

4. Metody řešení optimalizačních úloh algoritmy inspirovanými přírodou

Mezi nejznámější algoritmy inspirovanými přírodou patří:

- Genetické algoritmy
(GA – Genetic Algorithms)
- Optimalizace mravenčí kolonií
(ACO - Ant Colony Optimization)
- Optimalizace hejnem částic
(PSO - Particle Swarm Optimization)

Genetické algoritmy - GA

Biologická inspirace:

Každý živočišný druh má ve svých buňkách charakteristický počet a strukturu útvarů složených z DNA (kyseliny deoxyribonukleové) a proteinů (organických makromolekulárních látek), kterým se říká chromozomy.

Každý chromozom obsahuje tisíce až miliony genů, přičemž každý gen může nabývat různých forem, které se nazývají alely.

Soubor všech různých alel se nazývá genotyp a určuje soubor znaků jedince, tzv. fenotyp, který se pak projeví navenek (krevní skupina, barva očí, atd.).

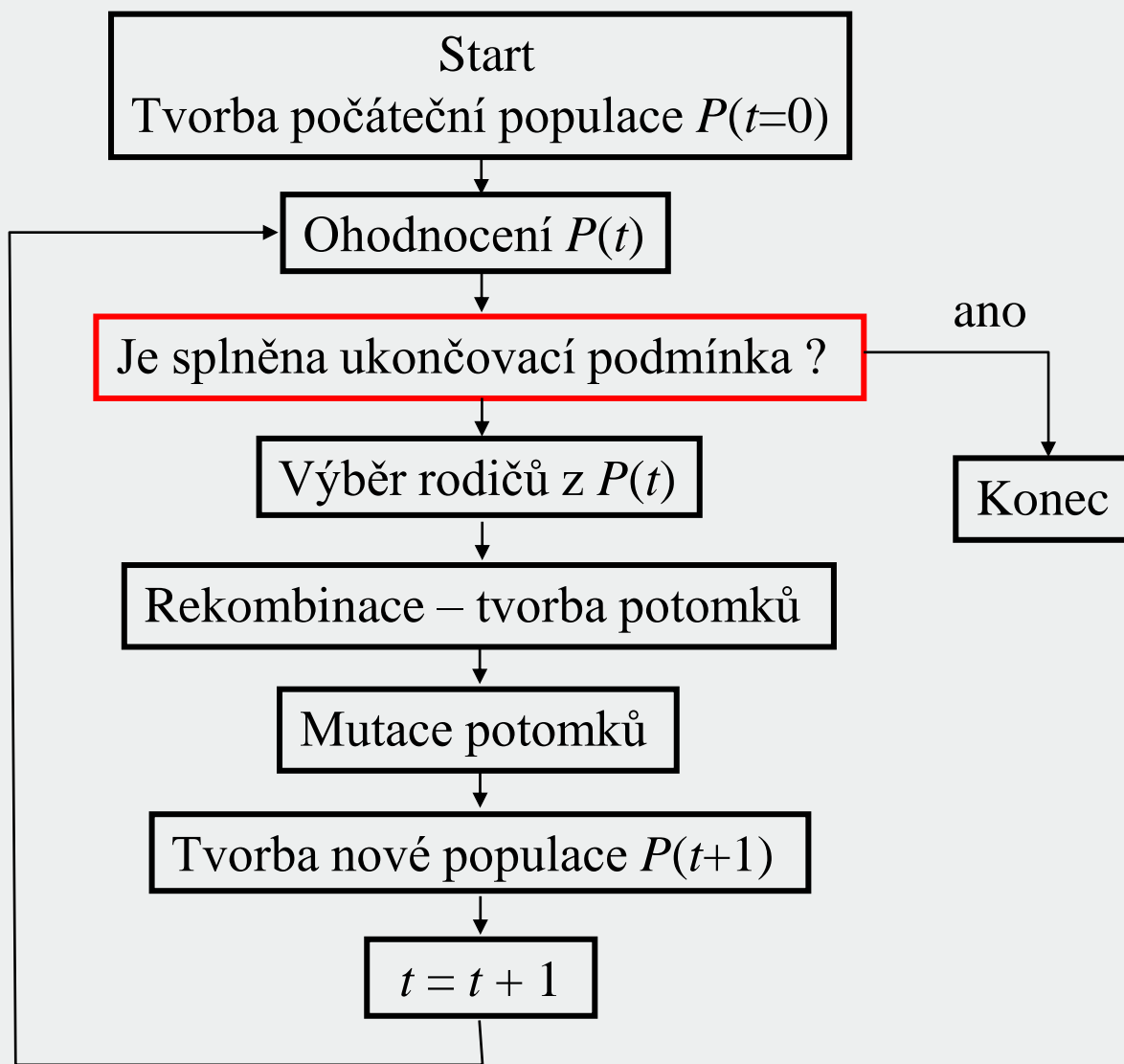
Buňky člověka mají 46 chromozomů, které tvoří 23 párů. V každém z těchto párů je jeden chromozom mateřského a druhý otcovského původu. Výjimku tvoří pohlavní buňky (ve vajíčkách a spermiích), které obsahují pouze jeden náhodně vybraný chromozom příslušného páru (princip dědičnosti).

Po oplodnění vytvoří odpovídající si chromozomy pár, a proto každý potomek má ve svých buňkách opět 23 párů chromozomů. Přitom není důležité, který gen je v těchto párech převzat od matky a který od otce.

Příležitostně, vlivem neobvyklých fyzikálních, chemických nebo biologických efektů, např. kosmického záření, může být některý gen náhodně změněn – tato změna se nazývá mutací.

Jedinci, kteří se lépe adaptují na okolní prostředí, mají větší šanci na přežití a tím na plození dalších potomků – Darwinova teorie vývoje, evoluce.

Programový model genetického vývoje – genetický algoritmus.



Základní pojmy GA

- 1) Jedinec (chromozom)
- 2) Populace
- 3) Hodnocení jedinců, kvalita populace, ukončení výpočtu
- 4) Reprodukce
 - Výběr rodičů
 - Křížení (rekombinace – tvorba potomků)
 - Mutace
 - Tvorba nové populace

Jedinec (chromozom)

Jedinci v GA jsou obvykle reprezentováni řetězci znaků, ale obecně jsou možné i jiné jejich reprezentace, například grafy, matice atp.

Řetězce znaků připomínají chromozomy, a proto se také tak nazývají. Délky řetězců jsou pak závislé na konkrétních řešených úlohách.

Kombinace jednotlivých znaků/genů v chromozomu nese informaci, která se nazývá řešením dané optimalizační úlohy – je zřejmé, že toto „řešení“ může mít ke skutečnému (tj. k optimálnímu) řešení velmi daleko.

Příklady chromozomů:

- Pokud se hledá maximum nějaké funkce v intervalu $\langle a, b \rangle$, může být chromozomem n -bitový binární řetězec. Pro převod hodnoty kladného binárního čísla x_{bin} reprezentovaného tímto řetězcem na číslo x reprezentované chromozomem lze použít například vztah:

$$x = a + (x_{bin}) / (2^n - 1) * (b - a)$$

Například pro interval $\langle -3, 5 \rangle$ představuje osmibitový chromozom (0,0,0,1,1,0,1,1) reálné číslo $x \cong -2.15294$.

- Pokud se řeší problém obchodního cestujícího, pak chromozomem je řetězec znaků. Například pro sedm míst A, B, C, D, E, F a G představuje chromozom (A,D,E,C,G,B,F) cestu

$$A \rightarrow D \rightarrow E \rightarrow C \rightarrow G \rightarrow B \rightarrow F \rightarrow A$$

Populace

Populaci tvoří předem daný počet chromozomů. Tento počet je závislý na řešené úloze, obvykle jde o desítky až tisíce jedinců.

Jedinci počáteční/výchozí populace se generují náhodně, může se však samozřejmě využít libovolná dostupná heuristika, která umožní generování jedinců s počátečními hodnotami (řešeními) blízkými optimálnímu řešení.

Hodnocení jedinců, kvalita populace, ukončení výpočtu

Hodnocení jedince/chromozomu spočívá ve vyhodnocení, jak se řešení představované tímto jedincem liší od správného/optimálního řešení dané úlohy (fitness of solution).

Funkce, jejíž hodnota udává kvalitu řešení chromozomu se nazývá fitness funkce. Hodnota fitness funkce se musí zvyšovat s kvalitou řešení, tj. musí být tím vyšší, čím je řešení představované ohodnocovaným chromozomem bližší správnému řešení.

Fitness funkce mají výrazný vliv jak na kvalitu řešení, tak i na délku výpočtu, a proto jejich návrhu je nutné věnovat velkou pozornost.

Kvalita populace je dána ohodnocením populace – průměrnou hodnotou hodnot fitness funkcí všech jedinců populace, neboli hodnotou fitness funkce průměrného jedince populace.

Pokud některý jedinec představuje správné/optimální řešení (což však u některých úloh nelze zjistit!), výpočet samozřejmě okamžitě končí.

Jinak se pro ukončení výpočtu používá dvou přístupů:

- kvalita populace se delší dobu nezvyšuje,
- bylo dosaženo předem stanoveného maximálního počtu iterací (tj. vytváření nových populací).

Řešení problému v těchto případech představuje nejlépe hodnocený chromozom.

Reprodukce (výběr rodičů, křížení, mutace, tvorba nové populace)

Výběr rodičů se provádí na základě ohodnocení všech jedinců populace fitness funkcí, a to buď podle jejich proporcí, nebo podle jejich pořadí.

Nejjednodušším výběrem je výběr elity, kdy se k rodičovství vybere potřebný počet nejlépe ohodnocených jedinců.

Dalším jednoduchým výběrem je turnajový výběr, kdy z náhodně vybraných jedinců vítězí (tj. vybírá se jako rodič) nejlépe ohodnocený jedinec. Počet „turnajů“ se samozřejmě musí rovnat potřebnému počtu rodičů. V turnajové výběru pak dostávají šanci stát se rodiči i hůře ohodnocení jedinci, protože některé turnaje mohou být díky náhodnému výběru „slaběji“ obsazené.

Křížení produkuje nové jedince z informací obsažených v genech rodičů a spočívá ve vzájemné výměně částí chromozomů. Místa, od kterých změny začínají, se nazývají místy křížení a bývají vybírána náhodně.

Konkrétních přístupů ke křížení je celá řada a závisí mj. i na použité reprezentaci jedinců.

Nejjednodušším křížením je tzv. jednobodové křížení: Stanoví se náhodně místo křížení – pro m genů v chromozomu může toto místo nabývat hodnot 1 až $m - 1$:

		← místo křížení
1. rodič:	a b c d e f g h i j k	
2. rodič:	a b c d e f g h i j k	
1. potomek:	a b c d e f g h i j k	
2. potomek:	a b c d e f g h i j k	

Mutace spočívá ve změně hodnoty náhodně vybraného genu náhodně vybraného potomka. Pravděpodobnost mutace je obvykle velmi nízká, například $P_m = 1/n$, kde n je počet jedinců v populaci.

U binárních hodnot se invertuje hodnota bitu, u reálných hodnot se obvykle k hodnotě genu přičte/odečte malá náhodná reálná hodnota (max. 10%).

Tvorba nové populace

- Generační model: všichni jedinci původní populace jsou v nové populaci nahrazeni potomky.
- Inkrementační model: v nové populaci je nahrazen potomkem jediný jedinec původní populace.
- Modely s překrytím generací: v nové populaci je nahrazena potomky část jedinců původní populace – je zřejmé, že předchozí dva typy modelů jsou extrémními případy tohoto modelu.

Příklad - Hledáme průchod nulovou hodnotou funkce $y = 5 - x$.

Necht':

- chromozomy mají čtyři binární geny (hodnotou je kladné celé číslo x),
- populace obsahuje čtyři chromozomy,
- fitness funkce je definována takto: $f = 8 - \text{abs}(y)$,
- počáteční populace: 1100, 0011, 0111, 1001.

i	chromozom $_i$	x_i	y_i	f_i
1	1100	12	-7	1
2	0011	3	2	6
3	0111	7	-2	6
4	1001	9	-4	4
Σ				17
ϕ				4.25

Hodnota ϕ (4.25) udává průměrnou hodnotu fitness funkce v populaci (kvalitu populace).

Použijeme generační model a turnajový výběr rodičů \Rightarrow potřebujeme čtyři rodiče, tj. 2 x 2 turnaje a necht' tyto turnaje mají vždy pouze dva účastníky.

Necht':

- pro první turnaj jsou náhodně vybráni jedinci 2 a 4, vítězem je jedinec 2.
- pro druhý turnaj jsou náhodně vybráni jedinci 1 a 3, vítězem je jedinec 3.
- pro třetí turnaj jsou náhodně vybráni jedinci 2 a 4, vítězem je jedinec 2.
- pro čtvrtý turnaj jsou náhodně vybráni jedinci 1 a 4, vítězem je jedinec 4.

První dvojici rodičů tvoří jedinci 2 a 3, druhou dvojici tvoří jedinci 2 a 4.

Dále necht' pro první pár bylo náhodně zvoleno místo křížení 3 a pro druhý pár místo křížení 1:

První pár:

R2	0 0 1 1
R3	0 1 1 1
P1	0 0 1 1
P2	0 1 1 1

Druhý pár:

R2	0 0 1 1
R4	1 0 0 1
P3	0 0 0 1
P4	1 0 1 1

Nová populace je pak následující:

i	chromozom $_i$	x_i	y_i	f_i
1	0011	3	2	6
2	0111	7	-2	6
3	0001	1	4	4
4	1011	11	-6	2
Σ				18
ϕ				4.5

Nechť v dalších čtyřech turnajích byly k reprodukci vybrány dvojice (3 a 2) a (1 a 2) a necht' místa křížení jsou 2 a 3:

První pár:

R3	0 0 0 1
R2	0 1 1 1
P1	0 0 1 1
P2	0 1 0 1

Druhý pár:

R1	0 0 1 1
R2	0 1 1 1
P3	0 0 1 1
P4	0 1 1 1

Další nová populace je následující:

i	chromozom _{i}	x_i	y_i	f_i	
1	0011	3	2	6	
2	0101	5	0	8	... hledané řešení
3	0011	3	2	6	
4	0111	7	2	6	
Σ				26	
ϕ				6.5	

Na příkladu je vidět postupné zlepšování kvality populace vedoucí nakonec k nalezení hledaného řešení.

Poznamenejme, že uvedený způsob křížení není použitelný obecně, nelze jej například použít pro permutační problémy jako je TSP. Popis jiných metod křížení přesahuje rámec předmětu IZU, zájemci však mohou tento popis nalézt v podkladech předmětu SFC.

Optimalizace mravenčí kolonií - ACO

Algoritmy ACO jsou inspirovány postupem, který používají mravenci při hledání potravy (nejkratší cesty mezi mraveništěm a zdrojem potravy):

- Mravenci nejprve náhodně prohledávají blízké okolí mraveniště.
- Mravenci během svého pohybu vypouští na zem chemickou látku, tzv. feromon.
- Mravenci tuto látku cítí a pravděpodobnost výběru jejich dalších cest je dána aktuálními koncentracemi feromonů na začátcích možných dalších cest.
- Když některý mravenec narazí na zdroj potravy, vyhodnotí její kvalitu i kvantitu a vrací se s částí této potravy do mraveniště.
- Během zpáteční cesty je množství vypouštěného feromonu tímto mravencem úměrné kvalitě a množství nalezené potravy.
- Feromonové stopy tak směřují ostatní mravence ke zdrojům potravy, s časem však tyto stopy vyprchávají.

Hlavní rozdíly mezi chováními skutečných mravenců a umělých mravenců v modelech ACO jsou tyto:

- Skuteční mravenci se pohybují v reálném/spojitém prostředí asynchronně, umělí mravenci/agenti se pohybují v diskrétním prostředí synchronně.
- Skuteční mravenci se při návratu do mraveniště řídí feromonovými stopami, umělí mravenci se v každém cyklu vrací do mraveniště po stejných cestách, kterými se v tomto cyklu pohybovali od mraveniště.
- Skuteční mravenci vypouští feromon neustále, umělí mravenci značkují cestu „feromonem“ pouze při návratu do mraveniště.
- Chování skutečných mravenců je založeno na implicitním vyhodnocení cest. To spočívá v tom, že pohyb po kratších cestách trvá mravencům kratší dobu, proto mohou tyto cesty opakovat častěji a tím se množství feromonu na těchto cestách zvyšuje. Umělí mravenci vyhodnocují cesty explicitně, a to při návratech do mraveniště podle kvality a délky cesty.
- Skuteční mravenci jsou prakticky slepí, umělí mravenci „zrak“ mají.
- Umělí mravenci mají paměť.

Původní ACO algoritmus byl navržen pro řešení TSP a nazývá se Ant-cycle algorithm:

1. Vynulujte čítač cyklů $c = 0$ a nastavte počáteční stav intenzit feromonů $\tau_{ij} = a$ na všech cestách $i-j$, $i = 1 \dots n$, $j = 1 \dots n$, $i \neq j$ (n je počet míst/uzlů, a je malé kladné číslo). Vytvořte m n -místných seznamů Tabu_k , $k = 1 \dots m$, kde m je uvažovaný počet mravenců.
2. Vynulujte přírůstky intenzity feromonů $\Delta \tau_{ij} = 0$ na všech cestách $i-j$.
3. Umístěte náhodně m mravenců/agentů do některých z n míst.
4. Nastavte index s na první položky seznamů Tabu_k (tj. $s = 1$) a do každého seznamu $\text{Tabu}_k(s)$ uložte index místa, ve kterém se právě nachází k -tý mravenec.
5. Inkrementujte hodnotu indexu s .
6. Pro každého mravence k (nachází se právě v místě s indexem $i = \text{Tabu}_k(s - 1)$) vyberte cestu do následujícího místa (například do místa s indexem j) a vložte index tohoto místa do seznamu $\text{Tabu}_k(s)$.

Cesta k -tého mravence z místa i do místa j se vybírá na základě pravděpodobnosti této cesty, která je dána vztahem

$$p_{ij}^k = \begin{cases} \frac{\tau_{ij}^\alpha \cdot \eta_{ij}^\beta}{\sum_{l=1, l \notin Tabu_k}^n \tau_{il}^\alpha \cdot \eta_{il}^\beta} & \text{pro } j \notin Tabu_k, \\ 0 & \text{jinak.} \end{cases}$$

Symbol η_{ij} označuje vzájemnou „viditelnost“ míst i a j , která je nepřímo úměrná jejich vzdálenosti d_{ij} ($\eta_{ij} = 1/d_{ij}$), α a β jsou parametry, které nastavují vlivy aktuální intenzity feromonové stopy a vzájemné viditelnosti míst.

7. Pokud nejsou seznamy $Tabu_k$ plné (nebyla dosud navštívena všechna místa), tj. platí-li vztah $s < n$, tak se vraťte na bod 5, jinak pokračujte.
8. Ze seznamů $Tabu_k$ vypočítejte délky cest L_k , které uskutečnili jednotliví mravenci v aktuálním cyklu a k těmto délkám připočtete délky cest z posledních do prvních míst v těchto seznamech.

9. Aktualizujte nejkratší nalezenou cestu (po prvním cyklu pouze uložte nejkratší nalezenou cestu, po každém dalším cyklu nejprve porovnejte aktuální nejkratší cestu s dříve uloženou nejkratší cestou a poté uložte kratší z nich).
10. Spočítejte přírůstky intenzit feromonových stop na všech hranách $i-j$ uložených jednotlivými mravenci k takto:

$$\Delta\tau_{ij}^k = \begin{cases} \frac{Q}{L_k} & \text{je-li hrana } i-j \text{ v seznamu } Tabu_k, \\ 0 & \text{jinak.} \end{cases}$$
$$\Delta\tau_{ij} = \sum_{k=1}^n \Delta\tau_{ij}^k$$

kde Q je konstanta reprezentující celkové množství feromonu vyloučeného jedním mravencem během jednoho cyklu.

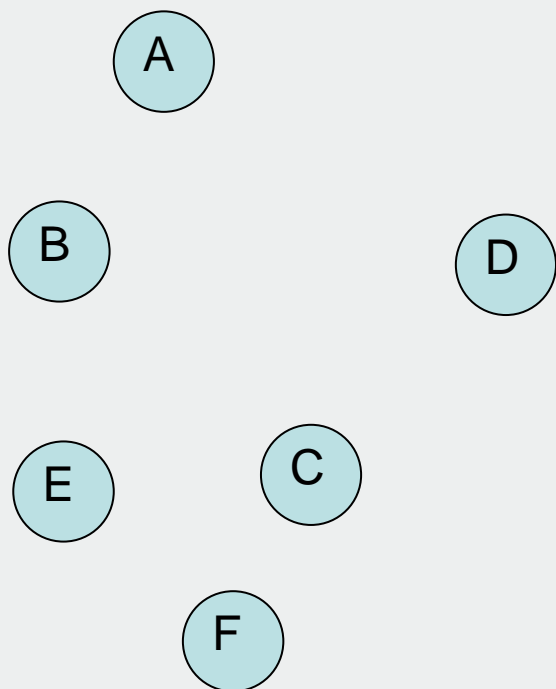
11. Upravte intenzitu feromonových stop na všech hranách $i-j$:

$$\tau_{ij} = (1 - \rho) \cdot \tau_{ij} + \Delta\tau_{ij}.$$

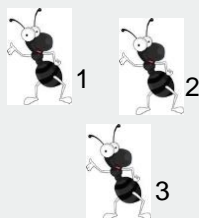
Symbolem $\rho \in \langle 0, 1 \rangle$ je označen koeficient, který reprezentuje vypařování feromonů.

12. Inkrementujte hodnotu čítače cyklů c a pokud platí podmínka $c < c_{max}$, kde c_{max} je předem stanovený maximální počet cyklů, tak se vraťte na bod 2, jinak vraťte nejkratší nalezenou cestu.

Příklad (uvažujme 6 míst a 3 mravence)



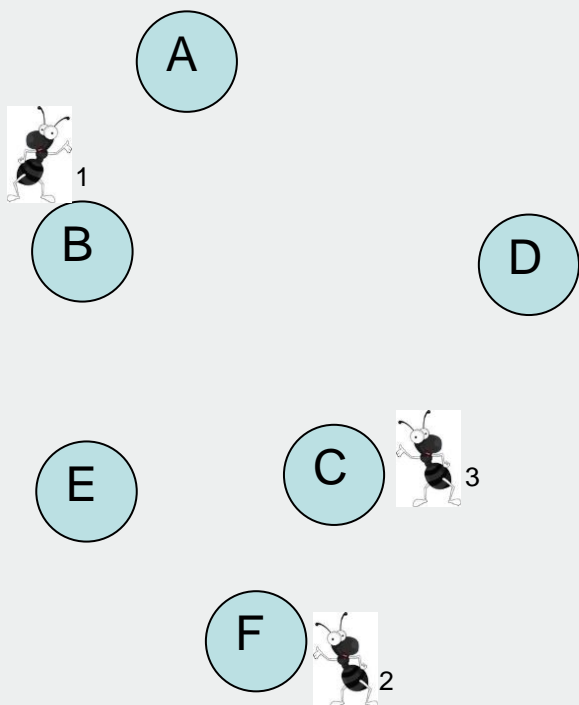
d_{ij}	A	B	C	D	E	F
A	0	20	40	35	40	55
B	20	0	30	40	20	40
C	40	30	0	25	20	15
D	35	40	25	0	45	45
E	40	20	20	45	0	20
F	55	40	15	45	20	0



$$\tau_{ij} = 1, Q = 100,$$

$$\text{Tabu}_1 = [], \text{Tabu}_2 = [], \text{Tabu}_3 = []$$

Mravenci se náhodně rozmístí



d_{ij}	A	B	C	D	E	F
A	0	20	40	35	40	55
B	20	0	30	40	20	40
C	40	30	0	25	20	15
D	35	40	25	0	45	45
E	40	20	20	45	0	20
F	55	40	15	45	20	0

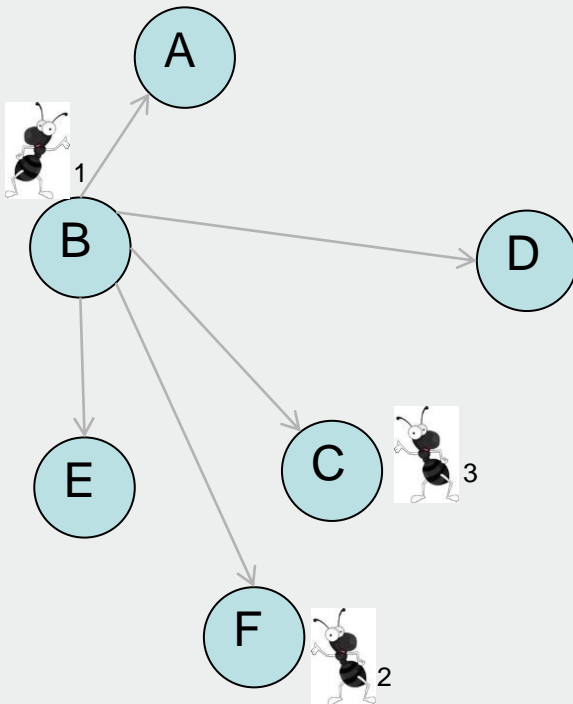
$$\Delta \tau_{ij} = 0$$

$$\text{Tabu}_1 = [\text{B}]$$

$$\text{Tabu}_2 = [\text{F}]$$

$$\text{Tabu}_3 = [\text{C}]$$

První mravenec se rozhoduje, kterou cestou půjde



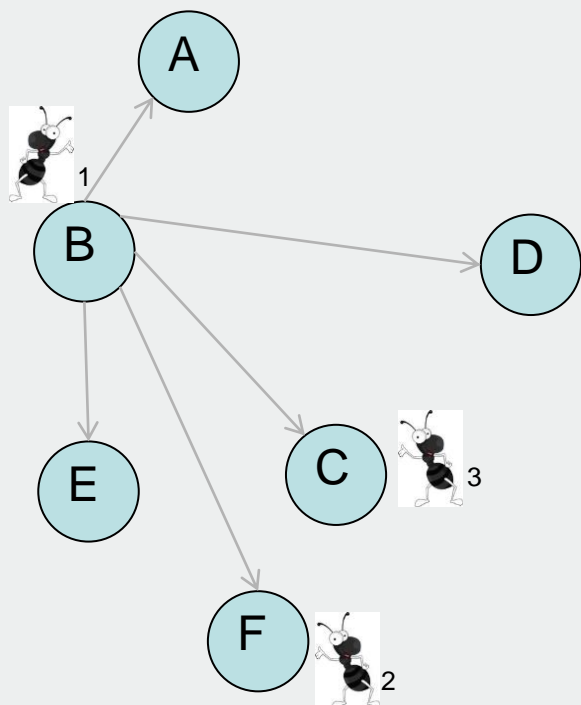
d_{ij}	A	B	C	D	E	F
A	0	20	40	35	40	55
B	20	0	30	40	20	40
C	40	30	0	25	20	15
D	35	40	25	0	45	45
E	40	20	20	45	0	20
F	55	40	15	45	20	0

$$p_{ij}^k = \frac{\tau_{ij}^\alpha \cdot \eta_{ij}^\beta}{\sum_{l=1, l \notin Tabu_k}^n \tau_{il}^\alpha \cdot \eta_{il}^\beta} \quad \text{pro } j \notin Tabu_k,$$

Nechť pro jednoduchost $\alpha = \beta = 1$

$$\begin{aligned}
 p_{BA}^1 &= \frac{1^1 \cdot 1/20^1}{1^1 \cdot 1/20^1 + 1^1 \cdot 1/30^1 + 1^1 \cdot 1/40^1 + 1^1 \cdot 1/20^1 + 1^1 \cdot 1/40^1} = \\
 &= \frac{0.05}{0.183} = 0.273
 \end{aligned}$$

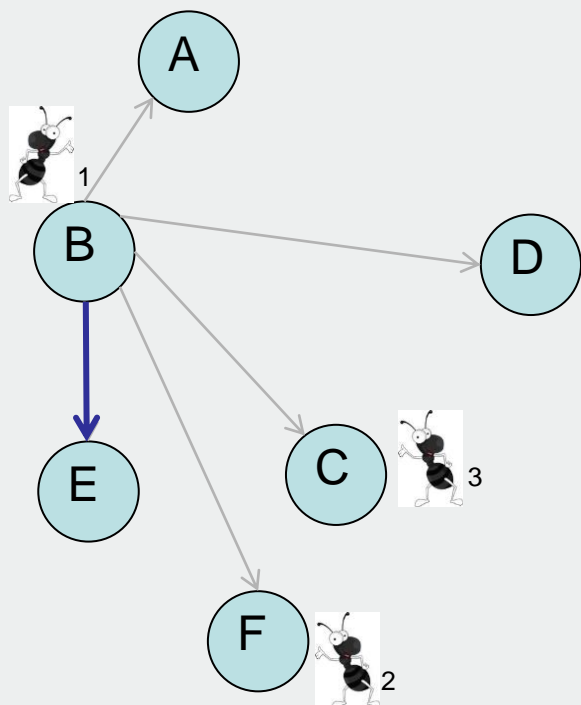
První mravenec se rozhoduje, kterou cestou půjde



d_{ij}	A	B	C	D	E	F
A	0	20	40	35	40	55
B	20	0	30	40	20	40
C	40	30	0	25	20	15
D	35	40	25	0	45	45
E	40	20	20	45	0	20
F	55	40	15	45	20	0

$$\begin{aligned}
 p_{BE}^1 &= p_{BA}^1 = 0.273 \\
 p_{BD}^1 &= p_{BF}^1 = \frac{1^1 \cdot 1/40^1}{0.183} \\
 &= 0.136 \\
 p_{BC}^1 &= \frac{1^1 \cdot 1/30^1}{0.183} = 0.182
 \end{aligned}$$

První mravenec se rozhoduje, kterou cestou půjde

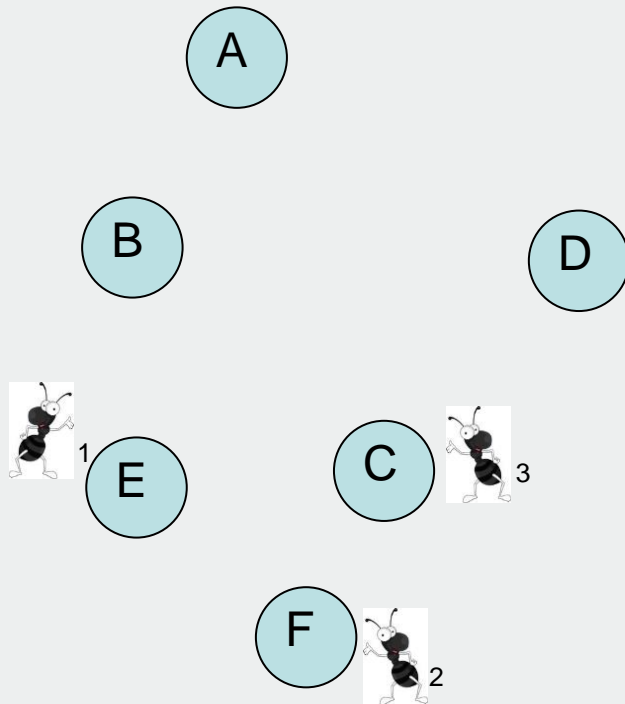


Interval pro generování náhodného čísla, například $\langle 0, 1 \rangle$, se rozdělí podle pravděpodobností $\langle 0, p_{BA} \rangle, \langle p_{BA}, p_{BA} + p_{BC} \rangle, \langle p_{BA} + p_{BC}, p_{BA} + p_{BC} + p_{BD} \rangle, \langle p_{BA} + p_{BC} + p_{BD}, p_{BA} + p_{BC} + p_{BD} + p_{BE} \rangle, \langle p_{BA} + p_{BC} + p_{BD} + p_{BE}, 1 \rangle$:

$\langle 0, 0.273 \rangle, \langle 0.273, 0.455 \rangle, \langle 0.455, 0.591 \rangle, \langle 0.591, 0.864 \rangle, \langle 0.864, 1 \rangle$.

Pak se generuje náhodné číslo $\langle 0, 1 \rangle$, např. 0.672, které odpovídá čtvrtému intervalu \Rightarrow první mravenec si tak vybere cestu do místa E: $\text{Tabu}_1 = [B, E]$.

Postup se opakuje

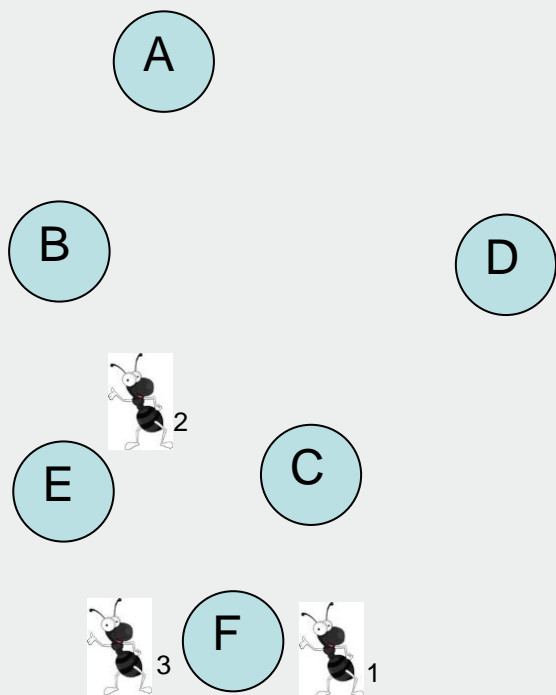


Podobně postupují zbývající dva mravenci a vyberou si například cesty do míst C a B:

$\text{Tabu}_2 = [F, C]$,

$\text{Tabu}_3 = [C, B]$.

Postup se opakuje, dokud mravenci neprošli všemi místy:



Nechť jsou cesty mravenců těsně před návratem do výchozích míst tyto (jsou dány seznamy Tabu):

$\text{Tabu}_1 = [B, E, D, A, C, F],$

$\text{Tabu}_2 = [F, C, D, B, A, E],$

$\text{Tabu}_3 = [C, B, D, A, E, F].$

Délky těchto cest včetně návratů do původních míst jsou tyto:

$$L_1 = 20 + 45 + 35 + 40 + 15 + 40 = 195$$

$$L_2 = 15 + 25 + 40 + 20 + 40 + 20 = 160$$

$$L_3 = 30 + 40 + 35 + 40 + 20 + 15 = 180$$

Nejkratší cestou je cesta druhého mravence:

$$T_{\text{nejkratší}} = \text{Tabu}_2$$

$$L_{\text{nejkratší}} = L_2$$

d_{ij}	A	B	C	D	E	F
A	0	20	40	35	40	55
B	20	0	30	40	20	40
C	40	30	0	25	20	15
D	35	40	25	0	45	45
E	40	20	20	45	0	20
F	55	40	15	45	20	0

Nyní se vypočítají přírůstky feromonových stop

$$\text{Tabu}_1 = [\text{B}, \text{E}, \text{D}, \text{A}, \text{C}, \text{F}], \quad L_1 = 195,$$

$$\text{Tabu}_3 = [\text{C}, \text{B}, \text{D}, \text{A}, \text{E}, \text{F}], \quad L_3 = 180,$$

$$\text{Tabu}_2 = [\text{F}, \text{C}, \text{D}, \text{B}, \text{A}, \text{E}], \quad L_2 = 160$$

$$Q = 100$$

$$\Delta\tau_{ij}^k = \begin{cases} \frac{Q}{L_k} & \text{je-li hrana } ij \text{ v seznamu } \text{Tabu}_k, \\ 0 & \text{jinak.} \end{cases}$$

$$\Delta\tau_{ij} = \Delta\tau_{ij} + \Delta\tau_{ij}^k.$$

$$\Delta\tau_{AB} = 0 + 100/160 + 0 = 0.625$$

$$\Delta\tau_{AD} = 100/195 + 0 + 100/180 = 1.069$$

$$\Delta\tau_{AF} = 0 + 0 + 0 = 0$$

$$\Delta\tau_{BC} = 0 + 0 + 100/180 = 0.556$$

$$\Delta\tau_{BE} = 100/195 + 0 + 0 = 0.513$$

$$\Delta\tau_{CD} = 0 + 100/160 + 0 = 0.625$$

$$\Delta\tau_{CF} = 100/195 + 100/160 + 100/180 = 1.694$$

$$\Delta\tau_{DE} = 100/195 + 0 + 0 = 0.513$$

$$\Delta\tau_{EF} = 0 + 100/160 + 100/180 = 1.181$$

$$\Delta\tau_{AC} = 100/195 + 0 + 0 = 0.513$$

$$\Delta\tau_{AE} = 0 + 100/160 + 100/180 = 1.181$$

$$\Delta\tau_{BD} = 0 + 100/160 + 100/180 = 1.181$$

$$\Delta\tau_{BF} = 100/195 + 0 + 0 = 0.513$$

$$\Delta\tau_{CE} = 0 + 0 + 0 = 0$$

$$\Delta\tau_{DF} = 0 + 0 + 0 = 0$$

Pak se aktualizují feromonové stopy

$$\begin{array}{lllll} \Delta\tau_{AB} = 0.625 & \Delta\tau_{AC} = 0.513 & \Delta\tau_{AD} = 1.069 & \Delta\tau_{AE} = 1.181 & \Delta\tau_{AF} = 0 \\ \Delta\tau_{BC} = 0.556 & \Delta\tau_{BD} = 1.181 & \Delta\tau_{BE} = 0.513 & \Delta\tau_{BF} = 0.513 & \\ \Delta\tau_{CD} = 0.625 & \Delta\tau_{CE} = 0 & \Delta\tau_{CF} = 1.694 & & \\ \Delta\tau_{DE} = 0.513 & \Delta\tau_{DF} = 0 & & & \\ \Delta\tau_{EF} = 1.181 & & & & \end{array}$$

$$\tau_{ij} = (1 - \rho) \cdot \tau_{ij} + \Delta\tau_{ij} \quad \text{Počáteční } \tau_{ij} = 1 \text{ a necht' } \rho = 0.1$$

$$\begin{aligned} \tau_{AB} &= 0.9 \cdot 1 + 0.625 = 1.525 \\ \tau_{AD} &= 0.9 \cdot 1 + 1.069 = 1.969 \\ \tau_{AF} &= 0.9 \cdot 1 + 0 = 0.9 \\ \tau_{BC} &= 0.9 \cdot 1 + 0.556 = 1.456 \\ \tau_{BE} &= 0.9 \cdot 1 + 0.513 = 1.413 \\ \tau_{CD} &= 0.9 \cdot 1 + 0.625 = 1.525 \\ \tau_{CF} &= 0.9 \cdot 1 + 1.694 = 2.594 \\ \tau_{DE} &= 0.9 \cdot 1 + 0.513 = 1.413 \\ \tau_{EF} &= 0.9 \cdot 1 + 1.181 = 2.081 \end{aligned}$$

$$\begin{aligned} \tau_{AC} &= 0.9 \cdot 1 + 0.513 = 1.413 \\ \tau_{AE} &= 0.9 \cdot 1 + 1.181 = 2.081 \\ \tau_{BD} &= 0.9 \cdot 1 + 1.181 = 2.081 \\ \tau_{BF} &= 0.9 \cdot 1 + 0.513 = 1.413 \\ \tau_{CE} &= 0.9 \cdot 1 + 0 = 0.9 \\ \tau_{DF} &= 0.9 \cdot 1 + 0 = 0.9 \end{aligned}$$

Nakonec se testuje, zda bude výpočet opakován, nebo zda se ukončí

Inkrementuje se hodnota čítače c a pokud platí nerovnost $c \leq 100$, pak se výpočet opakuje od snímku 26, jinak se vrací nejkratší nalezené řešení $(T_{\text{nejkratší}}, L_{\text{nejkratší}})$.

Optimalizace hejnem částic - PSO

Algoritmus PSO byl původně navržen pro simulaci pohybů ptačího hejna, dnes se však používá především k řešení spojitých optimalizačních úloh.

Algoritmus PSO má několik společných rysů s GA:

- Pracuje s populací jedinců, nazývanými částice, které představují „řešení“ podobně jako chromozomy v GA.
- Hodnocení kvality částic je prováděno pomocí hodnotících (fitness) funkcí.
- Počáteční rozložení částic v prohledávaném prostoru jsou náhodná.

Každá částice k je reprezentována polohou v n -rozměrném prostoru, vektorem rychlosti jejího pohybu a pamětí předchozích úspěchů při hledání.

V průběhu výpočtu se jak poloha, tak rychlost každé částice k mění, a to v závislosti na dosud nejlepší poloze této částice a na dosud nejlepší poloze nalezené hejnem (nejúspěšnější částicí hejna):

$$\vec{x}^k(t + \Delta t) = \vec{x}^k(t) + \vec{v}^k(t) \cdot \Delta t$$

$$\vec{v}^k(t + \Delta t) = \omega \cdot \vec{v}^k(t) + c_p \cdot r_p \cdot \left(\vec{x}_{best}^k - \vec{x}^k(t) \right) + c_g \cdot r_g \cdot \left(\vec{x}_{best} - \vec{x}^k(t) \right)$$

a pro prakticky vždy uvažovaný jednotkový přírůstek času ($\Delta t = 1$), pak

$$\vec{x}^k(t + 1) = \vec{x}^k(t) + \vec{v}^k(t)$$

$$\vec{v}^k(t + 1) = \omega \cdot \vec{v}^k(t) + c_p \cdot r_p \cdot \left(\vec{x}_{best}^k - \vec{x}^k(t) \right) + c_g \cdot r_g \cdot \left(\vec{x}_{best} - \vec{x}^k(t) \right)$$

Koeficient setrvačnosti ω se v původním algoritmu nevyskytoval (byl tedy roven jedné), dnes se doporučuje volit jeho hodnotu v rozmezí 0.4 až 0.9.

Koeficienty c_p (kognitivní) a c_g (sociální) jsou váhové koeficienty a udávají míru důležitosti individuální paměti a sociálního vlivu. Doporučuje se volit oba tyto koeficienty stejné s hodnotami mírně vyššími než 2, například $c_p = c_g = 2.05$.

Koeficienty r_p a r_g se nastavují náhodně v intervalu $(0,1)$ s rovnoměrným rozložením pravděpodobnosti.

Pro doposud nejlepší pozice nalezené částicí k a všemi částicemi hejna pak zřejmě platí vztahy (pro aktuální čas τ):

$$\begin{aligned} f(\vec{x}_{best}^k) &\geq f(\vec{x}^k(\tau)), & \tau &= 0 \dots t \\ f(\vec{x}_{best}) &\geq f(\vec{x}_{best}^k), & k &= 1 \dots m \end{aligned}$$

kde $f(.)$ je hodnotící/účelová (fitness) funkce a m počet částic hejna.

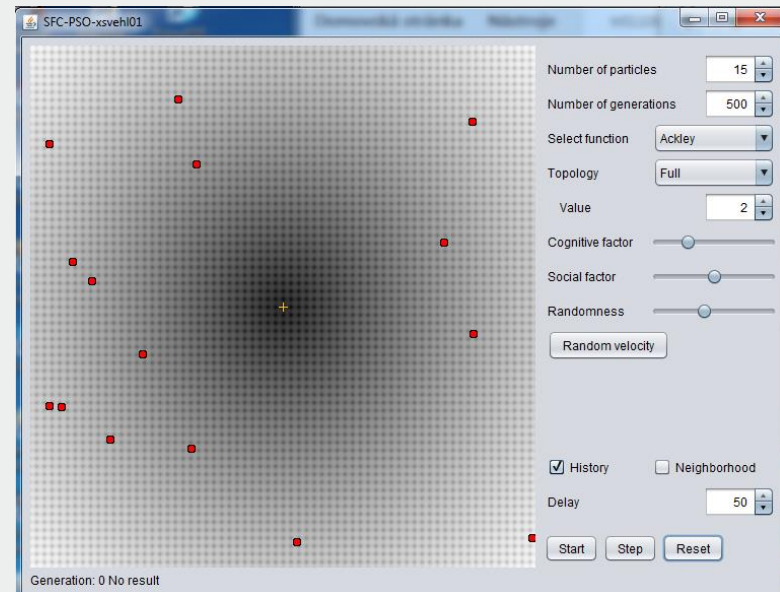
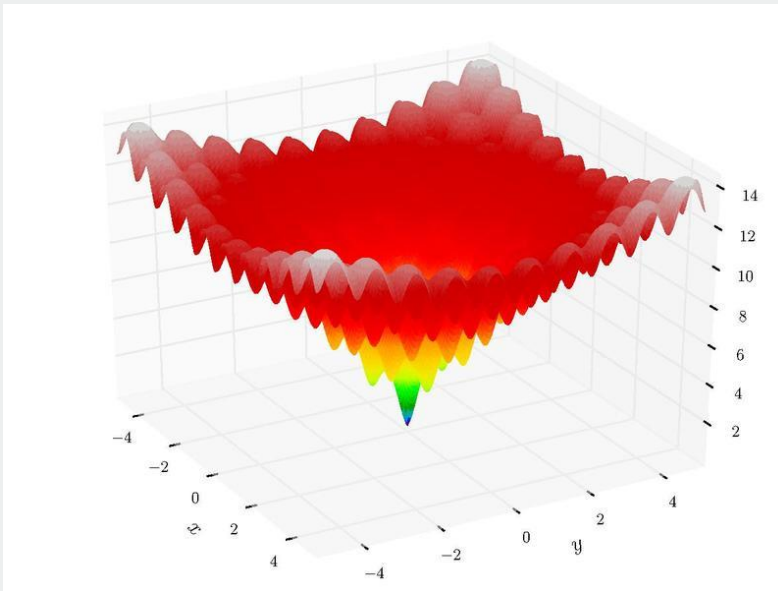
Agoritmus PSO

1. Zvolte počet částic m (obvykle v rozmezí 20 až 100).
2. Zvolte hodnoty koeficientů ω , c_p a c_g .
3. Nastavte náhodně počáteční hodnoty pozic a rychlostí částic:
 - $x_i^k = \text{random}(x_{ilow}, x_{iup})$ pro každou částici $k = 1 \dots m$ a pro každou souřadnici i n -rozměrného prostoru. Symboly x_{ilow} , resp. x_{iup} značí dolní a horní hranici prohledávaného prostoru v i -té souřadnici.
 - $v_i^k = \text{random}(-v_{imax}, v_{imax})$ pro každou částici $k = 1 \dots m$ a pro každou souřadnici i n -rozměrného prostoru. Symbol v_{imax} značí maximální rychlost ve směru i -té souřadnice.
4. Nastavte pozici \vec{x}^k částice k jako její dosud nejlepší pozici \vec{p}^k .
5. Určete nejlepší pozici hejna \vec{g} , pro které je dáno pozicí nejlépe hodnocené částice $\vec{g} \geq \vec{p}^k$, $k = 1 \dots m$.
6. Nastavte index k na první částici ($k = 1$).

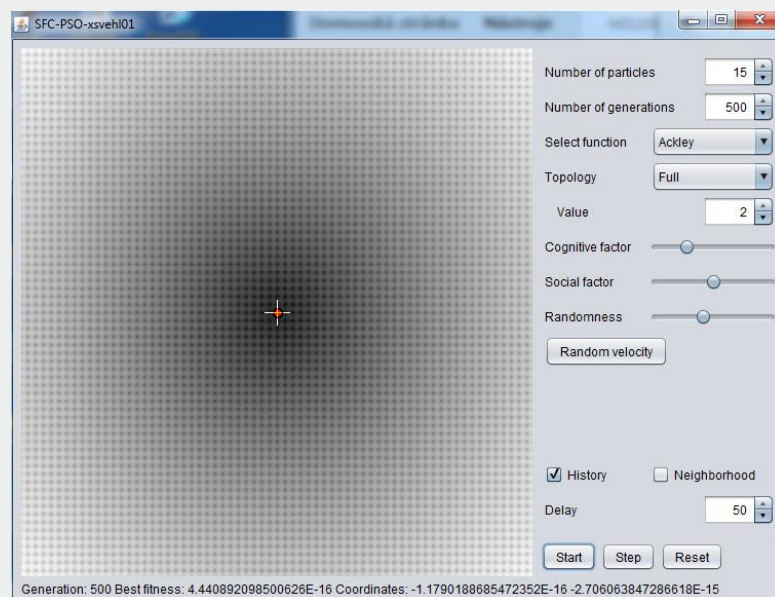
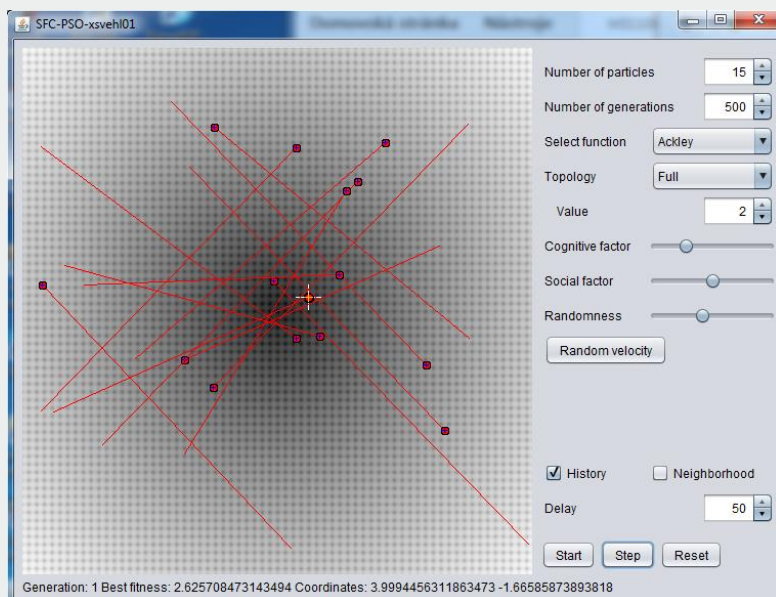
7. Určete náhodná čísla r_p a r_g (z intervalu $<0,1>$, rovnoměrné rozdělení pravděpodobnosti).
8. Vypočtete novou rychlost částice a její novou pozici podle vztahů
$$\vec{x}^k(t+1) = \vec{x}^k(t) + \vec{v}^k(t)$$
$$\vec{v}^k(t+1) = \omega \cdot \vec{v}^k(t) + c_p \cdot r_p \cdot (\vec{x}_{best}^k - \vec{x}^k(t)) + c_g \cdot r_g \cdot (\vec{x}_{best} - \vec{x}^k(t))$$
9. Ohodnoťte novou polohu částice \vec{x}^k a pokud je toto hodnocení lepší, než její dosavadní nejlepší ohodnocení, pak upravte její nejlepší pozici $\vec{p}^k = \vec{x}^k$.
10. Pokud je hodnocení částice lepší, než hodnocení dosud nejlepší pozice hejna, pak upravte nejlepší pozici hejna $\vec{g} = \vec{p}^k$.
11. Inkrementujte index k , a pokud platí $k \leq m$, tak se vraťte na bod 7, jinak pokračujte.
12. Pokud se řešení zlepšuje a pokud nebyl překročen zvolený maximální čas výpočtu, tak se vraťte na bod 6, jinak řešení ukončete (řešení je dáno nejlepší pozicí hejna \vec{g}).

Příklad (hledání globálního minima Ackleyho funkce):

$$f(x) = -20e^{-0.2 \cdot \sqrt{0.5(x^2+y^2)}} - e^{0.5(\cos(2\pi x) + \cos(2\pi y))} + e + 20$$



Příklad pokračování:



Pro řešení optimalizačních problémů byla navržena celá řada dalších algoritmů inspirovaných přírodou. Jejich přehled k roku 2013 lze nalézt v článku Fisher, I., et al.: A Brief Review of Nature-Inspired Algorithms for Optimization, Elektrotehnikvestnik80(3), 2013, ze kterého je převzata i tabulka uvedená na dalším snímku.

Swarm intelligence based algorithms			Bio-inspired (not SI-based) algorithms		
Algorithm	Author	Reference	Algorithm	Author	Reference
Accelerated PSO	Yang et al.	[69], [71]	Atmosphere clouds model	Yan and Hao	[67]
Ant colony optimization	Dorigo	[15]	Biogeography-based optimization	Simon	[56]
Artificial bee colony	Karaboga and Basturk	[31]	Brain Storm Optimization	Shi	[55]
Bacterial foraging	Passino	[46]	Differential evolution	Storn and Price	[57]
Bacterial-GA Foraging	Chen et al.	[6]	Dolphin echolocation	Kaveh and Farhoudi	[33]
Bat algorithm	Yang	[78]	Japanese tree frogs calling	Hernández and Blum	[28]
Bee colony optimization	Teodorović and Dell'Orco	[62]	Eco-inspired evolutionary algorithm	Parpinelli and Lopes	[45]
Bee system	Lucic and Teodorovic	[40]	Egyptian Vulture	Sur et al.	[59]
BeeHive	Wedde et al.	[65]	Fish-school Search	Lima et al.	[14], [3]
Wolf search	Tang et al.	[61]	Flower pollination algorithm	Yang	[72], [76]
Bees algorithms	Pham et al.	[47]	Gene expression	Ferreira	[19]
Bees swarm optimization	Drias et al.	[16]	Great salmon run	Mozaffari	[43]
Bumblebees	Comellas and Martinez	[12]	Group search optimizer	He et al.	[26]
Cat swarm	Chu et al.	[7]	Human-Inspired Algorithm	Zhang et al.	[80]
Consultant-guided search	Iordache	[29]	Invasive weed optimization	Mehrabian and Lucas	[42]
Cuckoo search	Yang and Deb	[74]	Marriage in honey bees	Abbass	[1]
Eagle strategy	Yang and Deb	[75]	OptBees	Maia et al.	[41]
Fast bacterial swarming algorithm	Chu et al.	[8]	Paddy Field Algorithm	Premaratne et al.	[48]
Firefly algorithm	Yang	[70]	Roach infestation algorithm	Havens	[25]
Fish swarm/school	Li et al.	[39]	Queen-bee evolution	Jung	[30]
Good lattice swarm optimization	Su et al.	[58]	Shuffled frog leaping algorithm	Eusuff and Lansey	[18]
Glowworm swarm optimization	Krishnanand and Ghose	[37], [38]	Termite colony optimization	Hedayatzadeh et al.	[27]
Hierarchical swarm model	Chen et al.	[5]	Physics and Chemistry based algorithms		
Krill Herd	Gandomi and Alavi	[22]	Big bang-big Crunch	Zandi et al.	[79]
Monkey search	Mucherino and Seref	[44]	Black hole	Hatamlou	[24]
Particle swarm algorithm	Kennedy and Eberhart	[35]	Central force optimization	Formato	[21]
Virtual ant algorithm	Yang	[77]	Charged system search	Kaveh and Talatahari	[34]
Virtual bees	Yang	[68]	Electro-magnetism optimization	Cuevas et al.	[13]
Weightless Swarm Algorithm	Ting et al.	[63]	Galaxy-based search algorithm	Shah-Hosseini	[53]
Other algorithms			Gravitational search	Rashedi et al.	[50]
Anarchic society optimization	Shayeghi and Dadashpour	[54]	Harmony search	Geem et al.	[23]
Artificial cooperative search	Civicioglu	[9]	Intelligent water drop	Shah-Hosseini	[52]
Backtracking optimization search	Civicioglu	[11]	River formation dynamics	Rabanal et al.	[49]
Differential search algorithm	Civicioglu	[10]	Self-propelled particles	Vicsek	[64]
Grammatical evolution	Ryan et al.	[51]	Simulated annealing	Kirkpatrick et al.	[36]
Imperialist competitive algorithm	Atashpaz-Gargari and Lucas	[2]	Stochastic diffusion search	Bishop	[4]
League championship algorithm	Kashan	[32]	Spiral optimization	Tamura and Yasuda	[60]
Social emotional optimization	Xu et al.	[66]	Water cycle algorithm	Eskandar et al.	[17]