

Pokroky matematiky, fyziky a astronomie

Jiří Pospíchal; Vladimír Kvasnička
Genetické algoritmy: nic pro biology

Pokroky matematiky, fyziky a astronomie, Vol. 40 (1995), No. 1, 1–10

Persistent URL: <http://dml.cz/dmlcz/138593>

Terms of use:

© Jednota českých matematiků a fyziků, 1995

Institute of Mathematics of the Czech Academy of Sciences provides access to digitized documents strictly for personal use. Each copy of any part of this document must contain these *Terms of use*.



This document has been digitized, optimized for electronic delivery and stamped with digital signature within the project *DML-CZ: The Czech Digital Mathematics Library* <http://dml.cz>

Genetické algoritmy: nic pro biology

Jiří Pospíchal a Vladimír Kvasnička, Bratislava

1. Úvod

Asi před dvěma desetiletími začaly porůznu mezi počítačovými odborníky klíčit myšlenky, že je třeba hledat poučení v přírodě. Vždyť zatím jen málo a jen specializovaných úkolů se vykonává stroji podstatně lépe než živočichy nebo rostlinami. Tyto myšlenky našly odraz především ve dvou oblastech — ve vytvoření určitého modelu „mozku“ — neuronových sítí, a ve stavbě genetických algoritmů [1–8]. Rozvoj obou těchto metod byl posílen jejich možným budoucím použitím v paralelních počítačích. Obě tyto metody nejsou deterministickými algoritmy, ale využívají náhodných čísel. Na rozdíl od neuronových sítí, které se snaží „polapit“ efektivitu jednotlivého „mozku“ (nebo spíše ganglie), genetické algoritmy svou stavbu spojují s vývojem celého společenství. Snaží se využít genetických představ o hnacích silách evoluce živé hmoty. Není zde tak důležitý jedinec, jako postupný vývoj, kooperace a fungování populace — souboru jedinců. Neúspěšní jedinci vymírají, úspěšní přežívají a množí se. Hybnou silou změn jsou mutace a křížení (výměna „genetické“ informace mezi jedinci). Každý jedinec je v algoritmu reprezentován svým lineárně uspořádaným informačním obsahem (formálně nazývaným chromozom).

Ovšem stejně jako neuronové sítě i genetické algoritmy se v rukou počítačových odborníků postupem času změnilly natolik, že už biologům téměř nic neříkají. Staly se spíše inženýrským nástrojem, a jako běžný případ míry jejich odloučení od biologických věd lze uvést použití genetických algoritmů pro řízení manipulačních robotů. Pro studium procesů v genetice mají už biologové jiné modely, které přesněji popisují specifika dějů v DNA.

V praxi se genetické algoritmy využívají např. pro nastavování parametrů neuronových sítí a v operačním výzkumu. Od biologických počátků se přerodily na negradientový optimalizační algoritmus používaný především pro hledání globálního optima (minima nebo maxima) funkcí s mnoha parametry a s mnoha extrémy. Tyto funkce jsou definovány na složitých, často diskrétních doménách. Výsledkem je jen jedna sada hodnot parametrů, pro kterou má funkce globální extrém. Tato sada představuje jen jeden chromozom z celé populace — ten nejlepší.

Další možností využití genetických algoritmů je použití pro strojové učení s klasifikačními systémy [3] a pro umělou inteligenci, kde je však možné za klasifikační

RNDr. JIŘÍ POSPÍCHAL, CSc. (1961), je odborným asistentem Katedry matematiky CHTF STU, Radlinského 9. 812 37 Bratislava.

Prof. Ing. VLADIMÍR KVASNIČKA, DrSc. (1941), je vedoucím Katedry matematiky tamtéž.

postup z nejobecnějšího hlediska považovat například jakýkoli počítačový program. Chromozomy zde představují pravidla (podmínky pro spuštění akce), kde jedna ze splněných podmínek spouští akci. Splněná pravidla spolu „soutěží“ o řízení akce s dalšími pravidly, jejichž podmínky jsou také splněny. V tomto použití se plně uplatňuje celý soubor zkoumaných chromozomů, neboť pro jednotlivé vstupy se pak vybírají co nejlépe odpovídající chromozomy a většinou neexistuje jedinec, který by byl ideální pro libovolnou možnou situaci. Tomuto přístupu se však zde nebudeme věnovat, neboť vzhledem ke své obsáhlosti by si zasloužil vlastní článek.

2. Základy genetických algoritmů

Ačkoliv genetické algoritmy nemají již prakticky s biologií nic společného, udržely si biologickou terminologii. Omezíme-li použití genetických algoritmů na optimalizaci funkce, pak se za evoluci považují postupné změny proměnných, které vedou k nalezení extrému funkce. Soubor vstupních hodnot proměnných funkce tvoří chromozom. Každému chromozomu je přiřazena nezáporná hodnota, která se nazývá síla chromozomu a odpovídá míře pravděpodobnosti úspěšného přežití. V případě hledání globálního minima, síla chromozomu je nepřímo úměrná funkční hodnotě optimalizované funkce. Mutace je malá náhodná změna jedné nebo několika hodnot proměnných (prvků chromozomu), která ovlivní řešení, ať už kladně nebo záporně. Je nutná k tomu, aby se zamezilo přílišné specializaci, zapadnutí celé populace řešení do jednoho suboptimálního minima, aby vždy byla možnost vytvoření zásadně nových chromozomů odpovídajících lepšímu řešení. Mutace přinášejí do chromozomů novou informaci. Kdyby však existovaly pouze mutace, genetický proces by se nelišil od metody náhodného prohledávání. Efektivita je zajištěna rekombinací neboli křížením, což je zmixování dvou rodičovských chromozomů (souborů proměnných funkce) tak, aby vytvořily jiné dva chromozomy vzájemnou výměnou některých hodnot proměnných. Reprodukci dvou silných jedinců (chromozomů odpovídajících lepším řešením) dostaneme s vysokou pravděpodobností i silné potomky. Populace chromozomů je sada souborů hodnot proměnných, kde každý soubor může dávat jinou funkční hodnotu. Jednotlivé chromozomy bojují o přežití, tj. do následující populace (nové generace) jsou vybírány s větší pravděpodobností soubory hodnot proměnných, které odpovídají lepšímu řešení (extrému funkce). Tento kvazináhodný výběr si můžeme představit pomocí „rulety“, kde každému chromozomu je přiřazen počet čísel na „ruletě“ odpovídající jeho síle. Vypočítejme výsledky funkce udávající sílu chromozomu pro všechny chromozomy z populace a sečteme je do jednoho čísla (celkové „síly“). Vygenerujme potom náhodné číslo z rozmezí mezi nulou a celkovou silou. Výsledkem výběru je první chromozom, jehož výsledek funkce připočtený k výsledkům předcházejících chromozomů je větší nebo roven vygenerovanému náhodnému číslu. Jako příklad si můžeme uvést pět chromozomů s výsledky funkce (3, 2, 1, 3, 4). Celková síla je pak 13. Vygenerujeme-li náhodné číslo rovné 8, pak $3 + 2 + 1 < 8 < 3 + 2 + 1 + 3$, a výsledkem výběru je tedy čtvrtý chromozom. Vybrané chromozomy vytěsňují z populace chromozomy odpovídající méně úspěšným řešením, tak, aby celkový počet chromozomů v populaci

zůstal zachován. Po nějakém počtu výměn generací program konverguje — většina řešení v populaci je blízka optimálnímu řešení, a nejlepší chromozom by měl přesně odpovídat globálnímu optimu.

Genetický algoritmus ve zjednodušené podobě vypadá takto:

1. *Náhodně vygeneruj první generaci chromozomů.*
2. *Vypočítej síly jednotlivých chromozomů na základě jim odpovídajících funkčních hodnot .*
3. *Vytvoř nové chromozomy křížením současných chromozomů vybraných náhodně pomocí rulety s pravděpodobností odpovídající jejich síle.*
4. *Proveď náhodné mutace nových chromozomů.*
5. *Odstraň staré chromozomy (všechny nebo alespoň ty s nejmenší silou), aby se vytvořil prostor pro nové chromozomy.*
6. *Vypočítej síly nových chromozomů a začleň nové chromozomy do populace.*
7. *V případě, že je překročen maximální počet generací nebo splněno jiné kritérium, vrať jako výsledek chromozom s největší silou. Jinak pokračuj krokem 3.*

Evoluční proces v populacích chromozomů odpovídá prohledávání prostoru potenciálních řešení. Takové prohledávání vyžaduje rovnováhu dvou cílů: nalezení nejlepšího řešení v malém okolí výchozího bodu a prohledávání celého prostoru řešení. Optimalizační metoda „hillclimbing“ je příkladem strategie hledající nejlepší řešení v „blízkém“ okolí výchozího bodu, kde se nově nalezené řešení vezme za výchozí bod, a znovu se prohledá „blízké“ okolí. Tento postup se opakuje tak dlouho, dokud se nám v okolí nového výchozího bodu podaří nalézt lepší bod. (Počáteční bod je zvolen náhodně.) Tento přístup však zanedbává prohledávání celého prostoru řešení. Proto velmi pravděpodobně bude nalezené řešení úlohy hledání globálního optima pouze suboptimální. Oproti tomu u stochastického (slepého) prohledávání se pokaždé vyhodnocuje jiný náhodně zvolený bod bez návaznosti na předchozí výsledky. Tak se prozkoumá velmi mnoho oblastí, ale zanedbává se podrobnější výzkum slibných lokalit. Nalezení globálního extrému funkce tímto způsobem by si vyžádalo absurdně dlouhou dobu. Genetické algoritmy představují rovnováhu mezi náhodným a směřovaným prohledáváním. Jsou vhodné tam, kde gradientové nebo simplexové metody selhávají buď pro nemožnost určit gradient, nebo pro příliš velké množství lokálních extrémů funkce.

Proč při definici genetických algoritmů stále zdůrazňujeme náhodnost výběru chromozomů? Kdybychom do reprodukčního procesu vybírali jen chromozomy s největší silou, potom bychom velmi pravděpodobně podstatně ohraničili doménu, na které hledáme optimální výsledek. Genetický algoritmus by se stal velmi „oportunistickým“, za dočasný lepší výsledek bychom velmi pravděpodobně dostali horší konečný výsledek. Chromozom s menší silou může stále ještě obsahovat důležitou informaci využitelnou v budoucí evoluci populace. V genetickém algoritmu nemůžeme zasahovat výběrem chromozomů na křížení a vytěšňování například tak, aby tyto chromozomy splňovaly určité podmínky. Kdybychom poznali takovou metodu, která nám řekne, které chromozomy dají v budoucnu lepší výsledek, nepotřebovali bychom genetický algoritmus. Všechny „částečně urychlující“ heuristiky jsou nejen nedostatečné, ale v konečném důsledku i zavádějící, proto se je ani nepokoušíme do genetického algoritmu zabudovat.

Genetický algoritmus je založen na vhodné reprezentaci dat potenciálního řešení problému a musí obsahovat definici výměny informací, která vytváří z rodičovských řešení nová řešení — potomky. K hlavním parametrům algoritmu patří velikost populace (počet chromozomů) a pravděpodobnost mutace a křížení, spolu s metodou (procentem) vymírání méně úspěšných chromozomů. Ty nejlepší chromozomy se mohou popřípadě převzít do nové populace všechny, a i nejlepší rodičovské chromozomy mohou případně přežít (ale jen malé procento z nich, abychom neohranili prostor prohledávání).

3. Bitová reprezentace proměnných a optimalizace funkce

Velká část řešení pomocí genetických algoritmů je umožněna zápisem hodnot proměnných do bitového řetězce. Pro převedení reálného čísla x na binární reprezentaci musíme nejprve zvolit interval $(-a, a)$, ve kterém se naše číslo nachází a přesnost na q desetinných míst. Vynásobíme-li interval $-a \leq x \leq a$ číslem 10^q a připočteme-li $a \cdot 10^q$, dostaneme $0 \leq x \cdot 10^q + a \cdot 10^q \leq 2a \cdot 10^q$. Číslo x je tak převedeno na nezáporné celé číslo $y = x \cdot 10^q + a \cdot 10^q$. Délka m binárního čísla, které bude reprezentovat číslo y je určena vztahem $m = \lceil \ln(2a \cdot 10^q) / \ln 2 \rceil$, kde $\lceil z \rceil$ je nejmenší celé číslo větší než z . Bitový řetězec složený z jedné nebo několika za sebou zařazených binárních hodnot proměnných je možno brát jako chromozom, a počet nul a jedniček v tomto bitovém řetězci nazveme délkou k chromozomu. Je-li počet proměnných optimalizované funkce n , potom $k = n \cdot m$. Počet chromozomů v populaci by měl být sice mnohem menší než celkový možný počet rozdílných chromozomů 2^k , ale mnohem větší než je délka chromozomu k , tak aby byl tento počet „statisticky významný“. Náhodnou výměnou některé nuly za jedničku nebo naopak se provádí mutace. Je-li pravděpodobnost mutace 0,01, pak pro každou pozici ve všech chromozomech populace vygenerujeme náhodné číslo z intervalu $(0,1)$. Je-li toto číslo větší než 0,01, pak neděláme nic, je-li vygenerované číslo menší než 0,01, pak u dané pozice zaměníme bit za jeho komplement (0 za 1 a obráceně, 1 za 0). Mutace je ilustrována tímto příkladem: $A = 0010101011$ je původní chromozom délky 10, to znamená, že desetkrát generujeme náhodné číslo z intervalu $(0,1)$. Když jenom osmé náhodně generované číslo je menší než 0,01, potom v osmé poloze změňme bit, v našem případě z 0 na 1, a dostaneme nový chromozom A' ,

$$A = 00101010\underline{1}1 \implies A' = 0010101\underline{1}11,$$

kde podtržený bit označuje místo mutace.

Vezmeme-li dva bitové řetězce A a B a vyměníme jejich části, dostaneme křížení — výměnu informací. To si můžeme znázornit takto: Náhodně si vybereme místo, ve kterém se budou bitové řetězce křížit (vygenerujeme číslo z intervalu od 1 do $k-1$) a podřetězce od začátku bitových řetězců až do místa křížení jednoduše vyměníme,

$$\begin{aligned} A = 0010101|011 &\implies A' = 1101010|011, \\ B = 1101010|101 &\implies B' = 0010101|101, \end{aligned}$$

kde vertikální čára označuje místo křížení. Jak je vidět z příkladu, křížením se může vyměnit třeba jen část binárního řetězce odpovídajícího jedné proměnné, není zde třeba rozlišovat podřetězce odpovídající jedné proměnné.

Otázka je, proč vlastně genetické algoritmy fungují. Příčinu je třeba hledat především ve výměně informací mezi chromozomy. Můžeme si představit, že některé pozice ovlivňují řešení více než jiné, a že kombinace těchto pozic může působit nelineárně, t.j. výsledek není součtem vlivu izolovaných změn, ale spíše jejich násobkem. Důvod, proč v tvorbě takto výhodně součinných pozic funguje tak dobře křížení, hledáme pomocí tzv. schémat. Ve schématu si v bitovém řetězci nahradíme ty hodnoty, na kterých hodnota funkce tolik nezáleží, hvězdičkami. U hvězdiček je jedno, jestli bitový řetězec nabývá hodnot 0 nebo 1. Genetický algoritmus se snaží prohledávat mnoho oblastí fázového prostoru zároveň. Nechť dva podřetězce R a S , které se vyskytují v dvou různých nepřekrývajících se částech řetězce, podstatně zvyšují sílu chromozomu. Jestli pravděpodobnost výskytu každého z nich je 10^{-6} , potom se pravděpodobnost jejich současného výskytu rovná součinu $10^{-6} \cdot 10^{-6} = 10^{-12}$. Současný výskyt R a S v chromozomu jen v důsledku mutací je tedy vysoce nepravděpodobný. Avšak vyskytují-li se chromozomy s izolovanými podřetězci R a S už v populaci, křížení sloučí tyto podřetězce s velkou pravděpodobností do jednoho chromozomu.

Důvodem výrazně vyšší efektivity genetických algoritmů oproti náhodnému výběru je fakt, že chromozom lze považovat za umístěný ve více schématech. Máme-li chromozom 11010, je tento jediný chromozom členem schématu 11***, ale také třeba schématu **0*0 a mnoha dalších. Relativně malý počet chromozomů tak mapuje velké množství schémat. Tento implicitní paralelizmus je patrně jedním z klíčových faktorů úspěchu genetických algoritmů. Křížení tento efekt implicitního paralelizmu poněkud komplikuje, poněvadž sice schémata spojuje, ale také často rozbíjí, tím pravděpodobněji, čím více jsou schémata rozložena po délce chromozomu. Tím, že dochází k rozbití schémat, dochází ale také k vytvoření nových schémat a výsledné chromozomy mohou patřit do oblasti fázového prostoru řešení, do které nepatřil ani jeden z rodičů. Vzhledem k tomu, že méně často dochází k rozbití schémat nul a jedniček umístěných blízko sebe, genetické algoritmy zvláště dobře prohledávají možná řešení obsahující v části chromozomu podřetězec nul a jedniček kladně ovlivňující sílu chromozomu. Dochází tak k zvýšené produkci nelineárně podporovaných podřetězců. To je vyjádřené Hollandovou větou z r. 1975, která tvoří teoretický základ genetických algoritmů [1]:

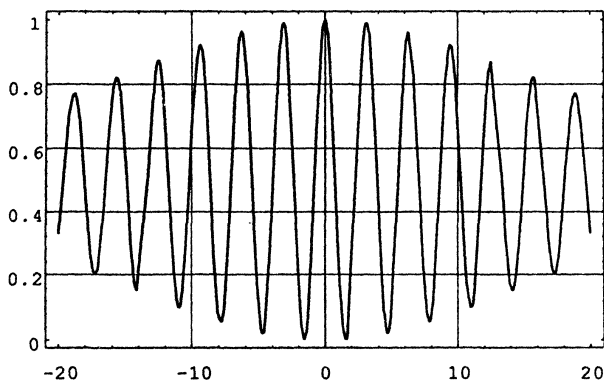
VĚTA. Nechť r je střední hodnota síly všech chromozomů v populaci, které obsahují schéma, n je počet těchto chromozomů a konečně nechť a je střední hodnota síly všech chromozomů v populaci. Potom očekávaný počet chromozomů obsahujících schéma v následující generaci je $n \cdot r/a$ – (počet zániků schématu v důsledku křížení a mutací).

Tato věta říká, že efektivní schéma se vyskytuje v následující generaci s rostoucí frekvencí, a naopak, neefektivní schéma se vyskytuje s klesající frekvencí. Efektivita schématu závisí na poměru r/a . Jestliže platí $r/a > 1$ (tj. $a < r$, čili střední síla všech chromozomů populace je menší než střední síla chromozomů obsahujících schéma), počet chromozomů obsahujících schéma roste. V opačném případě, když $r/a < 1$,

potom počet těchto chromozomů klesá. Tyto jednoduché úvahy jsou založeny na předpokladu, že počet zániků schémat v chromozomech v důsledků křížení nebo mutací je podstatně menší než počet jeho opakovaných vzniků v procesu reprodukce.

Chromozomy se postupně shromažďují v oblastech odpovídajících lepším řešením. To odpovídá schopnosti kombinovat v chromozomech potomků části řešení nacházející se v rodičovských chromozomech. Jako se kombinují výhodné části řešení, tak se samozřejmě také mohou zkombinovat nevýhodné části řešení, ale takový potomek pravděpodobně vymře. Mutace jako náhodná změna chromozomu tvoří jen neporovnatelně malou část změn v porovnání s křížením. Obvyčejně se udává, že vhodná průměrná frekvence mutací je jedno náhodné prohození nuly a jedničky na deset tisíc bitů. Mutace neurychluje nalezení řešení, ale spíše zajišťuje možnost další evoluce v případě, že se všechny chromozomy v průběhu generací následkem křížení nahnou do jedné oblasti.

Odstranění omezení průzkumu schémat na víceméně souvislé části chromozomů se snaží napravit operace tzv. inverze [1]. Ta může přeskupit chromozomy tak, že vstupy umístěné v původním chromozomu daleko od sebe, budou v novém chromozomu u sebe. To odpovídá předefinování schémat, která jsou víceméně souvislá (tj. sekvence nul a jedniček ve schématu není přerušovaná hvězdičkami) a ve kterých je proto průzkum intenzivnější, neboť nejsou tak často rozbíjena křížením. Avšak tento přístup se nejeví příliš úspěšný, neboť je třeba si pamatovat původní pozice před přeskupením chromozomu. Není předem jasné, zda nové souvislé oblasti budou úspěšnější, a prováděli se inverze příliš často, výsledkem je pak prohledávání zase „širší“, ale méně „hluboké“ a genetický algoritmus špatně konverguje.



Obr. 1. Graf funkce $f(x) = 0,5 - (\sin^2 |x| - 0,5)/(1,0 + 0,001x^2)^2$, která je použita pro testování efektivity hledání globálního minima pomocí genetického algoritmu.

Uvedeme si nyní jako příklad hledání globálního minima funkce [9]

$$f(x) = 0,5 - \frac{\sin^2 |x| - 0,5}{(1,0 + 0,001x^2)^2}.$$

Tato funkce má v blízkosti nuly velký počet lokálních minim (viz obr. 1), z toho důvodu je velmi vhodným testovacím příkladem pro použití genetického algoritmu na

hledání minima blízkého globálnímu. První tři minimální hodnoty funkce (vypočtené Newtonovou a Raphsonovou metodou):

$$f_{\min} = 0,0024559 \quad \text{pro } x_1 = \pm 1,56923,$$

$$f_{\min} = 0,0214679 \quad \text{pro } x_2 = \pm 4,70778,$$

$$f_{\min} = 0,0563647 \quad \text{pro } x_3 = \pm 7,84659.$$

Aplikace genetického algoritmu k minimalizaci této funkce byla úspěšná. Délka chromozomu byla zvolena 22 bitů, 22bitový podřetězec reprezentuje reálnou proměnnou x s přesností na 5 míst za desetinnou čárkou a hodnota reálné proměnné x je brána z intervalu $-20,07152 \leq x \leq 20,07152$. Populace obsahovala 200 chromozomů, výsledné minimální hodnoty funkce asi po 200–300 iteracích byly už rovné absolutnímu minimu funkce $f_{\min} = 0,00246$.

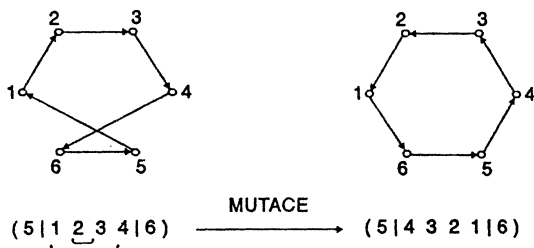
4. Optimalizace diskretních funkcí

Velkou většinu aplikací genetických algoritmů tvoří právě popsaná optimalizace reálných funkcí s proměnnými reprezentovanými bitovým řetězcem. To by nás však nemělo svést k přesvědčení, že to je jediný způsob využití genetických algoritmů. Jako příklad odlišné reprezentace dat, mutací i křížení můžeme uvést třeba řešení známého problému obchodního cestujícího [10]. Obchodní cestující má zadán seznam měst, které má navštívit. Každé město má navštívit pouze jednou, a po průchodu všemi městy se má vrátit do města, ze kterého vyšel. Předpokládá se, že každá dvojice měst je spojena silnicí, která neprochází žádným jiným městem (nebo obchodní cestující cestuje letadlem). Vzdálenost mezi dvojicemi měst je známa. Úkolem je určit co nejkratší trasu.

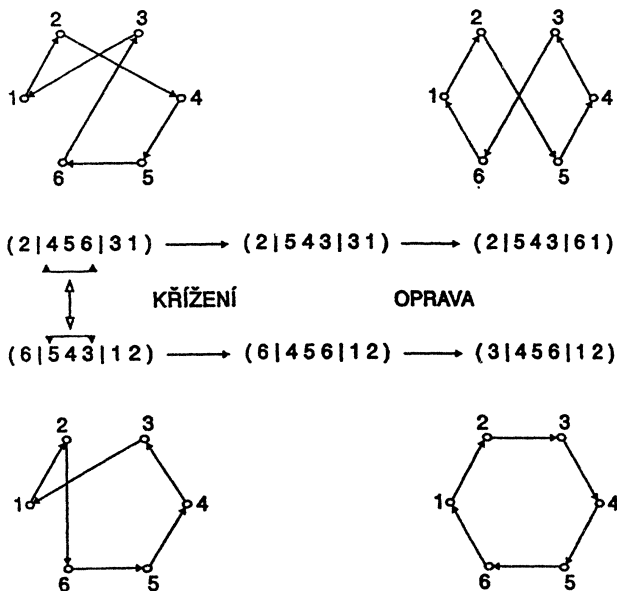
Máme-li N měst, můžeme si je očíslovat od 1 po N . Každá permutace N čísel nám pak dává jednu trasu (předpokládá se, že z města odpovídajícího poslednímu číslu v permutaci trasa pokračuje do města odpovídajícího prvnímu číslu v permutaci). Každou permutaci tedy můžeme brát jako jeden chromozom.

Mutaci můžeme definovat třeba tak, že náhodně určenou část permutace „vyřízeme“ a umístíme v opačném pořadí (viz obr. 2). Křížení si představíme jako vzájemnou výměnu částí dvou permutací od náhodně určené i -té pozice po náhodně určenou j -tou pozici (viz obr. 3). Po výměně těchto částí permutací se bohužel může stát, že výsledné chromozomy nejsou permutacemi, poněvadž některá čísla se opakují, a jiná nejsou v chromozomu obsažena. V takovém případě je třeba původní nevytěžené části chromozomů „opravit“ [3], aby se chromozomy zase staly permutacemi. Vyskytuje-li se po výměně číslo a v novém chromozomu dvakrát, a to na místě p v původní nevytěžené části a na místě q v nové části vzniklé výměnou, nahradíme jeho výskyt na místě p číslem b , které bylo v původním chromozomu na místě q . Pokud i číslo b je nyní v chromozomu dvakrát, postupujeme stejně jako v případě čísla a , dokud se na daném místě neocitne číslo, které se už v řetězci neopakuje. Tyto výměny provádíme v nevytěžených částech chromozomu tak dlouho, dokud se chromozom nestane zase permutací.

Určení nejkratší cesty je v obecném případě faktoriálový problém, poněvadž je nutné prozkoumat $\frac{1}{2}(N-1)!$ permutací. To způsobuje těžkosti při ověření správnosti výsledku. Pro snadné ověření výsledné nejkratší cesty lze využít speciálního umístění měst na pravoúhlé mřížce, kde vzdálenost mezi sousedními městy je jednotková. Ostatní vzdálenosti jsou určeny Hammingovou metrikou — vzdálenost počítáme, jako bychom se mohli pohybovat jen po dané pravoúhlé mřížce. V takovém případě je pro sudý

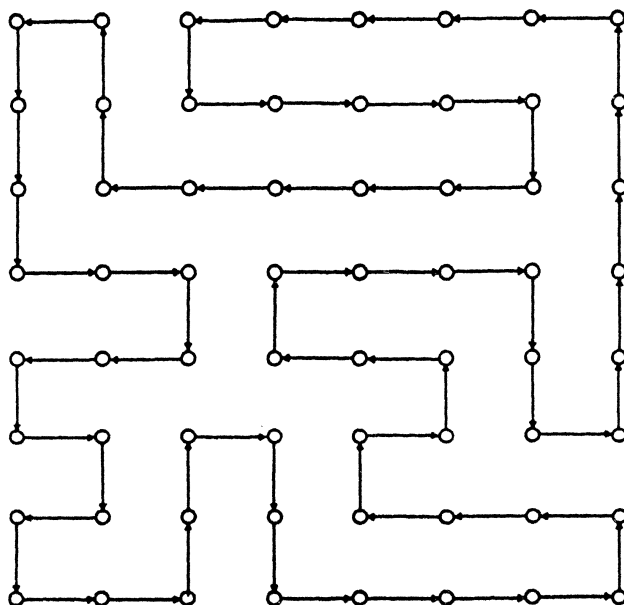


Obr. 2. Zobrazení mutace při hledání nejkratší cesty pro spojení šesti měst (na obr. kroužky) popsaných čísly od 1 do 6. Permutace těchto čísel tvoří jednu trasu — chromozom. Cesta jde od jednoho města (číslo) v permutaci k následujícímu v pořadí, a od posledního města v permutaci pokračuje k prvnímu. Část tohoto chromozomu označená vertikálními čárkami byla mutací „vyříznuta“ a umístěna v opačném pořadí.



Obr. 3. Zobrazení křížení při hledání nejkratší cesty pro spojení šesti měst. Dva chromozomy si vymění část svého řetězce (část permutace); protože výsledné chromozomy nejsou permutacemi, musí následovat oprava, aby chromozom určoval cestu obsahující všechna města bez opakování.

počet měst celková délka trasy rovna N^2 , všechna spojení jsou umístěna na pravoúhlé mřížce. Pro lichý počet měst je délka trasy $N^2 + 1$, jedno ze spojení je umístěno na diagonále základního čtverce mřížky. Výsledek algoritmu pro mřížku velikosti 8×8 můžeme vidět na obr. 4.



Obr. 4. Zobrazení jedné z optimálních cest nalezených genetickým algoritmem na mřížce 8×8 .

5. Závěr

Genetické algoritmy spolu s metodou „simulovaného žhání“ [2] tvoří nové pole výzkumu optimalizačních metod. Je třeba zdůraznit, že ve všeobecnosti mohou tyto metody skončit v suboptimu, i když většinou se jim daří nalézt globální optimum. Genetické algoritmy mají oproti simulovanému žhání výhodu možné budoucí výhodné implementace na paralelních počítačích. Fungují především tam, kde jiné metody pro velikost úkolu v praxi selhávají; pro jednodušší funkce, kde klasické metody fungují uspokojivě, nemá smysl genetických algoritmů používat.

Výhodou genetických algoritmů je jejich obecnost, dají se upravit pro řešení nej-různějších úkolů. Nevýhody genetických algoritmů vyplývají také z jejich obecné formulace, je zde spousta parametrů pro nastavení, široký výběr reprezentace dat, definicí mutace a křížení. Z důvodů této obecnosti neexistuje pro genetické algoritmy hlubší teorie, která by zásadně pomohla při výběru reprezentace dat a nastavování parametrů. To je třeba v praxi provádět spíše metodou pokusu a omylu. Přesto se genetické algoritmy stále více využívají — nic lepšího zatím k dispozici není.

L i t e r a t u r a

- [1] J. HOLLAND: *Adaptation in Natural and Artificial Systems*. University of Michigan Press, Ann Arbor, 1975.
- [2] L. DAVIS (ed): *Genetic Algorithms and Simulated Annealing*. Morgan Kaufman, Los Altos, 1987.
- [3] D. GOLDBERG: *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley, 1989.
- [4] L. DAVIS (ed): *Handbook of Genetic Algorithms*. Van Nostrand Reinhold, New York, 1991.
- [5] L. D. WHITLEY: *Foundation of Genetic Algorithms 2*. Morgan Kaufmann Publishers, San Mateo, CA, 1993.
- [6] J. J. GREFENSTETTE (ed): *Genetic Algorithms and Their Application*. Proceedings of the Second International Conference on Genetic Algorithms. Lawrence Erlbaum Associates, Hillsdale, NJ, 1987.
- [7] R. K. BELLEW, L. B. BOOKER (ed): *Proceedings of the Fourth International Conference on Genetic Algorithms*. Morgan Kaufmann Publishers, Los Altos, CA, 1991.
- [8] Z. MICHALEWICZ: *Genetic Algorithms + Data Structures = Evolution Programs*. Springer Verlag, Berlin, 1992.
- [9] J. D. SCHAFFER, R. A. CARUANA, L. J. ESHELMAN, R. DAS: *A study of control parameters affecting online performance of genetic algorithms for function optimization*. Ve J. D. SCHAFFER (ed): *Proceedings of the Third International Conference on Genetic Algorithms*. Morgan Kaufmann Publishers, Los Altos, CA, 1989.
- [10] D. GOLDBERG, R. LINGLE: *Alleles, loci, and the traveling salesman problem*. Ve J. J. GREFENSTETTE (ed): *Proceedings of an International Conference on Genetic Algorithms and Their Applications*. Lawrence Erlbaum Associates, Hillsdale, NJ, 1988. (original proceedings 1985).