天津工业大学

本 科 生 毕 业 设 计 （ 论 文 ）

学　　号: 1710820215

姓　　名: 田汝浩

专　　业: 信息与计算科学

学　　院: 数学科学学院

指 导 教 师: 谭建国

职　　称: 副教授

完 成 日 期: 2021.3.5

本人声明

我声明，本论文及其研究工作是由本人在导师指导下独立完成的，在完成论文时所利用的一切资料均已在参考文献中列出。

作者： 田汝浩

签字：

时间： 2020 年 02 月

Hopfield 神经网络在城市物流路径规划的应用研究

摘　要

物流企业对我国物流运输乃至经济的发展有着了不可磨灭的贡献，有了物流企业其他实体行业的商品才会有生产和大量销售的可能，运输互动作为物流的一个组成部分，近年来收到了广泛的研究与讨论，其较为热门的解决方法，遗传算法，模拟退火算法，Hopfield 网络也成为了各个领域研究的焦点。

本论文首先介绍了运输问题的数学模型，然后介绍了用于解决运输问题的算法，如 Hopfield 神经网络，智能算法以及确定性算法，如动态规划，分支限界法。本论文应用的方法是是模拟退火算法的改进型，下面统称为 GASA-Hopfield 算法，首先将模拟退火算法于 Hopfield 神经网络相结合，并且为了改善模拟退火的收敛速度，运行速度，解的质量，本论文重写并改进了“选择”函数部分，并且将变异率由定值改为变量，将程序的退出条件改进为一定代数解不改变即退出程序，改进后的程序收敛速度，运行时间，解的平均质量都有了明显改善。

为了将 GASA-Hopfield 应用于实例中，本论文首先采用爬虫技术动态的爬取指定经纬度地图数据，然后采用 Dijkstra 算法，交点求解算法生成了图的邻接矩阵。最后带入到程序中得到最终结果，并将结果在，最优解，平均解，求解时间，收敛代数四个维度与常见的解决 TSP 问题的算法进行对比，证明了算法的优越性。

关键词：物流运输；遗传算法；模拟退火算法；TSP；K-means；Hopfield 网络；Dijkstra

Research on Application of Hopfield Neural Network in Urban Logistics Path Planning

Abstract

Logistics companies have made an indelible contribution to the development of my country’s logistics and transportation and even the economy. Only with logistics companies’ products in other physical industries will it be possible to produce and sell in large quantities. Transportation interaction, as an integral part of logistics, has received extensive attention in recent years. Research and discussion, its more popular solutions, genetic algorithm, simulated annealing algorithm, and Hopfield network have also become the focus of research in various fields.

This paper first introduces the mathematical model of transportation problems, and then introduces the algorithms used to solve transportation problems, such as Hopfield neural networks, intelligent algorithms, and deterministic algorithms, such as dynamic programming and branch and bound methods. The method used in this paper is an improved type of simulated annealing algorithm, which is collectively referred to as the GASA-Hopfield algorithm below. First, the simulated annealing algorithm is combined with the Hopfield neural network, and in order to improve the convergence speed, running speed, and solution quality of simulated annealing, This paper rewrites and improves the ”selection” function part, and changes the mutation rate from a fixed value to a variable, and improves the exit condition of the program to a certain algebraic solution without changing the program. The improved program convergence speed, running time, The average quality of the solutions has improved significantly.

In order to apply GASA-Hopfield to an example, this paper first uses crawler technology to dynamically crawl the specified latitude and longitude map data, and then uses the Dijkstra algorithm, and the intersection solving algorithm to generate the adjacency matrix of the graph. Finally, bring it into the program to get the final result, and compare the results in the four dimensions of optimal solution, average solution, solution time, and convergence algebra with common algorithms for solving TSP problems,

which proves the superiority of the algorithm.

Key words: logistics and transportation; genetic algorithm; simulated annealing algorithm; TSP; K-means; Hopfield network; Dijkstra

目　　录

第一章　绪论

一、　引言

运输问题，一版被描述为寻找某一种最优的运输方案使得被运输品从始发地运输到多个目的地, 使得运输方利益最大化。我国的物流运输现状，近年来有着规模扩大, 运量增加的趋势. 截止到 2021 年, 我国近五年的社会物流总额年均增长率超 6.5%，物流业总收入超 10 万亿元. 但现阶段我国物流还存在着效率低，兼容性差，建设滞后，机制障碍的问题。所以在现阶段在面对大量的公路，铁路，海运等物流运输要求时，可以在时间和仓储方面消耗最少的资源无疑会更有优势，这也对现有的物流路径规划算法有了更高的要求。

二、　课题的应用背景

近几年来我国物流企业不断崛起，整体呈现出规模巨大，科技水平提升，物流发展格局不断优化的特点，物流企业对我国物流运输乃至经济的发展有着了不可磨灭的贡献，有了物流企业其他实体行业的商品才会有生产和大量销售的可能而商品生产出来的目的就是为了被消费者所消耗，而大学生作为聚集且消费能力强的群体，导致大学校园以及周边的外卖快递的物流运输十分的错综复杂。但本文以天津西青大学城为例，研究物流运输的路径最优问题。面积近 90 平方公里，入驻师生约 40 万人。入驻高校有天津工业大学、天津师范大学、天津理工大学、天津师范大学津沽学院、天津大学软件学院、天津公安警官职业学院、天津农学院、天津城建大学、天津商业大学宝德学院、天津教育招生考试院等高等教育机构、公安部天津消防研究所等科研机构、天津团泊体育中心等体育远动场所发展起来的郊区新城，近年来物流运输请求的逐年增加，如何让商品流转于各个地方最终到达消费者手中且能够在消费者可接受的时间内送达成为了平台和快递员与外卖员常常思考的问题。但在后文的讨论中我们就会发现物流运输在大学校园的应用其本质就是解决 TSP 问题。为了增加代码的复用性，本文所用的代码可以通过经纬度自动搜集任意地区的地图矢量数据，使用本文的方法对路径进行优化。

三、 本文的内容及结构安排

本文一共分为四章。

第一章绪论。主要介绍了本文的研究背景与相关领域的介绍。

第二章相关技术及其理论基础。介绍了运输问题的理论模型以及解决常见运输问题的研究算法。

第三章数据爬取介绍了本文例子如何从网站爬取并且对数据进行转化清洗最终转化为图的数据结构。

第四章西青大学城运用 GASA-Hopfield 算法求解运输问题。实现本文的例子，并且将集成的 Hopfield 路径规划算法与其余算法在时间空间性能稳定性上进行比较。

第二章　本项目的数学模型

一、　车辆路径运输问题模型

Dantzig 和 Ramser 于 1959 年首次提出了车辆路径规划问题 (VRP)，这个问题是指不定量的顾客，每个顾客有不同的需求，一个或多个物流中心向客户提供所需商品，并且由一个或多个配送员分配顾客的货物，并寻找合适的配送路径，在满足客户需求的大前提下，能够完成例如路径最短，时间花费最少等成本消耗最小的目的。

(一)　模型一：带有容量约束的车辆路径问题 (CVRP)

记顾客集合 $M = \{1, 2, \cdots, n\}$

记顶点集合 $V = M \cup \{0, n+1\}$

记车辆数量 D

记车辆总容量 K

记顾客 j 的需求 R_j

记路径成本 x_{ij}

目标函数如下：

$$min \quad f = \sum_{i=0}^{n+1} \sum_{j=0}^{n+1} x_{ij} \omega_{ij} \tag{2.1}$$

$$
st.\begin{cases}
\sum_{\substack{j=1\\ j\neq i}}^{n+1}\omega_{ij}=1 \quad \forall i\in M & (2.2)\\
\sum_{\substack{i=0\\ k\neq i}}^{n+1}\omega_{ik}=\sum_{\substack{j=1\\ k\neq i}}^{n}+1\omega_{kj} \quad \forall k\in M & (2.3)\\
\sum_{j=1}^{n}\omega_{0j}\Subset D & (2.4)\\
y_i+\omega_{ij}q_j-k(1-\omega_{ij})\leq y_i \quad \forall i,j\in V & (2.5)\\
q_j\leq y_i\leq Q \quad \forall i\in N & (2.6)\\
x_{ij}\in\{0,1\} \quad \forall i,j\in N & (2.7)
\end{cases}
$$

上式中 (2.1) 是目标函数，代表合法路径的成本最小化，(2.2) 代表每个点除起点外都要出一次，(2.3) 代表流量约束，指每个点出一次必须进一次，(2.4) 代表起点出弧数小于车辆数量，而 (2.5) 代表合法路径不存在回路，(2.6) 表示需求不超过车的容量。(2.7) 表示 ω_{ij} 满足 0，1 约束。

对于本文例子的应用，经过调查走访发现大学城南门各饭店小吃街的各个商户的外卖商品样式质量相似，所以各个顾客的需求可近似为 1，其店铺规模普遍偏小，每家商户通常配备 1 名配送员，一名配送员通常服务 1 家门店所以 D = 1，K = 1, 故模型可以进一步简化为模型二

(二) 模型二：TSP 问题模型

记赋权图 G=(V,E)，V 为顶点集 $V=\{1,2,\cdots,n\}$，E 为边集，各顶点间的距离 d_{ij} 已知 n 为地点数量。车辆路径运输模型如下：

$$
min \quad f=\sum_{i=1}^{n}\sum_{j=1}^{n}\omega_{ij}x_{ij} \tag{2.8}
$$

$$
st.\begin{cases}
\sum\limits_{\substack{i=1\\1\le j\le n}}^{n}\omega_{ij}=1 \quad \forall i\in V & (2.9)\\
\sum\limits_{\substack{j=1\\1\le i\le n}}^{n}\omega_{ij}=1 \quad \forall j\in V & (2.10)\\
\sum\limits_{i=1}^{n}\omega_{ij}=\sum\limits_{k=1}^{n}\omega_{jk} \quad \forall j\in V & (2.11)\\
\sum\limits_{i\in S}\sum\limits_{j\in\bar{S}}\omega_{ij}\ge 1 \quad \forall S\Subset V, S\neq\varnothing & (2.12)\\
\omega_{ij}\in\{0,1\} \quad \forall i,j\in V & (2.13)\\
\omega_{ii}\neq 1 \quad \forall i\in V & (2.14)
\end{cases}
$$

上式中 ω_{ij} 代表最终路径中是否存在从 i 到 j 的通路，x_{ij} 代表从 i 到 j 所需代价。(2.8) 是目标函数，即通过所有节点后代价最小。目标函数下面为各个约束条件，其中 (2.9) 说明任意节点的进度为 1, (2.10) 说明任意节点的出度为 1，(2.9) 和 (2.10) 规定合法路径中不存在分叉，(2.11) 是流量平衡约束，即合法路径流量从一个节点到另外一个节点不会存在断流情况。(2.12) 是消除子回路约束，其中 n 个地点被任意分为 S 与 $\bar{S}$，合法路径中应满足以存在于 S 中的 i 点为起点，以存在于 $\bar{S}$ 中的 j 点为终点的情况。(2.13) 说明 ω_{ij} 满足 0，1 约束。(2.14) 说明路径任意节点并不存在一条自己到自己的通路。

二、 运输问题算法研究概况

目前对于路径规划问题的算法分为四部分

(1) 精确式算法:

其精确式算法主要分为三部分即动态规划法，线性规划法，与分支定界法，但此三种方法的运行时间会随着问题的规模指数上升，对于大型问题普遍求解效率偏低，经过近几年算法的发展，学者们对其进行了改进，王晓琨利用混合启发式快速算法与整数规划解决了电车充电需求问题，张鹏乐提出了一种求解大规模 VCVRP 问题的快速动态规划模型即 DPM-MST 模型，并且改进后的算法求解质量受节点规模波动较小。

(2) 启发式算法:

启发式算法相对于最优化算法而被定义，通常是一个基于直观或经验构造

的算法，它可以在规定时间内给出一个问题的可行解，但求得的解与最优解的偏差无法被估计，可行解的区间也无法被确切估计。其在 VRP 的应用主要由三类算法构成，即插入检测法，节约法，与改进节约法。启发式算法能够解决大规模 VRP 问题，尽管往往求出的并非是最优解。此类算法的应用研究在十余年前较为流行，近几年研究文献逐步下降，例如孔媛设计了新的评价因子即最小评价因子并且结合了 VSPA(接送顾客到机场的车辆调度问题) 问题的特点，基于顺序插入的方法构造路径。李兵采用了重新优化法方法即将车辆的出发点看作任务点使得问题可以看作可以使用静态问题算法求解的普通车辆路径问题。但此阶段的启发式算法对于问题最优解的寻找作用较为薄弱。

(3) 元启发式算法:

元启发式算法是启发式算法的改进，它是随机算法与局部搜索算法相结合的产物，相对于启发式算法，它求得的可行解质量有了较大的提升，它由遗传算法，蚁群算法，模拟退火算法，粒子群算法，禁忌搜索算法与混合算法等六类算法组成。元启发式算法也是近几年研究成果最为丰富的一类算法。穆东等基于并行模型退火算法求解时间依赖型车辆路径问题，罗勇等针对遗传算法的选择交叉变异步骤，提出了基于序的选择算子、基于最小代价树的交叉算子和基于随机点长度控制的变异算子。其改进后的遗传算法收敛速度与全局搜索能力有了较大的提升。

(4) 神经网络:

除了以上常用的几种元启发式算法以外, 还有一些文献也提到了神经网络算法, 学者们将人工智能的方法应用到 VRP 问题中, 取得了较好的效果. 路径规划更加合理, 速度也更快. 随着环境的动态性和信息的未知性, 通过神经网络学习的方式, 对未知环境进行训练式探索, 是一种较好的求解方法。而本文的研究方法主要为神经网络，下面对该类方法进行较为详细的介绍。

第三章　项目所用数据的搜集与处理

本章的内容是为了构造程序所需要的数据结构，实现根据参数来爬取指定经纬度的 osm 地图数据，然后转化为 shp 地图数据，读取 shp 地图数据来构造图的数据结构—邻接矩阵。

一、　数据爬取

(一)　爬取 osm 数据

OPen Street Map 简称 OSM，是一个由网络大众共同打造的免费开源服务，在上面你可以下载指定经纬度的地图信息，该组织由史蒂夫·克斯特与 2004 年 7 月创立，2006 年 4 月 OSM 基金会成立，他鼓励社区的用户自由的发布地理数据，并与社区所有人分享并使用地理数据，为了搜集本项目的数据，我们在 osm 中采用爬虫技术爬取 osm 文件。下面是分析过程：

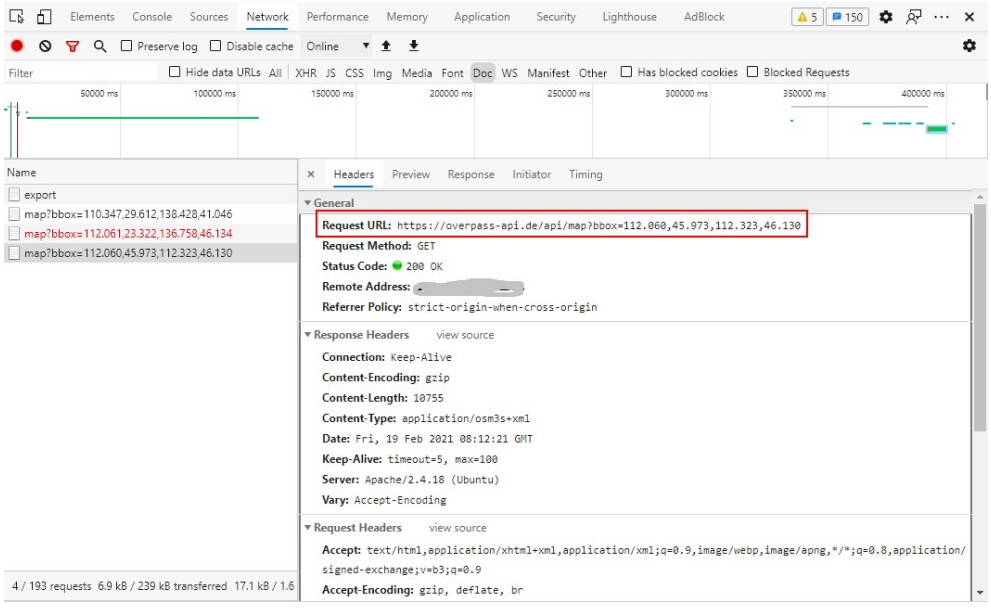

图 3-1　调试信息 General 项

我们从上图可以看出下载的 osm 文件的请求地址，其中请求类型为 GET 表明该数据的请求信息加载在网址中，且成功返回数据，状态码为 200，返回的数

据大小为 10755。然后我们查看该请求的请求头信息：

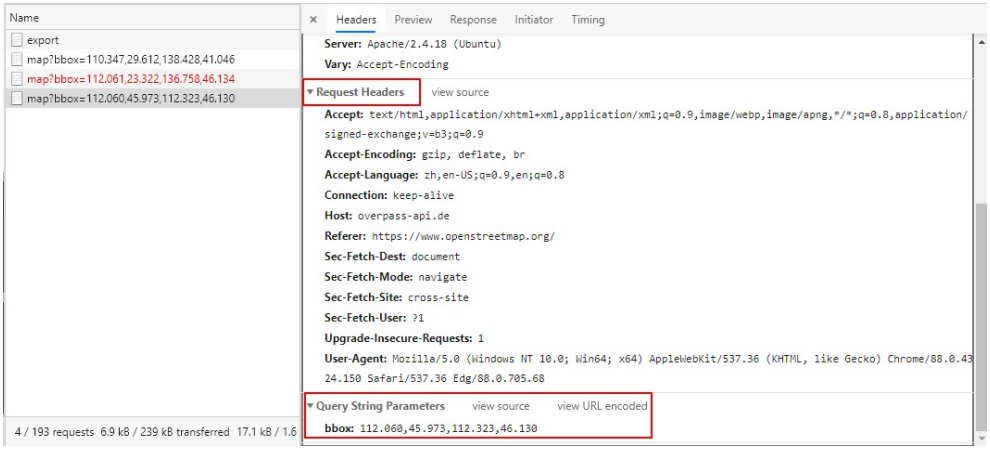

图 3-2　调试信息请求头

根据请求头我们发现该网站在请求头中并不存在验证本机的信息，说明该网址并不存在反爬虫，所需无需手动构造请求头，让程序请求时自动生成即可，所以程序为：

```
import requests
import json
import sys
import  zipfile
for  arg  in  sys.argv:
    bbox = arg # 经纬度信息作为参数传递
osm_path = '/TSP_hopfield/shp_distance/map'
zip_file  = '/TSP_hopfield/shp_distance/map.zip'
def  download_osm():
    url  = 'https://overpass-api.de/api/map?bbox='+ bbox #
        115.3141,38.8425,115.4748,38.9303
    myfile  = session.get(url  ,  allow_redirects=True)
    open ( osm_path, 'wb').write(myfile.content)
session  = requests.session()
# 下载 osm 文件
```

```
15 download_osm()
```

代码 **3.1**　爬取 **osm** 数据代码

为了使得其余程序能够直接调用此文件来下载特定区域内的 osm 数据，我们的程序会读取命令行列表，来动态下载地图数据，下面为了将地图数据转变为计算机好读取的格式我们将其转化为 shp 类型地图文件。

(二)　转化成 shp 数据

首先打开”https://geoconverter.hsr.ch/” 网站，如下所示：

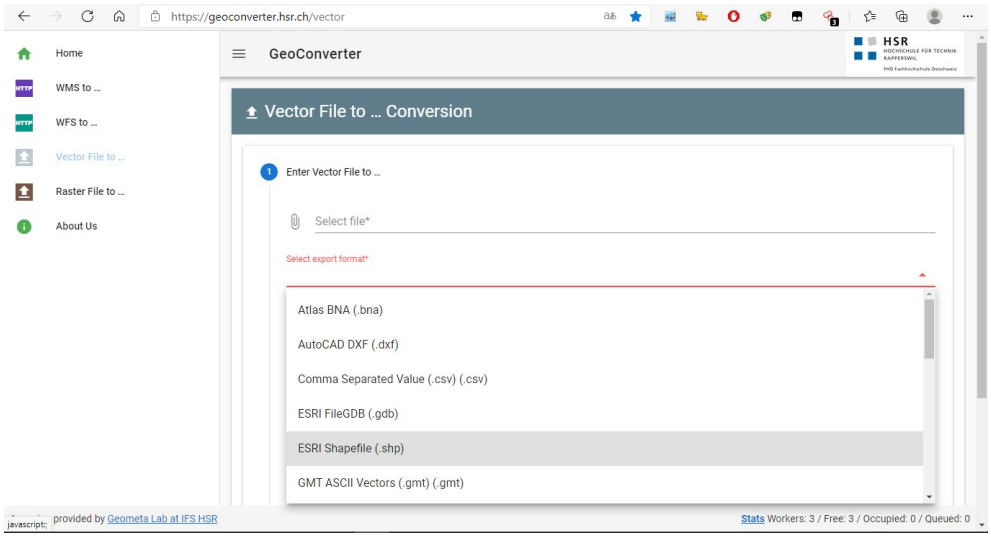

图 3-3　进入 geoconverter 网站

我们上传 osm 文件，并调整转化格式。然后点击转换，即可下载转换后的文件，但是为了能够自动化的进行文件转换，我们采用爬虫技术，来实现文件的上传，转换，下载的步骤，为了模拟该流程，我们首先进行一次数据的提交转换，来观察其过程。

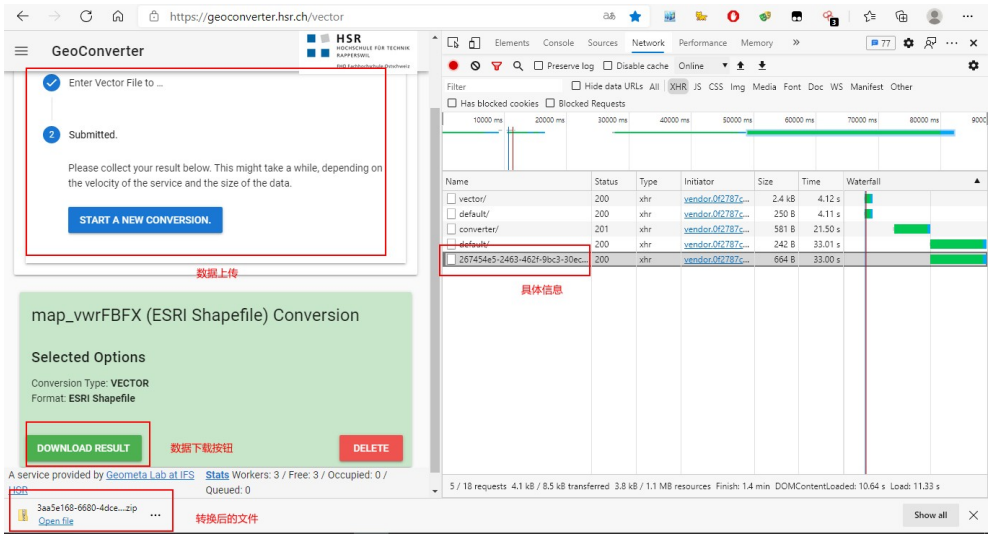

图 3-4　模拟提交转换文件

其中转化后的 shp 的文件在下载的 zip 压缩文件内，他被分为了 3 个部分分别为，路网，水网，建筑网三个文件，我们程序只需提取路网文件即可。再上图中我们在调试 XHR 栏目中可以看到许多请求过程文件，但是值得注意的只有 converter/请求与由长数字字母构成的 267454e5-2463-462f-9bc3-30ecbe41956d/请求，并且查看文件发送时间后发现，converter/请求发送时间较早，而 267454e5-2463-462f-9bc3-30ecbe41956d/请求发送较晚，并且在查看请求内容后我们发现，converter/请求时发送文件的并返回 267454e5-2463-462f-9bc3-30ecbe41956d/请求名称即：

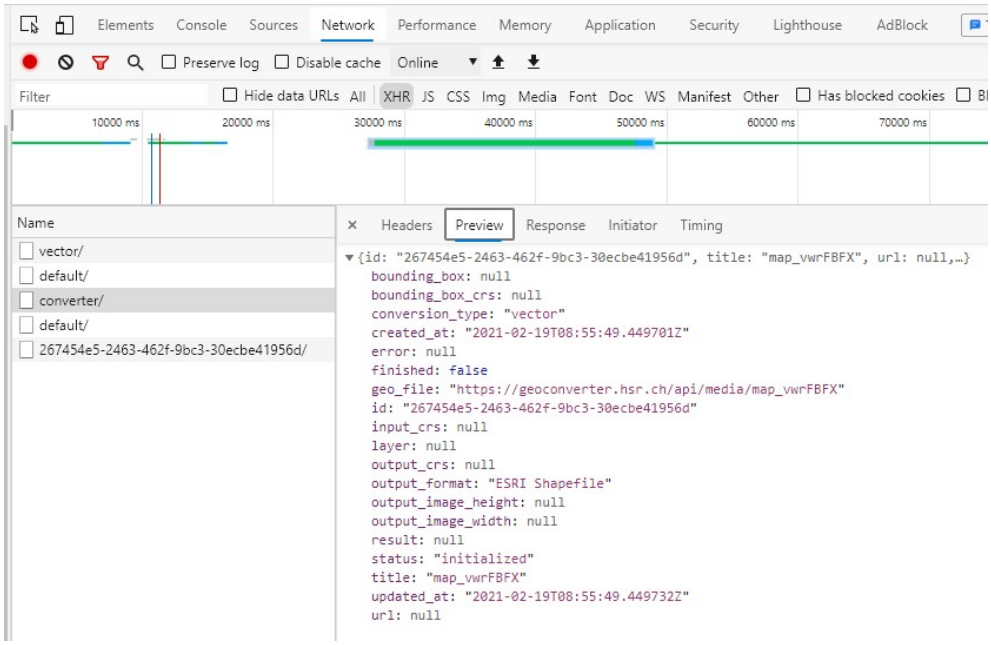

图 3-5　请求文件信息

所以我们的程序会对 multipart/form-data 请求进行模拟，通过 post 请求将文件发送到服务器。并保存服务器响应数据，读取响应数据的 id 值，并以此构造第二次请求的 url，这就是图中的 267454e5-2463-462f-9bc3-30ecbe41956d/请求，请求详细信息如下：

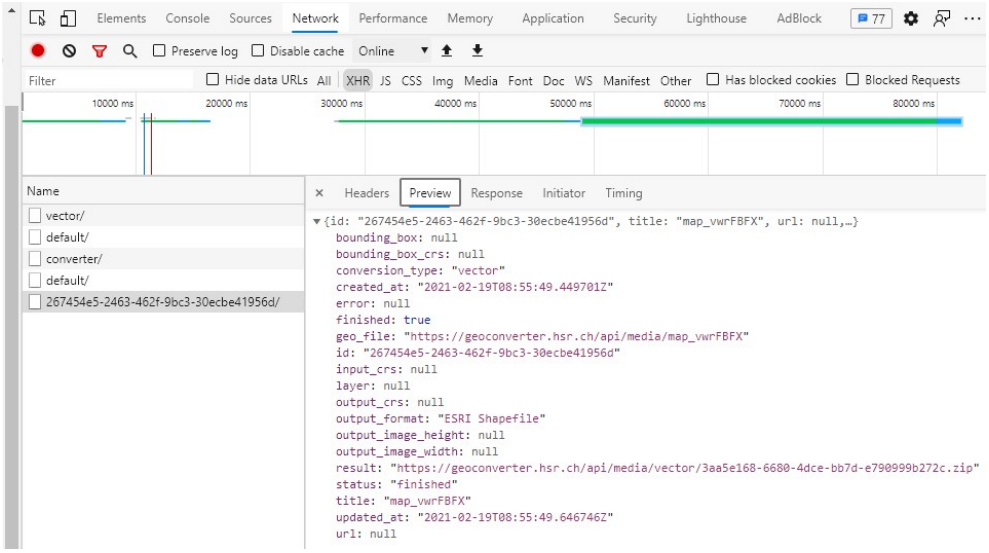

图 3-6　第二次请求信息

从以上的信息中提取到转化后的文件下载地址。在此次模拟提交中是：”https:

//geoconverter.hsr.ch/api/media/vector/3aa5e168-6680- 4dce-bb7d-e790999b272c.zip”，转换结束，然后我们以此来编写代码来代替人工进行自动化请求转化下载步骤，然后解压文件提取 shp 文件，代码如下：

```
    second_headers = {
    'authority': 'geoconverter.hsr.ch',
    'method': 'GET',
    'path': '',
    'scheme': 'https',
    'accept': 'application/json, text/plain, */*',
    'accept-encoding': 'gzip, deflate, br',
    'accept-language': 'zh,en-US;q=0.9,en;q=0.8',
    'referer': 'https://geoconverter.hsr.ch/vector',
    'sec-fetch-dest': 'empty',
    'sec-fetch-mode': 'cors',
    'sec-fetch-site': 'same-origin',
    'user-agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64)
        AppleWebKit/537.36 (KHTML, like Gecko) Chrome
        /87.0.4280.88 Safari/537.36 Edg/87.0.664.66',
}
# 将 osm 文件转化为 shp 文件

# 构造 multipart/form-data 用来上传 osm 文件
files = {'geo_file': ('file', open(osm_path, 'rb'),'application
    /octet-stream'),
        'conversion_type':( None, 'vector'),
        'output_format':( None, 'ESRI Shapefile')
        }
# 不断的请求上传直到响应状态码正确
while True:
```

```
    res = session.post(url, files=files)
    if res.status_code // 100 == 2:
        break
# 获取响应 id
id = res.text.split('"')[3]
# 构造请求头, 准备进行第二次请求, 下载转换的 zip 文件
second_url = 'https://geoconverter.hsr.ch/api/converter/' + id + '/'
path = '/api/converter/'+ id +'/'
second_headers['path'] = path
# 不断的进行第二次请求直到返回正确的 zip 文件下载地址
while True:
    res = session.get(second_url, headers = second_headers)
    if res.status_code // 100 == 2 and json.loads(res.text)['result'] != None:
        break
# 将 zip 下载地址从 text->json 中提取出来
res_dict = json.loads(res.text)
shp_download_url = res_dict['result']
# 压缩文件名称
yasuo_file = shp_download_url.split("/")[-1].split('.')[0]
# 下载 shp 的压缩文件并保存
myfile = session.get(shp_download_url , allow_redirects=True)
open( zip_file, 'wb').write(myfile.content)
with zipfile.ZipFile(zip_file) as zf:
    zf.extractall()
shp_path = '/TSP_hopfield/shp_distance/' + yasuo_file + '/converted.shp'
print(shp_path)
```

代码 **3.2** 转化为 **shp** 数据

二、 图数据结构的构建

(一) 读取 line 数据

在 matlab 中读取 shp 文件数据，我们可以得到 $n \times 1$ *struct* 的数据其中 *n* 表示线的组数，其 line 数据包含以下数据： 下面我们根据以上信息初步生成邻接

表 **3-1** **line** 字段

字段	含义
Geometry	拓扑结构
BoundingBox	边界框
X	线段各点的 x 坐标
Y	线段各点的 y 坐标
osm_id	路段 id
name	路段名称
highway	道路类别
waterway	水路类别
aerialway	空中路段类别
barrier	障碍物
man_made	人造物
z_order	高度
other_tags	其他标签

矩阵，其基本思想为遍历 *line* 数据结构的每个元素，将每个元素的 X,Y 项分别输入一个存储已知点集的数组中，我们称其为 x_{pos}, y_{pos}。然后对 *line* 中的每个元素 X,Y 相邻的坐标点进行距离计算并存储到邻接矩阵 distance 中。由于各个路段之间村子啊交点，但 line 数据结构并不会主动将这些交点标注出来，所以我们准备用快速排斥与跨立算法对线集中的交点进行求解。

(二) 求交点

在判断平米拿上两条线段是否相交时，我们采用快速排斥与跨立实验的算法，其基本思想为，首先通过快速排斥实验来对不满足条件的情况进行排除，然

后对剩余情况用跨立实验进行具体判断并求解。

快速排斥

假设二维平面上存在两个线段，$P_1, P_2 Q_1, Q_2$，以 P_1, P_2 组成的线段为对角线做矩形 R, 以 Q_1, Q_2 为对角线做矩形 T，当两个矩形不相交时两个线段肯定不相交，但两个线段不相交时，两个矩形可能会相交。

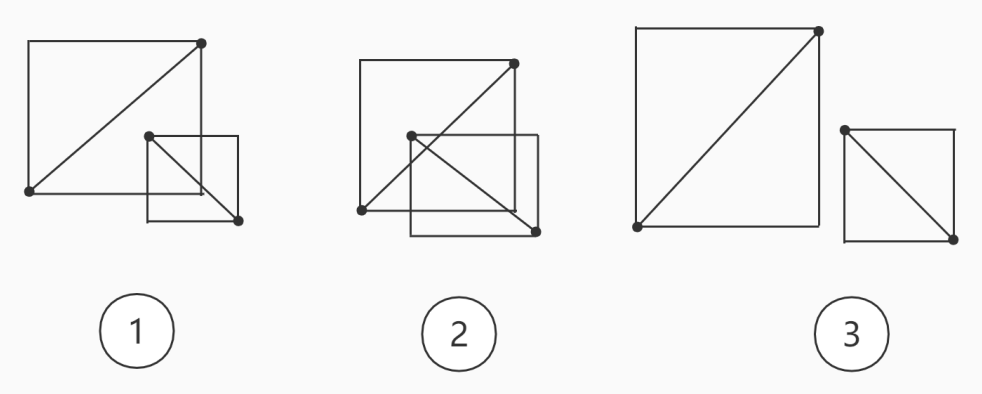

图 3-7　快速排斥与跨立实验

其中第一种情况通过了快速排斥实验但并未通过跨立实验，第二种情况通过了快速排斥实验，并且也通过了跨立实验，第三种情况并未通过快速排斥实验，直接可以得出结论，不存在交点。
在所以在快速排斥中的矩阵相交条件为：

$$min(p_1.x, p_2.x) \le max(q_1.x, q_2.x)\&\& \tag{3.1}$$

$$min(q_1.x, q_2.x) \le max(p_1.x, p_2.x)\&\& \tag{3.2}$$

$$min(p_1.y, p_2.y) \le max(q_1.y, q_2.y)\&\& \tag{3.3}$$

$$min(q_1.y, q_2.y) \le max(p_1.y, p_2.y); \tag{3.4}$$

如果通过了快速排斥实验，也不能直接得出结论二者有交点，而是在通过跨立实验得到结果。

跨立实验

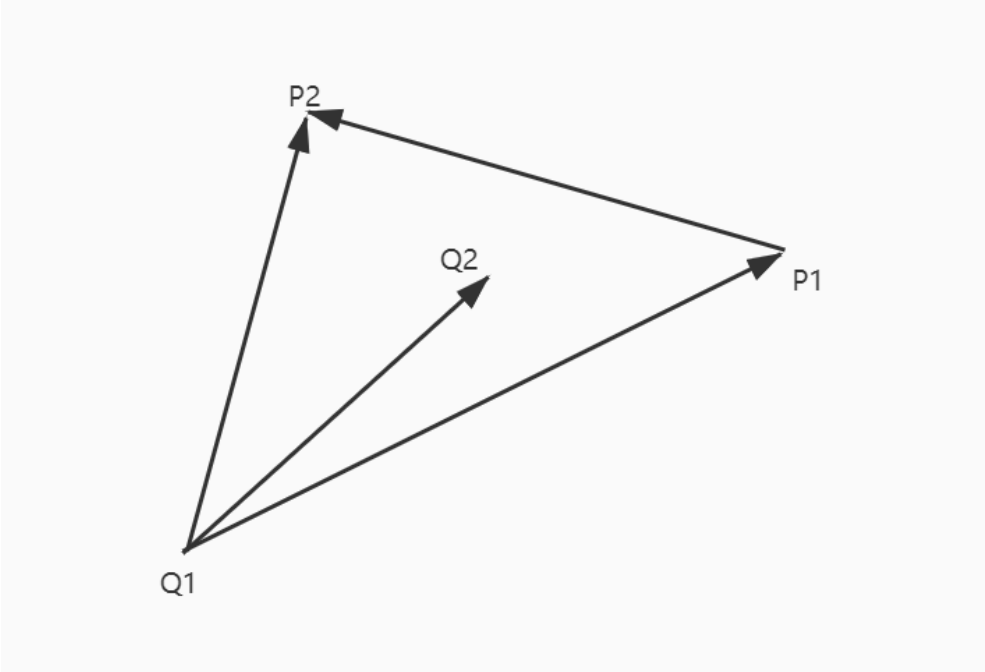

图 3-8　跨立实验

若线段 P_1P_2 跨过了线段 Q_1Q_2，那么 P_1P_2 就在 Q_1Q_2 的两侧，所以将：

$$((P_1 - Q_1) \times (Q_2 - Q_1)) * ((Q_2 - Q_1) \times (P_2 - Q_1)) > 0 \tag{3.5}$$

记为条件 1，同理 Q_1Q_2 分布在 P_1P_2 两侧的条件为：

$$((Q_1 - P_1) \times (P_2 - P_1)) * ((P_2 - P_1) \times (Q_2 - P_1)) > 0 \tag{3.6}$$

记为条件 2，满足以上两个条件两线段才会相交。运行程序我的得到运行前后邻接矩阵的点数对比：

表 **3-2**　运行前后点数对比

点数对比	点数
求交点前	3104
求交点后	4145

可以看出运行前后，在该图中共找到 1041 个交点，改善了图的连通性，使得路径求解称为可能，下面我们给出求解邻接矩阵算法的完整算法流程图，其中我们设 shp 文件读取后的数据为 *Map*，*Map* 的每个元素记为 *line*，*line* 中的元素构成如（3-1）所示。

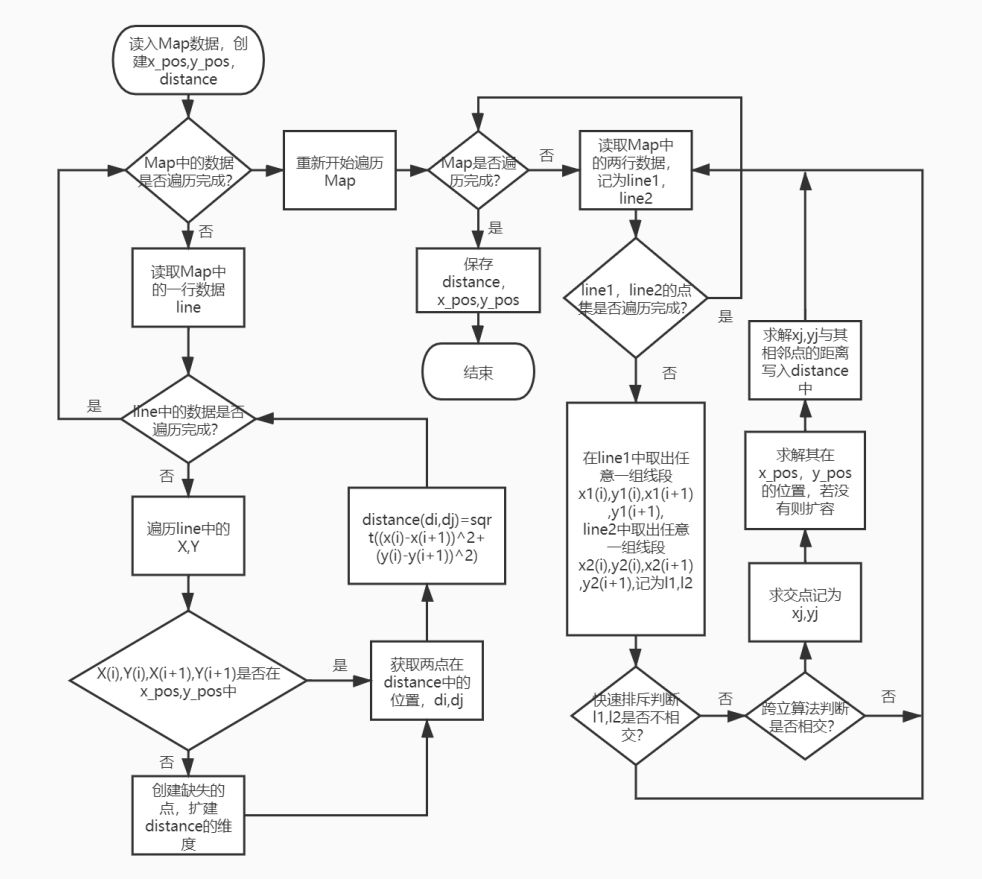

图 3-9　邻接矩阵构造算法

以上为数据的所有准备工作，下一章的项目实现将以此数据为依据对路径进行规划。

第四章　西青大学城运用 GASA-Hopfield 算法求解运输问题

一、　集成的必要性与可行性

我们前面介绍到 Hopfield 神经网络可以很好的解决 TSP 问题，但是他在解决优化问题时存在着容易求解得到非法解，或者容易收敛到局部最优解，前者随着 TSP 问题城市数量的增多，非法解出现的频率会逐渐增大。后者会导致最优解的质量得不到保证。本项目在算法上属于王银年在 2009 年提出的 3PM 交叉算子的模拟退火算法，但由于求解时间较长，收敛较慢，所以对算法的选择操作进行了重写，并且改变了算法参数，与结束条件。

二、　集成的方法

(一)　单源最短路

在本例中由于每个地点之间并不直接通达，而是中间经过其他地点后相通，所以为了求解任意两个城市之间的路径我们采用了单源最短路算法-Dijkstra 算法：

该算法设置一个集合 P, 来记录在其求解的图中的最短路顶点，在算法开始时，会将原点放入该集合中，然后算法依次遍历并且将新遍历的点并入集合中, 然后更新最短路径，与其路径权值。在算法运行过程中会创建两个中间变量，*dist*[] 数组与 *path*[] 数组。其中 *dist* 数组记录原点到图中各个顶点的路径长度，*path* 表示从原点到某一顶点最短路径的前驱节点的索引值。算法结束后也以此变量来反推最短路径。下面算法中 V 表示图中的点集，下面给出其算法步骤：

$Step\ \ 1:$　数据初始化，P 集合将原点包含进去，$dist[i] = distance[0][i],\quad i = \{123\cdots n-1\}$

$Step\ \ 2:$　在 $V-P$ 中找到某点 v_k，该点满足 $dist[j] = Min\{dist[i]|v_i \in V-P\}$，令 $S = S \cup k$。

$Step\ \ 3:$　求改从 v_0 出发到集合 $V-P$ 中任意一点 v_k 的距离，其更新公式为：

$$if \quad dist[j] + distance[j][k] < dist[k]:$$
$$dist[k] = dist[j] + distance[j][k] \tag{4.1}$$

$Step$ 4： 重复 $Step$ 2 ~ $Step$ 3 直到图中所有顶点都被包含到了 P 集合中。

(二) 主程序

本例的算法属于王银年在 2009 年提出的 3PM 交叉算子的模拟退火算法的变种，称为基于遗传模拟退火策略的 Hopfield 神经网络的优化算法，对适应度函数，遗传子程序到 Hopfield 程序的过渡部分，交叉，变异方式进行了添加与改动。由于本算法结合了多种算法所以为了方便叙述，我们将本算法简称为 GASA-Hopfield 算法, 下面给出算法的设计步骤：

$Step$ 1： 输入初始数据，如邻接矩阵，路径矩阵，数据初始化，如种群，网络参数，迭代步长，交叉，变异概率，初始温度，降温幅度，种植温度，退火迭代链长。

$Step$ 2： 进入遗传算法子程序，见（三），传入种群，邻接矩阵，网络参数，编译交叉概率。

$Step$ 3： 将子程序搜索到的最优路径翻译为置换矩阵，并作为 Hopfield 网络的输入。

$Step$ 4： 对神经网络输入 $U_{xi}(t)$ 进行初始化；即 $U_{xi}(t) = U_0' + \delta_{xi}$，其中 $U_0' = \frac{1}{2}U_0 ln(N-1)$，$\delta$ 为（-1，+1）区间的随机数。

$Step$ 5： 动态方程计算 $\frac{dU_{xi}}{dt}$;

$Step$ 6： 根据一阶欧拉法对 U_{xi} 即:

$$U_{xi}(t+1) = U_{xi}(t) + \frac{dU_{xi}(t)}{dt}\Delta T \tag{4.2}$$

$Step$ 7： 使用 sigmoid 函数计算 $V_{xi}(t)$ 即：

$$V_{xi}(t) = \frac{1}{2}(1 + tanh(\frac{U_{xi}(t)}{U_0})) \tag{4.3}$$

$Step$ 8： 计算能量函数 E;

Step 9： 检查路径合法性，判断是否达到规定迭代次数，若没达到返回 *Step* 5。

Step 10： 输出每次迭代的能量函数，最优路径。以及最终的路径图，最短路程。

(三) 遗传算法子程序

Step 1： 对每个种群求解其对应的适应度。

Step 2： 对种群进行选择，交叉，变异操作。

Step 3： 对种族的每个染色体执行模拟退火算法子程序，见（四），传入一个染色体，此温度下链长，邻接矩阵，初始温度，与网络参数。

Step 4： 对模拟退火得到的优解保留到下一代中，即继续执行选择操作。

Step 5： 对此次迭代中得到的新解进行适应度计算。

Step 6： 判断当前温度是否达到终止条件，如果达到则退出子程序，否则对温度进行更新，转 *Step* 3，更新算法为：$T_{i+1} = q \times T_i$，其中 q 代表降温速率，以此来调节算法迭代次数。

(四) 模拟退火算法子程序

Step 1： 初始化各个变量，将每次迭代的结果存储下来以便筛选最优解。

Step 2： 对当前解 S_1 使用交换法进行扰动来产生新解 S_2。

Step 3： 通过 Metropolis 准则来判断是否接受新解，即新解优于旧解则接受，若不优于旧解则根据 $p = exp(\frac{E(S_{new}-E(x_{old}))}{KT})$ 概率接受新解。

Step 4： 在规定的链长下进行迭代，若执行次数达到规定次数则执行退火操作退出子程序。若没有达到规定次数则返回 *Step* 2 继续执行产生新解并筛选的过程。

以上述算法编写程序，得到以下结果：

表 **4-1** 选择操作改进前后运行效率

情况	时间	最优解	平均解
选择修改前	41234s	0.26	0.28
选择修改后	53s	0.18121	0.218

根据表 (4-1) 后发现，无论是时间上，还是运行结果上都不理想，并且收敛速度较差，在运行时间上，通过分析程序可知，程序的大部分时间用于遗传算法的选择函数上，而收敛速度慢可以看出优质解并不容易遗传到下一代种群中，所以对程序的选择函数进行重写，重写后的选择步骤为：

$Step$ 1： 对种群中所有个体按照适应度进行排序。

$Step$ 2： 对该个体进行判断他有 0 ~ n 个个体进入下一代。

$Step$ 3： 对下一代个体进行填充。

$Step$ 4： 下一代种群是否填充完毕，如果填充完毕则退出子函数，若没有填充完毕则返回 $Step$ 2

其中算法中的 n 可以为定值，也可以是种群数与迭代数的一个变量，本论文中取值为定值，由于此选择程序会导致种群过快收敛而导致解空间搜索并不全面，所以本论文将变异概率改为 $min\{\frac{iter^s}{(iter+1)^s}, pm_{max}\}$，其中 s 为定值，$iter$ 为迭代次数，pm_{max} 为所允许的变异率最大值。

(五) k-means 算法

对于 Hopfield 网络引入了混合智能算法，虽然提升了算法局部寻优与全局寻优能力，但是解的求解时间也会增加，随着城市数量的增加其求解时间也会出现较长的问题。为了优化算法的求解速率，我们将引入 K-means 算法来优化求解时间，其算法思想采用了“分而治之”的思想，将大规模 TSP 问题分解为小规模多个 TSP 问题，一方面降低了解的规模，减少了求解时间；另一方面避免了大量无效的劣质解的搜索，优化了解的质量。多个小规模 TSP 问题解决后，采用类间连接将各个类的路径进行相连。下面我们给出 K-means 算法的流程图：

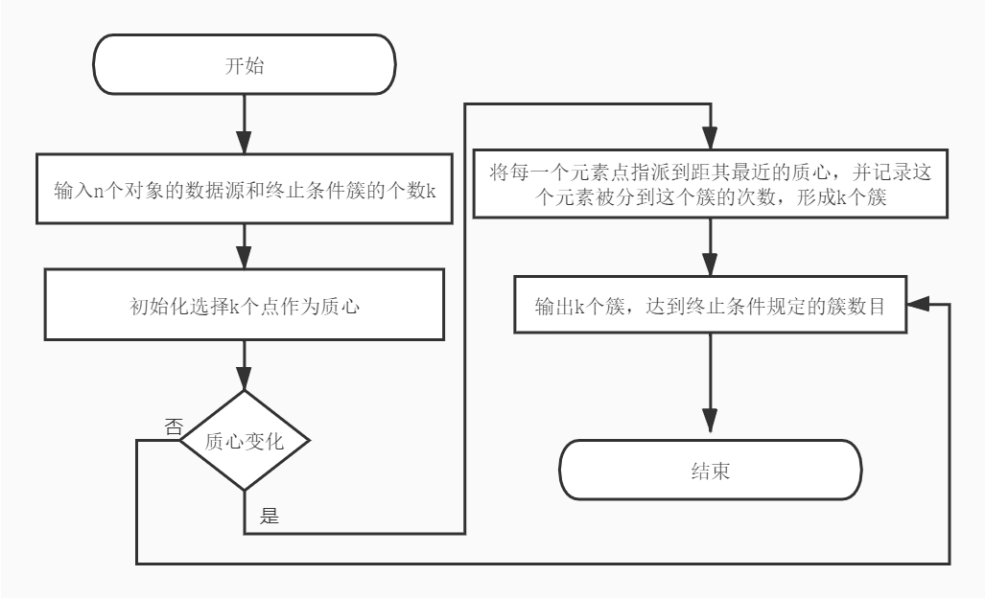

图 4-1　K-means 算法

其中对于最优 k 值的选取则采用轮廓系数的方法来确定：其轮廓系数的求解方法为：

$$s(i) = \frac{b(i) - a(i)}{max\{a(i), b(i)\}} \tag{4.4}$$

其中某一样本 i 到同簇的其余节点的平均距离被称为 a_i，我们称 a_i 为簇不相似度，即其值越小越属于该簇。一个簇中所有节点的簇不相似度称为该簇的不相似度，也是上式的 a_i。

某一样本 i 到其余某簇的所有样本的平均距离作为该样本与某簇的不相似度，样本 i 到簇 j 的不相似度称为 b_{ij}，样本 i 的簇间不相似度，定义为:$b_i = min\{b_{i1}, b_{i2}, \cdots b_{ik}\}$。其值越小样本越属于该簇。选取好最优分类值后我们开始准备簇间连接。

然后运行主程序对每个类的点进行性求解，然后采用类间连接算法对各个类进行相连，其主要思想为贪心思想，即以第一个簇作为起点簇，依次寻找离此簇最近的点所属于的簇即为簇链的下一个簇元素。

三、 针对运输问题集成 Hopfield 网络与其他优化算法进行求解

(一) 算法实现

在（二）中我们已经给出了算法流程图，接下来我们偏向于程序实现的角度具体的对改程序的步骤进行阐述。

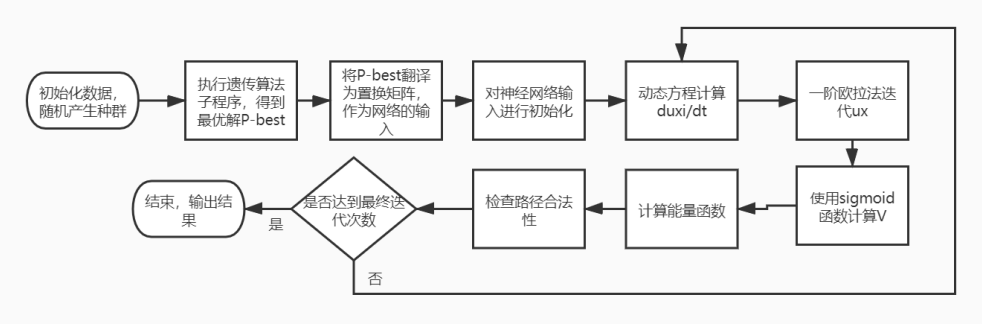

图 4-2 主程序

上图为主程序的实现，其中需要提前求出的变量为，该区域所有点集的邻接矩阵记为 $distance$，进行计算的点集的邻接矩阵 $distance_p$，存储所有点集坐标的 x_{pos}, y_{pos}，以及存储第 i 个点到第 j 个点的路径的 $path-group$，随机生成规定的种群后进入遗传算法子程序，即下图：

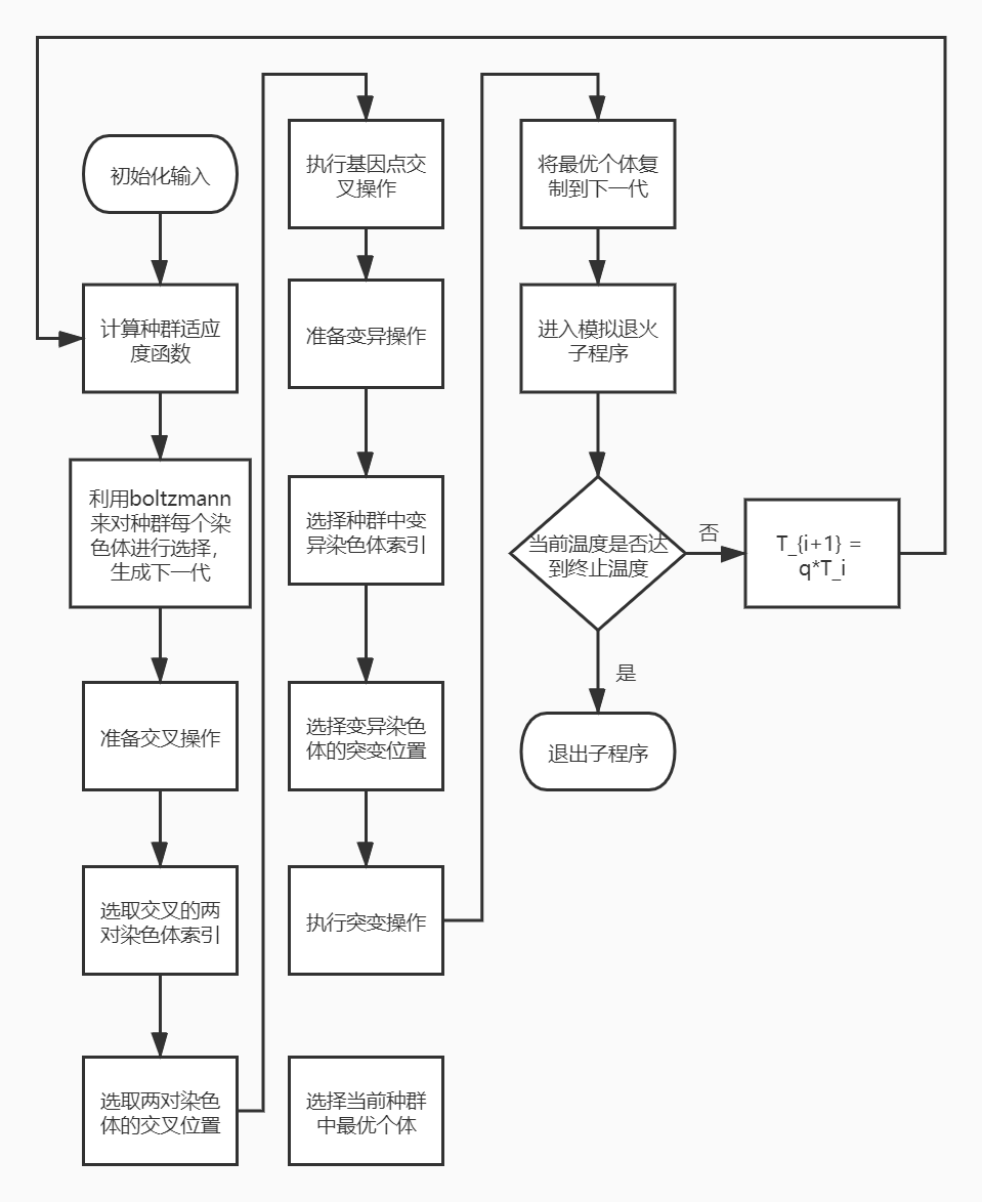

图 4-3　遗传算法子程序

在遗传算法子程序中我们的子程序终止条件为温度降为规定温度，并且子程序将种群，$distance_p$，变异率，交换率，网络指标作为初始输入。其中网络指标是由于需要计算能量函数来求解适应度函数。在每次迭代中程序依次执行选择，交叉，变异操作，然后进入模拟退火子程序，对当前代数的最优解附近的空间进行局部搜索，来弥补遗传算法的局部寻优的短板，经模拟退火子程序返回后，在将返回的最优解复制到下一代。然后根据迭代次数的要求判断是否继续迭代。下面我们介绍模拟退火子程序。

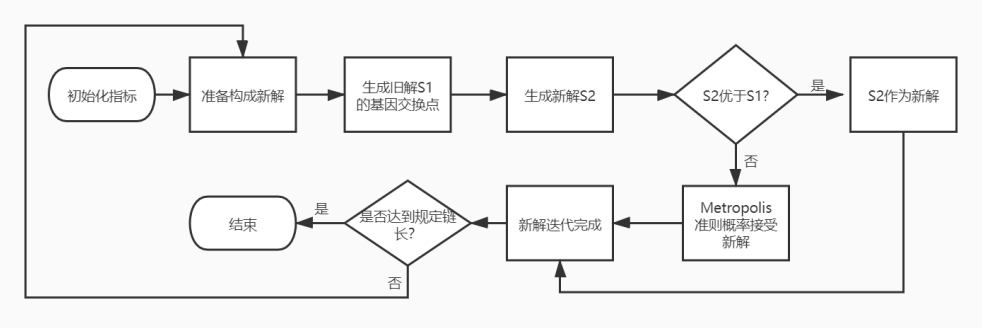

图 4-4　模拟退火子程序

模拟退火子程序的设计目的是为了弥补遗传算法局部寻优的不足，并且遗传算法也可以弥补模拟退火算法的全局寻优的短板。模拟退火程序的结束条件生成了某一温度固定次数的链长。首先对传入的初始解 S_1 进行随机扰动生成新解 S_2，若新解的质量比旧解的质量高则接受新解，否则根据 Metropolis 准则函数对新解进行概率接受。然后记录每次的解，迭代结束后将最优解作为其初始解空间附近寻找到的的最优解返回。

(二)　实例求解

表 **4-2**　程序参数

参数名称	大小
种群数量	1000
网络参数 A	1.5
网络参数 D	1
网络迭代步长	0.02
网络迭代次数	连续五十次解不变结束迭代
交叉概率	65%
变异概率	$\min(\frac{iter}{iter+1}, 0.8)$
初始温度	500
终止温度	1.00E-04
降温速率	0.98
每个温度链长	50

上表中 *iter* 为迭代次数，我们根据以上算法对天津西青大学城附近的地区随

机选取地点进行实验，以 50 个地点的 TSP 问题为例，我们首先运行 k-means 算法，并且根据轮廓系数选取最优的分类数，轮廓系数图如下：

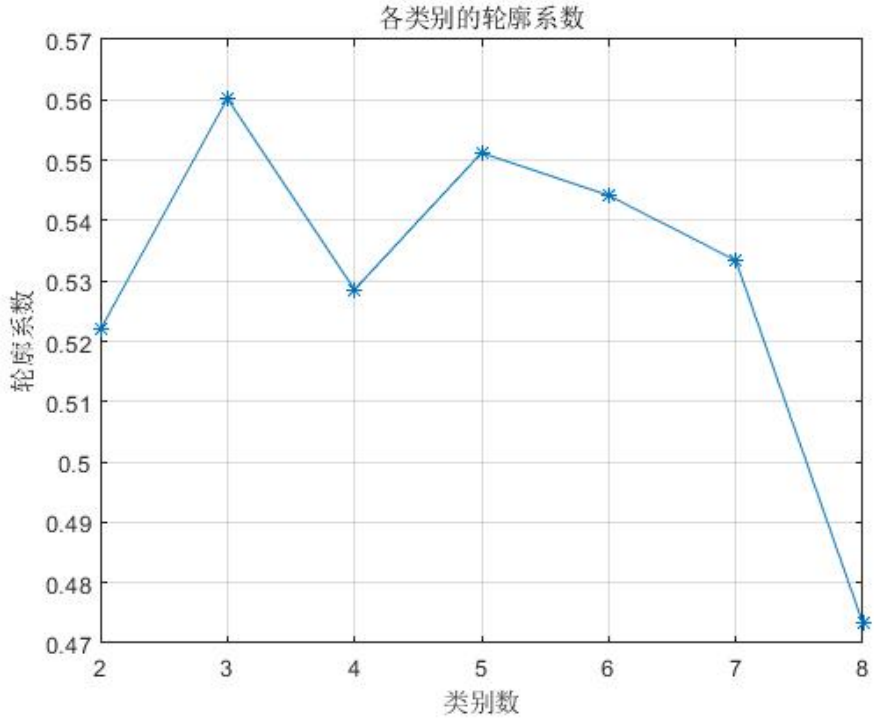

图 4-5　轮廓系数

则最优的分类数为 3，然后运行子程序，得到如下结果：

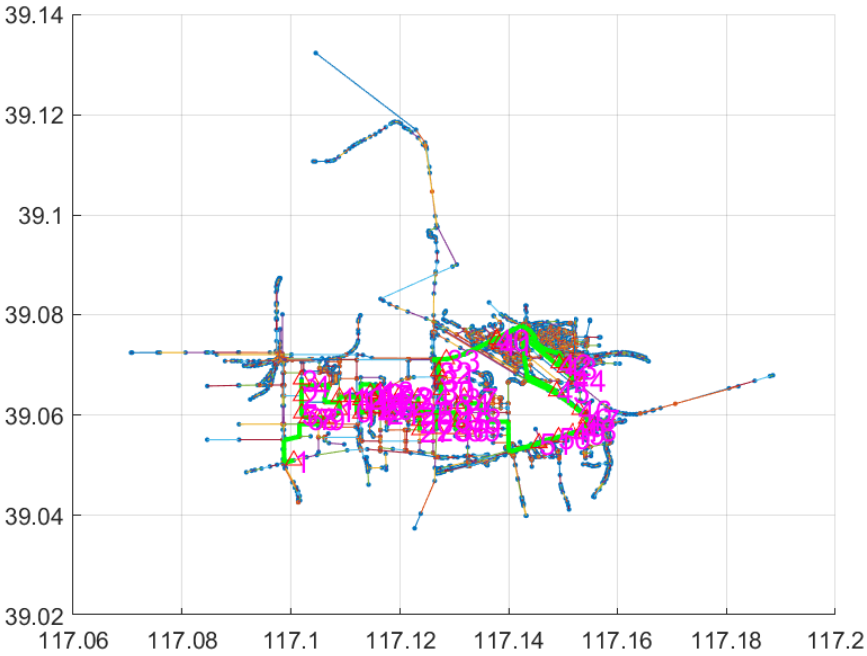

图 4-6　运输路线图

上面为城市路线图，正三角形标记了每一个需要送达的每一个地点，路径通过绿色路线来标注，由于地图点集过多，点集距离跨度较大，导致无法较为清晰的看到其运输路径，所以我们简化了了的路径图像如下：

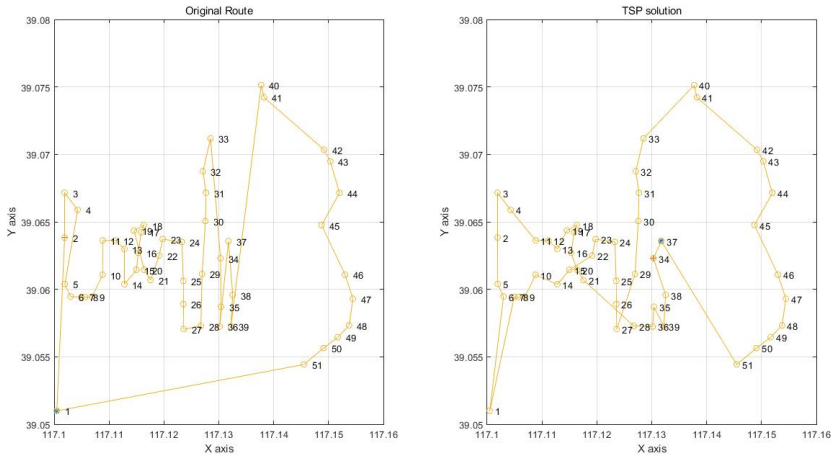

图 4-7　简化运输路线

上图为简化版的运输路线，从上面我们可以看出经过程序优化过的路线图相较于人工绘制的路线图路径更加简单，在配送时间上更加快速程序中 hopfield

的能量函数如下：

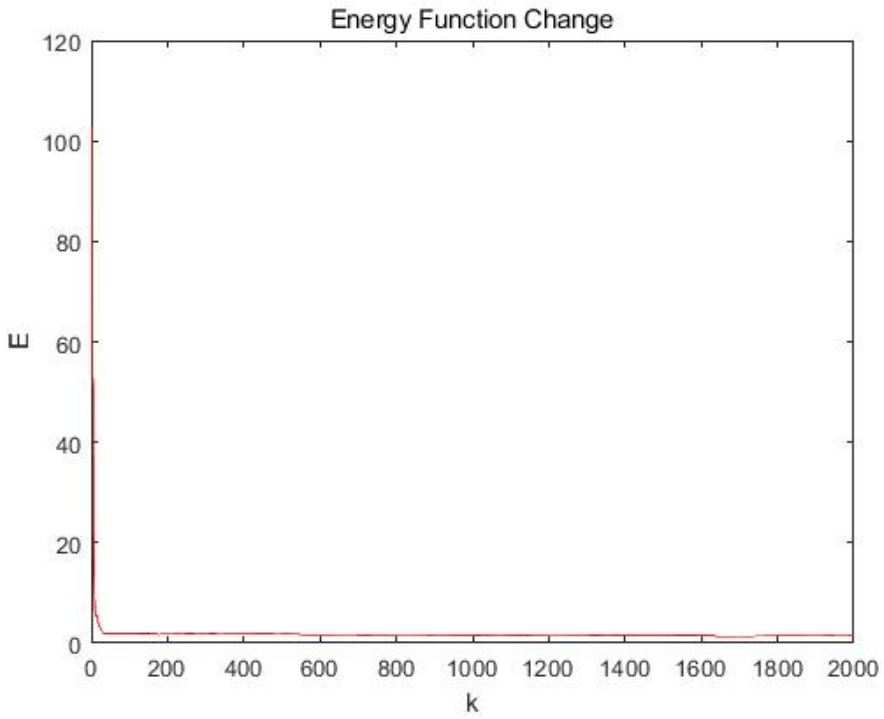

图 4-8　能量函数

此外为了对比此算法的优越性，将 GASA-hopfield 算法与常见的用于解决 TSP 问题的智能算法效率进行比较，结果如下：

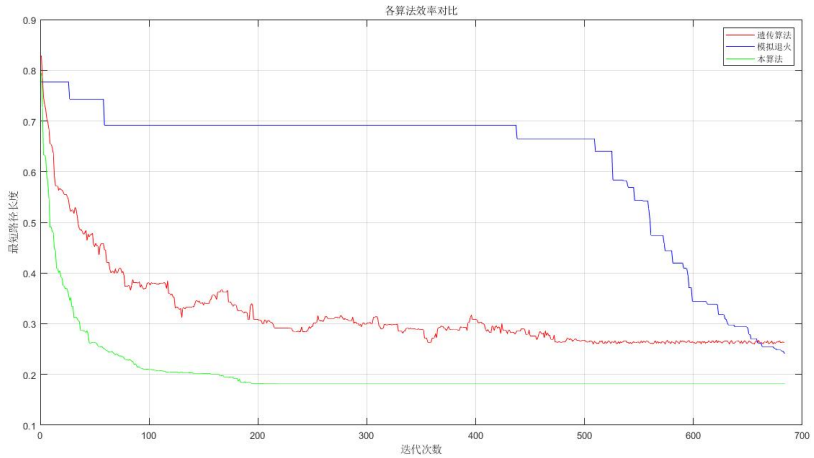

图 4-9　各算法效率对比图

可以看出 GASA-Hopfield 混合优化算法的收敛的比模拟退火与遗传更快，并且也更早的找到了最优解，有趣的是遗传算法由于参数设置的较差所以导致其

陷入了局部最优解，而模拟退火算法则到了 700 此迭代也没有彻底收敛，由此可见本算法的优越性。下面我们将比较的范围扩大到常见的解决 TSP 问题的算法，其对比维度为，求得的最优解，求得的平均解，与求解时间三个维度，运行程序，最后得到如下表格：

表 **4-3**　各算法运行效率

算法	最优解	平均解	运行时间
暴力穷举	0.18121	0.18121	1000s+
剪枝	0.18121	0.18121	1000s+
粒子群	0.25	0.27	119s
模拟退火	0.23	0.25	30s
遗传	0.27	0.28	729s
蚁群	0.1871	0.23	76s
动态规划	0.18121	0.18121	1000s+
GASA-Hopfield	0.18121	0.218	53s

我们可以看出本算法在最优解上达到了已知的最优解 0.18121，平均解的质量也是各智能算法中最优的，并且在求解时间上仅次于模拟退火算法，但仍然在可接受范围内，并且时间上也超过了蚁群遗传以及确定性算法，足以看出 GASA-Hopfield 算法的优越性。

总结

本文根据提出的遗传模拟退火进行改进与 Hopfield 网络进行结合，相对于遗传算法增强了算法的局部搜索能力，避免算法时间过长接质量较差的缺点，相对于模拟退火算法，增强了算法的全局搜索能力，避免了算法陷入局部最优解的状态，相对于 Hopfield 神经网络通过调整神经元初始状态避免了城市数量过大时频繁出现的迭代出非法路径的问题。虽然相对于模拟退火与 Hopfield 神经网络，混合优化算法时间上较长，但是在实际应用中属于可以接受的范围，并且通过改进程序的实现方式还有较大的优化空间。

致谢

感谢国家

参 考 文 献

[1] 仇旺令. 遗传算法和 Hopfield 神经网络集成求解运输优化问题[D]. [出版地不详: 出版者不详], 2005.

[2] 范立南, 吕鹏. 基于改进遗传算法的校园外卖配送路径规划[J]. 物流科技, 2021, 44(01):14-19.

[3] 何黎明. 我国物流业 2020 年发展回顾与 2021 年展望[J]. 中国流通经济, 2021: 1-6.

[4] 胡大伟, 陈海妹, 梁一为, 等. 车辆与无人机混合编队的路径优化问题模型构建[J]. 长安大学学报 (自然科学版), 2021, 41(01):78-89.

[5] 兰兆青. Hopfield 神经网络在 TSP 问题中的应用[D]. [出版地不详: 出版者不详], 2008.

[6] 李博, 颜靖艺. 基于剪枝算法解决非对称 TSP 问题的算法研究[J]. 桂林航天工业学院学报, 2020, 25(04):430-436.

[7] 刘宁钟, 杨静宇. 遗传算法和 Hopfield 模型求解货郎担问题的比较和分析[J]. 计算机工程与应用, 2003(04):95-97.

[8] 马俊, 董良雄, 李军. 一种基于 K-means 改进蚁群算法的船舶航线设计方法[J]. 中国修船, 2020, 33(03):38-41.

[9] 庞燕, 罗华丽, 邢立宁, 等. 车辆路径优化问题及求解方法研究综述[J]. 控制理论与应用, 2019, 36(10):1573-1584.

[10] 帅训波, 马书南. 一种基于遗传 Hopfield 神经网络求解 TSP 问题的算法[J]. 微型机与应用, 2009, 28(21):7-9+15.

[11] 田贵超, 黎明, 韦雪洁. 旅行商问题 (TSP) 的几种求解方法[J]. 计算机仿真, 2006(08):153-157.

[12] 王铁, 胡泓. 基于 K-means 信息挥发速率动态调整的改进蚁群算法[J]. 机械与电子, 2020, 38(02):25-29.

[13] 王银年. 遗传算法的研究与应用[D]. [出版地不详: 出版者不详], 2009.

[14] 王颖. 基于 Hopfield 网络的 TSP 路径优化研究[J]. 赤峰学院学报 (自然科学版), 2015, 31(12):31-33.

[15] 吴高航. Hopfield 神经网络解 TSP 问题及能量函数参数分析[J]. 现代计算机(专业版), 2016(09):9-12.

[16] 闫玉莲. 一种改进的 Hopfield 神经网络对 TSP 问题的求解方法[J]. 闽南师范大学学报 (自然科学版), 2014, 27(03):37-43.

[17] 于兆敏. 基于遗传模拟退火策略的霍普菲尔德神经网络求解 TSP 问题[J]. 中国水运 (下半月), 2019, 19(04):89-91+94.

[18] 于兆敏. 基于遗传模拟退火策略的霍普菲尔德神经网络求解 TSP 问题[J]. 中国水运 (下半月), 2019, 19(04):89-91+94.

[19] 余一娇. 用 Hopfield 神经网络与遗传算法求解 TSP 问题的实验比较与分析[J]. 华中师范大学学报 (自然科学版), 2001(02):157-161.

[20] 张广林, 胡小梅, 柴剑飞, 等. 路径规划算法及其应用综述[J]. 现代机械, 2011(05):85-90.

[21] 张露. 基于改进遗传算法求解带时间窗车辆路径规划问题[J]. 中国物流与采购, 2020(14):66-69.

[22] 朱献文, 张敬. 基于遗传算法的 Hopfield 神经网络应用[J]. 信息与电脑 (理论版), 2011(20):166-167.

附录 A　常见问题

附录 B　联系我们