Chaotic differential evolutionary algorithm

algorithm

---

chaotic systems

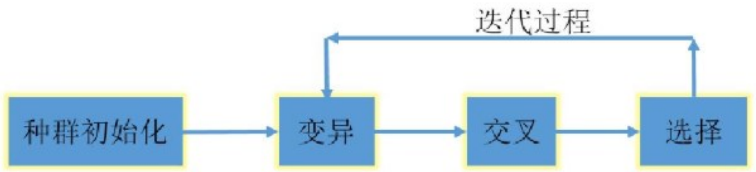
这里展示3个chaotic system

- logistic map
$$ch_{k+1} = uch_k(1-ch_k), ch_k \in [0,1], k=0,1,2,\ldots,K \tag{1}$$
- mapping drawn form chaotic neuron
$$ch_{k+1} = \eta ch_k - 2\tanh(\gamma ch_k)\exp[-3(ch_k)^2], ch_k \in [0,1], k=0,1,2,\ldots,K \tag{2}$$
- tent mapping
$$ch_{k+1} = \begin{cases} \frac{ch_k}{\alpha}, & ch_k \in [0,\alpha], \\ (1-\alpha)(1-ch_k), & ch_k \in (\alpha,1] \end{cases} \quad k=0,1,2,\ldots,K \tag{3}$$

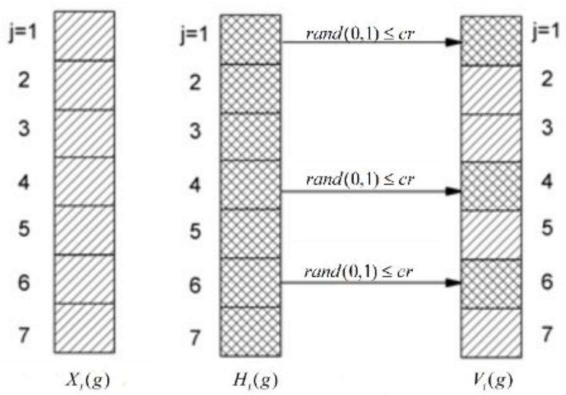
k是迭代计数，K预设最大值，u和γ 是控制参数，$\eta \in [0,1]$,在式（1）中，设定$\mu=4, ch_0 \notin \{0, 0.25, 0.5, 0.75, 1\}$,在式(2)中，$ch_0 \notin \{0, 0.25, 0.5, 0.75, 1\}$，这样便可利用(1)(2)(3)迭代，无规律，随机地分布式搜索。

Differential evolution algorithm

https://blog.csdn.net/qq_37423198/article/details/77856744]
这里讲的很详细，我也参考着写一下算法过程。
差分进化算法：类似于遗传算法，是其的一种改进。
算法过程



- 初始化
随机产生M个个体一，每个个体由n维向量组成（也说成染色体）
$$X_i(0) = (x_{i,1}(0), x_{i,2}(0), x_{i,3}(0)), \ldots, x_{i,n}(0), i=1,2,3,\ldots,M$$
第i个个体的第j维取值方式如下：
$$X_{i,j}(0) = L_{j\_min} + rand(0,1)(L_{j\_max} - L_{j\_min}), i=1,2,3,\ldots,M \ j=1,2,3,\ldots,n$$
- 变异
在第g次迭代中，从种群中随机选择3个个体$Xp_1(g), Xp_2(g), Xp_3(g)$,且p1≠p2≠p3≠i，生成的变异向量为：
$$H_i(g) = X_{p1}(g) + F \cdot (X_{p2}(g) - X_{p3}(g))$$
这里的F是缩放因子
- 交叉
$$v_{i,j} = \begin{cases} h_{ij}(g), rand(0,1) \leq cr \\ x_{ij}(g), else \end{cases}$$
其中，$cr \in [0,1]$ 为交叉概率



- 选择

$$x_i(g+1)=\begin{cases} v_i(g),\ if\ f(v_i(g))<f(X_i(g)) \\ x_i(g),\ else \end{cases}$$

- 参数
  f 是适应度函数，M一般介于$5\times n$到$10\times n$之间， 不能少于4;F一般在$[0,2]$ 间选择，通常取0.5,cr 一般在[0,1]间选择，一般选0.3

- 参数自适应调整
  将随机选择的个体从优到劣排序，得到X_b, X_m, X_w ,对应适应度f_b, f_m, f_w ,变异算子改为：

$$V_i=X_b+F_i\cdot(X_m-X_w)$$

  F的自适应变化：

$$F_i=F_l+(F_u-F_l)\cdot\frac{f_m-f_b}{f_w-f_b}$$

  其中，$F_l=0.1, F_u=0.9$

$$cr_i=\begin{cases} cr_l+(cr_u-cr_l)\frac{f_i-f_{min}}{f_{max}-f_{min}} & \text{if} \quad f_i>\bar{f} \\ cr_l & \text{if} \quad f_i<\bar{f} \end{cases}$$

其中f_i是个体X_i的适应度，f_{min}和f_{max}分别是当前种群中最差和最优个体的适应度，$\bar{f}$是当前种群适应度平均值，cr_l和cr_u分别是cr的下限和上限，一般$cr_l=0.1, cr_u=0.6$。

变异策略表示为$DE/a/b$，其中a表明被变异个体的选择方式，b表明差向量的个数。

1. **DE/rand/1:**
   $$V_i=X_{p1}+F\left(X_{p2}-X_{p3}\right)$$
2. **DE/best/1:**
   $$V_i=X_{best}+F\left(X_{p1}-X_{p2}\right)$$
3. **DE/current to best/1:**
   $$V_i=X_i+F\left(X_{best}-X_i\right)+F\left(X_{p1}-X_{p2}\right)$$
4. **DE/best/2:**
   $$V_i=X_{best}+F\left(X_{p1}-X_{p2}\right)+F\left(X_{p3}-X_{p4}\right)$$
5. **DE/rand/2:**
   $$V_i=X_{p1}+F\left(X_{p2}-X_{p3}\right)+F\left(X_{p4}-X_{p5}\right)$$

cpde

使用chaotic system初始化

Algorithm 2 (*Chaotically initializing population*)

Step 0. Set the maximum number of chaotic iteration $K \geqslant 300$, the population scale NP, and the individual counter $i = 0$.

While $(i \leqslant NP)$ **do**

Step 1. Randomly initialize variables $ch_0^j \in (0,1)$, $ch_0^j \notin \{0.25, 0.5, 0.75\}$, $j = 1,2,\ldots,n$ and set iteration counter $k = 0$.

Step 2. **While** $(k < K)$ **do**

Generate different chaotic variables $ch_k^j, j = 1,2,\ldots,n$ according to the formula (1), (2) or (3). Set $k = k + 1$.

End while.

Step 3. Mapping the chaotic variables ch_k^j to feasible region according to equation $x_{ij}^{(0)} = x_{\min,j} + ch_k^j(x_{\max,j} - x_{\min,j})$, $j = 1,2,\ldots,n$.

Step 4. Set $i = i + 1$

End while

Note that in Algorithm 2, a point (individual) in feasible region is generated through one chaotic system through K cycles of Step 1 to Step 3. And a population with NP individuals is formed after NP cycles of Step 1 to 4.

cpde算法

chaotic部分：初始化，在交叉后用chaotic产生的个体替换差的那一半个体，再用pattern search方法对最优个体进行优化，再使用获得的个体替换当前最着的那一半个体。

Algorithm 3 (*The new algorithm* (*CPDE*))

Step 0. Preset the population size, NP, the maximum number of iteration, *ktoal* within DE algorithm, and the maximum iteration $TT_{\max}$. Set $T = 0$.

Step 1. Chaotically initialize the population (see Algorithm 2) and evaluate it.

Step 2. **While** ($TT_{\max}$ or solution with preset precision not reached) **do**

Step 2.1. Execute first selection, mutation, crossover, second selection of DE search in Section 2.3 for *ktoal* iterations.

Step 2.2. Execute the pattern search in Section 2.2 with the best individual in the population as the initial point. Assume x^* is obtained.

Step 2.3. Use the chaotically initializing population method (Algorithm 2) to generate a subpopulation S_{sub} with scale $NP/2$, Use S_{sub} to replace the worst half part of the original population, while the best half part keeps steady in the population.

Step 2.4. Use x^* obtained in Step 2.2 to replace the worst individual in the current population and form a new population.

Step 2.5. Set $T = T + 1$ and turn to Step 2.1

End while

Step 3. Output the best results.

引用：

Wang Y J , Zhang J S . Global optimization by an improved differential evolutionary algorithm.[J]. Applied Mathematics & Computation, 2007, 188(1):669-680.