

1 引言

我们采用遗传算法 GA 优化 Griewank 函数，Griewank 函数的表达式如 (1) 式所示：

$$f(\vec{x}) = \sum_{i=1}^N \frac{x_i^2}{4000} - \prod_{i=1}^N \cos\left(\frac{x}{\sqrt{i}}\right) + 1, \quad x_i \in [-600, 600] \quad (1)$$

函数具有多个最小值，且函数可以拓展到高维空间上。最小值取在 $\vec{x} = 0$ 处。Griewank 函数是典型的非线性多模态函数，具有广泛的搜索空间，通常被认为是优化算法很难处理的复杂多模态问题。

如图 1-3 是一维情况下 Griewank 函数图像为了方便起见分别按照三个比例进行展示，从图 1 中我们可以看出，函数在 $[-600, 600]$ 区间段明显可以看出 $x = 0$ 为其最小值点，然而在图 2 中可以明显看出函数在任意位置存在着明显的震荡情况，非常不利于传统算法进行优化搜索。同时在图 3 中可以看到函数在零点附近的很大范围内均仅剩几乎等幅的震荡过程，而函数减除震荡之后所剩的基线基本都是平的。这给传统优化算法提出了相当大的困难。

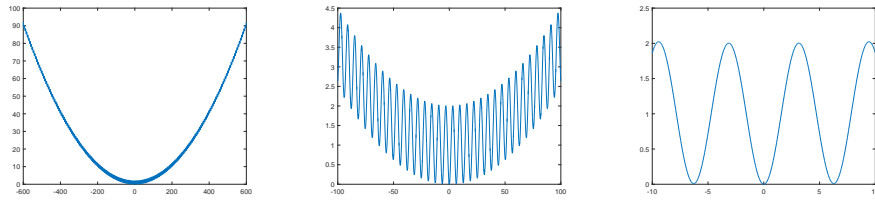


Fig. 1: $x \in [-600, 600]$ 时 Griewank 函数图像 Fig. 2: $x \in [-100, 100]$ 时 Griewank 函数图像 Fig. 3: $x \in [-10, 10]$ 时 Griewank 函数图像

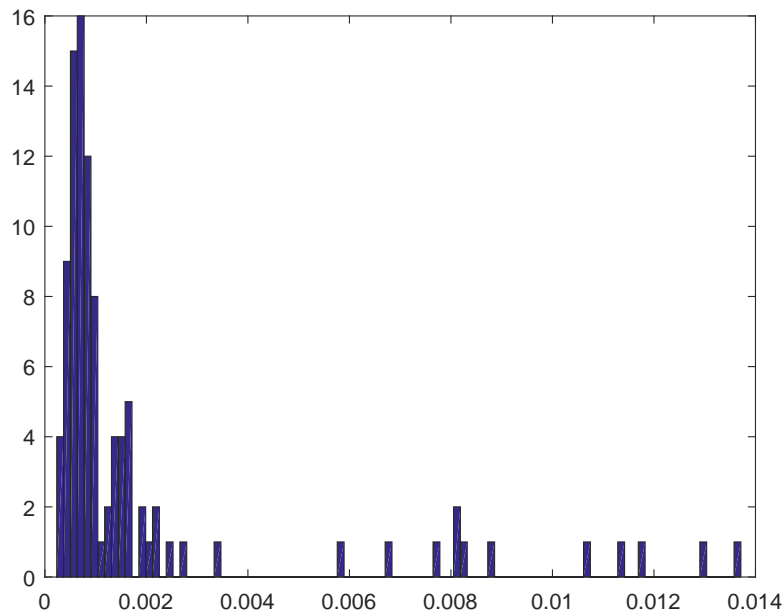


Fig. 4: 优化结果的直方图

2 遗传算法求取过程

我们采用了遗传算法 GA 对函数的最小值点进行求取，其中我们使用了 MATLAB 中自带的遗传算法函数库，这极大的方便了我们的设计过程。由于 Griewank 函数适合向量化并行计算，因此我们设计函数的评估体系为向量化评估，这从很大程度提升了计算速度。其他设置我们采用了 MATLAB 的默认设置。

我们采取的优化维度为 100 维，这个维度下，函数的震荡等问题非常明显，因此传统优化算法几乎不可能完成相关任务。

3 随机试验统计结果

我们对上述算法进行了 100 次随机试验，得到的优化最小值的直方图如图 4 所示。优化最小值的平均值为 0.0020, 标准差为 0.0030, 中位数为 0.00084, 极差为 0.0135。最优优化解达到 0.00023, 最差解达到 0.0137。说明优化效果还是很好的

4 某次试验优化目标变化曲线

如图 5,6 为某次试验优化目标的变化曲线,这次实验中将目标优化到了 0.00054 这个位置。可以看到如图 5 所示,开始迭代几轮之后优化目标迅速降到 1 以下,之后的下降逐渐开始趋于平缓,最优个体的下降速度开始放慢,而平均个体震荡趋近最优解。

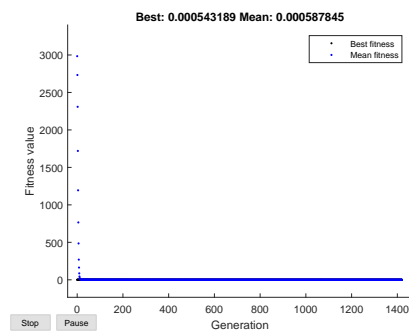


Fig. 5: 优化目标变化曲线

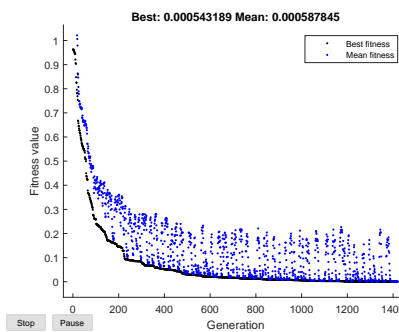


Fig. 6: 优化目标变化曲线

5 遗传算法特点讨论

遗传算法 GA 的特点使得他在求解这个问题时相对于其他的智能优化算法和传统优化算法有着比较大的优势。首先遗传算法受到邻域的影响较小,通过交叉变异等方式,可以很快的跳出局部极小解。另外,从这个问题中我们可以看出,各个维度之间的独立性相对较强,均是越趋近于 0 越容易给出较优的解。因此通过将统一维度上的染色体交叉,这些优势染色体会很好的保留下来,提升了搜索速度。

另外,遗传算法对优化函数的约束相对较小,同时可以做到并行计算,这使得 MATLAB 将问题向量化提升计算速度变得比较方便

6 实验心得和体会

这次实验通过使用 GA 算法对一个函数进行最小值优化,让我初步了解了 GA 算法的操作流程和其特点。相比于之前了解的一些经典优化算法,GA 算法对优化目标的要求很低,甚至可以将优化目标作为黑盒处理。然而,实际上通过加强对优化目标的了解,进行对 GA 算法的设置,可以得到更好的计算效果。当然,相对于传统优化算法,GA 算法在一般情况下计算时间也相对较长,这是由于 GA 算法没有获得更多的信息所致的。

因此，传统优化算法在一些一般的优化问题中仍然有他的实际意义与价值，在一些传统优化算法不能起到作用的地方，GA 等智能优化算法有着其不可或缺的重要作用。

具体代码实现可以见<https://github.com/ZeroWeight/IOAA/Func>