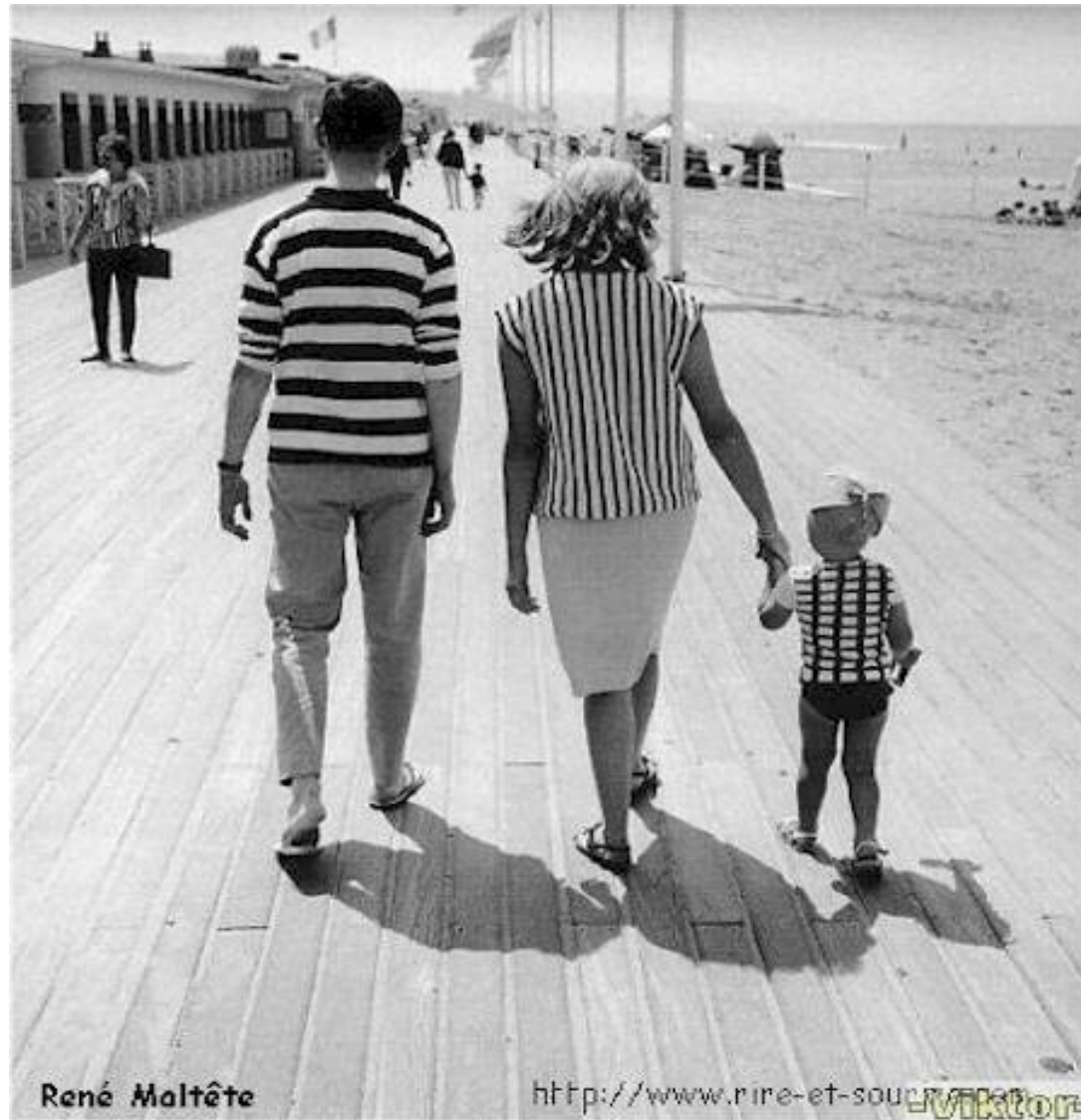


EVA II.



René Maltête

<http://www.nine-et-sour>

Teorie (ještě teoretičtější)

'Přesné' modely GA

- 90.léta: Vose, Lepins, Nix, Whitley, ...
- Snaha zachytit:
 - jak přesně vypadají populace
 - zobrazení přechodu k další populaci
 - vlastnosti tohoto zobrazení
 - asymptotické chování jednoduchého GA
- Nekonečné populace
- Konečné populace

Jednodušší jednoduchý GA

- Náhodná počáteční populace 1 bin. řetězců x
- Fitness $f(x)$
- Opakuj dokud nenaplníš novou:
 - Vybrat selekcí 2 jedince, zkřížit s p_c , 1 zahodit
 - Mutovat každý bit s p_M
 - Vložit do nové
- Opakuj dokud nenajdeš dost dobré x

Formalizujeme JJGA

- Každý řetězec je reprezentován číslem $0..2^l$
 - 00000111 je jako 7
- Populace t je reprezentována:
 - dvěma vektory: $p(t)$ a $s(t)$ délky 2^l
 - $p_i(t)$ část populace t , kterou zabírá řetězec i
 - $s_i(t)$ pravděpodobnost selekce řetězce i
- $p(t)$ definuje složení populace
- $s(t)$ shrnuje pravděpodobnosti výběru

Operátor Velké Gé

- Máme fitness f , definujeme matici $F(i,j)$:
 - $F(i,i) = f(i)$; $F(i,j) = 0$ pro $i \neq j$
- Pak $s(t) = F p(t) / (\sum F(j,j) p_j(t))$
 - (to je vlastně definice proporcionální selekce)
- Chceme definovat G realizující JJGA:
 - $s(t+1) = G s(t)$
- Pak iterujeme $G \dots G s(0)$ a všechno víme
- $G = F \circ M$ (F fitness, M křížení a mutace)

Necht' $G = F$

- $E(x)$ očekávána (střední hodnota)
- $E(p(t+1)) = s(t)$
- $s(t+1) \sim F p(t+1)$ (liší se jen o multiplikační c)
- Takže: $E(s(t+1)) \sim F s(t)$
 - V případě konečné populace nám výběrové chyby mohou způsobit odchylku od $E(.)$
 - Čím větší populace, tím menší odchylka
 - U nekonečné populace to je přesné

M

- M rekombinační op. (zahrnuje křížení i mutaci)
- Odvození je těžší:
 - $r(i,j,k)$... pravděpodobnost, že k vznikne z i a j
 - Když známe r , můžeme spočítat:
- $E(p_k(t+1)) = \sum_i \sum_j s_i(t) s_j(t) r(i,j,k)$

$\mathbf{r}(i,j,0)$

$$r_{i,j}(0) = \frac{(1 - p_m)^l}{2} \left[\eta^{|i|} \left(1 - p_c + \frac{p_c}{l-1} \sum_{c=1}^{l-1} \eta^{-\Delta_{i,j,c}} \right) + \eta^{|j|} \left(1 - p_c + \frac{p_c}{l-1} \sum_{c=1}^{l-1} \eta^{\Delta_{i,j,c}} \right) \right].$$

A zbytek analogicky

Výsledky

- JJGA pracující prostřednictvím G je dynamický systém, $s(t)$ jsou body (trajektorie).
- Jaké jsou pevné body? NEVÍME
 - Pevné body F jsou populace se stejnou fitness
 - Stabilní pevný bod F : maximální stejná fitness
 - (Jediný) pevný bod M : stejné pravděpodobnosti s (resp. stejné zastoupení jedinců p)
 - Kombinace míchání M a ostření F (punctuated equilibria)

Konečná populace

- Markovovské řetězce:
 - Stochastický proces v diskrétním čase
 - Systém má stavy, pravděpodobnost přechodu z jednoho stavu do druhého závisí jen na tom 1 stavu
 - Tabulka stavů
 - Tabulka přecodových pravděpodobností ze stavu i do stavu j
- Budeme modelovat GA nad konečnou populací jako markovovský řetězec

Stavy

- Stav Markovovského řetězce je konkrétní populace.
- Populací o n jedincích délky l je N :
 - $N = (n+2^l-1) \text{ nad } (2^l-1)$
- Matice Z
 - Sloupce jsou populace
 - $Z(y,i) = \text{počet jedinců } y \text{ v populaci } i$
- (takže sloupce Z jsou stavy M.ř.)

Přechodová matice

- Q matice pravd.přechodu mezi stavy:
- $N \times N$
- (např. pro $n=l=8$ má 10^{29} prvků!)
- Odvození komplikované:

$$Q_{i,j} = n! \prod_{y=0}^{2^l-1} \frac{\left[\mathcal{M} \left(F \vec{\phi}_i / |F \vec{\phi}_i| \right)_y \right]^{Z_{y,j}}}{Z_{y,j}!}.$$

K čemu to je?

- Přesná analýza GA pro danou f
- Ač prakticky těžko proveditelná
- Asymptotické výsledky – vzhled do konvergenčních vlastností a chování GA
- Korespondence ideálního JJGA s nek.pop. a kon. Pop. (když n jde k nekonečnu)

Evoluční programování

Evoluční programování

- L.Fogel, 60.léta (starší než Hollandovy GA)
- Snaha vyvinout „umělou inteligenci“:
 - předpověď stavu prostředí, ve kterém se agent nachází
 - odvození patřičné akce s ohledem na stav prostředí
- Agent: konečný automat
- Prostředí: posloupnosti symbolů konečné abecedy

Konečný automat

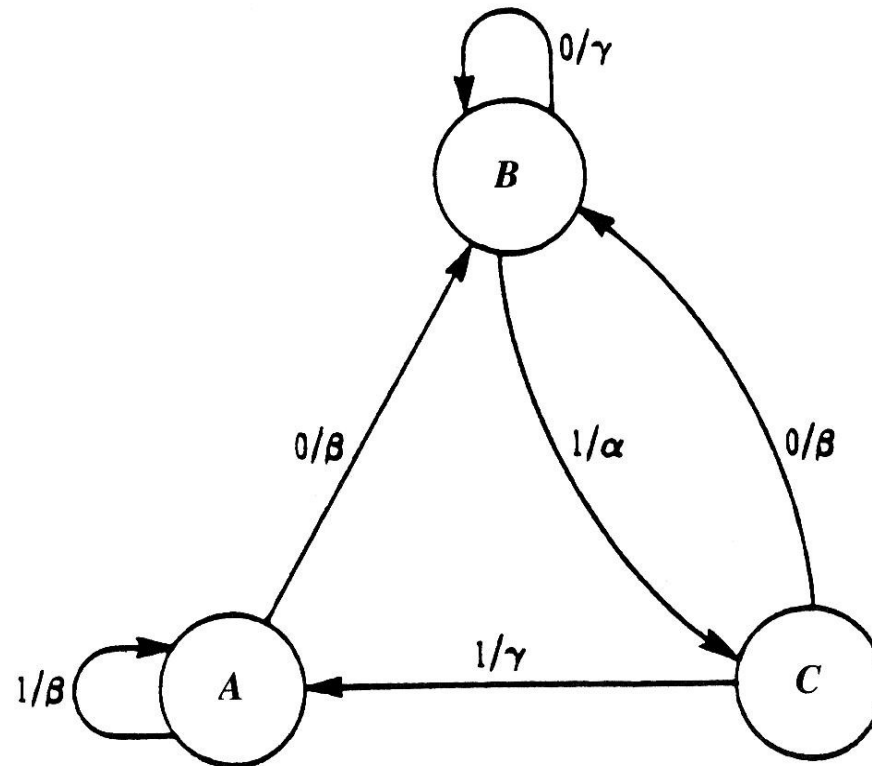


Table 3-1 The Response of the Finite-State Machine Shown in Figure 3-2 to a String of Symbols. In This Example, the Machine Starts in State C.

Present State	<i>C</i>	<i>B</i>	<i>C</i>	<i>A</i>	<i>A</i>	<i>B</i>
Input Symbol	0	1	1	1	0	1
Next State	<i>B</i>	<i>C</i>	<i>A</i>	<i>A</i>	<i>B</i>	<i>C</i>
Output Symbol	β	α	γ	β	β	α

EP

- Populace konečných automatů
- Předkládají se jim vstupy,
- Výstup automatu je porovnán s následujícím vstupem v posloupnosti
- Fitness: úspěšnost predikce (různě měřená)
 - Vše nebo nic
 - Absolutní chyba
 - Mean square error

EP pokr.

- Vznik nových automatů: mutace
 - Změna výstupního symbolu
 - Změna následného stavu
 - Přidání stavu
 - Odebrání stavu
 - Změna počátečního stavu
- Polovina populace nahrazena novými automaty
- Často do fitness zahrnuta i velikost automatu

Příklad: predikce prvočísel

- Učí se řetězce 0 a 1, 0 znamená číslo na dané pozici není prvočíslo.
- Iterativně: učí se pevnou délku, pak zvětšit vstup o 1 symbol
- Fitness: bod za každý správný symbol
- Záporný člen fitness: $- 0.01 * N$ (=počet stavů)
- Automaty se 'zjednodušovaly' až k jednomu stavu a 0.

Příklad pokr.

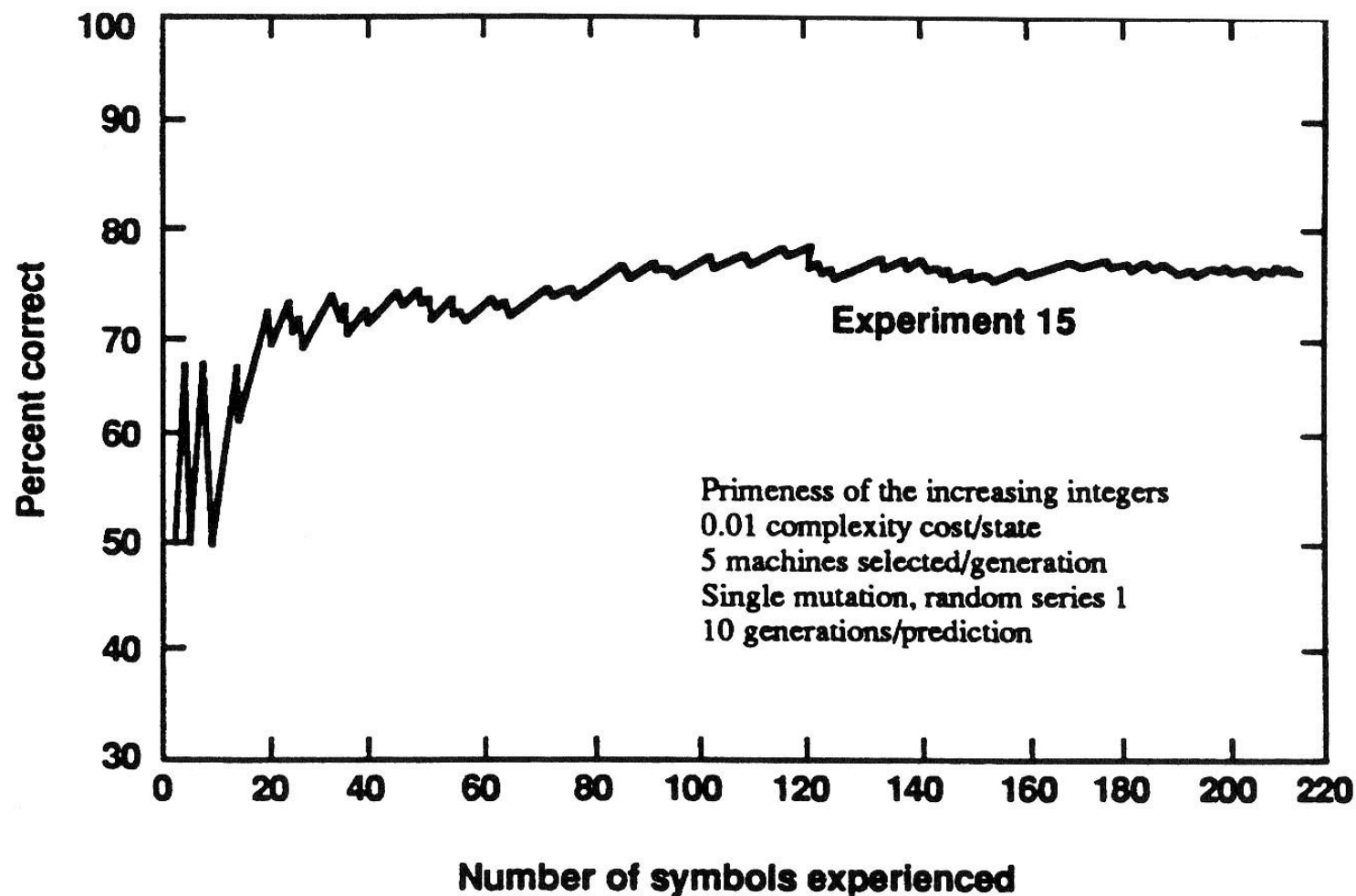


Figure 3-3. The cumulative percentage of correct predictions in the first 200 prime numbers (from Fogel et al., 1966, p. 37).

EP vs. GA

- Křížení:
 - Výměna bloků?
 - Divná mutace?
- Jones 1995: headless chicken experiment:
 - Tradiční křížení
 - Náhodné křížení (s náhodným řetězcem)
- There is no crossover without the idea of crossover, do not call headless chicken a chicken, although it has many chicken features.

Headless chicken

- V experimentech s jasnými bloky překoná křížení náhodné křížení
- Když je ale náhodné křížení lepší, jde vlastně o makromutaci
- V takovém případě nemáme jasné bloky
 - Bud' špatné zakódování
 - Nebo těžký problém pro GA s křížením
- Poučení: má naše kuře hlavu?

GA vs EP experiment

Table 4.1 The Number of Parameters, The Binary Coding Length, and The Functions Studied in Schraudolph and Belew (1992). These Follow Previous Efforts By De Jong (1975). The Operation $[x_i]$ in $f3$ Returns The Greatest Integer Less Than or Equal To x_i . The $N(0, 1)$ in $f4$ Represents A Standard Gaussian Random Variable.

Function	Dimension	Bit Length
$f1$	3	10
$f2$	2	12
$f3$	5	10
$f4$	30	8
$f5$	2	17

Function	Parameter Range
----------	-----------------

$$f1: F(x) = \sum_{i=1}^3 x_i^2 \quad [-5.12, 5.12]$$

$$f2: F(x) = 100(x_1^2 - x_2)^2 + (1 - x_1)^2 \quad [-2.048, 2.048]$$

$$f3: F(x) = \sum_{i=1}^5 [x_i], \quad [-5.12, 5.12]$$

$$f4: F(x) = \sum_{i=1}^{30} ix_i^4 + N(0, 1) \quad [-1.28, 1.28]$$

$$f5: F(x)^{-1} = 1/K + \sum_{j=1}^{25} f_j(x)^{-1} \quad [-65.536, 65.536]$$

$$f_j(x) = c_j + \sum_{i=1}^2 (x_i - a_{ij})^6.$$

where $K = 500$, $c_j = j$, and

$$[a_{ij}] = \begin{bmatrix} -32 & -16 & 0 & 16 & 32 & -32 & -16 & \dots & 0 & 16 & 32 \\ -32 & -32 & -32 & -32 & -32 & -16 & -16 & \dots & 32 & 32 & 32 \end{bmatrix}$$

$$f6: F(x, y) = x^2 + 2y^2 - 0.3\cos(3\pi x) - 0.4\cos(4\pi y) + 0.7,$$

$$f7: F(x, y) = x^2 + 2y^2 - 0.3(\cos(3\pi x)\cos(4\pi y)) + 0.3,$$

$$f8: F(x, y) = x^2 + 2y^2 - 0.3(\cos(3\pi x) + \cos(4\pi y)) + 0.3.$$

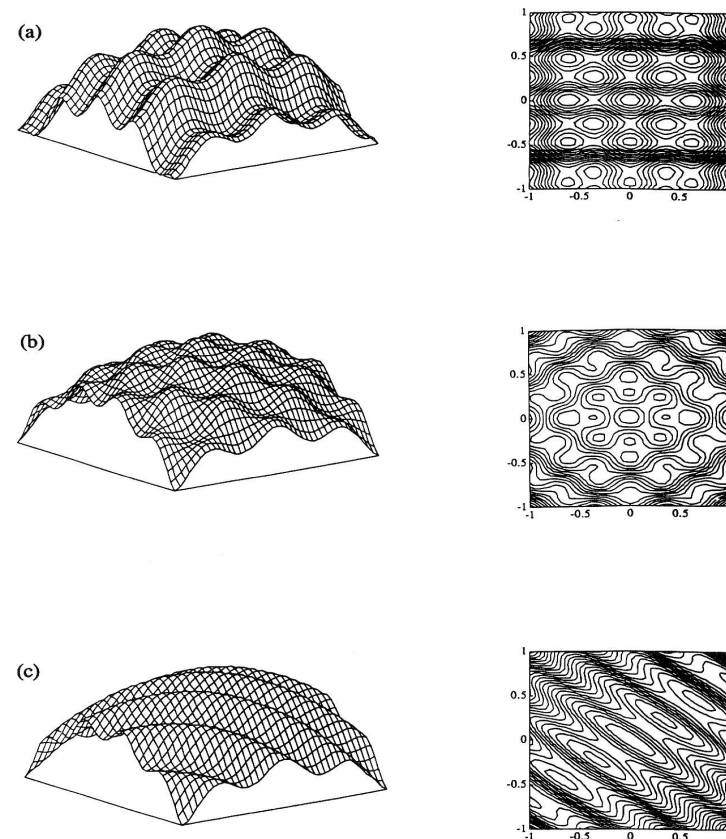


Figure 4-17 Inverted and contour plots of the three Bohachevsky functions studied in Fogel and Stayton (1994). (a) $F(x, y) = x^2 + 2y^2 - 0.3\cos(3\pi x) - 0.4\cos(4\pi y) + 0.7$. (b) $F(x, y) = x^2 + 2y^2 - 0.3(\cos(3\pi x)\cos(4\pi y)) + 0.3$. (c) $F(x, y) = x^2 + 2y^2 - 0.3(\cos(3\pi x) + \cos(4\pi y)) + 0.3$.

GA vs EP exp. pokr.

Table 4-3 Results for the Best Score in the Population and the Mean of All Parents' Scores After 10,080 Function Evaluations, Averaged Over 10 Trials With the Genetic Algorithm Techniques (Both With And Without Dynamic Parameter Encoding) and 500 Trials With "Evolutionary Programming" (after Fogel and Stayton, 1994). Evolutionary Programming Outperforms Both Genetic Methods On Functions $f1, f2$, and $f6-f8$, and Yields Comparable Performance on $f3-f5$. The Values in Parentheses Indicate the Standard Deviations

		Average Best	Average Mean
$f1$:	EP	3.149×10^{-66} (2.344×10^{-129})	1.087×10^{-65} (1.794×10^{-128})
	DPE	1.056×10^{-11} (1.072×10^{-21})	3.618×10^{-10} (1.060×10^{-18})
	GA	2.836×10^{-4} (4.587×10^{-8})	6.135×10^{-1} (1.627×10^{-1})
$f2$:	EP	1.215×10^{-14} (3.357×10^{-26})	8.880×10^{-14} (2.399×10^{-24})
	DPE	2.035×10^{-2} (3.315×10^{-3})	8.785×10^{-2} (8.090×10^{-3})
	GA	2.914×10^{-2} (6.280×10^{-4})	1.722×10^0 (2.576×10^0)
$f3$:	EP	0.0 (0.0)	0.0 (0.0)
	DPE	0.0 (0.0)	0.0 (0.0)
	GA	0.0 (0.0)	1.307×10^0 (1.172×10^{-1})
$f4$:	EP	-2.575×10^0 (7.880×10^{-1})	-4.274×10^{-1} (3.206×10^{-2})
	DPE	-2.980×10^0 (1.009×10^{-1})	4.704×10^{-1} (1.283×10^{-1})
	GA	-4.599×10^{-1} (4.265×10^{-1})	1.312×10^1 (2.941×10^0)
$f5$:	EP	4.168×10^0 (9.928×10^0)	4.194×10^0 (1.022×10^1)
	DPE	3.502×10^{09} (1.265×10^1)	1.642×10^1 (7.101×10^2)
	GA	9.980×10^{-1} (3.553×10^{-15})	1.021×10^1 (7.165×10^1)
$f6$:	EP	5.193×10^{-96} ($1.348 \times 10^{10-188}$)	9.392×10^{-94} (4.410×10^{-184})
	DPE	1.479×10^{-9} (1.460×10^{-18})	8.340×10^{-7} ($4.649 \times 10^{10-14}$)
	GA	2.629×10^{-3} (1.103×10^{-5})	4.022×10^1 (6.467×10^3)
$f7$:	EP	8.332×10^{-101} (3.449×10^{-198})	2.495×10^{-99} (3.095×10^{-195})
	DPE	2.084×10^{-9} (6.831×10^{-18})	6.520×10^{-7} ($7.868 \times 10^{10-14}$)
	GA	4.781×10^{-3} ($2.146 \times 10^{10-5}$)	3.541×10^1 (2.922×10^3)
$f8$:	EP	1.366×10^{-105} (4.479×10^{208})	3.031×10^{-103} ($2.122 \times 10^{10-203}$)
	DPE	1.215×10^{-5} (5.176×10^{-10})	3.764×10^{-1} ($9.145 \times 10^{10-1}$)
	GA	2.444×10^{-3} (4.511×10^{-5})	$2.788 \times 10^{10-1}$ (1.368×10^{-3})

Evoluční strategie

Evoluční strategie

- Rechenberg, Schwefel, 60.léta
- optimalizace multidimenzionálních funkcí
- 'evoluce evoluce'
- evolvovaný jedinec:
 - *Genetické* parametry ovlivňující chování
 - *Strategické* parametry ovlivňující evoluci
- nový jedinec akceptován jen je-li lepší
- na vzniku se může podílet více jedinců

ES notace

- Důležité parametry:
 - M počet jedinců v populaci
 - L počet vznikajících potomků
 - R počet 'rodičů'
- Zvláštní notace souvisí se selekcí:
 - (M+L) ES – M jedinců do nové populace je vybráno z M+L starých i nových jedinců
 - (M,L) ES – M nových jedinců je vybráno jen z L nových potomků
- Jedinec $C(i)=[G_n(i), S_n(i)]$

ES cyklus

1. $n=0$; Náhodně inicializuj populaci P_n M jedinců
2. Ohodnot' jedince P_n pomocí fitness
3. Dokud není řešení dostatečně dobré:
 - a) Opakuj L krát:
 - i. vyber R rodičů,
 - ii. zkříž, mutuj, ohodnot' nového
 - b) Vyber M nových (podle typu ES)
 - c) $++n$

ES jedinec a mutace

- $C(i)=[G_n(i), S_n(i)]$
- S_n jsou:
 - Standardní odchylky floating point mutací
 - Nekorelované mutace
 - Případně vylepšené o „rotace“:
 - Nemutuje se jen podle os dimenzí
 - Násobí se maticí rotace pro určení optimálních směrů
 - (stačí vektor: n-rozměrný)
 - Takže metaparametrů je $2n$
 - Korelované mutace

ES mutace

- Genetické parametry:
 - Přičtení náhodného čísla z normálního rozdělení (s příslušnou odchylkou (a rotací))
- Odchytky:
 - Zvětšovat nebo zmenšovat podle úspěšnosti mutace
 - 1/5 pravidlo heuristika
- Rotace:
 - Přičtení náhodného čísla z $N(0,1)$

ES Křížení

- Uniformní
- Gang bang více rodičů
 - Lokální ($R=2$)
 - Globální ($R=M$)
- Dvě verze
 - Diskrétní
 - Aritmetické (průměr)

PSO

Particle swarm optimization

- Populační prohledávací algoritmus
- Eberhart, Kennedy, 1995
- Inspirace hejny hmyzu/ryb
- Jedinec je typicky vektor reálných čísel
- Říká se mu *částice*
- Nejsou zde operace křížení, mutace
- Jedinci se pohybují v hejnu prostorem parametrů

PSO - algoritmus

For each particle: Initialize particle END

Do

For each particle

Calculate fitness value

If the fitness value is better than the best fitness value (pBest) in history
set current value as the new pBest

End

Choose the particle with the best fitness value of all the particles as the gBest

For each particle

Calculate particle velocity according equation (a)

Update particle position according equation (b)

End

While maximum iterations or minimum error criteria is not attained

PSO – rovnice pohybu

- $V[] := v[] +$
 $+ c1 * \text{rand}() * (\text{pbest}[] - \text{present}[]) + c2 *$
 $+ \text{rand}() * (\text{gbest}[] - \text{present}[]) \quad (a)$
- $\text{present}[] = \text{persent}[] + v[] \quad (b)$

$v[]$ je rychlost částice, $\text{present}[]$ je pozice částice.

$\text{pbest}[]$, $\text{gbest}[]$ viz alg.

$\text{rand}()$ náhodné číslo z $(0,1)$. $c1$,

$c2$ konstanty (učící faktory) často $c1 = c2 = 2$.

PSO - diskuse

- **Společné s GA:**
 - Začínají z náhodné konfigurace, prohledávají prostor, mají fitness jako ohodnocení, používají stochastické metody
- **Odlišné:**
 - Nejsou zde genetické operace
 - Částice mají paměť
 - Výměna informací je jen od nejlepších částic ostatním

Memetické algoritmy

Memetické algoritmy aneb Kulturní evoluce

- 70.-80. léta Dawkins „univerzální Darwinismus“:
- Evoluční principy se nemusí omezovat jen na biologii
- *Mém* – jednotka přenosu kulturní informace, nebo imitace, negenetickými prostředky
- Různé interpretace od filozofie, přes kulturní antropologii, po „urban legends“
- Využití v informatice - „hybridní EA“

MA - algoritmus

Initialize: Generate an initial population;

while Stopping conditions are not satisfied do

 Evaluate all individuals in the population.

 Evolve a new population using stochastic search operators.

 Select the subset of individuals, Ω_{ii} , that should undergo the individual improvement procedure.

 for each individual in Ω_{ii} do

 Perform individual learning using meme(s) with frequency or probability of f_{ii} , for a period of t_{ii} .

 Proceed with Lamarckian or Baldwinian learning.

 end for

 end while

MA - poznámky

- Lamarckovo učení – použiji genotyp změněný lokálním učení (což je často vhodné pro řešení optimalizačních úloh)
- Baldwinovo učení – nepoužiji nový genotyp, ale jen informaci o úspěšnosti nového fenotypu (což je politicky korektnější vzhledem k paradigmátům v biologii)
- Jak často, jak dlouho, a jak lokálně učit mémy, to je otázka konkrétního použití (např. back propagation jako chytrá mutace při evoluci NS)

Neuroevoluce

Učení NS pomocí EVA

- První pokusy od 80.let
- Učení parametrů (vah)
- Učení struktury (spoje, architektura)
- Učení vah i architektury najednou
- Použití v úlohách reinforcement learning – není možné učit metodami s učitelem (robotika)
- Hybridní metody – kombinace EA s lokálním prohledáváním apod

Učení vah

- Přímočaré,
 - floating point GA, standardní operátory
- Většinou je pomalejší než specializované gradientní algoritmy (často řádově)
- + Lze ho paralelizovat
- + Lze ho použít i pro úlohy, kde mám fitness, ale ne chybu v každém kroku, takže gradientní algoritmy nemohu použít

Učení struktury

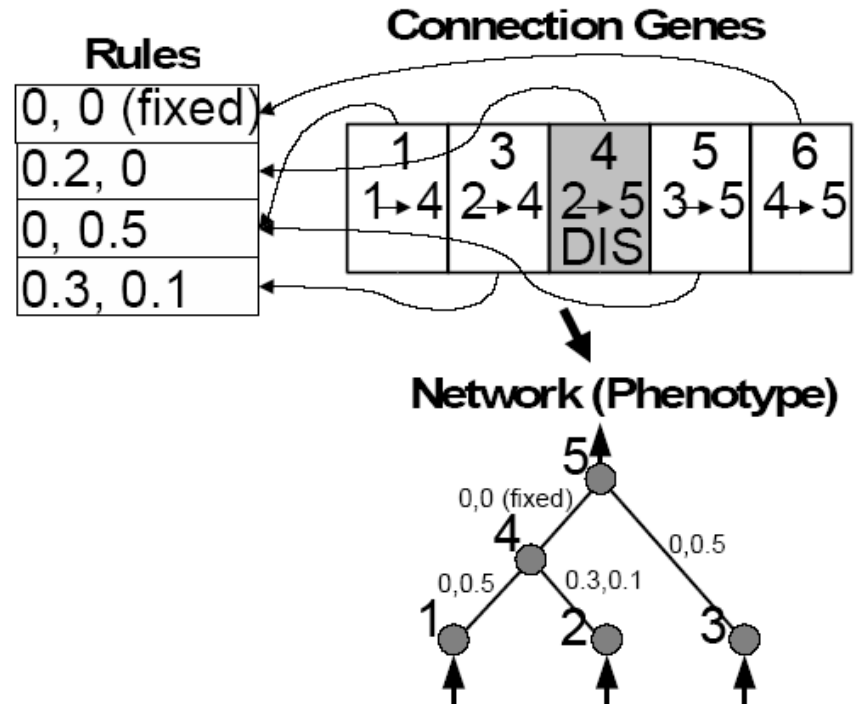
- Fitness = postavit síť, inicializovat, zkusit učit, nejlépe víckrát
- Přímé kódování
 - Realizují strukturu sítě např. jako binární matici, pracují s evolucí linearizovaných dlouhých binárních vektorů
- Gramatické kódování
 - Kitano navrhl vyvíjet 2D formální gramatiky, které jsou „programem“ pro vytvoření matice

Učení struktury 2

- Růst sítě ve 2D
 - Rané pokusy v evoluční robotice, velmi neefektivní
- Celulární kódování
 - Gruau navrhl použití Genetického programování
 - Program v GP je vlastně programem, jak nakreslit síť operacemi přidej neuron, rozděl neuron sériově, rozděl neuron paralelně, přepoj synapsi, apod.

NEAT

- Smiley – Neuroevolution of augmenting topologies
- Síť je seznam hran, každá hrana má informace o svých vrcholech, vahách, a *rodné číslo*.



NEAT pokr.

- Kříží se jen hrany se stejnými rodnými čísly, zbytek se přenáší do jedince beze změn
 - Tím se umožňuje křížení jen mezi hranami, které mají stejný evoluční původ
- Na vektorech hran s rodnými čísly se definuje podobnost (jak moc se dvě sítě od sebe liší hranami)
 - Při evoluci jsou podobné sítě zahrnuty do stejného druhu, fitness je relativní vzhledem k druhu
 - To umožní ochranu nových topologií než se jim dovyvinou váhy

Strojové učení

Strojové učení

- Učení pravidel na základě předkládaných dat
 - Data mining
 - Expertní systémy
 - Učení agentů, robotů
- Dva základní přístupy:
 - Michiganský (Holland): pravidlo je jedinec
 - Pittsburský: jedinec je množina pravidel

Michigan

- Holland v 80.letech: learning classifier systems
- Jedinec GA je pravidlo, celá populace pak funguje jako řídicí či expertní systém
- Jednoduchá pravidla:
 - Pravá strana: příznak nastal/ne/don't care
 - Levá strana: kód akce či klasifikace kategorie
- Pravidla mají váhu (úspěšnost),
- Evoluce probíhá jen občas a jen na části populace

Michigan pokr.

- Problém reaktivnosti (absence vnitřní paměti)
 - Pravidla mohou mít na pravé straně další „zprávy“ a na levé straně „receptory“ na jejich příjem, plus fronta zpráv
 - Pak ale jen některá pravidla vedou k akci, za kterou je odměna/trest, je nutno zkomplikovat rozdělení odměny – pro celý řetěz úspěšných pravidel
 - Pravidla musejí dát část svých peněz, když chtějí soupeřit o možnost být v cestě k řešení
 - *Bucket brigade algorithm*, v praxi velmi komplikované a těžkopádné

Pitt

- Jedinci jsou množiny pravidel
- Ohodnocení komplikovanější
 - Priority pravidel, konflikty
 - False positives, false negatives
- Genetické operátory komplikovanější
 - Typicky vede k desítkám operátorů pracujících na úrovni celých množin, jednotlivých pravidel, termů v pravidlech
 - Důraz na bohatou reprezentaci domén (množiny, výčtové typy, intervaly, ...)