进化算法简介

XXX XXX XXX XXX XXX XXX

March 21, 2018

浙江大学计算机科学与技术学院

目录

前言

定义

又叫演化计算，是模拟自然界中的生物的演化过程产生的一种群体导向的随机搜索技术和方法。

是一种通用的问题求解方法，具有自组织、自适应、自学习性和本质并行性等特点，不受搜索空间限制性条件的约束，也不需要其它辅助信息。

思想

进化算法是受生物进化过程中“优胜劣汰”的自然选择机制和遗传信息的传递规律的影响，通过程序迭代模拟这一过程，把要解决的问题看作环境，在一些可能的解组成的种群中，通过自然演化寻求最优解。

种类

- 遗传算法 (Genetic Algorithms)

种类

- 遗传算法 (Genetic Algorithms)
- 演化策略 (Evolution Strategy)

种类

- 遗传算法 (Genetic Algorithms)
- 演化策略 (Evolution Strategy)
- 演化规划 (Evolution Programming)

种类

- 遗传算法 (Genetic Algorithms)
- 演化策略 (Evolution Strategy)
- 演化规划 (Evolution Programming)
- 遗传程序设计 (Genetic Programming)

种类

- 遗传算法 (Genetic Algorithms)
- 演化策略 (Evolution Strategy)
- 演化规划 (Evolution Programming)
- 遗传程序设计 (Genetic Programming)
- 多种群协同进化 (multi-species cooperative)

种类

- 遗传算法 (Genetic Algorithms)
- 演化策略 (Evolution Strategy)
- 演化规划 (Evolution Programming)
- 遗传程序设计 (Genetic Programming)
- 多种群协同进化 (multi-species cooperative)
- 差分进化算法 (Differential Evolutionary)

演化规划 (EPA)

演化规划

定义

定义

演化规划 (Evolutionary Programming Algorithm, EPA) 是由美国学者 Lawrence J. Fogel 于 1960 年提出的，它适用于解决目标函数或约束条件不可微的复杂非线性实值连续优化问题。它与遗传算法类似，但要优化的程序结构是固定的，而其数值参数则可以进化。

思想

EP 模拟生物种群层次上的进化，因此在进化过程中主要强调生物种群行为上的联系，即强调种群层次上的行为进化而建立父、子代间的行为链。

EP 算法最重要的一个操作是变异操作。通过变异，父代群体中的每一个个体产生一个子代个体，父代和子代中最好的那一半被选择生存下来。

Algorithm 1: Evolutionary Programming Algorithm

Input: 个体表现型 X, 群体规模 N, 迭代次数 G 等

Output: 子代

随机产生初始群体并计算适应值 (含 N 个个体)

while *not done* **do**

  //终止条件：达到规定的进化代数，或若干代内种群中最好个体的函数值不再发生变化，则终止进化

  for $\mathrm{i} = 1; \mathrm{i} < \mathrm{N}; \mathrm{i}++$ **do**

    对 X_i 进行变异得到 X_i'

    对 X_i 进行可行性检查

    计算 X_i 的适应值

  end

  从 $2N$ 个个体中选择 N 个个体 //随机型 q-竞争法

end

return 子代;

q-竞争选择算法

演化规划算法的选择策略采用的是 q-竞争机制，这也是与进化策略算法最大的不同点，q-竞争说白了就是选择优质解的同时，以一定的随机概率接受较差的解。多数是优解，少数是比较差的解，共同组成一组父代，为下一次进化做准备。

q-竞争选择算法

思想

将 N 个父代进化的 N 个子代一起放在一起，从中随机选择不重复 q 个个体组成一个组，然后依次对 2N 个个体的每一个个体进行计分，将 2N 个依次每次一个与随机挑选出的群组的每一个成员进行比较，相比优的话，则会对应的个体的分数加 1，最后对分数进行排序，选择分数最高的 N 个个体！

缺点

大量的研究证明，由于过度选择、变异操作破坏有效模式、参数选取不当等原因，该算法存在一些亟待克服的**缺点**：

（1）容易出现早熟收敛现象

（2）进化后期，个体之间的竞争趋缓导致算法后期的搜索效率降低等。

改进

- 基于柯西变异的进化规划算法 1996

改进

- 基于柯西变异的进化规划算法 1996
- 基于 Lévy 概率分布的进化规划算法 (LEP) 2004

改进

- 基于柯西变异的进化规划算法 1996
- 基于 Lévy 概率分布的进化规划算法 (LEP) 2004
- 一种求解数值优化问题的快速进化规划算法 2004

改进

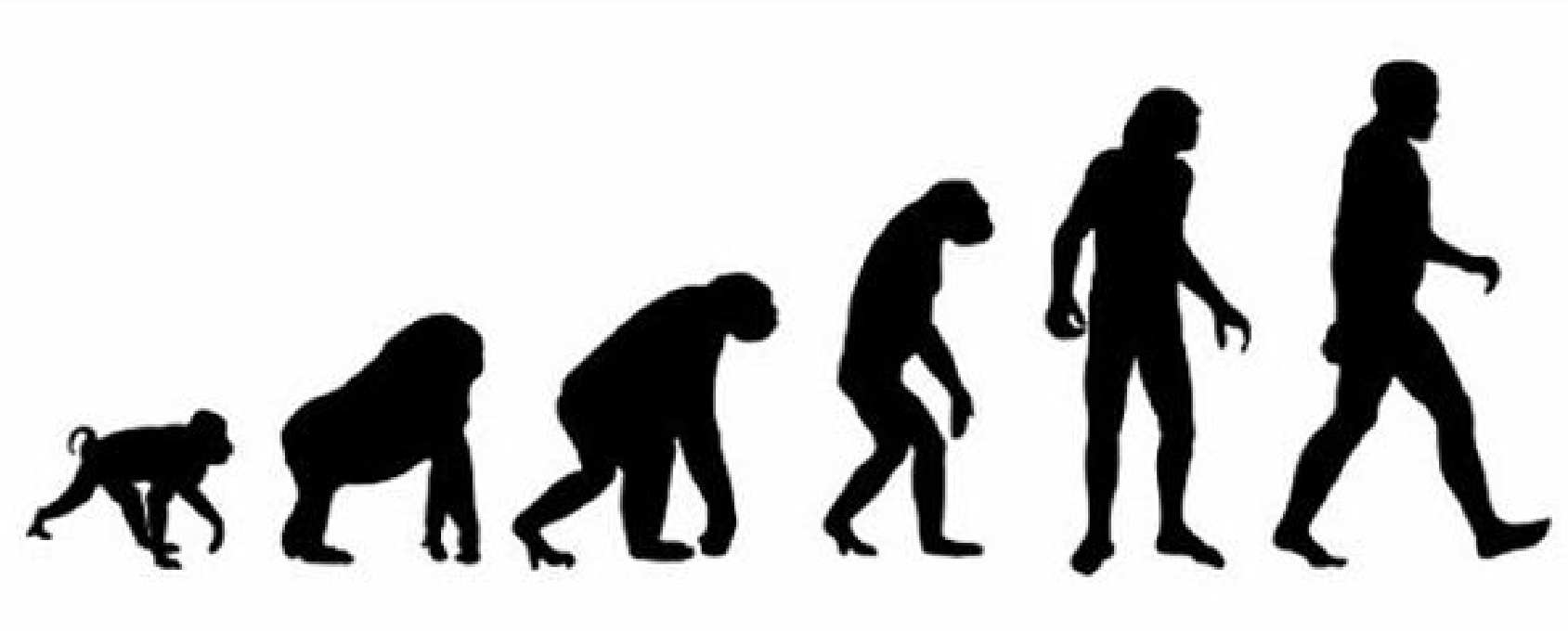
- 基于柯西变异的进化规划算法 1996
- 基于 Lévy 概率分布的进化规划算法 (LEP) 2004
- 一种求解数值优化问题的快速进化规划算法 2004
- ...

差分进化

定义

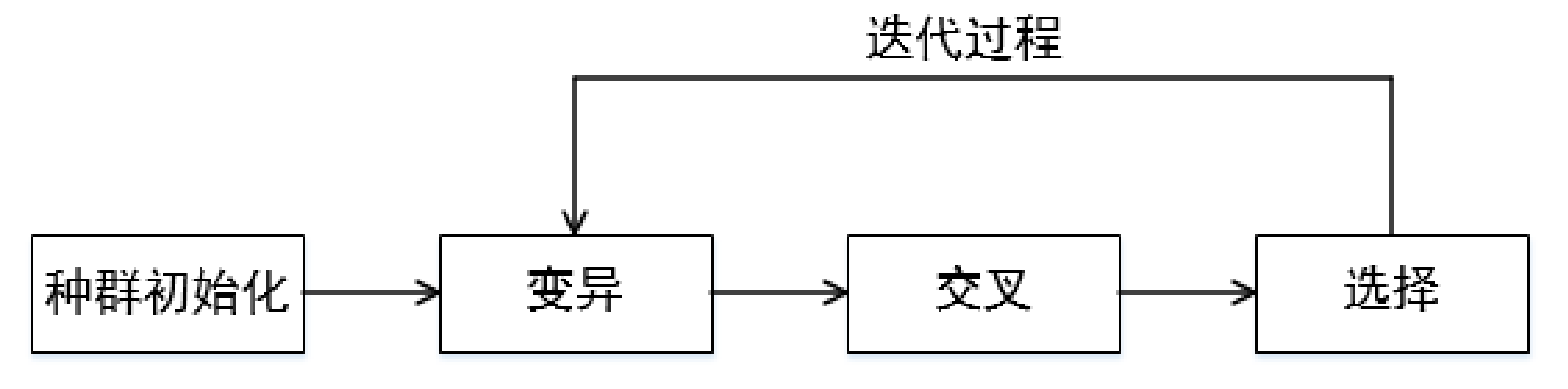
是一种基于群体智能的全局优化方法，其主要通过种群内个体之间的协同合作和相互竞争来产生群体智能，进一步指导进化过程的全局搜索。

算法思想



通过种群之间的**个体差异**和**优胜劣汰**的竞争策略产生新的个体，最终使种群接近或达到全局最优解。

算法框架



- **种群初始化**在解空间中随机、均匀地产生 M 个个体，每个个体由 n 个染色体组成，作为第 0 代种群，标记为

$$X_i(0) = (x_{i,1}(0), x_{i,2}(0), \cdots, x_{i,n}(0))$$

$$i = 1, 2, \cdots, M$$

- **变异、交叉、选择**三步操作迭代执行，直到算法收敛。第 g 次迭代的第 i 个个体标记为

$$X_i(g) = (x_{i,1}(g), x_{i,2}(g), \cdots, x_{i,n}(g))$$

$$i = 1, 2, \cdots, M$$

种群初始化

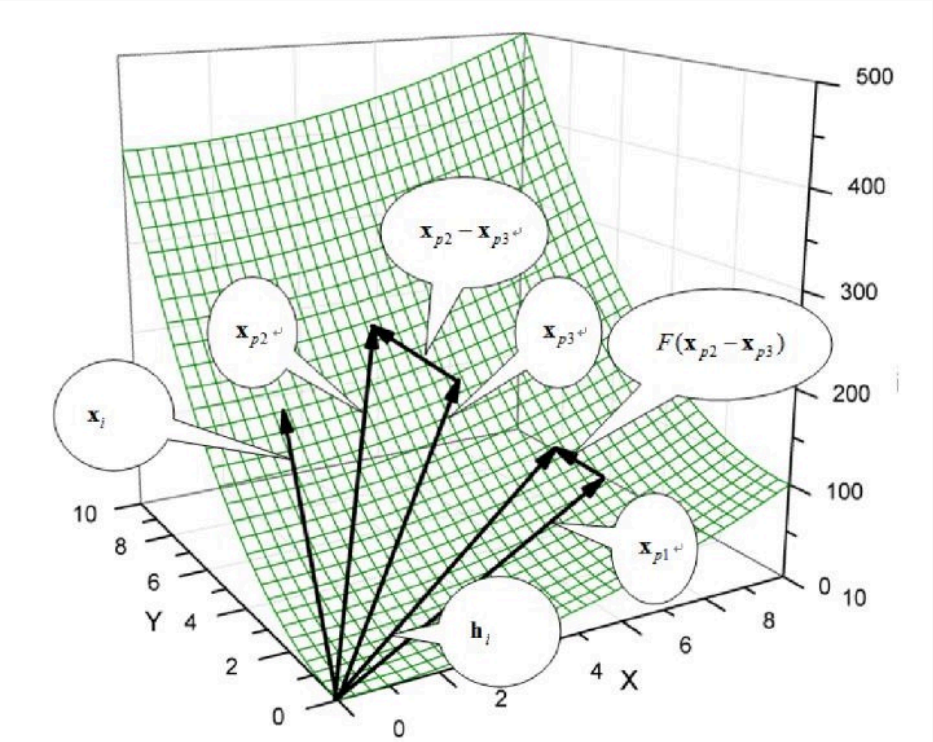
在 n 维空间里随机产生满足约束条件的 M 个染色体，第 i 个染色体的第 j 个维取值方式如下 (rand(0,1) 产生 0 到 1 的均匀分布的随机数)：

$$x_{i,j}(0) = L_j + rand(0,1)\,(U_j - L_j)$$
$$i = 1, 2, \cdots, M$$
$$j = 1, 2, \cdots, n$$

变异算子

在第 g 次迭代中，对个体 $X_i(g) = (x_{i,1}(g), x_{i,2}(g), \cdots, x_{i,n}(g))$，从种群中随机选择 3 个个体 $X_{p1}(g), X_{p2}(g), X_{p3}(g)$，且 $p1 \neq p2 \neq p3 \neq i$，则

$$H_i(g) = X_{p1}(g) + F \cdot (X_{p2}(g) - X_{p3}(g))$$

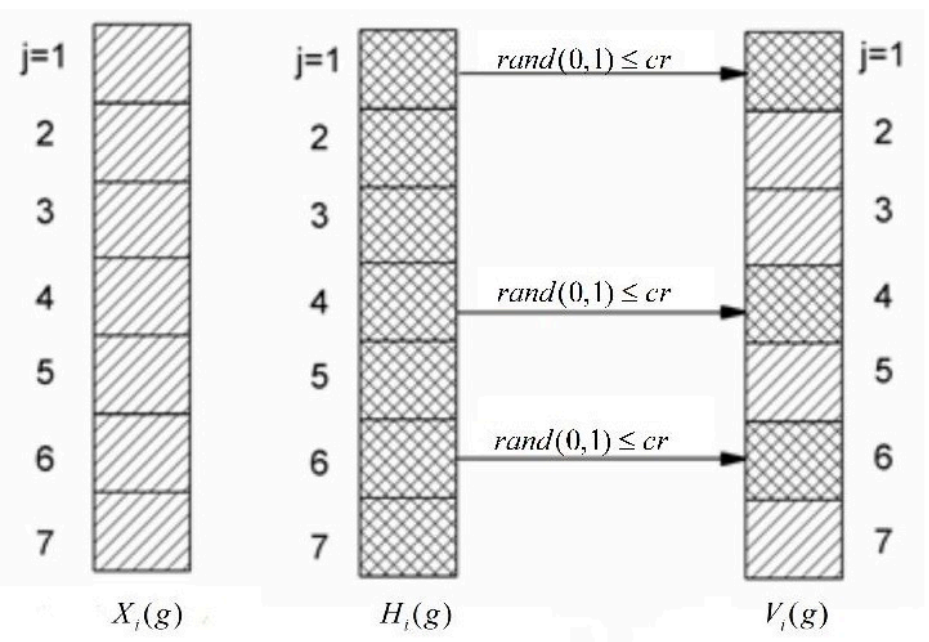


交叉算子

交叉操作可以增加种群的多样性，方法如下：

$$v_{i,j}(g) = \begin{cases} h_{i,j}(g), rand(0,1) \leq cr \\ x_{i,j}(g), else \end{cases}$$

其中 $cr \in [0,1]$ 为交叉概率，$rand(0,1)$ 是 $[0,1]$ 上服从均匀分布的随机数。



选择算子

首先查看根据评价函数选择 $V_i(g)$ 或 $X_i(g)$ 作为 $X_i(g+1)$

$$X_i(g+1) = \begin{cases} V_i(g), & \text{if} \quad f(V_i(g)) < f(X_i(g)) \\ X_i(g), & else \end{cases}$$

可以看出：

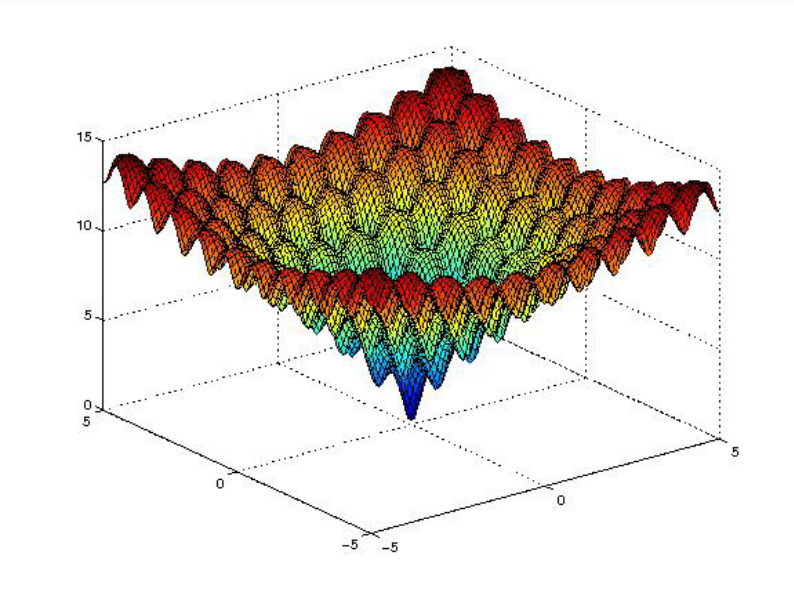
- 对每个个体，$X_i(g+1)$ 要好于或持平 $X_i(g)$。
- 肯定会收敛于最优点（可能是局部最优）。
- **变异、交叉** 操作有助于突破局部最优到达全局最优。

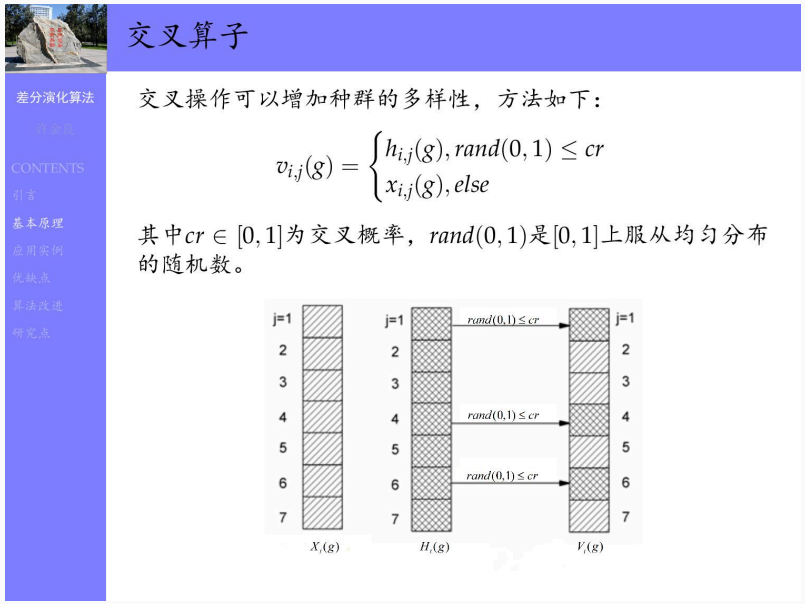
差分进化算法寻找函数最优解

定义关于参数 x, y 的函数，函数图像如左图所示

$$f(x,y) = -20e^{-0.2\sqrt{\frac{x^2+y^2}{2}}} - e^{\frac{\cos 2\pi x + \cos 2\pi y}{2}} + 20 + e$$

用差分进化算法求解，效果如右图所示（参数设置：$N = 20, F = 0.5, cr = 0.5$，迭代次数 $T = 300$）





优缺点

和其他进化算法相比，差分进化算法具有以下优点：

1. 在非凸、多峰、非线性、连续不可微函数优化问题上表现出极强的稳健性。
2. 收敛速度快。
3. 操作简单，容易实现。

缺点：

1. 算法后期个体间差异逐渐缩小，收敛速度慢，容易陷入局部最优。
2. 控制参数和学习策略对算法性能有着重要的影响，并且高度依赖于优化问题的本质。
3. 有时需要过多的迭代才能搜索到全局最优。

参数的选取

- M：一般介于 $5 \times n$ 与 $10 \times n$ 之间，但不能少于 4，否则变异算子无法进行；
- F：一般在 $[0, 2]$ 之间选择，通常取 0.5；
- cr：一般在 $[0, 1]$ 之间选择，比较好的选择应在 0.3 左右。cr 取值偏大，收敛速度会加快，但易发生早熟现象。

参数的自适应调整 (F)

将变异算子中随机选择的三个个体进行从优到劣的排序，得到 X_b, X_m, X_w，对应适应度 f_b, f_m, f_w，则变异算子改为：

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

同时，F 的取值根据生成差分向量的两个个体自适应变化，平衡全局搜索和局部搜索之间的矛盾。

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

其中，

$$F_l = 0.1, F_u = 0.9$$

参数的自适应调整 (cr)

对于适应度好的解，取较小的 cr，使得该解进入下一代的机会增大；对于适应度差的解，则取交大的 cr，加快改变该个体的结构，使该解被淘汰掉。

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

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

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