

Populacyjne algorytmy optymalizacji globalnej

Michał Okulewicz

Wydział Matematyki i Nauk Informatycznych
Politechnika Warszawska

Plan wykładu

- 1 Wstęp
- 2 Podejścia ewolucyjne
 - Schemat ogólny
 - Algorytmy
 - Operatory krzyżowania i mutacji
- 3 Podejścia inteligencji rojowej
 - Zastosowanie w przemyśle rozrywkowym
 - Zastosowanie w optymalizacji
- 4 Ewolucja Różnicowa
 - Wariant bazowy
 - Modyfikacje

Plan prezentacji

- 1 Wstęp
- 2 Podejścia ewolucyjne
 - Schemat ogólny
 - Algorytmy
 - Operatory krzyżowania i mutacji
- 3 Podejścia inteligencji rojowej
 - Zastosowanie w przemyśle rozrywkowym
 - Zastosowanie w optymalizacji
- 4 Ewolucja Różnicowa
 - Wariant bazowy
 - Modyfikacje

Algorytmy populacyjne

Definicja

Algorytmy populacyjne do podjęcia decyzji o kolejnej próbkce funkcji, wykorzystują informację z kilku poprzednich próbek.

Przykład (Operator krzyżowania)

Operator krzyżowania w algorytmie genetycznym uśrednia 2 próbki o wysokiej jakości funkcji, w nadziei że pomiędzy nimi funkcja również będzie wysokiej jakości.

Rodzaje podejść populacyjnych

Przykład (Podejścia ewolucyjne)

Jeden z głównych nurtów opiera się na mechanizmach obserwowanych w procesie rozmnażania i adaptacji gatunków - czyli na **procesie ewolucji**.

Przykład (Podejścia rojowe)

Drugi z głównych nurtów opiera się na mechanizmach obserwowanych w **zachowaniach społecznych/stadnych** obserwowanych wśród stad ptaków, ławic ryb czy rojów owadów.

Plan prezentacji

- 1 Wstęp
- 2 **Podejścia ewolucyjne**
 - Schemat ogólny
 - Algorytmy
 - Operatory krzyżowania i mutacji
- 3 **Podejścia inteligencji rojowej**
 - Zastosowanie w przemyśle rozrywkowym
 - Zastosowanie w optymalizacji
- 4 **Ewolucja Różnicowa**
 - Wariant bazowy
 - Modyfikacje

Operatory podejść ewolucyjnych

Definicja

Selekcją będziemy nazywać strategię wyboru argumentów funkcji, które będą wykorzystane w kolejnych iteracjach algorytmu.

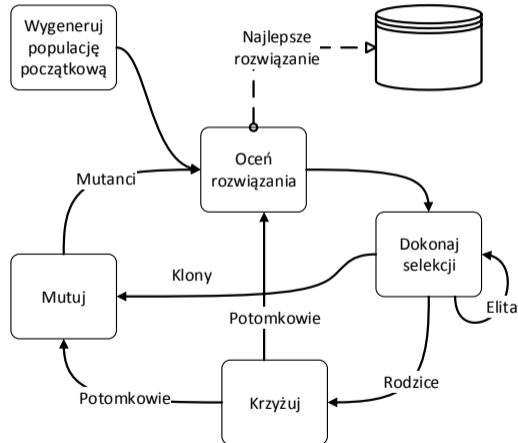
Definicja

Krzyżowaniem będziemy nazywać operator tworzący nową próbkę (osobnika) na podstawie pewnego rodzaju agregacji kilku (dwóch) próbek (osobników).

Definicja

Mutacją będziemy nazywać operator tworzący nową próbkę (osobnika) poprzez niewielką (zwykle losową) zmianę jej współrzędnych.

Schemat podejścia ewolucyjnego



Podejścia ewolucyjne

Definicja

Strategią ewolucyjną będziemy nazywać podejścia, w których operatorami są selekcja i mutacja.

Definicja

Algorytmem ewolucyjnym będziemy nazywać podejścia dla funkcji ciągłych wykorzystujących selekcję, krzyżowanie i mutację.

Definicja

Algorytmem genetycznym będziemy nazywać podejścia dla funkcji dyskretnych wykorzystujących selekcję, krzyżowanie i mutację.

(1+1) Evolution Strategy [5]

Ogólna koncepcja

(1+1)-ES opiera się na adaptacji zakresu mutacji w oparciu o stosunek liczby prób, które poprawiły wynik funkcji do wszystkich prób.

Reguła 1/5 sukcesów

Reguła 1/5 sukcesów została wyprowadzona eksperymentalnie na funkcji kwadratowej, jako metoda najszybszego dojścia do optimum - poprzez adaptację długości kroku.

Strategia Ewolucyjna

- 1: $0.85 \leq a < 1$
- 2: $x \leftarrow x_{init}$
- 3: **while** STOP criteria not met **do**
- 4: $improvements \leftarrow 0$
- 5: **for** $iter \in \{1, 2, \dots, iter_{max}\}$ **do**
- 6: $x' \leftarrow NormalSample(x, \sigma)$ ▷ Próbkę z sąsiedztwa
- 7: **if** $f(x') < f(x)$ **then**
- 8: $x \leftarrow x'$
- 9: $improvements \leftarrow improvements + 1$ ▷ Zliczamy sukcesy dla wartości σ
- 10: **end if**
- 11: **end for**
- 12: **if** $improvements/iter_{max} < 1/5$ **then** ▷ Zmieniamy zasięg mutacji na mniejszy ($0.85 \leq a < 1$)
- 13: $\sigma \leftarrow a * \sigma$
- 14: **else if** $improvements/iter_{max} > 1/5$ **then** ▷ Zmieniamy zasięg mutacji na większy
- 15: $\sigma \leftarrow \sigma/a$
- 16: **end if**
- 17: **end while**

Algorytm Ewolucyjny

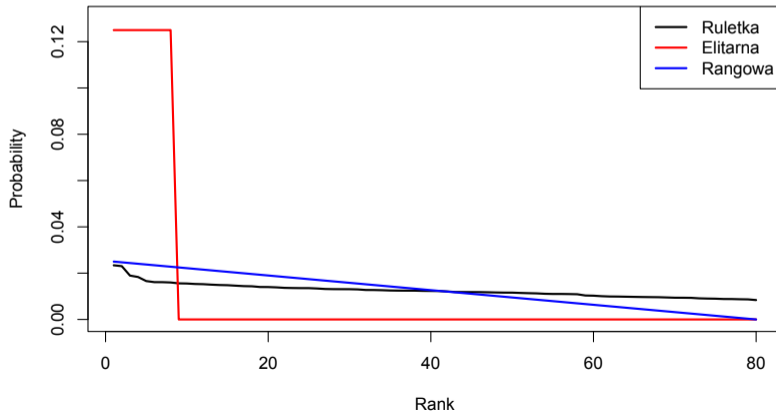
```
1:  $X \leftarrow X_{init}$   
2:  $Evaluate(X)$   
3: for  $iter \in \{1, 2, \dots, iter_{max}\}$  do  
4:    $X' \leftarrow Crossover(X)$   
5:    $X'' \leftarrow Mutate(X, X')$   
6:    $Evaluate(X', X'')$   
7:    $X \leftarrow Selection(X, X', X'')$   
8: end for
```

- ▷ Populacja początkowa
- ▷ Oceń osobniki
- ▷ Stwórz nowe osobniki przy pomocy agregacji
- ▷ Stwórz nowe osobniki przy pomocy losowego zaburzania
- ▷ Oceń nowe osobniki
- ▷ Wybierz kolejną populację

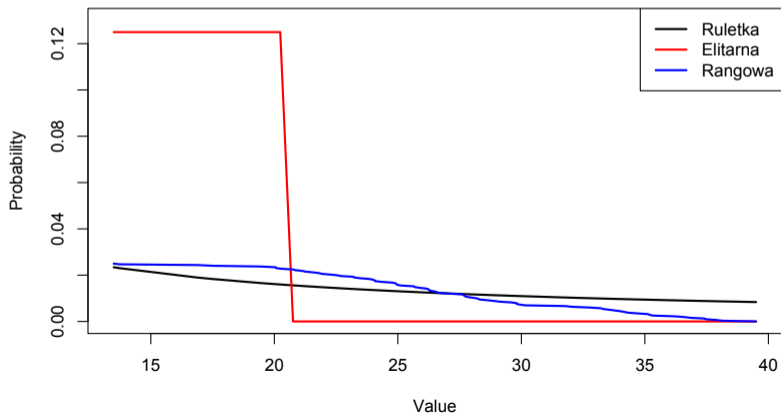
Metody selekcji

- Ruletka (popularna)
- Ruletkowa zrównoważona (popularna z poprawką)
- Rangowa (wzmocnienie nacisku przy płaskich funkcjach)
- Turniejowa (brak oceny bezwzględnej)
- Elitarna (mocna eksploatacja)
- Jednostajna/Prosta (mocna eksploracja) - „dziecko” za „rodzica”

Porównanie rozkładów selekcji

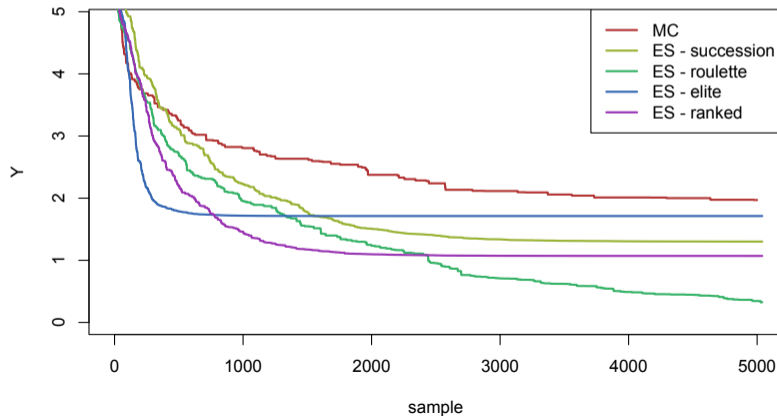


Porównanie rozkładów selekcji



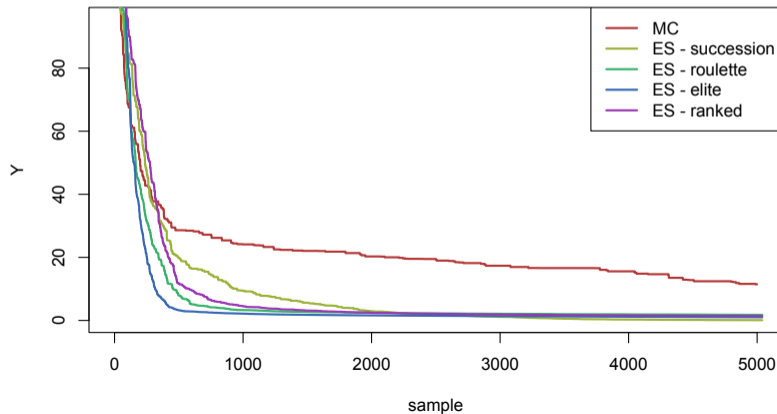
Porównanie zbieżności

Rastrigin 5



Porównanie zbieżności

Rosenbrock 5



Operatory krzyżowania

- Krzyżowanie wektorowe
 - jednopunktowe
 - dwupunktowe
 - wymieniające
- Krzyżowanie arytmetyczne
 - uśredniające
 - liniowe
 - prostokątne

Operatory krzyżowania

- Krzyżowanie wektorowe
 - jednopunktowe
 - dwupunktowe
 - wymieniające
- Krzyżowanie arytmetyczne
 - uśredniające
 - liniowe
 - prostokątne

Operatory krzyżowania

- Krzyżowanie wektorowe
 - jednopunktowe
 - dwupunktowe
 - wymieniające
- Krzyżowanie arytmetyczne
 - uśredniające
 - liniowe
 - prostokątne

Operatory mutacji

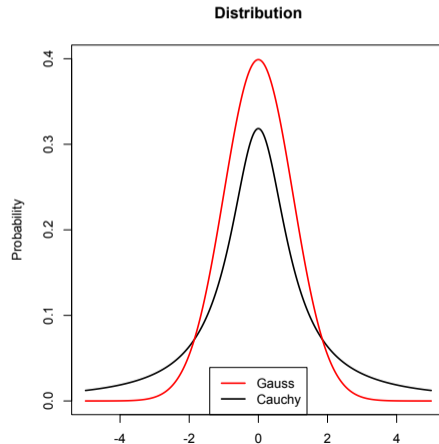
- Mutacja wektorowa
 - przełączanie bitów
- Mutacja arytmetyczna
 - rozkład Gaussa
 - rozkład Cauchy'ego

Operatory mutacji

- Mutacja wektorowa
 - przełączanie bitów
- Mutacja arytmetyczna
 - rozkład Gaussa
 - rozkład Cauchy'ego

Operatory mutacji

- Mutacja wektorowa
 - przełączanie bitów
- Mutacja arytmetyczna
 - rozkład Gaussa
 - rozkład Cauchy'ego



Oczekiwania względem algorytmu

- Eksploracja czy eksploatacja?
 - Makromutacje + niska presja selektywna = Eksploracja?
 - Krzyżowanie + wysoka presja selektywna = Eksploatacja?

Oczekiwania względem algorytmu

- Eksploracja czy eksploatacja?
 - Makromutacje + niska presja selektywna = Eksploracja?
 - Krzyżowanie + wysoka presja selektywna = Eksploatacja?

Inspiracje: Lamarck vs. Darwin

Teoria Ewolucji Lamarcka

Teoria Ewolucji Darwina nie była jedyną rozważaną - de Lamarck zaproponował następujące hipotezy:

- Używane organy rozwijają się, a nieużywane ulegają degeneracji
- Cechy nabyte przez organizmy podlegają przekazaniu potomstwu

Czy wiedza i doświadczenie podlegają dziedziczeniu?

A study shows that when mice are taught o fear an odor, both their offspring and the next generation are born fearing it. The gene for an olfactory receptor activated by the odor is specifically demethylated in the germ line and the olfactory circuits for detecting the odor are enhanced. [12]

Plan prezentacji

- 1 Wstęp
- 2 Podejścia ewolucyjne
 - Schemat ogólny
 - Algorytmy
 - Operatory krzyżowania i mutacji
- 3 **Podejścia inteligencji rojowej**
 - Zastosowanie w przemyśle rozrywkowym**
 - Zastosowanie w optymalizacji**
- 4 Ewolucja Różnicowa
 - Wariant bazowy
 - Modyfikacje

Definicja

Inteligencja Rojowa to grupa podejść z obszaru Inteligencji Obliczeniowej/Sztucznej Inteligencji, które wzorując się na obserwacji zachowań społecznych podążają za następującymi wytycznymi:

- Modelują proste byty o nieskomplikowanej logice
- Zapewniają tym bytom możliwość komunikacji

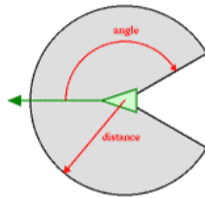
W efekcie uzyskujemy nietrywialne rozwiązania postawionych problemów.



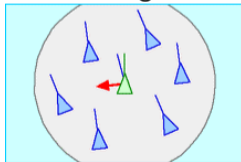
Ależ mój Panie, nie ma takiej armii



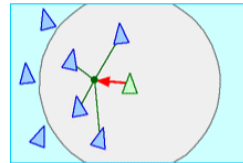
Boidy Reynoldsa [1]



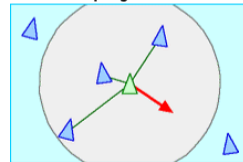
Postrzeganie



Wyrównanie

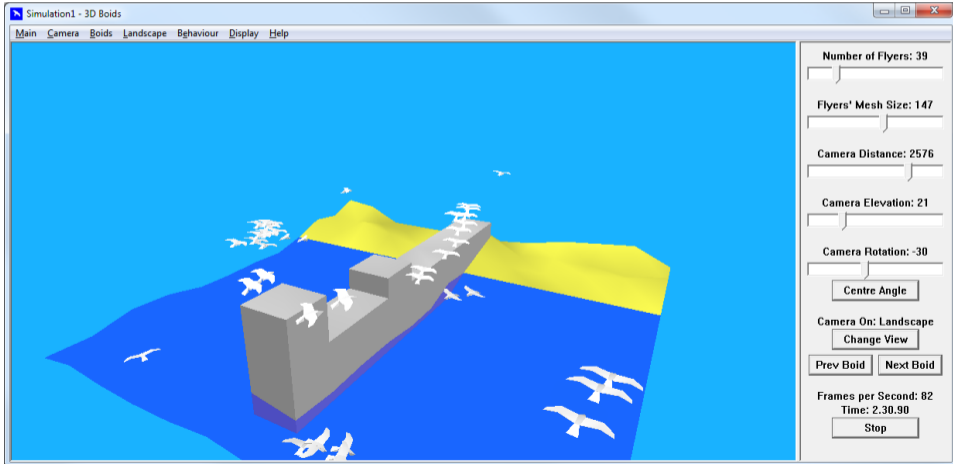


Spójność



Unikanie kolizji

Boidy Reynoldsa [1]



Przykłady

- <https://www.youtube.com/watch?v=jCVwdeAobYc> - Batman Powraca
- <https://www.youtube.com/watch?v=EmTz7EAYLrs&t=315> - Władca Pierścieni
- <https://www.youtube.com/watch?v=2D5x1xUense&t=3m50s> - Gra o Tron

Inteligencja Rojowa w optymalizacji

- Algorytm Mrówkowy (znajdowanie ścieżek w grafach)
- Optymalizacja Rojem Cząstek (optymalizacja ciągła)

Optymalizacja Rojem Cząstek [2]

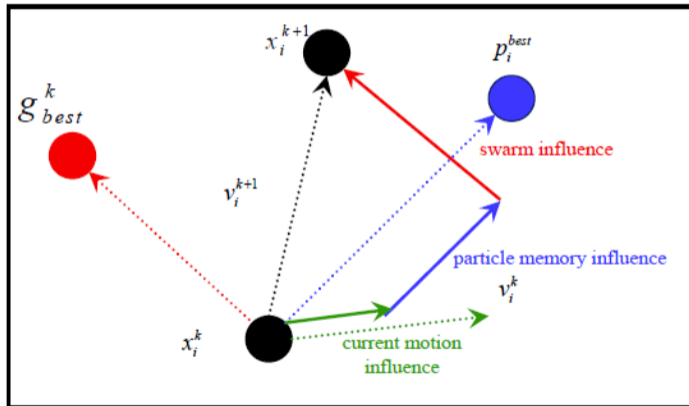
Definicja

Optymalizacja Rojem Cząstek (Particle Swarm Optimization) to algorytm iteracyjny, w którym optymalizacja prowadzona jest przez zbiór obiektów zwanych cząstkami.

Cząstka posiada następujące cechy:

- Aktualne położenie (i wartość) x_t, y_t
- Najlepsze dotychczasowe położenie (i wartość) p_{best}, y_{best}
- Aktualną prędkość v_t
- Strukturę sąsiedztwa i informację o najlepszym sąsiedzie g_{best}

Aktualizacja położenia cząstki



Źródło: <https://www.researchgate.net/figure/>

Velocity-component-construction-of-a-particle-swarm-optimization-algorithm-Mohamed_fig4_324470759

Optymalizacja Rojem Cząstek

- 1: $Particles.X \leftarrow X_{init}$
- 2: $Particles.V \leftarrow \frac{Particle_j.x - Particle_j.x}{2}$
- 3: $EvaluateParticles.X$
- 4: **for** $iter \in \{1, 2, \dots, iter_{max}\}$ **do**
- 5: **for** $p \in Particles$ **do**
- 6: $p.X_{iter} \leftarrow p.X_{iter-1} + p.V_{iter-1}$
- 7: $p.y_{iter} \leftarrow Evaluate(p.X_{iter})$
- 8: **if** $p.y_{iter} < p.y_{best}$ **then**
- 9: $p.y_{best} \leftarrow p.y_{iter}$
- 10: $p.p_{best} \leftarrow p.X_{iter}$
- 11: **end if**
- 12: $p.V_{iter} \leftarrow \omega * p.V_{iter-1} + c_1 * (p.p_{best} - p.X_{iter}) + c_2 * (p.g_{best} - p.X_{iter})$ ▷ c_1, c_2 - losowe wektory
- 13: **end for**
- 14: **end for**

▷ Inicjalizacja położenia roju

▷ Inicjalizacja prędkości roju

▷ Wersja asynchroniczna

▷ c_1, c_2 - losowe

Parametry PSO (Poli) [13]

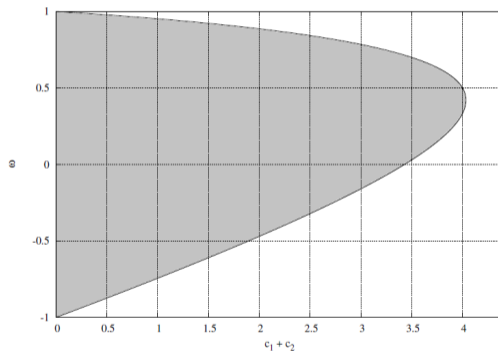
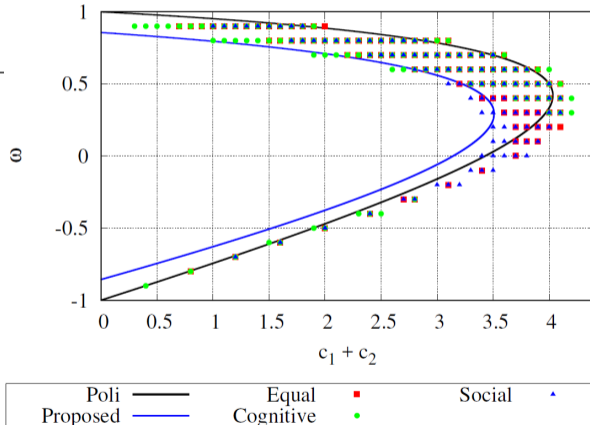


Fig. 1. Visualization of Poli's theoretically defined region for convergent behaviour of PSO parameters.

Parametry PSO (Harrison) [13]

CONTROL PARAMETER VALUES

ω	c_1	c_2
0.7298	1.49618	1.49618
0.729	2.0412	0.9477
0.6	1.7	1.7
0.721	1.193	1.193
0.715	1.7	1.7
0.724	1.468	1.468
0.785	1.331	1.331
0.837	1.255	1.255
0.42	1.55	1.55
0.711897	1.711897	1.711897
0.5	1.90	1.90
0.6	1.80	1.80
0.1	0.950	2.850
-0.1	0.875	2.625



Plan prezentacji

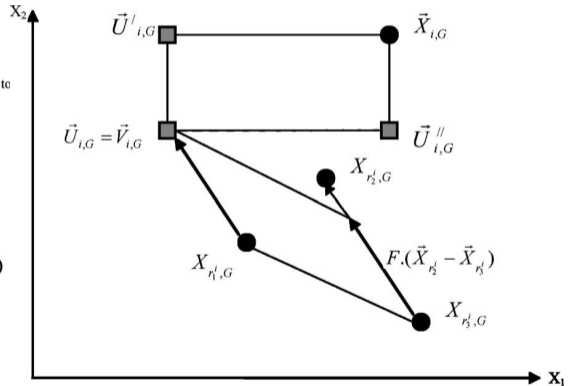
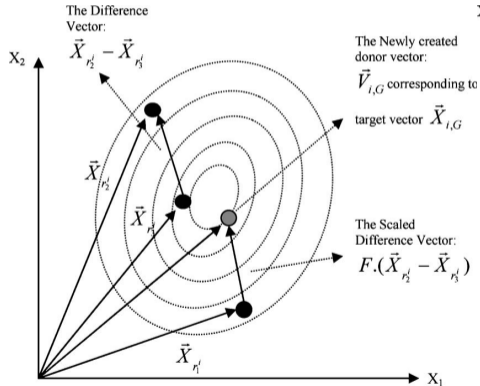
- 1 Wstęp
- 2 Podejścia ewolucyjne
 - Schemat ogólny
 - Algorytmy
 - Operatory krzyżowania i mutacji
- 3 Podejścia inteligencji rojowej
 - Zastosowanie w przemyśle rozrywkowym
 - Zastosowanie w optymalizacji
- 4 **Ewolucja Różnicowa**
 - Wariant bazowy
 - Modyfikacje

Ewolucja Różnicowa [3]

Definicja

Ewolucja Różnicowa (Differential Evolution) wprowadza w miejsce mutacji gausowskiej tzw. operator różnicowy. Położenie osobnika jest zmieniane zgodnie z wektorem różnicy w położeniu innych osobników.

Działanie operatorów w DE [9]



Ewolucja Różnicowa (DE/best/1/bin)

```
1:  $X \leftarrow X_{init}$ 
2: for  $iter \in \{1, 2, \dots, iter_{max}\}$  do
3:   for  $x \in X$  do
4:      $v \leftarrow x_{best} + F * (x_{r_1} - x_{r_2})$ 
5:      $u \leftarrow Crossover_{bin}(x, v)$ 
6:     if  $f(u) < f(x)$  then
7:        $x \leftarrow u$ 
8:     end if
9:   end for
10: end for
```

▷ Inicjalizacja połączenia

▷ Mutacja różnicowa

▷ Krzyżowanie wymieniające

▷ Aktualizacja połączenia

Kluczowe warianty metody DE

- DE/best/1/bin, DE/rand/1/bin [3]
- JADE [11]
- L-SHADE [15]

Praca domowa (podwójna za 20 pkt, do 11. maja 20:00 PL)

- Porównać działanie *Hill Climbingu*, strategii ewolucyjnych, algorytmu ewolucyjnego, optymalizacji rojem cząstek i ewolucji różnicowej
- Należy zwrócić uwagę na sprawiedliwe porównywanie metod jednopunktowych z populacyjnymi (nie wg iteracji, a wg ewaluacji)

Bibliografia I

- [1] Craig W Reynolds.
Flocks, herds and schools: A distributed behavioral model.
SIGGRAPH Comput. Graph., 21(4):25–34, 1987.
- [2] James Kennedy and Russell C. Eberhart.
Particle Swarm Optimization.
Proceedings of IEEE International Conference on Neural Networks. IV, pages 1942–1948, 1995.
- [3] Rainer Storn and Kenneth Price.
Differential Evolution – A Simple and Efficient Heuristic for global Optimization over Continuous Spaces.
Journal of Global Optimization, 11(4):341–359, 1997.

Bibliografia II

- [4] James Kennedy, Russell C. Eberhart, and Yuhui Shi.
Swarm Intelligence.
Swarm Intelligence, pages 133–185, 2001.
- [5] Hans-Georg Beyer and Hans-Paul Schwefel.
Evolution strategies—a comprehensive introduction.
Natural computing, 1(1):3–52, 2002.
- [6] Tim M Blackwell and Peter J Bentley.
Dynamic search with charged swarms.
In *Proceedings of the 4th Annual Conference on Genetic and Evolutionary Computation*, pages 19–26, 2002.

Bibliografia III

- [7] Rui Mendes, James Kennedy, and José Neves.
The Fully Informed Particle Swarm: Simpler, Maybe Better.
IEEE Transactions on Evolutionary Computation, 8(3):204–210, jun 2004.
- [8] Tim Blackwell and Jürgen Branke.
Multiswarms, exclusion, and anti-convergence in dynamic environments.
IEEE Transactions on Evolutionary Computation, 10(4):459–472, 08 2006.
- [9] Swagatam Das and Ponnuthurai Nagaratnam Suganthan.
Differential Evolution: A Survey of the State-of-the-Art.
IEEE Transactions on Evolutionary Computation, 15(1):4–31, feb 2011.
- [10] Maurice Clerc.
Standard particle swarm optimisation, 2012.

Bibliografia IV

- [11] Petr Poaík and Václav Klema.
JADE, an adaptive differential evolution algorithm, benchmarked on the BBOB noiseless testbed.
In Proceedings of the fourteenth international conference on Genetic and evolutionary computation conference companion - GECCO Companion '12, page 197, New York, New York, USA, 2012. ACM Press.
- [12] Moshe Szyf.
Lamarck revisited: epigenetic inheritance of ancestral odor fear conditioning.
Nature neuroscience, 17(1):2–4, 2014.
- [13] Kyle Robert Harrison, Andries P. Engelbrecht, and Beatrice M. Ombuki-Berman.
An adaptive particle swarm optimization algorithm based on optimal parameter regions.
In 2017 IEEE Symposium Series on Computational Intelligence (SSCI), pages 1–8. IEEE, nov 2017.

Bibliografia V

[14] Yan Wang, Huijie Liu, and Zhongsheng Sun.

Lamarck rises from his grave: parental environment-induced epigenetic inheritance in model organisms and humans.

Biological Reviews, 2017.

[15] Adam P. Piotrowski.

L-SHADE optimization algorithms with population-wide inertia.

Information Sciences, 468:117–141, nov 2018.