

第六章 进化策略

进化策略

1. 引言

- **Evolutionary Strategy**, 简称**ES**
- 最初是由**I. Rechenberg**和**H.-P. Schwefel**提出
- 主要应用于求解数值优化问题
- 主要特点
 - 快速
 - 相对较多的理论
 - 自适应（变异）参数
 - 强调个体级的行为变化

进化策略

2. 小例子

- 问题: minimize $f: \mathbf{R}^n \rightarrow \mathbf{R}$
- 算法: (1+1)-ES
 - 从 \mathbf{R}^n 直接产生一个实数向量作为初始染色体
 - 种群大小为1
 - 每次迭代只利用变异运算产生一个子代染色体
 - 种群更新时采用贪婪策略

进化策略

2. 小例子

- Set $t = 0$
- Create initial chromosome $x^t = (x_1^t, \dots, x_n^t)$
- **REPEAT UNTIL** (*TERMIN.COND* is satisfied) **DO**
 - Draw z_i from a normal distribution for all $i = 1, \dots, n$
 - $y_i^t = x_i^t + z_i$
 - **IF** $f(x^t) < f(y^t)$ **THEN** $x^{t+1} = x^t$
 - **ELSE** $x^{t+1} = y^t$
 - Set $t = t + 1$
- **END DO**

(1+1)ES算法的伪代码

进化策略

2. 小例子

- \mathbf{z} 值根据正态分布 $N(\xi, \sigma)$ 产生
 - 均值 ξ 设为 0
 - 方差 σ 被称为变异步长
- σ 值根据“**1/5 success rule**”进行重置，即根据如下公式每 k 次迭代更新一下 σ 的取值
 - $\sigma = \sigma/c$ 如果 $ps > 1/5$
 - $\sigma = \sigma \cdot c$ 如果 $ps < 1/5$
 - $\sigma = \sigma$ 如果 $ps = 1/5$

这里 ps 表示成功变异次数的比例, $0.8 \leq c \leq 1$

ES算法设计

1. 编码方法

- 染色体由三个部分组成
 - 解变量: x_1, \dots, x_n
 - 策略参数
 - ✓ 变异步长: $\sigma_1, \dots, \sigma_n$
 - ✓ 旋转角度: $\alpha_1, \dots, \alpha_k$
- 每一个部分并不是总是需要
- 完整形式: $(x_1, \dots, x_n, \sigma_1, \dots, \sigma_n, \alpha_1, \dots, \alpha_k)$
这里 $k = n(n-1)/2$ (i, j 对的数量)

ES算法设计

2. 变异算子

➤ 主要机制

- 根据正态分布产生的随机扰动更新染色体
- $\mathbf{x}'_i = \mathbf{x}_i + N(\mathbf{0}, \sigma)$

➤ 关键思想

- σ 是染色体编码 $(\mathbf{x}_1, \dots, \mathbf{x}_n, \sigma)$ 的一部分
- σ 也需要变异为 σ' (见后文)

➤ 这样变异步长 σ 同解变量 \mathbf{x} 一同进化迭代

ES算法设计

2. 变异算子

- 最终变异效果: $(\boldsymbol{x}, \boldsymbol{\sigma}) \rightarrow (\boldsymbol{x}', \boldsymbol{\sigma}')$
- 执行顺序很重要
 - 先执行 $\boldsymbol{\sigma} \rightarrow \boldsymbol{\sigma}'$
 - 后执行 $\boldsymbol{x} \rightarrow \boldsymbol{x}' = \boldsymbol{x} + N(\mathbf{0}, \boldsymbol{\sigma}')$
- 改变执行顺序会导致变异运算失效

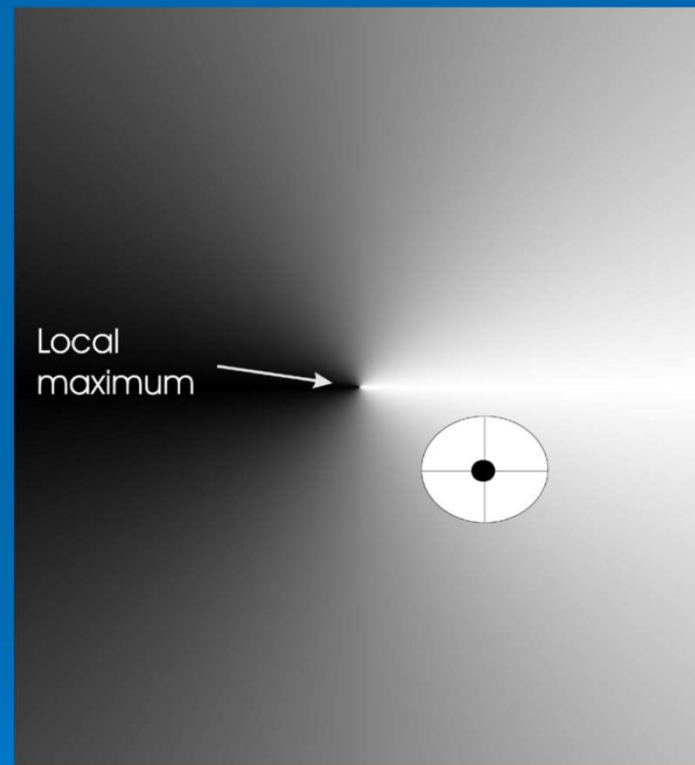
ES算法设计

变异算子示例: 只含一个变异率的非相关变异

- 染色体: $(x_1, \dots, x_n, \sigma)$
- 先执行 $\sigma' = \sigma \cdot \exp(\tau \cdot N(0, 1))$
- 后执行 $x'_i = x_i + \sigma' \cdot N(0, 1)$
- 学习率: $\tau \propto 1/n^{1/2}$
- 变异率边界值: $\sigma' < \varepsilon_0 \Rightarrow \sigma' = \varepsilon_0$

ES算法设计

Mutants with Equal Likelihood



Chromosome: (x, y, σ)

Circle: mutants having the same chance to be created

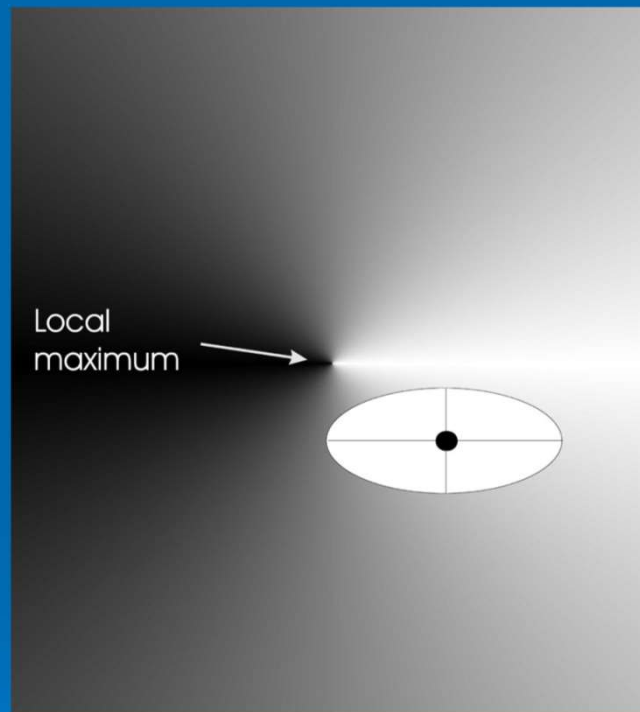
ES算法设计

变异算子示例: 含有 n 个变异率的非相关变异

- 染色体: $(x_1, \dots, x_n, \sigma_1, \dots, \sigma_n)$
- 先执行 $\sigma'_i = \sigma_i \cdot \exp(\tau' \cdot N(0, 1) + \tau \cdot N_i(0, 1))$
- 后执行 $x'_i = x_i + \sigma'_i \cdot N(0, 1)$
- 两个学习率参数
 - τ' : 总体学习率, 通常 $\tau' \propto 1/(2n)^{1/2}$
 - τ : 坐标位学习率, 通常 $\tau \propto 1/(2n^{1/2})^{1/2}$
- 变异率边界值: $\sigma'_i < \varepsilon_0 \Rightarrow \sigma'_i = \varepsilon_0$

ES算法设计

Mutants with Equal Likelihood



Chromosome: $(x, y, \sigma_x, \sigma_y)$

Ellipse: mutants having the same chance to be created

ES算法设计

变异算子示例: 相关性变异

- 染色体: $(x_1, \dots, x_n, \sigma_1, \dots, \sigma_n, \alpha_1, \dots, \alpha_k)$
- 协方差矩阵 C 定义如下:
 - $c_{ii} = \sigma_i^2$
 - $c_{ij} = 0$ 如果 i 和 j 不相关
 - $c_{ij} = 1/2 \cdot (\sigma_i^2 - \sigma_j^2) \cdot \tan(2 \alpha_{ij})$ 如果 i 和 j 相关

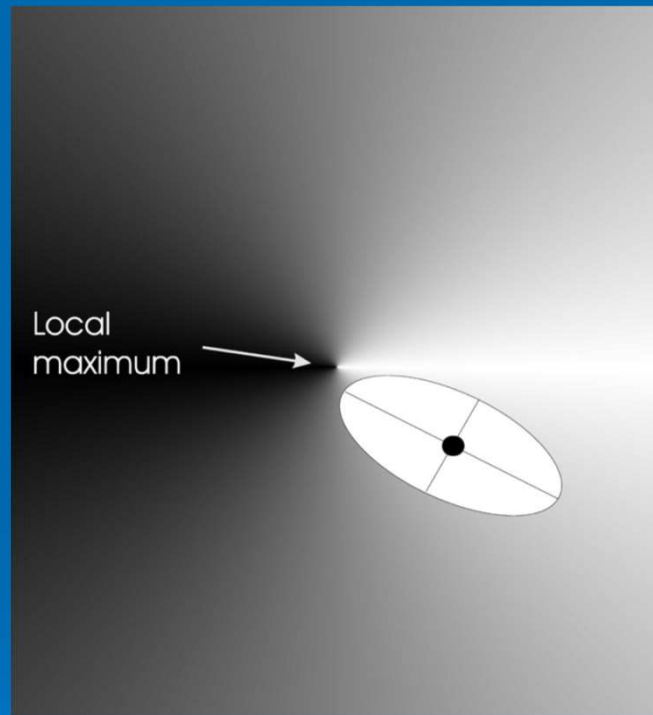
ES算法设计

变异算子示例: 相关性变异

- 先执行 $\sigma'_i = \sigma_i \cdot \exp(\tau' \cdot N(0, 1) + \tau \cdot N_i(0, 1))$
- 再执行 $\alpha'_j = \alpha_j + \beta \cdot N(0, 1)$
- 后执行 $\mathbf{x}' = \mathbf{x} + N(\mathbf{0}, \mathbf{C}')$
 - \mathbf{x} : 染色体向量 (x_1, \dots, x_n)
 - \mathbf{C}' : 对 α 执行变异后的协方差矩阵
- 参数设置:
 - $\tau' \propto 1/(2n)^{1/2}$, $\tau \propto 1/(2n^{1/2})^{1/2}$, $\beta \approx 5^\circ$
- 参数边界值:
 - $\sigma'_i < \varepsilon_0 \Rightarrow \sigma'_i = \varepsilon_0$
 - $|\alpha'_j| > \pi \Rightarrow \alpha'_j = \alpha'_j - 2 \cdot \pi \cdot \text{sign}(\alpha'_j)$

ES算法设计

Mutants with Equal Likelihood



Chromosome: $(x, y, \sigma_x, \sigma_y, \alpha_{x,y})$

Ellipse: mutants having the same chance to be created

ES算法设计

3. 重组算子

- 利用两个父代染色体产生一个子代染色体

	Two fixed parents	Two parents selected for each i
$z_i = (x_i + y_i)/2$	Local intermediary	Global intermediary
z_i is x_i or y_i chosen randomly	Local discrete	Global discrete

ES算法设计

4. 选择策略

- 父代个体总是利用均匀随机分布来选择
- 这意味着ES中父代染色体选择是无偏的，也就是说每个染色体都有相同的概率被选中
- 注意，在ES中，“父代个体”通常指整个种群（在GA中：被选择进行遗传操作的群体成员）

ES算法设计

5. 种群更新策略

- μ 个父代个体执行变异和重组操作后产生 λ 个子代，通过种群更新策略产生下一代父代种群
- 两种常用的种群更新策略
 - (μ, λ) 更新策略:: 仅从 λ 个子代中选择 μ 个最好个体构成下一代父种群
 - $(\mu + \lambda)$ 更新策略: 从当前所有的父代和子代(总计 $\mu + \lambda$ 个个体)中选择 μ 个最好个体构成下一代父种群

ES算法设计

5. 种群更新策略

- 通常 (μ, λ) 更新策略被认为更好
 - 使用 $(\mu + \lambda)$ 更新策略可能会导致差的变异率 σ 被继承
 - (μ, λ) -ES中选择压力通常要设得很高 (通常 $\lambda \approx 7 \cdot \mu$)

ES算法实验（选作）

Ackley Function

- The Ackley function (here used with $n=30$)

$$f(x) = -20 \cdot \exp\left(-0.2 \sqrt{\frac{1}{n} \cdot \sum_{i=1}^n x_i^2}\right) - \exp\left(\frac{1}{n} \sum_{i=1}^n \cos(2\pi x_i)\right) + 20 - e$$

- ES design

- Representation:
 - ✓ $-30 < x_i < 30$
 - ✓ 30 step size
- (30, 200) selection
- Termination: after 200000 fitness evaluations
- Results: average best solution is $7.48 \cdot 10^{-8}$