

Numeryczne metody optymalizacji inspirowane naturą

Piotr Dąbrowski

slajdy do kursu:
Optymalizacja systemów
prof. dr hab. inż. Jerzy Świątek

Optymalizacja kombinatoryczna

- rozwiązujemy zadanie optymalizacji
- zbiór rozwiązań jest dyskretny, skończony
 - lub możemy go sprowadzić do dyskretnego
- przeszukanie wszystkich rozwiązań jest często niemożliwe
 - bywa, że możliwych rozwiązań jest więcej niż atomów we wszechświecie
- najczęściej spotykane problemy
 - **problem komiwojażera**
 - problem plecakowy
 - problem szeregowania zadań



Problemy P i NP

- problem NP - rozwiązanie można zweryfikować w czasie wielomianowym
 - problem P - znalezienie rozwiązania ma złożoność wielomianową
 - wszystkie znane problemy P zawierają się w NP (choć nie wiadomo, czy tak jest zawsze: $P \stackrel{?}{=} NP$)
- przykładowe problemy
 - czy można wyznaczyć niepusty podzbiór $\{-2, 6, -3, 72, 10, -11\}$ sumujący się do zera?
 - znalezienie rozwiązania rośnie wykładniczo wraz z kolejnymi elementami zbioru
 - weryfikacja rozwiązania, np. $\{-2, 6, -3, 10, -11\}$ jest problemem liniowym (wielomianowym)
 - czy dane sudoku posiada choć jedno rozwiązanie?
 - odpowiedzenie na to pytanie wymaga rozwiązania danej łamigłówki
 - dla nierozwiązywalnych układów wymaga pełnego przeglądu
 - weryfikacja danego rozwiązania jest banalna (sprawdzamy reguły)
- w optymalizacji kombinatorycznej mamy do czynienia z problemami NP-trudnymi lub NP-kompletnymi, czyli problemami klasy NP niebędącymi problemami P

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

Heurystyki

- metoda rozwiązywania problemów, dla której nie ma gwarancji znalezienia rozwiązania optymalnego
- korzysta się z nich gdy pełny algorytm jest zbyt kosztowny lub nieznany
- najczęściej stosowane w przypadku zbadanych i znanych problemów
- w przeciwieństwie do algorytmów dokładnych
 - czas działania zazwyczaj nie jest dowolnie długi (w algorytmach dokładnych może być)
 - jedynie w specyficznych przypadkach znajduje rozwiązanie dokładne
 - może w ogóle nie znaleźć rozwiązania
- nie daje nam gwarancji przybliżenia do rozwiązania dokładnego (w przeciwieństwie do algorytmów aproksymacyjnych)
- stosowane w przypadku problemów NP-trudnych i NP-kompletnych

Metaheurystyki

- heurystyka, którą można stosować do rozwiązania dowolnego problemu, który da się opisać za pomocą pewnych zdefiniowanych przez algorytm pojęć
- najważniejsza różnica:
 - heurystyki często rozwiązują konkretny problem
 - metaheurystyki rozwiązują szersze spektrum problemów
- zazwyczaj opisują sposób przechodzenia między możliwymi rozwiązaniami
- często możliwe jest dostrajanie ich działania za pomocą różnych parametrów
 - nie istnieją uniwersalne wartości tych parametrów

Metody optymalizacji inspirowane naturą

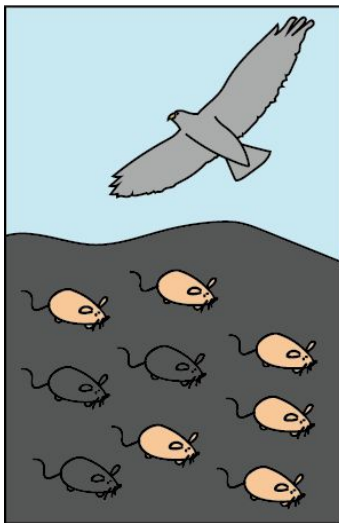
- są często metaheurystykami lub heurystykami
- wykorzystują analogię do procesów zachodzących w naturze w procesie przeszukiwania przestrzeni rozwiązań
- między innymi
 - błądzenie losowe (*random walk*)
 - symulowane wyżarzanie (*simulated annealing*)
 - **algorytmy genetyczne**
 - strategie ewolucyjne
 - **algorytmy mrówkowy (*ant colony optimization*)**
 - **algorytm pszczeli (*artificial bee colony*)**
 - **algorytm roju cząsteczek (*particle swarm optimization*)**
 - **algorytm świetlika (*firefly algorithm*)**
 - algorytm kukułki (*cuckoo search*)
 - algorytm nietoperza (*bat algorithms*)
 - algorytm zapylania kwiatów (*flower pollination algorithms*)

Metody

Klasyczny algorytm genetyczny - motywacja

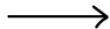
- zainspirowany mechanizmem dziedziczenia cech w populacji zaproponowanym w 1958 przez R.A. Fishera, którego głównymi założeniami są:
 - allele jako zbiór alternatywnych wartości dla danego genu
 - ciąg alleli tworzy genotyp (który wyznacza cechy osobnika)
 - każda generacja osobników tworzy kolejną generację potomstwa
 - istnieje funkcja oceniająca (*fitness function*), zwracająca liczbę potomstwa, które otrzyma dany ciąg alleli
 - istnieją operatory genetyczne, w szczególności operator mutacji, który modyfikuje potomków danego osobnika w taki sposób, że następna generacja zawsze różni się od poprzedniej
- symulowany jest proces selekcji naturalnej, w którym populacja osobników traktowana jest jako przestrzeń rozwiązań, w której iteracyjnie (co generację) poszukujemy osobników osiągających coraz wyższą wartość funkcji oceny
- selekcja naturalna jest procesem optymalizacji, w którym przeżywają lepiej dostosowani do życia osobnicy

Klasyczny algorytm genetyczny - motywacja



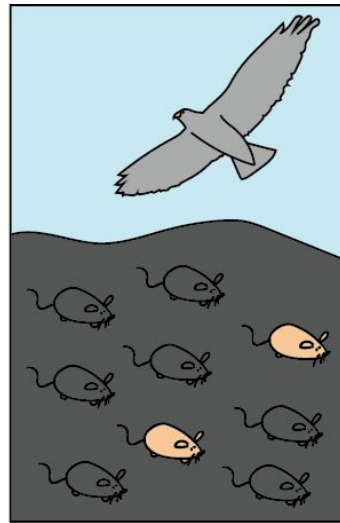
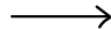
A population of mice has moved into a new area where the rocks are very dark. Due to natural genetic variation, some mice are black, while others are tan.

Some mice are eaten by birds



Tan mice are more visible to predatory birds than black mice. Thus, tan mice are eaten at higher frequency than black mice. Only the surviving mice reach reproductive age and leave offspring.

Mice reproduce, giving next generation

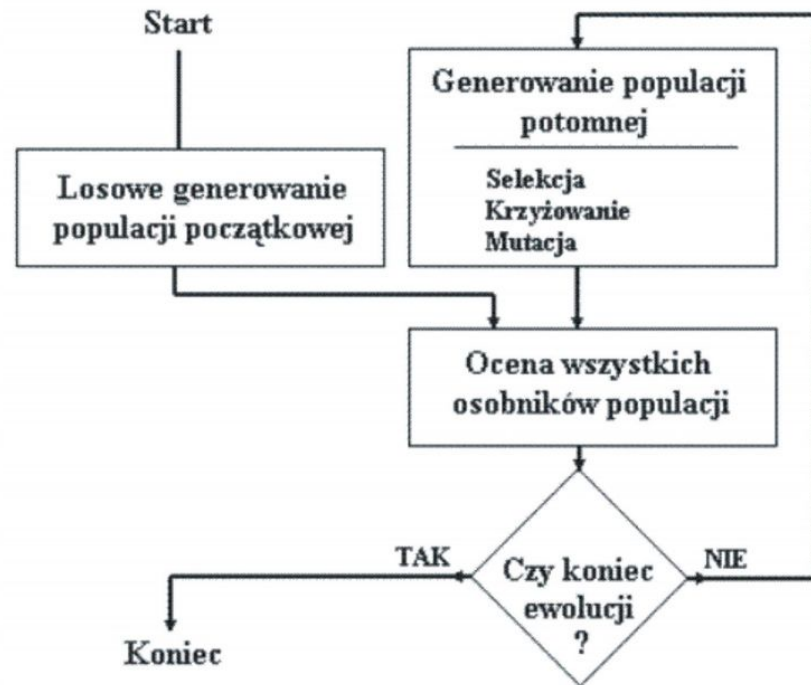
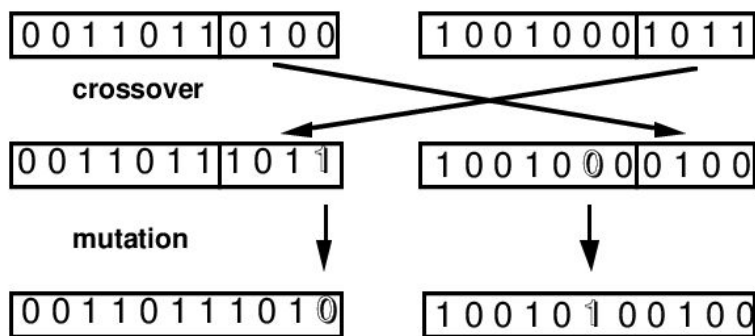


Because black mice had a higher chance of leaving offspring than tan mice, the next generation contains a higher fraction of black mice than the previous generation.

Klasyczny algorytm genetyczny

- zmienna decyzyjna: binarna reprezentacja (allel może przyjąć wartość 0 lub 1)
- funkcja celu: funkcja zwracająca ocenę danego rozwiązania
- zakładana jest populacja początkowa (wiele osobników - wiele rozwiązań)
- operatory:
 - selekcji
 - zwraca zbiór osobników z obecnej generacji, którzy wygenerują potomstwo
 - osobnicy wybierani są losowo, ale...
 - osobnicy o większej wartości funkcji oceny mają większe prawdopodobieństwo wyboru
 - krzyżowania (*crossover*)
 - tworzy nowych potomków łącząc dwóch osobników (ich genotypy)
 - wykorzystywany zawsze przy tworzeniu potomstwa
 - mutacji (*mutation*)
 - losowo modyfikuje pojedynczy allel
 - zachodzi bardzo rzadko

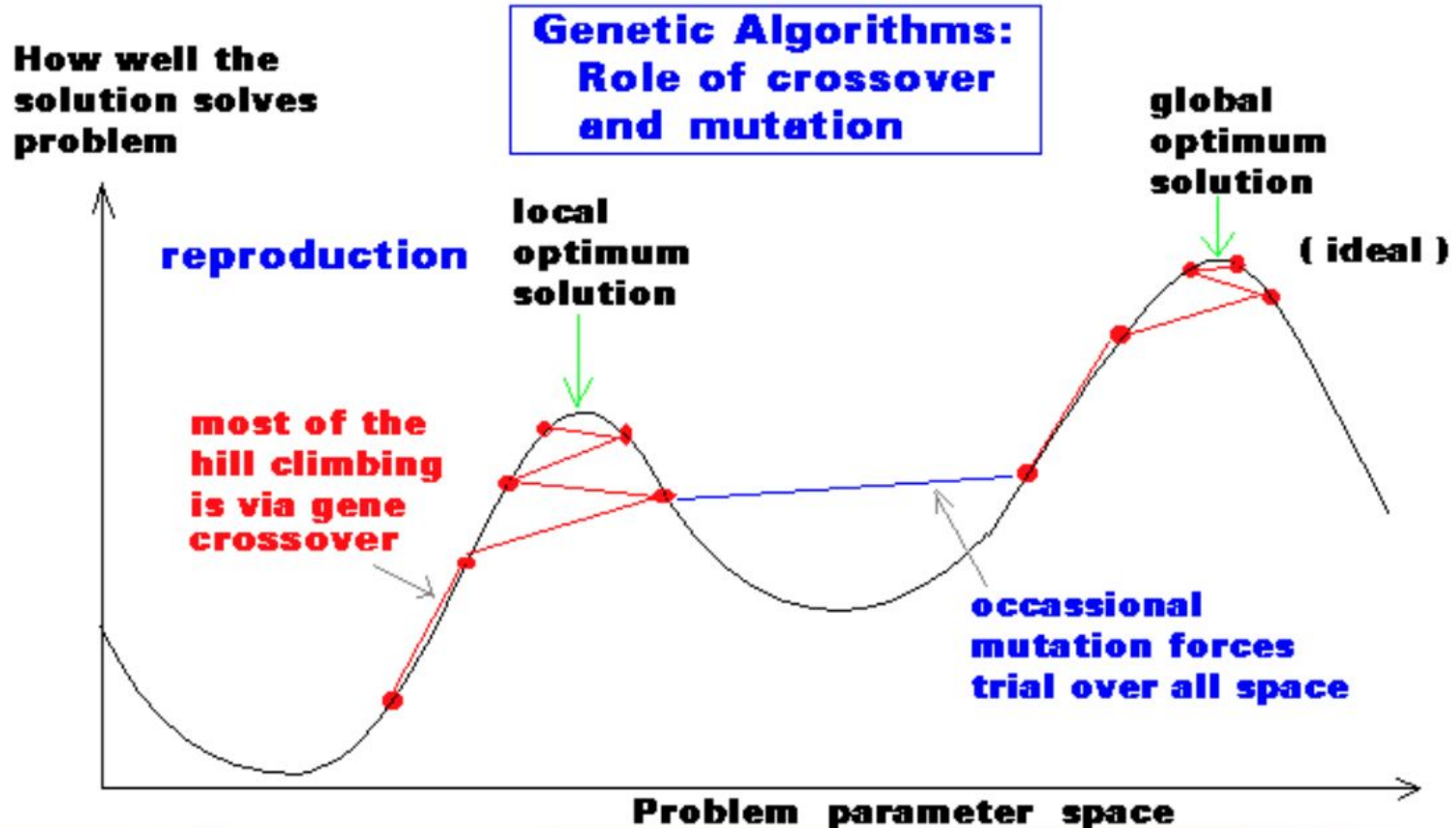
Klasyczny algorytm genetyczny



Algorytm genetyczny

- możliwość stosowania różnych kodowań reprezentacji (zmiennych decyzyjnych)
 - rezygnacja z ograniczenia się do binarnych alleli
 - możliwe reprezentacje: binarna, całkowitoliczbowa, rzeczywistoliczbowa, permutacyjna
- różnorodność technik stosowanych w ramach poszczególnych operatorów, w większości zależnych od reprezentacji, m.in.:
 - mutacja
 - binarna: negacja
 - całkowitoliczbowa: losowa wartość, zmiana o losową wartość
 - rzeczywistoliczbowa: zmiana o losową wartość z rozkładu jednostajnego lub innego wybranego
 - permutacyjna: zamiana dwóch losowych wartości, wstawianie wartości w losowej pozycji
 - krzyżowanie
 - binarna/liczbowa: przecięcie w losowym miejscu, przecięcie w kilku losowych miejscach
 - permutacyjna: zachowując jak najwięcej krawędzi lub cykli
 - selekcja: proporcjonalna do wartości funkcji oceny, rankingowa, turniejowa, *rosyjska ruletka*

Algorytm genetyczny - eksploracja vs eksploatacja



Algorytm genetyczny - wizualizacja

<https://ostap0207.github.io/web-ga-tsp/>

Algorytm genetyczny - zalety i wady

- zwraca dobre rozwiązania w stosunkowo krótkim czasie
- wymaga wyliczenia oceny każdego rozwiązania w populacji
- nie działa dobrze dla bardziej skomplikowanych problemów
 - wraz ze wzrostem rozmiaru reprezentacji, rośnie przestrzeń rozwiązań
 - rzadkie zmiany w mutacji występują wystarczająco często, by *psuć* rozwiązanie
- często utyka w optimach lokalnych
 - przez brak innych mechanizmów wyjścia z optimum lokalnego niż mutacja
 - mutacja z jednej strony nie może być za duża, żeby nie stracić dobrych rozwiązań
 - z drugiej strony zbyt mała mutacja uniemożliwia wyjście z optimów lokalnych

Evolved antenna

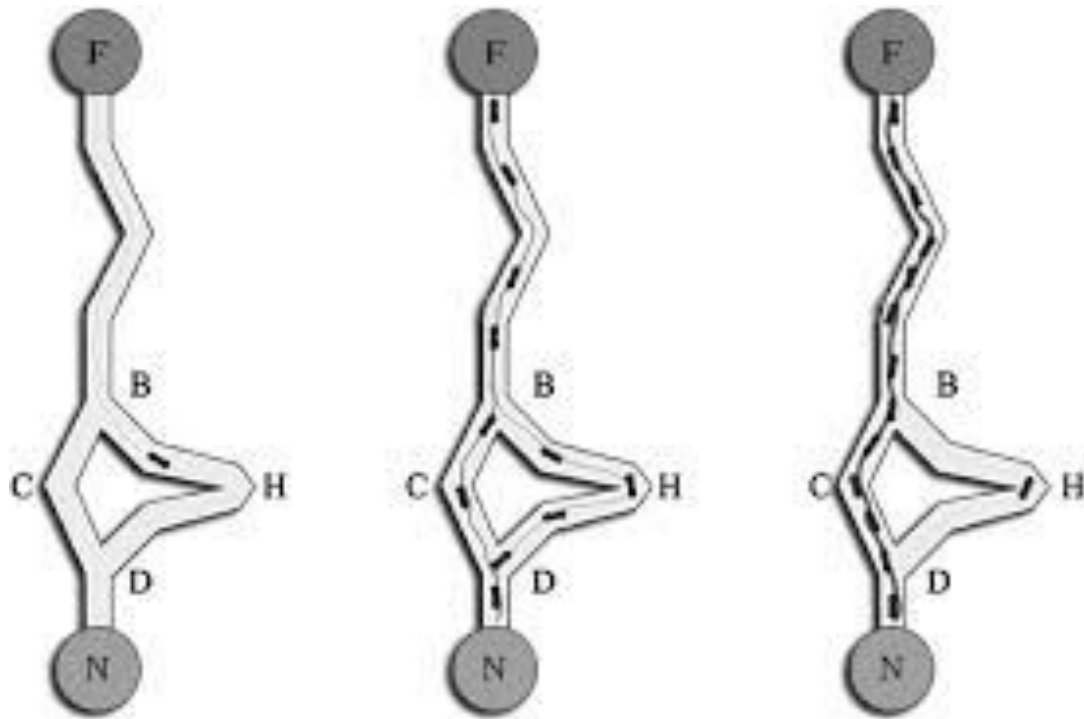
- antena opracowana przy pomocy algorytmu ewolucyjnego
- antena osiągnęła lepsze parametry w porównaniu z projektami opracowywanymi ręcznie
- posiada kształt, który byłby trudny do stworzenia w ręcznym procesie projektowania
- zdjęcie po prawej przedstawia antenę, która wykorzystana została przez NASA do komunikacji z satelitą pracującą w przestrzeni kosmicznej



Algorytm mrówkowy - motywacja

- zainspirowany zachowaniem mrówek poszukujących pożywienia
 - mrówki poszukując pożywienia poruszają się w sposób losowy
 - gdy znajdują pożywienie, wracają do swojej kolonii pozostawiając ślad z feromonów
 - gdy inna mrówka natknie się na ten ślad, przestaje poruszać się losowo i podąża za śladem
 - po pewnym czasie feromony wyparowują, więc ich siła maleje
 - im dłuższa trasa do pożywienia, tym więcej jest czasu, by feromony odparowały
 - krótsze trasy zapewniają zatem, że siła działania feromonów będzie większa
 - ostatecznie wszystkie mrówki poruszać się będą jedną, najlepszą drogą, a pozostałe drogi wyparują
- oparty na wiedzy, której dostarczył nam eksperyment Deneubourga z 1989 roku
- wykorzystywany do szukania dobrych dróg w grafach
 - zazwyczaj w problemie komiwojażera i pochodnych
 - siła feromonu zależy od jakości znalezionej drogi (np. jej długości)

Algorytm mrówkowy - motywacja



Algorytm mrówkowy

1. Inicjalizacja:

- a. wszystkie ścieżki w grafie inicjalizowane są niewielką ilością feromonu
- b. mrówki umieszczane są losowo w różnych miastach

2. Mrówki wędrują między miastami:

- a. wyznaczanie ścieżek
 - i. każda mrówka wędruje do sąsiadującego miasta wybierając losowo drogę na podstawie prawdopodobieństwa wyznaczanego bazując na ilości feromonu danej krawędzi
 - ii. mrówki posiadają pamięć, która nie pozwala im odwiedzić danego miasta ponownie
 - iii. mrówki kończą wędrowanie po osiągnięciu celu
- b. aktualizacja feromonów na poszczególnych krawędziach
 - i. dodawana jest ilość feromonu proporcjonalna do długości drogi przebytej przez mrówkę
 - ii. odejmowana jest ilość feromonu związana z odparowywaniem
- c. jeśli niespełniony jest warunek stopu, kroki są powtarzane

Algorytm mrówkowy

- wybór krawędzi
 - η - atrakcyjność krawędzi *a priori*
często przyjmuje się $\eta = 1 / d_{xy}$
(tj. odwrotnie proporcjonalna do długości/kosztu krawędzi)
 - τ - ilość feromonu na danej krawędzi (*a posteriori*)

$$p_{xy}^k = \frac{(\tau_{xy}^\alpha)(\eta_{xy}^\beta)}{\sum_{z \in \text{allowed}_x} (\tau_{xz}^\alpha)(\eta_{xz}^\beta)}$$

- aktualizacja krawędzi (ilości feromonu)

$$\tau_{xy} \leftarrow (1 - \rho)\tau_{xy} + \sum_k \Delta\tau_{xy}^k$$

- ilość pozostawianego feromonu

- Q - stała
- L_k - koszt drogi (np. długość)

$$\Delta\tau_{xy}^k = \begin{cases} Q/L_k & \text{if ant } k \text{ uses curve } xy \text{ in its tour} \\ 0 & \text{otherwise} \end{cases}$$

- parametry

- liczba mrówek
- α - wpływ ilości feromonu na wybór danej krawędzi ($\alpha \leq 0$)
- β - wpływ atrakcyjności krawędzi *a priori* na wybór krawędzi ($\beta \geq 1$)
- ρ - siła parowania feromonu

Algorytm mrówkowy - wizualizacja

<https://poolik.github.io/visual-aco/#/visualisation>

Algorytm mrówkowy - zalety i wady

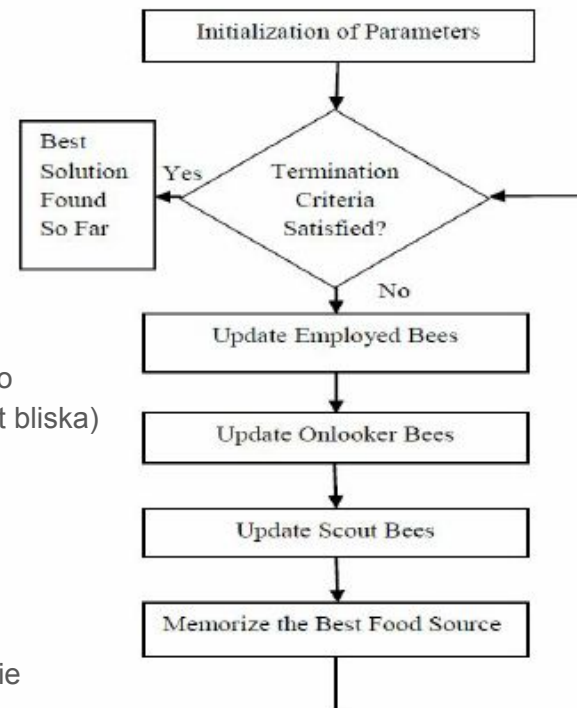
- stosowany w różnych problemach (nie tylko do poruszania się po grafie):
 - problem plecakowy
 - zwijanie białka
 - wykrywanie krawędzi na obrazie
- dostosowuje się do zmiennych warunków
 - może służyć do wyznaczania tras w przypadku zamknięć dróg
 - w tym zastosowaniu nie zapomina poprzednio znalezionych dobrych rozwiązań
- stosunkowo duża liczba parametrów

Algorytm pszczele

- zainspirowany sposobem wyszukiwania pokarmu przez pszczoły
 - pszczoły robotnice poszukują źródeł pokarmu wybierając losowy kierunek i obszar poszukiwań
 - po znalezieniu źródła pożywienia, wracają do ula, by poinformować o tym innych
 - pszczoły chcąc przekazać informację o innym, wykonują specyficzny taniec (*waggle dance*), który jest dłuższy i bardziej ekspresyjny im pokarmu jest więcej
 - pozostałe pszczoły oceniają taniec zwiadowców i decydują się podążać za najlepiej tańczącym
 - także pszczoły eksplorujące poprzednio inne źródło, mogą w pewnym momencie zmienić zdanie, gdy uznają, że inny obszar rokuje na większą ilość pokarmu niż ten, który obecnie eksploatują
- wyraźnie wyznacza dwa etapy: eksploracji i eksploatacji
- analogie zastosowane w algorytmie:
 - lokalizacja pokarmu - pojedyncze rozwiązanie problemu
 - ilość pokarmu - jakość danego rozwiązania (wartość funkcji oceny)
 - liczba pszczół - liczba rozwiązań w populacji

Algorytm pszczeli

- inicjalizacja - generowanie losowych rozwiązań
- pracownice (*employed bees*)
 - dla każdej pszczoły generowane są losowe rozwiązania w sąsiedztwie aktualnego (sąsiedztwo zdefiniowane jako rozwiązania, których reprezentacja wektorowa jest bliska)
 - pszczoła porzuca poprzednie rozwiązanie jeśli nowe jest lepsze
- obserwatorzy (*onlooker bees*) - eksploatacja
 - pszczoły wybierają jedno z rozwiązań proponowanych przez pracownice
 - wybór rozwiązania dokonywany jest losowo, lecz proporcjonalnie do jego oceny (lepsze rozwiązania mają większe prawdopodobieństwo wyboru)
 - każda pszczoła ewaluuje jedno rozwiązanie w sąsiedztwie wybranego przez siebie (sąsiedztwo zdefiniowane identycznie jak w przypadku pracownic)
- zwiadowcy (*scout bees*) - eksploracja
 - pracownice, które nie zauważyły poprawy w otoczeniu swojego rozwiązania stają się zwiadowcami
 - generują one całkowicie losowe rozwiązanie



Algorytm pszczeni

- wyznaczanie źródeł pożywienia - zwiadowcy

ub oraz lb definiują zakres wartości (minimalną i maksymalną dozwoloną wartość)

k - numer wymiaru w wektorze reprezentacji

$$x_{i,k} = lb_k + \Phi_{i,k} \times (ub_k - lb_k)$$

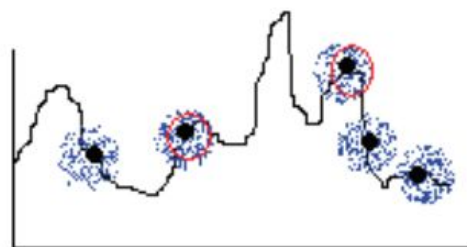
$$\Phi_{i,k} = \text{rand}(0, 1)$$

- poszukiwanie pożywienia - pracujące i obserwatorzy

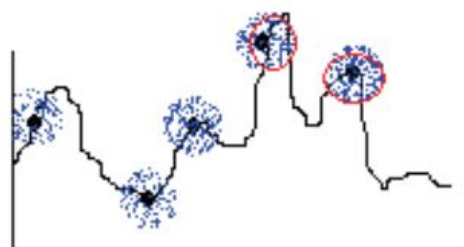
x_i - poprzednie rozwiązanie, x_j - losowe rozwiązanie, Φ - wektor losowych wartości z zakresu $[-1, 1]$

$$v_{i,k} = x_{i,k} + \Phi_{i,k} \times (x_{i,k} - x_{j,k})$$

Algorytm pszczele



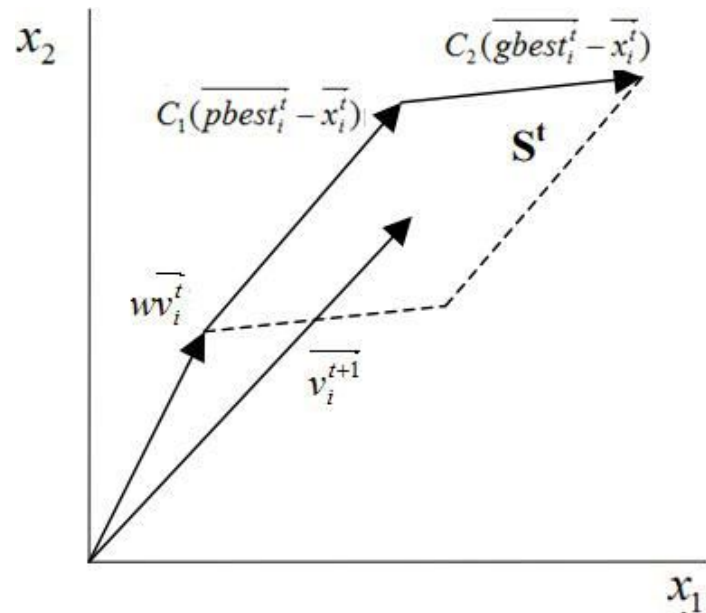
a) punkty - reprezentują wylosowane wartości początkowe, chmury - przeszukiwane sąsiedztwo punktów, koła - najlepsze sąsiedztwa



b) trzy najgorsze obszary zostały zastąpione nowo wylosowanymi, zmieniły się też najlepsze sąsiedztwa - pozostało jedno z poprzedniej iteracji, drugie zostało znalezione w bieżącej

Algorytm optymalizacji rojem cząstek

- zainspirowany ruchem cząstek w przestrzeni
- każda cząstka reprezentuje pojedyncze rozwiązanie
- cząstka ma przypisane prędkość oraz kierunek, w jakim się porusza
- dla każdej z cząstek zapamiętywane jest jej najlepsze rozwiązanie (lokalne)
- zapamiętywane jest także globalnie najlepsze rozwiązanie
- kierunek poruszania się cząstki w każdej iteracji aktualizowany jest na podstawie najlepszego rozwiązania lokalnego i globalnego



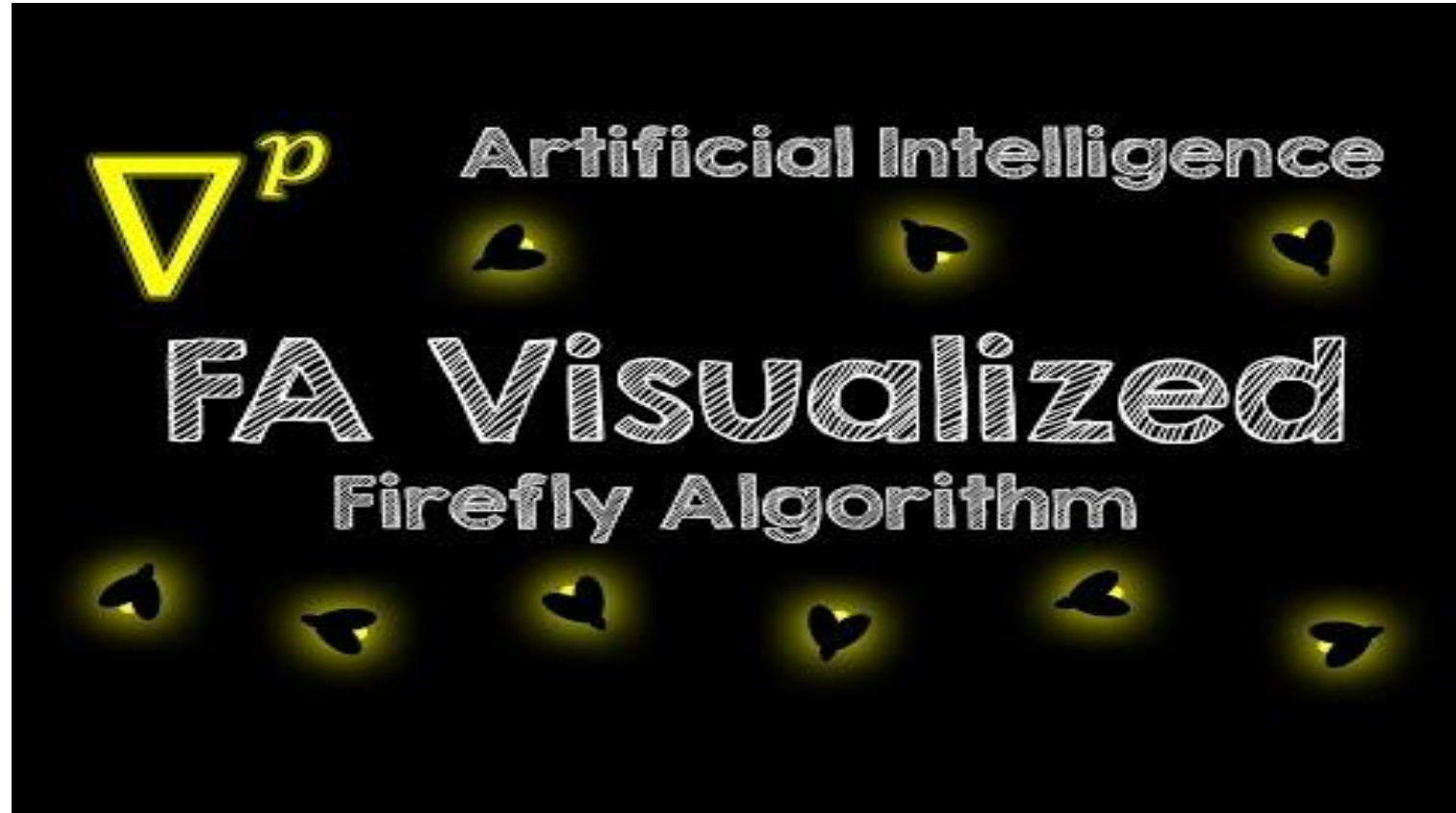
Algorytm optymalizacji rojem cząstek - wizualizacja



Algorytm świetlika

- zainspirowany zjawiskiem bioluminescencji robaczeków świętojańskich
 - jaśniej świecące świetliki przyciągają te o słabszym świetle
 - jeżeli w otoczeniu nie ma osobników o jaśniejszym świetle, wtedy porusza się losowo
- świetlik reprezentuje pojedyncze rozwiązanie
- jasność świecenia jest proporcjonalna do jakości rozwiązania
- bardzo zbliżony w swym działaniu do algorytmu optymalizacji rojem cząstek
 - obie cząstki poruszają się w kierunku globalnego rozwiązania
 - cząstka dodatkowo porusza się w kierunku swojego najlepszego lokalnego rozwiązania
 - świetlik może poruszać się w kierunku najlepszego rozwiązania innego świetlika

Algorytm świetlika - wizualizacja



Krytyka

- dopasowują zjawiska zachodzące w przyrodzie, które niekoniecznie mają cokolwiek wspólnego z optymalizacją
- metafory związane z tymi zjawiskami często są coraz bardziej absurdalne
- coraz nowsze metody nie wprowadzają nic nowego
(algorytm optymalizacji rojem cząstek vs algorytm świetlika)

Weyland, Dennis. "A critical analysis of the harmony search algorithm—How not to solve sudoku." *Operations Research Perspectives* 2 (2015): 97-105.

Źródła

Chrysostomou, D., Gasteratos, A., Nalpantidis, L., & Sirakoulis, G. C. (2012). Multi-view 3D scene reconstruction using ant colony optimization techniques. *Measurement Science and Technology*, 23(11), 114002

Comellas, Francesc, and Javier Ozon. "Graph coloring algorithms for assignment problems in radio networks." *Applications of Neural Networks to Telecommunications 2* (1995): 49-56

de Almeida, Bruno Seixas Gomes, and Victor Coppo Leite. "Particle swarm optimization: A powerful technique for solving engineering problems." *Swarm Intelligence-Recent Advances, New Perspectives and Applications*. IntechOpen, 2019

Holland, John H., "Genetic algorithms." *Scholarpedia* (2012), 7(12):1482

Karaboga, Dervis, "Artificial bee colony algorithm". *Scholarpedia* (2010) , 5(3):6915.

Kumar, Sandeep, Vivek Kumar Sharma, and Rajani Kumari. "Randomized memetic artificial bee colony algorithm." *arXiv preprint arXiv:1408.0102* (2014)

Rejer, Izabela, and Krzysztof Lorenz. "Classic genetic algorithm vs. genetic algorithm with aggressive mutation for feature selection for a brain-computer interface." *Przegląd Elektrotechniczny* 91.2 (2015): 98-102

Weyland, Dennis. "A critical analysis of the harmony search algorithm—How not to solve sudoku." *Operations Research Perspectives* 2 (2015): 97-105

Xin-She Yang, "Nature-Inspired Optimization Algorithms", 2nd Edition (2020)

Yang, X. S., Gandomi, A. H., Talatahari, S., & Alavi, A. H. (Eds.). (2012). *Metaheuristics in water, geotechnical and transport engineering*. Newnes

<http://www.alife.pl/optymalizacja-rojem-czastek>

http://www.zio.iar.pwr.wroc.pl/pea/w6_aco.pdf

http://yadda.icm.edu.pl/yadda/element/bwmmeta1.element.baztech-b91f2a0d-c6e1-499e-984e-ec518cb659e/c/49_L_RUTCZYNSKA-WDOWIAK_JABLONSKI.pdf

https://en.wikipedia.org/wiki/Ant_colony_optimization_algorithms

https://en.wikipedia.org/wiki/Artificial_bee_colony_algorithm

https://en.wikipedia.org/wiki/Combinatorial_optimization

https://en.wikipedia.org/wiki/Firefly_algorithm

https://en.wikipedia.org/wiki/Particle_swarm_optimization

https://eti.pg.edu.pl/documents/176546/25263566/SZ_wyklad2.pdf

<https://lilianweng.github.io/lil-log/2019/09/05/evolution-strategies.html>

<https://ostap0207.github.io/web-ga-tsp/>

https://pl.wikipedia.org/wiki/Algorytm_mr%C3%B3wkowy

[https://pl.wikipedia.org/wiki/Optymalizacja_\(matematyka\)](https://pl.wikipedia.org/wiki/Optymalizacja_(matematyka))

<https://poolik.github.io/visual-aco/#/visualisation>

<https://repozytorium.ukw.edu.pl/bitstream/handle/item/3505/Wieslaw%20Popielarski%20Algorytm%20pszczeli%20w%20optymalizacji%20modelu%20przeplywowego%20szeregowania%20zadan.pdf>

https://sound.eti.pg.gda.pl/student/isd/isd03-algorytmy_genetyczne.pdf

https://www.math.uni.lodz.pl/~marta/2012_2013_zzz/gen.pdf

https://www.youtube.com/watch?v=LPKg_luHxpc

<https://www.youtube.com/watch?v=OUHAypWn1Ro>