武汉大学计算机学院
本科生课程设计报告

基于分块及搜索的蚁群算法的改进
——旅行商问题

专　业　名　称: 计算机科学与技术

课　程　名　称: 智能计算

指　导　教　师: 王峰　教授

学　生　学　号: 2021302191650

学　生　姓　名: 于丰林

二〇二三年十二月

郑 重 声 明

本人呈交的设计报告，是在指导老师的指导下，独立进行实验工作所取得的成果，所有数据、图片资料真实可靠。尽我所知，除文中已经注明引用的内容外，本设计报告不包含他人享有著作权的内容。对本设计报告做出贡献的其他个人和集体，均已在文中以明确的方式标明。本设计报告的知识产权归属于培养单位。

本人签名: _____ 日期: _____

目　录

1 研究现状

1.1 蚁群算法

蚁群算法（Ant Colony Optimization, ACO）是一种模拟自然界蚂蚁觅食行为的启发式算法，由 Marco Dorigo 于 1990 年代初期提出。这种算法的核心思想基于蚂蚁在寻找食物源和返回蚁巢过程中释放信息素，从而形成一条最短路径。在自然界中，某些蚂蚁种类最初会随机漫游，发现食物后会在返回蚁巢的过程中留下信息素轨迹。其他蚁群成员发现这样的路径后，可能不会继续随机移动，而是选择跟随这条轨迹，并在最终找到食物时加强这条路径的信息素。

随着时间的推移，信息素轨迹会逐渐蒸发，减弱其吸引力。蚁群花费更长的时间在一条路径上往返，意味着信息素有更多的时间蒸发。相比之下，较短的路径会被更频繁地经过，因此在较短路径上的信息素密度会变得更高。信息素的蒸发还有助于避免算法收敛于局部最优解。如果没有信息素蒸发，最初由蚁群选择的路径可能过于吸引后续蚁群，限制了对解空间的探索。

蚁群算法通过模拟这一行为，用于解决诸如旅行商问题（TSP）、图的最短路径问题等复杂的组合优化问题。[3] 在计算机科学和运筹学中，ACO 是一种用于解决可以归结为在图中找到良好路径的计算问题的概率技术。人工蚂蚁代表由真实蚂蚁行为启发的多智能体方法。其中，生物蚂蚁的信息素基通信通常是主要的模仿范式。人工蚂蚁（例如模拟代理）通过在代表所有可能解决方案的参数空间中移动来寻找最优解。真实的蚂蚁在探索环境时会放下信息素，引导彼此找到资源，模拟的“蚂蚁”同样会记录它们的位置和解决方案的质量，以便在后续的模拟迭代中让更多的蚂蚁找到更好的解决方案。

1.2 蚁群算法的改进

对蚁群算法的改进集中在两个主要方面：

首先，研究者们致力于将蚁群算法与其他元启发式算法结合，例如模拟退火、

遗传算法、禁忌搜索和粒子群优化。[4] 这种结合的目的是利用不同算法的优势来改善蚁群算法的性能。例如，将蚁群算法与局部搜索算法相结合已成为许多涉及图的优化任务的首选方法，比如车辆路线规划和互联网路由。这种结合策略使算法能够更有效地解决涉及图结构的复杂问题。

其次，蚁群算法的改进也集中在其应用上，特别是在那些需要适应特定问题需求的场景中。例如，在蚁群算法中，可以采用修改后的全局信息素更新规则，允许每次迭代中的最佳蚂蚁来更新路径。此外，通过控制每条路径上的最大和最小信息素量，限制信息素量的范围，以避免搜索算法停滞，增强了算法的搜索能力和适应性。此外，连续正交蚁群（COAC）算法中的信息素存储机制使蚂蚁能够更有效地协作搜索解决方案。这种正交设计方法和自适应半径调整方法也可以扩展到其他优化算法中，以解决实际问题。

这些改进通常旨在解决蚁群算法在初始搜索阶段的盲目性和收敛速度慢等问题，从而提高算法的整体性能和效率。通过这些策略，蚁群算法能够更好地适应不同的应用场景，解决更广泛的优化问题。

2 近年来改进的算法

2.1 基本的蚁群算法

基本的蚁群算法在随机的城市中生成若干个蚂蚁，这些蚂蚁寻找从一个点开始，经过所有其他点各一次并返回起点的路径。对于下一个点的选择概率基于信息素强度（τ_{uv}）和路径的相对吸引力（η_{uv}），其中 α 和 β 是控制这两个因素重要性的参数。这一概率由以下公式表示：

$$P = \frac{\tau_{uv}^{\alpha} \times \eta_{uv}^{\beta}}{\sum_{i \in v} \tau_{ui}^{\alpha} \times \eta_{ui}^{\beta}}$$

每个蚂蚁根据当前位置可用的边的长度以及相应的信息素水平来选择下一条边。在算法的每一步中，蚂蚁从一种状态移动到另一种状态，代表一个更完整的中间解决方案。

信息素通常在所有蚂蚁完成其解决方案后更新，增加或减少与“好”或“坏”解决方案相对应的路径的信息素水平。全局信息素更新规则可表示为：

$$\tau_{xy} \leftarrow (1-\rho)\tau_{xy} + \sum_{k}^{m} \Delta\tau_{xy}^{k}$$

其中，τ_{xy} 是状态转换的信息素量，ρ 是信息素蒸发系数，m 是蚂蚁的数量，$\Delta\tau_{xy}^{k}$ 是第 k 只蚂蚁存放的信息素量。

通过这种方式进行迭代，最终在优秀的路径上会累积信息素，我们从蚂蚁种群中选择得到的最优路径。

2.2 基于信息素扩散的改进蚁群算法

在自然界中，蚂蚁留下的信息素确实可能会在相近的路径上扩散，这种现象可以被用于改进蚁群算法。在基于信息素扩散的改进蚁群算法中，模拟蚂蚁在路径选择时考虑周围路径上的信息素浓度。[1] 这种方法通过计算与当前路径相近的其他路径之间的相似度，如通过余弦相似度计算两条共顶点路径的相关度，来模拟信息素的扩散效应。

在这种改进方法中，当计算出一对路径（比如 (i,j) 和 (i,k)）的相关度后，算法会将 (i,j) 上的信息素的一部分扩散到 (i,k) 上。这种扩散是通过将相关度与一个较小的系数相乘来实现的。这样，算法不仅考虑了蚂蚁当前所在路径上的信息素浓度，而且还考虑了周围路径上的信息素，从而增加了算法在寻找最优解时的全局搜索能力和效率。

2.3 朴素基于信息素扩散的改进蚁群算法的不足

在深入分析基于信息素扩散的改进蚁群算法时，我们认识到了该方法的几个关键局限性。这些局限性不仅影响了算法的性能，而且在某些情况下限制了其应用的普遍性。

首先，当面对节点数量较多的图时，观察到节点倾向于形成簇状结构，即形成多个分散的群体或团簇。在这种情况下，朴素的信息素扩散算法主要在这些团簇内部进行信息素的扩散，而不涉及团簇之间的信息素交换。这种局限性可能与实际蚂蚁的行为模式相一致，但在计算上引入了显著的复杂性。具体来说，计算不同路径之间相关度的时间复杂度高达 $\mathcal{O}(n^3)$，成为算法性能的主要瓶颈。[2] 此外，对于大量的路径组合 (i,j,k)，其在信息素更新过程中并无显著贡献，导致算法在计算上的效率低下，浪费了大量的计算资源。

其次，对于不同长度的路径应用单一的蚁群算法策略可能并不理想。尽管算法通过衰减机制对长距离路径的信息素进行调整，但这种衰减比例——作为算法的一个关键超参数——会随着图的结构变化而发生显著改变。这意味着，基于单一的 Q/len 参数进行信息素更新可能不足以捕捉不同结构图中路径的真实特性，从而影响算法在各类问题上的通用性和效率。

总结来看，尽管基于信息素扩散的改进蚁群算法在某些方面模拟了自然界中蚂蚁的行为并展示出一定的优势，但其在处理大规模图结构和不同路径长度时所面临的挑战提示我们在未来的研究中需要进一步优化和调整算法。

3 基于分块及搜索的蚁群算法的改进

为了克服朴素基于信息素扩散的改进蚁群算法所存在的局限性，本研究提出了一种基于分块及搜索的改进策略。该策略旨在通过对原图的点集进行有效划分，以及对内部及点集间路径的优化处理，提高算法的整体性能和求解精度。

3.1 点集的分块

在改进策略的第一阶段，我们采用块划分的方法，将原图中的所有点分为若干个集合。这种分块策略可以通过多种方式实现，具体包括:

1. 为了提高蚁群算法的求解速度及收敛效率，可以将所有点分成约 $7 \sim 20$ 个点集。这种方法旨在通过减少每个子集内部的点数来加快搜索速度，并通过分散处理来加速算法的收敛。
2. 为了增强算法的求解精度，我们还可以设置一个比例因子 α。在此方法中，每次加入新点前，会判断该点与已有点集几何中心的距离，若超过 α 值设定的最远距离，则不加入该点集。如果在一次扫描中没有新点加入，我们将选择未加入任何点集的第一个点来创建新的集合。

在本研究中，我们采用了临近点选取原则，将所有点分成大约 $\sqrt{n}$ 个集合。这种方法在实验中表现出良好的效果，兼顾了速度和精度两方面的需求。

3.2 点集内部路径

对于每个分块中的点集，由于其中的点与点之间的距离普遍处于相同的数量级，我们采用朴素基于信息素扩散的改进蚁群算法来处理点集内部的路径。这种方法利用信息素扩散原理，能有效地在点集内部找到优化的路径。

3.3 点集与点集之间的路径

针对不同点集之间的路径问题，由于各点集间距离可能有显著差异，我们采用枚举搜索法进行处理。通过这种暴力求解方法，我们能够确保找到点集间的最优路径。虽然这种方法可能在计算上更为繁琐，但它确保了在点集间路径优化方面的准确性和有效性。

综上所述，这种基于分块及搜索的改进策略，通过对点集的有效划分和路径的优化处理，提升了蚁群算法在复杂图结构中的应用效果，展现了在处理大规模优化问题时的潜在优势。

4 仿真实验设计

在本研究中，我们旨在评估基于分块及搜索的改进蚁群信息素扩散算法在解决旅行商问题（TSP）上的效果。为此，我们进行了一系列的实验，并对实验结果进行了详细分析。

4.1 实验数据的选取

实验采用了与朴素基于信息素扩散的改进蚁群算法相同的数据集 `dantzig42.tsp`，该数据集详见附录文件。此数据集被广泛认为是评估 TSP 解决方案效率的标准测试。通过在此数据集上进行实验，我们能够准确评估改进算法相对于传统方法的性能提升。

4.2 点集的划分

在实验过程中，首先对 `dantzig42.tsp` 数据集进行了点集的划分。此过程的结果如图 4.1 所示。从图中可以明显看出，不同区域的点被有效地划分为了不同的点集，这对于后续的路径搜索和信息素扩散至关重要。

4.3 对比结果分析

为了全面评估改进算法的效果，我们进行了 20 次独立实验，并记录了每次实验的结果。实验结果包括每种算法求解的最短距离、最长距离和平均距离。我们将基于分块及搜索的蚁群信息素扩散算法的结果与传统蚁群算法及朴素基于信息素扩散的改进蚁群算法的结果进行了对比，如表 4.1 所示：

从表 4.1 中我们可以观察到，基于分块及搜索的蚁群信息素扩散算法在处理 TSP 问题上，不仅在平均距离上显著优于传统蚁群算法和朴素基于信息素扩散的改进蚁群算法，而且在最短和最长路径长度上同样表现出色。这一结果明确证明了我们所提出改进算法的有效性。

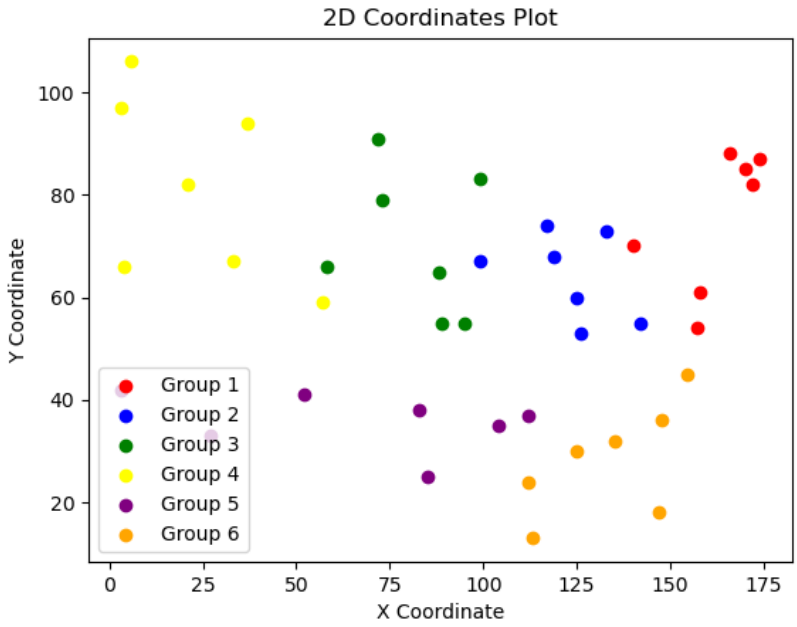

图 4.1　点集划分结果

算法	蚁群算法	朴素基于信息素扩散的改进蚁群算法	基于分块及搜索的蚁群信息素扩散算法
平均距离	745.79	733.03	712.32
最远距离	749.42	760.91	718.765
最近距离	742.91	716.68	710.708

表 4.1　实验结果

此外，值得注意的是，尽管在实验中我们使用了与之前实验相同数量的蚁群和迭代次数，改进算法在运行速度上仍然表现出显著提升。这一提升可能部分归因于使用 C++ 而非 Python 编写算法，导致运行效率的提高。然而，更重要的是，通过对数据集进行分块处理，我们实际上减少了每次迭代所需处理的数据量，从而加快了算法的运行速度。

综上所述，基于分块及搜索的蚁群信息素扩散算法在解决旅行商问题上展现了显著的优势，无论是在求解精度还是在运行效率方面都优于传统方法。

5 实验结论

在本次智能计算课程设计中，我对蚁群算法进行了深入研究和改进，旨在解决旅行商问题（TSP）。通过实验验证，改进算法在多个方面显著超越了传统蚁群算法及其基本改进版本，具体表现在以下几个关键方面：

首先，改进算法在求解 TSP 问题的性能上表现突出。通过引入基于分块及搜索的策略，算法不仅在平均路径长度上优于标准蚁群算法和基于信息素扩散的改进蚁群算法，而且在最短和最长路径长度上也表现出显著的改进。这一成果证实了我们对算法结构进行调整的有效性，突显了在处理复杂组合优化问题时，结合多种策略的重要性。

其次，我们的算法在运行效率上也有显著提升。通过对蚁群算法的分块处理和信息素扩散机制进行优化，算法在处理大规模数据时的运行速度得到了显著提升。除了由于编程语言效率的差异（在本实验中，C++ 相较于 Python 有更高的运行效率），更重要的是由于算法本身的优化，如减少迭代次数，使得算法更适合解决实际应用中遇到的大规模问题。

此外，我们的研究还表明，通过对算法进行适当的修改和优化，可以显著提高其在特定应用中的适用性。尤其是在点集的划分和信息素扩散机制方面的改进，为解决类似 TSP 这样的复杂优化问题提供了新的思路。这些改进不仅在理论上具有创新性，而且在实际应用中具有很高的实用价值。

然而，我们的研究也存在一定的局限性。例如，在点集的划分方法和信息素更新机制方面，可能需要根据不同类型的问题进行进一步的调整和优化。此外，我们的算法虽然在旅行商问题上表现优异，但其在其他类型的优化问题上的适用性还有待进一步探索和验证。

参考文献

[1] J. X. Deng, Wu and H. Zhao. An improved ant colony optimization algorithm based on hybrid strategies for scheduling problem. *IEEE access 7 (2019): 20281-20292*, 2019.

[2] 张纪会 and 徐心和. 一种新的进化算法——蚁群算法. *系统工程理论与实践*, 19(3):84–87, 1999.

[3] 贾燕花. 蚁群算法在旅行商问题（tsp）中的应用研究. *计算机与数字工程*, *计算机与数字工程*, Jan 2016.

[4] 黄挚雄, 张登科, and 黎群辉. 蚁群算法及其改进形式综述. *计算技术与自动化*, 25(3):35–38, 2006.

教师评语评分

评语:

评分:

评阅人:

年　　月　　日

(备注:对该实验报告给予优点和不足的评价,并给出百分制评分。)