

## 智能优化算法及其应用

# TSP 问题的优化

张蔚桐 2015011493 自 55

## 1 编码方式

我们使用 GA 算法来优化传统的 TSP 问题，这是一个组合优化问题。为了方便使用 MATLAB 自带的遗传算法工具箱，因此我们采用如下方式来对组合进行编码。

我们假设所有的城市按照他们的编号构成一个循环队列，每次当确定下一个访问的城市时，将其从循环队列中移除。我们引入表征组合情况的向量  $X$ ，其中  $x(i)$  表征从当前城市到下一个城市之间在循环队列中的距离。举下面六个城市 AF 为例。

$\{A,B,C,D,E,F\}$  此种访问顺序对应的向量即为  $[1,1,1,1,1,1]$  而  $\{B,D,F,C,A,E\}$  这种访问顺序对应的向量为  $[2,2,2,2,2,2]$  这样，就使得优化元素的向量的取值范围没有有了限制，方便了 GA 算法进行下一步的优化。GA 算法我们采用了 MATLAB 的标准遗传算法库，这样进一步减少了工作量。

## 2 GA 优化效果

如图 1 是 GA 算法优化的结果，我们使用了 GA 算法优化了 10 个城市，得到了很好的结果，在后来尝试对 30,50,75 个城市进行优化的过程中，我们发现单凭 GA 算法很难取得非常令人满意的结论，因此我们补充增加了另外一些优化方式，这一点将在后面说明。

## 3 补充优化方法

由于单凭 GA 算法我们很难获得尤其是 75 城市的最优优化效果，因此我们采用了一些补充优化方式来进一步获得更好的结果。

首先是反复执行 GA 算法若干次，寻找 GA 算法能够提供的最优的路径组合，通过这种方式，实验中我们发现环路的距离被明显的减少，算法方

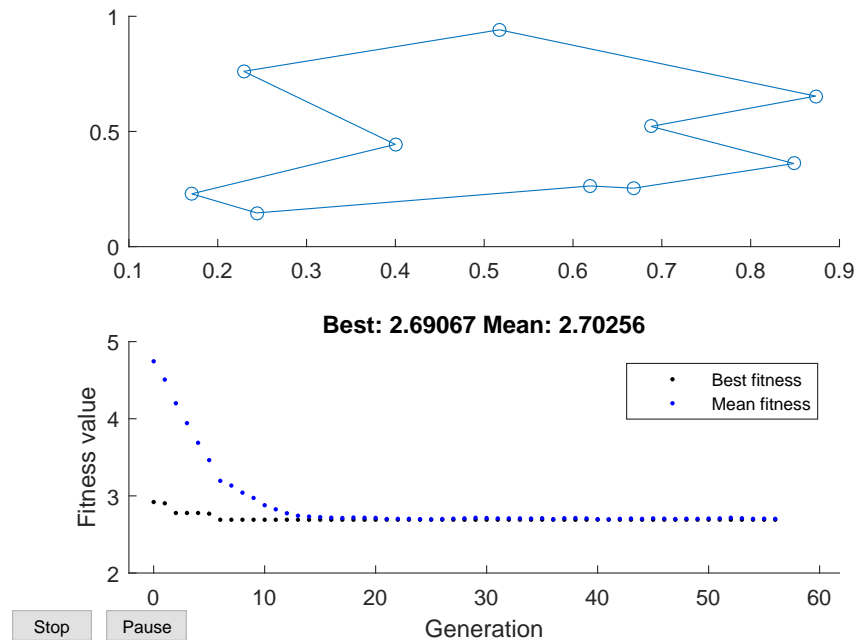


Fig. 1: GA 优化 10 城市 TSP 的最佳路线图和优化过程图

差也由于反复执行的过程而降低，这种方式带来的代价是，算法的执行时间被明显的增长了。

其次在得到 GA 能给出的最优路径之后，我们在这个路径的基础之上进一步的优化这条路径，首先我们进行交叉路径优化，也就是消除环路中存在的交叉点。其次，我们遍历整个环路中的每一个城市，试图交换每一个城市和他前一个城市的位置，如果路径变短则保留新的路径。通过这两种方式，我们在很短的时间内将 GA 算法已经给出的路径明显的变短了。经过简单的测试，我们得到的四种情况的最优路径为图 2-5所示。由于这种测试没有增加 GA 算法的迭代次数（为节省时间），这些算法得到了路径可能稍劣于下一节算法测试中得到的最优路径。

## 4 算法稳定性

我们花了大量时间重复执行了上述算法并尝试对于 10 城市，30 城市，50 城市，75 城市的 TSP 问题进行优化。每次优化过程我们先执行 100 次 GA 算法，找出 GA 算法的最优解，然后反复执行交叉点删除和相邻城市交叉方式对已有路径进行处理。通过这种方式得到了很好的 10 城市，30 城市，50 城市，75 城市的 TSP 问题解。我们对于每一个问题随机优化了 100 次

统计算法性能，算法性能如表 1 所示。四个问题的直方图如图 6 - 9 所示。

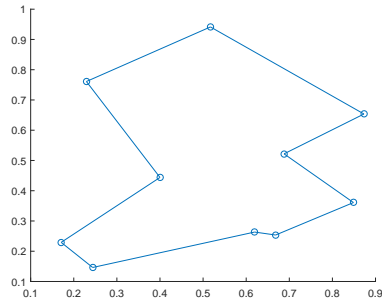


Fig. 2: 10 城市 TSP 路径，环路距离 2.69

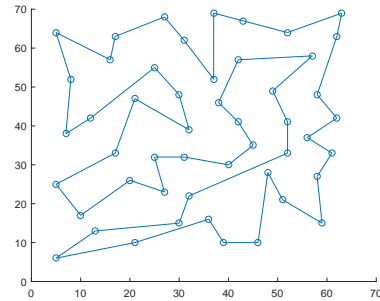


Fig. 4: 50 城市 TSP 路径，环路距离 537.83

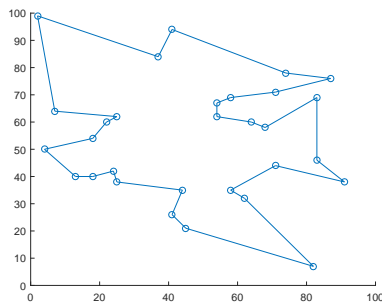


Fig. 3: 30 城市 TSP 路径，环路距离 461.78

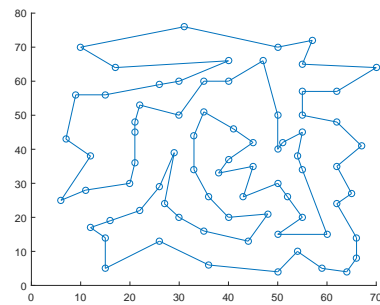


Fig. 5: 75 城市 TSP 路径，环路距离 656.427

## 5 算法特性分析

首先，GA 算法不需要考虑 TSP 问题的复杂性，在采用了上述相对合理的编码方式之后，GA 算法的交叉和变异均可以按照标准的 GA 算法流程进行设计。交叉实际上变化为某一组城市之间的访问次序，而保证其他某些城市之间的访问次序不变。因此，这种交叉有利于保留已有的优秀成分，有利于算法尽快收敛到最小值。

另外，GA 算法可以实现并行化，在实际测试中采用了 8 个线程同时评估子代的性能。相比于传统算法以及 SA 算法，实现并行化之后的 GA 算法相对较快。

然而，GA 算法仍然很难直接实现对大规模 TSP 问题的直接求取，需要引入一些需要数学推导的图论中的特性（包括欧几里得空间的假设等），

这些都是需要改进的地方。

Tab. 1: 算法性能统计表

城市数	最短优化距离	最长优化距离	距离中位数	距离平均值	距离方差
10	2.6907	2.6907	2.6907	2.6907	0.0000
30	444.9039	692.7088	546.0493	547.8904	46.7701
50	500.3703	712.2952	586.7363	587.6895	41.8671
75	670.2950	924.8517	781.7456	778.9307	51.9237

城市数	归一化标准差	城市数	归一化标准差
10	0.0000	30	0.0854
50	0.0712	75	0.0667

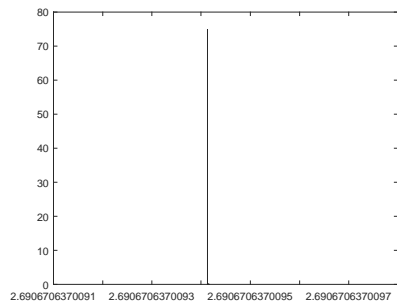


Fig. 6: 10 城市 TSP 优化长度直方图

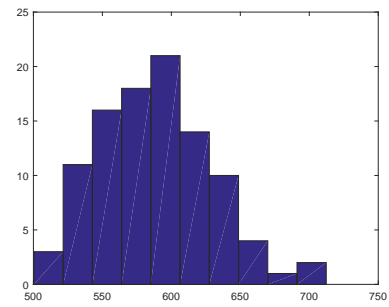


Fig. 8: 50 城市 TSP 优化长度直方图

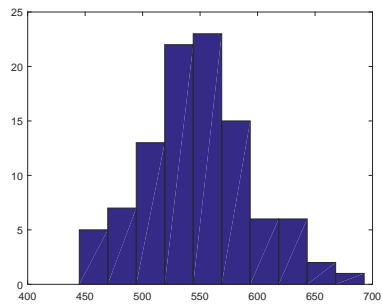


Fig. 7: 30 城市 TSP 优化长度直方图

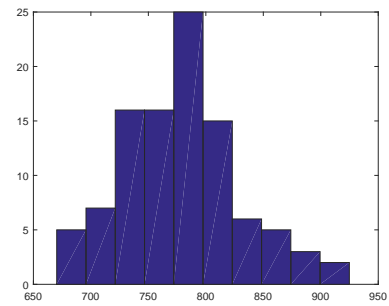


Fig. 9: 75 城市 TSP 优化长度直方图

## 6 实验感想

本次实验使用遗传算法 GA 实现组合优化 TSP 问题，相比于解决传统优化函数的问题，这种问题涉及到合理的编解码等问题，使得问题的难度上升。同时，GA 算法不依靠外部条件的特性一方面保证了设计的简单性，一方面却使得可以通过进一步的专家经验的引入得到更好的路径，因此我知道了在实际应用中，在使用尤其是像 GA 这种黑盒性很强的算法时，需要适当的在优化过程中和优化后引入专家经验，这样使得我们可以获得更优解。

具体代码可以见<https://github.com/ZeroWeight/IOAA/TSP>