

3. Umělý život (Artificial Life)

Umělý život

(Úvod do evolučních algoritmů a evolučních strategií)

8. března 2019

3. Umělý život (Artificial Life)

What is Life?

Webster dictionary:

- a) the quality that distinguishes a vital and functional being from a dead body or purely chemical matter
- b) the state of a material complex or individual characterized by the capacity to perform certain functional activities including metabolism, growth, and reproduction

Samuel Butler:

Life is one long process of getting tired.

Jacques Prevert:

Life too is an epidemic, sons catching it from fathers,
daughters from mothers.

3. Umělý život (Artificial Life)

Shakespeare (Macbeth V.v.):

Life's but a walking shadow, a poor player
That struts and frets his hour upon the stage
And then is heard no more: it is a tale
Told by an idiot, full of sound and fury
Signifying nothing.

John Keats (Sleep and Poetry):

Stop and consider! Life is but a day;
A fragile dew-drop on its perilous way
From a tree's summit;

3. Umělý život (Artificial Life)

What is Artificial Life?

"The study of man-made systems that exhibit behaviors characteristic of natural living systems."

C. G. Langton. "Artificial Life." In: C. G. Langton, editor. *Artificial Life*, Volume VI of *SFI Studies in the Sciences of Complexity*, pages 1-47, Addison-Wesley, Redwood City, CA, 1989.

**"Artificial Life is a field of study devoted to understanding life by attempting to abstract the fundamental dynamical principles underlying biological phenomena, and recreating these dynamics in other physical media - such as computers - making them accessible to new kinds of experimental manipulation and testing.
...**

In addition to providing new ways to study the biological phenomena associated with life here on Earth, *life-as-we-know-it*, Artificial Life allows us to extend our studies to the larger domain of "bio-logic" of possible life, *life-as-it-could-be ...*"

C. G. Langton. "Preface." In: C. G. Langton, C. Taylor, J. D. Farmer, and S. Rasmussen (Eds.), *Artificial Life II*, Volume X, *Studies in the Sciences*, pp. xiii-xviii, Addison-Wesley, 1992

3. Umělý život (Artificial Life)

"Among all of the things that artificial life is or will come to be, however, it is probably safe to say that the field as a whole represents an attempt to vastly increase the role of synthesis in the study of biological phenomena."

C. G. Langton. "Editor's introduction". *Artificial Life Journal*, Volume 1, Number 1/2, pp. v-viii, 1994. The MIT Press, Cambridge, MA.

"Artificial Life (AL) is the enterprise of understanding biology by constructing biological phenomena out of artificial components, rather than breaking natural life forms down into their component parts. It is the synthetic rather than the reductionist approach."

T. S. Ray. "An evolutionary approach to synthetic biology: Zen and the art of creating life." *Artificial Life Journal*, Volume 1, Number 1/2, pp. 179-209, 1994. The MIT Press, Cambridge, MA.

"... Artificial Life models ... are powerful enough to capture much of the complexity of living systems, yet in a form that is more easily manipulable, repeatable, and subject to precisely controlled experiment than are the corresponding natural systems."

C. Taylor and D. Jefferson: "Artificial life as a tool for biological inquiry." *Artificial Life Journal*, Vol. 1

3. Umělý život (Artificial Life)

"Alife is a constructive endeavor: Some researchers aim at evolving patterns in a computer; some seek to elicit social behaviors in real-world robots; others wish to study life-related phenomena in a more controllable setting, while still others are interested in the synthesis of novel lifelike systems in chemical, electronic, mechanical, and other artificial media. Alife is an experimental discipline, fundamentally consisting of the observation of run-time behaviors, those complex interactions generated when populations of man-made, artificial creatures are immersed in real or simulated environments. Published work in the field usually relates the conception of a model, its instantiation into real-world or simulated objects, and the observed behavior of these objects in a collection of experiments."

E. M. A. Ronald, M. Sipper, and M. S. Capcarrère. "Design, Observation, Surprise! A Test of Emergence." *Artificial Life Journal*, Volume 5, Number 3, pages 225-239, Summer 1999. The MIT Press, Cambridge, MA.

3. Umělý život (Artificial Life)

Artificial Life

(a basic definition)

Artificial Life ("AL" or "Alife") is the name given to a new discipline that studies "natural" life by attempting to recreate biological phenomena from scratch within computers and other "artificial" media. Alife complements the traditional analytic approach of traditional biology with a synthetic approach in which, rather than studying biological phenomena by taking apart living organisms to see how they work, one attempts to put together systems that behave like living organisms.

3. Umělý život (Artificial Life)

Artificial Life

(a basic definition)

Umělý život ("AL" or "Alife") je název pro novou (vědní?) disciplínu, která studuje "přirozený" život přístupem k vytváření biologických jevů od jejich počátku na bázi počítačů a dalších "artificial" medií. Metody umělého života kombinují tradiční analytický přístup klasické biologie se syntetickým přístupem, který spíše než studium biologických jevů rozebírá živé organismy z hlediska jejich jednotlivých funkcí a snaží se navrhovat umělé systémy, které se však chovají jako živé organismy.

3. Umělý život (Artificial Life)

Evoluční strategie, evoluční algoritmy

Cílem evolučních strategií a evolučních algoritmů je **modelování evolučních procesů** pro účely, které **nemají nic společného s biologií**.

V přírodě biologičtí jedinci jedné populace mezi sebou soutěží o přežití a možnost reprodukce na základě toho, jak dobře jsou přizpůsobeni prostředí. V průběhu mnoha generací se struktura dané populace vyvíjí na základě Darwinovy teorie o přirozeném výběru a přežívání jen těch jedinců, kteří mají největší sílu (fitness).

Evoluční strategie, evoluční algoritmy

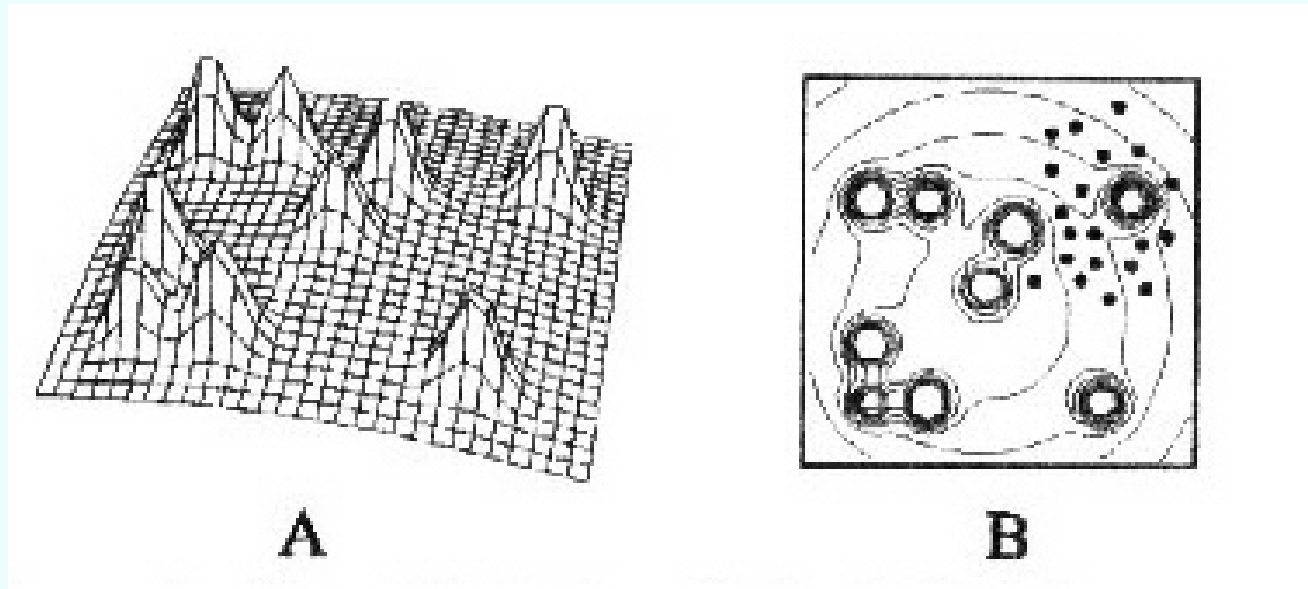
Cílem evolučních strategií a evolučních algoritmů je **modelování evolučních procesů** pro účely, které **nemají nic společného s biologií**.

V přírodě biologičtí jedinci jedné populace mezi sebou soutěží o přežití a možnost reprodukce na základě toho, jak dobře jsou přizpůsobeni prostředí. V průběhu mnoha generací se struktura dané populace vyvíjí na základě Darwinovy teorie o přirozeném výběru a přežívání jen těch jedinců, kteří mají největší sílu (fitness).

Evoluční algoritmy se snaží využít modelů evolučních procesů, aby našly řešení náročných a rozsáhlých úloh. Veškeré takové modely mají několik společných rysů:

- 1) pracují zároveň **s celou skupinou možných řešení** zadaného problému místo práce s jednotlivými řešeními;
- 2) řešení **postupně vylepšují** zařazováním nových řešení, získaných kombinací původních;
- 3) kombinace řešení jsou **následovány náhodnými změnami** a vyřazováním nevýhodných řešení.

3. Umělý život (Artificial Life)



Znázornění povrchu fitness (A), který vyjadřuje závislost síly (fitness) jedinců na složení jejich genotypu. V tomto jednoduchém případě se jedná o organismus s dvojgenovým chromozómem, přičemž jeho složky jsou reálná čísla. Konturový graf (B) odpovídá povrchu fitness, oblak bodů znázorňuje populaci jedinců.

3. Umělý život (Artificial Life)

Smysl orientace na využití evolučních strategií a algoritmů:

Existují úlohy, které příroda zvládá velmi snadno, zatímco člověkem navržené algoritmy se hroučí.

3. Umělý život (Artificial Life)

Smysl orientace na využití evolučních strategií a algoritmů:

Existují úlohy, které příroda zvládá velmi snadno, zatímco člověkem navržené algoritmy se hroutí.

Dva druhy evolučních algoritmů:

- algoritmy napodobující přirozené (přírodní) evoluční procesy založené na teorii přirozeného výběru a „zvládající“ problémy adaptace na měnící se podmínky a učení – **genetické algoritmy**

3. Umělý život (Artificial Life)

Smysl orientace na využití evolučních strategií a algoritmů:

Existují úlohy, které příroda zvládá velmi snadno, zatímco člověkem navržené algoritmy se hroutí.

Dva druhy evolučních algoritmů:

- algoritmy napodobující přirozené (přírodní) evoluční procesy založené na teorii přirozeného výběru a „zvládající“ problémy adaptace na měnící se podmínky a učení – **genetické algoritmy**
- algoritmy napodobující činnost mozku – **umělé neuronové sítě**, tvořené z neuronů a propojení mezi nimi, kterými se simuluje chování nervové soustavy

3. Umělý život (Artificial Life)

Genetické algoritmy

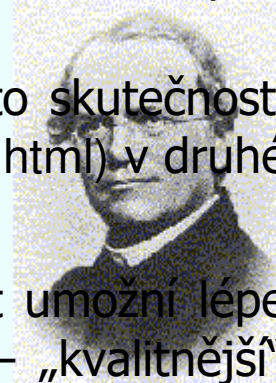
Základní myšlenkou genetických algoritmů je **snaha napodobit vývoj a učení** nějakého živočišného druhu a takto vzniklý algoritmus použít při řešení úloh, které se vyskytují ve složitém, případně i měnícím se prostředí, v němž člověk není schopen dopředu nadefinovat všechny vzniklé případy a správné reakce na ně.

3. Umělý život (Artificial Life)

Genetické algoritmy

Základní myšlenkou genetických algoritmů je **snaha napodobit vývoj a učení** nějakého živočišného druhu a takto vzniklý algoritmus použít při řešení úloh, které se vyskytují ve složitém, případně i měnícím se prostředí, v němž člověk není schopen dopředu nadefinovat všechny vzniklé případy a správné reakce na ně.

Informace se v živých organismech uchovává v **genech** – objevitelem této skutečnosti byl Johann Gregor **Mendel** (viz <http://www.mendelu.cz/univerzita/mendel.html>) v druhé polovině 19. století. Založeny jsou na třech různých „operacích“:



1. **Reprodukce** – každý jedinec je trochu jiný; pokud mu jeho odlišnost umožní lépe přežít, zanechá po sobě více potomků než jedinci takto nevybavení – „kvalitnější“ jedinci se více a častěji reprodukuje
2. **Křížení** – geny se při vytváření nové populace mohou křížit (anglicky crossover) – část informace se vezme z otce, část z matky – například z modrého a červeného hrachu vznikne hrach modročervený
3. **Mutace** – geny mohou vlivem okolí (např. radioaktivního záření) mutovat (mutation) – zpravidla dochází k malé náhodné změně. Tím se vytvářejí nové populace, které příroda buďto přijme nebo odsoudí k zániku

Mendelovy zákony dědičnosti

1. Při vzájemném křížení homozygotů (F_1 generace) vzniká potomstvo, které je svým genotypem i fenotypem jednotné – zákon o jednotnosti první generace kříženců.

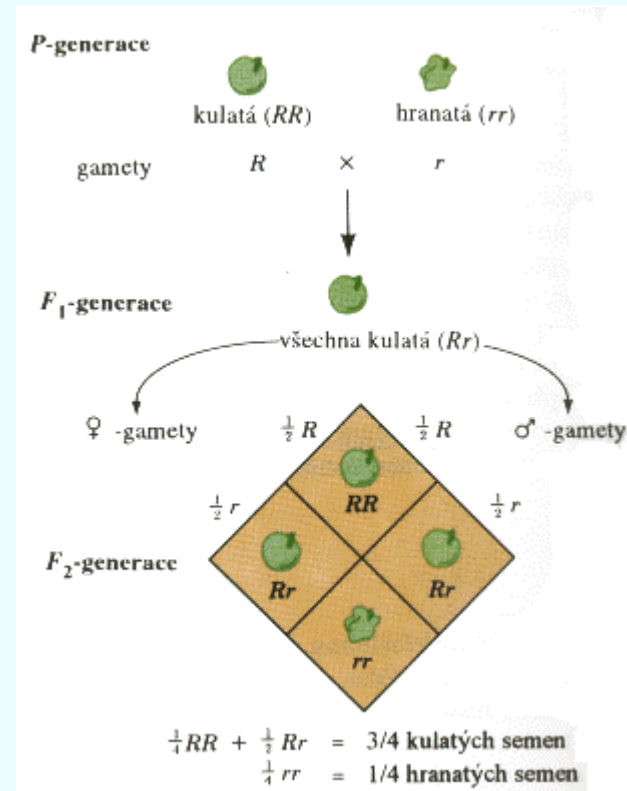
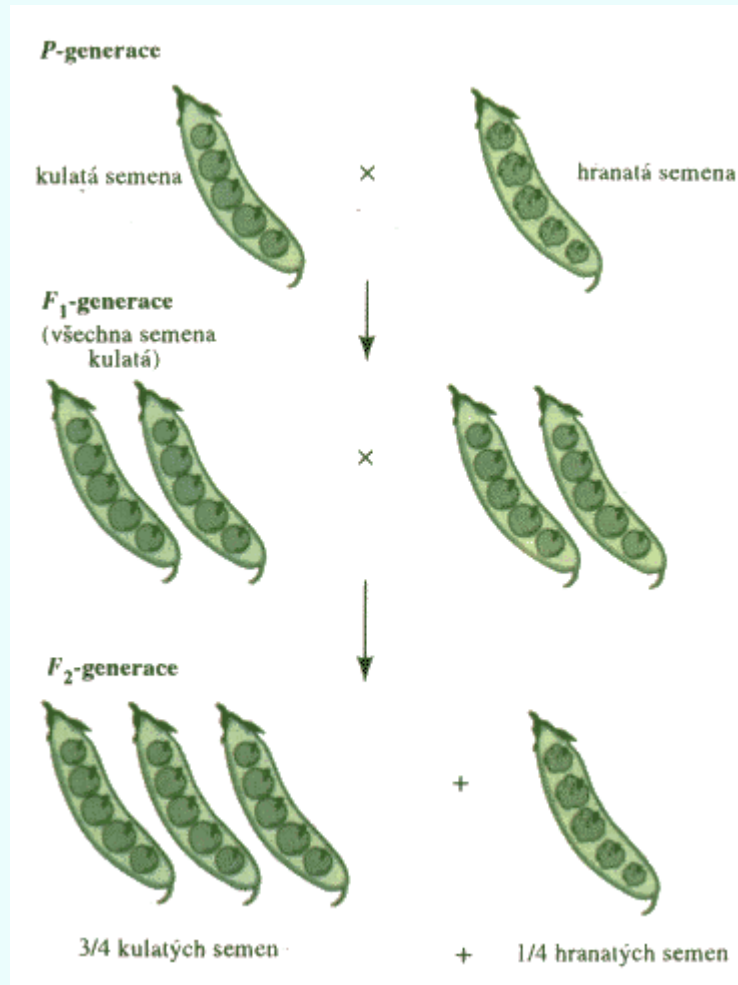
Mendelovy zákony dědičnosti

- 1.** Při vzájemném křížení homozygotů (F_1 generace) vzniká potomstvo, které je svým genotypem i fenotypem jednotné – zákon o jednotnosti první generace kříženců.
- 2.** Při vzájemném křížení heterozygotů (F_2 generace) vzniká potomstvo, které je genotypově i fenotypově různorodé, přičemž poměrné zastoupení homozygotů i heterozygotů v tomto potomstvu (proto i dominantních a recesivních fenotypů) je pravidelné a stálé – zákon o segregaci alely a jejich kombinaci ve druhé generaci kříženců.

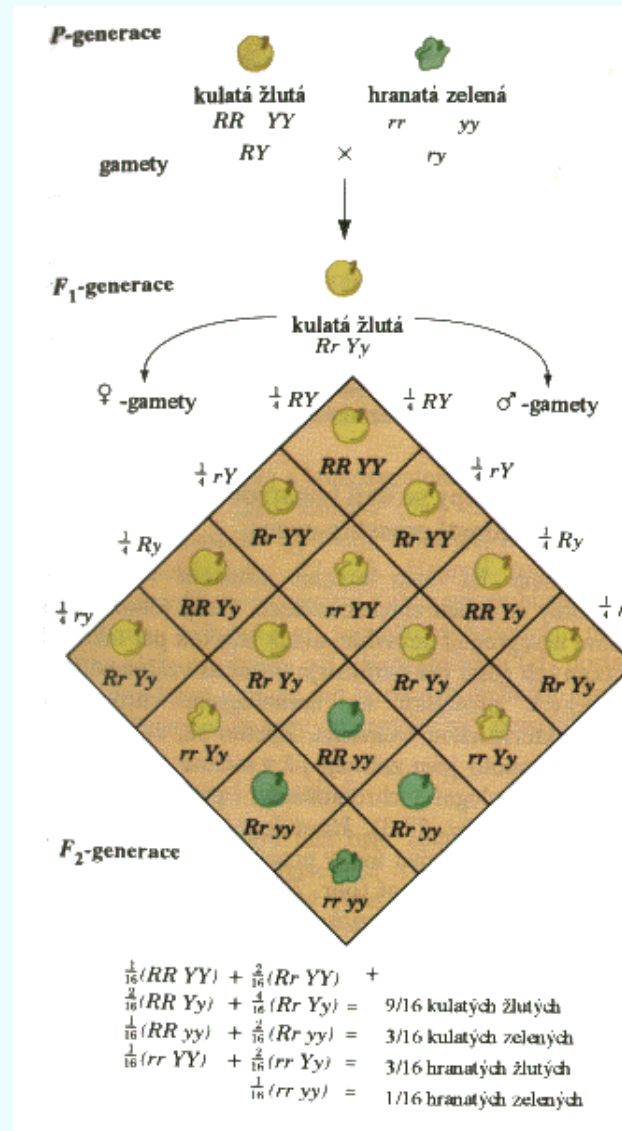
Mendelovy zákony dědičnosti

- 1.** Při vzájemném křížení homozygotů (F_1 generace) vzniká potomstvo, které je svým genotypem i fenotypem jednotné – zákon o jednotnosti první generace kříženců.
- 2.** Při vzájemném křížení heterozygotů (F_2 generace) vzniká potomstvo, které je genotypově i fenotypově různorodé, přičemž poměrné zastoupení homozygotů i heterozygotů v tomto potomstvu (proto i dominantních a recesivních fenotypů) je pravidelné a stálé – zákon o segregaci alely a jejich kombinaci ve druhé generaci kříženců.
- 3.** Při vzájemném křížení heterozygotů (F_3 generace) ve více genových párech vzniká genotypově i fenotypově různorodé potomstvo, v němž je pravidelné a stálé zastoupení (poměrné zastoupení) genotypů všech možných kombinací mezi rozdílnými alelami všech heterozygotních alelových párů (9 : 3 : 3 : 1) – zákon o volné (nezávislé) kombinovanosti alel různých alelových párů.

3. Umělý život (Artificial Life)



3. Umělý život (Artificial Life)



3. Umělý život (Artificial Life)

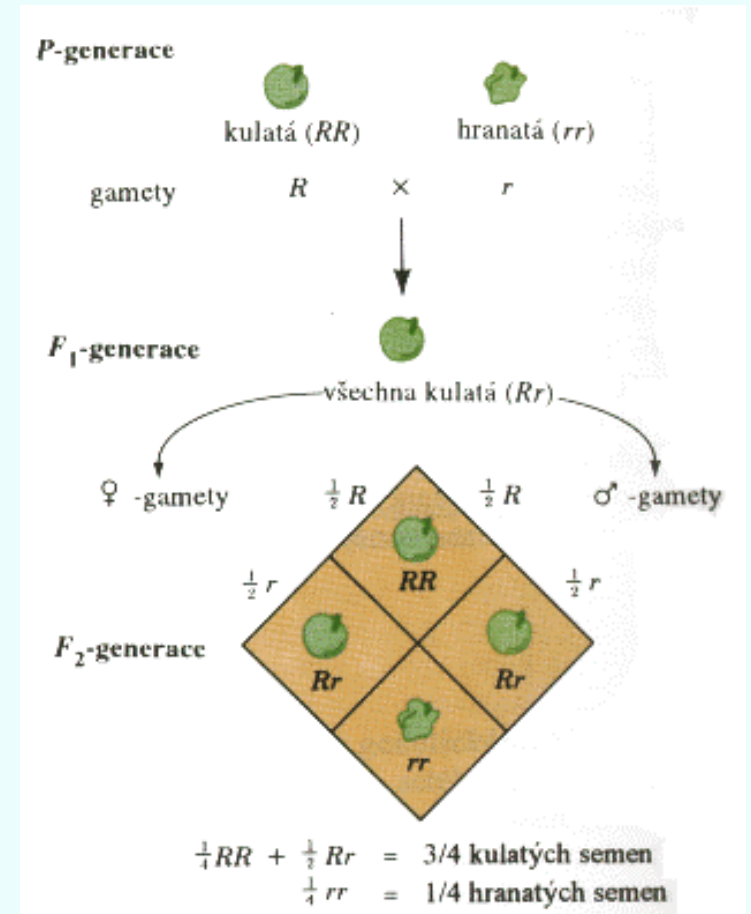
Mendel zjistil, že křížením rodičů (P) lišících se v jediném znaku (např. tvar semen) vzniká potomstvo (F_1 , první filiální generace), ve kterém mají všichni jedinci znak pouze jednoho z rodičů, v tomto případě kulatá semena. Znak, projevující se u F_1 generace, se jmenuje **dominantní**, alternativní znaky se nazývají **recesivní**. V generaci F_2 (potomci F_1 rodičů) se dominantní znak objevuje u tří čtvrtin potomstva, recesivní pak u jedné čtvrtiny.

Hrách s recesivním znakem poskytuje přímé potomstvo, tzn. výsledkem křížení mezi recesivními F_2 je potomstvo F_3 mající rovněž recesivní znak. Příslušníci F_2 generace vykazující dominantní znak se však dělí do dvou kategorií: jednu třetinu vytváří jednotné potomstvo, zatímco zbylá část poskytuje potomstvo s poměrem dominantních znaků k recesivním 3:1 (jako u generace F_2).

Mendel vysvětloval toto pozorování hypotézou, že různé páry kontrastních znaků jsou každý výsledkem faktoru (nyní nazývaného **gen**), který má alternativní formy (**alely**). Každá rostlina obsahuje pár genů určujících určitý znak, přičemž od každého z rodičů získala po jednom genu. Alely pro tvar semene mají symbol **R** pro kulatá semena a **r** pro semena hranatá. Tak vznikají dvě možnosti genotypů (složení genů):

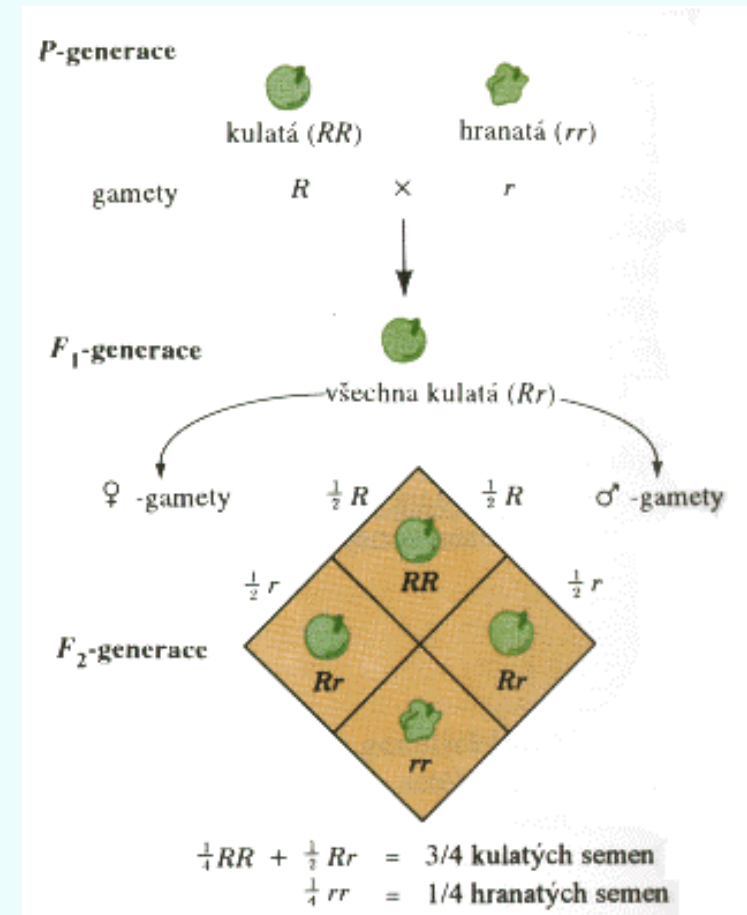
3. Umělý život (Artificial Life)

- a) rostliny čisté linie s kulatými nebo hranatými semeny mají genotyp RR a rr , jsou označovány jako **homozygoti** ve tvaru semen.



3. Umělý život (Artificial Life)

- a) rostliny čisté linie s kulatými nebo hranatými semeny mají genotyp RR a rr , jsou označovány jako **homozygoti** ve tvaru semen.
- b) rostliny s genotypem Rr jsou **heterozygoti** ve tvaru semen a jejich **fenotypem** (projev znaku) jsou kulatá semena, protože R je dominantní. Tyto dvě alely se žádným způsobem v rostlinách nemísí a prostřednictvím gamet se přenášejí na potomstvo.



3. Umělý život (Artificial Life)

Formálně platí:

Genetické algoritmy (GA), založené na principu přirozeného výběru (prvně publikovány v roce 1975), jsou dnes základem **evolučního programování** (EP), **evolučních strategií** (ES) a **genetického programování** (GP), přičemž platí

$$\mathbf{GP = EP + ES + GA}$$

3. Umělý život (Artificial Life)

Formálně platí:

Genetické algoritmy (GA), založené na principu přirozeného výběru (prvně publikovány v roce 1975), jsou dnes základem **evolučního programování** (EP), **evolučních strategií** (ES) a **genetického programování** (GP), přičemž platí

$$\mathbf{GP = EP + ES + GA}$$

Genetické algoritmy v informatice a genetické programování jsou sice založeny na Mendelově vývojové teorii, avšak v poněkud upraveném (kybernetickém, resp. informatickém) pojetí – viz dále

3. Umělý život (Artificial Life)

Genetické algoritmy – kybernetické pojetí:

Řešení úlohy popíšeme řetězcem vstupních parametrů nějaké funkce, parametry funkce se mění obvykle náhodně, kvalita řešení je určena výstupní hodnotou optimalizované funkce, nazývané *fitness*.

1. **Reprodukce** – kvalitní jedinci se častěji reprodukují, při reprodukci předávají potomkům část své genetické informace, déle přežívají, takže mají možnost déle se reprodukovat.
2. **Křížení** – zabraňuje vývoji nelišících se jedinců, vznikají jedinci s „prohozenými“ vlastnostmi (část informace se „dědí“ po otci, část po matce, a to většinou náhodně).
3. **Mutace** – druhý způsob zabránění vývoji nelišících se jedinců, umožňují adaptaci na měnící se prostředí, probíhají jen u malého procenta vždy náhodně vybraných vstupních parametrů.

Nositelem genetické informace u biologických systémů jsou **chromozómy**, které reprezentujeme jako řetězce číselných hodnot, nejčastěji binárních.

3. Umělý život (Artificial Life)

Implementace evolučních strategií a genetických algoritmů

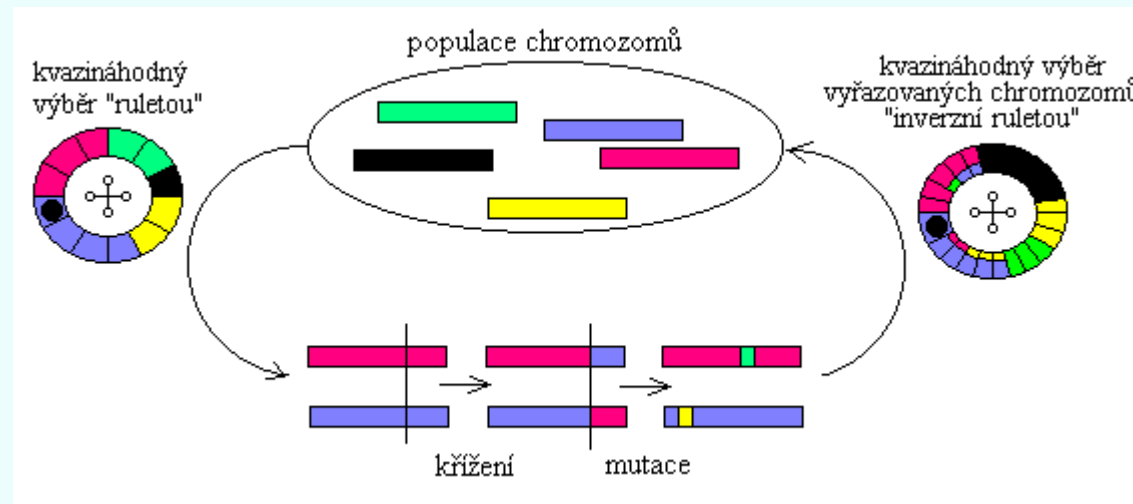
V přírodě a tedy i v genetických algoritmech platí, že kvalitnější jedinci se častěji rozmnožují a také déle přežívají, proto zanechávají více potomků, kteří nesou dál část jejich genetické informace. Přesto je tento výběr ovlivněn náhodou, neboť i kvalitní řešení se vybírají k dalšímu přežití sice úměrně své kvalitě, ale náhodně. V informatice pro tento tzv. kvazináhodný výběr podle kvality použijeme zařízení, nazvané **ruleta**.

Jedinec řešení (chromozóm) je nejčastěji implementován lineární posloupností symbolů, např. bitovým řetězcem. Každý chromozóm je ohodnocen funkcí, která mu přiřadí tzv. sílu, vyjádřenou kladným reálným číslem. Čím má chromozóm větší sílu, tím je kvalitnější.

Situaci, kdy se většina jedinců natolik zkříží, že by další křížení produkovalo stejné jedince a tudíž nemělo smysl, brání mutace. Mutace také nabízejí možnost adaptovat populaci na měnící se prostředí, ale musejí se provádět jen u malého procenta populace, protože často už dobrá řešení spíše pokazí.

3. Umělý život (Artificial Life)

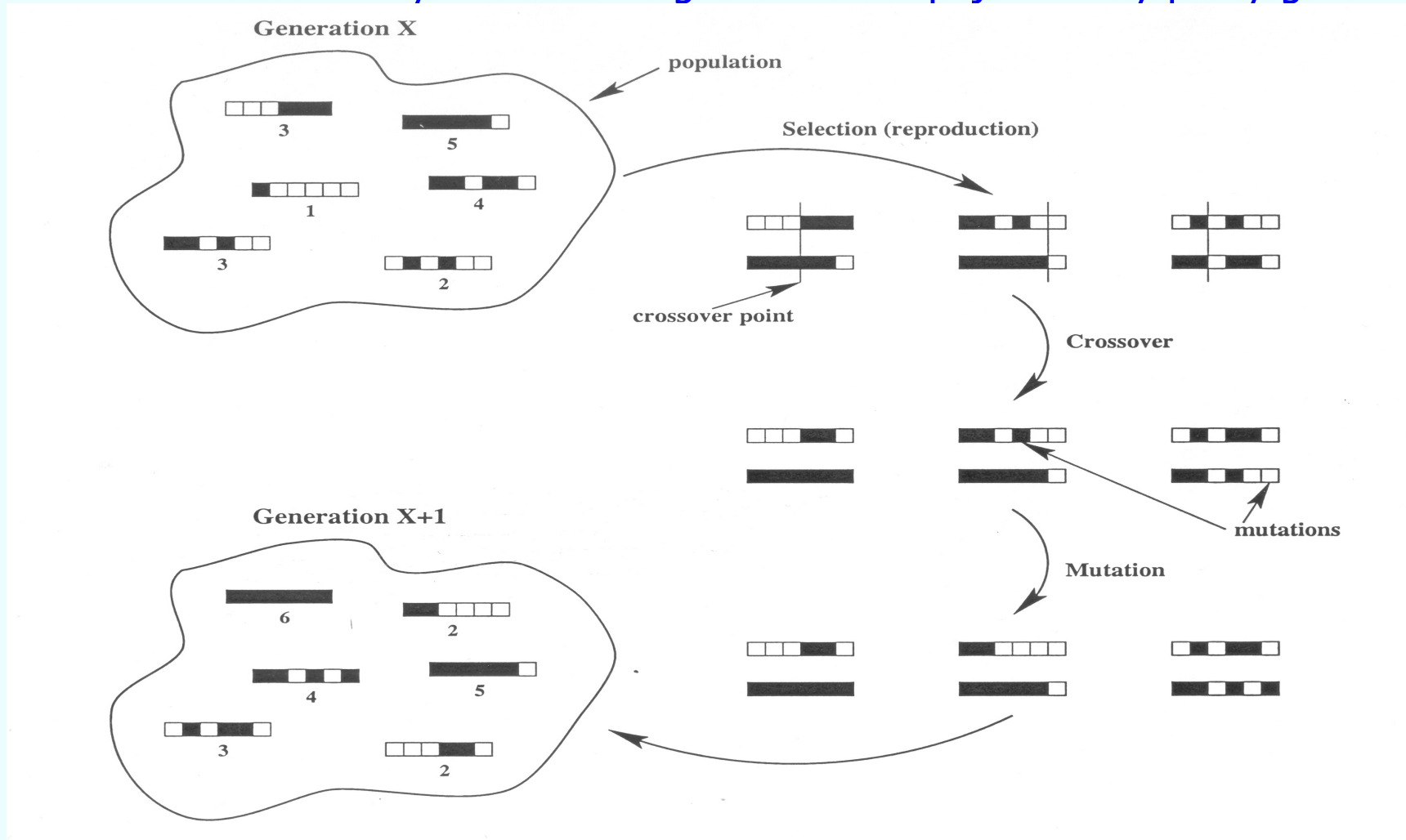
Schématické znázornění:



Podstata algoritmu: Z populace se (kvazi)náhodně vyberou dva chromozómy, které si křížením vymění opět (kvazi)náhodně vybranou část řetězců. Výsledné chromozómy se pak ještě podrobí mutaci, která překlopí náhodně zvolený(é) bit(y). Takto nově vytvořená dvojice se vrací do populace, kde vytěsňují dvojici kvazináhodně vybraných chromozómů s malou silou.

3. Umělý život (Artificial Life)

Druhá možnost – do vytvoření X+1. generace vstupují všechny prvky generace X:



3. Umělý život (Artificial Life)

Symbolický zápis algoritmu:

Create a random population.

Evaluate the fitness of the population.

Repeat

 Select pairs from the population to be parents.

 Crossover parents and mutate the resulting structures.

 Place the new structures in the population, evaluating their fitness.

Until Done.

Figure 2.1: Algorithm for a simple genetic algorithm

3. Umělý život (Artificial Life)

Př.: Implementace genetického algoritmu v pseudoPascalu:

```
t := 0;  
P0 := náhodně vygenerovaná populace chromozomů;  
ohodnot' každý chromozom z populace P silou;  
while t < tmax do  
begin  
  t := t + 1;  
  Q := nové chromozomy vzniklé reprodukcí kvazináhodně  
  vybraných chromozomů z Pt-1 s největší silou;  
  ohodnot' každý chromozom z Q silou;  
  R := kvazináhodně vybrané chromozomy z Pt-1 s nejmenší silou;  
  Pt := (Pt-1 \ R) ∪ Q;  
end;
```

P_t ... populace chromozomů v čase t

Q ... subpopulace potomků

R ... subpopulace náhodně vybraných chromozomů s nejmenší silou

operátor \ zde znamená vyloučení (vypuštění) množiny R z množiny P_{t-1}
(množinová exkluze)

3. Umělý život (Artificial Life)

Využití genetických algoritmů

Velké uplatnění nacházejí genetické algoritmy při

- vytváření rozvrhů práce pro stroje v továrnách,
- v teorii her a v managementu,
- při řešení optimalizačních problémů multimodálních funkcí,
- při řízení robotů,
- při návrzích rozpoznávacích systémů a
- při simulaci úloh umělého života.

Genetické algoritmy překvapivě dobře fungují při řešení problémů, kde téměř všechny ostatní algoritmy selhávají, např. pro NP-úplné problémy, tj. kde výpočetní čas je exponenciálně nebo faktoriálně závislý na počtu proměnných. Nemá však smysl je používat u relativně jednoduchých optimalizovaných funkcí nebo u funkcí, pro které existují specializované algoritmy.

Genetické programování

Na přelomu osmdesátých a devadesátých let americký informatik John Koza ze Stanfordské univerzity navrhl originální modifikaci genetického algoritmu, kterou nazval genetické programování. Při tomto přístupu je původní reprezentace chromozomů znakovými řetězci nahrazena složitějšími funkcemi. V nejjednodušší verzi genetického programování se funkce rovnají výrazům obsahujícím proměnné, konstanty, základní aritmetické operace a elementární funkce.

K tomu, aby genetické programování bylo schopné řešit také složitější algoritmické úlohy, musí obsahovat vhodné příkazy pro zápis do paměti, načítání z paměti a pro opakování příkazu. Pokud genetické programování rozšíříme o tyto tři příkazy, potom je tzv. Turingovsky úplné. To tedy znamená, že je schopné realizovat každý algoritmus, který je realizovatelný na Turingově počítači.

3. Umělý život (Artificial Life)

Příklad použití genetického programování:

V genetickém programování se chromozómy reprezentují složitějšími funkcemi. V nejjednodušší verzi genetického programování se funkce rovnají **výrazům** obsahujícím **proměnné, konstanty, základní aritmetické operace** a **elementární funkce**.

Jednoduchá funkce $x(1+x)$ budiž reprezentována pomocí **syntaktického stromu** (parse-tree). **Interpretace** vyhodnocení tohoto **syntaktického stromu** je pak velmi jednoduchá – průchod in-order. První problém, který musíme v rámci genetického programování vyřešit, je způsob vyhodnocení chromozómu stromu. Toto vyhodnocení je formálně chápáno jako funkce $y = f(x)$, přiřazující nezávislé proměnné (vstupu) x závislou proměnnou (výstup) y .

Mějme dānu **trénovací množinu** A , která obsahuje p bodů (x_i, y_i) tj. $A = (x_i, y_i)$; $i = 1, 2, \dots, p$. Naším cílem pak je najít takovou funkci $f(x)$ (reprezentovanou syntaktickým stromem), která by minimalizovala **účelovou funkci**, tj. např. součet absolutních hodnot rozdílů vypočtených a zadaných hodnot y z trénovací množiny:

$$F(f) = \sum_{i=1}^p |f(x_i) - y_i|$$

3. Umělý život (Artificial Life)

Je-li tento součet nulový, funkce $f(x)$ přesně vystihuje body z trénovací množiny. Najít takovou funkci se nám nejspíš nepodaří, ale pokusíme se najít alespoň takovou, pro níž bude uvedený součet co nejmenší.

Takto formulovaná úloha je blízká **regresní analýze**, kde hledáme vhodné koeficienty funkce, které minimalizují účelovou funkci. U regresní analýzy je však tvar funkce daný, na rozdíl od principu využitého v genetických algoritmech. John Koza proto nazval takový zevšeobecněný přístup k regresi slovy **symbolická regrese**.

3. Umělý život (Artificial Life)

Je-li tento součet nulový, funkce $f(x)$ přesně vystihuje body z trénovací množiny. Najít takovou funkci se nám nejspíš nepodaří, ale pokusíme se najít alespoň takovou, pro níž bude uvedený součet co nejmenší.

Takto formulovaná úloha je blízká **regresní analýze**, kde hledáme vhodné koeficienty funkce, které minimalizují účelovou funkci. U regresní analýzy je však tvar funkce daný, na rozdíl od principu využitého v genetických algoritmech. John Koza proto nazval takový zevšeobecněný přístup k regresi slovy **symbolická regrese**.

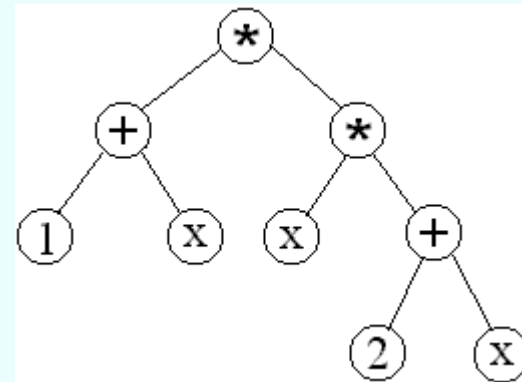
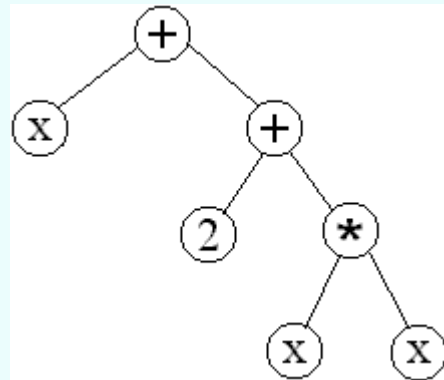
Křížení a mutace v syntaktických stromech

Pro **křížení** v syntaktických stromech vybereme náhodně dva uzly (vrcholy), z nichž žádný nesmí být kořenem. **Výměnou příslušných podstromů** těchto uzlů dostaneme **dva nové** syntaktické stromy, které jsou **výsledkem křížení** původních syntaktických stromů.

Mutace je definována obdobně. Náhodně se vybere uzel (vrchol), který není kořenem, a příslušný podstrom se **zamění za nový**, náhodně vygenerovaný podstrom.

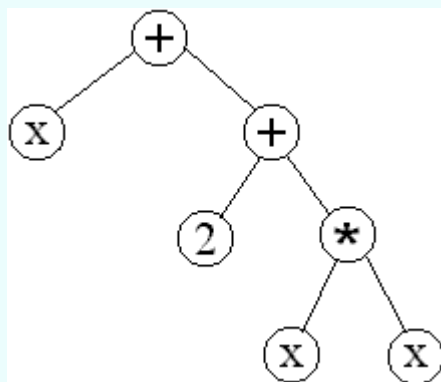
3. Umělý život (Artificial Life)

Příklad dvou syntaktických stromů:

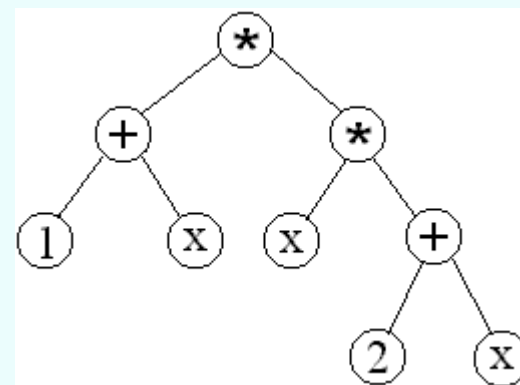


3. Umělý život (Artificial Life)

Příklad dvou syntaktických stromů:



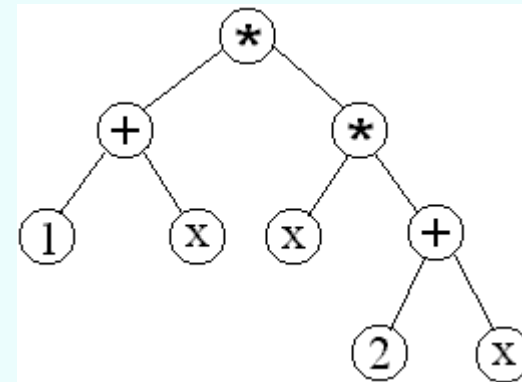
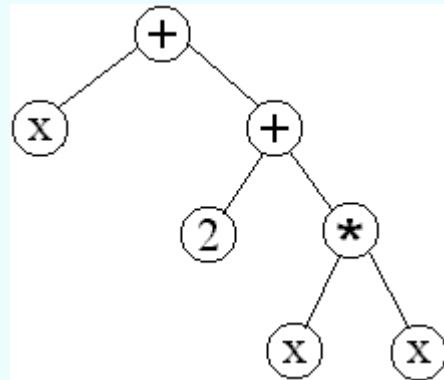
Interpretace: $x + (2 + x * x)$



$(1 + x) * x * (2 + x)$

3. Umělý život (Artificial Life)

Příklad dvou syntaktických stromů:



Interpretace: $x+(2+x*x)$

$(1+x)*x*(2+x)$

Ohodnocení syntaktických stromů silou:

Ohodnocení syntaktických stromů je vypočítáváno v souladu s hodnotou účelové funkce $F(f)$ pomocí nepřímé úměrnosti. To znamená, že chromozóm s největší hodnotou účelové funkce $F(f)$ je ohodnocen nejmenší silou a naopak.

Stochastické optimalizační algoritmy

Stochastické optimalizační algoritmy se používají pro optimalizaci mnohoparametrových funkcí s "divokým" průběhem, tj. s mnoha extrémy nebo s neznámým gradientem. Založeny jsou na mnohokrát opakované náhodné volbě počátečního řešení úlohy a za výsledné řešení se vezme nejlepší výsledek. Stochastičnost takového postupu spočívá jen v náhodném výběru počátečního řešení, následovně použitý optimalizační algoritmus je striktně deterministický.

Stochastické optimalizační algoritmy

Stochastické optimalizační algoritmy se používají pro optimalizaci mnohoparametrových funkcí s "divokým" průběhem, tj. s mnoha extrémy nebo s neznámým gradientem. Založeny jsou na mnohokrát opakované náhodné volbě počátečního řešení úlohy a za výsledné řešení se vezme nejlepší výsledek. Stochastičnost takového postupu spočívá jen v náhodném výběru počátečního řešení, následovně použitý optimalizační algoritmus je striktně deterministický.

K těmto algoritmům a metodám patří:

- stochastický "horolezecký" (hill climbing) algoritmus,
- genetické algoritmy,
- evoluční strategie,
- simulované žíhání (simulated annealing),
- zakázané prohledávání (tabu search)

Evoluční strategie a evoluční programování

Když kolem roku 1963 začali Hans-Paul Schwefel a Ingo Rechenberg, tehdy ještě studenti, na Technické univerzitě v Berlíně s napodobováním vývoje v přírodě, byli přesvědčeni, že jejich metoda nejlépe napodobuje evoluci v živé přírodě. Proto svoji metodu nazvali tak obecně – **evoluční strategie**. Postupem času se však ukázalo, že tento způsob řeší jen určitý typ úloh, a to zejména ve stavebním a strojním inženýrství. **Genetické algoritmy** tak nejsou podřazeny evolučním strategiím, ale naopak je svojí podstatou a možná také popularitou zastiňují.

Evoluční strategie a evoluční programování

Když kolem roku 1963 začali Hans-Paul Schwefel a Ingo Rechenberg, tehdy ještě studenti, na Technické univerzitě v Berlíně s napodobováním vývoje v přírodě, byli přesvědčeni, že jejich metoda nejlépe napodobuje evoluci v živé přírodě. Proto svoji metodu nazvali tak obecně – **evoluční strategie**. Postupem času se však ukázalo, že tento způsob řeší jen určitý typ úloh, a to zejména ve stavebním a strojním inženýrství. **Genetické algoritmy** tak nejsou podřazeny evolučním strategiím, ale naopak je svojí podstatou a možná také popularitou zastiňují.

Evoluční strategie, stejně jako evoluční programování a genetické algoritmy, jsou vhodné především **pro optimalizaci funkcí s mnoha maximy a minimy**, u nichž nejsme schopni určit analytické řešení a klasické gradientové techniky selhávají.

3. Umělý život (Artificial Life)

1) výběr potomků

Evoluční strategie – u původní implementace lepší potomek jednoduše vytěsňuje horšího rodiče, který je odstraněn z populace. Pokud potomek není lepší než rodič, pak rodič zůstává v populaci. Pozdější algoritmy generují například z 10 rodičů 100 potomků, potom seřadí rodiče i potomky podle síly (fitness) a vyberou deset nejlepších do další generace. Je možná i strategie “vymřeni” rodičů, kdy se nejlepší jedinci vybírají pouze z potomků.

3. Umělý život (Artificial Life)

1) výběr potomků

Evoluční strategie – u původní implementace lepší potomek jednoduše vytěsňuje horšího rodiče, který je odstraněn z populace. Pokud potomek není lepší než rodič, pak rodič zůstává v populaci. Pozdější algoritmy generují například z 10 rodičů 100 potomků, potom seřadí rodiče i potomky podle síly (fitness) a vyberou deset nejlepších do další generace. Je možná i strategie “vymřeni” rodičů, kdy se nejlepší jedinci vybírají pouze z potomků.

Evoluční programování – i horší potomci mají ještě šanci, pokud jsou vybráni do slabší skupiny. Skupinky jsou náhodně vybrány z nových (tj. mutovaných) jedinců i z rodičů. Ve skupinkách pak dochází k “turnaji”, potomci jsou seřazeni podle síly a ti nejlepší jsou vybráni do další populace. Počet jedinců v populaci nemusí být konstantní a jedinec může produkovat i více potomků.

3. Umělý život (Artificial Life)

2) křížení

Evoluční strategie napodobuje vývoj jedinců v druhu, a tudíž i jejich **křížení**. Vzhledem k tomu, že jedinci představují zakódované řešení problému, tj. směs reálných čísel, celých čísel a dalších parametrů, je křížení **specificky definováno pro každý typ problému**. Předpokládejme, že máme vybrány dva jedince charakterizované dvěma vektory reálných čísel. Diskrétní křížení vytváří nového jedince, jehož vektor je tvořen hodnotami převzatými buď z jednoho nebo z druhého vektoru, přičemž výběr hodnot z vektorů je náhodný.

Křížení "průměrem" vytváří jedince, jehož jednotlivé hodnoty vektoru jsou vždy průměrem odpovídajících složek vektorů rodičovských jedinců.

Ovšem – **evoluční programování typicky křížení nepoužívá**; napodobuje totiž vývoj druhů, které mezi sebou soutěží, a mezidruhové křížení většinou nenastává.

3. Umělý život (Artificial Life)

Poznámka: Z pohledu genetických algoritmů má evoluční programování výhodu v pseudonáhodném výběru jedinců, čímž může lehčeji překonat bariéry falešného lokálního minima. Evoluční strategie má zase výhodu v existenci křížení, kdy sdílení informace pomůže rychleji dospět k optimu.

křížení "průměrem"								
-3,91	8,84	-3,98	-2,51	-5,76	-1,40	-3,33	8,26	rodič 1
6,91	8,46	1,33	-5,66	6,66	-0,75	9,13	-0,40	rodič 2
1,50	8,65	-2,67	-4,09	0,45	-1,08	2,90	3,93	potomek
								$= \frac{\text{rodič 1} + \text{rodič 2}}{2}$
diskrétní křížení								
-3,91	8,84	-3,98	-2,51	-5,76	-1,40	-3,33	8,26	rodič 1
↓			↓		↓	↓		
-3,91	8,46	1,33	-2,51	6,66	-1,40	-3,33	-0,40	
	↑	↑		↑			↑	
6,91	8,46	1,33	-5,66	6,66	-0,75	9,13	-0,40	rodič 2

Příklad: Dva z mnoha druhů křížení používaných evolučními strategiemi

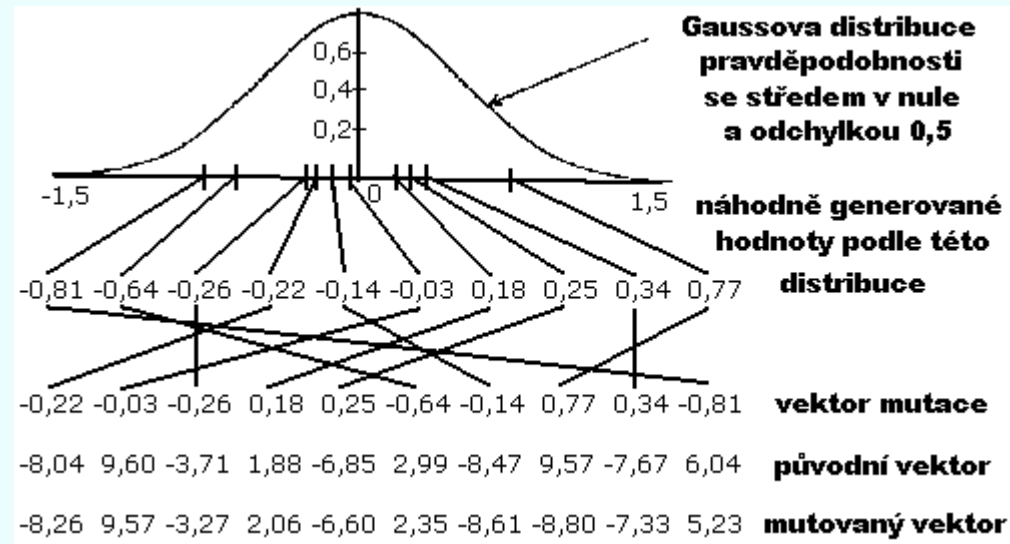
3. Umělý život (Artificial Life)

3) mutace

Mutace je prováděna u každého potomka (nejen u několika málo náhodně vybraných potomků, jako je tomu u genetických algoritmů). Spočívá v přičtení náhodného čísla s Gaussovým rozdělením k reálné hodnotě proměnné.

Uvědomíme-li si, že účelem mutace je vlastně drobná odchylka u jedince, vzniklá přičtením náhodného malého čísla blízkého nule k původní hodnotě, dosáhneme odchylky jistěji než překlopením bitu u proměnné – překlopením bitu na začátku kódu proměnné, jak se může stát u genetických algoritmů, bychom mohli získat mnohonásobnou reálnou hodnotu v porovnání s původní hodnotou. Střední Gaussova odchylka nám sice může dát velké číslo, avšak s mnohem menší pravděpodobností.

3. Umělý život (Artificial Life)



Příklad: Zobrazení Gaussova normálního rozdělení hodnot charakterizujících potomky (se střední směrodatnou odchylkou rovnou 0.5) a deseti náhodně vygenerovaných hodnot příslušejících potomkům, odpovídajících tomuto rozdělení. Hodnoty tvoří po patřičném promíchání vektor mutace – odpovídající hodnoty vektoru jsou přičteny k vektoru “rodiče” a spoluvytvářejí tak vektor “potomka”, nepatrně odlišného od rodiče.

3. Umělý život (Artificial Life)

Jeden možný algoritmus evoluční strategie:

Evoluční strategie:

```
1       $x :=$  náhodně generovaný vektor reálných proměnných;  
2       $t := 0$ ;  $\sigma := \sigma_{ini}$ ;  $x^* := x$ ;  
3      WHILE  $t < t_{max}$  DO  
4      BEGIN  $i := 0$ ;  $k := 0$ ;  
5          WHILE  $i < i_{max}$  DO  
6              BEGIN  $i := i + 1$ ;  $x' := x + r(0, \sigma)$ ;  
7                  IF  $f(x') < f(x)$  THEN  
8                      BEGIN  $k := k + 1$ ;  $x := x'$ ;  
9                          IF  $f(x) < f(x^*)$  THEN  $x^* := x$ ;  
10                     END;  
11                 END;  
12                 IF  $k/i_{max} < 0,2$  THEN  $\sigma := c_d \cdot \sigma$  ELSE IF  $k/i_{max} > 0,2$  THEN  $\sigma := c_i \cdot \sigma$ ;  
13      END;
```

Metoda simulovaného žíhání

- fyzikální interpretace

Při postupném žíhání nejprve zahřejeme těleso na vysokou teplotu. Tím se jeho atomům umožní překonávat lokální energetické hladiny a tak se dostat do rovnovážných poloh. Postupné snižování teploty pak má za následek, že se rovnovážné polohy atomů fixují, takže při konečné teplotě žíhání (podstatně nižší než byla počáteční) jsou všechny atomy v rovnovážných polohách a těleso neobsahuje žádné vnitřní defekty ani pnutí.

Metoda simulovaného žíhání

- fyzikální interpretace

Při postupném žíhání nejprve zahřejeme těleso na vysokou teplotu. Tím se jeho atomům umožní překonávat lokální energetické hladiny a tak se dostat do rovnovážných poloh. Postupné snižování teploty pak má za následek, že se rovnovážné polohy atomů fixují, takže při konečné teplotě žíhání (podstatně nižší než byla počáteční) jsou všechny atomy v rovnovážných polohách a těleso neobsahuje žádné vnitřní defekty ani pnutí.

- matematická interpretace

Počátkem osmdesátých let dostali Kirkpatrick, Gelatt, Vecchi a Černý nápad, že problém hledání globálního minima funkce může být realizovatelný podobným způsobem jako žíhání tuhého tělesa a nazvali ho **metodou simulovaného žíhání**.

Algoritmus simulovaného žíhání

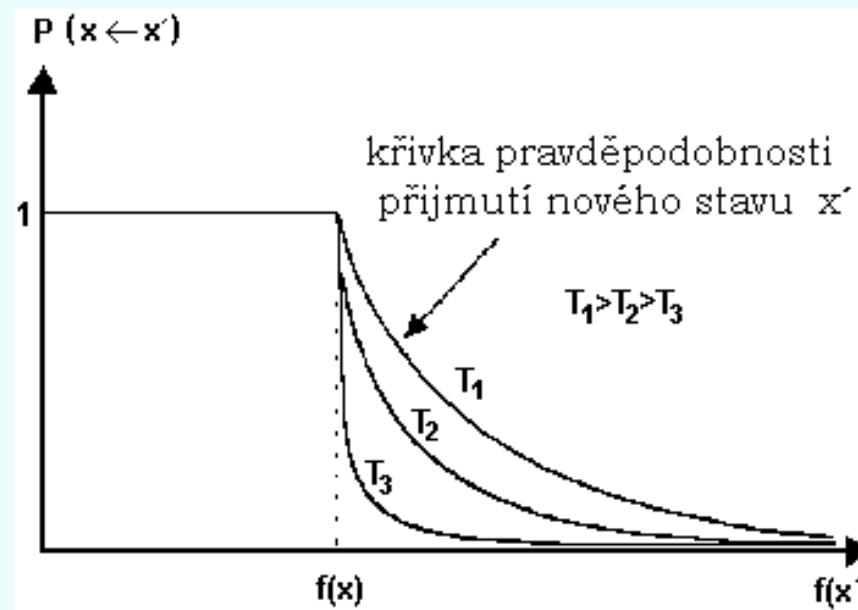
Uvažujme hypotetický systém popsaný n -rozměrným stavem $x = (x_1, x_2, \dots, x_n)$, kde x_i patří do nějakého intervalu. Stavy tohoto systému jsou ohodnoceny funkcí f , která přiřadí každému x hodnotu $y=f(x)$. Naší úlohou je najít takový stav x_{\min} , ve kterém nabývá funkce f globálního minima. Necht' x je přípustný stav systému, ten přejde do nového stavu x' , formálně $x' = O_{\text{pert}}(x)$, kde O_{pert} je "operátor" poruchy – perturbace. Otázka, zda bude nový stav akceptován do dalšího procesu, se řeší na základě tzv. Metropolisova kritéria, které určuje pravděpodobnost nahrazení starého stavu novým.

3. Umělý život (Artificial Life)

Celý tento vztah lze zapsat:

$$P(x \leftarrow x') = \begin{cases} 1 & \text{pro } f(x') \leq f(x) \\ \exp(-|f(x') - f(x)|/T) & \text{pro } f(x') > f(x) \end{cases},$$

kde T je parametr interpretovaný jako teplota.



3. Umělý život (Artificial Life)

V případě, že nový stav x' má menší nebo stejnou funkční hodnotu jako původní stav x , potom provedeme záměnu stavu x stavem x' . V opačném případě je nový stav x' akceptován s pravděpodobností $0 < P(x \leftarrow x') < 1$. Necht' random je náhodné číslo z intervalu $(0,1)$, potom nový stav je akceptován, pokud $\text{random} < P(x \leftarrow x')$, jinak se proces opakuje s původním stavem x .

Hodnota parametru T podstatně ovlivňuje hodnotu pravděpodobnosti $P(x \leftarrow x')$ pro případ $f(x') > f(x)$. Pro velké hodnoty T je tato pravděpodobnost blízká jedné (tj. přijmou se téměř všechny nové stavy), ale blíží-li se T k nule, potom pravděpodobnost přijmutí je velmi nízká, téměř nulová.

Při vhodném nastavení počáteční "teploty" T_{\max} (větší hodnota) metoda simulovaného žíhání prohledává nejprve prostor řešení D silně stochasticky, přijímá i stavy s horším ohodnocením než má současné řešení. Tato vlastnost simulovaného žíhání je velmi důležitým rysem této metody, protože poskytuje možnost snadno se dostat z lokálního extrému a prohledat i jiné oblasti v prostoru řešení. Tahle možnost se pak postupně snižuje úměrně se zmenšováním hodnoty T . Pro malé T připustí Metropolisovo kritérium akceptování už jen lepšího řešení než je současné.

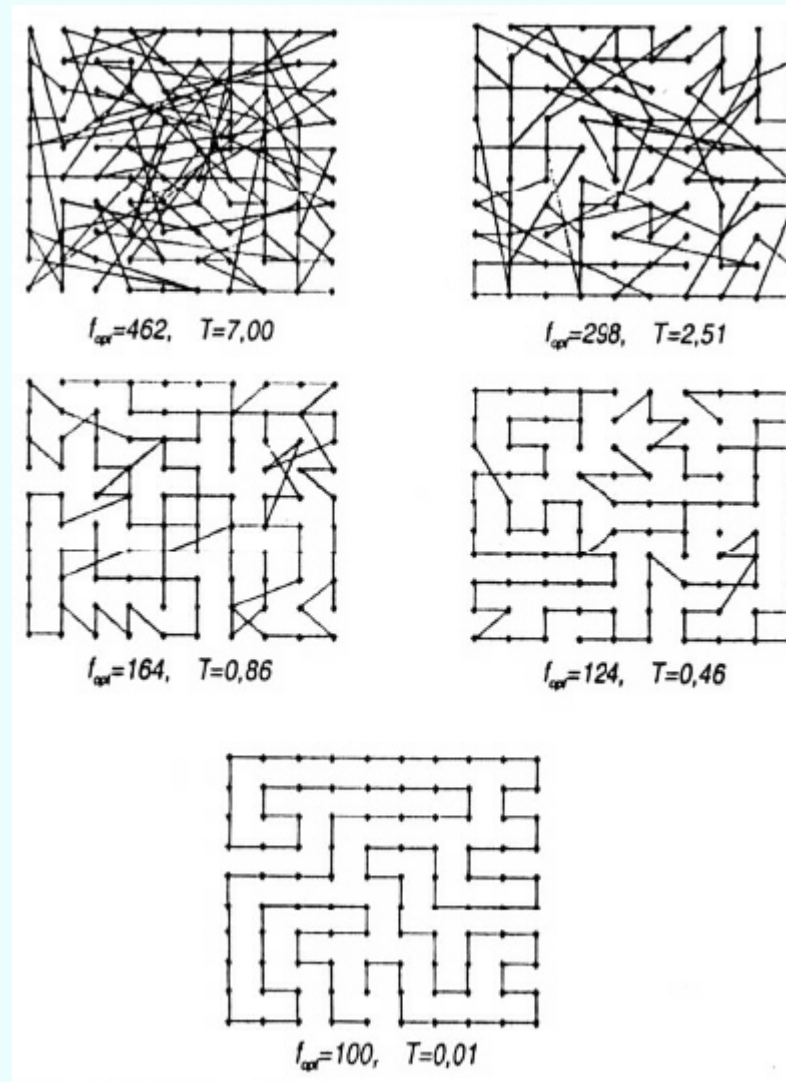
3. Umělý život (Artificial Life)

Praktické využití simulovaného žíhání

Metoda simulovaného žíhání byla testována už v době svého vzniku na známém problému **obchodního cestujícího**, kdy hledáme nejkratší uzavřenou cestu mezi N místy. Tento problém patří mezi NP-úplné problémy; čas nalezení řešení roste při zvětšování dimenze přinejlepším exponenciálně. Simulované žíhání je schopno nalézt optimální řešení i v případě takto složité úlohy (viz obrázek na následující folii).

Ukazuje se, že simulované žíhání poskytuje velmi efektivní algoritmus k řešení kombinatoriálních úloh, které jsou NP-úplné, přičemž získaná řešení jsou buď totožná nebo velmi blízká optimálním řešením.

3. Umělý život (Artificial Life)



3. Umělý život (Artificial Life)

Algoritmus "simulovaného žíhání" zapsaný jako procedura v PseudoPascalu:

```
procedure Simulated_Annealing ( $\mathbf{x}_{min}$ ,  $T_{min}$ ,  $T_{max}$ ,  $k_{max}$ ,  $\alpha$ );
  procedure Metropolis_Algorithm ( $\mathbf{x}_{ini}$ ,  $\mathbf{x}_{out}$ ,  $k_{max}$ ,  $T$ );
    begin  $k := 0$ ;  $\mathbf{x} := \mathbf{x}_{ini}$ ;
    while  $k < k_{max}$  do
      begin  $k := k + 1$ ;
         $\mathbf{x}' := \mathcal{O}_{pert}(\mathbf{x})$ ;
         $Pr := \min(1, \exp(-(f(\mathbf{x}') - f(\mathbf{x}))/T))$ ;
        if random < Pr then  $\mathbf{x} := \mathbf{x}'$ ;
      end;
       $\mathbf{x}_{out} := \mathbf{x}$ ;
    end;
  begin  $\mathbf{x}_{ini} :=$  náhodně vygenerovaný stav;
     $T := T_{max}$ ;
    while  $T > T_{min}$  do
      begin
        Metropolis_Algorithm ( $\mathbf{x}_{ini}$ ,  $\mathbf{x}_{out}$ ,  $k_{max}$ ,  $T$ );
         $\mathbf{x}_{ini} := \mathbf{x}_{out}$ ;
         $T := \alpha * T$ ;
      end;
       $\mathbf{x}_{min} := \mathbf{x}_{out}$ ;
    end;
end;
```

3. Umělý život (Artificial Life)

Metoda zakázaného prohledávání (Tabu Search)

Jednou z mnoha optimalizačních algoritmů a heuristik je metoda **zakázaného prohledávání**, na které zajímavé, že byla navržena teprve před několika lety a přitom je tak jednoduchá a efektivní, že se okamžitě zařadila mezi nejlepší algoritmy používané pro optimalizaci v operačním výzkumu. Navrhl ji koncem osmdesátých let prof. Fred Glover z University of Colorado a sám ji také nazval **tabu search**.

Metoda dokáže najít **přijatelné optimum** za velmi krátký čas, pokud je přijatelných minim ve sledované funkci více. Při prohledávání funkcí s mnoha minimy, ale jen málo přijatelnými minimy, má tato metoda trochu problémy. Její základní myšlenka totiž spočívá v logickém vylepšení algoritmu "**slepého horolezce**" (**hill climbing algorithm**). Jedná se v podstatě o variantu gradientní metody "bez gradientu", protože se směr nejprudšího spádu určí prohledáváním okolí.

3. Umělý život (Artificial Life)

Algoritmus však trpí základní nectností gradientových metod, tj. nejspíše skončí v lokálním extrému a nedosáhne globálního minima.

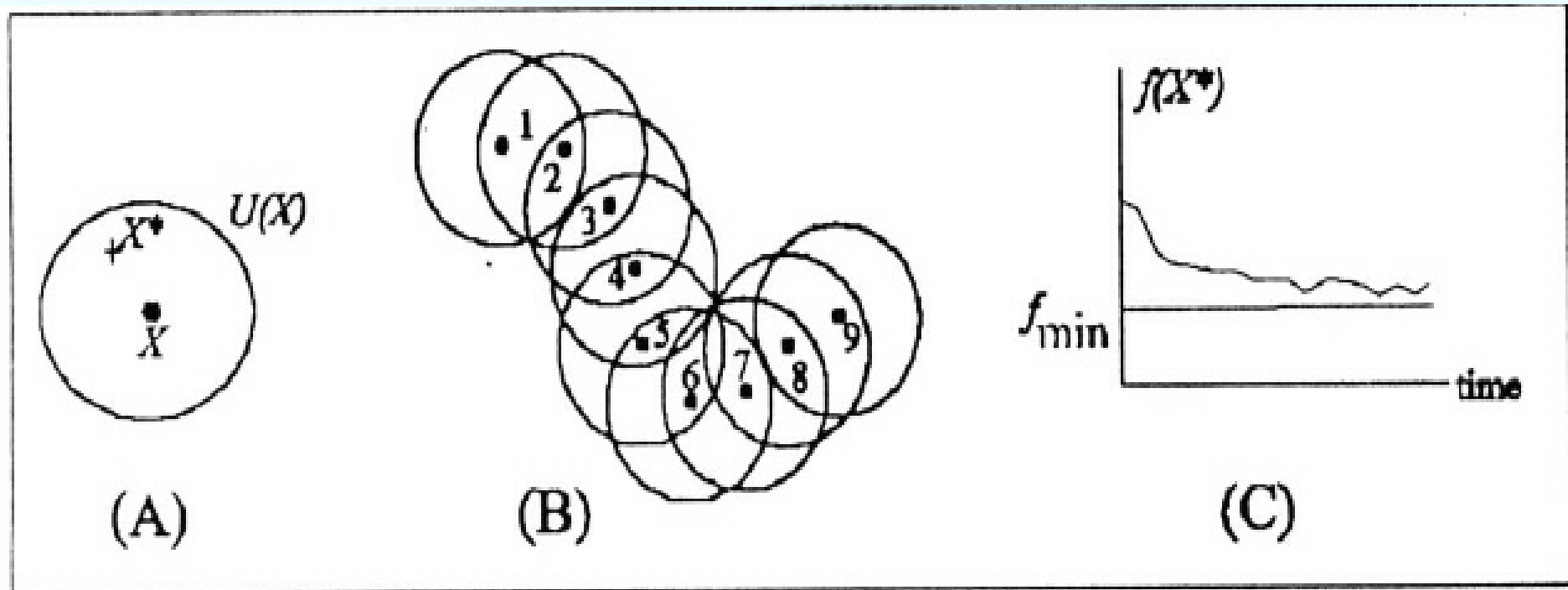
3. Umělý život (Artificial Life)

Algoritmus však trpí základní nectností gradientových metod, tj. nejspíše skončí v lokálním extrému a nedosáhne globálního minima.

Vychází z náhodně navrženého řešení:

Pro aktuální navržené řešení se generuje pomocí konečného souboru transformací **určité okolí** a **vybere se** z tohoto okolí **nejlepší minimum**. Získané lokální řešení se použije jako "**střed**" **nového okolí**, ve kterém se lokální minimalizace opakuje; tento proces projde předepsaným počtem opakování. V průběhu celé historie algoritmu se zaznamenává nejlepší řešení, které slouží jako výsledné minimum. Aby nedošlo k zacyklení, spustí se algoritmus několikrát s náhodně vygenerovanými počátečními stavy a poté se vybere nejlepší výsledek.

3. Umělý život (Artificial Life)



(A) Znáznornění okolí $U(X)$ momentálního řešení X a nejlepšího řešení z tohoto okolí X^* řešení, (B) Znáznornění iteračního postupu horolezeckého algoritmu, (C) Hodnoty momentálního nejlepšího řešení v závislosti na čase (iteraci). Ke konci graf začne pravidelně oscilovat.

3. Umělý život (Artificial Life)

Do algoritmu je zavedena tzv. **krátkodobá paměť**, která si pro určitý krátký interval předcházející historie algoritmu **pamatuje inverzní transformace k lokálně optimálním transformacím řešení**, použitým k získání nových "středů" pro jednotlivé iterace. Tyto inverzní transformace jsou **zakázány** (tabu) při tvorbě nového okolí pro dané aktuální řešení. Tímto jednoduchým způsobem je možné podstatně omezit výskyt zacyklení při pádu do lokálního minima.

Takováto **krátkodobá paměť** je realizována **zakázaným seznamem T (tabu list)** typu LIFO (zásobník), který je sestaven a obnovován v průběhu algoritmu. Jestliže tedy nějaká **transformace patří do zakázaného seznamu**, pak se **nemůže použít k lokální minimalizaci**. Numerické zkušenosti s metodou zakázaného prohledávání ukazují, že **velikost (délka) zakázaného seznamu je velmi důležitým parametrem** ovlivňujícím schopnost vymanit se z lokálního extrému. Příliš malá délka nemusí zabránit zacyklení a příliš velká může způsobit přeskočení nadějných lokálních minim.

3. Umělý život (Artificial Life)

Implementace metody tabu search:

```
Horolezecký algoritmus :  
X := náhodně vygenerovaný vektor;  
time := 0; fmin := ∞;  
WHILE time < timemax DO  
BEGIN  
  time := time + 1;  
  f(X*) := min f(X')  
           X' ∈ U(X)  
  IF f(X*) < fmin THEN  
  BEGIN  
    fmin := f(X*);  
    Xmin := X*;  
  END;  
  X := X*;  
END;
```

```
Zakázané prohledávání :  
X := náhodně vygenerovaný vektor;  
time := 0; fmin := ∞; T := ∅;  
WHILE time < timemax DO  
BEGIN  
  time := time + 1;  
  floc min := ∞;  
  FOR t ∈ S DO  
  BEGIN  
    X' := t X;  
    IF (t ∈ T and f(X') < floc min) OR f(X') < fmin  
    THEN BEGIN  
      X* := X';  
      t* := t;  
      floc min := f(X');  
    END;  
  END;  
  IF floc min < fmin THEN  
  BEGIN  
    fmin := floc min;  
    Xmin := X*;  
  END;  
  X := X*;  
  IF |T| < k THEN T := T ∪ {t*-1}  
  ELSE T := (T ∪ {t*-1}) \ {t};  
END;
```

3. Umělý život (Artificial Life)

Využití metody tabu search

Mezi problémy úspěšně řešitelné metodou zakázaného prohledávání patří problém obchodního cestujícího, problémy rozvrhu práce pro stroje v továrnách, návrh komunikačních sítí, strategické rozmístění skladů a logistické problémy v ekonomice a mnoho dalších.

3. Umělý život (Artificial Life)

Využití metody tabu search

Mezi problémy úspěšně řešitelné metodou zakázaného prohledávání patří problém obchodního cestujícího, problémy rozvrhu práce pro stroje v továrnách, návrh komunikačních sítí, strategické rozmístění skladů a logistické problémy v ekonomice a mnoho dalších.

Technika zakázaného prohledávání může být použita jak v klasickém spojení s horolezeckým algoritmem, tak i v kombinaci s jinými algoritmy, např. se simulovaným žíháním nebo genetickými algoritmy. U těchto metod však není natolik efektivní; tyto metody neprohledávají celé okolí aktuálního řešení a tak pravděpodobnost účinnosti metody zakázaného seznamu je značně malá.

3. Umělý život (Artificial Life)

Umělý život

Umělý život (Artificial Life) je rychle se rozvíjející oblast vědeckého výzkumu spojující **biologii a informatiku**. Snaží se včlenit procesy podobné životu do počítačových programů. Hlavním rysem tohoto oboru je **formulování základních pravidel chování jednotlivých "organismů"** a jejich replikace, přičemž cílem je dosáhnout vývoje organismů a jejich vzájemných vztahů.

3. Umělý život (Artificial Life)

Umělý život

Umělý život (Artificial Life) je rychle se rozvíjející oblast vědeckého výzkumu spojující **biologii a informatiku**. Snaží se včlenit procesy podobné životu do počítačových programů. Hlavním rysem tohoto oboru je **formulování základních pravidel chování jednotlivých "organismů"** a jejich replikace, přičemž cílem je dosáhnout vývoje organismů a jejich **vzájemných vztahů**.

Ve vztahu ke klasické umělé inteligenci je umělý život orientován opačným směrem. Zatímco se **umělá inteligence** snaží o přístup shora dolů, tj. **program musí být v první řadě inteligentní** a je na programátorovi, jak toho dosáhnout, **umělý život se snaží vytvořit jen základní pravidla**, která budou **tak vhodně zvolena**, že se složitější chování vytvoří po čase samo.

3. Umělý život (Artificial Life)

Umělá inteligence a umělý život mají k sobě nejbližší v typu programů nazvaných **autonomní agent**. To je program, který obsahuje nějaký druh vnímání okolního prostředí a návazně ovlivňuje vlastní stav, stav okolních agentů nebo prostředí samotné. V Americe už se autonomní agenti využívají na praktické úlohy: řízení letového provozu nebo provozu továrny, kdy každý stroj má svého agenta starajícího se o dodávky energie, surovin a odvoz hotových výrobků. Toto využití je však stále ještě velmi nákladné. Za autonomního agenta může být považován i počítačový virus, ale v praxi se od agenta liší svojí jednoduchostí a tím, že má jen jeden hlavní cíl – rozšiřovat se.

3. Umělý život (Artificial Life)

Umělá inteligence a umělý život mají k sobě nejbliže v typu programů nazvaných **autonomní agent**. To je program, který obsahuje nějaký druh vnímání okolního prostředí a návazně ovlivňuje vlastní stav, stav okolních agentů nebo prostředí samotné. V Americe už se autonomní agenti využívají na praktické úlohy: řízení letového provozu nebo provozu továrny, kdy každý stroj má svého agenta starajícího se o dodávky energie, surovin a odvoz hotových výrobků. Toto využití je však stále ještě velmi nákladné. Za autonomního agenta může být považován i počítačový virus, ale v praxi se od agenta liší svojí jednoduchostí a tím, že má jen jeden hlavní cíl – rozšiřovat se.

Mnoho aplikací umělého života vybavuje své organismy **umělou neuronovou sítí**, která slouží jako umělý mozek schopný natrénovat se z příkladů s předem zadaným výsledkem. Neuronová síť je tvořena vstupními neurony, napojenými na informace získávané z okolního prostředí skrytými vrstvami neuronů, provádějícími většinu výpočtů, a výstupními neurony, které určují další akci organismu – agenta.

Závěrečné porovnání jednotlivých algoritmů

Z pohledu genetických algoritmů má evoluční programování výhodu v pseudonáhodném výběru jedinců, protože tím může snadněji překonat falešné lokální minimum. Evoluční strategie má zase výhodu v existenci křížení, kdy sdílení informace pomůže rychleji dospět k optimu.

Závěrečné porovnání jednotlivých algoritmů

Z pohledu genetických algoritmů má evoluční programování výhodu v pseudonáhodném výběru jedinců, protože tím může snadněji překonat falešné lokální minimum. Evoluční strategie má zase výhodu v existenci křížení, kdy sdílení informace pomůže rychleji dospět k optimu.

Poznámka:

I když jsou si jak genetické algoritmy, tak i evoluční programování a evoluční strategie natolik podobné, že je téměř nemá smysl od sebe striktně rozlišovat, vzhledem k historickému vývoji si různé vědecké komunity, fandící jednotlivým metodám, udržují od sebe nesmyslně velký odstup.

3. Umělý život (Artificial Life)

Genetický algoritmus

```
t := 0;
P0 := {náhodně vygenerovaná populace chromozomů, většinou binární reprezentace};
Ohodnot' každý chromozom z populace P0 silou;
while t < tmax do
begin
  t := t + 1;
  Q := {nové chromozomy vzniklé křížením a případně následnou mutací kvazináhodně
  vybraných chromozomů z Pt-1 s největší silou};
  Ohodnot' každý chromozom z Q silou;
  R := {kvazináhodně vybrané chromozomy z Pt-1 s nejmenší silou};
  Pt := (Pt-1 \ R) ∪ Q;
end;
```

P_t v algoritmu označuje populaci chromozomů v čase t , t_{\max} počet opakování cyklu,

Q je subpopulace potomků - chromozomů,

R označuje subpopulaci náhodně vybraných chromozomů s nejmenší silou

$P_t := (P_{t-1} \setminus R) \cup Q$; tvorba nové populace P_t z populace P_{t-1} tak, že nové chromozomy vytěsňují část původních chromozomů

Počty chromozomů subpopulací Q a R jsou ohraničeny podmínkami $|Q| \ll |P_t|$ a $|Q| = |R|$

3. Umělý život (Artificial Life)

Evoluční strategie

```
t := 0;  
P0 := {náhodně vygenerovaná populace chromozomů, reálná reprezentace};  
Ohodnot' každý chromozom z populace P0 silou;  
while t < tmax do  
begin  
  t := t + 1;  
  Q := {nové chromozomy vzniklé křížením a vždy použitou mutací náhodně  
  vybraných chromozomů z Pt-1};  
  Ohodnot' každý chromozom z Q silou;  
  Pt := {nejlepší chromozomy z Q; může být použita i strategie výběru nejlepších  
  chromozomů z Pt-1 ∪ Q};  
end;
```

Počet chromozomů subpopulace Q je ohraničen podmínkou $|Q| \geq |P_t|$

3. Umělý život (Artificial Life)

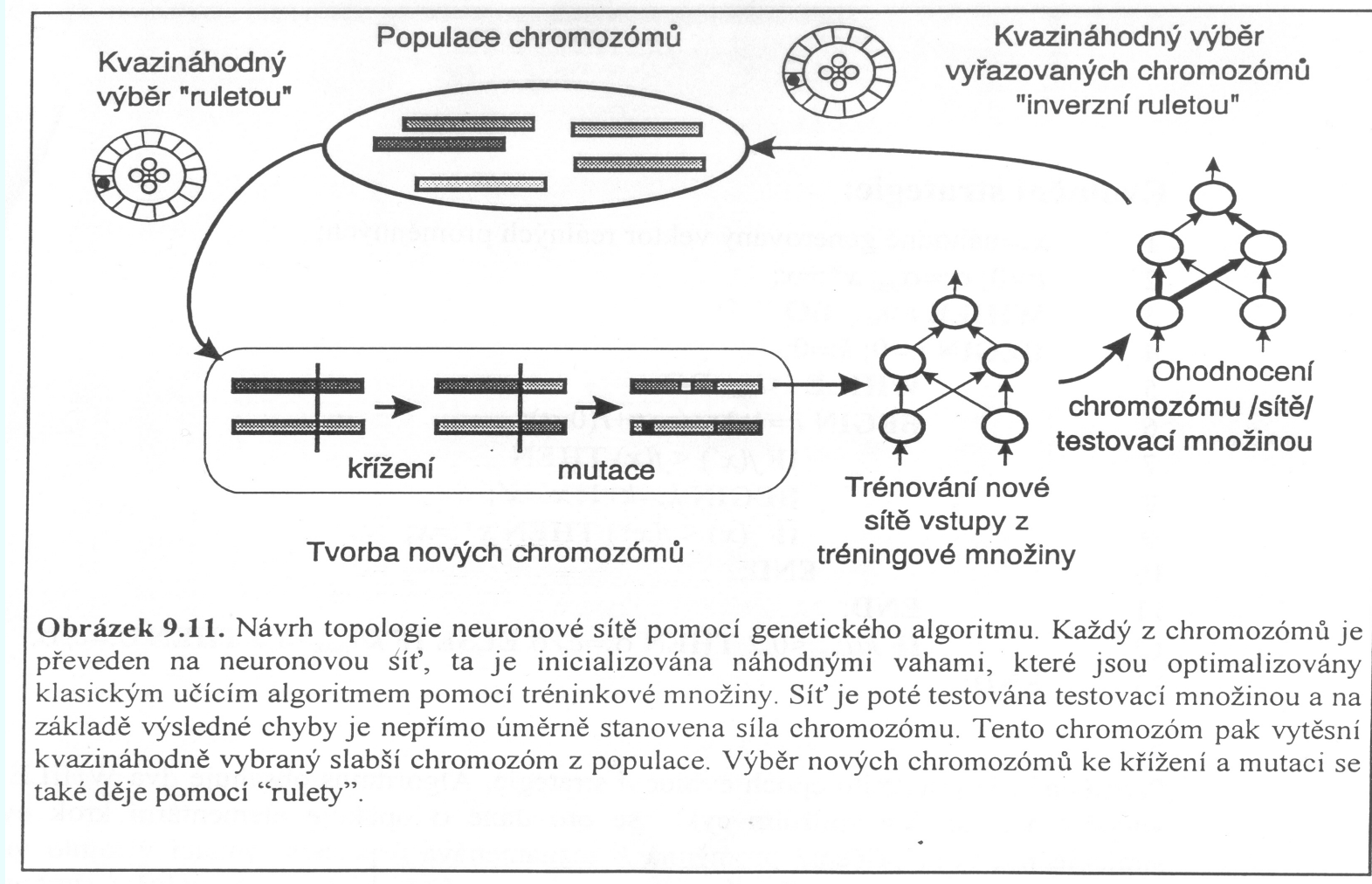
Evoluční programování

```
t := 0;  
P0 := {náhodně vygenerovaná populace chromozomů, reálná reprezentace};  
Ohodnot' každý chromozom z populace P0 silou;  
while t < tmax do  
begin  
  t := t + 1;  
  Q := {nové chromozomy vzniklé mutací ze všech chromozomů z Pt-1};  
  Ohodnot' každý chromozom z Q silou;  
  Pt := {chromozomy z Q jsou porovnány "turnajem" s náhodně vybranými q chromozomy  
  z Pt-1 u Q (chromozomy mohou tedy skončit v turnaji na 1 až q+1 místě), vybírají se  
  nejlepší chromozomy podle výsledků dosažených v turnaji};  
end;
```

Počet chromozomů subpopulace Q je ohraničen podmínkou $|Q| = |P_t|$

3. Umělý život (Artificial Life)

Závěrem příklad: návrh topologie neuronové sítě genetickým algoritmem:



Obrázek 9.11. Návrh topologie neuronové sítě pomocí genetického algoritmu. Každý z chromozómů je převeden na neuronovou síť, ta je inicializována náhodnými vahami, které jsou optimalizovány klasickým učícím algoritmem pomocí trénigové množiny. Síť je poté testována testovací množinou a na základě výsledné chyby je nepřímo úměrně stanovena síla chromozómu. Tento chromozóm pak vytěsňuje kvazináhodně vybraný slabší chromozóm z populace. Výběr nových chromozómů ke křížení a mutaci se také děje pomocí "rulety".