakú oblasť a zachovávaním možnosti prenášať pozornosť z jednej oblasti priestoru na inú oblasť.

Samotný mechanizmus vytvárania nových jedincov je založený na imitácii dvoch prírodných procesov – Darwinovho procesu evolučného výberu a Mendelovho procesu génovej dedičnosti. Miera napodobňovania vynálezov prírody však kolíše pre rôzne evolučné algoritmy v značnom rozsahu – od pomerne presnej simulácie prírodných procesov až k (veľmi) voľnej inšpirácii týmito procesmi. Napodobňovanie prírodných procesov však nie je cieľom ale iba prostriedkom – cieľom je využitie mechanizmov vymyslených prírodou pre riešenie úloh.

Vďaka tomu, že evolučné algoritmy vyžadujú iba málo informácií o riešenej úlohe (skryté v jednej číselnej hodnote ohodnocujúcej celkovú kvalitu toho-ktorého kandidáta na riešenie, resp. v tvare výsledného riešenia), sú typickým reprezentantom slabých metód. Na základe tejto príslušnosti je možné získať predstavu o výkonnosti evolučných algoritmov pri riešení úloh (obr. 1.3).

Slabé metódy sú charakterizované malým objemom znalostí, ktoré vyžadujú o riešených úlohách. Pretože slabé metódy kladú iba málo podmienok,



Obr. 1.3: Porovnanie výkonnosti slabých a silných metód

ktoré musia byť splnené úlohami aby mohli byť riešené týmito metódami, je možné tieto metódy použiť pre riešenie širokej škály problémov.

Ideálna slabá metóda by dokázala vyriešiť všetky úlohy, pričom každú z nich by riešila s rovnakou výkonnosťou (presnosťou, rýchlosťou, atď.). Keďže reálna slabá metóda (napr. evolučný algoritmus) predsa len kladie nejaké podmienky na riešené úlohy, už nedokáže riešiť všetky úlohy, ale iba nejakú (aj keď značne veľkú) podmnožinu týchto úloh. Navyše, nie je rovnako vhodná pre riešenie každej úlohy, takže jej výkonnosť pri riešení rôznych úloh kolíše.

Naproti tomu silná metóda vyžaduje, aby riešená úloha spĺňala mnoho obmedzení. Následkom toho môže byť aplikovaná iba na veľmi úzku podmnožinu úloh – oveľa užšiu ako slabá metóda. Toto jej však umožňuje, že nemusí byť príliš všeobecnou metódou ale môže byť špecializovanou na riešenie podmnožiny úloh, berúc do úvahy špecifické vlastnosti danej podmnožiny úloh. To vedie k tomu, že výkonnosť silnej metódy je, pri vhodnej implementácii, značne priaznivejšia ako pri metóde slabej.

Z porovnania vyplýva, že evolučné algoritmy z princípu nemôžu súťažiť s omnoho špecializovanejšími silnými metódami. Ich použitie je prirodzené v týchto situáciách:

- neexistuje žiadna silná metóda, použitie slabej metódy je jediná možnosť ako sa pokúsiť úlohu riešiť,

- doménová oblasť je zložitá – ak aj existujú silné metódy na riešenie úloh z tejto domény, sú ťažko použiteľné a ich využívanie je veľmi náročné,
- existuje síce použiteľná silná metóda, avšak požiadavky na riešenie úlohy vyžadujú takú výkonnosť riešiacej metódy, ktorá môže byť dosiahnutá aj slabou metódou<sup>2</sup>.

V prvom prípade môžu evolučné algoritmy reprezentovať jedinú šancu ako niečo vyriešiť, v druhom prípade zase umožnia zjednodušiť riešenie (napr. v situácii, keď niektoré údaje o úlohe nie sú dostupné vôbec alebo ich získanie je spojené s obtiažami). Dôvodom pre použitie evolučných algoritmov v treťom prípade môže byť snaha nahradiť používanie niekoľkých rôznych špecifických metód jednou všeobecnejšou metódou.

---

<sup>2</sup>A navyše ak implementácia silnej metódy vyžaduje viac času a úsilia ako aplikácia generickej slabej metódy.



---

Časť I

Štruktúra evolučného  
algoritmu



## Kapitola 2

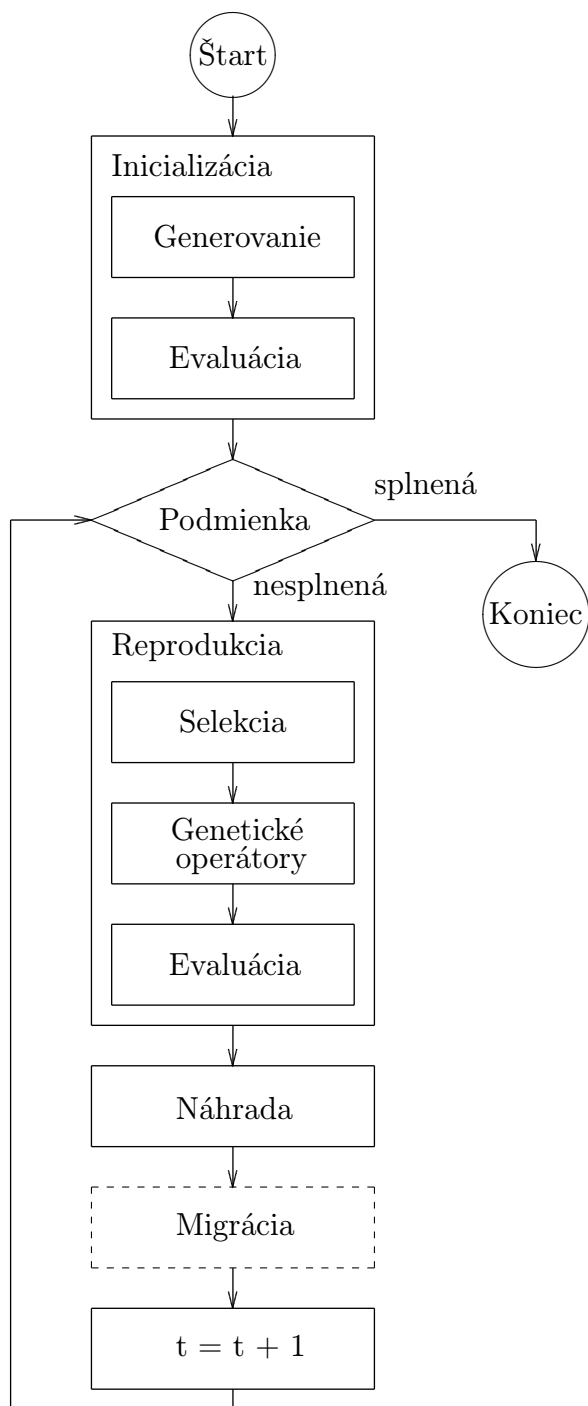
# Všeobecná štruktúra

Pretože evolučný algoritmus je zároveň prehľadávacím algoritmom, aj jeho štruktúra musí zodpovedať štruktúre prehľadávacieho algoritmu uvedenej na obr. 1.2. Rozdiel je však v detailnosti zobrazenia - keďže evolučný algoritmus je špeciálnym prípadom prehľadávacieho algoritmu, tak jednotlivé všeobecné bloky prehľadávacieho algoritmu sú nahradené špeciálnejšími blokmi, bližšie reflektujúcimi vlastnosti evolučných algoritmov. Táto štruktúra je zobrazená na obr. 2.1.

Evolučný algoritmus je populačným prehľadávacím algoritmom – teda naraz skúma viacero kandidátov na riešenie (bodov priestoru prehľadávania). Každý skúmaný kandidát sa označuje ako jedinec – individuum, ktorého vnútorná štruktúra definuje jeho polohu v priestore kandidátov. Tie jedince, ktoré sú algoritmom skúmané v určitom čase, vytvárajú populáciu jedincov.

Každý jedinec, s ktorým algoritmus pracuje, má priradenú svoju individuálnu vhodnosť, ktorá ho charakterizuje z hľadiska cieľa evolučného algoritmu (ktorým je nájdenie riešenia určitého problému). Vnútorná štruktúra jedinca spolu s jeho vhodnosťou definujú polohu daného jedinca na ploche vhodnosti (obr. 1.1).

Na začiatku činnosti algoritmu je vytvorená prvotná populácia jedincov, ktorá postupom času prechádza vývojom v rámci evolučného cyklu. Čas  $t$  je meraný v diskretných jednotkách – generáciách. Každá generácia zodpovedá jednému prechodu evolučným cyklom – teda v každom prechode je pozornosť algoritmu sústredená na tú istú generáciu populácie jedincov, označovanú ako populácia v čase  $t$  alebo populácia  $t$ -tej generácie. Počas jedného prechodu evolučným cyklom sa teda čas nemení, iba na jeho konci dochádza k jeho inkrementácii. Prechod medzi generáciami (prechod z jedného prechodu



Obr. 2.1: Všeobecná štruktúra evolučného algoritmu



cyklom k prechodu nasledujúcemu) je na obrázku preto reprezentovaný blokom  $t = t + 1$ . V súlade s týmto značením, prvotná populácia jedincov sa obvykle označuje ako populácia 0-tej generácie.

V rámci evolučného cyklu dochádza k tvorbe nových a nových populácií jedincov<sup>1</sup>. Pri vytváraní nových generácií jedincov sa evolučný algoritmus snaží kombinovať skúmanie (prehľadávanie) priestoru prehľadávania s využívaním známej informácie získanej počas predchádzajúcich generácií. Pre túto činnosť sa inšpiruje<sup>2</sup> nasledujúcimi teóriami:

- teóriou evolúcie a prirodzeného výberu Charlesa Darwina (1809-1882), ktorá si hneď od svojho publikovania našla zástancov aj odporcov,
- teóriou génovej dedičnosti Gregora Mendela (1822-1884), ktorej publikovanie nevyvolalo záujem súčasníkov a ktorá bola prijatá ostatnými vedcami až po jeho smrti.

Myšlienky evolučného výberu sú používané všade tam, kde je potrebné z populácie dostupných jedincov vybrať iba jedného alebo niekoľkých jedincov. Vtedy výber poskytuje väčšiu šancu tým jedincom, ktorí sú “lepšie prispôsobení životným podmienkam” – pričom pod životným prostredím sa chápe úloha nájsť riešenie nejakého problému. Typicky sú na myšlienke evolučného výberu založené bloky *Selekcia*, *Náhrada* a *Migrácia*.

Princípy dedičnosti vlastností medzi jedincami pomocou génového prenosu sa využívajú pri tvorbe nových jedincov. Pri tejto tvorbe je snaha preniesť čo najviac tých vlastností, ktoré sa podľa doterajšej skúsenosti javia ako žiaduce z hľadiska riešenia problému. Typicky sú na myšlienke génového prenosu založené *Genetické operátory*.

Na začiatku algoritmu je potrebné inicializovať populáciu 0-tej generácie. Teda simulovaný čas začína a platí  $t = 0$ . Samotná inicializácia pozostáva z dvoch krokov. Najprv je potrebné vygenerovať jedince, ktoré budú vytvárať prvotnú populáciu  $P(t = 0)$ . K tomuto účelu slúži blok *Generovanie*. Ak  $a_i(t = 0)$  bude označovať  $i$ -teho jedinca v 0-tej generácii, a počet požadovaných jedincov bude  $\mu(t = 0)$ , tak je potrebné vygenerovať množinu jedincov

$$P(0) = \{a_1(0), a_2(0), \dots, a_{\mu(0)}(0)\} \quad (2.1)$$

<sup>1</sup>Jednotlivé populácie môžu ale nemusia byť disjunktné. Nejaký jedinec môže byť súčasťou populácie v rôznych generáciách, či už nasledujúcich tesne za sebou alebo oddelených populáciami, v ktorých sa daný jedinec nevyskytuje.

<sup>2</sup>Inšpirácia uvedenými teóriami je pomerne voľná, majúca za cieľ iba využiť hlavné princípy týchto teórií pre prehľadávanie priestoru kandidátov a pohyb po ploche vhodnosti. V žiadnom prípade sa nejedná o pokus modelovať prírodné procesy takým spôsobom, ako prebiehajú v prírode – ani čo do komplexnosti ani čo do vernosti.

kde pod generovaním jedinca sa chápe vytvorenie štruktúry jedinca vrátane určenia hodnôt všetkých atribútov vytvárajúcich štruktúru hľadaného riešenia (kap. 4). Každý jedinec takto bude reprezentovať nejakého kandidáta na riešenie.

Jednou z možností je náhodné generovanie – náhodné priradenie hodnôt jednotlivým atribútom. Inou možnosťou je generovanie prvotnej populácie jedincov tak, aby rovnomerne pokrývali priestor prehľadávania. V prípade, že algoritmus má k dispozícii informáciu o nejakej dôležitej kombinácii hodnôt atribútov (dodanej napríklad expertom ako vhodný štartovací bod hľadania), je možné perturbácie týchto hodnôt použiť pre vytvorenie prvotnej populácie jedincov.

V nasledujúcom bloku *Evaluácia* (kap. 5) sa pre každého jedinca v populácii určí jeho individuálna vhodnosť. Táto bude reprezentovať, akými dobrými riešeniami sú kandidáti riešení reprezentovaní jednotlivými jedincami. Účelom je získať predstavu (aj keď limitovanú iba na niekoľko bodov) o tvare plochy vhodnosti, ktorá algoritmu nie je známa. Výstupom evaluácie bude teda množina (distribúcia) hodnôt vhodnosti

$$\{\Phi(a_1(0)), \Phi(a_2(0)), \dots, \Phi(a_{\mu(0)}(0))\} \quad (2.2)$$

Takto inicializovaná prvotná populácia vstupuje do evolučného cyklu. Ten začína blokom *Podmienka*, rozhodujúcim o pokračovaní alebo ukončení evolučného cyklu. Tento blok vlastne definuje ukončovaciu podmienku, ktorej splnenie signalizuje, že už nemá zmysel ďalej pokračovať v evolučnom cykle. Ilustráciou triviálnej ukončovacej podmienky môže byť napríklad to, že hľadané riešenie už bolo nájdené (teda bol identifikovaný jedinec reprezentujúci globálne maximum plochy vhodnosti). Inou podmienkou môže byť uplynutie maximálne povoleného počtu generácií, zvýšenie stupňa konverencie populácie nad zadaný prah (kap. 10), alebo situácia, keď počet generácií, počas ktorých nedošlo k zlepšeniu (napr. priemernej vhodnosti jedincov v populácii), prekročil nejaký prah.

Nesplnenie ukončovacej podmienky vyžaduje realizáciu ďalšieho prechodu evolučným cyklom, ktorého cieľom je vytvoriť novú generáciu jedincov. Nové jedince sú pritom vytvárané v procese reprodukcie, ktorá pozostáva z troch oddelených blokov.

Najprv v bloku *Selekcia* (kap. 6) sú identifikované tie jedince z aktuálnej populácie  $P(t)$ , ktorých vlastnosti sú považované za dostatočne atraktívne pre to, aby boli prenesené na nových jedincov. Tie jedince, ktoré budú selektované aby sa zúčastnili reprodukčného procesu, sa označujú ako rodičia.

Výstupom selekčného procesu bude teda množina rodičov

$$P'(t) = \{b_1(t), b_2(t), \dots, b_{\varrho(t)}(t)\} \quad (2.3)$$

kde  $\varrho(t)$  označuje počet selektovaných rodičov v generácii  $t$ . Zároveň platí  $b_j(t) = a_i(t), i \in \langle 1, \mu(t) \rangle$  pre všetky  $j$ , pričom jedinec z aktuálnej populácie jedincov môže byť za rodiča vybraný raz, viackrát alebo vôbec.

Následne v bloku *Genetické operátory* (kap. 8) sú generované nové jedince. Tie sú vytvárané z rodičov  $\{b_1(t), \dots, b_{\varrho(t)}(t)\}$  použitím jedného alebo niekoľkých operátorov, umožňujúcich transformácie využívajúce hodnoty atribútov jednotlivých rodičov. Novo generované jedince sú označované ako potomkovia. Výstupom tohto kroku evolučného cyklu je preto množina potomkov

$$P''(t) = \{a_{\mu(t)+1}(t), a_{\mu(t)+2}(t), \dots, a_{\mu(t)+\lambda(t)}(t)\} \quad (2.4)$$

kde  $\lambda(t)$  označuje počet potomkov generovaných v  $t$ -tom prechode evolučným cyklom, na ktorého začiatku bola populácia tvorená  $\mu(t)$  jedincami.

Záverečným krokom reprodukčného procesu generovania nových jedincov je určenie vhodnosti nových potomkov v bloku *Evaluácia*, pričom spravidla činnosť tohto bloku je rovnaká ako činnosť podobného bloku pri vyhodnocovaní jedincov prvej populácie.

Nasledujúcim blokom v evolučnom cykle je blok *Náhrada* (kap. 7). Jeho úlohou je z existujúcich jedincov sformovať novú populáciu, ktorá nahradí existujúcu aktuálnu populáciu a stane sa aktuálnou populáciou v nasledovnej generácii. Na vstupe tohto bloku je  $\mu(t)$  jedincov vytvárajúcich aktuálnu populáciu  $P(t)$  a  $\lambda(t)$  jedincov vytvárajúcich skupinu  $P''(t)$  nových potomkov<sup>3</sup>. Výstupom tohto bloku bude  $\mu(t+1)$  jedincov tvoriacich novú populáciu  $P(t+1)$ . Všetky tie jedince, pre ktorých sa nenašlo miesto v tejto novej populácii, upadajú do zabudnutia.

Novo sformovaná nová populácia môže byť ešte upravená v nasledujúcom bloku *Migrácia*. Migrácia jedincov sa používa v prípade, že v algoritme sa uvažuje nie iba jeden evolučný cyklus ale niekoľko oddelených evolučných cyklov, pričom každý z nich vyvíja svoju populáciu jedincov. Aby sa zabezpečila interakcia medzi týmito oddelenými samostatne sa vyvíjajúcimi populáciami, tieto majú možnosť vymieňať si jedného alebo niekoľkých jedincov navzájom. Triviálnou podobou je napríklad nahradiť najhorších jedincov v jednom evolučnom cykle najlepšími riešeniami vyvinutými v ostatných evolučných cykloch.

<sup>3</sup>Tieto dve skupiny nemusia byť disjunktné, rodič mohol byť transformovaný na potomka rovnakého ako je on sám.

Záverene je v poslednom bloku evolučného cyklu inkrementované číslo generácie, čím sa novo vyformovaná populácia  $P(t + 1)$  stáva aktuálnou populáciou  $P(t)$  v nasledujúcej generácii.

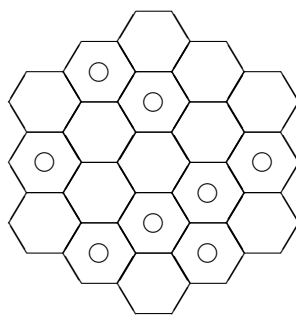
Aj keď je možné použiť evolučný algoritmus v popísanej podobe, keď jednotlivé parametre algoritmu môžu nadobúdať rôzne hodnoty v rôznych generáciách, pri praktickom použití sa zväčša ich hodnoty nemenia a platí

$$\begin{aligned}\mu(t + 1) &= \mu(t) = \mu \\ \varrho(t + 1) &= \varrho(t) = \varrho \\ \lambda(t + 1) &= \lambda(t) = \lambda\end{aligned}\tag{2.5}$$

čo prináša zjednodušenie implementácie algoritmu a zmenšenie počtu parametrov, ktoré je potrebné nastaviť na vhodnú hodnotu.

## 2.1 Ilustračná evolúcia

Aplikovanie princípov, ukotvených v uvedenej všeobecnej štruktúre evolučného algoritmu, pre riešenie úloh je možné demonštrovať na jednoduchom príklade. Predpokladajme, že je daná šesťuholníková sieť miestností, v ktorej je možné prechádzať voľne z miestnosti do miestnosti – teda v každej z miestností sa nachádza šesť dverí, ktorými sa možno dostať do jednej zo šiestich susedných miestností. Niektoré z miestností sú prázdne, zatiaľ čo v iných miestnostiach je uložený poklad. Jeho rozloženie je predpokladané podľa obr. 2.2. Na obrázku je zobrazený iba fragment danej siete, pričom o okolitých miestnostiach, ktoré už nie sú zobrazené, sa predpokladá že sú prázdne.



Obr. 2.2: Rozloženie miestností s pokladom

V strednej miestnosti sa nachádza zberač pokladov, ktorého cieľom je navštíviť čo najviac miestností s pokladom. Jeho pohyb v sieti je však ob-

medzený iba na desať krokov, takže okrem strednej miestnosti môže navštíviť aspoň jednu ďalšiu miestnosť avšak najviac desať iných miestností (je povolený aj návrat do už navštívenej miestnosti).

Aby pre riešenie danej úlohy bolo možné použiť evolučný algoritmus, musí byť jej predpokladané riešenie rozložiteľné. Keďže zberač pokladov má vykonať pohyb, ktorý pozostáva z daného počtu krokov, je možné jeho pohyb rozložiť na:

- miestnosti, ktoré pri svojom pohybe navštívi,
- dvere, ktorými bude pri svojom pohybe prechádzať z jednej miestnosti do druhej,
- smery, ktorými sa pohybuje (prechod z jednej miestnosti do druhej možno považovať za priamočiary pohyb v nejakom smere).

Pre ilustráciu nech sa používa posledná možnosť, kde jednotlivé smery pohybu sa budú označovať podľa svetových strán. Keďže sa jedná o šesťuholníkovú štruktúru, budeme rozoznávať šesť rôznych smerov pohybu:

	S (sever)	
SZ (severozápad)		SV (severovýchod)
JZ (juhozápad)		JV (juhovýchod)
	J (juh)	

Jednotlivé smery pohybu budú v štruktúre jedinca hrať úlohu atribútov, ktoré budú vytvárať štruktúru riešenia, a konkrétne smery (SZ, J, SV, atď.) zase budú reprezentovať možné hodnoty týchto atribútov. Inak povedané, priestor, ktorý je potrebné prehľadávať, bude definovaný desiatimi atribútmi, pričom každý z nich nadobúda jednu zo šiestich možných hodnôt – teda je potrebné prehľadať  $6^{10} = 60466176$  rôznych kandidátov na riešenie.

Okrem rozložiteľnosti musí byť predpokladané riešenie aj ohodnotiteľné. Vhodnosť jednotlivých kandidátov je možné definovať jednoduchým spôsobom ako počet miestností s pokladom, ktoré to-ktoré riešenie navštívilo.

Pri hľadaní riešenia pomocou evolučného algoritmu predpokladajme evolúciu populácie troch jedincov<sup>4</sup>. V rámci inicializácie prvotnej populácie boli náhodným spôsobom vygenerované jedince

<sup>4</sup>Hodnota  $\mu = 3$  bola zvolená iba pre prehľadnosť, pri praktickom použití sa používajú vyššie počty súčasne spracovávaných jedincov, pričom hodnota  $\mu$  je jedným z dôležitých parametrov, ovplyvňujúcich celkovú činnosť algoritmu.

$$\begin{aligned} a_1(0) &= \text{JZ SV JZ S JV SZ JZ JV JV SV} \\ a_2(0) &= \text{SZ J J SV SZ JZ S J S S} \\ a_3(0) &= \text{SV JZ S JZ SZ JV JV JZ SV JV} \end{aligned}$$

ktorých vhodnosť bola určená ako  $\Phi(a_1(0)) = 3$ ,  $\Phi(a_2(0)) = 3$  a  $\Phi(a_3(0)) = 2$ . Teda priemerná vhodnosť populácie v 0-tej generácii je 2.67.

Keďže ešte nebolo nájdené vyhovujúce riešenie (napríklad také, ktoré by našlo aspoň šesť miestností s pokladom), populácia sa bude ďalej vyvíjať v evolučnom cykle. Nasledujúcim krokom je teda vygenerovanie nových jedincov – potomkov. Najprv je potrebné z aktuálnej populácie selektovať rodičov. Tí môžu byť vyberaní napríklad na základe svojej vhodnosti, pričom čím je vyššia vhodnosť jedinca, tým je vyššia pravdepodobnosť jeho výberu. V danom prípade teda jedince  $a_1(0)$  a  $a_2(0)$  budú vyberané s rovnakou pravdepodobnosťou ( $3/8$ ), zatiaľ čo tretí jedinec bude mať menšiu pravdepodobnosť stať sa rodičom ( $2/8$ ). Nech daný pravdepodobnostný výber určí za rodičov jedince  $b_1(0) = a_2(0)$  a  $b_2(0) = a_1(0)$ .

V nasledujúcom kroku z dvoch rodičov sú vytvorení dvaja potomkovia  $a_4(0)$  a  $a_5(0)$ . Najprv títo potomkovia budú mať rovnaký tvar ako ich rodičia

$$\begin{aligned} a_4(0) &= b_1(0) = \text{SZ J J SV SZ JZ S J S S} \\ a_5(0) &= b_2(0) = \text{JZ SV JZ S JV SZ JZ JV JV SV} \end{aligned}$$

Potom je na ne aplikovaný operátor, ktorý v ich kóde vymení posledné tri hodnoty (počet menených hodnôt bol zvolený náhodným spôsobom). Pri tejto výmene sa teda vymenia trojice “J S S” a “JV JV SV”.

$$\begin{aligned} a_4(0) &= \text{SZ J J SV SZ JZ S JV JV SV} \\ a_5(0) &= \text{JZ SV JZ S JV SZ JZ J S S} \end{aligned}$$

Následne je na jedinca  $a_5(0)$  (výber jedinca bol náhodný) aplikovaný druhý operátor, ktorý druhú a tretiu hodnotu “SV JZ” presunie na koniec (výber hodnôt na presun bol náhodný).

$$\begin{aligned} a_4(0) &= \text{SZ J J SV SZ JZ S JV JV SV} \\ a_5(0) &= \text{JZ S JV SZ JZ J S S SV JZ} \end{aligned}$$

Reprodukčný proces sa ukončí ohodnotením novo vygenerovaných potomkov. Ich vhodnosť môže byť určená ako  $\Phi(a_4(0)) = 4$  a  $\Phi(a_5(0)) = 2$ .

Záverne je potrebné nahradiť aktuálnu populáciu  $P(0)$  majúcu tvar  $\{a_1(0), a_2(0), a_3(0)\}$  novou populáciou. Táto nová populácia by mala hrať rolu aktuálnej populácie  $P(1)$  v nasledujúcej generácii. Tvorba novej populácie sa opäť bude realizovať na pravdepodobnostnom princípe, kde čím má

jedinec vyššiu vhodnosť, tým bude mať väčšiu pravdepodobnosť ocitnúť sa v novej populácii. Najväčšiu hodnotu pravdepodobnosti výberu bude mať v našom prípade jedinec  $a_4(0)$  (4/14), zatiaľ čo najmenšiu hodnotu (2/14) budú mať jedince  $a_3(0)$  a  $a_5(0)$ . Príkladom aplikácie takéhoto stochastického výberu je populácia  $P(1)$  pozostávajúca z jedincov  $\{a_1(0), a_3(0), a_4(0)\}$ .

$$\begin{aligned} a_1(1) &= a_1(0) = \text{JZ SV JZ S JV SZ JZ JV JV SV} \\ a_2(1) &= a_3(0) = \text{SV JZ S JZ SZ JV JV JZ SV JV} \\ a_3(1) &= a_4(0) = \text{SZ J J SV SZ JZ S JV JV SV} \end{aligned}$$

Pri takomto tvare novej populácie priemerná vhodnosť vyvíjajúcej sa populácie vzrástla na 3.0.

Uvedeným spôsobom sa bude populácia vyvíjať počas mnohých nasledujúcich prechodov evolučným cyklom. Počas tohto vývoja sa vo všeobecnosti očakáva nachádzanie stále lepších a lepších jedincov, a teda sa očakáva postupné vzrastanie priemernej vhodnosti populácie. Keďže však jednotlivé bloky sú realizované stochastickým spôsobom, vzrast vhodnosti populácie nie je garantovaný – môže dôjsť aj k jej poklesu.

Evolučný vývoj bude aktívny až do doby, kým bude splnená ukončovacia podmienka. Vtedy ako výsledné riešenie bude poskytnutý ten jedinec, ktorý mal najvyššiu vhodnosť v rámci celého evolučného procesu (teda nemusí sa nachádzať v poslednej generácii).

Predpokladajme, že okrem uvedenej vyvíjajúcej sa populácie daná úloha zbierania pokladu je riešená paralelne aj inou vyvíjajúcou sa populáciou. Táto populácia bola inicializovaná pomocou “semena”  $a_1(0)$  – kandidátom riešenia získaného z predchádzajúcich skúseností s riešením daného (alebo podobného) problému.

$$a_1(0) = \text{SZ SZ SV JV J SV JZ J J SZ}$$

Ostatné jedince v prvotnej populácii boli generované z tohto vloženého jedinca pomocou zmien poradia jednotlivých krokov, napríklad výmenou dvoch susedných hodnôt ( $a_2(0)$ ) alebo presunutím nejakej hodnoty na koniec ( $a_3(0)$ ):

$$\begin{aligned} a_2(0) &= \text{SZ SV SZ JV J SV JZ J J SZ} \\ a_3(0) &= \text{SZ SZ SV JV SV JZ J J SZ J} \end{aligned}$$

Vhodnosť všetkých jedincov bola určená rovnakým spôsobom ako v prípade predchádzajúcej populácie. Samotný evolučný cyklus má však na rozdiel od predchádzajúceho menej stochastický charakter, pretože niektoré jeho bloky sú realizované deterministickým výberom.

Napríklad selekcia rodičov je pomerne jednoduchá – každý jedinec aktuálnej populácie sa stáva povinne rodičom, pričom jeden rodič slúži pre vygenerovanie  $\lambda/\mu$  potomkov. Podobne to je aj s náhradou aktuálnej populácie populáciou novou. Jedince z  $P(t)$  aj z  $P''(t)$  sa spoja do jednej množiny a zotriedia podľa klesajúcej hodnoty individuálnych vhodností. Z tejto zotriedenej postupnosti je následne vybraných  $\mu$  najlepších jedincov, ktoré spolu vytvoria novú generáciu  $P(t + 1)$ .

Stochastický charakter si zachováva iba generovanie potomkov, kde potomok vzniká z rodiča náhodným výberom nejakého kroku a jeho náhodnou zmenou na inú hodnotu. Takýmto spôsobom napríklad z rodiča  $b_1(0) = a_1(0)$  s vhodnosťou  $\Phi(a_1(0)) = 4$  môžu vzniknúť potomkovia

$$\begin{aligned} a_{\mu+1}(0) &= \text{SZ SZ SV JV JV SV JZ J J SZ} \\ a_{\mu+2}(0) &= \text{SZ SZ SV JV SV SV JZ J J SZ} \end{aligned}$$

s vhodnosťami  $\Phi(a_{\mu+1}(0)) = 5$  a  $\Phi(a_{\mu+2}(0)) = 3$ .

Vzhľadom na použitý deterministický spôsob realizácie náhrady aktuálnej populácie, je garancia, že v aktuálnej populácii sa nebudú objavovať jedince horšie ako tie, ktoré sa v nej už vyskytujú. Následkom toho priemerná vhodnosť aktuálnej populácie nebude klesať, ale iba buď porastie alebo ostane nezmenená.

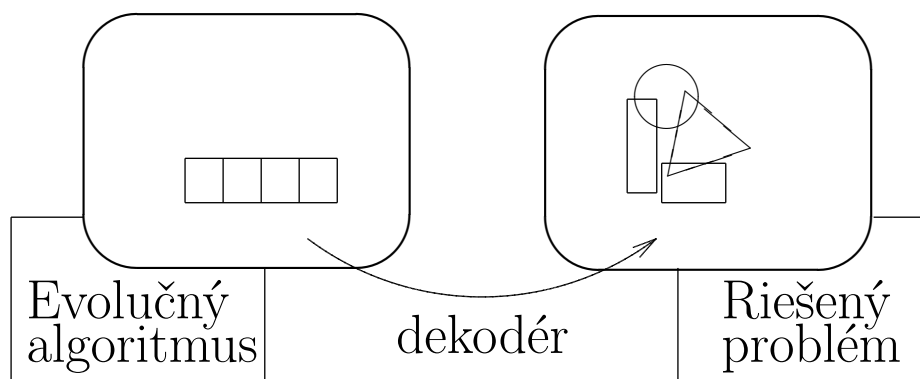
Aby sa obe populácie nevyvíjali úplne nezávisle od seba, ale aby jedna vyvíjajúca sa populácia mohla pri svojom vývoji zúžitkovať aj informáciu získanú počas vývoja druhej populácie, je možné použiť migráciu jedincov pre vzájomnú výmenu informácií. Napríklad každú desiatu generáciu sa najlepší jedinec z každej populácie sprístupní populácii druhej, v ktorej nahradí najhoršieho jedinca.



## Kapitola 3

# Priestor kandidátov riešení a priestor prehľadávania

V predchádzajúcich častiach boli použité pojmy priestor prehľadávania a priestor kandidátov riešení, pričom oba pojmy vystupovali ako pojmy navzájom zameniteľné. V skutočnosti je situácia o niečo zložitejšia – jedná sa o priestory dva, medzi ktorými je potrebné rozlišovať. Situácia, ilustrujúca vzťah týchto dvoch priestorov, je zobrazená na obr. 3.1.



Obr. 3.1: Priestor kandidátov na riešenie (vpravo) kontra priestor prehľadávania (vľavo)

Príčinou pre toto rozlišovanie sú dva odlišné pohľady na to, čo je vlastne riešením (či už riešením skutočným alebo potenciálnym – kandidátom na riešenie):

- pohľad z hľadiska riešeného problému,

- pohľad z hľadiska evolučného algoritmu.

Z hľadiska riešeného problému sa jedná o entitu, ktorá do určitej miery vyhovuje vďaka určitému stupňu splňania všetkých podmienok, požiadaviek a ohraničení definovaných riešeným problémom. Jej štruktúra a/alebo vlastnosti sú vyjadrené pomocou doménovo závislých pojmov, používaných daným problémom.

Naproti tomu z pohľadu evolučného algoritmu sa jedná o štruktúru, ktorá reprezentuje dané riešenie (táto štruktúra musí byť dekomponovateľná na zložky, ako už bolo uvedené v predošlých častiach). Každé riešenie musí mať takú reprezentáciu, ktorej evolučný algoritmus rozumie a dokáže s ňou narábať.

Uvedený dualizmus má za následok oddelenie priestoru prehľadávania od priestoru kandidátov. Všetky možné riešenia, popísané doménovými pojmami, vytvárajú priestor kandidátov riešení (na obrázku znázornený vpravo). Cieľom celého snaženia je vybrať jedného alebo viac z týchto kandidátov ako konečné riešenie riešeného problému. Keďže však je možné riešiť úlohy z rozličných domén, popis týchto riešení môže nadobúdať rozličné formy – a teda nie je možné zaručiť, že evolučný algoritmus bude schopný narábať priamo s týmito riešeniami.

Preto evolučný algoritmus nepracuje priamo s priestorom kandidátov. Jednotlivé riešenia sú reprezentované nejakým vhodným spôsobom, a tieto ich reprezentácie vytvárajú priestor, s ktorým je algoritmus schopný narábať. Toto je priestor prehľadávania (na obrázku znázornený vľavo), ktorý evolučný algoritmus prehľadáva a snaží sa v ňom identifikovať tie reprezentácie, ktoré reprezentujú hľadané riešenia.

Samotná reprezentácia niekedy môže mať tvar totožný (alebo veľmi podobný) ako má reprezentované riešenie, inokedy zase jej tvar môže byť značne rozdielny. Niekedy totiž môže byť reprezentované priamo riešenie, inokedy zase iba jeho časti alebo vlastnosti, a niekedy dokonca sú reprezentované iba artefakty, ktoré síce nie sú súčasťou riešenia avšak pomocou ktorých je možné samotné riešenie odvodiť.

Vo všetkých prípadoch musí existovať spôsob, ako dekodovať reprezentáciu riešenia na samotné riešenie. Formálne, očakáva sa existencia dekodéra, ktorý je vlastne zobrazením

$$g : \mathcal{M}_S \longrightarrow \mathcal{M}_C \tag{3.1}$$

kde  $\mathcal{M}_S$  reprezentuje priestor prehľadávania a  $\mathcal{M}_C$  zase priestor kandidátov riešení.

V ideálnom prípade by mal dekodér každému bodu z  $\mathcal{M}_S$  priradiť jeden bod z  $\mathcal{M}_C$ , pričom rôznym reprezentáciám by mal priradiť rôznych kandidátov. Navyše, čím by boli reprezentácie podobnejšie, tým by mali byť podobnejšie aj priradení kandidáti. Pri praktickom použití sa však nezriedka stáva, že priestory  $\mathcal{M}_S$  a  $\mathcal{M}_C$  sú nerovnako veľké, a teda jedno riešenie môže mať viac reprezentácií (resp. rôzne riešenia môžu mať rozličný počet reprezentácií) alebo niektoré riešenia nie sú vôbec reprezentované<sup>1</sup>. Návrh vhodného spôsobu reprezentácie riešení je jedným z najdôležitejších krokov pri praktickom použití evolučných algoritmov.

Zložitosť dekodéra závisí od rozdielnosti kandidátov riešení a ich reprezentácií. Napríklad, ak kandidát riešenia je sada hodnôt numerických parametrov, a toto riešenie je reprezentované ako vektor reálnych čísel, tak podoba dekodéra je triviálna. Iná situácia je napríklad v prípade, že úlohou je vyfarbiť uzly grafu (aby žiadne dva uzly spojené hranou neboli vyfarbené jednou farbou). Dekodér má jednoduchú podobu, ak riešenie je reprezentované priamo farbami – ako vektor hodnôt, kde hodnota  $i$ -teho prvku vektora reprezentuje farbu ktorou je zafarbený  $i$ -ty uzol grafu. Ak by však riešenie nebolo reprezentované farbami ale zotriedením uzlov grafu – vektorom udávajúcim poradie uzlov grafu (hodnota  $i$ -teho prvku vektora reprezentuje poradie  $i$ -teho uzla grafu), dekodér by bol zložitejší<sup>2</sup>.

Aj keď v prípade triviálneho dekodéra môžu priestor prehľadávania a priestor kandidátov splyvať, v prípade zložitejšieho dekódovania je ich odlišnosť zrejmá. Jedinca, s ktorými evolučný algoritmus manipuluje, sú vždy reprezentácie v priestore prehľadávania. Sú to reprezentácie, ktoré sú selektované do úlohy rodičov, zmenami ktorých sú vytváraní noví potomkovia, a ktoré vytvárajú nové populácie jedincov.

Jediným prípadom, keď je potrebné opustiť priestor prehľadávania a vrátiť sa k priestoru kandidátov, je ohodnocovanie jedincov ich vhodnosťou. Nositeľom vhodnosti je totiž kandidát na riešenie a nie jeho reprezentácia. Iba riešenie môže byť ohodnotený z hľadiska ako dobre alebo zle spĺňa po-

---

<sup>1</sup>Dokonca je možný aj prípad, keď nejaká reprezentácia môže byť dekódovaná na viac riešení, pričom výber medzi nimi sa deje náhodne. V tomto prípade sa zohľadňujú iba najdôležitejšie parametre problému, nastavenie menej dôležitých je odkladané do neskorších etáp.

<sup>2</sup>V tomto prípade by dekodér musel riešenie konštruovať. Bolo by to možné týmto postupom: zoberal by nasledujúci uzol zo zotriedenej postupnosti uzlov grafu a snažil by sa ho vyfarbiť prvou farbou. Ak by to nebolo možné (lebo daná farba bola už použitá niektorým susedným uzlom), skúšal by to s druhou farbou, treťou, štvrtou ... atď. až by našiel vyhovujúcu farbu a postúpil na farbenie nasledujúceho uzla. Ak by sa žiadna vyhovujúca farba nenašla, vrátil by sa na predchádzajúci uzol a hľadal by alternatívne možnosti jeho vyfarbenia.

žiadavky riešeného problému. Preto vo všeobecnosti sa určovanie vhodnosti chápe ako zobrazenie

$$\Phi : \mathcal{M}_C \longrightarrow \mathfrak{R} \quad (3.2)$$

kde  $\mathfrak{R}$  reprezentuje množinu reálnych čísel. Pre určenie vhodnosti nejakého jedinca je teda potrebné reprezentáciu riešenia, vyjadrenú daným jedincom, dekodovať do podoby riešenia, a pre toto dekodované riešenie určiť vhodnosť. Táto hodnota bude nielen vhodnosťou daného kandidáta riešenia, ale sa súčasne stane aj vhodnosťou reprezentácie tohto riešenia.

Keďže neustále dekodovanie môže vyžadovať značné zdroje (výpočtové resp. časové), je snaha určovať vhodnosť ako

$$\Phi : \mathcal{M}_S \longrightarrow \mathfrak{R} \quad (3.3)$$

čo by umožnilo eliminovať opakované dekodovanie bodov z priestoru prehľadávania na body z priestoru kandidátov. Toto však nie je možné vždy, iba pre určité prípady (zvyčajne ak dekodér je pomerne jednoduchý).

Príkladom môže byť problém hľadania cesty v sústave miestností podľa obr. 2.2 s cieľom navštíviť čo najviac miestností s pokladom. Riešením tohto problému je cesta a jej vhodnosť je daná ako počet navštívených miestností s pokladom.

Ak by miestnosti boli určitým spôsobom číslované (poklad sa podľa tohto číslovania<sup>3</sup> nachádza v miestnostiach 2, 4, 5, 11, 13, 15, 17 a 19) a cesta by bola reprezentovaná ako postupnosť čísel miestností v poradí, v akom boli navštívené, tak z reprezentácie cesty

6 1 6 7 1 7 17 6 5 4

by sa dalo priamo určiť, že boli navštívené tri miestnosti s pokladom a teda vhodnosť jedincov by sa dala určiť bez dekodovania. Ak by však cesta bola reprezentovaná pomocou smerov pohybu rovnako ako v kapitole 2.1, tak reprezentáciu

JZ SV JZ S JV SZ JZ JV JV SV

(reprezentujúcu rovnakú cestu ako bola reprezentovaná pomocou čísel miestností) je potrebné najprv dekodovať do tvaru navštívených miestností a až

---

<sup>3</sup>Miestnosti číslované v smere hodinových ručičiek v kružniciach s narastajúcou vzdialenosťou od strednej miestnosti.

na základe toho je možné určiť vhodnosť riešenia a priradiť ju k danej reprezentácii. V tomto prípade nie je možné obísť krok dekódovania pri určovaní vhodnosti.

Z uvedeného vyplýva, že návrh spôsobu určovania vhodnosti jedincov je závislý aj od návrhu spôsobu reprezentácie riešení a ich dekódovania na samotné riešenia.



---

## Časť II

# Plocha vhodnosti





## Kapitola 4

# Reprezentácia

Jedince musia byť vhodným spôsobom reprezentované (kódované), aby evolučný algoritmus s nimi vedel účelne manipulovať. Voľba vhodnej reprezentácie je jedným z centrálnych faktorov pri aplikácii evolučných algoritmov. Rozdiel medzi vhodnou a menej vhodnou reprezentáciou môže rozhodnúť o tom, či algoritmus bude úspešný pri riešení nejakého problému alebo nie. Napriek dôležitosti, ktorá je kladená na voľbu spôsobu reprezentácie riešení, neexistuje všeobecne použiteľný návod<sup>1</sup>, ako voliť reprezentáciu pre nejaký konkrétny problém.

Použitá reprezentácia vlastne definuje priestor prehľadávania a tým zároveň aj stanovuje:

- veľkosť priestoru prehľadávania,
- jeho vzťah k priestoru kandidátov riešení,
- požiadavky na genetické operátory.

Veľkosť priestoru prehľadávania priamo ovplyvňuje úsilie, ktoré je potrebné vynaložiť na jeho prehľadanie. Čím je priestor, ktorý je potrebné prehľadať, väčší, tým väčšie úsilie (merané v spotrebovaných zdrojoch) je potrebné vynaložiť. Aj keď vyhovujúci jedinec môže byť objavený hneď na začiatku prehľadávania, v priemernom prípade (a samozrejme ešte vypuklejšie je to v najhoršom prípade) sa zväčšená veľkosť priestoru prehľadávania prejaví v negatívnom zmysle.

---

<sup>1</sup>Jediné, čo možno poskytnúť, je niekoľko všeobecných rád a spôsoby reprezentácie, ktoré boli úspešné pri riešení konkrétnych problémov v minulosti (čo však neznamená, že v minulosti použité reprezentácie sú pre dané problémy optimálnymi).

Vzťah medzi priestorom prehľadávania a priestorom kandidátov je sprostredkovaný použitým spôsobom dekódovania. Návrh reprezentácie ovplyvňuje jednak zložitosť dekódovania (ako aj zložitosť návrhu vhodného dekodéra) a jednak vzťah medzi reprezentáciami riešení a samotnými riešeniami (kandidátmi riešení).

Intuitívne sa dá chápať viacnásobná reprezentácia (každé riešenie je reprezentované viacerými spôsobmi) ako redundantná so zbytočným zväčšovaním priestoru prehľadávania. Prípade, keď rôzne riešenia sú reprezentované rôznym počtom reprezentácií, zase vedie na nerovnováhu medzi riešeniami a preferovanie riešení s väčším počtom reprezentácií, pretože budú častejšie objavované vďaka svojmu početnejšiemu zastúpeniu v priestore prehľadávania.

Vhodným návrhom reprezentácie je možné vylúčiť niektorých kandidátov na riešenia tým, že žiadna reprezentácia nebude dekódovaná na žiadneho z týchto kandidátov. Toto môže urýchliť hľadanie riešenia v prípade vylúčenia neperspektívnych kandidátov, avšak v prípade vylúčenia skutočného riešenia môže mať za následok neúspech hľadania. Na druhej strane výskyt bodov v priestore prehľadávania, ktoré nie sú dekódované na žiadne riešenia, vyžaduje príslušnú zmenu algoritmu aby bol schopný narábať aj s týmito neinformatívnymi časťami priestoru prehľadávania.

Spôsob reprezentácie, pri ktorom existujú závislosti medzi jednotlivými časťami reprezentácie, vyžaduje prítomnosť takých genetických operátorov, ktoré s týmito závislosťami vedia manipulovať a dokážu produkovať také nové jedince, v ktorých vyžadované závislosti sú zachované<sup>2</sup>. Naproti tomu reprezentácia bez vzájomných závislostí umožňuje použiť jednoduchú formu operátorov.

Zdalo by sa teda, že ideálnou reprezentáciou je taká, ktorá má za následok čo najmenší priestor prehľadávania (rovnako veľký alebo menší ako je priestor kandidátov riešení) s jednoduchým dekódovaním a so žiadnymi požiadavkami na genetické operátory. Avšak nie je to celkom pravda – podmienky pre činnosť evolučného algoritmu definuje synergia medzi spôsobom reprezentácie a spôsobom určovania vhodnosti. Ich kombinácia totiž vytvára plochu vhodnosti, po ktorej sa algoritmus pohybuje a snaží sa nájsť globálny extrém (maximum). Pri riešení konkrétneho problému môže nastať prípad, keď dodržanie uvedených zásad vedie k pre algoritmus menej “priateľskej” ploche vhodnosti ako ich porušenie.

Pri výbere vhodného spôsobu kódovania sú k dispozícii v podstate dva medzné prístupy, z ktorých každý má svojich obhajcov aj odporcov. Prvý

---

<sup>2</sup>Ak to genetické operátory nedokážu, je to nutné riešiť úpravou/doplnením algoritmu.

prístup považuje za prioritnú reprezentáciu. Pri tomto prístupe sa vyberá nejaká kódovacia schéma (viac menej bez ohľadu na vlastnosti riešeného problému) a hľadá sa spôsob, ako namapovať riešenie problému na selektovanú reprezentáciu čo najlepším spôsobom. Prednosťou je použitie štandardnej schémy, pre ktorú sú k dispozícii všetky potrebné zložky evolučného algoritmu, ktoré mu umožňujú manipuláciu s danou reprezentáciou.

Druhý prístup za prioritný považuje riešený problém a snaží sa použiť takú reprezentačnú schému, ktorá reprezentuje riešenie daného problému čo najprirodzenejším spôsobom. A to aj za cenu, že nemôže byť použitá žiadna zo štandardne známych reprezentácií ale musí byť vytvorená nová reprezentácia. Nevýhodou je nutnosť navrhnuť nielen reprezentáciu, ale aj všetky tie zložky algoritmu, ktoré mu umožňujú s danou schémou manipulovať (typicky sa jedná o návrh genetických operátorov). Prednosťou je lepšie prispôbenie riešenému problému.

## 4.1 Štandardné kódovacie schémy

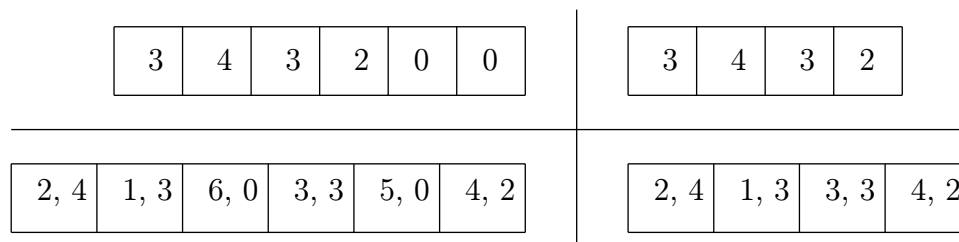
Dôležitými vlastnosťami, ktoré charakterizujú možné reprezentácie, sú dĺžka reprezentácie a usporiadanie prvkov reprezentácie. Obe môžu byť v zásade dvojakeho typu:

- pevné,
- variabilné.

Pri pevnej dĺžke sú reprezentácie všetkých možných riešení rovnako dlhé. Najčastejším prípadom je, keď všetky obsahujú rovnaké zložky (s rôznymi hodnotami) – vtedy každá z reprezentácií reprezentuje jeden bod priestoru prehľadávania (každá zložka definuje jednu os tohto priestoru a možné hodnoty danej zložky zase možné hodnoty na danej osi).

Pri premenlivej dĺžke dve rôzne riešenia môžu byť reprezentované nerovnako dlhými reprezentáciami. Rôzne dlhé reprezentácie neobsahujú rovnaké zložky – čím je reprezentácia kratšia, tým menej zložiek obsahuje. Ak sa neprítomnosť nejakej zložky bude považovať za signál, že na jej hodnote nezáleží (respektíve každá je rovnako prijateľná), tak potom reprezentácie riešení v priestore prehľadávania reprezentujú množiny bodov. Ak by neprítomnosť niektorej zložky znamenala, že daná zložka nadobúda nejakú konkrétnu hodnotu, tak opäť každá reprezentácia vyjadruje jeden bod z priestoru prehľadávania.

Pri pevnom usporiadaní sú zložky v reprezentácii riešenia zoradené vždy v tom istom poradí, zatiaľ čo pri variabilnom sa toto usporiadanie môže líšiť



Obr. 4.1: Rôzne reprezentácie jedného riešenia: s pevným usporiadaním (hore), s variabilným usporiadaním (dole), s pevnou dĺžkou (vľavo) a s variabilnou dĺžkou (vpravo)

pre reprezentácie rôznych riešení (dokonca môže byť odlišné aj pri dvoch reprezentáciách toho istého riešenia). Pretože poradie nie je pevne stanovené, je potrebné kódovať nielen hodnoty jednotlivých zložiek ale aj to, ktorá zložka sa kde v reprezentácii nachádza. Príklady takejto reprezentácie sú použité napríklad v messy genetických algoritmoch [15].

Najjednoduchšie sa manipuluje s reprezentáciou, ktorá má pevnú dĺžku a pevné usporiadanie svojich zložiek. Každá variabilita zvyšuje nároky na prácu s danou reprezentáciou (avšak tie môžu byť vyvážené priateľskejšou výslednou plochou vhodnosti).

Pre ilustráciu uvedených vlastností uvažujme úlohu kúpiť sadu strojov a usporiadať ich do 3 až 6 buniek, pričom v bunke má byť 2 až 6 strojov<sup>3</sup>. Jedno možné riešenie je vytvoriť 4 bunky s 3, 4, 3 a 2 strojmi. Reprezentácie tohto riešenia sú znázornené na obr. 4.1.

Riešenie pozostáva zo zložiek (atribútov), ktoré reprezentujú jednotlivé bunky. Hodnota každej zložky určuje, koľko strojov vytvára danú bunku. Je potrebné uvažovať šesť možných buniek, pričom obsadenosť každej bunky môže byť jedna zo šiestich možností (vrátane prípadu, že bunka bude obsahovať 0 strojov – teda že bunka nebude vytvorená). Pri pevnej dĺžke a pevnom usporiadaní bude teda priestor prehľadávania obsahovať  $6^6 = 46656$  bodov.

Ak by nebolo podstatné, ktorá bunka je ktorá, potom je možné použiť reprezentáciu s premenlivou dĺžkou, ktorá bude obsahovať iba skutočne vytvorené bunky. V tomto prípade sa počet hodnôt, ktoré môže nadobúdať každá zložka, zníži na päť, čím sa veľkosť priestoru prehľadávania zníži. Spolu bude existovať  $5^3 + 5^4 + 5^5 + 5^6 = 19500$  rôznych kandidátov na

<sup>3</sup>Úloha môže zahŕňať aj ďalšie prvky, ktoré momentálne nie sú zaujímavé, ako napr. optimalizácia počtu strojov a ich druhového zloženia voči cene alebo výrobnému programu.

riešenie, spomedzi ktorých je možné vyberať.

Ak by bolo dôležité zohľadňovať nielen rozmiestnenie strojov do buniek ale aj pripustiť rôzne vzájomné usporiadanie jednotlivých buniek v reprezentácii, bolo by možné použiť variabilné usporiadanie, ktoré však pridáva ďalšie informácie do kódu a tým zväčšuje priestor, ktorý je potrebné prehľadať. V prípade variabilného usporiadania s pevnou dĺžkou by sa každá distribúcia strojov medzi bunky dala vyjadriť  $6!$  rôznymi spôsobmi a teda veľkosť tohto priestoru prehľadávania by bola  $6!6^6$  bodov.

V praxi pri riešení problémov rôznych typov sa niektoré spôsoby reprezentácie ukázali ako vhodné pre riešenie pomerne širokej triedy úloh. Z nich vďaka svojej jednoduchosti vynikajú usporiadané lineárne štruktúry pevnej dĺžky. V rámci tejto skupiny je v podstate možné rozlišovať štyri základné typy takýchto štruktúr:

**Binárne kódovanie.** Reprezentácia riešenia má tvar binárneho reťazca – reťazca, ktorý na každej pozícii môže obsahovať iba jednu z dvoch možných hodnôt (zvyčajne zobrazovaných ako 0 a 1). Pri predpoklade, že dĺžka reťazca je  $l$ , je veľkosť priestoru prehľadávania daná ako

$$\|\mathcal{M}_S\| = 2^l \quad (4.1)$$

Tento spôsob reprezentácie je z historických dôvodov<sup>4</sup> najrozšírenejší. Preto aj väčšina teórií, snažiacich sa popísať činnosť evolučných algoritmov, ako aj heuristik pre konfigurovanie evolučného algoritmu je orientovaná na tento spôsob reprezentácie.

**Mnohoznakové kódovanie.** Jedná sa o rozšírenie binárneho kódovania použitím  $n$  znakovkej abecedy. Na každej pozícii reťazca sa môže vyskytovať jedna z  $n$  možných hodnôt, pričom priestor prehľadávania sa zväčší na  $n^l$  bodov. Toto kódovanie nahrádzalo binárne kódovanie najmä v prípadoch, keď počet možných hodnôt sa líšil od mocniny čísla 2. Variantom je prípad, keď rôzne pozície povoľujú rôzne počty možných hodnôt.

**Reálne kódovanie.** Jedinec je pri tomto spôsobe kódovania reprezentovaný ako vektor reálnych čísel. Na každej pozícii sa teda vyskytuje reálne číslo, ktoré môže byť buď neohraničené alebo je z určitého stanoveného intervalu. Priestor prehľadávania má mohutnosť kontinua.

<sup>4</sup>Binárne kódovanie bolo používané genetickými algoritmi – v minulosti najrozšírenejšou formou evolučných algoritmov.

Pôvodne bolo používanie tohto spôsobu reprezentácie reálnym kódovaním geograficky ohraničené na Nemecko<sup>5</sup>, v súčasnosti sa už jedná o štandardný široko rozšírený spôsob reprezentácie.

**Permutačné kódovanie.** Táto kódovacia schéma je určená na reprezentáciu poradia v rámci nejakej množiny objektov. Riešenie je kódované ako jedno z možných zotriedení objektov. Vizuálne má rovnakú podobu ako mnohoznačné kódovanie, pričom pre počet použitých hodnôt platí  $n = l$ . Navyše ešte platí, že žiadne dve rôzne pozície nemajú priradenú rovnakú hodnotu. Veľkosť priestoru prehľadávania je pri tomto spôsobe reprezentácie daná pomocou vzťahu

$$\|\mathcal{M}_S\| = n! \quad (4.2)$$

Okrem uvedených lineárnych štruktúr sa v praktickom použití objavujú aj iné štruktúry. Z nich ako pomerne rozšírené možno spomenúť používanie stromových štruktúr<sup>6</sup> ako príklad reprezentačnej štruktúry s variabilnou dĺžkou alebo maticovo usporiadané štruktúry ako príklad reprezentačnej štruktúry s pevnou dĺžkou.

## 4.2 Vybrané ukážky kódovania

### 4.2.1 Reprezentácia reálneho čísla

**Reálne kódovanie.** Reálne číslo z nejakého intervalu  $\langle min, max \rangle$  sa najprirodzenejším spôsobom dá reprezentovať pomocou reálneho kódovania, kde pre svoju reprezentáciu využíva jednu pozíciu. Ak na tejto pozícii je dané číslo reprezentované ako číslo z iného ako požadovaného intervalu, tak je ho potrebné ešte dodatočne transformovať do požadovaného rozsahu.

V prípade, že reprezentované číslo nie je ohraničené, tak ho je možné pomocou vzťahu

$$\frac{e^x - 1}{e^x + 1} \quad (4.3)$$

transformovať na číslo z intervalu  $\langle -1, 1 \rangle$ , resp. ho je možné transformovať na číslo z požadovaného intervalu podľa

$$min + (max - min) \frac{e^x}{e^x + 1} \quad (4.4)$$

---

<sup>5</sup>V minulosti sa požíval výhradne v oblasti evolučných stratégií – forme evolučných algoritmov rozšírenej v Nemecku.

<sup>6</sup>Stromové reprezentačné schémy boli dominantne využívané vo forme evolučných algoritmov nazývanej genetické programovanie.

**Binárne kódovanie.** Pri tomto druhu kódovania je reálne číslo reprezentované pomocou  $l_i$  pozícií v binárnom reťazci [5].

$$\dots 1 \overbrace{1 \dots 0 1}^{l_i} \overbrace{0 1 0 0 \dots 0 1}^{l_i} \overbrace{1 1 \dots 1}^{l_i} 0 \dots$$

Na danom počte binárnych pozícií je možné zobraziť nezáporné celé číslo v rozsahu od 0 až po  $2^{l_i} - 1$ . Keďže zobrazené číslo (reprezentované v danom segmente binárnym kódom) je ešte potrebné pretransformovať do požadovaného rozsahu, tak výsledné reálne číslo je potrebné určiť podľa vzťahu

$$v_i = \min + \frac{\max - \min}{2^{l_i} - 1} \left( \sum_{j=0}^{l_i-1} h_j 2^j \right) \quad (4.5)$$

kde  $h_j$  reprezentuje hodnotu zobrazenú na  $j$ -tej pozícii v rámci segmentu určeného pre reprezentáciu daného čísla. Pre transformáciu celého čísla na reálne je použitá lineárna transformácia na požadovaný interval hodnôt. Je však potrebné si uvedomiť, že maximálne dosiahnuteľná presnosť tohto spôsobu reprezentácie je  $(\max - \min) / (2^{l_i} - 1)$ . Povedané inak, z požadovanej presnosti je možné odvodiť minimálnu hodnotu  $l_i$ .

Binárne kódovanie čísel má tú nevýhodu, že zmena lokalizovaná v malej oblasti bitového reťazca má nepredvídateľné následky – od prechodu k susednému číslu až po skokovú zmenu cez polovicu rozsahu, v závislosti na tom, ktorá pozícia bola zmenená. Pre zmenu čísla na susedné číslo niekedy stačí zmeniť hodnotu na jednej pozícii, inokedy je zase potrebné zmeniť hodnoty na všetkých pozíciách. Pre odstránenie tejto nerovnomernosti sa často používa Grayov binárny kód, v ktorom ľubovoľné dve susedné čísla sa líšia vždy iba v hodnote jednej pozície. V tomto prípade sa hodnoty v príslušnom segmente reťazca pokladajú za Grayov kód nezáporného čísla a najprv sa pretransformujú do binárneho kódu a až po transformácii čísla do binárneho kódu sa následne určí hodnota reálneho čísla už uvedeným postupom.

Číslo v Grayovom kóde je možné transformovať na binárnu reprezentáciu podľa vzťahu<sup>7</sup>

$$h_i^b = \bigoplus_{j=0}^i h_j^g \quad \text{pre } i = 0, 1, \dots, l_i - 1 \quad (4.6)$$

kde  $\oplus$  označuje súčet modulo 2. Tak napríklad číslo 111 v Grayovom kóde bude zodpovedať číslu 101 v binárnom kóde a číslo 100 sa zase bude transformovať na číslo 111. Pre zvýšenie účinnosti hľadania je možné doplniť Grayov

<sup>7</sup>Hodnota  $i = 0$  označuje najvyšší rád a  $i = l - 1$  zase najnižší rád daného čísla.

kód posunmi, umožňujúcimi využívať malú podmnožinu rôznych Grayových reprezentácií [37].

**Mnohoznakové kódovanie.** Keďže toto kódovanie je iba rozšírením binárneho kódovania na  $n$  použitých symbolov ( $n$  je väčšie než 2), stačí vo vzťahu (4.5) zameniť hodnotu 2 za hodnotu  $n$ .

#### 4.2.2 Reprezentácia binárnej/ $n$ -árnej hodnoty

**Binárne kódovanie.** Binárna hodnota sa najprirodzenejším spôsobom dá reprezentovať pomocou binárneho kódovania. Pre reprezentáciu sa využíva jedna pozícia, ktorej hodnota priamo určuje hodnotu reprezentovanej binárnej hodnoty.

$N$ -árnu hodnotu (pre  $n$  väčšie ako 2) je možné pomocou binárneho kódovania reprezentovať rovnakým spôsobom ako binárne kódované kladné celé číslo. Inou možnosťou je použiť pre každú z možných hodnôt jednu pozíciu, pričom je potrebné pri dekódovaní zabezpečiť riešenie konfliktu, keď súčasne je signalizovaná prítomnosť viacerých hodnôt.

**Mnohoznakové kódovanie.** Binárna hodnota sa dá reprezentovať pomocou tohto kódovania prostredníctvom jednej pozície, na ktorej sa môže vyskytovať jedna z párneho počtu možných hodnôt. Polovica z týchto hodnôt je dekódovaných na jednu hodnotu a druhá polovica zase na druhú možnú hodnotu reprezentovanej binárnej premennej.

$N$ -árna hodnota sa v tejto reprezentácii prirodzeným spôsobom reprezentuje použitím jednej pozície.

**Reálne kódovanie.** Binárna hodnota sa pomocou tohto spôsobu kódovania môže reprezentovať pomocou jednej pozície, na ktorej sa nachádza reálne číslo z nejakého intervalu  $\langle min, max \rangle$ . Tento interval je rozdelený prahom na dva podintervaly, a podľa toho, do ktorého podintervalu patrí reálna hodnota, vyskytujúca sa na danej pozícii, je táto hodnota dekódovaná na jednu alebo druhú binárnu hodnotu.

Reprezentácia  $n$ -árnej hodnoty je podobná, jediný rozdiel je ten, že interval, z ktorého môže byť číslo na danej pozícii, je rozdelený prahmi na  $n$  podintervalov.

**Permutačné kódovanie.** Binárnu hodnotu je možné kódovať aj pomocou tohto spôsobu kódovania (aj keď účelnosť takéhoto kódovania môže byť dis-



kutabilná). Pre jej kódovanie je potrebné použiť dve pozície, každá z nich pritom zodpovedá jednej možnej binárnej hodnote. Usporiadanie týchto dvoch pozícií, dané celkovou reprezentovanou permutáciou, môže byť dvojaké – a výsledkom je teda reprezentácia jednej alebo druhej hodnoty binárnej premennej.

Podobným spôsobom je možné reprezentovať  $n$ -árnu hodnotu, pričom tá je reprezentovaná pomocou  $n$  pozícií. Každá z nich zodpovedá jednej povolenej hodnote z  $n$  možných hodnôt. Vo výslednej permutácii tá z týchto pozícií, ktorá je z nich prvá podľa usporiadania reprezentovaného permutáciou, je dekódovaná na hodnotu, ktorú reprezentuje.

### 4.2.3 Reprezentácia poradia

**Permutačné kódovanie.** Poradie môže byť týmto spôsobom kódovania veľmi prirodzene reprezentované ako permutácia poradia objektov. Každý objekt je zviazaný s určitou pozíciou a hodnota, reprezentovaná na tejto pozícii, priamo určuje poradové číslo daného objektu.

**Reálne kódovanie.** Poradie  $n$  objektov je možné reprezentovať pomocou vektoru reálnych čísel. Dĺžka tohto vektoru je rovná hodnote  $n$  a čísla na všetkých pozíciách sú z toho istého rozsahu. Dekódovanie sa deje na základe porovnávania jednotlivých reálnych hodnôt na jednotlivých pozíciách. Takýmto spôsobom napríklad vektor  $[5.14, 4.62, -0.19, 2.17, -6.09]$  môže byť dekódovaný na poradie  $(5\ 4\ 2\ 3\ 1)$ , pretože najmenšie číslo sa nachádza na piatej pozícii, druhé najmenšie na tretej pozícii, atď.

**Mnohoznakové kódovanie.** Pomocou tohto spôsobu reprezentácie je taktiež možné reprezentovať poradie, pričom pre reprezentáciu poradia  $n$  objektov je potrebný vektor  $n$  hodnôt. Na prvej pozícii tohto vektora sa bude vyskytovať číslo z množiny  $\{1, 2, \dots, n\}$ , na druhej pozícii bude číslo z množiny  $\{1, 2, \dots, n - 1\}$ , na predposlednej pozícii bude hodnota 1 alebo 2 a na poslednej pozícii bude vždy 1 (a teda vlastne túto poslednú pozíciu ani nie je potrebné reprezentovať).

Dekódovanie sa deje prostredníctvom takzvaného referenčného zoznamu a hodnoty v reprezentačnom vektore hrajú úlohu odkazov do tohto referenčného zoznamu (resp. odkazov na prvky tohto zoznamu, ktoré ešte neboli použité). Tak napríklad pri referenčnom zozname  $(5\ 4\ 3\ 2\ 1)$  je reprezentačný vektor  $[1, 1, 2, 1, 1]$  dekódovaný na výsledné poradie  $(5\ 4\ 2\ 3\ 1)$ . Na začiatku je hodnota 5, pretože prvá jednotka v reprezentačnom vektore ukazuje na hodnotu 5 (ktorá je prvá z doposiaľ ešte nepoužitých hodnôt v

referenčnom zozname). Druhá jednotka z vektoru poskytne hodnotu 4 (čo je teraz prvá z nepoužitých hodnôt v referenčnom zozname). Hodnota 2 z vektora vedie na hodnotu 2, pretože v tomto okamihu má referenčný zoznam tvar (3, 2, 1).

#### 4.2.4 Reprezentácia stromových štruktúr

**Mnohoznakové kódovanie.** Strom, ktorý môže mať až  $n$  uzlov, je možné kódovať ako vektor, ktorého dĺžka je aspoň  $n$ . Každá pozícia svojou hodnotou reprezentuje jeden z uzlov grafu. Pri dekódovaní sa vektor prechádza z jednej strany na druhú, pričom vždy dvojica pozícií reprezentuje možnú hranu. Hrana je však akceptovaná iba vtedy, ak vo výslednom grafe by jej pridaním nevznikla cyklická štruktúra. Takýmto spôsobom z jedinca (1, 3, 1, 2, 4, 5, 5, 3) možno získať strom s hranami (1,3), (1,2), (2,4) a (4,5).

Ak dĺžka vektora je  $2(n-1)$  a každý uzol je v ňom reprezentovaný aspoň raz, tak dekódovaním možno získať kosť grafu o  $n$  uzloch [40].

**Permutačné kódovanie.** Toto je spôsob reprezentácie stromových štruktúr, ktorý sa hodí iba pre grafy s nie príliš veľkým počtom uzlov (teda množstvo potenciálnych hrán, ktoré môžu byť brané do úvahy, nie je nevládnuteľne veľké). Pri tomto spôsobe reprezentácie strom, ktorý môže mať až  $n$  uzlov, je možné kódovať ako permutáciu, ktorej dĺžka je maximálne  $n(n-1)/2$ . Každá pozícia reprezentuje jednu možnú hranu (alebo jednu dvojicu uzlov) a hodnota na tejto pozícii udáva poradie vo výslednom zotriedení všetkých potenciálnych hrán.

Pri dekódovaní sa jednotlivé hrany prechádzajú v poradí danom reprezentovanou permutáciou. Hrana je akceptovaná iba vtedy, ak vo výslednom grafe jej pridaním nevzniká cyklická štruktúra.

### 4.3 Ilustrácia alternatívnych reprezentácií

Ako ilustračný príklad pre demonštráciu použiteľnosti rôznych spôsobov reprezentácie použijeme problém známy ako farbenie grafov. Budeme uvažovať štáty Afriky a rozdelíme tieto štáty do určitého počtu skupín tak, aby žiadne dva štáty, ktoré navzájom susedia (pod susedstvom sa bude chápať bezprostredná blízkosť, teda ostrovný štát sa považuje za štát, ktorý nemá suseda), nepatrili do jednej a tej istej skupiny. Je teda potrebné uvažovať 53 štátov a 103 susediacich dvojíc.

Aby sa dali získané výsledky porovnať z hľadiska vplyvu voľby reprezentácie, bolo by potrebné aby sa jednotlivé varianty algoritmu líšili iba v

použitej reprezentácii. To však bohužiaľ nie je možné, pretože pre rôzne reprezentácie je potrebné použiť aj rôzne genetické operátory, ktoré sú schopné s danou reprezentáciou narábať. Výsledkom je teda súhrnný vplyv voľby reprezentácie a zodpovedajúcej sady genetických operátorov (pri inej voľbe operátorov by vzájomné porovnanie mohlo dopadnúť inak). Výsledky je preto z hľadiska porovnania rôznych spôsobov reprezentácie potrebné považovať iba za indikatívne<sup>8</sup>, ilustrujúce fakt, že voľba spôsobu reprezentácie ovplyvňuje celkovú výkonnosť algoritmu.

Riešením problému bude také priradenie štátov do skupín, pre ktoré každá susediaca dvojica štátov má rôzne priradenia. Pre reprezentáciu tohto riešenia boli použité tieto štyri možnosti:

**Binárne kódovanie.** Riešenie je reprezentované ako vektor binárnych hodnôt. Dĺžka vektora je prirodzeným násobkom počtu uvažovaných štátov.

*Dekódovanie (číselné):* Každému štátu zodpovedá úsek  $l_i$  bitov, pričom toto priradenie je pevne dané počas celého behu algoritmu. Hodnota  $l_i$  je minimálne taká, aby umožňovala kódovanie všetkých skupín, ktoré môžu byť použité pre rozdeľovanie štátov. Hodnoty z určitého úseku sa dekodujú na číslo skupiny podľa vzťahu

$$\left[ \left( \sum_{j=0}^{l_i-1} h_j 2^j \right) \% NG \right] + 1 \quad (4.7)$$

kde  $NG$  reprezentuje počet uvažovaných skupín a  $\%$  zase operáciu modulo. Daný vzťah umožňuje čo najrovnomernejšie kódovanie jednotlivých skupín.

*Dekódovanie (pozičné):* Opäť každému štátu zodpovedá úsek  $l_i$  bitov, pričom toto priradenie je opäť pevne dané počas celého behu algoritmu. Hodnota  $l_i$  je teraz o jednotku menšia než je počet povolených skupín. Jednotka na prvej pozícii znamená prvú skupinu, jednotka na druhej pozícii zase druhú skupinu, atď. Ak sú naraz nastavené viaceré skupiny, výsledná skupina sa vyberie na základe dominancie – druhá skupina je dominantná voči prvej, tretia je dominantná voči prvej a druhej, atď. A ak na žiadnej pozícii nie je jednotka, znamená to poslednú skupinu, ktorá nemá priradenú reprezentatívnu pozíciu.

**Mnohoznakové kódovanie.** Riešenie je reprezentované ako vektor celých čísel, kde na každej pozícii sa vyskytuje číslo z množiny  $\{1, 2, \dots, NG\}$ , kde

<sup>8</sup>Je potrebné si uvedomiť, že porovnanie nemá absolútnu platnosť, ale iba relatívnu vzhľadom na daný riešený problém. Viac o tom v časti o No Free Lunch teoréme.

*NG* opäť reprezentuje maximálny počet skupín, do ktorých je potrebné štáty rozdistribúovať. Dĺžka vektora je rovná počtu uvažovaných štátov.

*Dekódovanie:* Každá pozícia zodpovedá jednému štátu, pričom priradenie je pevne dané počas celého behu algoritmu. Hodnota na nejakej pozícii zodpovedá skupine, do ktorej je reprezentovaný štát zaradený.

**Reálne kódovanie.** Riešenie je reprezentované ako vektor reálnych čísel, kde na každej pozícii sa vyskytuje číslo z intervalu  $(0, NG)$ , uzavretého zľava a otvoreného sprava. Dĺžka vektora je rovná počtu všetkých štátov.

*Dekódovanie:* Každý štát je reprezentovaný práve jednou pozíciou, pričom toto priradenie sa počas behu algoritmu nemení. Hodnota na každej pozícii je transformovaná na skupinu, do ktorej je zaradený príslušný štát, podľa vzťahu  $\lfloor h_i \rfloor + 1$ , v ktorom je použité zaokrúhlenie nadol.

**Permutačné kódovanie.** Riešenie je reprezentované ako permutácia množiny čísel  $\{1, 2, \dots, NS\}$ , kde  $NS$  je počet uvažovaných štátov. Inak povedané, dĺžka vektora je rovná  $NS$ .

*Dekódovanie:* Každý štát je reprezentovaný jednou pozíciou, pričom toto priradenie je pevné počas celého behu algoritmu. Hodnota na každej pozícii reprezentuje poradie, v ktorom daný štát bude uvažovaný. Riešenie sa konštruuje podobným postupom, ako bol uvedený v kapitole 3. Postupne sa berie zo zotriedenej postupnosti štátov nasledujúci, doposiaľ nepriradený štát, a priradí sa do prvej skupiny. Ak to nie je vhodné priradenie (lebo jeden z jeho susedov už je v danej skupine), tak sa priradí do ďalšej skupiny, atď. Ak ho nie je možné priradiť do žiadnej skupiny, tak ostáva bez priradenia a prechádza sa na nasledujúci štát v danom zotriedení.

Jednotlivé spôsoby reprezentácie boli testované pri štyroch líšiach sa nastaveniach (odlišnosť bola vždy iba v hodnote jedného parametra). Základným bolo rozdeliť všetky štáty do štyroch skupín, čo bolo najťažšou úlohou (úloha nie je riešiteľná pri troch skupinách). Ostatné varianty nastavenia boli o niečo ľahšie (aj keď nie každý z algoritmov si to myslel), vo všeobecnosti umožňujúce nájsť riešenie rýchlejšie. Boli to:

- väčšia populácia poskytuje viac stavebného materiálu pri tvorbe nových jedincov (jej veľkosť bola zdvojnásobená),
- väčší selekčný tlak (pojem definovaný v kapitole 10) umožňuje rýchlejšiu konvergenciu,

Reprezentácia	Variant nastavenia			
	4 skupiny	4 skupiny (+populácia)	4 skupiny (+tlak)	5 skupín
Binárna (číselná)	11 050	11 100	7 000	2 850
Binárna (pozičná)	13 350	19 400	10 850	2 350
Mnohoznaková	8 400	8 500	6 600	2 050
Reálna	17 600	12 100	10 150	3 700
Permutačná	60	100	60	50

Tabuľka 4.1: Priemerné počty generovaných jedincov potrebných pre nájdenie riešenia pri použití rôznych spôsobov reprezentácie

- zvýšený počet povolených skupín robí úlohu jednoduchšou, zvyšuje sa počet prijateľných riešení.

Parametre genetických operátorov v rôznych verziách algoritmu boli nastavené rôzne – tak, aby výkon algoritmu bol čo najlepší. Ostatné časti algoritmu a ich nastavenie boli rovnaké vo všetkých verziách.

Pre porovnanie rôznych (variantov) evolučných algoritmov sa často používa počet vyhodnotení, teda počet generovaných jedincov, potrebných pre nájdenie riešenia. Aj keď toto kritérium je pomerne diskutabilné, pretože vygenerovanie a vyhodnotenie jedného jedinca pri použití rozličných reprezentácií a dekodérov kladie rôzne nároky na výpočtové zdroje, nie je známe lepšie kritérium pre všeobecné porovnanie.

Výsledky získané pre jednotlivé spôsoby reprezentácie pri rôznych variantoch nastavenia sú zhrnuté v tabuľke 4.1. Keďže evolučný algoritmus patrí medzi stochastické algoritmy, v tabuľke uvedené hodnoty sú priemernými hodnotami – každá bola určená ako priemer z tisíc pokusov.

Binárnej reprezentácii sú venované dva riadky, spojené s oboma spôsobmi dekodovania. Pri prvom spôsobe (kde binárna hodnota je dekodovaná na celé číslo), bola použitá “minimálna” reprezentácia, keď  $l_i$  malo najmenšiu možnú dĺžku – 2 bity pre rozdelenie do štyroch skupín a 3 pri rozdeľovaní do piatich skupín.

Z tabuľky vidno, že pri niektorých reprezentáciách zväčšenie populácie neprineslo adekvátny efekt v podobe zníženia počtu vyhodnotení, potrebných pre nájdenie riešenia. Aj keď došlo k zmenšeniu celkového počtu generácií, kvôli nutnosti spracovávať viac jedincov v každej generácii celkový počet generovaných jedincov vzrástol.

Binárna reprezentácia (redundantná)	Variant nastavenia		
	4 skupiny	4 skupiny (+populácia)	4 skupiny (+tlak)
nerovnomerná	27 350	46 800	27 950
rovnomerná	21 000	39 900	24 400
prekódovaná	11 800	8 200	6 050

Tabuľka 4.2: Priemerné počty generovaných jedincov potrebných pre nájdenie riešenia pri použití rôznych spôsobov dekodovania pri binárnej redundantnej reprezentácii

S výnimkou permutačnej reprezentácie, výkon algoritmu príliš nekolíše, pomer najlepšieho a najhoršieho výkonu sa pohybuje okolo 1:2 v prvej dvojici stĺpcov, zatiaľ čo v druhej dvojici sa rozdiely medzi nimi ešte znižujú. Dramatická zmena sa objaví, ak sa do porovnania zahrnie aj permutačná reprezentácia. Pre ňu je riešený problém príliš ľahký, a riešenie v drvivej väčšine prípadov (84%-100%) bolo dekodované už z náhodne vytváranej 0-tej generácie, bez toho aby sa evolučný proces hľadania vôbec rozbehol.

Pre ilustráciu vplyvu redundancie pri binárnom kódovaní na celkový výkon algoritmu bola použitá väčšia hodnota  $l_i$  pre kódovanie skupiny, do ktorej bol zaradený nejaký štát. Získané dáta sú zobrazené v tabuľke 4.2. Táto “redundantná” forma používa  $l_i$  rovné hodnote 4, pričom vo všetkých bol použitý číselný spôsob dekodovania. Najľahšie riešiteľný variant nastavenia pre delenie štátov do piatich skupín nebol použitý.

Pri nerovnomernej reprezentácii<sup>9</sup> bolo binárne číslo, reprezentované na štyroch pozíciách, dekodované na skupiny tak, že tri skupiny mali každá po jednej reprezentácii, zatiaľ čo ostatné reprezentácie boli dekodované na štvrtú skupinu (ktorej teda zodpovedalo 13 rôznych reprezentácií). Táto nerovnomernosť sa zreteľne prejavila na dosiahnutom výkone.

Pri rovnomernej reprezentácii bolo cieľom poskytnúť každej skupine rovnaký počet reprezentácií. Aby každá z nich mala štyri rôzne reprezentácie, bolo použité dekodovanie podľa vzťahu (4.7). Keďže intuitívne by mali byť získané výsledky (priemerný počet generovaných jedincov) porovnateľné s “minimálnou” reprezentáciou pri  $l_i = 2$ , hodnoty uvedené v tabuľke sú viac než prekvapivé.

Je ich však možné vysvetliť pomerne jednoducho – ako pomerne nešťastnú zhodu, že počet reprezentovaných skupín je mocninou čísla 2. Pri

---

<sup>9</sup>Pri nerovnomernej reprezentácii náhodné vzorkovanie asymetricky preferuje niektoré riešenia na úkor iných.

kód	skupina	kód	skupina	kód	skupina	kód	skupina
0000	1	0100	4	1000	3	1100	2
0001	2	0101	3	1001	4	1101	1
0010	3	0110	2	1010	1	1110	4
0011	4	0111	1	1011	2	1111	3

Tabuľka 4.3: Použité prekódovanie celých čísel

danom dekódovaní napríklad skupine 2 zodpovedajú tieto štyri reprezentácie

0 0 0 1    0 1 0 1    1 0 0 1    1 1 0 1

Pri zmenách materiálu, z ktorého sú jedince zostavené, sa veľmi často mení iba hodnota na jednej pozícii, zatiaľ čo hodnoty na ostatných pozíciách nejakého úseku, reprezentujúceho nejaký objekt, ostávajú nezmenené. A pri danej reprezentácii zmena na prvej alebo druhej pozícii síce zmení reprezentáciu, avšak nie skupinu, na ktorú je daná reprezentácia dekódovaná. Inak povedané, evolučný algoritmus polovicu svojho času venoval činnosti, z ktorej nebol adekvátny efekt<sup>10</sup>. Navyše, zmenou na tretej alebo štvrtej pozícii sa skupina 2 zmení na skupinu 1 alebo 4, avšak nie na skupinu 3.

Preto bolo použité upravené dekódovanie, kde binárne kódované číslo bolo dekódované na číslo skupiny podľa tabuľky 4.3. Výhodou použitého prekódovania je jednak to, že zmena na jednej pozícii má za následok vždy aj zmenu výslednej skupiny, a jednak to, že každá skupina sa zmenou nejakej pozície môže zmeniť na ľubovoľnú inú skupinu.

Uvedený príklad ilustruje fakt, že voľba spôsobu reprezentácie, vhodnej pre riešenie určitého konkrétneho problému, dokáže ovplyvniť výkon výsledného algoritmu. Z toho dôvodu čas venovaný tejto voľbe je strávený zmysluplne. Aj keď na prvý pohľad “samozrejmé” riešenie (v tomto prípade to je mnohoznačná reprezentácia) nie je najhoršie, neznamená to že neexistuje od neho voľba výhodnejšia.

<sup>10</sup>Je potrebné varovať pred generalizáciou, pretože pre nejaký iný riešený problém daný spôsob dekódovania môže byť vyhovujúci.





## Kapitola 5

# Vhodnosť

Kandidáti na riešenie predstavujú jednotlivé alternatívy pre hľadané riešenie nejakého problému. O tom, ktorý z nich je alebo nie je skutočným riešením, rozhoduje účelová funkcia, ktorá vlastne zosobňuje vlastnosti, kladené na hľadané riešenie. Je ju možné chápať aj ako odchýlku, chybu, ktorá bola dosiahnutá pri riešení zadaného problému v konkrétnych jednotlivých prípadoch. Je vždy udávaná v hodnotách a pojmoch prirodzených v doméne, do ktorej patrí riešený problém.

Na druhej strane evolučný algoritmus používa vhodnosť jedincov (vo všeobecnosti určovanú ako vhodnosť tých kandidátov na riešenie, ktorí sú reprezentovaní týmito jedincami) pre riadenie resp. ovplyvňovanie prehľadávania bodov v priestore prehľadávania. Algoritmus v nejakom čase (generácii) má informáciu o dvoch množinách bodov na ploche vhodnosti. Jedna množina je reprezentovaná jedincami aktuálnej populácie, zatiaľ čo druhá množina bodov má tvar novo vygenerovaných potomkov. Na základe tejto dostupnej informácie sa algoritmus snaží identifikovať tie oblasti plochy vhodnosti (alebo smer k týmto oblastiam), ktoré by mal v ďalšom preskúmať.

S ohľadom na funkciu vhodnosti ako jediný dostupný informačný zdroj, by “priateľská” plocha vhodnosti preto mala mať vlastnosti:

- mala by reprezentovať čo “najhladší” terén bez rozoklaných pohorí, strmých úbočí, úzkych a hlbokých roklín<sup>1</sup> pre jednoduchý pohyb po tomto teréne,
- nemala by obsahovať veľké plošiny, kde body majú rovnakú vhodnosť a teda sa nedá nijako odhadnúť smer, sledovaním ktorého sa dá dostať

<sup>1</sup>Ideálnou je unimodálna funkcia s jedným extrémom bez výskytu lokálnych extrémov – pre takýto prípad sú však k dispozícii jednoduchšie metódy ako sú evolučné algoritmy.

k lepším riešeniam.

Keďže funkcia vhodnosti je vlastne jediným zdrojom, ktorý evolučnému algoritmu poskytuje informácie pre efektívny pohyb v priestore prehľadávania, tak nie každý jej tvar je rovnako vhodný. Rozdiel medzi lepším a horším návrhom (viac či menej priateľským) funkcii vhodnosti môže rozhodnúť o tom, či algoritmus bude úspešný pri riešení nejakého problému alebo nie.

Podobne ako pri návrhu spôsobu reprezentácie, ani pri návrhu funkcie vhodnosti, napriek dôležitosti kladenej na danú voľbu, neexistuje všeobecne použiteľný návod (okrem všeobecných rád a zoznamu tvarov funkcie vhodnosti, ktoré boli úspešné pri riešení konkrétnych problémov v minulosti), ako voliť jej správny tvar pre nejaký konkrétny problém.

Zdá sa prirodzeným, aby konštrukcia funkcie vhodnosti bola priamo založená na tvare účelovej funkcie. Jednou takou možnosťou je postup zostrojenia funkcie vhodnosti priamo z hodnôt účelovej funkcie. Tento postup je založený na predpoklade, že účelová funkcia nadobúda hodnoty z reálneho intervalu  $\langle U_{min}, U_{max} \rangle$ . Zakladá sa na jednoduchom lineárnom zobrazení, ktoré maximálnu a minimálnu hodnotu účelovej funkcie zobrazí na minimálnu a maximálnu hodnotu funkcie vhodnosti, zatiaľ čo ostatné funkčné hodnoty sú lineárne<sup>2</sup> interpolované medzi týmito dvoma limitnými hodnotami.

V prípade minimalizačnej úlohy je možné postupovať obdobne a dané zobrazenie realizovať podľa [24]

$$\Phi(a_i) = \frac{\Phi_{max} - \Phi_{min}}{U_{min} - U_{max}} U(a_i) + \frac{U_{min} \Phi_{min} - U_{max} \Phi_{max}}{U_{min} - U_{max}} \quad (5.1)$$

kde  $\Phi_{min}$  a  $\Phi_{max}$  zastupujú minimálnu a maximálnu hodnotu vhodnosti a  $U(a_i)$  a  $\Phi(a_i)$  reprezentujú hodnotu účelovej funkcie a vhodnosť kandidáta na riešenie, reprezentovaného jedincom  $a_i$ .

Aj keď vzťahy (3.2) a (5.1) zobrazujú proces určovania vhodnosti ako jednokrokový proces, vo všeobecnosti sa v ňom zvyčajne dajú rozpoznať dve fázy (aj keď v konkrétnom prípade obe môžu splývať do jednej). Preto je vhodnejšie ho znázorniť v tvare

$$\Phi : \mathcal{M}_c \longrightarrow \mathfrak{R} \longrightarrow \mathfrak{R} \quad (5.2)$$

Výsledkom prvej fázy je určenie hodnoty vhodnosti<sup>3</sup>, zatiaľ čo v druhej fáze sa táto vhodnosť transformuje na štandardizovanú vhodnosť. Vhodnosť býva

<sup>2</sup>Nič nebráni pokusom s nelineárnou interpoláciou hodnôt.

<sup>3</sup>Niekedy sa táto hodnota označuje ako "surová" vhodnosť.

nejakým vhodným spôsobom odvodená z účelovej funkcie a je to vlastne reálne číslo, určujúce akým dobrým riešením je ten-ktorý kandidát na riešenie.

Štandardizovaná vhodnosť je tiež reprezentovaná reálnym číslom a voči vhodnosti má navyše aj vlastnosti, ktoré umožňujú jej jednoduché použitie evolučným algoritmom. Z týchto vlastností to je najmä to, že lepšie riešenia majú vyššiu hodnotu vhodnosti než riešenia horšie – teda transformácia optimalizačného problému na maximalizačný. To je možné napríklad pri minimalizačnom probléme pomocou vzťahov

$$\frac{k_1}{k_2 + \Phi(a_i)} \quad , \quad k_3 - \Phi(a_i) \quad (5.3)$$

kde  $\Phi(a_i)$  reprezentuje vhodnosť  $i$ -tého riešenia a hodnoty  $k_i$  reprezentujú konštanty (pričom  $k_2$  je volené tak, aby nedošlo k deleniu nulou).

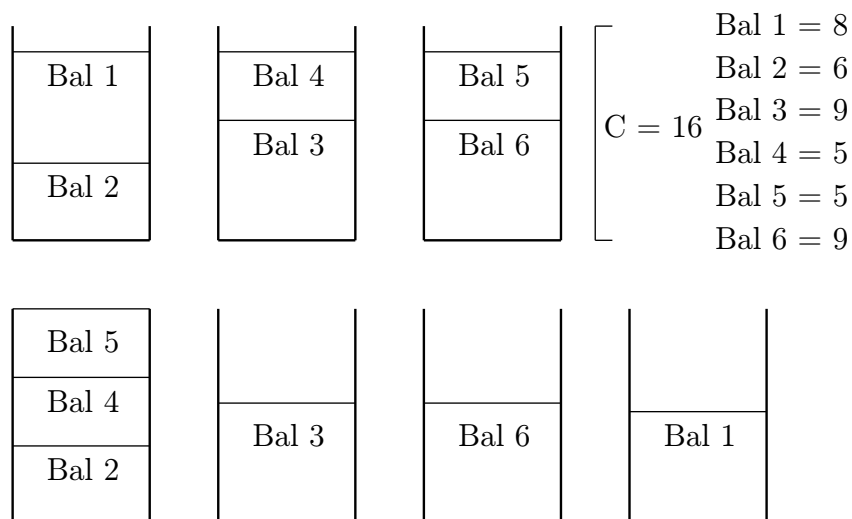
V menšej miere sa vyskytujú aj požiadavky na to, aby hodnoty vhodnosti neboli záporné alebo aby ležali v preddefinovanom intervale možných hodnôt, resp. aby boli nejakým spôsobom normalizované. Keďže v evolučnom algoritme sa používa práve štandardizovaná vhodnosť, tak v ďalšom texte sa pre jednoduchosť bude používať iba vhodnosť, zatiaľ čo pod ňou sa bude chápať štandardizovaná vhodnosť.

## 5.1 Aproximatívna vhodnosť

Z pohľadu praktického použitia evolučného algoritmu pre riešenie nejakého konkrétneho problému je niekedy vhodnejšie použiť funkciu vhodnosti, ktorá je skôr aproximáciou než priamym vyjadrením účelovej funkcie. V podstate pre to existujú dva dôvody (okrem očividného dôvodu, keď použitie účelovej funkcie je nemožné napríklad pre náročné požiadavky na použité zdroje):

- určovanie hodnôt vhodnosti využívajúce účelovú funkciu je zaťažené prítomnosťou veľkej dávky šumu,
- použitie vhodnosti, vyjadrujúcej účelovú funkciu, vedie na pre evolučný algoritmus nepríjemnú plochu vhodnosti.

Pri prvom type je zvyčajne vytváraný nejaký model (napríklad v tvare umelej neurónovej siete), ktorý aproximuje účelovú funkciu. Na základe tohto modelu je potom určovaná aproximatívna vhodnosť. Buď je táto vhodnosť následne používaná samostatne pre riadenie pohybu evolučného algoritmu



Obr. 5.1: Optimálne (hore) a suboptimálne riešenie (dole) baliaceho problému

po ploche vhodnosti alebo je používaná v kombinácii s vhodnosťou určovanou priamo bez sprostredkovania modelom.

Príkladom je prístup podľa [18], ktorý väčšinou používa aproximatívnu vhodnosť a iba malú časť jedincov vyhodnocuje prostredníctvom (zašumenej) neaproximatívnej vhodnosti. Jedince, ktorých vyhodnotenie je založené priamo na účelovej funkcii bez jej sprostredkovania prostredníctvom modelu, tvoria buď iba malú časť populácie v každej generácii alebo s nejakou periódou vytvárajú celú populáciu. Dôvodom pre používanie aj zašumenej neaproximatívnej vhodnosti je snaha obmedziť konvergenciu algoritmu do falošných extrémov (extrémov zavedených vytvoreným modelom avšak neprítomných v pôvodnej účelovej funkcii).

Príkladom druhého typu môže byť riešenie baliaceho problému [12]. Ten je vo svojej najjednoduchšej definícii daný nasledovne: “Je daná množina balíkov rôznej veľkosti a sada rovnakých kontajnerov. Balíky je potrebné uložiť do kontajnerov tak, aby pre zabalenie všetkých balíkov bolo použitých čo najmenej kontajnerov. Úlohou je zistiť minimálne potrebný počet kontajnerov.” Jednorozmerná verzia tohto problému je ilustrovaná na obr. 5.1 so šiestimi balíkmi, pre zabalenie ktorých sú potrebné aspoň tri kontajnery.

Na prvý pohľad sa zdá, že vhodnosť riešení by mala korešpondovať s počtom použitých kontajnerov<sup>4</sup>, rovnako ako je definovaná účelová funkcia.

<sup>4</sup>S vhodnou úpravou pre transformáciu minimalizačného problému na maximalizačný, napr. počet všetkých balíkov zmenšený o počet použitých kontajnerov.

Avšak takáto funkcia vhodnosti je nepriateľská pre praktické použitie, pretože popri malom počte optimálnych riešení existuje veľmi mnoho suboptimálnych riešení, ktoré by mali rovnaké vhodnosti. Plocha vhodnosti by mala tvar veľkých plošín nad ktorými je roztrúsených niekoľko bodov reprezentujúcich optimálne riešenia. Takáto plocha vhodnosti by mala slabú schopnosť viesť algoritmus a prehľadávanie by malo charakter hľadania ihly v kope sena.

Preto je vhodné uvažovať o aproximatívnej funkcii vhodnosti, ktorá by podobným riešeniam, vyžadujúcim rovnaký počet kontajnerov (z hľadiska účelovej funkcie sa jedná o rovnako dobré resp. zlé riešenia), priradzovala podobné vhodnosti – avšak nie rovnaké! Inak povedané, pri rovnako dobrých riešeniach by sa jedno z nich “umelo” preferovalo na úkor iného. Týmto spôsobom by plocha vhodnosti neobsahovala nepríjemné plošiny resp. by boli menšie.

Namiesto uvažovania počtu kontajnerov možno brať do úvahy ich naplnenosť, ktorú pre  $i$ -ty kontajner možno určiť ako pomer  $V_i/V$ , kde  $V_i$  reprezentuje zaplnenú časť kontajnera a  $V$  zase jeho kapacitu. Čím je viac kontajnerov, tým ich priemerná naplnenosť bude menšia. Najväčšiu hodnotu dosiahne práve hľadané optimálne riešenie. Avšak namiesto priemernej naplnenosti je možné zohľadňovať aj spôsob naplnenosti jednotlivých kontajnerov. V prípade dvoch kontajnerov je riešenie, keď jeden z nich je skoro plný a druhý skoro prázdny, považované za lepšie ako riešenie, pri ktorom sú oba kontajnery naplnené rovnako do polovice. Je to preto, lebo skoro prázdny kontajner dokáže ľahšie prijať ďalší balík než kontajner naplnený do polovice (pretože dokáže prijať aj balíky, ktoré sa už nezmestia do spolu naplneného kontajnera). Toto je možné vyjadriť ako

$$\frac{\sum_{i=1}^n (V_i/V)^k}{n} \quad (5.4)$$

kde  $n$  označuje počet použitých kontajnerov a  $k$  zase stupeň uvažovania naplnenosti kontajnerov. Pre hodnotu  $k = 1$  sa vhodnosť stáva úmerná  $1/n$ , čím priamo reflektuje účelovú funkciu. Čím má  $k$  väčšiu hodnotu, tým viac sú preferované viac naplnené balíky oproti menej naplneným, a tým viac sú preferované riešenia s nerovnomerným naplnením voči riešeniam s rovnomerným naplnením kontajnerov.

Voľba hodnoty  $k$  má však vplyv na ešte jednu, dosiaľ nespomenutú, avšak požadovanú vlastnosť aproximatívnej funkcie vhodnosti. Prirodzenou požiadavkou je, aby jej globálne maximum malo rovnakú polohu ako v prípade funkcie vhodnosti priamo vychádzajúcej z účelovej funkcie (aby optimálne riešenie v oboch prípadoch bolo rovnaké – teda aby sme v oboch prípadoch

hľadali to isté riešenie). A to pri veľkých hodnotách parametra  $k$  neplatí. V extrémnom prípade  $k \rightarrow \infty$  sa vhodnosť stáva úmernou  $\max(V_i/V)$ , kedy maximálna vhodnosť je priradená konfigurácii s aspoň jedným plne zaplneným kontajnerom bez ohľadu na to, koľko ďalších (hoci aj temer prázdnych) kontajnerov je použitých. Pri veľkých hodnotách parametra by teda mohlo dôjsť k situácii, keď je uprednostnené riešenie s väčším počtom kontajnerov oproti riešeniu vyžadujúcemu menší počet kontajnerov<sup>5</sup>.

## 5.2 Súťažná vhodnosť

Funkcia vhodnosti v tvare, v akom bola uvažovaná v predchádzajúcej časti, reprezentuje úplné usporiadanie v priestore kandidátov a jej hodnota pre nejakého konkrétneho kandidáta môže byť chápaná ako odkaz do tohto usporiadania. No môžu existovať prostredia, kde takáto funkcia je obtiažne vypočítateľná (určenie vhodnosti vyžaduje príliš mnoho zdrojov) alebo kde úplné usporiadanie kandidátov neexistuje.

Príkladom prostredia bez úplného usporiadania je prípad troch stratégií, kde prvá poráža druhú, druhá poráža tretiu a tretia zase poráža prvú. Riešením nie je stanovenie nejakého poradia medzi stratégiami, pretože by vždy existovala dvojica stratégií, ktorých sila by bola nesprávne ohodnotená. A riešením nie je ani stanovenie rovnakej vhodnosti pre všetky stratégie, pretože pri výskyte iba dvoch z nich ich sila opäť nie je adekvátne vyjadrená.

Pre tieto prípady vo funkcii vhodnosti možno použiť súťažnú vhodnosť [5]. Je to typ relatívnej vhodnosti poskytujúcej parciálne usporiadanie, ktorej hodnoty závisia na riešeniach, ktoré sú aktuálne k dispozícii. K jej výpočtu stačí, aby pre dvojicu súťažiacich kandidátov bolo možné rozhodnúť, ktorý z nich je lepším riešením.

Existujú tri spôsoby určovania hodnôt súťažnej vhodnosti. Sú to tieto súťaže: *plná súťaž*, *bipartitná súťaž* a *turnaj*. Plná súťaž a turnaj sú určené pre vzájomné súťaženie medzi jedincami populácie. Bipartitná súťaž môže byť použitá rovnakým spôsobom, ale aj v prípade, keď jedince populácie súťažajú s jedincami z prostredia (čo môže byť nejaká pevne daná množina súťažiacich alebo to môžu byť jedince vyvíjané iným evolučným algoritmom<sup>6</sup>).

Pri plnej súťaži každý účastník súťaží s každým iným účastníkom. Keďže

---

<sup>5</sup>V príklade podľa obr. 5.1 pre uprednostnenie dolného (suboptimálneho) riešenia pred horným (optimálnym) stačí pre  $k$  zvoliť ľubovoľnú hodnotu spĺňajúcu nerovnicu  $\sqrt[k]{4} < 8/7$ .

<sup>6</sup>Toto sa označuje ako súťažná koevolučná schéma, pri ktorej dochádza k interakcii (avšak nie k výmene materiálu) v rámci evaluačného bloku algoritmu.

jeho vhodnosť je potom daná ako počet duelov, z ktorých vyšiel víťazne, tak vhodnosť každého účastníka bude celým číslom z množiny  $\{0, 1, 2, \dots, n-1\}$ , kde  $n$  reprezentuje počet všetkých súťažiacich. Počet všetkých duelov, ktoré je potrebné zrealizovať, je  $n(n-1)/2$ . Pre zníženie počtu duelov jedinec nemusí súťažiť voči všetkým ostatným, ale iba voči určitej podmnožine náhodne vybraných jedincov.

Pri bipartitnej súťaži existujú dva tímy a súťaž je organizovaná medzi nejakým účastníkom z jedného tímu a všetkými súťažiacimi z tímu druhého. Ak účastníci sú rozdelení na rovnako veľké tímy, tak vhodnosť každého bude číslo z množiny  $\{0, 1, \dots, n/2\}$  a počet všetkých duelov je  $(n/2)^2$ . Pre zníženie počtu duelov je možné, aby súťažiaci súperil iba s podmnožinou druhého tímu (ktorá môže byť opäť vybraná náhodným spôsobom).

Turnajová súťaž je organizovaná ako eliminačný turnaj. Vždy súťažia dvaja súťažiaci, pričom víťaz postupuje do ďalšej súťaže, zatiaľ čo porazený vypadáva. Keďže rôzni súťažiaci absolvujú rôzny počet duelov, tento spôsob je vhodný najmä na určenie jedného víťaza. Výhodou je pokles potrebných duelov na  $n-1$ .

Nevýhodou súťažnej vhodnosti (okrem času potrebného na absolvovanie potrebného počtu duelov) je to, že pri zmene skupiny súťažiacich môže dochádzať aj k zmene vhodnosti – a teda v každej generácii je potrebné opakovane určovať vhodnosť aj tých jedincov, ktoré sa vyskytovali už v predchádzajúcich generáciách.

### 5.3 Vhodnosť založená na komplexnosti

Tento typ vhodnosti sa používa pri problémoch, keď je zadaná nejaká množina vstupných dát a je potrebné vyvinúť model, ktorý by tieto dáta vysvetľoval (napríklad je potrebné vytvoriť rozhodovací strom, pomocou ktorého by sa dali dáta klasifikovať rovnakým spôsobom ako je zadané). Problémom je totiž to, že nie je pevne zadaná veľkosť modelu, ale ten môže ľubovoľne rásť. V prípade, keď by ako vhodnosť bola použitá zhoda modelu s dátami, hľadanie vhodného modelu by prerástlo až do fázy preučenia – model by verne kódoval nielen všeobecné vzťahy platiace v dátach, pre ktoré je vyvíjaný, ale aj vzťahy, ktoré sú síce platné v dostupnej vzorke dát ale vo všeobecnosti neplatia.

Jedným z kritérií, založených na komplexnosti vytváraného modelu, je kritérium minimálnej dĺžky popisu, dobre známe v oblasti strojového učenia. Toto kritérium je dané ako

$$\text{dĺžka popisu( reziduálne dáta )} + \text{dĺžka popisu( model )}$$

pričom cieľom je vytvoriť model, ktorý hodnotu tohto kritérium bude minimalizovať.

Vo všeobecnosti, model niektoré dáta vysvetľuje správne, zatiaľ čo niektoré vysvetľuje chybné. Tieto chybné vysvetlené dáta vlastne reprezentujú výnimku k modelu a hrajú úlohu reziduálnych dát. Ak by model bol perfektný vzhľadom na zadané dáta a všetky vysvetľoval správne, tak by neexistovali žiadne reziduálne dáta (a popisovala by sa iba ich neprítomnosť).

Pod dĺžkou popisu sa rozumie dĺžka kódu potrebného pre zakódovanie toho-ktorého objektu, pričom pri kódovaní modelu je potrebné kódovať ako jeho štruktúru, tak aj jeho parametre. Pretože dáta a model sú rozdielne entity, pre ich zakódovanie sú potrebné rozdielne spôsoby kódovania informácií. A v tom spočíva úskalie tejto metódy – voľba rovnako efektívnych kódov pre obe časti. Ak by totiž kód pre kódovanie modelu bol efektívnejší ako kód pre dáta, tak by bol vyvíjaný zbytočne veľký model. Pri opačnom pomere efektívnosti kódov by zase bol preferovanejší model menší, až v limitnom prípade by vyhrával model žiadny resp. triviálny model s neakceptovateľne veľkou chybou.

## 5.4 Kombinovanie vhodností

Je možné kombinovať dve alebo viaceré funkcie vhodnosti do jednej výslednej. Pritom je možné skúšať rôzne prístupy k tvorbe výslednej funkcie vhodnosti, či už všeobecné alebo ad-hoc prístupy.

Jednou z všeobecne použiteľných možností je kombinácia majoritnej a minoritnej vhodnosti. Pri tomto prístupe sa uvažujú dve vhodnosti  $\Phi_I$  a  $\Phi_J$ . Ak  $\Phi_I$  je považovaná za majoritnú, tak potom pre ľubovoľné dva jedince  $a_i$  a  $a_j$ ,  $i \neq j$  bude platiť

$$\begin{aligned} \Phi_I(a_i) < \Phi_I(a_j) &\implies \Phi_{IJ}(a_i) < \Phi_{IJ}(a_j) \\ \Phi_I(a_i) > \Phi_I(a_j) &\implies \Phi_{IJ}(a_i) > \Phi_{IJ}(a_j) \end{aligned} \quad (5.5)$$

kde  $\Phi_{IJ}$  je výsledná funkcia vhodnosti vzniknutá kombináciou dielčích funkcií  $\Phi_I$  a  $\Phi_J$ .

Inak povedané, ak majoritná vhodnosť dokáže určiť, ktorý z dvoch jedincov je vhodnejší, tak minoritná vhodnosť toto poradie nesmie zmeniť. Iba v prípade, že majoritná vhodnosť oboch jedincov považuje za rovnako vhodné riešenia, bude výsledné poradie jedincov určené prostredníctvom minoritnej vhodnosti.

Uvedeným spôsobom možno jednoducho kombinovať napríklad také dve vhodnosti, kde majoritná vhodnosť nadobúda hodnoty z oboru celých čísel,



zatiaľ čo obor funkčných hodnôt minoritnej vhodnosti je interval  $(0, 1)$ , ktorý je zľava uzavretý a sprava otvorený. V tomto prípade je možné výslednú vhodnosť jednoducho vytvoriť ako súčet týchto dielčích vhodností.

Aby mohol byť daný prístup úspešne použitý, musia byť splnené nasledujúce podmienky:

- obe vhodnosti musia byť založené na rôznych princípoch,
- majoritná funkcia vhodnosti mnohým riešeniam priraduje rovnakú vhodnosť.

Ak by obe funkcie vhodnosti vyjadrovali to isté, ich kombináciou nedôjde k výraznému zlepšeniu, pretože v prípade, že nevie zmysluplne rozhodnúť jedna funkcia vhodnosti, nedokáže to ani druhá. A ak by neplatila druhá podmienka, tak isto by neprišlo k zlepšeniu, pretože minoritná vhodnosť by nemala dostatočný priestor, v ktorom by sa mohla prejaviť.

Na druhej strane, je možné v úlohe minoritnej vhodnosti použiť aj takú funkciu, ktorej globálne maximum nie je považované za riešenie daného problému, a jej použitie samostatne je neprijateľné. To preto, lebo úlohou minoritnej funkcie vhodnosti nie je do viesť evolučný algoritmus k hľadanému riešeniu, ale pôsobiť iba lokálne v určitom podprieštore (kde plocha vhodnosti tvorená majoritnou funkciou vhodnosti má tvar neinformatívnej plošiny) a vyviesť algoritmus z tohto podprieštora vhodným<sup>7</sup> smerom.

## 5.5 Ilustrácia alternatívnych vhodností

Pre ilustráciu použitia rozličných spôsobov určovania vhodností jedincov použijeme ten istý problém ako pri ilustrácii alternatívnych reprezentácií. Opäť je potrebné štáty Afriky zaradiť do skupín tak, aby žiadne dva susediace štáty neboli v tej istej skupine.

Aj keď v tomto prípade sa prezentované algoritmy líšia iba spôsobom výpočtu vhodnosti a ostatné časti majú rovnaké, cieľom nie je určiť, ktorý spôsob určovania vhodnosti je lepší a ktorý horší. Skôr toto porovnanie je potrebné chápať ako ilustráciu, že pre jeden a ten istý problém možno navrhnúť rozličné tvary funkcie vhodnosti a že tento návrh môže ovplyvniť celkovú výkonnosť algoritmu.

Všetky použité tvary funkcie vhodnosti priradujú lepším riešeniam vyššie hodnoty, takže už nebolo nutné hodnoty vhodnosti ďalej standardizovať. Boli použité nasledovné spôsoby:

<sup>7</sup>Nesprávne zvolená minoritná funkcia vhodnosti môže aj spomaliť či znemožniť hľadanie riešenia.

**Počet splnených ohraničení ( $\Phi_1$ ).** Základom tohto spôsobu určenia vhodnosti sú dvojice susediacich štátov, pričom každá z nich je považovaná za ohraničenie, ktorého splnenie sa vyžaduje. Postupne je kontrolovaná každá takáto dvojica. Testuje sa, či hodnotené riešenie priradilo každý zo štátov danej dvojice do rôznych skupín, a teda či dané ohraničenie je splnené. Riešeniu sa ako jeho vhodnosť priradí počet splnených ohraničení.

Funkcia vhodnosti nadobúda hodnoty z  $\{0, 1, \dots, ND\}$ , kde  $ND$  označuje počet všetkých susediacich dvojíc.

**Počet správne zaradených štátov ( $\Phi_2$ ).** Základom tohto spôsobu sú štáty, ktoré je potrebné zaradiť do jednej zo skupín. Za správne zaradený štát sa považuje taký, ktorý so žiadnym svojim susedom nezdieľa rovnakú skupinu. Postupne je kontrolovaný každý štát, pričom sa testuje, či jeho zaradenie spĺňa všetky ohraničenia. Riešeniu sa ako jeho vhodnosť priradí počet správne zaradených štátov.

Funkcia vhodnosti nadobúda hodnoty z  $\{0, 1, \dots, NS\}$ , kde  $NS$  označuje počet všetkých štátov.

**Vážený počet správne zaradených štátov ( $\Phi_3$ ).** Tento spôsob je rozšírením  $\Phi_2$ , ktorý každý správne zaradený štát uvažoval s váhou rovnou 1. Na rozdiel od neho, teraz sa rozlišuje, či štát nemá žiadneho suseda, má málo susedov alebo má veľa susedov. Správne zaradenie štátu s veľkým počtom susedov je považované za cennejšie ako správne zaradenie štátu s menším počtom susedov. Riešeniu sa ako vhodnosť priradí súčet váh správne zaradených štátov, kde pod váhou štátu sa rozumie počet jeho susedov.

Funkcia vhodnosti nadobúda hodnoty z  $\{0, 1, \dots, 2ND\}$ .

**Vážený počet nesprávne zaradených štátov ( $\Phi_4$ ).** Podobne ako v prípade  $\Phi_3$  aj teraz sa zohľadňujú váhy štátov. Dôraz je však kladený na nesprávne zaradenie štátov, pričom za vážnejšie sa považuje, ak štát je v rovnakej skupine s viacerými zo svojich susedov, ako keby bol v rovnakej skupine iba s malým percentom svojich susedov (v ideálnom prípade toto percento je nulové). Váha štátu je určená ako pomer počtu jeho susedov, s ktorými zdieľa skupinu, a celkového počtu jeho susedov (ak štát nemá susedov, tak jeho váha je nulová, rovnako ako váha správne zaradeného štátu). Riešeniu sa priradí celkový počet štátov zmenšený o súčet váh štátov.

Funkcia vhodnosti nadobúda hodnoty z  $\{NN, \dots, NS\}$ , kde  $NN$  označuje počet štátov, ktoré nemajú susedov.

**Počet splnených ohraňení a správne zaradených štátov ( $\Phi_5$ ).** Je to kombinácia  $\Phi_1$  a  $\Phi_2$  zohľadňujúca súčasne spĺňanie ohraňení aj správnosť zaradovania štátov. Riešeniu sa ako vhodnosť priradí pomer počtu splnených ohraňení a nesprávne zaradených štátov v tvare  $\Phi_1/(NS - \Phi_2 + \varepsilon)$ .

Funkcia vhodnosti nadobúda hodnoty z  $\{0, \dots, ND/\varepsilon\}$ , kde  $\varepsilon$  označuje kladné číslo menšie ako 1.

**Plná súťaž ( $\Phi_6$ ).** Základom tohto spôsobu je vzájomné porovnávanie jedincov v populácii. Každé riešenie je porovnávané so všetkými ostatnými riešeniami. Ako porovnávacie kritérium bol použitý počet splnených ohraňení ( $\Phi_1$ ). Riešeniu sa priradí súčet výsledkov jednotlivých duelov, pričom za vyhratý duel sú 2 body, za remízu 1 bod a za prehru nie je žiadny bod.

Funkcia vhodnosti nadobúda hodnoty z  $\{0, 1, \dots, 2(\mu - 1)\}$ , kde  $\mu$  označuje počet jedincov v populácii.

**Náhodná súťaž ( $\Phi_7$ ).** Modifikácia spôsobu  $\Phi_6$  s tým, že počet duelov  $NZ$  je pre každého jedinca obmedzený na malý počet (bola použitá hodnota 5) a súper sú pre každého jedinca vyberaní náhodným spôsobom.

Funkcia vhodnosti nadobúda hodnoty z  $\{0, 1, \dots, 2NZ\}$ , kde  $NZ$  označuje počet zápasov, ktoré každý jedinec absolvuje.

Z hľadiska globálneho extrému sú funkcie  $\Phi_1 - \Phi_5$  rovnocenné. Ak nejaké riešenie je globálnym maximom v jednej funkcii, tak je súčasne globálnym maximom aj v ostatných. Iná situácia je však pri funkciách  $\Phi_6$  a  $\Phi_7$ . Maximálnu hodnotu vhodnosti tu môže dosiahnuť aj jedinec, ktorý z hľadiska funkcií  $\Phi_1 - \Phi_5$  nie je globálnym extrémom (stačí, ak jeho súper budú horšími riešeniami ako on sám).

Iná situácia je v oblasti lokálnych extrémov. Výsledkom je, že pre dve vhodnosti  $\Phi_I$  a  $\Phi_J$ ,  $I \neq J$  môžu existovať dva jedince  $a_i$  a  $a_j$ ,  $i \neq j$ , pre ktoré bude súčasne platiť

$$\Phi_I(a_i) < \Phi_I(a_j) \quad a \quad \Phi_J(a_i) > \Phi_J(a_j) \quad (5.6)$$

teda že dve rôzne funkcie vhodnosti vyhodnotia vzťah dvoch rôznych jedincov opačným spôsobom.

Príkladom<sup>8</sup> je  $\Phi_2$ , ktoré uprednostní správne zaradenie väčšieho počtu štátov vystupujúcich v nejakom počte susedských vzťahov oproti správne zaradeniu menšieho počtu štátov figurujúcich vo väčšom počte susedských vzťahov, zatiaľ čo názor  $\Phi_3$  je presne opačný.

<sup>8</sup>Objavenie všetkých takýchto dvojíc je ponechané na čitateľa.

$\Phi$	Variant nastavenia			
	4 skupiny	4 skupiny (+populácia)	4 skupiny (+tlak)	5 skupín
1	8 400	8 500	6 600	2 050
2	8 300	8 700	6 750	2 000
3	4 400	4 500	3 350	1 750
4	13 500	22 300	12 850	2 750
5	7 850	8 200	6 450	2 000
6	7 900	8 900	6 850	2 100
7	9 200	10 500	7 359	2 450

Tabuľka 5.1: Priemerné počty generovaných jedincov potrebných pre nájdenie riešenia pri použití rôznych spôsobov určovania vhodnosti

Je potrebné si uvedomiť, že v prípade  $\Phi_6$  a  $\Phi_7$  vhodnosť jedinca nie je pevne daná ale sa môže meniť v rôznych generáciách. Preto podmienka pre ukončenie evolučného procesu nemôže byť založená na sledovaní maximálnej vhodnosti v populácii, pretože výskyt maximálnej hodnoty vhodnosti neznamená, že sa našlo hľadané riešenie, a naopak, to hľadané riešenie sa môže vyskytovať v populácii aj keď v nej nebola detekovaná maximálna hodnota vhodnosti (ak sa v populácii nachádza viacnásobne a súťaží aj samo so sebou s nerozhodným výsledkom).

Alternatívne spôsoby určovania vhodnosti boli testované pri rovnakých štyroch nastaveniach ako v prípade alternatívnych reprezentácií. Priemerné hodnoty počtu vyhodnotení, potrebných pre nájdenie hľadaného riešenia, sú uvedené v tabuľke 5.1, kde každej vhodnosti je venovaný jeden riadok.

Výsledky ukazujú, že nie je rozdiel v tom, či je funkcia vhodnosti založená na počtoch susediacich dvojíc alebo samotných štátov ( $\Phi_1$  a  $\Phi_2$ ). Malý rozdiel preferujúci samotné štáty pred dvojicami nie je signifikantný.

Použitie vážených súčtov pred neváženými bolo vedené snahou o vzájomné odlišenie prípadov, ktoré pri použití nevážených súčtov boli ohodnotené rovnakou vhodnosťou. Cieľom bolo poskytnúť algoritmu dodatočnú informáciu o tom, ktorým smerom sa má pohybovať po tých častiach plochy vhodnosti, ktoré by v prípade použitia nevážených súčtov mali tvar plošín s rovnakou vhodnosťou. Ako ukazujú výsledky dosiahnuté v prípadoch  $\Phi_3$  a  $\Phi_4$ , konečným výsledkom môže byť značné zlepšenie ale aj zhoršenie výkonnosti hľadania.

Pri kombinovaní viacerých kritérií do jedného podľa  $\Phi_5$  nebol pozorovaný podstatný nárast výkonnosti, teda kombináciou dvoch priemerných kritérií

Majoritná vhodnosť	Minoritná vhodnosť				
	$\Phi_1$	$\Phi_2$	$\Phi_8$	$\Phi_8 - opak$	$\emptyset$
$\Phi_1$	×	7 300	5 750	11 500	8 400
$\Phi_2$	7 650	×	6 200	10 200	8 300
$\Phi_3$	4 500	4 550	4 350	4 450	4 400

Tabuľka 5.2: Priemerné počty generovaných jedincov potrebných pre nájdenie riešenia pri použití rôznych spôsobov kombinovania vhodnosti

nevniklo nadpriemerne úspešné kritérium. Podobný výsledok sa získal ak boli v  $\Phi_5$  použité vážené sumy ktoré nahradili nevážené súčty.

Úspešnosť určovania vhodnosti jedincov súťažným spôsobom závisí na dvoch faktoroch – na počte duelov, ktoré každý jedinec absolvuje, a na kritériu, použitom pre vzájomné porovnávanie jedincov. Pokiaľ je použitý dostatočný počet duelov (prípade  $\Phi_6$ ), výsledný výkon algoritmu je podobný ako keby v úlohe funkcie vhodnosti bolo samotné použité kritérium (v tomto prípade  $\Phi_1$ ).

Pokiaľ je počet duelov menší ( $\Phi_7$  – každý jedinec absolvoval iba 5 zápasov), celková výkonnosť sa začne znižovať, pričom však toto zhoršenie nie je dramatické ani pri takom malom počte duelov ako bolo použité. Rovnaké chovanie algoritmu je aj pri použití  $\Phi_3$  v úlohe porovnávacieho kritéria, keď pre variant “4 skupiny” bola dosiahnutá hodnota 5 700 vyhodnotení.

Dokonca pri účasti každého jedinca iba v troch dueloch sa, pri použití  $\Phi_3$  ako kritéria pre porovnanie, počet potrebných vyhodnotení zvýšil na 7 500, čo je hodnota lepšia ako väčšina hodnôt v prvom stĺpci tabuľky.

Pri kombinovaní viacerých vhodností možno použiť princíp majoritnej a minoritnej vhodnosti. Dosiahnuté výsledky (iba pre jeden variant nastavenia – “4 skupiny”) sú zobrazené v tabuľke 5.2, pričom posledný stĺpec reprezentuje použitie majoritnej vhodnosti samostatne. V prípade, keď v úlohe minoritnej vhodnosti bola použitá  $\Phi_1$  ( $\Phi_2$ ), jej hodnoty boli násobené hodnotou  $1/ND$  ( $1/NS$ ) aby bol zaistený obor funkčných hodnôt, požadovaný pre minoritnú funkciu vhodnosti.

Z tabuľky vidno mierne zlepšenie pri kombinovaní vhodností  $\Phi_1$  a  $\Phi_2$  (obomi spôsobmi) voči spôsobu ich kombinácie  $\Phi_5$ . To, že zlepšenie oproti ich použitiu samostatne nie je nijako dramatické, je zapríčinené tým, že obe vhodnosti sú viac-menej založené na tom istom princípe (aj keď lokálne extrémny nemajú rovnaké) – ak sa v nejakom riešení správne zaradia ďalšie štáty, počet splnených ohraničení sa taktiež zvýši. Kombinovanie  $\Phi_1$  a  $\Phi_2$  ako minoritných vhodností s  $\Phi_3$  dokonca malo za následok zhoršenie výkonu

algoritmu, čo, napriek tomu že zhoršenie je pomerne malé, ilustruje fakt, že výsledok kombinovania nemusí byť bezpodmienečne vždy pozitívny.

Doposiaľ sa uvažovala iba vhodnosť založená na štátoch alebo dvojiciach štátov. No úvahu možno založiť aj na skupinách. Predpokladajme, že susedné štáty nejakého štátu boli zaradené do  $NG$  skupín (z maximálneho počtu  $NG_{max}$ ). Ak daný štát bol zaradený do zlej skupiny (lebo v nej už bol niektorý z jeho susedov), je ho potrebné preradiť do skupiny inej. A čím je hodnota  $NG$  menšia, tým je väčšia šanca, že sa to podarí (pretože tým je väčší počet skupín, ktoré prichádzajú do úvahy). Samozrejme, že znižovanie hodnoty  $NG$  je výhodné iba po nejakú hranicu – po prekročení tejto hranice síce daný štát bude možné jednoducho preradiť do vhodnej skupiny, avšak dôjde k porušeniu ohraničenia medzi jeho susedmi navzájom. Na tejto úvahe možno vytvoriť vhodnosť  $\Phi_8$ , ktorá je daná ako

$$\frac{1}{NSP} \sum_{j=1}^{NSP} \left( \frac{NG_{max} - NG_j}{NG_{max}} \right)^k \quad (5.7)$$

kde  $NG_j$  označuje, do koľkých skupín boli zaradení susedia  $j$ -teho štátu, zatiaľ čo  $NSP$  reprezentuje počet štátov, ktoré porušujú aspoň jedno ohraničenie. Konštanta  $k$  umožňuje nastaviť preferenciu menšej alebo väčšej variability hodnôt. Vďaka rozsahu oboru funkčných hodnôt je možné  $\Phi_8$  použiť priamo vo funkcii minoritnej funkcie vhodnosti.

Funkcia  $\Phi_8$  nemôže byť použitá samostatne, pretože jej optimalizáciou by sa znižoval počet použitých skupín, až v limitnom prípade by štáty boli spolu so všetkými svojimi susedmi zaradené do jednej jedinej skupiny.

Vzhľadom na to, že princíp tejto vhodnosti je iný ako princíp ostatných, je vhodným kandidátom do úlohy minoritnej vhodnosti. To ukazujú aj dosiahnuté výsledky v tabuľke 5.2, prezentujúcej zvýšenie výkonu algoritmu pri kombinovaní tejto vhodnosti s vhodnosťami  $\Phi_1$  a  $\Phi_2$ . V kombinácii s  $\Phi_3$  došlo iba k nesignifikantnému zlepšeniu, pretože vďaka váženiu štátov boli rôzne riešenia oveľa menej krát ohodnotené prostredníctvom  $\Phi_3$  ako rovnaké riešenia než pri použití  $\Phi_1$  a  $\Phi_2$ .

Je zaujímavé použiť vhodnosť  $\Phi_8$  v opačnom význame (v čitateli iba s  $NG_j$  namiesto  $NG_{max} - NG_j$ ), keď zvyšujúca sa hodnota vhodnosti signalizuje pokles šance pre preradenie zle zaradeného štátu do vhodnej skupiny. Ani v takomto tvare ju nemožno použiť samostatne, pretože globálne maximum, rovnako ako v predchádzajúcom prípade, nie je akceptovateľným riešením. Z tabuľky (stĺpec  $\Phi_8$  – *opak*) jasne vidieť negatívny vplyv takejto minoritnej funkcie vhodnosti – čo zároveň môže byť chápané aj ako svedectvo o robustnosti použitých majoritných funkcií vhodnosti, ktoré dokázali

doviesť algoritmus k hľadanému riešeniu napriek deštrukčnému pôsobeniu minoritnej funkcie vhodnosti.

Ilustratívny príklad jasne ukazuje, že tvar funkcie vhodnosti má podstatný vplyv na výkonnosť evolučného algoritmu (aj keď v uvedenom ilustračnom príklade to neboli rozdiely rádovo veľké, napriek tomu boli podstatné). Inak povedané, pri riešení konkrétneho problému má zmysel venovať čas hľadaniu vhodného tvaru funkcie vhodnosti, pretože prvý nápad (čo ako sa zdá byť logickým) nemusí byť nijako nadpriemerným.





---

Časť III

**Základné bloky algoritmu**



## Kapitola 6

# Selekcia

Selekcia slúži pre vytvorenie skupiny rodičov z aktuálnej populácie jedincov. Cieľom je vybrať  $\rho$  jedince do funkcie rodičov. Počet potrebných rodičov je daný počtom potomkov, ktorých je potrebné vygenerovať v danej generácii evolučného cyklu, a použitými genetickými operátormi. Aj keď vo všeobecnosti platí, že počet vybratých rodičov nemusí byť totožný s počtom jedincov v populácii, rovnako ako nemusí byť totožný s počtom potomkov ktorí budú vygenerovaní, rovnosť je veľmi častým prípadom.

To, či nejaký jedinec z aktuálnej populácie sa môže stať aj viacnásobným rodičom, závisí od pomeru veľkostí populácie a skupiny rodičov (napríklad v prípade skupiny rodičov väčšej než populácia vlastne ani iná možnosť nie je), ako aj od použitej selekčnej metódy.

Výber jedincov sa realizuje na základe šancí na reprodukciu, ktoré jedince, vyskytujúce sa v populácii, majú. Jedince s malou šancou nemusia byť vôbec zaradené do skupiny rodičov. Naopak, tie jedince, ktorých šanca na reprodukciu je vysoká, môžu byť medzi rodičov zaradení viackrát. Teda jedinec s väčšou šancou na reprodukciu sa stáva častejšie rodičom ako jedinec s menšou šancou. Toto však v prípade tých selekčných metód, ktoré sú založené na stochastických princípoch, predstavuje iba priemerný prípad – pri konkrétnej realizácii jedinec s menšou šancou sa môže stať rodičom, zatiaľ čo jedinec so šancou väčšou sa rodičom nestane.

Otázkou ostáva, ako určiť šance jedincov na reprodukciu. Pretože jedince nemajú explicitne priradenú žiadnu šancu na reprodukciu, je nutné ju vyjadriť na základe tých vlastností, ktoré jedince majú. Pre tento účel sa používa vhodnosť jedincov<sup>1</sup>. Intuitívne vyplýva, že ak už je potrebné rozlišovať

---

<sup>1</sup>Šanca by sa však dala vyjadriť aj na základe iných vlastností jedincov. Príkladom by mohla byť závislosť šance na reprodukciu od veku jedinca resp. od jeho vzťahu k iným

rôzne šance jedincov, potom tie jedince, ktoré sú z hľadiska riešenia úlohy nadpriemerné, by mali mať šancu väčšiu ako tie jedince, ktoré sú z tohto hľadiska podpriemerné. Pre splnenie tejto podmienky stačí, ak šancu jedincov na reprodukciu nahradíme ich vhodnosťou. V prípade, že nie je potrebné rozlišovať rôzne šance jedincov, nie je potrebné ani zväzovať tieto šance s vhodnosťou jedincov – stačí každému jedincovi priradiť rovnakú šancu (teda z hľadiska riešenia úlohy nadpriemerné jedince budú mať rovnaké šance ako jedince podpriemerné).

Na základe tohto je možné definovať selekciu ako vzorkovanie populácie jedincov a výber rodičov tak, že vhodnejšie jedince nemajú menšiu šancu na reprodukciu ako menej vhodné jedince. Ak ako selekčná metóda je použitá nejaká metóda, ktorá preferuje jednu skupinu jedincov oproti inej skupine (napr. vhodnejšie jedince majú väčšiu šancu ako menej vhodné), potom selekcia je zdrojom selekčného tlaku. Čím je táto preferencia väčšia, tým je aj selekčný tlak väčší.

Selekčné metódy môžu byť delené do rôznych skupín. Jedným z kritérií, podľa ktorých je možné definovať rozličné delenia selekčných metód, je fakt či daná metóda pre nejakého jedinca garantuje, že daný jedinec bude alebo nebude vybraný do skupiny rodičov. Napríklad, ak každý jedinec z populácie má šancu, že bude vybraný, ale zároveň to pre žiadneho jedinca nie je garantované, potom sa jedná o uchovávaciu selekciu. Naproti tomu vymierajúce stratégie vopred deklarujú, že niektorý jedinec (resp. niektoré jedince) sa v skupine rodičov vyskytnúť nebude. Ak sa to týka najvhodnejších jedincov, potom sa hovorí o vymieraní zľava, a ak najmenej vhodných jedincov, potom zase o vymieraní sprava<sup>2</sup> [4]. A ak nejaká selekčná metóda zaručuje, že najvhodnejší jedinec (alebo viac najvhodnejších jedincov) sa stane rodičom, potom takáto metóda sa označuje ako elitistická selekcia.

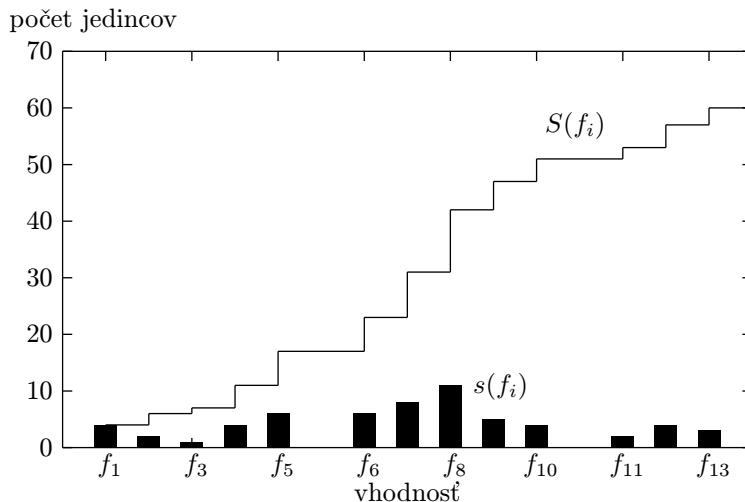
Selekčné metódy je možné skúmať z rôznych hľadísk, napr. z hľadiska najlepšej vhodnosti alebo z hľadiska priemernej vhodnosti. Napríklad, elitistická selekcia prenáša najvyššiu hodnotu vhodnosti z populácie jedincov do skupiny rodičov, zatiaľ čo pri uchovávačnej stratégii sa táto hodnota môže preniesť ale rovnako sa môže aj stratiť (to sa práve deje pri vymieraní zľava). Čo sa týka priemernej vhodnosti, tá má tendenciu sa zvyšovať vďaka preferovaniu vhodnejších jedincov na úkor menej vhodných. Avšak následkom stochastickej podstaty niektorých selekčných metód pri realizácii vzorkovania populácie sa priemerná hodnota vhodnosti môže znížiť, pričom to je viac

---

jedincom (napríklad výber čo naj(ne)podobnejších jedincov).

<sup>2</sup>Pri definovaní týchto pojmov ich autor očividne pracoval s populáciou jedincov zostavenou zostupne podľa vhodnosti.

pravdepodobné pri uchovávaajúcej selekcii ako pri vymieraní sprava.



Obr. 6.1: Príklad diskkrétnej distribučnej funkcie  $s(f_i)$  a kumulatívnej distribučnej funkcie  $S(f_i)$

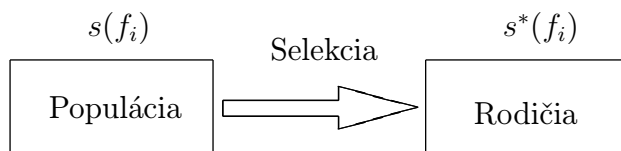
My pre účel charakterizácie jednotlivých selekčných metód použijeme distribúciu vhodnosti (histogram vhodnosti). Pri tomto prístupe sa berie do úvahy, že v nejakej skupine jedincov (či už v populácii, skupine vybratých rodičov, alebo nejakej inej skupine) sa vo všeobecnosti vyskytujú jedince s rôznymi vhodnosťami, pričom niektoré jedince môžu mať vhodnosť rovnakú. To znamená, že v tejto skupine je možné nájsť niekoľko rôznych hodnôt vhodnosti  $f_1 < f_2 < \dots < f_n$ , pričom počet týchto hodnôt je menší alebo rovný počtu jedincov. Distribúcia vhodnosti je charakterizovaná diskretnou distribučnou funkciou  $s(f_i)$  a kumulatívnu distribučnou funkciou  $S(f_i)$  (obr. 6.1), pričom prvá reprezentuje histogram hodnôt vhodnosti a druhá je definovaná ako

$$S(f_i) = \begin{cases} 0 & i < 1 \\ \sum_{j=1}^i s(f_j) & 1 \leq i \leq n \\ \mu & i > n \end{cases} \quad (6.1)$$

kde  $\mu$  predstavuje počet jedincov v skupine (teda v tomto prípade v populácii).

Selekcia mení distribúciu vhodnosti – vo všeobecnosti platí, že distribúcia vhodnosti je iná v populácii jedincov a iná v skupine rodičov (a to aj vtedy, ak platí rovnosť  $\mu = \rho$ ). Na základe znalosti distribúcie vhodnosti

v populácii a znalosti použitej selekčnej metódy je možné určiť, aká bude distribúcia vhodnosti v skupine rodičov (obr. 6.2). To však bude iba očakávaná distribúcia vhodnosti  $s^*(f_i)$  (teoretická distribúcia), na rozdiel od skutočnej distribúcie  $s'(f_i)$  získanej realizáciou vzorkovania populácie. Analogicky, kumulatívna distribúcia vhodnosti sa selekciou mení na očakávanú kumulatívnu distribúciu vhodnosti  $S^*(f_i)$ .



Obr. 6.2: Zmena distribúcie vhodnosti prostredníctvom selekcie

Pomer distribúcií vhodnosti po a pred selekciou

$$\frac{s^*(f_i)}{s(f_i)} \quad (6.2)$$

sa označuje ako miera reprodukcie a vyjadruje, koľkokrát sa selektuje jedinec s vhodnosťou  $f_i$ . Za rozumnú mieru reprodukcie sa považujú hodnoty  $\geq 1$  pre nadpriemerné hodnoty vhodnosti a hodnoty  $\leq 1$  pre podpriemerné hodnoty vhodnosti.

## 6.1 Vzorkovanie populácie

Vzorkovaním populácie sa chápe proces výberu rodičov z jedincov aktuálnej populácie. Pri použití ľubovoľnej selekčnej metódy sú jedince vyberané do úlohy rodičov s nejakou pravdepodobnosťou  $p_s$ . Rozličné pravdepodobnosti selekcie jedincov s rôznymi hodnotami funkcie vhodnosti odrážajú preferovanie niektorých jedincov na úkor iných jedincov.

Podľa toho, či jedince sú vzorkované stále s tými istými pravdepodobnosťami alebo tieto pravdepodobnosti sú rôzne v jednotlivých generáciách (fázach evolučného vývoja), sa selekčné metódy delia na statické a dynamické. Pri statických selekčných metódach platí, že najvhodnejší jedinec je v každej generácii selektovaný s rovnakou pravdepodobnosťou, rovnako aj druhý najvhodnejší jedinec je tiež vyberaný s rovnakou pravdepodobnosťou (vo všeobecnosti inou ako je pravdepodobnosť výberu najvhodnejšieho jedinca), atď. Formálnejšie to možno vyjadriť ako podmienku

$$\forall i \in \{1, \dots, \mu\}, \forall t \geq 0 : p_s(a_i(t)) = c_i \quad (6.3)$$

kde hodnoty  $c_i$  reprezentujú konštanty a  $a_i(t)$  zase  $i$ -teho jedinca (v populácii zotriedenej podľa vhodnosti<sup>3</sup>) v generácii  $t$ . Statické selekčné metódy sú teda previazané iba s pozíciou jedinca v zotriedenej postupnosti jedincov.

Naopak, ak sa pravdepodobnosť selekcie aspoň jedného jedinca mení v priebehu evolúcie, a teda platí podmienka

$$\exists i \in \{1, \dots, \mu\}, \exists t_1 \geq 0, \exists t_2 \geq 0, t_2 \neq t_1 : p_s(a_i(t_1)) \neq p_s(a_i(t_2)) \quad (6.4)$$

tak daná selekčná metóda je dynamická. Aj keď stačí zmena iba u jedného jedinca, zvyčajne, keď sa pripúšťa zmena pravdepodobnosti selekcie, tak sa táto prejavuje u všetkých jedincov (alebo skoro všetkých – niekedy sa pre toho najhoršieho stanovuje nulová hodnota pravdepodobnosti selekcie).

V uvádzaných dynamických selekčných metódach vyplýva dynamika z previazanosti pravdepodobnosti selekcie jedincov a ich vhodnosti (kapitola 6.1.1.2). Pretože populácia jedincov je v rôznych generáciách rôzna (iba vtedy má použitie evolučného algoritmu praktický zmysel), mení sa aj distribúcia vhodnosti v populácii – a keďže sa menia vhodnosti jedincov, tak potom sa menia aj pravdepodobnosti ich selekcie. Z teoretického hľadiska sú uprednostňované statické selekčné metódy, pretože ak sú pravdepodobnosti selekcie konštantné (a tieto konštanty sa dajú určiť), potom chovanie takýchto metód je predikovatelné a je ich možné použiť v teoretických modeloch činnosti evolučného algoritmu. Naproti tomu, keďže pri dynamických metódach pravdepodobnosti selekcie jedincov závisia od konkrétnych hodnôt vhodnosti, ktoré nie sú vopred známe, nie je možné tieto pravdepodobnosti vopred určiť a použiť pri modelovaní. To však neznamená, že by tieto metódy pri riešení praktických úloh neboli použiteľné.

Aj keď každá selekčná metóda vyberá rodičov s nejakou pravdepodobnosťou, nie každá s touto pravdepodobnosťou narába v explicitnej podobe. Pri mnohých metódach pravdepodobnosť selekcie je iba dôsledkom činnosti mechanizmu vzorkovania populácie, použitého na výber rodičov. Pri explicitnej práci s pravdepodobnosťami selekcie sa tieto pravdepodobnosti určujú pred samotnou realizáciou vzorkovania populácie (to sú požadované pravdepodobnosti selekcie) a toto vzorkovanie je potom realizované tak, aby jedince boli vyberané s danými pravdepodobnosťami. Tie sú však vyberané s aktuálnou pravdepodobnosťou selekcie, ktorá sa môže líšiť od požadovanej. Ich zhoda je meraná ako odchýlka – absolútny rozdiel medzi požadovanou a aktuálnou pravdepodobnosťou selekcie jedinca. Optimálnou je nulová od-

<sup>3</sup>Ak v ďalšom nebude uvedené inak, tak pri zotriedení populácie podľa vhodnosti budeme uvažovať jej vzostupné zotriedenie – jedinec s indexom 1 bude mať najnižšiu vhodnosť a jedinec s indexom  $\mu$  zase najvyššiu vhodnosť v populácii.

chýlka, keď aktuálna pravdepodobnosť vzorkovania jedinca súhlasí s požadovanou.

Keďže každý jedinec je selektovaný s určitou pravdepodobnosťou, tak je možné pre každého jedinca určiť očakávaný počet rodičov<sup>4</sup> – koľkokrát daný jedinec bude vybraný do skupiny rodičov

$$\eta(a_i(t)) = \rho p_s(a_i(t)) \quad (6.5)$$

Realizácia nejakej metódy však spôsobí, že aktuálny počet  $\eta'(a_i(t))$ , vyjadrujúci koľkokrát sa  $i$ -ty jedinec v  $t$ -tej generácii stal rodičom, sa bude líšiť od očakávaného počtu rodičov pre daného jedinca (čo je takmer vždy, nakoľko očakávaný počet rodičov je vo všeobecnosti reálne a nie celé číslo). Výber iného ako očakávaného počtu kópií jedincov reprezentuje chybu vzorkovania, ktorá pôsobí ako podstatný zdroj šumu.

Rozsah možných hodnôt, koľkokrát sa môže nejaký jedinec stať rodičom, sa označuje ako rozpätie. Čím je rozpätie nejakej metódy menšie, tým sú výsledky tejto metódy predikovateľnejšie. Minimálne rozpätie<sup>5</sup> je

$$\{\lfloor \eta(a_i(t)) \rfloor, \lceil \eta(a_i(t)) \rceil\} \quad (6.6)$$

kde  $\lfloor \cdot \rfloor$  znamená zaokrúhlenie nadol a  $\lceil \cdot \rceil$  zaokrúhlenie nahor. Maximálne rozpätie je pritom definované ako  $\{0, \dots, \rho\}$ .

Zatiaľ čo odchýlku je možné považovať za mieru správnosti použitej selekčnej metódy, rozpätie indikuje presnosť danej metódy a teda je možné ho považovať za mieru konzistencie selekčnej metódy [2].

### 6.1.1 Vzorkovanie s explicitnými pravdepodobnosťami

Aj keď v literatúre tento typ selekcie je často prezentovaný ako jedna metóda, v skutočnosti tento typ selekcie pozostáva z dvoch samostatných fáz – určenia cieľovej pravdepodobnosti selekcie jedincov a výberu jedincov do úlohy rodičov. Pre každú z týchto dvoch fáz existuje niekoľko metód ako ich je možné realizovať, pričom ich možno vzájomne kombinovať.

<sup>4</sup>Je to vlastne stredná hodnota počtu výberov daného jedinca.

<sup>5</sup>S rozpätím súvisí aj otázka genetického driftu. Pri tomto jave aj v prípade, že všetky jedince majú rovnakú pravdepodobnosť selekcie, niektoré jedince nie sú vybrané vôbec zatiaľ čo iné sú vybrané viackrát – a teda dochádza k redukcii rôznorodosti materiálu v populácii, s hrozbou konverencie celej populácie k jednému jedincovi. Príčinou je stochastický charakter selekcie. Selekcčná metóda s minimálnym rozpätím netrpí genetickým driftom.



### 6.1.1.1 Výber jedincov

Vychádza sa z toho, že pre každého jedinca je známa jeho pravdepodobnosť selekcie, čo pri známom počte rodičov, ktorých je potrebné vybrať, umožňuje pre každého jedinca určiť, koľkokrát daný jedinec má byť selektovaný (vzťah (6.5)). Problémom je však to, že tento očakávaný počet rodičov je vo všeobecnosti reálne a nie celé číslo. Musí byť teda nejakým spôsobom konvertovaný na celočíselnú hodnotu.

V minulosti boli často používanými metódy založené na analógii s ruletou. Pri týchto metódach sa vychádzalo z jednoduchého princípu:

Na rulete sa vyznačilo toľko úsekov, koľko bolo jedincov v populácii. Veľkosť každého úseku korešpondovala s očakávaným počtom rodičov, prislúchajúcim danému jedincovi (a teda celková dĺžka obvodu rulety sa zhodovala s požadovaným počtom rodičov  $\varrho$ ). Ruleta sa náhodne rozkrútila a po jej zastavení bol za rodiča vybraný ten jedinec, ktorému prislúchal ten dielik rulety na ktorý ukazoval jazýček. Tento proces bol opakovaný toľkokrát, koľko bolo neobsadených miest rodičov.

Realizácia tohto princípu sa označuje ako stochastické vzorkovanie s náhradou. Jedná sa o správnu vzorkovaciu metódu, nakoľko jej odchýlka je nulová – pri každom zatočení rulety sú jedince vyberané s pravdepodobnosťami rovnými požadovaným pravdepodobnostiam selekcie. Na druhej strane táto metóda trpí veľmi malou presnosťou, pretože produkuje maximálne možné rozpätie. Aj keď očakávaný počet rodičov  $i$ -teho jedinca je konkrétna hodnota  $\eta(a_i(t)) > 0$ , konkrétna realizácia môže vyprodukovať ľubovoľný počet selekcií tohto jedinca – od prípadu, že nebude vybraný ani raz, až po prípad, že celkom vyplní skupinu rodičov svojimi kópiami.

Pre zníženie rozpätia a tým aj zlepšenie presnosti slúžili stochastické vzorkovanie bez náhrady, zvyškové stochastické vzorkovanie s náhradou a zvyškové stochastické vzorkovanie bez náhrady.

Variant “zvyškového” vzorkovania sa snažil zmenšiť rozpätie stanovením spodnej hranice koľkokrát daný jedinec bude aktuálne vybraný. Všetky očakávané počty rodičov prislúchajúce jedincom sa rozdelili na celú a desatinnú časť. Celá časť sa realizovala deterministicky, zatiaľ čo na výber pomocou rulety sa použili iba desatinné časti očakávaných hodnôt. Teda v tomto prípade dĺžka rulety bola

$$\varrho - \sum_{i=1}^{\mu} \lfloor \eta(a_i(t)) \rfloor = \sum_{i=1}^{\mu} (\eta(a_i(t)) - \lfloor \eta(a_i(t)) \rfloor) \quad (6.7)$$

čo súčasne udáva, koľko miest rodičov bolo obsadzovaných stochasticky. Tento variant teda zabezpečil

$$\eta'(a_i(t)) \geq \lfloor \eta(a_i(t)) \rfloor \quad (6.8)$$

čím zmenšil rozpätie, zatiaľ čo naďalej zachoval nulovú odchýlku.

Variant vzorkovania “bez náhrady” sa naopak snažil zmenšiť rozpätie stanovením hornej hranice koľkokrát daný jedinec bude vybratý. Keď zatočenie rulety rozhodlo o výbere nejakého jedinca, tak dĺžka toho dielika rulety ktorý zodpovedal tomuto jedincovi bol zmenšený o jednotku (v prípade, že pred týmto zmenšením jeho dĺžka bola menšia ako 1, tak bol zmenšený o túto dĺžku). Skrátene nejakého dielika na nulovú dĺžku znamenalo, že príslušný jedinec už viac nebol vybraný. Tento variant teda zabezpečil

$$\eta'(a_i(t)) \leq \lceil \eta(a_i(t)) \rceil \quad (6.9)$$

čím zmenšil rozpätie. Na rozdiel od predchádzajúceho variantu, v tomto sa dĺžka rulety skracovala – po výbere každého rodiča bola kratšia. Ak bol nejaký jedinec vybraný do úlohy prvého rodiča s pravdepodobnosťou  $p_s(a_i(t))$  a bol skutočne vybraný (a obvod rulety sa skrátil na hodnotu  $\varrho - 1$ ), tak potom do úlohy druhého rodiča je už vybraný s menšou pravdepodobnosťou ( $p_s(a_i(t)) - 1/\varrho$ ). Ak by nebol vybraný do úlohy prvého rodiča, tak pre výber na miesto druhého rodiča sa jeho pravdepodobnosť po rovnakom skrátene obvodu rulety zvýšila na  $\varrho p_s(a_i(t))/(\varrho - 1)$ . To znamená, že okrem výberu prvého rodiča boli jedince vzorkované s inou pravdepodobnosťou ako bola požadovaná pravdepodobnosť selekcie. Cenou za zníženie rozpätia bol teda vzrast odchýlky.

Z rodiny uvedených ruletových metód sa tak najpopulárnejšou stalo zvyškové stochastické vzorkovanie bez náhrady, ktoré spojením vzťahov (6.8) a (6.9) zabezpečuje minimálnu hodnotu rozpätia s hodnotou odchýlky, o ktorej sa verilo, že je malá. Avšak [2] experimentálne potvrdilo, že odchýlka tejto metódy (v tomto prípade preferovanie hodnôt s malými desatinnými časťami) je značná v porovnaní s niektorými inými metódami vzorkovania.

Jednou z týchto iných metód<sup>6</sup> je stochastické univerzálne vzorkovanie. Je tiež založené na existencii rulety s obvodom dĺžky  $\varrho$ , na ktorej každému jedincovi zodpovedá dielik, ktorého dĺžka je rovná očakávanému počtu rodičov vytvorených z daného jedinca. Modifikácia pôvodného ruletového princípu je v tom, že neexistuje iba jeden jazýček ale toľko jazýčkov, koľko rodičov je

---

<sup>6</sup>Inou metódou je zvyškové stochastické nezávislé vzorkovanie [2], ktoré síce pri zabezpečení minimálnej hodnoty rozpätia dosahuje malú odchýlku, avšak je vhodným kandidátom na paralelnú implementáciu.

potrebné vybrať. Tieto jazýčky sú rozmiestnené po obvode rulety v ekvidiš-tantných pozíciách, pričom vzdialenosť medzi ľubovoľnými dvomi susednými jazýčkami je jednotková. Pre selekciu rodičov stačí jedno zatočenie rulety – a každý jazýček vyberá jedného rodiča. Vďaka takémuto usporiadaniu sa dosahuje minimálna hodnota rozpätia, pričom metóda dosahuje nulovú od-chýlku.

Nevýhodou tejto metódy je to, že selekcia kombinácií jedincov závisí od rozloženia im zodpovedajúcich dielikov na rulete. Ak totiž sú vedľa seba umiestnené dva dieliky, súčet dĺžok ktorých je menší ako 1, potom môže byť selektovaný jeden alebo druhý z nich, nie však obaja spolu. Preto pred roz-miestňovaním dielikov na ruletu je potrebné jedince v populácii premiešať (tak aby sa ľubovoľný jedinec mohol dostať na ľubovoľnú pozíciu s rovna-kou pravdepodobnosťou). Toto zvyšuje celkovú zložitosť tejto vzorkovacej metódy.

Jedna z možností, ako náhodne zamiešať/zotriediť populáciu, je použiť nasledujúci algoritmus pre vygenerovanie náhodného poradia

```
for( i=1; i< k; i++ ) {
    j = random_int( i, k );
    temp = sorted_pop[j];
    sorted_pop[j] = sorted_pop[i];
    sorted_pop[i] = temp;
}
```

Stochastické univerzálne vzorkovanie je považované za optimálny sek-venčný vzorkovací mechanizmus. Umožňuje vyberať rodičov presne podľa teoretických špecifikácií, čo sa s výhodou využíva pri experimentoch ove-rujúcich teoretické modely práce evolučných algoritmov. Na druhej strane nie je jasné, do akej miery výskyt nenulovej odchýlky ovplyvňuje výkonnosť evolučného algoritmu ako celku pri riešení praktických úloh.

### 6.1.1.2 Určenie pravdepodobnosti selekcie

Cieľom tejto fázy je určenie cieľových pravdepodobností selekcie. Buď sa tieto určujú podľa aktuálneho stavu v populácii (napr. podľa distribúcie vhodnosti) alebo podľa toho, aký model chovania je žiadané dosiahnuť. Ukážkou nastavenia určitého modelu môže byť prípad podľa [30], keď prav-depodobnosti selekcie sa určia na základe poradia jedincov v populácii, kde

$i$ -temu poradiu bude zodpovedať pravdepodobnosť<sup>7</sup>

$$q(1 - q)^{i-1} \quad (6.10)$$

kde parameter  $q \in (0, 1)$ . V tomto prípade poradie  $i = 1$  bude znamenať najpreferovanejšieho jedinca, zatiaľ čo  $i = \mu$  zase najmenej preferovaného jedinca. Čím je hodnota parametra  $q$  väčšia, tým viac sú preferované vhodnejšie jedince oproti menej vhodným jedincom.

Pri voľbe nejakého modelu chovania je taktiež možné napr. zobrať pravdepodobnosti selekcie odvodené v sekcii 6.1.2 a získať tak rovnaké výsledky ako výsledky tam popisovaných selekčných metód (vlastne nie rovnaké, pretože pri vzorkovaní s explicitne vyjadrenými pravdepodobnosťami je možné dosiahnuť minimálne rozpätie, čo nie je prípad všetkých metód v tej sekcii).

Z aktuálnej distribúcie vhodnosti vychádza proporcionálna selekcia, pri ktorej pravdepodobnosť selekcie je založená na aktuálnej vhodnosti. V prípade tejto metódy sa pravdepodobnosti selekcie určujú ako relatívne vhodnosti jedincov podľa vzťahu

$$p_s(a_i(t)) = \frac{\Phi(a_i(t))}{\sum_{j=1}^{\mu} \Phi(a_j(t))} = \frac{\Phi(a_i(t))}{\mu \bar{\Phi}(t)} \quad (6.11)$$

kde  $\bar{\Phi}(t)$  reprezentuje priemernú vhodnosť populácie v  $t$ -tej generácii.

Vzťah určuje, že jedince s priemernou vhodnosťou budú selektované s pravdepodobnosťou  $1/\mu$  (a teda v prípade, že počet rodičov je rovnako veľký ako počet jedincov v populácii, každý jedinec s priemernou vhodnosťou by mal získať jednu kópiu v skupine rodičov). Jedince s nadpriemernou vhodnosťou sú selektované s nadpriemernou pravdepodobnosťou a jedince s podpriemernou vhodnosťou majú zase pravdepodobnosť selekcie podpriemernú. Ak vhodnosť nejakého jedinca je dvojnásobná ako vhodnosť iného jedinca, tak ten vhodnejší bude vyberaný s dvojnásobnou pravdepodobnosťou oproti tomu menej vhodnému.

Oponenti tejto selekčnej metódy jej vyčítajú niektoré nevýhody. Predovšetkým, nie je možné pracovať s negatívnymi hodnotami vhodnosti. Druhým vážnym problémom je situácia, ak v populácii s jedincami s pomerne nízkou

---

<sup>7</sup>Aby súčet pravdepodobností selektovania všetkých jedincov bol rovný 1, je potrebné do vzťahu pridať ešte konštantu  $k$ , čím sa získa  $kq(1 - q)^{i-1}$ . Pre hodnotu tejto konštanty musí platiť (s využitím vzťahu (6.47) pre spočítanie pravdepodobnosti všetkých jedincov)  $k = \frac{1}{1 - (1 - q)^\mu}$ .

Použitím substitúcie  $q = 1 - c$  možno dokázať, že tieto pravdepodobnosti selekcie sú rovnaké ako pri premapovaní vhodnosti exponenciálnym zotriedením s následným použitím proporcionálnej selekcie (sekcia 6.2.2) – ibaže jedince sú zotriedené opačným smerom (zmenu zotriedenia možno dosiahnuť náhradou  $i - 1$  v exponente za  $\mu - i$ ).

vhodnosťou sa zrazu objaví jedinec s veľmi vysokou vhodnosťou. Preferencia tohto jedinca bude taká veľká, že zaplní veľký počet rodičovských miest (v extrémnom prípade celú skupinu rodičov), čím môže nežiadúco ovplyvniť zloženie nasledujúcich generácií. Ich rôznorodosť poklesne, a následkom toho celé hľadanie predčasne skonverguje do oblasti v okolí daného jedinca. Takýto jedinec sa označuje ako superjedinec.

Aj keď sa zachováva rozdiel medzi vhodnosťami dvoch jedincov, preferencia jedného voči druhému sa mení, pretože závisí aj od veľkosti hodnôt vhodnosti. Napríklad, pri vhodnostiach 1 a 2 bude mať druhý jedinec dvojnásobnú pravdepodobnosť selekcie, zatiaľ čo pri hodnotách 11 a 12 preferencia bude menšia a pri hodnotách 1001 a 1002 skoro zanedbateľná. Následkom toho v neskorších fázach evolučného vývoja, keď vhodnosti všetkých jedincov v populácii sú vysoké, sa začína strácať tlak na preferovanie lepších jedincov a hľadanie sa začína podobať náhodnému prehľadávaniu.

Keďže presné hodnoty pravdepodobnosti závisia od konkrétnej distribúcie vhodnosti, ktorá sa mení počas evolučného procesu, jedná sa o typického predstaviteľa dynamickej selekcie. V spojení s metódou výberu jedincov, zabezpečujúcou minimálne rozpätie, sa chová ako elitistická selekcia. Ak však vzorkovacia metóda má maximálne rozpätie, tak potom nie je zaručený výber najvhodnejšieho jedinca.

Vďaka závislosti na aktuálnej distribúcii vhodnosti nie je možné predikovať výsledok selekcie, ktorý vo všeobecnosti je pri rôznych distribúciách vhodnosti rôzny. Predikcia chovania je možná vždy iba pre nejakú konkrétnu distribúciu vhodnosti. Napríklad, ak by táto bola daná lineárnym vzťahom

$$\Phi(a_i(t)) = \left( \xi + 2(1 - \xi) \frac{i - 1}{\mu - 1} \right) \quad (6.12)$$

kde  $i$  označuje poradie jedinca v podľa vhodnosti vzostupne zotriedenej postupnosti jedincov, tak potom by priemerná vhodnosť bola

$$\bar{\Phi}(t) = \frac{\Phi(a_1(t)) + \Phi(a_\mu(t))}{2} = \frac{\xi + (2 - \xi)}{2} = 1 \quad (6.13)$$

Dosadením do vzťahu (6.11) by sa zistilo, že v tomto prípade by  $i$ -ty jedinec bol selektovaný s pravdepodobnosťou

$$p_s(a_i(t)) = \frac{1}{\mu} \left( \xi + 2(1 - \xi) \frac{i - 1}{\mu - 1} \right) \quad (6.14)$$

V tomto prípade očakávaný počet jedincov s vhodnosťou  $f_i$  alebo menšou

je (jedince sú zotriedené vzostupne podľa svojej vhodnosti)

$$S^*(f_i) = \varrho \sum_{j=1}^{S(f_i)} p_s(a_j(t)) = S(f_i) \frac{\varrho}{\mu} \xi + \frac{\varrho}{\mu} 2(1 - \xi) \frac{1}{\mu - 1} \sum_{j=1}^{S(f_i)} (j - 1) \quad (6.15)$$

S uvážení, že  $\sum_{j=0}^n j = (j + 1)j/2$  potom získame

$$S^*(f_i) = \frac{\varrho}{\mu} \xi S(f_i) + \frac{\varrho}{\mu} \frac{1 - \xi}{\mu - 1} S(f_i)(S(f_i) - 1) \quad (6.16)$$

Potom pre očakávanú distribúciu vhodnosti

$$\begin{aligned} s^*(f_i) &= S^*(f_i) - S^*(f_{i-1}) \\ &= \frac{\varrho}{\mu} \xi S(f_i) - \frac{\varrho}{\mu} \xi S(f_{i-1}) \\ &\quad + \frac{\varrho}{\mu} \frac{1 - \xi}{\mu - 1} [S(f_i)(S(f_i) - 1) - S(f_{i-1})(S(f_{i-1}) - 1)] \\ &= \frac{\varrho}{\mu} \frac{\mu \xi - 1}{\mu - 1} s(f_i) + \frac{\varrho}{\mu} \frac{1 - \xi}{\mu - 1} (S(f_i)^2 - S(f_{i-1})^2) \end{aligned} \quad (6.17)$$

V špeciálnom prípade, ak za konštantu  $\xi$  dosadíme hodnotu  $1/\mu$ , potom očakávaná distribúcia vhodnosti bude

$$s^*(f_i) = \frac{\varrho}{\mu^2} (S(f_i)^2 - S(f_{i-1})^2) \quad (6.18)$$

čo je rovnaká hodnota ako pre binárny turnaj v prípade, že v populácii nie sú žiadne dva jedince s rovnakou vhodnosťou (sekcia 6.1.2.1) – to znamená že pri danom špeciálnom tvare distribúcie vhodnosti proporcionálna selekcia založená na vhodnosti môže byť ekvivalentná binárnemu turnaju.

## 6.1.2 Vzorkovanie s implicitnými pravdepodobnosťami

Do tejto triedy selekčných metód patria tri skupiny metód – turnaje, orezania, a náhodný výber. Ich spoločnou charakteristikou je to, že v popise činnosti týchto metód sa nevyskytuje slovo “pravdepodobnosť” – avšak je ju možné určiť na základe toho, ako jednotlivé jedince sú selektované.

### 6.1.2.1 Turnaje

Pri turnajových metódach jedince spolu súťažia. O víťazovi rozhoduje vhodnosť tých jedincov, ktorí sa zúčastňujú turnaja. Výhodou turnajov oproti

orezaniu (a aj niektorým variantom náhodného výberu) je to, že sa nevyžaduje aby jedince boli zotriedené podľa svojej vhodnosti. Týmto spôsobom odpadá fáza triedenia jedincov, čím sa znižuje časová náročnosť turnajových metód, čo prispieva k ich popularite pri praktických aplikáciách.

Aj keď existuje viacero turnajových metód (napríklad pravdepodobnostný turnaj<sup>8</sup>), dominantné postavenie zaujíma  $q$ -árny turnaj. Princíp tejto metódy je pomerne jednoduchý:

Náhodne sa z populácie vyberie  $q$  jedincov a ten z nich, ktorý má najvyššiu vhodnosť, sa stáva víťazom turnaja – a zároveň aj rodičom. Ak jedincov s najlepšou vhodnosťou je v turnaji viac, tak víťaz sa vyberie náhodne spomedzi nich. Realizáciou jedného turnaja je teda možné vybrať jedného rodiča. Celá selekcia potom pozostáva z toľkých turnajov, koľko rodičov je potrebné selektovať (pre každého rodiča jeden osobitný turnaj).

Parameter  $q$  sa nazýva veľkosť turnaja a umožňuje ovplyvňovať preferenciu vhodnejších jedincov oproti menej vhodným jedincom. Pomocou neho je možné (v určitej obmedzenej miere, nakoľko hodnota  $q$  musí byť prirodzené číslo) regulovať veľkosť selekčného tlaku produkovaného selekčnou metódou. Jedinec s najvyššou vhodnosťou sa stane víťazom vo všetkých turnajoch, ktorých sa zúčastní. Naopak, jedinec s najnižšou vhodnosťou nevyhrá ani jeden turnaj v ktorom by súťažil s jedincami iných vhodností. Priemerný jedinec v binárnom turnaji ( $q = 2$ ) s jedincami iných vhodností ako daný jedinec vyhrá priemerne polovicu svojich turnajov, avšak v ternárnom turnaji ( $q = 3$ ) ich už vyhráva priemerne iba štvrtinu. Inak povedané, čím väčšia hodnota  $q$ , tým silnejšia preferencia najvhodnejších jedincov.

To, že z populácie bude náhodne vybratý jedinec s vhodnosťou rovnou alebo menšou ako nejaká hodnota  $f_i$ , má pravdepodobnosť

$$\frac{S(f_i)}{\mu} \quad (6.19)$$

Keďže výbery v rámci turnaja sú nezávislé, potom pravdepodobnosť toho, že turnaj vyhrá jedinec s vhodnosťou rovnou alebo menšou ako hodnota  $f_i$  (teda všetky jedince, ktoré sa zúčastnia turnaja, majú vhodnosť menšiu nanajvýš rovnú tejto hodnote) je

$$\left(\frac{S(f_i)}{\mu}\right)^q \quad (6.20)$$

<sup>8</sup>Pri pravdepodobnostnom turnaji lepší jedinec nemusí vyhrať deterministicky, ale iba s určitou pravdepodobnosťou.

Teda  $\varrho$  opakovaní turnaja by malo vyprodukovať

$$\varrho \left( \frac{S(f_i)}{\mu} \right)^q \quad (6.21)$$

rodičov s vhodnosťou menšou alebo rovnou hodnote  $f_i$ , čo je vlastne očakávaná kumulatívna distribúcia vhodnosti  $S^*(f_i)$ . Keďže miera reprodukcie podľa (6.2)

$$\frac{s^*(f_i)}{s(f_i)} = \frac{S^*(f_i) - S^*(f_{i-1})}{S(f_i) - S(f_{i-1})} = \frac{\varrho \left( \left( \frac{S(f_i)}{\mu} \right)^q - \left( \frac{S(f_{i-1})}{\mu} \right)^q \right)}{S(f_i) - S(f_{i-1})} \quad (6.22)$$

je súčasne očakávaný počet vybratých kópií jedného jedinca (s danou vhodnosťou), tak potom jedinec s vhodnosťou  $f_i$  je selektovaný s pravdepodobnosťou

$$\frac{1}{\mu^q} \frac{S(f_i)^q - S(f_{i-1})^q}{S(f_i) - S(f_{i-1})} \quad (6.23)$$

Pravdepodobnosť selekcie jedinca teda závisí od toho, koľko jedincov v populácii má rovnaké vhodnosti. Keby sa tento počet nemenil (napr. všetky jedince by mali navzájom rôzne vhodnosti), tak potom by sa tieto pravdepodobnosti nemenili počas evolučného vývoja. V tomto prípade by  $q$ -árny turnaj reprezentoval statickú selekčnú metódu.

Keďže jedince sú do turnaja vyberané s rovnakou pravdepodobnosťou  $1/\mu$ , tak počas celej selekcie by v priemere každý jedinec mal byť vybraný  $\varrho q/\mu$  krát za súťažiaceho (počet turnajov ktorých sa zúčastní je menší, lebo v jednom turnaji môže súťažiť nielen s inými jedincami ale aj sám so sebou – a čím častejšie súťaží sám so sebou, tým v menšom počte turnajov účinkuje). Priemerný počet turnajov, v ktorých jedince účinkujú, je rovnaký pre každého jedinca. A keďže to je súčasne aj počet rodičov, ktorých vyprodukoval najvhodnejší jedinec v prípade, že všetky ostatné jedince majú horšiu vhodnosť, je to možné určiť z (6.23) ako

$$\frac{1}{\mu^q} (\mu^q - (\mu - 1)^q) \quad (6.24)$$

Z popisu  $q$ -árneho turnaja však vyplýva, že pri skutočnej realizácii selekcie tento najvhodnejší jedinec vôbec nemusí byť vybraný do žiadneho turnaja, a teda nevyprodukuje žiadneho rodiča. Alebo naopak, môže byť účastníkom všetkých turnajov a teda celá skupina rodičov bude pozostávať iba z jeho kópií. Inak povedané, rozpätie  $q$ -árneho turnaja je maximálne.

Pre zmenšenie rozpätia je možné použiť variant –  $q$ -árny turnaj bez náhrady (vyššie popísaný turnaj sa niekedy označuje aj ako turnaj s náhradou).



Teraz sa jedinec nemôže zúčastniť v ľubovoľnom počte turnajov, pretože môže byť maximálne  $\lceil \rho q/\mu \rceil$  krát súťažiacim. Po každej účasti v nejakom turnaji sa tento počet znižuje o hodnotu 1. Možno to prakticky realizovať tak, že ak počet požadovaných rodičov je rovný počtu jedincov v populácii, tak sa z každého jedinca najprv vytvorí  $q$  kópií a selektovať sa bude z tejto rozšírenej skupiny jedincov. Po skončení každého turnaja všetky tie jedince, ktoré sa ho zúčastnili, sú z tejto rozšírenej skupiny odstránené. Uvedený prístup možno použiť vždy v prípade, že hodnota  $\rho q/\mu$  je prirodzeným číslom. Tento spôsob realizácie turnaja reprezentuje elitistický variant selekcie.

Výhodou použitia turnajových metód sa prejaví aj pri použití evolučných algoritmov pre riešenie minimalizačných úloh. Stačí, ak víťazom turnaja sa stane ten jedinec, ktorého vhodnosť je najmenšia – nie je potrebná dodatočná transformácia minimalizačnej úlohy na maximalizačnú, ako napr. pri použití metód pracujúcich s explicitne vyjadrenou pravdepodobnosťou selekcie.

### 6.1.2.2 Orezanie

Na rozdiel od iných selekčných metód, pri selekcii orezaním sa vymieranie sprava vyskytuje spravidla vo väčšej miere – nielen najhorší jedinec ale zvyčajne niekoľko najhorších jedincov nemá žiadnu šancu stať sa rodičom. Základná schéma selekcie orezaním je:

Vychádza sa z populácie jedincov zotriedených podľa ich vhodnosti. Populácia sa rozdelí na dve časti. Tá časť, v ktorej sa nachádzajú jedince s nižšou vhodnosťou, sa už ďalej na selekcii nezúčastňuje. Jedince z druhej časti (tie s vyššími vhodnosťami) sú kandidátmi na úlohu rodičov – sú za rodičov vyberané s rovnakou pravdepodobnosťou. To znamená, že medzi jedincami z druhej časti nie je žiadna preferencia jedných jedincov voči druhým.

Delenie populácie na dve časti je ovplyvňované hodnotou parametra  $T \in \langle 1/\mu, 1 \rangle$  – prahom orezania, ktorý udáva z akej časti populácie sa budú vyberať rodičia. Jeho extrémne hodnoty znamenajú priamy výber iba najvhodnejšieho jedinca ( $T = 1/\mu$ ) resp. výber z celej populácie pričom každý jedinec má rovnakú šancu stať sa rodičom nezávisle na jeho vhodnosti ( $T = 1$ ). Keďže tento parameter určuje pomer veľkostí oboch častí, na ktoré sa populácia rozdelí, je ním možné regulovať veľkosť selekčného tlaku produkovaného selekciou. Pretože nadobúda reálne hodnoty, nastavovanie selekčného tlaku je možné realizovať jemnejšie ako pri  $q$ -árnom turnaji.

Ak teda rodičov je možné vybrať iba spomedzi  $T\mu$  najlepších jedincov, každý z týchto jedincov bude mať očakávaný počet kópií v skupine rodičov

$$\frac{1}{T\mu} \varrho \quad (6.25)$$

Ostatných  $(1 - T)\mu$  jedincov bude mať pravdepodobnosť selekcie nulovú. Ak sa v jednotlivých generáciách nemení hodnota  $T$  (nie je snaha adaptívne prispôbovať veľkosť selekčného tlaku), tak orezanie reprezentuje statickú selekčnú metódu.

Očakávaná distribúcia vhodnosti po realizácii selekcie orezaním je

$$s^*(f_i) = \begin{cases} 0 & S(f_i) \leq (1 - T)\mu \\ \varrho \frac{S(f_i) - (1 - T)\mu}{T\mu} & S(f_{i-1}) \leq (1 - T)\mu < S(f_i) \\ \varrho \frac{s(f_i)}{T\mu} & \text{inak} \end{cases} \quad (6.26)$$

Prvý prípad popisuje jedince, ktoré neboli selektované, pretože vytvárajú orezanú časť. Druhý prípad zodpovedá situácii, keď prah rozdelí skupinu jedincov s rovnakou vhodnosťou do oboch častí – z časti tejto skupinky sa budú vyberať rodičia, zatiaľ čo na druhú časť sa zabudne. V poslednom prípade sa môžu selektovať všetky jedince s danou vhodnosťou.

Samotný výber rodičov z vhodnejšej časti populácie je možné realizovať dvojako. V prvom prípade sa vyberá deterministicky. Podmienkou je, aby požadovaný počet rodičov bol násobkom veľkosti skupiny kandidátov na rodičov. Vtedy očakávaný počet rodičov každého jedinca je celé číslo a tak jednoducho je možné každého kandidáta príslušný počet krát preniesť do skupiny rodičov. Vzhľadom na charakter výberu sa jedná o elitistickú selekciu. Najčastejšie sa používa v konfigurácii, keď  $\varrho = T\mu$  a teda každý kandidát sa stáva presne raz rodičom. Tento spôsob sa v literatúre zvykne označovať ako “top n” selekcia (a zvyčajne bez akéhokoľvek spomínania orezania).

Druhý spôsob je založený na pravdepodobnostnom princípe. Vtedy sa spomedzi kandidátov náhodne vyberá potrebný počet rodičov. Podobne ako pri  $q$ -árnom turnaji, aj teraz je možné znižovať rozpätie metódy použitím variantu bez náhrady.

Zložitejším usporiadaním orezania je stupňovité orezávanie, ktoré reprezentuje vlastne akúsi kombináciu viacerých orezaní. V tomto prípade sa predsa len zavádza preferenciu aj medzi kandidátmi na rodičov. Príkladom môže byť orezanie s dvomi prahmi  $T_1 < T_2$ , ktoré rozdelia populáciu na tri rôzne skupiny jedincov. Náhodne s nejakou pravdepodobnosťou  $p$  sa určuje, či sa použije prvý prah alebo druhý. Pri použití  $T_1$  sa rodičia vyberajú iba z tretej skupiny jedincov, zatiaľ čo pri použití väčšieho  $T_2$  sú vyberané z

tretej aj druhej skupiny. Pri takomto usporiadaní je pravdepodobnosť selekcie jedincov z prvej (najmenej vhodnej) skupiny nulová, jedince z druhej (strednej) skupiny majú pravdepodobnosť selekcie

$$(1 - p) \frac{1}{T_2 \mu} \tag{6.27}$$

a jedince z tretej (najvhodnejšej) skupiny zase

$$p \frac{1}{T_1 \mu} + (1 - p) \frac{1}{T_2 \mu} \tag{6.28}$$

V extrémnom prípade, ak by tých prahov bol maximálny možný počet (teda  $\mu - 1$ ) a každý z nich by bol rovnako pravdepodobný, by bola pravdepodobnosť selekcie

$$p_s(a_i) = p \sum_{j=1}^{i-1} \frac{1}{\mu - j} = \frac{1}{\mu - 1} \sum_{j=1}^{i-1} \frac{1}{\mu - j} \tag{6.29}$$

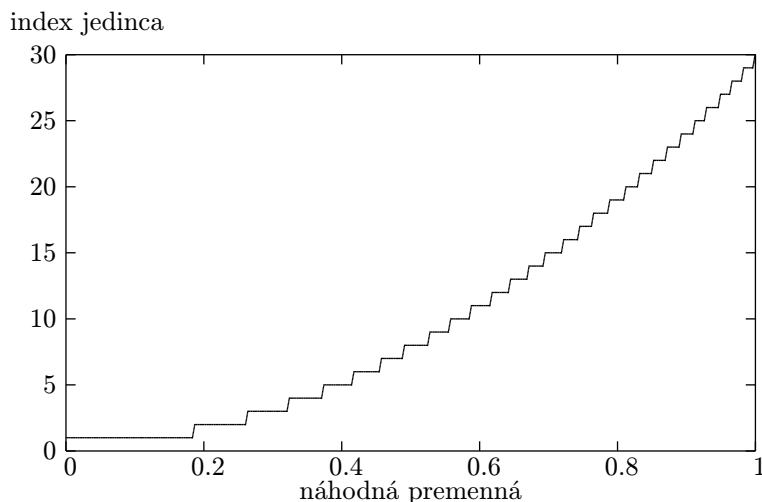
Na rozdiel od turnajov, pri orezaní sa vyžaduje zotriedenie populácie jedincov, čo zvyšuje výpočtové nároky. Čo môže byť ešte väčším problémom, je to, že pri zotriedení pre jedince s rovnakou vhodnosťou (ktoré sú rovnako prijateľné) je potrebné umelo stanoviť, ktorý z nich je lepší. Následkom toho jeden z nich môže pripadnúť do kandidátskej skupiny zatiaľ čo druhý nie.

Rovnako ako pri turnajoch je možné jednoducho riešiť minimalizačné úlohy. Stačí zotriediť populáciu opačným smerom, a ako kandidátov na rodičov uvažovať jedincov s najmenšou vhodnosťou.

### 6.1.2.3 Náhodný výber

Najčastejšie sa vyskytujúcim spôsobom náhodného výberu je prípad, keď všetky jedince majú rovnakú pravdepodobnosť výberu. Vtedy sa náhodným spôsobom vyberie príslušný počet rodičov, pričom sa vlastne neuvažuje vhodnosť jedincov, a teda vhodnejšie jedince nie sú preferované pred menej vhodnými. Je to vlastne zároveň aj prípad stochastického variantu orezania s hodnotou parametra  $T = 1$ .

Okrem tohto triviálneho prípadu je možná aj selekcia, založená na existencii nejakej prevodovej funkcie realizujúcej zobrazenie, kde nezávisle premenná reprezentuje náhodnú premennú a závisle premenná zase index jedinca v postupnosti zotriedenej podľa vhodnosti jedincov. Od toho, či daná funkcia bude mať konkávny alebo konvexný charakter, závisí preferencia medzi skupinou vhodnejších a menej vhodných jedincov.



Obr. 6.3: Príklad prevodovej funkcie preferujúcej jedinca s nižším indexom

Napríklad, ak prevodová funkcia má tvar podľa obr. 6.3 (“schodovitost” je zapríčinená tým, že index jedinca nemôže byť reálne číslo) a jedinca sú usporiadané zostupne podľa svojej vhodnosti (čím menší index, tým vhodnejší jedinec), tak vhodnejšie jedinca sú viac preferované ako menej vhodné – pravdepodobnosť výberu jedinca z prvej (najlepšej) tretiny jedincov je väčšia ako pravdepodobnosť výberu jedinca z poslednej (najhoršej) tretiny jedincov.

Príkladom takejto selekcie (statickej, pretože opäť nebude záležať na konkrétnych hodnotách vhodnosti) je funkcia

$$i = \left\lfloor \frac{\mu}{2(k-1)} \left( k - \sqrt{k^2 - 4(k-1)\chi} \right) \right\rfloor \quad (6.30)$$

kde  $\chi \in (0, 1)$  reprezentuje náhodnú premennú<sup>9</sup>. Konštanta  $k$  umožňuje zvyšovať alebo znižovať preferovanie jedincov s nadpriemernou vhodnosťou voči jedincom s podpriemernou vhodnosťou. Pre jej hodnotu platí  $1 < k \leq 2$ .

Ak by sme  $\chi$  považovali za kumulatívnu distribučnú funkciu hodnôt indexov  $F(x)$  [4] a predchádzajúci vzťah by sme zapísali v spojitom tvare

$$x = \frac{\mu}{2(k-1)} \left( k - \sqrt{k^2 - 4(k-1)F(x)} \right) \quad (6.31)$$

<sup>9</sup>V uvedenom tvare vzťah dokáže produkovať indexy z  $\{0, 1, \dots, \mu-1\}$ , čo pripočítaním jednotky možno upraviť na nami používaný rozsah indexov.

potom jednoduchou úpravou by sa odvodilo

$$F(x) = \frac{x}{\mu} \left( k - \frac{x}{\mu}(k-1) \right) \quad (6.32)$$

Spojité hustota pravdepodobnosti by sa určila z distribučnej funkcie ako

$$\frac{dF(x)}{dx} = \frac{1}{\mu} \left( k - 2\frac{x}{\mu}(k-1) \right) \quad (6.33)$$

čo je nápadne podobné pravdepodobnosti selekcie v prípade lineárnej distribúcie vhodnosti a proporcionálnej selekcie založenej na vhodnosti (6.14) – len s tým rozdielom, že jedince sú teraz zoradené zostupne.

Takáto selekcia sa jednoducho implementuje, avšak jej nevýhodou je maximálne rozpätie, a nutnosť triedenia populácie so zavedením preferencie aj medzi rovnako vhodné jedince. Rovnako ako predchádzajúce metódy umožňuje jednoduché riešenie minimalizačných úloh zmenou poradia jedincov.

## 6.2 Premapovanie vhodnosti

Nie vždy je vhodné, aby selekčné metódy pracovali priamo s vhodnosťou jedincov. Dôvodom pre to je, že distribúcia vhodnosti svojim tvarom nezodpovedá požiadavkám na ňu kladeným pre to, aby sa dosiahlo nejaké špecifické chovanie selekcie (napríklad vhodný selekčný tlak alebo konvergencia populácie). Takými požiadavkami môžu byť napríklad výskyt hodnôt vhodnosti iba v rámci nejakého intervalu, potláčanie prítomnosti rovnakých vhodností v populácii, alebo dokonca presný tvar distribúcie vhodnosti. Preto pred samotným vzorkovaním môže dôjsť k zmene vhodnosti jedincov, ktorá sa označuje ako premapovanie vhodnosti.

Pri tomto premapovaní je vhodnosť všetkých jedincov zmenená. Pre nejakého jedinca  $a_i$  sa jeho vhodnosť  $\Phi(a_i(t))$  nahradí novou vhodnosťou  $\Phi'(a_i(t))$  a táto nová vhodnosť sa použije počas selekcie. Životnosť tejto novej vhodnosti je značne obmedzená – platí iba po dobu selekcie. Ak daný jedinec je vybraný do úlohy rodiča (či už raz alebo viackrát), v skupine rodičov sa vracia k pôvodnej hodnote svojej vhodnosti. Podobne, ak jedinec ešte bude figurovať v niektorej časti evolučného cyklu (napríklad počas formovania novej generácie), tak iba so svojou pôvodnou vhodnosťou.

Principiálne je možné použiť premapovanie vhodnosti v spojení s ľubovoľnou selekčnou metódou. Ak však premapovanie vhodnosti je rýdzomonotónne (a to prevažná väčšina premapovacích postupov je), teda ak platí

$$\Phi(a_i(t)) > \Phi(a_j(t)) \implies \Phi'(a_i(t)) > \Phi'(a_j(t)) \quad (6.34)$$

tak má zmysel iba v kombinácii so selekčnou metódou uvažujúcou absolútne hodnoty vhodnosti jedincov a nie iba relatívne hodnoty (teda nemá zmysel pre metódy založené na porovnávaní vhodností jedincov alebo ich triedení podľa vhodnosti).

Príkladom nemonotónneho premapovania vhodnosti je disruptívna selekcia [23], kde premapovanie sa deje podľa vzťahu

$$\Phi'(a_i(t)) = |\Phi(a_i(t)) - \bar{\Phi}(t)| \quad (6.35)$$

Takéto premapovanie potláča replikáciu jedincov s priemernou vhodnosťou a preferuje extrémnejšie jedince – či už v dobrom alebo zlom zmysle. Je vhodná pre úlohy, kde hľadané riešenie je obklopené veľmi zlými riešeniami (takýto problém sa označuje ako problém hľadania ihly v kope sena), a teda získanie hľadaného riešenia je najjednoduchšie možné získať modifikáciou zlého riešenia.

Monotónne premapovania sa z vyššie uvedeného dôvodu prakticky používajú výhradne v kombinácii s proporcionálnou selekciou založenou na vhodnosti. Cieľom je odstránenie alebo potlačenie niektorých nevýhod, uvedených v kapitole 6.1.1.2. Všetky tieto metódy je možné rozdeliť do dvoch skupín: skupina metód zabezpečujúcich zmenu mierky a metód nahrádzajúcich pôvodnú vhodnosť novou vhodnosťou.

### 6.2.1 Premapovanie škálovaním

Vo všeobecnosti sa jedná o transformáciu hodnôt vhodnosti nejakým lineárnym spôsobom<sup>10</sup>. Cieľom je udržiavať hodnoty vhodnosti v nejakých stanovených medziach, ktoré predstavujú vhodný kompromis medzi snahou neustále udržiavať dostatočný selekčný tlak pre preferovanie nadpriemerných jedincov a medzi snahou kontrolovať prítomnosť superjedincov. Inak povedané, preferencia by mala byť čo najväčšia, avšak zase nie taká veľká, aby hľadanie predčasne skonvergovalo bez nájdania vhodného riešenia.

**Windowing** Pri tejto metóde sa od vhodnosti každého jedinca odpočíta nejaká konštanta. V tom najjednoduchšom prípade sa vhodnosť najhoršieho jedinca v populácii považuje za novú základňu a o túto hodnotu sa znížia vhodnosti ostatných:

$$\Phi'(a_i(t)) = \Phi(a_i(t)) - \Phi_{min}(t) \quad (6.36)$$

---

<sup>10</sup>Samozrejme, škálovať možno aj nelineárnym spôsobom, napr. logaritmicky alebo exponenciálne, avšak kvôli nepatrnému rozšíreniu týchto metód sa nimi nebudeme zaoberať.

Takýmto spôsobom sa neustále udržiava dostatočná hodnota selekčného tlaku. Príjemným je aj vyriešenie problému záporných vhodností.

Pri použití proporcionálnej selekcie založenej na vhodnosti by teraz pravdepodobnosti selekcie neboli určované podľa (6.11) ale podľa

$$p_s(a_i(t)) = \frac{\Phi(a_i(t)) - \Phi_{min}(t)}{\sum_{j=1}^{\mu} \Phi(a_j(t)) - \mu\Phi_{min}(t)} = \frac{\Phi(a_i(t)) - \Phi_{min}(t)}{\mu(\bar{\Phi}(t) - \Phi_{min}(t))} \quad (6.37)$$

Z porovnania tohto vzťahu a (6.11) vyplýva, že po premapovaní pravdepodobnosť selekcie nadpriemerných jedincov je väčšia než pred premapovaním, a pravdepodobnosť selekcie podpriemerných jedincov zase nižšia než pred premapovaním – teda zvýši sa preferencia vhodnejších jedincov voči menej vhodným.

Najhorší jedinec bude mať po premapovaní nulovú vhodnosť, čo ho vylúči z možnosti stať sa rodičom. Nie vždy je to považované za vhodné, niekedy by bolo vhodné dať aj tomu najhoršiemu jedincovi nejakú (aj keď veľmi malú) šancu stať sa rodičom. Preto sa vytvára okno  $w$  posledných generácií (vrátane aktuálnej) a v rámci tohto okna sa hľadá najhorší jedinec. Jeho vhodnosť je potom použitá ako nová základňa. Pretože vo všeobecnosti sa počas jednotlivých generácií vhodnosti jedincov zvyšujú, je pravdepodobné, že najhorší jedinec nájdený v rámci okna nebude z aktuálnej generácie. Následkom toho potom aj najhorší jedinec v aktuálnej generácii bude mať pozitívnu vhodnosť. Čím väčšia šírka okna sa použije, tým nižšie bude položená základňa a tým väčšie vhodnosti budú mať jedince v populácii.

**Sigma škálovanie** Vychádza z podobnej transformácie ako predchádzajúca metóda, avšak pri hľadaní novej základne zohľadňuje štatistické vlastnosti distribúcie vhodnosti v populácii. Takýto postup umožňuje prekonať problém obzvlášť zlých jedincov, ktorí by udržiavali základňu príliš nízko, čím by redukovali tlak na výber nadpriemerných jedincov.

Jedna z možných realizácií stanovuje novú základňu  $sf$  smerodajných odchýlok pod priemernou hodnotou. Teda jedince po premapovaní budú mať vhodnosť

$$\Phi'(a_i(t)) = \Phi(a_i(t)) - (\bar{\Phi}(t) - \sigma(t) sf) \quad (6.38)$$

kde  $\sigma$  reprezentuje smerodajnú odchýlku. Priemerná vhodnosť populácie sa v tomto prípade zmení na hodnotu  $\Phi'(t) = \sigma(t) sf$ .

Jedinec, ktorého vhodnosť je jednu smerodajnú odchýlku nad priemernou vhodnosťou, bude mať po premapovaní vhodnosť  $(sf + 1)\sigma(t)$  a pri použití proporcionálnej selekcie založenej na vhodnosti bude jeho očakávaný počet

rodičov

$$\frac{\rho \text{ sf} + 1}{\mu \text{ sf}} \quad (6.39)$$

Parameter  $\text{sf}$  sa označuje ako škálovací faktor a je pomocou neho možné ovplyvňovať hodnotu selekčného tlaku. Typickými hodnotami sú hodnoty z intervalu  $< 1,5 >$ , pričom menšia hodnota parametra  $\text{sf}$  znamená silnejší selekčný tlak.

Táto metóda premapovania explicitne rieši problém stagnácie hľadania. V neskorších fázach evolúcie typicky všetky jedince majú pomerne vysokú vhodnosť, pričom rozdiely vo vhodnosti jedincov sú malé. Následkom toho aj hodnota smerodajnej odchýlky je malá, čo pri vysokej priemernej vhodnosti umožňuje položiť základňu dostatočne vysoko. Aby jedinec získal viac kópií ako priemerný jedinec, pri vysokej základni nepotrebuje mať vhodnosť až o toľko väčšiu od priemerného jedinca ako pri veľkých hodnotách smerodajnej odchýlky, spôsobujúcich umiestnenie základne pomerne nízko. A teda na konci behu algoritmu aj malé rozdiely vo vhodnosti jedincov umožňujú vyvinúť dostatočne silný selekčný tlak a tým predísť náhodnému hľadaniu.

Na rozdiel od predchádzajúcej metódy premapovania, teraz do určitej miery je riešený aj problém výskytu superjedincov. Na začiatku hľadania je smerodajná odchýlka pomerne veľká, a až v neskorších generáciách sa zmenšuje. Keďže problém superjedincov sa vyskytuje v ranných štádiách hľadania (keď vhodnosť ostatných jedincov je nízka), tak práve vtedy veľká hodnota smerodajnej odchýlky udržuje základňu dostatočne nízko, takže preferencia superjedince nie je príliš veľká. Navyše, výskytom superjedince sa ešte hodnota smerodajnej odchýlky zväčšuje, čo posúva základňu ešte o niečo nižšie.

Metóda však nezaručuje nezápornosť vhodností. V tomto prípade je možné pre jedince, ktoré po premapovaní budú mať zápornú vhodnosť, zmeniť vhodnosť na nulovú. Alebo im priradiť nejakú malú pozitívnu vhodnosť, aby aj tieto jedince mali šancu stať sa rodičmi.

Iným spôsobom ako realizovať sigma škálovanie je použiť

$$\Phi'(a_i(t)) = \begin{cases} 1 + \frac{\Phi(a_i(t)) - \bar{\Phi}(t)}{\sigma(t) \text{ sf}} & \sigma(t) \neq 0 \\ 1 & \text{inak} \end{cases} \quad (6.40)$$

Opticky síce tento spôsob vyzerá inak ako predchádzajúci (v tomto prípade je priemerná vhodnosť po premapovaní 1), avšak v oboch prípadoch pomer  $\Phi'(a_i(t))/\bar{\Phi}'(t)$  je pre každého jedinca ten istý. To pri použití proporčionalnej selekcie založenej na vhodnosti zaručuje rovnaké chovanie.



**Lineárne škálovanie** Pri tomto škálovaní sú hodnoty vhodnosti transformované tak, že najlepší jedinec získa špecifikovaný očakávaný počet rodičov. Ostatné hodnoty sú zmenené tak, aby vhodnosť jedinca s priemernou vhodnosťou bola po premapovaní rovná 1. Takýmto spôsobom sa nielenže zabezpečí dostatočný selekčný tlak, ale zároveň sa aj obmedzí preferovanie superjedincov, čím sa zabráni ich prílišnej dominancii v populácii.

Vhodnosť po premapovaní je možné určovať pomocou vzťahu

$$\Phi'(a_i(t)) = 1 + (sf - 1) \frac{\Phi(a_i(t)) - \bar{\Phi}(t)}{\Phi_{max}(t) - \bar{\Phi}(t)} \quad (6.41)$$

kde  $\Phi_{max}(t)$  označuje vhodnosť najvhodnejšieho jedinca v populácii. Vzhľadom na to, že priemerná vhodnosť po premapovaní bude 1 a vhodnosť najlepšieho jedinca  $sf$ , tak tento parameter zároveň určuje očakávaný počet rodičov najlepšieho jedinca, ak požadovaný počet rodičov sa rovná veľkosti populácie a ako selekčná metóda sa použije proporcionálna selekcia založená na vhodnosti. Typicky je hodnota tohto parametra z intervalu  $< 1.2, 2 >$  a jeho zmenou je možné ovplyvňovať veľkosť selekčného tlaku.

Problémom môže pri tomto premapovaní byť to, že jedince, ktorých vhodnosť je menšia ako priemer, môžu nadobudnúť zápornú vhodnosť aj keď ich vhodnosť pred premapovaním nebola záporná. Okrem dodatočného priradenia nulovej vhodnosti týmto jedincom je možné nastaviť hodnotu parametra  $sf$  tak, aby k takémuto prípadu nedošlo. Ak by sme pripustili, že najhoršiemu jedincovi priradíme nulovú vhodnosť, tak potom pre hodnotu parametra stačí zvoliť

$$sf = 1 + \frac{\Phi_{max}(t) - \bar{\Phi}(t)}{\bar{\Phi}(t) - \Phi_{min}(t)} = \frac{\Phi_{max}(t) - \Phi_{min}(t)}{\bar{\Phi}(t) - \Phi_{min}(t)} \quad (6.42)$$

kde  $\Phi_{min}(t)$  označuje vhodnosť najhoršieho jedinca v populácii. Táto hodnota je hraničnou hodnotou, keď ešte nedochádza ku generovaniu zápornej vhodnosti.

Potom pri samotnom premapovaní vhodnosti sa zvolená hodnota  $sf$  porovná s hodnotou určenou podľa predchádzajúceho vzťahu. Ak je väčšia, tak je ju potrebné znížiť na hraničnú hodnotu. Ak nie je väčšia, nie je ju potrebné upravovať. Takýmto nastavením sa však znižuje miera schopnosti regulovať selekčný tlak. Môže totiž nastať prípad, že neustály výskyt príliš zlého jedinca v populácii môže dlhodobo brániť nastaveniu vhodného selekčného tlaku. V tomto prípade môže pomôcť monitorovanie populácie a odstránenie tohto jedinca.

### 6.2.2 Premapovanie zotriedením

Pri zotriedeniach sa pôvodná vhodnosť jedincov využije na to, že tieto sa zotriedia podľa nej. Každý jedinec získa určité poradie v postupnosti jedincov. Pre naše účely predpokladajme, že zoradenie bude vzostupné, teda že index 1 bude patriť jedincovi s najhoršou vhodnosťou a naopak jedinec s najlepšou vhodnosťou bude mať index  $\mu$ . Dôležité je to, že jedno poradie môže patriť iba jednému jedincovi – a to aj v prípade, že viaceré jedince majú rovnakú vhodnosť. V tomto prípade jeden jedinec bude preferovaný viac ako iný jedinec, napriek ich rovnakej vhodnosti.

Po zotriedení jedince získajú novú vhodnosť, ktorá nie je odvodená od absolútnej hodnoty ich pôvodnej vhodnosti ale od ich poradia. Toto umožňuje zvoliť také hodnoty, aby sa kompenzovali všetky nevýhody proporcionálnej selekcie založenej na vhodnosti – odstrániť záporné hodnoty vhodnosti, zabezpečiť dostatočný selekčný tlak ako v ranných tak aj neskorších fázach evolučného vývoja, ako aj odstrániť nebezpečenstvo predčasnej konvergenie vplyvom superjedincov.

Tento spôsob premapovania zachováva usporiadanie jedincov podľa pôvodnej vhodnosti (s výnimkou už spomenutého prípadu rovnakých vhodností, keď po premapovaní budú mať jedince vhodnosti rôzne).

**Lineárne zotriedenie** Pri tomto spôsobe premapovania sa stanoví vhodnosť  $\tau^+$  pre najlepšieho jedinca, rovnako aj vhodnosť  $\tau^-$  pre najhoršieho jedinca. Vhodnosti ostatných jedincov sa potom určia podľa lineárneho vzťahu

$$\Phi'(a_i(t)) = \left( \tau^- + (\tau^+ - \tau^-) \frac{i-1}{\mu-1} \right) \quad (6.43)$$

kde  $i$  označuje poradie jedinca v zotriedenej populácii jedincov. Keďže smernica umožňuje riadiť veľkosť preferencie lepších jedincov voči horším, pomocou dvoch volených parametrov je možné ovplyvňovať veľkosť selekčného tlaku.

Navyše sa kladie ešte dodatočná podmienka, aby nová vhodnosť priemerného jedinca (nie toho, ktorý mal pôvodne priemernú vhodnosť, ale toho ktorý bude priemerný po premapovaní) bola rovná 1

$$\frac{\tau^+ + \tau^-}{2} = 1 \quad (6.44)$$

z čoho vyplýva vzájomná závislosť oboch volených hodnôt. Navyše, aby sa zachovala podmienka nezáporných vhodností, musí platiť  $\tau^- \in \langle 0, 1 \rangle$  a  $\tau^+ \in \langle 1, 2 \rangle$ .

Pri zachovaní týchto podmienok distribúcia vhodnosti po premapovaní bude rovnaká ako podľa vzťahu (6.12). Stačí, ak sa položia rovnosti  $\tau^- = \xi$  a  $\tau^+ = 2 - \xi$ . Znamená to teda, že kombinácia premapovania lineárnym zotriedením s následným použitím proporcionálnej selekcie založenej na vhodnosti dokáže produkovať rovnaké chovanie selekcie ako binárny turnaj (pri vhodnej voľbe hodnoty  $\tau^-$ ).

Lineárne zotriedenie teda vedie na statickú selekciu, ktorá v závislosti od toho, či parameter  $\tau^-$  nadobúda nulovú alebo nenulovú hodnotu, spôsobuje vymieranie alebo uchovávaciu selekciu.

**Exponenciálne zotriedenie** Pri tomto type zotriedenia vhodnosti jedincov budú exponenciálne vážené. Najlepší jedinec bude mať po premapovaní vhodnosť 1, druhý najlepší vhodnosť  $c$ , tretí najlepší vhodnosť  $c^2$ , atď. Premapovanie je teda možné znázorniť pomocou vzťahu

$$\Phi'(a_i(t)) = c^{\mu-i} \quad (6.45)$$

Pre účel bázy zotriedenia sa teda použije hodnota parametra  $c$ , ktorá je väčšia ako 0 a menšia ako 1. Čím viac sa jeho hodnota bude blížiť číslu 1, tým menej sa bude prejavovať exponenciálny charakter premapovania.

Pri použití proporcionálnej selekcie založenej na vhodnosti je možné pravdepodobnosti selekcie jedincov vyjadriť ako

$$p_s(a_i(t)) = \frac{c^{\mu-i}}{\sum_{j=1}^{\mu} c^{\mu-j}} \quad (6.46)$$

čo pri využití vzťahu

$$\sum_{j=0}^n c^j = \frac{c^{n+1} - 1}{c - 1} \quad (6.47)$$

vedie na pravdepodobnosti

$$p_s(a_i(t)) = \frac{c - 1}{c^{\mu} - 1} c^{\mu-i} \quad (6.48)$$

ktoré nie sú závislé ani na pôvodnej vhodnosti jedincov ani na konkrétnej generácii – teda takéto premapovanie vedie na statickú uchovávaciu selekciu.

Ak by sa ako báza uvažovala hodnota  $(1 - 1/\mu)^q$ , tak dosadením do predchádzajúceho vzťahu by sa získalo:

$$p_s(a_i(t)) = \frac{\left(1 - \frac{1}{\mu}\right)^q - 1}{\left[\left(1 - \frac{1}{\mu}\right)^q\right]^{\mu} - 1} \left[\left(1 - \frac{1}{\mu}\right)^q\right]^{\mu-i}$$

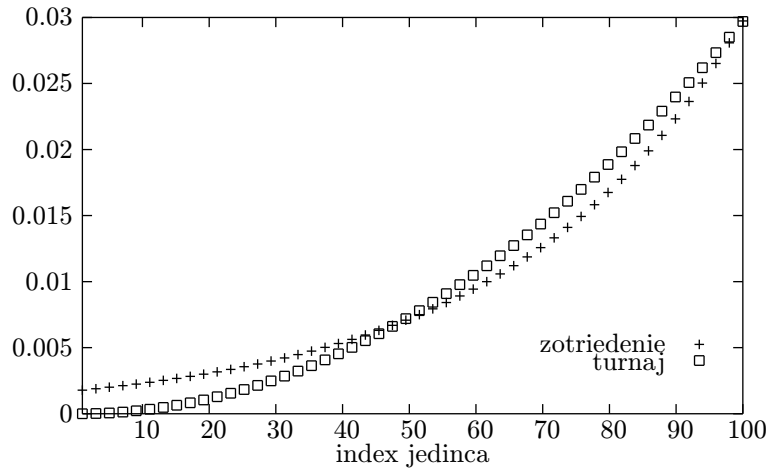
$$= \frac{\left[\left(1 - \frac{1}{\mu}\right)^{\mu-i+1}\right]^q - \left[\left(1 - \frac{1}{\mu}\right)^{\mu-i}\right]^q}{\left[\left(1 - \frac{1}{\mu}\right)^\mu\right]^q - 1} \quad (6.49)$$

Pri uvažovaní aproximácie  $(1 - x)^y \approx 1 - xy$  pre malé hodnoty  $x$  je možné vzťah ďalej upraviť na

$$p_s(a_i(t)) \approx \frac{\left(1 - \frac{\mu-i+1}{\mu}\right)^q - \left(1 - \frac{\mu-i}{\mu}\right)^q}{\left(1 - \frac{\mu}{\mu}\right)^q - 1} = \frac{1}{\mu^q} ((i)^q - (i-1)^q) \quad (6.50)$$

čo je vlastne pravdepodobnosť selekcie q-árneho turnaja v prípade, že všetky jedince majú navzájom rôznu vhodnosť (kap. 6.1.2.1). Premapovanie exponenciálnym zotriedením (s následným použitím proporcionálnej selekcie založenej na vhodnosti) je teda možné považovať za aproximáciu q-árneho turnaja – tým presnejšiu, čím bude viac jedincov v populácii.

pravdep. selekcie



Obr. 6.4: Porovnanie pravdepodobností selekcie pre exponenciálne zotriedenie ( $c = 0.972$ ) a ternárny turnaj.

Exponenciálne zotriedenie sa vyznačuje tým, že nie je také prísne k najhorším jedincom ako ostatné metódy. Toto ilustruje obr. 6.4, kde je porovnané s ternárnym turnajom. Parametre oboch metód boli nastavené tak, aby pravdepodobnosť selekcie najlepšieho jedinca bola rovnaká v oboch prípadoch (teda nie aby exponenciálne premapovanie bolo čo najlepšou apro-

ximáciou ternárneho turnaja). Šance stať sa rodičom sú pre najhorších jedincov značne väčšie pri exponenciálnom zotriedení – a to na úkor jedincov priemerných a lepších ako priemerné jedince. Toto platí aj pre porovnanie exponenciálneho zotriedenia voči lineárnemu zotriedeniu či orezaniu.

### 6.3 Porovnanie selekčných metód

Pri veľkom počte použiteľných metód vystupuje do popredia otázka ich porovnania, ktorá rezonuje v odbornej literatúre, venovanej problematike selekcie, už značne dlhú dobu. Objavili sa mnohé takéto pokusy. Niektoré sa snažili stavať svoje závery na výsledkoch teoretických odvodení, iné zase empirickým spôsobom overovali chovanie týchto metód pri riešení konkrétnych úloh – či už “laboratórnych” alebo reálnych z praxe. Výsledkom boli rôzne (často sa rozchádzajúce) doporučenia, ktoré si v žiadnom prípade nerobili nárok na všeobecnú platnosť. Problémom totiž je to, že vlastne pri riešení konkrétnej úlohy ani vopred nie je jasné, ktoré vlastnosti selekčných metód by bolo vhodné pre danú úlohu preferovať.

Najčastejšie sa ako porovnávacie kritérium uvažuje veľkosť selekčného tlaku, ktorý je tá-ktorá metóda schopná vyvinúť. Pritom sa však nedá vopred určiť, aká hodnota selekčného tlaku by mala byť preferovaná – aká hodnota je vhodná pre riešenie nejakého problému. V [4] sú ilustrované experimenty s rôznymi tvarmi plochy vhodnosti, pričom pri niektorých tvaroch zvyšovanie selekčného tlaku poskytlo lepšie výsledky zatiaľ čo pri iných viedlo k horším riešeniam.

Nevýhodou selekčného tlaku je, že sa nemeria priamo, ale prostredníctvom faktorov, ktorých hodnoty majú súvis s veľkosťou produkovaného selekčného tlaku.

Jedným z takýchto faktorov<sup>11</sup> je selekčná intenzita, ktorá sa zameriava

<sup>11</sup> Iným často využívaným faktorom je doba prevzatia – počet generácií nutných k tomu, aby pri opakovaných selekciách najlepší jedinec z pôvodnej populácie celkom ovládol zloženie aktuálnej populácie – teda aby počet jeho kópií v aktuálnej populácii dosiahol  $\mu$  (v niektorých definíciách sa požaduje iba  $\mu - 1$ ). Vychádza sa pritom z populácie obsahujúcej dve skupiny jedincov – jedného najlepšieho jedinca a  $\mu - 1$  horších jedincov.

Pri samotnom experimente sa zo skupiny  $\mu$  jedincov pomocou zvolenej selekčnej metódy vytvorí nová skupina jedincov rovnakej veľkosti, z nej potom ďalšia, z nej ďalšia, atď. Až pokiaľ sa nedosiahne požadované zloženie vytvorenej skupiny jedincov. Pritom pri použití selekčnej metódy, ktorá nie je elitistická, sa môže stať, že najlepší jedinec sa z populácie stratí. Vtedy ho je nutné naspäť do populácie vložiť, napr. namiesto aktuálne najlepšieho jedinca. Počet takýchto zásahov tiež môže slúžiť ako kritérium pre výber vhodnej selekčnej metódy.

Čím bude selekčný tlak väčší, tým bude aj najlepší jedinec viac preferovaný voči os-

na zmenu priemernej vhodnosti skupiny jedincov následkom použitia nejakej selekčnej metódy. Podľa [6] je definovaná ako pomer

$$\frac{\bar{\Phi}^* - \bar{\Phi}}{\sigma} \quad (6.51)$$

kde  $\bar{\Phi}^*$  reprezentuje očakávanú hodnotu priemernej vhodnosti v skupine rodičov. Prítomnosť smerodajnej odchýlky v menovateli zabezpečuje, že selekčná intenzita je bezrozmernou veličinou.

Čím väčšia je hodnota selekčnej intenzity, tým viac sa zvýšila hodnota priemernej vhodnosti, a tým viac boli preferované vhodnejšie jedince voči menej vhodným. A tým väčšia hodnota selekčného tlaku bola dosiahnutá. Teda veľké hodnoty selekčnej intenzity signalizujú súčasne aj veľký selekčný tlak, a naopak.

Keďže však selekčná intenzita závisí od štatistických vlastností distribúcie vhodnosti, iná intenzita bude dosahovaná tou istou metódou pri riešení rôznych úloh, dokonca aj v rôznych fázach riešenia tej istej úlohy. To znamená, že konkrétna hodnota selekčnej intenzity nemá nejaký praktický význam. Je ju však možné použiť pre porovnanie dvoch selekčných metód – ktorá z nich (pre tú istú distribúciu vhodnosti) dosiahne väčšiu selekčnú intenzitu, tá vyvíja aj väčší selekčný tlak.

Na rozdiel od v literatúre často používanej distribúcie vhodnosti v tvare normálnej distribúcie s nulovou strednou hodnotou a jednotkovou smerodajnou odchýlkou (v tomto prípade sa meraná hodnota selekčnej intenzity označuje ako štandardizovaná selekčná intenzita), použijeme distribúciu vhodnosti, ktorá sa vyskytla pri riešení konkrétneho problému<sup>12 13</sup>.

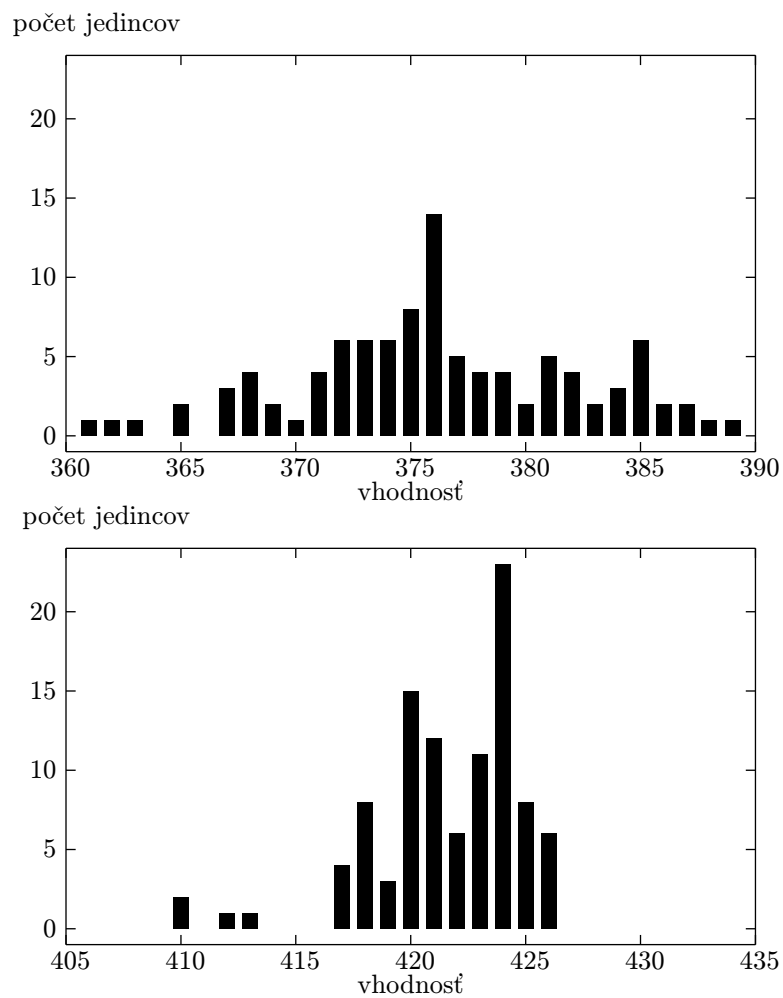
Obr. 6.5 zobrazuje dve distribúcie vhodnosti. Horná reprezentuje prvotnú populáciu ( $t = 0$ ), ktorá ešte nebola modifikovaná v priebehu evolučného procesu. Dolná distribúcia naopak zobrazuje pomery v populácii po určitej dobe pôsobenia evolučného algoritmu. Takýto tvar distribúcie vhodnosti vzniká pomerne často pri riešení ťažkých úloh z pôvodne zvonovitého tvaru

---

tatným jedincom, a tým skôr vyplní celú populáciu jedincov. Teda malé hodnoty doby prevzatia zodpovedajú veľkým hodnotám selekčného tlaku, a naopak dlhá doba prevzatia signalizuje malú hodnotu selekčného tlaku. Vzťahy pre výpočet doby prevzatia pre niektoré metódy sú odvodené v [4]. Empirické merania realizované v rámci rozsiahleho experimentálneho porovnávania selekčných metód možno nájsť zase v [16].

<sup>12</sup>Jednalo sa o riešenie 3SAT problému, kde bolo potrebné nastaviť hodnoty premenných tak, aby boli splnené všetky definované ohraničenia. Vhodnosť určovala počty ohraničení, splnených kombináciami hodnôt premenných reprezentovaných jedincami.

<sup>13</sup>Robíme tak s nádejou, že prezentované výsledky budú použiteľné pri riešení praktických úloh.



Obr. 6.5: Distribúcie vhodnosti použité pre porovnanie selekčných metód

vtedy, ak zlepšenie menej vhodného jedinca je pravdepodobnejšie ako zlepšenie vhodnejšieho jedinca. Body sa postupne presúvajú z ľavej na pravú stranu, a keďže riešenie ešte nebolo nájdené, tak sa tam hromadia, a čaká sa, kedy nastane zlepšenie najlepšej vhodnosti v populácii.

Sel.int.	0.0057	0.0228	0.0398	0.0563	0.0844	0.1028	0.1160
$q$				2	3	4	5
$T$	0.98	0.88	0.77	0.65	0.46	0.37	0.32
$\tau^-$	0.9	0.6	0.3	0.01			
$c$	0.998	0.992	0.986	0.979	0.966	0.955	0.944
$sf$ (lš)	1.12	1.49	1.85				
$sf$ (sš)	1.76	0.44	0.25				

Tabuľka 6.1: Hodnoty parametrov selekčných metód pre dosiahnutie vybraných hodnôt selekčnej intenzity (použitá horná distribúcia z obr. 6.5).

Tabuľka 6.1 reprezentuje selekčné intenzity dosahované jednotlivými selekčnými metódami pri rôznych hodnotách ich riadiacich parametrov, pričom ako prvotná distribúcia vhodnosti bola použitá horná distribúcia na obr. 6.5. Tabuľka umožňuje jednak určiť, pri ktorých hodnotách parametrov selekčné metódy dosahujú rovnakú selekčnú intenzitu, a jednak indikuje, pre aký rozsah hodnôt selekčnej intenzity sú tieto metódy použiteľné. Prázdne políčko v tabuľke indikuje, že daná selekčná intenzita sa nedala dosiahnuť<sup>14</sup>. Jednotlivé metódy sú reprezentované svojimi parametrami:  $q$  je veľkosť  $q$ -árneho turnaja,  $T$  je prah pri orezaní,  $\tau^-$  je vhodnosť najhoršieho jedinca po premapovaní pomocou premapovania lineárnym zotriedením, a  $c$  je báza použitá pri premapovaní vhodnosti exponenciálnym zotriedením. Pre ilustráciu sú uvedené aj dve metódy premapovania vhodnosti pomocou škálovania ( $sf$  je škálovací faktor pre lineárne škálovanie resp. pre sigma škálovanie).

Z tabuľky je zrejmé, že v určitých rozsahoch hodnôt selekčnej intenzity sú jednotlivé metódy navzájom zameniteľné (pri vhodnom nastavení príslušných parametrov)<sup>15</sup>. Pri pohľade na tabuľku je vhodné si však uvedomiť:

<sup>14</sup>Pri tvorbe tabuľky sa postupovalo tak, že najprv boli volené hodnoty parametra  $q$ , a v tých stĺpcoch, kde tento nemá žiadnu hodnotu, zase hodnoty parametra  $\tau^-$ . Pomocou týchto hodnôt sa určili všetky hodnoty selekčnej intenzity. Ostatné hodnoty parametrov metód boli nastavené tak, aby dané metódy čo najpresnejšie dosahovali príslušné hodnoty selekčnej intenzity (teda nedosahujú presne uvedenú hodnotu selekčnej intenzity, ale hodnotu blízku).

<sup>15</sup>Zameniteľné kombinácie hodnôt parametrov z tabuľky z veľkej časti súhlasia s kombináciami hodnôt parametrov v [6], kde bola skúmaná štandardizovaná selekčná intenzita.



- Výsledky pre premapovanie vhodnosti škálovacími metódami slúžia skutočne iba pre ilustráciu, pretože sú to dynamické selekčné metódy závisiace nielen od tvaru distribúcie vhodnosti ale aj od konkrétnych hodnôt vhodnosti.
- Pre orezanie netreba zabúdať, že súčin  $T\mu$  je potrebné zaokrúhliť na celé číslo – a teda pri malých populáciách hodnota  $T = 0.98$  môže znamenať, že sa uvažujú všetky jedince, čo však znamená úplnú absenciu selekčného tlaku.
- Pri premapovaní lineárnym škálovaním by pri snahe ďalej zvyšovať škálovací faktor bolo nutné riešiť problém výskytu záporných vhodností po premapovaní.
- Sigma škálovanie sa nezvykne používať s hodnotou škálovacieho faktoru menšou ako jedna (pri hodnote 0.25 už bolo nutné po premapovaní zápornú vhodnosť nahrádzať vhodnosťou nulovou).

Okrem uvedených hodnôt, samotná proporcionálna selekcia založená na aktuálnych hodnotách vhodnosti dosiahla veľmi nízku hodnotu selekčnej intenzity 0.0016, ktorá sa po premapovaní vhodnosti prostredníctvom windowingu zvýšila na 0.0398 (keďže veľkosť okna bola jednotková, najhoršia vhodnosť po premapovaní bola nulová – to je to isté ako keď sa pri premapovaní vhodnosti lineárnym škálovaním použije taká hodnota škálovacieho faktora, že najhorší jedinec nadobúda nulovú hodnotu).

V tabuľke nie je zachytený celý možný rozsah selekčných intenzít. Je možné vyvinúť aj vyššie intenzity (napr. q-árnym turnajom, orezaním alebo premapovaním vhodnosti exponenciálnym zotriedením) resp. intenzity nižšie (napr. premapovaním vhodnosti pomocou lineárneho zotriedenia či pomocou škálovania).

Tabuľka 6.2 reprezentuje selekčné intenzity dosahované jednotlivými selekčnými metódami pri rôznych hodnotách ich riadiacich parametrov, pričom však ako prvotná distribúcia vhodnosti bola použitá dolná distribúcia na obr. 6.5. Keďže sa zmenil tvar použitej distribúcie vhodnosti, aj dosahované hodnoty selekčnej intenzity sú iné ako v predchádzajúcom prípade. Kombinácie hodnôt parametrov jednotlivých selekčných metód pre dosahovanie selekčnej intenzity sa zmenili iba nepatrne. Najväčšie zmeny podľa očakávania postihli dynamické selekčné metódy:

---

Tam sa však vďaka použitej prvotnej distribúcii dosahovali rádovo väčšie hodnoty selekčnej intenzity.

Sel.int.	0.0053	0.0214	0.0374	0.0530	0.0742	0.0861	0.0939
$q$				2	3	4	5
$T$	0.985	0.91	0.80	0.68	0.53	0.41	0.32
$\tau^-$	0.9	0.6	0.3	0.01			
$c$	0.998	0.992	0.985	0.977	0.963	0.952	0.942
$sf$ (lš)	1.07	1.29					
$sf$ (sš)	1.89	0.47	0.23				

Tabuľka 6.2: Hodnoty parametrov selekčných metód pre dosiahnutie vybraných hodnôt selekčnej intenzity (použitá dolná distribúcia z obr. 6.5).

- Pri premapovaní vhodnosti lineárnym škálovaním už nebolo možné dosiahnuť rovnakú selekčnú intenzitu ako pri premapovaní vhodnosti lineárnym zotriedením s hodnotou  $\tau^- = 0.3$  – záporná vhodnosť pri premapovaní sa teraz začala objavovať pri menších hodnotách škálovacieho faktoru než v predchádzajúcom prípade.
- Pri najmensej hodnote škálovacieho faktoru pre sigma škálovanie sa objavilo viac jedincov so zápornou vhodnosťou po premapovaní vhodnosti ako v predchádzajúcom prípade.

Selekčná intenzita proporcionálnej selekcie založenej na aktuálnych hodnotách vhodnosti klesla na hodnotu 0.0007, čo je v porovnaní s hodnotami dosahovanými inými metódami skoro zanedbateľná hodnota.

Z hľadiska veľkosti produkovanej selekčnej intenzity (a teda vlastne aj selekčného tlaku) medzi metódami selekcie je menej rozdielov, než by sa na prvý pohľad zdalo. Preto je možné pre výber použiť aj iné kritériá. Jedným z nich je strata rôznorodosti, založená na tom, že výber viacerých kópií nejakého jedinca vedie k tomu, že iný jedinec nie je selektovaný vôbec. Takýmto spôsobom dochádza k strate “materiálu” – k určitej redukcii rôznorodosti, nakoľko skupina rodičov vykazuje menšiu rôznorodosť (väčšiu uniformnosť) ako populácia jedincov z ktorej sú rodičia vyberaní<sup>16</sup>. Podľa [6] je strata rôznorodosti definovaná ako proporcia jedincov populácie, ktorí nie sú vybraní do úlohy rodičov. Vo všeobecnosti platí, že jej veľkosť je úmerná veľkosti selekčného tlaku. Čím je silnejšia preferencia jednej skupiny jedincov, tým menej jedincov z druhej skupiny sa stane rodičom. No nemožno očakávať,

<sup>16</sup>Niekedy je situácia tak zlá, že je nutné použiť špeciálne metódy pôsobiace proti strate rôznorodosti (kapitola 10).

že dve selekčné metódy produkujúce rovnako veľký selekčný tlak budú spôsobovať rovnakú stratu rôznorodosti.

V [45] je táto strata rôznorodosti určovaná podľa vzťahu

$$\frac{1}{\mu} \sum_{j=1}^{\mu} (1 - p_s(a_i))^{\varrho} \quad (6.52)$$

Tento vzťah platí pre prípad, keď selekcia je realizovaná ako séria  $\varrho$  výberov – jeden výber pre každého rodiča (napr.  $q$ -árny turnaj alebo stochastické vzorkovanie s náhradou). Iná situácia je však, ak jedným výberom sú selektovaní všetci rodičia odrazu, ako v prípade použitia stochastického univerzálneho vzorkovania. Pre toto vzorkovanie platí vzťah

$$\frac{1}{\mu} \sum_{j=1}^{\mu} \max(1 - p_s(a_i) * \varrho, 0) \quad (6.53)$$

Pre oba vzťahy platí, že ak je použitá statická selekčná metóda, potom pravdepodobnosti selekcie jedincov sa nemenia a hodnota straty rôznorodosti nezávisí ani od času, ani od konkrétnych hodnôt vhodností jedincov.

	Selekčná intenzita						
	0.0057	0.0228	0.0398	0.0563	0.0844	0.1028	0.1160
metóda	Strata rôznorodosti						
Turnaj				0.248	0.385	0.472	0.535
				0.430	0.502	0.559	0.604
Orezanie	0.020	0.120	0.230	0.350	0.540	0.630	0.680
	0.371	0.401	0.438	0.488	0.591	0.654	0.693
Lin. zotr.	0.025	0.101	0.177	0.250			
	0.367	0.376	0.397	0.431			
Exp. zotr.	0.025	0.099	0.172	0.250	0.375	0.459	0.525
	0.367	0.375	0.395	0.425	0.494	0.549	0.598
Lin. škál.	0.022	0.089	0.155				
	0.367	0.376	0.397				
Sigma škál.	0.022	0.088	0.156				
	0.367	0.376	0.397				

Tabuľka 6.3: Strata rôznorodosti selekčných metód pre vybrané hodnoty selekčnej intenzity.

Nakoľko je obtiažne porovnávať stratu rôznorodosti dosiahnutú rôznymi metódami (pretože každá z nich má iný riadiaci parameter), tak sa často

strata rôznorodosti uvažuje ako funkcia nejakého parametra ktorý je relevantný pre každú metódu (a je ovplyvniteľný riadiacim parametrom každej metódy). Keďže takýmto parametrom je selekčná intenzita, tak tab. 6.3 obsahuje hodnoty straty rôznorodosti rôznych metód pri rovnakých hodnotách selekčnej intenzity. Keďže opäť sú ilustrované aj dynamické selekčné metódy, ich hodnoty sú iba indikatívne – sú platné pre použitú distribúciu vhodnosti (bola použitá horná distribúcia z obr. 6.5).

V tabuľke sú každej metóde venované dva riadky – v prvom riadku sú hodnoty pre prípad voľby všetkých rodičov jedným výberom, zatiaľ čo v druhom riadku je voľba realizovaná osobitným výberom pre každého rodiča. Rozdiel medzi týmito dvomi spôsobmi je markantný najmä v oblasti nižších selekčných intenzít. Preferencia stochastického univerzálneho vzorkovania pri vzorkovaní s explicitnými pravdepodobnosťami (či už bez alebo s premapovaním vhodnosti) bola uvedená skôr v súvislosti s problémami odchýlky a rozpätia. No tabuľka potvrdzuje aj zmyslupnosť realizovať napr. q-árny turnaj a orezanie nie ako turnaj či orezanie, ale iba použitím príslušnej pravdepodobnosti selekcie jedincov explicitným spôsobom.

Aj z hľadiska straty rôznorodosti medzi uvedenými selekčnými metódami nie je veľa rozdielov<sup>17</sup>. V podstate zaujímavými rozdielmi sa zdajú:

- Škálovacie premapovacie metódy (v kombinácii so stochastickým univerzálnym vzorkovaním) spôsobujú menšiu stratu rôznorodosti pri vyšších (z ich pohľadu) selekčných intenzitách.
- Orezanie pre dosiahnutie nejakej selekčnej intenzity spôsobuje vyššiu stratu rôznorodosti ako iné metódy pri produkcii rovnako veľkej selekčnej intenzity.

Problémom však môže byť to, že sa nedá jednoznačne určiť, či je lepšou väčšia alebo menšia strata rôznorodosti. Väčšia hodnota vedie k rýchlejšej konvergencii – čo pri niektorých úlohách urýchli nájdenie hľadaného riešenia, zatiaľ čo pri iných povedie k predčasnej konvergencii s nájdením iba lokálneho extrému funkcie vhodnosti.

Pri riešení praktických úloh môže nastať prípad, keď pri určovaní vhodnosti jedincov sa vyskytuje šum. Tento zabraňuje presnému určeniu vhodnosti, čím negatívne ovplyvňuje činnosť selekčných metód. Samotný šum môže byť spôsobený rôznymi faktormi, vrátane zašumených vstupných dát.

---

<sup>17</sup>Menšie rozdiely v hodnotách v rovnakom stĺpci sú spôsobené tým, že pri použitých hodnotách riadiacich parametrov metódy dosahovali približne rovnakú hodnotu selekčnej intenzity, nie však úplne rovnakú vo všetkých prípadoch.

Môže byť však aj výsledkom dobrovoľnej voľby uprednostniť rýchlo určiteľnú aproximáciu funkcie vhodnosti pred presnou funkciou vhodnosti vyžadujúcou značné časové zdroje pre určovanie vhodnosti jedincov. Evolučné algoritmy sú síce považované za relatívne imúnne voči výskytu šumu, avšak prítomnosť šumu spôsobuje spomalenie konvergenzie populácie. Preto napríklad, aj keď z teoretického hľadiska binárny turnaj a lineárne zotriedenie s proporcionálnou selekciou založenou na vhodnosti môžu byť ekvivalentné, pri realizácii turnaja ako turnaja s náhradou je konvergenzia turnaja pomalšia [16]. Dôvodom je to, že rozpätie takto realizovaného turnaja je maximálne, a táto chyba vzorkovania pôsobí ako zdroj šumu.

Výber metódy teda môže spočívať v požiadavke vyvinúť skôr nižší (vyšší) selekčný tlak resp. selekčnú intenzitu alebo v potrebe poskytovať široký rozsah možných selekčných tlakov. Rolu môže hrať aj otázka rýchlosti konvergenzie a s ňou súvisiaca redukcia rôznorodosti populácie (či už ako výsledok vytvárania selekčného tlaku alebo následkom prítomnosti genetického driftu). V niektorých prípadoch môže byť rozhodujúcou zložitosť metódy a jej implementácie a požadované časové nároky. A vplyv má samozrejme aj tvar a vlastnosti samotnej funkcie vhodnosti.



## Kapitola 7

# Náhrada

Cieľom tohto kroku je vyformovať nasledujúcu generáciu jedincov na základe tých jedincov, ktoré sú aktuálne k dispozícii. Vo všeobecnosti sú to tieto dve skupiny jedincov:

- jedince vytvárajúce pôvodnú populáciu v  $t$ -tej generácii  
 $P(t) = \{a_1(t), \dots, a_{\mu(t)}(t)\},$
- nové jedince (potomkovia) vytvorené v  $t$ -tej generácii aplikovaním genetických operátorov na selektovaných rodičov  
 $P''(t) = \{a_{\mu(t)+1}(t), \dots, a_{\mu(t)+\lambda(t)}(t)\}.$

Z týchto dvoch skupín je potrebné vhodným spôsobom vybrať skupinu jedincov  $P(t+1) = \{a_1(t+1), \dots, a_{\mu(t+1)}(t+1)\},$  ktoré budú reprezentovať novú populáciu v  $(t+1)$ -vej generácii. Na rozdiel od selekcie, teraz každý z jedincov môže byť vybraný maximálne raz (t.j. realizuje sa výber bez náhrady).

Aj keď počet jedincov v oboch skupinách sa môže meniť v rôznych generáciách, najčastejším prípadom je ako skupina potomkov rovnakej veľkosti v každej generácii ( $\lambda(t) = \lambda(t+1) = \lambda$ ), tak aj populácia stabilnej veľkosti ( $\mu(t) = \mu(t+1) = \mu$ ).

### 7.1 Spôsoby náhrady

Pri jednoduchšom spôsobe vytvorenia novej populácie sa použijú iba novo vytvorení potomkovia ( $P(t+1) \subseteq P''(t)$ ). Jedince vytvárajúce populáciu v  $t$ -tej generácii sa na tvorbe novej populácie nezúčastňujú. Tento model má teda jednogenečný charakter, jedince vo všeobecnosti prežívajú iba jednu

generáciu (výnimočne však môžu aj viac – jedinec má šancu na účasť pri tvorbe novej populácie, ak sa replikuje takým spôsobom, že nejaký vytvorený potomok je rovnaký ako daný jedinec). V populácii sa teda vyskytujú iba rovnako staré jedince.

Pre veľkosti zúčastnených skupín jedincov musí v tomto prípade očividne platiť  $\mu(t+1) \leq \lambda(t)$ . Rovnosť vedie na jednoduchú náhradu aktuálnej populácie potomkami, pričom všetci potomkovia prechádzajú do nasledujúcej generácie. Takáto náhrada sa označuje ako generačná. V prípade nerovnosti je k dispozícii viac vygenerovaných potomkov ako voľných miest a teda je nutné vybrať iba príslušný počet jedincov zo skupiny potomkov. Tento spôsob tvorby novej populácie býva označovaný ako  $(\mu, \lambda)$ , alebo ako čiarková stratégia (kde čiarka symbolizuje striktné oddelenie aktuálnej populácie a potomkov).

Pri zložitejšej realizácii náhrady sa nová populácia vytvára nielen z potomkov, ale aj z členov aktuálnej populácie ( $P(t+1) \subset P(t) \cup P''(t)$ ). V tomto prípade teda v populácii typicky koexistujú jedince rôzneho veku, pretože jedinec môže ale nemusí byť vyradený z populácie po uplynutí jednej alebo určitého počtu generácií (v extrémnom prípade nejaký jedinec, nachádzajúci sa v prvotnej generácii, môže v populácii prežívať až do konca behu evolučného algoritmu).

V tomto prípade počet miest v nasledujúcej generácii je menší ako počet jedincov, ktorí sú k dispozícii. Preto dochádza k súťaži medzi jedincami o to, kto prežije a kto zahynie. Organizácia tejto súťaže môže byť dvojaká:

- oddelená súťaž,
- spoločná súťaž.

Pri oddelenej súťaži je vopred presne dané, koľko miest v novej populácii bude obsadených jedincami pôvodnej populácie, a koľko zase potomkami. O tieto miesta súťažia jedince aktuálnej populácie medzi sebou a potomkovia tiež iba medzi sebou. Tieto počty sú dané generačnou medzerou  $G$ , kde

$$(1 - G)\mu(t + 1) \tag{7.1}$$

miest<sup>1</sup> je vyhradených jedincom aktuálnej populácie. Čím je teda generačná medzera väčšia, tým menej jedincov prejde z aktuálnej populácie do novej populácie. Extrémnymi hodnotami generačnej medzery sú hodnoty:

---

<sup>1</sup>Aj keď  $G$  je vo všeobecnosti reálne číslo z intervalu  $< 0, 1 >$ , počet voľných miest (či už pre staršie jedince alebo pre nováčikov) musí byť nezáporné celé číslo.



- $1 - G$  – žiadny jedinec aktuálnej populácie neprežije, nová populácia bude vytvorená iba z potomkov (náhrada sa teda redukuje na prípad generáčnej náhrady alebo čiarkovej stratégie),
- $1/\mu(t)$  – populácia sa zmení iba minimálne, iba jeden<sup>2</sup> jej jedinec bude nahradený novovygenerovaným potomkom. Tento extrémny spôsob náhrady býva označovaný ako “steady state” alebo ako inkrementálny.

Keďže počet miest vyhradených jedincom aktuálnej populácie býva menší, ako je počet týchto jedincov, a teda typicky platí  $(1 - G)\mu(t + 1) < \mu(t)$ , je potrebné vybrať tie, ktoré prežijú (resp. tie ktoré zahynú). Pre počet miest rezervovaných pre potomkov platí  $G\mu(t + 1) \leq \lambda(t)$ . Pri rovnosti sa jednoducho využijú všetci potomkovia, pri nerovnosti je potrebné vybrať spomedzi nich potrebný počet jedincov.

K súťaži teda môže dôjsť ako medzi jedincami aktuálnej populácie, tak aj medzi potomkami – nedochádza k súťaži medzi oboma skupinami navzájom. Môže dôjsť však k tomu, že výsledok výberu v jednej skupine ovplyvní výber v skupine inej – napríklad potomkovia, vkladani do populácie, nahradia tie jedince v populácii, ktorí sú k nim v nejakom vzťahu<sup>3</sup>.

Naproti tomu pri spoločnej súťaži jedince aktuálnej populácie a potomkovia spolu vytvárajú skupinu jedincov, z ktorej sa vyberajú jedince do novej populácie. Nie je stanovený počet, koľko pôvodných jedincov alebo potomkov má byť prenesených do novej populácie.

Dokonca nie je ani garantované, že v novej populácii bude aspoň jeden jedinec z aktuálnej populácie či aspoň jeden novo vygenerovaný potomok – teda nová populácia môže v extrémnom prípade reprezentovať totálnu výmenu jedincov v populácii alebo odmietnutie potomkov a zotrvanie v predchádzajúcom stave. Pri tejto stratégii vo všeobecnosti súťažia všetky jedince medzi sebou, resp. každý jedinec s každým<sup>4</sup>. Takýto spôsob tvorby novej populácie sa označuje ako  $(\mu + \lambda)$  alebo plus stratégia (kde plus symbolizuje spojenie aktuálnej populácie a potomkov do spoločného zdroja jedincov novej populácie).

<sup>2</sup>Niekedy sa v tejto úlohe uvažuje hodnota  $2/\mu$ , pretože pri použití sexuálneho genetického operátora výsledkom sú zvyčajne dvaja potomkovia.

<sup>3</sup>Môžu to byť ich rodičia, alebo jedince ktoré sú vkladným potomkom naj(ne)podobnejšie, ap.

<sup>4</sup>V niektorých špeciálnych schémach dochádza k obmedzeniu súťaže iba na súťaž v rámci určitej špecifickej skupiny jedincov, napríklad pri jednom spôsobe potlačania konvergencie populácie sa využíva rodinná súťaž – potomkovia súťažia so svojimi rodičmi a o zaradení do novej populácie rozhoduje ich vhodnosť.

## 7.2 Výber jedincov a selekčný tlak

Vo všetkých tých prípadoch, keď je potrebné vybrať jedince pre zaradenie do novej populácie (čiarková stratégia, oddelená súťaž potomkov, plus stratégia) alebo pre vyradenie z populácie (oddelená súťaž jedincov aktuálnej populácie), sa tento výber realizuje na základe rovnakého princípu ako selekcia – vhodnejšie jedince sú preferované oproti menej vhodným, a teda pravdepodobnosť prežitia vhodnejších jedincov nie je menšia ako pravdepodobnosť prežitia jedincov menej vhodných.

To znamená, že pre tento účel je možné využiť metódy používané pre selekciu rodičov. Aj keď v princípe je možné použiť ľubovoľnú selekčnú metódu, v praxi sa najviac používajú metódy vzorkovania s implicitnými pravdepodobnosťami – najmä orezanie či už v deterministickej podobe (deterministický výber určitého počtu najlepších alebo najhorších jedincov) alebo v pravdepodobnostnej podobe (napr. náhodný výber v rámci lepšej alebo horšej polovice skupiny jedincov).

Ak pri formovaní novej populácie dochádza k výberu jedincov, má to za následok vznik selekčného tlaku (s rovnakým následkom pre rýchlosť a spôsob konverencie algoritmu ako pri selekcii). Jeho veľkosť závisí nielen od použitej selekčnej metódy, ale aj od toho, aká časť novej populácie je vyberaná na základe preferovania vhodnejších jedincov.

Preto napríklad pri generačnej náhrade nie je uvažovaný žiadny selekčný tlak, zatiaľ čo pri čiarkovej stratégii áno – a čím je menší pomer  $\mu/\lambda$ , tým je veľkosť selekčného tlaku väčšia.

Pri porovnávaní čiarkovej stratégie s plus stratégiou, ak sa ako selekčná metóda použije deterministické<sup>5</sup> orezanie (pri rovnakých hodnotách  $\mu$  a  $\lambda$ ), väčší selekčný tlak vzniká pri plus stratégii<sup>6</sup>. Priemerná vhodnosť novej populácie bude pri použití plus stratégie rovnaká alebo vyššia ako pri použití čiarkovej stratégie, pričom rovnosť nastane iba v tom prípade, ak v rámci aktuálnej populácie nie je jedinec s vhodnosťou lepšou než vhodnosť najhoršieho potomka vybraného pomocou čiarkovej stratégie. Podobne najlepšia vhodnosť sa pri použití plus stratégie nezníži oproti možnosti jej zníženia pri použití čiarkovej stratégie.

Podobne napríklad pri náhrade s generačnou medzerou nie je jedno, či sa

---

<sup>5</sup>Pri stochastických metódach sa výsledok porovnania líši podľa toho, či potomkovia majú vyššie vhodnosti ako ich rodičia alebo nižšie.

<sup>6</sup>Pri určovaní doby prevzatia (s uvažovaním iba selekcie a náhrady, bez pôsobenia genetických operátorov) sa v prípade plus stratégie v novej populácii vyskytne viac kópií najlepšieho jedinca ako v prípade čiarkovej stratégie – a teda najlepší jedinec skôr zaplní celú populáciu.

vygeneruje iba toľko potomkov, koľko je pre potomkov rezervovaných miest v novej populácii, alebo viac s následným výberom. A samozrejme, čím bude skupina potomkov väčšia (pri rovnakom počte rezervovaných miest), tým menšia časť potomkov bude vybratá a tým silnejší bude selekčný tlak.

Prax ukazuje, že pri použití inkrementálnej náhrady (teda náhrady s minimálnou generačnou medzerou) je selekčný tlak vyšší ako pri použití generačnej náhrady (maximálnej generačnej medzery). Je to spôsobené jednak tým, že z aktuálnej populácie sú vyradované najhoršie jedince, a jednak tým že pri inkrementálnom spôsobe potomok je okamžite zaradzovaný do populácie a jeho vplyv môže byť okamžitý, nemusí dlho čakať kým sa prejaví<sup>7</sup>.

Podobne ako pri selekcii, aj teraz pri porovnávaní jednotlivých metód je možné vziať do úvahy aj iné faktory, nielen selekčný tlak. Z pohľadu prítomnosti šumu pri určovaní vhodnosti je potrebné spomenúť veľkú citlivosť inkrementálnej náhrady na prítomnosť šumu [16]. Keďže pri vyradovaní jedincov z aktuálnej populácie sa vplyvom šumu môžu vyradovať lepšie jedince zatiaľ čo horšie v populácii prežívajú (pretože vďaka šumu horšie jedince môžu mať vhodnosť vyššiu ako jedince lepšie), následkom je rapídne spomalenie konvergencie algoritmu. V tomto prípade sa ukazuje ako vhodnejšie pri inkrementálnej náhrade vyradovanie jedincov z populácie nie na základe nepresne určovanej vhodnosti ale na základe ich veku – potomok nahradí vždy najstaršieho jedinca z aktuálnej populácie. Vychádza sa pritom z predpokladu, že keďže vo všeobecnosti sa vhodnosti jedincov v populácii zvyšujú, tak najstaršie jedince vlastne viac-menej reprezentujú jedince s najhoršou vhodnosťou.

Dôležitou vlastnosťou niektorých spôsobov náhrady aktuálnej populácie populáciou novou je elitizmus – elitistická náhrada zabezpečuje, že najlepšia vhodnosť v novej populácii nebude horšia ako najlepšia vhodnosť v aktuálnej populácii. V tomto prípade je garantovaná monotónna evolúcia najlepšieho jedinca v populácii. Najlepší jedinec môže byť nahradený iba lepším potomkom – bude prežívať dovtedy, pokiaľ takýto lepší potomok nebude nájdený. Príkladom elitistickej schémy je plus stratégia alebo prípad generačnej medzery  $G < 1$  ak je zaručené, že najlepší jedinec z aktuálnej populácie nebude vyradený.

Naproti tomu náhrada bez elitizmu pripúšťa aj zhoršenie vhodnosti najlepšieho jedinca v populácii. Takými schémami sú napríklad generačná náhrada alebo čiarková stratégia. Schopnosť zabudnúť aktuálne najlepšie riešenie v princípe poskytuje nástroj pre opustenie oblasti lokálneho extrému a

<sup>7</sup>Pre porovnanie jednotlivých metód sa ako čas nepoužíva počet generácií ale skôr počet vyhodnotení (teda počet generovaných potomkov).

teda pre zabránenie uviaznutia v ňom.

Zdrojom selekčného tlaku teda môže byť nielen selekcia, ale aj vytváranie novej populácie. Aj keď je možné vytvárať selekčný tlak na oboch miestach súčasne (pričom výsledný selekčný tlak vzniká zložením oboch parciálnych), pomerne častými sú schémy, ktoré vytvárajú selekčný tlak iba na jednom mieste (zatiaľ čo druhé je riešené jednoduchým spôsobom nevytvárajúcim selekčný tlak):

- v selekcii (s použitím generačnej náhrady),
- v náhrade (buď každý jedinec sa povinne stáva rodičom alebo každý má rovnakú pravdepodobnosť sa stať rodičom nezávisle na svojej vhodnosti).

V týchto prípadoch je potom možnosť výslednú veľkosť selekčného tlaku regulovať iba pomocou jedného parametra, pričom vhodná voľba príslušnej metódy poskytuje dostatočne veľký rozsah dosiahnuteľných hodnôt.

## Kapitola 8

# Genetické operátory

Cieľom genetických operátorov je poskytnúť nástroj, umožňujúci zužitkovať štrukturálne elementy, z ktorých sú vytvorení rodičia, pre vytvorenie nových potomkov (v priestore prehľadávania). Keďže cieľom tvorby nových potomkov je získanie nových vzoriek plochy vhodnosti, aplikácia genetických operátorov na štruktúru rodičov musí poskytovať potomkov, ktorých štruktúra je odlišná od štruktúry ich rodičov (či už sa jedná o odlišnosť menšiu alebo väčšiu).

Najčastejšie používané delenie operátorov je založené na počte rodičov, potrebných pre vygenerovanie nových jedincov. Z tohto hľadiska sa rozlišujú tri triedy operátorov [4]:

- asexuálne operátory – potomok má iba jedného rodiča,
- sexuálne operátory – potomok má dvoch rodičov,
- panmiktické operátory – potomok má viac ako dvoch rodičov.

Keďže asexuálny operátor používa k vytvoreniu nového jedinca stavebný materiál iba jedného rodiča, musí zmeniť aspoň časť štrukturálnych elementov, z ktorých je vytvorený daný rodič – jeden alebo viacero stavebných elementov je nahradených inými elementami. To znamená, že štruktúra novo vytvoreného jedinca bude obsahovať aspoň jeden taký element, ktorý sa nenachádzal v štruktúre rodiča.

Tento princíp je použiteľný aj pre sexuálny a panmiktický operátor – potomok môže obsahovať stavebný element, ktorý nie je súčasťou ani jedného z rodičov. Avšak prítomnosť viacerých rodičov (z ktorých každý je nositeľom nejakej sady stavebných elementov) umožňuje budovať nových potomkov aj na základe iného princípu – kombinovaním štrukturálnych elementov rodičov

do nových kombinácií<sup>1</sup>. Výsledkom je, že štruktúra novo vytvoreného jedinca obsahuje iba také elementy, z ktorých každý sa nachádza v štruktúre aspoň jedného z rodičov.

V analógii k biologickým procesom, asexuálny operátor sa zvykne označovať ako *mutačný operátor* resp. mutácia a sexuálny operátor zase ako *krížiaci operátor* alebo kríženie. Tieto dva typy operátorov sú najčastejšie používanými genetickými operátormi (v rôznom pomere v rôznych typoch evolučných algoritmov), zatiaľ čo používanie tretieho typu je pomerne zriedkavé.

Aby novo generované jedince mali možnosť kombinovať stavebné elementy rodičov (s nádejou na lepšiu kombináciu ako je v rodičovských štruktúrach) ako aj možnosť zapojiť v rodičoch nepoužité stavebné elementy (s nádejou nájsť vhodnejšiu alternatívu), je bežným zvykom kombinovať rôzne operátory navzájom.

Najčastejšou kombináciou je súčasné použitie mutačného aj krížiaceho operátora, pričom krížiaci operátor realizuje iba premiešanie<sup>2</sup> stavebných elementov rodičov. Oba operátory sa podieľajú na produkcii potomkov, pričom je použité jedno z usporiadaní:

- každý rodič je najprv spracovaný mutačným operátorom a následne dvojice mutovaných rodičov sú spracovávané krížiacim operátorom,
- dvojice rodičov sú spracované krížiacim operátorom a následne každý výsledok tohto kríženia je spracovaný mutačným operátorom.

Iným spôsobom použitia viacerých operátorov je ich oddelené používanie – každý z použitých operátorov slúži pre vytvorenie časti populácie nových potomkov osobitne, pričom je pre rôzne operátory možné použiť rôznych rodičov alebo tých istých rodičovských jedincov.

Ak však všetky stavebné elementy, potrebné pre zostavenie hľadaného riešenia, sú prítomné v rodičovských štruktúrach, nie je potrebné používať mutačný operátor. Podobne nie je potrebné používať krížiaci operátor v prípade, ak je možné vytvoriť hľadané riešenie iba zmenou rodičovských štruktúr bez ich vzájomného kombinovania.

Genetické operátory sú na rodičov aplikované nie deterministicky, ale iba s nejakou pravdepodobnosťou. Mutačný operátor je tak aplikovaný s pravdepodobnosťou  $p_m$  a krížiaci operátor s pravdepodobnosťou  $p_c$ . Zatiaľ

---

<sup>1</sup>Sexuálny a panmiktický operátor sa niekedy spoločne označujú ako rekombinačný operátor.

<sup>2</sup>Ak krížiaci operátor umožní aj vkladanie elementov, ktoré sa nevyskytujú v rodičovskej dvojici na vstupe operátora, potreba mutačného operátora silne klesá.

čo  $p_c$  určuje, či sa dvaja rodičia budú krížiť alebo nie,  $p_m$  sa v závislosti od konkrétneho operátora môže chápať dvojako. Pri operátoroch, kde jednotlivé štrukturálne elementy môžu byť mutované nezávisle od seba, sa chápe ako pravdepodobnosť mutácie jedného štrukturálneho elementu rodiča. V prípade, že mutácia vyžaduje súčasnú zmenu viacerých elementov štruktúry rodiča, rozhoduje o tom, či daný rodič bude mutovaný alebo nie. Vhodnou voľbou týchto pravdepodobností je možné určiť, že použité operátory sú navzájom rovnocenné, alebo že niektorý z nich je majoritným zatiaľ čo iný je minoritným operátorom.

Ak operátor nie je aplikovaný, tak rodičia sa stávajú svojimi potomkami bez toho, aby ich vnútorná štruktúra bola zmenená. Toto poskytuje šancu realizovať iba časť zmien (niektorý operátor je aplikovaný zatiaľ čo iný nie) alebo dokonca zachovať aktuálny stav (žiadny operátor nie je aplikovaný).

Počet nových jedincov, ktorých je schopný vyprodukovať nejaký operátor, môže byť rôzny. Najčastejším prípadom je produkcia rovnakého počtu jedincov ako je počet rodičov, avšak môže to byť aj menší (typicky jeden potomok) alebo väčší počet. Keďže toto závisí od konkrétneho použitého operátora, je potrebné prispôbiť výberu operátorov aj počet rodičov, vybraných pomocou selekcie.

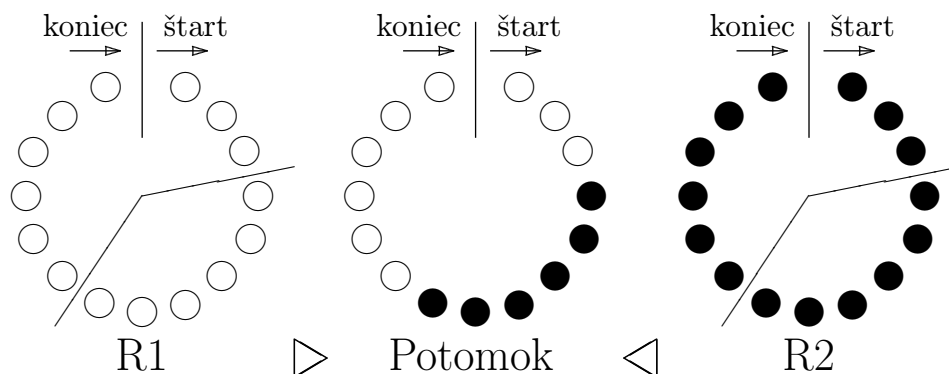
Genetické operátory pre zmenu stavebných elementov rodičov alebo pre ich miešanie potrebujú znalosti o použitom spôsobe reprezentácie. Sú teda závislé na použitej reprezentácii a pre rôzne spôsoby reprezentácie je potrebné použiť rôzne operátory. V ďalšom uvádzame niektoré operátory, určené pre lineárne kódovacie schémy pevnej dĺžky a pevného usporiadania.

## 8.1 Operátory pre binárne kódovanie

**Doplnková mutácia.** Jednotlivé pozície rodičovského kódu sú menené nezávisle na sebe. Mutácia hodnoty na nejakej pozícii znamená náhradu tejto hodnoty druhou hodnotou. Keďže na danej pozícii sa môže vyskytovať iba jedna z dvoch možných hodnôt, určenie náhradnej hodnoty je jednoznačné.

**Mnohobodové kríženie.** Štruktúry rodičov, chápané ako postupnosti stavebných elementov, sa rozdelia pomocou deliacich bodov na niekoľko za sebou nasledujúcich disjunktných segmentov. Keďže na oboch rodičov sa aplikujú tie isté deliace body, obaja rodičia budú rozdelení na rovnaké segmenty. Potomok je potom konštruovaný tak, že striedavo preberá segmenty elementov z jedného a druhého rodiča. Takýmto spôsobom je možné vyprodukovať dvoch potomkov, ktoré sú navzájom komplementárne. Jeden z nich

preberá z prvého rodiča nepárne segmenty a z druhého párne segmenty, zatiaľ čo druhý potomok preberá nepárne segmenty z druhého rodiča a párne segmenty z prvého rodiča. Obr. 8.1 ilustruje kríženie pri použití dvoch deliacich bodov.



Obr. 8.1: Ilustrácia dvojbodového kríženia

Deliaci bod sa generuje vždy medzi dve susedné pozície, pričom nie je preferovaná žiadna z možných polôh deliaceho bodu. Ak pre reprezentáciu jedincov je použitých  $l$  pozícií, tak je k dispozícii  $l - 1$  možných umiestnení deliacich bodov. V prípade, že jedinec je chápaný ako cyklická štruktúra (tak ako to je na obrázku), tak potom je k dispozícii  $l$  možných umiestnení – jedna možná pozícia pribudla medzi začiatkom a koncom jedinca.

Ak deliace body sú generované tak, že musia byť umiestnené na rôznych pozíciách, a súčasne žiadny deliaci bod nesmie byť umiestnený medzi začiatkom a koncom jedinca (jedinec je uvažovaný ako lineárna a nie cyklická štruktúra), potom počet segmentov, na ktoré sú jedince rozdelené, je presne daný ako počet deliacich bodov zvýšený o jednotku. V rámci tohto variantu je najčastejšie používaný jeden deliaci bod (jednobodové kríženie s výmenou zadných častí rodičov) alebo dva deliace body (dvojbodové kríženie s výmenou stredných častí rodičov).

V prípade, že deliace body môžu byť umiestňované aj medzi začiatok a koniec jedinca alebo na miesto iného deliaceho bodu, tak počet segmentov nie je presne daný ale závisí na náhodnom umiestňovaní deliacich bodov – vzniká efekt premenlivého počtu segmentov, na ktoré sú rodičia rozdelení. Tento variant sa používa v súvislosti s vyšším počtom deliacich bodov.



**Miešajúce kríženie.** Je to rozšírenie mnohobodového kríženia. Pred vykonaním samotného kríženia sú pozície náhodne popreskupované (u oboch rodičov rovnakým spôsobom). Po vykonaní kríženia sa zase pozície potomkov preskupia do pôvodného poradia [7].

Vďaka prvotnému premiešaniu je možné, že pozície, ktoré sú od seba aj značne vzdialené, sa stanú susednými a majú väčšiu šancu byť spolu prenesené do toho istého potomka. Tento postup sa snaží eliminovať výhody a nevýhody toho, že niektoré pozície sú blízko seba a niektoré sú od seba vzdialenejšie, a zrovnoprávniť ich šance na spoločný prenos.

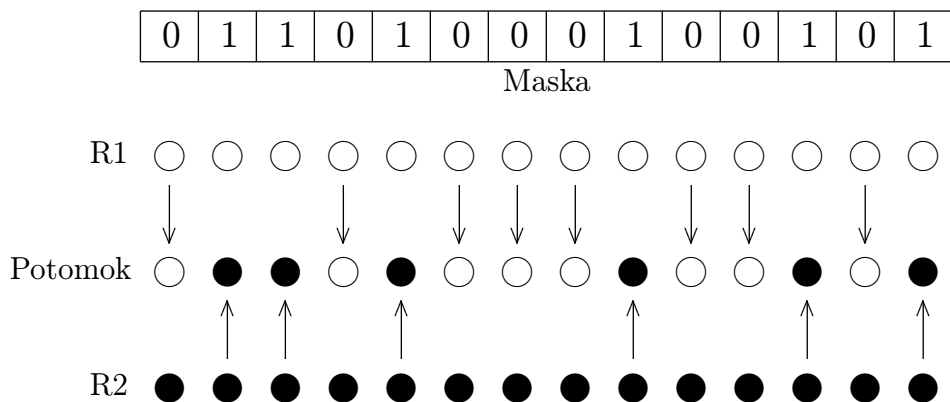
Miesto náhodného preskupovania pozícií je možné pred samotným krížením aplikovať operátor preusporiadania a po jeho vykonaní zase inverzný operátor. V minulosti bol populárnym preusporiadavacím operátorom operátor inverzie – pomocou dvoch náhodných pozícií sa určil nejaký segment v kóde rodiča a v rámci tohto segmentu sa poradie jednotlivých pozícií invertovalo.

**Segmentové kríženie.** Podobne ako pri mnohobodovom krížení, aj teraz je lineárna štruktúra rodiča rozdelená na segmenty pomocou deliacich bodov (oba ja rodičia sú delení rovnakým spôsobom).

Rozdiel je v tom, že nie je presne zadaný počet deliacich bodov a teda ani segmentov. Namiesto toho je zadaná pravdepodobnosť zmeny segmentu. Pri definovaní segmentov sa postupne skúmajú jednotlivé pozície a pre každú z nich sa určuje (stochasticky s využitím zadanej pravdepodobnosti), či patrí k tomu istému segmentu ako predchádzajúca pozícia alebo k novému segmentu.

**Uniformné kríženie.** Pre riadenie prenosu hodnôt z rodičov do potomkov slúži maska [42]. Je to vektor rovnakej dĺžky ako je počet stavebných elementov, z ktorých je vytvorený každý rodič. Každá pozícia v tejto maske je binárna – jej hodnota signalizuje jedného z rodičov. Obsah masky je generovaný náhodne a umožňuje z dvoch rodičov vyprodukovať dvoch potomkov. Prvý potomok preberá z prvého rodiča hodnoty na tých pozíciách, pre ktoré je v maske hodnota 0, a z druhého rodiča hodnoty na tých pozíciách, pre ktoré je v maske hodnota 1. Druhý potomok je vytváraný komplementárne – hodnota 0 v maske signalizuje použitie druhého rodiča a hodnota 1 zase použitie prvého rodiča. Proces tvorby potomka je ilustrovaný na obr. 8.2.

Operátor sa chová ako mnohobodové kríženie s premenlivým počtom deliacich bodov. Ak hodnoty 0 a 1 sú v maske generované s rovnakou pravdepodobnosťou, tak priemerný počet deliacich bodov je  $l/2$ . Ak hodnoty v



Obr. 8.2: Ilustrácia uniformného kríženia

maske sú generované s rôznymi pravdepodobnosťami, tak priemerný počet deliacich bodov sa znižuje. Je tým menší, čím je väčší rozdiel medzi pravdepodobnosťami generovania hodnôt 1 a 0.

**Skenovací operátor.** Na vstupe očakáva niekoľko rodičov. Potomka vytvára postupne po jednotlivých pozíciách. Pre určenie, akú hodnotu vložiť do potomka na nejakú pozíciu, najprv preskenuje všetkých rodičov a zistí, aké hodnoty sa vyskytujú v rodičoch na danej pozícii. Potom vyberie hodnotu pre potomka pre danú pozíciu pomocou jednej z nasledujúcich možností:

- deterministického výberu,
- náhodného výberu.

Pri deterministickom výbere sa vyberie tá hodnota, ktorá sa najčastejšie vyskytuje v štruktúre rodičov. Rodičia svojimi hodnotami vlastne robia hlasovanie. Pre zabezpečenie jednoznačného rozhodnutia je potrebné, aby počet rodičov bol nepárny. Nevýhodou tohto variantu je, že umožňuje z jednej skupiny rodičov vygenerovať iba jedného potomka.

Pri náhodnom výbere sa na základe frekvencie výskytu rôznych hodnôt na danej pozícii v rámci skupiny rodičov určí pravdepodobnosť pre každú z hodnôt. Výsledná hodnota pre potomka sa určuje stochasticky s využitím týchto pravdepodobností. Tento variant nevyžaduje nepárny počet rodičov a ani nie je obmedzený na produkciu iba jedného potomka. Pri znížení počtu

rodičov na dvoch, tento operátor degraduje na uniformné kríženie s rovnakými pravdepodobnosťami generovania hodnôt v maske.

**Diagonálny operátor.** Aj tento operátor na vstupe očakáva viacerých rodičov. Rodičia sú pomocou náhodne generovaných deliacich bodov rozdelení na segmenty, pričom každý rodič je rozdelený rovnakým spôsobom. Počet deliacich bodov je taký, aby v rámci každého jedinca vzniklo toľko segmentov, koľko je rodičov.

Jednotlivé segmenty vytvárajú maticu – segmenty  $i$ -teho rodiča reprezentujú  $i$ -ty riadok tejto matice, zatiaľ čo  $i$ -ty stĺpec matice reprezentuje  $i$ -te segmenty všetkých rodičov.

Potomkovia sú vytváraní postupným skladaním segmentov, ktoré ležia na diagonálach tejto matice. Napríklad jeden potomok bude vytvorený spojením segmentov ležiacich na hlavnej diagonále. Druhý potomok použije prvý segment druhého rodiča, druhý segment tretieho rodiča, ..., predposledný segment posledného rodiča a posledný segment prvého rodiča. Analogicky je možné vytvoriť ostatných potomkov. Takýmto spôsobom je možné vytvoriť toľko potomkov, koľko je rodičov.

Pri znížení počtu rodičov na dvoch tento operátor degraduje na mnohobodové kríženie s jedným deliacim bodom.

## 8.2 Operátory pre mnohoznačné kódovanie

Pre tento druh kódovania nie je potrebné vytvárať špeciálne operátory, pretože je možné využiť všetky operátory<sup>3</sup>, používané pre binárnu kódovaciu schému.

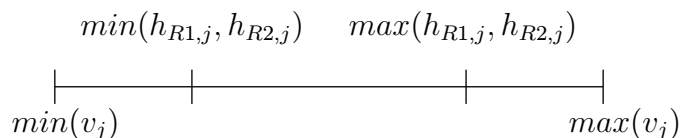
Jediný rozdiel je ten, že pri mutačnom operátore hodnota, aktuálne prítomná v rodičovskej štruktúre, sa môže zmeniť na jednu z niekoľkých hodnôt ktoré prichádzajú do úvahy. Každú z nich možno uvažovať s rovnakou pravdepodobnosťou (a teda výber náhradnej hodnoty sa deje náhodne) alebo je možné zaviesť preferenciu hodnôt prostredníctvom distribúcie pravdepodobnosti nad týmito hodnotami.

## 8.3 Operátory pre reálne kódovanie

Z hľadiska kombinovania hodnôt rodičov je reálne kódovanie nadmnožinou binárneho kódovania. Situácia pri použití dvoch rodičov je ilustrovaná na

<sup>3</sup>Pri použití skenovacieho operátora s deterministickým výberom nepárny počet rodičov nezaručuje jednoznačné rozhodnutie v prospech jednej z prípustných hodnôt.

obr. 8.3.



Obr. 8.3: Alternatívy pre voľbu reálnej hodnoty

Na nejakej  $j$ -tej pozícii každého jedinca je prípustná iba hodnota z intervalu  $< min(v_j), max(v_j) >$  a prvý rodič má na tejto pozícii hodnotu  $h_{R1,j}$ , zatiaľ čo hodnota  $h_{R2,j}$  sa nachádza na tejto pozícii v štruktúre druhého rodiča.

Pre reálnu kódovaciu schému je možné využiť krížiacie operátory pre binárne kódovanie – teda operátory, výsledkom činnosti ktorých bude výber jednej z hodnôt  $h_{R1,j}$  alebo  $h_{R2,j}$ <sup>4</sup>. Navyše k tomuto výberu jednej z dvoch hodnôt však výsledkom môže byť aj nejaká kombinácia oboch hodnôt – hodnota z povoleného intervalu ktorá sa nenachádza ani v jednom z rodičov. Teda reálna reprezentácia umožňuje definovať k operátorom, presúvajúcim hodnotu z jedného z rodičov do potomka, aj operátory, kombinujúce hodnoty rodičov do novej hodnoty pre potomka.

**Uniformná mutácia.** Je rozšírením doplnkovej mutácie pre reálnu kódovaciu schému. Mutácia hodnoty na nejakej pozícii znamená náhradu tejto hodnoty inou hodnotou. Keďže na danej pozícii sa môže vyskytovať ľubovoľná hodnota z určitého intervalu hodnôt, určenie náhradnej hodnoty je realizované pomocou uniformnej distribúcie pravdepodobnosti nad týmto intervalom (poskytujúcej rovnakú šancu pre každú z prípustných hodnôt).

**Extremálna mutácia.** Táto mutácia znamená náhradu hodnoty na nejakej pozícii inou hodnotou, ktorou môže byť teraz iba jedna z hraničných hodnôt definujúcich interval povolených hodnôt pre danú pozíciu.

**Neuniformná mutácia.** Vzniklo niekoľko modifikácií uniformnej mutácie, zameraných na odstránenie uniformnosti distribúcie pravdepodobnosti

---

<sup>4</sup>Namiesto priameho prenosu hodnoty  $h_{i,j}$  z rodiča je možné do potomka umiestniť hodnotu z okolia tejto hodnoty, z intervalu  $< h_{i,j} - k, h_{i,j} + k >$ , pričom sa použije trojuholníková distribúcia pravdepodobnosti s maximom v hodnote  $h_{i,j}$  a minimom v hraničných bodoch daného intervalu [43].

nad možnými hodnotami, ktoré môžu nahradiť aktuálne sa vyskytujúcu hodnotu v štruktúre rodiča.

Jedna trieda takýchto vylepšení spočíva na myšlienke, že hodnoty blízko aktuálnej hodnoty by mali byť preferovanejšie voči hodnotám, ktoré sú od nej vzdialenejšie. Teda menšie zmeny hodnoty by sa mali diať s väčšou pravdepodobnosťou ako zmeny väčšie (ako dôsledok tohto princípu, najväčšiu pravdepodobnosť by mala mať žiadna zmena, teda zachovanie aktuálneho stavu). Jedným zosobnením tejto myšlienky je normálna mutácia, ktorá novú hodnotu určuje podľa

$$h'_j = h_j + \alpha N_j(0, 1) \quad (8.1)$$

kde  $N(0, 1)$  reprezentuje hodnotu generovanú podľa normovaného normálneho rozdelenia pravdepodobnosti (s nulovou strednou hodnotou a jednotkovou smerodajnou odchýlkou) a  $\alpha$  je parameter, upravujúci veľkosť mutačnej zmeny. Iným variantom je Cauchyho mutácia, ktorá namiesto  $N(0, 1)$  používa Cauchyho náhodnú premennú so škálovacím parametrom nastaveným na hodnotu 1. Tvar hustoty pravdepodobnosti je podobný ako pri normálnom rozdelení<sup>5</sup>, rozdiel je iba v tom, že funkcia sa približuje k vodorovnej osi pomalšie ako pri normálnom rozdelení – väčšie zmeny sú pri Cauchyho mutácii o niečo pravdepodobnejšie ako pri normálnej mutácii, menšie zmeny zase o niečo menej pravdepodobné.

Iným zosobnením uvedenej myšlienky je Mühlenbeinova mutácia [32]. V tomto prípade nová hodnota na  $j$ -tej pozícii sa určuje podľa vzťahu

$$h'_j = \begin{cases} h_j + \Delta(y), & \chi = 0 \\ h_j - \Delta(y), & \chi = 1 \end{cases} \quad (8.2)$$

kde  $\chi$  je náhodné číslo z množiny  $\{0, 1\}$  a  $y$  je mutačný rozsah (určený napríklad ako nejaké percento z  $\max(v_j) - \min(v_j)$ ). Funkcia  $\Delta$  je funkciou možnej zmeny. Je definovaná ako

$$\Delta(y) = y \sum_{i=0}^{15} 2^{-i} \chi \quad (8.3)$$

<sup>5</sup>Jednorozmerná Cauchyho hustota pravdepodobnosti je definovaná pomocou

$$\frac{1}{\pi} \frac{t}{t^2 + \Delta^2} \quad -\infty < \Delta < \infty$$

kde  $t > 0$  je škálovací parameter a  $\Delta$  zase označuje zmenu, ktorej pravdepodobnosť je určená danou funkciou.

kde  $\chi$  je opäť náhodné číslo z množiny  $\{0, 1\}$  – avšak na rozdiel od predchádzajúceho vzťahu, v (8.3) je hodnota 0 generovaná s inou (väčšou) pravdepodobnosťou<sup>6</sup> ako hodnota 1. Následkom toho menšie hodnoty sú generované častejšie ako hodnoty väčšie.

Iná myšlienka úpravy uniformnej mutácie bola vedená snahou pridať uniformnej mutácii schopnosť adaptácie na aktuálnu fázu evolučného hľadania. Vychádzala z predstavy, že v ranných štádiách hľadania by si mal algoritmus urobiť predstavu o celej ploche vhodnosti a teda mutácia by mala byť vykonávaná v celom intervale povolených hodnôt. Naproti tomu v neskorších fázach hľadania by algoritmus mal podrobnejšie preskúmať vytipovaný podpriestor a teda mutácia by sa mala odohrávať iba v zmenšenom rozsahu.

Na základe tejto predstavy bola navrhnutá adaptívna mutácia podľa vzťahu, ktorý umožňuje určiť novú hodnotu na  $j$ -tej pozícii nasledovne [30]:

$$h'_j = \begin{cases} h_j + \Delta(t, \max(v_j) - h_j), & \chi = 0 \\ h_j - \Delta(t, h_j - \min(v_j)), & \chi = 1 \end{cases} \quad (8.4)$$

kde  $\chi$  je opäť náhodné číslo z množiny  $\{0, 1\}$ . Funkcia  $\Delta$  je teraz funkciou času a maximálnej možnej zmeny. Je definovaná ako

$$\Delta(t, y) = y \left( 1 - \chi \left( 1 - \frac{t}{t_{max}} \right)^\alpha \right) \quad (8.5)$$

kde  $\chi$  je opäť náhodné číslo, tentokrát z intervalu  $< 0, 1 >$ ,  $t_{max}$  je maximálny počet generácií a  $\alpha$  je parameter, udávajúci stupeň závislosti na čísle generácie. Pripočítavať resp. odpočítavať sa bude hodnota z intervalu  $< 0, y >$ , pričom so stúpajúcim číslom generácie čoraz viac narastá pravdepodobnosť výberu z minimálneho konca tohto intervalu.

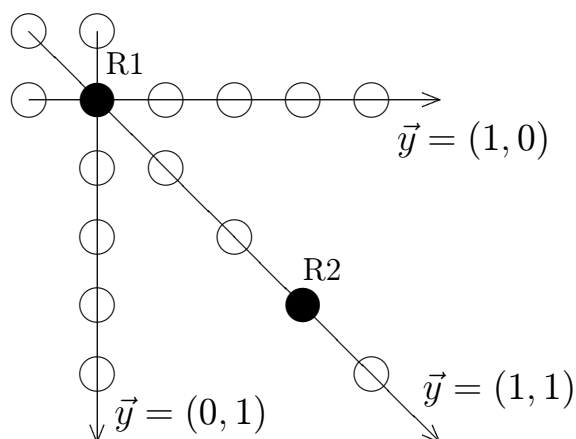
**Aritmetické kríženie.** Obidvaja rodičia sú chápaní ako body v mnohorozmernom priestore. Potomok je určovaný ako nový bod v tomto priestore, pričom pre jeho určenie sa používajú aritmetické operácie s hodnotami rodičov. Obr. 8.4 ilustruje tento typ kríženia, pričom prázdne krúžky znázorňujú možné polohy potomkov.

Najjednoduchším typom aritmetického kríženia je lineárna kombinácia hodnôt rodičov. Potomok sa určuje podľa vzťahu

$$h'_j = \chi h_{R1,j} + (1 - \chi) h_{R2,j} \quad (8.6)$$

---

<sup>6</sup>Možno použiť napr. pravdepodobnosť 1/16 pre generovanie hodnoty 1 a 15/16 pre generovanie hodnoty 0. V tomto prípade v priemere bude pri výpočte  $\Delta(y)$  generovaná iba jedna jednotka a teda bude použitá jedna zo zmien  $\{2^0, \dots, 2^{-15}\}$ .



Obr. 8.4: Ilustrácia aritmetického kríženia

kde  $\chi$  je náhodné číslo z intervalu  $\langle 0, 1 \rangle$ . Tento typ kríženia dokáže produkovať potomkov na úsečke, ohraničenej bodmi reprezentujúcimi oboch rodičov.

Potomkov s väčším rozsahom hodnôt dokáže produkovať lineárne kríženie, produkujúce ľubovoľný nepárny počet potomkov na základe vzťahov

$$\begin{aligned}
 & 0.5h_{R1,j} + 0.5h_{R2,j} \\
 & 1.5h_{R1,j} - 0.5h_{R2,j} \\
 & -0.5h_{R1,j} + 1.5h_{R2,j} \\
 & 2.5h_{R1,j} - 1.5h_{R2,j} \\
 & \dots
 \end{aligned} \tag{8.7}$$

Nepárny počet je možný upraviť na párný jednoduchým deterministickým výberom potrebného počtu generovaných jedincov. Tento typ umiestňuje potomkov na priamke prechádzajúcej oboma bodmi reprezentujúcimi rodičov, pričom nie je obmedzený iba na oblasť medzi oboma rodičmi. Čím sa používajú vyššie koeficienty vo vzťahu, tým ďalej od oboch rodičov je potomok umiestňovaný.

Podobné chovanie je možné dosiahnuť diferencovaním rodičov – jeden z nich bude štartovací (v obr. 8.4 to je rodič R1) a druhý vodiaci, udávajúci smer pohybu od štartovacieho rodiča. Hodnoty potomka je možné určovať podľa

$$h_j = h_{R1,j} + (h_{R2,j} - h_{R1,j})k \tag{8.8}$$

kde hodnota  $k$  označuje veľkosť kroku. Krok veľkosti 0 alebo 1 umožňuje reprodukovať jedného z rodičov, jeho hodnota z otvoreného intervalu  $(0, 1)$  umiestňuje potomka na úsečku medzi oboch rodičov, a hodnota menšia ako 0 alebo väčšia ako 1 lokalizuje potomka na priamke prechádzajúcej obomi rodičmi mimo úseku medzi nimi.

Vzťah pre určovanie potomkov je možné doplniť o perturbačný vektor  $\vec{y}$  na tvar

$$h_j = h_{R1,j} + y_j(h_{R2,j} - h_{R1,j})k \quad (8.9)$$

kde  $y_j$  reprezentuje  $j$ -tu zložku tohto vektora. Každá zložka tohto vektora môže mať hodnotu 0 alebo 1, pričom 1 povoľuje pohyb pozdĺž  $j$ -tej súradnej osi, zatiaľ čo hodnota 0 tento pohyb zakazuje. Vektor tvaru  $(1, 1, \dots, 1)$  umožňuje pohyb v smere medzi obomi rodičmi,  $(0, 0, \dots, 0)$  stotožňuje potomka so štartovacím rodičom, a všetky ostatné podoby vektora umiestňujú potomkov na priamke prechádzajúcej štartovacím rodičom ale nie vodiacim rodičom a kolmej na tú súradnú os, ktorej zodpovedá hodnota 0 v perturbačnom vektore. Perturbačný vektor je generovaný náhodným spôsobom.

Podobný vzťah ako (8.8) používa heuristické kríženie. To produkuje potomka podľa vzťahu

$$h_j = h_{R1,j} + (h_{R1,j} - h_{R2,j})\chi \quad (8.10)$$

kde  $\chi$  reprezentuje náhodné číslo z intervalu  $< 0, 1 >$ . Navyše sa predpokladá, že z hľadiska vhodnosti je rodič  $R1$  vhodnejším riešením ako rodič  $R2$ . Takto definovaný operátor umiestňuje potomka na priamku prechádzajúcu oboma rodičmi do časti začínajúcej lepším rodičom a smerujúcej od oboch rodičov preč.

**FCB kríženie.** Pri produkcii hodnoty pre potomka sa vychádza z rozdelenia intervalu možných hodnôt, ktoré sa môžu vyskytnúť na  $j$ -tej pozícii, na tri disjunktné podintervaly podľa obr. 8.3. Pre každý tento podinterval sa produkuje jedna hodnota – takýmto spôsobom je možné vygenerovať tri alternatívy pre každú pozíciu novo generovaného potomka.

Samotné hodnoty je možné určovať z hodnôt vyskytujúcich sa v rodičoch pomocou vzťahu

$$h_j = \min(v_j) + [\max(v_j) - \min(v_j)] \Delta(h'_{R1,j}, h'_{R2,j}) \quad (8.11)$$

kde  $h'_{R1,j}$  a  $h'_{R2,j}$  reprezentujú hodnoty rodičov v normovanej podobe, pričom normovanie sa robí podľa



$$h'_{i,j} = \frac{h_{i,j} - \min(v_j)}{\max(v_j) - \min(v_j)} \quad (8.12)$$

a zabezpečuje, že tieto normované hodnoty rodičov patria vždy do intervalu  $\langle 0, 1 \rangle$ .

Definícia funkcie zmeny  $\Delta(y_1, y_2)$  zabezpečuje, aby výsledná hodnota bola z intervalu<sup>7</sup>  $\langle 0, \min(y_1, y_2) \rangle$ ,  $\langle \min(y_1, y_2), \max(y_1, y_2) \rangle$ , alebo z intervalu  $\langle \max(y_1, y_2), 1 \rangle$  – stačí vhodne zvoliť tvar použitej funkcie<sup>8</sup>. Jednou z možností je použiť tzv. algebraickú skupinu funkcií

$$\begin{aligned} \Delta_1(y_1, y_2) &= y_1 y_2 \\ \Delta_2(y_1, y_2) &= y_1 + y_2 - y_1 y_2 \\ \Delta_3(y_1, y_2) &= y_1^{(1-\alpha)} y_2^\alpha \end{aligned} \quad (8.13)$$

kde  $\alpha$  je parameter z intervalu  $\langle 0, 1 \rangle$ , vážiaci vplyv hodnôt vyskytujúcich sa v štruktúre rodičov. Kombinovaním funkcií dokopy je možné získať hodnotu z viacerých intervalov

$$\Delta_4(y_1, y_2) = \Delta_3(\Delta_1(y_1, y_2), \Delta_2(y_1, y_2)) \quad (8.14)$$

Inou z možností je použiť tzv. Hamacherovu, Einsteinovu či logickú skupinu funkcií [17] alebo inú príhodnú skupinu funkcií.

**Diferenciálny operátor.** Tento operátor pre svoju činnosť používa troch rodičov. Jeho cieľom je upraviť štruktúru jedného rodiča, pričom pre určenie veľkosti tejto úpravy využíva ďalších dvoch rodičov. Novú hodnotu pre  $j$ -tu pozíciu potomka určuje podľa

$$h_j = h_{R1,j} + \alpha(h_{R2,j} - h_{R3,j}) \quad (8.15)$$

teda k hodnote jedného rodiča pripočíta vážený rozdiel druhého a tretieho rodiča, pričom parameter  $\alpha$  určuje, s akou váhou sa daný rozdiel má uvažovať.

<sup>7</sup>Za predpokladu, že oba argumenty funkcie sú z intervalu  $\langle 0, 1 \rangle$ .

<sup>8</sup>Skratka FCB reprezentuje “fuzzy connectives-based” – inšpiráciou pre názov bolo zrejme to, že v úlohe vhodných funkcií je možné použiť fuzzy t-normy, t-konormy a spriemerňujúce operátory.

## 8.4 Operátory pre permutačné kódovanie

**Výmenná mutácia.** Keďže pri permutačnom kódovaní hodnoty na jednotlivých pozíciách nie sú navzájom nezávislé (ľubovoľná dvojica pozícií musí obsahovať rôzne hodnoty), nie je možné mutovať nejakú pozíciu nezávisle od ostatných, ale je nutné meniť štruktúru rodiča ako celok. Prirodzenou zmenou je výmena hodnôt dvoch pozícií navzájom – výber týchto pozícií je ponechaný na náhodu. Pri potrebe rozsiahlejších zmien je možné takýchto výmen vykonať niekoľko.

**Inverzia.** Je to mutačný operátor, ktorý umožňuje realizovať masívnejšie zmeny než predchádzajúci operátor. Pre štruktúru rodiča sa generujú dva deliace body, ktoré definujú medzi sebou postupnosť pozícií. A v rámci tohto segmentu je dané poradie invertované, napríklad segment (5–4–7–2) sa zmení na tvar (2–7–4–5). Rodiča je možné chápať ako cyklickú štruktúru, keď vybraný segment môže zahŕňať súčasne koniec aj začiatok kódu rodiča.

**PMX kríženie** Toto kríženie<sup>9</sup> chápe permutačnú reprezentáciu schému ako reprezentáciu poradia objektov, pričom každý objekt je reprezentovaný jednou pozíciou.

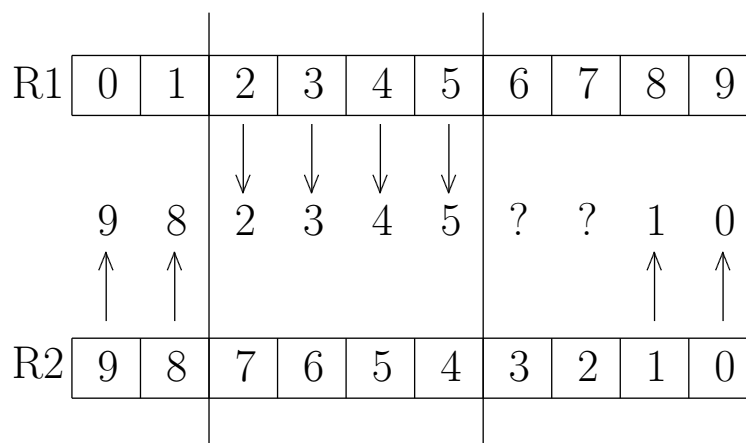
Potomka buduje tak, že z jedného rodiča vyberá súvislý segment niekoľkých pozícií (určený dvomi deliacimi bodmi generovanými náhodne) a zvyšné pozície dopĺňa z druhého rodiča tak, aby sa čo najviac zachovalo *absolútne* poradie objektov dané v druhom rodičovi. V prípade výskytu nejakého konfliktu využíva mapovanie medzi oboma rodičmi.

Predpokladajme existenciu dvoch rodičov podľa obr. 8.5. Keďže deliace body boli náhodne vygenerované medzi druhou a treťou pozíciou a medzi šiestou a siedmou pozíciou, segment štyroch pozícií (pozície 3-6) bol prenesený z prvého rodiča do potomka. Ostatné pozície sú prenášané z druhého rodiča, pričom však nebolo možné preniesť hodnotu vyskytujúcu sa na siedmej (hodnota “3”) a ôsmej (hodnota “2”) pozícií v štruktúre druhého rodiča, pretože tieto hodnoty už boli vložené do budovaného potomka.

Hodnota “3” bola prenesená z prvého rodiča z jeho štvrtej pozície – a druhý rodič má na svojej štvrtej pozícii hodnotu “6”. Keďže táto hodnota ešte nie je obsiahnutá v potomkovi, vloží sa na siedmu pozíciu namiesto hodnoty “3” vyskytujúcej sa na tejto pozícii v druhom rodičovi. Analogicky, na ôsmu pozíciu bude vložená hodnota “7”. Výsledkom teda bude permutácia (9–8–2–3–4–5–6–7–1–0).

---

<sup>9</sup>PMX je skratkou pre “Partially matched crossover”.



Obr. 8.5: Ilustrácia činnosti PMX kríženia

**OX kríženie** Toto kríženie<sup>10</sup> chápe permutačnú reprezentáciu schému ako reprezentáciu cyklického poradia objektov (po poslednom objekte opäť nasleduje objekt prvý), kde každý objekt je opäť reprezentovaný jednou pozíciou.

Potomka buduje tak, že z jedného rodiča vyberá súvislý segment niekoľkých pozícií (určený dvomi deliacimi bodmi generovanými náhodne) a zvyšné pozície doplní z druhého rodiča tak, aby sa čo najviac zachovalo *relatívne* poradie objektov dané v druhom rodičovi.

Predpokladajme opäť existenciu dvoch rodičov podľa obr. 8.5. Keďže deliace body boli náhodne vygenerované medzi druhou a treťou pozíciou a medzi šiestou a siedmou pozíciou, segment štyroch pozícií (pozície 3-6) bol prenesený z prvého rodiča do potomka.

Keďže poradie sa chápe v cyklickom zmysle, druhý rodič reprezentuje aj poradie hodnôt (3-2-1-0-9-8-7-6-5-4) – v prípade, že vyjadrovanie tohto poradia štartuje druhým deliacim bodom. Keďže hodnoty “2”, “3”, “4” a “5” už boli prenesené do potomka z prvého rodiča, tak tieto hodnoty budú z uvedeného poradia vypustené. Redukované poradie bude mať tvar (1-0-9-8-7-6). Keďže toto redukované poradie má toľko členov, koľko je ešte neobsadených pozícií v potomkovi, a žiadny z nich sa v potomkovi ešte nenachádza, je ho možné priamo vložiť do potomka – počnúc pozíciou nasledujúcou po druhom deliacom bode. Výsledkom teda bude permutácia (7-6-2-3-4-5-1-0-9-8).

<sup>10</sup>OX je skratkou reprezentujúcou “Order crossover”.

## 8.5 Porovnanie rekombinačných operátorov

Hodnotiť činnosť genetických operátorov je možné z rôznych hľadísk. Ak bude činnosť evolučného algoritmu chápaná ako snaha skladať nových jedincov z tých stavebných blokov (elementov), z ktorých je vytvorené hľadané riešenie<sup>11</sup>, tak pre činnosť rekombinačných operátorov sú dôležité tri vlastnosti:

- rekombinačný potenciál
- deštrukčný potenciál
- prehľadávací potenciál

Rekombinačný potenciál reprezentuje schopnosť operátora kombinovať jednoduchšie stavebné bloky, vyskytujúce sa v rôznych jedincoch (hrajúcich úlohu rodičov), do zložitejších stavebných blokov tvoriacich súčasť štruktúry novo vytváraných potomkov. Deštrukčný potenciál reprezentuje opačnú vlastnosť – schopnosť operátora rozbiť zložitejšie stavebné bloky, vyskytujúce sa v rodičovských jedincoch, a prenášať ich časti do rôznych novo vytvorených jedincov. Vo všeobecnosti vysoký rekombinačný potenciál favorizuje genetický operátor zatiaľ čo vysoký deštrukčný potenciál ho diskvalifikuje<sup>12</sup>.

Prehľadávací potenciál operátora vyjadruje, koľko rôznych jedincov je možné vytvoriť z dvojice rodičov. Operátor s menším potenciálom síce môže dokázať vytvoriť rovnakých potomkov ako operátor s potenciálom vyšším – ale na to potrebuje byť aplikovaný viackrát<sup>13</sup>.

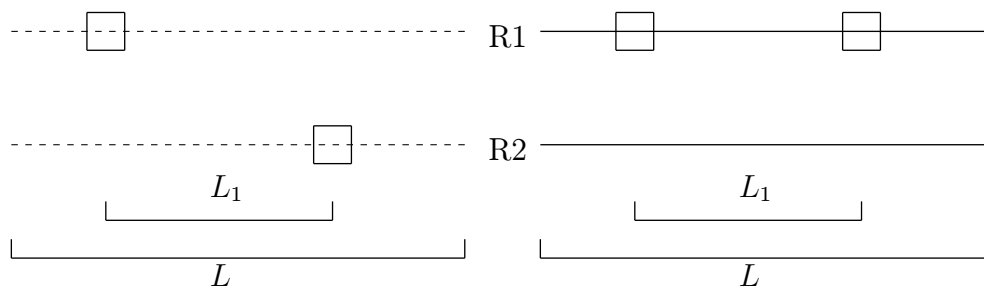
V ďalšom texte sa zameriame na porovnanie operátorov pre binárnu reprezentáciu. Pre účely analýzy jednotlivých krížiacich operátorov uvažujme situáciu podľa obr. 8.6. Sú dané dva rodičovské jedince, každý je tvorený ako lineárna postupnosť  $L$  pozícií. Budú nás zaujímať stavebné bloky (každý z nich zaberá iba jednu pozíciu) nachádzajúce sa na pozíciách, vymedzujúcich v jedincoch segment dĺžky  $L_1$ .

---

<sup>11</sup>Za identifikáciu a selekciu týchto stavebných elementov sú zodpovedné tie časti algoritmu, ktoré sú zdrojom selekčného tlaku.

<sup>12</sup>To platí najmä v prípade, ak algoritmus je schopný preferovať tie stavebné elementy, z ktorých je vytvorené hľadané riešenie. Ak však algoritmus selektuje a preferuje aj stavebné bloky, z ktorých hľadané riešenie nie je možné poskladať, tak kombinovanie týchto blokov bez ich rozbiťania nemusí byť vítanou vlastnosťou.

<sup>13</sup>Napríklad ak jeden rodič obsahuje samé jednotky a druhý samé nuly, tak potomok, v ktorom sa pravidelne striedajú obe hodnoty, môže byť vytvorený minimálne jednou aplikáciou uniformného kríženia ale minimálne toľkými aplikáciami jednobodového kríženia, koľko je možných miest na umiestnenie deliaceho bodu.



Obr. 8.6: Kombinovanie stavebných blokov (vľavo) a deštrukcia kombinovaného bloku (vpravo)

Ľavá strana zobrazuje situáciu, keď sledované stavebné bloky sa nachádzajú v rôznych rodičoch, a teda pre získanie kombinovaného bloku je potrebné, aby krížiaci operátor preniesol oba vyznačené bloky do jedného (ľubovoľne ktorého) potomka. Pravdepodobnosť tejto rekombinácie vyjadruje rekombinačný potenciál operátora.

Pravá strana reprezentuje situáciu, keď oba bloky sú už skombinované v jednom jedincovi. V tomto prípade je potrebné, aby operátor preniesol oba bloky do toho istého jedinca a teda nerozdelil ich medzi oboch potomkov. Komplement pravdepodobnosti tohto zachovania môže byť použitý pre vyjadrenie deštruktívneho potenciálu príslušného operátora.

Nasledujúca analýza krížiacich operátorov je založená na [41]. Pri *mnohobodovom krížení* výsledok závisí od toho, či medzi pozície, na ktorých sa nachádzajú dané stavebné bloky, prípadne párny alebo nepárny počet deliacich bodov. Pri zohľadnení rôznych spôsobov distribúcie deliacich bodov do jednotlivých oblastí jedincov potom výsledná pravdepodobnosť (rekombinácie blokov alebo zachovania existujúcej kombinácie blokov) môže byť určená podľa vzťahu

$$\sum_{i=0}^n \binom{n}{i} \left( \frac{L_1 - 1}{L - 1} \right)^i \left( \frac{L - L_1}{L - 1} \right)^{n-i} K \quad (8.16)$$

kde  $n$  reprezentuje počet použitých deliacich bodov. Keďže  $L - 1$  reprezentuje počet všetkých možných miest, na ktoré je možné umiestniť deliace body a  $L_1 - 1$  zase počet miest pre deliace body medzi vyznačenými blokmi, tak druhý faktor uvedenej sumy vyjadruje pravdepodobnosť, že medzi blokmi bude umiestnených  $i$  deliacich bodov. Analogicky, tretí faktor reprezentuje

pravdepodobnosť, že zvyšné deliace body budú umiestnené mimo segmentu ohraničeného sledovanými blokmi. Prvý term je kombinačný a vyjadruje koľkými spôsobmi je možné vybrať  $i$  deliacich bodov spomedzi všetkých  $n$  deliacich bodov.

Posledný faktor  $K$  reprezentuje korekčný člen hovoriaci, akým spôsobom sa má uvažovať tá-ktorá distribúcia deliacich bodov. Ak sa jedná o kombinovanie blokov z rôznych rodičov (obr. 8.6 vľavo), tak

$$K = \begin{cases} 1, & i \% 2 = 1 \\ 0, & i \% 2 = 0 \end{cases} \quad (8.17)$$

a vzťah (8.16) vyjadruje pravdepodobnosť rekombinácie. Ak by sa jednalo o prežitie existujúcej kombinácie blokov (obr. 8.6 vpravo), tak

$$K = \begin{cases} 0, & i \% 2 = 1 \\ 1, & i \% 2 = 0 \end{cases} \quad (8.18)$$

a vzťah (8.16) by vyjadroval pravdepodobnosť zachovania.

V predchádzajúcich definíciách korekčného člena sa uvažovalo, že potrebný stavebný blok sa vždy nachádza iba v jednom z rodičov. V skutočnosti sa však môže nachádzať aj v druhom rodičovi – prípad, keď obaja rodičia majú na nejakej pozícii rovnaké hodnoty. Vtedy dva bloky môžu byť kombinované/zachované aj pri párnom/nepárnom počte deliacich bodov pripadajúcich medzi uvažované pozície. Ak  $p_{=}$  bude pravdepodobnosť, že obaja rodičia majú na nejakej pozícii rovnaké hodnoty, potom namiesto nulovej hodnoty korekcie by sa použila hodnota

$$p_{=}(1 - p_{=}) + (1 - p_{=})p_{=} + p_{=}p_{=} = 2p_{=} - p_{=}^2 \quad (8.19)$$

vyjadrujúca pravdepodobnosť toho, že prvý stavebný blok sa nachádza v oboch rodičoch, alebo druhý stavebný blok sa nachádza v oboch rodičoch, alebo oba stavebné bloky sa nachádzajú v oboch rodičoch.

Pri *miešajúcom krížení* je situácia podobná ako pri mnohobodovom krížení. Rozdiel je iba v tom, že aj keď uvažované stavebné bloky vymedzujú segment dĺžky  $L_1$ , vďaka náhodnému premiešaniu pozícií sa táto dĺžka môže zmeniť na ľubovoľnú hodnotu z  $\{2, \dots, L\}$ , pričom všetky možnosti sú rovnako pravdepodobné. Preto výsledná hodnota pravdepodobnosti už nebude závisieť na  $L_1$  ale bude priemerom cez všetky možnosti

$$\frac{1}{L-1} \sum_{L_1=2}^L \sum_{i=0}^n \binom{n}{i} \left(\frac{L_1-1}{L-1}\right)^i \left(\frac{L-L_1}{L-1}\right)^{n-i} K \quad (8.20)$$

príčom hodnota korekčného člena ostáva rovnaká ako v predchádzajúcom prípade.

Pri *uniformnom krížení* výsledok závisí od toho, či pozície v maske, zodpovedajúce pozíciám uvažovaných blokov v jedincoch, obsahujú rovnaké hodnoty alebo hodnoty rôzne. Pri zohľadnení rôznych spôsobov distribúcie hodnôt v maske potom výsledná pravdepodobnosť môže byť určená podľa vzťahu

$$\sum_{i=0}^2 \binom{2}{i} (p_0)^i (1-p_0)^{2-i} K \quad (8.21)$$

kde  $p_0$  je pravdepodobnosť, že v maske je generovaná hodnota 0, zatiaľ čo hodnota 1 je v maske generovaná s komplementárnou pravdepodobnosťou  $1-p_0$ .

Druhý člen uvedenej sumy reprezentuje pravdepodobnosť, že  $i$  pozícií (zo sledovaných dvoch pozícií obsahujúcich uvažované stavebné bloky) bude preberaných z jedného rodiča. Tretí člen vyjadruje pravdepodobnosť, že zvyšné pozície (zo sledovaných pozícií) budú preberané z druhého rodiča. Kombinačný prvý term vyjadruje, koľkými spôsobmi je možné vybrať  $i$  pozícií z dvoch sledovaných pozícií.

Posledný korekčný člen  $K$  opäť hovorí, akým spôsobom sa má uvažovať daná distribúcia hodnôt v maske. Ak sa jedná o kombinovanie blokov z rôznych rodičov (obr. 8.6 vľavo), tak

$$K = \begin{cases} 0, & i \in \{0, 2\} \\ 1, & i \in \{1\} \end{cases} \quad (8.22)$$

a vzťah (8.21) vyjadruje pravdepodobnosť rekombinácie. Ak by sa jednalo o prežitie kombinácie oboch blokov (obr. 8.6 vpravo), tak

$$K = \begin{cases} 1, & i \in \{0, 2\} \\ 0, & i \in \{1\} \end{cases} \quad (8.23)$$

a vzťah (8.21) by vyjadroval pravdepodobnosť zachovania.

Ak sa opäť uvažuje aj situácia, že rodičovské jedince môžu mať na nejakej pozícii rovnaké hodnoty, tak namiesto nulovej hodnoty korekcie by sa použila hodnota podľa (8.19).

Pri *segmentovom krížení* výsledok závisí od toho, či medzi pozíciami uvažovaných blokov v rodičovských jedincoch dôjde k párnemu alebo nepárnemu počtu zmien segmentov. Pri zohľadnení rôznych spôsobov distribúcie zmien

segmentov v úseku medzi danými blokmi potom výsledná pravdepodobnosť môže byť určená podľa vzťahu

$$\sum_{i=0}^{L_1-1} \binom{L_1-1}{i} (p_z)^i (1-p_z)^{L_1-1-i} K \quad (8.24)$$

kde  $p_z$  je pravdepodobnosť, že dochádza k zmene segmentu a teda nejaká nasledujúca pozícia bude patriť k inému segmentu ako pozícia predchádzajúca, zatiaľ čo pokračovanie segmentu je umožnené s pravdepodobnosťou  $1 - p_z$ .

Druhý člen uvedeného vzťahu reprezentuje pravdepodobnosť, že  $i$  krát dôjde k zmene segmentu (na  $i$  pozíciách spomedzi pozícií vymedzených danými blokmi), pričom každá zmena zapríčiňuje zmenu rodiča z ktorého je preberaná hodnota do štruktúry potomka. Tretí člen vyjadruje pravdepodobnosť, že na zvyšných pozíciách (v úseku medzi sledovanými blokmi) nedochádza k zmene segmentu, a teda nedochádza k zmene rodiča z ktorého budú preberané hodnoty. Kombinačný prvý term vyjadruje, koľkými spôsobmi je možné vybrať  $i$  pozícií v úseku sledovaných pozícií medzi danými blokmi.

Posledný korekčný člen  $K$  teraz vyjadruje, akým spôsobom sa má uvažovať daná distribúcia prepínacích pozícií. Ak sa jedná o kombinovanie blokov z rôznych rodičov (obr. 8.6 vľavo), tak

$$K = \begin{cases} 0, & i \% 2 = 0 \\ 1, & i \% 2 = 1 \end{cases} \quad (8.25)$$

a vzťah (8.24) vyjadruje pravdepodobnosť rekombinácie. Ak by sa jednalo o prežitie kombinácie blokov (obr. 8.6 vpravo), tak

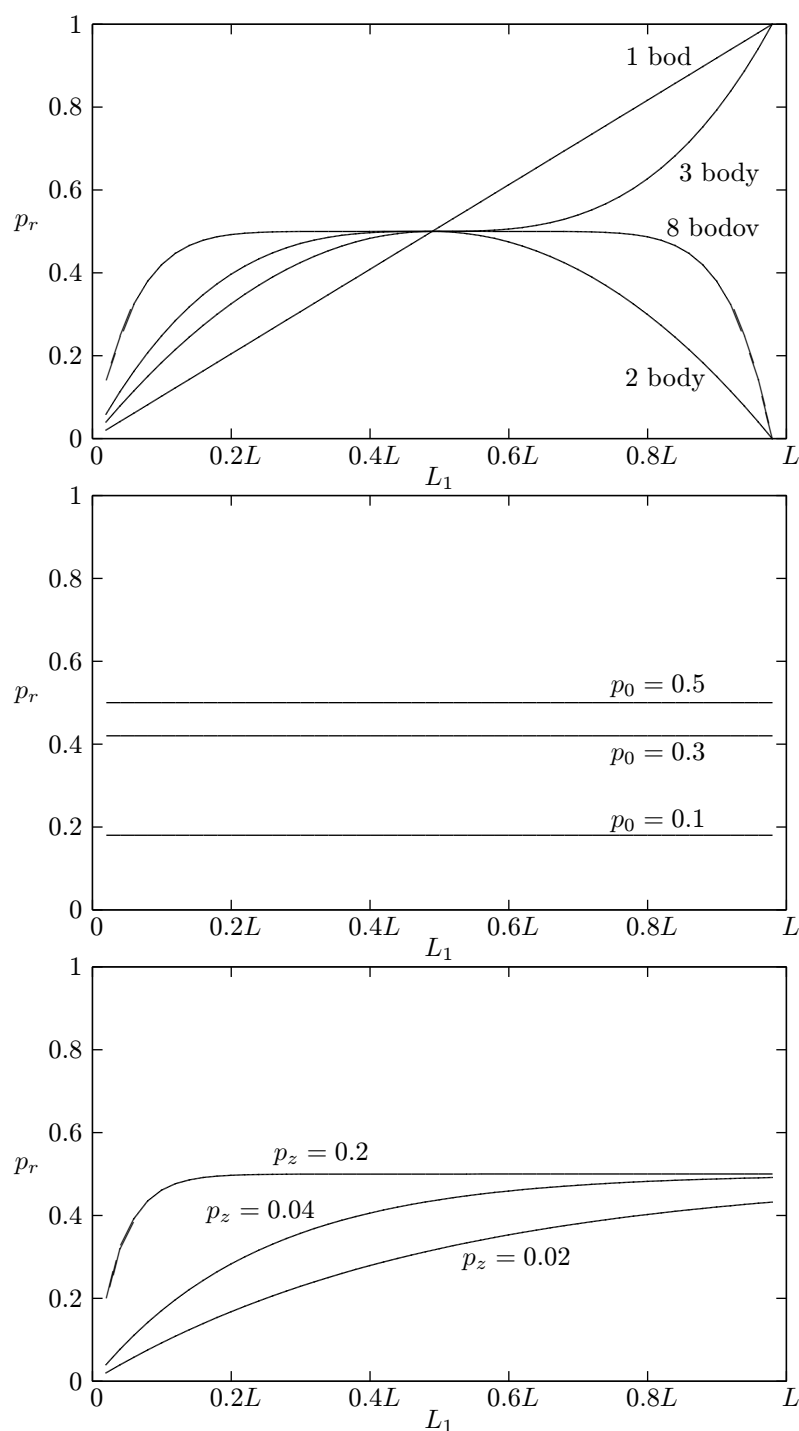
$$K = \begin{cases} 1, & i \% 2 = 0 \\ 0, & i \% 2 = 1 \end{cases} \quad (8.26)$$

a vzťah (8.24) by vyjadroval pravdepodobnosť zachovania.

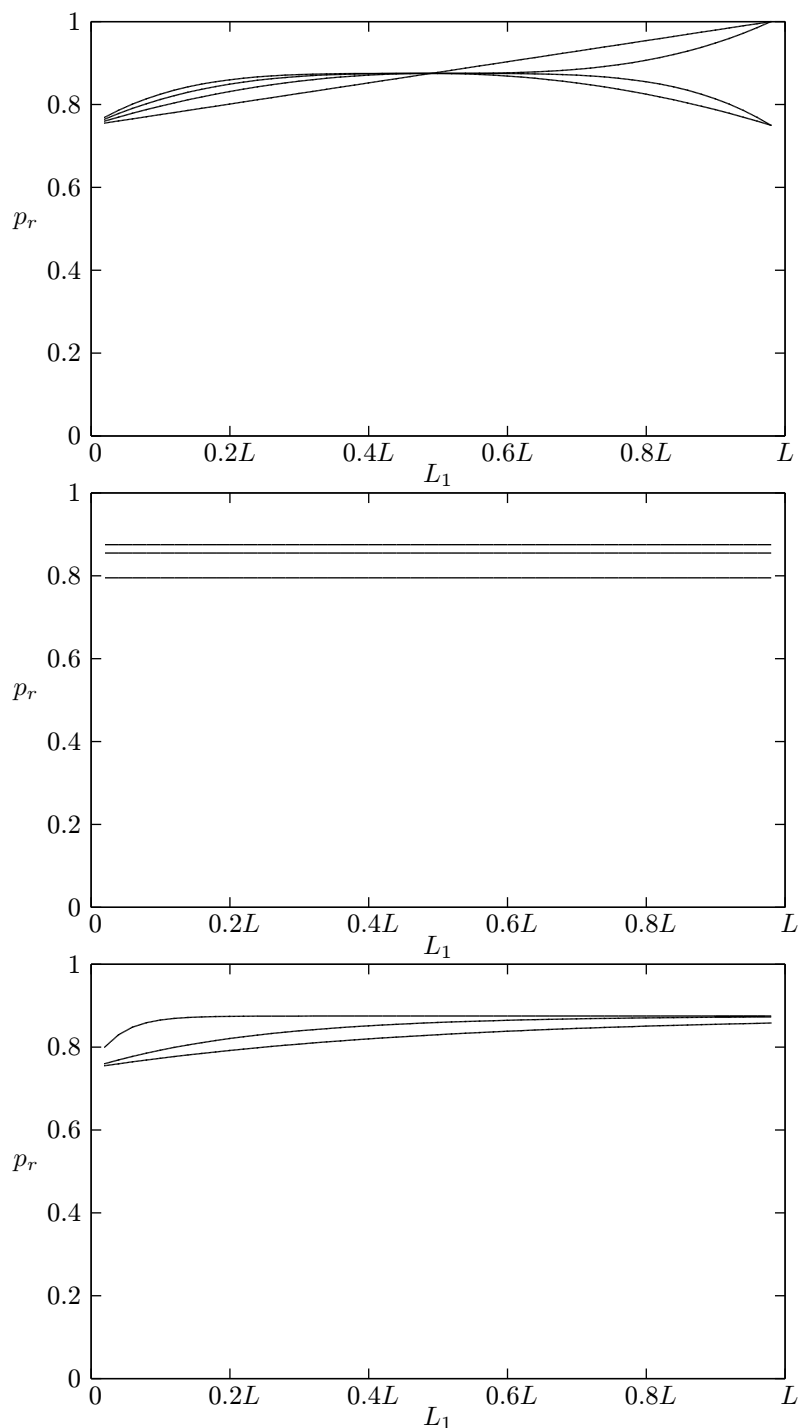
Ak sa opäť uvažuje aj situácia, že rodičovské jedince môžu mať na nejakej pozícii rovnaké hodnoty, tak namiesto nulovej hodnoty korekcie by sa opäť použila hodnota podľa (8.19).

Obr. 8.7 ilustruje rekombinačný potenciál krížiacich operátorov, pričom sa neuvažoval prípad, keď kombinácia blokov môže byť výsledkom aplikácie operátora aj vtedy, ak sa hodnoty na sledovaných pozíciách preberajú iba z jedného rodiča (v prípade takýchto distribúcií sa použili nulové hodnoty





Obr. 8.7: Kombinácia dvoch blokov pri mnohobodovom krížení (hore), uniformnom krížení (v strede) a segmentovom krížení (dole) – popis v texte



Obr. 8.8: Kombinácia dvoch blokov pri mnohobodovom (hore), uniformnom (v strede) a segmentovom krížení (dole) pri nenulových korekciách

korekcie). Následkom toho uvedené priebehy reprezentujú spodný odhad rekombinačného potenciálu.

Pri mnohobodovom krížení (obr. 8.7 hore) má použitie nepárneho počtu deliacich bodov väčšiu nádej kombinovať stavebné bloky navzájom než použitie párneho počtu. Celkový potenciál (reprezentovaný ako plocha pod krivkou) je rovnaký pre nepárne počty deliacich bodov, pri párnom počte je tým väčší, čím je použitý väčší počet deliacich bodov.

Pri uniformnom krížení (obr. 8.7 v strede) pravdepodobnosť kombinovania je invariantná od vzdialenosti kombinovaných blokov. Celkový potenciál je najväčší pri generovaní hodnôt v maske s rovnakou pravdepodobnosťou a je rovnako veľký ako pri mnohobodovom krížení využívajúcom nepárny počet deliacich bodov.

Rekombinačný potenciál pri segmentovom krížení (obr. 8.7 dole) je citlivý na vzdialenosť medzi kombinovanými blokmi, pričom táto citlivosť je výraznejšia pri blízko umiestnených blokoch než pri blokoch vzdialenejších.

Keďže v prípade korekcií neuvažujúcich výskyt rovnakých hodnôt v oboch rodičoch je korekcia pre rekombináciu blokov komplementom ku korekcii pre zachovanie bloku, tak aj pravdepodobnosti zachovania sú komplementom k uvedeným pravdepodobnostiam kombinovania blokov.

Obr. 8.8 ilustruje rekombinačný potenciál krížiacich operátorov pri uvažovaní prípadu, keď kombinácia blokov môže byť výsledkom aplikácie operátora aj vtedy, ak sa hodnoty na sledovaných pozíciách preberajú iba z jedného rodiča (v prípade takýchto distribúcií sa použili nenulové hodnoty korekcie).

Z dôvodu prehľadnosti nie sú v obrázku uvedené hodnoty jednotlivých parametrov – sú však použité tie isté hodnoty ako na obr. 8.7. Je zrejmé, že kvôli nahradeniu nulových korekcií nenulovými došlo k posunutiu priebehov do vyšších hodnôt so zachovaním základného tvaru kriviek.

Pre výpočet korekcií bola použitá hodnota  $p_{=} = 0.5$  indikujúca rovnomernú distribúciu hodnôt v populácii. Toto reprezentuje stav na začiatku evolučného hľadania. Neskoršie fázy hľadania sú poznačené konvergenciou, keď na nejakej pozícii sa častejšie vyskytuje nejaká hodnota zatiaľ čo výskyt hodnoty opačnej je menej častý. Ak by operátor mal kombinovať v tejto neskoršej fáze bloky, ktoré sú zastúpené v populácii nadpriemerne, tak hodnota  $p_{=}$  by bola vyššia a krivky v grafoch by boli posunuté ešte viac nahor (rekombinačný potenciál operátora by bol vyšší). Ak by sa však kombinovali bloky s minoritným zastúpením v populácii, tak hodnota  $p_{=}$  by bola menšia a krivky by boli posunuté nahor menej ako na obr. 8.8.



---

## Časť IV

# Pripraviť sa, pozor, štart



## Kapitola 9

# Nastavovanie parametrov

Zatiaľ čo napríklad pre voľbu plochy vhodnosti neexistuje veľa doporučení, voľba hodnôt parametrov je už pomerne rozpracovanou<sup>1</sup> súčasťou výskumu a praxe v oblasti používania evolučných algoritmov. Aj keď pod nastavením parametra sa najčastejšie rozumie určenie jeho numerickej hodnoty (prah orezania, pravdepodobnosť mutácie, počet krížiacich bodov, veľkosť populácie, ap.), patrí tu aj voľba realizácie nejakého bloku algoritmu (výber selekčnej metódy, voľba spôsobu inicializácie, ap.) ako aj rozhodnutie o tom, či nejaká súčasť vôbec bude použitá alebo nie (premapovanie vhodnosti, použitie nejakého typu operátora, podpora rôznorodosti populácie, ap.).

Jednotlivé možné spôsoby nastavovania parametrov môžu byť delené do skupín podľa rôznych kritérií. Z hľadiska pôvodcu nastavenia je možné rozoznávať dva typy nastavenia:

- exogénne,
- endogénne.

Pri exogénnom nastavení parametrov je zdrojom nastavenia vonkajší agent (najčastejšie to je človek, ale môže to byť aj softvérový agent). Tento agent nastaví parametre algoritmu na vhodné hodnoty, ktoré sú následne používané evolučným algoritmom pri jeho realizácii. Pri endogénnom nastavení je zdrojom nastavenia samotný algoritmus, ktorý pre nastavenie hodnôt svojich parametrov nepotrebuje vonkajší zdroj.

Iným kritériom je dynamika hodnôt parametrov. Z tohto hľadiska je možné spôsoby nastavovania deliť na:

---

<sup>1</sup>Kedže problematika svojím rozsahom značne presahuje priestor tejto kapitoly, je uvedených iba niekoľko ukážok ilustrujúcich jednotlivé triedy prístupov.

- statické,
- dynamické.

Pri statickom nastavení je parameter na začiatku nastavený na nejakú fixnú hodnotu, ktorá ostáva nemenná počas celej realizácie algoritmu. Pri dynamickom nastavení sa hodnoty parametrov menia v závislosti na nejakej charakteristike odvodenej z činnosti algoritmu (niekedy sa toto označuje ako riadenie parametrov).

Exogénne nastavovanie je najčastejšie kombinované so statickým nastavovaním, aj keď je možná aj jeho kombinácia s nastavovaním dynamickým. V dynamickom prípade sa aktuálna hodnota parametra môže viazať na čas, vyjadrený napríklad číslom aktuálnej generácie – hodnota parametra sa deterministicky mení podľa nejakej vopred určenej časovej závislosti bez zohľadňovania spätnej väzby z procesu prehľadávania.

Naproti tomu endogénne nastavovanie je používané výhradne v kombinácii s dynamickým nastavovaním, keď evolučný algoritmus modifikuje hodnoty parametrov podľa aktuálneho stavu prehľadávania. K tomu má k dispozícii dva mechanizmy – adaptívne a samoadaptívne nastavovanie.

## 9.1 Exogénne nastavovanie

V minulosti boli vyskúšané štyri základné spôsoby, ako je možné nastaviť hodnoty parametrov evolučného algoritmu:

- overené hodnoty,
- vzorce,
- pokus-omyl,
- metaevolúcia.

**Overené hodnoty.** Tento spôsob je najjednoduchší a aj najrýchlejší pre ľudského agenta – jednoducho sa použijú hodnoty, ktoré sa osvedčili v minulosti (či už sa jedná o vlastnú skúsenosť alebo o skúsenosť iných). Ukazuje sa, že spoliehanie sa na robustnosť evolučných algoritmov je schodnou stratégiou – avšak je to stratégia očividne suboptimálna.



**Vzorcie.** Určovanie vzorcov pre priamy výpočet hodnôt parametrov môže byť založené na rôznych ideách. Jednou z nich je empirické hľadanie závislostí medzi jednotlivými parametrami pri riešení nejakej triedy úloh. Ukážkou takéhoto prístupu je [38], určujúci pravdepodobnosť mutácie podľa

$$p_m = \frac{1.76}{\mu\sqrt{l}} \quad (9.1)$$

kde mutácia je dávaná do súvisu s veľkosťou populácie a počtom kódovacích pozícií (dimenzionalitou priestoru prehľadávania).

Iným prípadom je výpočet založený na myšlienke zabezpečiť určitú kvalitu. Príkladom je výpočet veľkosti populácie [19], určujúci vhodnú hodnotu pre permutačnú kódovaciu schému. Konzervatívne sa predpokladá, že existuje iba jedno optimálne poradie a hľadá sa taká minimálna hodnota  $\mu$ , pre ktorú je pravdepodobnosť, že náhodne vybraná populácia bude obsahovať všetky susedné dvojice z tohto optimálneho usporiadania, rovná aspoň hodnote  $p$ .

Keďže jedno poradie obsahuje práve  $n$  dvojíc susedov (chápané cyklicky – posledný v poradí susedí s prvým) a možných dvojíc je  $n(n-1)/2$ , tak pravdepodobnosť, že nejaká dvojica nebude súčasťou poradia reprezentovaného jedným jedincom, je

$$1 - \frac{n}{n(n-1)/2} = \frac{n-3}{n-1} \quad (9.2)$$

Ak výskyt dvojice susedov v nejakom jedincovi nebude závisieť od jej výskytu v inom jedincovi, tak pravdepodobnosť, že nejaká dvojica bude reprezentovaná aspoň jedným jedincom, je

$$1 - \left(\frac{n-3}{n-1}\right)^\mu \quad (9.3)$$

Ak výskyt jednotlivých dvojíc nie je navzájom závislý, tak pravdepodobnosť, že všetky dvojice optimálneho riešenia sú obsiahnuté v populácii, je

$$\left(1 - \left(\frac{n-3}{n-1}\right)^\mu\right)^n \quad (9.4)$$

a táto pravdepodobnosť musí byť väčšia alebo rovná než požadovaná pravdepodobnosť  $p$  (zvyčajne sa pre  $p$  používajú hodnoty 0.95, 0.99 alebo 0.999). Odtiaľ pre minimálnu veľkosť populácie vyplýva

$$\mu \geq \frac{\ln(1 - \sqrt[p]{p})}{\ln((n-3)/(n-1))} \quad (9.5)$$

Iným príkladom je dynamické nastavenie parametra, meniace sa podľa deterministicky predpísaného vzťahu. Ilustráciou je nastavenie pravdepodobnosti mutácie  $p_m$  podľa [1]

$$p_m(t) = \left(2 + (l - 2) \frac{t}{t_{max}}\right)^{-1} \quad (9.6)$$

reprezentujúce hyperbolickú závislosť na generácii, pričom sa začína v situácii, že je mutovaných priemerne polovica pozícií v jedincovi, a končí sa nastavením, pri ktorom sa v priemere v každom jedincovi mutuje iba jedna pozícia použitej reprezentácie. Vychádza sa z predstavy, že veľké mutačné zmeny môžu byť dobré v skorších etapách prehľadávania, pretože podporujú skúmanie plochy vhodnosti. V neskorších etapách je však cieľom skôr jemné preskúmanie okolia bodu, reprezentovaného jedincom, a teda sú žiadané iba malé mutačné zmeny.

**Pokus–omyl.** Asi najviac používaný prístup v praktickom nasadení. Jeho cieľom je jednoducho otestovať možné hodnoty parametrov priamo na riešenom probléme, identifikovať najvhodnejšiu kombináciu hodnôt a použiť túto kombináciu pre finálnu realizáciu evolučného algoritmu a vyriešenie daného problému.

Samotné testovanie má podobu spustenia algoritmu v rámci nejakého definovaného časového intervalu (počtu generácií alebo vygenerovaní či vyhodnotení jedincov). Tento interval musí byť dostatočne krátky, aby bolo možné opakovane realizovať algoritmus s viacerými nastaveniami, a súčasne dostatočne dlhý, aby bolo možné získať obraz o chovaní sa algoritmu pri testovanej kombinácii hodnôt parametrov.

Po uplynutí daného testovacieho intervalu je testovaná kombinácia hodnôt ohodnotená. Aj keď je možné použiť hodnoty viažúce sa k okamihu zastavenia algoritmu (napr. dosiahnutá vhodnosť), častejšie sa používajú charakteristiky integrujúce celý dovtedajší priebeh prehľadávania. Príkladmi takýchto charakteristík sú:

- *on-line vhodnosť* v nejakom čase je priemer vhodností všetkých jedincov, ktoré boli vygenerované a začlenené do populácie do daného času,
- *off-line vhodnosť* v nejakom čase je priemer vhodností najlepších jedincov z prechádzajúcich generácií.

Problémom tohto prístupu je to, že zvyčajne nie je možné preskúmať všetky kombinácie hodnôt parametrov, pretože by to bolo z časového hľadiska neakceptovateľné (obzvlášť v prípadoch, keď počet týchto kombinácií

je nekonečne veľký). Preto je nutné vybrať iba určitú (pomerné malú) podmnožinu možných kombinácií a zamerať úsilie iba na ňu.

Alternatívnou stratégiou je v každom okamihu sa zamerať iba na jeden parameter a skúmať jeho rôzne hodnoty, zatiaľ čo hodnoty ostatných parametrov ostávajú fixované. Po nájdení najlepšej hodnoty sa daný parameter stáva fixovaným a prechádza sa na skúmanie hodnôt iného parametra. Takto je možné realizovať zvládnuteľný počet testov – avšak je nutné si uvedomiť, že výsledkom bude pravdepodobne iba suboptimálne nastavenie parametrov, pretože jednotlivé parametre v skutočnosti navzájom interagujú nelineárnym spôsobom.

**Metaevolúcia.** Hľadanie vhodných hodnôt parametrov evolučného algoritmu je zložitý optimalizačný problém, ako stvorený pre riešenie evolučným algoritmom. Inak povedané, je možné použiť dva evolučné algoritmy:

- “riešiaci” (primárny) pre samotné riešenie úlohy,
- “nastavovací” (sekundárny) pre nastavenie hodnôt parametrov riešiacieho algoritmu.

Každý jedinec nastavovacieho algoritmu bude reprezentovať jednu možnú kombináciu hodnôt parametrov riešiacieho algoritmu (pre kvantitatívne parametre je potrebné vhodne reprezentovať reálne čísla, pre kvalitatívne parametre zase binárne alebo n-árne hodnoty).

Pre zistenie kvality nejakej kombinácie hodnôt parametrov (vhodnosti jedinca nastavovacieho algoritmu) je potrebné danú kombináciu otestovať rovnakým spôsobom ako v predchádzajúcom prístupe. Hodnoty sa vložia do riešiacieho algoritmu a ten sa realizuje počas nejakého časového intervalu. Po jeho skončení sa chovanie riešiacieho algoritmu s danou kombináciou hodnôt parametrov ohodnotí numerickou mierou a táto hodnota sa použije ako vhodnosť príslušného jedinca nastavovacieho algoritmu.

Vlastnosťou nastavovacieho algoritmu je, že určovanie vhodnosti jedincov je časovo najnáročnejšou operáciou, ktorá limituje počet jedincov generovaných nastavovacím algoritmom. A samozrejme, pribúda ešte jeden problém – ako nastaviť vhodné hodnoty parametrov nastavovacieho algoritmu. Aj keď aj pre tento prípad je možné použiť metaevolúciu (a rekurzívne navyšovať počet použitých evolučných algoritmov), zvyčajne sa používajú overené hodnoty, vhodné pre úlohu numerickej optimalizácie.

## 9.2 Adaptívne nastavovanie

Pri tomto spôsobe sa algoritmus snaží nájsť hodnoty parametrov, ktoré sú najlepšie pre aktuálnu situáciu – nevyhnutným faktorom je využívanie spätnej väzby z procesu prehľadávania, poskytujúcej informáciu o aktuálnom stave prehľadávania a účinnosti aktuálneho nastavenia algoritmu v tomto prehľadávaní.

Parametre algoritmu sa takto adaptujú na aktuálny stav, pričom adaptačný mechanizmus môže mať rôznu podobu. Dôležitou vlastnosťou použitého adaptačného mechanizmu je to, že pôsobí ako dodatočná súčasť algoritmu, pridaná k pôvodnej štruktúre algoritmu – nie je prítomná v algoritme, ktorý nerealizuje adaptáciu svojich parametrov.

Základné typy používaných adaptačných mechanizmov sú najčastejšie založené na niektorom z princíпов:

- distribúcia kreditu,
- reakcia na udalosti.

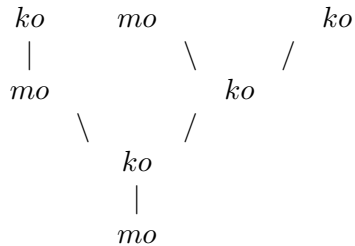
Prvý princíp je používaný v prípade výberu jednej z viacerých alternatív. Dostupné alternatívy sú porovnávané s ohľadom na ich úspešnosť (meranú ako nejaká charakteristika, závislá na aplikácii jednotlivých alternatív počas prehľadávania). Za dosahované výsledky sú odmeňované kreditom a na základe akumulovaného kreditu sa rozhoduje, ktorá alternatíva sa použije.

Typickým príkladom adaptácie založenej na distribúcii kreditu je rozhodovanie, ako kombinovať operátory generujúce nových potomkov. Pri rozhodovaní je možné sa riadiť ich úspešnosťou v danej fáze prehľadávania a preferovať ten operátor, ktorý je aktuálne úspešnejší a produkuje vhodnejších jedincov.

Jedna z možných schém je prezentovaná v [20]. Pri tomto spôsobe je jedinec produkovaný vždy iba jedným genetickým operátorom – mutáciou alebo krížením. Ak vyprodukovaný jedinec je lepší ako aktuálny medián vhodnosti v populácii, tak operátor, ktorý ho vytvoril, získava kredit (a rovnako aj operátor, ktorý vytvoril jeho rodičov, rodičov jeho rodičov, ... až do určitej hĺbky). Za vytvorenie jedinca je určitý kredit, za vytváranie jeho predchodcov je kredit menší – je tým menší, čím je daný predchodca vzdialenejší v histórii.

Pre ilustráciu predpokladajme, že uvažovaná história vytvárania jedinca

je daná schémou (je uvažovaná hĺbka štyroch generácií)



teda jedinec bol vytvorený pomocou mutačného operátora, jeho rodič pomocou krížiaceho operátora, jeden z jeho prarodičov pomocou mutačného operátora (a rodič tohto prarodiča zase vznikol krížením) zatiaľ čo druhý prarodič vznikol krížením (a jeho rodičia krížením a mutáciou). Operátor získava kredit za účasť na tvorbe jedinca: 1.0 za samotného jedinca, 0.8 za rodiča,  $0.8^2$  za prarodiča a  $0.8^3$  za praprarodiča. Teda v tomto prípade mutačný operátor získava celkový kredit  $1.0 + 0.8^2 + 0.8^3$  zatiaľ čo krížiaci operátor získava  $0.8 + 0.8^2 + 2(0.8^3)$ .

Algoritmus udržiava informáciu o určitom počte posledne generovaných jedincov a kreditoch priradených oboj genetickým operátorom za generovanie týchto jedincov a ich predchodcov. Pri vložení nového záznamu sa najstarší záznam odstráni. Pravdepodobnosť výberu operátora pre vytvorenie potomka je potom daná pomerom celkového kreditu daného operátora a súčtu kreditov oboch operátorov (algoritmicke sa môže strážiť, aby pravdepodobnosti boli z nejakého intervalu napríklad  $\langle 0.05, 0.95 \rangle$  aby bola garantovaná minimálna šanca pre každý z operátorov).

Iným princípom adaptácie je reakcia na udalosti. Je definovaná nejaká sada udalostí a pravidiel. Každá udalosť je zviazaná s jednou alebo viacerými charakteristikami procesu prehľadávania, napríklad hodnota nejakej kvantitatívnej charakteristiky presiahne resp. klesne pod vopred definovaný prah alebo je detekovaná prítomnosť nejakej kvalitatívnej charakteristiky.

Počas procesu prehľadávania sa sleduje, či nastala niektorá z preddefinovaných udalostí alebo nie. Ak nastala, potom sa aktivuje príslušné pravidlo resp. procedúra a riadený parameter zmení svoje nastavenie. Akcie sú teda vopred známe, to čo je neznáme a závisí na aktuálnom stave je kedy sú tieto akcie vykonávané.

Tento prístup môže byť ilustrovaný dynamickým kódovaním parametrov [39]. Pri reprezentácii reálnej hodnoty pomocou binárnej kódovacej schémy je pre kódovanie hodnoty potrebné použiť sekvenciu bitov. Voľba konkrétnej hodnoty počtu použitých bitov by mala byť čo najmenšia (aby priestor pre-

hľadávania bol čo najmenší) a súčasne dostatočne veľká pre reprezentáciu daného čísla s vyhovujúcou presnosťou.

Pri adaptívnom kódovaní je možné použiť menší pevný počet bitov, avšak meniť ich význam pri mapovaní na reálnu hodnotu. Celé hľadanie prebieha v jednoduchom cykle:

- prehľadávanie až populácia skonverguje,
- prekódovanie populácie s väčším rozlíšením.

Tento cyklus sa opakuje dovtedy, pokiaľ sa nezíska výsledok s požadovanou presnosťou.

Pre kódovanie reálneho čísla sa bude používať  $l_i$  bitov a dekódovanie na číslo z intervalu  $\langle min, max \rangle$  bude realizované vzťahom (4.5). Predpokladá sa použitie hodnoty  $l_i$  menšej ako je potrebné pre dosiahnutie požadovanej rozlišovacej presnosti. Pri takomto nastavení sa bude prehľadávať priestor prehľadávania, až pokiaľ populácia neskonverguje. Pod konvergenciou sa tu rozumie situácia, že hodnoty reprezentované rôznymi jedincami, ktoré pôvodne boli roztrúsené v intervale  $\langle min, max \rangle$ , sa presunuli (buď všetky alebo aspoň ich nejaká vopred definovaná časť) do jedného z cieľových podintervalov

$$\begin{aligned} & \langle min, min + (max - min)/2 \rangle \\ & \langle min + (max - min)/4, max - (max - min)/4 \rangle \\ & \langle min + (max - min)/2, max \rangle \end{aligned} \quad (9.7)$$

Použitie prekrývajúcich sa intervalov má za cieľ odstrániť riziko posunu na nesprávnu stranu intervalu, ak optimum leží blízko stredu daného intervalu.

Po výskyte tejto udalosti sa hodnoty vo všetkých jedincoch prekódujú tak, aby boli z nového zúženého podintervalu (napr. ak hodnota bola z daného podintervalu, tak sa zachová; ak bola mimo neho, tak je generovaná náhodná hodnota z daného podintervalu alebo sa vhodne upraví prvý alebo druhý bit).

Ak je použitý Grayov kód, tak všetky jedince po prekódovaní majú na určitej pozícii hodnotu, ktorá identifikuje daný podinterval: prvý a tretí podinterval je charakterizovaný hodnotou 0 alebo 1 na prvej pozícii, druhý podinterval je charakterizovaný hodnotou 1 na druhej pozícii. Keďže túto pozíciu už nie je potrebné reprezentovať, pretože všetky jedince majú na nej rovnakú hodnotu, tak pozícia a hodnota sa odpamätá, reprezentované hodnoty sa posunú o jednu pozíciu doľava a hodnota na poslednej uvoľnenej pozícii sa doplní náhodne. Takýmto spôsobom dôjde virtuálne k zväčšeniu segmentu pre kódovanie reálneho čísla o jeden bit a teda rozlišovacia presnosť kódovania sa dvojnásobne zväčší.

### 9.3 Samoadaptívne nastavovanie

Aj pri tomto spôsobe sa algoritmus snaží nájsť hodnoty parametrov, ktoré sú najlepšie pre danú situáciu. Na rozdiel od adaptívneho prístupu však k tomu nepoužíva žiadny dodatočný mechanizmus – vystačí si s tými prostriedkami, ktoré má inherentne k dispozícii.

Parametre, ktoré je potrebné nastaviť, sa stávajú súčasťou reprezentácie jedincov. Jedna časť štruktúry jedinca potom kóduje kandidáta na riešenie úlohy, zatiaľ čo druhá časť reprezentuje hľadané hodnoty parametrov. Pre určovanie vhodnosti jedincov sa využíva iba prvá časť kódu (kódujúca riešenie), druhá časť (hodnoty parametrov) slúži pre ovplyvňovanie činnosti algoritmu.

Takýmto spôsobom dochádza k rozšíreniu priestoru prehľadávania, keď niektoré súradné osi reprezentujú atribúty hľadaného riešenia, zatiaľ čo iné osi reprezentujú parametre, ktorých hodnoty je potrebné vhodným spôsobom nastaviť. Algoritmus prehľadávaním priestoru prehľadávania nehľadá iba riešenie úlohy ale súčasne aj hodnoty parametrov, umožňujúce mu toto riešenie nájsť.

V princípe, každý jedinec môže poskytnúť iné hodnoty parametrov. Pokiaľ sa jedná o lokálne parametre, ovplyvňujúce iba jedného jedinca, tak je možné priamo použiť hodnoty dodané nejakým jedincom pre manipuláciu s ním. Príkladom lokálneho použitia samoadaptívne nastavovaných parametrov je mutácia reálnych čísel založená na smerodajných odchýlkach a rotačných uhloch. Normálna mutácia mení hodnotu neuniformným spôsobom (čím je zmena menšia, tým je jej pravdepodobnosť väčšia), pričom pravdepodobnosť nejakej zmeny  $\Delta$  je daná vzťahom

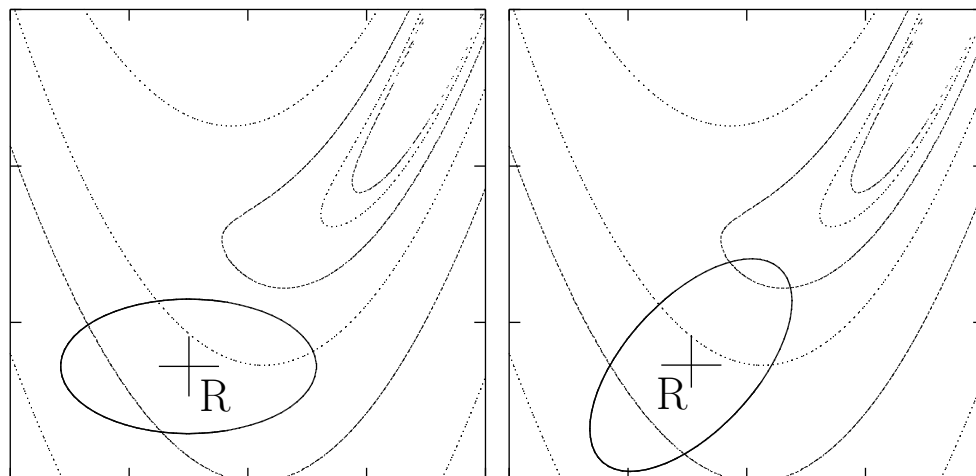
$$\frac{1}{\sqrt{2\pi}\sigma^2} e^{-\Delta^2/(2\sigma^2)} \quad (9.8)$$

čo je hustota pravdepodobnosti normálneho rozdelenia s nulovou strednou hodnotou a nenulovou smerodajnou odchýlkou.

Pri mutácii každá hodnota jedinca môže byť zmenená, pričom pri určovaní zmeny  $\Delta_i$  upravujúcej hodnotu na  $i$ -tej pozícii je možné použiť osobitnú hodnotu  $\sigma_i$ . Každá pozícia môže byť teda mutovaná s inou hodnotou smerodajnej odchýlky.

Všetky tie body v priestore, ktoré môžu vzniknúť z nejakého rodiča s tou istou hodnotou pravdepodobnosti, vytvárajú mutačný elipsoid. Situácia pre prípad dvoch premenných je znázornená na obr. 9.1 vľavo.

Obrázok ilustruje situáciu rôznych smerodajných odchýlok  $\sigma_1 \neq \sigma_2$ . V prípade rovnosti by mal elipsoid kruhový tvar, rôznosť smerodajných od-



Obr. 9.1: Umiestnenie mutačného elipsoidu v rovine bez rotácie (vľavo) a s rotáciou (vpravo)

chýlok však tento kruhový tvar deformuje na tvar eliptický. Vďaka rôznym hodnotám smerodajných odchýlok je maximálna zmena v jednom smere iná než v smere inom (pri tej istej pravdepodobnosti celkovej zmeny) – dochádza k preferencii jedného smeru voči smeru druhému. Rodič  $R$ , vytvárajúci stred elipsoidu, má šancu sa posúvať v jednom smere rýchlejšie než v smere inom pri tej istej pravdepodobnosti.

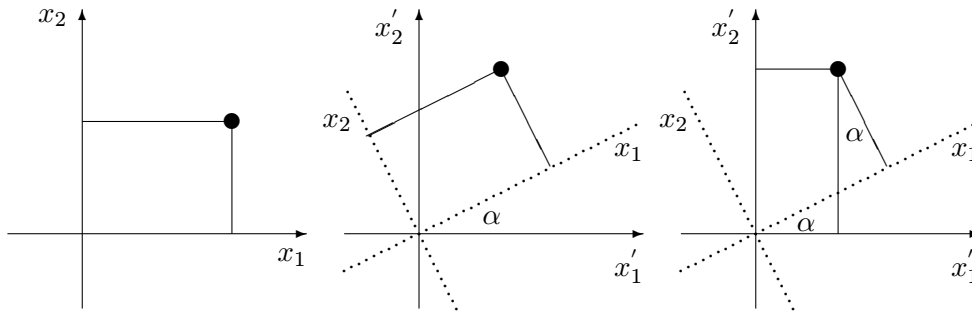
Osi elipsoidu sú rovnobežné so súradnými osami, a teda preferovaný smer je vždy smer rovnobežný s nejakou osou a kolmý na ostatné osi. Toto však nie je vždy najvhodnejšie, ako vidno z obrázku, na ktorom znázornené vrstevnice reprezentujú tie body priestoru, ktorým zodpovedá rovnaká hodnota vhodnosti. Z rozloženia týchto vrstevníc je zrejmé, že najzaujímavejšia časť priestoru je v pravom hornom rohu, kde ležia body s najlepšou vhodnosťou.

Aktuálna voľba smerodajných odchýlok preferuje vodorovný smer. Pohyb vľavo však vedie k okamžitému zhoršeniu, zatiaľ čo pohyb vpravo pri malom kroku prinesie malé zlepšenie, avšak pri väčšom kroku vedie opäť k zhoršeniu. Lepšia situácia by bola v prípade, ilustrovanom pravou časťou obr. 9.1. V tomto prípade je orientácia elipsoidu taká, že preferovaný smer je priamym smerom k hľadaným riešeniam a rodič má šancu rýchleho priblíženia sa k nim.

Aby bol umožnený aj takýto prípad, mutačný elipsoid nemôže byť orientovaný iba v smere osí, ale musí mu byť umožnená ľubovoľná orientácia. Je to



možné zavedením rotácie priestoru – mutačný elipsoid naďalej bude orientovaný v smere osí ale tieto budú v priestore pootočené, čím sa dosiahne vhodné natočenie elipsoidu. Celý prístup je ilustrovaný na obr. 9.2.



Obr. 9.2: Pootočenie súradných osí

Ľavá časť znázorňuje pôvodný priestor s pôvodnou polohou súradných osí. Stredná časť ilustruje pootočenie priestoru o rotačný uhol  $\alpha$ , ktorý môže byť z intervalu  $\langle -\pi, \pi \rangle$ . Pôvodné osi  $x_1$  a  $x_2$  zmenili svoju polohu spolu s polohou všetkých bodov v priestore – poloha týchto bodov voči pôvodným súradným osiam ostáva zachovaná, nemení sa. Ak však chceme vyjadriť polohu z pohľadu súradných osí  $x'_1$  a  $x'_2$ , tak evidentne došlo k zmene hodnôt súradníc každého bodu okrem počiatku, čo je ilustrované v pravej časti obrázku. Transformácia hodnôt súradníc z priestoru  $x_1 - x_2$  do priestoru  $x'_1 - x'_2$  je podľa pravej časti obrázku vyjadrená vzťahmi

$$\begin{aligned} x'_1 &= x_1 \cos(\alpha) - x_2 \sin(\alpha) \\ x'_2 &= x_1 \sin(\alpha) + x_2 \cos(\alpha) \end{aligned} \quad (9.9)$$

Pri viacerých premenných by bolo možné robiť rotácie postupne v jednotlivých rovinách [24]. Napríklad najprv by sa vykonala rotácia v rovine  $x_1 - x_2$  o rotačný uhol  $\alpha_{1,2}$ , pričom  $x_1$  by sa zmenilo na  $x'_1$  a  $x_2$  zase na  $x'_2$ , zatiaľ čo  $x_3$  by sa nezmenilo. Potom by sa vykonala rotácia v rovine  $x'_1 - x_3$  o rotačný uhol  $\alpha_{1,3}$ , pričom  $x'_1$  by sa zmenilo na  $x''_1$  a  $x_3$  na  $x'_3$ . Bez zmeny by teraz ostalo zachované  $x'_2$ . A nakoniec by sa vykonala rotácia v rovine  $x'_2 - x'_3$  o uhol  $\alpha_{2,3}$  s výslednou zmenou  $x'_2$  na  $x''_2$  a  $x'_3$  na  $x''_3$ . Pri viacerých premenných by sa postupovalo obdobným spôsobom. Pre  $n$  premenných by sa muselo vykonať  $n(n-1)/2$  rotácií, každá z nich by zmenila práve dve premenné.

Každá z rotácií môže byť vyjadrená rotačnou maticou [4]. Rotačná matica je štvorcovou maticou s rozmerom daným počtom premenných. Matica  $R_{i,j}$  bude reprezentovať rotáciu v rovine danej osami  $x_i$  a  $x_j$  o rotačný uhol  $\alpha_{i,j}$ . Je to jednotková matica (na hlavnej diagonále sú jednotky, mimo nej nuly) s výnimkou štyroch prvkov:  $r_{i,i} = r_{j,j} = \cos(\alpha_{i,j})$ ,  $r_{i,j} = -\sin(\alpha_{i,j})$ , a  $r_{j,i} = \sin(\alpha_{i,j})$ .

Pri použití týchto matíc sa najprv vygeneruje vektor zmien  $(\Delta_1, \dots, \Delta_n)$  podľa vzťahu (9.8) použitím smerodajných odchýlok  $(\sigma_1, \dots, \sigma_n)$ . Potom sa realizuje rotácia podľa rotačných uhlov použitím príslušných rotačných matíc podľa

$$\left( \prod_{i=1}^{n-1} \prod_{j=i+1}^n R_{i,j} \right) (\Delta_1, \dots, \Delta_n)^T \quad (9.10)$$

a nakoniec sa zmeny upravené rotáciou zložia s hodnotami rodičovského jedinca. Pre mutovanie jedinca s tromi pozíciami by teda bolo potrebné určiť zmeny jednotlivých hodnôt podľa

$$\begin{array}{c} \left| \begin{array}{l} \Delta_1'' \\ \Delta_2'' \\ \Delta_3'' \end{array} \right| = \left| \begin{array}{ccc|ccc} \cos(\alpha_{1,2}) & -\sin(\alpha_{1,2}) & 0 & \cos(\alpha_{1,3}) & 0 & -\sin(\alpha_{1,3}) \\ \sin(\alpha_{1,2}) & \cos(\alpha_{1,2}) & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & \sin(\alpha_{1,3}) & 0 & \cos(\alpha_{1,3}) \end{array} \right| \\ \left| \begin{array}{ccc|l} 1 & 0 & 0 & \Delta_1 \\ 0 & \cos(\alpha_{2,3}) & -\sin(\alpha_{2,3}) & \Delta_2 \\ 0 & \sin(\alpha_{2,3}) & \cos(\alpha_{2,3}) & \Delta_3 \end{array} \right| \end{array} \quad (9.11)$$

Vykonanie mutácie jedinca, reprezentovaného ako vektor  $n$  reálnych čísel, vyžaduje pre svoju realizáciu  $n$  smerodajných odchýlok  $(\sigma_1, \dots, \sigma_n)$  a  $n(n-1)/2$  rotačných uhlov  $\alpha_{1,2}, \dots, \alpha_{n-1,n}$ . Aj keď jedna sada smerodajných odchýlok by mohla byť použitá pre mutovanie všetkých jedincov, rotačné uhly musia byť dané osobitne pre každého jedinca, pretože vhodná veľkosť rotácie závisí na aktuálnej polohe jedinca voči hľadanému riešeniu.

Toto predstavuje veľké množstvo parametrov, pričom niektoré z nich vlastne ani nie je možné nastaviť vopred. Preto určenie týchto parametrov je vhodným kandidátom pre samoadaptačné nastavovanie. Jedinca tak bude pozostávať až z  $n(n+3)/2$  reálnych čísel<sup>2</sup>.

Jedinca však môžu poskytnúť aj hodnoty pre parametre, ktoré sú globálne, ovplyvňujúce viacerých jedincov resp. celú populáciu naraz. V tomto

---

<sup>2</sup>Je možné znižovať počet parametrov použitím jednej smerodajnej odchýlky pre viac hodnôt a/alebo jedného rotačného uhla pre rotáciu vo viacerých rovinách.

prípade z lokálnych hodnôt, poskytnutých rôznymi jedincami, je potrebné určiť hodnoty globálne platné na úrovni celej populácie. Príkladom globálneho použitia samoadaptívne nastavovaných parametrov je spôsob odvodzovania použiteľných hodnôt podľa [10].

Podľa tohto prístupu každý jedinec reprezentuje daný parameter ako reálne číslo z intervalu  $(0, 1)$  umiestnené na pozícii  $j$  vo svojom tele. Toto číslo je mutované podľa vzťahu

$$h'_{ij} = \left( 1 + \frac{1 - h_{ij}}{h_{ij}} e^{-\alpha N(0,1)} \right)^{-1} \quad (9.12)$$

kde  $h_{ij}$  reprezentuje hodnotu parametra podľa  $i$ -teho jedinca a  $\alpha$  riadi rýchlosť adaptačných zmien<sup>3</sup>. Tento mechanizmus zabezpečuje, že mutovaná hodnota naďalej ostáva vo vymedzenom intervale, že menšie zmeny sú pravdepodobnejšie ako veľké, a že modifikácia nejakým faktorom  $\xi$  je rovnako pravdepodobná ako modifikácia recipročným faktorom  $1/\xi$ .

Výsledná hodnota parametra je určená ako súčet lokálnych hodnôt, ktorý je následne vhodne upravený. Príkladom takejto úpravy môže byť transformácia na nejaký interval  $(y_1, y_2)$ . Hodnota z tohto intervalu môže byť chápaná ako absolútna (a priamo vyjadrovať hľadanú hodnotu parametra). Alebo môže byť  $y_1 < 1 < y_2$  a hodnota bude chápaná ako relatívna (a vyjadrovať pomer, v akom sa má parameter zmeniť – pri hodnote väčšej ako 1 hodnota parametra rastie a pri hodnote menšej ako 1 sa zase znižuje). Iným príkladom úpravy je diskretizácia v prípadoch, keď ako hodnota parametra sa očakáva prirodzené číslo.

## 9.4 “No free lunch” teoréma

Každý prehľadávací algoritmus vzorkuje funkciu vhodnosti a zo známych bodov, ktoré už navštívil, extrapoluje<sup>4</sup> nové body ako kandidátov na ďalšie skúmanie. Za nejakú dobu takto preskúma  $n$  rôznych bodov, ktoré vytvárajú histogram vhodností  $H$ , ktorý reprezentuje závislosť medzi nájdenými hodnotami vhodnosti a ich početnosťou v rámci skúmaných  $n$  bodov. Tento histogram môže byť použitý na vyhodnotenie kvality prehľadávania (napr. ak sa hľadá maximum funkcie, tak sa bude z histogramu uvažovať iba bod zodpovedajúci najvyššej hodnote vhodnosti).

<sup>3</sup>V citovanom zdroji bola použitá hodnota 0.22.

<sup>4</sup>Keďže použitá extrapolácia môže byť deterministická alebo stochastická, nasledujúca teoréma bude platná aj pre evolučné algoritmy.

Otázkou je, s akou pravdepodobnosťou bude nájdený nejaký konkrétny histogram vhodnosti. Túto pravdepodobnosť je možné vyjadriť ako podmienenú pravdepodobnosť  $p(H|\Phi, n, \zeta)$ , závisiacu na použitej funkcii vhodnosti  $\Phi$ , počte rôznych vyhodnotených bodov  $n$  a použitom prehľadávacom algoritme  $\zeta$ .

Ak sa porovnávajú dva algoritmy, je možné porovnať histogramy vyprodukované týmito algoritmi. Alebo ešte lepšie, porovnať pravdepodobnosti toho, že vyprodukujú nejaký žiadaný histogram (napr. taký ktorý bude obsahovať hľadaný globálny extrém). Pre tieto účely môže slúžiť nasledujúca “No free lunch” teoréma [47].

**Teoréma:** Pre ľubovoľný pár algoritmov  $\zeta_1$  a  $\zeta_2$  platí

$$\sum_{\Phi} p(H|\Phi, n, \zeta_1) = \sum_{\Phi} p(H|\Phi, n, \zeta_2) \quad (9.13)$$

kde uvedené sumy reprezentujú sumáciu pravdepodobností cez všetky možné funkcie vhodnosti. Toto platí nielen pre samotný histogram, ale aj pre ľubovoľnú mieru založenú na histograme.

Inak povedané, pre dva rôzne algoritmy<sup>5</sup> platí, že ak by výkonnosť jedného z nich bola lepšia na nejakej množine funkcií vhodnosti, tak druhý musí byť lepší na množine zvyšných funkcií vhodnosti. O žiadnom algoritme sa nedá povedať, že je vo všeobecnosti lepší ako nejaký iný algoritmus – s ohľadom na všetky možné funkcie vhodnosti sú všetky algoritmy rovnako výkonné. Pri spriemernení cez všetky funkcie vhodnosti sú všetky algoritmy rovnaké, sú ekvivalentné.

Keďže jedným z prehľadávacích algoritmov je aj náhodné prehľadávanie, tak pre každý algoritmus existuje aspoň jedna funkcia vhodnosti, na ktorej je jeho výkonnosť horšia ako čisto náhodné hľadanie.

Bezprostredným dôsledkom tejto teorémy je rovnosť

$$p(H|n, \zeta_1) = p(H|n, \zeta_2) \quad (9.14)$$

indikujúca, že ľubovoľná miera výkonnosti algoritmov, odvodená z histogramu produkovaného algoritmom, je v prípade, že nie je nič známe o funkcii vhodnosti  $\Phi$ , nezávislá na použitom algoritme. Toto je možné odvodiť

---

<sup>5</sup>Za rôzne algoritmy sa považujú aj inštancie toho istého algoritmu, líšiace sa v hodnote aspoň jedného parametra. Rôzne nastavenie parametrov toho istého algoritmu takto vlastne produkuje rôzne algoritmy.

jednoduchým spôsobom:

$$p(H|n, \zeta_1) = \sum_{\Phi} p(H|\Phi, n, \zeta_1)p(\Phi|n, \zeta_1) = \sum_{\Phi} p(H|\Phi, n, \zeta_1)p(\Phi) = \dots \quad (9.15)$$

keďže funkcia vhodnosti nie je závislá od použitého algoritmu a počtu vyhodnotených bodov. Ak nevieme nič o  $\Phi$ , potom všetky funkcie vhodnosti sú rovnako pravdepodobné a teda možno pokračovať

$$\dots = p(\Phi) \sum_{\Phi} p(H|\Phi, n, \zeta_1) = p(\Phi) \sum_{\Phi} p(H|\Phi, n, \zeta_2) = p(H|n, \zeta_2) \quad (9.16)$$

Iným dôsledkom je rovnosť

$$E(H|n, \zeta_1) = E(H|n, \zeta_2) \quad (9.17)$$

indikujúca, že očakávaný histogram (daný ako stredná hodnota všetkých histogramov) získaný vyhodnotením  $n$  rôznych bodov je rovnaký pre všetky algoritmy (opäť v prípade, že nie je nič známe o funkcii vhodnosti). Toto je možné odvodiť nasledovne

$$E(H|n, \zeta_1) = \sum_H H p(H|n, \zeta_1) = \dots \quad (9.18)$$

čo možno s použitím predchádzajúceho dôsledku transformovať na

$$\dots = \sum_H H p(H|n, \zeta_2) = E(H|n, \zeta_2) \quad (9.19)$$

Oba dôsledky hovoria, že ak pri výbere (nastavení) algoritmu pre prehľadávanie nejakej konkrétnej funkcie vhodnosti znalosti o tejto funkcii budú ignorované, potom nie je nijaká záruka, že zvolená verzia algoritmu bude danú plochu vhodnosti prehľadávať efektívnym spôsobom – je to jednoducho spoliehanie sa na šťastnú zhodu vlastností použitého algoritmu a riešeného problému.

Neexistuje žiadny “free lunch” pre efektívne prehľadávanie. Každý algoritmus prehľadáva nejakú plochu vhodnosti iba tak dobre, do akej miery boli doň vložené znalosti o danej funkcii vhodnosti. V podstate sú dve možnosti, ako dosiahnuť, aby bol vybratý taký algoritmus pre nejakú funkciu vhodnosti, ktorého výkon na danej funkcii je nadpriemerný:

- skonštruovať algoritmus špeciálne cieleň pre daný problém, zohľadňujúci charakteristiky daného problému,

- funkciu vhodnosti  $\Phi$  nahraď distribúciou  $p(\Phi)$  a použiť algoritmus, o ktorom je známe, že je vhodný pre danú distribúciu.

Prvá možnosť by vlastne znamenala pretransformovať slabú metódu (všeobecný evolučný algoritmus) na metódu silnú (evolučný algoritmus špecializovaný na určitú doménu úloh).

V praxi je schodnejšou druhá možnosť – na základe známych vlastností funkcie vhodnosti zúžiť množinu potenciálnych funkcií. Už to nebude ľubovoľná funkcia vhodnosti s uniformnou pravdepodobnosťou. Čím bude toto zúženie väčšie, tým bude daná distribúcia lepšou náhradou za konkrétnu funkciu.

## Kapitola 10

# Udržiavanie rôznorodosti populácie

Počas činnosti evolučného algoritmu sa populácia jedincov vyvíja počas jednotlivých generácií. Jedince reprezentujúce kandidátov riešenia sa presúvajú do sľubnejších častí priestoru (charakterizovaných lepšou vhodnosťou) a postupne sa sústreďujú v nich. Výsledkom je snaha jedincov okupovať iba jeden podpriestor tvoriaci iba malú časť celkového priestoru.

Jedna z predstáv o činnosti evolučného algoritmu hovorí, že algoritmus kombinuje vhodné stavebné bloky do väčších celkov až po vytvorenie výsledného riešenia. Teda vhodné stavebné bloky začínajú v populácii prevládať, zatiaľ čo menej vhodné z nej začínajú miznúť. Schopnosť evolučného algoritmu hľadať lepšie riešenia je postupne znižovaná následkom nedostatku stavebného materiálu z ktorého môžu byť nové jedince budované.

Takáto situácia sa označuje ako konvergencia populácie – pôvodná rôznorodosť populácie sa znižuje tým viac ako populácia konverguje. Podľa predchádzajúceho je zrejmé, že navonok sa konvergencia prejavuje ako konvergencia priestorová a štrukturálna:

**Priestorová konvergencia** – zmenšenie vzdialeností medzi jedincami, nakopenie jedincov v malom podpriestore (redukcia veľkosti skúmaného podpriestoru),

**Štrukturálna konvergencia** – zníženie rôznorodosti materiálu z ktorého sú vytvárané nové jedince (redukcia variability dostupných reprezentatívnych elementov).

Oba typy konvergenzie sú navzájom závislé – objavenie sa jedného typu môže mať za následok súčasný výskyt druhého typu konvergenzie. V prípade,

ak dekódovanie podobných reprezentácií vedie na podobných kandidátov riešenia, tak sa konvergencia súčasne prejaví v priestore prehľadávania aj v priestore kandidátov – zúženie oblasti prehľadávania v jednom z nich na iba malý podpriestor sa prejaví podobným zúžením pozornosti aj v druhom priestore.

Konvergencia je neoddeliteľne spätá s evolučným algoritmom a je celkom užitočná vlastnosť umožňujúca populácii skonvergovať do úzkeho okolia hľadaného riešenia respektíve umožňujúca vyselektovať iba tie stavebné bloky, z ktorých je potrebné skonštruovať hľadané riešenie. Problémom však je v situácii, keď sa konvergencia vyskytne skôr ako je algoritmus schopný lokalizovať hľadané riešenie a populácia následkom toho skonverguje k nejakému lokálnemu extrému, z ktorého vďaka nedostatku stavebného materiálu nedokáže uniknúť. Takáto forma konvergenzie sa označuje ako predčasná.

Keďže populácia začína konvergovať hneď po štarte algoritmu, je vhodnejšie hovoriť o stupni resp. miere konvergenzie ako o tom či sa konvergencia vyskytla alebo nie. Tento stupeň môže byť meraný viacerými mierami. Príkladom miery štrukturálnej konvergenzie pre binárnu kódovaciu schému je:

$$\frac{1}{l\mu} \sum_{j=1}^l \max \left\{ \sum_{i=1}^{\mu} (1 - h_{ij}), \sum_{i=1}^{\mu} h_{ij} \right\} \quad (10.1)$$

kde  $\mu$  je veľkosť populácie,  $l$  je dĺžka reprezentácie a  $h_{ij}$  je hodnota na  $j$ -tej pozícii  $i$ -teho jedinca [4]. Miera nadobúda hodnoty z intervalu  $< 0.5, 1.0 >$  a násobená hodnotou 100 vyjadruje priemerné percento najprominentnejšej hodnoty na každej pozícii vo všetkých jedincoch. Hodnota 0.5 reprezentuje rovnomerné zastúpenie každej hodnoty na každej pozícii v populácii, hodnota 1.0 predstavuje situáciu, keď pre každú pozíciu je prístupná iba jedna alternatíva.

Iným príkladom vhodnej miery je entropia nadobúdajúca hodnoty z intervalu  $< 0.0, 1.0 >$ , kde rovnomerné zastúpenie všetkých hodnôt je reprezentované hodnotou 1.0 zatiaľ čo výskyt iba jednej hodnoty na každej pozícii je signalizovaný hodnotou 0.0. Hodnotu možno určiť podľa vzťahu

$$-\frac{1}{l} \sum_{j=1}^l \left( \frac{\sum_{i=1}^{\mu} (1 - h_{ij})}{\mu} \log_2 \frac{\sum_{i=1}^{\mu} (1 - h_{ij})}{\mu} + \frac{\sum_{i=1}^{\mu} h_{ij}}{\mu} \log_2 \frac{\sum_{i=1}^{\mu} h_{ij}}{\mu} \right) \quad (10.2)$$

vyjadrujúceho priemernú entropiu jednej pozície použitej binárnej kódovacej schémy.

Priestorová konvergencia je meraná pomocou vzdialeností. Na vzdialenosti môže byť založená priemerná variancia populácie (indikujúca hodno-



tou 0.0 konvergenciu materiálu pre každú pozíciu iba na jednu alternatívu) určená vzťahom

$$\frac{1}{\mu} \sum_{i=1}^{\mu} \sum_{j=1}^l \left| h_{ij} - \frac{\sum_{k=1}^{\mu} h_{kj}}{\mu} \right| \quad (10.3)$$

alebo ako priemerná vzdialenosť medzi jedincami v populácii určovaná podľa vzťahu

$$\frac{1}{\mu(\mu-1)/2} \sum_{i=1}^{\mu-1} \sum_{j=i+1}^{\mu} \sum_{k=1}^l |h_{ik} - h_{jk}| \quad (10.4)$$

Stupeň konvergenzie v priestore kandidátov môže byť tiež založený na vzdialenosti medzi jednotlivými kandidátmi riešenia. Jediným rozdielom je použitie Euklidovej vzdialenosti<sup>1</sup> namiesto Hammingovej<sup>2</sup> pre výpočet vzdialenosti dvoch bodov v priestore. Inou možnosťou je veľkosť podpriestoru obsahujúceho kandidátov. Avšak s kvantitatívnym vyjadrením konvergenzie v priestore prehľadávania sa možno stretnúť oveľa častejšie.

Všetky uvedené miery sú ilustrované na obr. 10.1, pričom je sledovaný vývoj konvergenzie počas určitého počtu generácií. Obrázok ilustruje typický časový priebeh konvergenzie - na začiatku rýchla strata variabilnosti stavbných elementov spôsobujúca priblíženie jedincov navzájom, nasledovaná obdobím stagnácie alebo iba pomalého zvyšovania stupňa konvergenzie.

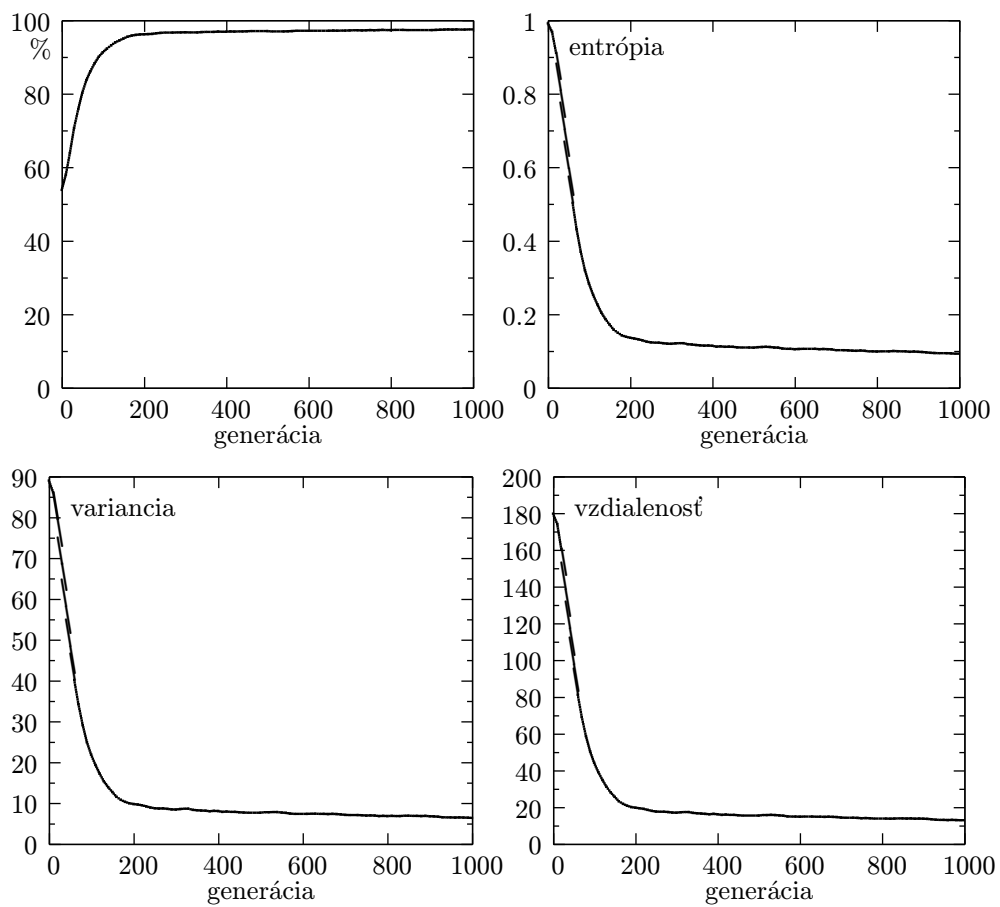
Pre ovplyvňovanie konvergenzie je potrebné poznať hlavné faktory konvergenzie. Za tieto sa považujú:

- selekčný tlak – daný selekčnou metódou. Čím viac sú preferované nejaké jedince voči iným jedincom (napr. nadpriemerné voči podpriemerným), tým je rýchlosť konvergenzie väčšia.
- selekčný šum – daný genetickým driftom použitej vzorkovacej metódy (výskyt driftu spôsobuje konvergenciu bez toho aby bola opodstatnená vhodnosťou jedincov v populácii).
- operátorová deštrukcia – genetické operátory majú deštrukčný vplyv na vytvorených jedincov, pretože dokážu meniť elementy z ktorých sú tieto vytvorené (čím pôsobia proti konvergencii).

Keďže k dispozícii sú selekčné metódy ktoré netrpia genetickým driftom a operátory pôsobia skôr proti konvergencii, hlavným dôvodom, v praxi neodstrániteľným, je selekčný tlak. Ten je od evolučného algoritmu neoddeliteľný

<sup>1</sup>Druhá odmocnina súčtu druhých mocnín rozdielov zodpovedajúcich zložiek.

<sup>2</sup>Súčet absolútnych hodnôt rozdielov zodpovedajúcich zložiek.



Obr. 10.1: Ukážka konvergenzie meranej ako percento (vľavo hore), entropia (vpravo hore), variancia (vľavo dole) a vzdialenosť (vpravo dole).

– a tak evolučný algoritmus je vlastne o hľadani rovnováhy medzi selekčným tlakom a rýchlosťou konvergencie. V ideálnom prípade selekčný tlak by mal byť minimálne tak veľký, aby hľadané riešenie bolo nájdené vo vyhovujúcom čase, a súčasne menší ako tlak, ktorý by už spôsobil predčasnú konvergenciu hľadania v lokálnom extréme.

Skúsenosti ukazujú, že intuitívne riešenie – zväčšenie veľkosti populácie – vo väčšine prípadov nefunguje. Je pravda, že pri väčšej populácii je k dispozícii viac materiálu a teda trvá dlhšie kým nastane konvergencia populácie. Avšak rozdiel je prekvapivo malý a preto neprináša želaný efekt.

Preto je vhodnejšie použiť pri zostavení evolučného algoritmu prvky pôsobiace proti faktorom konvergencie. V prípade, že aj tak dochádza k predčasej konvergencii, je nutné použiť niektorú z modifikácií špeciálne určených pre udržiavanie rôznorodosti populácie.

## 10.1 Infúzia materiálu

Myšlienkou tejto triedy metód je priame vkladanie tých stavebných prvkov do populácie, ktoré v nich chýbajú. Vďaka tomuto prístupu sú do populácie neustále pridávané tie prvky, ktoré z nej konvergencia vytláča. Nedochádza k absencii žiadnych hodnôt, ktoré by mohli byť použité pre tvorbu nových jedincov. Následkom toho je variabilita stavených elementov dostatočná na to, aby algoritmus mohol tvoriť jedince ľubovoľných tvarov.

Pri príprave nového materiálu, ktorý má byť vložený do populácie, je možné použiť dva prístupy:

- štatistický,
- náhodný.

Pri štatistickom prístupe sa skúma aktuálna distribúcia jednotlivých hodnôt (stavebných blokov) v populácii a na základe aktuálneho stavu sa rozhodne, ktoré hodnoty je potrebné do populácie vložiť. Je to možné urobiť deterministickým spôsobom (napríklad sa vloží hodnota, ktorá je momentálne minoritnou) alebo na pravdepodobnostnom základe – určí sa pravdepodobnosť vloženia každej hodnoty a konkrétna hodnota sa vyberie stochastickým spôsobom. Napríklad, ak pri použití binárnej reprezentácie sa zistí, že aktuálne sa v populácii  $\mu$  jedincov na nejakej pozícii vyskytuje  $N_0$  hodnôt 0, tak táto hodnota bude vkladaná do populácie s pravdepodobnosťou

$$\frac{\mu - N_0}{\mu} \tag{10.5}$$

zaručujúcou tým väčšiu šancu pre danú hodnotu, čím je zastúpenie tejto hodnoty v populácii menšie.

Pri druhom spôsobe sa neskúma čo v populácii chýba, ale náhodne sa vyberajú hodnoty ktoré sú do nej vložené. Výsledkom je, že sú vkladané aj hodnoty, ktorých výskyt nebol redukovaný konvergenciou. Na druhej strane odpadá analýza populácie, ktorá si môže vyžadovať značný výpočtový čas.

Príkladmi metód patriacich do tejto triedy sú prístupy založené na eliminácii duplikácií alebo na reinicializácii.

**Eliminácia duplikácií.** Pred tým, ako je nejaký novo vytvorený jedinec vložený do populácie, je tento jedinec porovnávaný s jedincami ktoré sa už nachádzajú v populácii. Ak sa zistí, že v populácii už existuje rovnaký jedinec, tak nový jedinec sa do populácie nevkladá. Buď sa priamo zahodí alebo sa urobí pokus zmutovať tohto jedinca tak, aby sa líšil od jedincov v populácii. Táto základná schéma môže byť modifikovaná niekoľkými spôsobmi:

- jedinec sa neporovnáva voči všetkým jedincom v populácii, ale iba voči menšej náhodne vybratej vzorke z populácie čo umožní znížiť výpočtové nároky na úkor presnosti,
- aby bol jedinec vložený do populácie, tak vzdialenosť<sup>3</sup> medzi ním a každým z ostatných jedincov musí byť väčšia ako zadaný prah. Tento môže byť fixne daný alebo sa postupne môže zmenšovať (napr. ak určitú dobu sa nepodarilo zaradiť do populácie žiadneho nového jedinca), čo umožňuje jemnejšie preskúmavanie okolí extrémov v neskorších fázach hľadania.
- pokusov o mutáciu nového jedinca možno vykonať v prípade opakovaného neúspechu viac. Ak ani po vyčerpaní všetkých možností sa nepodarilo jedinca upraviť vhodným spôsobom, tak je buď zahodený alebo vložený do populácie napriek nesplneniu podmienky minimálnej vzdialenosti.

**Reinicializácia.** V určitých okamihoch sa populácia rozdelí na dve časti – prežívajúcu a reinicializovanú. Prežívajúca časť (zvyčajne to sú jedince s najlepšou vhodnosťou) ostáva nezmenená. Jedince v reinicializovanej časti sa masívne menia s cieľom zabezpečiť dostatočnú variabilitu reprezentačných

---

<sup>3</sup>Zvyčajne sa používa Hammingova vzdialenosť.

elementov. Pre počty jedincov v oboch častiach platia vzťahy

$$\begin{aligned} 0 &\leq N_P < \mu \\ 1 &\leq N_R \leq \mu \end{aligned} \tag{10.6}$$

kde  $N_P$  označuje veľkosť prežívajúcej časti a  $N_R$  zase veľkosť reinitializovanej časti.

Samotná zmena obsahu reinitializovanej časti sa môže diať viacerými spôsobmi:

- náhodné vygenerovanie nových jedincov (bez ohľadu alebo s ohľadom na distribúciu hodnôt v prežívajúcej časti),
- hypermutácia (mutácia s vysokou pravdepodobnosťou mutácie, zvyčajne  $p_m \sim 0.5$ ) jedincov z reinitializovanej časti,
- hypermutácia jedincov z prežívajúcej časti, ktoré slúžia ako vzory. Tieto vzory sú modifikované pre získanie dostatočnej variability.

Podobne určenie okamihu vhodného pre reinitializáciu časti populácie môže byť založené na niekoľkých spôsoboch:

- periodická reinitializácia s pevnou periódou (čím je perióda menšia, tým je postačujúce reinitializovať menšiu časť populácie),
- až populácia úplne skonverguje (v tomto prípade je potrebné reinitializovať veľkú časť populácie, často všetky jedince alebo všetkých okrem najlepšieho jedinca),
- ak stupeň konvergenzie presiahne vopred definovaný prah, ktorý môže byť fixný alebo sa môže meniť v závislosti na počte vygenerovaných generácií,
- ak dlho nedošlo k zlepšeniu najlepšej vhodnosti (založené na predpoklade že príčinou je prílišné skonvergovanie populácie).

Rozhodnutie o vykonaní reinitializácie nemusí byť deterministické ale môže byť aj stochastické, keď uvedené faktory ovplyvňujú pravdepodobnosť reinitializácie (napr. doba od poslednej reinitializácie, stupeň konvergenzie) [28].

Algoritmus využívajúci reinitializáciu môže pracovať aj s menšou populáciou (tzv. mikro algoritmy používali  $\mu = 5$ ) a nemusí využívať operátor mutácie.

## 10.2 Udržiavanie subpopulácií

Každá zo subpopulácií je považovaná za nositeľku/reprezentantku nejakého stavebného materiálu. Kým je subpopulácia prítomná v celkovej populácii, aj daný stavebný materiál je k dispozícii. Preto cieľom tejto skupiny metód je udržiavať v rámci populácie dostatočný počet subpopulácií.

Jednou z možností ako reprezentovať príslušnosť jedinca k subpopulácii je použitie  $k$ -bitového príznaku. Všetky jedince, ktoré majú rovnakú hodnotu svojho príznaku, tak patria k tej istej subpopulácii. Voľbou hodnoty  $k$  možno voľiť maximálny počet subpopulácií, ktoré môžu byť explicitne reprezentované. Pokiaľ subpopulácie sú uvažované oddelene pri vytváraní novej generácie, ich počet sa v populácii nemení. Ak pri tvorbe novej generácie sa pracuje nad populáciou ako celkom, tak počet subpopulácií sa môže znižovať (prípade keď ani jeden jedinec nejakej subpopulácie nie je prenesený do nasledujúcej generácie). Naopak, keď jedinec môže nadobudnúť príznak odlišný od príznaku jeho rodičov (napr. jeho mutáciou), tak počet subpopulácií sa môže zvyšovať.

Aj keď je možné explicitne definovať subpopulácie, dominantne sa používajú metódy, keď subpopulácie nie sú explicitne identifikované (a tým pádom nie je známy ich počet v rámci celkovej populácie). V tomto prípade nie je možné pracovať priamo so subpopuláciami ale iba používať postupy, ktoré podporujú vytváranie a prítomnosť subpopulácií – avšak bez garantovania, že v konkrétnom prípade skutočne nejaké subpopulácie existujú v aktuálnej populácii.

**Obmedzené spájanie.** V prírode *druh* označuje triedu organizmov schopných páriť sa navzájom s následnou produkciou nových jedincov. Aj keď môže dochádzať taktiež k medzidruhovému spojeniu, to je oveľa menej častým prípadom ako spojenie vnútrodrohové. Z pohľadu genetickej stavby jedincov, oveľa častejšie dochádza k spájaniu málo rozdielneho genetického materiálu ako k spájaniu genetického materiálu s väčšími odlišnosťami (vnútrodrohová genetická stavba jedincov je menej variabilná ako medzidrohová).

Tento princíp môže byť v evolučnom algoritme premietnutý dvojaký spôsobom. Pri výbere rodičov je možné preferovať

- čo najväčšiu podobnosť rodičov (analógia vnútrodrohového kríženia),
- čo najmenšiu podobnosť rodičov (analógia medzidrohového kríženia).

V prvom prípade je veľká šanca, že potomok bude podobný obom rodičom – bude reprezentovať tú istú subpopuláciu. Vplyvom selekčného tlaku

však bude dochádzať k znižovaniu počtu subpopulácií. V druhom prípade pri spojení rodičov z rôznych subpopulácií je šanca, že výsledný potomok bude z inej subpopulácie ako jeho rodičia – môže dochádzať k vytváraniu nových subpopulácií, ktoré dovtedy neboli zastúpené v aktuálnej populácii.

Keďže cieľom nie je udržiavať tie isté subpopulácie v populácii ale skôr udržiavať ich počet na čo najvyššej hodnote, prístup založený na čo najväčšej nepodobnosti rodičov je preferovaný z hľadiska udržiavania rôznorodosti stavebného materiálu v populácii.

Z hľadiska modifikácie základnej podoby evolučného algoritmu je možné princíp obmedzeného spájania realizovať dvojako:

- vykonanie krížiaceho genetického operátora nad selektovanými rodičmi závisí na podobnosti rodičov,
- selekcia rodičov je upravená pre uvažovanie podobnosti selektovaných rodičov.

Príkladom prvého spôsobu je prípad, keď dvaja rodičia sa krížia iba ak ich Hammingova vzdialenosť je nad zadaným prahom. Inak sa krížiaci operátor neaplikuje [11]. Na začiatku je tento prah nastavený na nejakú hodnotu. Ak sa v rámci určitého počtu generácií nevyskytlo žiadne kríženie rodičov, tak je tento prah dekrementovaný. Toto umožňuje zohľadniť presunutie záujmu iba na určitý podpriestor v neskorších fázach hľadania riešenia.

Príkladom druhého spôsobu je, keď jeden rodič je vybraný použitím nejakej štandardnej selekčnej metódy (preferujúcej vhodnejších jedincov). Druhý rodič je vybraný zo skupiny jedincov (vybraných náhodne z populácie alebo pomocou selekčnej metódy preferujúcej vhodnejších jedincov) ako najmenej podobný s prvým rodičom. Inou možnosťou je opakovaný výber druhého rodiča pokiaľ nie je vybraný jedinec s vyhovujúcou vzdialenosťou voči prvému rodičovi alebo kým sa nedosiahne maximálny povolený počet opakovaných výberov.

**Obmedzená súťaž.** Inšpiráciou je ekologický fenomén súťaže o obmedzené zdroje medzi jedincami. Podobné jedince populácie (najčastejšie jedince rovnakého druhu) majú tendenciu okupovať rovnaký priestor a preto musia navzájom súťažiť o dostupné zdroje. Naproti tomu rôzne jedince (najčastejšie patriace k rôznym druhom) majú tendenciu okupovať rôzne priestory a teda navzájom nesúťažia.

Analógiou súťaže o zdroje je v evolučnom algoritme fáza formovania novej generácie - súťaž novo generovaných jedincov s pôvodnými jedincami

populácie [26]. Noví potomkovia nebudú súťažiť so všetkými pôvodnými jedincami ale ich súťaž sa zameriava iba na jedince, ktoré sú s nimi podobné (s úmyslom nahrádzať iba jedincov patriacich do rovnakej subpopulácie ako tieto novo generované jedince).

Jednoduchým zosobnením tohto postupu by bolo pre každého potomka zistiť, ktorý z pôvodných jedincov, tvoriacich populáciu, je mu najviac podobný. S týmto by potom súťažil o svoje zaradenie do populácie. V najjednoduchšom prípade by toho jedinca jednoducho v populácii nahradil, avšak je možný aj deterministický alebo stochastický súboj [29] založený na ich vhodnostiach<sup>4</sup>. Z dôvodu zníženia výpočtovej náročnosti je možné nového jedinca porovnávať iba so vzorkou populácie namiesto s celou populáciou. Túto vzorku je možné vybrať náhodne (s alebo bez prihliadnutia k vhodnosti jedincov) alebo deterministicky (napr. najhoršie jedince).

Ešte väčším zjednodušením je predpoklad, že novo vygenerované jedince sú najpodobnejší svojim rodičom. A tak buď priamo nahradia svojich rodičov v populácii alebo s nimi súťažia v deterministických alebo stochastických súbojoch.

Keďže pri krížení z dvoch rodičov vznikli dvaja potomkovia, existujú dve možnosti organizácie súťaže

rodič 1	—	potomok 1	rodič 1	—	potomok 2
rodič 2	—	potomok 2	rodič 2	—	potomok 1

V prípade väčšieho počtu rodičov a potomkov sa počet možných usporiadaní súťaže zvyšuje. Použije sa to usporiadanie, ktoré minimalizuje priemernú vzdialenosť “rodič-potomok” v rámci súťaže.

### 10.3 Penalizácia

Cieľom je potláčať zhukovanie jedincov s podporou ich rovnomernejšieho pokrytia priestoru. Spočíva na predstave, že jedince, obývajúce nejaký priestor, majú k dispozícii iba obmedzené zdroje tohto priestoru. Čím viac jedincov sa nahromadí v nejakom podpriestore, tým menej zdrojov pripadá na jedného jedinca. Ak koncentrácia jedincov prekročí kapacitu daného podpriestoru, tak výsledkom je nemožnosť plne uspokojovať potreby týchto jedincov a následkom toho dochádza iba k obmedzenému pridelovaniu zdrojov.

Analogicky k tejto predstave úlohu zdrojov hrá vhodnosť. Obmedzené pridelovanie zdrojov je reprezentované situáciou, keď vhodnosť jedincov je

---

<sup>4</sup>Pri deterministickom súboji by vyhral jedinec s lepšou vhodnosťou, pri stochastickom by na vhodnosti jedincov záviseli pravdepodobnosti ich výberu.



znižovaná na menšiu hodnotu než by prislúchalo podľa schopnosti jedincov reprezentovať hľadaných kandidátov riešenej úlohy. Jedinca sú penalizované za zhlukovanie sa v malom podpriestore. Čím dochádza k väčšej koncentrácii jedincov v nejakom podpriestore, tým viac je vhodnosť týchto jedincov penalizovaná.

Ak teda časť populácie skonverguje do malého podpriestoru, jedincom nachádzajúcim sa v tomto podpriestore je ich vhodnosť dočasne znížená (na obdobie realizácie selekcie skupiny rodičov). Vďaka tomu jedinca z tohto podpriestoru majú menšiu šancu stať sa rodičmi a tým sa zároveň znižuje pravdepodobnosť, že aj novo generovaní potomkovia budú patriť to daného podpriestoru. Následkom toho v nasledujúcej generácii možno očakávať menej jedincov patriacich do daného podpriestoru ako v prípade, ak by jedinca neboli penalizované za svoju vzájomnú blízkosť.

Najznámejším reprezentantom tohto prístupu je metóda *zdieľania* [9]. Pri tejto metóde je vhodnosť každého jedinca v populácii premapovaná na novú hodnotu, použitú pri selekcii, podľa vzťahu

$$\Phi'(a_i(t)) = \frac{\Phi(a_i(t))}{\sum_{j=1}^{\mu(t)} sh(d(a_i(t), a_j(t)))} \quad (10.7)$$

kde zdieľacia funkcia  $sh$  je funkciou vzdialenosti medzi dvomi jedincami populácie. Nadobúda hodnotu 1.0 v prípade, že jedinca sú identické (teda zdieľajú ten istý bod priestoru a ich vzdialenosť je minimálna) a hodnotu 0.0 ak vzdialenosť prekročí nejaký zadaný prah. Tento prah určuje veľkosť uvažovaných podpriestorov v ktorých sa uvažuje penalizácia – jedinca nie je ovplyvňovaný tými jedincami od ktorých je vzdialenejší ako tento zadaný prah. Najčastejšie používaná zdieľacia funkcia má tvar

$$sh(d) = \begin{cases} 1 - \left(\frac{d}{\theta}\right)^\alpha, & d \leq \theta \\ 0, & d > \theta \end{cases} \quad (10.8)$$

kde  $\theta$  reprezentuje zadný prah označovaný aj ako polomer zdieľania. Parameter  $\alpha$  je využiteľný k regulovaniu tvaru funkcie, avšak často je použité  $\alpha = 1.0$  aby sa nezvyšoval zbytočne počet parametrov, ktoré je potrebné nastaviť.

Iný spôsob použitia penalizácie je rozdeliť celú populáciu na explicitné zhluky jedincov a pri určovaní penalizácie uvažovať vždy iba jedinca určitého zhluku. Keďže zhluk je explicitne daný, potom penalizácia môže byť variabilnejšia. Môže byť založená na:

- vzdialenosti jedincov (či už navzájom alebo od centra zhluku),

- počte jedincov prislúchajúcich k danému zhluku,
- selekcii, keď iba niekoľko jedincov (zvyčajne tých najlepších) z každého zhluku môže byť vybraných do funkcie rodičov [34].

Taktiež pre určovanie zhlukov je možné použiť rozličné prístupy. Je možné napríklad použiť niektorý z existujúcich zhlukovacích algoritmov. Ilustráciou môže byť adaptívny algoritmus [44], keď sa začína s  $k$  zhlukmi a okrem nich sú dané dva prahy  $\theta_{min}$  a  $\theta_{max}$ . Samotný algoritmus má tvar:

1.  $k$  náhodných jedincov je vybratých ako centrá budúcich zhlukov,
2. všetky ostatné jedince sú zaradené do toho zhluku, k centru ktorého sú najbližšie,
3. pre každý zhluk sa určí nové centrum ako priemer hodnôt jedincov v danom zhluku,
4. ak centrá dvoch zhlukov sú bližšie ako  $\theta_{min}$ , tak tie dva zhluky sa spoja do jedného; ak nejaký jedinec je od najbližšieho centra vzdialený viac ako  $\theta_{max}$ , tak daný jedinec bude tvoriť centrum nového zhluku,
5. ak je splnená podmienka pre ukončenie činnosti (napr. žiadna zmena v definícii zhlukov, uplynutie maximálne povoleného počtu iterácií, atď.), tak algoritmus končí. Inak prechod na bod 2.

Pretože však realizácia takéhoto zhlukovania môže predstavovať veľkú výpočtovú záťaž, je možné používať aj jednoduchšie techniky na určenie zhlukov. Jedna z nich je

1. Populácia jedincov sa zotriedi podľa vhodnosti od lepších k horším hodnotám.
2. Postupne sa prechádza populáciou a definujú sa centrá zhlukov. Ak jedinec je dostatočne blízko centra niektorého zo zhlukov, tak sa prechádza na ďalšieho jedinca. Inak sa daný jedinec stáva centrom nového zhluku.
3. Každý jedinec je zaradený do toho zhluku, k centru ktorého sa nachádza najbližšie.

Tento postup je možné modifikovať tak, že sa vytvára iba maximálne určitý počet zhlukov alebo jedinec je zaradený do zhluku iba ak jeho vzdialenosť od centra daného zhluku je menšia ako nejaký zadaný prah.

## 10.4 Preferencia odlišnosti

V populácii existujú jedince, ktoré sú si navzájom podobné (vzdialenosť medzi nimi je malá) ako aj jedince, ktoré sú navzájom od seba odlišné (vzdialenosť medzi nimi je veľká). Konvergencia populácie znamená, že populácia je tvorená podobnými jedincami, je v nej núdza o jedince odlišné. Preto pre potlačenie konvergenzie je postačujúce zabezpečiť, aby populácia bola vytváraná odlišnými jedincami resp. aby v nej bol dostatočne veľký počet odlišných jedincov.

Prítomnosť odlišných jedincov v populácii je možné podporovať prístupom, keď rozhodnutia, majúce za následok výber jedincov (t.j. selekcia rodičov a formovanie novej generácie), budú brať do úvahy podobnosť a odlišnosť jedincov – keď budú realizované takým spôsobom, aby počet odlišných jedincov ostával v populácii čo najväčší.

Pre výber rodičov to znamená identifikovať odlišných jedincov v populácii a preferovať ich do úlohy rodičov. Keďže však je súčasne potrebné zachovať aj preferenciu vhodnejších jedincov oproti menej vhodnejším pre zachovanie tlaku vyhľadávať čo najlepších jedincov, je potrebné kritérium odlišnosť jedincov kombinovať s vhodnostným kritériom. Je to možné v zásade dvoma spôsobmi:

- sekvenčný výber,
- kombinácia kritérií.

Pri sekvenčnom výbere selekcia pozostáva z dvoch fáz. V každej z týchto fáz je použité iné kritérium. Najprv sa použitím prvého kritéria vyberie skupina kandidátov na rodičov a z tejto skupiny sa následne určí skupina rodičov použitím druhého kritéria.

Druhý spôsob predpokladá selekciu štandardnou selekčnou metódou avšak použitím modifikovanej vhodnosti. Vhodnosť jedincov je modifikovaná údajom o ich podobnosti/odlišnosti voči iným jedincom. Jednou z možností je pre praktické použitie nahradiť podobnosť vzdialenosťou daného jedinca voči zvolenému referenčnému jedincovi [46] (ktorý môže byť jedincom prítomným v aktuálnej populácii ale tak isto môže byť jedincom fiktívnym, napr. s náhodne vybranými hodnotami atribútov alebo hodnotami reprezentujúcimi priemerné hodnoty vyskytujúce sa v populácii). Keďže cieľom referenčného jedinca je iba zabezpečiť rôznorodosť populácie a nie ovplyvňovať smer hľadania, tento jedinec by sa mal dostatočne často meniť.

## 10.5 Porovnanie metód pre udržiavanie rôznorodosti

Pre empirické porovnanie popisovaných metód pre udržiavanie rôznorodosti v populácii jedincov sme použili úlohu numerickej optimalizácie – hľadanie minima multimodálnej funkcie. V úlohe testovacej funkcie bola použitá Ackleyho funkcia [4] – exponenciálna funkcia modulovaná kosínusovou zložkou, reprezentujúca stredne komplikovanú topológiu. Funkcia bola definovaná vzťahom

$$f(\vec{x}) = -20 \exp\left(-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2}\right) - \exp\left(\frac{1}{n} \sum_{i=1}^n \cos(2\pi x_i)\right) + 20 + e \quad (10.9)$$

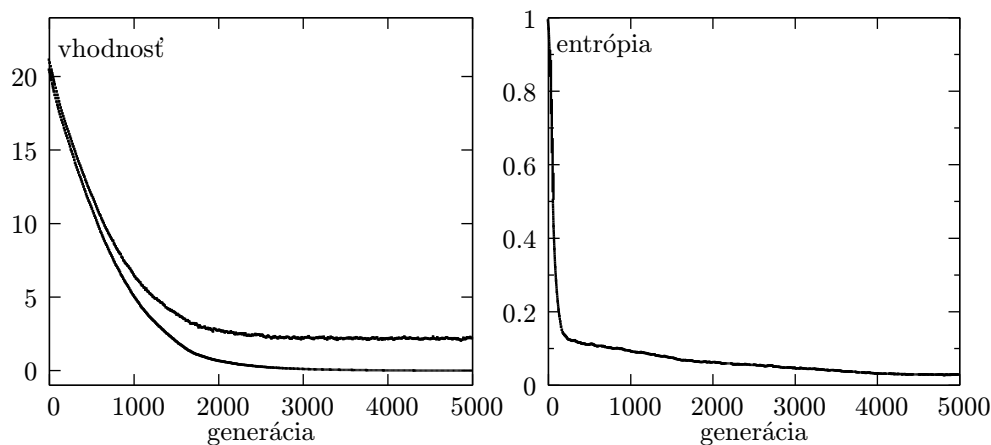
pričom sa použilo  $n = 30$  premenných a každá premenná sa skúmala v intervale  $-31 \leq x_i \leq 32$ .

Ako testovacie prostredie bol použitý jednoduchý tvar algoritmu, ktorý využíval nasledujúce prvky:

- náhodnú inicializáciu populácie (o veľkosti 100 jedincov),
- Grayov binárny kód pre reprezentáciu hodnôt premenných (každá premenná bola reprezentovaná pomocou 12 bitov),
- selekciu 10 rodičov binárnym turnajom bez náhrady,
- tvorbu 10 potomkov dvojbodovým krížením ( $p_c = 0.5$ ) a mutáciou ( $p_m = 1/360$ ),
- náhradu náhodných jedincov v populácii novo generovanými potomkami (s chránenou elitou 20 najlepších jedincov).

Štruktúra algoritmu umožňovala použitie metód pre udržiavanie rôznorodosti sústreďujúcich sa rozdielne fázy evolučného cyklu – ako na fázu selekcie, tak aj na fázu kríženia či nahradzovanie aktuálnej populácie novou generáciou.

Chovanie tohto testovacieho algoritmu je zobrazené na obr. 10.2, pričom kvôli získaniu zobrazených dát bolo vykonaných 100 opakovaní a dosiahnuté výsledky boli spriemernené. Sledoval sa priebeh vhodnosti v závislosti na čase (vľavo) a vývoj stupňa konvergenzie aktuálnej populácie (vpravo). Pri vhodnosti horný priebeh reprezentuje priemernú vhodnosť populácie, dolný priebeh zobrazuje vhodnosť najlepšieho jedinca v populácii. Stupeň konvergenzie je reprezentovaný entropiou populácie.



Obr. 10.2: Spriemerené výsledky pre testovací algoritmus.

Do použitého testovacieho algoritmu boli pridávané jednotlivé skúmané metódy pre potlačanie konvergenencie, pričom cieľom bolo zistiť, ako sa zmenia uvedené priebehy vhodností a stupňa konvergenencie.

Očakávaná bola, že zvýšenie rôznorodosti populácie bude charakterizované zvýšenou entropiou populácie ale aj zhoršením (zvýšením<sup>5</sup>) priemernej vhodnosti a pomalším poklesom najlepšej vhodnosti k nulovej hodnote. Vhodná metóda poskytne viditeľné zvýšenie entropie (čím väčšie tým lepšie) a zároveň nebude mať vplyv na vhodnosť najlepšieho jedinca. Metóda, ktorá zvýši entropiu iba nepatrne, alebo spôsobí väčšie spomalenie pri poklese najlepšej vhodnosti, je menej vhodná<sup>6</sup>.

V porovnávaní bolo použitých deväť metód reprezentujúcich triedy prístupov uvedené v predchádzajúcich častiach. Boli testované nasledujúce metódy:

**Operátor jedinečnosti (Eliminácia duplikácií).** Prah pre minimálnu vzdialenosť<sup>7</sup> bol na začiatku nastavený na hodnotu 8. V prípade, ak sa počas 100 generácií nepodarilo zaradiť žiadneho nového jedinca, bol tento prah dekrementovaný o jednotku. Nový jedinec bol kontrolovaný voči náhodnej

<sup>5</sup>Úlohou je minimalizácia – nájdenie globálneho minima funkcie.

<sup>6</sup>Opäť je potrebné pripomenúť, že prezentované výsledky nemajú globálnu platnosť a slúžia iba pre informatívne porovnanie metód – pri inej štruktúre algoritmu a/alebo testovacej úlohe, dosiahnuté výsledky sa môžu líšiť od tu prezentovaných.

<sup>7</sup>Vo všetkých testovaných metódach bola použitá Hammingova vzdialenosť.

vzorke 10 jedincov z populácie a v prípade prílišnej podobnosti bolo maximálne 5 pokusov o jeho vhodné mutovanie. V prípade neúspechu bol daný jedinec zahodený.

**Permanentné vkladanie (Reinicializácia).** V každej generácii bolo 10 najhorších jedincov nahradených novými jedincami. Tie boli generované náhodne s uvažovaním pravdepodobností pre generovanie jednotlivých hodnôt založených na aktuálnej distribúcii týchto hodnôt v populácii.

**Obmedzené spájanie (Obmedzené spájanie).** Prvý rodič je vyberaný normálnou selekciou. Druhý je vyberaný ako k prvému rodičovi najnepodobnejší jedinec zo skupiny 4 náhodne vybratých jedincov.

**Crowding (Obmedzená súťaž).** Potomok nahrádza najpodobnejšieho jedinca vybraného zo vzorky 3 jedincov náhodne vybratých z populácie. Podobnosť je meraná ako vzdialenosť medzi jedincami.

**Deterministický crowding (Obmedzená súťaž).** Potomkovia súťažšia so svojimi rodičmi, z každého súťažiaceho páru postupuje jedinec s lepšou vhodnosťou. Rodičia a potomkovia sú zadeľovaní do súťažných párov tak, aby sa minimalizovala priemerná vzdialenosť týchto párov.

**Zdieľanie (Penalizácia).** Pri penalizácii vhodnosti jedincov bola použitá lineárna zdieľacia funkcia ( $\alpha = 1.0$ ) s polomerom zdieľania 3.0.

**Dynamické zhlukovanie (Penalizácia).** Prvým prechodom cez zotriedenú populáciu jedincov boli vytvárané centrá zhlukov, ktorých počet nebol obmedzený. Jedinec vytváral nový zhluk, ak k centru žiadneho existujúceho zhluku nebol bližšie ako 9, pričom vzdialenosť sa určovala ako vzdialenosť medzi daným jedincom a jedincom predstavujúcim centrum zhluku. Následne každý jedinec bol zaradený do najbližšieho zhluku. Vhodnosť jedinca bola penalizovaná v závislosti od vzdialenosti k centru príslušného zhluku.

**Seceder selekcia (Preferencia odlišností).** Pre výber každého rodiča sa štandardnou selekčnou metódou vyberú 3 jedince a z nich sa stane ten jedinec rodičom, ktorého vzdialenosť od ostatných dvoch je najväčšia.

**Pohyblivý cieľ (Preferencia odlišnosti).** Vhodnosť jedincov bola modifikovaná vzťahom  $\phi(a_i)/[w + (1 - w)d(a_i)]$  kde  $d$  reprezentuje vzdialenosť uvažovaného jedinca voči referenčnému jedincovi, ktorým bol najlepší jedinec v aktuálnej generácii. Pre váhu  $w$  bola použitá hodnota 0.4.

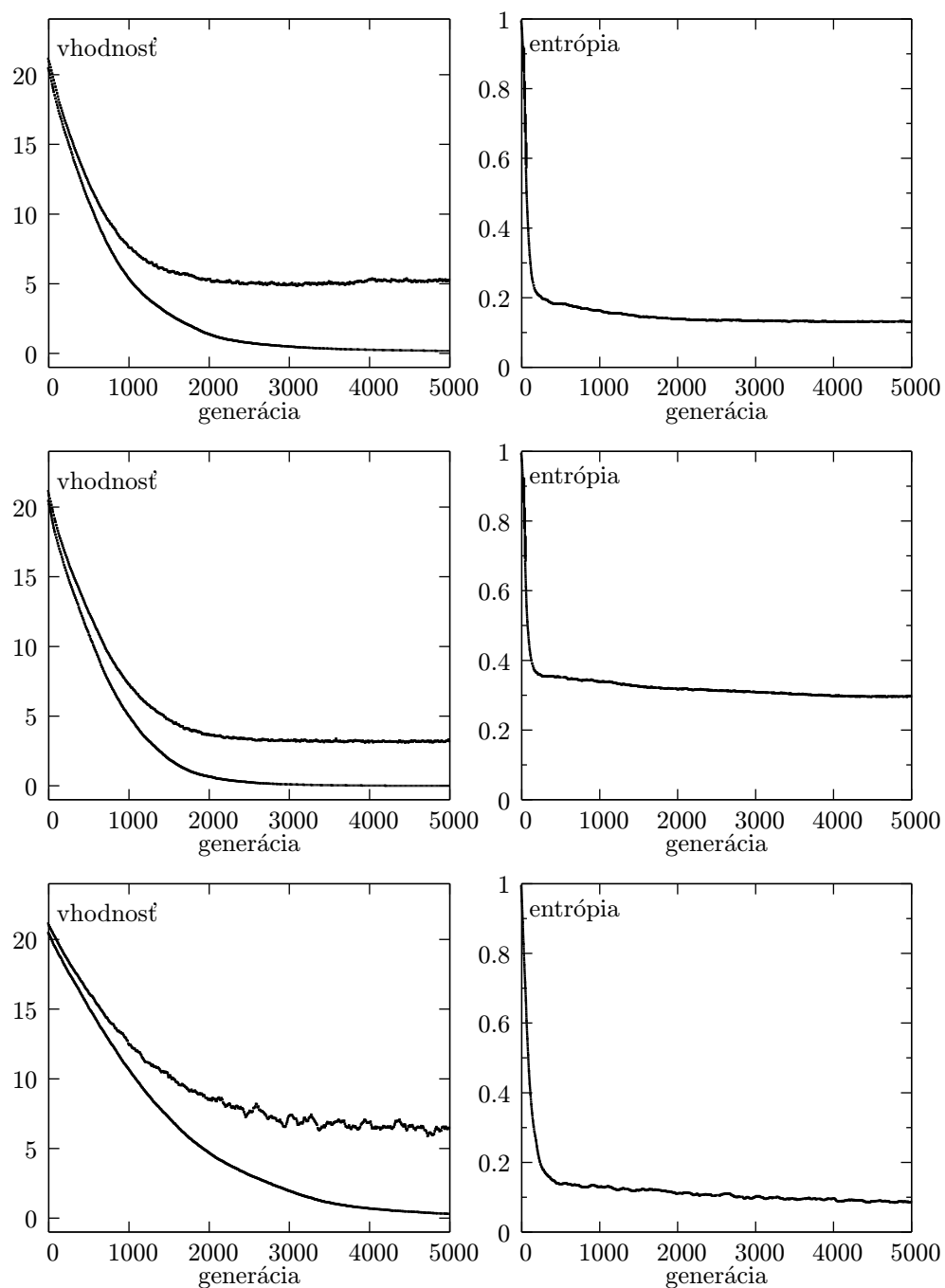
Výsledky dosiahnuté jednotlivými metódami sú prezentované na obrázkoch 10.3, 10.4 a 10.5. V prezentovaných výsledkoch je možné vidieť rozličné vzory chovania. Je možné pozorovať:

- Z hľadiska priemernej vhodnosti sú prítomné zhoršenia malé aj väčšie, pričom neexistuje jednoznačná korelácia medzi zhoršením priemernej vhodnosti a zvýšením entropie. Možno vidieť dvojice metód, kde metóda s väčšou zmenou priemernej vhodnosti poskytuje aj väčšiu zmenu entropie, avšak aj dvojice kde je tento vzťah opačný.
- Priebeh najlepšej vhodnosti sa pri niektorých metódach príliš nezhoršil, zatiaľ čo pri iných došlo k očividnému spomaleniu poklesu vhodnosti najlepších jedincov k hodnotám hľadaného globálneho extrému.
- Z hľadiska podpory rôznorodosti populácie je možné metódy rozdeliť do dvoch základných skupín:
  - metódy schopné udržať rôznorodosť populácie dlhodobo na stabilnej úrovni,
  - metódy podporujúce rôznorodosť populácie iba dočasne.

Prvý typ metód poskytuje stabilnú úroveň entropie (rozdielne metódy) ako v počiatočnej fáze evolučného hľadania tak aj v neskorších fázach.

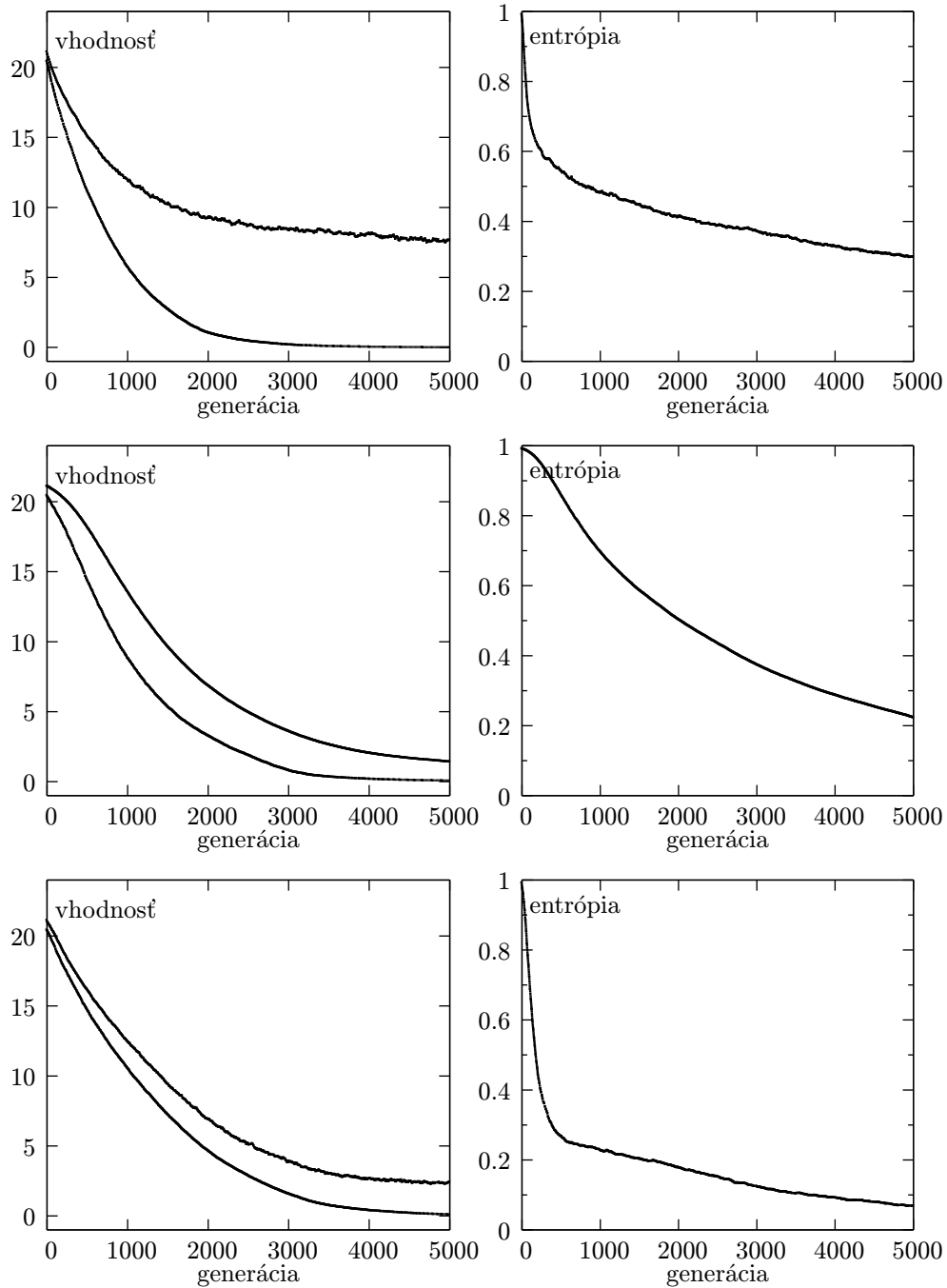
Druhý typ metód bol schopný poskytnúť väčšiu mieru rôznorodosti populácie, avšak iba v počiatočných fázach hľadania - s postupujúcim časom sa stupeň konvergence zvyšoval a entropia klesala (s rôznou rýchlosťou u rôznych metód). Ak algoritmus beží dostatočne dlhú dobu, tak entropia poskytovaná týmto typom metód klesne pod hodnoty entropie poskytované metódami prvého typu.

Niektoré z prezentovaných metód majú jeden alebo viac parametrov, ktorými je možné ovplyvniť účinnosť týchto metód. Vhodným nastavením je možné získať inú úroveň entropie čo je súčasne doprevádzané aj zmenami v priebehu vhodností. Menší dôraz na zachovanie priebehu najlepšej vhodnosti umožní zvýšenie rôznorodosti. Na druhej strane, menšie požiadavky

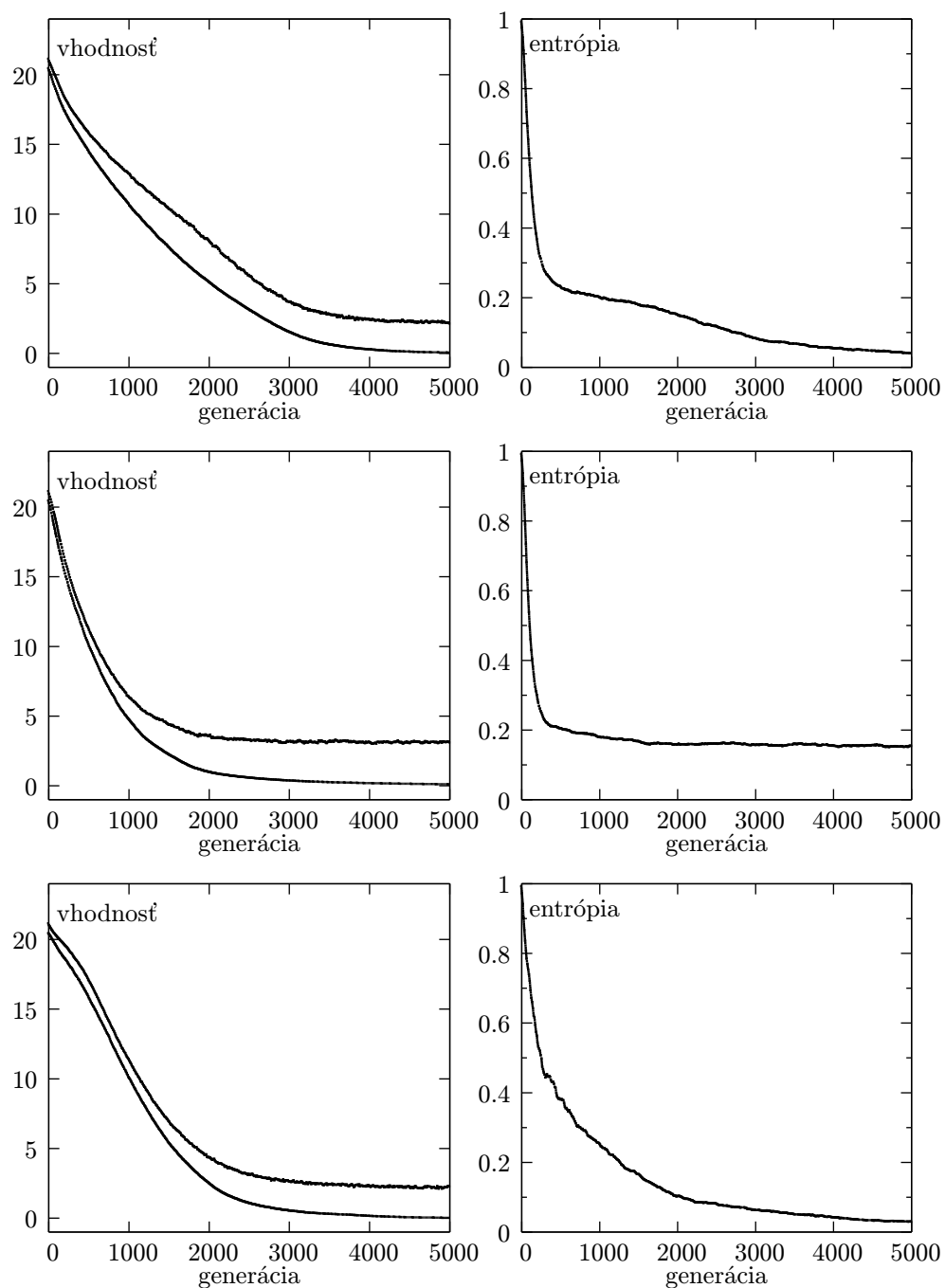


Obr. 10.3: Spriemerené výsledky pre operátor jedinečnosti (hore), permanentné vkladanie (v strede) a obmedzené spájanie (dole).





Obr. 10.4: Spriemerené výsledky pre crowding (hore), deterministický crowding (v strede) a zdieľanie (dole).



Obr. 10.5: Spriemernené výsledky pre dynamické zhukovanie (hore), seceder selekcii (v strede) a pohyblivý cieľ (dole).

na rôznorodosť umožnia menšiu degradáciu priebehu vhodností najlepších jedincov.

Ako vidno, je možné aplikáciou vhodnej metódy získať rozličné chovanie a rozličnú mieru podpory rôznorodosti stavebných blokov v populácii. Je tak už iba na návrhárovi, ktorý typ chovania uprednostní a v závislosti na tom ktorú metódu použije.



---

Časť V

**Pohľad pod kapotu**



## Kapitola 11

# Hypotéza stavebných blokov

Evolučný algoritmus rieši problémy v rozloženom tvare – každý problém je reprezentovaný množinou atribútov a ich hodnôt. Hodnoty atribútov majú charakter stavebných blokov, z ktorých je zostavený každý jedinec. Cieľom je nájsť takú kombináciu stavebných blokov, z ktorej je zložené hľadané riešenie. Hypotéza stavebných blokov [14] sa snaží podať predstavu o činnosti evolučného algoritmu:

Činnosť evolučného algoritmu je založená na vzorkovaní, rekombinovaní a opätovnom vzorkovaní krátkych stavebných blokov. Namiesto skúmania všetkých možných kombinácií stavebných blokov, vzorkovaním sú identifikované stavebné bloky s vysokou vhodnosťou a následne sú kombinované do zložitejších stavebných blokov s potenciálne vyššou vhodnosťou. Opakovaním sú konštruované stále lepšie a lepšie stavebné bloky.

Pre vysvetlenie tejto hypotézy začneme najjednoduchším prípadom, keď jedince síce budú kódovať hodnoty všetkých atribútov, avšak sústredíme sa na hodnotu iba jedného atribútu – a navyše to bude atribút binárny. Teda sú k dispozícii dva stavebné bloky (dve hodnoty daného atribútu) a je potrebné sa rozhodnúť, ktorý z nich má byť preferovaný resp. selektovaný.

Tento problém môže byť modelovaný pomocou pákového výherného automatu. Automat po vhození mince a potiahnutí páky vracia hráčovi jeho výhru, pričom výška výhry je náhodnou premennou s určitou strednou hodnotou a varianciou. Uvažujme automat s dvomi pákami. Prvá páka pri  $i$ -tom potiahnutí vracia výhru  $y_1(i)$ , zatiaľ čo druhá páka vracia  $y_2(i)$ . Stredná hodnota výhry pri použití prvej páky je  $E(y_1)$ , pri použití druhej páky je

stredná hodnota výhry  $E(y_2)$ . Obe stredné hodnoty ako aj variancie sú statické, počas hry sa nemenia.

Hráč má k dispozícii  $n$  mincí (pokusov) a jeho úlohou je získať čo najväčšiu celkovú výhru, pričom však môže pozorovať iba hodnoty  $y_1$  a  $y_2$ , ale stredné hodnoty ani variancie nepozná. Nemá žiadnu apriórnu informáciu, ktorá z pák mu s väčšou pravdepodobnosťou poskytne väčšiu výhru. Jediné, čo môže robiť, je postupne vkladať mince do automatu, potiahnuť jednou z pák, na základe pozorovaných výhier odhadovať, ktorá z pák poskytuje väčšiu výhru, a na základe svojho odhadu sa rozhodovať ktorú páku následne použiť.

V ideálnom prípade by bolo možné používať iba štedrejšiu páku. V tomto prípade by očakávaný zisk bol

$$n \max(E(y_1), E(y_2)) \quad (11.1)$$

Keďže však informácia o tom, ktorá páka je lepšia, je neznáma, je potrebné použiť obe páky. Ak páka s menšou strednou hodnotou bude použitá  $N_l$  krát a páka s väčšou strednou hodnotou zase  $N_h = n - N_l$  krát, tak očakávaná strata oproti ideálnemu prípadu bude

$$N_l |E(y_1) - E(y_2)| \quad (11.2)$$

Je možné použiť rôzne stratégie. Pri pesimistickej stratégii “na istotu” by sa pri polovici pokusov použila jedna páka a pri druhej polovici druhá páka. Takto získaná očakávaná strata by mala byť hornou hranicou – od sofistikovanejších stratégií sa očakáva menšia hodnota straty.

Pri použití ľubovoľnej stratégie je možné očakávanú stratu vyjadriť ako súčet

$$p_z(n - N_l)|E(y_1) - E(y_2)| + (1 - p_z)N_l|E(y_1) - E(y_2)| \quad (11.3)$$

Tento vzťah identifikuje dva možné zdroje straty:

- na základe vyplatených výhier bola páka s menšou strednou hodnotou výhry chybné identifikovaná ako páka lepšia, a teda všetkých  $n - N_l$  použití tejto chybné identifikovanej páky prinieslo stratu,
- identifikácia pák bola správna, strata bola zapríčinená  $N_l$  použitiami páky s menšou strednou hodnotou výhry, potrebnými pre správnu identifikáciu pák.



V uvedenom vzťahu  $p_z$  reprezentuje pravdepodobnosť chybnjej identifikácie pák, a môže byť daná ako

$$p_z = p \left( \frac{\sum_{i=1}^{n-N_l} y_1(i)}{n - N_l} < \frac{\sum_{i=1}^{N_l} y_2(i)}{N_l} \right) \quad (11.4)$$

v prípade, že lepšou je prvá páka, avšak pozorovaná priemerná výhra je vyššia pri druhej páke.

Optimálna stratégia je taká, ktorá použije hodnotu  $N_l$  minimalizujúcu očakávanú stratu. Po aproximovaní pravdepodobnosti  $p_z$  pomocou funkcie závislej na  $N_l$  (a samozrejme aj na stredných hodnotách výhier a smerodajných odchýlok oboch pák) a následnom derivovaní vzťahu (11.3) podľa  $N_l$  a položení tejto derivácie rovnaj nule je možné získať [31]

$$n - N_l \approx e^{kN_l} \quad (11.5)$$

kde  $k$  reprezentuje konštantu. To, čo je podstatné, nie je hodnota konštanty ale tvar závislosti – optimálna alokácia pokusov medzi obe páky je taká, keď počet pokusov pre tú páku, ktorá sa na základe pozorovaní javí byť lepšou, sa exponenciálne zvyšuje s počtom pokusov venovaných páke, ktorá sa na základe pozorovaní javí ako horšia.

Činnosť evolučného algoritmu je analogická činnosti hráča – musí sa rozhodnúť, ktorý stavebný blok má preferovať (t.j. ktorú páku zvoliť) a táto preferencia je založená na predchádzajúcich pozorovaniach vhodnosti jedincov s rôznymi stavebnými blokmi (t.j. pozorovaniach výhry pri použití rôznych pák)<sup>1</sup>.

Ak v populácii  $\mu$  jedincov je  $N_0$  takých jedincov, ktoré sú nositeľmi stavebného bloku “0” na sledovanej pozícii, tak po selekcii  $\varrho$  rodičov bude medzi nimi

$$\varrho \sum_{i=1}^{N_0} p_s(a_i) \quad (11.6)$$

jedincov, ktoré budú obsahovať tento stavebný blok na sledovanej pozícii. Ak budeme uvažovať priemernú pravdepodobnosť selekcie jedincov, ktoré obsahujú daný stavebný blok, a navyše táto bude o nejakú hodnotu  $k$  posunutá voči priemernej pravdepodobnosti selekcie, tak predchádzajúci vzťah je možné upraviť na

$$\varrho(1 + k)\overline{p}_s N_0 \quad (11.7)$$

kde  $\overline{p}_s$  reprezentuje priemernú selekčnú pravdepodobnosť. Ak by sa selektovalo toľko rodičov ako je jedincov v populácii a použila sa selekčná metóda

---

<sup>1</sup>Táto preferencia je úlohou selekčného tlaku.

s priemernou pravdepodobnosťou selekcie rovnou  $1/\mu$ , tak medzi rodičmi možno očakávať

$$(1 + k)N_0 \tag{11.8}$$

nositeľov daného stavebného bloku. Po  $n$  opakovaných selekciách by bolo  $k$  dispozícií

$$(1 + k)^n N_0 \tag{11.9}$$

nositeľov tohto stavebného bloku. Teda počet jedincov by sa exponenciálne zmenšoval pri podpriemernej pravdepodobnosti selekcie a naopak exponenciálne zväčšoval pri nadpriemernej pravdepodobnosti selekcie. Keďže sa opäť jedná o exponenciálne alokovanie podobne ako v prípade výherného automatu, evolučný algoritmus identifikuje a alokuje relevantné stavebné bloky spôsobom blízkym optimálnej alokácii pokusov.

Záveru uvedenej analógie sú platné aj pre prípad, že na danej pozícii sa môže vyskytnúť viac stavebných blokov – jednoducho by sa urobilo porovnanie s rozšíreným výherným automatom s viacerými pákami<sup>2</sup>. V prípade reálnej reprezentácie možno chápať hľadanie reálnej hodnoty ako identifikáciu podintervalu, v ktorom sa táto hodnota nachádza. Keďže obor hodnôt atribútu možno rozdeliť na konečný počet takýchto podintervalov, je možné uvedenú analógiu použiť aj pre reálnu kódovaciu schému.

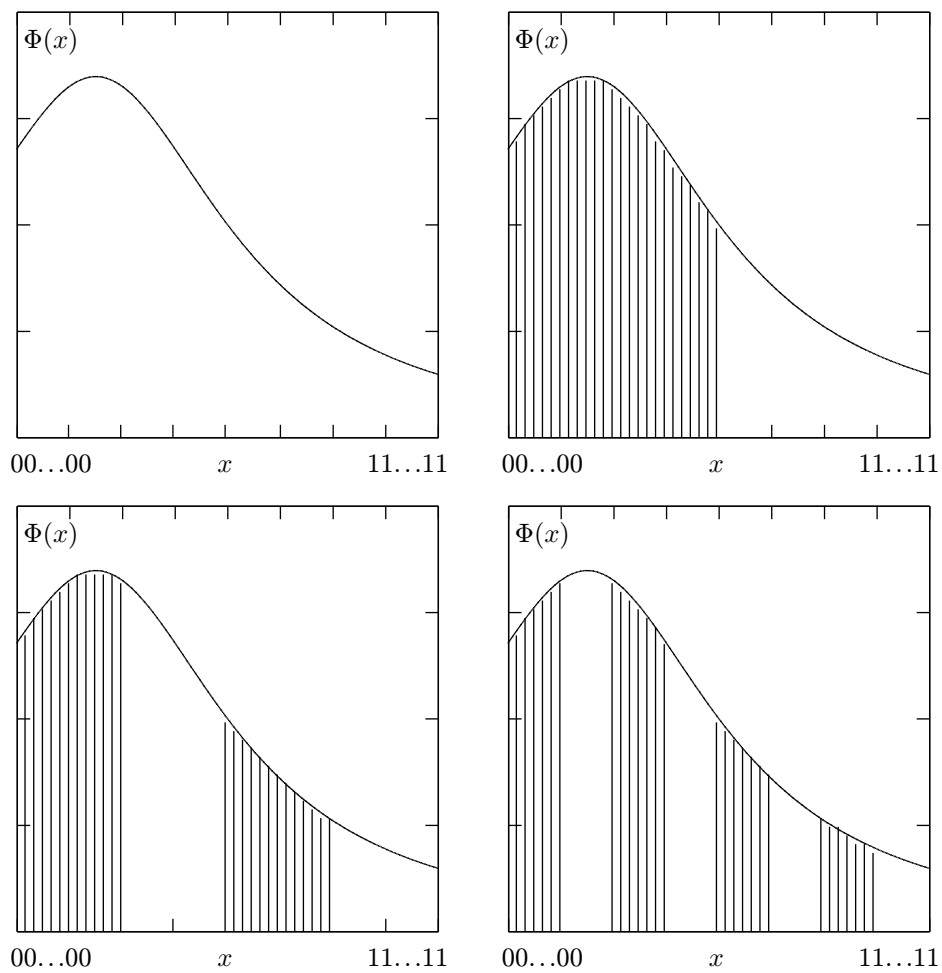
Evolučný algoritmus však nepracuje iba s jednou pozíciou, ale naraz uvažuje stavebné bloky na všetkých pozíciách štruktúry jedincov. Tomu je analogická predstava viacerých navzájom nezávislých výherných automatov, pričom každý z nich zodpovedá jednej pozícii. Osobitne pre každú pozíciu prebieha skúmanie, ktorý z možných stavebných blokov je lepší a ktorý je horší. Počet nositeľov tých stavebných blokov, ktoré boli identifikované ako lepšie, narastá v populácii, zatiaľ čo počet nositeľov blokov, identifikovaných ako horšie, postupne klesá. Identifikácia vhodných stavebných blokov na jednotlivých pozíciách prebieha paralelne.

Toto paralelné prehľadávanie je možné ilustrovať pomocou obr. 11.1. Obrázok ilustruje priebeh funkcie vhodnosti, pričom použitý priestor prehľadávania je zložený z binárnych atribútov – na každej pozícii sa môže vyskytovať iba jeden z dvoch stavebných blokov “0” alebo “1”.

Hodnota na prvej pozícii rozdeľuje priestor prehľadávania na dve polovice (obr. 11.1 vpravo hore). Keďže jedince nesúce “0” na prvej pozícii majú vyššiu vhodnosť ako jedince obsahujúce “1” na tejto pozícii, hodnota “0” je identifikovaná ako vhodnejší stavebný blok a bude preferovaná. Následkom

---

<sup>2</sup>Optimálna alokácia pokusov je podobná prípadu s dvomi pákami – opäť je potrebné alokovať exponenciálne rastúci počet pokusov pre páku, ktorá sa javí byť najlepšou.



Obr. 11.1: Plocha vhodnosti a jej segmentovanie podľa stavebných blokov na prvej pozícii (vpravo hore), druhej pozícii (vľavo dole) a tretej pozícii (vpravo dole)

toho bude v populácii stúpať počet jedincov ktoré budú mať tvar “0\*\*\*...\*”, kde znak \* reprezentuje ľubovoľnú hodnotu.

Podobne aj hodnota na druhej pozícii rozdeľuje priestor prehľadávania na dve polovice (obr. 11.1 vľavo dole). Aj keď teraz jedinec s hodnotou “1” na druhej pozícii môže mať vyššiu vhodnosť než jedinec s hodnotou “0”, priemerná vhodnosť jedincov nesúcich “0” je vyššia než priemerná vhodnosť jedincov obsahujúcich “1”. Pri dostatočnom vzorkovaní je toto odhalené a hodnota “0” bude identifikovaná ako vhodnejší stavebný blok. Jej preferencia spôsobí, že v populácii bude stúpať počet jedincov tvaru “\*0\*\*...\*”.

Preferencia vhodnejších stavebných blokov na prvej a druhej pozícii bude mať následok aj pre ich kombinovanie. V populácii sa najčastejšie vyskytne kombinácia “00\*...\*”, menej časté budú kombinácie “01\*...\*” a “10\*...\*”, a kombinácia “11\*...\*” bude iba zdriedkavosťou. Prehľadávanie algoritmu sa teda zameria najmä na oblasť, charakterizovanú hodnotou “0” na prvých dvoch pozíciách (čo je žiadaným chovaním).

Evolučný algoritmus teda identifikuje pre každú pozíciu, ktorý stavebný blok na tejto pozícii má vyššiu vhodnosť<sup>3</sup>. Tieto identifikované stavebné bloky sú preferované a v konečnom dôsledku je z nich skladané výsledné riešenie. Tento pohľad na činnosť evolučného algoritmu má však trochu obmedzenú platnosť. Podľa obr. 11.1 je zrejmé, že na tretej pozícii má stavebný blok “0” vyššiu vhodnosť než blok “1” (vpravo dole). Teda budú preferované jedince tvaru “\*\*0\*...\*” – čo by však zabránilo nájdeniu skutočného riešenia a poskytlo iba suboptimálne riešenie.

Problém je v tom, že medzi jednotlivými stavebnými blokmi na rôznych pozíciách môžu existovať vzájomné závislosti – epistáza. Táto epistáza môže mať rozličný stupeň:

- žiadna – medzi stavebnými blokmi neexistuje závislosť; zámena stavebného bloku, ktorý nie je súčasťou riešenia, za blok, ktorý je súčasťou hľadaného riešenia, má za následok nárast vhodnosti,
- slabá – zámena stavebného bloku, ktorý nie je súčasťou riešenia, za blok, ktorý je súčasťou hľadaného riešenia, sa nemusí prejavíť zmenou vhodnosti,
- silná – zámena stavebného bloku, ktorý nie je súčasťou riešenia, za blok, ktorý je súčasťou hľadaného riešenia, môže mať za následok aj pokles vhodnosti.

---

<sup>3</sup>Vhodnosť stavebného bloku sa uvažuje ako priemer vhodností tých jedincov, ktoré sú nositeľmi daného bloku.

Deficit epistázy sa môže vyskytnúť iba pri unimodálnych funkciách vhodnosti<sup>4</sup>. Prípad výskytu nielen globálneho extrému ale aj lokálnych extrémov korešponduje s výskytom silnej epistázy. Keďže aj takéto úlohy sú riešiteľné evolučným algoritmom, mechanizmus jeho činnosť musí byť zložitejší než mechanizmus jednoduchého modelu uvedeného vyššie.

Evolučnému algoritmu by nestačilo teda skúmať iba stavebné bloky na jednotlivých pozíciách. Potrebuje skúmať aj kombinácie stavebných blokov na viacerých pozíciách súčasne. Pri takomto skúmaní potom v prípade podľa obr. 11.1 síce identifikuje hodnotu “0” ako vhodný stavebný blok pre každú z prvých troch pozícií, avšak na základe skúmania dvojíc stavebných blokov rozdiel medzi štruktúrami jedincov “\*01 \* ...\*” a “\*00 \* ...\*” už nie je veľký<sup>5</sup>, a pri skúmaní všetkých troch pozícií naraz bude dochádzať k jasnej preferencii “001 \* ...\*” pred “000 \* ...\*”.

Ak teda riešenie je kódované na  $l$  pozíciách, potom si možno evolučný algoritmus predstaviť ako analógiu  $2^l - 1$  výherných automatov identifikujúcich vhodnosť jednotlivých stavebných blokov a ich kombinácií, pričom  $l$  automatov bude skúmať stavebné bloky na jednej pozícii,  $l(l - 1)/2$  automatov bude skúmať kombinácie stavebných blokov na dvoch pozíciách, atď.

Nie všetky automaty sú však uvažované v rovnakej miere. Čím totiž automat skúma väčšie kombinácie stavebných blokov, tým viac sa prejavuje:

- činnosť genetických operátorov, ktoré môžu zmeniť nejakú kombináciu stavebných blokov za inú a tým porušiť exponenciálne rozširovanie danej kombinácie v populácii,
- neuniformná distribúcia blokov v populácii, spôsobujúca že vzorkovanie nejakej kombinácie stavebných blokov nebude naprieč celým priestorom prehľadávania ale iba v nejakom limitovanom podpriestore.

Následkom toho je, že počet automatov, ktoré efektívne pôsobia na proces prehľadávania, je menší než uvedený počet.

---

<sup>4</sup>Príkladom je ‘one-max’ problém keď vhodnosť je daná ako počet jednotiek pri použití binárnej kódovacej schémy.

<sup>5</sup>Obzvlášť, ak vplyvom preferencií na prvej pozícii sa vzorkovanie sústreďuje najmä do oblasti s hodnotou “0” na prvej pozícii.



## Kapitola 12

# Vizualizácia

Princípy, na ktorých sú evolučné algoritmy založené, sú pomerne jednoduché. Napriek tomu vizualizácia ich činnosti je pomerne zložitou úlohou, pretože pri svojej činnosti produkujú veľké množstvo dát. Produkované dáta, s ktorými evolučné algoritmy pracujú, samé osebe nič nepredstavujú. Tieto dáta musia byť vhodným spôsobom interpretované, aby z nich boli extrahované informácie o vyskytujúcich sa stavoch a prebiehajúcich procesoch.

Úlohou vizualizácie je produkované dáta nejakým spôsobom predspracovať (selektovať, agregovať, odvodiť nové dáta a pod.) a následne upraviť do grafickej formy, ktorá je pre ľudského pozorovateľa jednou z najvhodnejších. Užívateľ tým získava prehľad o práci algoritmu, ktorý ho nielen informuje o aktuálnom dianí, ale mu aj umožňuje vykonávať informované zásahy do činnosti algoritmu [8]. Takýmto spôsobom mu vizualizácia umožňuje realizovať spätnú väzbu, ktorá je podstatná najmä pri návrhu a doladovaní algoritmu. Pomocou vizualizácie má možnosť získať informácie, umožňujúce najmä vnímanie:

- dynamiky hľadania lepších riešení,
- efektu nejakého konkrétneho nastavenia hodnôt riadiacich parametrov,
- stavu konverencie procesu prehľadávania,
- vytváraných preferencií (napr. na úrovni hodnôt atribútov),
- výsledku aplikovania jednotlivých operátorov a ich podielu na progrese procesu prehľadávania,
- stupňa a spôsobu pokrytia priestoru prehľadávania.

V oblasti vizualizácie dát, produkovaných evolučnými algoritmami, dominuje vizualizácia tých dát, ktoré reprezentujú výsledky prehľadávania priestoru. Avšak je možné sa stretnúť aj s inými spôsobmi vizualizácie, napríklad s demonštráciou činnosti genetických operátorov alebo so sledovaním príbuzenských vzťahov medzi jedincami alebo hodnotami atribútov.

Podľa úrovne použitej detailnosti je možné vizualizačné techniky rozdeľovať do troch skupín. Tieto skupiny sa koncentrujú na vizualizáciu:

- populácie jedincov ako celku,
- jednotlivých indivíduí,
- atribútov a ich hodnôt.

## 12.1 Vizualizácia populácie

Pri tomto druhu vizualizácie sa uvažuje populácia ako celok (bez rozoznávania konkrétnych jedincov a/alebo hodnôt atribútov), pričom v centre pozornosti je vždy nejaká charakteristika, ktorá je schopná vypovedať o populácii ako celku. Je možné vizualizovať túto charakteristiku pre nejakú konkrétnu generáciu a teda skúmať stav populácie v určitom čase. Dôraz sa však môže klásť nie na stav ale na dynamiku vývoja populácie a objektom záujmu je potom vývoj danej charakteristiky počas určitého obdobia (určitého počtu generácií).

Základným vizualizačným cieľom býva vhodnosť populácie. Pre jej zobrazovanie sa najčastejšie používa dvojrozmerný graf, ktorého vodorovná os reprezentuje “čas” (vyjadrený napríklad ako číslo generácie) a zvislá os zase konkrétnu hodnotu v danom čase. Samotná vhodnosť populácie býva reprezentovaná jednou z týchto vhodností:

- najlepšia vhodnosť v populácii,
- najhoršia vhodnosť v populácii,
- priemerná vhodnosť jedincov v populácii,
- on-line vhodnosť,
- off-line vhodnosť.

Príkladom takéhoto zobrazenia je obr. 10.2 vľavo. Samozrejme, v tom istom grafe je možné zobrazovať viacero vhodností. Najčastejšou voľbou sú



prvé tri možnosti alebo posledné dve. Je takisto možné zobraziť priemernú vhodnosť v kombinácii s indikáciou rozptylu či smerodajnej odchýlky hodnôt vhodnosti, prítomných v populácii. Kombinovanie viacerých vhodností môže v konkrétnom prípade naraziť na potrebu kombinovať hodnoty vyžadujúce diametrálne rozličné mierky (zvyčajne sa to stáva pri použití najhoršej vhodnosti).

Iným cieľom záujmu býva konvergencia populácie. Keďže stupeň konvergenzie populácie v určitej generácii je možné vyjadriť ako numerickú hodnotu (napr. podľa vzťahov (10.1) až (10.4)), nie je problémom zostrojiť dvojrozmerný graf, zobrazujúci túto hodnotu v závislosti od generácie. Príkladom je obr. 10.2 vpravo.

**Histogram distribúcie vzdialenosti.** Poskytuje o čosi detailnejší pohľad na (priestorovú) konvergenciu než iba použitie jednej numerickej hodnoty. Pri tejto metóde sa počítajú vzájomné vzdialenosti všetkých dvojíc jedincov. V histograme sa zobrazuje frekvencia existujúcich vzdialeností. Sekvencia histogramov pre rôzne generácie umožňuje sledovať vývoj vzdialeností v čase.

Takéto zobrazenie poskytuje názorný pohľad. Čím existuje viac väčších vzdialeností v populácii, tým je populácia rôznorodejšia. Ak sa populácia homogenizuje, ťažisko histogramu sa posúva smerom k zvislej osi (v úplne homogénnej populácii existujú iba nulové vzdialenosti). Histogram dokáže do určitej miery indikovať aj výskyt zhlukov jedincov prostredníctvom objavenia sa viacerých oddelených vrcholov v histograme (malé vzdialenosti medzi jedincami v rámci zhľuku a väčšie medzi jedincami rôznych zhľukov).

## 12.2 Vizualizácia jedincov

Táto skupina vizualizačných techník sa sústreďuje na jedince vytvárajúce populáciu, pričom typicky sa pre jednotlivé jedince uvažujú ich dve charakteristiky – poloha jedinca v priestore prehľadávania (daná hodnotami atribútov reprezentujúcich súradné osi tohto priestoru) a jeho vhodnosť. Samotné vizualizačné techniky pre zobrazovanie informácií používajú rôzne grafické elementy, pričom takýto element reprezentuje celého jedinca.

Je prirodzeným vizualizovať distribúciu jedincov (alebo jej zmeny) v priestore prehľadávania. Takýto spôsob vizualizácie umožňuje sledovať také charakteristiky ako cestu generovanú (niekoľkými najlepšimi) jedincami v tomto priestore, konvergenciu populácie, pokrytie priestoru prehľadávania, a pod.

Bohužiaľ, priestor prehľadávania reprezentuje mnohorozmerný dátový priestor. Pretože ľudské pozorovacie schopnosti sú limitované na tri alebo menej dimenzií, všetky relevantné vizualizačné metódy sú založené na transformácii mnohodimenzionálnych dát na menší počet dimenzií. Typicky, dáta sú transformované do roviny ktorá môže byť zobrazená jednoducho na obrazovke monitora.

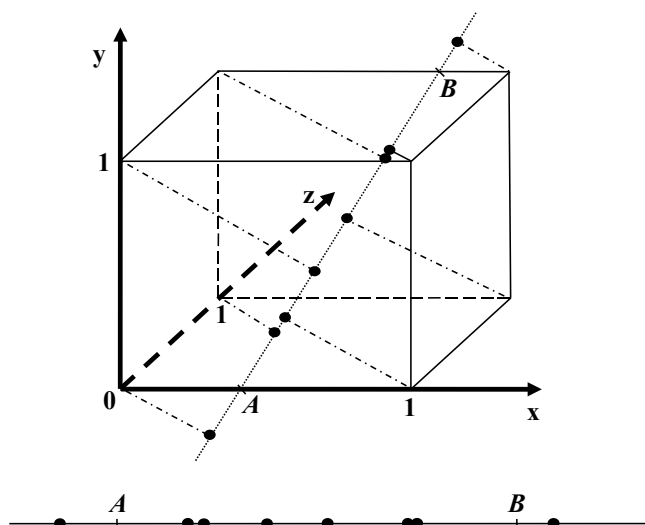
Metód pre transformáciu mnohorozmerného priestoru do priestoru s menším počtom rozmerov existuje značné množstvo. Pomerne známou je napr. metóda analýzy hlavných komponentov. Pre túto metódu (a mnohé ďalšie) však neexistuje iba jedno všeobecne použiteľné mapovanie, ale pre každú generáciu (charakterizovanú svojím vlastným rozložením bodov v mnohorozmernom priestore) je potrebné vytvárať nové mapovanie, ktoré zodpovedá aktuálnej distribúcii bodov. Následkom toho nie je garantovaná priestorová konzistencia – ten istý bod môže byť v rôznych generáciách mapovaný rozličným spôsobom. Prednosť sa zvykne dávať priestorovo konzistentným metódam.

**Premietanie.** Túto techniku možno ilustrovať na premietaní na priamku [27], hoci je možné použiť aj premietanie na rovinu, ktoré má však väčšie výpočtové nároky. Pri tomto prístupe sa predpokladá umiestnenie priamky (nazývanej “zobrazovacej”) v mnohorozmernom priestore prehľadávania, nasledované projekciou všetkých bodov tohto priestoru, ktoré reprezentujú jedincov, na túto zobrazovaciu priamku.

Tento proces je znázornený na obr. 12.1, kde je pre jednoduchosť priestor prehľadávania znázornený ako trojrozmerný, avšak metódu je možné použiť aj pri vyšších rozmerostiach priestoru. Zobrazovacia priamka je daná bodmi  $A$  a  $B$ . Výsledok priemetov ôsmich bodov (vrcholov kocky) je množina bodov na zobrazovacej priamke (ich počet je závislý na polohe priamky v priestore – v situácii podľa obrázku to bude minimálne dvojica bodov a maximálne bude na priamke toľko bodov, koľko ich je premietaných z priestoru na priamku).

Pre určenie priemetov bodov na zobrazovacej priamke je možné odvodiť matematický vzťah. Na definovanie priamky v  $n$ -rozmernom priestore postačia dva body  $A$  a  $B$  so súradnicami  $[A_1, A_2, \dots, A_n]$  a  $[B_1, B_2, \dots, B_n]$ . Smerový vektor definovaný týmito bodmi je:

$$\vec{u} = (u_1, u_2, \dots, u_n) = (B_1 - A_1, B_2 - A_2, \dots, B_n - A_n) \quad (12.1)$$



Obr. 12.1: Premietanie na zobrazovaciu priamku

Parametrická rovnica zobrazovacej priamky má potom tvar:

$$\begin{aligned}
 x_1 &= A_1 + u_1 z = A_1 + (B_1 - A_1)z \\
 x_2 &= A_2 + u_2 z = A_2 + (B_2 - A_2)z \\
 &\vdots \\
 x_n &= A_n + u_n z = A_n + (B_n - A_n)z
 \end{aligned}
 \tag{12.2}$$

kde parameter  $z$  je reálnym číslom, pričom zároveň reprezentuje polohu na priamke (hodnota 0 reprezentuje bod  $A$  a hodnota 1 zase bod  $B$ ).

Nech  $C$  je ľubovoľný bod v danom  $n$ -rozmernom priestore so súradnicami  $[C_1, C_2, \dots, C_n]$ . Výsledkom jeho priemetu na priamku je bod  $D$  so súradnicami  $[D_1, D_2, \dots, D_n]$ . Keďže tento leží na priamke, hodnoty jeho súradníc musia vyhovovať parametrickej rovnici priamky (12.2) a hodnota parametra  $z$  (ktorú je potrebné určiť) slúži ako indikátor jeho polohy na zobrazovacej priamke.

Keďže sa jedná o kolmý priemet, vektor daný bodmi  $C$  a  $D$  musí byť kolmý na smerový vektor priamky a teda ich skalárny súčin musí byť rovný nule:

$$(C_1 - D_1)u_1 + (C_2 - D_2)u_2 + \dots + (C_n - D_n)u_n = 0 \tag{12.3}$$

Dosadením parametrických rovníc za súradnice  $D_i$  je možné určiť hodnotu

hľadaného parametra  $z$ :

$$z = \frac{(C_1 - A_1)u_1 + (C_2 - A_2)u_2 + \dots + (C_n - A_n)u_n}{u_1^2 + u_2^2 + \dots + u_n^2} \quad (12.4)$$

Použitie priamky na zobrazovanie namiesto roviny spôsobuje úbytok informácií o umiestnení bodov. Preto ako prídavnú informáciu pri samotnej vizualizácii je vhodné použiť aj vzdialenosti bodov od priamky. Vzdialenosť bodu  $C$  od priamky je vlastne vzdialenosť bodu  $C$  a jeho priemetu  $D$  a je daná ako:

$$\sqrt{(C_1 - D_1)^2 + (C_2 - D_2)^2 + \dots + (C_n - D_n)^2} \quad (12.5)$$

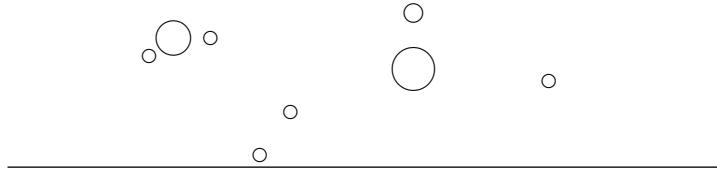
Výhodou metódy je možnosť voliť polohu zobrazovacej priamky a tým získavať rozličné pohľady. Táto voľba však môže byť obtiažna – preto sa doporučuje najprv použiť nejakú štandardnú polohu (napr. “krížom” cez priestor) a iba v prípade, že táto poloha z nejakého dôvodu nevyhovuje, experimentovať s polohou priamky<sup>1</sup>.

Napriek použitiu roviny na zobrazovanie nie je vylúčená (a v praxi aj veľmi často nastáva) situácia, keď viacero bodov priestoru, reprezentujúcich jedince v populácii, je premietaných do toho istého bodu (teda sa nachádzajú na tej istej ploche kolmej na zobrazovaciu priamku a sú od nej rovnako vzdialené, pričom ich vzájomná vzdialenosť môže byť väčšia ako ich vzdialenosť k zobrazovacej priamke). Aj keď priemety takýchto bodov sa navzájom nerozlišujú, je vhodné informovať o tom, že nastala takáto situácia. Táto dodatočná informácia môže byť poskytovaná viacerými spôsobmi, napr. pomocou číselnej informácie doprevádzajúcej každý zobrazený bod, doplnkového histogramu alebo pomocou využitia farieb (či už použitie viacerých farieb alebo použitie založené na rôznych intenzitách jednej farby). Na obr. 12.2 je táto dodatočná informácia zobrazovaná veľkosťou bodu reprezentujúceho priemet (keďže vzdialenosť je vždy nezáporná, zobrazovanie sa robí iba nad priamkou).

**Sammonovo mapovanie.** Sammonovo mapovanie [21] je nelineárna mapovacia technika. Je založená na základnej myšlienke zachovať vzdialenosti medzi jednotlivými bodmi. Snaží sa zobraziť body v nízkorozmernom priestore tak, aby ich vzájomné vzdialenosti čo najlepšie zodpovedali vzdialenostiam v pôvodnom mnohorozmernom priestore.

---

<sup>1</sup>Ak zobrazovacia priamka bude stotožnená s niektorou z osí priestoru, tak vznikne toľko priemetov na nej, koľko hodnôt atribútu, reprezentovaného danou osou, sa v populácii vyskytuje. Zo zobrazenia (v spojitosti s technikou použitou na obr. 12.2) je potom možné určiť prítomnosť a frekvenciu jednotlivých hodnôt daného atribútu v populácii.



Obr. 12.2: Zobrazenie s dodatočnou informáciou

K zachovaniu vzdialeností medzi bodmi aj po zobrazení dochádza iba vo výnimočných prípadoch, vo všeobecnosti sa však nové vzdialenosti nerovnajú vzdialenostiam pôvodným. Dochádza k ich zmene. Ak je potrebné zobraziť  $\mu$  bodov a vzdialenosť medzi  $i$ -tým a  $j$ -tým bodom je  $d_{ij}^{IN}$  v pôvodnom mnohorozmernom priestore a  $d_{ij}^{OUT}$  v priestore cieľovom, potom zmena vzdialeností je charakterizovaná pomocou kritéria:

$$\frac{1}{\sum_{i=1}^{\mu-1} \sum_{j=i+1}^{\mu} d_{ij}^{IN}} \sum_{i=1}^{\mu-1} \sum_{j=i+1}^{\mu} \frac{(d_{ij}^{IN} - d_{ij}^{OUT})^2}{d_{ij}^{IN}} \quad (12.6)$$

kde hodnota 0 indikuje bezstratové zobrazenie. Teda optimálna konfigurácia bodov minimalizuje hodnotu tohto kritéria.

Pri hľadaní spôsobu projekcie minimalizujúcej uvedené kritérium sa začína s nejakou inicializačnou konfiguráciou bodov (zvolenou napríklad náhodne) a určí sa hodnota kritéria. V nasledujúcich krokoch konfigurácia je upravovaná zmenami pozície jednotlivých bodov s cieľom aproximovať stále lepšie a lepšie pôvodnú distribúciu vzdialeností. Tento proces má iteračný charakter. V pôvodnej verzii metódy sa používala Newtonova metóda (gradientová metóda najstrmšieho zostupu). Pretože však zložitosť celého procesu bola pre praktické použitie príliš vysoká, boli navrhnuté metódy pre urýchlenie celého procesu [33].

Aby sa dosiahla priestorová konzistencia, nerobí sa hľadanie projekcie pre každú generáciu zvlášť. V podstate sú možné dva základné spôsoby:

- hľadá sa projekcia s uvažovaním všetkých bodov, ktoré sa budú zobrazovať, pričom sa vždy naraz zobrazia iba tie body, ktoré tvoria niektorú generáciu,
- vytvorí sa projekcia pre jednu generáciu a do tejto sa pre ďalšiu generáciu dopĺňajú nové body, ktoré v predchádzajúcej neboli, pričom poloha bodov z predchádzajúcej generácie sa nemení.

**Lineárne mapovanie.** V prípade, že atribúty sú nominálne (každý môže nadobudnúť iba určitý počet hodnôt), počet bodov priestoru prehľadávania je konečný. Pre tento prípad je možné dopredu pripraviť nejaké mapovanie všetkých možných bodov mnohorozmerného priestoru do nižšieho počtu dimenzií. Príkladmi sú mapy pokrytia alebo matica priestoru prehľadávania [8], ktoré reprezentujú lineárny spôsob mapovania.

8	02020	02021	02120	02121	12020	12021	12120	12121
7	02010	02011	02110	02111	12010	12011	12110	12111
6	02000	02001	02100	02101	12000	12001	12100	12101
5	01020	01021	01120	01121	11020	11021	11120	11121
4	01010	01011	01110	01111	11010	11011	11110	11111
3	01000	01001	01100	01101	11000	11001	11100	11101
2	00020	00021	00120	00121	10020	10021	10120	10121
1	00010	00011	00110	00111	10010	10011	10110	10111
0	00000	00001	00100	00101	10000	10001	10100	10101
	0	1	2	3	4	5	6	7

Obr. 12.3: Matica priestoru prehľadávania

Príklad takéhoto mapovania do roviny je na obr. 12.3 (matica priestoru prehľadávania). Pri tomto spôsobe párne pozície (presnejšie hodnoty atribútov na párnych pozíciách) slúžia na určenie pozície v rámci jednej dimenzie a podľa nepárnych pozícií sa určuje druhá dimenzia. Rovina (alebo jej časť) sa postupne rozdeľuje v horizontálnom aj vertikálnom smere vždy na toľko častí, koľko hodnôt môže nadobúdať práve spracovávaný atribút.

Samotné mapovanie je možné realizovať pomocou jednoduchého vzťahu. Ak je potrebné zobrazit jedinca s  $l$  atribútmi, ktorých hodnoty sú  $At_1 = v_1, At_2 = v_2, \dots, At_l = v_l$ , potom pre jednu dimenziu je možné použiť vzťah

$$\sum_{0 \leq i < l} v_{l-i} \prod_{0 \leq j < i} \| At_{l-j} \| \quad (12.7)$$

kde indexy  $i$  a  $j$  sa menia s krokom 2 a  $\| At_i \|$  reprezentuje kardinalitu množiny možných hodnôt  $i$ -teho atribútu. Analogicky je možné určiť druhú súradnicu (použijúc zostávajúce atribúty)

$$\sum_{1 \leq i \leq l} v_{l-i} \prod_{1 \leq j < i} \| At_{l-j} \| \quad (12.8)$$

pričom opäť indexy  $i$  a  $j$  sa menia s krokom 2.

Ak by sme v súlade s obr. 12.3 uvažovali päť atribútov ( $l = 5$ ), pričom druhý a štvrtý sú ternárne a ostatné sú binárne ( $\| At_1 \| = \| At_3 \| = \| At_5 \| = 2$  a  $\| At_2 \| = \| At_4 \| = 3$ ), tak jedinec “10120” by bol mapovaný na

$$\begin{aligned} v_5 + v_3 \| At_5 \| + v_1 \| At_5 \| \| At_3 \| &= 0 + 1 \cdot 2 + 1 \cdot 2 \cdot 2 = 6 \\ v_4 + v_2 \| At_4 \| &= 2 + 0 \cdot 3 = 2 \end{aligned} \quad (12.9)$$

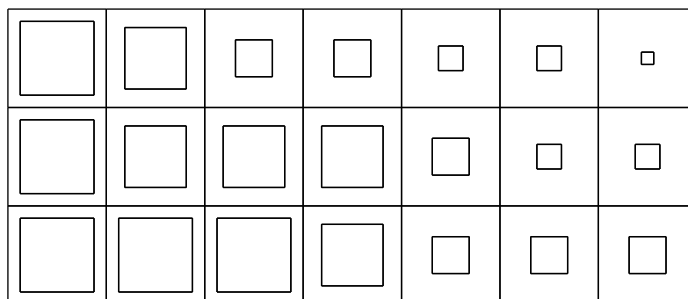
zatiaľ čo jedinec “02111” by bol mapovaný na 3 a 7.

Metóda môže byť s určitou stratou presnosti použitá aj pre vizualizáciu reálnych atribútov. Stačí, ak reálny atribút bude pre účely zobrazenia diskretizovaný, napr. jeho hodnoty zo spodnej časti rozsahu budú nahradené hodnotou 0 a hodnoty z hornej časti zase hodnotou 1 (samozrejme je možné použiť nielen dve ale aj viac hodnôt).

Dynamiku evolučného vývoja pohybu jedincov v priestore prehľadávania možno zachytiť prostredníctvom animovanej sekvencie “snímok”, pričom každá snímka zobrazuje pokrytie priestoru prehľadávania jedincami jednej generácie. Použitím takejto sekvencie, zobrazujúcej chovanie sa populácie jedincov z generácie na generáciu, je možné zachytiť trendy pohybu jedincov. V prípade, že snímka je jednorozmerná (teda zobrazovanie sa deje na priamku), celú sekvenciu snímok možno zobraziť naraz v rovine. Aby bolo možné dosiahnuť časovú kontinuitu, je potrebné dodržať jednu podmienku – je možné použiť iba takú vizualizačnú metódu, ktorá toho istého jedinca zobrazuje rovnakým spôsobom vo všetkých generáciách (a teda zobrazenie jedinca nezávisí na polohe ostatných jedincov).

Mensšie nároky na vizualizáciu kladie sledovanie vhodnosti jedincov. Pri vizualizácii ich vhodnosti je každý jedinec reprezentovaný nejakým grafickým elementom, pričom skutočná hodnota vhodnosti jedinca je vyjadrená prostredníctvom nejakej charakteristiky daného grafického elementu. Veľmi jednoduchým spôsobom je použitie stĺpcového grafu, kde výška stĺpca vyjadruje vhodnosť jedinca reprezentovaného daným stĺpcom. Používajú sa však aj zložitejšie techniky.

**Hintonov diagram.** Tento spôsob vizualizácie je známy z oblasti neurónových sietí. V modifikovanej podobe je ho možné použiť aj pre vizualizáciu vhodností jedincov podľa obr. 12.4. Každý jedinec je reprezentovaný štvorcem. Veľkosť štvorca predstavuje vhodnosť daného jedinca – čím je jedinec vhodnejší, tým je daný štvorec väčší. Rovnaké štvorce reprezentujú jedince s rovnakou vhodnosťou (v rámci zvoleného stupňa detailnosti).



Obr. 12.4: Príklad Hintonovho diagramu

Táto zobrazovacia technika umožňuje reprezentovať nielen jedincov v rámci jednej generácie, ale aj v rámci generácií viacerých a tak sledovať dynamiku vývoja aktuálnych hodnôt vhodnosti. V tomto prípade každá generácia je zobrazená na samostatnom riadku (a teda obr. 12.4 poskytuje informácie o troch generáciách jedincov).

Jeden štvorec v nejakom riadku teda predstavuje informáciu o nejakom jedincovi v určitej generácii. Bolo by však chybou počítať so štvorcami ako s nejakým konkrétnym vyvíjajúcim sa jedincami (napr. štvorce na rovnakých pozíciách vo viacerých riadkoch), pretože zobrazenie nepodáva informáciu o tom, ktoré jedince prežívajú medzi generáciami a ktoré sú novo vzniknuté. Je možné vnímať iba celkovú distribúciu hodnôt vhodnosti v populácii a jej zmeny pri prechode cez viaceré generácie.

Zaujímavou je otázka usporiadania jedincov v rámci jednej generácie. Kvôli vizuálnemu efektu sa používa usporiadanie jedincov podľa vhodnosti, čo pozorovateľovi poskytuje určitý stupeň kontinuity v rámci diagramu – hoci sa konkrétne jedince v populácii môžu meniť dosť radikálne, ich vhodnosti sa menia v rastúcom trende (aj keď môže dochádzať k dočasným depresiám – závisí to na použitej metóde tvorby nových generácií). Vizuálne diagram vyzerá tak, ako by sa z jednej strany doň vlievali veľké štvorce a vyláčali z neho štvorce malé.

Ak vopred nie je známa hodnota maximálnej vhodnosti, ktorú jedinec môže dosiahnuť, môže v neskorších generáciách (v rámci jednej generácie nie je problémom zistiť maximálnu hodnotu vhodnosti v populácii) nastať situácia, keď jedinec by mal byť reprezentovaný štvorcami väčším ako je vyhradený priestor v zobrazení. V tomto prípade je nutné zmeniť mierku a celý obraz prekresliť podľa nej.

Alternatívnou možnosťou je pre indikáciu vhodnosti nepoužívať štvorce rôznych veľkostí, ale každého jedinca reprezentovať rovnako veľkým štvor-



com a hodnotu vhodnosti indikovať farbou alebo textúrou.

Nie je nutné polohu jedincov a ich vhodnosť vždy zobrazovať oddelene, ale je možné obe zobrazenia spojiť. V tomto prípade sa zvyčajne jedince zobrazia v rovine a vhodnosť je pridaná ako ďalší rozmer, ktorý sa najčastejšie zobrazuje farbou.

## 12.3 Vizualizácia atribútov

Aj keď tieto metódy poskytujú najpodrobnejšie informácie, práve kvôli vysokému stupňu (prílišnej) detailnosti sú využívané zriedkavejšie než predchádzajúce skupiny metód.











Aj keď je možné sledovať hodnoty atribútov s väzbou na konkrétnych jedincov, takéto sledovanie sa používa buď iba v špeciálnych prípadoch (napr. pri detailnom skúmaní lokálneho účinku rôznych genetických operátorov) alebo sa viaže iba na jedného (zvyčajne najlepšieho) jedinca v každej generácii. Na vizualizáciu uvedeného druhého prípadu stačí dvojrozmerný graf hodnôt v závislosti od čísla generácie. Aby hodnoty všetkých atribútov bolo možné zobrazovať v tom istom grafe, sú tieto hodnoty normované do intervalu  $< 0, 1 >$  [35]. Pre skúmanie činnosti algoritmov v globálnom pohľade sa však konkrétne jedince nevizualizujú.

Ako jedna z najjednoduchších metód v rámci tejto skupiny sa často používa metóda (často pod rôznymi názvami v závislosti od toho, akú vlastnosť je potrebné vizualizovať), pri ktorej sa zobrazuje obrázok matice, reprezentujúcej nejakú vlastnosť hodnôt atribútov. Každé pole tejto matice reprezentuje jednu z možných hodnôt niektorého z atribútov (a teda metóda je použiteľná iba pre nominálne atribúty; nemožno ju použiť pre atribúty s príliš veľkým resp. nekonečným počtom možných hodnôt). Samotná vlastnosť je indikovaná grafickým elementom, umiestneným vo vnútri príslušného poľa. Ukážka takéhoto zobrazenia je na obr. 12.5.

Aj keď všetky atribúty na obrázku sú binárne, nie je problémom zobraziť viacero hodnôt (a dokonca rôzne atribúty môžu nadobúdať rôzne počty možných hodnôt). Hodnotu sledovanej vlastnosti hodnôt atribútov je možné zobraziť ako veľkosť nejakého grafického elementu (prípád na obrázku), numerickou hodnotou, farbou v rámci zvolenej farebnej škály ap.

Vlastnosťou, ktorá je vizualizovaná takýmto spôsobom, najčastejšie býva:

- frekvencia,
- vhodnosť.

Hodnota					
prvá					
druhá					
	prvý	druhý	tretí	štvrtý	piaty
	Atribúty				

Obr. 12.5: Matica vlastností hodnôt atribútov

V prípade frekvencie hodnôt stĺpec matice vlastne reprezentuje distribúciu tých hodnôt, ktoré daný atribút môže nadobúdať. V prevažnej miere sa distribúcia uvažuje iba pre jednu generáciu (keď je jednoduchou identifikácia konvergenencie populácie s ohľadom na hodnotu jedného alebo viacerých atribútov), avšak je možná aj aplikácia na viacero generácií naraz.

V prípade vhodnosti je potrebné danú hodnotu najprv odvodiť z dostupných údajov, nakoľko vhodnosti sú priraďované jedincom a nie hodnotám atribútov. Preto sa v tejto úlohe používa priemerná vhodnosť tých jedincov, v ktorých daný atribút nadobúda danú hodnotu. Opäť možno záujem sústrediť iba na jednu generáciu, avšak často sa takýmto spôsobom zohľadňujú všetky generácie. Vhodnosť nejakej hodnoty nejakého atribútu totiž reprezentuje odhad vhodnosti jednotlivých stavebných blokov – a čím viac jedincov sa zahrnie do tohto odhadu, tým je väčšia šanca vyššej presnosti tohto odhadu.

Podobne ako pri vizualizácii jedincov, aj pri vizualizácii hodnôt je možné v jednom zobrazení uvažovať iba informácie o jednej generácii a vytváraním sekvencií snímok pre rôzne generácie možno vizualizáciu obohatiť aj o časové hľadisko.

---

Časť VI  
Čo je čo



## Kapitola 13

# Varianty evolučného algoritmu

V predchádzajúcich kapitolách boli uvedené rozličné prvky, z ktorých je možné vytvoriť evolučný algoritmus. Rôznymi variáciami týchto prvkov je možné získať pomerne veľký počet alternatívnych podôb algoritmu. Pritom nebolo rozlišované, či tá-ktorá konkrétna kombinácia prvkov je hypotetická a ešte iba čaká na svoje vyskúšanie alebo či už bola vyskúšaná v teoretickej alebo praktickej rovine.

V skutočnosti niektoré kombinácie prvkov sú tak známe, že daná kombinácia má svoje vlastné pomenovanie a je prezentovaná (najmä autormi tej kombinácie) ako samostatný algoritmus. Aj keď predkladaný text takéto kombinácie nepovažuje sa samostatné algoritmy ale iba za jednotlivé inštancie evolučného algoritmu, pre úplnosť tieto známe inštancie sú uvedené v nasledujúcom texte (uvedené sú typické tvary týchto inštancií).

### 13.1 Klasické prístupy

V minulosti sa vyskytlo niekoľko pokusov využiť evolúciu ako základ pre tvorbu algoritmov – či už s cieľom modelovať a študovať evolučné procesy alebo riešiť praktické úlohy. Mnohé z nich skončili neúspechom alebo mali iba malú odozvu. Tri takéto pokusy si však vybojovali väčšiu či menšiu popularitu a prežili (niekoľko desiatok rokov) až dodnes:

- evolučné stratégie,
- evolučné programovanie,

- genetické algoritmy.

Tieto tri smery sa dlhú dobu vyvíjali izolovane, nevediac o sebe navzájom. K prvým kontaktom komunit, stojacich za týmito prístupmi, začalo dochádzať až na začiatku deväťdesiatych rokov dvadsiateho storočia.

### 13.1.1 Evolučná stratégia

Tento typ evolučného algoritmu vznikol na pôde Technickej univerzity v Berlíne v prvej polovici šesťdesiatych rokov minulého storočia. Podnetom bola snaha riešiť optimalizačné úlohy, ktoré nebolo možné riešiť analyticky alebo využitím tradičných optimalizačných metód. Prvé aplikácie boli spojené s optimalizáciou štruktúry a tvaru v hydrodynamickej oblasti.

Vznik evolučnej stratégie je spojený s dvomi menami: I. Rechenberg a H. P. Schwefel. Schwefel vo svojej diplomovej práci (1965) prvýkrát simuloval rôzne verzie stratégie na počítači. Aj keď výsledná dvojčlenná evolučná stratégia ešte nepoužívala populačný princíp, potomok bol vytváraný z rodiča aplikáciou mutačného operátora a súťažil s ním o prežitie. Populačný princíp bol prvýkrát zavedený Rechenbergom v tvare mnohočlennej stratégie, ktorá ho však aplikovala iba na rodičov ale nie na potomkov. Rozšírenie populačného princípu aj na potomkov bolo zavedené onedlho na to Schwefelom.

Nasledujúci opis algoritmu je založený na plne adaptívnom variante uvedenom v [4].

**Reprezentácia.** Usporiadaná lineárna štruktúra pevnej dĺžky používajúca reálne kódovanie.

Jedinec kóduje nielen riešenie ale aj parametre algoritmu. Ak riešenie problému je reprezentované pomocou  $l$  atribútov resp. ich hodnôt, potom jedinec pozostáva z  $l(l+3)/2$  reálnych čísel, z ktorých

- $l$  čísel reprezentuje priamo hodnoty atribútov  $h_1, \dots, h_l$  (v princípe to sú ľubovoľné reálne hodnoty),
- $l$  čísel reprezentuje smerodajné odchýlky  $\sigma_1, \dots, \sigma_l$  (kladné reálne čísla),
- a  $l(l-1)/2$  čísel reprezentuje rotačné uhly  $\alpha_{1,2}, \dots, \alpha_{l-1,l}$  (čísla z intervalu  $\langle -\pi, \pi \rangle$ ).

**Vhodnosť.** V úlohe vhodnosti sa použije priamo hodnota účelovej funkcie bez akejkoľvek úpravy.

**Selekcia.** Selekcia je realizovaná ako uchovávajúca selekcia bez produkcie selekčného tlaku. Jedinca sú vyberané do úlohy rodičov náhodným spôsobom s rovnakou pravdepodobnosťou.

**Genetické operátory.** Mutácia je hlavným operátorom pre hodnoty atribútov. Rekombinácia je považovaná za operátor nutný pre samoadaptatívny proces parametrov stratégie.

Mutačný operátor je realizovaný ako trojica za sebou nasledujúcich mutačných krokov, pričom každý z nich operuje nad inou časťou jedinca:

$$\begin{aligned}\sigma'_i &= \sigma_i e^{\tau N(0,1)} e^{\tau_i N_i(0,1)} \\ \alpha'_i &= \alpha_i + \beta N_i(0,1) \\ h'_i &= h_i + \text{rot}_i(\vec{N}(\vec{0}, \vec{\sigma}'), \vec{\alpha}')\end{aligned}\tag{13.1}$$

Najprv sa určí nová hodnota smerodajných odchýlok, ktoré sú mutované multiplikatívnym spôsobom. Term založený na  $N(0,1)$  (normálnom rozdelení s nulovou strednou hodnotou a jednotkovou smerodajnou odchýlkou) pritom reprezentuje globálny faktor, zatiaľ čo term založený na  $N_i(0,1)$  zastupuje lokálny faktor, umožňujúci individuálnu zmenu každej smerodajnej odchýlky osobitne. Použitie exponenciálnej funkcie je inšpirované snahou zabezpečiť nezápornosť odchýliek. Keďže postupným násobením by sa odchýlky mohli stať príliš malými, sú algoritmicky kontrolované, aby neklesli pod nejakú stanovenú hodnotu  $\varepsilon$ .

Potom sa určuje nová hodnota rotačných uhlov, ktoré sú mutované aditívnym spôsobom. Ak sa hodnota nejakého uhla dostane mimo požadovaný interval, tak je postupne zvyšovaná resp. znižovaná o hodnotu  $2\pi$  dovtedy, pokiaľ opäť nebude v požadovaných medziach.

Nakoniec dochádza k zmene hodnôt atribútov pomocou aditívnej mutácie. Vygenerujú sa hodnoty odchýlok pre všetky atribúty, pričom pre každú z nich sa použije normálne rozdelenie pravdepodobnosti s nulovou strednou hodnotou a novou hodnotou smerodajnej odchýlky, prislúchajúcej danému atribútu. Tieto odchýlky sú upravené pomocou rotačných matíc, využívajúcich zmenené hodnoty rotačných uhlov. Napokon každá hodnota atribútu je upravená o odchýlku vyprodukovanú rotačným procesom.

Pri rekombinácii sú jednotlivé skupiny hodnôt (atribúty, smerodajné odchýlky a rotačné uhly) rekombinované nezávisle na sebe. Rekombinácia hodnôt atribútov je variantom<sup>1</sup> uniformného kríženia, keď pre každú pozíciu sa náhodne vyberá, z ktorého rodiča bude hodnota prenesená do potomka.

---

<sup>1</sup>V ES komunite je toto kríženie označované ako diskrétné kríženie.

Operátor je realizovaný ako sexuálny, využívajúci dvoch rodičov pre úplnú produkciu potomka.

Rekombinácia parametrov stratégie je realizovaná ako variant<sup>2</sup> aritmetického kríženia podľa vzťahu (8.6). Operátor je realizovaný ako panmiktický, kde počas generovania jedného potomka jeden rodič je predstavovaný tým istým selektovaným jedincom, zatiaľ čo do úlohy druhého rodiča sa vyberá nový jedinec osobitne pre každú pozíciu v štruktúre jedinca.

**Náhrada.** Charakteristickou je deterministická náhrada produkujúca selekčný tlak. V tzv. čiarkovej stratégii  $(\mu, \lambda)$  je síce k dispozícii  $\mu$  jedincov aktuálnej populácie a  $\lambda$  novo vyprodukovaných potomkov ( $\mu < \lambda$ ), avšak nová generácia sa formuje iba z potomkov – je tvorená príslušným počtom najlepších potomkov.

**Inicializácia.** Jednotlivé časti jedinca sa inicializujú rozdielnym spôsobom. Pri inicializácii hodnôt atribútov sa z jedného štartovacieho bodu (môže byť napr. volený užívateľom algoritmu alebo náhodne vybraný v prípustnej oblasti použitím uniformnej distribúcie pravdepodobnosti) vytvorí pomocou mutácie príslušný počet jedincov.

Hodnoty parametrov stratégie sa inicializujú nastavením na doporučené hodnoty.

**Podmienka ukončenia.** Stratégia je ukončená, ak rozdiel medzi najlepšou a najhoršou hodnotou vhodnosti v aktuálnej populácii klesne pod stanovený prah.

**Typické hodnoty.** Pre parametre, ovplyvňujúce mutačný proces, sa doporučujú hodnoty

$$\tau \propto 1/\sqrt{2l} \quad \tau_i \propto 1/\sqrt{2\sqrt{l}} \quad \beta \approx 0.0873 \quad (13.2)$$

Obyčajne konštanty proporcionality sú rovné jednej a doporučená hodnota pre  $\beta$  zodpovedá  $5^\circ$  v radiánoch.

Smerodajné odchýlky sa inicializujú na hodnoty približne rovné  $\Delta_i/\sqrt{l}$ , kde čitateľ reprezentuje odhad vzdialenosti (v smere  $i$ -tej osi) medzi štartovacím bodom a optimom. Experimenty však dokazujú, že vhodným nastavením môže byť aj hodnota 3.0 v prípade, neznámych funkcií.

---

<sup>2</sup>Označované ako medziľahlé kríženie.



Pre zabezpečenie vhodného selekčného tlaku sa používa pomer  $\mu/\lambda \approx 1/7$ , pri zachovaní dostatočne veľkej populácie napr.  $\mu = 15$ .

### 13.1.2 Evolučné programovanie

Evolučné programovanie vzniklo v USA a jeho začiatky boli spojené s menami L. J. Fogel, A. J. Owens a M. J. Walsh. Vo svojej práci sa zameriavali na evolúciu automatov pre účely predikcie postupností – schopnosť predpovedať budúce stavy a reagovať na ne považovali za podmienku inteligencie. Evolučný prístup používali na tvorbu tabuliek prechodov stavov konečných automatov. Ich prístup, spoločne publikovaný v roku 1966, už využíval populačný princíp, bol založený na produkcii selekčného tlaku a mutačných zmenách. Keďže sa snažili napodobniť evolúciu druhov, rekombináciu ignorovali.

Idea si nezískala mnoho pozornosti a ostala dlhý čas v zabudnutí až do konca osemdesiatych rokov, keď D. B. Fogel (syn jedného zo zakladateľov) rozšíril tento prístup pre použitie v aplikáciách, ktoré vyžadovali optimalizáciu spojitých parametrov. Vo svojej dizertačnej práci (1992) prezentoval niekoľko variantov vrátane variantu, zahŕňajúceho samoadaptáciu parametrov algoritmu.

Nasledujúci opis algoritmu je založený na variante známom ako meta-evolučné programovanie [3].

**Reprezentácia.** Usporiadaná lineárna štruktúra pevnej dĺžky používajúca reálne kódovanie.

Jedinec kóduje nielen riešenie ale aj parametre algoritmu. Ak riešenie problému je reprezentované pomocou  $l$  atribútov resp. ich hodnôt, potom jedinec pozostáva z  $2l$  reálnych čísel, z ktorých

- $l$  čísel reprezentuje priamo hodnoty atribútov  $h_1, \dots, h_l$ ,
- a  $l$  čísel reprezentuje variancie  $\sigma_1^2, \dots, \sigma_l^2$ .

Každá z hodnôt atribútov je obmedzená na nejaký podinterval reálnych čísel  $\langle \min(v_i), \max(v_i) \rangle$ . Toto obmedzenie je však platné iba vo fáze inicializácie, počas behu algoritmu ho operátory nemusia dodržiavať – priestor prehľadávania nie je obmedzený.

Podobne aj variancie sú obmedzené na interval  $\langle 0, k \rangle$ , kde  $k > 0$ , iba počas inicializácie. Počas realizácie algoritmu môžu variancie nadobúdať ľubovoľné kladné reálne hodnoty.

**Vhodnosť.** Vhodnosť je odvodená z hodnoty účelovej funkcie. Hodnota tejto účelovej funkcie sa najprv zmení pomocou nejakej náhodnej alternácie a následne môže byť škálovaná do kladných hodnôt. Táto transformácia môže byť použitá aj s nulovou alternáciou hodnoty účelovej funkcie.

**Selekcia.** Selekcia je realizovaná ako elitistická selekcia bez produkcie selekčného tlaku. Jedince sú vyberané do úlohy rodičov deterministickým spôsobom – každý jedinec sa stáva rodičom.

**Genetické operátory.** Mutácia je jediným operátorom použitým v tomto prístupe. Rekombinácia nie je použitá v žiadnej podobe.

Mutačný operátor je realizovaný ako dvojica za sebou nasledujúcich mutačných krokov, pričom každý z nich operuje nad inou časťou jedinca:

$$\begin{aligned}h'_i &= h_i + \sqrt{\sigma_i^2} N_i(0, 1) \\ (\sigma_i^2)' &= \sigma_i^2 + \sqrt{\xi \sigma_i^2} N_i(0, 1)\end{aligned}\tag{13.3}$$

Najprv dochádza k zmene hodnôt atribútov pomocou aditívnej mutácie. Zmena hodnoty sa určí prostredníctvom normálnej mutácie s využitím aktuálnej hodnoty variancie príslušnej k danej mutovanej hodnote<sup>3</sup>.

Potom sa určí nová hodnota smerodajných odchýlok, ktoré sú mutované aditívnym spôsobom, taktiež využitím normálnej mutácie, pričom použitie škálovacieho parametra  $\xi$  je inšpirované snahou o zabezpečenie pozitívnosti hodnôt variancií.

Keďže však postupným mutovaním by sa variancie mohli stať zápornými alebo nulovými, sú algoritmicky kontrolované, aby neklesli pod nejakú stanovenú hodnotu  $\varepsilon$ .

**Náhrada.** Používa sa kombinácia pravdepodobnostne deterministického výberu (produkujúceho selekčný tlak) jedincov novej generácie, pričom výsledná náhrada má charakter elitistickej náhrady.

Keďže každý rodič vyprodukoval jedného potomka, je k dispozícii  $\mu$  jedincov aktuálnej populácie a  $\mu$  novo vyprodukovaných potomkov. Výber je založený na vhodnosti – avšak nie na pôvodnej vhodnosti jedincov, ale na vhodnosti premapovanej. Vhodnosť založená na účelovej funkcii je nahradená súťažnou vhodnosťou. Používa sa variant plnej súťaže – ibaže jedinec

---

<sup>3</sup>Variant EP bez samoadaptívneho upravovania variancií hodnotu  $i$ -teho atribútu upravuje o  $\sqrt{\Phi} N_i(0, 1)$ , teda využíva vhodnosť jedinca – v tomto prípade vhodnosti musia byť škálované do kladných hodnôt.

nesúťaží so všetkými ostatnými  $2\mu - 1$  jedincami, ale pre každého jedinca sa náhodne vyberie (spomedzi pôvodných jedincov aj novo vytvorených potomkov)  $q$  jedincov, s ktorými bude súťažiť. Výsledná vhodnosť jedinca bude daná počtom tých jedincov v súťaži, nad ktorými zvíťazil.

Podľa hodnoty súťažnej vhodnosti sú všetky jedince zoradené podľa tejto vhodnosti a následne  $\mu$  najlepších vytvorí nasledujúcu generáciu.

**Inicializácia.** Obe časti jedinca sa inicializujú podobným spôsobom. Prvotná populácia je vytvorená náhodným vzorkovaním každej pozície v rámci intervalu povolených hodnôt pre danú pozíciu. Pri tomto vzorkovaní sa používa uniformná distribúcia pravdepodobnosti.

**Podmienka ukončenia.** Pevne stanovený počet generácií. Po jeho uplynutí sa evolučný proces zastaví.

**Typické hodnoty.** Pre zabezpečenie vhodného vzorkovania plochy vhodnosti sa používa  $\mu = 200$ .

Hodnoty atribútov sa inicializujú z intervalu  $\langle -50, 50 \rangle$ , hodnota variancií je zhora ohraničená hodnotou  $k = 25$ .

Na jednej strane pre zabezpečenie dostatočného selekčného tlaku, ale na strane druhej pre ponechanie dostatočného priestoru pre náhodu, sa pri určovaní súťažnej vhodnosti používa súboj jedinca s desiatimi ďalšími jedincami ( $q = 10$ ).

Pre nastavenie parametra  $\xi$ , ovplyvňujúceho mutačný proces, sa doporčuje hodnota 6.0.

### 13.1.3 Genetický algoritmus

Genetický algoritmus sa zrodil na pôde Michiganskej univerzity ako výsledok práce J. Hollanda. Jeho práca (zosumarizovaná knižne v 1975) bola štartovacím bodom všetkých ďalších prác na tému genetických algoritmov.

Hollandov cieľ nebol vytvoriť algoritmus pre riešenie konkrétnych úloh, ale formálne študovať fenomén adaptácie tak, ako sa vyskytuje v prírode, a vytvoriť spôsob, ako tento mechanizmus prirodzenej adaptácie importovať do umelých (počítačových) systémov. Holland vytvoril všeobecnú teóriu adaptívnych systémov, ktoré komunikujú so svojím prostredím prostredníctvom svojich binárnych detektorov. Prezentoval algoritmus (vtedy ešte nazvaný adaptívny a reprodukčný plán) ako abstrakciu biologickej evolúcie.

K. A. De Jong vo svojej dizertačnej práci (1975) položil základy pre použitie genetického algoritmu pre riešenie reálnych úloh. Pretože genetický

algoritmus vychádza z adaptačných procesov v prírode, jedince sú binárne reťazce a pre tvorbu potomkov sa používajú reprodukčné postupy blízke prirodzeným.

Nasledujúci opis algoritmu je založený na základnom variante uvedenom v [14].

**Reprezentácia.** Usporiadaná lineárna štruktúra pevnej dĺžky používajúca binárne kódovanie.

Jedinec kóduje priamo iba riešenie bez akýchkoľvek dodatočných informácií. Ak riešenie problému je reprezentované pomocou  $n$  atribútov resp. ich hodnôt, a pre zakódovanie hodnoty  $i$ -teho atribútu je potrebné použiť  $l_i$  bitov, potom jedinec pozostáva z

$$l = \sum_{i=1}^n l_i \quad (13.4)$$

bitových pozícií, vytvárajúcich lineárnu štruktúru jedinca.

**Vhodnosť.** Vhodnosť je odvodená z hodnoty účelovej funkcie. V závislosti od charakteru účelovej funkcie a jej oboru hodnôt môže byť potrebné pre určovanie vhodnosti vykonať nasledovné transformácie:

- ak účelová funkcia vyžaduje minimalizáciu, tak je nutné minimalizačný problém transformovať na maximalizačný,
- ak obor hodnôt účelovej funkcie zahŕňa aj záporné hodnoty, tak je ho potrebné transformovať na interval nezáporných hodnôt.

**Selekcia.** Selekcia je realizovaná ako uchováajúca selekcia s produkciou selekčného tlaku.

Ako selekčná metóda sa používa stochastické vzorkovanie s náhradou<sup>4</sup>. Pre každého jedinca sa určí pravdepodobnosť jeho selekcie použitím proporcionálnej selekcie. Následne sa každému jedincovi vypočíta očakávaný počet rodičov a ten sa realizuje danou selekčnou metódou.

Selekčná metóda nepracuje priamo s vhodnosťou jedincov, tá je najprv premapovaná na hodnoty, zabezpečujúce dostatočný selekčný tlak počas celého behu algoritmu. Samotné premapovanie je realizované lineárnym škálovaním tak, aby maximálna hodnota vhodnosti v aktuálnej populácii bola premapovaná na dvojnásobok hodnoty, na ktorú je premapovaná priemerná

---

<sup>4</sup>Modernejšie verzie používajú vzorkovacie metódy s menším rozpätím.

vhodnosť populácie. Ak by toto viedlo k situácii, že minimálna vhodnosť v populácii je premapovaná na zápornú hodnotu, tak sa transformačný vzťah upraví do tvaru, umožňujúceho premapovanie minimálnej vhodnosti na nulovú hodnotu.

**Genetické operátory.** Genetický algoritmus používa oba typy operátorov – mutáciu pre zmenu informácie kódovanej v rodičoch a kríženie pre kombinovanie informácií obsiahnutých v rôznych jedincoch. Tieto operátory nie sú chápané ako rovnocenné – primárnym operátorom je kríženie, zatiaľ čo mutácia zastáva sekundárnu pozíciu.

Mutačný operátor je realizovaný ako doplnková mutácia s pravdepodobnosťou  $p_m$ .

Krížiaci operátor je realizovaný ako jednobodové kríženie s pravdepodobnosťou  $p_c$ .

Oba operátory sú kombinované za sebou, pričom najprv sa na dvoch selektovaných rodičov aplikuje krížiaci operátor a na výsledky kríženia je následne aplikovaný mutačný operátor. Vďaka tomu, že operátory sú aplikované na pravdepodobnostnom princípe, novo vygenerovaný potomok môže ale aj nemusí byť kombináciou oboch rodičov a súčasne môže alebo nemusí byť nositeľom novej informácie, ktorá sa nenachádzala ani v jednom z rodičov. V extrémnom prípade dokonca môže byť presnou kópiou jedného z rodičov.

**Náhrada.** Z hľadiska produkcie selekčného tlaku je náhrada menej významným blokom algoritmu ako blok selekcie. Jednou z možností je generačná náhrada, pričom najčastejšie sa generuje iba toľko potomkov, koľko je ich potrebných pre vytvorenie novej generácie.

**Inicializácia.** Prvotná populácia je vytvorená náhodným vzorkovaním každej pozície. Pri tomto vzorkovaní sa používa rovnaká pravdepodobnosť pre obe možné hodnoty.

**Podmienka ukončenia.** Pevne stanovený počet generácií. Po jeho uplynutí sa evolučný proces zastaví.

**Typické hodnoty.** Pre zabezpečenie vhodného vzorkovania plochy vhodnosti sa používa  $\mu \approx 100$ , avšak je možné sa stretnúť aj s menším resp. rádovo väčším počtom.

Pre nastavenie pravdepodobnosti kríženia  $p_c$  sa používa hodnota z intervalu  $\langle 0.6, 1.0 \rangle$ .

Pre pravdepodobnosť mutácie  $p_m$  sa používa jednoduché pravidlo – pravdepodobnosť sa nastaví tak, aby v priemere v potomkovi došlo k jednej zmene ( $p_m = 1/l$ ).

## 13.2 Novšie prístupy

Klasické alternatívy evolučných algoritmov inšpirovali vznik ďalších alternatívnych algoritmov, využívajúcich prvky evolučných algoritmov. Príčiny ich vzniku boli rôznorodé. Niektoré vznikli ako pokus prispôsobiť klasický algoritmus riešeniu úloh z nejakej oblasti, iné ako následok snahy nahradiť niektorý z princípov, na ktorých spočívali klasické varianty, iným princípom (či už prírodným alebo umelým) a niektoré ako výsledok pokusu zohľadniť vzťahy a závislosti, ktoré v klasických variantoch zohľadnené neboli.

### 13.2.1 Genetické programovanie

Pokus využiť stromové štruktúry pre generovanie funkcií, ktoré by boli spracovávané genetickými operátormi, bol publikovaný N. L. Cramerom (1985). Neskôr sa hnacím motorom vývoja tohto prístupu stal J. R. Koza. Prístup sa zameriava nie priamo na riešenie úloh ale na tvorbu programov, ktoré by danú úlohu dokázali vyriešiť.

Nasledujúci opis algoritmu je založený na [22].

**Reprezentácia.** Jedinice majú tvar stromových štruktúr, reprezentujúcich hierarchické programy. Tieto štruktúry nemajú pevne daný rozmer, ale ich veľkosť je premenlivá. Je však obmedzená maximálna dĺžka vetiev (hĺbka stromu).

Na pozície listových uzlov stromu je možné umiestňovať terminálne symboly, ktoré zvyčajne reprezentujú atómy alebo konštanty. Atómy zastupujú hodnoty, ktoré sú z hľadiska vyvíjaného programu externé, konštanty môžu reprezentovať napr. číselné či logické hodnoty.

Pozície nelistových uzlov obsadzujú funkcie. Árnosť funkcie rozhoduje o tom, koľko potomkov bude mať uzol, reprezentujúci danú funkciu. Vzťah medzi nadradeným a podradeným uzlom tak reprezentuje vzťah medzi funkciou a jej argumentom, ktorým môže byť premenná/konštanta (potomok je listový uzol) alebo volanie inej funkcie (potomok nie je listovým uzlom).

Konkrétne použitá množina funkcií a terminálnych symbolov závisí od aktuálneho problému, pre riešenie ktorého je potrebné vytvoriť program.

**Vhodnosť.** Vhodnosť môže byť určovaná pomocou množiny príkladov, ktoré slúžia pre vyhodnotenie programu reprezentovaného jedincom. Vhodnosť potom môže reprezentovať chybu, ktorá vzniká pri aplikácii programu na dané príklady. Táto chyba môže byť následne transformovaná a normalizovaná.

**Selekcia.** Selekcia je realizovaná ako uchováajúca selekcia s produkciou selekčného tlaku.

Ako selekčná metóda sa používa niektorý z variantov stochastického zorkovania. Pravdepodobnosti selekcie môžu byť určené proporcionálnou selekciou. Pred tým ešte vhodnosť môže byť premapovaná pomocou zotriedenia jedincov.

**Genetické operátory.** Genetické programovanie používa oba typy operátorov – mutáciu pre zmenu informácie kódovanej v rodičoch ako aj kríženie pre kombinovanie informácií obsiahnutých v rôznych jedincoch. Kríženie je považované za hlavný operátor, mutácia je označovaná ako sekundárny operátor.

Mutačný operátor môže byť realizovaný ako náhodná náhrada podstromu. Najprv je náhodne vyberaný mutačný bod. Následne sa z jedinca odstráni podstrom, ktorého koreň je predstavovaný daným mutačným bodom. Záverečne sa k jedincovi v mieste mutačného bodu pripojí náhodne generovaný podstrom.

Krížiaci operátor je realizovaný ako vzájomná výmena tzv. krížiacich fragmentov (podstromov). Vzhľadom k tomu, že rodičia nemusia byť rovnako veľké jedince, je potrebné krížiaci bod generovať osobitne pre každého rodiča, a teda vymieňané fragmenty môžu pochádzať z rôznych častí rodičov. Preto aj v prípade, že obaja rodičia sú rovnaké jedince, potomkovia rovnakí nemusia byť.

Kríženie (a mutácia) sa dejú iba s nejakou pravdepodobnosťou – teda existuje aj pravdepodobnosť, že rodič bude transformovaný na potomka bez toho, aby došlo k jeho zmene.

**Náhrada.** Jednou z možností je generačná náhrada, pričom sa generuje iba toľko potomkov, koľko je ich potrebných pre vytvorenie novej generácie.

**Inicializácia.** Prvotná populácia je tvorená náhodným generovaním stromových štruktúr zhora nadol.

Najprv sa na pozíciu koreňového uzla stromu vyberá jedna z funkcií. V závislosti na árnosti vybratej funkcie sa pod koreňovým uzlom vytvorí príslušný počet uzlov. Tento postup sa opakuje vždy pri výbere funkcie na nejakú pozíciu v strome. Ak sa na pozíciu vyberie terminálny symbol, daný uzol sa stáva listovým a generovanie v danej vetve stromu končí.

Je možné generovať plné stromy, ktorých dĺžka v každej vetve dosahuje maximálne dovolenú dĺžku, alebo stromy s premenlivou dĺžkou.

**Podmienka ukončenia.** Pevne stanovený počet generácií. Po jeho uplynutí sa evolučný proces zastaví. Alternatívne je možné použiť doménovo závislý indikátor úspechu, ktorý signalizuje dosiahnutie stavu vhodného pre ukončenie evolúcie.

**Typické hodnoty.** Oproti ostatným metódam sa používa väčšia populácia. Ako vhodná veľkosť, s ktorou je možné začať experimentovať, sa doporučuje 500. Komplexnejšie problémy vo všeobecnosti vyžadujú ešte väčšiu populáciu.

Pri generovaní polohy krížiaceho bodu sa odporúča tento bod v 90% prípadov umiestňovať rovnomerne medzi nelistovými uzlami a v 10% prípadov zase rovnomerne umiestňovať medzi listovými uzlami stromu. Takáto distribúcia podporuje rekombináciu väčších štruktúr.

### 13.2.2 Šľachtiteľský algoritmus

Šľachtiteľský algoritmus<sup>5</sup> vznikol na začiatku deväťdesiatych rokov a bol predstavený H. Mühlenbeinom a D. Schlierkamp-Voosenom. Nesnaží sa modelovať prirodzenú evolúciu ale evolúciu umelú – racionálnu selekciu realizovanú šľachtiteľmi pri šľachtení zvierat. Stavia na znalostiach nahromadených v oblasti kvantitatívnej genetiky.

Nasledujúci opis algoritmu je založený na [32].

**Reprezentácia.** Usporiadaná lineárna štruktúra pevnej dĺžky používajúca reálne kódovanie.

Jedinec kóduje priamo iba riešenie bez akýchkoľvek dodatočných informácií.

**Vhodnosť.** Vhodnosť je odvodená z hodnoty účelovej funkcie. Nie je potrebné vykonať žiadne dodatočné transformácie.

---

<sup>5</sup>Breeder Genetic Algorithm



**Selekcia.** Selekcia je zdrojom selekčného tlaku. Je realizovaná ako selekcia s vymieraním sprava.

Ako selekčná metóda sa používa orezanie s následným stochastickým vzorkovaním vhodnejšej časti populácie. Jedinca sú do úlohy rodičov vyberané a párované náhodne, pričom však nie je dovolené párovať dve kópie toho istého jedinca.

**Genetické operátory.** Šľachtiteľský algoritmus používa oba typy operátorov pre ich synergický efekt – mutáciu pre zmenu informácie kódovanej v rodičoch a kríženie pre kombinovanie informácie obsiahnutej v rôznych jedincoch.

Mutačný operátor je realizovaný ako Mühlenbeinova mutácia. Umožňuje meniť hodnotu premennej iba v diskretných krokoch, následkom čoho je schopný pracovať iba s presnosťou  $2^{-15}$  - násobku mutačného rozsahu. Mutácia každej hodnoty sa deje s pravdepodobnosťou  $p_m$ .

Krížiaci operátor je realizovaný ako uniformné kríženie<sup>6</sup> s rovnakou pravdepodobnosťou prenosu hodnoty z prvého alebo druhého rodiča do vytváraného potomka. Inou alternatívou je použitie aritmetického kríženia podľa (8.6), pričom však  $\chi$  je náhodné číslo z intervalu  $\langle -0.25, 1.25 \rangle$ . Pritom hodnotu kombinačného parametra  $\chi$  je možné generovať osobitne pre každú z pozícií alebo je možné použiť tú istú hodnotu pre všetky pozície.

Oba operátory sú kombinované pre spoločné vytváranie potomka, pričom nie sú definované pravdepodobnosti ich aplikovania – oba operátory sú aplikované vždy. Avšak vzhľadom na to, že každá pozícia je mutovaná iba s určitou pravdepodobnosťou, je možné že hodnoty jedinca nebudú zmenené mutačným operátorom.

**Náhrada.** Je realizovaná ako generačná náhrada, pričom sa generuje iba toľko potomkov, koľko je ich potrebných pre vytvorenie novej generácie.

Tento mechanizmus je obohatený o elitistické chovanie – najlepší jedinec, nájdený do danej chvíle, ostáva v populácii aj naďalej.

**Inicializácia.** Prvotná populácia môže byť vytvorená napríklad náhodným vzorkovaním každej pozície s uniformnou distribúciou pravdepodobnosti.

**Podmienka ukončenia.** Možno použiť napríklad pevne stanovený počet generácií. Po jeho uplynutí sa evolučný proces zastaví.

---

<sup>6</sup>V jazyku používanom autormi algoritmu označované ako diskretná rekombinácia.

**Typické hodnoty.** V prípade, že rozsah tých hodnôt, ktoré sa môžu vyskytnúť na nejakej  $j$ -tej pozícii jedinca, je obmedzený na nejaký interval  $\langle \max(v_j), \min(v_j) \rangle$ , tak ako mutačný rozsah pre mutačnú funkciu zmeny sa používa hodnota  $0.1(\max(v_j) - \min(v_j))$ . Pokiaľ takto určená hodnota spôsobuje neprijateľnú presnosť mutačného operátora alebo rozsah hodnôt na danej pozícii nie je obmedzený, je možné hodnotu mutačného rozsahu nastaviť v závislosti od požadovanej presnosti.

Pravdepodobnosť mutácie jednej hodnoty  $p_m$  sa nastavuje na takú hodnotu, aby v priemere bola zmenená jedna pozícia v každom mutovanom jedincovi.

Pre prah orezania sa doporučuje hodnota z intervalu  $\langle 0.1, 0.5 \rangle$ . Aby vďaka produkovanej selekčnej intenzite algoritmus neskonvergoval predčasne, pre veľkosť populácie sa doporučuje<sup>7</sup> použiť aspoň 50 (pre  $T = 0.5$ ) až 100 (pre  $T = 0.1$ ) jedincov.

### 13.2.3 Diferenciálna evolúcia

Za vznikom prvých verzií tohto algoritmu v polovici deväťdesiatych rokov stáli K. Price a R. Storn. Spočiatku hlavným faktorom vývoja bola snaha využiť princíp žihania v simulovanej evolúcii. Po náhrade binárnej reprezentácie reálnou a logických operácií aritmetickými vektorovými operáciami sa tento princíp ukázal nadbytočný a bol odstránený. A tak sa hlavným znakom algoritmu stalo používanie vektorových diferencií.

Nasledujúcu opis je založený na [13] a je známy ako variant DE/rand/1.

**Reprezentácia.** Usporiadaná lineárna štruktúra pevnej dĺžky používajúca reálne kódovanie.

Jedinec kóduje priamo iba riešenie bez akýchkoľvek dodatočných informácií.

**Vhodnosť.** Vhodnosť je odvodená z hodnoty účelovej funkcie. Nie je potrebné vykonať žiadne dodatočné transformácie.

**Selekcia.** Selekcia nie je zdrojom selekčného tlaku. Je realizovaná ako elitistická selekcia.

Ako selekčná metóda sa používa kombinácia deterministického výberu (deterministický variant orezania s prahom orezania rovným jednej) a stochastického výberu (náhodný výber s rovnakou pravdepodobnosťou výberu

---

<sup>7</sup>Na základe experimentov s unimodálnou funkciou.

pre každého jedinca). Každý jedinec z aktuálnej populácie sa povinne stáva rodičom a k tomuto jedincovi sú pridané ešte tri jedince náhodne vybrané z populácie. Každá takáto štvorica umožní vyprodukovať jedného nového potomka.

**Genetické operátory.** Algoritmus používa iba jeden typ operátora - rekombinačný operátor. V skutočnosti sa súčasne používajú dva varianty rekombinačného operátora<sup>8</sup>:

- diferenciálny operátor produkujúci z troch rodičov jedného potomka,
- uniformné kríženie kombinujúce dvoch jedincov.

Oba operátory sú kombinované v sekvenčnom usporiadaní. Najprv sa aplikuje diferenciálny operátor (aplikuje sa na tri náhodne selektované jedince) a vyprodukuje jedinca, ktorý sa označuje ako šumový vektor.

Následne sa použije uniformné kríženie (aplikuje sa na štvrtého, deterministicky vybraného rodiča a na šumový vektor) a vyprodukuje výsledného potomka.

Oba operátory sú vždy aplikované, neexistuje žiadne stochastické určovanie, či sa operátor použije alebo nie.

**Náhrada.** Je realizovaná ako deterministický turnaj, pričom novo generovaný potomok súťaží s jedným zo svojich rodičov (tým, čo bol selektovaný deterministickým spôsobom). Ten jedinec, ktorý má lepšiu vhodnosť, sa stáva členom nasledovnej generácie.

**Inicializácia.** Prvotná populácia je vytvorená náhodným vzorkovaním každej pozície. Pri tomto vzorkovaní sa vyberá jedna z možných hodnôt pri použití uniformnej distribúcie pravdepodobnosti.

**Podmienka ukončenia.** Pevne stanovený počet generácií. Po jeho uplynutí sa evolučný proces zastaví.

**Typické hodnoty.** Parameter diferenciálneho operátora sa zvykne voliť z intervalu  $< 0, 2 >$ , pričom hodnota 0 vlastne eliminuje pôsobenie operátora.

Pravdepodobnosti, použité v uniformnom krížení, sa nastavujú tak, aby dochádzalo k zmene – teda aby potomok nebol totožný s rodičom ale aspoň

---

<sup>8</sup>V jazyku DE sa jeden rekombinačný operátor označuje ako mutácia.

jedna jeho hodnota bola prenesená zo šumového vektora. Toto je možné strážiť algoritmicke – náhodne sa určí jedna pozícia, ktorá sa bude preberať zo šumového vektora deterministicky.

Pre horný odhad veľkosti populácie možno použiť hodnotu rovnú  $100l$ , kde  $l$  je počet atribútov.

### 13.2.4 Samo-organizujúci sa migračný algoritmus

Tento algoritmus existuje od roku 1999. Jeho hlavným znakom je, že pri tvorbe nových potomkov sa nesnaží imitovať prírodné princípy, ale využíva geometrické princípy – traverzovanie priestoru po priamkach.

Nasledujúci opis je založený na [48] a je známy ako variant AllToOne.

**Reprezentácia.** Usporiadaná lineárna štruktúra pevnej dĺžky používajúca reálne kódovanie.

Jedinec kóduje priamo iba riešenie bez akýchkoľvek dodatočných informácií.

**Vhodnosť.** Vhodnosť je odvodená z hodnoty účelovej funkcie. Nie je potrebné vykonať žiadne dodatočné transformácie.

**Selekcia.** Selekcia je<sup>9</sup> zdrojom selekčného tlaku. Je realizovaná ako elitistická selekcia.

Ako selekčná metóda sa používa deterministický výber  $\mu - 1$  dvojíc rodičov. Jedinec s najlepšou vhodnosťou je umiestnený ako prvý rodič do každej z týchto dvojíc. Ostatné jedince hrajú úlohu druhých rodičov, pričom každý z nich vystupuje v jednej rodičovskej dvojici.

**Genetické operátory.** Algoritmus používa iba jeden typ operátora - aritmetické kríženie, využívajúce perturbačný vektor. Úlohu štartovacieho rodiča hrá jedinec s horšou vhodnosťou, vodiaci rodič zastupuje rodiča s vhodnosťou lepšou.

Operátor generuje body v úseku medzi oboma rodičmi a v úseku za vodiacim rodičom, pričom vplyvom perturbačného vektora smer pohybu môže byť odklonený od smeru prechádzajúceho vodiacim rodičom (obr. 8.4). Hustota generovaných bodov a ich počet je daný nastavením parametrov. Novo

---

<sup>9</sup>Priemerná vhodnosť v skupine rodičov je vyššia alebo rovná priemernej vhodnosti aktuálnej populácie, pričom rovnosť nastáva iba v prípade, že všetky jedince majú rovnakú vhodnosť.

vygenerovaný potomok je reprezentovaný vygenerovaným bodom s najlepšou hodnotou vhodnosti.

**Náhrada.** Je realizovaná ako generačná náhrada doplnená elitistickým prenosom. Keďže bolo vygenerovaných  $\mu - 1$  potomkov, tieto jedince po doplnení najlepším jedincem z aktuálnej populácie vytvoria novú generáciu.

**Inicializácia.** Prvotná populácia je vytvorená náhodným vzorkovaním každej pozície. Pri tomto vzorkovaní sa vyberá jedna z možných hodnôt pri použití uniformnej distribúcie pravdepodobnosti.

**Podmienka ukončenia.** Pevne stanovený počet generácií. Po jeho uplynutí sa evolučný proces zastaví. Inou možnosťou je pokles rozdielu vhodností najlepšieho a najhoršieho jedinca v populácii pod stanovenú hranicu.

**Typické hodnoty.** Pre hodnotu kroku aritmetického kríženia sa odporúča voliť hodnoty z intervalu  $\langle 0, 3 \rangle$ , pričom horná hodnota bola zistená ako postačujúca pre praktické úlohy. Pre zachovanie vhodnej “zrnitosti” prehľadávania sa odporúča používať ekvidištantné hodnoty kroku, pričom vzdialenosť medzi nimi by mala byť dostatočne malá a súčasne taká, aby krok nenadobudol hodnotu 1 (aby sa vylúčil prípad, že vodiaci jedinec svojimi kópiami v populácii spôsobí predčasnú konvergenciu).

Pri tvorbe perturbačného vektora sa doporučuje hodnota 0.1 pre pravdepodobnosť generovania hodnoty 1.

### 13.2.5 Harmonické prehľadávanie

Algoritmus harmonického prehľadávania bol navrhnutý v roku 2001. Bol odvodený z fenoménu, keď muzikanti, improvizujúci na svojich hudobných nástrojoch, vytvárajú zvukovú harmóniu.

Nasledujúci opis<sup>10</sup> algoritmu je založený na [25] a reprezentuje verziu algoritmu vhodnú pre numerickú optimalizáciu.

**Reprezentácia.** Usporiadaná lineárna štruktúra pevnej dĺžky používajúca reálne kódovanie. Jedinec kóduje priamo iba riešenie bez akýchkoľvek dodatočných informácií.

---

<sup>10</sup>Komunita harmonického prehľadávania používa vlastné terminologické označenia. Tu je ich terminológia mapovaná na pojmy, použité v predchádzajúcich častiach.

Každá z hodnôt atribútov je obmedzená na nejaký podinterval reálnych čísel  $\langle \min(v_i), \max(v_i) \rangle$ , reprezentujúci definičný interval hodnôt pre danú pozíciu. Toto obmedzenie je platné vo všetkých fázach realizácie algoritmu.

**Vhodnosť.** Vhodnosť je odvodená z hodnoty účelovej funkcie. Nie je potrebné vykonať žiadne dodatočné transformácie.

**Selekcia.** Selekcia nie je zdrojom selekčného tlaku. Je realizovaná ako elitistická selekcia.

Ako selekčná metóda sa používa deterministický výber – každý jedinec v aktuálnej populácii je súčasne aj rodičom.

**Genetické operátory.** Algoritmus používa oba typy operátorov – mutáciu pre zmenu informácie kódovanej v rodičoch a rekombináciu pre kombinovanie informácií obsiahnutých v rôznych jedincoch. Mutačná zmena môže mať dve formy. Jednou z nich je modifikácia hodnoty, ktorá sa nachádzala v niektorom z rodičov, zatiaľ čo druhou je náhrada takejto hodnoty. Tomu zodpovedá aj použitie dvoch rôznych mutačných operátorov.

Rekombinačný operátor má podobu panmiktického operátora, ktorý používa  $\mu$  rodičov (všetky jedince aktuálnej populácie sú rodičmi pri aplikácii tohto operátora). Je realizovaný ako skenovací operátor založený na náhodnom výbere, pričom pri fyzickej realizácii sa náhodne určuje rodič, z ktorého sa preniesie hodnota do vytváraného potomka.

Mutačný operátor pre náhradu hodnoty je realizovaný ako uniformná mutácia, vyberajúca novú hodnotu z definičného oboru prislúchajúcemu danej pozícii.

Mutačný operátor pre modifikáciu hodnoty je realizovaný ako uniformná mutácia, vyberajúca novú hodnotu z nejakého intervalu  $\langle h_{ij} - \Delta, h_{ij} + \Delta \rangle$ . Ak nová hodnota je menšia ako minimálna hranica definičného intervalu, prislúchajúceho danej pozícii, tak je nahradená touto dolnou hranicou. Ak je väčšia ako horná hranica definičného intervalu, tak je nahradená zase touto hornou hranicou.

Operátory sú kombinované sekvenčným spôsobom. Najprv sa aplikuje rekombinačný operátor, ktorý vytvorí jedného potomka. Na tohto potomka je aplikovaný mutačný operátor pre modifikáciu hodnôt. Záverečne je aplikovaný na potomka mutačný operátor pre náhradu hodnôt<sup>11</sup>.

---

<sup>11</sup>Pri praktickej realizácii je vhodné prelínanie operátorov pre zabránenie zbytočných operácií – ak na nejakú pozíciu má byť aplikovaný mutačný operátor pre náhradu, je

Rekombinačný operátor je aplikovaný vždy, mutačné operátory iba pravdepodobnostne (každý z nich má svoju vlastnú pravdepodobnosť mutácie).

**Náhrada.** Náhrada je zdrojom selekčného tlaku. Je realizovaná ako náhrada s generačnou medzerou doplnená deterministickým turnajom.

Keďže v každej generácii je vygenerovaný iba jeden potomok, generačná medzera má hodnotu  $1/\mu$ . Novo vygenerovaný potomok súťaží s najhorším jedincom v populácii. Ak má lepšiu vhodnosť, potom je vložený do populácie, pričom nahradí aktuálne najhoršieho jedinca. V opačnom prípade je zahodený.

**Inicializácia.** Prvotná populácia je vytvorená náhodným vzorkovaním každej pozície s využitím definičného intervalu prislúchajúceho danej pozícii.

**Podmienka ukončenia.** Pevne stanovený počet generácií (resp. generovaných potomkov). Po jeho uplynutí sa evolučný proces zastaví.

**Typické hodnoty.** Pre zlepšenie konvergencie k optimálnemu riešeniu sa používa dynamické nastavenie parametrov mutačného operátora pre modifikáciu hodnoty – jeho pravdepodobnosti mutácie a veľkosti maximálnej zmeny. Pravdepodobnosť mutácie pre zmenu hodnoty sa lineárne zvyšuje podľa

$$p_m^{min} + (p_m^{max} - p_m^{min}) \frac{t}{t_{max}} \quad (13.5)$$

kde  $p_m^{min}$  a  $p_m^{max}$  sú jej hraničné hodnoty a  $t_{max}$  reprezentuje povolený počet generácií, po uplynutí ktorého sa proces zastaví.

Hodnota maximálnej zmeny  $\Delta$  sa exponenciálne znižuje podľa nasledovného vzťahu

$$\Delta_{max} e^{\ln(\Delta_{min}/\Delta_{max})t/t_{max}} \quad (13.6)$$

kde  $\Delta_{min}$  a  $\Delta_{max}$  sú opäť hraničné hodnoty.

### 13.2.6 Eugenická evolúcia

Cieľom eugenickej evolúcie bolo zohľadniť epistázu – závislosť medzi hodnotami na rôznych pozíciách (vplyv hodnoty umiestnenej na nejakej pozícii na vhodnosť jedinca môže byť rôzny v závislosti od toho, aká hodnota sa nachádza na nejakej inej pozícii).

---

zbytočné na danú pozíciu vyberať hodnotu z potomka a následne ju modifikovať, keď aj tak následne bude nahradená náhodnou hodnotou.

Nasledujúci opis algoritmu je založený na [36].

**Reprezentácia.** Usporiadaná lineárna štruktúra pevnej dĺžky používajúca binárne kódovanie.

Jedinec kóduje priamo iba riešenie bez akýchkoľvek dodatočných informácií.

**Vhodnosť.** Vhodnosť je odvodená z hodnoty účelovej funkcie. Nie je potrebné vykonať žiadne dodatočné transformácie.

**Selekcia.** Selekcia nie je zdrojom selekčného tlaku. Je realizovaná ako elitistická selekcia.

Ako selekčná metóda sa používa deterministický výber – každý jedinec v aktuálnej populácii je súčasne aj rodičom.

**Genetické operátory.** Algoritmus používa iba jeden typ operátora - rekombinačný operátor, ktorý umožňuje kombinovanie informácie obsiahnutej v rôznych jedincoch.

Rekombinačný operátor má podobu panmiktického operátora, ktorý používa  $\mu$  rodičov (všetky jedince aktuálnej populácie sú rodičmi pri aplikácii tohto operátora). Je realizovaný ako skenovací operátor, založený na náhodnom výbere. Oproti štandardnému skenovaciemu operátoru má však tieto odlišnosti:

- pozície v potomkovi sú obsadzované hodnotami v poradí, ktoré je dané dôležitosťou jednotlivých pozícií,
- po priradení hodnoty na nejakú pozíciu môže dôjsť k redukcii rodičov, ktorí budú použítí pre obsadenie ďalšej pozície.

Pri určovaní poradia pozícií sa vychádza z pravdepodobnosti jednotlivých hodnôt pre jednotlivé pozície. Tá pozícia z neobsadených pozícií je najdôležitejšia (a teda jej bude priradená hodnota najskôr), ktorá maximalizuje

$$|p(h_j = 0) - p(h_j = 1)| \quad (13.7)$$

kde  $p(h_j = 0)$  ( $p(h_j = 1)$ ) reprezentuje pravdepodobnosť toho, že pre  $j$ -tu pozíciu bude vybraná hodnota 0 (hodnota 1).



Pri rozhodovaní o redukcii používaných rodičov sa používa pravdepodobnosť redukcie, ktorá sa určuje ako

$$1.0 - \max_j |p(h_j = 0) - p(h_j = 1)| \quad (13.8)$$

kde  $j$  reprezentuje indexy aktuálne neobsadených pozícií. Ak k redukcii dôjde, tak pre obsadenie ostávajúcich pozícií sa ponechajú iba tí rodičia, ktorí sú relevantní pre posledne priradenú hodnotu.

**Náhrada.** Náhrada je zdrojom selekčného tlaku. Je realizovaná ako náhrada s generačnou medzerou.

Keďže v každej generácii je vygenerovaný iba jeden potomok, generačná medzera má hodnotu  $1/\mu$ . Novo vygenerovaný potomok vždy nahradí aktuálne najhoršieho jedinca.

**Inicializácia.** Prvotná populácia je vytvorená náhodným vzorkovaním každej pozície s uniformnou distribúciou pravdepodobnosti.

**Podmienka ukončenia.** Pevne stanovený počet generácií. Po jeho uplynutí sa evolučný proces zastaví.

**Typické hodnoty.** Pravdepodobnosť výberu možných hodnôt pre nejakú pozíciu je založená na vhodnosti. Pravdepodobnosť hodnoty 0 pre  $j$ -tu pozíciu sa určí ako

$$p(h_j = 0) = \frac{\bar{\Phi}(h_j = 0)}{\bar{\Phi}(h_j = 0) + \bar{\Phi}(h_j = 1)} \quad (13.9)$$

kde  $\bar{\Phi}(h_j = 0)$  ( $\bar{\Phi}(h_j = 1)$ ) je priemerná vhodnosť tých rodičov, ktorí majú na  $j$ -tej pozícii hodnotu 0 (hodnotu 1).



---

**Časť VII**

**Prílohy**



# Matematické symboly

Symbol	Krátky popis
$\lfloor \cdot \rfloor$	zaokrúhlenie nadol
$\lceil \cdot \rceil$	zaokrúhlenie nahor
$(\cdot)$	vektor daný súradnicami
$\{ \cdot \}$	množina
$[\cdot]$	bod v priestore daný súradnicami
$\  \cdot \ $	kardinalita
$\propto$	proporcionálne
$\approx$	približne
$\Sigma$	súčet
$\Pi$	súčin
$\implies$	implikácia
$\%$	modulo
$\oplus$	súčet modulo
$\vec{\cdot}$	vektor
$\alpha$	parameter, uhol
$\beta$	parameter
$\Delta$	zmena, funkcia zmeny
$\eta$	očakávaný počet rodičov
$\eta'$	aktuálny počet rodičov

Symbol	Krátky popis
$\varepsilon$	malé číslo
$\chi$	náhodná premenná
$\lambda$	počet potomkov
$\mu$	veľkosť populácie
$\sigma$	smerodajná odchýlka
$\tau$	parameter
$\tau^-, \tau^+$	koeficienty lineárneho premapovania vhodnosti
$\varrho$	veľkosť skupiny rodičov
$\Phi$	vhodnosť
$\Phi_{min}$	najhoršia vhodnosť
$\Phi_{max}$	najlepšia vhodnosť
$\Phi'$	vhodnosť po premapovaní
$\bar{\Phi}$	priemerná vhodnosť
$\bar{\Phi}^*$	očakávaná priemerná vhodnosť
$\bar{\Phi}'$	priemerná vhodnosť po premapovaní
$\pi$	Ludolfovo číslo (3,14159...)
$\xi$	konštanta/parameter
$\theta$	prah, prahová hodnota
$\zeta_i$	i-ty algoritmus
$a_i, b_i$	i-ty jedinec
$A, B, C, D$	body v priestore
$A_i, B_i, C_i, D_i$	i-te súradnice bodov v priestore
$At_i$	i-ty atribút
$c$	báza exponencionálneho premapovania vhodnosti
$c_i$	konštanta
$d$	vzdialenosť
$e$	Eulerovo číslo (2.71828...)

Symbol	Krátky popis
$E(.)$	stredná hodnota
$f_1, f_2, \dots$	hodnoty vhodnosti
$F$	kumulatívna distribúcia indexov
$g$	zobrazenie (dekodér)
$G$	generačná medzera
$h_j (h_{ij})$	hodnota zobrazená na j-tej pozícii (v i-tom jedincovi)
$H$	histogram
$k$	konštanta
$K$	korekcia
$l$	počet atribútov, dĺžka reprezentácie
$l_i$	dĺžka i-teho segmentu
$L, L_1$	vzdialenosť, dĺžka
$\mathcal{M}_S, \mathcal{M}_C$	priestor (prehľadávania, kandidátov)
$min, max$	minimálna a maximálna hodnota
$n$	počet
$Nx$	počet objektov typu "x"
$N(., .)$	normálne rozdelenie pravdepodobnosti
$p_c$	pravdepodobnosť kríženia
$p_m$	pravdepodobnosť mutácie
$p_s$	pravdepodobnosť selekcie
$p_ =$	pravdepodobnosť rovnakých hodnôt
$p_0$	pravdepodobnosť výskytu hodnoty 0
$p_z$	pravdepodobnosť zmeny
$p(.)$	pravdepodobnosť nejakého javu
$P$	populácia
$P'$	rodičia
$P''$	potomkovia

Symbol	Krátky popis
$q$	veľkosť turnaja/konštanta
$r_{ij}$	prvok matice
$R$	rotačná matica
$\Re$	reálne čísla
$s$	distribúcia vhodnosti
$s^*$	očakávaná distribúcia vhodnosti
$s'$	skutočná distribúcia vhodnosti
$sf$	škálovací faktor
$S$	kumulatívna distribúcia vhodnosti
$S^*$	očakávaná kumulatívna distribúcia vhodnosti
$t, t_1, t_2$	generácia
$t_{max}$	maximálna generácia
$T, T_1, T_2$	prah orezania
$u_i$	zložka vektora
$U$	účelová funkcia
$U_{min}$	minimálna hodnota účelovej funkcie
$U_{max}$	maximálna hodnota účelovej funkcie
$v_i$	hodnota (i-teho atribútu)
$V$	objem, kapacita
$w$	šírka okna
$x_i$	i-ta súradnica
$y, y_i$	hodnota
$z$	parameter



## Slovensko – anglický slovník

Nedostatok publikácií v slovenskom jazyku z oblasti evolučných algoritmov má za následok neustálenosť slovenských podôb jednotlivých odborných termínov. Predložený text vychádza z anglickej terminológie, pričom sa však nesnaží o doslovný preklad – je uprednostnený taký preklad, ktorý daný pojem vyjadruje čo najlepšie. Pre zorientovanie sa v použitej terminológii a uľahčenie konfrontovania predkladaných informácií s informáciami prezentovanými v iných publikáciách je v tejto prílohe uvedený krátky slovensko – anglický slovník najdôležitejších odborných termínov, ktoré boli použité v texte.

Slovenský termín	Anglický termín
A algoritmus	
– evolučný	evolutionary algorithm
– populačný	population algorithm
B baliaci problém	bin packing problem
báza zotriedenia	ranking base
D deliaci bod	crossover position
distribúcia	
– kreditu	credit assignment
– vhodnosti	fitness distribution
– – kumulatívna	cumulative fitness distribution
– – očakávaná	expected fitness distribution
doba prevzatia	takeover time
dvojpákový výherný automat	two-armed bandit
E epistáza	epistasis
evolučný cyklus	evolutionary cycle

F	funkcia vhodnosti	fitness function
G	generácia generačná medzera genetický drift	generation generation gap genetic drift
H	histogram distribúcie vzdialeností hodnota – binárna – na pozícii hranica – dolná – horná hypermutácia hypotéza stavebných blokov	distance distribution histogram  binary value allele  lower bound upper bound hypermutation building block hypothesis
I	infúzia materiálu inverzia	genetic load inversion
J	jedinec	individual, chromosome
K	kandidát riešenia kódovanie – binárne – mnohoznačné – parametrov dynamické – permutačné – reálne konvergencia – populácie – predčasná konzistencia priestorová kríženie – aritmetické – dvojbodové – heuristické – jednobodové – lineárne – miešajúce – mnohobodové – segmentové – uniformné	solution candidate  binary encoding n-ary encoding dynamic parameter encoding permutation encoding floating point encoding convergence population convergence premature convergence space consistency crossover arithmetical crossover two-point crossover heuristic crossover one-point crossover linear crossover shuffle crossover multi-point crossover segmented crossover uniform crossover

M	mapa pokrytia	coverage map
	matica priestoru prehľadávania	search space matrix
	metaevolúcia	meta-evolution
	metóda	
	– selekčná	selection method
	– vizualizačná	visualisation method
	– analýzy hlavných komponentov	principal component analysis method
	mera reprodukcie	reproduction rate
	mutácia	mutation
	– doplnková	flip mutation
	– neuniformná	non-uniform mutation
	– normálna	normal mutation
	– uniformná	uniform mutation
	– výmenná	swap mutation
N	náhodný výber	random selection
	náhrada	replacement
	– elitistická náhrada	elitist replacement
	– generačná náhrada	generational replacement
	normálna distribúcia	normal distribution
O	obmedzená súťaž	restricted competition
	obmedzené spájanie	restricted mating
	odchýlka	bias
	operátor genetický	genetic operator
	– asexuálny	asexual operator
	– panmiktický	panmictic operator
	– – diagonálny	diagonal operator
	– – diferenciálny	differential operator
	– – skenovací	scanning operator
	– sexuálny	sexual operator
	orezanie	truncation
	– deterministické	deterministic truncation
P	plocha vhodnosti	fitness landscape
	počet	
	– rodičov	number of parents
	– – očakávaný	expected number of parents
	– vyhodnotení	number of evaluations

polomer zdieľania	sharing radius
populácia	population
– aktuálna	actual population
– prvotná	initial population
potenciál	potential
– deštrukčný	disruption potential
– prehľadavací	exploration potential
– rekombinačný	recombination potential
potomok	offspring
pozícia v jedincovi	gene, feature, locus
prah orezania	truncation threshold
pravdepodobnosť	probability
– kríženia	crossover probability
– mutácie	mutation probability
– selekcie	selection probability
preferencia pozičná	positional bias
prehľadávanie	search
– náhodné	random search, random walk
premapovanie vhodnosti	fitness remapping
priestor	
– kandidátov riešení	candidate space, phenotype space
– prehľadávania	search space, genotype space
problém ihly v kope sena	needle in a haystack problem
R	
reinizializácia	reinitialization
rekombinácia	recombination
reprezentácia	
– s pevnou dĺžkou	fixed length representation
– s pevným usporiadaním	fixed order representation
– s premenlivou dĺžkou	variable length representation
– s premenlivým usporiadaním	variable order representation
reprodukcia	reproduction
rodič	parent
rotačný uhol	rotation angle
rozpätie hodnôt	spread
rôznorodosť populácie	population diversity
ruleta	spinning wheel, roulette wheel
rýchlosť konvergenencie	convergence velocity

S	samoadaptácia	self-adaptation
	selekcia	selection
	– deterministická	deterministic selection
	– disruptívna	disruptive selection
	– dynamická	dynamic selection
	– elitistická	elitist selection
	– s vymieraním	extinctive selection
	– – sprava	right extinctive selection
	– – zľava	left extinctive selection
	– statická	static selection
	– uchovávajúca	preservative selection
	– založená na vhodnosti	fitness based selection
	– – proporcionálna	fitness proportionate selection
	selekčná intenzita	selection intensity
	– štandardizovaná	standardized selection intensity
	selekčný šum	selection noise
	selekčný tlak	selection pressure
	skúmanie	exploration
	strata rôznorodosti	loss of diversity
	stratégia	
	– čiarková	comma strategy
	– plus	plus strategy
	stupeň konverencie	convergence rate
	superjedinec	superindividual
	súťaž	
	– bipartitná	bipartite competition
	– náhodná	random competition
	– plná	full competition
Š	škálovací faktor	scaling factor
	škálovanie	scaling
	– lineárne	linear scaling
	– sigma	sigma scaling/sigma truncation
T	turnaj	tournament
	– binárny	binary tournament
	– pravdepodobnostný	probabilistic tournament
	– q-árny	q-ary tournament
	– – bez náhrady	q-ary tournament without replacement

U	účelová funkcia	object function
	udržiavanie	
	– rôznorodosti	diversity maintaining
	– subpopulácií	subpopulation maintaining
	ukončovacia podmienka	termination criterion
V	veľkosť	
	– populácie	population size
	– turnaja	tournament size
	vhodnosť	fitness
	– aproximatívna	approximative fitness
	– populácie	population fitness
	– – najhoršia	population worst fitness
	– – najlepšia	population best fitness
	– – priemerná	population average fitness
	– surová	raw fitness
	– súťažná	competitive fitness
	– štandardizovaná	standardised fitness
	– založená na komplexnosti	complexity-based fitness
	využívanie	exploitation
	vzorkovanie	sampling
	– stochastické	stochastic sampling
	– – zvyškové	remainder stochastic sampling
	– – bez náhrady	stochastic sampling without replacement
	– – s náhradou	stochastic sampling with replacement
	– – univerzálne	stochastic universal sampling
Z	základňa	baseline
	zdieľanie	sharing
	zotriedenie	ranking
	– exponenciálne	exponential ranking
	– lineárne	linear ranking

# Literatúra

- [1] Aguirre, H.E. – Tanaka, K.: Parallel Varying Mutation in Deterministic and Self-adaptive GAs. In: Proc. of the 7th Int. Conference Parallel Problem Solving from Nature - PPSN VII, Granada, Spain, (7-11 September 2002), 111–121.
- [2] Baker, J.E.: Reducing Bias and Inefficiency in the Selection Algorithm. In: Proc. of the 2nd Int. Conference on Genetic Algorithms, Cambridge, Massachusetts, October 1987. Lawrence Earlbaum Associates, Hillsdale, New Jersey, 14–21.
- [3] Bäck, T. – Schwefel, H-P.: An Overview of Evolutionary Algorithms for Parameter Optimization. *Evolutionary Computation*, vol. 1, 1993, no. 1, 1–23.
- [4] Bäck, T.: *Evolutionary Algorithms in Theory and Practice*. Oxford University Press, New York, 1996, ISBN 0-19-509971-0, 314 s.
- [5] Bäck, T. – Fogel, D.B. – Michalewicz, Z.: *Evolutionary Computation 2: Advanced Algorithms and Operators*. IOP Publishing, Bristol, 2000, ISBN 0-7503-0665-3, 270 s.
- [6] Blickle, T. – Thiele, L.: A Comparison of Selection Schemes used in Evolutionary Algorithms. *Evolutionary Computation*, vol. 4, 1996, no. 4, 361–394.
- [7] Caruana, R.A. – Eshelman, L.J. – Schaffer, J.D.: Representation and hidden bias II: Eliminating defining length bias in genetic search via shuffle crossover. In: Proc. of the 11th Int. Joint Conference on Artificial Intelligence, Detroit, Michigan, 1989, 750–755.
- [8] Collins, T.D.: Using Software Visualisation Technology to Help Evolutionary Algorithm Users Validate their Solutions. In: Proc. of the 7th Int. Conference on Genetic Algorithms, Michigan, 1997, 307-314.

- [9] Deb, K. – Goldberg, D.E.: An investigation of niche and species formation in genetic function optimization. In: Proc. of the 3rd Int. Conference on Genetic Algorithms, Morgan Kaufmann, San Mateo, CA, 1989, 40–50.
- [10] Eiben, A.E. – Schut, M.C. – de Wilde, A.R.: Is Self-adaptation of Selection Pressure and Population Size Possible? – A Case Study. In: Proc. of the 9th Int. Conference Parallel Problem Solving from Nature PPSN IX, Reykjavik, (September 2006), 900–909.
- [11] Eshelman, L.J. – Schaffer, J.D.: Preventing Premature Convergence in Genetic Algorithms by Preventing Incest. In: Proc. of the 4th Int. Conference on Genetic Algorithms, Morgan Kaufmann, San Mateo, 1991, 115–122.
- [12] Falkenauer, E.: Genetic Algorithms and Grouping Problems. John Wiley & Sons, Chichester, 1998, ISBN 0-471-97150-2, 220 s.
- [13] Feoktistov, V.: Differential Evolution: In Search of Solutions. Springer, 2006, ISBN 0-387-36896-5, 195 s.
- [14] Goldberg, D.E.: Genetic Algorithms in Search, Optimization, and Machine Learning. Addison-Wesley, 1989, ISBN 0-201-15767-5.
- [15] Goldberg, D.E. – Deb, K. – Kargupta, H. – Harik, G.: Rapid, accurate optimization of difficult problems using fast messy genetic algorithms. In: Proc. of the 5th Int. Conference on Genetic Algorithms ICGA'93, Urbana-Champaign, IL, 1993, 56–64.
- [16] Hancock, P.J.B.: Selection Methods for Evolutionary Algorithms. In: Chambers, L. (ed.): Practical Handbook of Genetic Algorithms. CRC Press, New York, 1995, 67–92.
- [17] Herrera, F. – Lozano, M. – Verdegay, J.L.: Fuzzy connectives based crossover operators to model genetic algorithms population diversity. Fuzzy Sets and Systems, vol. 92, 1997, 21–30.
- [18] Jin, Y. – Olhofer, M. – Sendhoff, B.: On Evolutionary Optimization with Approximate Fitness Functions. In: Proc. of the Genetic and Evolutionary Computation Conference GECCO'2000, Las Vegas, Ne, (10–12 July, 2000), 786–793.
- [19] Julstrom, B. A.: Population Size in GA for TSP. In: Proc. of the 2nd NWGA, Vaasa, Fínsko, (19–23 August 1996).



- [20] Julstrom, B. A.: Adaptive Operator Probabilities. In: Proc. of the 2nd NWGA, Vaasa, Fínsko, (19-23 August 1996).
- [21] Kim, Y.H. – Moon, B.R.: New Usage of Sammon’s Mapping for Genetic Visualisation. In: Proc. of the Genetic and Evolutionary Computation Conference, Chicago, 2003, 1136-1147.
- [22] Koza, J. R.: Genetic Programming. On the Programming of Computers by Means of Natural Selection. The MIT Press, Cambridge, Massachusetts, 5. vydanie, 1996, ISBN 0-262-11170-5, 819 s.
- [23] Kuo, T. – Hwang, S-Y.: A Genetic Algorithm with Disruptive Selection. IEEE Trans. on Systems, Man, and Cybernetics – part B: Cybernetics, vol. 26, 1996, no. 2, 299–307.
- [24] Kvasnička, V. – Pospíchal, J. – Tiňo, P.: Evolučné algoritmy. Vydavateľstvo STU, Bratislava, 2000, ISBN 80-227-1377-5, 215 s.
- [25] Mahdavi, M. – Fesanghary, M. – Damangir, E.: An improved harmony search algorithm for solving optimization problems. Applied Mathematics and Computation, vol. 188, 2007, 1567–1579.
- [26] Mahfound, S.W.: Niching methods for genetic algorithms. PhD dissertation, Univ. of Illinois, Urbana-Champaign, USA, 1995.
- [27] Mach, M. – Zeťáková, Z.: Visualising Genetic Algorithms: A Way through the Labyrinth of Search Space. In: Intelligent Technologies – Theory and Applications. IOS Press, Amsterdam, 2002, 279–285.
- [28] Maresky, J. et al.: Selectively Destructive Re-start. In: Proc. of the 6th Int. Conference on Genetic Algorithms, Morgan Kaufmann, San Francisco, 1995, 144–150.
- [29] Mengshoel, O.J. – Goldberg, D.E.: Probabilistic Crowding: Deterministic Crowding with Probabilistic Replacement. In: Proc. of the Genetic and Evolutionary Computation Conference GECCO-99, Orlando, Florida, (13-17 July, 1999), 409–416.
- [30] Michalewicz, Z.: Genetic Algorithms + Data Structures = Evolution Programs (3. vydanie). Springer, New York, 1996, ISBN 3-540-60676-9, 387 s.
- [31] Mitchell, M.: An Introduction to Genetic Algorithms. MIT Press, Cambridge, MA, 1996, ISBN 0-262-13316-4.

- [32] Mühlenbein, H. – Schlierkamp-Voosen, D.: Predictive Models for the Breeder Genetic Algorithm I. Continuous Parameter Optimization. *Evolutionary Computation*, Vol. 1, 1993, No. 1, 25–49.
- [33] Pekalska, E. et al.: A new method of generalizing Sammon mapping with application to algorithm speed-up. In: *Proc. of the 5th Annual Conf. of the Advanced School for Computing and Imaging, 1999*, 221–228.
- [34] Petrowski, A.: A new selection operator dedicated to speciation. In: *Proc. of the 7th Int. Conference on Genetic Algorithms, Morgan Kaufmann, San Francisco, 1997*, 144–151.
- [35] Pohlheim, H.: Vizualization of Evolutionary Algorithms – Set of Standard Techniques and Multidimensional Visualization. In: *Proc. of the Genetic and Evolutionary Computation Conference, Orlando, 1999*, 533–540.
- [36] Prior, J.W.: *Eugenic Evolution for Combinatorial Optimization*. Master of Arts Thesis, University of Texas, Austin, 1998.
- [37] Rana, S.B. – Whitley, L.D.: Bit Representations with a Twist. In: *Proc. of the 7th Int. Conference on Genetic Algorithms ICGA'97, East Lansing, MI, (19-23 July, 1997)*, 188–195.
- [38] Schaffer, J.D. et al.: A Study of Control Parameters Affecting Online Performance of Genetic Algorithms for Function Optimization. In: *Proc. of the 3rd Int. Conference on Genetic Algorithms ICGA'89, Arlington, VA, (4-7 June 1989)*, 51–60.
- [39] Schraudolph, N.N. – Belew, R.K.: Dynamic Parameter Encoding for Genetic Algorithms. *Machine Learning*, vol. 9, 1992, no. 1, 9–21.
- [40] Soak, S.M. – Corne, D. – Ahn, B.H.: A Powerful New Encoding for Tree-Based Combinatorial Optimisation Problems. In: *Proc. of the 8th Int. Conference on Parallel Problem Solving from Nature PPSN VIII, Birmingham, UK, (September 2004)*, 430–439.
- [41] Spears, W.M. – De Jong, K.A.: On the virtues of parameterized uniform crossover. In: *Proc. of the 4th Int. Conference on Genetic Algorithms ICGA'91, Morgan Kaufmann, San Mateo, 1991*, 230–229.

- [42] Syswerda, G.: Uniform crossover in genetic algorithms. In: Proc. of the 3rd Int. Conference on Genetic Algorithms ICGA'89, Morgan Kaufmann, San Mateo, 1989, 2–9.
- [43] Voigt, H.M.: Soft genetic operators in evolutionary algorithms. In: Evolution and Biocomputation: Computational Models of Evolution. Springer, Berlin, 1995, 123–141.
- [44] Walton, N. – Smith, G.D.: The Origination of Diversity by Adaptive Clustering. In: Proc. of the 6th Int. Conference on Parallel Problem Solving from Nature PPSN VI, Paris, France, (18-20 September, 2000), 415–424.
- [45] Wierzch, W. – Czech, Z.J.: Selection schemes in evolutionary algorithms. In: Proc. of the Intelligent Information Systems XI - IIS'02, Sopot, Poland, (3-6 June, 2002), 185–194.
- [46] Willis-Ford, C. – Soule, T.: Non-stationary Subtasks can Improve Diversity in Stationary Tasks. In: Proc. of the Genetic and Evolutionary Computation Conference GECCO 2004, part II, Seattle, WA, USA, (26-30 June, 2004), 307–317.
- [47] Wolpert, D.H. – Macready, W.G.: No Free Lunch Theorems for Search. Technical Report SFI-TR-95-02-010, The Santa Fe Institute, 1996.
- [48] Zelinka, I.: Umělá inteligence v problémech globální optimalizace. Ben, Praha, 2002, ISBN 80-7300-069-5, 192 s.



# Register

- algoritmus
  - evolučný, 4
    - diferenciálna evolúcia, 206
    - eugenická evolúcia, 211
    - evolučná stratégia, 194
    - evolučné programovanie, 197
    - genetické programovanie, 202
    - genetický algoritmus, 199
    - harmonické prehľadávanie, 209
    - soma, 208
    - šľachtiteľský algoritmus, 204
  - horolezecký, 4
  - individuálny, 4
  - náhodného hľadania, 4
  - populačný, 4
  - prehľadávací, 3
- alokácia pokusov, 174
  - optimálna, 173
- baliaci problém, 48
- báza zotriedenia, 87
- dekodér, 22
- deliaci bod, 107
- diagram Hintonov, 187
- distribúcia vhodnosti, 14, 65
  - kumulatívna, 65
  - očakávaná, 65
  - očakávaná, 65
- doba prevzatia, 89
- eliminácia duplikácií, 152
- entropia populácie, 160
- epistáza, 176, 211
  - silná, 176
  - slabá, 176
- evaluácia jedincov, 14
- evolučný cyklus, 13, 14
- faktory konvergenencie, 149
- funkcia
  - Ackleyho, 160
  - distribučná, 65
  - kumulatívna, 65
  - účelová, 45
  - vhodnosti, 45
- generácia, 11
- generovanie jedincov, 13
- genetický drift, 68
- genetický operátor, 15, 105
  - asexuálny, 105
  - panmiktický, 105
  - sexuálny, 105
  - analýza, 120
- histogram
  - distribúcie vzdialenosti, 181
  - pravdepodobnosti, 143
  - vhodnosti, 65
- hustota pravdepodobnosti
  - Cauchyho, 113
  - normálneho rozdelenia, 139
- hypermutácia, 153

- hypotéza stavebných blokov, 171
- chyba vzorkovania, 68
- jedinec, 5, 11
- kód
  - binárny, 35
  - Grayov, 35
- kódovanie
  - binárne, 33
  - mnohoznakové, 33
  - parametrov dynamické, 137
  - permutačné, 34
  - reálne, 33
- konvergencia
  - predčasná, 96
  - priestorová, 147
  - štruktúrálna, 147
- konvergencia populácie, 147
  - predčasná, 148
- konzistencia priestorová, 182
- korekcia, 122–124
- kritérium
  - minimálnej dĺžky popisu, 51
- kríženie
  - aritmetické, 114
  - dvojbodové, 108
  - FCB, 116
  - heuristické, 116
  - jednobodové, 108
  - lineárne, 115
  - miešajúce, 108, 122
  - mnohobodové, 107, 121
  - OX, 119
  - PMX, 118
  - segmentové, 109, 123
  - uniformné, 109, 123
- mapa pokrytia, 186
- mapovanie
  - lineárne, 186
  - Sammonovo, 184
- matica
  - priestoru prehľadávania, 186
  - rotačná, 141
  - vlastností hodnôt atribútov, 189
- medzera generačná, 100
- metóda
  - ruletová, 69
  - selektie
    - dynamická, 66
    - statická, 66
  - silná, 5
  - slabá, 5
- miera
  - konvergenzie, 148
  - reprodukcie, 66
- migrácia, 15
- množina
  - potomkov, 15
  - rodičov, 14
- mutácia
  - Cauchyho, 113
  - doplnková, 107
  - extremálna, 112
  - inverzná, 118
  - Mühlenbeinova, 113
  - neuniformná, 112
  - normálna, 113
  - uniformná, 112
  - výmenná, 118
- náhodný výber, 79
- náhrada, 15
  - elitistická, 103
  - generačná, 100
  - inkrementálna, 101
- nastavenie parametrov
  - adaptácia, 136
  - distribúcia kreditu, 136

- reakcia na udalosti, 137
- dynamické, 132
- endogénne, 131
- exogénne, 131
  - metaevolúcia, 135
  - overené hodnoty, 132
  - pokus-omyl, 134
  - vzorce, 132
- samoadaptácia, 139
- statické, 132
- odchýlka, 67
- operátor
  - diagonálny, 111
  - diferenciálny, 117
  - krížiaci, 106
  - mutačný, 106
  - preusporiadania, 109
    - inverzný, 109
  - rekombinačný, 106
  - skenovací, 110
    - deterministický, 110
    - náhodný, 110
- orezanie, 77
  - deterministické, 102
  - stochastické, 102
  - stupňovité, 78
- plocha vhodnosti, 2
- počet
  - rodičov, 63
  - vyhodnotení, 41
- počet rodičov
  - očakávaný, 68
- polomer zdieľania, 157
- populácia jedincov, 11
  - aktuálna, 14
  - prvotná, 11, 13
- potenciál
  - deštruktívny, 120
  - prehľadávací, 120
  - rekombinačný, 120
- potomok, 105
- prah orezania, 77
- pravdepodobnosť
  - kríženia, 106
  - mutácie, 106
  - rekombinácie, 122–124
  - selekcie, 67, 71
  - zachovania, 122–124
- prehľadávanie náhodné, 73
- premapovanie vhodnosti, 81
  - nemonotónne, 82
  - škálovaním, 82
  - zotriedením, 86
    - exponenciálne, 87
    - lineárne, 86
- premietanie, 182
- priestor
  - kandidátov riešení, 1, 22
  - prehľadávania, 1, 22, 29
- reinizializácia, 152
- reprezentácia, 29
  - hodnoty
    - binárnej, 36
    - reálnej, 34
  - poradia, 37
  - riešenia, 22
  - s dĺžkou
    - pevnou, 31
    - premenlivou, 31
  - s usporiadaním
    - pevným, 31
    - variabilným, 31
  - stromovej štruktúry, 38
- riadenie parametrov, 132
- rodič, 63
- rotačný uhol, 141
- rozdelenie pravdepodobnosti

- normálne, 139
- rozložiteľnosť riešenia, 1, 17
- rôznorodosť
  - jedincov, 5
  - populácie, 147
- rozpätie, 68
  - minimálne, 68
- rýchlosť konvergencie, 149
- samoadaptácia parametrov
  - globálnych, 142
  - lokálnych, 139
- selekcia, 14, 63
  - elitistická, 64
  - proporcionálna, 72
  - s vymieraním
    - sprava, 64
    - zľava, 64
  - uchovávajúca, 64
- selekčná intenzita, 89
  - štandardizovaná, 90
- selekčný
  - šum, 149
  - tlak, 64, 89, 102
- skúmanie
  - priestoru prehľadávania, 4, 13
- spájanie obmedzené, 154
- stavebný blok, 171
- strata rôznorodosti, 95
- stratégia
  - čiarková, 100
  - plusová, 101
- stupeň konvergencie, 148
  - entropia, 148
  - per cento, 148
  - priemerná vzdialenosť, 149
  - variancia, 148
- superjedinec, 72
- súťaž
  - bipartitná, 51
  - obmedzená, 155
  - oddelená, 100
  - plná, 50
  - spoločná, 101
  - turnajová, 51
- šanca na reprodukciu, 63
- škálovací faktor, 84
- škálovanie
  - lineárne, 85
  - sigma, 83
- štruktúra
  - evolučného algoritmu, 11
  - jedinca, 13, 17
- teoréma “No free lunch”, 143
- teória
  - evolučného výberu, 5, 13
  - génovej dedičnosti, 5, 13
- transformácia
  - čísla
    - binárneho na reálne, 35
    - Grayovho na binárne, 35
  - intervalu, 34
- triedenie populácie
  - náhodné, 71
  - vzostupné, 86
- turnaj
  - binárny, 96
  - q-árny, 75
    - bez náhrady, 76
- tvorba novej generácie, 99
- udržiavanie rôznorodosti
  - infúziou materiálu, 151
  - penalizáciou, 156
  - preferenciou odlišnosti, 159
  - udržiavaním subpopulácií, 154
- ukončovacia podmienka, 14
- veľkosť



- 
- populácie, 133, 148
  - turnaja, 75
  - vhodnosť, 2, 45
    - aproximatívna, 47
    - majoritná, 52
    - minoritná, 52
    - off-line, 134
    - on-line, 134
    - populácie
      - najlepšia, 64
      - priemerná, 18, 72
    - relatívna, 50
    - surová, 46
    - súťažná, 50
    - štandardizovaná, 47
    - založená na komplexnosti, 51
  - vizualizácia
    - atribútov, 189
    - jedincov, 181
    - konvergencie, 181
    - populácie, 180
    - vhodnosti, 180
  - výber
    - deterministický, 19
    - pravdepodobnostný, 18
  - výherný automat pákový, 171
  - využívanie známej informácie, 4, 13
  - vzdialenosť
    - Euklidova, 149
    - Hammingova, 149
  - vzorkovanie
    - plochy vhodnosti, 3
    - populácie, 66
    - s pravdepodobnosťou
      - explicitnou, 68
      - implicitnou, 74
    - stochastické
      - bez náhrady, 69
      - s náhradou, 69
      - univerzálne, 70
      - zvyškové, 69
  - windowing, 82
  - zdieľanie, 157

Autor: Marián Mach  
Názov: Evolučné algoritmy: prvky a princípy  
Vydanie: prvé  
Náklad: 100  
Rozsah: 250 strán  
Vydavateľ: elfa s.r.o., Letná 9, Košice  
Sadzba: L<sup>A</sup>T<sub>E</sub>X+ gnuplot + xfig  
Formát: b5  
Tlač: elfa s.r.o., Letná 9, Košice  
Vytlačené: November 2009

ISBN 978-80-8086-123-0  
EAN 9788080861230