

## 基于迁移知识选择和种群削减的进化多任务优化算法

李二超, 黄鹏飞

引用本文

李二超, 黄鹏飞. 基于迁移知识选择和种群削减的进化多任务优化算法[J]. 计算机科学, 2026, 53(2): 349-357.

LI Erchao, HUANG Pengfei. [Evolutionary Multi-task Optimization Algorithm Based on Transfer Knowledge Selection and Population Reduction](#) [J]. Computer Science, 2026, 53(2): 349-357.

---

## 相似文章推荐 (请使用火狐或 IE 浏览器查看文章)

Similar articles recommended (Please use Firefox or IE to view the article)

### [基于图染色混合进化算法的长期多智能体任务分配](#)

Lifelong Multi-agent Task Allocation Based on Graph Coloring Hybrid Evolutionary Algorithm  
计算机科学, 2025, 52(7): 262-270. <https://doi.org/10.11896/jsjcx.240600016>

### [自扰动和极性维度交互的自适应差分进化算法](#)

Adaptive Differential Evolution Based on Self-guided Perturbation and Extreme DimensionExchange  
计算机科学, 2025, 52(6A): 240800100-14. <https://doi.org/10.11896/jsjcx.240800100>

### [云边协同环境下面向负载时间窗口的无服务器应用资源分配方法](#)

Resource Allocation Method with Workload-time Windows for Serverless Applications inCloud-edge Collaborative Environment  
计算机科学, 2025, 52(6): 336-345. <https://doi.org/10.11896/jsjcx.240400073>

### [面向工业动态取送货问题的分解多目标进化算法](#)

Decomposition-based Multi-objective Evolutionary Algorithm for Industrial Dynamic Pickup andDelivery Problems  
计算机科学, 2025, 52(1): 331-344. <https://doi.org/10.11896/jsjcx.231200132>

### [均衡加权图着色问题与启发式算法](#)

Balanced Weighted Graph Coloring Problem and Its Heuristic Algorithms  
计算机科学, 2024, 51(11A): 231200103-7. <https://doi.org/10.11896/jsjcx.231200103>

# 基于迁移知识选择和种群削减的进化多任务优化算法

李二超 黄鹏飞

兰州理工大学电气工程与信息工程学院 兰州 730050

**摘要** 进化多任务优化是近年来计算智能领域的研究热点之一,其原理是通过任务间的知识迁移提高算法同时求解多个任务的效率。不合理的迁移知识选择会降低任务间的正向知识迁移,因此如何合理选择迁移知识成为了当前的重点研究方向。此外,在算法进化过程中,单层种群削减难以长期维持算法的高效优化性能。基于此,提出了一种基于迁移知识选择和种群削减的进化多任务优化算法(MTDE-MCT)。首先,初始化任务种群并进行适应度评估,采用基于曼哈顿距离和适应度值的联合指标进行迁移知识的选取。其次,通过子群体对齐策略消除任务间迁移个体的特征差异。最后,提出了一种多层种群削减策略,根据算法的进化阶段对任务种群进行线性规模的削减。为验证所提算法的性能,在 CEC2017 问题测试集和 WCCI2020 问题测试集上将其与近几年的经典算法进行了比较。实验结果证明,该算法在求解多任务优化问题时具有较强的竞争力。

**关键词:** 进化算法;多任务优化;迁移知识选取;子群体对齐;多层种群削减

中图分类号 TP18

## Evolutionary Multi-task Optimization Algorithm Based on Transfer Knowledge Selection and Population Reduction

LI Erchao and HUANG Pengfei

College of Electrical Engineering and Information Engineering, Lanzhou University of Technology, Lanzhou 730050, China

**Abstract** Evolutionary multi-task optimization has emerged as one of the research hotspots in the field of computational intelligence in recent years, with its principle being to enhance the efficiency of algorithms in simultaneously solving multiple tasks through knowledge transfer between tasks. Since improper selection of transfer knowledge can reduce positive knowledge transfer between tasks, how to appropriately select transfer knowledge has become a key research direction. Additionally, during the algorithm's evolutionary process, single-layer population reduction is insufficient to sustain the algorithm's efficient optimization performance over the long term. Based on this, this paper proposes an evolutionary multi-task optimization algorithm (MTDE-MCT) based on transfer knowledge selection and population reduction. Firstly, the task population is initialized, and fitness evaluation is conducted, utilizing a combined index based on Manhattan distance and fitness values for the selection of transfer knowledge. Next, a subpopulation alignment strategy is applied to eliminate feature differences in transfer individuals between tasks. Finally, a multi-layer population reduction strategy is proposed, which linearly reduces the task population size based on the algorithm's evolutionary stage. To validate the performance of the proposed algorithm, comparisons are made with classic algorithms from recent years using the CEC2017 and WCCI2020 problem test sets. The experimental results demonstrate that the proposed algorithm exhibits strong competitiveness in solving multi-task optimization problems.

**Keywords** Evolutionary algorithm, Multi-task optimization, Transfer knowledge selection, Subpopulation alignment, Multi-layer population reduction

### 1 引言

多任务优化问题普遍存在于科学、工程和工业的各个领域,并随着科技的发展日益复杂化,如大规模优化问题<sup>[1]</sup>、调度优化问题<sup>[2]</sup>、车辆路径规划<sup>[3]</sup>和背包问题<sup>[4]</sup>等。求解此类优化问题的目标是在给定的约束条件下,从可行域中寻找一

个或一组最优解。在这种复杂背景下,进化算法(EA)应运而生<sup>[5]</sup>,EA是受生物自然进化过程的启发而提出的一种基于种群的搜索算法,因其卓越的搜索性能,被广泛用于解决实际生活中的复杂优化问题。

尽管EA具有强大的隐式并行计算能力,但现有的EA主要集中于优化单一任务,而现实生活中常常面临着多个任

到稿日期:2025-06-24 返修日期:2025-09-28

基金项目:国家自然科学基金(62063019);甘肃省科技计划重点研发计划(25YFGA030);甘肃省自然科学基金重点项目(24JRRA173)

This work was supported by the National Natural Science Foundation of China(62063019), Key Research and Development Program of Gansu Province Science and Technology Plan(25YFGA030) and Key Project of Natural Science Foundation of Gansu Province(24JRRA173).

通信作者:李二超(lecstarr@163.com)

务需要同时优化的情况,传统的单任务进化算法已经无法满足实际需求。在现实世界中,很多问题之间都具有潜在的关联性<sup>[6]</sup>,一个优化问题中获得的信息可以用于优化另外一个问题。受任务间关联性的启发,Gupta 等<sup>[7]</sup>首先提出了一种多因子进化算法(Multi-Factorial Evolutionary Algrithm, MFEA),该算法利用了多种群的隐式并行性,仅使用一个种群就能表示多个任务的解。相比单任务优化,该优化算法能更快收敛。在多任务进化过程中,不同的优化任务具有不同的适应度地形,因此需要不同的进化机制和搜索策略进行解<sup>[8]</sup>。在单种群框架下,每个任务的进化方式相同,这在一定程度上会影响进化效率。为了保证每个任务的进化过程,Li 等<sup>[9]</sup>开发了明确的多种群进化框架(MPEF),此框架不仅便于对不同任务采用不同的优化方法,还能使用多种跨任务的知识迁移方法。

现有的多任务优化算法研究主要集中于单一任务的优化方法、任务间知识的表达方法以及迁移策略的方法。在单一任务的优化方法上,Feng 等<sup>[10]</sup>将粒子群优化(PSO)和差分进化(DE)引入多因子进化算法(MFEA),提出了多因子粒子群优化(MFPSO)和多因子差分进化(MFDE)算法。Wu 等<sup>[11]</sup>提出了一种模糊系统优化的多任务遗传算法(MTGA),通过消除最优值之间的偏差,改善了处理差异较大任务的性能。在任务间知识的表达方法上,多数算法将任务中表现较好的解作为传递的知识。Xue 等<sup>[12]</sup>将任务中有效的搜索方向作为知识。Ma 等<sup>[13]</sup>先对个体进行变换,将转换后的个体作为知识,使得迁移知识更适合于当前任务。在迁移策略的方法上,Wang 等<sup>[14]</sup>认为任务中表现较好的个体可能是有效迁移解,采用直接注入的方式进行迁移。Chen 等<sup>[15]</sup>使用不同的交叉算子来进行任务之间的知识迁移,增强种群的多样性。Zhao 等<sup>[16]</sup>提出了一种混合知识迁移策略的多任务进化算法,解决了迁移目标的问题,其通过种群变化分析来确定多个迁移目标,提高了多任务优化算法的性能。

随着多任务优化的进一步发展,部分学者还考虑到了知识转移(KT)技术对多任务优化的作用。传统的方法是基于统一搜索空间<sup>[17]</sup>来进行决策空间在任务之间的映射。另一项突出的方法是基于领域适应的知识转移,通过适当的转换使源任务中的信息适应目标任务,以促进任务之间的知识转移。Liang 等<sup>[18]</sup>提出了一种基于子空间对齐的多目标多任务算法,使用子空间学习得到的映射矩阵来改变个体的决策向量,减小任务间负知识转移的概率。Gao 等<sup>[19]</sup>考虑到子空间分布信息的发散性,提出了一种同时对齐碱基和分布的方法,最小化了源域和目标域之间的差异。Wang 等<sup>[20]</sup>利用深度学习模型生成迁移个体并引导算法种群进化轨迹,其利用变分自编码器(VAE)沿着特征维度进行采样来产生迁移个体,增强了算法的全局搜索能力。Zhang 等<sup>[21]</sup>提出了一种具有自适应双知识迁移的多任务差分进化算法。

任务间迁移个体随机选择会导致知识负迁移,不利于算法的收敛,设计合理的迁移个体选取策略对促进任务间知识正迁移具有重要作用。此外,单层种群削减无法长期保持算法高效的优化性能。基于此,本文提出了一种基于迁移知识选择和种群削减的进化多任务优化算法(Evolutionary Multi-

task Optimization Algorithm Based on Transfer Knowledge Selection and Population Reduction, MTDE-MCT),用于求解多任务优化问题。

本文的主要贡献如下:

1)提出了一种基于曼哈顿距离和适应度值的联合指标,并将其作为任务间选择迁移知识的基准,综合考虑了迁移知识的质量与多样性;

2)提出了一种改进的子群体对齐策略,利用任务间具体子群体的特征来消除差异,确保促进任务间知识的正迁移;

3)提出了一种多层种群削减策略,根据合理的进化区间对任务种群进行线性规模削减,从而集中任务种群的优势。

## 2 相关工作

### 2.1 多任务优化问题

多任务优化问题的目标是通过单一种群同时求解多个任务的最优解,利用种群搜索的隐式并行性,发掘不同任务间的潜在互补关系,实现任务在多任务环境下的高效优化。一般来说,MTO的目标是同时处理 $k$ 个任务,以在单次运行中获得每个任务的最佳解决方案。MTO模型如式(1)所示:

$$\begin{cases} x_1^* = \operatorname{argmin} f_1(x_1), x_1 \in \Omega_1 \\ x_2^* = \operatorname{argmin} f_2(x_2), x_2 \in \Omega_2 \\ \dots \\ x_k^* = \operatorname{argmin} f_k(x_k), x_k \in \Omega_k \end{cases} \quad (1)$$

其中, $x_i$ 是任务 $T_i$ 的最优解, $f_i$ 是任务 $T_i$ 的目标函数, $\Omega_i$ 是任务 $T_i$ 的决策空间, $i=1,2,3,\dots,k$ 。当MTO问题中的任务数量超过3时,将其称为多任务优化问题。

### 2.2 自适应双知识迁移的多任务差分进化算法

自适应双知识迁移的多任务差分进化算法(Multitask Differential Evolution with Adaptive Dual Knowledge Transfer, MTDE-ADKT)是Zhang 等<sup>[21]</sup>于2024年提出的一种MTO算法。其采用了基于多种群的任务框架(见图1),每个群体处理特定任务。

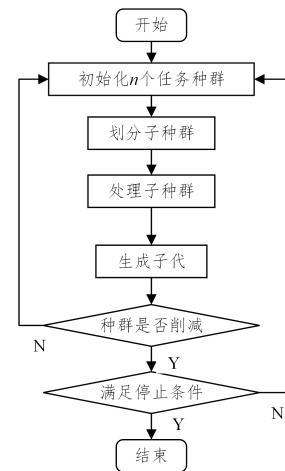


图1 自适应双知识迁移的多任务差分进化算法

Fig. 1 Multi-task differential evolution method with self-adaptive dual knowledge transfer

该算法框架的具体流程为:首先为每个任务随机初始化种群,依据自适应知识迁移率将每个任务群体划分为每代

3个子群体,  $RSP$  的更新发生在每代知识迁移后。子群体 1 和子群体 2 分别使用了领域自适应技术(DA)和统一搜索空间技术(USS)来消除任务间的差异。随后更新  $RSP$  值,判断种群是否削减。最后,判断是否达到预定义的最大函数评估数量,若判断为是,则输出结束,若判断为否,则重新初始化种群。该算法框架能够有效消除任务之间的知识差异,增强其处理多样化和复杂的优化挑战的能力,这对于快速求解多任务优化问题是极其重要的。

### 3 本文方法

本文提出了一种基于迁移知识选择和种群削减的进化多任务优化算法(MTDE-MCT)。与 MFEA 依靠技能因子划分的隐式多种群结构不同, MTDE-MCT 采用了显式多种群框架,为每个任务分配了一个单独的种群。MTDE-MCT 算法在 MTDE-ADKT 的基础上进行了改进,加入了基于曼哈顿距离和适应度值的联合指标进行迁移知识选择,随后利用子种群对齐策略消除目标任务子群体和迁移知识间的差异。最后为兼顾算法的探索和开发能力,达到更好的收敛速度和更稳定的性能,依据多层种群削减策略来线性削减种群中表现较差的个体。MTDE-MCT 算法流程框架如图 2 所示。

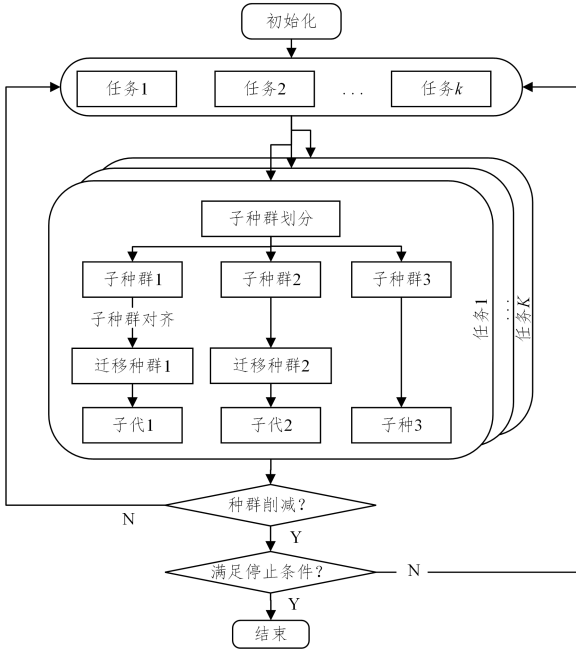


图 2 MTDE-MCT 算法的流程图

Fig. 2 Flowchart of the MTDE-MCT algorithm

MTDE-MCT 算法主要由种群初始化、迁移知识选择、子群体对齐策略、多层种群削减和条件停止判别 5 部分组成。

1) 种群初始化。在统一搜索空间为每个任务随机生成大小为  $N$  的种群和存档 Arc。

2) 迁移知识选择。联合指标对目标任务子群体 1 和 2 进行知识迁移个体的选取,在选取过程中避免重复选取。

3) 子群体对齐策略。将目标任务子群体以及迁移个体进行归一化处理,将迁移个体进行对齐处理。

4) 多层种群削减。根据算法迭代次数是否达到预定种群削减的迭代次数,将种群进行线性的削减,选取部分最优解作

为下一代种群并更新档案 Arc;否则重复执行第 2—4 阶段。

5) 条件停止判别。达到评估的最大次数时,终止循环。

#### 3.1 迁移知识选择

本文提出的多任务优化算法通过利用不同任务种群个体之间的联系选择迁移知识,采用曼哈顿距离之和衡量个体间的差异,从而保证迁移个体的多样性。同时,为了确保迁移个体的质量,以适应度值来衡量所选个体在其任务上的优势。为减少任务间相似度较低导致的负向知识迁移引起的性能下降,本文依据曼哈顿距离和适应度值的联合指标对个体进行排序,选取排名靠前的个体作为迁移个体,进一步提升迁移效果。有关指标的计算式为:

$$d_i(x_i, x_j) = \sum_{i=1}^n \sum_{j=1}^k |x_i - x_j| \quad (2)$$

$$\hat{f}(x) = f(x_j) \quad (3)$$

其中,  $d_i(x_i, x_j)$  为目标任务子群体中的个体和其他任务种群个体之间的曼哈顿距离之和,  $x_i$  为目标任务子群体中的个体,  $x_j$  为其他任务种群中的个体,  $\hat{f}(x)$  为个体适应度值。

联合指标的计算式为:

$$f = \alpha \cdot d_i + (1 - \alpha) \cdot \hat{f} \quad (4)$$

其中,  $f$  为联合指标值,  $\alpha$  为权重系数。

算法 1 给出了迁移知识选择策略的伪代码。首先计算曼哈顿距离和适应度值,将其作为联合指标对迁移知识个体进行排序。其在选取完第一个子群体的迁移知识后,排除已选知识再进行第二个子群的迁移知识选取。避免所选知识的重复,增加了种群的多样性。

#### 算法 1 迁移知识选择算法

输入: 目标任务子群体  $SP_1, SP_2$ ; 权重系数  $\alpha$ ; 辅助任务种群  $P_k$

输出: 迁移个体集合  $OP_1, OP_2$

1. if  $\text{length}(SP_1) > 0$
2. for  $i=1$  to  $SP_1$  do
3. 根据式(2)计算子种群  $SP_1$  中个体与辅助任务种群个体的曼哈顿距离之和
4. end
5. 根据式(3)计算辅助任务种群个体适应度
6. 根据式(4)对迁移的个体进行升序排序
7. 选择排序后的前  $\text{length}(SP_1)$  个个体作为迁移个体集合  $OP_1$ ;
8. 排除已选的辅助任务种群个体
9. if  $\text{length}(SP_2) > 0$
10. for  $i=1$  to  $SP_2$  do
11. 根据式(2)计算子种群  $SP_2$  中个体与辅助任务种群剩余个体的曼哈顿距离之和
12. end
13. 根据式(3)计算辅助任务种群剩余个体的适应度
14. 根据式(4)对迁移个体进行升序排序
15. 选择排序后的前  $\text{length}(SP_2)$  个个体作为迁移个体集合  $OP_2$ ;
16. end
17. end

#### 3.2 子群体对齐

本文细化了领域自适应技术(DA),从减小任务种群间的差异将其改进为减小任务子群体和迁移个体之间的差异。为了减少在任务间相似度较低的情况下造成的知识负迁移现

象,首先计算迁移个体子群体和目标任务子种群决策变量的均值。其具体计算式如下:

$$\begin{cases} \mu_s = \frac{1}{n} \sum_{i=1}^n D_i \\ \mu_t = \frac{1}{n} \sum_{i=1}^n P_i \end{cases} \quad (5)$$

其中,  $\mu_s$  是所选的迁移个体子群体决策变量的均值,  $D_i$  是迁移个体子群体中第  $i$  个个体的决策变量,  $\mu_t$  是目标任务子群体决策变量的均值,  $P_i$  是目标任务子群体中第  $i$  个个体的决策变量,  $n$  为子群体中的个体数量。随后对迁移个体、迁移个体构成的迁移知识子群体和当前任务子群体进行归一化处理。其具体计算式如下:

$$\begin{cases} m_i = D_i - \mu_s \\ D = D - \mu_s \\ P = P - \mu_t \end{cases} \quad (6)$$

其中,  $m_i$  为归一化后的迁移个体,  $D$  和  $P$  分别为归一化后的迁移个体子群体决策变量和当前任务子群体决策变量。为了减小任务间差异的影响,对迁移个体进行子群体对齐处理。迁移个体的对齐计算式为:

$$\begin{cases} c_s = \frac{1}{n} \sum_{i=1}^n (D_i - \mu_s)(D_i - \mu_s)^T \\ c_t = \frac{1}{n} \sum_{i=1}^n (P_i - \mu_t)(P_i - \mu_t)^T \\ D_i^* = m_i \cdot C_s^{-\frac{1}{2}} \cdot C_t^{\frac{1}{2}} + \mu_t \end{cases} \quad (7)$$

其中,  $c_s$  为迁移个体构成子群体的决策变量协方差矩阵,  $c_t$  为目标任务子群体的决策变量协方差矩阵,  $m_i$  为归一化后的迁移个体,  $n_i$  为对齐后的迁移个体。

子群体对齐算法的伪代码如算法 2 所示。

#### 算法 2 子群体对齐算法

输入: 迁移个体  $D_i$ , 目标任务子群体个体  $P_i$

输出: 对齐后的迁移个体  $D_i^*$

1. for  $i=1$  to  $P_i$  do
2. 根据式(5)计算迁移个体子群体和目标任务子群体决策变量均值
3. 根据式(6)对迁移个体、迁移个体子群体和目标任务子群体进行归一化
4. 根据式(7)生成对齐后的迁移个体  $D_i^*$
5. end

### 3.3 多层种群削减

本文提出了一种多层种群削减策略,有助于算法长期保持高效的收敛性。在算法运行过程中,依据具体的进化代数对任务种群进行线性规模的削减,有助于提高种群聚焦收敛。在算法运行的过程中,随着迭代次数的增加,当前任务种群逐渐趋向于最优,需要随着算法运行而线性改变削减的种群数量,进一步保证任务种群个体的质量。其具体计算式为:

$$P^* = P \cdot \frac{1}{\frac{1}{\alpha} + \frac{gen}{g}} \quad (8)$$

其中,  $P^*$  表示削减后的种群个数,  $P$  表示目标任务当前种群个体数,  $\alpha$  表示联合指标的权重系数,  $gen$  表示当前的迭代次数,  $g$  表示间隔代数。

算法 3 给出了多层种群削减策略的伪代码。首先判断当前迭代次数是否达到执行种群削减策略的迭代次数,其次识

别削减种群规模的数量。当判断完成后,将当前任务种群个体按适应度值排序,从当前任务种群中削减排名靠后的  $N+1$  到末尾的个体,并将其加入当前任务的存档中。判断任务存档中的个体数量是否超过阈值,并更新存档中的个体,直到输出削减后的种群和更新后的存档。

#### 算法 3 多层种群削减策略

输入: 当前迭代次数  $gen$ ; 当前任务种群个体数  $P$ ; 当前任务存档  $arc$ ; 任务数  $T$ ; 最大迭代次数  $max\ gen$

输出: 缩减后的种群; 更新后的存档

1. if  $gen < maxgen$
2. if 间隔了  $g$  代
3. 依据式(8)削减任务种群
4. end
5. for  $t=1$  to  $T$  do
6. 对任务  $k$  中的个体进行适应度值排序
7. 将排名靠后的个体添加到存档  $arc$
8. if  $length(arc) > N$
9. 随机选择  $N$  个个体进入存档
10. end
11. 将当前任务种群削减为排名前  $N$  的个体
12. end
13. end

### 3.4 计算复杂度分析

MFEA 算法是经典的多因子进化算法,属于单种群的多任务框架。其任务种群规模为  $N$ , 任务数为  $T$ , 决策变量维度为  $D$ , 其交叉变异生成子代的复杂度为  $O(N \cdot D)$ , 迭代次数为  $G$ , 故 MFEA 算法的总体运算复杂度为  $O(G \cdot N \cdot D)$ 。

MTDE-MCT 算法基于多种群框架,可以确定,算法的计算复杂度主要取决于迁移知识选择策略,其复杂度为  $O(T \cdot N^2)$ 。其中,  $T$  为任务数,  $N$  为种群规模。然后每个种群都要生成子代,其复杂度为  $O(T \cdot N \cdot D^2)$ ,  $D$  为决策变量维数。子群体对齐策略的运算复杂度为  $O(N^2 \cdot D^2)$ , 种群削减策略的计算复杂度为  $O(T \cdot N \log N)$ 。由于主循环体每次迭代的复杂度由知识转移  $O(T \cdot N^2)$  和子代生成  $O(T \cdot N \cdot D^2)$  主导,因此本文算法的总体计算复杂度为  $O(G \cdot T \cdot N^2 \cdot D)$ , 其中  $G$  为最大迭代次数。

## 4 实验与分析

为了验证 MTDE-MCT 算法对解决多任务优化问题的有效性,本章分别在 *mult-task* 测试集 CEC2017 和 *mult-task* 复杂测试集 WCCI2020 上进行了实验对比,实验结果证明了 MTDE-MCT 算法的优越性。

### 4.1 测试问题集和比较算法

CEC2017 测试集包含 9 个多任务单目标测试问题,每个问题由两个最小化连续优化任务组成,依据全局最优解的交集和地形相似性被划分为不同的类别,具体性质如表 1 所列。根据全局最优解的相交程度,CEC2017 测试集被划分为完全相交(Complete Intersection, CI)、部分相交(Partial Intersection, PI)和不相交(No Intersection, NI)3 类。其次,根据函数适应度外观之间的联系,这些问题也可以被分为高相似性

(High Similarity, HS)、中等相似性(Medium Similarity, MS)和低相似性(Low Similarity, LS)<sup>[22]</sup> 3组。

表1 CEC2017测试集中的9个测试问题的特性

Table 1 Characteristics of the nine test problems in the CEC2017 benchmark set

种类	任务函数	交叉度	相似度	
CI+HS	Griewank(T1)	完全相交	1.0000	
	Rastrigin(T2)			
CI+MS	Ackley(T1)		0.2261	
	Rastrigin(T2)			
CI+LS	Ackley(T1)		0.0002	
	Schwefel(T2)			
PI+HS	Rastrigin(T1)		部分相交	0.8670
	Sphere(T2)			
PI+MS	Ackley(T1)			0.2154
	Rosenbrock(T2)			
PI+LS	Ackley(T1)	0.0725		
	Weierstrass(T2)			
NI+HS	Rosenbrock(T1)	无相交		0.9434
	Rastrigin(T2)			
NI+MS	Griewank(T1)			0.3669
	Weierstrass(T2)			
NI+LS	Rastrigin(T1)		0.0016	
	Schwefel(T2)			

WCCI2020-MTSO测试集包含10个复杂多任务单目标基准问题。WCCI2020-MTSO的任务地形更为复杂,任务的组合具有多样化的特征,相似性差异较大,甚至包含高度不相关的任务。

本文选择MFEA<sup>[6]</sup>,MFEA-II<sup>[23]</sup>,EMEA<sup>[24]</sup>,MEEA-AKT<sup>[25]</sup>,MTEA-Sao<sup>[26]</sup>,MTDE-ADKT<sup>[19]</sup>作为对比算法,与MTDE-MCT进行仿真对比实验。

表2 MTDE-MCT与其余6种对比算法在CEC2017上的实验结果

Table 2 Experimental results of MTDE-MCT and six comparative algorithms on CEC2017

类别	任务	MFEA	EMEA	MFEA-II	MFEA-AKT	MTEA-Sao	MTDE-ADKT	MTDE-MCT
CI+HS	T1	3.2150×10 <sup>-2</sup> -	3.6724×10 <sup>-2</sup> -	5.8135×10 <sup>-2</sup> -	2.5457×10 <sup>-2</sup> -	3.0277×10 <sup>-2</sup> -	<b>0.0000=</b>	<b>0.0000</b>
	T2	1.5173×10 <sup>2</sup> -	4.4095×10 <sup>2</sup> -	8.1723×10-	1.1235×10 <sup>2</sup> -	1.4770×10 <sup>2</sup> -	<b>0.0000+</b>	1.9125×10 <sup>-14</sup>
CI+MS	T1	1.1157-	4.1309×10 <sup>-1</sup> -	8.8292×10 <sup>-2</sup> -	2.2078-	5.2078×10 <sup>-1</sup> -	2.0967×10 <sup>-12</sup> -	<b>3.7964×10<sup>-13</sup></b>
	T2	1.3357×10 <sup>2</sup> -	4.3746×10 <sup>2</sup> -	3.1550-	1.1894×10 <sup>2</sup> -	8.8692×10-	<b>0.0000+</b>	6.5016×10 <sup>-4</sup>
CI+LS	T1	2.0288×10-	2.1084×10-	2.1188×10-	2.0312×10-	2.1205×10-	1.0327×10-	<b>2.3401×10<sup>-1</sup></b>
	T2	2.3620×10 <sup>3</sup> -	1.1733×10 <sup>4</sup> -	1.1764×10 <sup>3</sup> -	2.1377×10 <sup>3</sup> -	2.2175×10 <sup>3</sup> -	7.6517×10 <sup>2</sup> -	<b>4.2765</b>
PI+HS	T1	2.7218×10 <sup>2</sup> -	2.1687×10 <sup>2</sup> -	3.8101×10 <sup>2</sup> -	2.8063×10 <sup>2</sup> -	2.1686×10 <sup>2</sup> -	<b>7.0295×10=</b>	8.3320×10
	T2	1.6286×10 <sup>-2</sup> -	4.2981×10 <sup>-1</sup> -	3.3224×10 <sup>-3</sup> -	1.1202×10 <sup>-2</sup> -	6.7424×10 <sup>-2</sup> -	<b>1.4081×10<sup>-20</sup>+</b>	5.6195×10 <sup>-11</sup>
PI+MS	T1	6.8264×10 <sup>-1</sup> -	4.0717×10 <sup>-1</sup> -	1.7480×10 <sup>-1</sup> -	2.7794-	4.0087×10 <sup>-1</sup> -	4.4043×10 <sup>-11</sup> -	<b>3.6066×10<sup>-12</sup></b>
	T2	1.2642×10 <sup>2</sup> -	1.2457×10 <sup>3</sup> -	1.2087×10 <sup>2</sup> -	1.8857×10 <sup>2</sup> -	1.4704×10 <sup>2</sup> -	<b>4.3894×10+</b>	6.9423×10
PI+LS	T1	1.3524×10-	3.4276×10 <sup>-1</sup> -	4.3258×10 <sup>-1</sup> -	1.7035-	5.2787×10 <sup>-1</sup> -	1.3366×10 <sup>-10</sup> -	<b>4.7539×10<sup>-12</sup></b>
	T2	1.1954×10-	3.6461×10 <sup>-1</sup> -	1.1642-	2.2216-	8.9943×10 <sup>-1</sup> -	<b>1.5063×10<sup>-5</sup>+</b>	2.5410×10 <sup>-4</sup>
NI+HS	T1	2.2728×10 <sup>2</sup> -	9.2666×10 <sup>2</sup> -	2.1523×10 <sup>2</sup> -	2.3360×10 <sup>2</sup> -	2.3369×10 <sup>2</sup> -	3.8562×10-	<b>3.6632×10</b>
	T2	1.7555×10 <sup>2</sup> -	4.4368×10 <sup>2</sup> -	2.6389×10 <sup>2</sup> -	1.7556×10 <sup>2</sup> -	2.3420×10 <sup>2</sup> -	4.4980-	<b>2.0858</b>
NI+MS	T1	3.7892×10 <sup>-2</sup> -	3.2071×10 <sup>-2</sup> -	1.4323×10 <sup>-2</sup> -	3.2548×10 <sup>-2</sup> -	3.6058×10 <sup>-2</sup> -	2.4653×10 <sup>-4</sup> -	<b>1.9437×10<sup>-13</sup></b>
	T2	1.8994×10-	3.1701×10-	4.7611-	1.2381×10-	1.5719×10-	1.4110-	<b>1.1673</b>
NI+LS	T1	2.9774×10 <sup>2</sup> -	2.1603×10 <sup>2</sup> -	3.8111×10 <sup>2</sup> -	2.8646×10 <sup>2</sup> -	2.1637×10 <sup>2</sup> -	1.4539×10 <sup>2</sup> -	<b>8.7224×10</b>
	T2	2.2125×10 <sup>3</sup> -	1.1545×10 <sup>4</sup> -	1.0338×10 <sup>3</sup> -	2.1925×10 <sup>3</sup> -	2.0939×10 <sup>3</sup> -	8.4312×10 <sup>2</sup> -	<b>1.0874×10<sup>2</sup></b>
+/-/=		0/18/0	0/18/0	0/18/0	0/18/0	0/18/0	5/11/2	

从图3的收敛曲线可以观察到,MTDE-MCT的收敛速度在对比的多任务优化算法中总体上是最好的,具体表现如下。

1) CI+LS为完全相交的低相似度问题,从图3(a)一图3(b)可以看出,MTDE-MCT分别在任务T1和任务T2上获得了最好的收敛性,其主要原因是运用迁移知识选取策略有

## 4.2 参数设置

实验中所有对比算法的参数设置如下。

1)种群大小设置为100,对于测试集中的每个问题,所有算法独立运行30次,为保证公平性,所有对比算法的特定参数设置都严格遵循原文献中的设置。

2)CEC2017基准问题上的最大函数评估次数 $FE_{max}=100000$ ,WCCI2020基准问题上的最大函数评估次数 $FE_{max}=200000$ 。

3)根据原文献参数设定和大量实验,最后取得MTDE-MCT中的 $\alpha=0.4$ ,种群削减的间隔代数 $g$ 设置为150代。

## 4.3 实验结果分析

### 4.3.1 CEC2017测试集上的结果及综合分析

表2列出了MTDE-MCT与其他6种算法在CEC2017测试问题上独立运行30次的结果,目标函数适应度的平均值被用来评价算法的性能。表2中,加粗数据表示每个任务的最优值,为了进行结果比较,符号“+”“-”和“=”分别表示对比算法优于、劣于和相似于MTDE-MCT算法。

从表2中的实验结果可知,MTDE-MCT在18个任务中的12个任务上表现出了比其他算法更好的性能,超过了任务总数的一半,特别是在NI+HS,NI+MS,NI+LS的所有测试任务中表现最佳。MTDE-ADKT在部分任务上与MTDE-MCT具有竞争性,这可能是由于MTDE-ADKT使用了基于无监督域对齐技术的知识迁移策略。因此,分析任务获得的最优解数量可知,MTDE-MCT在最优解数量的获取上优于对比算法MFEA,MFEA-II,EMEA,MEEA-AKT,MTEA-Sao和MTDE-ADKT。

效促进了低相似度任务中的知识正迁移。

2)PI+MS和PI+LS为部分相交的低相似度问题,从图3(c)一图3(d)可以看出,此时在统一搜索空间中,每个任务的全局最优值相互距离较远,导致优化难度显著增加。但MTDE-MCT仍表现出了较强的竞争性。

3)NI+MS为不相交问题,从图3(e)一图3(f)可以看

出, MTDE-MCT 击败了所有对比算法, 在不相交问题上均取得了最优的收敛性能, 进一步验证了本文提出机制的优越性。

现最佳, 在算法运行的整体进化过程中均具有良好的收敛速度。这是因为算法前期种群规模较大, 促进了算法的前期探索能力。算法在后期聚焦于开发性, 任务种群的个体与当前任务更加契合, 促进任务快速收敛。

从表 2 和图 3 可知, MTDE-MCT 算法在整体性能上表

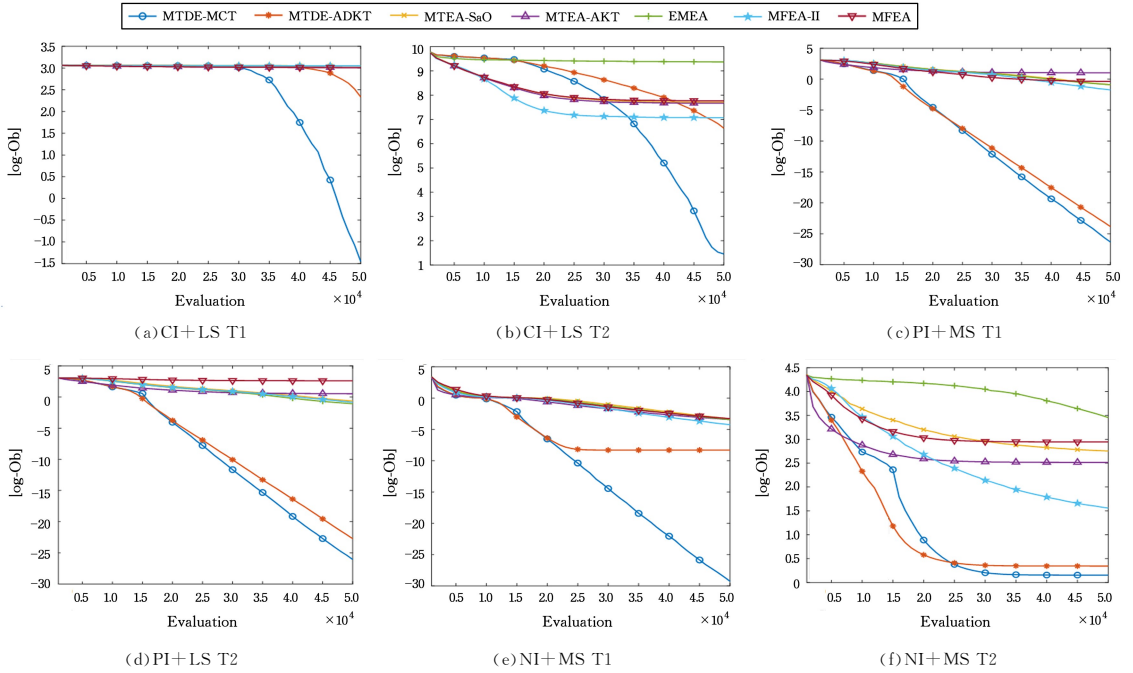


图 3 MTDE-MCT 和其余 6 种对比算法在部分任务中的收敛曲线

Fig. 3 Convergence curves of MTDE-MCT and six competing algorithms on selected benchmark problems

4.3.2 WCCI2020 测试集上的结果及综合分析

表 3 是 MTDE-MCT 与其他 6 种算法在 WCCI2020 测试问题上独立运行 30 次所获得的目标函数适应度平均值的对比结果。由表 3 的实验结果可知, MTDE-MCT 在总计 20 个任务中

的 11 个任务上表现最好, 超过了任务总数的一半。这表明 MTDE-MCT 的迁移知识选择策略在复杂的多任务测试集上表现良好, 说明了迁移知识选择策略和子群体对齐策略的有效结合, 使得 MTDE-MCT 在处理复杂问题时具有较强的竞争力。

表 3 MTDE-MCT 与其余 6 种对比算法在 WCCI2020 上的实验结果

Table 3 Experimental results of MTDE-MCT and six competing algorithms at WCCI2020

类别	任务	MFEA	EMEA	MFEA-II	MFEA-AKT	MTEA-SaO	MTDE-ADKT	MTDE-MCT
MTSO1	T1	6.4624 × 10 <sup>2</sup> -	6.1509 × 10 <sup>2</sup> -	6.0423 × 10 <sup>2</sup> -	6.1675 × 10 <sup>2</sup> -	6.2384 × 10 <sup>2</sup> -	<b>6.0085 × 10<sup>2</sup> +</b>	6.0223 × 10 <sup>2</sup>
	T2	6.4583 × 10 <sup>2</sup> -	6.0353 × 10 <sup>2</sup> =	6.0375 × 10 <sup>2</sup> =	6.1698 × 10 <sup>2</sup> -	6.2341 × 10 <sup>2</sup> -	<b>6.0085 × 10<sup>2</sup> +</b>	6.0363 × 10 <sup>2</sup>
MTSO2	T1	7.0001 × 10 <sup>2</sup> -	7.0000 × 10 <sup>2</sup> -	7.0000 × 10 <sup>2</sup> -	7.0001 × 10 <sup>2</sup> -	7.0000 × 10 <sup>2</sup> -	7.0000 × 10 <sup>2</sup> -	<b>7.0000 × 10<sup>2</sup></b>
	T2	7.0001 × 10 <sup>2</sup> -	7.0001 × 10 <sup>2</sup> -	7.0000 × 10 <sup>2</sup> +	7.0001 × 10 <sup>2</sup> -	7.0001 × 10 <sup>2</sup> -	<b>7.0000 × 10<sup>2</sup> +</b>	7.0000 × 10 <sup>2</sup>
MTSO3	T1	2.0507 × 10 <sup>6</sup> -	1.8919 × 10 <sup>6</sup> -	2.2583 × 10 <sup>6</sup> -	3.6552 × 10 <sup>5</sup> -	7.4529 × 10 <sup>5</sup> -	8.1831 × 10 <sup>3</sup> -	<b>6.7590 × 10<sup>3</sup></b>
	T2	2.4898 × 10 <sup>6</sup> -	5.3039 × 10 <sup>7</sup> -	1.5344 × 10 <sup>6</sup> -	4.2243 × 10 <sup>5</sup> -	5.8807 × 10 <sup>5</sup> -	7.7106 × 10 <sup>3</sup> =	1.1516 × 10 <sup>4</sup>
MTSO4	T1	1.3005 × 10 <sup>3</sup> -	1.3004 × 10 <sup>3</sup> -	1.3005 × 10 <sup>3</sup> -	1.3005 × 10 <sup>3</sup> -	1.3005 × 10 <sup>3</sup> -	1.3003 × 10 <sup>3</sup> =	<b>1.3003 × 10<sup>3</sup></b>
	T2	1.3005 × 10 <sup>3</sup> -	1.3006 × 10 <sup>3</sup> -	1.3004 × 10 <sup>3</sup> -	1.3004 × 10 <sup>3</sup> -	1.3005 × 10 <sup>3</sup> -	1.3003 × 10 <sup>3</sup> =	<b>1.3003 × 10<sup>3</sup></b>
MTSO5	T1	1.5362 × 10 <sup>3</sup> -	1.5124 × 10 <sup>3</sup> -	1.5268 × 10 <sup>3</sup> -	1.5282 × 10 <sup>3</sup> -	1.5121 × 10 <sup>3</sup> -	1.5100 × 10 <sup>3</sup> -	<b>1.5087 × 10<sup>3</sup></b>
	T2	1.5330 × 10 <sup>3</sup> -	1.5347 × 10 <sup>3</sup> -	1.5296 × 10 <sup>3</sup> -	1.5275 × 10 <sup>3</sup> -	1.5118 × 10 <sup>3</sup> =	1.5113 × 10 <sup>3</sup> =	<b>1.5108 × 10<sup>3</sup></b>
MTSO6	T1	1.5506 × 10 <sup>6</sup> -	1.1024 × 10 <sup>6</sup> -	1.2793 × 10 <sup>6</sup> -	1.2536 × 10 <sup>6</sup> -	4.4588 × 10 <sup>5</sup> -	1.0443 × 10 <sup>4</sup> =	<b>9.9640 × 10<sup>3</sup></b>
	T2	9.2307 × 10 <sup>5</sup> -	2.0156 × 10 <sup>7</sup> -	7.4272 × 10 <sup>5</sup> -	7.9839 × 10 <sup>5</sup> -	2.9706 × 10 <sup>5</sup> -	<b>8.5350 × 10<sup>3</sup> =</b>	1.2119 × 10 <sup>4</sup>
MTSO7	T1	3.2976 × 10 <sup>3</sup> -	3.0204 × 10 <sup>3</sup> -	2.8644 × 10 <sup>3</sup> -	3.0652 × 10 <sup>3</sup> -	3.0160 × 10 <sup>3</sup> -	<b>2.3577 × 10<sup>3</sup> +</b>	2.4414 × 10 <sup>3</sup>
	T2	3.2905 × 10 <sup>3</sup> -	4.4283 × 10 <sup>3</sup> -	3.0850 × 10 <sup>3</sup> -	3.0572 × 10 <sup>3</sup> -	3.1984 × 10 <sup>3</sup> -	2.4314 × 10 <sup>3</sup> -	<b>2.4280 × 10<sup>3</sup></b>
MTSO8	T1	5.2030 × 10 <sup>2</sup> +	5.2104 × 10 <sup>2</sup> -	5.2118 × 10 <sup>2</sup> -	<b>5.2027 × 10<sup>2</sup> +</b>	5.2118 × 10 <sup>2</sup> -	5.2078 × 10 <sup>2</sup> -	5.2066 × 10 <sup>2</sup>
	T2	<b>5.2027 × 10<sup>2</sup> +</b>	5.2118 × 10 <sup>2</sup> -	5.2118 × 10 <sup>2</sup> -	5.2029 × 10 <sup>2</sup> +	5.2117 × 10 <sup>2</sup> -	5.2078 × 10 <sup>2</sup> -	5.2047 × 10 <sup>2</sup>
MTSO9	T1	8.1691 × 10 <sup>3</sup> -	7.7682 × 10 <sup>3</sup> -	1.4240 × 10 <sup>4</sup> -	7.9946 × 10 <sup>3</sup> -	7.8251 × 10 <sup>3</sup> -	7.0405 × 10 <sup>3</sup> -	<b>6.5320 × 10<sup>3</sup></b>
	T2	1.6215 × 10 <sup>3</sup> -	1.6225 × 10 <sup>3</sup> -	1.6222 × 10 <sup>3</sup> -	1.6213 × 10 <sup>3</sup> -	1.6218 × 10 <sup>3</sup> -	1.6197 × 10 <sup>3</sup> =	<b>1.6194 × 10<sup>3</sup></b>
MTSO10	T1	3.5774 × 10 <sup>4</sup> -	2.4360 × 10 <sup>4</sup> -	2.9856 × 10 <sup>4</sup> -	3.0606 × 10 <sup>4</sup> -	1.4191 × 10 <sup>4</sup> -	2.6008 × 10 <sup>3</sup> -	<b>2.3187 × 10<sup>3</sup></b>
	T2	1.6760 × 10 <sup>6</sup> -	2.2154 × 10 <sup>7</sup> -	1.7552 × 10 <sup>6</sup> -	1.9508 × 10 <sup>6</sup> -	4.7103 × 10 <sup>5</sup> -	1.0384 × 10 <sup>4</sup> +	1.4222 × 10 <sup>4</sup>
		+/ - / =	2/18/0	0/19/1	1/18/1	2/18/0	0/19/1	5/7/8

4.4 策略有效性分析

为了探讨 MTDE-MCT 中每种策略对算法性能的影响,

在 CEC2017 测试集上对 MTDE-MCT 中的主要组成部分进行消融实验, 以证明这些策略的合理性。为此, 将 MTDE-

MCT 与 3 个变体进行比较;用随机选择迁移个体代替迁移知识选取策略,记为 MTDE-XZ;未使用子群体对齐策略,记为 MTDE-DQ;未使用多层种群削减策略,记为 MTDE-XJ。参数设置与上述实验保持一致,3 个变体均在每个问题上独立

运行 30 次。实验结果如表 4 所列。从表 4 的实验结果可知,MTDE-MCT 与其他 3 个变体相比始终表现出卓越的性能。这些结果证明了迁移知识选取策略和多层种群削减策略在增强多任务优化算法性能方面的有效性。

表 4 MTDE-MCT 与 3 个变体在 CEC2017 上的实验结果

Table 4 Experimental results of MTDE-MCT and its three variants on CEC2017

类别	任务	MTDE-XZ	MTDE-DQ	MTDE-XJ	MTDE-MCT
CI+HS	T1	$2.8866 \times 10^{-16} =$	$4.8110 \times 10^{-17} =$	<b>0.0000 =</b>	<b>0.0000</b>
	T2	$2.4237 \times 10^{-2} =$	$1.3901 \times 10^{-5} =$	$3.6711 \times 10^{-15} =$	$3.0731 \times 10^{-14}$
CI+MS	T1	$2.5648 \times 10^{-13} =$	$2.7756 \times 10^{-13} =$	$1.6099 \times 10^{-11} =$	<b><math>1.7240 \times 10^{-13}</math></b>
	T2	$1.3534 \times 10^{-5} =$	$7.1622 \times 10^{-4} =$	<b>0.0000 =</b>	$5.3197 \times 10^{-6}$
CI+LS	T1	$1.5269 \times 10^{-1} =$	$1.1747 =$	$1.6339 \times 10 =$	<b><math>7.6758 \times 10^{-2}</math></b>
	T2	$1.0245 =$	$2.0302 \times 10 =$	$9.2245 \times 10^2 =$	<b><math>4.0117 \times 10^{-1}</math></b>
PI+HS	T1	$8.7909 \times 10 =$	<b><math>7.6900 \times 10 +</math></b>	$1.4201 \times 10^2 =$	$8.3001 \times 10$
	T2	$1.5897 \times 10^{-11} =$	$1.8738 \times 10^{-2} =$	$9.7992 \times 10^{-11} =$	<b><math>1.2141 \times 10^{-11}</math></b>
PI+MS	T1	<b><math>2.0235 \times 10^{-12} =</math></b>	$4.5572 \times 10^{-12} =$	$5.3583 \times 10^{-11} =$	$3.2138 \times 10^{-12}$
	T2	$7.2735 \times 10 =$	$6.6585 \times 10 =$	<b><math>5.6681 \times 10 =</math></b>	$6.9972 \times 10$
PI+LS	T1	<b><math>5.9216 \times 10^{-12} =</math></b>	$1.1983 \times 10^{-11} =$	$1.5489 \times 10^{-10} =$	$8.8324 \times 10^{-12}$
	T2	$2.3207 \times 10^{-3} =$	$1.3738 \times 10^{-3} =$	<b><math>3.5450 \times 10^{-10} +</math></b>	$2.0433 \times 10^{-3}$
NI+HS	T1	$3.3260 \times 10 =$	$3.5776 \times 10 =$	$3.7713 \times 10 =$	<b><math>3.3255 \times 10</math></b>
	T2	$2.0151 =$	$3.3827 =$	$4.0189 =$	<b><math>6.6331 \times 10^{-2}</math></b>
NI+MS	T1	$8.1224 \times 10^{-14} =$	<b><math>7.2398 \times 10^{-14} =</math></b>	$4.0258 \times 10^{-13} =$	$1.9473 \times 10^{-13}$
	T2	$1.4797 =$	$3.6142 =$	$1.2872 =$	<b>1.0978</b>
NI+LS	T1	<b><math>7.3278 \times 10 +</math></b>	$8.8569 \times 10 =$	$1.4974 \times 10^2 =$	$8.9503 \times 10$
	T2	$7.7431 \times 10 =$	<b><math>7.5145 \times 10 =</math></b>	$9.6363 \times 10^2 =$	$8.9037 \times 10$
+/-/=		1/3/14	1/2/15	1/2/5	

图 4 可视化了 MTDE-MCT 和 3 个变体在 CEC2017 测试集中的收敛曲线。从图 4(a)可以看出,在处理 CI+HS T1 问题时,随着算法运行,具备知识迁移选取策略的 MTDE-MCT 在算法运行后期更占优势,这是因为测试问题 CI+HS 相似性较高,进化前期迁移知识的选择对算法性能的影响较小,随着进化次数递增,知识迁移选取策略逐渐显示其有效性。图 4(b)显示,基于任务间对齐 MTDE-DQ 和 MTDE-

MCT 在算法前中期具有类似的趋势,但在算法后期考虑迁移个体和目标任务子群体的 MTDE-MCT 显示了明显的优势,证明了子群体对齐策略的优势。从图 4(c)可以看出,多层种群削减策略有助于算法更快收敛到最优解,说明在算法的进化过程中,线性种群削减有助于集中优势探索未知的区域,加快优化任务的收敛速度。综合分析,MTDE-MCT 在提出策略的帮助下显著提升了任务优化效果,收敛速度更快。

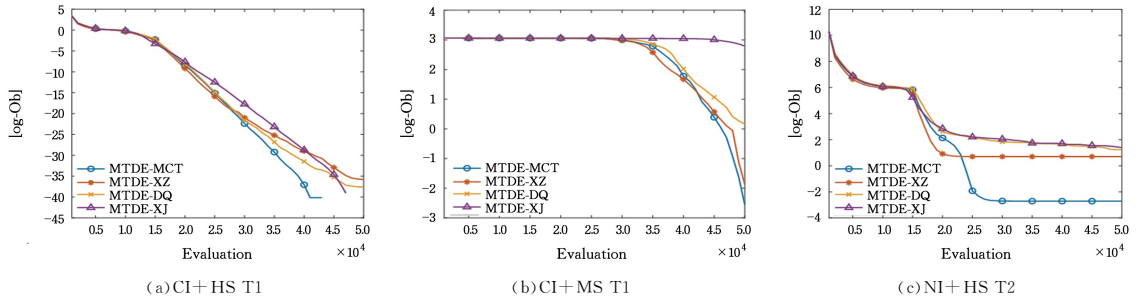


图 4 MTDE-MCT 和 3 个变体在 CEC2017 上的收敛曲线

Fig. 4 Convergence curves of MTDE-MCT and its three variants on CEC2017

#### 4.5 参数敏感性分析

##### 4.5.1 权重系数

在 MTDE-MCT 算法中,需要在迁移知识选择策略中设置曼哈顿距离和适应度值之间的权重系数,权重系数的设置进一步影响到选择迁移知识的优劣,因此设计了 5 种权重系

数不同的变体进行比较实验。

表 5 中的实验结果证明,权重系数  $\alpha$  的选择会影响算法的性能, $\alpha=0.4$  时的任务效果最好。通过有效选取迁移知识,进一步促进了任务之间的知识正迁移。该结果凸显了参数寻优在知识迁移过程中的重要作用。

表 5 MTDE-MCT 与 4 个变体在 CEC2017 上的实验结果

Table 5 Experimental results of MTDE-MCT and its four variants on CEC2017

类别	任务	$\alpha=0.1$	$\alpha=0.2$	$\alpha=0.3$	$\alpha=0.4$	$\alpha=0.5$
CI+HS	T1	$1.4904 \times 10^{-8} =$	$7.1942 \times 10^{-15} =$	$5.6899 \times 10^{-15} =$	<b>0.0000</b>	<b>0.0000 =</b>
	T2	$3.4471 \times 10^{-2} =$	$2.5685 \times 10^{-4} =$	$3.6175 \times 10^{-3} =$	<b>0.0000</b>	$4.4555 \times 10^{-11} =$
CI+MS	T1	$7.8562 \times 10^{-7} =$	$3.5365 \times 10^{-12} =$	<b><math>4.6638 \times 10^{-13} +</math></b>	$2.5971 \times 10^{-12}$	$5.6692 \times 10^{-13} +$
	T2	$4.7907 \times 10^{-1} =$	$2.1685 \times 10^{-3} =$	$1.0454 =$	<b><math>4.2789 \times 10^{-7}</math></b>	$8.9378 \times 10^{-4} =$
CI+LS	T1	$1.6124 \times 10 =$	$8.5560 =$	<b><math>5.2242 \times 10^{-3} +</math></b>	$6.0324$	$1.0407 +$
	T2	$6.9556 \times 10^2 =$	$1.9548 \times 10^2 =$	<b><math>1.0094 \times 10^{-2} +</math></b>	$2.5023 \times 10^2$	$3.0328 \times 10 +$

(续表)

类别	任务	$\alpha=0.1$	$\alpha=0.2$	$\alpha=0.3$	$\alpha=0.4$	$\alpha=0.5$
PI+HS	T1	$8.9309 \times 10=$	<b><math>8.0882 \times 10=</math></b>	$8.2064 \times 10=$	$8.4607 \times 10$	$8.4407 \times 10=$
	T2	$2.6028 \times 10^{-2}=$	$2.6937 \times 10^{-7}=$	$5.7365 \times 10^{-11}=$	<b><math>1.0169 \times 10^{-11}</math></b>	$3.3359 \times 10^{-11}=$
PI+MS	T1	$1.0348 \times 10^{-6}-$	$8.4085 \times 10^{-11}=$	$5.7638 \times 10^{-12}+$	$2.0880 \times 10^{-11}$	<b><math>5.1538 \times 10^{-12}=</math></b>
	T2	$8.4740 \times 10=$	$7.7052 \times 10-$	<b><math>6.8322 \times 10=</math></b>	$6.8554 \times 10$	$7.3858 \times 10=$
PI+LS	T1	$4.6185 \times 10^{-6}-$	$1.3106 \times 10^{-10}-$	$6.0238 \times 10^{-11}=$	$2.2480 \times 10^{-11}$	<b><math>8.1907 \times 10^{-12}=</math></b>
	T2	$3.7236 \times 10^{-2}-$	$1.9860 \times 10^{-2}-$	$2.0231 \times 10^{-3}-$	<b><math>1.8403 \times 10^{-13}</math></b>	$2.5112 \times 10^{-6}-$
NI+HS	T1	$4.1851 \times 10-$	$3.5695 \times 10=$	$3.4665 \times 10=$	<b><math>3.4611 \times 10</math></b>	$3.6106 \times 10=$
	T2	$1.1714 \times 10-$	<b><math>6.9503 \times 10^{-1}=</math></b>	$4.3287=$	$6.7315$	$5.1285-$
NI+MS	T1	$7.1044 \times 10^{-7}-$	$6.1605 \times 10^{-4}-$	$3.6630 \times 10^{-13}=$	<b><math>1.1066 \times 10^{-13}</math></b>	$3.5444 \times 10^{-13}=$
	T2	$4.3056=$	$2.1653=$	$1.6741=$	$1.3455$	<b><math>9.3806 \times 10^{-1}+</math></b>
NI+LS	T1	$6.3480 \times 10+$	<b><math>6.2241 \times 10-</math></b>	$6.6870 \times 10+$	$1.2714 \times 10^2$	$9.7500 \times 10+$
	T2	$8.8683 \times 10^2-$	$3.0794 \times 10^2-$	$1.1844 \times 10^2+$	$2.4117 \times 10^2$	<b><math>8.4861 \times 10+</math></b>
+/-/=		1/13/4	0/9/9	6/4/8	Base	6/14/8

## 4.5.2 种群削减间隔代数

在 MTDE-MCT 算法中,需要在多层种群削减策略中设置种群削减的间隔代数,以控制种群削减的进程,进一步高效地集中种群优势,故设计了 5 种间隔代数不同的

变体进行比较实验。

表 6 中的实验结果证明,种群削减间隔代数的选择会影响算法的性能,实验结果表明, $g=150$  时任务效果最好,能够在算法进化过程中有效集中任务种群的优势。

表 6 MTDE-MCT 与 4 个变体在 CEC2017 上的实验结果

Table 6 Experimental results of MTDE-MCT and its four variants on CEC2017

类别	任务	$g=50$	$g=100$	$g=150$	$g=200$	$g=250$
CI+HS	T1	$1.2315 \times 10^{-3}=$	<b><math>3.3307 \times 10^{-17}=</math></b>	$3.6980 \times 10^{-4}$	$5.906410 \times 10^{-15}+$	$4.363210 \times 10^{-15}=$
	T2	$6.3240=$	<b><math>1.9578 \times 10^{-5}=</math></b>	$2.7400$	$2.1611+$	$4.2584 \times 10^{-3}+$
CI+MS	T1	$8.2376 \times 10^{-1}-$	$3.2232 \times 10^{-13}=$	<b><math>2.9035 \times 10^{-13}</math></b>	$1.1290 \times 10^{-11}-$	$5.5896 \times 10^{-12}-$
	T2	$6.2971-$	<b><math>6.8175 \times 10^{-5}=</math></b>	$4.9192 \times 10^{-3}$	$1.0549-$	$1.6450 \times 10^{-4}=$
CI+LS	T1	$1.8182-$	$1.6894=$	<b><math>5.0495 \times 10^{-2}</math></b>	$3.4887-$	$1.4250-$
	T2	$3.9348 \times 10-$	$1.8878 \times 10=$	<b><math>3.3518 \times 10^{-1}</math></b>	$4.3916 \times 10-$	$2.2118 \times 10-$
PI+HS	T1	$1.0960 \times 10^2-$	$9.1514 \times 10=$	$8.8702 \times 10$	$8.6167 \times 10=$	<b><math>8.2809 \times 10=</math></b>
	T2	<b><math>3.5112 \times 10^{-12}+</math></b>	$6.5435-$	$2.0280 \times 10^{-11}$	$7.1872 \times 10^{-1}-$	$9.1252 \times 10^{-10}-$
PI+MS	T1	$6.1778 \times 10^{-1}-$	$4.3953 \times 10^{-2}=$	<b><math>2.8488 \times 10^{-12}</math></b>	$2.0912 \times 10^{-11}-$	$1.0804 \times 10^{-10}-$
	T2	$7.7696 \times 10=$	$7.8309 \times 10-$	<b><math>6.5214 \times 10</math></b>	$9.8342 \times 10-$	$6.6438 \times 10=$
PI+LS	T1	$1.5257-$	$4.3310 \times 10^{-1}=$	<b><math>6.1273 \times 10^{-12}</math></b>	$2.2133 \times 10^{-11}-$	$3.1384 \times 10^{-10}-$
	T2	$2.5848 \times 10^{-1}-$	$1.5313 \times 10^{-1}-$	<b><math>9.6277 \times 10^{-14}</math></b>	$2.0910 \times 10^{-1}-$	$6.3391 \times 10^{-3}-$
NI+HS	T1	$4.3310 \times 10=$	<b><math>3.3366 \times 10=</math></b>	$3.4035 \times 10$	$3.5607 \times 10-$	$3.5917 \times 10-$
	T2	$1.9331 \times 10=$	$2.7859=$	$6.4672 \times 10^{-1}$	$2.3596 \times 10-$	<b><math>9.9514 \times 10^{-2}+</math></b>
NI+MS	T1	$2.3415 \times 10^{-3}-$	<b><math>1.1896 \times 10^{-14}+</math></b>	$6.1617 \times 10^{-14}$	$7.3856 \times 10^{-13}-$	$4.6663 \times 10^{-12}-$
	T2	$1.2358=$	<b><math>1.1627=</math></b>	$1.4539$	$2.0916=$	$1.9050=$
NI+LS	T1	$1.2188 \times 10^2-$	$9.3343 \times 10=$	$8.5739 \times 10$	<b><math>7.9817 \times 10=</math></b>	$8.7595 \times 10=$
	T2	$8.7061 \times 10=$	$1.0304 \times 10^2=$	$1.2243 \times 10^2$	<b><math>4.2790 \times 10+</math></b>	$7.4242 \times 10=$
+/-/=		1/10/7	1/3/14	Base	3/12/3	2/9/7

**结束语** 本文提出了一种基于迁移知识选择和种群削减的进化多任务优化算法 MTDE-MCT,用于求解多任务优化问题。该算法将曼哈顿距离和适应度值作为联合指标进行迁移知识选取,同时兼顾个体的多样性和质量。其次利用子群体对齐策略消除迁移个体在不同任务间的差异,确保了知识的正迁移,最后利用多层种群削减策略进行线性的种群削减,提高了任务种群的优势。在 CEC2017 问题测试集和 WC-CI2020 问题测试集上与相应的 MTO 算法进行对比,实验结果验证了 MTDE-MCT 在多任务优化中的优越性能,尤其在低相似度任务中表现突出。

在未来的研究中,计划从以下两个方面展开工作。一方面,针对任务数量增多的情况,加入任务间相似性测量策略,设计更高效的多任务算法。另一方面,本文的种群削减策略目前仅基于固定进化代数进行削减,未来计划设计一种自适应削减机制,以更好地满足多任务优化算法的需求。

## 参 考 文 献

- [1] LIANG J, LIU R, ZHAI B Y, et al. Overview of the application of evolutionary Algorithms in Large-Scale Optimization Problems[J]. Journal of Zhengzhou University, 2018, 39(3): 15-21.
- [2] SONG Q L, CHE A D. Overview of the application of quantum evolutionary algorithms in production scheduling[J]. Computer Applications and Research, 2012, 29(5): 1601-1605.
- [3] SONG X B, GAO J W, ZHANG C X. Research on off-road vehicle path planning based on improved ant colony algorithm[J]. Computer Simulation, 2023, 40(10): 200-204, 325.
- [4] WANG Y, WANG Z G. Solving the multi-choice knapsack problem using differential evolution algorithm[J]. Science Technology and Engineering, 2011, 11(34): 8405-8408.
- [5] BACK T, HAMMEL U, SCHWEFEL H P. Evolutionary computation: Comments on the history and current state[J]. IEEE

- Transactions on Evolutionary Computation, 1997, 1(1):3-17.
- [6] ZHANG X, ZHANG Y, WANG W, et al. Transfer Boosting with Synthetic Instances for Class Imbalanced Object Recognition[J]. IEEE Transactions on Cybernetics, 2016(1):357-370.
- [7] GUPTA A, ONG Y, FENG L. Multifactorial Evolution: Toward Evolutionary Multitasking[J]. IEEE Transactions on Evolutionary Computation, 2016, 20(3):343-357.
- [8] MUHAMMAD I, BING X, HARITH S A, et al. Cross-Domain Reuse of Extracted Knowledge in Genetic Programming for Image Classification [J]. IEEE Transactions on Evolutionary Computation, 2017, 21(4):569-587.
- [9] LI G, ZHANG Q, GAO W. Multipopulation evolution framework for multifactorial optimization[C]//Proceedings of the Genetic and Evolutionary Computation Conference Companion, 2018:215-216.
- [10] FENG L, ZHOU W, ZHOU L, et al. An empirical study of multifactorial PSO and multifactorial DE[C]//2017 IEEE Congress on Evolutionary Computation(CEC). IEEE, 2017:921-928.
- [11] WU D R, TAN X F. Multitasking genetic algorithm (MTGA) for fuzzy system optimization[J]. IEEE Transactions on Fuzzy Systems, 2020, 28(6):1050-1061.
- [12] XUE X, ZHANG K, TAN K C, et al. Affine transformation-enhanced multifactorial optimization for heterogeneous problems [J]. IEEE Transactions on Cybernetics, 2020, 52(7):6217-6231.
- [13] MA X, ZHENG Y, ZHU Z, et al. Improving evolutionary multitasking optimization by leveraging inter-task gene similarity and mirror transformation [J]. IEEE Computational Intelligence Magazine, 2021, 16(4):38-53.
- [14] WANG C, LIU J, WU K, et al. Solving multitask optimization problems with adaptive knowledge transfer via anomaly detection [J]. IEEE Transactions on Evolutionary Computation, 2021, 26(2):304-318.
- [15] CHEN Y, ZHONG J, FENG L, et al. An adaptive archive-based evolutionary framework for many-task optimization [J]. IEEE Transactions on Emerging Topics in Computational Intelligence, 2019, 4(3):369-384.
- [16] ZHAO B, CUI Z, YANG J, et al. A multi-task evolutionary algorithm for solving the problem of transfer targets[J]. Information Sciences, 2024, 681:121214-121214.
- [17] CHEN K, XUE B, ZHANG M, et al. Evolutionary multitasking for feature selection in high-dimensional classification via particle swarm optimization[J]. IEEE Transactions on Evolutionary Computation, 2021, 26(3):446-460.
- [18] LIANG Z, DONG H, LIU C, et al. Evolutionary multitasking for multiobjective optimization with subspace alignment and adaptive differential evolution [J]. IEEE Transactions on Cybernetics, 2020, 52(4):2096-2109.
- [19] GAO W, CHENG J, GONG M, et al. Multiobjective multitasking optimization with subspace distribution alignment and decision variable transfer [J]. IEEE Transactions on Emerging Topics in Computational Intelligence, 2021, 6(4):818-827.
- [20] WANG R, FENG X, YU H. Contrastive variational auto-encoder driven convergence guidance in evolutionary multitasking [J]. Applied Soft Computing, 2024, 163:111883.
- [21] ZHANG T Y, GONG W Y, LI Y C. Multitask differential evolution with adaptive dual knowledge transfer [J]. Applied Soft Computing, 2024, 165:112040.
- [22] DA B, ONG Y S, FENG L, et al. Evolutionary multitasking for single-objective continuous optimization: Benchmark problems, performance metric, and baseline results[J]. arXiv:1706.03470, 2017.
- [23] BALI K K, ONG Y S, GUPTA A, et al. Multifactorial evolutionary algorithm with online transfer parameter estimation: MFEA-II[J]. IEEE Transactions on Evolutionary Computation, 2019, 24(1):69-83.
- [24] FENG L, ZHOU L, ZHONG J, et al. Evolutionary multitasking via explicit autoencoding [J]. IEEE Transactions on Cybernetics, 2018, 49(9):3457-3470.
- [25] ZHOU L, FENG L, TAN K C, et al. Toward adaptive knowledge transfer in multifactorial evolutionary computation [J]. IEEE Transactions on Cybernetics, 2020, 51(5):2563-2576.
- [26] LI Y, GONG W, LI S. Multitasking optimization via an adaptive solver multitasking evolutionary framework[J]. Information Sciences, 2023, 630:688-712.



**LI Erchao**, born in 1980, Ph.D, professor. His main research interests include intelligent optimization theory, methods, and applications, environmental perception, modeling, and control of intelligent robots, modeling and operational optimization of integrated energy systems.

(责任编辑:喻黎)