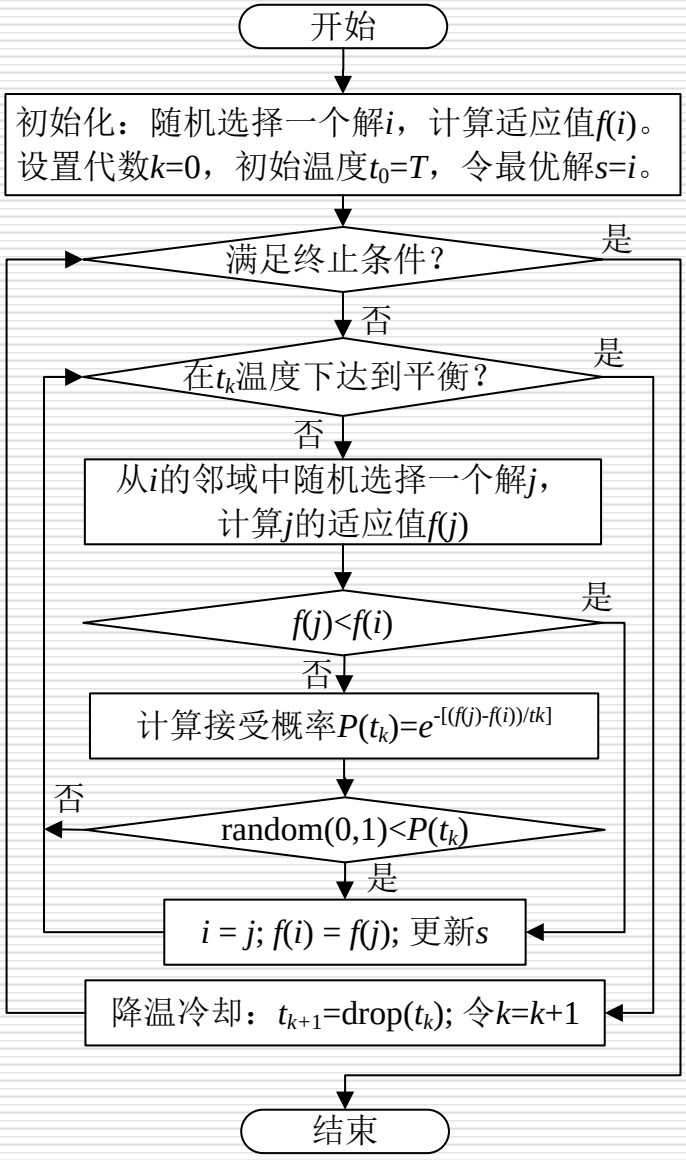
Artificial Intelligence
人工智能实验

高级搜索

中山大学计算机学院
2024年春季

目录

1.1 模拟退火算法



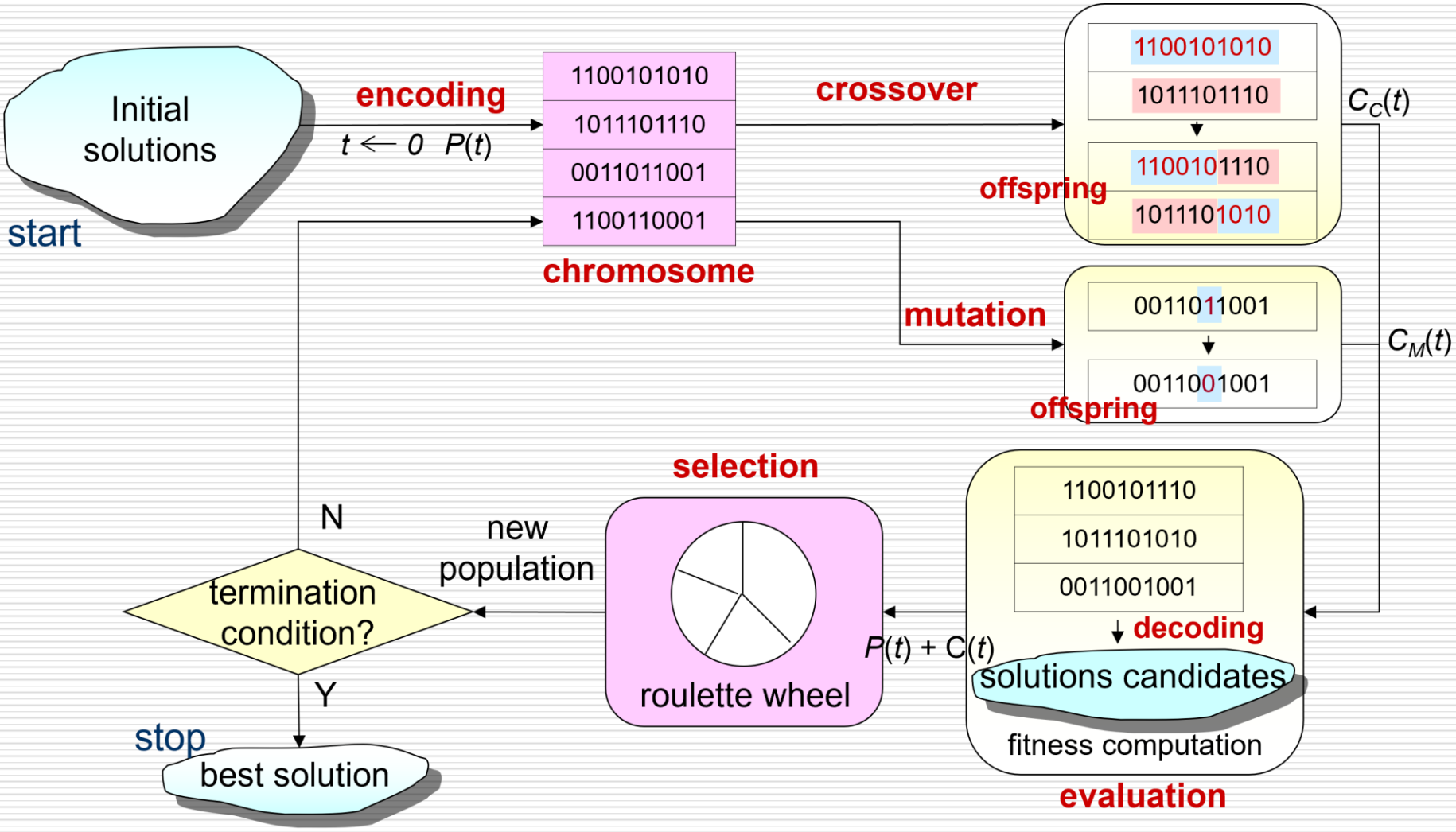

```
//功能：模拟退火算法伪代码
//说明：本例以求问题最小值为目标
//参数：T为初始温度; L为内层循环次数

procedure SA
  //Initialization
    Randomly generate a solution X0, and calculate its
    fitness value f(X0);
  Xbest=X0; k=0; tk=T;
  while not stop
    //The search loop under the temperature tk
    for i=1 to L  //The loop times
      Generate a new solution Xnew based on the current
      solution Xk, and calculate its fitness value f(Xnew).
      if f(Xnew) < f(Xk)
          Xk = Xnew;
          if f(Xk) < f(Xbest) Xbest=Xk;
          continues;
      end if
      Calculate P(tk)=e^-[(f(Xnew)-f(Xk))/tk];
       if random(0,1) < P
          Xk = Xnew;
      end if
    end for
    //Drop down the temperature
    tk+1=drop(tk); k=k+1;
  end while
  print Xbest
end procedure
```

1.1 模拟退火算法

功能意义	基本要素	设置方法
影响模拟退火算法全局搜索性能的重要因素之一。 实验表明，初温越大，获得高质量解的几率越大，但花费的计算时间将增加。	初始温度	1、均匀抽样一组状态，以各状态目标值的方差定初温 2、随机产生一组状态，以两两状态间最大差值定初温 3、利用经验公式给出初温
状态空间与状态产生函数。 邻域函数（状态产生函数）应尽可能保证产生的候选解遍布全部解空间。	邻域函数	候选解一般采用按照某一概率密度函数对解空间进行随机采样来获得。 概率分布可以是均匀分布、正态分布、指数分布等等
指从一个状态X_k(一个可行解)向另一个状态X_{new}(另一个可行解)的转移概率，通俗的理解是接受一个新解为当前解的概率	接受概率	一般采用Metropolis准则 $P_{ij}^{T}=\begin{cases}1, & \text{if } E(j)\le E(i)\\ e^{-\left(\frac{E(j)-E(i)}{KT}\right)}=e^{-\left(\frac{\Delta E}{KT}\right)}, & \text{otherwise}\end{cases}$
指从某一较高温状态t_0向较低温状态冷却时的降温管理表，或者说降温方式	冷却控制	1、经典模拟退火算法的降温方式 $t_k=\frac{t_0}{\lg(1+k)}$ 2、快速模拟退火算法的降温方式 $t_k=\frac{t_0}{1+k}$
内层平衡也称Metropolis抽样稳定准则，用于决定在各温度下产生候选解的数目	内层平衡	1、检验目标函数的均值是否稳定 2、连续若干步的目标值变化较小 3、预先设定的抽样数目，内循环代数
算法的终止条件	终止条件	1、设置终止温度的阈值 2、设置外循环迭代次数 3、算法搜索到的最优值连续若干步保持不变 4、检验系统熵是否稳定

1.2 遗传算法


Initial solutions
start
encoding
$t \leftarrow 0$ $P(t)$
1100101010
1011101110
0011011001
1100110001
chromosome
crossover
1100101010
1011101110
1100101110
1011101010
offspring
$C_C(t)$
mutation
0011011001
0011001001
offspring
$C_M(t)$
selection
1100101110
1011101010
0011001001
decoding
solutions candidates
fitness computation
evaluation
$P(t) + C(t)$
roulette wheel
new population
termination condition?
N
Y
stop
best solution

1.2 遗传算法

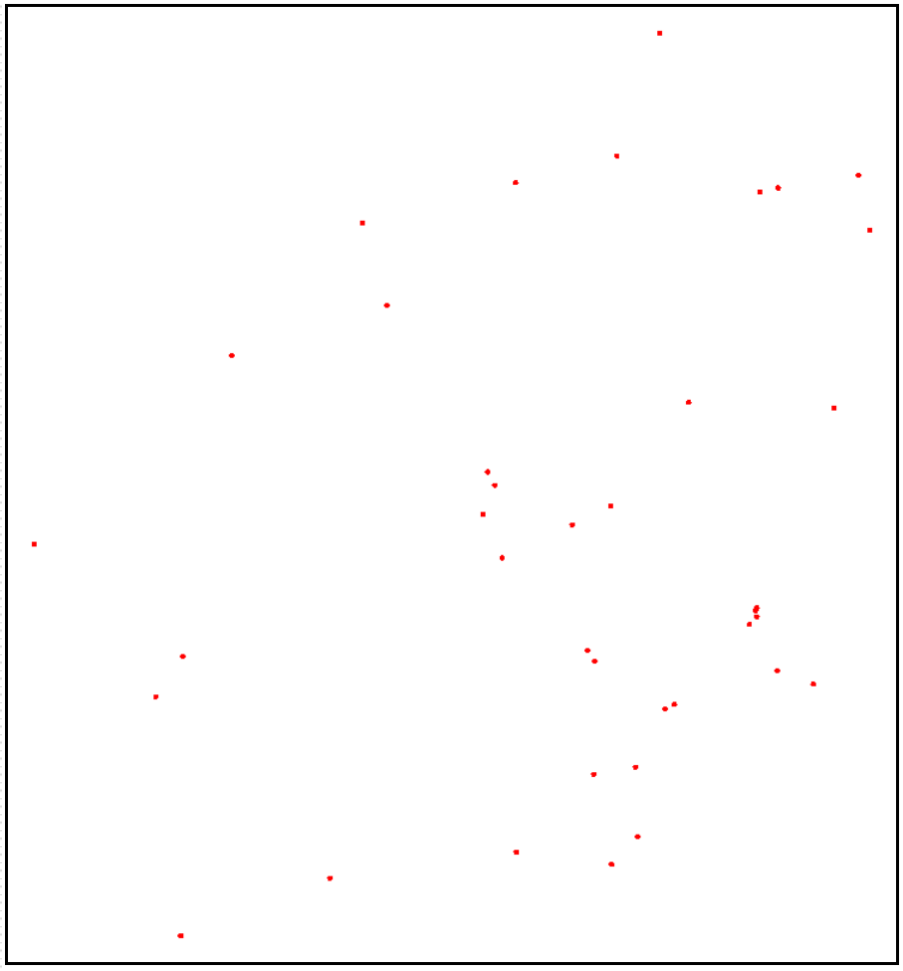
算法 遗传算法解决 TSP

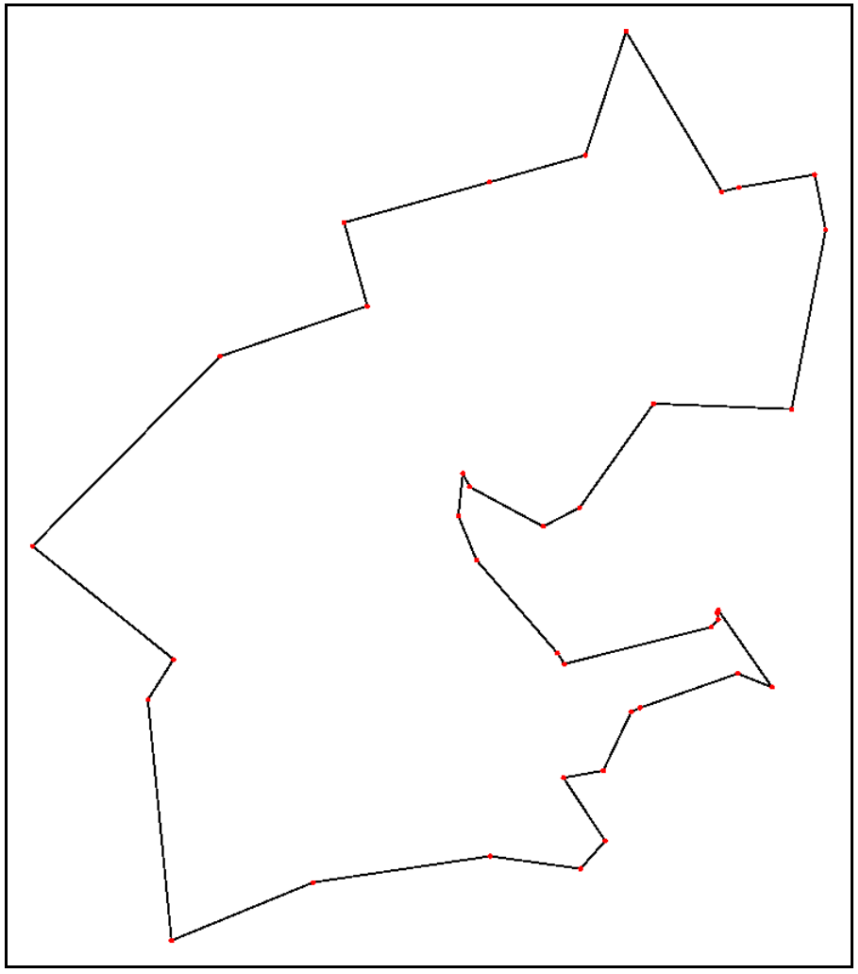
输入: n 个城市的坐标

随机生成 10 个不同个体的初始种群 $P(0)$
for $t = 0$ **to** 最大迭代轮数 **do**
    初始化后代种群 $C(t)$
    for $i = 1$ **to** 每一代后代数目 **do**
        依据适应度, 从 $P(t)$ 中选取一对父母 p_1, p_2
        利用交叉操作, 从 p_1, p_2 产生后代 c_1, c_2
        以一定概率对 c_1, c_2 进行变异操作
        $C(t) \leftarrow \{c_1, c_2\}$
    end for
    依据适应度, 从 $P(t) \cup C(t)$ 选取优异个体作为 $P(t+1)$
end for
依据适应度, 从最后一代种群中选择最优个体作为问题的解

1.3 旅行商问题(TSP)

- 旅行商问题(TSP)是最为广泛研究的组合优化问题之一.
- 问题描述: 一个旅行商寻找环游n个城市(不重复)的最短路径.





1.3.1 解的表示

- **表示方法一：随机排列**
  - 对于n个城市的TSP问题，解表示为1-n的排列

染色体表示

5	4	6	9	2	1	7	8	3

环游顺序: 5-4-6-9-2-1-7-8-3-5

- **表示方法二：随机数排序**
  - 生成n个(0,1)范围内的随机数
  - 这n个随机数的排序对应旅游城市的顺序

染色体表示

1	2	3	4	5	6	7	8	9
0.23	0.82	0.45	0.74	0.87	0.11	0.56	0.69	0.78

环游顺序: 6-1-3-7-8-2-9-4-5-6

1.3.2 染色体交叉操作

部分映射交叉(Partial-Mapped Crossover ,PMX)

算法 染色体交叉操作

输入: 染色体 v_1, v_2 及其长度 l

1: $R \leftarrow \varnothing$

切割染色体

2: 随机生成两个下标 s, t, 满足 $0 < s < t < l - 1$

交叉染色体

3: $v_1' \leftarrow [v_1[0], \cdots, v_1[s-1], v_2[s], \cdots, v_2[t], v_1[t+1], \cdots, v_1[l-1]]$

4: $v_2' \leftarrow [v_2[0], \cdots, v_2[s-1], v_1[s], \cdots, v_1[t], v_2[t+1], \cdots, v_2[l-1]]$

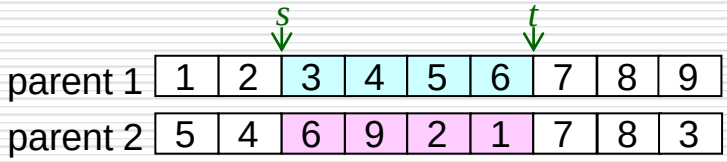
基因映射

5: 依据 $[v_1[s], \cdots, v_1[t]]$ 与 $[v_2[s], \cdots, v_2[t]]$ 建立映射关系

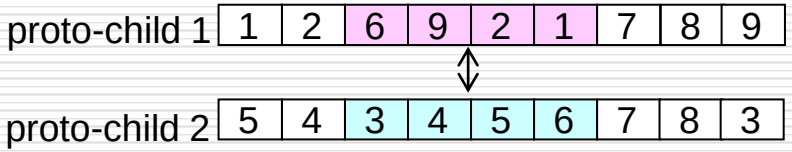
生成后代染色体

6: 依据映射关系修改 v_1', v_2' 非交叉部分

step 1 : 随机选择下标 s,t

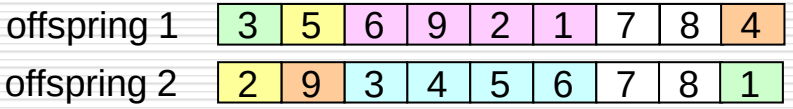


step 2: 交叉子串



step 3 : 确定映射关系

step 4 : 生成后代染色体

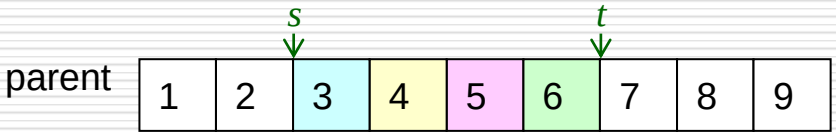


其他交叉操作: order crossover (OX), cycle crossover (CX), position-based crossover, order-based crossover等. 感兴趣的同学可自行尝试.

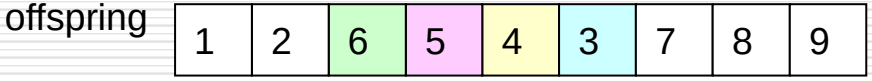
1.3.3 染色体变异操作

倒置变异(Inversion Mutation)

step 1: 随机选择下标 s,t



step 2 : 将s-t中间部分倒置



算法 染色体变异操作

输入: 染色体 v 及其长度 l

1: $R \leftarrow \varnothing$

切割染色体

2: 随机生成两个下标 s, t, 满足 $0 < s < t < l - 1$

倒置

3: $v' \leftarrow [v[0], \cdots, v[s-1], v[t], v[t-1], \cdots, v[s+1], v[s], v[t+1], \cdots, v[l-1]]$

4: **return** v'

2. 实验任务

用遗传算法求解 TSP 问题

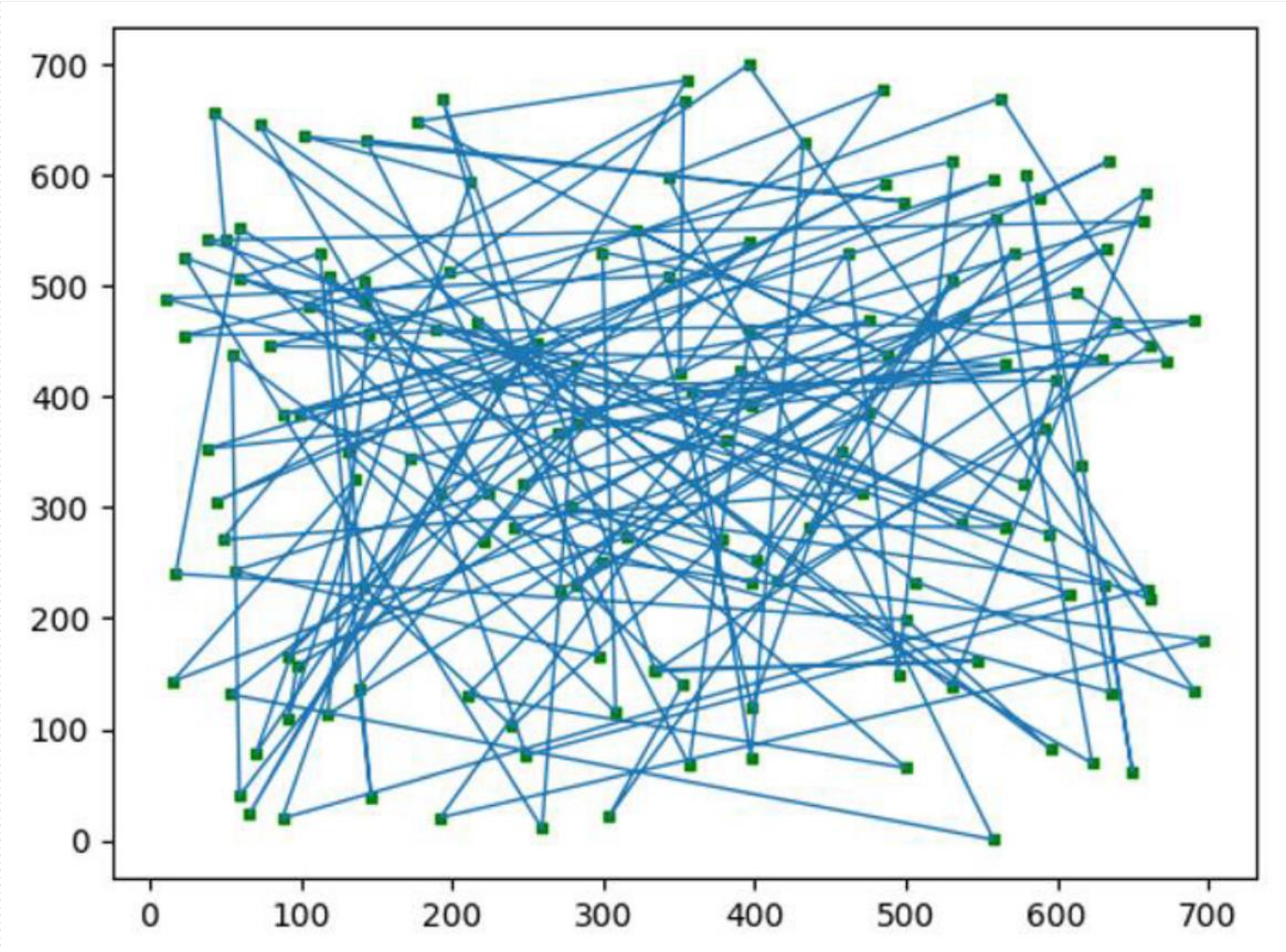
- 在National Traveling Salesman Problems (uwaterloo.ca)（https://www.math.uwaterloo.ca/tsp/world/countries.html）中任选两个TSP问题的数据集。
- 建议：
  - 对于规模较大的TSP问题, 遗传算法可能需要运行几分钟甚至几个小时的时间才能得到一个比较好的结果. 因此建议先用城市数较小的数据集测试算法正确与否, 再用城市数较大的数据集来评估算法性能。
  - 由于遗传算法是基于随机搜索的算法, 只运行一次算法的结果并不能反映算法的性能. 为了更好地分析遗传算法的性能, 应该以不同的初始随机种子或用不同的参数(例如种群数量, 变异概率等)多次运行算法, 这些需要在实验报告中呈现.

3. 作业提交说明

- 压缩包命名为：“学号_姓名_作业编号”，例：20240402_张三_实验6。
- 每次作业文件下包含两部分：code文件夹和实验报告PDF文件。
  - code文件夹：存放实验代码；
  - PDF文件格式参考发的模板。
- 如果需要更新提交的版本，则在后面加_v2，_v3。如第一版是“学号_姓名_作业编号.zip”，第二版是“学号_姓名_作业编号_v2.zip”，依此类推。
- 截至日期：**2024年4月30日晚24点**。
- 提交邮箱：zhangyc8@mail2.sysu.edu.cn。

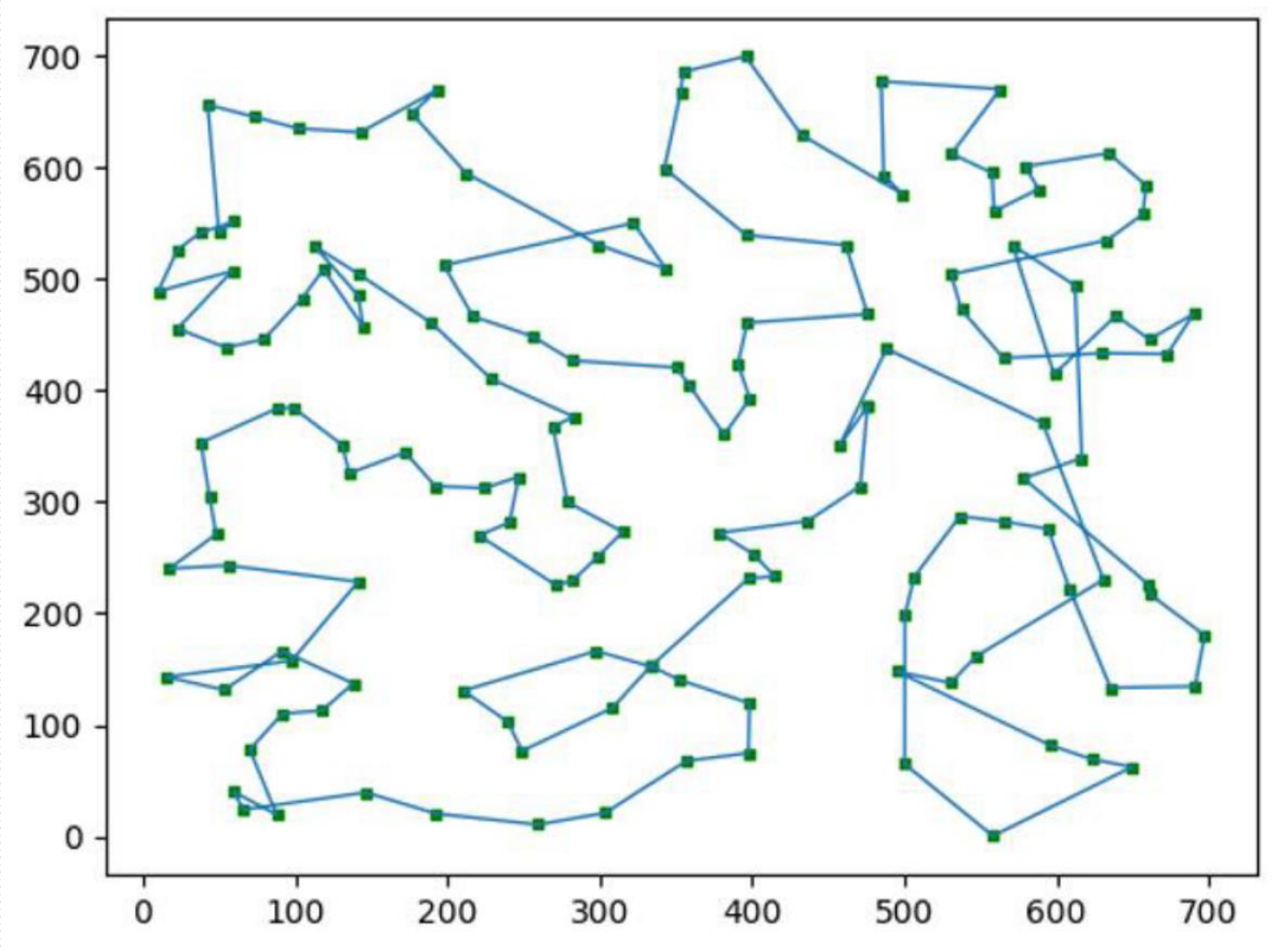
往届学生作品演示

- 路径可视化



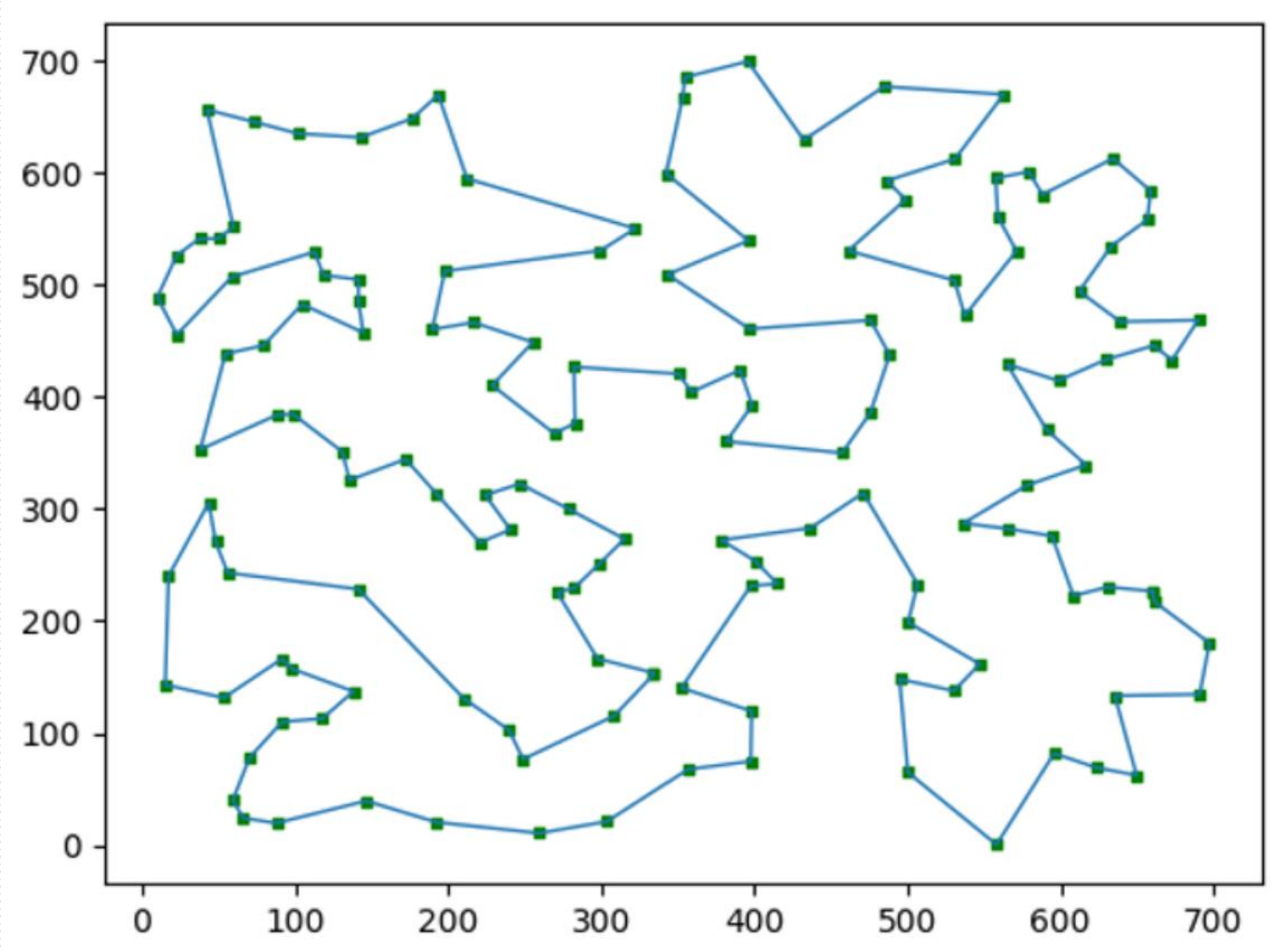
往届学生作品演示

- 路径可视化



往届学生作品演示

- 路径可视化



往届学生作品演示

- 收敛曲线