人工智能课程期末大作业

——————旅行路线TSP问题

计算机2001 袁子宸[[1]](#footnote-1) 20205962

指导老师：张恩德

摘要

在这篇文章中，我使用了一种不同于以往遗传算法、粒子群算法等的启发式算法的方式来求解TSP问题。我使用了Hopfield神经网络的方法，对于这个最优化的NP问题，我通过构建能量函数，状态函数，在34个中国城市组成的图上 来求解最终的稳定状态，即最短总路径。这个方法简单有效，相比于传统的启发式算法可以在较短的时间内根据随机初始化的参数推导出相对应的次优解，并且通过多次的实验最终无限逼近全局最优解。

1 介绍

旅行商问题 [1]，即TSP问题（Traveling Salesman Problem）又译为旅行推销员问题、[货郎担问题](https://baike.baidu.com/item/%E8%B4%A7%E9%83%8E%E6%8B%85%E9%97%AE%E9%A2%98/4844663?fromModule=lemma_inlink" \t "_blank)，是数学领域中著名[问题](https://baike.baidu.com/item/%E9%97%AE%E9%A2%98/1067365?fromModule=lemma_inlink" \t "_blank)之一。假设有一个旅行[商人](https://baike.baidu.com/item/%E5%95%86%E4%BA%BA/1243610?fromModule=lemma_inlink)要拜访个城市，他必须选择所要走的路径，路径的限制是每个城市只能拜访一次，而且最后要回到原来出发的城市。路径的选择目标是要求得的路径路程为所有路径之中的最小值。

在[计算复杂性理论](https://en.wikipedia.org/wiki/Computational_complexity_theory)中，TSP的决策版本（给定长度L，任务是决定图是否最多有L的巡视）属于[NP完全](https://en.wikipedia.org/wiki/NP-completeness)问题。因此，TSP 的任何算法[的最坏情况](https://en.wikipedia.org/wiki/Best,_worst_and_average_case)[运行时间](https://en.wikipedia.org/wiki/Time_complexity)都可能随着城市数量的[增加而超多](https://en.wikipedia.org/wiki/Time_complexity#Superpolynomial_time),但不超过[指数级](https://en.wikipedia.org/wiki/Exponential_time_hypothesis)。

在历史上已知的最早时间是在1832年，一本德国旅行推销员手册中非正式地描述了问题，但没有包含任何数学处理。在之后的几十年中，美国的经济社会去的了巨大的发展，公路交通快速建设，与此同时诞生了一种广泛存在的职业，旅行推销员，其工作内容是在不同的地点、城市中穿梭，挨家挨户上门推销公司的产品，并且这种工作一直流行到二十世纪六七十年代才逐渐消失。在这个产业的早期1883年的时候美国估计就有20万名旅行推销员在奔波的路上。

TSP的一般形式被认为是在20世纪30年代在维也纳和哈佛大学首次由数学家研究的，特别是[卡尔·门格尔（Karl Menger](https://en.wikipedia.org/wiki/Karl_Menger)），他定义了这个问题，考虑了明显的蛮力算法，并观察到最近邻启发式的非最优性，是优化中研究最深入的问题之一 [2]。它被用作许多优化方法的[基准](https://en.wikipedia.org/wiki/Benchmark_(computing))。在后来的比较长的时间内这个问题在计算上很困难，但许多[启发式算法](https://en.wikipedia.org/wiki/Heuristic)和[精确的算法](https://en.wikipedia.org/wiki/Exact_algorithm)是的提出，使得现在可以完全解决一些具有数万个城市的实例，甚至数百万个城市的问题也可以在1%的小部分内近似。

2 问题定义

最简单的作为图论问题，其可以描述为[无向加权图](https://zh.wikipedia.org/wiki/%E5%9B%BE_(%E6%95%B0%E5%AD%A6))来对TSP建模，则城市是图的[顶点](https://zh.wikipedia.org/wiki/%E9%A1%B6%E7%82%B9_(%E5%9B%BE%E8%AE%BA))，道路是图的[边](https://zh.wikipedia.org/wiki/%E5%9B%BE%E8%AE%BA%E6%9C%AF%E8%AF%AD)，道路的距离就是该边的长度。它是起点和终点都在一个特定[顶点](https://zh.wikipedia.org/wiki/%E9%A1%B6%E7%82%B9_(%E5%9B%BE%E8%AE%BA))，访问每个顶点恰好一次的最小化问题。通常，该模型是一个[完全图](https://zh.wikipedia.org/wiki/%E5%AE%8C%E5%85%A8%E5%9C%96)（即每对顶点由一条边连接）。如果两个城市之间不存在路径，则增加一条非常长的边就可以完成图，而不影响计算最优回路。

在考虑了不同节点之间的通行的能力和消耗，我们需要从非对称和对称角度来理解。在对称TSP问题中，两座城市之间来回的距离是相等的，形成一个[无向图](https://zh.wikipedia.org/wiki/%E5%9B%BE_(%E6%95%B0%E5%AD%A6))。这种对称性将解的数量减少了一半。在非对称TSP问题中，可能不是双向的路径都存在，或是来回的距离不同，形成了[有向图](https://zh.wikipedia.org/wiki/%E6%9C%89%E5%90%91%E5%9B%BE)。[交通事故](https://zh.wikipedia.org/wiki/%E8%BB%8A%E7%A6%8D)、[单行道](https://zh.wikipedia.org/wiki/%E5%8D%95%E8%A1%8C%E9%81%93)和出发与到达某些城市机票价格的不同等都是打破这种对称性的例子。

在本次的作业中我们定义了中国地图上的34个城市节点，并且任意两点都是可达的，并且两点指点的通行开销或者距离式简单的直线距离，也就是欧式距离。

故我们需要求解的问题是 [3] ：

* 输入将是 个点的集合,
* 解空间：所有可能的走遍所有节点且不重复的回路，也就是点集的哈密顿图。
* 旅行的成本：旅行的总长度，也就是旅行中两两点之间的距离总和。
* 目标：找到成本（总长度）最小的路径。

3 相关工作

旅行商问题(TSP)最早提出于1930年，是运筹学(OR)领域研究最深入的组合优化问题之一。找到最优TSP解决方案是NP困难的，即使在二维欧几里德情况下，节点是二维点，边权是点对之间的欧几里德距离 [4] 。在实践中，TSP求解者依赖于精心设计的启发式来指导他们的搜索过程，以高效地找到具有数千个节点的图的近似解。1959年，[吉莉安·比尔德伍德](https://en.wikipedia.org/wiki/Jillian_Beardwood)、J.H.哈尔顿和[约翰·哈默斯利](https://en.wikipedia.org/wiki/John_Hammersley)在剑桥哲学学会的期刊上发表了一篇题为“穿越许多要点的最短路径”的文章。德伍德-荷顿-哈默斯利定理为旅行推销员问题提供了一个实用的解决方案。作者推导出一个渐近公式来确定从家中或办公室开始并在返回起点之前访问固定数量的位置的销售员的最短路线的长度。并且在之后的几年诞生了一个具有里程碑意义的工作，1962年的赫尔德-卡普算法，提出了一个复杂度为的算法。如今，最先进的TSP求解器，如Concorde [5]，使用线性规划的方法，以最佳的方式解决了大规模的TSP问题，它是一个由85900个城市节点构成的复杂网络。除此之外减少解搜索的分支定界方法外，还利用切平面算法[6]，迭代求解TSP的线性规划弛缓空间。

因此，为组合优化问题设计良好的启发式通常需要大量的专业知识和多年的研究工作。由于这些问题的高度结构化性质，神经网络已被用来学习近似策略，特别是对于非微不足道的问题，以设计启发式[7]。历史上的工作主要集中在使用变形模板模型(deformable template models)[8]。然而，这些方法的基准性能在速度和解决方案质量方面都比不上算法方法 [9]。

4 我的方法

我在这里使用Hopfield Neural Network来解决这个问题。对于TSP问题，其本质上是一个最优化问题，一个通用的方法流程是构建目标函数，找到可行解，最后在解空间内找到最优解。我在这里可以分别对应到连续Hopfield网络中定义的三种工具，能量函数，状态函数，和最终的稳定状态。

4.1 Hopfield Neural Network

作为记忆模型的神经网络的[伊辛模型](https://en.wikipedia.org/wiki/Ising_model)由[William A. Little](https://en.wikipedia.org/w/index.php?title=William_A._Little_(physicist)&action=edit&redlink=1)于1974年首次提出，霍普菲尔德在他1982年的论文中得到了认可。具有连续动力学的网络是由霍普菲尔德在他1984年的论文中开发的。2016年，克罗托夫和霍普菲尔德通过网络动力学和能量函数的变化，开发了内存存储容量的重大进步。这个想法在2017年由德米西吉尔和合作者进一步扩展。大内存容量模型的连续动力学是在2016年至2020年间的一系列论文中开发的。大内存存储容量的霍普菲尔德网络现在被称为密集关联记忆或[现代霍普菲尔德网络](https://en.wikipedia.org/wiki/Modern_Hopfield_network)。

4.1.1 Hopfield Neural Network的结构

Hopfield网络的灵感来自于生物的神经网络，并用二进制和阈值函数来还有形式上的神经元表示生物电和电流的传递逻辑来模拟生物的神经网络。在Hopfield网络中的单位是二进制阈值单位，即这些单位在其状态下仅具有两个不同的值，并且该值由单位的输入是否超过其阈值决定，离散霍普菲尔德网络描述了二元（放电或不放电）神经元之间的关系在特定时间，神经网络的状态由向量描述。，它记录哪些神经元正在以二进制字输出位数据。

互动的神经元之间的单位通常采用 -1 和 1 值的单位。这些相互作用是通过赫布的关联定律(Hebb's law of association )“学习”的，因此，对于某种状态，

一旦网络被训练， 不再改变。如果神经元的新状态被引入神经网络，网络作用于神经元，使得：

其中是第个神经元的阈值（通常取为 0）。Hopfield网络能够“记住”存储在交互矩阵中的状态，因为如果出现新的状态{\displaystyle V^{s'}}受交互矩阵的影响，每个神经元都会发生变化，直到它与原始状态匹配。{\displaystyle V^{s}}

另外霍普菲尔德网络中的连接通常具有以下限制：

*，*表示为没有神经元与自己连接

*，*表示为两个神经元之间连接对等

权重对称的约束保证了能量函数在遵循激活规则时单调递减。[[12]](https://en.wikipedia.org/wiki/Hopfield_network#cite_note-12)权重不对称的网络可能会表现出一些周期性或混沌行为;然而，Hopfield发现这种行为仅限于相空间的相对较小的部分，并且不会损害网络作为内容可寻址关联记忆系统的能力。

Hopfield还为连续值对神经网络进行了建模，其中每个神经元的电输出不是二进制的，而是0到1之间的某个值。他发现这种类型的网络也能够存储和再现记忆状态。

总结来说，Hopfield 网络中的每对单元 和 都有一个由连接权重描述的连接。从这个意义上说，霍普菲尔德网络可以正式描述为一个完整的无向图。其中是一组[麦卡洛克-皮茨神经元](https://en.wikipedia.org/wiki/Artificial_neuron)([McCulloch–Pitts neurons](https://en.wikipedia.org/wiki/Artificial_neuron))，并且是将单元对链接到实际值（连接权重）的函数。

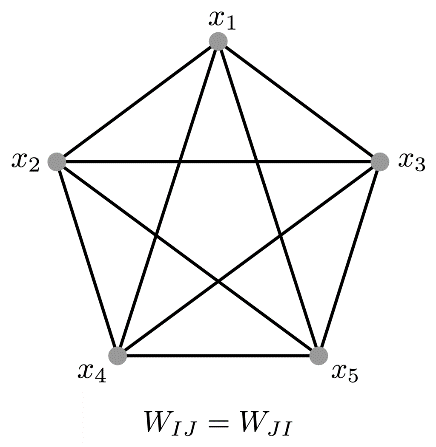


图1 一般Hopfield网络的拓扑结构

4.1.2 连续型Hopfield网络

首先，我们需要抽象地定义一下连续性Hopfield神经网络的结构。Hopfield神经网络等效为放大电子电路，并且每一个神经元等效为一个电子放大器元件。同时，每一个神经元的输入和输出，等效为电子元件的输入电压和输出电压，每一个电子元件又称神经元输出的电信号有正负值，正值代表兴奋，负值代表抑制。每一个电子元件的输入信息，包含恒定的外部电流输入，和其它电子元件的反馈连接。

根据上面的表述，我们可以具象的画出其拓扑结构(图 2)：

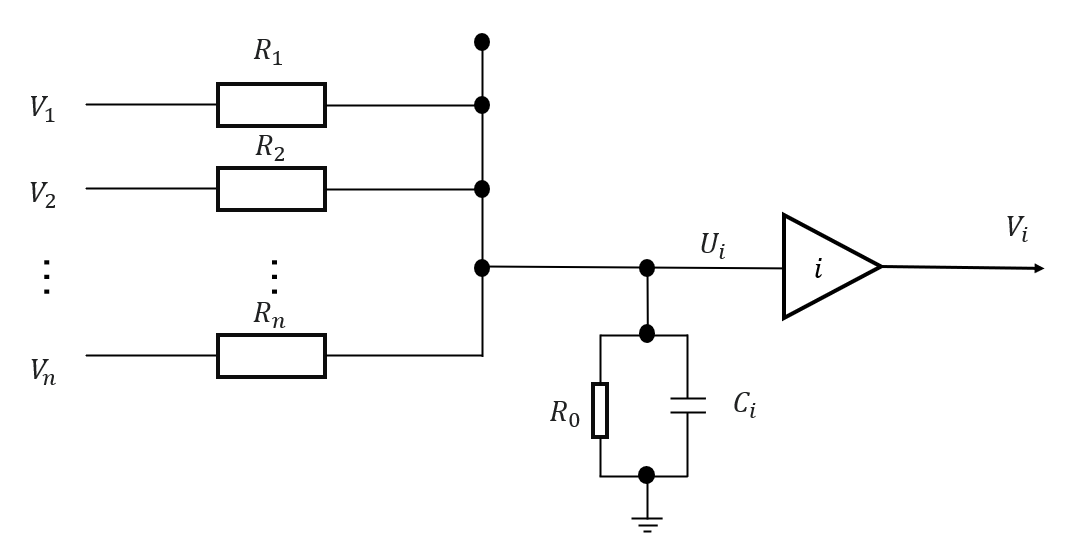


图 2

其中，表示放大电子元件的输入电压，表示输出电压，运算放大器表示第个神经元。

除了结构上的定义，我下面从功能那个上定义其是如何运行的：

设电容两端的电压为，存储的电荷量为，则

则经过电容的电流为:

根据基尔霍夫电流定律，CHNN等效电路的电流关系为:

令表示神经元之间连接的权值:

则电流关系化简为：

上式就是关于CHNN模型中输入电压和的微分方程关系，也就是连续型Hopfield网络的状态方程，其中输入电压满足非线性映射规则。

4.1.2 使用Hopfield网络解决TSP的算法

首先对于最优化问题中的最重要的问题之一就是构建目标函数，Hopfield网络中一般的能量函数定义如下：

由计算可知能量函数是单调下降的，故此定义下的网络是稳定的。

我们的案例中就34个中国城市，我们需要通过构造置换矩阵，将每一个城市对应于神经网络的每一个神经元，即案例的TSP问题转换成可用网络种神经元表示的方式来求解。

为了满足TSP的规则，我们设计置换矩阵的形式规则如下：

* 一个城市只能被访问一次，转换为矩阵表达为：矩阵每行有且只有一个1，其余元素均为0。
* 一次只能访问一个城市，转换为：矩阵每列有且只有一个1，其余元素均为0
* 共访问过个城市，翻译为：矩阵的全部元素中1的数量为。

在神经网络迭代优化过程中，每次神经元输出的状态集合只要满足上述置换矩阵的规则，则证明该组输出状态就是一个TSP问题的解，我们只要在这些解中找到最小代价的解即可

对于TSP问题中的能量函数，在这里我们的函数需要完成的两个子任务是：

* TSP的能量函数需要量化的翻译置换矩阵的规则。
* 在TSP问题中的个合法路线中，能量函数要有利于量化表示最短路线的解。

首先对每一行中的每一个城市，必须有且只有一个1，我们定义函数的第一项：

再者是每一列中的每一个城市，必须有且只有一个1，我们定义函数的第二项：

为了符合第三条规则，整个矩阵有个1，定义函数的第三项：

最后设计函数的最后一项，：

其中，是城市到城市的距离，这里使用欧几里得距离。

把三项通过相加组合,并通过化简可以得到最终的能量函数：

然后根据能量函数，我们能推导出求可行解所利用的状态函数：

根据上文描述的动态函数，我们可以写出输入状态的更新表达式：

和输出状态更新表达式：

至此，我们总结出利用Hopfield神经网络来求解TSP问题的步骤:

1. 初始化Hopfield神经网络的初值（如输入电压、迭代次数）和权值（A、D）
2. 计算个城市之间的距离矩阵
3. 初始化神经网络的输入状态
4. 利用CHNN动态方程计算输入状态的增量
5. **由一阶欧拉方法更新神经网络下一个时刻的输入状态**
6. 由sigmoid函数更新神经网络下个时刻的输出状态
7. 计算当前的能量函数E
8. 检查当前神经网络的输出状态集合，是否满足TSP置换矩阵的规则

5 实验

本此实验我使用的数据是老师指定的34个中国城市的经纬度坐标并在结果上做了可视化。

实验的环境是：

Python 3.8， matplotlib 3.5.0, Numpy 1.21.2

IDE：Pycharm和JupyterNotebook

在实验过程中，因为初始化设置的是电压的随机生成的，故程序只能生成在这个随机数下的局部最优解。但是通过多次重复实验，我可以逼近全局最优解(结果参见附录)。

经过多次重复实验，最后得到的结果是：生成的路径总距离变化在210 – 171之间，能量在5300 – 4500之间。

相比于普通的启发式算法，包括蚁群算法，粒子群算法和遗传算法等，我的方法在速度上远胜于他们，在略微损失求解精度的同时，相对于启发式算法大概分钟级的运算速度，我的算法的求解出答案的时间仅需小于10秒的时间，是一个效率上的巨大飞跃。

6 结论

Hopfield神经网络解决TSP问题的算法

**输入**：城市坐标,初始电压,步长,

迭代次数

**初始化**：电压

**计算**：

**For in** range(

*# 利用动态方程计算du*

,

*# 更新下一个时间的输入状态*

*#由sigmoid函数更新下一个时间的输出状态*

*# 计算当前网络的能量E*

*# 检查路径的合法性*

**返回：**最终最优化路径

我在这里使用了一种基于神经网络的方法，利用Hopfield神经网络近似求解二维欧几里德旅行推销员问题。对于一个固定的图尺寸，具有更好的解决能力，并且我的模型在求解质量、推理速度方面具有不错的成绩。如果由未来的工作，可以尝试将其迁移到更大的网络，使用最先进的技术，如图卷积网络和强化学习，将其纳入我的框架，以推广到大规模的问题实例，并解决TSP之外的以前未遇到的组合问题。

致谢

最后，非常感谢老师上课辛勤地为我们传授知识，我从一个更深的角度理解了机器学习的知识，包括聚类、回归、SVM还有若干启发式算法，这对我以后的学习非常有帮助。

附录



图 3 近似全局最优解

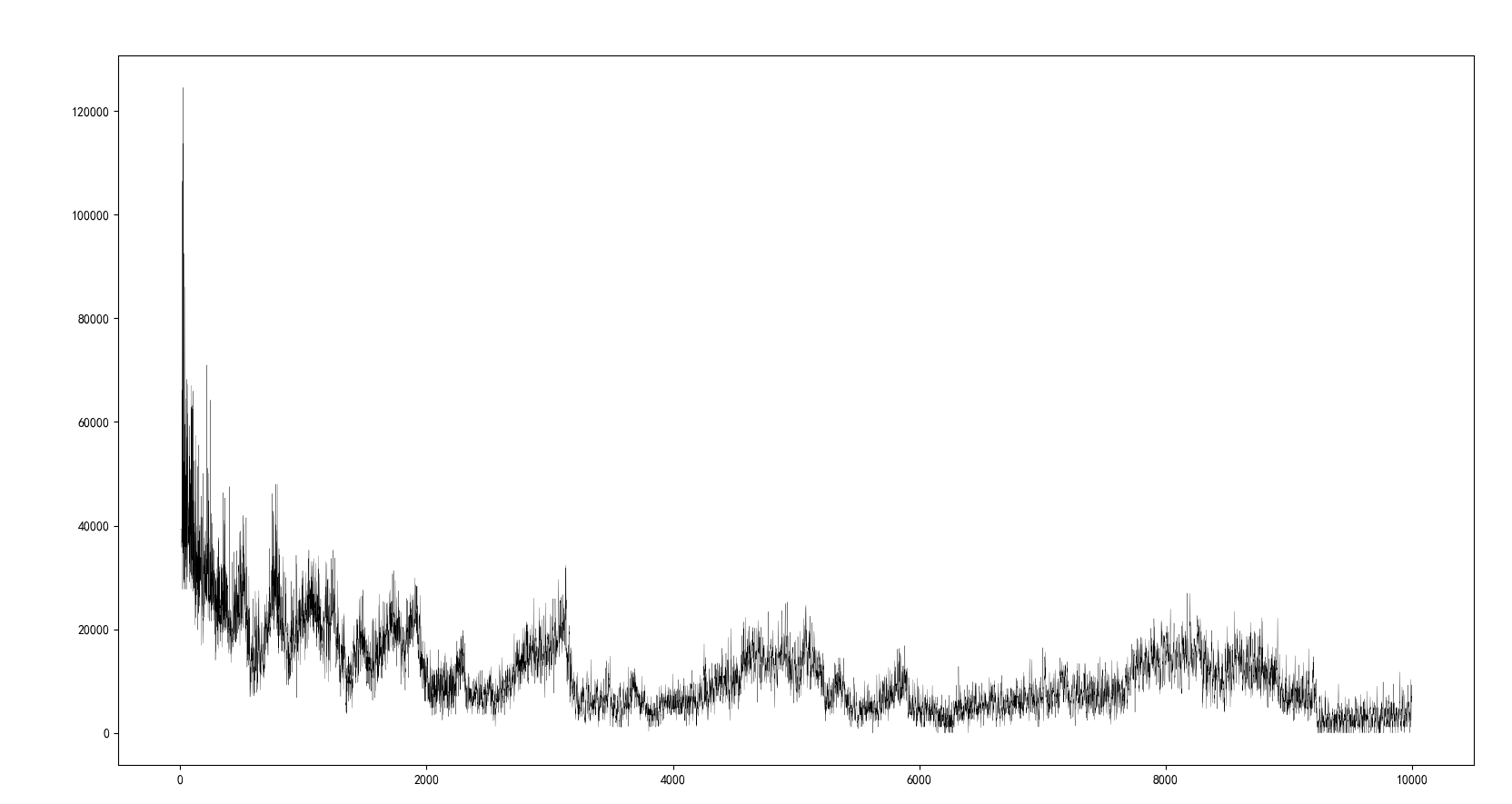


图 4 最优解时的迭代时的能量变化

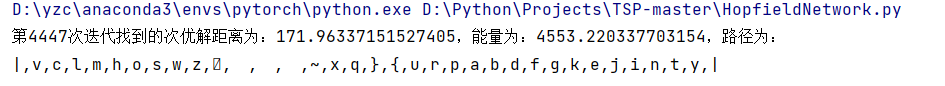


图 5 最优解时的输出结果



图 6 某次优解之1



图 7 某次优解之2

参考文献

1. [TSP问题\_百度百科 (baidu.com)](https://baike.baidu.com/item/TSP%E9%97%AE%E9%A2%98/840008)
2. [Travelling salesman problem - Wikipedia](https://en.wikipedia.org/wiki/Travelling_salesman_problem)
3. [TSP问题浅析 - 知乎 (zhihu.com)](https://zhuanlan.zhihu.com/p/105349630)
4. Christos H Papadimitriou. The euclidean travelling salesman problem is np-complete. Theoretical computer science, 4(3):237–244, 1977.
5. David L Applegate, Robert E Bixby, Vasek Chvatal, and William J Cook. The traveling salesman problem: a computational study. Princeton university press, 2006.
6. Manfred Padberg and Giovanni Rinaldi. A branch-and-cut algorithm for the resolution of large-scale symmetric traveling salesman problems. SIAM review, 33(1):60–100, 1991.
7. Kate A Smith. Neural networks for combinatorial optimization: a review of more than a decade of research. INFORMS Journal on Computing, 11(1):15–34, 1999.
8. JC Fort. Solving a combinatorial problem via self-organizing process: An application of the kohonen algorithm to the traveling salesman problem. Biological cybernetics, 59(1):33–40, 1988.
9. Bert FJ La Maire and Valeri M Mladenov. Comparison of neural networks for solving the travelling salesman problem. In 11th Symposium on Neural Network Applications in Electrical Engineering, pages 21–24. IEEE, 2012.

1. 代码同步公开在我的GitHub仓库： [↑](#footnote-ref-1)