Algorytm SHADE

Dawid Płudowski

Wydział Matematyki i Nauk Informatycznych

18-03-2023

Differential Evolution (DE)

Pomysł: Wylosujmy punkty w przestrzeni (populacja). Pozwólmy im, zgodnie z ustaloną strategią, losowo łączyć się ze sobą (mutacja) z każdą kolejną iteracją (generacja). Każda mutacja to dodanie do oryginalnego punktu przeskalowanego kierunku wyznaczonego poprzez inne punkty. To, czy mutacja zajdzie, jest wyznaczane losowo.

Notacja

- N - liczność populacji
- U - zbiór strategii mutacji
- F - skala mutacji
- CR - *Crossover rate*, częstość mutacji
- G - numer generacji mutacji

Przykładowe strategie

Częścią wspólną każdej strategii jest losowanie permutacji populacji $(r_1, \ldots, r_N)$. Zmutowane punkty oznaczamy $(\nu_1, \ldots, \nu_N)$ i wyznaczamy za pomocą strategii:

Strategie

- **rand/1**: $\nu_{i,G} = x_{r_1,G} + F(x_{r_2,G} - x_{r_3,G})$
- **best/1**: $\nu_{i,G} = x_{best,G} + F(x_{r_1,G} - x_{r_2,G})$
- **current-to-best/1**:
  $\nu_{i,G} = x_{i,G} + F(x_{best,G} - x_{i,G}) + F(x_{r_1,G} - x_{r_2,G})$

Crossover rate

Po dokonaniu mutacji decydujemy się, które z nich uwzględnimy w kolejnej generacji. **Uwaga**: decyzja dotyczy elementu wektora, nie całego punktu!

$$u_{j,i,G} = \begin{cases} \nu_{j,i,G}, & CR > unif(0,1) \\ x_{j,i,G}, & w.p.p \end{cases} \tag{1}$$

Jeżeli mutacja była dla nas pomyślna to ją zachowujemy, w przeciwnym wypadku powracamy do oryginalnego punktu:

$$x_{i,G+1} = \begin{cases} u_{i,G}, & f(u_{i,G}) < f(x_{i,G}) \\ x_{i,G}, & w.p.p \end{cases} \tag{2}$$

Problemy DE

Dla całego procesu wybieramy liczność (N), strategię (U_i), skalę mutacji (F) i częstość mutacji (CR). Parametry globalne mogą być w pewnych miejscach przestrzeni nieadekwatne, niektóre mogą charakteryzować się złym dobraniem do konkretnego problemu optymalizacyjnego.

JADE

JADE to prekursor algorytmu **SHADE**. Rozwiązuje problem globalnych parametrów przydzielając dla każdego osobnika z osobna dynamicznie parametry mutacji.

Nowa strategia

current-to-*p*best/1: $\nu_{i,G} = x_{i,G} + F(x_{pbest,G} - x_{i,G}) + \mathrm{F}(\mathrm{x}_{r_1,G} - x_{r_2,G})$

x_{pbest} oznacza losowo wybrany punkt z p najlepszych.

Dynamiczna zmiana parametrów

Dla każdego osobnika z populacji stosowany jest algorytm:

- w pierwszej iteracji zainicjalizuj $F_i = 0.5$ i $CR_i = 0.5$
- wyznacz średnią wartość $mean_L(F)$ i $mean_A(CR)$ dla tych osobników, których mutacja zakończyła się sukcesem (nie została odrzucona w ostatnim kroku)
- wyznacz średnią wartość $\mu_{F_G} = c\mu_{F_{G-1}} + (1 - c)mean_L(F)$ i analogicznie μ_{CR} [n - tzw. *learning_rate*, parametr algorytmu]
- wylosuj nowe F_i i CR_i z rozkładów $Cauchy(\mu_F, 0.1)$ i $N(\mu_{CV}, 0.1)$ odpowiednio
- powtarzaj kroki 2-4 w kolejnych iteracjach

$mean_A$ oznacza średnią arytmentyczną, a $mean_L$ średnią Lehmera.

Opcjonalne archiwum

W tradycyjnym alogrytmie podczas każdej generacji "usuwamy" poprzednią populację. Jeżeli korzystamy z archiwum (P), osobnik, który nie ptrzetrwał (znaleźliśmy lepszy punkt na jego podstawie), trafia do archiwum. Punkt $x_{r_2,G}$ z poprzedniego slajdu losowany jest z sumy zbioru X i P. Co daje nam taka strategia:

- zachowujemy punkty, które mogłyby być potencjalnie obiecujące, ale źle zmutowały
- zwiększamy wielkość zbioru, z którego losujemy mutacje (większa różnorodność)

Problemy JADE

W algorytmie JADE dobieramy dynamicznie parametry na podstawie podpopulacji, której udało się ewoluować z sukcesem. Probabilistycznie możliwe jest, że taka podpopulacja miała bardzo złe parametry i jej "sukces" był przypadkowy. Potrzebujemy algorytmu, który będzie odporny na pojedyncze "szczęśliwe" generacje, które bazują na złych parametrach.

SHADE

SHADE (*Success History DE*) rozwija algroytm JADE o zachowywanie informacji o poprzednich sukcesach.

- jeżeli parametry sprawdziły się w pewnej generacji, kolejne (nawet odległe) generacje mogą z nich ponownie skorzystać
- "szczęsliwe" parametry, które w rzeczywistości nie są odpowiednie do zadania nie determinują kolejnych generacji
- zwiększenie entropii algrytmu - mniejsza szansa na zatrzymanie się na niekorzystnych parametrach

Pamięć poprzednich generacji

Zamiast:

$$\mu_F, \ \mu_{CR}$$

Zdefiniujmy:

$$M_F = (M_{F,1}, \ldots, M_{F,H}), \ M_{CR} = (M_{CR,1}, \ldots, M_{CR,H})$$

Podczas tworzenia nowej generacji wybierzmy parametry μ_F, μ_{CR} ze zbiorów M_F, M_{CR}. Wyznaczmy, tak samo jak w JADE, nowe parametry F i CR i nadpiszmy nim $M_{F,k}$, $M_{CR,k}$, gdzie $k = mod(G, H)$.

Wyznaczanie średniej średniej

Zamiast wyznaczać zwykłą średnią, możemy nadać jej odpowiednie wagi, które będą faworyzowały szczególnie "wybitne" osobniki, tzn. te, które w trakcie trwania generacji najbardziej zmniejszyły wartość $f(x)$.

Średnia ważona

$$mean_{WA}(S_{CR}) = \Sigma_{k=1}^{|S_{CR}|} \omega_k * S_{CR,k}$$

$$mean_{WL}(S_F) = \frac{\Sigma_{k=1}^{|S_{CR}|} \omega_k * S_{F,k}^2}{\Sigma_{k=1}^{|S_{CR}|} \omega_k * S_{F,k}}$$

$$\omega_k = \frac{\Delta f_k}{\Sigma_{i=1}^{|S_{CR}|} \Delta f_i}$$

Dynamiczna zmiana current-to-$p$best

Z jednej strony zmiana w kierunku najlepszego punktu jest intuicyjnie uzasadniona; z drugiej strony ryzykujemy, że najlepszy punkt utknął w minimum lokalnym i cała reszta populacji również tam trafi. Kompromisem pomiędzy tymi podejściami jest losowanie p dla każdego punktu z pewnego rozkładu $rand[p_{min}, 0.2]$.

Benchamark

F	SHADE Mean (Std Dev)	CoDE Mean (Std Dev)	EPSDE Mean (Std Dev)	JADE Mean (Std Dev)	dynNP-jDE Mean (Std Dev)
F_1	**0.00e+00 (0.00e+00)**	**0.00e+00 (0.00e+00)**≈	**0.00e+00 (0.00e+00)**≈	**0.00e+00 (0.00e+00)**≈	**0.00e+00 (0.00e+00)**≈
F_2	9.00e+03 (7.47e+03)	9.78e+04 (4.81e+04)−	1.37e+06 (5.23e+06)−	**7.67e+03 (5.66e+03)**≈	9.52e+04 (4.09e+04)−
F_3	**4.02e+01 (2.13e+02)**	1.08e+06 (3.03e+06)−	1.75e+08 (5.39e+08)−	4.71e+05 (2.35e+06)−	1.71e+06 (2.54e+06)−
F_4	**1.92e-04 (3.01e-04)**	8.18e-02 (1.09e-01)−	8.08e+03 (2.56e+04)−	6.09e+03 (1.33e+04)−	4.76e+01 (4.75e+01)−
F_5	**0.00e+00 (0.00e+00)**	**0.00e+00 (0.00e+00)**≈	**0.00e+00 (0.00e+00)**≈	**0.00e+00 (0.00e+00)**≈	**0.00e+00 (0.00e+00)**≈
F_6	**5.96e-01 (3.73e+00)**	4.16e+00 (9.00e+00)−	9.27e+00 (1.33e+00)−	2.07e+00 (7.17e+00)≈	1.19e+01 (1.66e+00)−
F_7	4.60e+00 (5.39e+00)	9.32e+00 (6.34e+00)−	5.88e+01 (4.29e+01)−	3.16e+00 (4.13e+00)≈	**2.62e+00 (1.59e+00)**≈
F_8	**2.07e+01 (1.76e-01)**	2.08e+01 (1.18e-01)≈	2.09e+01 (5.32e-02)−	2.09e+01 (4.93e-02)−	2.10e+01 (3.98e-02)−
F_9	2.75e+01 (1.77e+00)	**1.45e+01 (2.90e+00)**+	3.50e+01 (4.21e+00)−	2.65e+01 (1.96e+00)+	2.20e+01 (5.12e+00)+
F_{10}	7.69e-02 (3.58e-02)	**2.71e-02 (1.50e-02)**+	1.02e-01 (5.65e-02)−	4.04e-02 (2.37e-02)+	3.63e-02 (2.34e-02)+
F_{11}	**0.00e+00 (0.00e+00)**	**0.00e+00 (0.00e+00)**≈	1.95e-02 (1.39e-01)≈	**0.00e+00 (0.00e+00)**≈	**0.00e+00 (0.00e+00)**≈
F_{12}	2.30e+01 (3.73e+00)	3.98e+01 (1.21e+01)−	4.94e+01 (9.28e+00)−	**2.29e+01 (5.45e+00)**≈	4.07e+01 (8.81e+00)−
F_{13}	5.03e+01 (1.34e+01)	8.04e+01 (2.74e+01)−	7.68e+01 (1.72e+01)−	**4.67e+01 (1.37e+01)**≈	7.10e+01 (1.72e+01)−
F_{14}	3.18e-02 (2.33e-02)	3.60e+00 (4.09e+00)−	3.99e-01 (6.00e-01)−	2.86e-02 (2.53e-02)≈	**9.39e-03 (1.40e-02)**+
F_{15}	**3.22e+03 (2.64e+02)**	3.36e+03 (5.31e+02)−	6.75e+03 (7.60e+02)−	3.24e+03 (3.17e+02)≈	4.39e+03 (4.72e+02)−
F_{16}	9.13e-01 (1.85e-01)	**3.38e-01 (2.03e-01)**+	2.48e+00 (2.88e-01)−	1.84e+00 (6.27e-01)−	2.32e+00 (2.83e-01)−
F_{17}	**3.04e+01 (3.83e-14)**	3.04e+01 (1.17e-02)−	3.04e+01 (2.51e-02)−	3.04e+01 (1.95e-14)−	3.04e+01 (1.78e-03)≈
F_{18}	7.25e+01 (5.58e+00)	**6.69e+01 (9.23e+00)**+	1.37e+02 (1.12e+01)−	7.76e+01 (5.91e+00)−	1.35e+02 (1.24e+01)−
F_{19}	1.36e+00 (1.20e-01)	1.61e+00 (3.58e-01)−	1.84e+00 (2.00e-01)−	1.44e+00 (8.71e-02)−	**1.27e+00 (1.09e-01)**+
F_{20}	1.05e+01 (6.04e-01)	1.06e+01 (6.69e-01)≈	1.30e+01 (6.33e-01)−	**1.04e+01 (5.82e-01)**≈	1.13e+01 (4.14e-01)−
F_{21}	3.09e+02 (5.65e+01)	3.02e+02 (9.02e+01)≈	3.05e+02 (8.06e+01)≈	3.04e+02 (6.68e+01)≈	**2.94e+02 (8.29e+01)**≈
F_{22}	9.81e+01 (2.52e+01)	1.17e+02 (9.96e+00)−	3.09e+02 (1.12e+02)−	**9.39e+01 (3.08e+01)**≈	1.03e+02 (2.57e+01)−
F_{23}	3.51e+03 (4.11e+02)	3.56e+03 (6.12e+02)≈	6.74e+03 (8.20e+02)−	**3.36e+03 (4.01e+02)**≈	4.36e+03 (4.61e+02)−
F_{24}	2.05e+02 (5.29e+00)	2.21e+02 (9.28e+00)−	2.91e+02 (7.08e+00)−	2.17e+02 (1.57e+01)−	**2.04e+02 (4.22e+00)**≈
F_{25}	2.59e+02 (1.96e+01)	2.57e+02 (6.55e+00)≈	2.99e+02 (3.29e+00)−	2.74e+02 (1.06e+01)−	**2.55e+02 (7.91e+00)**≈
F_{26}	2.02e+02 (1.48e+01)	2.18e+02 (4.48e+01)−	3.56e+02 (6.49e+01)−	2.15e+02 (4.11e+01)≈	**2.00e+02 (3.06e-03)**+
F_{27}	**3.88e+02 (1.09e+02)**	6.20e+02 (1.01e+02)−	1.21e+03 (7.42e+01)−	6.70e+02 (2.40e+02)−	3.90e+02 (9.12e+01)≈
F_{28}	**3.00e+02 (0.00e+00)**	**3.00e+02 (0.00e+00)**≈	**3.00e+02 (0.00e+00)**≈	**3.00e+02 (0.00e+00)**≈	**3.00e+02 (0.00e+00)**≈
	+	4	0	2	5
	-	15	23	10	13
	≈	9	5	16	10

Rysunek: Benchmark algorytmów.

Bibliografia

- *Success-History Based Parameter Adaptation for Differential Evolution*, Ryoji Tanabe, Alex Fukunaga, http://metahack.org/CEC2013-SHADE.pdf