

# 智能优化算法 大作业

2017011560 周映佟 自 74  
2017011954 吾尔开希 自 76

2020 年 12 月 25 日

## 目录

<b>1 求解多极小函数</b>	<b>2</b>
1.1 SA 算法	2
1.1.1 算法流程	2
1.1.2 20 次随机实验的统计结果	3
1.1.3 目标函数的变化曲线	3
1.2 GA 算法	6
1.2.1 编码	6
1.2.2 算法流程	6
1.2.3 算法参数	7
1.2.4 20 次随机实验的统计结果	7
1.2.5 目标函数的变化曲线	7
<b>2 TSP 问题</b>	<b>8</b>
2.1 SA 算法	8
2.1.1 算法流程	8
2.1.2 20 次随机实验的统计结果	8
2.2 GA 算法	13
2.2.1 编码	13
2.2.2 算法流程	13
2.2.3 算法参数	13
2.2.4 20 次随机实验的统计结果	13
2.2.5 一次实验结果	13
<b>3 算法特点</b>	<b>17</b>
3.1 SA	17
3.2 GA	17
<b>4 TSP 可视化应用程序</b>	<b>17</b>
<b>5 实验分工</b>	<b>18</b>

# 1 求解多极小函数

求解多极值函数问题中, 我们需要求出函数的最值, 我们以函数  $f(x_0, x_1) = -x_1 * \sin(x_0) - x_0 * \cos(x_1)$  为例, 求它在  $x_0, x_1 \in [-10, 10]$  区间内的最小值。

使用 MATLAB 画出的函数图像如图1所示, 其最小值可用 matlab 库函数求出, 约为-17.394, 同时该函数有多个局部极小值。

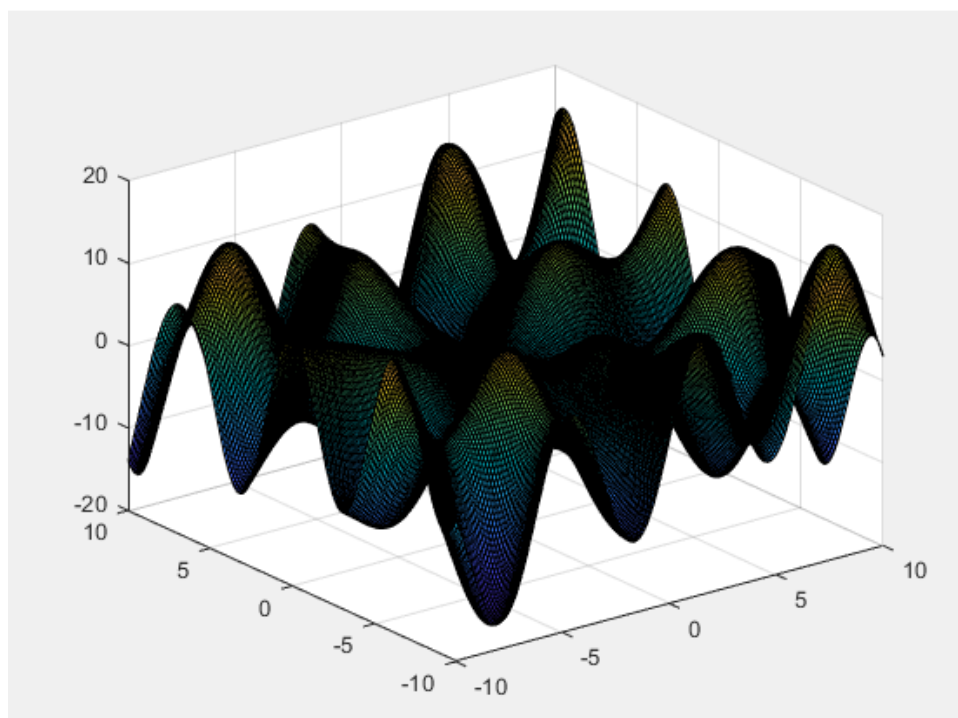


图 1: 函数  $f(x)$  图像

## 1.1 SA 算法

### 1.1.1 算法流程

如步骤图2、参数设置图3、流程图4, 步骤 1 中, 初始温度可以有两种生成方式: 按照经验设定固定初温或随机选择指定数量的状态计算目标值并把方差作为初温; 初始解在连续函数中为函数值。

步骤 2 中, 在外循环中, 温度随着迭代而发生变化, 收敛准则分为基于时间的收敛 (指定迭代次数, 需要指定迭代次数) 和基于性能的收敛 (连续若干步的目标值之差小于阈值, 需要指定阈值和迭代步数)。

步骤 3 中, 抽样稳定准则是在内循环中用来决定同一个温度下产生解的个数, 温度不随着内循环迭代而发生改变。收敛准则分为基于时间的收敛 (指定迭代次数, 需要指定迭代次数) 和基于性能的收敛 (连续若干步的目标值之差小于阈值, 需要指定阈值和迭代步数)。

步骤 4 为状态更新, 在连续函数的优化问题中, 需要指定自变量更新的步长和移动长度的分布类型 (高斯分布或柯西分布其中退温为对外循环的温度进行降温, 分为指数退温 (需要指定温度衰减系数) 和对数退温)。

步骤 5 为状态判断，如果新解的目标值更优，则用它来替换当前解，否则保持当前状态不变，并继续循环。

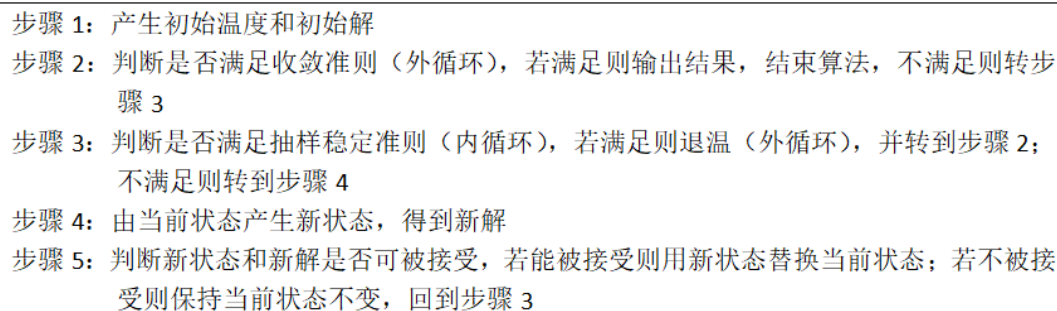


图 2: 模拟退火算法流程

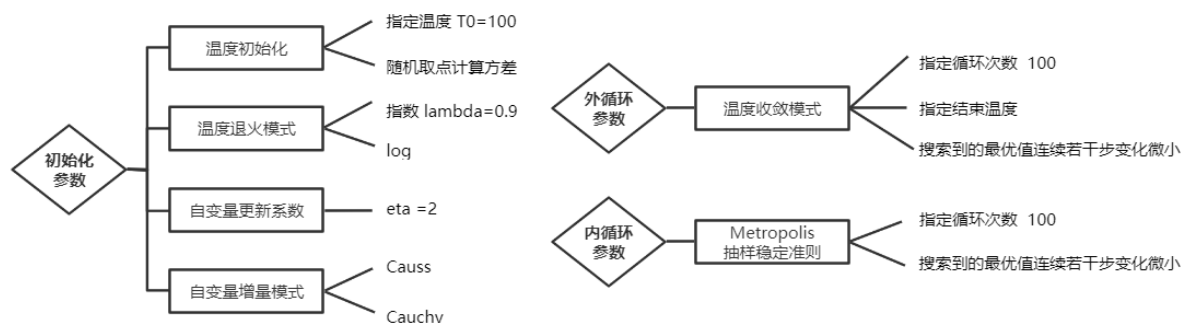


图 3: 模拟退火算法参数设置

### 1.1.2 20 次随机实验的统计结果

由于可选择的参数较多，这里的演示为指定了部分模式和参数来进行比较。不建议使用基于优化的收敛准则，因为需要指定阈值和迭代步数，在未知目标值范围的情况下随意迭代难以找到理想的解，有可能无法跳出循环。（改进思路：或许可以采用自适应阈值的算法。）

如图1.1.2, 修改外循环迭代次数，依次为 50、100、150。在状态转换步骤中，即使新状态并非最优，也有一定的概率可以跳转到新状态上。在这里，外循环迭代次数设置得不宜过高，因为最终温度会变得非常小，导致跳转概率的计算超过计算机所能存储的 64 位，故在退温步骤中设置温度不能小于  $1e-4$ 。

在迭代次数为 50 时，20 次循环也找不到最小值；迭代次数为 100 和 150 时能够找到最小值，说明适当增加迭代次数有助于进行更大范围的搜索；但 150 次的平均性能未必比 100 次好，找到的最差解和平均解都不如 100 次。这可能是因为在循环中设置了温度的下限，在 (130, 150] 次中进行迭代时，温度一直被限制在  $1e-4$ ，使得在出现了性能并非更优的解时的跳转概率很大，即无论解如何都要跳转，导致性能相对于 100 次时发生下降。

### 1.1.3 目标函数的变化曲线

如图8，算法在 35 个回合之后趋向于收敛。

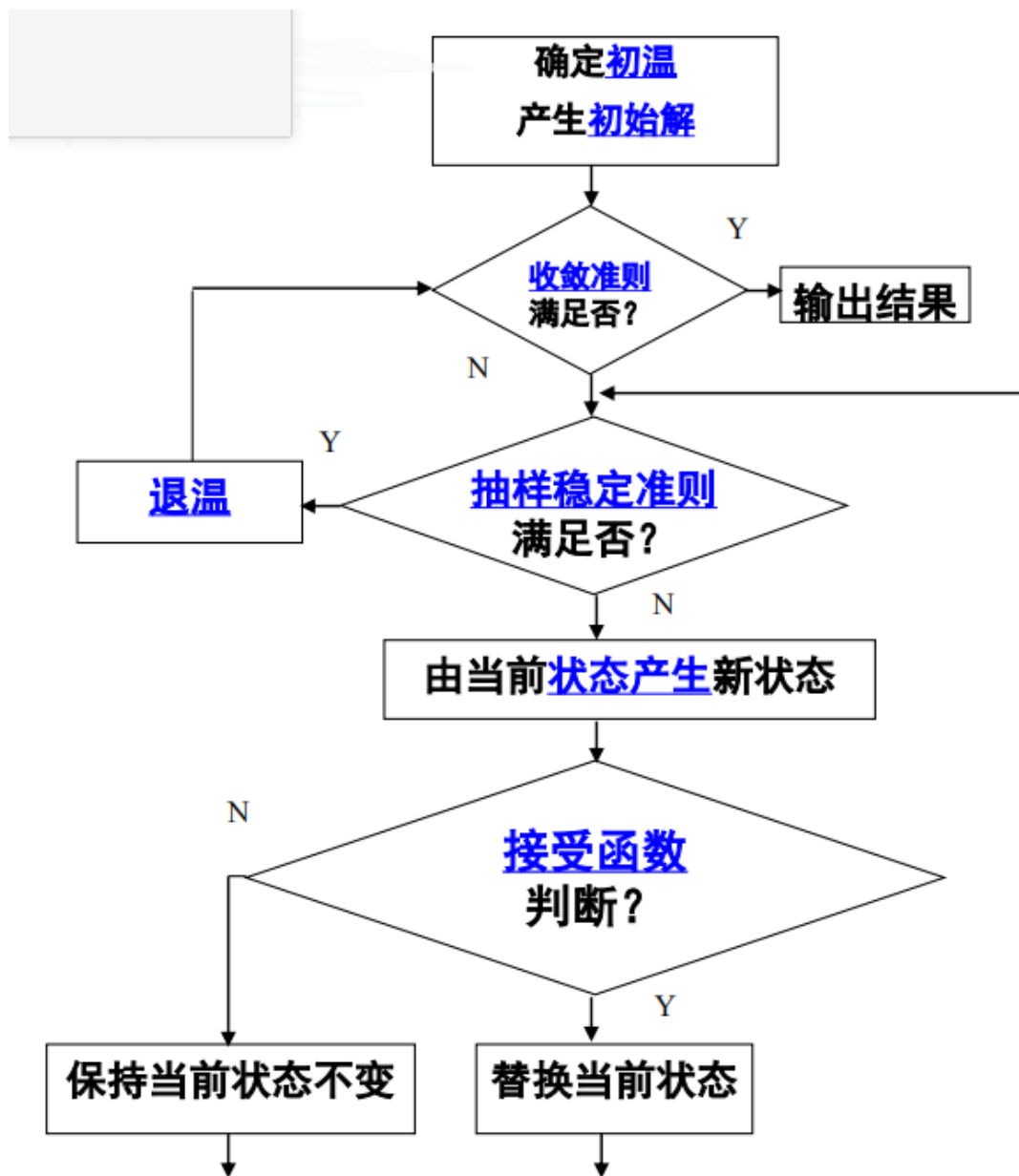


图 4: 模拟退火算法流程图

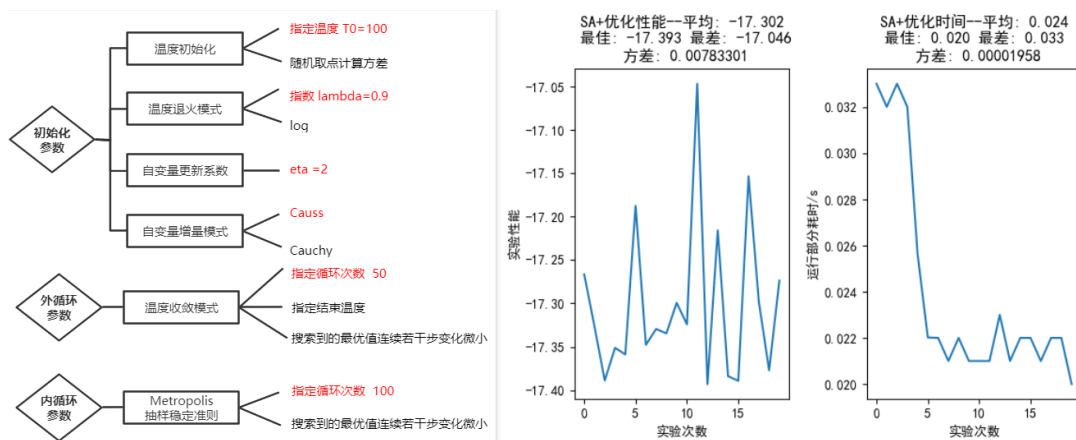


图 5: 外循环迭代 50 次

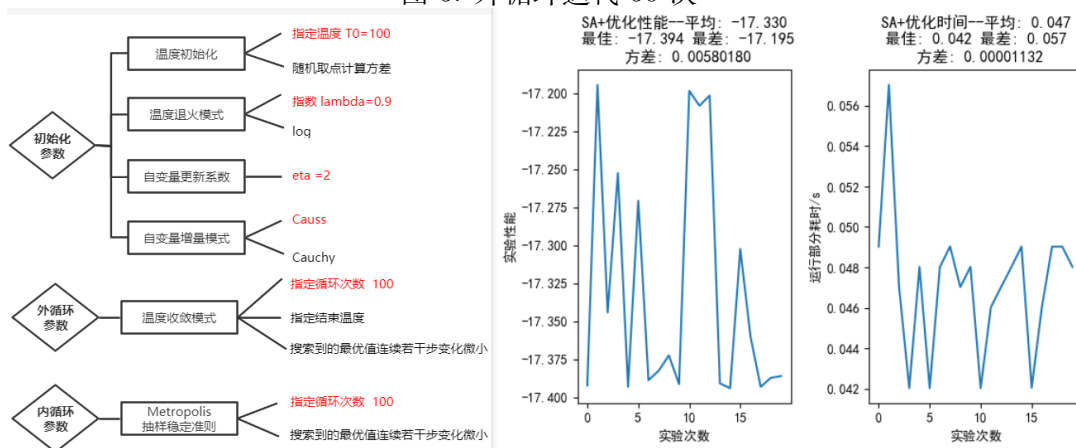


图 6: 外循环迭代 100 次

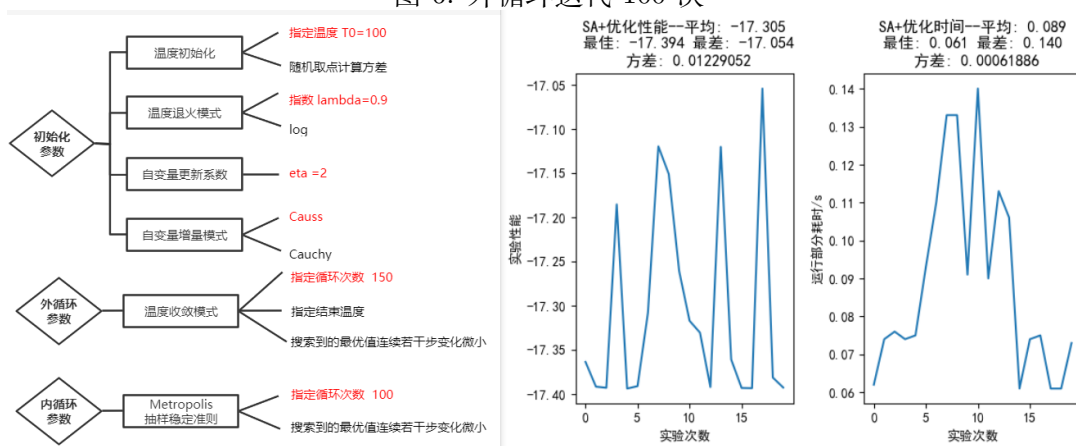


图 7: 外循环迭代 150 次

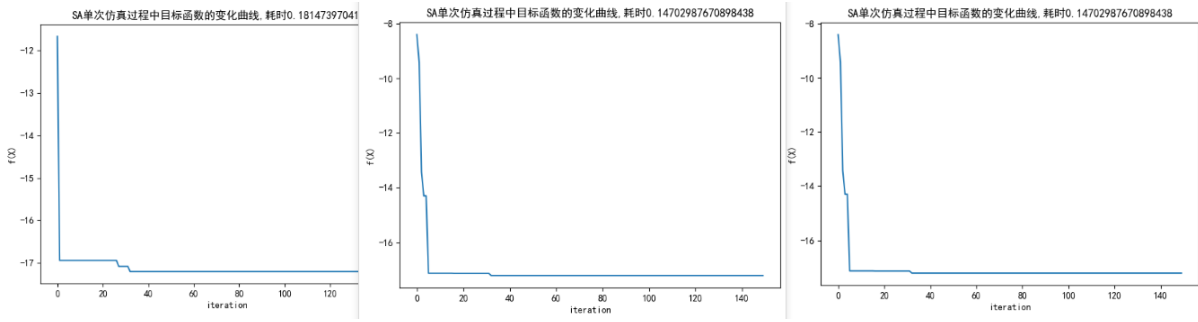


图 8: 连续优化问题, SA, 目标函数的变化曲线

## 1.2 GA 算法

### 1.2.1 编码

遗传算法需要对问题的解进行编码, 对此, 我们使用 32 位二进制码表示问题的解, 前 16 位表示  $x_0$ , 后 16 位表示  $x_1$ 。  $x_0, x_1$  的取值范围都为  $[-10, 10]$ , 使用 16 位二进制码表示可以使解精确到小数点后 3 位。二进制码  $[S]_2$  与实数  $x$  的关系为:

$$x = -10 + \frac{[S]_2}{2^{16} - 1} * (10 + 10)$$

### 1.2.2 算法流程

算法流程与标准的 GA 算法相同, 步骤与课件中相同, 如图9所示。其中, 适配值为函数值。在产生新种群时, 步骤 5 之前, 先将上一代种群排序, 保留一部分较优个体, 再从中生成下一代种群。

- 步骤 1: 令  $k=0$ , 随机产生  $N$  个初始个体构成初始种群  $P(0)$ 。
- 步骤 2: 评价  $P(k)$  中各个体的适配值(fitness value)。
- 步骤 3: 判断算法收敛准则是否满足。若满足则输出搜索结果; 否则执行以下步骤。
- 步骤 4: 令  $m = 0$ 。
- 步骤 5: 根据适配值大小以一定方式执行复制操作来从  $P(k)$  中选取两个个体。
- 步骤 6: 若交叉概率  $p_c > \xi \in [0, 1]$ , 则对选中个体执行交叉操作来产生两个临时个体; 否则将所选中父代个体作为临时个体。
- 步骤 7: 按变异概率  $p_m$  对临时个体执行变异操作产生两个新个体并放入  $P(k+1)$ , 并令  $m = m + 2$ 。
- 步骤 8: 若  $m < N$ , 则返回步骤 5; 否则令  $k = k + 1$  并返回步骤 2。

图 9: 遗传算法流程

算法的选择步骤中, 我们使用基于排名的方法, 先将种群个体按适配值排名, 以线性关系给每个排名的个体分配一个概率值, 然后归一化。倒数排名为  $i$  的个体被选择的概率值为  $\frac{i}{\sum_{j=1}^N j}$ 。此外, 选择操作之前, 父代种群中的适配值最高的两个个体会直接继承到子代种群。

交叉操作中, 我们对  $x_0$  染色体和  $x_1$  染色体分别进行交叉。为交叉一对染色体, 我们随机产生一个交叉点, 交换该点左右的基因, 生成一对新染色体。

变异操作中，我们对染色体中的每个基因（01 二进制数）按变异概率进行随机改变。

我们采用保优操作，即保存算法出现过的适配值最高的解。如果保优值在若干回合中一直保持不变，认为算法收敛，提前结束迭代，同时我们也有最高迭代次数的限制。

### 1.2.3 算法参数

- 最大迭代次数：2000
- 种群规模：30
- 交叉概率：0.95
- 变异概率：0.1
- 保优值保持不变提前结束迭代的回合数：100
- 产生新种群时先从上一代种群选出一部分较优个体，其比例：1.0

### 1.2.4 20 次随机实验的统计结果

统计结果如图10所示，平均性能：17.394，最佳性能：17.394，最差性能：17.393，方差：0.00000019。

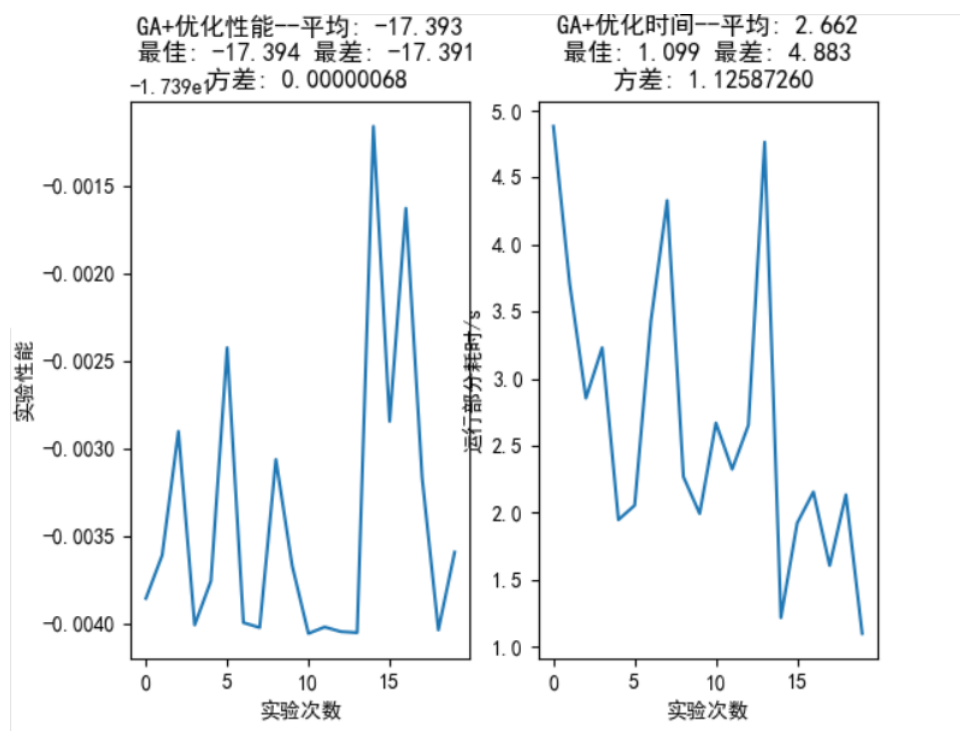


图 10: 函数优化问题中，GA 算法 20 次随机实验的统计结果

### 1.2.5 目标函数的变化曲线

在一次实验中目标函数的变化曲线如图11所示。



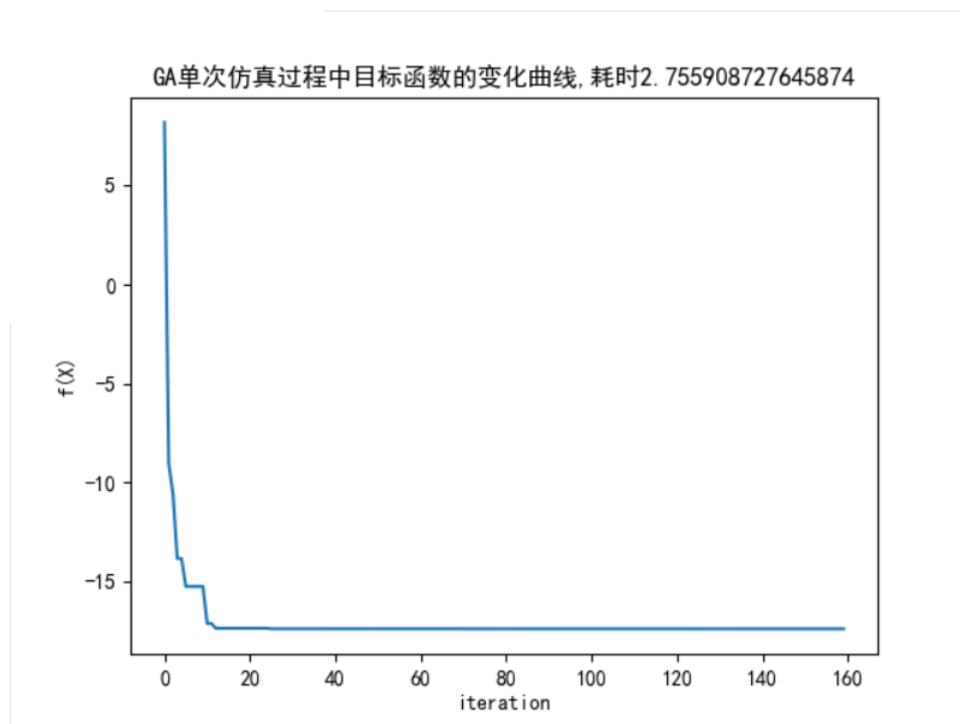


图 11: 函数优化问题中, GA 算法目标函数的变换曲线

## 2 TSP 问题

我们以网络学堂中 TSP30 文件中的点信息做测试。

### 2.1 SA 算法

#### 2.1.1 算法流程

算法流程与上文基本类似, 不同之处在于:

步骤 1 中, 初始温度可以有两种生成方式: 按照经验设定固定初温要更大, 随机选择指定数量计算方差时指定的元素数目也大得多; 初始解在离散的 TSP 问题中为旅行商所经城市的路线。

步骤 4 为状态更新, 在 TSP 问题中, 可以通过进行两位置元素交换、两位置之间的元素逆序、把指定某一段序列插到第三个位置之后或者把上述操作按照概率混合的方法产生新的状态, 即新的旅行商路线。

#### 2.1.2 20 次随机实验的统计结果

##### 1)30 个城市:

最短距离为 423.74, 0.7-3.6s 可找到最优解。

如图12, 从上到下依次指定外循环迭代 100 次、内循环迭代 200 次; 外循环迭代 100 次、内循环迭代 400 次; 外循环迭代 200 次、内循环迭代 400 次; 外循环迭代 200 次、内循环迭代 500 次, 所有参数的设置见图像12左半部分。



在外循环迭代 100 次、内循环迭代 200 次中，虽然随机 20 次实验中能够找到最优解（距离 423.741），但从图中可以看出这样的次数并不多，且距离方差很大。增大内循环迭代次数到 400，明显在 20 次实验中找到最优解的次数增加，距离方差也变小很多。

在外循环迭代 200 次、内循环迭代 400 次中，20 次随机试验中能够找到最优解的次数高于外循环迭代 100 次、内循环迭代 400 次；而将内循环迭代次数继续提高到 500 次时，性能又有了明显提升——最差距离与最优解仅仅相差不到 2 个单位，方差迅速降低到 0.24。

上述实验说明，在一定范围之内提高迭代次数有利于提升性能，但迭代次数设置得是否合理，是需要结合初始温度来进行考虑。迭代次数越多耗时越长，需要在性能与耗时方面做出权衡。

## 2)50 个城市：

最短距离为 428.10，0.9-9.6s 内可找到最优解。

如图13，参数设置与 30 个城市基本相同，仅改变内外循环次数进行实验。从上到下依次为外循环 100 次、内循环 200 次；外循环 100 次、内循环 400 次；外循环 200 次、内循环 400 次；外循环 200 次、内循环 1000 次。图中右半部分为 20 次实验中的最优路径，均没有发生路径交叉。随着迭代次数的增加，搜寻到的最短路径在减小，平均搜寻路径和方差也在减小，运行时间在增加。但由于状态更新的随机性，有时候未必能够得到性能优化。

## 2)75 个城市：

最短距离为 544.20，37-173s 内可找到最优解。

如图14，从上到下依次为初始温度 700、外循环迭代 200 次、内循环迭代 3000 次；初始温度 700、外循环迭代 200 次、内循环迭代 5000 次；初始温度 700，温度收敛到  $1e-5$ 、内循环迭代 5000 次。图中左侧为 20 次实验所得结果，右图为每 20 次实验的最短路径。到了 75 个城市时，使用之前的参数已经无法获取最优解，因此需要进行参数调整。为了增加搜寻的路径范围，此时设置的参数值较 TSP30、TSP50 有大幅度增加。尽管如此，搜寻到最优解的概率仍然很小，运行多次也只能得到一次 544.20，连 546、547、548 这样的数值都很少能够搜寻得到。

## 4) 总结：

在 50 个城市与 75 个城市中，由于城市较多，形成闭合路径的方法有很多，此时已经无法通过目测来判断每 20 次实验的最优解是否为最短路径。增加内循环次数的本质是在状态转换概率没有太大改变的情况下增加尝试的路径，以便尽可能覆盖较优解。增加外循环次数的本质是改变解的跳转概率，能够跳出局部最优，继续寻找全局最优解。

随着城市数量的增加，问题的复杂度变得很高，每次实验的运行时间变得很长，更难以搜寻到最优解，也无法通过肉眼判断是否搜寻到了最优，因为每次实验的路径都没有出现交叉。如图15，随着问题规模的增长，距离开始明显下降的迭代次数开始上升（TSP30 在 25 前后，TSP50 在 30-40 之间，TSP75 在 30-40 之间），距离趋向于收敛的次数也在上升（TSP30 在 50 之前，TSP50 在 50-60 之间，TSP75 大于 60）。说明盲目增大外循环迭代次数意义不大。问题规模对于 SA 算法的影响很大，在较复杂的问题下，SA 性能并不算理想；中小规模下 SA 算是一种兼顾了时间和性能的计算法。

**注：**实验表明，在产生新状态的过程中，新状态应当基于前一个产生的新状态而不是当前状态来产生，即使前一个新状态没有被接受，不能成为当前状态。这样运行多次，新状态与当前状态将有较大不同，更有可能找到最优路径。若所有新状态都是基于当前状态产生，则路径

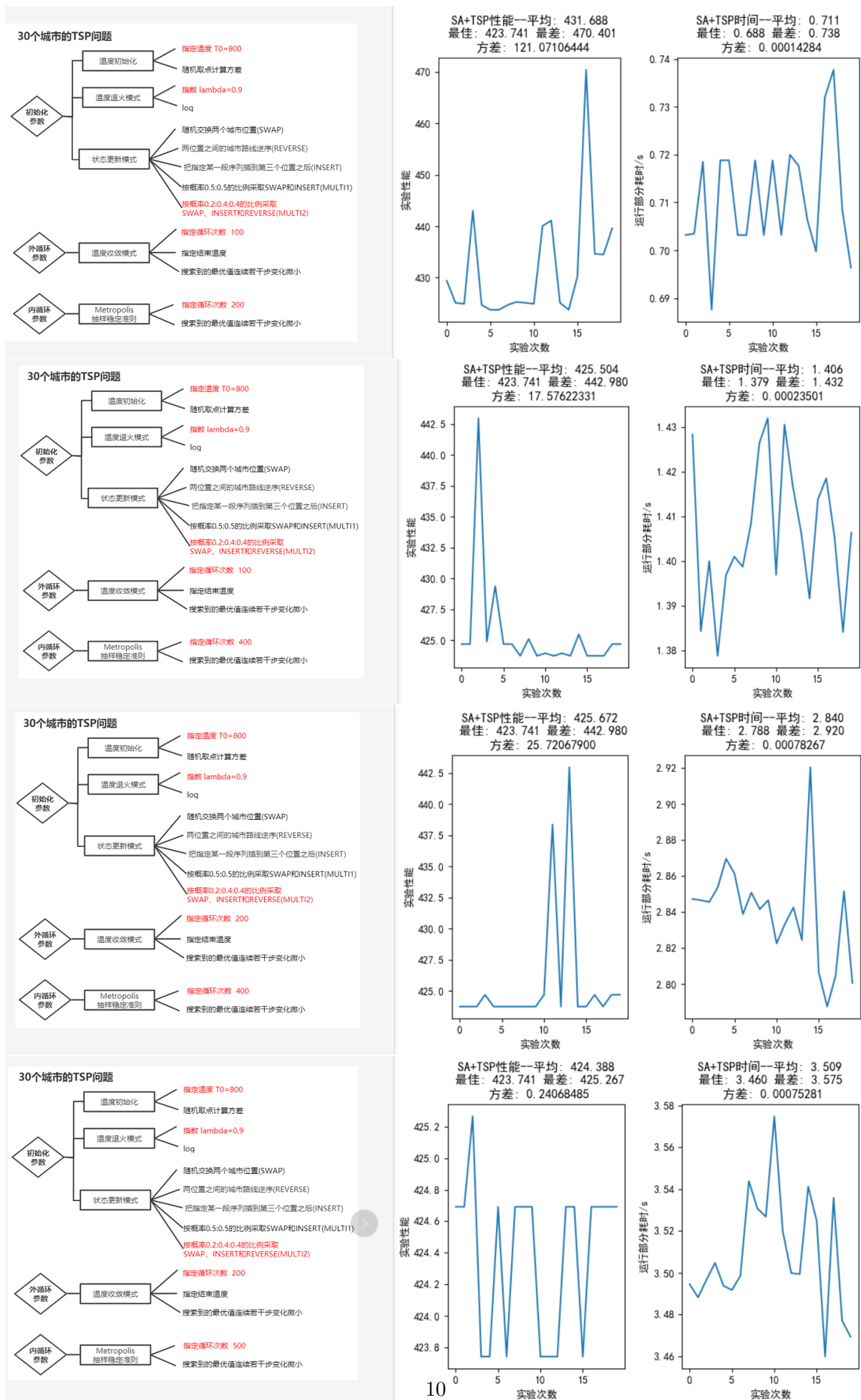


图 12: TSP30, SA 算法

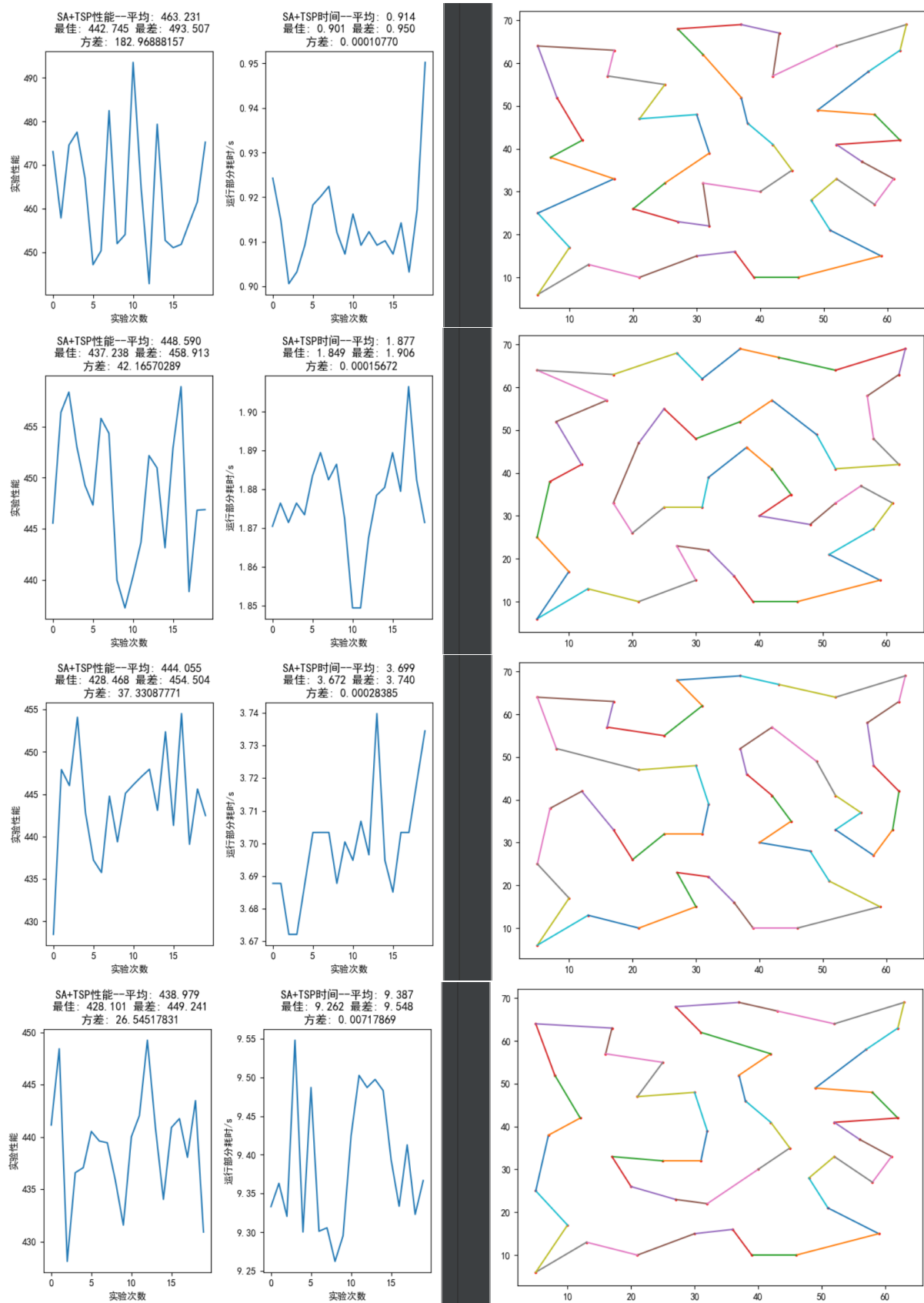


图 13: TSP50, SA 算法

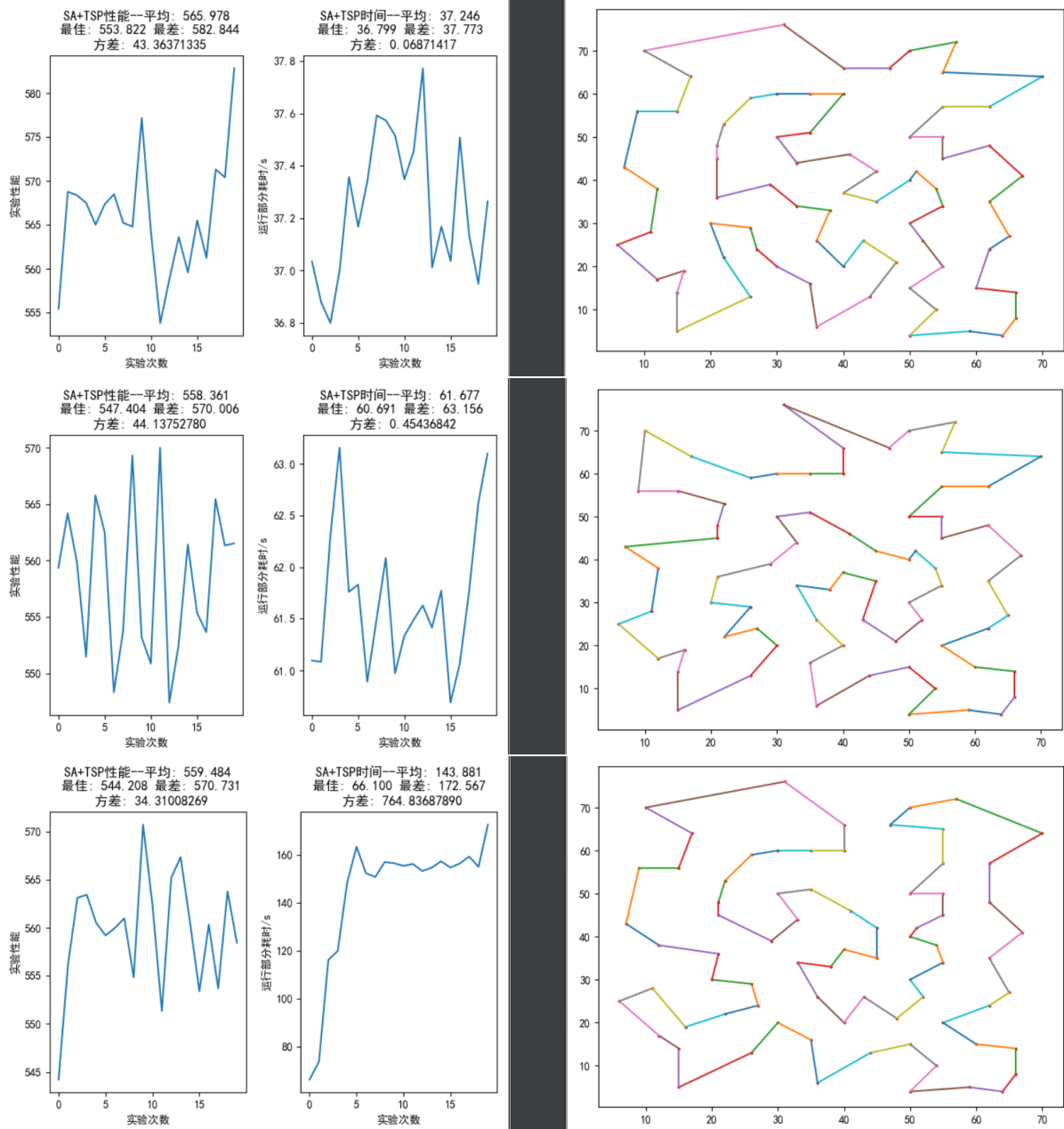


图 14: TSP75, SA 算法

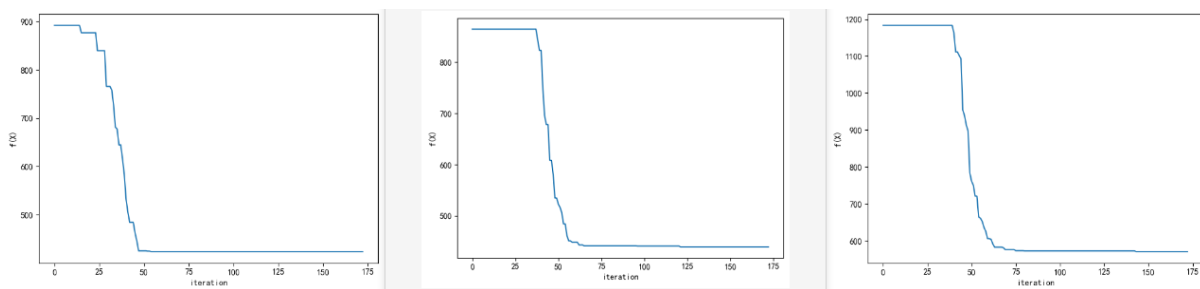


图 15: TSP30、50、75 最短距离随着外循环迭代次数的变化

是否变短会十分依赖于初始产生的路径，多次迭代也相当于只在当前路径上尝试小的城市变动，路径很难缩短，难以找到最优。

## 2.2 GA 算法

### 2.2.1 编码

用路径顺序进行编码，比如 5 个城市路径：[4,2,3,0,1]。

### 2.2.2 算法流程

算法流程与前一问题中描述的流程相同。

对于交叉操作，我们尝试了部分映射交叉和单位位置次序交叉两种方法，实验发现单位制次序交叉方法的性能更优。

变异操作中，我们对染色体中的每个基因（某城市）按变异概率随机与路径中另一城市交换次序。

与函数优化问题相同，我们在该问题中同样采用保优操作。

### 2.2.3 算法参数

- 最大迭代次数：2000
- 种群规模：60
- 交叉概率：0.95
- 变异概率：0.2
- 保优值保持不变提前结束迭代的回合数：500
- 产生新种群时先从上一代种群选出一部分较优个体，其比例：0.2

### 2.2.4 20 次随机实验的统计结果

30 个城市的统计结果如图16所示，平均性能：441.690，最佳性能：423.741，最差性能：478.440，方差：281.369。

50 个城市的统计结果如图16所示，平均性能：456.699，最佳性能：430.346，最差性能：474.503，方差：135.855。

75 个城市的统计结果如图16所示，平均性能：593.437，最佳性能：567.233，最差性能：635.567，方差：250.265。

### 2.2.5 一次实验结果

在某次实验中，目标函数（总距离）变化过程如图19所示。

最优结果的总距离为 423.741，路径图如图20所示，路径顺序如图21。

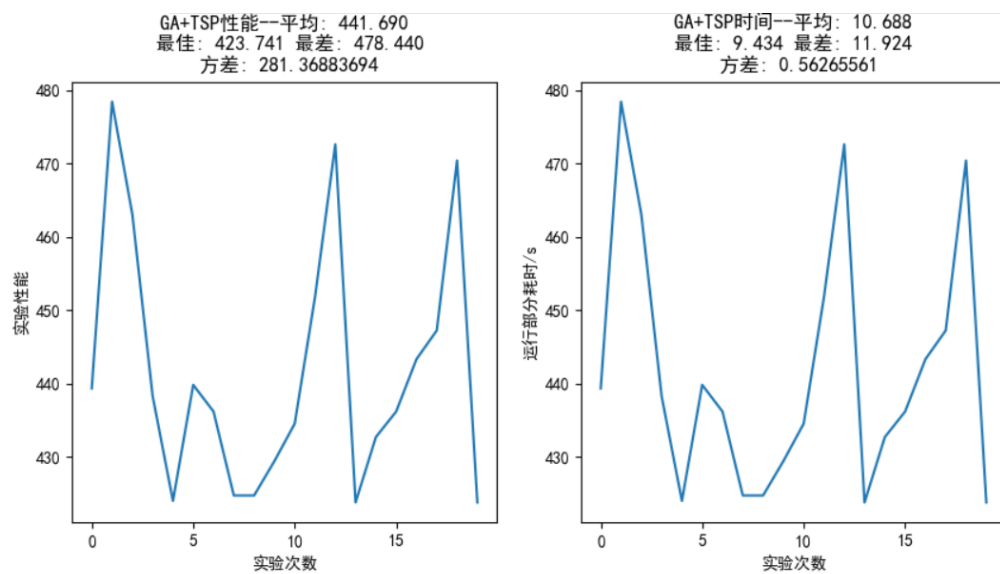


图 16: TSP 30 个城市问题中, GA 算法 20 次随机实验的统计结果

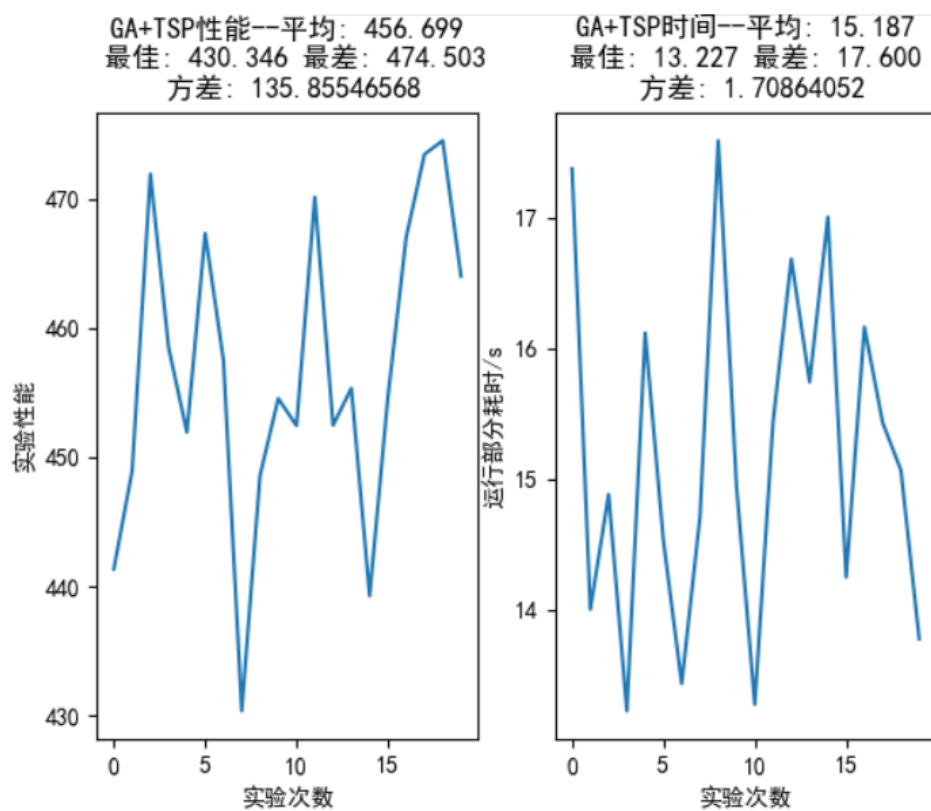


图 17: TSP 50 个城市问题中, GA 算法 20 次随机实验的统计结果

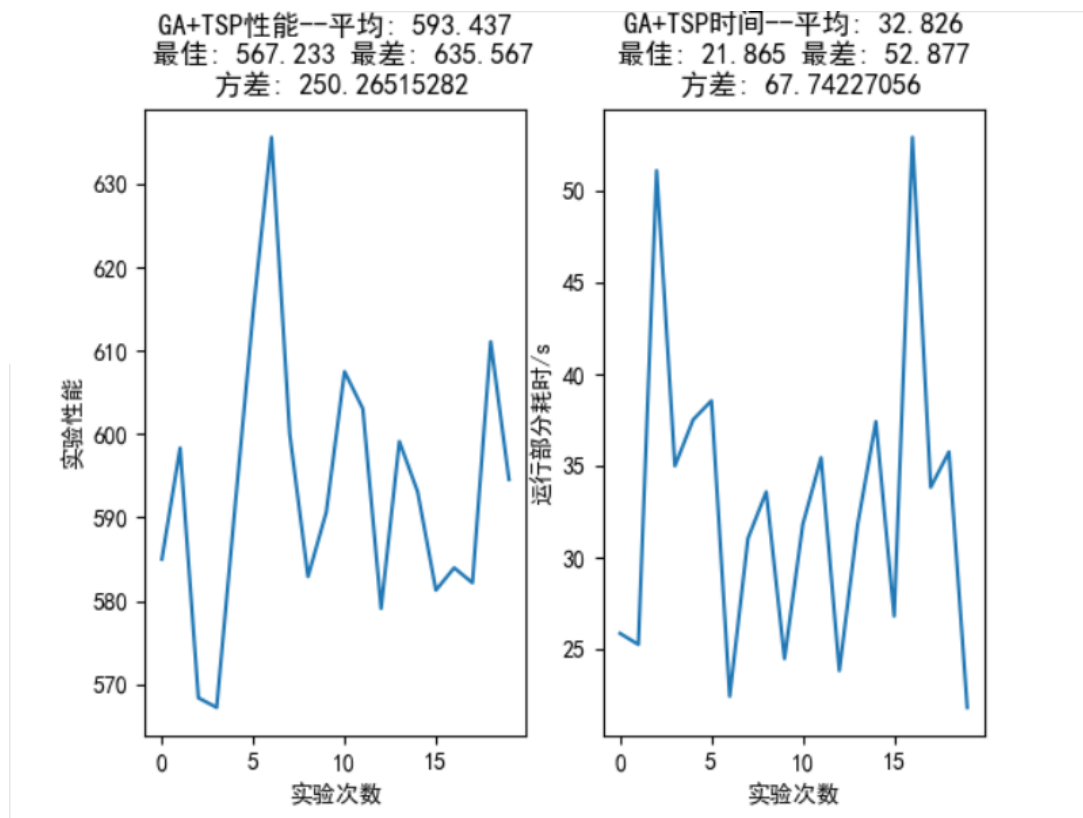


图 18: TSP 75 个城市问题中, GA 算法 20 次随机实验的统计结果

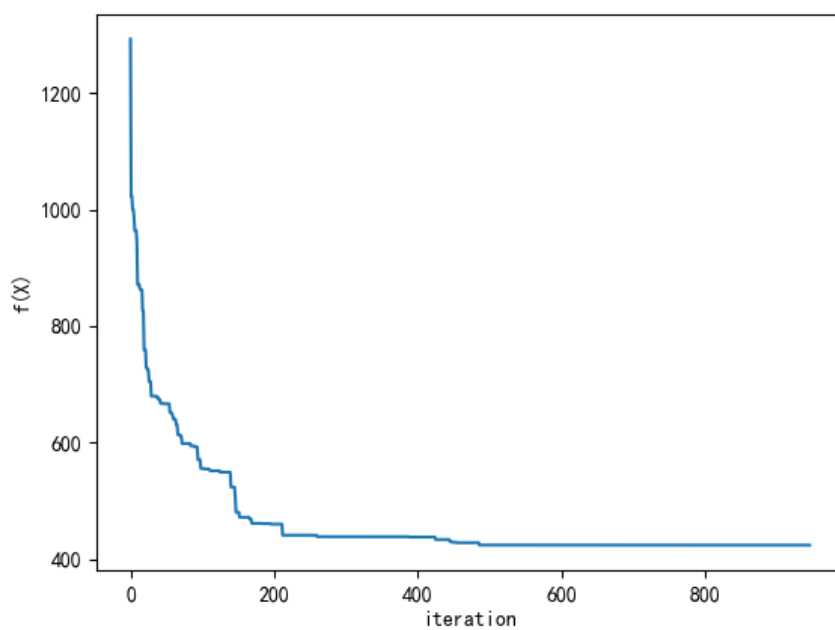


图 19: TSP 问题中, GA 算法一次实验目标函数的变化



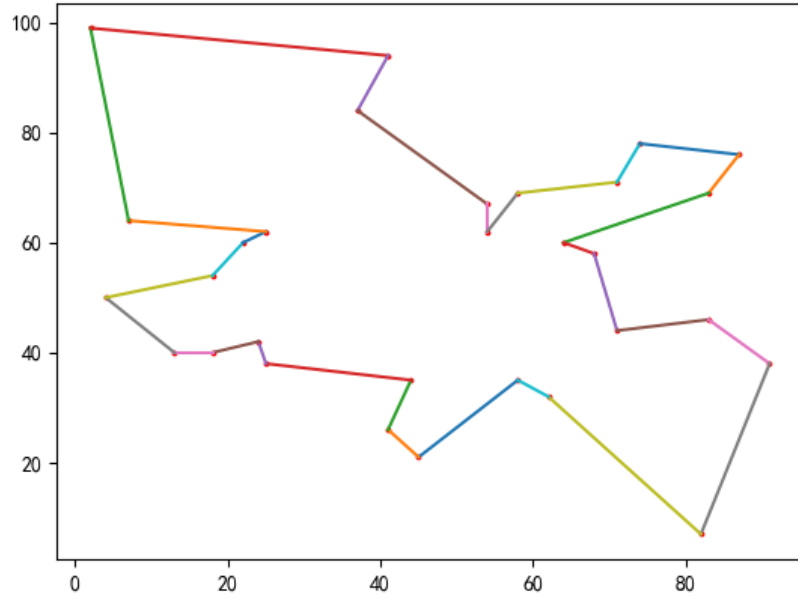


图 20: TSP 问题中, GA 算法最优结果的路径图

```

Iteration: 933 best_individual: [25, 26, 27, 28, 15, 16, 21, 22, 29, 11, 12, 3, 4, 5, 0, 1, 2, 8, 17, 18, 19, 28, 9, 10, 6, 7, 13, 14, 23, 24] best_fitness: -423.741 nochange_iter:11
Iteration: 934 best_individual: [25, 26, 27, 28, 15, 16, 21, 22, 29, 11, 12, 3, 4, 5, 0, 1, 2, 8, 17, 18, 19, 28, 9, 10, 6, 7, 13, 14, 23, 24] best_fitness: -423.741 nochange_iter:10
Iteration: 935 best_individual: [25, 26, 27, 28, 15, 16, 21, 22, 29, 11, 12, 3, 4, 5, 0, 1, 2, 8, 17, 18, 19, 28, 9, 10, 6, 7, 13, 14, 23, 24] best_fitness: -423.741 nochange_iter:9
Iteration: 936 best_individual: [25, 26, 27, 28, 15, 16, 21, 22, 29, 11, 12, 3, 4, 5, 0, 1, 2, 8, 17, 18, 19, 28, 9, 10, 6, 7, 13, 14, 23, 24] best_fitness: -423.741 nochange_iter:8
Iteration: 937 best_individual: [25, 26, 27, 28, 15, 16, 21, 22, 29, 11, 12, 3, 4, 5, 0, 1, 2, 8, 17, 18, 19, 28, 9, 10, 6, 7, 13, 14, 23, 24] best_fitness: -423.741 nochange_iter:7
Iteration: 938 best_individual: [25, 26, 27, 28, 15, 16, 21, 22, 29, 11, 12, 3, 4, 5, 0, 1, 2, 8, 17, 18, 19, 28, 9, 10, 6, 7, 13, 14, 23, 24] best_fitness: -423.741 nochange_iter:6
Iteration: 939 best_individual: [25, 26, 27, 28, 15, 16, 21, 22, 29, 11, 12, 3, 4, 5, 0, 1, 2, 8, 17, 18, 19, 28, 9, 10, 6, 7, 13, 14, 23, 24] best_fitness: -423.741 nochange_iter:5
Iteration: 940 best_individual: [25, 26, 27, 28, 15, 16, 21, 22, 29, 11, 12, 3, 4, 5, 0, 1, 2, 8, 17, 18, 19, 28, 9, 10, 6, 7, 13, 14, 23, 24] best_fitness: -423.741 nochange_iter:4
Iteration: 941 best_individual: [25, 26, 27, 28, 15, 16, 21, 22, 29, 11, 12, 3, 4, 5, 0, 1, 2, 8, 17, 18, 19, 28, 9, 10, 6, 7, 13, 14, 23, 24] best_fitness: -423.741 nochange_iter:3
Iteration: 942 best_individual: [25, 26, 27, 28, 15, 16, 21, 22, 29, 11, 12, 3, 4, 5, 0, 1, 2, 8, 17, 18, 19, 28, 9, 10, 6, 7, 13, 14, 23, 24] best_fitness: -423.741 nochange_iter:2
Iteration: 943 best_individual: [25, 26, 27, 28, 15, 16, 21, 22, 29, 11, 12, 3, 4, 5, 0, 1, 2, 8, 17, 18, 19, 28, 9, 10, 6, 7, 13, 14, 23, 24] best_fitness: -423.741 nochange_iter:1
Iteration: 944 best_individual: [25, 26, 27, 28, 15, 16, 21, 22, 29, 11, 12, 3, 4, 5, 0, 1, 2, 8, 17, 18, 19, 28, 9, 10, 6, 7, 13, 14, 23, 24] best_fitness: -423.741 nochange_iter:0
Iteration: 945 best_individual: [25, 26, 27, 28, 15, 16, 21, 22, 29, 11, 12, 3, 4, 5, 0, 1, 2, 8, 17, 18, 19, 28, 9, 10, 6, 7, 13, 14, 23, 24] best_fitness: -423.741 nochange_iter:-1
libpng warning: iccP: cHRM chunk does not match sRGB

```

图 21: TSP 问题中, GA 算法最优结果的路径

## 3 算法特点

### 3.1 SA

- 1) 有概率突跳性，能够以一定概率接受较差的解，从而跳出局部最优，找到全局最优解。
- 2) 串行运行，当问题规模较大时，运行时间显著增加。
- 3) 是一种迭代寻优的方法，运行性能与迭代次数密切相关。
- 4) 是一种随机寻优方法，最优解能否找到与问题规模相关，具有较大随机性。

### 3.2 GA

- 1) 算法通过模仿自然界中遗传进化的模式，来实现寻优，适用范围较广，在离散优化问题和连续优化问题中都能使用。
- 2) 是一种并行算法，多个个体同时搜索，易于实现算法的并行化处理。
- 3) 由于遗传算法使用多个个体同时搜索，所以算法不易陷入局部极值点，寻优能力较强。
- 4) 可以通过控制算法的变异交叉概率来平衡算法的全局搜索能力和局部探索能力，变异交叉概率较高时，全局搜索能力较强，反之局部探索能力较强。

## 4 TSP 可视化应用程序

用途：用来显示单次实验中 TSP 路线随着迭代次数增加的实时变化情况。可以使用老师提供的 TSP30、50、75 个城市的 txt 文件，也可以随机生成指定点数的城市，如图22

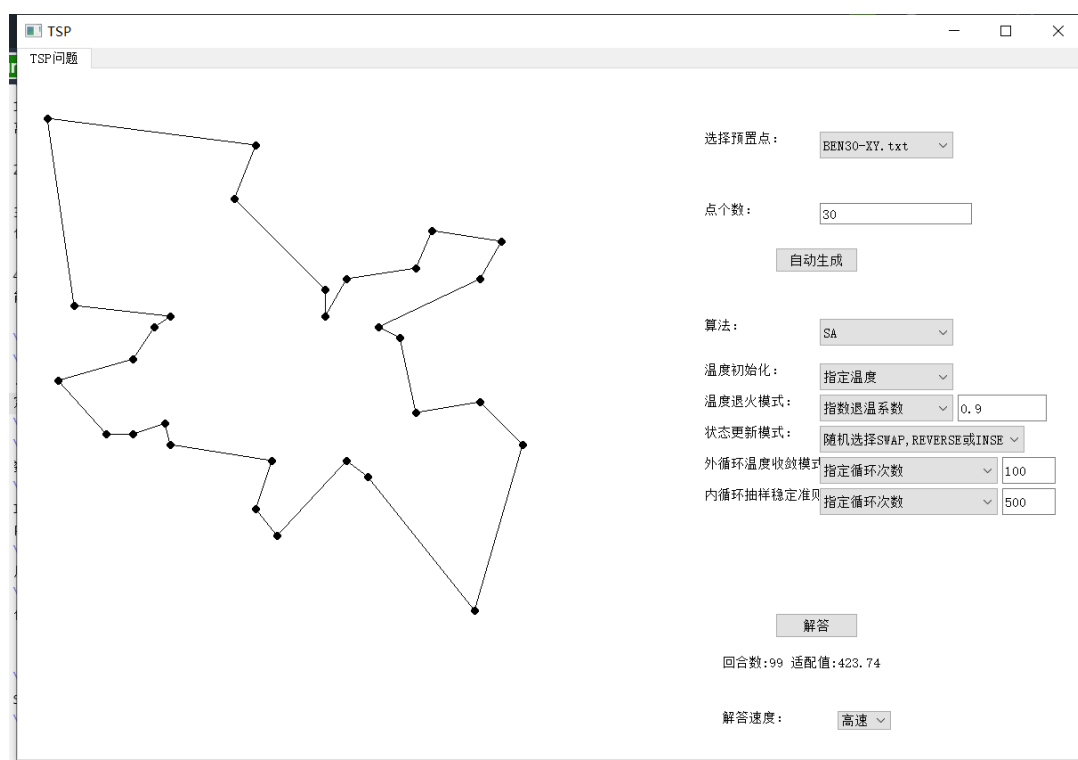


图 22: TSP 应用程序

应用程序的用法及其可执行文件的下载链接见 README.md 文件。

注：

1) 由于 SA 可选模式较多，ui 难以呈现，故 ui 中仅显示 SA 算法基于迭代次数收敛的选项 (也是最理想的参数设置方案)。

2) 在 SA 中，状态更新模式对结果影响较大，默认模式为效果较好的 MULTI2；循环迭代次数对结果影响也很大。建议参数：TSP30/50(外循环 200 次、内循环 500 次)、TSP75(外循环 200 次、内循环 3000 次)。

3) 由于报告中的数据分析是在非 ui 界面下运行的 (main\_TSP.py)，与使用 ui(ui\_TSP.py) 运行同一个算法的时间会有出入。

4) 运行非 ui(Optimization-algorithm-main/main\_TSP.py): 可改参数位于 394-422 行。运行 ui 可执行程序: 显示图形界面。

## 5 实验分工

SA 算法：周映佟

GA 算法：吾尔开希

应用程序：吾尔开希 周映佟