Statistic

for machine learning

Tran Trong Khiem

AI lab tranning

2024/05/29

1. Differential Evolution
2. JADE
3. Multiple populations DE

Introduction

Differential Evolution (DE) is a novel parallel direct search:

- Population for each generation G as $\{x_{i,G}\}_0^{\mathrm{NP}-1}$
- Size of population does't change during optimization process.
- Generates new trail vector by calculate the weighted sum of three different members.
- $x_{\text{best },\mathrm{G}}$ is evaluated for every generation G in order to keep track of the optimization progress.
- Basics scheme:
  1. Scheme DE1
  2. Scheme DE2

Scheme DE1

For each vector $x_{i,G}$, new vector v is generated according to:

$$v = x_{r_1,G} + F(x_{r_2,G} - x_{r_3,G})$$

- $r_1, r_2, r_3 \in \{0, 1, 2, \ldots, \mathrm{NP} - 1\}$, integer and mutually different.
- F is a real and constant factor.

In order to increase the diversity of the parameter vectors, the child vector $u = \left(u_1, u_2, \ldots, u_D\right)^T$.

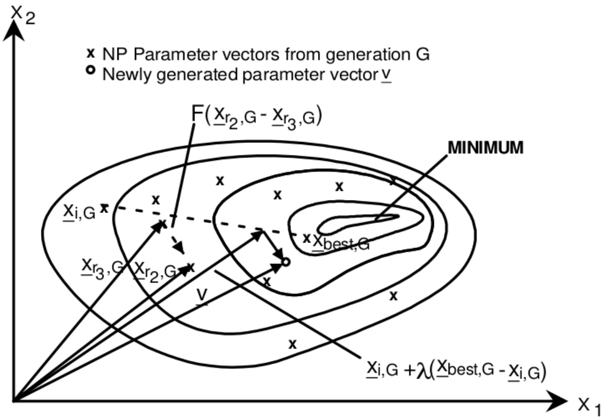
- Choose n random from $[0, D - 1]$.
- L is drawn from the interval $[0, D - 1]$ with the probability $\Pr(L = \nu) = (\mathrm{CR})^\nu$, where $\mathrm{CR} \in [0, 1]$.

$$u_j = \begin{cases} v_j, & \text{for } j = n \bmod D, (n+1) \bmod D, \cdots (n+L-1) \bmod D \\ (x_{i,G})_j, & \text{otherwise} \end{cases}$$

Scheme DE2

Basically, scheme DE2 works the same way as DE1 but generates the vector v according to

$$\mathbf{v} = \mathbf{x}_{i,G} + \lambda \cdot \left(\mathbf{x}_{\text{best},G} - \mathbf{x}_{i,G}\right) + F \cdot \left(\mathbf{x}_{r2,G} - \mathbf{x}_{r3,G}\right)$$



JADE: Adaptive differential evolution with optional external archive

For each vector $x_{i,G}$, new vector v is generated as:

$$\mathbf{v} = \mathbf{x}_{i,G} + \lambda \cdot (\mathbf{x}_{\text{pbest},G} - \mathbf{x}_{i,G}) + F \cdot (\mathbf{x}_{r2,G} - \mathbf{x}_{r3,G})$$

x_i is associated with its own CR_i and F_i parameters $CR_i \sim \mathcal{N}(\mu_{CR}, 0.1)$ and $F_i \sim \mathcal{C}(\mu_F, 0.1)$.

- If CR_i is generated outside of the interval $[0, 1]$, it is replaced by the limit value (0 or 1) closest to the generated value.
- When $F_i > 1$, F_i is truncated to 1, and when $F_i \leq 0$, the sampling is repeatedly applied to try to generate a valid value.
- $x_{\text{pbest},G}$ is randomly selected from the top $N \times p$.

At the end of the generation,if v is better than x, then CR_i and F_i are recoded as S_{CR} and S_F, μ_{CR} and μ_F are updated as:

$$\mu_{CR} = (1 - c) \cdot \mu_{CR} + c \cdot \text{mean}_A(S_{CR})$$

$$\mu_F = (1 - c) \cdot \mu_F + c \cdot \text{mean}_L(S_F)$$

Appendix

Lehmer mean is computed as:

$$\text{mean}_L(S_F) = \frac{\sum_F F^2}{\sum_F F}$$

Weighted mean is computed as:

$$\text{mean}_W(S_{CR}) = \sum_k w_k \cdot S_{CR,k}$$

$$w_k = \frac{\Delta f_k}{\sum_k \Delta f_k}$$

SHADE:Success-History Based Parameter Adaptation for Differential Evolution

SHADE maintains a historical memory with H entries for both of the DE control parameters CR and F , M_{CR}, M_F . In each generation, for each x_i :

- Select index r_i randomly from the interval $[1, H]$.
- sample $CR_i \sim \mathcal{N}(M_{CR,r_i}, 0.1)$ and $F_i \sim \mathcal{C}(M_{C,r_i}, 0.1)$ and $p_{i,G} = \text{rand}[p_{\min}, 0.2]$
- Generate new trail vector as JADE.

If $u_{i,G}$ better than $x_{i,G}$:

- CR_i and F_i are recorded in S_{CR} and S_F.
- The contents of memory are updated as follows:

$$M_{CR,k,G+1} = \begin{cases} \text{mean}_{AW}(S_{CR}) & \text{if } S_{CR} \neq \emptyset \\ M_{CR,k,G} & \text{otherwise} \end{cases}$$

Update the same for $M_{F,k,G+1}$

1. Differential Evolution
2. JADE
3. Multiple populations DE

Multiple population DE

Ideal: Combine Scheme DE1, SchemeDE2, JADE and SHADE to generate multiple populations. Then choose the best solution from them.

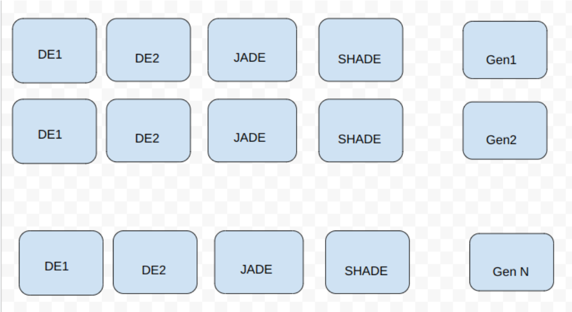


Figure 1: Multiple populations DE