遗传算法欺骗问题

及其优化策略

junxun-luo@outlook.com

School of mathmatics
Zhejiang University

2024 年 12 月 11 日

遗传算法欺骗问题

将所有妨碍适应度高的个体的生成从而影响算法正常工作的问题统称为遗传算法的欺骗问题

- 性质不好的函数 (例如不连续，多峰值，有跳跃点)
- 编码设计问题
- ...

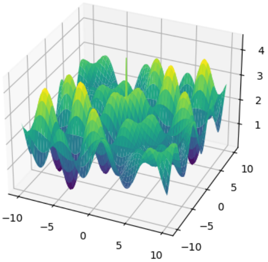
下面都以求解这个函数的最大值为例来说明如何提升遗传算法的求解性能

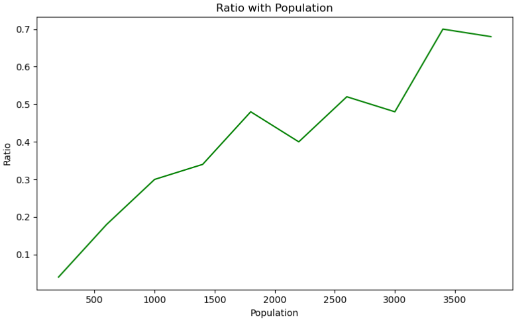
$$f(x,y)=\begin{cases}\sin(\frac{\pi}{3}x)\sin(\frac{\pi}{3}y)+\cos(\frac{\pi}{4}x)\cos(\frac{\pi}{4}y)+4 & |x|,|y|\leq\epsilon\\ \sin(\frac{\pi}{3}x)\sin(\frac{\pi}{3}y)+\cos(\frac{\pi}{4}x)\cos(\frac{\pi}{4}y)+2 & \epsilon<|x|,|y|\leq 10\end{cases}$$

其中 $\sin(\frac{\pi}{3}x)\sin(\frac{\pi}{3}y)+\cos(\frac{\pi}{4}x)\cos(\frac{\pi}{4}y)$
在 $(4.3222, 4.3222)$ 处取到最值 1.91
f 在 $x=y=\epsilon$ 时取到最值

种群大小改进

采用浮点数编码、单点交叉、单点变异求解上述函数的最大值（取 $\epsilon = 0.1$）．考察种群大小对求解效果的影响： $ratio = \frac{\text{算法搜索到}\delta(0,\epsilon)^2\text{的次数}}{\text{算法执行次数}}$



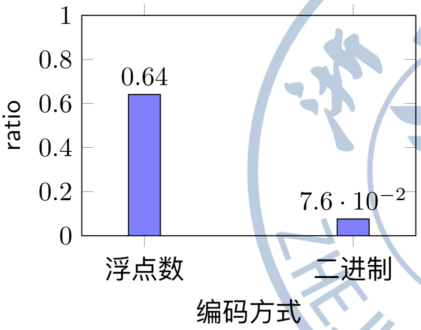


可以看出，随着种群大小的增加，算法的搜索空间变得更大，更容易搜索到全局最优解

编码方式改进举例

比较二进制编码和浮点数编码的效果.
在设定交叉概率 $p_c = 1$，变异概率 $p_m = 0.05$，种群数量 600，迭代次数 200 的情况下
分别运行 1000 次程序比较二者求出最优解的比率：

- 浮点数编码：**ratio: 0.641**
- 二进制编码：**ratio: 0.076**



变异策略改进

对单点变异、均匀变异和高斯变异进行比较.

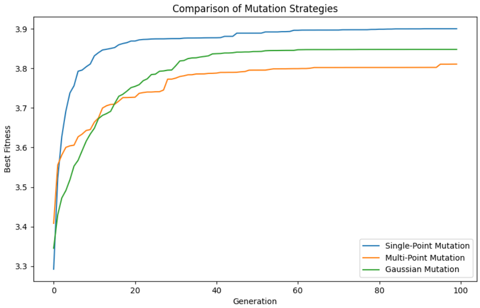


图: 不同变异策略的性能对比

- $0.1_{(10)} = 0.00011001100\ldots_{(2)}$
- 实际上，对求解有利的模式为：
  - $0.00010 * \ldots$
  - $0.0000 * \ldots$
  - $0.00011000 * \ldots$
- 在关键部分进行变异是最重要的.

动态变异策略

设定种群大小为 20，迭代次数 200，交叉概率 $p_c = 0.8$，固定变异概率 $p_{m_{fixed}} = 0.05$，动态变异概率 $p_{m_{dynamic}} = 0.5 \cdot (1 - t/T)$。

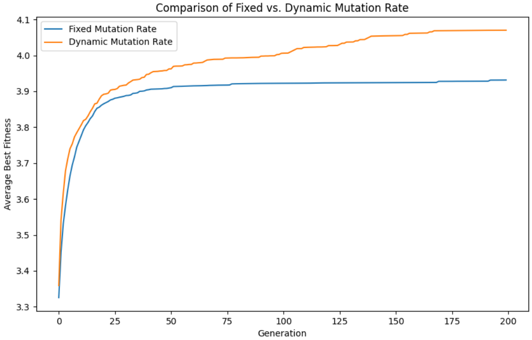


图: 动态变异策略的性能

总结

- 种群大小对搜索广度影响显著，尤其体现在算法运行初期（后期趋同）.
- 采用浮点数编码能显著提高算法性能，因其有助于群体稳定性.
- 变异策略需根据编码方式及问题特性进行调整.
- 动态变异策略能有效提升模型搜索能力.