控制与决策
Control and Decision

一种基于超粒子引导的自适应知识迁移多任务差分进化算法

孙倩, 王磊, 徐庆征, 夏坤, 李薇

引用本文:
孙倩,王磊,徐庆征,夏坤,李薇. 一种基于超粒子引导的自适应知识迁移多任务差分进化算法[J]. 控制与决策, 2024, 39(1): 26–38.

在线阅读 View online: https://doi.org/10.13195/j.kzyjc.2022.0432

一种基于超粒子引导的自适应知识迁移多任务差分进化算法

孙 倩[1], 王 磊[1,2†], 徐庆征[3], 夏 坤[1], 李 薇[1]

(1. 西安理工大学 陕西省网络计算与安全技术重点实验室，西安 710048；
2. 陕西理工大学 陕西省工业自动化重点实验室，陕西 汉中 723001；
3. 国防科技大学 信息通信学院，武汉 430035)

摘 要: 针对传统多任务优化算法(MTEA)存在负向知识迁移、迁移算子效率低下等问题,提出一种基于超粒子引导的自适应知识迁移的多任务差分进化算法(SAKT_MFDE). 首先,通过任务之间的相似程度自适应地调节任务之间的交配概率,增大任务之间的正向迁移;其次,利用超粒子引导算法的搜索方向,进一步提升算法整体的优化效率;最后,通过多任务基准函数进行仿真实验来评价改进算法的寻优性能. 实验结果表明,所提出算法可以有效规避任务之间的负向迁移,提高相似度较低的任务组的优化性能.

关键词: 多任务进化算法；知识迁移；超粒子

中图分类号: TP273 **文献标志码:** A

DOI: 10.13195/j.kzyjc.2022.0432

引用格式: 孙倩,王磊,徐庆征,等. 一种基于超粒子引导的自适应知识迁移多任务差分进化算法[J]. 控制与决策, 2024, 39(1): 26-38.

A super-particle guided multifactorial differential evolution algorithm with adaptive knowledge transfer

SUN Qian[1], *WANG Lei*[1,2†], *XU Qing-zheng*[3], *XIA Kun*[1], *LI Wei*[1]

(1. Shaanxi Key Laboratory of Network Computing and Security Technology，Xi'an University of Technology，Xi'an 710048，China；2. Shaanxi Key Laboratory of Industrial Automation，Shaanxi University of Technology，Hanzhong 723001，China；3. College of Information and Communication，National University of Defense Technology，Wuhan 430035，China)

Abstract: Aiming at the problems of negative knowledge transfer and low efficiency of the traditional multi-tasking optimization algorithm (MTEA), a multi-tasking differential evolution algorithm based on super-particle guided adaptive knowledge transfer (SAKT_MFDE) is proposed. Firstly, the algorithm adaptively adjusts the mating probability between tasks by the similarity degree between tasks to increase the forward migration between tasks. Secondly, the super-particle is used to guide the search direction of the algorithm, which further improves the overall optimization efficiency of the algorithm. The optimization performance of the improved algorithm is evaluated by the simulation of the multi-task benchmark function. The experimental results show that the proposed algorithm can effectively avoid the negative migration between tasks and improve the optimization performance of the task group with low similarity.

Keywords: multi-tasking evolution algorithm；knowledge transfer；super-particle

0 引 言

进化算法是一种传统的启发式算法[1-2],在解决传统数学和优化方法解决不了的复杂优化问题上非常有优势,可以高效处理困难的工程问题. 但这类启发式算法处理优化问题需要较高的计算代价,算法运行一次只能够解决一个优化问题,忽略了多个优化问题之间有用信息的挖掘,算法的优化性能并不理想. 多任务进化算法的提出弥补了这一缺陷. Gupta等[3]提出了多任务进化算法(multi-tasking evolution algorithm, MTEA)[3],相较于传统进化算法,其主要优势在于运行一次可以解决多个任务,算法优化性能高,收敛速度快,近年来成为进化算法邻域的研究热

收稿日期: 2022-03-19；**录用日期:** 2022-08-24.
基金项目: 国家自然科学基金项目(62176146).
责任编委: 李少远.
†通讯作者. E-mail: wanglei_sut@163.com.

点,已成功被应用于作业车间调度[4]、选矿问题[5]、光伏模型的参数优化问题[6]和车辆路径规划问题[7],解决了云计算中大规模虚拟机布局的优化问题[8]以及多层网络的构建问题[9].

MTEA算法也能够灵活地与其他算法进行结合. Feng等[10]将DE和PSO算法与MTEA结合起来,改进DE和PSO的搜索算子,增强了传统多任务算法的性能. 此外,其他文献对不同任务之间的迁移方式也进行了改进. Tang等[11]提出一种自适应多因子粒子群优化算法,可以在不同的演化阶段自适应调节迁移概率. Cai等[12]提出了一种用于多任务差分进化的差分向量共享机制,目的是在不同的任务中捕获、共享和利用有用的知识,提高迁移效率. Zheng等[13]提出一种自我调节的MTEA,将知识转移的强度与任务之间的相关性结合起来. Li等[14]提出了一个显式的多种群进化框架迁移有效信息,从而有效减少负向信息传递. Wang等[15]提出一种新的多群MTEA算法,并设计了新的跨种群交叉方法避免种群漂移. 在其他工作中,Wang等[16]以分群的MTEA为基础,探究解变量次序对于多任务优化性能的影响,研究发现其没有显著影响. 针对异维问题中存在的负迁移现象,Zhang等[17]利用有用信息替换存在负向迁移的基因位,有效规避了负向迁移产生的影响,Xu等[18]提出了一种自适应的信息传递策略解决多目标优化问题. 但这些研究工作在提升MTEA的性能时并没有将迁移时机与迁移效率分开进行探究,因此这一方面的研究仍需改进和完善.

本文将迁移时机与迁移效率分开进行讨论,首先,多任务进化算法同时优化多个任务的核心在于如何将对目标任务有利的信息进行迁移,但并不是所有的迁移都是正向的. 基于此,提出一种自适应调整迁移概率的策略,能够最大程度地挖掘不同任务之间合适的迁移时机. 此策略根据源任务(被迁移的任务)与目标任务(迁移任务)之间的相似程度判断任务之间是否进行知识迁移. 当任务之间的相似程度较高时,源任务相对于目标任务的可利用信息较多,增加迁移概率可以充分利用辅助信息帮助目标任务跳出局部最优或者更加接近最优位置,反之任务之间的相似程度较低时,应该降低知识迁移概率,以防止产生负迁移现象. 其次,如何利用有利信息实现有效的知识迁移的方式,也是MTEA需要解决的问题. 本文提出一种新的差分迁移策略,采用在两个任务上表现都有优势的超粒子引导种群的搜索,利用弱迁移实现有利信息的强转移,加快算法的收敛速度. 主要贡献集中在以下两个方面: 1) 构建一种自适应知识迁移的方法动态控制不同任务之间的交叉概率,进而规避负迁移现象; 2) 设计一种基于超粒子引导的差分变异方式,进而充分利用有潜力的迁移信息.

1　多任务进化算法

多任务进化算法通过编解码的方式实现跨域任务个体之间的知识迁移. 种群中的个体被编码到统一空间表示,假设待优化的任务有K个,分别为T_1, $T_2,\dots,T_K$,问题对应的维度分别表示为$D_1, D_2,\dots,D_K$,那么统一搜索空间的维度为多任务优化环境中是最大的任务维度,即$D_{\text{mulfitasisspace}} = \max(D_j)$, $j \in (1,2,\dots,K)$. 初始化包含N个个体的种群P, $P = \{x_1, x_2,\dots,x_N\}$. 种群$x_i$个体初始化为rand(1, $D_{\text{Multitasitipace}}$),解码操作是指当个体$x_i$在任务$T_K$上评估时,将统一空间中表示的个体$x_i$通过$y_i = L_i + (U_i - L_i) \times x_i (i \in \{1,2,\dots,N\})$映射到某个实际优化任务的搜索空间,其中$U$、$L$分别为实际优化任务的上下界. 再计算每个个体在任务上的目标值,假设K个任务对应的目标函数分别为$f_1(x), f_2(x),\dots,f_K(x)$,则多任务优化的形式化定义为

$$\{x_1, x_2,\dots,x_K\} = \text{argmin}\{f_1(x), f_2(x),\dots,f_K(x)\}, \tag{1}$$

其中x_j为$f_j(x)$的最优解. MTEA 中每个个体都具有如下4个属性.

属性1 (因子开销)　每个个体x_i的因子开销ψ_j^i为对应任务T_j上的目标函数值或者由某种评价指标得到的值,一般定义为$\psi_j^i = \lambda \cdot \delta_j^i + f_j^i$. 其中: f_j^i、δ_j^i分别为x_i在T_j上的目标值和总的约束违反系数,λ为惩罚项. 当δ_j^i为0时,表示个体x_i的因子开销为x_i在任务T_j上的目标函数值.

属性2 (因子排名)　每个个体x_i的因子排名r_j^i是在给定任务T_j上根据个体因子开销的排名,如果$r_j^a = r_j^b$,则表示个体x_a和x_b在T_j上的因子开销相同.

属性3 (技能因子)　每个个体的技能因子代表个体最具优势的任务.

属性4 (标量适应度值)　每个个体x_i的标量适应度值φ_i是个体x_i在多任务环境中搜索能力的标量化,定义为$\varphi_i = 1/\min\limits_{j\in\{1,2,\dots,K\}}\{r_j^i\}$. 在环境选择时,标量适应度值作为个体存活到下一代的竞争力指标.

MTEA算法描述如下.

step 1: 初始化种群,随机产生N个个体组成初始种群P,初始化随机交配概率(random mating probability, RMP),计算种群P中每一个个体p_i的因

子排名r_i^k和技能因子τ_i.

step 2: 进入迭代,通过选型交配产生子代种群O,在当前种群中任意选择两个个体p_a、p_b作为父代,如果双亲的技能因子不同,且rand < RMP (rand为随机数),则双亲交叉产生两个子代个体o_a、o_b;若rand > RMP,则双亲独立地通过突变产生子代o_a、o_b. 双亲技能因子相同则直接交叉产生子代o_a、o_b.

step 3: 进行垂直文化传播,为子代o_c个体分配技能因子,子代o_c个体如果为双亲p_a、p_b交叉所产生,则继承任意双亲的技能因子,子代o_c个体如果为单亲突变产生,则技能因子与其单亲父代保持一致.

step 4: 合并父代种群P和子代种群O,更新合并种群中个体的标量适应度值和技能因子.

step 5: 选择前N个较优个体为下一代种群.

step 6: 判断是否符合结束条件,若符合则输出当前全局最优解的位置及其适应度值,若不符合则转至step 2进行下一轮迭代.

2 多任务差分进化算法

多任务差分进化算法将DE的搜索算子嵌入到MTEA中,基本框架与MTEA一致,但在选型交配时不再采用交叉算子产生子代,而是使用DE的差分策略. MFDE中的选型交配操作描述如下.

step 1: 产生一个0～1之间的随机数rand,有

$$o_{i,g} = x_{r1,g} + F \times (\tilde{x}_{r2,g} - \tilde{x}_{r3,g}), \tag{2}$$

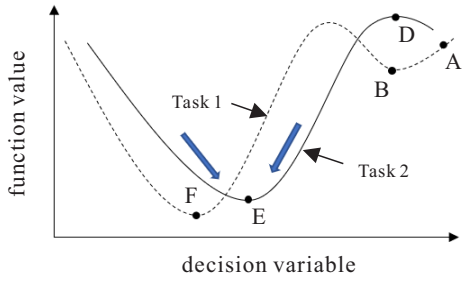
$$o_{i,g} = x_{r1,g} + F \times (x_{r2,g} - x_{r3,g}). \tag{3}$$

step 2: 若rand < RMP,则在满足迁移概率时,通过式(2)产生子代$o_{i,g}$,采用DE突变算子进行选型交配,选取其他任务上的个体构成差异向量,其中$\tilde{x}_{r2,g}$、$\tilde{x}_{r3,g}$为两个任意的技能因子不同于$x_{r1,g}$的个体. 当rand > RMP时,采用式(3) DE/rand/1生成子代个体,$x_{r2,g}$、$x_{r3,g}$个体的技能因子与$x_{r1,g}$相同.

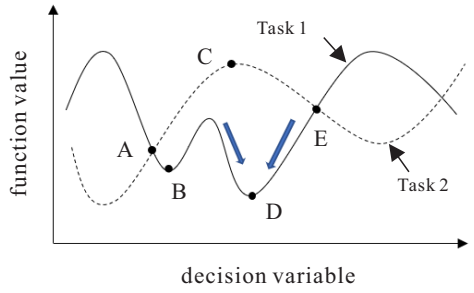
3 所提出算法

3.1 基于RMP的自适应知识迁移

多任务进化算法得益于利用不同任务之间潜在的互利信息相互促进演化,因此相比其他算法性能更优. 然而,任务之间的信息并不总是互利的, 如图1所示,假设搜索空间只有两个最小化优化任务,Task1为源任务,Task 2为目标任务. 图1(a)中,从A点开始搜索,Task 2上的搜索点A很容易陷入局部最优点B,如果可以将Task 1上点E的信息迁移到Task 2,则Task 2可以很快找到更优的点F. Task 1上大部分解对于Task 2都是有利的指导信息,因此当源任务中存在相对于目标任务有利信息较多时,任务之间的迁移概率应该设置为较大的值. 图2(b)的情况完全相反,只有点B～D之间的信息可以帮助Task 2找到最优点,其他区间的点反而会对Task 2产生消极影响. 例如A、C区间的点会导致Task 2陷入局部最优点B,在这种情况下应该减小任务之间的迁移概率,避免任务之间的负向迁移.



(a) 情况1



(b) 情况2

图 1 多任务优化环境中正向迁移和负向迁移

在原始的MTEA算法中,控制不同任务间的知识迁移参数由固定参数RMP决定,并没有根据任务间的有用信息动态调整. 文献[19]研究多任务之间相似度的衡量方法,提出了一种基于适应度地形[20]的测量方法(cross-task fitness distance correlation, CTFDC),即

$$\mathrm{CTFDC}_{T_1 \to T_2} = \frac{\dfrac{1}{n}\displaystyle\sum_{i=1}^{n}(f_1^i - \bar{f}_i)(d_2^i - \bar{d}_2)}{\sigma(f_1)\sigma(d_2)}. \tag{4}$$

其中:T_1、T_2分别为任务1和任务2;n为T_1任务上采样个体的数量;f_1为采样个体在T_1任务上评估的适应度值向量,f_1^i为第i个个体在任务T_1上的适应度值,$i \in [1, n]$;d_2为采样个体在T_2任务上评估的适应度值最好的个体x_{best}与所有采样个体之间的距离组成的向量,d_2^i为第i个个体与x_{best}之间的欧氏距离;$\overline{f_i}$、$\overline{d_2}$和$\sigma(f_1)$、$\sigma(d_2)$分别为f_1和d_2的均值和方差. 适应性地形分析方法采用全局视角衡量统一搜索空间中不同问题之间的相似程度,通过任务之间的相似程度动态调整RMP可以有效增强任务之间有用信息的迁移.

本文通过如下步骤实现基于RMP策略的自适应知识迁移.

step 1: 在源任务与目标任务之间随机选取种群中 1/4 个体,利用 CTFDC 并由式(4)计算任务间的相似性.

step 2: 利用任务之间的相似程度自适应调整 RMP. 若通过 step 1 测量得到任务之间的相似性为负值,表示任务之间的相似度较低,则 RMP 设置为 0,否则通过下式重新计算 RMP 的值:

$$\mathrm{RMP} = f_m(s) = a_1 \cdot s + b_1, \tag{5}$$

$$N_{x,y} = \frac{N_{\max} - N_{\min}}{O_{\max} - O_{\min}} \times (O_{x,y} - O_{\min}) + N_{\min}. \tag{6}$$

其中: a_1、b_2 分别为区间 $[0,1]$、$[0.3,1]$ 内由式(6)所示的区间映射关系得到的计算参数, $[N_{\min}, N_{\max}]$ 为目标区间, $[O_{\min},, O_{\max}]$ 为原始区间, $O_{x,y}$ 为原始区间内任意一点.

3.2　基于超粒子引导的差分迁移策略

MFDE 将 DE 的搜索算子与 MTEA 相结合,在知识迁移时简单地将差分向量替换为其他任务中的任意个体,其引导的搜索方向并不一定正确. 当满足迁移概率时,源任务与目标任务之间虽然具有较高的相似程度,但不能保证任意迁移到目标任务上的个体对目标任务优化时总是有益的,这种迁移策略并不能充分利用源任务上的有利信息,因此迁移效率并不高.

本文提出一种基于超粒子(super particle, Sp)引导的突变策略,此处超粒子是在目标任务上具有优势同时隐式地在源任务上也相对具有一定优势的个体. 这种迁移方式本质上是一种弱迁移,因为该方法并没有直接地将源任务上的优势个体迁移到目标任务上引导种群的搜索方向,而是将源任务的指导信息与目标任务上的指导信息结合起来,通过引导搜索方向减少负向迁移. 本文提出的基于超粒子引导的弱迁移突变策略如下所示:

$$o_{i,g} = x_{i,g} + F \times (x_{\mathrm{Sp}} - x_{i,g}) + F \times (\tilde{x}_{r2,g} - \tilde{x}_{r3,g}). \tag{7}$$

其中: x_{Sp} 为在源任务和目标任务上均表现较好的超粒子, $\tilde{x}_{r2,g}$、$\tilde{x}_{r3,g}$ 为源任务中选择的任意个体, F 为控制变异步长的缩放因子. 如图 2 所示, S_1 和 S_2 分别为目标任务和源任务的最优解. 目标任务和源任务具有较高的相似性,灰色阴影地带中的个体为目标任务上表现较优的个体,在其中选取源任务上表现较好的个体作为 x_{Sp},在 x_{Sp} 的引导下进行知识迁移,搜索得到更加接近 S_1 的 $o_{i,g}$.

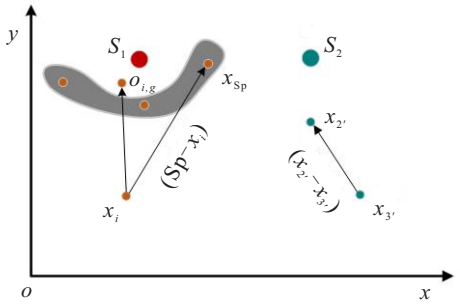


图 2　基于超粒子引导的迁移策略

3.3　SAKT_MFDE 算法

将自适应知识迁移策略与基于超粒子引导的弱迁移策略相结合,提出基于超粒子引导的自适应迁移的多任务差分进化算法 SAKT_MFDE (a super-pa1rticle guided multifactorial differential evolution algorithm with adaptive knowledge transfer, SAKT_MFDE),算法流程如图 3 所示, 具体描述如下.

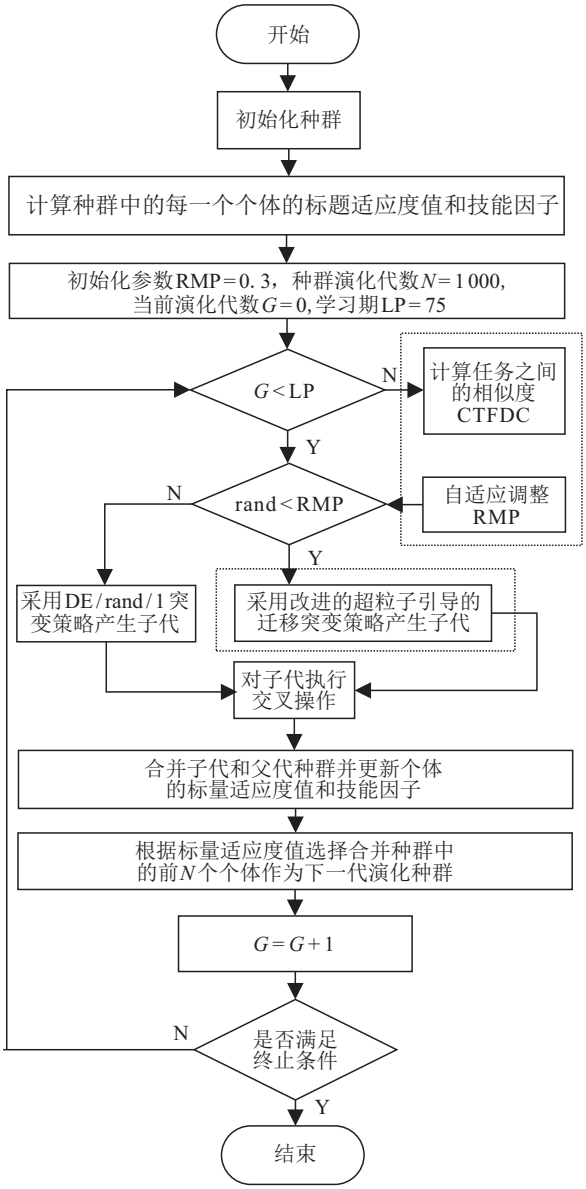


图 3　SAKT_MFDE 算法流程

step 1: 初始化种群:产生 N 个个体组成初始种群 P. 初始化参数:种群当前的演化代数 G、学习期 LP、随机交配概率 RMP、缩放因子 F. 计算种群中每个个

体p_i的因子排名r_i^k和技能因子τ_i.

step 2: 进入迭代,当$G > \mathrm{LP}$时采用3.1节所提出的自适应知识迁移策略CTFDC更新RMP.

step 3: 当rand $<$ RMP时采用3.2节提出的迁移策略产生子代,否则通过DE/rand/1产生子代.

step 4: 进行选型交配. 子代如果为超粒子引导的突变策略迁移产生,则子代任意继承源任务和目标任务的技能因子;子代如果由DE/rand/1产生,则直接继承父代的技能因子.

step 5: 合并父代种群P和子代种群O,并更新合并种群中个体的标量适应度值和技能因子.

step 6: 选择前N个较优个体为下一代种群.

step 7: 增加当前迭代次数$G = G + 1$,判断是否符合结束条件. 若符合,则输出当前全局最优解的位置及其适应度值;若不符合,则转至step 2进行下一轮迭代.

4 实验结果与分析

4.1 实验环境及参数配置

在多任务基准函数[21]上进行实验验证SAKT_MFDE算法的有效性. 如表1所示,多任务基准测试函数中包含9组多任务优化问题,测试问题由7个经典的单目标函数配对组成,分别是Sphere、Rosenbrock、Ackley、Weierstrass、Griewank、Rastrigin和Schwefel,其中Sphere为单峰函数,其他均为多模函数. 根据问题全局最优解的相交程度,可将多任务问题组分为完全相交(complete intersection, CI)、部分相交(partial intersection, PI)和无相交(no intersection, NI)三类. 此外,根据任务之间的相似性还可以将其分为高相似度(high similarty, HS)、中等相似度(middle similarty, MS)和低相似度(low similarty, LS)三类. 根据函数的分布特性和相交程度组成9组基础测试集. 实验环境为Intel(R) Core(TM) i7-4790 CPU,主频3.60 GHz和内存为4 GB的PC主机,操作系统为64位的Windows10,相关算法采用Matlab 2016a编写实现. 选取的对比算法包括基础多任务算法MFEA (multi-factorial evolution algorithm, MFEA)、MFEA_beta[3]、MFDE (multifactorial differential evolution algorithm, MFDE)以及改进的多任务算法MPEA[15] (factorial evolutionary algorithm as multi-population evolution mode, MPEA)、MFARR[22] (multifactorial evolutionary algorithm with resource reallocation, MEFARR)、MDE_DVSM[12] (multi-tasking differential evolution with difference vector sharing mechanism, MDE_DVSM)、SREMTO[13] (self-regulated evolutionary multi-tasking optimization algorithm, SREMTO)、MFEA_AKT (MFEA with adaptive knowledge transfer, MFEA_AKT)[23]. 其中:MPEA算法采用分群的方式保持不同任务的多样性;MFARR算法通过累计生存率检测迁移个体的搜索能力,自适应地调整任务之间的迁移概率;MDE_DVSM提出了一种差分共享机制以有效捕获多个任务之间的有利信息; SREMTO算法能够自适应不同任务间知识转移的强度;MFEA_AKT研究了交叉算子对知识迁移的影响,利用进化搜索过程中收集的信息引导交叉算子进行知识迁移. 这些对比算法都从提高迁移效率方面进行改进,与所提出算法具有较强的竞争性.

本文采用均值和方差评估算法的优化性和稳定性,方差结果在括号中表示. 均值越小表示算法性能越好,方差越小表示算法越稳定. 另外采用了排名方法,评价标准是先比较算法获得的均值,均值越小算法名次越好;在均值相同的情况下再比较方差,方差越小算法名次越好. 结果中“Count”表示排名为第1的总次数,“Ave Rank”表示平均排名情况,“Total Rank”表示在“Ave Rank”基础上的总排名情况,最优者用加粗表示.

为了保证结果的公平性,所有实验种群规模Np为100,演化代数为1 000,独立运行20次. 此外,所提出SAKT_MFDE和其他算法的参数设置与原始算法保持一致.

4.2 学习期对自适应知识迁移策略的影响

多任务进化算法随机初始化种群,导致在演化初期种群搜索空间的分布并不具有问题的特征. 因此,SAKT_MFDE算法在演化初期设置了学习期. 学习期内种群之间不进行任何信息交流,即RMP为0,各项任务独立演化. 经过学习期的演化后,种群分布能够较好地体现问题的特征,CTFDC能够准确地反映两个任务之间的相似程度. 然后通过自适应迁移策略根据任务之间的相似程度动态调节RMP,提高多个任务之间的正向迁移.

自适应迁移策略对学习期LP的设置非常敏感,学习期太短CTFDC测量的相似度不准确,学习期太长会降低自适应迁移策略对算法性能的提升效果. 本文在MFDE的基础上结合所提出的自适应知识迁移策略,将LP进行线性增加,运行20次得到均值(方差)见表2. 由表2可直观看出,学习期对自适应知识迁移策略的性能有所影响. 当LP为50代时,算法在7个问题上均优于LP设置为100代、150代、200代和250代,算法性能显著提升. 当LP超过100代时,

表 1　进化多任务优化问题的性质

类目	函数	分布特性	相交程度	任务间相似性
CI+HS	Griewank(T_1)	多模,不可分	完全相交	1.000
	Rastrigin(T_2)	多模,不可分		
CI+MS	Ackley(T_1)	多模,不可分	完全相交	0.226 1
	Schwefel(T_2)	多模,不可分		
CI+LS	Ackley(T_1)	多模,不可分	完全相交	0.002
	Rastrigin(T_2)	多模,可分		
PI+HS	Rastrigin(T_1)	多模,不可分	部分相交	0.867 0
	Sphere(T_2)	单峰,可分		
PI+MS	Ackley(T_1)	多模,不可分	部分相交	0.215 4
	Rosenbrock(T_2)	多模,不可分		
PI+LS	Ackley(T_1)	多模,不可分	部分相交	0.072 5
	Weierstrass(T_2)	多模,不可分		
NI+HS	Rosenbrock(T_1)	多模,不可分	不相交	0.943 4
	Rastrigin(T_2)	多模,不可分		
NI+MS	Griewank(T_1)	多模,不可分	不相交	0.366 9
	Weierstrass(T_2)	多模,不可分		
NI+LS	Rastrigin(T_1)	多模,不可分	不相交	0.001 6
	Schwefel(T_2)	多模,可分		

算法性能开始下降. 这表明当学习期充足、种群分布形态趋于成熟时,自适应迁移策略对算法的性能是显著提高的. 但是,当学习期饱和时,任务之间一直处于独立搜索的状态并不进行信息交流,导致异源有效信息流失,自适应的知识迁移并没有发挥积极的效应. 因此,合理设置LP有利于自适应迁移策略提高多任务优化之间的正向迁移. 通过实验探究,LP设置为50代时算法的性能处于上升阶段,100代时性能慢慢下降. 因此,在后续实验中将LP设置为75代,种群的分布充分体现问题的特征,同时最大程度地保留不同任务之间的有利信息.

4.3　自适应知识迁移策略的普适性探究

原始MTEA的主要缺陷是不能自适应地调节知识迁移. 针对该问题提出自适应知识迁移策略(adaptive knowledge transfer, AKT),能够有效捕捉任务间有利信息较多的敏感时间带,提高有利信息的利用率,从而提高多任务优化算法的效率. 为了验证自适应知识迁移策略的普适性,分别在相关基础多任务进化算法上添加自适应知识迁移策略,并与原始算法进行比较,结果见表3. 如表3所示,AKT_MFEA_alpha、AKT_MFEA_beta、AKT_MFDE分别为AKT策略与MFEA_alpha、MFEA_beta、MFDE结合后的优化算法. 实验结果表明,改进后的算法性能得到了不同程度的提升,其中AKT_MFEA_alpha算法在14个问题上比原算法更具优势,AKT_MFEA_beta算法在13个问题上比原算法性能更好,AKT_MFDE算法在16个问题上比原算法更接近最优值. 上述结果表明,所提出的AKT策略对多任务进化算法具有普适性.

4.4　AKT策略对相似度较低的多任务组的影响

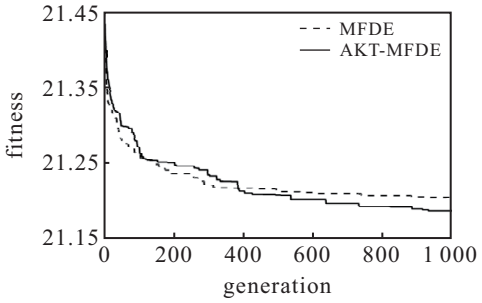
为了验证AKT策略可以提高多任务优化的正向迁移同时规避负向迁移,对AKT_MFDE算法和MFDE算法在相似度较低的多任务组收敛性方面进行对比分析. 图4～图6分别为MFDE和AKT_MFDE在多任务组CI+LS、PI+LS、NI+LS上运行20次每一代均值的收敛曲线.

表 2 学习期对自适应知识迁移策略的影响

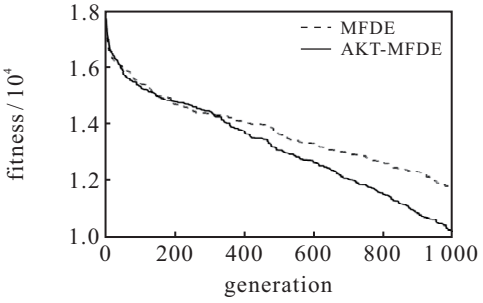
组别	任务	评价指标	LP = 0	LP = 50	LP = 100	LP = 150	LP = 200
			ATS_MFDE (MFDE with adaptive transfer strategy)				
CI+HS	T_1	Mean	1.501 9e-03	6.129 6e-04	**2.759 7e-05**	3.922 9e-04	2.510 4e-03
		Std	(3.456 8e-03)	(2.225 0e-03)	(2.654 2e-05)	(1.652 2e-03)	(4.805 4e-03)
		Rank	4	3	1	2	5
	T_2	Mean	5.218 6e+00	1.954 8e+00	**4.742 8e-02**	1.377 8e+00	8.666 4e+00
		Std	(1.280 8e+01)	(7.515 9e+00)	(4.860 4e-02)	(5.999 1e+00)	(1.589 2e+01)
		Rank	4	3	1	2	5
CI+MS	T_1	Mean	5.991 5e-02	**2.008 4e-03**	2.094 0e-01	1.671 4e-01	1.317 7e-01
		Std	(1.975 9e-01)	(1.357 9e-03)	(4.360 3e-01)	(4.078 7e-01)	(3.716 5e-01)
		Rank	2	1	5	4	3
	T_2	Mean	4.765 8e-01	**3.272 5e-03**	2.938 3e+00	1.099 3e+00	8.218 4e-01
		Std	(1.313 0e+00)	(4.692 3e-03)	(8.339 4e+00)	(3.830 5e+00)	(2.201 9e+00)
		Rank	2	1	5	4	3
CI+LS	T_1	Mean	**2.118 9e+01**	2.120 1e+01	2.120 9e+01	2.119 7e+01	2.120 7e+01
		Std	(4.568 3e-02)	(4.167 7e-02)	(4.102 0e-02)	(5.045 5e-02)	(3.585 5e-02)
		Rank	1	5	3	4	2
	T_2	Mean	1.105 0e+04	1.082 0e+04	1.040 4e+04	1.074 2e+04	**1.004 9e+04**
		Std	(1.543 7e+03)	(1.775 9e+03)	(1.501 9e+03)	(1.485 5e+03)	(2.074 6e+03)
		Rank	5	4	2	3	1
PI+HS	T_1	Mean	**8.378 9e+01**	8.971 1e+01	8.575 3e+01	8.456 4e+01	1.022 5e+02
		Std	(1.759 5e+01)	(6.592 9e+01)	(2.382 5e+01)	(2.160 8e+01)	(2.413 7e+01)
		Rank	1	4	3	2	5
	T_2	Mean	4.660 8e-05	3.084 9e-05	8.997 4e-05	1.206 5e-05	**7.029 3e-06**
		Std	(6.848 5e-05)	(3.959 8e-05)	(2.911 6e-04)	(2.345 0e-05)	(6.436 7e-06)
		Rank	4	3	5	2	1
PI+MS	T_1	Mean	4.080 9e-03	3.779 3e-03	2.670 3e-03	**2.653 5e-03**	3.413 2e-03
		Std	(1.413 5e-03)	(2.008 0e-03)	(8.781 9e-04)	(1.102 9e-03)	(3.023 7e-03)
		Rank	5	4	2	1	3
	T_2	Mean	**6.819 4e+01**	7.357 7e+01	7.235 4e+01	7.832 5e+01	8.196 7e+01
		Std	(2.016 2e+01)	(2.061 9e+01)	(2.384 7e+01)	(2.167 5e+01)	(2.075 5e+01)
		Rank	1	2	3	4	5
PI+LS	T_1	Mean	4.657 8e-01	5.302 6e-01	**3.639 4e-01**	4.356 3e-01	3.829 0e-01
		Std	(5.816 9e-01)	(6.336 5e-01)	(4.957 3e-01)	(5.063 3e-01)	(6.134 4e-01)
		Rank	4	5	1	3	2
	T_2	Mean	1.808 4e-01	**4.943 8e-02**	9.876 0e-02	1.988 2e-01	1.706 5e-01
		Std	(3.165 3e-01)	(4.902 4e-02)	(1.315 5e-01)	(3.526 9e-01)	(2.856 9e-01)
		Rank	4	1	2	5	3
NI+HS	T_1	Mean	8.670 9e+01	**6.112 3e+01**	8.143 4e+01	9.098 3e+01	8.571 6e+01
		Std	(3.508 3e+01)	(2.812 7e+01)	(3.751 9e+01)	(4.963 8e+01)	(4.268 7e+01)
		Rank	4	1	2	5	3
	T_2	Mean	2.403 1e+01	**1.566 0e+01**	2.599 0e+01	2.122 1e+01	2.093 4e+01
		Std	(1.589 5e+01)	(1.461 2e+01)	(1.563 7e+01)	(1.648 2e+01)	(1.571 8e+01)
		Rank	4	1	5	3	2
NI+MS	T_1	Mean	1.478 7e-03	**5.489 1e-05**	2.750 4e-03	1.764 2e-03	2.642 6e-03
		Std	(4.622 9e-03)	(4.099 9e-05)	(5.703 1e-03)	(3.680 2e-03)	(4.838 2e-03)
		Rank	2	1	5	3	4
	T_2	Mean	**2.675 4e+00**	2.720 8e+00	3.583 3e+00	4.761 2e+00	7.240 9e+00
		Std	(1.106 0e+00)	(1.201 9e+00)	(8.869 6e-01)	(1.617 2e+00)	(1.944 1e+00)
		Rank	1	2	3	4	5
NI+LS	T_1	Mean	1.510 9e+02	**1.138 3e+02**	1.659 3e+02	2.140 4e+02	2.013 9e+02
		Std	(1.008 6e+02)	(6.503 6e+01)	(1.026 9e+02)	(1.156 3e+02)	(1.183 4e+02)
		Rank	2	1	3	5	4
	T_2	Mean	**4.203 7e+03**	4.460 6e+03	4.511 5e+03	4.424 8e+03	4.817 4e+03
		Std	(6.666 9e+02)	(8.565 9e+02)	(7.985 1e+02)	(6.684 3e+02)	(8.179 4e+02)
		Rank	1	3	4	2	5
Count			**5**	**7**	**4**	**1**	**1**
Ave Rank			**2.83**	**2.5**	**3.05**	**3.2**	**3.39**
Total Rank			**2**	**1**	**3**	**4**	**5**

表 3　自适应知识迁移策略的普适性探究

组别	任务	评价指标	MFEA_alpha	AKT_MFEA_alpha	MFEA_beta	AKT_MFEA_beta	MFDE	AKT_MFDE
CI+HS	T_1	Mean	1.074 6e+00	**1.072 8e+00**	6.581 8e-01	**5.598 7e-01**	3.899 4e-04	**5.596 3e-05**
	T_1	Std	(2.680 3e-02)	(1.629 4e-02)	(1.235 6e-01)	(1.391 4e-01)	(1.663 6e-03)	(8.705 4e-05)
	T_2	Mean	3.313 6e+02	**3.249 7e+02**	2.325 0e+02	**2.074 0e+02**	1.439 7e+00	**1.090 7e-01**
	T_2	Std	(7.945 6e+01)	(3.365 1e+01)	(4.634 7e+01)	(5.055 6e+01)	(6.231 0e+00)	(1.824 5e-01)
CI+MS	T_1	Mean	8.180 8e+00	**7.897 0e+00**	2.938 7e+00	**3.111 7e+00**	1.899 1e-01	**3.201 6e-02**
	T_1	Std	(7.394 8e-01)	(7.866 1e-01)	(4.284 7e-01)	(3.930 4e-01)	(4.794 4e-01)	(1.963 9e+03)
	T_2	Mean	4.720 7e+02	**4.563 4e+02**	**2.194 1e+02**	2.222 9e+02	2.248 9e+00	**4.529 3e-01**
	T_2	Std	(9.664 2e+01)	(8.530 0e+01)	(6.219 2e+01)	(4.473 7e+01)	(7.428 1e+00)	(1.398 5e+00)
CI+LS	T_1	Mean	2.113 5e+01	**2.111 3e+01**	**2.026 9e+01**	2.120 6e+01	2.120 6e+01	**2.118 9e+01**
	T_1	Std	(8.734 7e-02)	(9.031 1e-02)	(4.226 3e+00)	(3.368 2e-02)	(4.066 8e-02)	(5.470 6e-02)
	T_2	Mean	9.779 1e+03	**9.724 5e+03**	3.757 1e+03	**3.757 0e+03**	1.175 0e+04	**1.023 6e+04**
	T_2	Std	(1.113 7e+03)	(1.068 5e+03)	(1.232 5e+03)	(9.662 9e+02)	(1.217 5e+03)	(1.963 9e+03)
PI+HS	T_1	Mean	7.932 4e+02	**7.468 5e+02**	3.544 6e+02	**3.463 7e+02**	**7.410 0e+01**	8.372 5e+01
	T_1	Std	(7.965 5e+01)	(8.362 4e+01)	(5.997 7e+01)	(4.850 3e+01)	(1.668 0e+01)	(1.838 7e+01)
	T_2	Mean	2.183 0e+02	**2.149 1e+02**	1.273 2e+02	**1.223 1e+02**	2.713 4e-05	**2.607 1e-05**
	T_2	Std	(1.005 7e+02)	(5.793 6e+01)	(5.841 2e+01)	(5.744 0e+01)	(2.918 9e-05)	(4.501 0e-05)
PI+MS	T_1	Mean	6.975 9e+00	**5.793 6e+01**	2.878 9e+00	**2.860 4e+00**	**1.363 5e-03**	2.705 2e-03
	T_1	Std	(6.797 5e-01)	(4.683 9e-01)	(4.495 2e-01)	(6.042 1e-01)	(4.676 6e-04)	(1.323 8e-03)
	T_2	Mean	**5.263 0e+04**	7.187 8e+04	**3.710 7e+03**	4.086 4e+03	8.203 1e+01	**7.200 9e+01**
	T_2	Std	(1.937 2e+04)	(3.810 4e+04)	(4.030 5e+03)	(5.260 5e+03)	(1.659 0e+01)	(2.335 0e+01)
PI+LS	T_1	Mean	**2.093 3e+01**	2.096 6e+01	4.624 5e+00	**4.515 5e+00**	4.429 7e-01	**4.358 7e-01**
	T_1	Std	(1.259 1e-01)	(1.373 1e-01)	(7.338 0e-01)	(6.547 1e-01)	(5.530 4e-01)	(5.295 6e-01)
	T_2	Mean	2.114 2e+01	**2.106 1e+01**	1.235 7e+01	**9.323 1e+00**	2.028 2e-01	**1.462 0e-01**
	T_2	Std	(2.488 2e+00)	(2.292 2e+00)	(5.136 3e+00)	(5.471 8e+00)	(3.542 6e-01)	(2.123 5e-01)
NI+HS	T_1	Mean	7.425 2e+04	**7.134 6e+04**	4.269 3e+03	**4.171 0e+03**	9.343 4e+01	**7.985 5e+01**
	T_1	Std	(3.627 1e+04)	(4.150 1e+04)	(3.083 4e+03)	(3.851 9e+03)	(3.900 8e+01)	(3.794 0e+01)
	T_2	Mean	**4.357 9e+02**	4.471 8e+02	**2.610 1e+02**	2.665 5e+02	2.608 4e+01	**2.450 6e+01**
	T_2	Std	(4.757 2e+01)	(1.309 7e+02)	(4.972 8e+01)	(4.279 5e+01)	(1.192 6e+01)	(1.878 3e+01)
NI+MS	T_1	Mean	1.037 4e+00	**1.035 3e+00**	**9.707 3e-01**	9.978 6e-01	4.275 6e-04	**1.899 1e-03**
	T_1	Std	(3.277 0e-02)	(3.244 6e-02)	(1.118 9e-01)	(8.027 3e-02)	(1.662 6e-03)	(4.508 9e-03)
	T_2	Mean	2.756 3e+01	**2.732 6e+01**	1.877 7e+01	**1.842 7e+01**	3.548 2e+00	**3.228 8e+00**
	T_2	Std	(2.661 5e+00)	(3.028 0e+00)	(9.903 5e+00)	(9.425 9e+00)	(1.396 5e+00)	(1.189 6e+00)
NI+LS	T_1	Mean	7.962 5e+02	**7.914 3e+02**	3.811 7e+02	**3.576 1e+02**	**1.023 7e+02**	1.445 5e+02
	T_1	Std	(9.460 1e+01)	(1.071 5e+02)	(4.809 6e+01)	(5.219 6e+01)	(2.034 5e+01)	(1.145 4e+02)
	T_2	Mean	**9.736 5e+03**	9.754 7e+03	3.659 3e+03	**3.481 3e+03**	4.262 0e+03	**4.207 9e+03**
	T_2	Std	(6.739 7e+02)	(8.134 9e+02)	(7.727 8e+02)	(8.143 1e+02)	(1.081 9e+03)	(8.855 7e+02)
Count			**4**	**14**	**5**	**13**	**3**	**16**

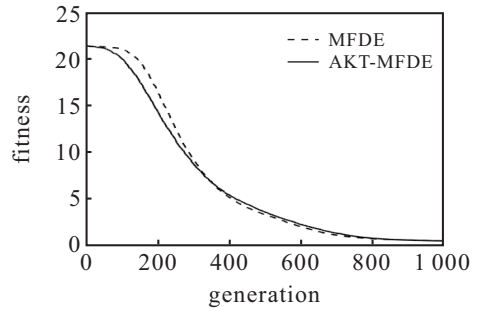


(a) T_1 in CI+LS

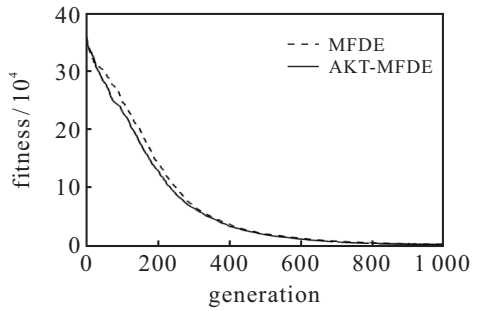


(b) T_2 in CI+LS

图 4 MFDE和AKT_MFDE在CI+LS多任务组的收敛性



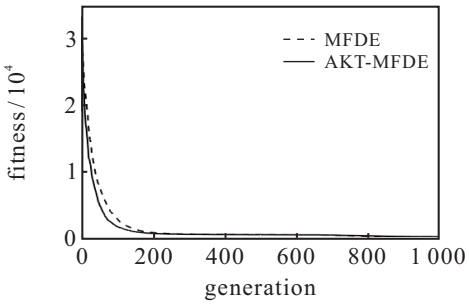
(a) T_1 in PI+LS



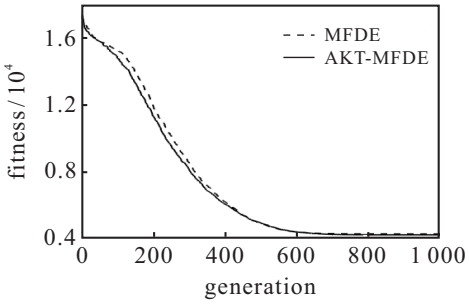
(b) T_2 in PI+LS

图 5 MFDE和AKT_MFDE在PI+LS多任务组的收敛性

如图4所示,在最优值完全相交的CI+LS中,即使任务间的相似度比较低,通过AKT策略也能有效规避负迁移,CI+LS任务组的两个优化问题都得到提升. 如图5和图6所示,在最优解不完全相交和完全不相交的PI+LS、NI+LS中,加入AKT策略后优化性能的提升并没有CI+LS显著. 这是因为在低相似度的情况下,任务之间的最优值也有很大差异,AKT策略捕捉到的互益信息较少,任务之间的知识迁移相对



(a) T_1 in NI+LS



(b) T_2 in NI+LS

图 6 MFDE和AKT_MFDE在NI+LS多任务组的收敛性

较少. 总体而言,AKT_MFDE算法比MFDE算法的曲线更凹,收敛性能更优,表明所提出自适应迁移策略通过任务间相似度的计算有效地减少了相似度较低任务中产生的负向迁移,进而提高了多任务进化算法在相似度较低任务组中的优化性能.

4.5 SAKT_MFDE与其他多任务进化算法性能对比

为了进一步探究SAKT_MFDE的优化性能,将SAKT_MFDE与具有代表性的MTEA算法变体进行对比. 5种算法变体分别为MFEA_AKT、 MPEA、MFARR、MDE_DVSM和SREMTO,数值实验结果见表4. 从均值和方差的角度分析,SAKT_MFDE算法在18个优化问题中11次优于其他5个多任务进化算法. 其中SAKT_MFDE算法在15个优化问题上优于MFEA_AKT算法,在16个优化问题上优于MPEA算法,在15个优化问题上优于MFARR算法,在12个优化问题上优于MDE_DVSM算法,在14个优化问题上优于SREMTO算法.

上述6个对比算法中,SAKT_MFDE算法的平均排名为1.95,MPEA算法、MFARR算法、MDE_DVSM算法、SREMTO算法和MFEA_AKT算法分别为4.67、5.45、2.28、2.94和3.78,总名次分别为5、6、2、3和4. SAKT_MFDE算法的排名位列第一,这表示在6种多任务进化算法中,SAKT_MFDE算法的优化性能最好,充分说明所提出算法是有效的. 在5%显著性水平下,Wilcoxon秩检验结果如表5所示,SAKT_MFDE明显优于MPEA、 MFARR、 MFEA_AKT,与MDE_DVSM、SREMTO无明显差异.

表 4　SAKT_MFDE与5个多任务算法的实验结果对比

组别	任务	评价指标	MPEA	MFARR	MDE_DVSM	SREMTO	MFEA_AKT	SAKT_MFDE
CI+HS	T_1	Mean	7.921 1e-01	9.529 2e-01	1.0189e-03	1.3579e-02	2.814 2e-01	**2.5099e-08**
		Std	(9.003 2e-02)	(1.631 2e-01)	(2.538 0e-03)	(1.253 1e-02)	(5.016 2e-02)	(5.567 7e-08)
		Rank	5	6	2	3	4	1
	T_2	Mean	3.831 5e+02	2.836 8e+02	5.082 4e+00	3.769 4e+01	1.972 8e+02	**4.498 0e-05**
		Std	(1.667 9e+01)	(7.628 5e+01)	(1.512 5e+01)	(2.986 3e+01)	(4.030 9e+01)	(9.731 2e-05)
		Rank	6	5	3	2	4	1
CI+MS	T_1	Mean	3.832 3e+00	7.678 8e+00	3.343 4e-02	4.173 1e+00	4.780 8e+00	**7.768 2e-03**
		Std	(3.961 5e-01)	(8.251 3e-01)	(8.397 3e-02)	(8.658 0e-01)	(7.179 4e-01)	(1.472 7e-02)
		Rank	3	6	2	4	5	1
	T_2	Mean	3.983 8e+02	4.086 0e+02	2.594 5e-01	8.928 7e+01	2.305 3e+02	**1.279 3e-01**
		Std	(2.356 2e+01)	(8.342 3e+01)	(6.074 3e-01)	(3.699 8e+01)	(6.692 7e+01)	(3.689 3e-01)
		Rank	5	6	2	3	4	1
CI+LS	T_1	Mean	2.121 7e+01	2.075 4e+01	2.120 4e+01	2.119 7e+01	**2.018 6e+01**	2.121 1e+01
		Std	(4.526 0e-02)	(1.274 8e-01)	(4.293 8e-02)	(3.612 0e-02)	(8.980 6e-02)	(3.528 8e-02)
		Rank	6	2	4	3	1	5
	T_2	Mean	1.418 8e+04	9.158 0e+03	5.886 8e+03	6.672 9e+03	**3.558 4e+03**	1.086 9e+04
		Std	(2.651 1e+01)	(8.372 9e+02)	(8.370 8e+02)	(8.459 5e+02)	(4.407 6e+02)	(1.857 5e+03)
		Rank	6	4	2	3	1	5
PI+HS	T_1	Mean	3.700 5e+02	5.671 7e+02	4.321 5e+02	1.948 4e+02	5.017 0e+02	**1.706 4e+02**
		Std	(2.651 1e+01)	(3.472 6e+01)	(2.537 3e+01)	(6.076 3e+01)	(9.467 1e+01)	(1.331 3e+02)
		Rank	3	6	4	2	5	1
	T_2	Mean	3.556 8e+01	1.344 6e+02	3.016 9e-04	5.493 5e-02	6.577 9e+00	**7.866 7e-05**
		Std	(9.577 3e+00)	(4.336 4e+01)	(3.871 7e-04)	(1.530 5e-01)	(1.260 9e+00)	(3.311 4e-04)
		Rank	5	6	2	3	4	1
PI+MS	T_1	Mean	1.633 7e+03	6.048 2e+00	**2.161 0e-01**	2.839 0e+00	2.998 2e+00	4.787 6e-01
		Std	(1.800 7e+03)	(7.603 3e-01)	(4.229 6e-01)	(6.953 1e-01)	(3.581 9e-01)	(6.634 7e-01)
		Rank	6	5	1	3	4	2
	T_2	Mean	**9.965 5e-01**	2.948 6e+04	8.050 7e+01	1.704 8e+02	3.541 7e+02	1.003 0e+02
		Std	(8.085 5e-01)	(1.909 5e+04)	(2.871 0e+01)	(6.488 9e+01)	(7.644 1e+01)	(2.522 5e+01)
		Rank	1	6	2	4	5	3
PI+LS	T_1	Mean	9.501 5e+00	1.865 1e+01	**7.458 7e-02**	4.189 0e+00	4.644 0e+00	2.020 4e-01
		Std	(2.622 2e+00)	(4.978 6e+00)	(2.555 6e-01)	(7.157 7e-01)	(6.426 2e-01)	(4.147 3e-01)
		Rank	5	6	1	3	4	2
	T_2	Mean	6.597 6e+00	1.675 9e+01	4.819 8e-02	3.843 7e+00	4.888 7e+00	**1.333 3e-02**
		Std	(2.643 6e+00)	(4.610 4e+00)	(4.440 6e-02)	(1.013 6e+00)	(9.157 3e-01)	(2.091 8e-02)
		Rank	5	6	2	3	4	1
NI+HS	T_1	Mean	3.292 3e+03	3.278 3e+04	8.409 8e+01	2.994 2e+02	4.733 0e+02	**7.736 7e+01**
		Std	(1.595 3e+03)	(2.449 4e+04)	(3.678 8e+01)	(4.713 2e+02)	(1.131 2e+02)	(3.071 0e+01)
		Rank	5	6	2	3	4	1
	T_2	Mean	3.832 0e+02	3.718 2e+02	3.586 5e+01	1.025 0e+02	2.206 2e+02	**1.203 2e+01**
		Std	(1.839 1e+01)	(1.033 3e+02)	(6.879 9e+01)	(4.262 7e+01)	(5.451 2e+01)	(9.402 0e+00)
		Rank	6	5	2	3	4	1
NI+MS	T_1	Mean	6.510 4e-02	9.924 0e-01	2.799 1e-03	1.900 1e-02	3.417 4e-01	**2.245 3e-03**
		Std	(2.760 7e-02)	(7.211 2e-02)	(4.150 8e-03)	(1.926 9e-02)	(6.562 6e-02)	(4.714 4e-03)
		Rank	4	6	2	3	5	1
	T_2	Mean	8.076 0e+00	2.403 3e+01	5.186 1e+00	1.589 5e+01	2.364 5e+01	**3.378 4e+00**
		Std	(2.670 4e+00)	(3.100 7e+00)	(1.052 5e+00)	(2.927 2e+00)	(3.617 3e+00)	(8.214 9e-01)
		Rank	3	6	2	4	5	1
NI+LS	T_1	Mean	1.903 0e+03	6.453 2e+02	4.185 7e+02	**2.069 7e+02**	5.730 1e+02	3.972 9e+02
		Std	(1.091 2e+03)	(2.358 7e+01)	(2.641 3e+01)	(5.405 1e+01)	(1.064 3e+02)	(1.704 3e+01)
		Rank	6	5	3	1	4	2
	T_2	Mean	7.357 5e+03	1.033 3e+04	6.746 8e+03	7.136 3e+03	**3.667 3e+03**	8.337 0e+03
		Std	(1.321 2e+03)	(6.732 2e+02)	(1.079 4e+03)	(1.331 0e+03)	(5.333 9e+02)	(2.463 8e+03)
		Rank	4	6	2	3	1	5
Count			**1**	**0**	**2**	**1**	**3**	**11**
Ave Rank			**4.67**	**5.45**	**2.28**	**2.94**	**3.78**	**1.95**
Total Rank			**5**	**6**	**2**	**3**	**4**	**1**

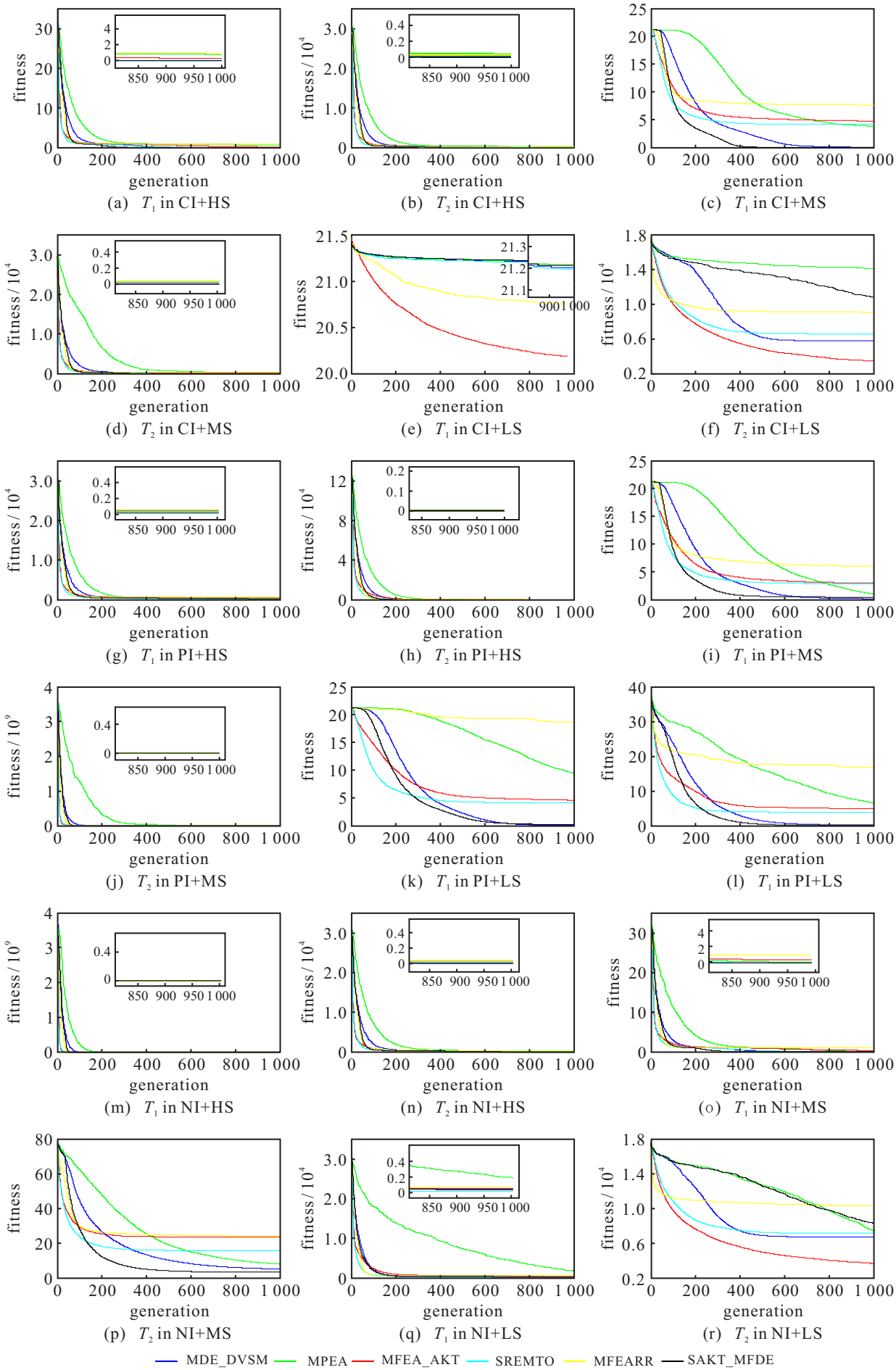


图 7 6种算法在多任务单目标测试集50维函数的收敛曲线

表 5　SAKT_MFDE与5个多任务算法的Wilcoxon符号秩检验结果

SAKT_MFDE vs	MPEA	MFARR	MDE_DVSM	SREMTO	MFEA_AKT
P_value	3.021 6e-02	9.049 3e-03	6.464 1e-01	1.412 4e-01	3.823 5e-02

为了进一步验证SAKT_MFDE算法在提升算法性能方面的优势,绘制6种算法在多任务单目标测试集50维函数的收敛曲线如图7所示. 在优化后期,由于函数优化的数值较小,放大显示了重叠部分的曲线. 由图7可以看出,在3种不同相似程度的多任务组中,SAKT_MFDE算法在相似度较高的任务组中性能最具优势,相比其他算法,SAKT_MFDE算法的收敛速度更快,这是由于SAKT_MFDE算法可以充分利用高相似度任务组上的有利信息.

同时,中度相似的任务组中SAKT_MFDE算法在前期的收敛速度优于MPEA、MDE_DVSM算法,但慢于MFEA_AKT、MFEARR和SREMTO,这是由于SAKT_MFDE算法在搜索初期存在学习期,不同任务之间不进行信息交流,导致一部分有效信息的流失,算法收敛速度较慢. 随着搜索进程的加快,SAKT_MFDE算法的收敛速度明显超过SREMTO算法.

在相似度较低的任务组中,SAKT_MFDE算法的收敛性相对其他5种算法不具明显的竞争力. 在相似度较低的情况下,自适应的迁移策略会减少任务之间的信息迁移,种群的多样性会减弱,更容易陷入局部极值点,导致SAKT_MFDE算法的收敛性减慢,算法的优化性能减弱.

5　结　论

本文针对MTEA算法存在的缺陷,提出一种基于超粒子引导的自适应知识迁移的多任务差分进化算法(SAKT_MFDE). 首先,提出自适应知识迁移算法改进传统MTEA的固定迁移概率,根据任务之间的相似程度动态调整不同任务之间的迁移概率,进而提高正向迁移效率;然后,改进MFDE算法的迁移算子,提出一种基于超粒子引导的差分迁移策略,在任务间进行知识迁移时能够充分利用迁移信息,使得目标任务可以较快地找到有利的搜索方向. 数值结果表明,所提出SAKT_MFDE能够更好地提高任务之间的正向迁移,提高多任务优化的性能.

本文算法仍存在以下不足:1)任务之间有用信息的利用仍不充分. 由于在AKT策略中设置学习期,忽略了前期演化过程中的有利信息,有利信息利用率较低. 2)在相似度较低的任务组中,SAKT_MFDE的收敛性能不够理想,虽然SAKT_MFDE算法相比MTEA在相似度较低的任务组中收敛性能有所提升,但由于SAKT_MFDE算法在任务间相似度较低的情况下会减少甚至不进行知识迁移,种群多样性减弱,导致SAKT_MFDE算法收敛性能相比其他同类型的改进算法不具优势. MTEA算法是一种比较新的算法,尚有许多地方需要探讨和完善,未来将进一步改进 MTEA,对其进行深入的理论研究,并拓展其应用领域.

参考文献(References)

[1] 付华, 刘昊. 多策略融合的改进麻雀搜索算法及其应用[J]. 控制与决策, 2022, 37(1): 87-96.
(Fu H, Liu H. Improved sparrow search algorithm with multi-strategy integration and its application[J]. Control and Decision, 2022, 37(1): 87-96.)

[2] 刘振, 鲁华杰, 刘文彪. 自适应协同进化蝙蝠算法[J]. 控制与决策, 2019, 34(8): 1626-1634.
(Liu Z, Lu H J, Liu W B. Adaptive cooperation evolutionary bat algorithm[J]. Control and Decision, 2019, 34(8): 1626-1634.)

[3] Gupta A, Ong Y S, Feng L. Multifactorial evolution: Toward evolutionary multitasking[J]. IEEE Transactions on Evolutionary Computation, 2016, 20(3): 343-357.

[4] Zhang F F, Mei Y, Nguyen S, et al. Surrogate-assisted evolutionary multitask genetic programming for dynamic flexible job shop scheduling[J]. IEEE Transactions on Evolutionary Computation, 2021, 25(4): 651-665.

[5] Yang C E, Ding J L, Jin Y C, et al. Multitasking multiobjective evolutionary operational indices optimization of beneficiation processes[J]. IEEE Transactions on Automation Science and Engineering, 2019, 16(3): 1046-1057.

[6] Liang J, Qiao K J, Yuan M H, et al. Evolutionary multi-task optimization for parameters extraction of photovoltaic models[J]. Energy Conversion and Management, 2020, 207: 112509.

[7] Feng L, Zhou L, Gupta A, et al. Solving generalized vehicle routing problem with occasional drivers via evolutionary multitasking[J]. IEEE Transactions on Cybernetics, 2021, 51(6): 3171-3184.

[8] Liang Z P, Zhang J, Feng L, et al. Multi-factorial optimization for large-scale virtual machine placement in

cloud computing[J/OL]. 2020, arXiv: 2001.06585.

[9] Wu K, Wang C, Liu J. Evolutionary multitasking multilayer network reconstruction[J]. IEEE Transactions on Cybernetics, 2022, 52(12): 12854-12868.

[10] Feng L, Zhou W, Zhou L, et al. An empirical study of multifactorial PSO and multifactorial DE[C]. IEEE Congress on Evolutionary Computation. Donostia, 2017: 921-928.

[11] Tang Z D, Gong M G. Adaptive multifactorial particle swarm optimisation[J]. CAAI Transactions on Intelligence Technology, 2019, 4(1): 37-46.

[12] Cai Y Q, Peng D N, Fu S K, et al. Multitasking differential evolution with difference vector sharing mechanism[C]. IEEE Symposium Series on Computational Intelligence. Xiamen, 2020: 3039-3046.

[13] Zheng X L, Qin A K, Gong M G, et al. Self-regulated evolutionary multitask optimization[J]. IEEE Transactions on Evolutionary Computation, 2020, 24(1): 16-28.

[14] Li G H, Lin Q Z, Gao W F. Multifactorial optimization via explicit multipopulation evolutionary framework[J]. Information Sciences, 2020, 512: 1555-1570.

[15] Wang N, Xu Q Z, Fei R, et al. Rigorous analysis of multi-factorial evolutionary algorithm as multi-population evolution model[J]. International Journal of Computational Intelligence Systems, 2019, 12(2): 1121-1133.

[16] Wang L, Sun Q, Xu Q, et al. Analysis of multitasking evolutionary algorithms under the order of solution variables[J]. Complexity, 2020, 2020: 1-18.

[17] Zhang D Q, Jiang M Y. Hetero-dimensional multitask neuroevolution for chaotic time series prediction[J]. IEEE Access, 2020, 8: 123135-123150.

[18] Xu Z W, Liu X M, Zhang K, et al. Cultural transmission based multi-objective evolution strategy for evolutionary multitasking[J]. Information Sciences, 2022, 582: 215-242.

[19] Zhou L, Feng L, Zhong J, et al. Astudy of similarity measure between tasks for multi factorial evolutionary algorithm[C]. Genetic and Evolutionary Computation Conference. NewYork, 2018: 229-230.

[20] Pitzer E, Affenzeller M. A comprehensive survey on fitness landscape analysis[M]. Berlin: Springer-Verlag, 2012: 161-191.

[21] Yuan Y, Ong Y S, Feng L, et al. Evolutionary multitasking for multiobjective continuous optimization: Benchmark problems, performance metrics and baseline results[J/OL]. 2017, arXiv: 1706.02766.

[22] Wen Y W, Ting C K. Parting ways and reallocating resources in evolutionary multitasking[C]. IEEE Congress on Evolutionary Computation. Donostia, 2017: 2404-2411.

[23] Zhou L, Feng L, Tan K C, et al. Toward adaptive knowledge transfer in multifactorial evolutionary computation[J]. IEEE Transactions on Cybernetics, 2021, 51(5): 2563-2576.

作者简介

孙倩(1997−), 女, 硕士生, 从事智能计算的研究, E-mail: qiansun1395@163.com;

王磊(1972−), 男, 教授, 博士生导师, 从事多任务、多目标优化计算、知识图谱应用等研究, E-mail: wanglei_sut@163.com;

徐庆征(1980−), 男, 副研究员, 博士, 从事进化计算、群体智能等研究, E-mail: xuqingzheng@hotmail.com;

夏坤(1998−), 男, 硕士生, 从事智能计算的研究, E-mail: xiakun129@163.com;

李薇(1973−), 女, 副教授, 博士, 从事进化计算、神经网络和数据挖掘等研究, E-mail: liwei@xaut.edu.cn.