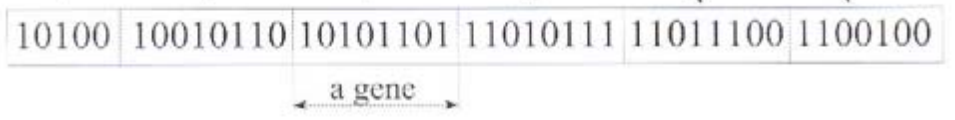


The set of an individual's attributes (genes)

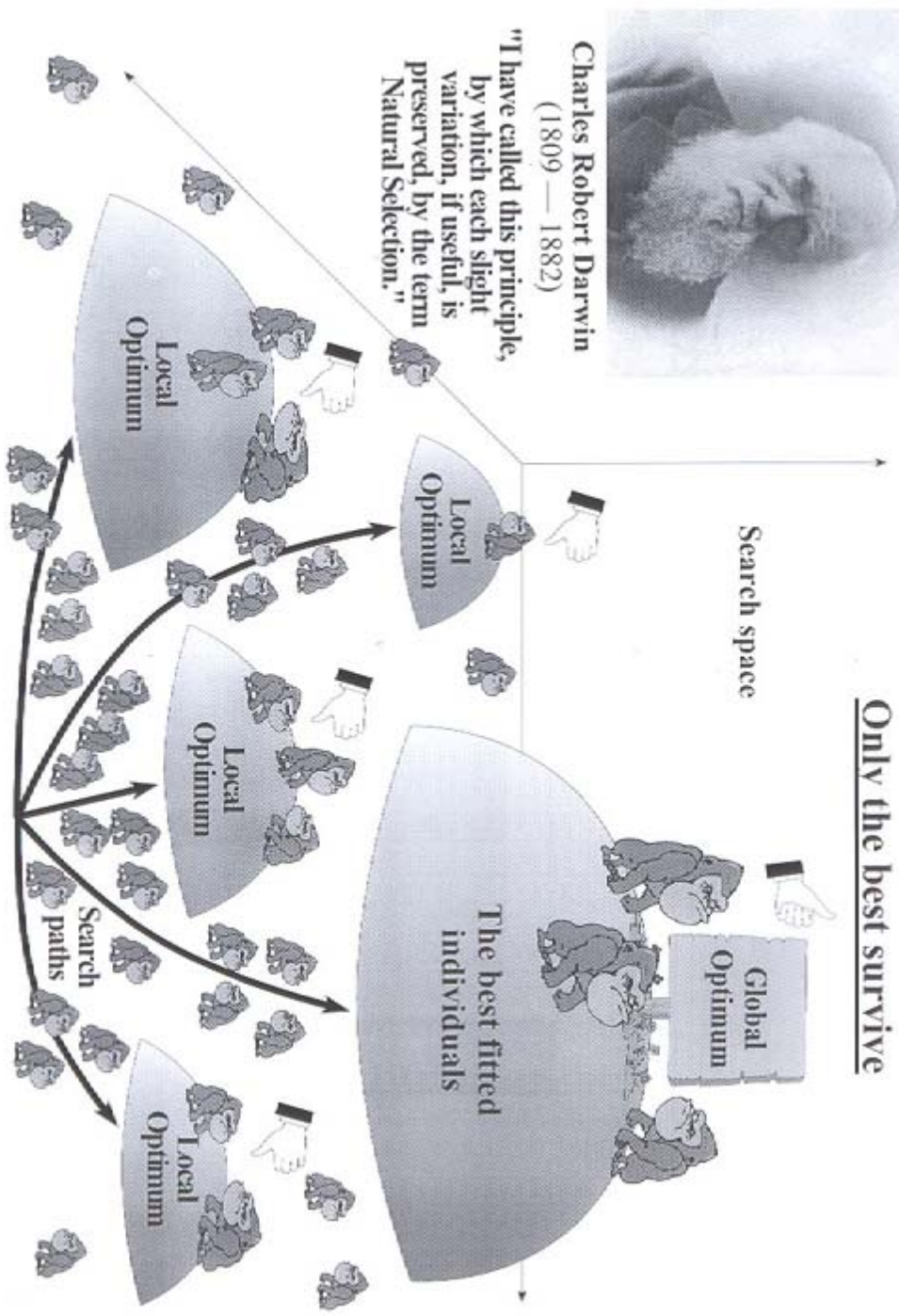
Age	IQ	Width	Height	Weight	Vigour
20	150	173.5	215.0	220.6	100%

Encoding into a bit-string (a chromosome)

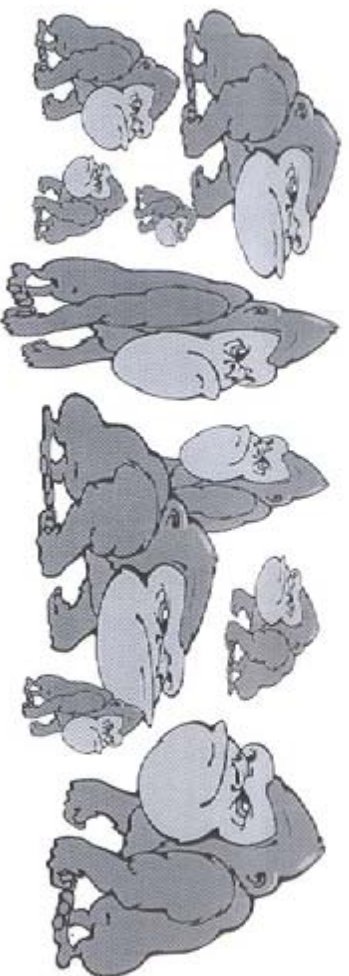


Charles Robert Darwin  
(1809 — 1882)

"I have called this principle, by which each slight variation, if useful, is preserved, by the term Natural Selection."

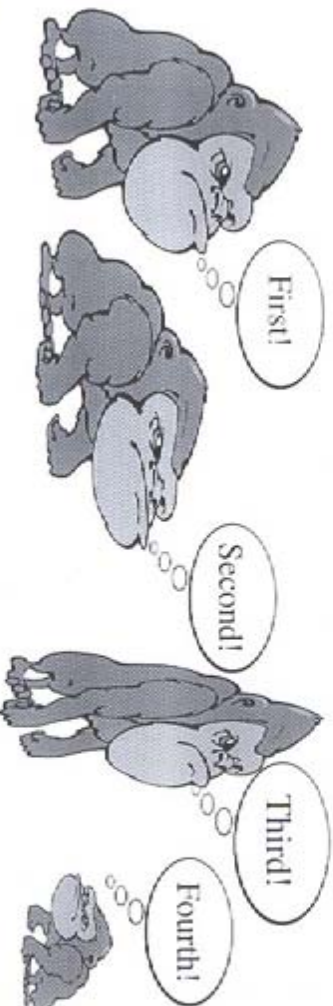


**Only the best survive**



1. Randomly generate the initial population (as large and diverse as possible)

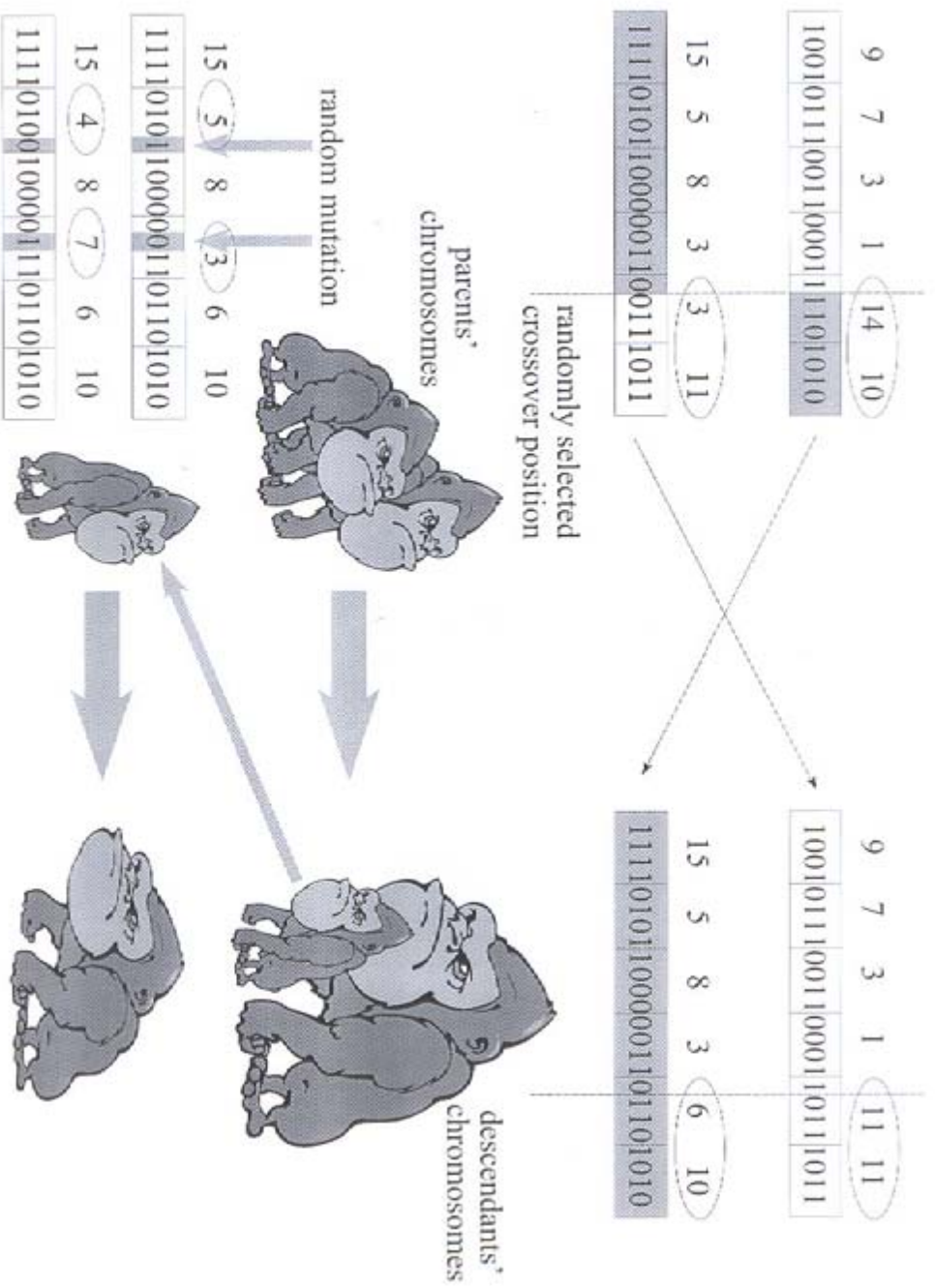
2. Perform the necessary genetic operations (crossover, mutation)



3. Evaluate individual solutions by calculating their scores using an objective function

4. Select the best (the fittest) individuals with the highest scores

5. If necessary, repeat again from the step No. 2



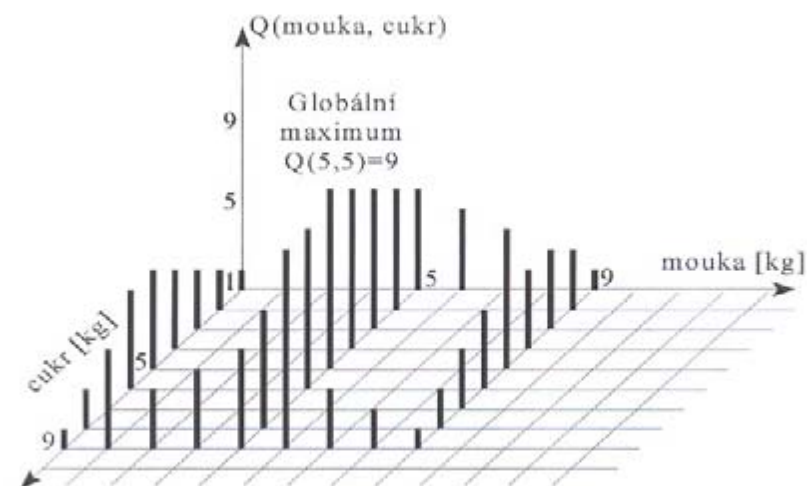


## GENETICKÉ ALGORITMY (GA)

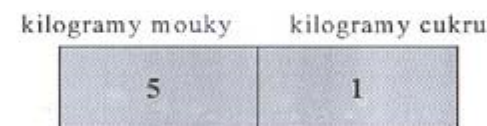
- Pro ilustraci činnosti GA uvažme problém, kdy kuchař hledá k upečení koláčů optimální poměr cukru a mouky, tj. hledá se řešení v dvourozměrném prostoru, kde na jedné ose je množství mouky v kg a na druhé ose množství cukru v kg. Kvalita výsledku je dána jako tabelovaná funkce  $Q(\text{mouka}, \text{cukr})$ . Hodnoty jsou nějak určeny, což na tomto místě není důležité. Funkce kvality má formu pahorku (1 = nejnižší kvalita, 9 = nejvyšší kvalita):

cukr[kg]	9	1	2	3	4	5	4	3	2	1
8	2	3	4	5	6	5	4	3	2	
7	3	4	5	6	7	6	5	4	3	
6	4	5	6	7	8	7	6	5	4	
5	5	6	7	8	9	8	7	6	5	
4	4	5	6	7	8	7	6	5	4	
3	3	4	5	6	7	6	5	4	3	
2	2	3	4	5	6	5	4	3	2	
1	1	2	3	4	5	4	3	2	1	
		1	2	3	4	5	6	7	8	9
										mouka [kg]

- Bylo by ovšem zde možné vyzkoušet každou kombinaci (těch je pouze  $9 \times 9 = 81$ ), tj. úplné prohledání prostoru řešení, avšak *s rostoucím počtem hodnot každé proměnné a se zvyšující se dimensionalitou prostoru složitost prudce roste* a testování "hrubou silou" se stává prakticky nepoužitelné.



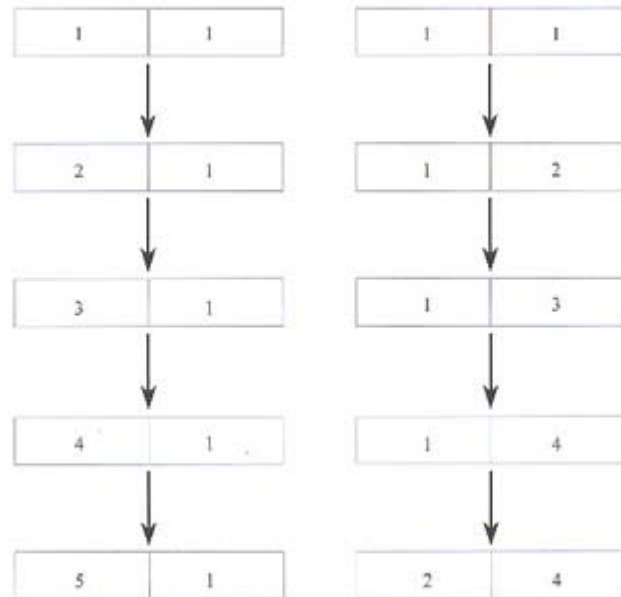
- Kuchař chce najít optimální složení směsi bez zkoušení každé kombinace. Protože také studoval umělou inteligenci, dozvěděl se o metodě *genetických algoritmů* a rozhodl se ji aplikovat na své koláče. Nejprve je nutno vytvořit analogii individuí, chromosomů, mutace, křížení, měření přizpůsobenosti, a přirozeného výběru:
  - *Individuum* (jedinec) je každá dávka upečených koláčů.
  - *Chromosom* se skládá ze dvou "genů", z nichž každý může nabývat hodnoty 1...9. První gen určuje množství mouky, druhý množství cukru:



**chromosom** je reprezentace s následujícími vlastnostmi:

- ★ obsahuje seznam elementů zvaných *geny*;
- ★ určuje celkovou přizpůsobenost vyjadřovanou nějakým mechanismem, používajícím geny jako plán (program);
- ★ disponuje konstruktory jež umí vytvářet chromosomy pomocí elementů, dále pomocí křížení párů již existujících chromosomů;
- ★ má k dispozici operátory pro mutaci existujících chromosomů změnou genů;
- ★ umí vytvářet pomocí chromosomu specifikovaný gen.

- ▶ Kuchař se dále rozhodne, že každý jedinec bude mít pouze jednu kopii určitého chromosomu (což je obvyklé u nižších rostlin a živočichů).
- ▶ K napodobení mutace chromosomů zde bude náhodně vybrán jeden z genů a náhodně změněn přičtením nebo odečtením čísla 1 (s ohledem na interval 1...9).
- ▶ Následující obrázek ukazuje, jak by se mohly dva chromosomy vyvíjet během série čtyř šťastných mutací, které by produkovaly jedince se stoupající kvalitou:



- Po provedení 1000 opakovaných pokusů dle uvedených bodů lze např. zjistit, že nalezení optimální směsi obou ingrediencí nastane průměrně v 16. generaci. Nejrychlejší (nejšťastnější) sekvence pokusů dala optimum v 8. generaci:

- ★ Generace 0 (chromosom, kvalita):  
1-1      1  
(mutace vytvořila chromosom 1-2)
- ★ Generace 1  
1-2      2  
1-1      1  
(1-2 mutoval na 1-3, 1-1 na 1-2 a nebyl přidán)
- ★ Generace 2  
1-3      3  
1-2      2  
1-1      1  
(poprvé počet jedinců převýšil povolené maximum 4 — přidáním nových mutantů vznikla populace s 6 jedinci, kteří byli omezeni na 4)  
1-4      4 (nejlepší přežije vždy)  
1-3      3 (plus tři další vybrání náhodně ze zbytku)  
1-2      2  
2-1      2  
2-2      3 (nepřežil)  
1-1      1 (nepřežil)  
(dále mutace přidala 3 nové chromosomy)
- ★ Generace 4  
2-4      5  
2-3      4  
3-1      3  
(z celkem 7 byli vybráni 4 jedinci)  
(dále již bez detailů . . .)
- ★ Generace 4  
2-4      5  
1-4      4  
1-3      3  
2-1      2

★	Generace 5	
	2-5	6
	1-5	5
	2-3	4
	2-2	3

★	Generace 6	
	3-5	7
	1-5	5
	3-2	4
	1-4	4

★	Generace 7	
	4-5	8
	1-5	5
	1-4	4
	3-1	3

(nyní 4-5 mutoval na 5-5, což je optimum, a experiment skončil)

★	Generace 8	
	5-5	9
	4-5	8
	2-5	6
	2-1	2

- Je zřejmé, že pro jednoduchý "terén" s jedním pahorkem není křížení zapotřebí.

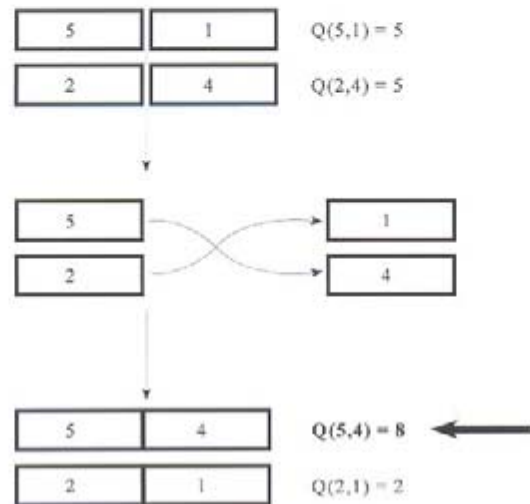
- GA obecně zahrnují nutnost volby mnoha parametrů:
  - ▶ Jak mnoho chromosomů má být v populaci? Je-li počet příliš nízký, všechny chromosomy se brzy stanou téměř či úplně identické a křížení nepřinese nic nového. Je-li počet příliš vysoký, výpočetní doba bude neúnosně dlouhá.
  - ▶ Jaký má být poměr mutací? Příliš nízký poměr způsobí, že se nové vlastnosti v populaci budou objevovat velmi pomalu. V opačném případě bude mít nová generace málo společného s předchozí.
  - ▶ Používat spojování chromosomů? Když ano, jak vybírat dvojice a jak určovat bod křížení?
  - ▶ Může se libovolný chromosom vyskytnout v populaci více než jedenkrát?

Všeobecně je přínosem vědět co nejvíce o prohledávaném prostoru (což odpovídá principu, že je nejlepší znát odpověď před tím, než se začne problém řešit).

- Vždy je snadné hledat optimum bez křížení:
  - ▶ začne se s chromosomem na pozici 1-1
  - ▶ žádný chromosom se nesmí vyskytnout více než jedenkrát v každé generaci
  - ▶ nanejvýš čtyři chromosomy přežívají z jedné generace do druhé
  - ▶ přeživší jedinci mohou spolu s novými přežít do následující generace
  - ▶ jeden gen je vybrán náhodně v každém přeživším ~~genu~~ <sup>chromosomu</sup> a je náhodně mutován; je-li mutant odlišný od ostatních kandidátů na přežití, je zařazen do populace
  - ▶ nepoužívá se křížení
  - ▶ chromosom s nejvyšším skóre vždy přežívá
  - ▶ ostatní jsou vybráni náhodně.



- ▶ Chromosomy mají tzv. *skóre kvality* a vytvářejí *populace*, i když ve skutečnosti jsou populace vytvářeny jedinci — nositeli chromosomů.
- ▶ K napodobení *křížení chromosomů* kuchař chromosomy rozdělí uprostřed na dvě části a tyto části spojí:



**přírodní výběr** se obecně napodobuje následujícím postupem (s možnými variacemi):

- ★ vytvoř počáteční "populaci" z jednoho chromosomu;
- ★ mutuj jeden či více genů v jednom či více existujících genů, což vytvoří jednoho nového potomka pro každý z mutovaných chromosomů;
- ★ spáruj jeden či více chromosomů;
- ★ přidej potomky a mutanty do populace;
- ★ vytvoř novou generaci zachováním nejlepších jedinců současné populace spolu s náhodně vybranými jedinci z horšího zbytku; výběr směřuj vzhledem k přizpůsobenosti.

- Standardní metoda určování přizpůsobenosti používá *relativní kvalitu*: Je-li zvolen způsob, jak napodobovat mutaci a křížení, je zapotřebí najít analogii k *přizpůsobenosti* a *přírodnímu výběru*.

Všeobecně platí, že míra přizpůsobenosti chromosomu je pravděpodobnost jeho přežití do další generace. Je nutné tedy stanovit vzorec pro výpočet vztahu mezi přizpůsobeností  $i$ -tého chromosomu  $f_i$  (což je pravděpodobnost nabývající hodnot mezi 0.0 a 1.0), a jeho kvalitou  $q_i$  (což je v uvedeném příkladě s koláči číslo mezi 1 a 9). Jednou z možností je následující vztah:

$$f_i = \frac{q_i}{\sum_j q_j}$$

což je poměr kvality jedince k celkové kvalitě populace. Pro uvedený vztah se vžil název *standardní metoda*.

Příklad: populace složená z jedinců s chromosomy **1-4**, **3-1**, **1-2** a **1-1** bude mít [s využitím tabulky  $Q(\text{mouka}, \text{cukr})$ ] následující kvality a z nich vypočítané pravděpodobnosti přežití do další generace:

chromosomy	kvalita $q$	přizpůsobenost $f$
<b>1 — 4</b>	4	0.40
<b>3 — 1</b>	3	0.30
<b>1 — 2</b>	2	0.20
<b>1 — 1</b>	1	0.10

- Seřazovací metoda spojuje přizpůsobenost a pořadí kvality:

Standardní metoda neumožňuje ovlivnit výběr. *Seřazovací metoda* nabízí prostředek k usměrnění výběru směrem k nejlepšímu chromosomu a eliminuje implicitní tendence dané nevhodnými výběry standardní funkce metriky.

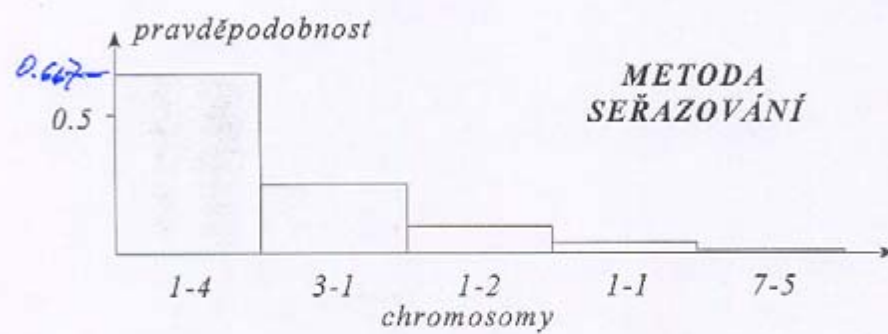
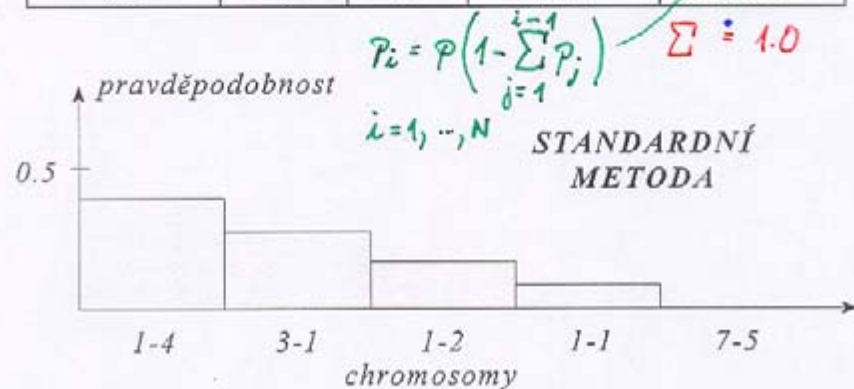
V principu seřazovací metoda ignoruje míru kvality, využívá ji pouze pro seřazování kandidátů od nejvyšší kvality po nejnižší. Přizpůsobenost nejlepšího kandidáta je pak rovna nějaké konstantě  $p$ . Není-li ten nejlepší vybrán, pak další v pořadí dostane tuto hodnotu  $p$ . Tento proces pokračuje tak dlouho, dokud není konečně nějaký jedinec vybrán nebo pokud nezůstává již jen jeden (a ten je pak zvolen):

- ★ seřaď  $n$  jedinců podle kvality;
- ★ necht' pravděpodobnost výběru  $i$ -tého kandidáta (za předpokladu, že prvních  $i-1$  kandidátů nebylo vybráno) je  $p$  s výjimkou kandidáta posledního, jenž je vybrán pokud žádný z předchozích jedinců vybrán nebyl;
- ★ vyber kandidáta pomocí vypočítaných pravděpodobností.

Příklad: necht'  $p=0.667$  a necht' se jedná o tytéž chromosomy jako v již uvažované situaci (viz standardní metodu), tj. 1-4, 3-1, 1-2, 1-1. Předpokládejme existenci příkopu a dále, že uvedené 4 chromosomy byly rozšířeny o 7-5 (jenž produkuje nulovou kvalitu koláčů).

Následující tabulka a graf ilustrují pořadí chromosomů podle přizpůsobenosti (v porovnání se standardní metodou):

chromosom	kvalita	pořadí	přizpůsobenost	
			standardní metoda	seřazovací metoda
1-4	4	1	0.40	0.667
1-3	3	2	0.30	0.222
1-2	2	3	0.20	0.074
1-1	1	4	0.10	0.025
7-5	0	5	0.00	0.012



**Změna:** Seřazovací metoda poskytuje chromosomu 7-5 nenulovou šanci na přežití, takže může být křížen.



- Křížení chromosomů umožňuje GA prohledávat mnohazměrné prostory efektivně. K určení, které chromosomy vybrat, zvolme následující postup:
  - ▶ uvažujeme pouze ty chromosomy, které přežily z předchozí generace
  - ▶ pro každý takový chromosom vybereme náhodně do páru některý z ostatních přeživších
  - ▶ každý pár překřížíme uprostřed; pokud je nově vzniklý potomek odlišný od ostatních kandidátů na přežití, je zařazen mezi tyto kandidáty.
- Uvedená metoda zde průměrně našla optimum ve 14. generaci, tedy o 2 generace dříve než bez použití křížení. Důvod urychlení tkví v tom, že křížení může spojit jedince, který “to umí dobře s moukou” s jedincem, který “se vyzná v cukru”, tj. každý si vede dobře ve své dimenzi. Spojením dvou dobrých genů vznikne kvalitní jedinec.
- Aby křížení dobře fungovalo, je zapotřebí, aby prohledávaný prostor umožňoval hledání globálního maxima pomocí vyhledávání maxim v každé dimenzi nezávisle. Křížení v podstatě redukuje dimensionalitu prohledávaného prostoru.

- Křížení umožňuje GA dostat se přes překážkové příkopy:

cukr[kg]	9	1	2	3	4	5	4	3	2	1
	8	2	0	0	0	0	0	0	0	2
	7	3	0	0	0	0	0	0	0	3
	6	4	0	0	7	8	7	0	0	4
	5	5	0	0	8	9	8	0	0	5
	4	4	0	0	7	8	7	0	0	4
	3	3	0	0	0	0	0	0	0	3
	2	2	0	0	0	0	0	0	0	2
	1	1	2	3	4	5	4	3	2	1
		1	2	3	4	5	6	7	8	9
										mouka [kg]

Předpokládejme nyní, že funkce kvality je dána modifikovanou tabulkou obsahující “příkop” reprezentovaný oblastí s nulami. Pomocí pouhých jednotlivých mutací v generacích nelze příkop překonat, protože potřebné chromosomy produkující koláče v příkopu mají nulovou kvalitu a tedy nulovou přizpůsobenost, tudíž i nulovou šanci na přežití do další generace. Na druhé straně, pomocí dvou dobrých rodičů s **1-5** a **5-1** chromosomy, jediné spárování vytvoří potomka schopného překonat překážkový příkop, aniž by bylo zapotřebí mít přechodné jedince přežívající i s nulovou pravděpodobností. Pro uvedený typ “terénu” je křížení více než pouhou pomocí. Bohužel, stále je efektivita nízká, neboť pokusy ukazují, že navzdory aplikaci operátoru křížení se populace jako celek spíše “plazí” podél jedné či druhé osy, přičemž všechny čtyři chromosomy střídavě vedou ke kombinaci **5-1** nebo **1-5**. “Příkopové” mutace hynou ihned. Mutace chromosomu **1-1** a výše mají sklon zhynout dříve, než se dostanou do výhodnější pozice ke křížení. 1000 pokusů nalézt optimum z **1-1** dalo výsledek průměrně ve 155. generaci.



- **Přežití nejrozmanitějších:** Přizpůsobenost, tak jak byla dosud měřena, ignoruje *rozmanitost*, což si lze předsavit jako míru odlišnosti genů chromosomů. Chromosomy mají tendenci vymizet, pokud jejich skóre je jen nepatrně nižší než u chromosomů blízkých nejlepšímu. To vede nekonec i ve velkých populacích k uniformitě, což je svým způsobem degenerace, protože populace se může vyvíjet jen omezeným směrem a nedosáhne už vývojem svého cíle, tj. globálního optima.
- V přírodní velké škále však nepřizpůsobeně vypadající jedinci a druhy přežívají docela dobře v ekologických prostředích, které leží mimo ostatní — přizpůsobeně vyhlížející — populaci.
- **Princip diversity:** Být odlišný může být stejně dobré jako být přizpůsobený.
- **Prostorově-seřazovací metoda** spojuje přizpůsobenost s kvalitou a diversitou (různorodostí): při výběru chromosomů pro novou generaci lze měřit i odlišnost, kterou kandidát přispěje populaci. Míra odlišnosti se počítá jako součet převrácených hodnot čtverců vzdáleností kandidáta od již vybraných chromosomů:

$$diversita = \sum_i \frac{1}{d_i^2}$$

- **Příklad:** uvažme opět soubor šesti kandidátů {5-1, 1-4, 3-1, 1-2, 1-1, 7-5}. Nejvyšší skóre má chromosom 5-1. Ostatních pět se seřadí podle kvality a diversity takto:

chromosom	skóre	1/d <sup>2</sup>	pořadí dle diversity	pořadí dle kvality
1-4	4	0.040	1	1
3-1	3	0.250	5	2
1-2	2	0.059	3	3
1-1	1	0.062	4	4
7-5	0	0.050	2	5

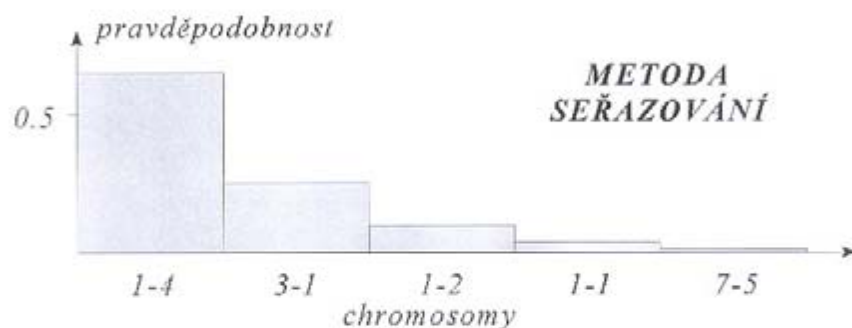
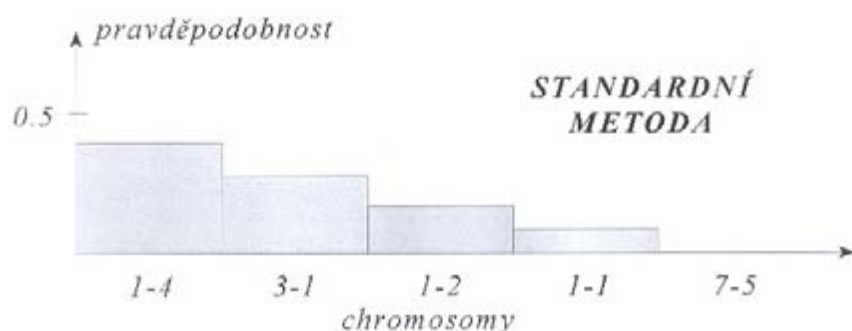
Např.  $d^2(1-4,5-1) = (1-5)^2 + (4-1)^2 = (-4)^2 + (3)^2 = 16+9 = 25$ , a tedy  $1/d^2 = 1/25 = 0.040$ .

Jednoduchý způsob kombinace pořadí dle kvality a dle diversity spočívá v zařazení každého chromosomu podle součtu jeho kvality a diversity za použití následující procedury:

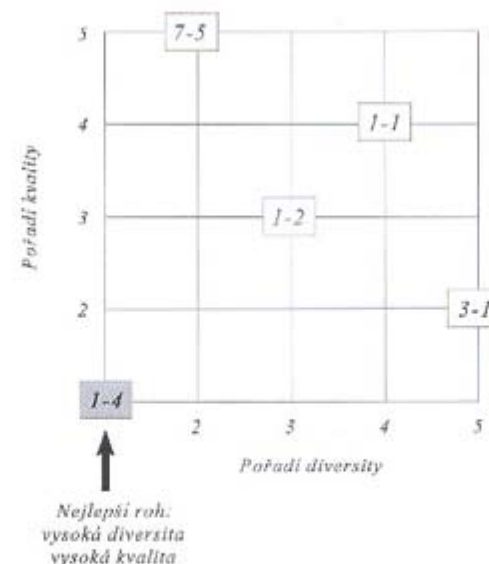
- ▶ seříd'  $n$  jedinců podle kvality
- ▶ seříd'  $n$  jedinců podle diversity
- ▶ seřaď jedince podle součtu kvality a diversity

Následující obrázek tuto proceduru ilustruje:

chromosom	kvalita	pořadí	přizpůsobenost	
			standardní metoda	seřazovací metoda
1-4	4	1	0.40	0.667
1-3	3	2	0.30	0.222
1-2	2	3	0.20	0.074
5-2	1	4	0.10	0.025
7-5	0	5	0.00	0.012



**Změna:** Seřazovací metoda poskytuje chromosomu 7-5 nenulovou šanci na přežití, takže může být křížen.



- Uvedený diagram, kde na jedné ose je kvalita a na druhé diversity, se nazývá **prostor pořadí** a vztahuje se k metodě **prostorového seřazení**. Nejlepší chromosom je v levém dolním rohu, kde je jak kvalita, tak i odlišnost nejlepší.
- Pomocí kombinovaného seřazení chromosomů lze při výběru dále postupovat již uvedenou metodou, tj. nastavením hodnoty přizpůsobenosti nejlepšího kandidáta na  $p$ .
- Výsledek ukazuje následující tabulka a diagram pravděpodobností. Za povšimnutí stojí, že chromosomy 3-1 a 7-5 mají stejnou kombinovanou hodnotu. V takovém případě je nutno rozhodnout, např. dát přednost chromosomu s lepší diversitou.



- Předpokládejme, že chromosom s nejvyšší pravděpodobností — tj. chromosom **1-4** — je vybrán k doprovodu nejlepšího chromosomu **5-1** do následující generace. Je nutno vybrat ještě dva další. *Nyní ovšem by další v pořadí měl být vzdálen od obou již vybraných jedinců.* Proto místo měření převrácené hodnoty čtverce vzdálenosti k pouze jednomu referenčnímu chromosomu se použije součet převrácených hodnot čtverců vzdáleností od **dvou** referenčních chromosomů, tj. **5-1** a **1-4**. V tomto okamžiku je součtové pořadí všech chromosomů totéž, tj. 5, ale využití pořadí dle diversity jako kritéria pro výběr dává následující tabulku:

chromosom	$\sum_i 1/d_i^2$	pořadí dle diversity	pořadí dle kvality	kombinované pořadí	přizpůsobenost
<b>3-1</b>	0.327	4	1	4	0.037
<b>1-2</b>	0.309	3	2	3	0.074
<b>1-1</b>	0.173	2	3	2	0.222
<b>7-5</b>	0.077	1	4	1	0.667

- Nyní předpokládejme, že je vybrán chromosom **7-5**. Pro výběr posledního chromosomu se použije následující přepočtená tabulka. Tentokrát je nejlepším kandidátem **1-1**:

chromosom	$\sum_i 1/d_i^2$	pořadí dle diversity	pořadí dle kvality	kombinované pořadí	přizpůsobenost
<b>3-1</b>	0.358	3	1	3	0.111
<b>1-2</b>	0.331	2	2	2	0.222
<b>1-1</b>	0.190	1	3	1	0.667

- Metoda prostorového seřazování je vhodná pro úlohy mající charakter "pahorku s propadlinami":

Při 1000 pokusných bězích algoritmu hledajícího optimum s použitím prostorového seřazení, s východiskovým bodem **1-1** a  $p=0.667$ , najde kuchař nejlepší složení mouky a cukru průměrně po 15 generacích. To je podstatné zlepšení vůči standardní metodě a metodě prostého seřazení (čísla udávají počet generací):

typ pahorku	standardní metoda	seřazení dle kvality	prostorové seřazení
hladký hrbol	14	12	12
kopec s propadlinami	155	75	15

Při testování metody prostorového seřazení (1000 simulačních běhů) bylo v nejlepším případě zapotřebí 7 generací k nalezení nejlepší koláčové směsi.

- Postup evoluce populace ukazuje následující sekvence obrázků:

1	2	3	4	5	4	3	2	1
2	0	0	0	0	0	0	0	2
3	0	0	0	0	0	0	0	3
4	0	0	7	8	7	0	0	4
5	0	0	8	9	8	0	0	5
4	0	0	7	8	7	0	0	4
3	0	0	0	0	0	0	0	3
2	0	0	0	0	0	0	0	2
1	2	3	4	5	4	3	2	1

1	2	3	4	5	4	3	2	1
2	0	0	0	0	0	0	0	2
3	0	0	0	0	0	0	0	3
4	0	0	7	8	7	0	0	4
5	0	0	8	9	8	0	0	5
4	0	0	7	8	7	0	0	4
3	0	0	0	0	0	0	0	3
2	0	0	0	0	0	0	0	2
1	2	3	4	5	4	3	2	1

- Obrázek ukazuje, jak výpočty postupovaly. Za povšimnutí stojí, jak má metoda prostorového seřazování chromosomů tendenci používat diversitu vzhledem ke standardní metodě (jež by nikdy nevybrala 7-5) a vzhledem k prosté seřazovací metodě (jež by vybrala 7-5 jako poslední spíše než jako třetí).

9	1	2	3	4	5	4	3	2	1
8	2	0	0	0	0	0	0	0	2
7	3	0	0	0	0	0	0	0	3
6	4	0	0	7	8	7	0	0	4
5	5	0	0	8	9	8	0	0	5
4	4	0	0	7	8	7	0	0	4
3	3	0	0	0	0	0	0	0	3
2	2	0	0	0	0	0	0	0	2
1	1	2	3	4	5	4	3	2	1
	1	2	3	4	5	6	7	8	9

1 Vybrán jeden chromosom

9	1	2	3	4	5	4	3	2	1
8	2	0	0	0	0	0	0	0	2
7	3	0	0	0	0	0	0	0	3
6	4	0	0	7	8	7	0	0	4
5	5	0	0	8	9	8	0	0	5
4	4	0	0	7	8	7	0	0	4
3	3	0	0	0	0	0	0	0	3
2	2	0	0	0	0	0	0	0	2
1	1	2	3	4	5	4	3	2	1
	1	2	3	4	5	6	7	8	9

2 Vybrány dva chromosomy

9	1	2	3	4	5	4	3	2	1
8	2	0	0	0	0	0	0	0	2
7	3	0	0	0	0	0	0	0	3
6	4	0	0	7	8	7	0	0	4
5	5	0	0	8	9	8	0	0	5
4	4	0	0	7	8	7	0	0	4
3	3	0	0	0	0	0	0	0	3
2	2	0	0	0	0	0	0	0	2
1	1	2	3	4	5	4	3	2	1
	1	2	3	4	5	6	7	8	9

3 Vybrány tři chromosomy

9	1	2	3	4	5	4	3	2	1
8	2	0	0	0	0	0	0	0	2
7	3	0	0	0	0	0	0	0	3
6	4	0	0	7	8	7	0	0	4
5	5	0	0	8	9	8	0	0	5
4	4	0	0	7	8	7	0	0	4
3	3	0	0	0	0	0	0	0	3
2	2	0	0	0	0	0	0	0	2
1	1	2	3	4	5	4	3	2	1
	1	2	3	4	5	6	7	8	9

4 Vybrány čtyři chromosomy

1	2	3	4	5	4	3	2	1
2	0	0	0	0	0	0	0	2
3	0	0	0	0	0	0	0	3
4	0	0	7	8	7	0	0	4
5	0	0	8	9	8	0	0	5
4	0	0	7	8	7	0	0	4
3	0	0	0	0	0	0	0	3
2	0	0	0	0	0	0	0	2
1	2	3	4	5	4	3	2	1

1	2	3	4	5	4	3	2	1
2	0	0	0	0	0	0	0	2
3	0	0	0	0	0	0	0	3
4	0	0	7	8	7	0	0	4
5	0	0	8	9	8	0	0	5
4	0	0	7	8	7	0	0	4
3	0	0	0	0	0	0	0	3
2	0	0	0	0	0	0	0	2
1	2	3	4	5	4	3	2	1

1	2	3	4	5	4	3	2	1
2	0	0	0	0	0	0	0	2
3	0	0	0	0	0	0	0	3
4	0	0	7	8	7	0	0	4
5	0	0	8	9	8	0	0	5
4	0	0	7	8	7	0	0	4
3	0	0	0	0	0	0	0	3
2	0	0	0	0	0	0	0	2
1	2	3	4	5	4	3	2	1

1	2	3	4	5	4	3	2	1
2	0	0	0	0	0	0	0	2
3	0	0	0	0	0	0	0	3
4	0	0	7	8	7	0	0	4
5	0	0	8	9	8	0	0	5
4	0	0	7	8	7	0	0	4
3	0	0	0	0	0	0	0	3
2	0	0	0	0	0	0	0	2
1	2	3	4	5	4	3	2	1

1	2	3	4	5	4	3	2	1
2	0	0	0	0	0	0	0	2
3	0	0	0	0	0	0	0	3
4	0	0	7	8	7	0	0	4
5	0	0	8	9	8	0	0	5
4	0	0	7	8	7	0	0	4
3	0	0	0	0	0	0	0	3
2	0	0	0	0	0	0	0	2
1	2	3	4	5	4	3	2	1

1	2	3	4	5	4	3	2	1
2	0	0	0	0	0	0	0	2
3	0	0	0	0	0	0	0	3
4	0	0	7	8	7	0	0	4
5	0	0	8	9	8	0	0	5
4	0	0	7	8	7	0	0	4
3	0	0	0	0	0	0	0	3
2	0	0	0	0	0	0	0	2
1	2	3	4	5	4	3	2	1



- S lokálními maximy se lépe vypořádáme, když použijeme diversitu:

- ▶ Většina přístupů považuje lokální maxima za "pasti". Takové metody vyžadují zabudování mechanismů umožňujících se pastem vyhnout (návrat po vlastní stopě — backtracking, či zpočátku velký, postupně se zmenšující krok). Jiné přístupy zahrnují paralelní prohledávání s velkým počtem počátečních startovacích pozic v naději, že jedna z paralelních větví bude chycena do lokálního maxima, které se rovněž ukáže být také maximem globálním.
- ▶ Oproti tomu, použijí-li GA diversitu jako složku přizpůsobivosti, pak někteří jedinci získají vlastnost "vznést se" nad již objevenými lokálními maximy pomocí kvality nebo diversity (či obojího) a "strhnout" za sebou ostatní dosud putující jedince.
- ▶ Dokud je dost jedinců pro dostatečné obydlení lokálních maxim, existuje velmi rozumná šance, že nějaké individuum najde svou cestu ke globálnímu maximu.

- Princip *obsad'-a-zabydli se*:

- ▶ Lokální maxima by měla být obsazena, nikoliv obcházena, na cestě hledání globálního maxima.

- Genetické algoritmy prokázaly svou užitečnost v mnoha komerčních aplikacích. Např. při návrhu lopatek turbín je tvar lopatky navržen s pomocí standardních metod, lopatka je popsána numerickými parametry. Tyto parametry jsou považovány za východiskový stav jakési lopatky "střední kvality". Řešení je dále optimalizováno pomocí GA, což vede ke zvýšení kvality konečného řešení.
- GA vyžadují velké množství výpočtů. Počet chromosomů může dosahovat několik tisíc, každý chromosom může být tvořen řetězcem s délkou řádově stovky. Počet rekombinací může být desítky tisíc.
- GA zahrnují určitou režii, která spočívá ve výpočtu funkce kvality.

## Evoluční optimalizace

Prototypem optimalizačního problému je tzv. problém obchodního cestujícího (POC / TSP).

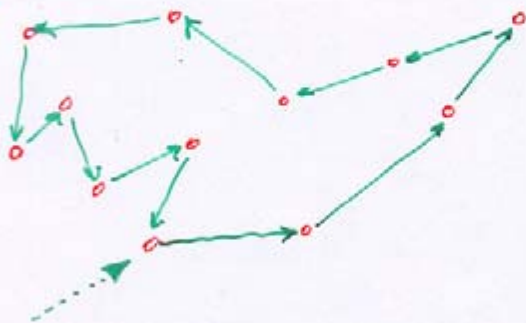
TSP lze jednoduše definovat:

Je dána množina  $n$  měst a prostředek k získání hodnoty vzdálenosti mezi kteroukoliv dvojicí měst. Likolem je najít cestu o minimální délce, kde je každé město navštíveno pouze jednou, a nakonec se vrátit do výchozího města.

Přestože se jedná o jednoduchý problém, nepodařilo se dosud nalézt optimalizační nebo alespoň dostatečně efektivní algoritmus na řešení POC.

Řešení POC je atraktivní z hlediska tzv. kombinatorické optimalizace.

Z praktického hlediska je důležité, že mnoho významných problémů lze formulovat jako instance problému POC.



POC se často řeší jako symetrický problém, tj. nezáleží na směru cesty (ani na výchozím místě).

Celkový počet možných řešení je dán jako permutace všech možných propojení, tj.:

$$\frac{(n-1)!}{2} \text{ různých cest.}$$

Pro malá  $n$  (počet měst) lze problém řešit snadno různými metodami, avšak pro větší hodnoty je řešení obtížné:

$n$	počet cest	Jedná se o NP problém.
100	$4.661 \times 10^{155}$	
200	$1.972 \times 10^{372}$	

## Formální definice problému

Je dána množina  $C = \{c_1, c_2, \dots, c_n\}$  měst a pro každou dvojici  $c_i, c_j$  ( $i \neq j$ ) jejich vzdálenost  $d(c_i, c_j)$ . Likolem je najít permutaci měst  $T = (c_{\pi(1)}, c_{\pi(2)}, \dots, c_{\pi(n)})$

$T = (c_{\pi(1)}, c_{\pi(2)}, \dots, c_{\pi(n)})$  takovou, že

$$f(T) = \sum_{i=1}^n d(c_{\pi(i)}, c_{\pi(i+1)}) + d(c_{\pi(n)}, c_{\pi(1)}) \rightarrow \min$$

$T$  se nazývá cesta a  $f(T)$  je délka cesty, tj. kvantita, jež má být minimalizována.

$$d(c_i, c_j) = d(c_j, c_i)$$

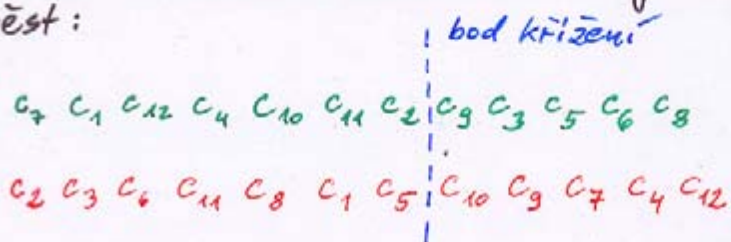


## Genetická reprezentace POC

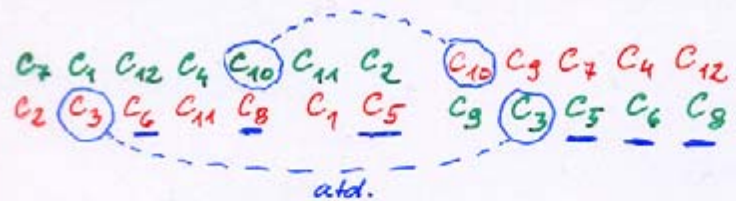
Reprezentace musí vhodným způsobem odrazet strukturu problému.

Je nutno v konkrétních případech mít na mysli specifikou problému, tj. aby chromozomy představovaly pouze platná řešení.

Jednou z možností zakódování POC je sekvence měst:



Po křížení vznikne nekorektní řešení:



Může být narušena podmínka zakazující návrat do již navštíveného místa (vznik smyčky v grafu).

Vzniklé kolizní situace je nutno řešit vhodným návrhem genetických operátorů (zde operátor křížení).

Nejlépe dosud existující algoritmy používají modifikované kódování – místo prosté sekvence měst (nejpřirozenější reprezentace) se používá reprezentace založená na využití hran grafu (spojnic dvojic měst):

$$T = \left( (c_{\pi(1)}, c_{\pi(2)}), (c_{\pi(2)}, c_{\pi(3)}), \dots, (c_{\pi(n-1)}, c_{\pi(n)}), (c_{\pi(n)}, c_{\pi(1)}) \right)$$

Tímto způsobem vznikne řetězcová reprezentace.

## Operátor mutace

Nejjednodušší modifikací je využití 2 hran:

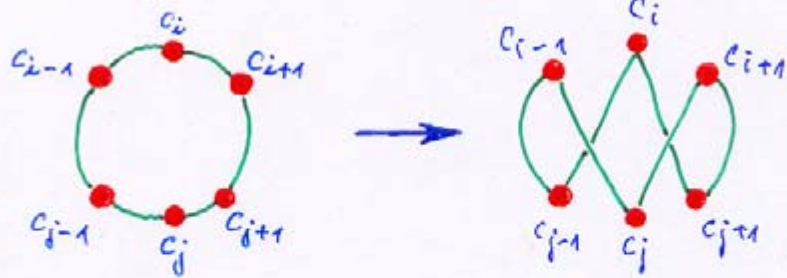
1. Vyřadí 2 náhodně vybrané hrany, např.  $(c_i, c_{i+1})$  a  $(c_j, c_{j+1})$ .
2. Nahradí tyto hrany pomocí  $(c_i, c_j)$  a  $(c_{i+1}, c_{j+1})$ .  
Tím se vytvoří 2 cesty  $T$  cesta  $T$ .

Je též možno využít více hran, např. 4:

1. Náhodně zvol 2 města, např.  $c_i$  a  $c_j$ .
2. Prohodí tato dvě města. Tím dojde k odstranění hran  $(c_{i-1}, c_i)$ ,  $(c_i, c_{i+1})$ ,  $(c_{j-1}, c_j)$ ,  $(c_j, c_{j+1})$ . Tyto hrany jsou nahrazeny hranami  $(c_{j-1}, c_i)$ ,  $(c_i, c_{j+1})$ ,  $(c_{i-1}, c_j)$ ,  $(c_j, c_{i+1})$

Viz obrázek:





### Operátor křížení

Je možné navrhnout různé operátory vzhledem k podmínkám úlohy POC.

Důležité je co nejefektivněji přenášet hrany na potomky tak, aby nedocházelo ke ztrátě informace a v genetickém zásobníku se udržovalo co nejvíce hran.

Dobrým řešením je např. tzv. MPX (Maximal Preservative Crossover, neboli křížení s maximálním uchováním). MPX je založen na dvou pojmech: dárce a příjemce. (tj. rodiče potomka)

**Dárce:** poskytuje příjemci spojitý řetěz hran. Hrana se nazývá přijatelná, pokud v procesu vytváření potomka nevznikne jejím přidáním cyklus (smyčka).

Následující zápis v pseudokódu popisuje algoritmus MPX:

rodiče

PROC křížení (příjemce, dárce, potomek)

Vyber pozici  $0 \leq i \leq n$  a délku  $l_{\min} \leq k \leq l_{\max}$  náhodným výběrem.

Extrahuj řetězec měst od pozice  $i$  do pozice  $j = (i+k) \bmod n$  z dárce. Tím se získá křížící řetězec.

Ekopiruj tento řetězec do potomka.

Přidej postupně další hrany dokud potomek nepředstavuje platnou cestu:

```

IF hrana příjemce začínající na konci poslední hrany potomka je možná
THEN přidej tuto hranu
ELSE IF hrana dárce začínající v posledním městě potomka je možná
THEN přidej tuto hranu
ELSE přidej město z příjemce, které je další v pořadí v řetězci (vznikne tak nová hrana, kterou považujeme za implicitní mutaci)

```

Pseudokód algoritmu MPX



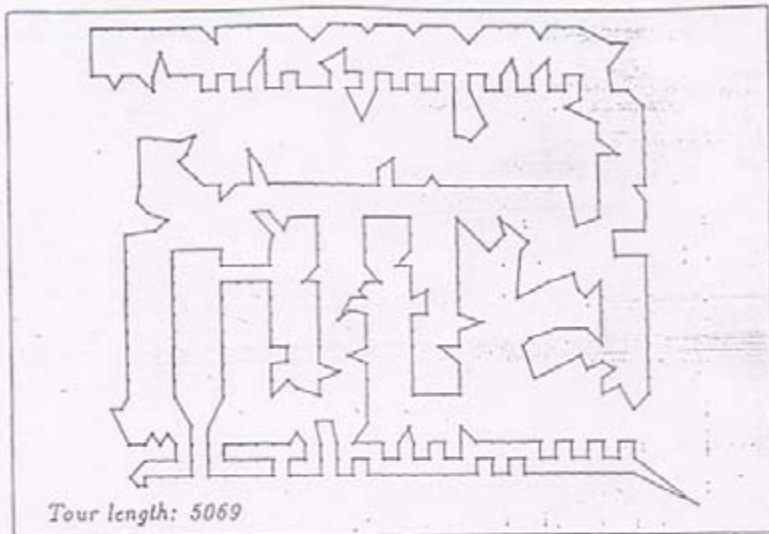


Figure 16.7 The optimal solution of the 442 drilling problem.

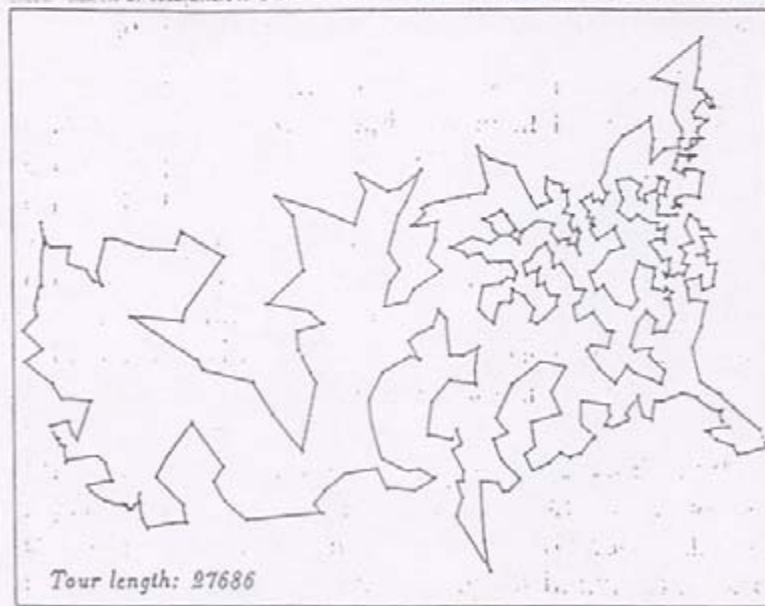


Figure 16.8 The optimal solution of the 532 AT&T problem.

## Genetické programování

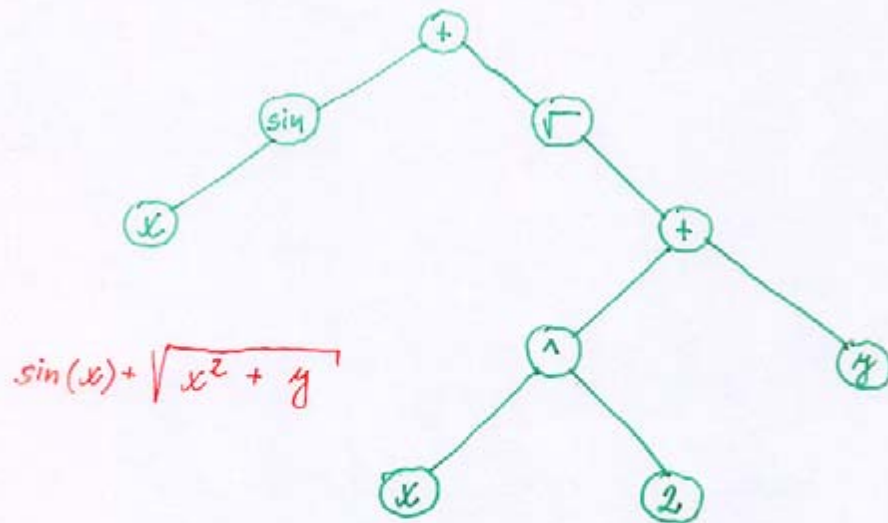
Genetické programování (GP) je forma evolučních výpočtů, v nichž jednotlivci vyvíjející se populace jsou počítačové programy.

### Representace programů:

Programy, s nimiž GP pracuje, jsou typicky reprezentovány jako stromy, které odpovídají stromům, na něž lze programy rozložit.

Každé volání funkce je ve stromu reprezentováno uzlem, přičemž argumenty funkce jsou obsaženy v uzlech "potomků".

Příklad stromové reprezentace programu v GP:



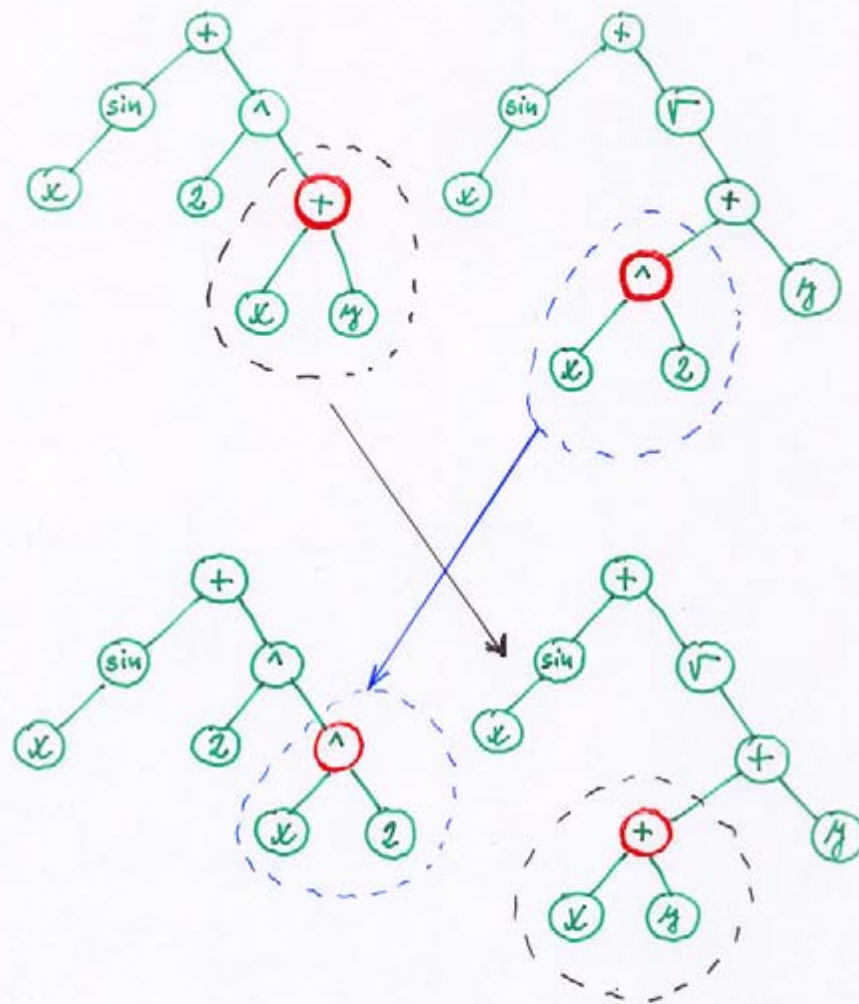
Pro aplikaci GP na nějakou oblast je zapotřebí, aby uživatel definoval primitivní funkce (např.  $\sin$ ,  $\cos$ ,  $\sqrt{\quad}$ ,  $+$ ,  $-$ , exponenty, ...) a terminály (např.  $x$ ,  $y$ , konstanty jako  $2$ , ...).

Algoritmus GP pak využívá evolučního pohledávání v velmi rozsáhlém prostoru možných programů, které mohou být popsány zmíněnými primitivami.

Podobně jako v genetických algoritmech (GA), GP pracují s populací jedinců (tj. programovými stromy). Každý iterací krok vytvoří novou generaci individuí použitím výběru, křížení a mutace.

Přizpůsobenost (fitness) daného individuálního programu v celé populaci je typicky určena provedením programu s použitím množiny trénovacích dat. Operace křížení se provádí pomocí přemístění náhodně vybraného podstromu jednoho rodiče na náhodně vybrané místo podstromu druhého rodiče (výměna podstromů).

Příklad aplikace křížení dvou individuí a vznik nových potomků:





## Ilustrativní příklad (John Koza, 1992):

Jedná se o naučení se algoritmu pro sestavení kostek tak, aby se objevilo slovo UNIVERSAL:



Úkolem je tedy sestavit "věž" z kostek, přičemž pořadí je dáno písmeny. John Koza použil 166 obdobných počátečních "tréninkových" pozic pro ohodnocení jednotlivých vyřizovaných programů (stanovení hodnoty funkce přizpůsobivosti). Úkolem pro vyvinutý program je sestavit správnou "věž" nezávisle na počáteční konfiguraci kostek.

Podmínky řešení: pro manipulaci s kostkami je k dispozici pouze 1 přesun v 1 časovém kroku. Např. vrchní kostka na věži je přesunuta na stůl; nebo zvolená kostka na stole je přesunuta na vrchol věže).

Tak jako ve většině GP aplikací, i zde je významný výběr způsobu reprezentace. Primitivní funkce použité pro sestavení programu pro daný úkol obsahují následující terminální argumenty:

Terminální argumenty:

- CS** (current stack): odkazuje na jméno vrchního bloku (kostky) v zásobníku (věži), případně udává, že CS neexistuje.
- TB** (top correct blok): odkazuje na jméno nejvnitřnějšího bloku v zásobníku, který má tu vlastnost, že on a bloky pod ním jsou ve správném pořadí.
- NN** (next necessary): odkazuje na jméno dalšího bloku potřebného nad TB k tomu, aby vzniklo slovo UNIVERSAL; resp. udává, že žádné další bloky nejsou zapotřebí.

Je zřejmé (intuitivně), že uvedený výběr terminálních argumentů poskytuje přirozenou reprezentaci pro popis programů manipulujících s kostkami v dané úloze. (Lze si představit obtíže, kdy by byly jako terminální argumenty použity např. souřadnice  $x$  a  $y$  každého bloku.)

k uvedeným terminálním argumentům přidáme ještě následující primitivní funkce:



## Primitivní funkce:

- (MS  $x$ ) (move to stack): je-li blok  $x$  na stole, tento operátor přesune  $x$  na vrchol zásobníku a vrátí hodnotu T; jinak neudělá nic a vrátí hodnotu F.
- (MT  $x$ ) (move to table): je-li blok  $x$  někde v zásobníku, přesune blok na vrcholu zásobníku na stůl a vrátí T; jinak nic a vrátí F.
- (EQ  $x$   $y$ ) (equal): vrací T, když  $x = y$ , jinak F.
- (NOT  $x$ ): vrací T, když  $x = F$  a F, když  $x = T$ .
- (DU  $x$   $y$ ) (do until): provádí opakovaně výraz  $x$  dokud výraz  $y$  vrací hodnotu T.

Pozn.: T = true, F = false.

Pokusy J. Kozy využívaly k ohodnocení programů počet správně sestavených „věží“ ze souboru 166 trénovacích příkladů, reprezentujících široké možnosti počátečních konfigurací a zahrnujících různé stupně obtížnosti.

Počáteční populace byla zvolena tak, že obsahovala 300 jedinců (programů). Po 10 generacích byl nalezen program, řešící všech 166 problémů:

(EQ (DU (MT CS) (NOT CS)) (DU (MS NN) (NOT NN)))

Tento program obsahuje sekvenci dvou DU (do until). První DU opakovaně přesouvá vrchol zásobníku na stůl, dokud není zásobník prázdný. Druhý DU opakovaně přesouvá NN (následně nutný) blok ze stolu do zásobníku.

Role EQ tkví v poskytování syntakticky správného způsobu uspořádání obou cyklů DU.

## Poznámky k GP

GP byly již úspěšně použity k velmi složitým úkolům:

- návrh obvodů elektronických filtrů
- klasifikace segmentů proteinových molekul

Např. při návrhu filtrů byl vyvinut pomocí GP program, který transformoval počáteční jednoduchý „náhradový“ elektronický obvod na finální obvod. Mezi primitivní funkce náleželo např. vkládání či rušení prvků obvodů a spojování prvků (součástek). Přizpůsobenost programů byla měřena simulací výstupů vyvinutých obvodů (určovalo se, jak daleko či blízko jsou navržené obvody ke specifikacím daného filtru). Přizpůsobenost se vyčíslovala jako velikost součtu chyb mezi požadovaným a skutečným výstupem pomocí 101 různých vstupních frekvencí. Pracovalo se s populací 640 000 jedinců (!), přičemž výběr tvořil 10% následné populace, křížení poskytl 89% a mutace 1%. Tento systém běžel na 64-uzlovém paralelním procesoru.



V první, náhodně vygenerované populaci, byly tak špatné obvody, že jejich chování nebyl simulátor schopen na 98% vyhodnotit. V dalších generacích podíl neryhodnotitelných obvodů klesal na 84.9%, 75.0%, až po průměrných 9.6% v následujících generacích. Počáteční kumulovaná chyba byla 159, po 20 generacích klesla jen na 39 a po 137 generacích na pouhých 0.8.

Nejlepší obvod, získaný po 137 generacích, měl výkonnost velice blízkou požadovanému chování.