2023-2024-1学期  
《人工智能》课程论文

|  |  |
| --- | --- |
| **题 目：** | **基于改进蚁群算法的多元函数求极值问题研究** |
| **姓 名：** | **冯运佳** |
| **学 号：** | **2021234110701** |
| **班 级：** | **数据2103** |
| **提交日期：** | **2023年12月17日** |

基于蚁群算法的多元函数求极值问题研究

冯运佳

**摘要：**传统的蚁群算法在函数优化问题上存在着很多问题，如需要占用过多的内存对于多峰函数的最值容易陷入局部最优解的情况，可以编码的求解空间较小，不易与其他优化算法结合。本文先是介绍了传统的蚁群算法的应用场景，算法原理和依据，然后就传统的蚁群算法提出了一些解决方案，例如合理编排使用二进制编码，同时，本文采取了经典的十个多峰函数来凸显算法的效果，同时与传统的蚁群算法的效果进行了比较。实验证明，因采用并行化策略，此方法大大降低了时间复杂度，提高了代码求解的精度，算法有优良的表现。本算法在处理峰数较少的情况下有较好的表现，结果表明本算法有较好的收敛性，算法性能良好。最后还给本算法提供了继续研究的思路，即选取合适的分隔区间作为接下来研究的方向。

**关键词：**蚁群算法，优化算法，多元函数，信息素

Research on the problem of finding extremums of multivariate functions based on ant colony algorithm

Feng Yunjia

**Abstract**: Traditional ant colony algorithms have many problems in function optimization problems, such as occupying too much memory and easily getting stuck in local optima for multimodal functions. The solution space that can be encoded is small, making it difficult to combine with other optimization algorithms. This article first introduces the application scenarios, algorithm principles, and basis of traditional ant colony algorithms, and then proposes some solutions for traditional ant colony algorithms, such as reasonable arrangement and use of binary encoding. At the same time, this article adopts the classic ten multimodal functions to highlight the effectiveness of the algorithm, and compares the effectiveness with traditional ant colony algorithms. Experimental results have shown that by adopting a parallelization strategy, this method greatly reduces time complexity and improves the accuracy of code solving, resulting in excellent performance of the algorithm. This algorithm performs well when dealing with fewer peaks, and the results show that it has good convergence and good performance. Finally, a further research approach is provided for this algorithm, which is to select appropriate separation intervals as the direction for further research.

**Key words**: Ant colony algorithm, Optimization algorithm, Multivariate function, Pheromone

**1 引言**

在实际问题中，求解多元函数的极值是一个重要而具有挑战性的任务。多元函数的极值求解是数学中的经典问题，广泛应用于自然科学、工程技术和经济管理等领域。在实际问题中，我们常常需要确定一个多元函数在一定约束条件下的最大值或最小值，以优化生产过程[1]、设计最佳方案或做出合理决策。传统的优化算法在处理复杂的多元函数时可能面临局部最优解、收敛速度慢等问题。为了克服这些限制，人们开始借鉴自然界中生物群体的行为，并将其应用到优化算法中。

蚁群算法（Ant Colony Optimization, ACO）是一种基于模拟蚁群觅食行为的启发式优化算法。它最初由意大利学者马科·德·戈洛（Marco Dorigo）于1992年提出，通过模拟蚂蚁在寻找食物过程中留下的信息素和伪随机选择路径的方式来寻找问题的最优解。蚁群算法在求解多元函数的极值问题上展现出了良好的性能和鲁棒性。

本研究旨在探讨蚁群算法在多元函数求极值问题中的应用。首先，我们将介绍蚁群算法的基本原理和算法流程，包括信息素更新规则、路径选择策略等[1]。其次，我们将详细阐述多元函数求极值问题的数学模型和目标函数定义。然后，我们将设计并实现基于蚁群算法的优化算法，并与其他传统优化算法进行对比实验，评估其性能和效果。最后，我们将讨论实验结果，并提出可能的改进方向和未来的研究方向。

本研究的意义在于提供一种新颖而有效的方法来解决多元函数求极值问题。通过借鉴蚁群行为，我们可以更好地探索问题空间，并找到全局最优解。同时，该研究还可为其他实际问题的优化提供参考，例如机器学习、图像处理等领域。

**2 问题描述与计算模型**

考虑一个多元函数，其中是自变量，是一个关于这些自变量的函数。我们的目标是找到这个多元函数在一定约束条件下的最大值或最小值[3]。

根据仿生学家的研究结果，蚂蚁的寻优能力找到蚂蚁巢穴到食物的最短距离，其基本原理是：蚂蚁会在寻找事物的路上留下一种具有挥发性的信息素，后来的蚂蚁会感知这种信息素及其强度来寻找自己的运动方向，信息素会随着时间的推移而逐渐消失。最终最短路径的信息素浓度最高，找到最短路径。

蚂蚁行为模型描述了蚂蚁在搜索空间中移动的策略，它由以下几个要素组成：

解空间表示：定义了问题的解空间，每个解对应于一个蚂蚁所在的位置[4]。

启发函数：基于问题的特点，为蚂蚁提供关于选择路径的启发信息。启发函数通常使用问题特定的启发式规则来评估解的质量。一经提出，引起了全球的广大关注，先后被运用到TSP问题，二次分配问题，图着色问题等多个经典。都有不错的效果，但是文献一[2]表明蚁群算法对于单峰函数具有好的有着良好的效果，但是对于多峰函数却无法找到所有的极值。事实上，无论是数学上还是工程上中的问题都是多峰问题。本文改进了二进制的编码方式[5]，对10个经典多峰函数都进行了检测。

选择策略：蚂蚁根据信息素和启发信息来选择下一步的移动方向。常见的选择策略有轮盘赌选择、最大值选择等。

路径更新：蚂蚁在搜索过程中通过选择路径来更新自己的解，并将其存储在解空间中。

前进规则：确定蚂蚁前进的方式，例如按概率选择、按照某种规则选择等。

通过蚂蚁行为模型，蚂蚁可以在解空间中进行探索，并根据信息素和启发信息来选择最优路径。

信息素更新模型：信息素更新模型描述了信息素在蚂蚁搜索过程中的变化规律，它由以下几个要素组成：

信息素表示：定义了问题的解空间中每条路径上的信息素量，通常用矩阵表示。

信息素初始化：在算法开始时，初始信息素值可以是相等的或根据问题特点进行设置。

信息素挥发：为了避免信息素陷入局部最优解，需要让信息素逐渐挥发。挥发率决定了信息素在每次迭代中的衰减程度。

信息素增强：当蚂蚁选择某条路径后，会在该路径上释放信息素，增强该路径的信息素值。增强量通常与路径的质量相关。

通过信息素更新模型，信息素在蚂蚁搜索过程中可以动态地调整，以引导蚂蚁更好地探索解空间，并集中于全局最优解的路径上。

本文将并行引入了算法[6]，相较传统的蚁群算法，避免了传统算法容易陷入多峰函数局部最优解的情况。实验通过几个不同的多峰函数证明算法效果较好。

**3. 蚁群算法设计**

**3.1 编码**

一个好的编码可以减少搜索空间维度，降低问题的复杂度，从而提高算法的搜索效率。可以引导着算法朝着优解方向搜索，减少局部最优解。

本文采取的是二进制编码[7]的编码方式，编码长度为，其中是单个变量的二进制编码长度，是该函数变量个数。

相较传统的蚁群算法编码方式，本文采取的编码方式是蚂蚁选择路径时只有0,1两种选择，只需选择编码长度的次数即可找到一个解。

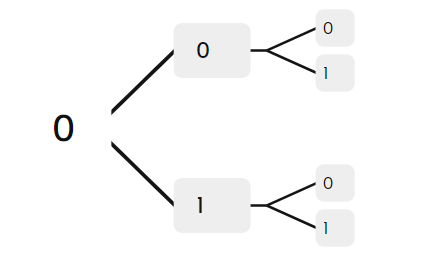


图1 蚂蚁个体寻优方式

**3.2 解码**

对候选集中的每一个变量用字长为N的二进制编码串表示，其中其中为该实数编码首位，为该实数编码末位。该值的范围是[,]，该二进制串对应的十进制数为，解码的公式为：

 (1)

**3.3 初始化种群**

相较传统的蚁群算法，本文采用多个蚁群进行搜索，通过区间划分的方式将每个区间的极大值搜索出来，进而取出这些极大值里面的最大值，作为该种群所求解出来的最优解。将所有变量定义域所组成的空间分成M个，在每个空间里随机产生一个种群，每个种群中随机产生N个个体，并依据编码规则随机进行编码，计算每个个体的适应度，将信息素浓度设为统一一个常数。

**3.4 路径选择**

每个蚂蚁将每一个二进制编码数字作为节点来进行路径选择，只需根据面对的两条边留下的信息素大小进行选择，选择信息素浓度高的位置的概率较大[8]。

**3.5 信息素更新**

信息素的更新分为两个过程：信息素的挥发；信息素的释放。信息素的挥发是指信息素随着时间的推移，浓度逐渐变低的过程。

 (2)

其中指的是t时刻路径i的第j个元素的信息素存量，是信息素的持久性（），经过多次迭代后，最优路径的信息素浓度会最高，是每一次迭代的最优解，同时也是全局最优解。

**3.6 协同操作策略[9]**

蚂蚁之间通过信息素浓度来达到间接通信的目的，若某个蚂蚁找到最优路径，则可以通过协同策略使得该路径上的信息素增加，进而通知到其他蚂蚁。

实际操作中，根据搜索空间划分出来的合适的区间，，在每一个区间内产生一个种群，，初始化每个种群和信息素；

DO{

种群在区间搜索路径，计算其适应度，得到最优个体；

更新各个种群信息素[10]；

统计各个种群最优解；

}WHILE（达到最大循环次数）

**4 仿真实验与结果分析**

对于一个算法性能的评估，需要选择合适的评估方式及参数进行评估，本实验根据实际的需求情况，选择从算法的预判决准确率及对编码复杂度和编码性能影响等方面对本实验算法进行评估。

测试函数1：Rastrigin函数



该函数的最大值是一个无限不循环小数，最大值的点也是一个无限不循环小数。因此，没有绝对具体的取得最大值的点，只有不断逼近最大值。经过算法的不断逼近，在多个区间找出其最大值，其中最大的值即是我们所要求解的答案，但是这个无限不循环小数是一个不固定的值，因此，我们在求解该函数极大值的时候只能知道大概值，在某区间内。使用本文改进的蚁群算法的结果如下：

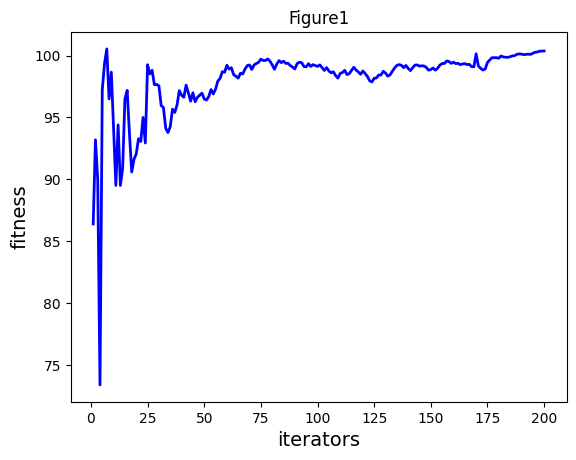


图2 测试函数f1(x)迭代过程中每一代的最优解

最大值为100.69239694565677，此时的x为[4.5315409 4.52336501]

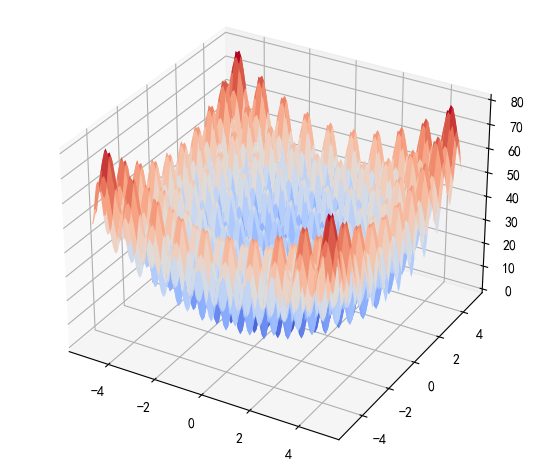


图3 测试函数f1(x)的三维函数图像

测试函数2：Ackley函数





该函数的最大值是一个无限不循环小数使用本文的蚁群算法求解后得到最大的函数值为14.302624162861365，当取得此最大值时的的值为 [-4.60001902 ， 4.59910193]，此函数与实验函数一一样，同样结果是一个无限不循环小数，因此此函数的结果也是一个不固定的值，我们在求解函数的极大值的大概值。这次我们将我们搜索到的最优解用红点标在函数图里，可以看出搜索结果很好。

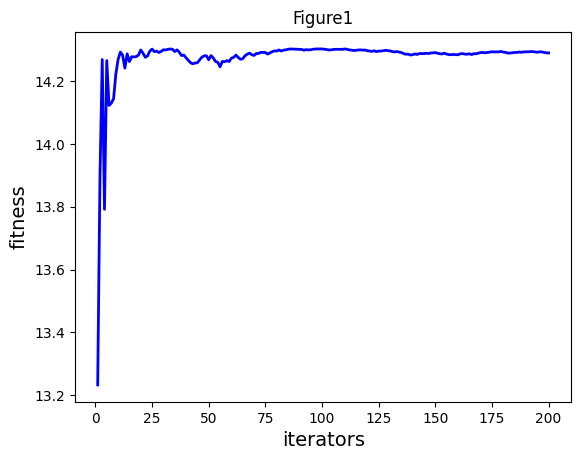


图4 测试函数f2(x)迭代过程中每一代的最优解

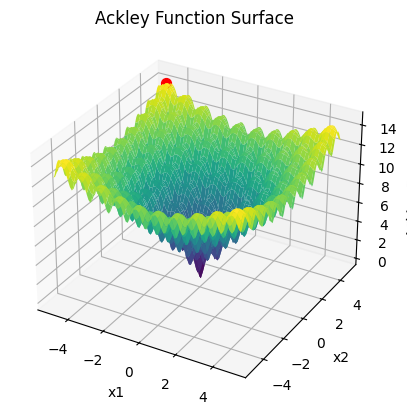


图5 测试函数f1(x)的三维函数图像

接下来的8个函数以表格形式体现，如表x：

表1 不同函数求解值和最大值差距

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 函数名 | 求解值 | X1 | X2 | 差距 |
| Michalewicz | 1.967263 | 4.96841 | -1.5715 | 0.0003 |
| Griewank | 2.004036 | -0.0408 | -4.4642 | 0.0034 |
| Branin RCOS | 0 | -3.1433 | 12.279 | 0.3979 |
| Kimono | 8.386894 | 2.32727 | -0.8202 | 0.0342 |
| Weierstrass | 3.984721 | 1.49993 | 5 | 0.3345 |
| De Jong | 25 | 5 | -5 | 0 |
| Rosenbrock | 90036.0 | -5 | -5 | 0 |
| Dixon-Price | 3636.0 | -5 | 5 | 0.0001 |
| Bukin | 227.7698 | 4.25932 | -5. | 0.234 |

经过300次迭代后，每个函数都找到了最优解，精度较为理想，相比传统的蚁群算法，精度较为理想，效率有了进一步的提升。

**5 结论**

本文采用二进制编码的蚁群算法对于多峰函数最大值的求解，有效的克服了蚁群不能求解连续参数的优化问题的缺陷，同时并行进行搜索，提高了算法的效率，也扩大了搜索范围，同时速度较为理想，一般来说只需要200到300次循环即可获得较为理想的答案。具有很广的应用范围，还可以推广到一般化的全局化问题。但是，对于不同的函数拥有多个极值时，需要对函数有一定水平的区间划分，需要相应的专业知识。对于区间划分方面的研究可以成为接下来的重点。

**参考文献：**

1. 王海超.基于集群优化蚁群算法的加工空行程优化方法[J/OL].计算机集成制造系统:1-21[2023-12-18].https://doi.org/10.13196/j.cims.2023.0315.
2. 李敏强, -. and -. 寇纪淞, - 多模态函数优化的协同多群体遗传算法. - 自动化学报, 2002. - 28(- 4): p. - 497.
3. 熊伟清, -. and -. 魏平, - 二进制蚁群进化算法. - 自动化学报, 2007. - 33(- 3): p. - 259.
4. 吴 斌, -. and -.史忠植, -一种基于蚁群算法的TSP问题分段求解算法. – 计算机学报 2001
5. 米泽辉,郭肃丽,秦固平等.基于蚁群优化算法的非均匀子阵划分技术[J/OL].计算机测量与控制:1-10[2023-12-18].http://kns.cnki.net/kcms/detail/11.4762.TP.20231115.1541.002.html
6. 马晓平,赵学涛,王炬成.加入动态搜索模型的蚁群算法及其应用[J].计算机工程与设计,2023,44(11):3462-3468.DOI:10.16208/j.issn1000-7024.2023.11.034..
7. 张玉杰,高元楼,李铎等.改进蚁群算法优化的密集装配路径规划[J].兵工自动化,2023,42(11):30-32.
8. 潘志安,王小英,王茂发.基于改进蚁群算法的多控制器负载均衡仿真[J].计算机仿真,2023,40(10):321-325.
9. 王昱钦.基于粒子群和蚁群混合算法的柔性车间调度算法[J].电子设计工程,2023,31(17):65-69.DOI:10.14022/j.issn1674-6236.2023.17.013.
10. 李炫秋,黄斐君,景鹏飞.基于量子蚁群算法的旅行商问题求解及算法评估[J/OL].