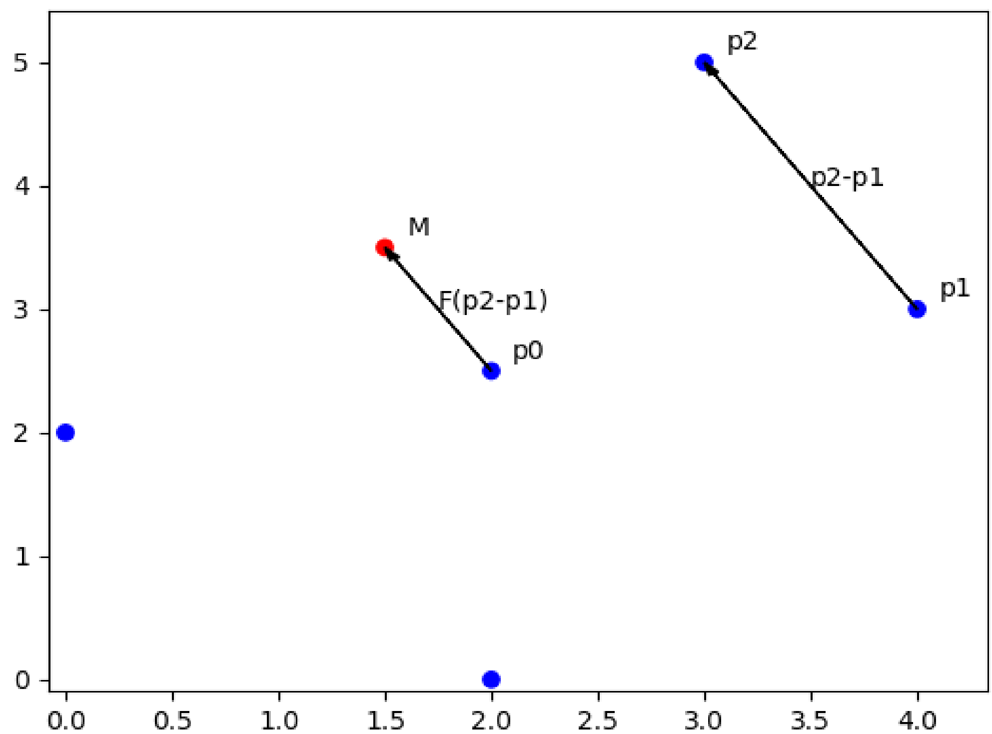
Ewolucja Różnicowa (Differential Evolution)

Skuteczna wersja algorytmu ewolucyjnego, którą można zastosować dla przestrzeni wektorów liczb rzeczywistych

Algorytm Ewolucji Różnicowej		
1.	Inicjuj $P^0 \leftarrow \{x_1, x_2 \dots x_\mu\}$	Inicjalizacja populacji początkowej P^0 μ osobnikami
2.	$H \leftarrow P^0$	Inicjalizacja historii H populacją początkową
3.	$t \leftarrow 0$	
4.	while !stop	
5.	for $(i \in 1{:}\mu)$	Dla każdego i-tego punktu w populacji:
6.	$r^t_i \leftarrow \text{select}(P^t)$	Wybór punktu roboczego r^t_i
7.	$d_i^t, e_i^t \leftarrow \text{sample}(P^t, 2)$	Wybór dwóch losowych punktów d_j^t i e_k^t na podstawie których będzie dokonana modyfikacja
8.	$M_i^t \leftarrow r^t_i + F(e_i^t - d_i^t)$	Nowy punkt M_i^t jest modyfikacją roboczego na podstawie różnicy między dwoma wylosowanymi
9.	$O_i^t \leftarrow \text{crossover}(r^t_i, M_i^t)$	Krzyżowanie zmodyfikowanego punktu z oryginalnym
10.	$H \leftarrow H \cup \{O_i^t\}$	Dodanie punktu do historii
11.	$P_i^{t+1} \leftarrow \text{tournament}(p_i^t, O_i^t)$	Aktualizacja populacji: podmiana i-tego punktu jeżeli nowy jest lepszy
12.	$t \leftarrow t + 1$	



Wyznaczanie nowego punktu jako M = p0+F(p2-p1) dla F = 0.5

Czyli wybrany punkt z populacji jest modyfikowany o przeskalowaną różnicę dwóch, możliwe że innych, punktów. Nowo utworzony punkt jest następnie krzyżowany i porównywany z kolejnym punktem z populacji i lepszy przechodzi do następnej populacji.

Warianty

Krzyżowanie

Typowe krzyżowania używane przy ewolucji różnicowej polegają na losowaniu czy dana współrzędna punktu O_i^t będzie pochodziła z punktu r^t_i czy M_i^t. Rodzaje krzyżowań:

1. Binarne – źródło każdej współrzędnej jest losowane niezależnie
2. Wykładnicze – losowany jest punkt przejścia między źródłami współrzędnych. Oznacza to że zaczyna mieć znaczenie kolejność wymiarów.

Krzyżowanie binarne	Krzyżowanie wykładnicze

binary_crossover(x, y):
  for($i \in 1$: n)
    if $a < c_r$
      $z_i \leftarrow y_i$
    else
      $z_i \leftarrow x_i$
  return z
	exponential_crossover(x, y):
	  $i \leftarrow 1$
	  while ($i \le n$)
	    if $a < c_r$
	      $z_i \leftarrow y_i$
	    else break
	  while ($i \le n$)
	    $z_i \leftarrow x_i$
	  return z

Gdzie a jest zmienną losową o rozkładzie jednostajnym z przedziału (0, 1) a c_r jest parametrem

Dla $c_r = 1$ zawsze zostanie wybrany y, więc nie będzie następowało krzyżowanie

Wybór punktu roboczego

1. Losowy – punkt roboczy jest wybrany losowo spośród całej populacji
2. Najlepszy – punkt roboczy jest najlepszym z całej populacji – jest więc ten sam dla każdego nowego punktu w jednej iteracji, nowe różnią się tylko wektorem $F(e_i^t - d_i^t)$

Liczba par używanych przy wyliczaniu nowego punktu

W krokach 7-8 może zostać wybrana więcej niż 1 para punktów – jeżeli używanych jest N par ten fragment algorytmu wygląda tak:

$M_i^t \leftarrow r^t_i$
do N times:
  $d_i^t, e_i^t \leftarrow \text{sample}(P^t, 2)$
  $M_i^t \leftarrow M_i^t + F(e_i^t - d_i^t)$

Notacja

Wariant ewolucji różnicowej oznacza się przy użyciu następującej notacji:

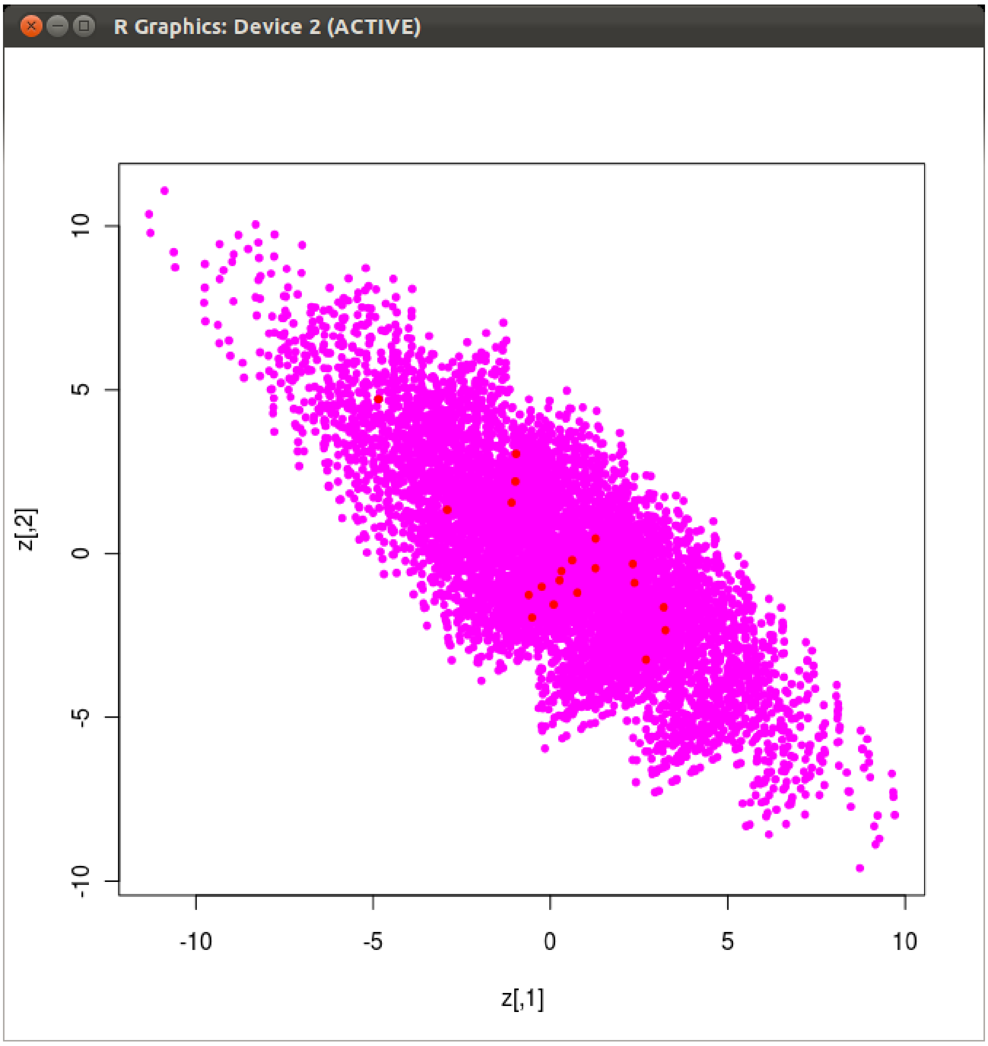
DE/<sposób wyboru punktu losowego>/<liczba krzyżowanych par>/<typ krzyżowania>
na przykład:

- DE/rand/1/bin – losowy punkt jako roboczy, jedna para punktów użyta do modyfikacji punktu roboczego, krzyżowanie binarne
- DE/best/2/exp – najlepszy punkt jako roboczy, dwie pary punktów użyte do modyfikacji punktu roboczego, krzyżowanie wykładnicze

Powyższa notacja nie przekazuje informacji o parametrach:

- Liczebności populacji μ
- Współczynnik skalujący wektor przesunięcia F
- Parametr krzyżowania c_r

Statystyka ewolucji różnicowej



Przykładowa populacja przed (czerwone) i po (różowe) mutacji dla DE/rand/1

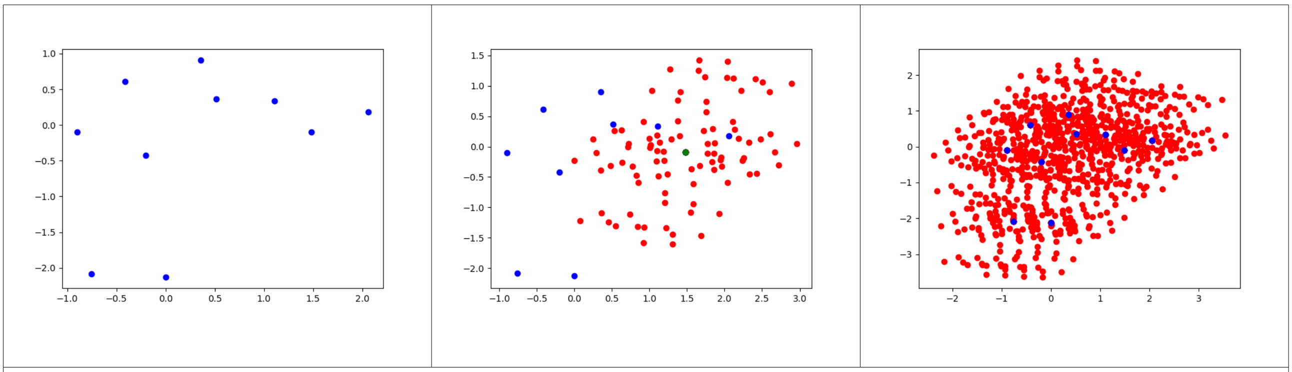
Warto zauważyć, że możliwe punkty wylosowane z kolejnej populacji są dyskretne – dla μ punktów w populacji jest maksymalnie μ^2 par punktów o których przeskalowaną różnicę można przesunąć punkt roboczy. Potencjalnych punktów roboczych jest μ, więc, nie licząc krzyżowania, nie da się wylosować więcej niż μ^3 punktów. Jest to różnica w stosunku do algorytmu ewolucyjnego, gdzie losując nowy punkt modyfikuje się losowo jego współrzędne – co daje nieskończenie wiele potencjalnych nowych punktów.

Jak również widać z powyższego obrazka, populację punktów można przybliżyć pewnym rozkładem zbliżonym do normalnego o pewnej wartości średniej oraz pewnej macierzy kowariancji.

Dla macierzy kowariancji punktów po selekcji v_P:

- Punkt startowy dla DE/rand/*/* jest wybrany z rozkładu z macierzą kowariancji v_P
- Punkty do wyznaczania wektora przesunięcia są wybierane z wariancją v_P a następnie wektor jest skalowany przez, F więc wektor przesunięcia dla DE/*/1/* ma macierz kowariancji $F^2(v_p + v_p) = 2F^2v_p$.

W DE/rand/1 daje to macierz kowariancji punktów po mutacji, przed krzyżowaniem $v_P + 2F^2v_p = (1+2F^2)v_p$.



Kolejne obrazki przedstawiają: populację 10 punktów, możliwe modyfikacje 1 przykładowego punktu populacji dla DE/*/1/*, F=0.5, możliwe mutacje całej populacji dla DE/rand/1/*. Kolejne macierze kowariancji punktów wyliczone eksperymentalnie:

$$\begin{bmatrix} 0.953 & 0.377 \\ 0.377 & 1.10 \end{bmatrix},$$

$$\begin{bmatrix} 0.433 & 0.171 \\ 0.171 & 0.501 \end{bmatrix} = 0.455 \begin{bmatrix} 0.953 & 0.377 \\ 0.377 & 1.10 \end{bmatrix},$$

$$\begin{bmatrix} 1.29 & 0.510 \\ 0.510 & 1.49 \end{bmatrix} = 1.35 \begin{bmatrix} 0.953 & 0.377 \\ 0.377 & 1.10 \end{bmatrix}$$

Właściwości DE jako metaheurystyki w $\mathbb{R}^n$

Uwaga – ewolucja różnicowa nie jest metaheurystyką w stricte sensie tego słowa ponieważ jest ograniczona do przestrzeni liczb rzeczywistych

Poinformowanie

Tak, przy podmianie punktu w populacji brana jest pod uwagę funkcja celu. Ponadto w wariancie DE/best/*/* jako punkt roboczy wybierany jest ten najlepszy.

Determinizm

Nie, losowane są punkty o których różnicę będzie zmodyfikowany punkt roboczy oraz krzyżowanie korzysta ze zmiennej losowej. Ponadto w wariancie DE/rand/*/* punkt roboczy jest wybierany losowo.

Typ stanu

Pamięciowy, stanem jest obecna populacja

Wielkość modelu

Liczność populacji

Lokalność generacji

Nie, maksymalna różnica odległości między nowym punktem a starym nie może przekroczyć maksymalnej różnicy odległości między punktami

Miękkość selekcji

Nie, jako że punkt w populacji może zostać zamieniony tylko na lepszy od siebie

Okno historii

Nieskończone, w obecnej populacji mogą znajdować się dowolne punkty z historii – w tym te którymi została ona zainicjalizowana

Zupełność

Nie, ani nie ma miękkości selekcji ani globalnej generacji punktów – na przykład jeżeli cała populacja znajdzie się w okolicy szczytu otoczonego względnie dużym obszarem o niższej wartości funkcji celu to nigdy z niego nie zejdzie

SHADE

SHADE jest jednym ze skuteczniejszych (wygrywających konkursy) algorytmów bazującym na ewolucji różnicowej, skupiającym się na dostosowywaniu parametrów F i c_r na podstawie poprzednich iteracji.

Nazwa jest skrótem od Success-History based Adaptive Differential Evolution.

Liczba parametrów jest ograniczona z 3 dla ewolucji różnicowej (liczność populacji, F i c_r) do 2: μ, czyli liczności populacji i H – liczba poprzednich iteracji branych pod uwagę przy ustalaniu bieżących wartości F i c_r.

Wartości F i c_r są losowane niezależnie dla każdego osobnika w populacji z rozkładów o parametrach bazujących na ich wartościach które pozwolły uzyskać poprawę we wcześniejszych iteracjach: dla każdej iteracji zapamiętuje się średnie wartości F oraz c_r które pozwoliły uzyskać poprawę, a ważone średnie tych średnich z H poprzednich iteracji są wartościami oczekiwanymi F oraz c_r w nowej iteracji.

Rozkłady używane do losowania wartości F oraz c_r to kolejno rozkład Cauchy’ego przeskalowany o 0.1 i rozkład normalny ze stałym odchyleniem standardowym 0.1.

W przeciwieństwie do ewolucji różnicowej pozwala to na osiągnięcie dowolnego punktu.

W algorytmie SHADE została też użyta inna metoda mutacji, określana jako “current-to-*p*best/1” – przypomina ona rand/2, ale pierwsza różnica nie jest między punktami losowymi a pomiędzy jednym z punktów najlepszych a punktem roboczym, co daje następujący wzór:

$$M_i^t \leftarrow r_i^t + F(x_{pbest} - r_i^t) + F(e_i^t - d_i^t)$$

i-tym punktem roboczym r_i^t w SHADE jest zawsze *i*-ty punkt w populacji, a punkty d^t_i e^t_i są losowane, jak w DE/rand/1

Punkt x_{pbest} jest losowany z rozkładem jednostajnym z μ*p najlepszych punktów populacji, gdzie p jest wartością losową, niezależną dla każdego losowania, losowaną z rozkładem jednostajnym z przedziału $[\frac{\mu}{2};2]$

Ponadto zostało nałożone ograniczenie, że wszystkie 4 punkty muszą być różnymi osobnikami z populacji.