

数值函数优化方法和多目标优化

何宇青

2024 年 9 月 25 日

为什么需要遗传算法

- 在函数是连续, 可求导, 低阶的简单情况下可解析地求最优解外, 大部分情况下, 只能通过数值方法求近似最优解.
- 至今还没有一种既能处理各种不同的复杂函数, 有具有良好求解结果的数值计算方法

需要研究出一种能够在可接受的时间和可接受的精度范围内求出数值函数近似最优解的方法或通用框架, 遗传算法提供了一种求解这类优化问题的通用框架.

定义

在环境 e 下策略 s 的在线性能 $X_e(s)$ 定义为:

$$X_e(s) = \frac{1}{T} \sum_{t=1}^T f_e(t)$$

其中 $f_e(t)$ 是在环境 e 下在时刻 t 的平均目标函数或平均适应度.

算法的在线性能指标表示算法从开始运行一直到当前为止的时间段内性能平均值, 反映了算法的动态性能.

定义

在环境 e 下策略 s 的离线性能 $X_e^*(s)$ 定义为:

$$X_e^*(s) = \frac{1}{T} \sum_{t=1}^T f_e^*(t)$$

其中 $f_e^*(t)$ 是在环境 e 下在时间段 $[0, t]$ 内最好目标函数或最大适应度.

算法的离线性能表示算法运行过程中各代的最佳性能值的累积平均, 反映算法的收敛性能.

算法性能测试函数

定义 (De Jong 函数 F1)

$$f_1(x_1, x_2, x_3) = \sum_{i=1}^3 x_i^2$$
$$-5.12 \leq x_i \leq 5.12 \quad i = 1, 2, 3$$

一个简单的平方和函数, 只有一个极小点 $f_1(0, 0, 0) = 0$

定义 (De Jong 函数 F2)

$$f_2(x_1, x_2) = 100(x_1^2 - x_2)^2 + (1 - x_1)^2$$
$$-2.048 \leq x_i \leq 2.048 \quad i = 1, 2$$

具有一个全局最小点 $f_2(1, 1) = 0$, 该函数虽然是单峰函数, 但他是病态的, 难以全局极小化.

算法性能测试函数

定义 (De Jong 函数 F3)

$$f_3(x_1, \dots, x_5) = \sum_{i=1}^5 [x_i] \\ -5.12 \leq x_i \leq 5.12 \quad i = 1, 2, 3, 4, 5$$

$[\cdot]$ 为向下取整函数

不连续, 对于 $x_i \in [-5.12, -5]$ 区域的每个点, 都取全局极小值 -30

定义 (De Jong 函数 F4)

$$f_4(x_1, \dots, x_{30}) = \sum_{i=1}^{30} ix_i^4 + \mathbf{Gauss}(0, 1) \\ -1.28 \leq x_i \leq 1.28 \quad i = 1, 2, \dots, 30$$

不考虑噪音影响时, 有全局极小值 $f_4(0, 0, \dots, 0) = 0$

定义 (De Jong 函数 F5)

$$f_5(x_1, x_2) = 0.02 + \sum_{j=1}^{25} 1/[j + \sum_{i=1}^2 (x_i - a_{ij})^6]$$
$$-65.536 \leq x_i \leq 65.536 \quad i = 1, 2$$

$$[a_{ij}] =$$

$$\begin{bmatrix} -32 & -16 & 0 & 16 & 32 & -32 & -16 & \cdots & 0 & 16 & 32 \\ -32 & -32 & -32 & -32 & -32 & -16 & -16 & \cdots & 32 & 32 & 32 \end{bmatrix}$$

这是一个多峰函数, 共 25 个局部极小点, 其中只有一个全局极小点

$$f_5(-32, -32) = 0.998$$

- **编码方法:** 二进制编码符号串表示个体
- **算法的影响参数:**
 - 群体大小 M
 - 交叉概率 P_c
 - 变异概率 P_m
 - 代沟 $G(0 < G < 1$ 指子代与父代之间的重叠程度)
- **算法种类:**
 - 基本遗传算法 (比例选择, 单点交叉, 基本位变异)
 - 保留最佳个体模型
 - 期望值模型
 - 排挤因子模型
 - 广义交叉模型

- 群体的规模越大, 遗传算法的离线性能越好, 越容易收敛.
- 规模较大的群体, 遗传算法的初始在线性能较差, 规模较小的群体相反.
- 变异概率的增大会增加群体的多样性, 但降低了算法的离线性能和在线性能, 并随着变异概率的增大, 算法性能越接近随机搜索的性能.
- 使用保留最佳个体模型或期望值模型的算法比基本遗传算法的性能有明显改进.
- 广义交叉算子会随着交叉点的增加降低算法的在线性能和离线性能.

使多个目标在给定区域都尽可能地好的优化

数学模型

$$\begin{aligned} \mathbf{V}\text{-min} \quad & f(x) = [f_1(x), f_2(x), \dots, f_p(x)]^T \\ \text{s.t.} \quad & \begin{cases} x \in X \\ X \subseteq \mathbb{R}^m \end{cases} \end{aligned}$$

$\mathbf{V}\text{-min}$ 表示向量最小化.

很多情况下各个子目标可能相互冲突, 一个子目标的改善往往以其他子目标的恶化为代价.

定义

$X \subseteq \mathbb{R}^m$ 是多目标优化模型的约束集, $f(x) \in \mathbb{R}^p$ 是多目标优化时的目标向量目标函数, $x_1, x_2 \in X$, 若 $f_k(x_1) \leq f_k(x_2), \forall k = 1, \dots, p$, 且 $f_k(x_1) < f_k(x_2), \exists k = 1, \dots, p$, 则称 x_1 优于 x_2 .

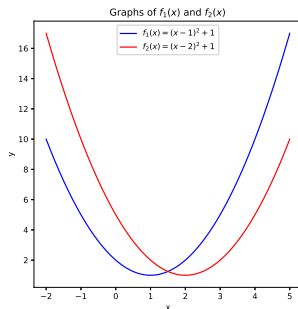
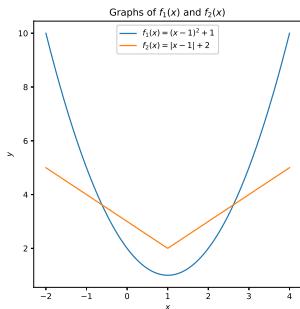
定义

$X \subseteq \mathbb{R}^m$ 是多目标优化模型的约束集, $f(x) \in \mathbb{R}^p$ 是多目标优化时的目标向量目标函数, 若 $x^* \in X$, 且对于任意的 $x \in X$, 都有 x^* 优于 x , 则称 x^* 是多目标优化问题的最优解.

大多数情况下最优解并不存在, 因此需要寻找一组解, 使得这组解在某种意义上是“最优的”.

定义 (Pareto 最优解)

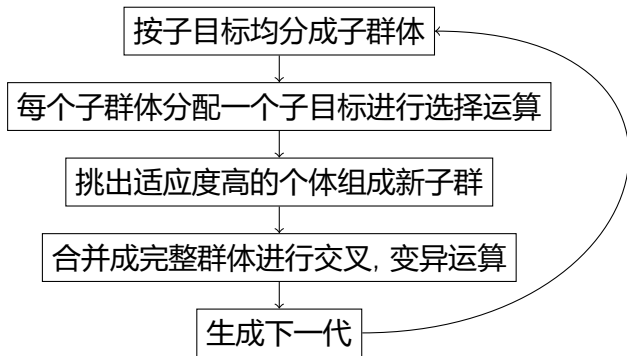
$X \subseteq \mathbb{R}^m$ 是多目标优化模型的约束集, $f(x) \in \mathbb{R}^p$ 是多目标优化时的目标向量目标函数, 若 $\tilde{x} \in X$, 且不存在 $x \in X$, 使得 x 优于 \tilde{x} , 则称 \tilde{x} 是多目标优化问题的 Pareto 最优解.



权重系数变化法

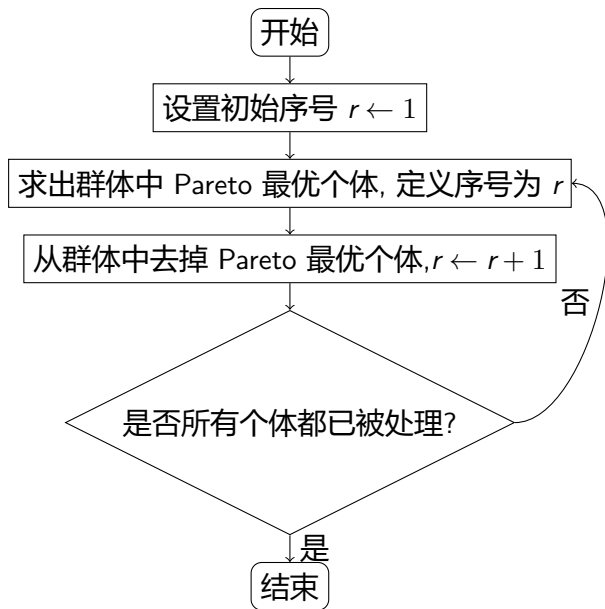
给每一个子目标函数 $f_i(x)$ 赋予权重 w_i , 将 $u[f](x) = \sum_{i=1}^P w_i f_i(x)$ 作为评价函数, 将其转化为单目标优化问题. 权重系数变化法就是在此基础上对每个个体取不同的权重系数, 用通常的遗传算法求解多个 Pareto 最优解.

并列选择法



容易产生个别子目标函数的极端最优解, 找到所有函数的协调最优解比较困难

排序选择法



排序选择法

基本思想

基于 Pareto 最优个体排序, 依据排列次序进行选择运算, 使得排在前面的 Pareto 最优个体有更多的机会遗传到下一代群体中, 经过循环得到 Pareto 最优解.

一般希望解分散在整个 Pareto 最优集合内, 而不是集中在某一部分, 因此利用小生境遗传算法, 让小生境数较小的个体有更多的机会遗传到下一代群体中. 增加了群体的多样性.

定义 (小生境数)

$m_X = \sum_{Y \in P} s(d(X, Y))$, $s(d)$ 为共享函数, 关于 d 单调递减.

Algorithm 1 Tournament Sharing Selection

- 1: 给定 k
 - 2: 群体中随机选 k 个个体组成集合 C
 - 3: 群体中随机选 2 个个体组成集合 T
 - 4: 比较 T 中个体于 C 中个体优越关系
 - 5: **if** T 中一个个体 (X) 比 C 中所有个体优越, 另一个体不比 C 中所有个体优越 **then**
 - 6: X 遗传到下一代
 - 7: **else**
 - 8: T 中小生境数较小的个体遗传到下一代
 - 9: **end if**
-

优点

能够得到多种不同的 Pareto 最优解

缺点

需要进行大量评价和比较运算, 搜索效率较低

Algorithm 2 Hybrid Selection

- 1: **并列选择过程**: 按子目标个数将群体均分成子群体, 并产生下一代子群体
 - 2: **保留 Pareto 最优个体过程**: 子群体中的 Pareto 最优个体不进行交叉, 变异运算, 直接保留到下一代子群体
 - 3: **共享函数处理过程**: 将子群体中非 Pareto 最优个体组成集合, 进行共享函数选择运算, 产生下一代子群体
-

主要思想

主体使用并列选择法, 通过引用保留最佳个体和共享函数的思想弥补并列选择法的不足.

算例

$$\begin{aligned} \min f_1 &= x_1^2/4 \\ \min f_2 &= x_1(1 - x_2) + 5 \\ \text{s.t. } &\begin{cases} 1 \leq x_1 \leq 4 \\ 1 \leq x_2 \leq 2 \end{cases} \end{aligned}$$

变量用 8 为二进制编码串表示, 交叉运算使用单点交叉算子, 变异运算使用基本位变异算子.

运行参数为: $\{M, T, p_c, p_m\} = \{100, 100, 0.8, 0.01\}$

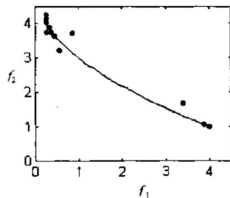


图 7-3 并列选择法的运行结果

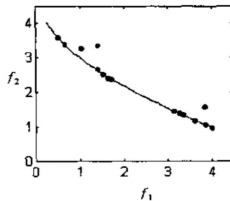


图 7-4 排序选择法的运行结果

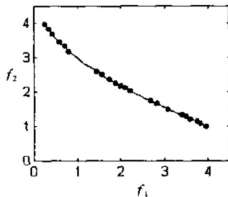


图 7-5 共享函数法的运行结果

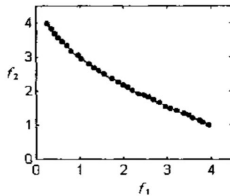
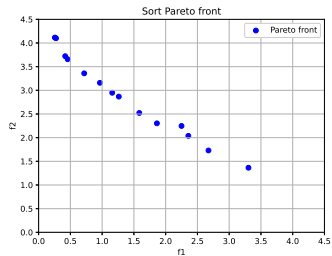
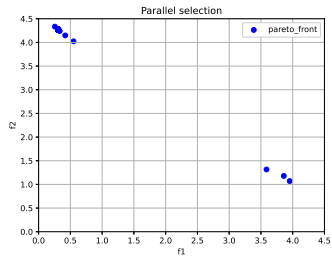


图 7-6 混合法的运行结果



Thank you all!