系统工程 张云佳 学号：1800900

|  |  |
| --- | --- |
| 禁忌表 | |
| 1 |  |
| 2 |  |
| 3 |  |

第二题：T 表长度设为 3，NG 设为 5。

初始解：1-2-3-4-5

历史最优值c() = 45，当前值c(x) = 10

1. 第一次迭代分析：

领域解集：

|  |  |
| --- | --- |
| 移动s(x) | 移动后的当前值c(x) |
| 2 1 3 4 5 (1,2) | 59 |
| 3 2 1 4 5 (1,3) | 44 |
| 4 2 3 1 5 (1,4) | 43 |
| 5 2 3 4 1 (1,5) | 45 |
| 1 3 2 4 5 (2,3) | 43 |
| 1 4 3 2 5 (2,4) | 45 |
| 1 5 3 4 2 (2,5) | 60 |
| 1 2 4 3 5 (3,4) | 60 |
| 1 2 5 4 3 (3,5) | 59 |
| 1 2 3 5 4 (4,5) | 44 |

|  |  |
| --- | --- |
| 禁忌表 | |
| 1 | (1,4) |
| 2 |  |
| 3 |  |

当前领域中目标函数值改善最大的移动是 (1,4)

和(2,3)，都不在禁忌表中，选择(1,4)

第一次迭代后，当前解变为：4-2-3-1-5

历史最优值c() = 43，当前值c(x) = 43

禁忌表如右图所示

1. 第二次迭代分析：

领域解集：

|  |  |
| --- | --- |
| 移动s(x) | 移动后的当前值c(x) |
| 2 4 3 1 5 (2,4) | 59 |
| 3 2 4 1 5 (3,4) | 44 |
| 1 2 3 4 5 (1,4) | 45 |
| 5 2 3 1 4 (4,5) | 43 |
| 4 3 2 1 5 (2,3) | 45 |
| 4 1 3 2 5 (1,2) | 43 |
| 4 5 3 1 2 (2,5) | 58 |
| 4 2 1 3 5 (1,3) | 58 |
| 4 2 5 1 3 (3,5) | 59 |
| 4 2 3 5 1 (1,5) | 44 |

|  |  |
| --- | --- |
| 禁忌表 | |
| 1 | (4,5) |
| 2 | (1,4) |
| 3 |  |

当前领域中目标函数值改善最大的移动是 (4,5)

和(1,2)，都不在禁忌表中，选择(4,5)

第二次迭代后，当前解变为：5-2-3-1-4

历史最优值c() = 43，当前值c(x) = 43

新的禁忌表如右图所示

1. 第三次迭代分析：

领域解集：

|  |  |
| --- | --- |
| 移动s(x) | 移动后的当前值c(x) |
| 2 5 3 1 4 (2,5) | 58 |
| 3 2 5 1 4 (3,5) | 45 |
| 1 2 3 5 4 (1,5) | 44 |
| 4 2 3 1 5 (4,5) | 43 |
| 5 3 2 1 4 (2,3) | 44 |
| 5 1 3 2 4 (1,2) | 43 |
| 5 4 3 1 2 (2,4) | 59 |
| 5 2 1 3 4 (1,3) | 59 |
| 5 2 4 1 3 (3,4) | 58 |
| 5 2 3 4 1 (1,4) | 45 |

当前领域中目标函数值改善最大的移动是 (4,5)

|  |  |
| --- | --- |
| 禁忌表 | |
| 1 | (1,2) |
| 2 | (4,5) |
| 3 | (1,4) |

和(1,2)，(4,5)在禁忌表中，选择(1,2)

第三次迭代后，当前解变为：5-1-3-2-4

历史最优值c() = 43，当前值c(x) = 43

新的禁忌表如右图所示

1. 第四次迭代分析：

领域解集：

|  |  |
| --- | --- |
| 移动s(x) | 移动后的当前值c(x) |
| 1 5 3 2 4 (1,5) | 44 |
| 3 1 5 2 4 (3,5) | 59 |
| 2 1 3 5 4 (2,5) | 58 |
| 4 1 3 2 5 (4,5) | 43 |
| 5 3 1 2 4 (1,3) | 58 |
| 5 2 3 1 4 (1,2) | 43 |
| 5 4 3 2 1 (1,4) | 45 |
| 5 1 2 3 4 (2,3) | 45 |
| 5 1 4 2 3 (3,4) | 44 |
| 5 1 3 4 2 (2,4) | 59 |

当前领域中目标函数值改善最大的移动是 (4,5)

|  |  |
| --- | --- |
| 禁忌表 | |
| 1 | (1,5) |
| 2 | (1,2) |
| 3 | (4,5) |

和(1,2)，都在禁忌表中，且没有达到渴望水平，

所以选择次优的移动(1,5)

第四次迭代后，当前解变为：1-5-3-2-4

历史最优值c() = 43，当前值c(x) = 44

新的禁忌表如右图所示

1. 第五次迭代分析：

领域解集：

|  |  |
| --- | --- |
| 移动s(x) | 移动后的当前值c(x) |
| 5 1 3 2 4 (1,5) | 43 |
| 3 5 1 2 4 (1,3) | 60 |
| 2 5 3 1 4 (1,2) | 58 |
| 4 5 3 2 1 (1,4) | 44 |
| 1 3 5 2 4 (3,5) | 57 |
| 1 2 3 5 4 (2,5) | 44 |
| 1 4 3 2 5 (4,5) | 45 |
| 1 5 2 3 4 (2,3) | 45 |
| 1 5 4 2 3 (3,4) | 43 |
| 1 5 3 4 2 (2,4) | 60 |

|  |  |
| --- | --- |
| 禁忌表 | |
| 1 | (3,4) |
| 2 | (1,5) |
| 3 | (1,2) |

当前领域中目标函数值改善最大的移动是 (1,5)

和(3,4)，(1,5)在禁忌表中， 所以选移动(3,4)

第五次迭代后，当前解变为：1-5-4-2-3

历史最优值c() = 43，当前值c(x) = 43

新的禁忌表如右图所示

迭代完成，最终结果1-5-4-2-3，c(x) = 43

Python代码如下：

city\_number = 5 *# 城市总数*table\_length = 3 *# 禁忌表长度*taboo\_table = [] *# 禁忌表  
  
# 两个城市之间的距离矩阵*distance\_matrix = [[0, 10, 15, 6, 2], [10, 0, 8, 13, 9], [15, 8, 0, 20, 15], [6, 13, 20, 0, 5], [2, 9, 15, 5, 0]]  
  
*# 计算领域解中所有路径的距离***def** cal\_path(distance\_matrix, path):  
 dis\_list = []  
 **for** each **in** path:  
 dis = 0  
 **for** j **in** range(city\_number - 1):  
 dis = distance\_matrix[each[j]][each[j + 1]] + dis  
 dis = distance\_matrix[each[4]][each[0]] + dis *# 回到起点* dis\_list.append(dis)  
 **return** dis\_list  
  
*# 寻找上一个最优路径对应的所有领域解.***def** find\_path(path\_best):  
 path\_new = []  
 **for** i **in** range(city\_number - 1):  
 **for** j **in** range(i + 1, city\_number):  
 path = path\_best.copy() *# 复制* path[i], path[j] = path[j], path[i] *# 2-opt* path\_new.append(path)  
 **return** path\_new  
  
*# 设置初始解*path\_initial = []  
firstValue = [1, 2, 3, 4, 5] *# 初始解为1,2,3,4,5*list = [x - 1 **for** x **in** firstValue] *# Python中转为0,1,2,3,4*initial = list  
path\_initial.append(initial)  
  
*# 加入禁忌表  
# 将移动后的结果放入禁忌表，等同于将移动放入禁忌表*taboo\_table.append(initial)  
  
*# 求初始解的路径长度*dis\_list = cal\_path(distance\_matrix, path\_initial)  
dis\_best = min(dis\_list) *# 最短距离*path\_best = path\_initial[dis\_list.index(dis\_best)] *# 对应的最短路径方案  
  
# 初始渴望*expect\_dis = dis\_best  
expect\_path = path\_best  
print(**'禁忌表迭代过程：'**)  
**for** iter **in** range(5): *# 迭代5次  
 # 寻找新的领域解* path\_new = find\_path(path\_best)  
 *# 求出所有新解的路径长度* dis\_new = cal\_path(distance\_matrix, path\_new)  
 *# 选择新的最短路径* dis\_best = min(dis\_new) *# 最短距离* path\_best = path\_new[dis\_new.index(dis\_best)] *# 对应的最短路径方案* **if** dis\_best < expect\_dis: *# 最短的路径如果小于期望* expect\_dis = dis\_best  
 expect\_path = path\_best *# 更新两个渴望（可以冲破渴望水平）* **if** path\_best **in** taboo\_table:  
 taboo\_table.remove(path\_best)  
 taboo\_table.append(path\_best)  
 **else**:  
 taboo\_table.append(path\_best) *# 更新禁忌表* **else**: *# 如果最短路径的还是不能改善历史最优值  
 # 以次优解代替最优解，for循环观察禁忌表中的每个元素，保证次优解不在禁忌表中* **for** i **in** range(table\_length):  
 **if** path\_best **in** taboo\_table: *# 如果在禁忌表里，则移除* dis\_new.remove(dis\_best)  
 path\_new.remove(path\_best)  
 dis\_best = min(dis\_new) *# 求新的次优解* path\_best = path\_new[dis\_new.index(dis\_best)] *# 对应新次优解的最短路径方案* taboo\_table.append(path\_best) *# 不在禁忌表中则选中  
 # 超过禁忌表规定长度则把先进入禁忌表的数据踢出* **if** len(taboo\_table) > table\_length:  
 **del** taboo\_table[0]  
 *#输出禁忌表迭代过程* **for** list **in** taboo\_table:  
 tab = [x + 1 **for** x **in** list]  
 print(tab)  
 print(**' '**)  
  
print(**'初始解：'**, firstValue)  
print(**'最短距离'**, expect\_dis)  
result = [x + 1 **for** x **in** expect\_path]  
print(**'最短路径：'**, result)

|  |  |
| --- | --- |
| 一些测试结果： |  |
| 初始解： [1, 2, 3, 4, 5]  最短距离 43  最短路径： [4, 2, 3, 1, 5]  禁忌表迭代过程：  [1, 2, 3, 4, 5]  [4, 2, 3, 1, 5]  [1, 2, 3, 4, 5]    [5, 2, 3, 1, 4]  [4, 2, 3, 1, 5]  [1, 2, 3, 4, 5]    [5, 1, 3, 2, 4]  [5, 2, 3, 1, 4]  [4, 2, 3, 1, 5]    [4, 1, 3, 2, 5]  [5, 1, 3, 2, 4]  [5, 2, 3, 1, 4]    [4, 2, 3, 1, 5]  [4, 1, 3, 2, 5]  [5, 1, 3, 2, 4] | 初始解： [3, 2, 5, 4, 1]  最短距离 43  最短路径： [3, 2, 5, 4, 1]  禁忌表迭代过程：  [3, 2, 5, 4, 1]  [3, 1, 5, 4, 2]  [3, 2, 5, 4, 1]    [3, 1, 4, 5, 2]  [3, 1, 5, 4, 2]  [3, 2, 5, 4, 1]    [3, 2, 4, 5, 1]  [3, 1, 4, 5, 2]  [3, 1, 5, 4, 2]    [3, 2, 5, 4, 1]  [3, 2, 4, 5, 1]  [3, 1, 4, 5, 2]    [3, 1, 5, 4, 2]  [3, 2, 5, 4, 1]  [3, 2, 4, 5, 1] |