5. 差分进化

堵威

华东理工大学 自动化系

2021.4.15

回顾

- **(μ+1)进化策略：** 每一代用μ个父代，组合形成1个子代，从μ个父代和这个子代中选出最好的μ个父代作为下一代的父代；离散性交叉、中间性交叉、离散全局交叉、中间全局交叉
- **(μ+λ)进化策略：** 每一代用μ个父代，组合形成λ个子代，精英
- **(μ, λ)进化策略：** 每一代用μ个父代，组合形成λ个子代，λ≥μ
  - 当适应度函数有噪声或是时变的，(μ, λ)-ES常常比(μ+λ)-ES好
  - 对于搜索空间无界的问题：推荐(μ, λ)-ES
  - 对于搜索空间离散的问题：推荐(μ+λ)-ES

回顾

- **自身自适应进化策略：**

$$\sigma_i' \leftarrow \sigma_i' \exp(\tau' \rho_0 + \tau \rho_i) \qquad x_i' \leftarrow x_i' + \sigma_i' r_i$$

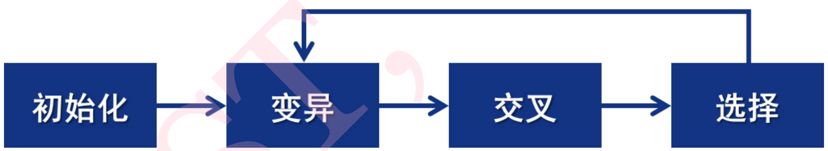
– **ρ_0，ρ_i有可能为正数和负数，导致指数会大于1或小于1，即σ'会增大或减小**

$$\tau = P_1 \left(\sqrt{2\sqrt{n}}\right)^{-1} \qquad \tau' = P_2 \left(\sqrt{2n}\right)^{-1}$$

- GA：对候选解编码，ES：连续参数上操作
- GA：强调交叉，ES：强调变异

回顾

- **差分进化：** 针对连续优化问题提出的基于种群差异的元启发式随机搜索算法
- **基本差分进化：** 变异、交叉、选择



- **变异：** 取两个个体之间的差分向量，将这个差分向量的一个伸缩版加到第三个个体上从而产生一个新的候选解

$$\mathbf{v}_i \leftarrow \mathbf{x}_{r1} + F(\mathbf{x}_{r2}-\mathbf{x}_{r3})$$

回顾

- **交叉：** 在建立变异向量v_i之后，它与一个差分进化个体x_i结合，产生一个试验向量u_i，按如下方式实施交叉：对于j∈[1,n]，

$$u_{ij} = \begin{cases} v_{ij} & \text{if } (r_{cj} < c) \text{ or } (j = \mathcal{J}_r) \\ x_{ij} & \text{otherwise} \end{cases}$$

- **选择：** 在生成试验向量u_i之后，比较向量u_i和x_i，适应性更强的向量留下来作为差分进化的下一代
- **调试参数：** 种群N，尺度参数F，交叉率c；参数设定的经验值：Storn和Price建议N的取值为[5D，10D]，50,100；F的取值[0.4,0.9]；c的取值[0.1,1]

回顾

- **命名：** DE/x/y/z；DE：differential evolution；x：被扰动的基向量；y：扰动x的差分向量个数；z：交叉的方式

- **变异向量的变种：**

变异向量：DE/rand/1, DE/rand/2, DE/best/1, DE/best/2, DE/target/1, DE/target/2, DE/target-to-best/1(DE/current-to-best/1)

试验向量：DE/x/y/bin, DE/x/y/L

尺度参数：颤振（每一个个体）、抖动（每一个个体的每一维）

本章内容

1. 背景知识

2. 基本差分进化

3. 差分进化变种

4. 离散优化

5. 差分进化与遗传算法

差分进化变种

- jDE
  - Janez Brest等人2006年提出差分进化变体 (IEEE Transactions on Evolutionary Computation, 2006, 10(6))
  - 主要思想：通过对尺度参数F和交叉率c进行自身自适应调整，从而实现对经典差分进化的改进
  - 参数控制方式：确定性(deterministic) & 自适应(adaptive) & 自身自适应(self-adaptive)

确定性：用特定的方式调整参数（不同的问题，同样的变化）

自适应：利用搜索中的反馈确定参数的调整的方向/大小

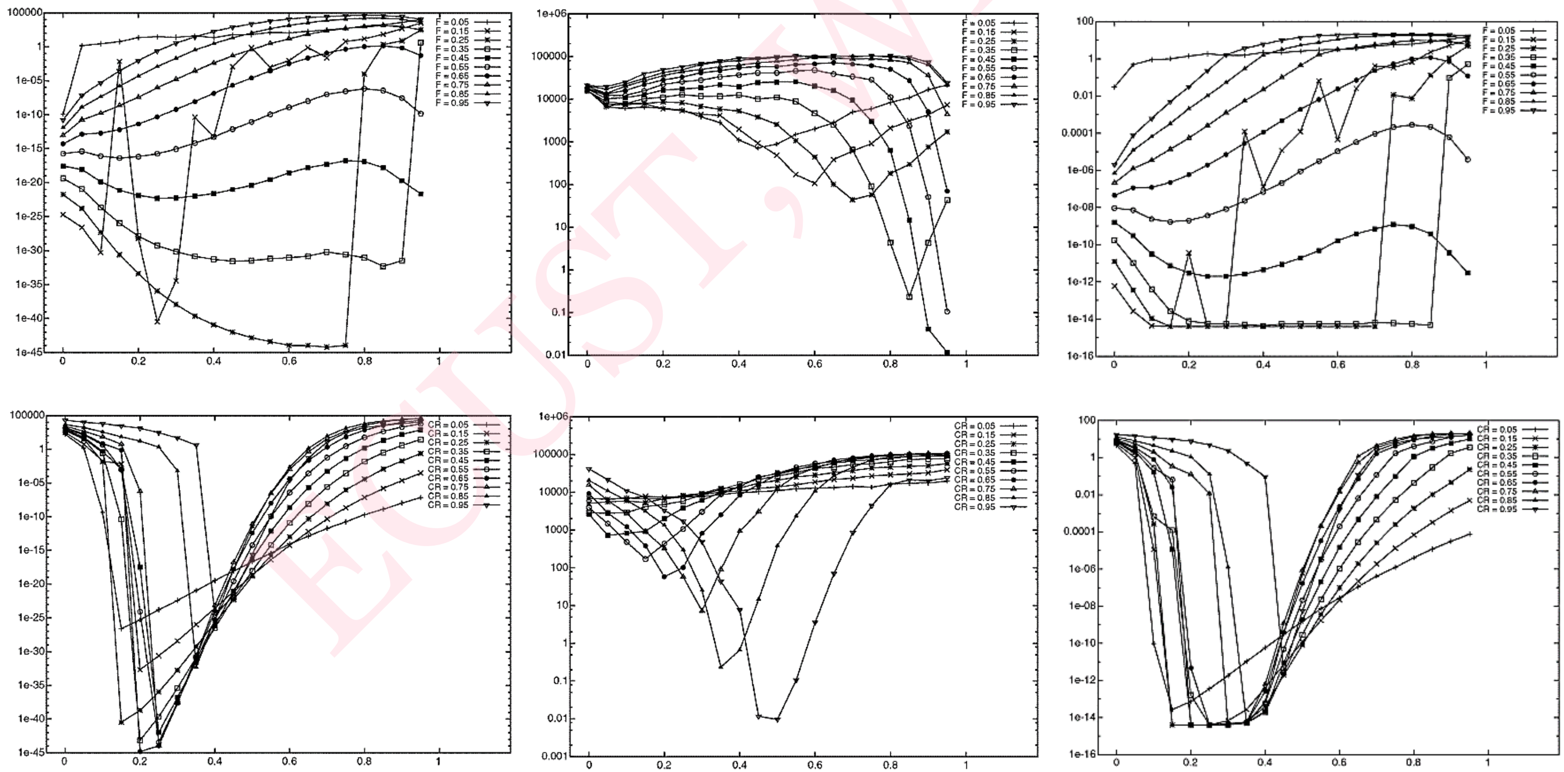
自身自适应：参数自身也参与进化

差分进化变种

- jDE
  - 在不同优化问题中，差分进化中尺度参数F和交叉率c的变化

F

c



差分进化变种

- jDE
  - jDE中使用了自身自适应的参数控制方式，将尺度参数F和交叉率c在种群中进行编码：

$\mathbf{x}_{1,G}$	$F_{1,G}$	$c_{1,G}$
$\mathbf{x}_{2,G}$	$F_{2,G}$	$c_{1,G}$
…	…	…
$\mathbf{x}_{N,G}$	$F_{N,G}$	$c_{N,G}$

N为种群大小，G表示进化代数。

差分进化变种

- jDE
  - jDE中使用了自身自适应的参数控制方式，将尺度参数F和交叉率c在种群中进行编码：

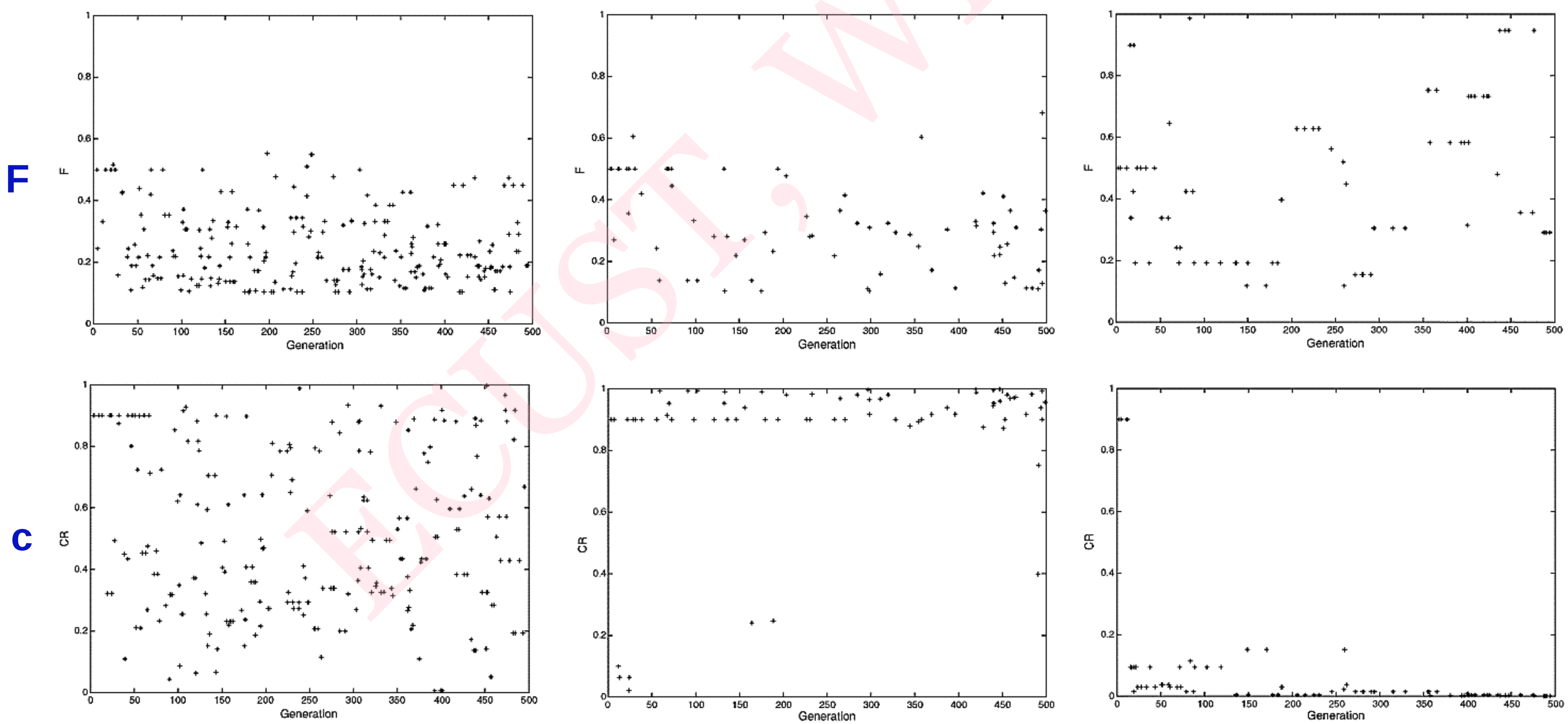
$$F_{i,G+1}=\begin{cases} F_l + rand_1 * F_u, & \text{if } rand_2 < \tau_1 \\ F_{i,G}, & \text{otherwise} \end{cases}$$

$$c_{\ i,G+1}=\begin{cases} rand_3, & \text{if } rand_4 < \tau_2 \\ c_{\ i,G}, & \text{otherwise} \end{cases}$$

其中，$rand_j$, $j\in\{1,2,3,4\}$是[0,1]区间的均匀随机数；τ_1和τ_2为控制F和c的概率，设为$\tau_1=0.1$，$\tau_2=0.1$；$F_l=0.1$，$F_u=0.9$。

差分进化变种

- jDE
  - 实验：观测不同优化问题中，F和c的变化



差分进化变种

- JADE
  - Zhang等人于2009年提出的自适应差分进化 (差分进化变体、IEEE Transactions on Evolutionary Computation, 2009, 13(5))
  - 主要创新：
  1. 利用了种群存档 (archive)
  2. 提出了新的变异算子DE/current-to-*p*best
  3. 尺度参数F和交叉率c的自适应更新
  - 是非常经典的差分进化变体，迄今为止已被广泛应用于求解众多应用优化问题

差分进化变种

- JADE
  - 利用archive，提出变异算子：DE/current-to-*p*best
  - DE/target-to-best/1或DE/current-to-best/1

$$\mathbf{v}_i \leftarrow \mathbf{x}_i + F(\mathbf{x}_b-\mathbf{x}_i)+F(\mathbf{x}_{r2}-\mathbf{x}_{r3})$$

  - 差分进化的第一个变异算子：DE/rand/1
  - 利用最好个体的信息：DE/best/1，DE/best/2，DE/current-to-best/1，......

利用种群中最好个体的信息，能够有效地加速种群的收敛，但是也会导致种群的早熟。

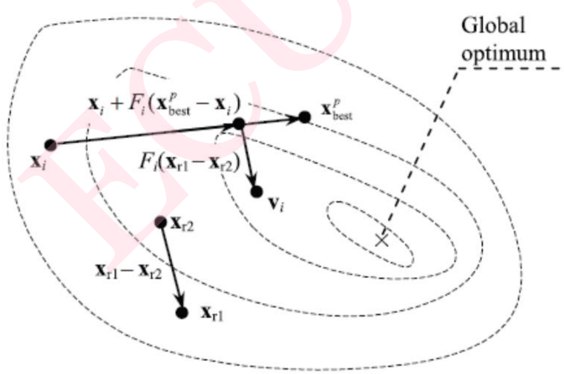
  - 利用最好的一部分个体的信息？

差分进化变种

- JADE
  - DE/current-to-*p*best

$$\mathbf{v}_{i,g} = \mathbf{x}_{i,g} + F_i(\mathbf{x}^p_{\text{best},g} - \mathbf{x}_{i,g}) + F_i(\mathbf{x}_{r1,g} - \tilde{\mathbf{x}}_{r2,g})$$

  - $\mathbf{x}^p_{\text{best},g}$ ： 利用了每一代最好的100*p*%个个体，p∈(0,1]
  - $\tilde{\mathbf{x}}_{r2,g}$ ： 从种群**P**和archive **A**并集中（**P**∪**A**）随机选择，其中archive A保存了搜索过的种群中差的个体；**A**的尺寸



DE/current-to-*p*best算子搜索示意图

差分进化变种

- JADE
  - 尺度参数F和交叉率c的自适应更新

尺度参数F的更新：

$$F_i = \mathrm{randc}_i(\mu_F, 0.1)$$

若$F_i \geq 1$，则截断取1；若$F_i \leq 0$，则重新生成F_i。

柯西分布的位置参数μ_F初始化为0.5，在每一代后按照下式更新：

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

$c \in [0,1]$，$\mathrm{mean}_L(\cdot)$是Lehmer平均，S_F是第g代中成功变异的F集合：

$$\mathrm{mean}_L(S_F) = \frac{\sum_{F \in S_F} F^2}{\sum_{F \in S_F} F}$$

差分进化变种

- JADE
  - 尺度参数F和交叉率c的自适应更新

交叉率c的更新：

$$c_i = \mathrm{randn}_i(\mu_c, 0.1)$$

在[0,1]区间内截断生成。

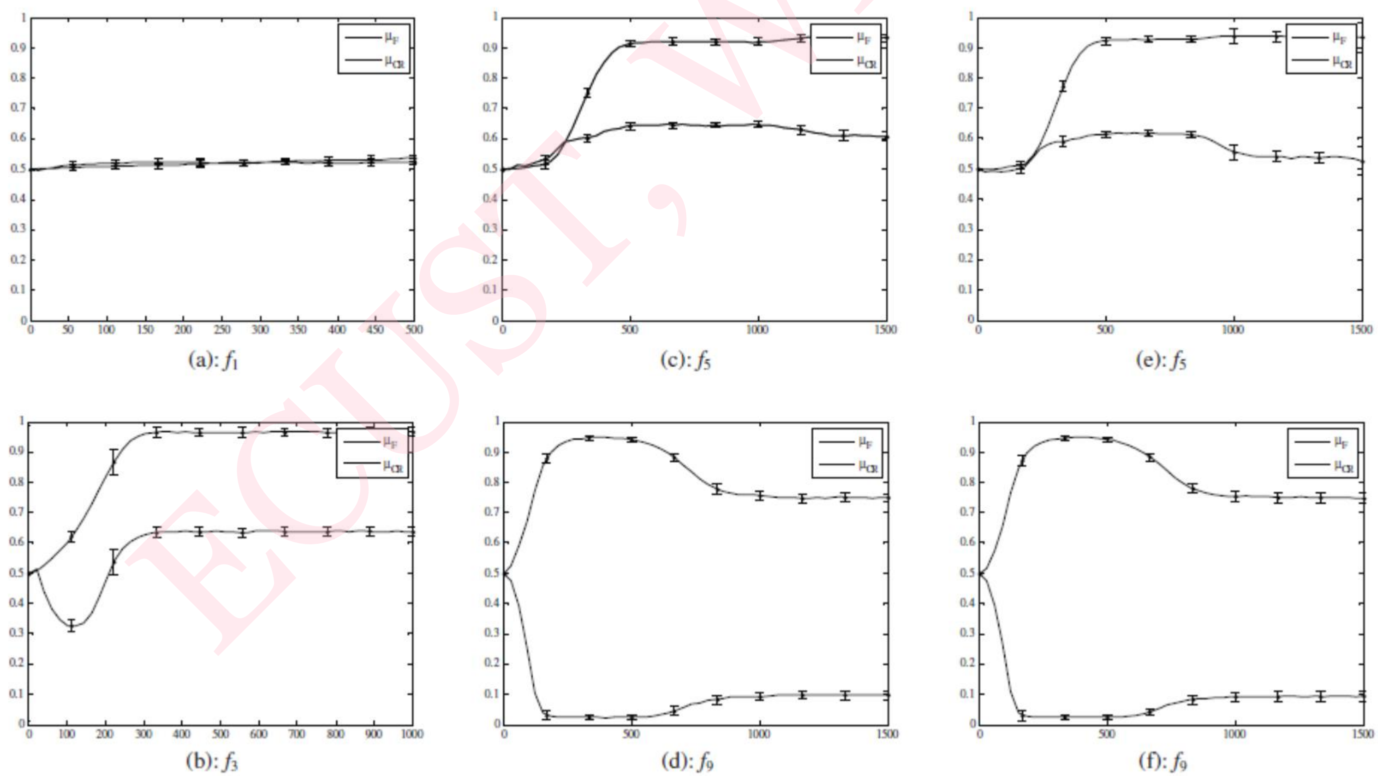
高斯分布的位置参数μ_c初始化为0.5，在每一代后按照下式更新：

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

$c\in[0,1]$， $\mathrm{mean}_A(\cdot)$是算术平均。

差分进化变种

- JADE
  - 尺度参数F和交叉率c的自适应更新



(a): f_1 (c): f_5 (e): f_5

(b): f_3 (d): f_9 (f): f_9

差分进化变种

- CoDE
  - Wang等人于2011年提出的差分进化变体 (IEEE Transactions on Evolutionary Computation, 2011, 15(1))
  - **主要创新：**

1. Composite DE
2. 多个试验向量生成策略组合
3. 多个控制参数（尺度参数F、交叉率c）的组合

  - 是非常经典的差分进化变体，迄今为止已被广泛应用于求解众多应用优化问题

差分进化变种

- CoDE
  - 试验向量生成策略：变异算子+交叉算子

DE/rand/1：　$\mathbf{v}_i \leftarrow \mathbf{x}_{r1} + F(\mathbf{x}_{r2}-\mathbf{x}_{r3})$

DE/best/1：　$\mathbf{v}_i \leftarrow \mathbf{x}_{best} + F(\mathbf{x}_{r2}-\mathbf{x}_{r3})$

DE/rand/2：　$\mathbf{v}_i \leftarrow \mathbf{x}_{r1} + F(\mathbf{x}_{r2}-\mathbf{x}_{r3}) + F(\mathbf{x}_{r4}-\mathbf{x}_{r5})$

DE/best/2：　$\mathbf{v}_i \leftarrow \mathbf{x}_{best} + F(\mathbf{x}_{r2}-\mathbf{x}_{r3}) + F(\mathbf{x}_{r4}-\mathbf{x}_{r5})$

DE/current-to-best/1：$\mathbf{v}_i \leftarrow \mathbf{x}_i + F(\mathbf{x}_b-\mathbf{x}_i)+F(\mathbf{x}_{r2}-\mathbf{x}_{r3})$

二项式交叉

差分进化变种

- CoDE
  - 控制参数设置：尺度参数F、交叉率c

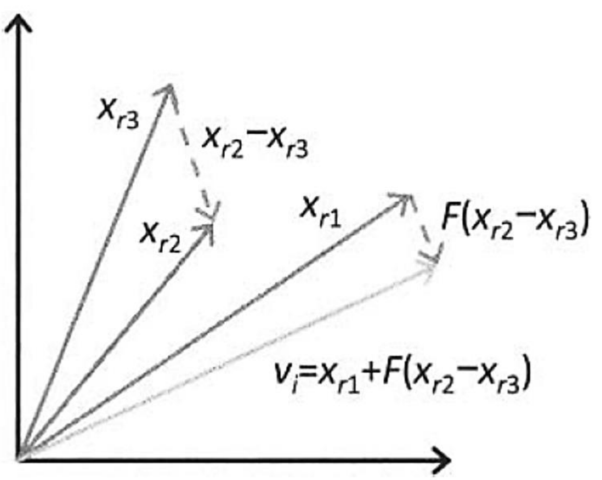
    F的取值[0.4,0.9]；c的取值[0.1,1]

    F大、F小：作用？

    c大、c小：作用？

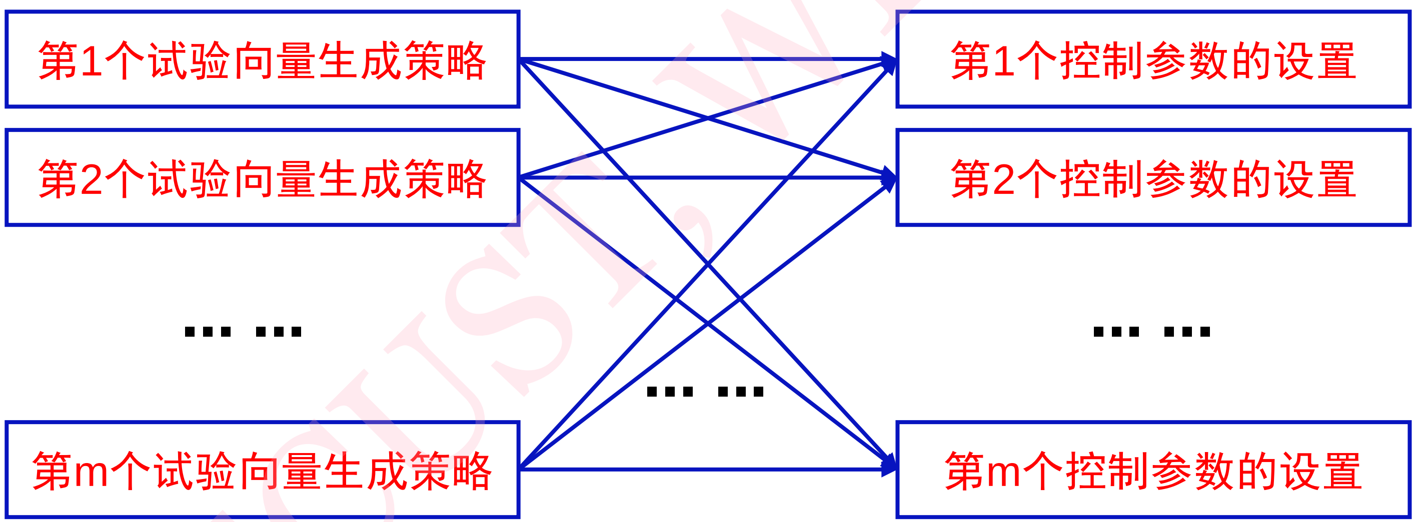
    共识：F的取值0.4~1

    c的取值靠近0或靠近1



差分进化变种

- CoDE
  - 主要思想：



DE/rand/1/bin

DE/rand/2/bin

DE/current-to-rand/1

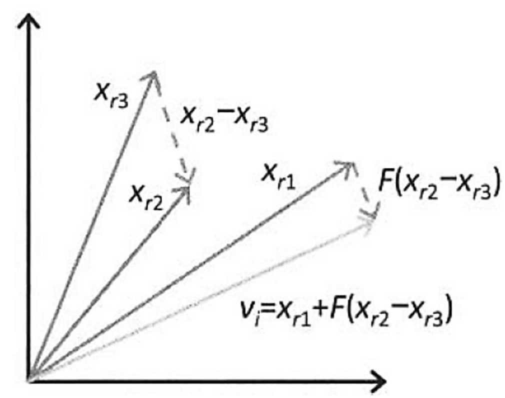
F=1.0, c=0.1
F=1.0, c=0.9
F=0.8, c=0.2

差分进化变种

- CoDE
  - DE/rand/1/bin：没有特定制定某个搜索方向，随机地挑选新的搜索方向
  - DE/rand/2/bin：相较于DE/rand/1/bin，两个差分向量增加了更多的扰动
  - DE/current-to-rand/1：$\mathbf{u}_i \leftarrow \mathbf{x}_i + \text{rand}(\mathbf{x}_{r1}-\mathbf{x}_i)+F(\mathbf{x}_{r2}-\mathbf{x}_{r3})$

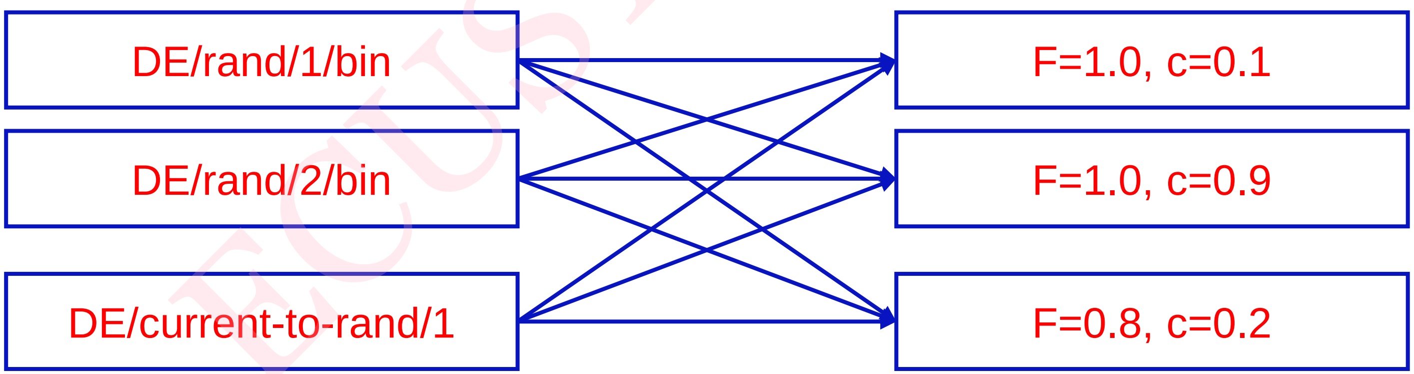
DE/rand/1 + arithmetic crossover

$\mathbf{v}_i \leftarrow \mathbf{x}_{r1} + F(\mathbf{x}_{r2}-\mathbf{x}_{r3})$，$\mathbf{u}_i \leftarrow \mathbf{x}_i + \text{rand}(\mathbf{v}_i - \mathbf{x}_i)$



差分进化变种

- CoDE
  - F=1.0, c=0.1：针对可分问题（separable problems）
  - F=1.0, c=0.9：提升探索性能，种群多样性
  - F=0.8, c=0.2：提升开发性能



差分进化变种

- CoDE

(1) G=0;

(2) Generate an initial population $P_0 = \{\vec{x}_{1,0},\ldots,\vec{x}_{NP,0}\}$ by uniformly and randomly sampling from the feasible solution space;

(3) Evaluate the objective function values $f(\vec{x}_{1,0}),\ldots,f(\vec{x}_{NP,0})$;

(4) FES=NP;

(5) **while** FES<Max_FES **do**

(6) $P_{G+1} = \varnothing$;

(7) **for** i=1:NP **do**

(8) Use the three trial vector generation strategies, each with a control parameter setting randomly selected from the parameter candidate pool, to generate three trial

vectors $\vec{u}_{i\_1,G}$, $\vec{u}_{i\_2,G}$, and $\vec{u}_{i\_3,G}$ for the target vector $\vec{x}_{i,G}$;

(9) Evaluate the objective function values of the three trial vectors $\vec{u}_{i\_1,G}$, $\vec{u}_{i\_2,G}$, and $\vec{u}_{i\_3,G}$;

(10) Choose the best trial vector (denoted as $\vec{u}^*_{i,G}$) from the three trial vectors $\vec{u}_{i\_1,G}$, $\vec{u}_{i\_2,G}$, and $\vec{u}_{i\_3,G}$;

(11) $P_{G+1} = P_{G+1} \cup select(\vec{x}_{i,G}, \vec{u}^*_{i,G})$;

(12) FES=FES+3;

(13) **end for**

(14) G=G+1;

(15) **end while**

差分进化变种

- SaDE
  - Qin等人于2009年提出的差分进化变体 (IEEE Transactions on Evolutionary Computation, 2009, 13(2))
  - **主要创新：**

1. 多个试验向量生成策略组合
2. 历史信息利用
3. 控制参数自适应

  - 是非常经典的差分进化变体，迄今为止已被广泛应用于求解众多应用优化问题

差分进化变种

- SaDE
  - 试验向量生成策略：变异算子+交叉算子

DE/rand/1/bin： $\mathbf{v}_i \leftarrow \mathbf{x}_{r1} + F(\mathbf{x}_{r2}-\mathbf{x}_{r3})$

DE/current-to-best/2/bin： $\mathbf{v}_i \leftarrow \mathbf{x}_i + F(\mathbf{x}_{best}-x_i) + F(\mathbf{x}_{r1}-\mathbf{x}_{r2}) + F(\mathbf{x}_{r3}-\mathbf{x}_{r4})$

DE/rand/2/bin： $\mathbf{v}_i \leftarrow \mathbf{x}_{r1} + F(\mathbf{x}_{r2}-\mathbf{x}_{r3}) + F(\mathbf{x}_{r4}-\mathbf{x}_{r5})$

DE/current-to-best/1： $\mathbf{v}_i \leftarrow \mathbf{x}_i + F(\mathbf{x}_{best}-\mathbf{x}_i)+F(\mathbf{x}_{r2}-\mathbf{x}_{r3})$

利用过往若干代的“成功经验”，挑选本代使用的试验向量生成策略

差分进化变种

- SaDE
  - 历史信息利用：

成功

Index	DE/rand/1/bin	DE/current-to-best/2/bin	DE/rand/2/bin	DE/current-to-best/1
1	$ns_{1,G-LP}$	$ns_{2,G-LP}$	$ns_{3,G-LP}$	$ns_{4,G-LP}$
2	$ns_{1,G-LP+1}$	$ns_{2,G-LP+1}$	$ns_{3,G-LP+1}$	$ns_{4,G-LP+1}$
⋮	⋮	⋮	⋮	⋮
LP	$ns_{1,G-1}$	$ns_{2,G-1}$	$ns_{3,G-1}$	$ns_{4,G-1}$

失败

Index	DE/rand/1/bin	DE/current-to-best/2/bin	DE/rand/2/bin	DE/current-to-best/1
1	$nf_{1,G-LP}$	$nf_{2,G-LP}$	$nf_{3,G-LP}$	$nf_{4,G-LP}$
2	$nf_{1,G-LP+1}$	$nf_{2,G-LP+1}$	$nf_{3,G-LP+1}$	$nf_{4,G-LP+1}$
⋮	⋮	⋮	⋮	⋮
LP	$nf_{1,G-1}$	$nf_{2,G-1}$	$nf_{3,G-1}$	$nf_{4,G-1}$

$$p_{k,G} = \frac{S_{k,G}}{\sum_{k=1}^{K} S_{k,G}}$$

差分进化变种

- SaDE
  - –尺度参数F和交叉率c的自适应更新：

尺度参数F：以高斯分布估计，N(0.5, 0.3)

交叉率c：以高斯分布估计，但是均值为动态

N(c_m，std)，cm初始化为0.5，后以成功的历史信息生成，std为0.1

第G代 第G+1代 第G+2代

本章内容

1. 背景知识

2. 基本差分进化

3. 差分进化变种

4. 离散优化

5. 差分进化与遗传算法

离散优化

- 经典差分进化
  - 用于连续域优化
  - 若应用于离散域的优化问题，在变异操作需进行修改：

$$\mathbf{v}_i \leftarrow \mathbf{x}_{r1} + F(\mathbf{x}_{r2}-\mathbf{x}_{r3})$$

因为F∈[0,1]，导致$\mathbf{v}_i$可能不属于问题域D

  - 两种修改方式：

1. 用经典差分进化算法生成变异向量$\mathbf{v}_i$，然后修改$\mathbf{v}_i$让它落在问题域D中；
2. 修改变异向量的生成方法，直接生成处于D中的变异向量

离散优化

- 混合整数差分进化
  - 要保证$\mathbf{v}_i \in D$的一个方法是将它投影到D上
  - 为了能在离散域上优化，按这种方式修改得到的差分进化方法被称为混合整数差分进化：

如果D是n维整数向量的集合，则

$$\mathbf{v}_i \leftarrow \mathbf{P}[\mathbf{x}_{r1} + F(\mathbf{x}_{r2}-\mathbf{x}_{r3})]$$

其中**P**是投影算子，对所有**x**，**P**(**x**)∈D。

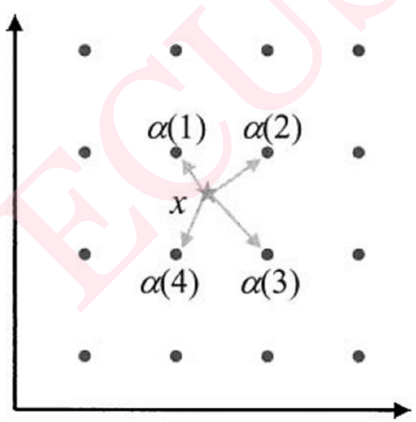
例：**P**可以为round函数，$\mathbf{v}_i \leftarrow \mathrm{round}[\mathbf{x}_{r1} + F(\mathbf{x}_{r2}-\mathbf{x}_{r3})]$

离散优化

- 混合整数差分进化
  - 通常情况下，**P**会更复杂：假设问题域D是n维整数向量的集合，可以定义P为

$$\mathbf{P}(x) = \arg\min_{\boldsymbol{\alpha}} f(\boldsymbol{\alpha}) : \boldsymbol{\alpha} \in D,\ |x_j - \alpha_j| < 1 \text{对所有} j \in [1,n]$$

将取实数值的向量**x**投影到取整数值的向量**α**，让目标函数最小并且它的每一个元素与**x**相应的元素在一个单位之内。



将连续值的向量**x**投影到离散值向量**α**

离散优化

- 离散差分进化
  - 修改变异向量的生成方法，直接生成处于D中的变异向量，这种修改方式称为离散差分进化
  - 将 $\mathbf{v}_i \leftarrow \mathbf{x}_{r1} + F(\mathbf{x}_{r2}-\mathbf{x}_{r3})$ 替换为 $\mathbf{v}_i \leftarrow \mathbf{G}(\mathbf{x}_{r1}, \mathbf{x}_{r2}, \mathbf{x}_{r3})$，其中 $\mathbf{G}(\cdot) \in D$
  - 编写函数 $\mathbf{G}(\cdot)$，可以有如下方案：

方案1： $\mathbf{v}_i \leftarrow \mathbf{x}_{r1} + \mathrm{round}[F(\mathbf{x}_{r2}-\mathbf{x}_{r3})]$

方案2： $\mathbf{v}_i \leftarrow \mathbf{x}_{r1} + \mathrm{sign}(\mathbf{x}_{r2}-\mathbf{x}_{r3})$

其中，round和sign的函数运算按向量的元素逐个进行

本章内容

1. 背景知识

2. 基本差分进化

3. 差分进化变种

4. 离散优化

5. 差分进化与遗传算法

差分进化与遗传算法

- 对差分进化的一点探讨
  - 均匀交叉

假设有两个父代$\mathbf{x}_a$和$\mathbf{x}_b$，均匀交叉得到子代y，其中对每个k∈[1,n], y的第k个特征为：

$$y(k) \leftarrow x_{i(k)}(k)$$

这里从集合{a, b}随机地选择i(k)，即随机地以50%的概率从它的两个父代选择每一个子代的特征

  - 在差分进化中定义：$\mathbf{x}_a=\mathbf{x}_i$，$\mathbf{x}_b=\mathbf{v}_i$，$\mathbf{y}=\mathbf{u}_i$
  - 如果子代更好，用子代$\mathbf{u}_i$替换$\mathbf{x}_i$：(1+1)-ES

差分进化与遗传算法

- 对差分进化的一点探讨
  - 差分进化可以看成遗传算法/进化策略的修改版

遗传算法的修改版1，最小化f(**x**)，c是交叉率

初始化候选种群{$\mathbf{x}_i$}, i∈[1,N]
While not (终止准则)
    For 每一个个体$\mathbf{x}_i$，i∈[1,N]
        r_1←随机整数，r_1∈[1,N]: r_1≠i
        $\mathbf{v}_i \leftarrow \mathbf{x}_{r1}$
        For 每一维j∈[1,n]
            If rand(0,1)<c then
                $u_{ij} \leftarrow v_{ij}$
            else
                $u_{ij} \leftarrow x_{ij}$
            Endif
        下一维
    下一个个体
    For 每一个i∈[1,N]，if f($\mathbf{u}_i$)<f($\mathbf{x}_i$)，then $\mathbf{x}_i \leftarrow \mathbf{u}_i$ End if
下一代

差分进化与遗传算法

- 对差分进化的一点探讨
  - 差分进化可以看成遗传算法/进化策略的修改版

遗传算法的修改版2，最小化f($\mathbf{x}$)，F是步长，c是交叉率

初始化候选种群{$\mathbf{x}_i$}, $i \in [1,N]$
While not (终止准则)
    For 每一个个体$\mathbf{x}_i$， $i \in [1,N]$
        $r_1 \leftarrow$随机整数，$r_1 \in [1,N]$: $r_1 \neq i$
        $r_2 \leftarrow$随机整数，$r_2 \in [1,N]$: $r_2 \notin \{i, r_1\}$
        $r_3 \leftarrow$随机整数，$r_3 \in [1,N]$: $r_3 \notin \{i, r_1, r_2\}$
        $\mathbf{v}_i \leftarrow \mathbf{x}_{r1} + F(\mathbf{x}_{r2} - \mathbf{x}_{r3})$
        $Jr \leftarrow$随机整数，$Jr \in [1,n]$
        For 每一维$j \in [1,n]$
            If (rand(0,1)<c) or (j=Jr) then
                $u_{ij} \leftarrow v_{ij}$
            else
                $u_{ij} \leftarrow x_{ij}$
            Endif
        下一维
    下一个个体
    For 每一个$i \in [1,N]$，if f($\mathbf{u}_i$)<f($\mathbf{x}_i$)，then $\mathbf{x}_i \leftarrow \mathbf{u}_i$ End if
下一代

差分进化与遗传算法

- 对差分进化的一点探讨
  - 遗传算法是差分进化的一种特殊情况？
  - 差分进化是遗传算法的一个变种？
  - 遗传算法作为进化算法的重要基础，标签永远不会过时
  - 差分进化的独特性，足以将其看成是一个独立的进化算法，而不是其他算法的特殊情况

结束