

3 适应度评价

适应度是指种群个体”适应环境的能力”。适应度的计算同样也是使用遗传算法中的极为关键的要素。它指定了问题解的搜索方向，并直接关系到搜索效率和最终解的质量好坏。

对于已建立的数学模型，一般可以采用目标函数值作为适应度值。为了应对最小化和最大化的两种相反的优化目标，一般遵循最小化约定，即”目标函数值越大，适应度越小”。某些版本是倒过来的，在算法内固定一种标准即可。

适应度的值必须是非负实数。某些问题中求出来的个体目标函数值有正有负，甚至是复数。因此有必要对目标函数与适应度函数之间建立合理的映射关系，保证适应度值是非负的，并且适应度增大的方向与目标函数的优化方向一致。

我们还可以对目标函数作线性变换、指数变换、幂指数变换、截断处理等得到适应度。这些变换都会对种群的多样性和算法的收敛速度带来影响。例如：

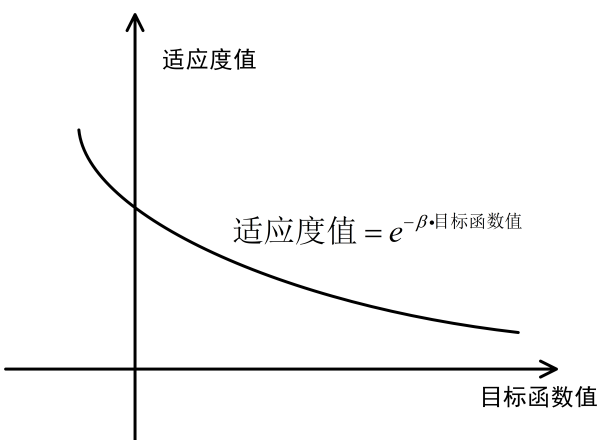


图 1 指数变换

上图进行了目标函数和适应度函数的指数变换，并且使其符合”目标函数值越大，适应度越小”的约定，同时，适应度大的个体将会”更加优秀”。指数变换可以突出优秀个体，加速搜索最优解的速度，但不利于种群多样性的保留。

在适应度计算中我们常常引入”罚函数”来解决带复杂约束的优化问题。”罚函数”通过对非可行解施加惩罚，以此来降低不符合约束条件的”非可行解”个体在下一代中的生存概率。在遗传算法中我们不完全否定非可行解。因为在搜索空间中非可行解有可能非常接近最优可行解。

下面介绍一种基于等级划分的适应度分配计算 (Rank-based fitness assignment):

这种适应度计算方法不是直接对目标函数进行变换来计算的个体的适应度的，而是先根据目标函数值来对个体进行排序，然后根据个体在排序中的位置来确定其适应度。

这种适应度计算方法克服了基于目标函数变换来计算适应度值所带来的放缩问题(因适应度放缩不当而导致的选择压力(其概念详见下一节)过小导致的搜索收敛过慢或过分放大优势个体的适应度而导致的搜索收敛过快)。基于等级划分的适应度分配计算在整个种群中设定了统一的比例，并提供了一种控制选择压力的简单有效的办法。

基于等级划分的适应度分配计算比简单的基于比例或目标函数变换的适应度计算更加健壮，所以，它是值得选择的方法。

这种适应度计算中涉及单目标函数值的线性排序 [1] 与非线性排序 [2] 两种方式：

1) 线性排序：

设种群规模(即种群个体数)为 N_{ind} ， i 为个体在种群中的位置 $i = 1, 2, \dots, N_{ind}$ ，根据个体的目标函数值从大到小对个体进行排序，设 SP 为选择压力。则个体的适应度计算如下：

$$Fitness_i = 2 - SP + 2(SP - 1) \frac{i - 1}{N_{ind} - 1}$$

线性排序中选择压力 SP 的值必须在 [1.0,2.0] 之间。

文献 [1] 中有这种线性排序的详细分析。其选择强度、多样性损失、选择方差(这些概念详见下一节)的计算如下：

选择强度：

$$SelInt(SP) = \frac{SP - 1}{\sqrt{\pi}}$$

多样性损失：

$$LossDiv(SP) = \frac{SP - 1}{4}$$

选择方差：

$$SelVar(SP) = 1 - \frac{(SP - 1)^2}{\pi} = 1 - SelInt(SP)^2$$

2) 非线性排序：

文献 [2] 中引入了一种使用非线性排序的新方法。使用非线性排序比线性排序方法允许设定更高的选择压力。

$$Fitness_i = \frac{N_{ind} \cdot X^{i-1}}{\sum_{i=1}^{N_{ind}} X^{i-1}}$$

其中 X 为下面多项式的根：

$$(SP - N_{ind}) \cdot X^{N_{ind}-1} + SP \cdot X^{N_{ind}-2} + \dots + SP \cdot X + SP = 0$$

非线性排序允许选择压力 SP 的值在 [1, $N_{ind} - 2$] 之间。

上面的适应度计算方法都是对单目标问题而言的。然而，在许多现实问题中，为评估个体的质量好坏，必须考虑多个标准才能确定个体的优越性。这就涉及到多目标的优化问题。我们可以通过线性加权法把多个单目标的目标函数值加权得到多目标的目标函数值，进而使用上面所述的一些单目标问题的适应度计算方法。也可以采用特殊的方法，如帕累托非支配排序法等。

参考文献

[1] Back T , Hoffmeister F . Extended Selection Mechanisms in Genetic Algorithms[J]. Belew & Booker, 1991, 30:92–99.

[2] Pohlheim H . Ein Genetischer Algorithmus Mit Mehrfachpopulationen Zur Numerischen Optimierung[J]. at - Automatisierungstechnik, 1995.