

基于自组织映射网络的 TSP 问题的算法分析与研究

李志

(大连理工大学 电子信息与电气工程学院, 辽宁省 大连市 116024)

摘要: 随着经济社会的发展, 人们对提升生产效率的需求日益强烈, 因而旅行商问题 (Traveling Salesman Problem) 的应用越来越广泛。但是作为 NP 完全问题, 旅行商问题无法用多项式算法解决, 因此许多现代智能优化算法应运而生, 诸如遗传算法, 蚁群算法, 神经网络等, 本文对求解旅行商问题的基于自组织映射网络的算法进行了详细的分析和改进。运用 python 语言设计出相关的计算程序, 分别对 5 组 TSPLIB 中的 TSP 实例 (城市规模从 70 到 200) 进行实验, 通过传统 SOM, ORC_SOM, ORCTS_SOM (Overall-Regional Competitive & Tabu Search for SOM) 对 TSP 求解结果的对比分析。发现在中小规模的 TSP 问题中, ORCTS_SOM 算法不但更加接近于理想最优解, 而且有更好的稳定性。

关键词: 旅行商问题; 自组织映射网络; ORCTS_SOM

Algorithm Analysis and Research of TSP Problem Based on Self-Organizing Map Network

Li Zhi

(Dalian University of Technology, Dalian 116024, China. contact:Li Zhi, E-mail:201782016@mail.dlut.edu.cn)

Abstract: With the development of economy and society, people's demand for improving production efficiency is becoming stronger and stronger, so the traveling salesman problem (Traveling Salesman Problem) is more and more widely used. However, as an NP-complete problem, the traveling salesman problem cannot be solved with polynomial algorithms, so many modern intelligent optimization algorithms emerge, such as genetic algorithms, ant colony algorithms, neural networks, etc. The algorithm has been analyzed and improved in detail. Using Python language to design related calculation programs, experiment on TSP instances (city scale from 70 to 200) in 5 groups of TSPLIB respectively, through traditional SOM, ORC_SOM, ORCTS_SOM (Overall-Regional Competitive & Tabu Search for SOM) to TSP Comparative analysis of solution results. It is found that in the small and medium-scale TSP problem, the ORCTS_SOM algorithm is not only closer to the ideal optimal solution, but also has better stability.

Key words: TSP; Self-Organizing Maps; ORCTS_SOM

1 绪论

1.1 本文的研究背景

TSP(Traveling Salesman Problem)即旅行商问题, 是数学领域中著名问题之一。这个问题是这样的: 假设有一个旅行商人要拜访 n 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长度为所有路径之中的最小值。

TSP 问题的数学模型为: 设有一个城市集合 $C=\{1,2,\dots,n\}$ (C 的元素表示 n 个城市的编号), $d(V_i, V_{i+1})$ 表示城市 V_i 到城市 V_{i+1} 的距离, 寻找集合 C 的一个排列

$R(C)=\{V_1, V_2, \dots, V_n\}$ 使得遍历所有城市的总路程 $T_d = \sum_{i=1}^{n-1} d(V_i, V_{i+1}) + d(V_n, V_1)$ 取最小值。

TSP 问题不但在现实工业生产的许多领域中都有着广泛的应用, 诸如物流路径优化, PCB 钻孔行程优化, 煤气管道和石油管道的架设等, 而且在理论研究方面也有重大的意义。TSP 问题是一个典型的组合优化问题, 且是一个 NP 完全难题, 也就是说没有办法用任何已知的多项式算法来求解, 该问题是所有组合优化问题的范例, 它已经并将继续作为测试新算法的标准问题。

目前求解 TSP 问题的方法一般可以归纳为两类: 精确算法和近似算法。其中精确算法是比较传统的方法, 比如穷举

搜索法、贪心法等,这些算法当城市数目比较大时,会产生所谓的“组合爆炸”问题,在有限的时间内难以收敛。而近似算法则是以启发式算法为主,比如遗传算法,模拟退火,禁忌搜索,神经网络等,其中神经网络凭借其优秀的学习性能和较快的收敛速度近年来备受关注。

1.2 本文主要工作及结构

本文首先将介绍标准 SOM 求解 TSP 问题的算法,然后引入泛化竞争和局部渗透这两个新的策略,并结合禁忌搜索,对标准 SOM 算法进行改进,形成两种新的算法,称为基于泛化竞争和局部渗透机制的 SOM 算法(Overall-Regional Competitive SOM, ORC_SOM)。和基于泛化竞争和局部渗透机制并结合禁忌搜索的 SOM 算法(Overall-Regional Competitive & Tabu Search SOM, ORCTS_SOM)。接着使用 python 语言对以上 3 种算法进行编程实现,最后对 TSPLIB 标准库中的 5 组 TSP 实例(城市规模从 70 到 200)进行大量的实验,并且对实验结果进行详细的分析,发现使用 ORCTS_SOM 算法求得的解具有更小的平均最优偏离率(偏离最优路径长度的百分率);而且整体最优偏离率都在 5% 以内,有较好的精确性与稳定性。

2 标准 SOM 算法简介

2.1 标准 SOM 的原理

自组织映射(Self-Organizing Maps, SOM)由芬兰学者 Teuvo Kohonen 提出,是一种无监督的聚类方法,受人类大脑神经元的自组织和侧抑制现象启发。不同外部刺激引起大脑皮层各个区域的反应是不同的;类似地,自组织映射在接收外界输入的模式时,会分为不同的应对区域,各区域对输入模式有着不同的响应。这种分区对应是在训练中逐渐学到的,是一个自组织的过程。另一方面,生物神经元之间存在侧抑制现象,临近的神经元相互激励,较远的神经元相互抑制。响应最强的神经元即获胜神经元,以它为中心,会形成一个区域,距离中心越远激励越弱。

2.2 标准 SOM 如何求解 TSP 问题

自组织映射(SOM)是一种无监督学习的神经网络用来解决旅行商问题(TSP)时,二维城市坐标是神经网络的输入,城市空间位置关系是神经网络要学习的模式,一个环形的神经元结构是神经网络的输出。

2.2.1 概述

标准 SOM(Self-Organizing Maps)算法包含三个过程:竞争,合作和自适应。竞争体现在每次迭代中各神经元针对与某个随机城市的欧式距离进行竞争产生获胜神经元。我们称获胜神经元周围的区域为优胜邻域,合作就体现在优胜邻域中的神经元响应修正系数服从高斯分布,即逻辑上与优胜神经元越近的神经元的响应越积极,越远的神经元的响应越保守,也就是对生物神经元之间存在侧抑制现象的模拟,同时优胜邻域也会随着迭代次数的增加而变窄,以加强算法的局部搜索能力和保证算法的收敛性。自适应则体现为神经网络各神经元在每次迭代中就竞争合作机制以相应的学习率更新权值,实现各神经元向城市的整体靠拢,随着迭代次数的增加和学习率的衰减最终实现算法对 TSP 问题中的城市空间位置的学习,并以神经网络拓扑结构形式拟合出 TSP 问题的最优解。

SOM 神经网络迭代过程可查考图 1, (测试实例为 st70,

绘图记录间隔为 1000 次迭代)

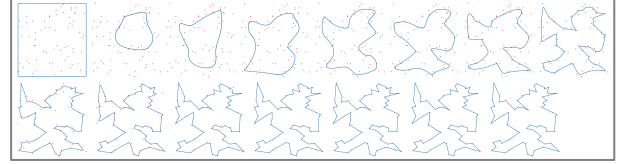


图 1 神经网络迭代过程

2.2.2 算法步骤

步骤一: 将输入的二维城市坐标(m 个城市)标归一化(x, y 坐标等比例缩小到区间(0,1)内)

步骤二: 根据输入的城市信息初始化神经网络, 神经元数目 $n=K*m$, 神经元倍率 k 一般取 $2 \sim 10$, 本文取 8。神经网络初始拓扑结构设置为一个能完全包围所有城市的矩形, 即对脚点坐标为城市坐标的极值, 大致分步如图 2。

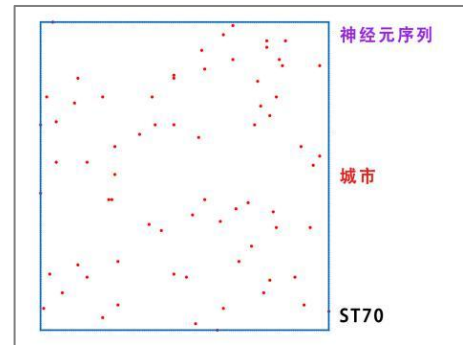


图 2 神经网络初始拓扑结构

步骤三: 在城市序列中, 随机选择一个城市 e , 计算各神经元与该城市的欧式距离, 选择与该城市距离最短的神经元记为获胜神经元 w_e 。

步骤四: 所有神经元权值按以下公式进行更新。

$$n_{t+1} = n_t + \alpha_t * g(w_e, h_t) * \Delta(e, n_t)$$

n_t : 第 t 次迭代后的神经元权值矩阵。

α_t : 第 t 次迭代后的学习率。

$g(w_e, h_t)$: 第 t 次迭代后的优胜邻域神经元响应修正系数,

本文设置为以获胜神经元 w_e 为中心, 标准差为 h_t 的高斯分布概率函数, 即

$$g(w_e, h_t) = \exp\left(-\frac{d(n, w_e)^2}{2 * h_t^2}\right)$$

$d(n, w_e)$: 各神经元与获胜神经元的逻辑距离(把神经网络想像成一个封闭的圆, 神经元按标号顺序均匀分布在圆上, 彼此间隔为单位一, 假设神经元数目为 100, 譬如标号 12 与标号 52 的神经元逻辑距离为 50, 而标号 12 与标号 92 的神经元逻辑距离为 20)

$\Delta(e, n_t)$: 第 t 次迭代后各神经元与城市 e 的欧式距离矩阵, 该过程形象表示见图 3:

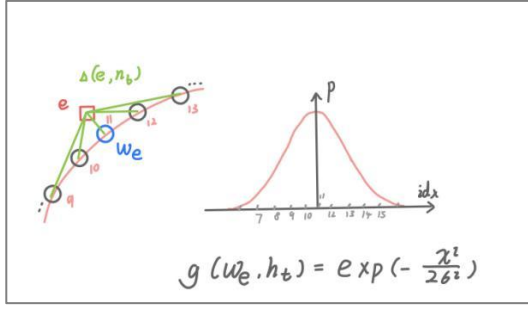


图 3 优胜邻域神经元响应修正系数分布

步骤五：学习率 α_t & 优胜邻域标准差 h_t 衰减

$$\alpha_{t+1} = \phi_{\alpha} * \alpha_t \quad \phi_{\alpha} = 0.99997$$

$$h_{t+1} = \phi_h * h_t \quad \phi_h = 0.9997$$

步骤六：收敛检测,出现以下 3 种情况认为算法已经收敛,前往步骤七。

1. 学习率收敛: $\alpha_t < 0.001$, 学习率过低, 神经网络几乎不再优化。
 2. 优胜领域标准差收敛: $h_t < \frac{k}{10}$, 优胜领域神经元响应修正系数过低, 除获胜神经元之外, 其他神经元权值几乎不产生改变。
 3. 迭代次数收敛: $i = 100000$, 控制算法完成时间。
- 如果不符合以上 3 种情况, 即算法未收敛, 重复步骤三至步骤六。

步骤七：此时算法已收敛, 即自组织映射神经网络已经完成对输入的 TSP 问题中的城市空间分布的学习, 神经网络的拓扑结构可以近似表征实际的路径选择。可以将城市序列根据各城市对应的最终获胜神经元的标号进行重新排列, 即可得到算法最终收敛的解。神经网络的拓扑结构到实际的路径选择的映射见图 4。

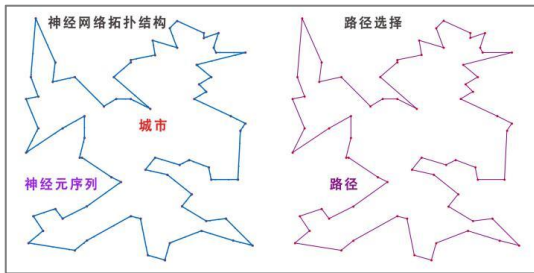


图 4 神经网络的拓扑结构到实际的路径选择的映射

2.2.3 参数选择及说明

1. 学习率初值 α_0 : 建议在区间[0.1,0.8]内, 本文设置为 0.3
2. 优胜领域标准差初值 h_0 : 由于优胜领域神经元响应修正

系数满足高斯分布, 即 $g(w_e, h_t) = \exp(-\frac{d(n, w_e)^2}{2h_t^2})$, 标准差 h_t

近似表征了优胜邻域的宽窄程度, 而在算法初期, 我们希望考虑全局, 所以需要优胜邻域尽量宽一些, 即距离获胜神经

元最远的神经元也参与城市空间位置的学习, 故而有以下算式:

$$f(\frac{n}{2}) = \exp(-\frac{n^2}{8 * h_0^2}) > p$$

取 $p = 10^{-5}$, 可得 $h_0 > 0.104 * n$, 故选择 $h_0 = \frac{n}{10}$ 。

3. 优胜领域标准差收敛值 h_f : 随着迭代次数的增加, 优胜

领域标准差 h_t 不断衰减, 即优胜邻域不断变窄, 参与城市空间位置的学习的神经元越来越少, 若某次迭代时, 仅有获胜神经元及其周围 k (神经元倍率) 个神经元参与学习, 而其他神经元权值几乎不产生改变, 则认为算法优胜领域标准差已收敛, 故而有以下算式

$$f(\frac{k}{2}) = \exp(-\frac{k^2}{8 * h_f^2}) > p$$

取 $p = 10^{-5}$, 可得 $h_f < 0.104 * k$, 故选择 $h_f = \frac{k}{10}$ 。

3 标准 SOM 算法的改进

3.1 ORC_SOM 算法

3.1.1 概述

标准 SOM 算法虽然通过学习率 α_t 和优胜领域标准差 h_t 随迭代次数增加而衰减来实现算法的收敛, 但是该算法在全局寻优和局部搜索方面能力比较薄弱, 只是单一的随着迭代次数的增加而收缩优胜领域的宽度, 一定程度的实现了对算法后期的局部搜索能力的提升, 而并没有针对性, 也没有在前期强化全局寻优, 故而引入两个新的学习准则——泛化竞争和局部渗透, 并结合这两个准则提出一个标准 SOM 的改进算法——ORC_SOM 算法。

所谓泛化竞争和局部渗透就是, 在标准 SOM 算法在对所有神经元权值进行更新时, 不仅考虑学习率和优胜邻域神经元修正系数, 同时引入泛化竞争和局部渗透算子 z , 通过计算获胜神经元 w_e 与随机输入城市 e 的距离 $d(e, w_e)$, 设置一个半径门限 λ_t , λ_t 随迭代次数增加而衰减, 若 $d(e, w_e) > \lambda_t$, 者说明所有神经元距离输入城市多足够远, 神经网络对于输入城市周围区域城市空间分布的学习才刚开始, 故而更应该强调全局寻优, 即根据实际距离降低此次迭代的学习率, 泛化竞争和局部渗透算子 z 取 0 到 1 之间的数。而与之相反, 若 $d(e, w_e) < \lambda_t$, 则表明神经网络对于输入城市周围区域城市空间分布的已经完成了部分认知, 故而更应该强调局部搜索, 即根据实际距离提高此次迭代的学习率, 泛化竞争和局部渗透算子 z 取大于 1 的数。具体的泛化竞争和局部渗透算子 z 可以使用以下算式求得。

$$z(d(e, w_e), \lambda_t) = \exp(-\frac{d(e, w_e)^2}{2 * \lambda_t^2} + \frac{1}{2})$$

易知该算子有如下性质:

$$z(d(e, w_e), \lambda_t) \begin{cases} \leq 1, & d(e, w_e) \geq \lambda_t, \text{ 泛化竞争} \\ > 1, & d(e, w_e) < \lambda_t, \text{ 局部渗透} \end{cases}$$

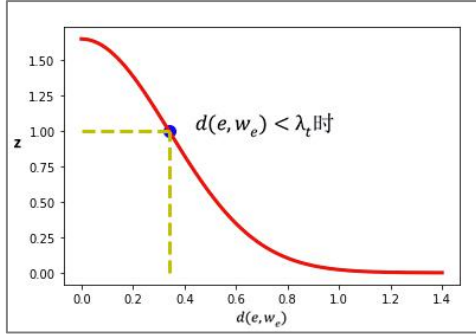


图 5 泛化竞争和局部渗透算子 z 性质

其中半径门限按照如下方式进行衰减:

$$\lambda_t = \lambda_0 * (1 - \exp(-\frac{t}{\tau})), t > 0$$

3.1.2 算法步骤

步骤一: 将输入的二维城市坐标(m 个城市)标归一化 (x, y 坐标等比例缩小到区间(0,1)内)

步骤二: 根据输入的城市信息初始化神经网络, 神经元数目 $n = K * m$, k 一般取 $2 \sim 10$, 本文取 8。神经网络初始拓扑结构设置为一个能完全包围所有城市的矩形, 即对脚点坐标为城市坐标的极值。

步骤三: 在城市序列中, 随机选择出一个城市 e , 计算各神经元与该城市的欧式距离, 选择与该城市距离最短的神经元记为获胜神经元 w_e 。

步骤四: 所有神经元权值按以下公式进行更新。

$$n_{t+1} = n_t + \alpha_t * z(d(e, w_e), \lambda_t) * g(w_e, h_t) * \Delta(e, n_t)$$

n_t : 第 t 次迭代后的神经元权值矩阵。

α_t : 第 t 次迭代后的学习率。

$z(d(e, w_e), \lambda_t)$: 第 t 次迭代后的泛化竞争和局部渗透算子,

$$z(d(e, w_e), \lambda_t) = \exp(-\frac{d(e, w_e)^2}{2 * \lambda_t^2} + \frac{1}{2}).$$

即 $d(e, w_e) \geq \lambda_t$ 时, $z(d(e, w_e), \lambda_t) \leq 1$ 且与 $d(e, w_e)$ 成反比, 算法强调泛化竞争。

而 $d(e, w_e) < \lambda_t$ 时, $z(d(e, w_e), \lambda_t) > 1$ 且与 $d(e, w_e)$ 成反比, 算法强调局部渗透。

$g(w_e, h_t)$: 第 t 次迭代后的优胜邻域神经元响应修正系数,

本文设置为以获胜神经元 w_e 为中心, 标准差为 h_t 的高斯分布概率函数, 即

$$g(w_e, h_t) = \exp(-\frac{d(n, w_e)^2}{2 * h_t^2})$$

$d(n, w_e)$: 各神经元与获胜神经元的逻辑距离 (把神经网络想像成一个封闭的圆, 神经元按标号顺序均匀分布在圆上, 彼此间隔为单位一, 假设神经元数目为 100, 譬如标号 12 与标号 52 的神经元逻辑距离为 50, 而标号 12 与标号 92 的神经元逻辑距离为 20)

$\Delta(e, n_t)$: 第 t 次迭代后各神经元与城市 e 的欧式距离矩阵。

步骤五: 学习率 α_t , 优胜邻域标准差 h_t , 半径门限 λ_t 衰减

$$\alpha_{t+1} = \phi_\alpha * \alpha_t \quad \phi_\alpha = 0.99997$$

$$h_{t+1} = \phi_h * h_t \quad \phi_h = 0.9997$$

$$\lambda_t = \lambda_0 * (1 - \exp(-\frac{t}{\tau}))$$

步骤六: 收敛检测, 出现以下 3 种情况认为算法已经收敛, 前往步骤七

1. 学习率收敛: $\alpha_t < 0.001$, 学习率过低, 神经网络几乎不再优化。

2. 优胜邻域标准差收敛: $h_t < \frac{k}{10}$, 优胜邻域神经元响应修正系数过低, 除获胜神经元之外, 其他神经元权值几乎不产生改变。

3. 迭代次数收敛: $i = 100000$, 控制算法完成时间。

如果不符合以上 3 种情况, 即算法未收敛, 重复步骤三至步骤六。

步骤七: 此时算法已收敛, 即自组织映射神经网络已经完成对输入的 TSP 问题中的城市空间分布的学习, 神经网络的拓扑结构可以近似表征实际的路径选择。可以将城市序列根据各城市对应的最终获胜神经元的标号进行重新排列, 即可得到算法最终收敛的解。

3.1.3 参数选择及说明

1. 初始半径门限 λ_0 : 由半径门限衰减公式 $\lambda_t = \lambda_0 * (1 - \exp(-\frac{t}{\tau}))$ 可知半径门限是单调递减的, 实际上初始半径

门限 λ_0 表征着算法最终局部与全局的判断标准。由于初始神经网络的拓扑结构为一个完全包围所有城市的矩形, 所以从实际意义出发, 半径门限应该与矩形对角线长度 D 有关,

经过对比实验, 本文选取为 $\lambda_0 = \frac{D}{4}$ 。

2. 半径门限衰减时间常数 τ : 实验表明算法大致在迭代 12000 次左右收敛, 为了匹配算法现有特性, 选取 $\tau = 3000$, 使得半径门限大概在 12000 次基本收敛到初始半径门限 λ_0 。

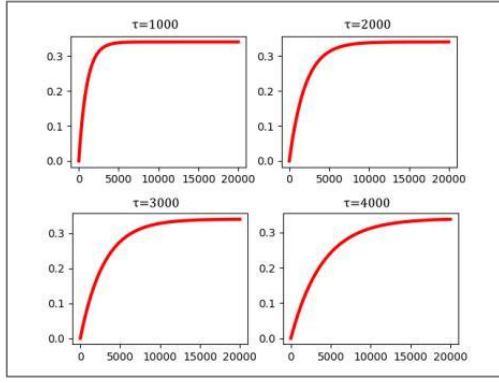


图 6 半径门限衰减时间常数 τ 的性质

3. 其他参数选择与标准 SOM 算法相同。

3.2 ORCTS_SOM 算法

3.2.1 概述

显然由于引入了泛化竞争和局部渗透算子，ORC_SOM 的全局寻优和局部搜索能力都有所提升，但是注意到算法施加给自组织映射神经网络的刺激是随机的，即每次选择的城市 e 是随机的，因此产生的获胜神经元也受随机性影响，可能连续几次集中在一边区域对神经网络持续刺激，影响了全局寻优，这是我们在算法前期不希望看到的，故而为了进一步提升算法的全局寻优能力，我们采取禁忌搜索的方式选取城市 e 以刺激神经网络。算法开始时所有城市禁忌长度设置为 0，每次迭代在城市序列中随机选取一个当前禁忌长度为 0 的城市 e ，随即更新禁忌表，城市 e 的禁忌长度更新为初始禁忌长度 T_0 ，其余城市如果其禁忌长度为 0 则保持不变，否则减 1。即如下算式：

$$(T_{t+1})_i = \begin{cases} T_0, & i = e \\ T_t - 1, & T_t > 0 \\ 0, & T_t = 0 \end{cases}$$

禁忌长度按照如下方式进行衰减：

$$T_{t+1} = \text{int}(\phi_T * T_t) \quad \phi_T = 0.97$$

3.2.2 算法步骤

步骤一：将输入的二维城市坐(m 个城市)标归一化 (x,y 坐标等比例缩小到区间(0,1)内)

步骤二：根据输入的城市信息初始化神经网络，神经元数目 $n=K*m$ ， k 一般取 2~10，本文取 8。神经网络初始拓扑结构设置为一个能完全包围所有城市的矩形，即对脚点坐标为城市坐标的极值。

步骤三：在城市序列中，随机选择一个禁忌长度为 0 城市 e ，将计算各神经元与该城市的欧式距离，选择与该城市距离最短的神经元记为获胜神经元 w_e 。更新禁忌表，城市 e 的禁忌长度更新为初始禁忌长度 T_0 ，其余城市如果其禁忌长度为 0 则保持不变，否则减 1。

步骤四：所有神经元权值按以下公式进行更新。

$$n_{t+1} = n_t + \alpha_t * z(d(e, w_e), \lambda_t) * g(w_e, h_t) * \Delta(e, n_t)$$

n_t : 第 t 次迭代后的神经元权值矩阵。

α_t : 第 t 次迭代后的学习率。

$z(d(e, w_e), \lambda_t)$: 第 t 次迭代后的泛化竞争和局部渗透算子，

$$z(d(e, w_e), \lambda_t) = \exp\left(-\frac{d(e, w_e)^2}{2 * \lambda_t^2} + \frac{1}{2}\right)$$

即 $d(e, w_e) \geq \lambda_t$ 时， $z(d(e, w_e), \lambda_t) \leq 1$ 且与 $d(e, w_e)$ 成反比，算法强调泛化竞争。

而 $d(e, w_e) < \lambda_t$ 时， $z(d(e, w_e), \lambda_t) > 1$ 且与 $d(e, w_e)$ 成反比，算法强调局部渗透。

$g(w_e, h_t)$: 第 t 次迭代后的优胜邻域神经元响应修正系数，

本文设置为以获胜神经元 w_e 为中心，标准差为 h_t 的高斯分布

概率函数，即

$$g(w_e, h_t) = \exp\left(-\frac{d(n, w_e)^2}{2 * h_t^2}\right)$$

$d(n, w_e)$: 各神经元与获胜神经元的逻辑距离（把神经网络想像成一个封闭的圆，神经元按标号顺序均匀分布在圆上，彼此间隔为单位一，假设神经元数目为 100，譬如标号 12 与标号 52 的神经元逻辑距离为 50，而标号 12 与标号 92 的神经元逻辑距离为 20)

$\Delta(e, n_t)$: 第 t 次迭代后各神经元与城市 e 的欧式距离矩阵。

步骤五：禁忌长度 T_t ，学习率 α_t ，优胜邻域标准差 h_t ，半径门限 λ_t 衰减

$$T_{t+1} = \text{int}(\phi_T * T_t) \quad \phi_T = 0.97$$

$$\alpha_{t+1} = \phi_\alpha * \alpha_t \quad \phi_\alpha = 0.99997$$

$$h_{t+1} = \phi_h * h_t \quad \phi_h = 0.9997$$

$$\lambda_t = \lambda_0 * (1 - \exp(-\frac{t}{\tau}))$$

步骤六：收敛检测，出现以下 3 种情况认为算法已经收敛，前往步骤七

1. 学习率收敛: $\alpha_t < 0.001$ ，学习率过低，神经网络几乎不再优化。

2. 优胜邻域标准差收敛: $h_t < \frac{k}{10}$ ，优胜邻域神经元响应修正系数过低，除获胜神经元之外，其他神经元权值几乎不产生改变。

3. 迭代次数收敛: $i = 100000$ ，控制算法完成时间。

如果不符合以上 3 种情况，即算法未收敛，重复步骤三至步骤六。

步骤七：此时算法已收敛，即自组织映射神经网络已经完成对输入的 TSP 问题中的城市空间分布的学习，神经网络的

拓扑结构可以近似表征实际的路径选择。可以将城市序列根据各城市对应的最终获胜神经元的标号进行重新排列,即可得到算法最终收敛的解。

3.2.3 参数选择及说明

1. 初始禁忌长度 T_0 : 由于初始神经网络的拓扑结构为一个完全包围所有城市的矩形,且每个边上分布钝神经元数目相等,可以认为在算法初期神经网络大致从4个方向通过输入城市的刺激来学习城市空间位置关系,所以为提升算法前期的全局寻优性能,可以取 $T_0 = \frac{m}{4}$ 。
2. 其他参数选择与 ORC_SOM 算法相同。

4 TSP 问题求解实验与比较

4.1 最优偏移率

在开始实验之前,本文首先引入一个参数:最优路径偏移率 δ ,这样可以用来更直观地评价各个算法之间性能的优劣。

因为不论通过任何改进 SOM 算法所得的 TSP 问题的最优路径都只能是无限接近于理想最优路径,都是真实城市在神经元上的一一映射而得到的虚拟城市的最优路径,与理想最优路径有一定差距,所以本文利用与理想最优路径之间的差距来描述各个算法的优劣,于是就提出了最优路径偏移率 δ 。

δ 是由通过各种求解 TSP 的算法所得到的最优路径和理想状态下的最优路径之差,再与理想最优路径比率,从而得到的 δ ,可以很直观的得到各个求解 TSP 问题的算法间的性能比较。

最优路径偏移率 δ 可以出下面公式得到:

$$\delta = \frac{l - l_{opt}}{l_{opt}} \times 100\%$$

式中 l 是通过 TSP 算法求得的最优路径长度, l_{opt} 是理性状态下的最优路径长度。由此可以看出 δ 越小,说明利用该算法求解 TSP 数据所得的最优路径越接近理想的最优路径,即在这种情况下,利用该算法求的 TSP 问题的最优路径精度最好。

4.2 TSP 数据实验比较分析

本文通过对 TSPLIB 标准库取得 5 组 TSP 实例(城市数目从 70 到 200)进行实验,每个算法对各问题都进行 5 次运算,最优路径偏移率取 5 次平均值,作为分析各算法性能的标准。

实验硬件环境:

cpu: 1.8 GHz 四核 Intel Core i5,

内存: 8 GB 2400 MHz DDR4

实验软件环境:

macOS 10.15.4, Visual Studio Code, python 3.7.6

实验结果见表 1,表中加粗表红的表示为本组实验数据中性能较好的。

表 1 最优路径偏移率对比分析

TSP	理想最优路径 长度	SOM δ	ORC_SOM δ	ORCTS_SOM δ
st70	675	3.03%	2.91%	1.97%
pr76	108159	2.19%	3.65%	2.73%
rat99	1211	6.36%	4.91%	4.51%
ch130	6110	2.47%	3.07%	4.07%
kroA200	29368	4.24%	4.14%	3.94%
平均值		3.65%	3.74%	3.43%

由实验结果可知,在求解 5 种 TSP 实例的结果中,ORCTS_SOM 的最优路径偏移率 δ 在 5 次中有 3 次最优,且平均最优路径偏移率优于其他两种算法,同时整体最优路径偏移率都在 5%以内,而标准 SOM 算法在求解 rat99 时最优路径偏移率为 6.36%,超过 5%。所以可以说,通过引入泛化竞争和局部渗透算子 z 和采用禁忌搜索方式生成城市 e 来刺激神经网络,ORCTS_SOM 在求解中小规模 TSP 问题的计算精度和稳定性方面相对于标准 SOM 算法都有一定的优势

5 结语

在信息技术飞速发展的今天,人工智能算法被应用于人类生产实践和日常生活的各个领域。自组织映射网络(Self-Organizing Maps, SOM)作为人工智能算法的一种,在解决传统旅行商问题方面相对于其他启发式算法在求解时间和精度方面都有一定的优势。在本文中,对标准 SOM 算法做了基本的介绍,然后根据算法的原理分析对标准 SOM 算法的不足,在强化全局寻优和局部搜索方面对进行改进,形成两种新的算法,并使用 python 语言编写相关程序,最后进行相关实验探究各算法的性能优劣。

主要工作包括以下几个方面:

1. 首先介绍了本文的研究背景。
2. 其次介绍了标准 SOM 算法的原理,以及如何用该算法求解 TSP 问题,给出来程序步骤并分析了相关参数的选择方法。
3. 接着对标准 SOM 的不足进行了讨论,提出两种改进算法,其一 ORC_SOM (Overall-Regional Competitive for SOM),相对于标准 SOM 引入了泛化竞争和局部渗透算子 z ,实现针对获胜神经元与城市距离和半径门限关系来修正学习率,以实现在神经网络学习初期强化全局寻优,在学习后期强化局部搜索;其二 ORCTS_SOM (Overall-Regional Competitive & Tabu Search for SOM),在 ORC_SOM 算法的基础上,以禁忌搜索的方式在每次迭代中选择城市 e 来刺激神经网络,进一步加强算法初期的全局寻优能力。
4. 在实验部分,来自 TSPLIB 标准库的 5 组 TSP 实例进行了相关实验,通过标准 SOM 算法及其两种改进算法求解这 5

组 TSP 实例的实验结果对比分析,发现 ORCTS_SOM 在求解中小规模 TSP 问题的计算精度和稳定性方面都有一定的优势。

当然,由于本人知识水平的限制,本文的工作也还存在着许多不足之处,比如:对 ORCTS_SOM 算法参数的选择仅是凭借实验的经验,而没有上升到理论推导。而且目前参数的选择对实验结果的优劣有一定的影响,此外本文所有代码和数据已开源于 Github 上,更详细的内容可浏览此存储库 <https://github.com/willwoodson/som-tsp>。

参考文献 (References)

[1]Wen-bao Qiao.Massive Parallel Self-organizing Map and 2-Opt on GPU to Large Scale TSP[J].International

Work-Conference on Artificial and Natural Neural Networks,2017:471-482

[2]Carsten Mueller.Hybrid Approach for TSP Based on Neural Networks and Ant Colony Optimization[J].IEEE Symposium Series on Computational Intelligence,2015:1431-1435

[3]刘少杰.TSP 问题的神经网络求解实验与比较分析[D].西安电子科技大学,2010.

[3] Liu Shaojie. Experiment and comparative analysis of neural network solution to TSP problem [D]. Xidian University, 2010.

[4]郭庆田.基于泛化竞争和局部渗透机制的自组织网[D].西安电子科技大学,2009.

[4] Guo Qingtian. Self-organizing network based on generalized competition and local penetration mechanism[D].Xidian University, 2009.