**哈尔滨工业大学**

**硕士学位论文中期报告**

**题 目：基于HNN分治策略的大规模TSP问题研究**

**院 （系） 机电工程及自动化**

**学 科 机械工程**

**导 师 胡泓教授**

**研 究 生 柴培林**

**学 号 16S153565**

**中期报告日期 2018年3月14日**

**研究生院制**

**二〇一八年三月**

目 录

[1 课题主要研究内容及进度 1](#_Toc508718783)

[1.1课题主要研究内容 1](#_Toc508718784)

[1.2进度介绍 2](#_Toc508718785)

[2 目前主要完成的研究工作及结果 3](#_Toc508718786)

[2.1 暂态混沌神经网络算法的改进 3](#_Toc508718787)

[2.1.1 参数时变增益项改进 3](#_Toc508718788)

[2.1.2 模拟退火函数改进 4](#_Toc508718789)

[2.1.3 暂态混沌神经网络算法的改进综述 5](#_Toc508718790)

[2.1.4 暂态混沌神经网络改进算法的仿真分析 5](#_Toc508718791)

[2.2 Chaotic Hopfield神经网络算法的分治策略 6](#_Toc508718792)

[2.2.1 大规模TSP问题的拆分策略 7](#_Toc508718793)

[2.2.2 凝聚式层次聚类的改进 7](#_Toc508718794)

[2.2.3 小规模TSP的连接策略 11](#_Toc508718795)

[2.3 路径优化可视化程序 12](#_Toc508718796)

[3 后期拟完成的研究工作及进度安排 13](#_Toc508718797)

[3.1后期拟完成的研究工作 13](#_Toc508718798)

[3.2进度安排 13](#_Toc508718799)

[4 存在的问题及解决方案 14](#_Toc508718800)

[4.1存在的问题与困难 14](#_Toc508718801)

[4.2解决方案 14](#_Toc508718802)

[5 如期完成论文工作的可能性 15](#_Toc508718803)

# 1 课题主要研究内容及进度

## 1.1课题主要研究内容

TSP问题(Traveling Salesmen Problem)，即旅行商问题，是一个经典的组合优化问题，其描述是: 给定N个城市，每个城市必须经历一次而且只能经历一次，最后再回到出发点，使所经过的路程最短。TSP问题在实际生产生活中的应用十分广泛，而且随着交通运输线路越来越复杂，电子元器件的集成度越来越高，生产生活中对大规模TSP问题的应用需求也越来越大。

在传统TSP问题的求解过程中，Hopfield神经网络算法(HNN)是一种较为优秀的算法，但是当TSP问题的规模变得复杂时，传统的Hopfield神经网络算法由于自身结构的限制以及其并行运算能力未能得到充分利用，导致算法变得有些力不从心，求解过程变的越来越困难，收敛速度也会变的缓慢。而且由于能量函数使用梯度下降的方法，导致算法容易陷入局部最优解，而不是全局最优解。其原因在于Hopfield神经网络算法自身结构的限制以及其并行运算能力未能得到充分利用。

为更好地解决大规模TSP问题，并且跳出Hopfield算法固有的缺点，本课题将从算法内容、算法结构和算法计算速度三个方面对Hopfield神经网络算法进行改进：

1、算法内容方面

为解决Hopfeild神经网络算法容易陷入局部最优解的缺陷，本课题采用随机性的方法，引入暂态混沌，Hopfield算法收敛时，加入随分岔参数自发衰减而逐渐消失的暂态混沌随机扰动，使算法可以跳出局部最优解，当暂态混沌衰减为零时，算法将收敛到全局最优解。

在前人的研究基础上，本课题添加了增益参数，在算法的混沌搜索阶段，添加一个较小的增益参数，使混沌搜索阶段保持足够长的时间，以便搜索到一个较好的位于全局最优解附近的初值；在算法的稳定收敛阶段，添加一个较大的增益参数，使算法得到一个较快的收敛速度，以便使算法能够从全局最优解附近的初值迅速地跌入全局最优解。

2、算法结构方面

为解决Hopfield算法在大规模TSP问题上的不足，本课题将分治策略和聚类思想引入Hopfield神经网络算法，将大规模TSP问题动态地分解为若干个小规模的TSP问题，求解完成每一个小规模TSP 问题之后，再根据一定的策略，将这些优化之后的小规模TSP网络连接起来还原成原先的大规模TSP问题。算法主要分为两个阶段：

第一阶段为小规模TSP问题的分解与求解。首先，结合分治与层次聚类，根据每个大规模TSP问题自身的特征，动态地将其拆分成若干个小规模TSP问题，小规模TSP问题的个数和各自的规模，由原先大规模TSP问题的特征决定；然后，在每个小规模TSP问题内部，利用Hopfield神经网络算法进行求解。

第二阶段为小规模TSP网络的连接。将提出一种连接策略，把经过优化之后的小规模TSP网络连接起来，还原成原先的大规模TSP问题。

3、算法计算速度方面

对改进过后的Hopfield神经网络算法的结构进行并行化改进，使之能够应用在CUDA计算框架上，利用CUDA并行计算，在计算模式层面对算法继续加速。

完成对算法的改进之后，将搭建全自动压接机实验平台，以验证算法的路径优化效果。

## 1.2进度介绍

目前，已经完成了：

（1）算法内容方面的改进，结合混沌动力学对Hopfield算法提出了改进策略，实现了改进之后的Chaotic Hopfield神经网络算法，完成对改进Chaotic Hopfield神经网络算法的仿真分析，并与为改进算法进行了对比，验证了算法的有效性；

（2）算法结构方面的改进，提出了分治、聚类和连接策略，实现了Chaotic Hopfield神经网络分治改进算法，以一组60城市数据为例，完成对算法的仿真分析，验证了算法的有效性。

# 2 目前主要完成的研究工作及结果

## 2.1 暂态混沌神经网络算法的改进

### 2.1.1 参数时变增益项改进

暂态混沌神经网络求解TSP的过程分为两个阶段，即混沌搜索阶段和稳定收敛阶段。要使算法有较高的寻优率，就必须使其混沌搜索阶段保持足够长的时间，这样才能为接下来的稳定收敛阶段提供一个较好的有可能位于全局最优解附近的初始值，而要使该算法有较快的收敛速度，就必须使网络在其稳定收敛阶段能迅速地从位于全局最优解附近的初始值跌落到全局最优解。因此，如何使混沌搜索阶段保持足够长的时间以便可以搜索到一个较好的位于全局最优解附近的初值，同时接下来的网络稳定收敛阶段又有较快的收敛速度，这对求解的质量有重要的影响。

通过对常规Hopfield神经网络模型的研究可以知道，神经元输出函数即函数的增益参数对神经网络的收敛动力学特性有着重要的影响。即增益过大时，神经元输出函数趋于离散阶跃函数，导致神经网络的收敛速度过快，从而不能获得最优解当增益过小时，又会使神经元输出函数的曲线过于平坦，神经元的状态不易收敛到“0”或“1”，网络不能在规定的迭代次数内收敛。因此，如果模型在其混沌搜索阶段有一个较小的增益参数，可以使网络的收敛动力学特性较弱而在接下来的稳定收敛阶段有一个较大的增益参数，则可以使网络有较强的收敛动力学特性。这样，就可以在获得较好的混沌搜索能力的同时也加快了原有算法的稳定收敛速度。

根据上述的讨论，提出了如下形式的改进模型：

 式（2-1）

 式（2-2）

 式（2-3）

 式（2-4）

其中，t为离散时间步长（t=0, 1, 2, …）；为神经元i的输出；为神经元的内部状态；为神经元i的动态时变的控制参数; 为输出函数的增益参数；为从神经元j到i的连接权值；为神经元i的输入偏置值；为正常数；k为神经元的衰减因子；为输入的正值缩放因子；为的衰减因子。

### 2.1.2 模拟退火函数改进

暂态混沌神经网络模型的动态特性很敏感地依赖于自反馈连接权值，它类似于随机模拟退火中的温度，是动态变化的，代表着退火方案。

从混沌到平衡点的演变分为两个过程动态搜索和收敛过程。在前半段表现为混沌现象，有丰富的动态特征，这一过程完成相空间的搜索行为，当下降到某一程度时，混沌现象消失，转到收敛阶段。而的下降速度直接影响到优化算法的两个重要指标准确性和速度。当下降速度很快时，将通过短暂的搜索阶段直接进入收敛过程，因此算法的整体速度很快，但因为没有充分利用混沌的丰富动态特征，容易陷入局部最小值，准确性大大降低。相反地，如果变化小，可以提高准确性但降低收敛的速度。

常用的三种退火函数为指数退火、线性退火函数和王凌退火函数：

1. 线性退火函数

 式（2-5）

1. 指数退火函数

 式（2-6）

1. 王凌退火函数

 式（2-7）

本课题在这几个常用退火函数的基础上，提出了新的分段指数退火函数，既充分利用混沌的动态特性进行搜索，使算法可以从局部最优的“陷阱”中跳出，又克服其带来的速度问题，减小收敛时间。

（4）改进分段指数退火函数

 式（2-8）

### 2.1.3 暂态混沌神经网络算法的改进综述

结合参数时变增益的改进和模拟退火函数的改进，最终模型修改为：

 式（2-1）

式（2-2） 式（2-5）

 式（2-4）

其中，t为离散时间步长（t=0, 1, 2, …）；为神经元i的输出；为神经元的内部状态；为神经元i的动态时变的控制参数; 为输出函数的增益参数；为从神经元j到i的连接权值；为神经元i的输入偏置值；为正常数；k为神经元的衰减因子；为输入的正值缩放因子；为的衰减因子。

### 2.1.4 暂态混沌神经网络改进算法的仿真分析

利用基本Hopfield神经网络算法和暂态混沌神经网络算法求解10城市旅行商问题，其中10城市的坐标如下：

表2-1 10城市坐标

|  |  |
| --- | --- |
| 城市 | 坐标 |
| City1 | (0.4000, 0.4439) |
| City2 | (0.2439, 0.1463) |
| City3  City4  City5  City6  City7  City8  City9  City10 | (0.4000, 0.4439)  (0.4000, 0.4439)  (0.4000, 0.4439)  (0.4000, 0.4439)  (0.4000, 0.4439)  (0.4000, 0.4439)  (0.4000, 0.4439)  (0.4000, 0.4439) |

采用基本Hopfield神经网络模型和暂态混沌神经网络模型对上文提到的 TSP问题进行优化仿真，分别进行30次实验，其中网络的基本参数取为：A=B=D=500，C=1000，μ=0.02，运行程序并统计实验结果，如表2-2中所示。

表2-2 10城市优化结果对比

|  |  |  |
| --- | --- | --- |
| 选项 | Hopfield | Chaotic Hopfield |
| 运行次数 | 30 | 30 |
| 最优解次数  搜索到有效路径次数 | 2  5 | 25  29 |
| 最优解路径长度 | 2.6907 | 2.6907 |

由表 2-2中可以看出，无论是基本Hopfield神经网络算法还是暂态混沌神经网络算法，都搜索到了最优路线，这说明两种算法在TSP问题中都是有效的。但在30次仿真结果中，暂态混沌神经网络算法搜索到了25次最优解，而基本Hopfield神经网络算法只搜索到 2次；从搜索到的总的有效路线来看，暂态混沌神经网络算法共搜索到 29次有效路线，远远高于基本Hopfield神经网络算法搜索到的5次。由此可以得出，暂态混沌神经网络搜索效率明显高于基本Hopfield神经网络。在求得每一个合法解时所经过的非法解次数中，暂态混沌神经网络算法小于基本Hopfield神经网络算法，说明暂态混沌神经网络算法的优化速度明显快于基本Hopfield神经网络算法。

## 2.2 Chaotic Hopfield神经网络算法的分治策略

算法的整体思想是，根据每个大规模TSP问题自身的分布特征，动态地划分为若干个小规模的TSP问题，对每一个小规模TSP使用Hopfield神经网络进行优化求解，然后使用某种连接策略将优化之后的小规模TSP连接，完成对大规模TSP问题的求解。

### 2.2.1 大规模TSP问题的拆分策略

大规模TSP问题的拆分，就是按照空间距离的分布，将其包含的所有城市划分为若干个组，这个过程是一种典型的聚类过程。将物理或抽象对象的集合分成相似的对象类的过程称为聚类，通过聚类能够识别对象空间中的稠密和稀疏区域，从而发现全局分布模式和个体属性之间的相互关系。

本课题讨论一类特殊的大规模TSP问题，即顶点数N较大且有些点比较聚集，各聚集类相互距离比较远的特殊类型，适合用聚类方法来减小问题规模，方便求解。

依据各簇内对象相似性最大化和各簇间对象相似性最小化的基本聚类分析原则，学术界提出了许多种不同的聚类方法，主要有以下几种：划分的方法、层次的方法、基于密度的方法，基于网格的方法和基于模型的方法，这些方法根据自身的特点分别应用于不同的领域中。

在各种聚类方法中，最常用的方法是划分方法和层次方法。而划分聚类的算法中最经典的是K-means，K-medoids及它们的变种。K-means算法采用某种邻近性度量函数将将每个对象指派到与其最为相似的簇中，同时更新每个簇的中心，重复指派和更新步骤，直至邻近性度量函数收敛，即每个簇中的对象都不再发生变化为止。K-means算法虽具有算法简单和收敛速度快的特点，但K值的预先未知性及初始中心位置选择的随机性都将有可能产生不同的聚类结果，进而导致结果簇集的不确定性。而层次聚类方法按照树状图的形成方式是自顶向下的分解还是自底向上的合并，分为分裂层次聚类算法和凝聚层次聚类算法。凝聚层次聚类算法最初将每个对象作为一个簇，然后依据某种准则(如单链、全链、组平均、War方法等)依次合并邻近性最优的簇，直至所有对象都被包含于一个簇中或某个终结条件被满足。层次聚类算法具有较好的聚类质量，聚类结果也较为稳定。

与分裂式方法相比，凝聚式算法从一个层次到另一个层次所需要的计算方法比较简单快捷，而且能够得到较好的聚类结果。因此，本课题就将使用凝聚式层次聚类的方法，确定合理的聚类数目，并进行聚类

### 2.2.2 凝聚式层次聚类的改进

传统的K均值法在进行聚类时，需要预先指定聚类数目，才能继续执行聚类操作，层次聚类法只有指定聚类数目，才能确定聚类终止条件。有没有一种方法和度量准则能够根据TSP问题的实际情况来确定合理的聚类数目呢？

以凝聚式层次聚类为基础，本文提出一种新的聚类方法，以确定聚类的最佳数目：首先，使用经典的层次聚类算法进行聚类，得到一个树形图，保存每一层的信息；然后，提出一种新的聚类有效性指标，对每一层进行判断，来确定最佳聚类层，得到聚类的最佳数目。

1、使用经典层次聚类法，将原始点集进行层次聚类得到一个树形图。

凝聚层次聚类法的主要思想是：开始将每个对象形成单独的组，然后逐次合并相近的对象或组，直到所有的组合并为一个整体，或者满足某个终止条件，设点集中元素的个数为N，层次聚类法算法如下：

Step1 将每个对象归为一类，共得到N 类，每类仅包含一个对象，类与类之间的距离就是他们所包含对象之间的距离；

Step2 找到距离最近的两个类合并为一类，这时总类数少一个；

Step3 重新计算所有新类和旧类之间的距离；

Step4 重复 Step2 和 Step3 直到最终合并为一个类为止。

层次聚类法如果没有指定聚类数目，所有的点最终将聚为一类，得到一个带有聚类过程的树形图，规模为N的点集在使用凝聚层次聚类法后，得到一个如图2-1所示类似的一个树形图，每聚一层，类的数目减少1。这样，通过凝聚层次聚类法得到的树形图共有N层，从第一层到第N层，簇的数目从N到1。如何判断哪一层是最优的聚类呢？下面提出了一个在层次聚类法基础上判断最优聚类数目的聚类有效性指标。

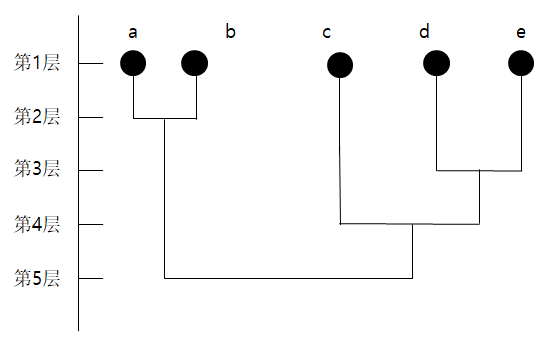


图 2-1凝聚式层次聚类树状图

2、聚类有效性指标

评价聚类结果优劣的过程称为聚类有效性分析。一般来说，一个好的聚类划分应尽可能反映数据集的内在结构，使簇内样本尽可能相似，簇间样本尽可能不相似。从距离测度考虑，就是使簇内距离极小化而簇间距离最大化的聚类是最优聚类。簇内和簇间，我们使用凝聚度的概念来判断聚类质量的好坏

若表示单个簇i，且簇内元素大于1，是簇中的一个对象，是簇 的中心，表示对象距离簇中心的距离，是簇中对象的个数，则簇的簇内凝聚度为：

 式（2-3）

对式（2-3），若某簇是个单元素簇，则值为0。

根据式（2-3）关于簇内凝聚度的定义，我们提出了以下关于簇间凝聚度的概念：

定义2.1：若某点集分为K个簇，K≥2，表示单个簇 i，任意簇的中心为，点集中各个簇的中心点的中心为g，则点集的簇间凝聚度为:

 式（2-4）

式(2-4)所定义的点集的簇间凝聚度要求聚类数目K≥2，因为所有点聚为一类时，与g重合，的值为零，且不符合所求的带有聚类特征的TSP问题的实际意义。

定义2.2：若某点集共包含N个点，分为K个簇，K≥2，则点集的簇间离散度为:

 式（2-5）

性质2.1：由式(2-3)可知，当的值越小，单个簇的簇内凝聚度越小，簇越紧密，单个簇的聚类质量越好。由定义2.1可知，当越大，各个簇之间越分散，簇集的聚类质量越好。由定义2.2可知，当越小，各个簇之间越分散，簇集的聚类质量越好。

计算出了树形图中每一层的簇内凝聚度、簇间凝聚度以及簇间离散度，由性质2.1可以得到如下性质：

性质2.2：若在树形图某层点集聚为K个簇，设参数

 式（2-6）

其中，m、n、p分别为、、的指数。

由性质2.1知，t值越大，聚类质量越好，t值越小，聚类质量越差。

由于凝聚层次聚类法是从每个对象为单元素簇开始聚类的，因此，从第一层起所有元素是单元素簇，从第二层，才至少有一个多元素簇，由式(2-6)可知，分母为，在第一层，所有簇集均为单元素簇，，不满足对参数t的限制条件，因此计算各层的聚类质量，需要从L≥2开始，且式(2-6)所定义的点集的簇间凝聚度要求聚类数目

K≥2，即L≤N−1。所以，对层次聚类法得到的有N个层的树形图各层进行判定时，判定层次的范围是L≥2且L≤N−1。

通过上面介绍的定义和性质，可以通过层次聚类的树形图得到最佳的聚类数目K。

算法2.2：（层次聚类法聚类及确定合理的聚类数目）

Step1首先对点集用算法5.1 进行聚类，得到一个树状图；

Step2树状图中若某层簇集个数为K，在L≥2且L≤N−1区间计算各层的各个簇的簇内凝聚度(i =1 ~ k)、簇间凝聚度值及簇间离散度；

Step3计算；

Step4确定最大的t值所在层的聚类数目K，并且得到各个点的聚类情况，即每个点属于哪一簇集，这样，点集的聚类过程结束。

利用算法2.2进行聚类，能够在事先不确定聚类数目的情况下，确定聚类最佳数目及合理聚类。本文提出的式(2-6)对聚类质量进行了合理的判定，并对算法进行了实现，证明了该有效性指标的合理性。

### 2.2.3 小规模TSP的连接策略

采用以上聚类算法5.2 后，得到最佳的聚类数目和聚类簇集，将大规模旅行商问题中的顶点分成了若干小类，各个小类之间距离较远，每个类内的城市距离又较近，每个类的规模已经减小，由2.1小节介绍的Chaotic Hopfield神经网络算法可以求解出每个簇集内的最优回路，那么怎么样将这些簇集内的近似最优回路连接起来，得到一条整个点集的近似最优回路呢？下面介绍一种连接策略。

算法2.3（各子回路之间的连接算法）

Step1首先找到距离最近的两个簇，这里的距离是指任意两个簇中最近的点之间的距离，即，其中表示簇r中任意一点，表示簇s中任意一点。

Step2设簇r与簇s中距离最近的两个点分别为和，连接，与相邻的两个点为和，与相邻的两个点为和，计算和，的大小，若较小，则连接，移除,；否则，连接，移除,，两个回路合并为一个回路；

Step3将连接后的回路作为一个整体，再来计算距离最近的两个点所在的两个簇，按照Step2来进行连接，直到所有簇集连接为一个整体。

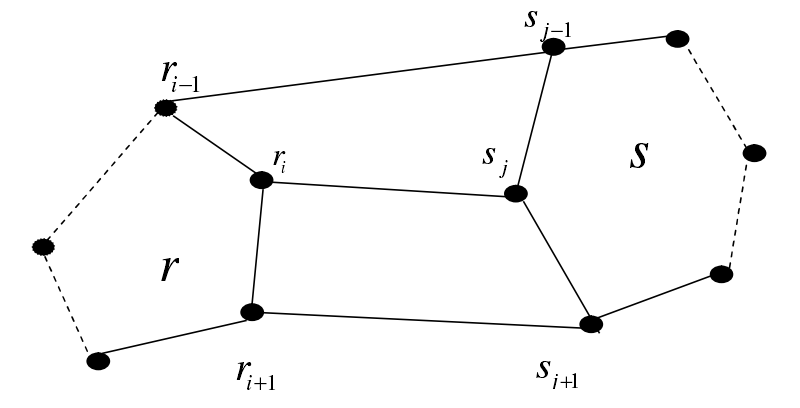


图2-2 回路之间连接方法简单示意图

## 2.3 路径优化可视化程序

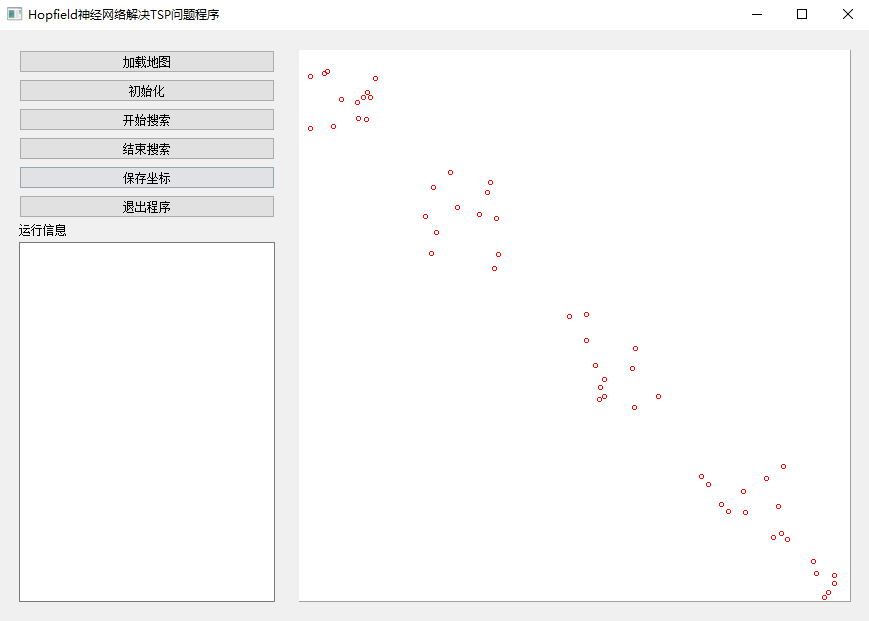


图2-3 Hopfield神经网络解决TSP问题程序界面

以一组随机产生的包含60个城市的数据集为例，未使用路径优化时，初始总距离未3710.92，优化之后总距离为2198.88，优化率为40.7%。

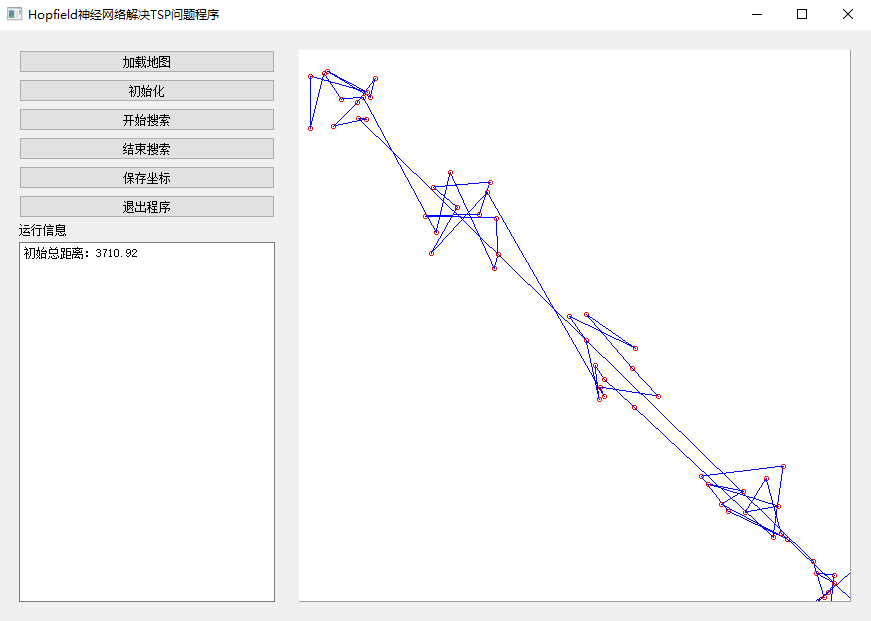
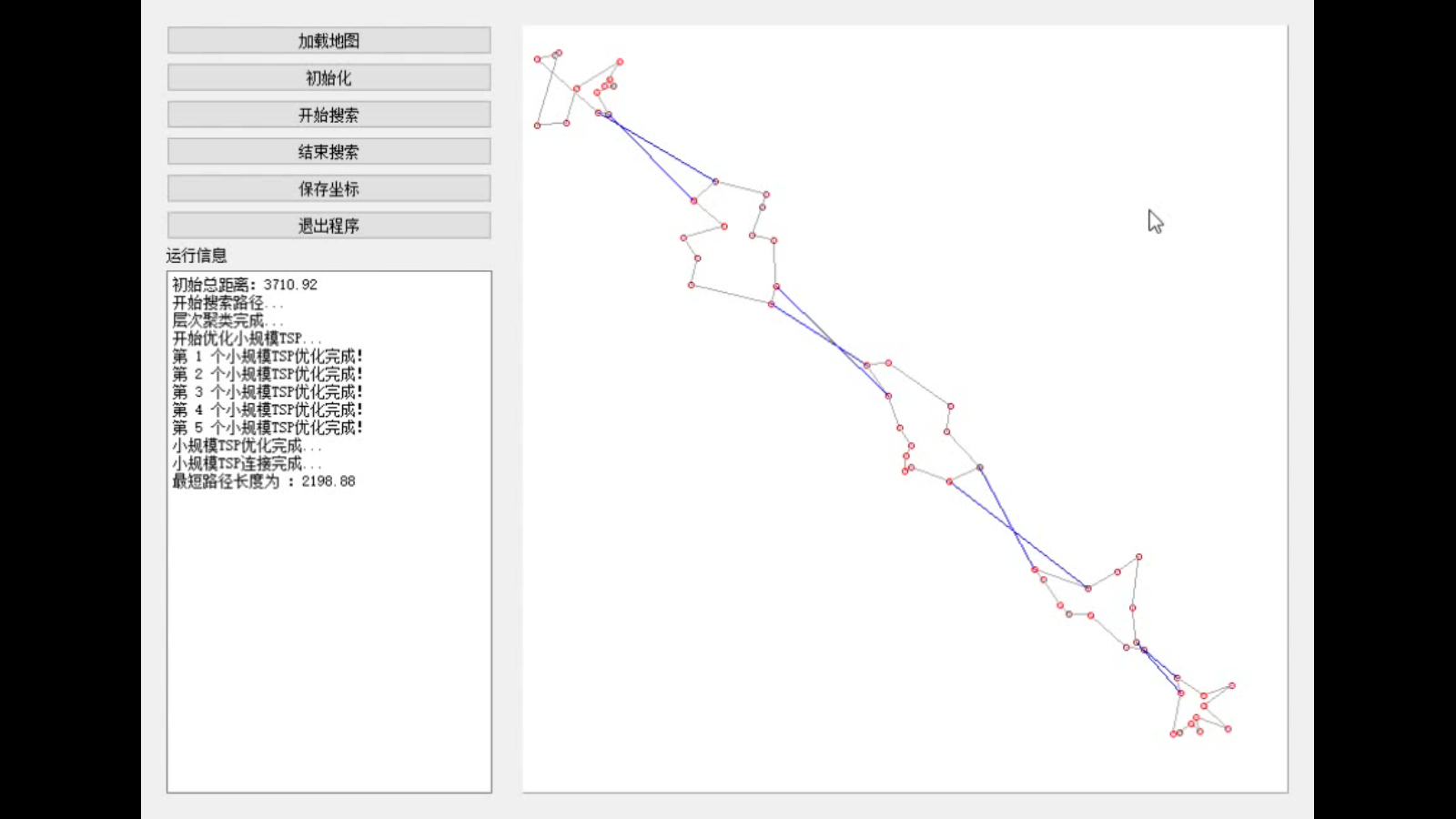
 

图2-4 60个城市初始总距离 图2-4 60个城市优化后距离

# 3 后期拟完成的研究工作及进度安排

## 3.1后期拟完成的研究工作

后续需要完成的工作主要有：

1、对连接策略进行优化；

2、尝试对任意的大规模TSP问题的求解，使用卷积神经网络迁移学习的方法，训练样本进行尝试；

3、将算法进行分布式并行加速，结合英伟达公司推出的CUDA框架改进程序算法；

4、结合压接机平台，验证算法的效果；

5、完善和优化整个系统，撰写论文，准备答辩。

## 3.2进度安排

根据目前课题的完成情况及自身情况，制定出以下研究进度：

表3-1 研究进度

|  |  |
| --- | --- |
| 时间 | 进度安排 |
| 2018.03——2018.04 | 改进算法的连接策略 |
| 2018.04——2018.05 | 搭建迁移学习框架，使用CNN对数据集进行分类 |
| 2018.06——2018.07  2018.08——2018.09 | 对算法进行并行改进，使用CUDA改造程序  搭建全自动压接机实验平台，验证算法的效果 |
| 2018.10——2018.12 | 完善和优化整个系统，撰写论文，准备答辩 |

# 4 存在的问题及解决方案

## 4.1存在的问题与困难

目前，课题主要存在以下问题：

1、小规模TSP的连接不尽合理；

2、TSP规模变大时，聚类算法的速度变慢；

3、对CUDA并行加速知识掌握的不够。

## 4.2解决方案

对目前存在的问题，提出以下解决方案：

1. 继续对小规模TSP的连接策略进行优化；
2. 对聚类的算法进行优化，减小其时间复杂度；
3. 继续查阅资料和文献，学习CUDA的原理与用法。

# 5 如期完成论文工作的可能性

到目前为止，课题初步完成了对Hopfield神经网络算法的算法内容和算法结构的改进。目前正在对改进内容进行优化，同时正在尝试从卷积神经网络迁移学习的角度去解决TSP问题的拆分。待完成这些内容之后，将会结合CUDA对算法进行并行处理，以提高算法计算速度，然后将搭建全自动压接机实验平台，验证本课题算法的效果。

综上所述，论文能够按期完成，并取得一定的研究成果。