

文章编号:1671-6833(2023)05-0024-08

基于联合分布适配的单向迁移差分进化算法

李 晰^{1,2}, 李 帅^{1,2}, 冯艳红¹, 李明亮¹

(1. 河北地质大学 信息工程学院, 河北 石家庄 050031; 2. 河北地质大学 人工智能与机器学习研究室, 河北 石家庄 050031)

摘 要:传统的差分进化算法求解优化问题一般从零知识开始,独立搜索,没有利用已求解过的相似问题信息,针对这一问题,在传统的差分进化算法中引入迁移学习技术。首先,利用存在相关性的源问题的优化种群和目标问题的当前种群抽取关键信息,通过联合分布适配的方法映射到高维希尔伯特空间。其次,用映射后得到的矩阵构建新种群,代替目标问题的种群,完成后续进化任务。实现了2种迁移模式:在目标问题求解初始化时,将源问题的有效信息进行迁移,引导算法搜索方向;目标问题求解迭代一定的次数后,再利用迁移的有效信息,加快种群收敛速度。最后,采用9组多任务测试函数对算法进行了测试,与无迁移的差分进化算法以及直接迁移种群的无适配技术的差分进化算法进行对比。结果表明:在求解质量方面,所提算法有7组优于传统的无迁移差分进化算法;在求解速度方面,所提算法有7组比传统差分进化算法收敛速度更快;基于迁移学习的差分进化算法对提高目标优化问题的求解精度和收敛速度是有效的。

关键词:优化算法; 迁移学习; 联合分布适配; 单向迁移; 差分进化算法

中图分类号: O234; TP301

文献标志码: A

doi: 10.13705/j.issn.1671-6833.2023.05.005

研究者模仿自然界中生物的群体行为,或者受社会实践中某些活动的启发,提出了演化算法(evolutionary algorithm, EA)^[1-3],它在求解优化问题时表现出了优越的性能。其中,差分进化算法(differential evolutionary, DE)^[4-5]因其参数设置简单、搜索能力强,成为最有效的演化算法之一。DE算法同其他演化算法类似,生成随机的初始种群,通过交叉、变异、选择算子进行种群的迭代更新,在决策空间中进行搜索,直到找到最优解或者满足结束条件为止。传统的差分进化算法是从问题零知识的情况下开始搜索,在搜索过程中始终保持种群的独立性,各种交叉变异操作在种群个体之间进行,信息的交互在种群内部完成。不管是在求解的初始阶段还是在求解过程中,传统的差分进化算法没有利用已求解过的相似问题信息获得有效知识。

迁移学习^[6]是机器学习中最热门的研究领域之一。迁移学习的基本思想是将已经解决的问题的结果或者方法运用到未解决的问题中,以加快新问

题的求解速度或提高问题的求解精度。一般地,将已经学习过的知识领域称作源域,将待应用或待求解的新领域称作目标域。迁移学习已经在图像分类^[7]、自动驾驶^[8]、推荐系统^[9]等领域均展现出良好的有效性。迁移学习与演化计算相结合的研究工作近期受到诸多学者的关注^[10-11]。Jiang等^[12]应用迁移学习机制求解动态优化问题,设计了一个集成迁移成分分析和多目标优化的算法框架,并将该算法框架与3个典型的多目标优化算法NSGA-II、MOPSO和RM-MEDA相结合,求解动态多目标优化问题。Jiang等^[13]又提出一种基于领域自适应和非参数估计的分布估计算法DANE-EDA,该算法将蒙特卡洛方法与迁移学习相结合,用以维持在动态优化中勘探与开发的平衡。张勇等^[14]提出一种基于相似历史信息迁移学习的进化优化框架,该算法从历史模型库中寻找与新问题相匹配的历史问题,将历史问题对应的知识迁移到新问题中,以提高种群的搜索效率。刘璐等^[15]提出Tr-NSGA II算法,该算

收稿日期:2023-01-16;修订日期:2023-03-20

基金项目:国家自然科学基金资助项目(61806069);河北省高校科学技术研究资助项目(ZD2022083);河北省重点研发计划项目(22375415D)

作者简介:李晰(1977—),女,河北深州人,河北地质大学副教授,博士,主要从事人工智能、演化计算的研究,E-mail: lixi_sjz@foxmail.com。

引用本文:李晰,李帅,冯艳红,等. 基于联合分布适配的单向迁移差分进化算法[J]. 郑州大学学报(工学版), 2023, 44(5): 24-31. (LI X, LI S, FENG Y H, et al. Unidirectional transfer differential evolution algorithm based on joint distribution adaptation[J]. Journal of Zhengzhou University (Engineering Science), 2023, 44(5): 24-31.)

法将 NSGA-II 算法与迁移成分分析方法相结合,对含有历史信息的种群进行搜索,以提高种群的搜索效率与算法的收敛性能。尚青霞等^[16]提出一种基于降噪自动编码器的进化多任务算法,通过显式的遗传迁移方法,实现跨任务的迁移学习。Feng 等^[17]将基于自动编码器的显式迁移技术扩展到了多目标多任务优化,针对多目标问题的特征,提出了多目标种群排序和映射矩阵的方法,用于优选解的迁移。

现有算法中使用的迁移学习技术有迁移成分分析(transfer component analysis, TCA)^[18]、测地线流式核法(geodesic flow kernel, GFK)^[19]等。本文将另一种迁移学习技术——联合分布适配(joint distribution adaptation, JDA)^[20]方法引入到演化计算中。JDA 是一种更巧妙、更精准的求解方法,对不同问题的适配程度更好,因此本文提出一种基于联合分布适配的单向迁移差分进化算法。该算法利用已求解过的简单单目标优化问题(源问题)的种群信息,通过联合分布适配的迁移学习技术,将有效知识迁移到具有相似性的复杂单目标优化问题(目标问题)中,以提高求解目标问题的收敛速度与求解精度。

1 迁移学习基础

1.1 最大均值差异

最大均值差异是迁移学习中应用较多的一种损失函数,用来衡量 2 个不同但存在一定相似性的随机变量的分布距离。这里假设存在一个源域 $\mathbf{X}_s = [x_{s_1}, x_{s_2}, \dots, x_{s_n}]$ 满足 P 分布,存在一个目标域 $\mathbf{X}_t = [x_{t_1}, x_{t_2}, \dots, x_{t_m}]$, 假设 H 为再生核希尔伯特空间(reproducing kernel Hilbert space, RKHS), $\Phi(\cdot): \mathbf{X} \rightarrow H$ 表示原始特征空间映射到再生核希尔伯特空间的映射函数,当 $n, m \rightarrow \infty$ 时, \mathbf{X}_s 和 \mathbf{X}_t 在再生核希尔伯特空间中的最大均值差异(maximum mean discrepancy, MMD)^[15]表示如下:

$$\text{dist}(\mathbf{X}_s, \mathbf{X}_t) = \left\| \frac{1}{n} \sum_{i=1}^n f(\mathbf{x}_{s_i}) - \frac{1}{m} \sum_{j=1}^m f(\mathbf{x}_{t_j}) \right\|_{\mathcal{H}}. \quad (1)$$

实际上, MMD 所求即源域与目标域映射到高维向量空间后的均值差异。目前 MMD 应用的实例不在少数,并且证明其具有良好的效果。但将其运用在迁移学习中并与演化计算相结合的实例仍然很少出现。

1.2 联合分布适配

联合分布适配是迁移学习中一种常用的方法,其目的是通过学习源域数据中有用的相关知识,进而对目标域的求解过程提供可靠有力的帮助。

1.2.1 边缘分布适配

为了使源域数据与目标域数据的边缘概率即 $P(\mathbf{A}^T \mathbf{X}_s)$ 和 $P(\mathbf{A}^T \mathbf{X}_t)$ 之间的距离尽可能小,采用最大均值差异 MMD 距离来度量源域与目标域的边缘概率分布差异,具体的表达式为

$$\left\| \frac{1}{n} \sum_{i=1}^n \mathbf{A}^T \mathbf{x}_{s_i} - \frac{1}{m} \sum_{j=1}^m \mathbf{A}^T \mathbf{x}_{t_j} \right\|_{\mathcal{H}}^2 = \text{tr}(\mathbf{A}^T \mathbf{X} \mathbf{M}_0 \mathbf{X}^T \mathbf{A}). \quad (2)$$

式中: \mathbf{X} 为源域与目标域整合在一起的数据; \mathbf{M}_0 为一个 MMD 矩阵,表达式为

$$(\mathbf{M}_0)_{ij} = \begin{cases} \frac{1}{n^2}, & \mathbf{x}_i, \mathbf{x}_j \in \mathcal{D}_s; \\ \frac{1}{m^2}, & \mathbf{x}_i, \mathbf{x}_j \in \mathcal{D}_t; \\ -\frac{1}{mn}, & \text{其他}. \end{cases} \quad (3)$$

式中: \mathcal{D}_s 为有标注的源域; \mathcal{D}_t 为无标注的目标域; n 和 m 分别为源域和目标域的样本个数。通过式(3),使得边缘分布 $P(\mathbf{A}^T \mathbf{X})$ 在某种形式下距离尽可能地接近。

1.2.2 条件分布适配

同样地,需要对源域和目标域的条件概率分布进行适配。根据贝叶斯公式 $P(y_i | \mathbf{X}_t) = P(y_i)P(\mathbf{X}_t | y_i)$ 和充分统计量的定义,在样本中存在较多的未知量并且样本足够好的情况下,便可以从样本中选择一些统计量用来近似替代实际需要估计的分布,这里可用 $P(\mathbf{X}_t | y_i)$ 来近似 $P(y_i | \mathbf{X}_t)$ 。使用 (\mathbf{X}_s, y_s) 训练一个 K 近邻分类器,利用这个分类器,对已知的 \mathbf{X}_t 进行预测,得到预测标签 \hat{y}_t , 作为目标域标注进行计算。则此时类间的 MMD 距离可以表示为

$$\sum_{c=1}^C \left\| \frac{1}{n_c} \sum_{\mathbf{x}_{s_i} \in \mathcal{D}_s^{(c)}} \mathbf{A}^T \mathbf{x}_{s_i} - \frac{1}{m_c} \sum_{\mathbf{x}_{t_i} \in \mathcal{D}_t^{(c)}} \mathbf{A}^T \mathbf{x}_{t_i} \right\|_{\mathcal{H}}^2 = \text{tr}(\mathbf{A}^T \mathbf{X} \mathbf{M}_c \mathbf{X}^T \mathbf{A}). \quad (4)$$

式中: n_c, m_c 分别为源域和目标域中来自第 c 类的样本个数。 \mathbf{M}_c 为

$$(\mathbf{M}_c)_{ij} = \begin{cases} \frac{1}{n_c^2}, & \mathbf{x}_i, \mathbf{x}_j \in \mathcal{D}_s^{(c)}; \\ \frac{1}{m_c^2}, & \mathbf{x}_i, \mathbf{x}_j \in \mathcal{D}_t^{(c)}; \\ -\frac{1}{m_c n_c}, & \begin{cases} \mathbf{x}_i \in \mathcal{D}_s^{(c)}, \mathbf{x}_j \in \mathcal{D}_t^{(c)}; \\ \mathbf{x}_i \in \mathcal{D}_t^{(c)}, \mathbf{x}_j \in \mathcal{D}_s^{(c)}; \end{cases} \\ 0, & \text{其他}. \end{cases} \quad (5)$$

通过条件分布适配,可以使得源域与目标域之间的条件分布 $P(y | \mathbf{A}^T \mathbf{X})$ 尽可能地减小差异。

联合分布适配要求源域和目标域之间的边缘分

布适配与条件分布适配均尽可能地小,故将上述 2 种适配方法相结合,得到了联合分布适配的目标函数:

$$\min \sum_{c=1}^C \text{tr}(\mathbf{A}^T \mathbf{X} \mathbf{M}_c \mathbf{X}^T \mathbf{A}) + \lambda \|\mathbf{A}\|_F^2. \quad (6)$$

式中: $\lambda \|\mathbf{A}\|_F^2$ 为正则项。同时,需要一个限定条件为一个变换前后数据的方差维持不变,增加另外的一个优化目标 $\max \mathbf{A}^T \mathbf{X} \mathbf{H} \mathbf{X}^T \mathbf{A}$, 其中 \mathbf{H} 为中心矩阵。将其与原来的优化目标合并,并使用拉格朗日法可得

$$\left(\mathbf{X} \sum_{c=0}^C \mathbf{M}_c \mathbf{X}^T + \lambda \mathbf{I} \right) \mathbf{A} = \mathbf{X} \mathbf{H} \mathbf{X}^T \mathbf{A} \Phi. \quad (7)$$

式中: Φ 为拉格朗日乘子; $\mathbf{I} = \mathbf{A}^T \mathbf{X} \mathbf{H} \mathbf{X}^T \mathbf{A}$ 。经过多次迭代过程,使得预测出的标签 \hat{y}_i 与真实标签之间的误差逐渐减小,且可以求出变换矩阵 \mathbf{A} , 最终实现源域和目标域适配的目的。

2 基于联合分布适配的单向迁移差分进化算法

当目标域数据完全无标注时,一般可以通过迁移成分分析的方法进行适配。该方法实现简单,但计算消耗时间多,且因为只对边缘概率分布进行适配,可能会存在领域间欠适配的问题,进而影响到对目标域问题的求解准确度。Long 等^[20]于 2014 年提出了联合分布适配,较好地解决了上述问题。联合分布适配方法要解决的是找到一种变换矩阵 \mathbf{A} , 使得源域数据与目标域数据的边缘概率(分别为 $P(\mathbf{A}^T \mathbf{X}_s)$ 和 $P(\mathbf{A}^T \mathbf{X}_t)$)和条件概率(分别为 $P(y_s | \mathbf{A}^T \mathbf{X}_s)$ 和 $P(y_t | \mathbf{A}^T \mathbf{X}_t)$)的距离均尽可能地减小。

因此,本文提出基于联合分布适配的单向迁移差分进化算法(JDA based transfer differential evolution algorithm, JDA_TrDE),实现了两种迁移模式:第一种是对初始种群的迁移;第二种是对迭代过程中生成种群的迁移。

2.1 初始迁移的 DE 算法 JDA_TrDE₁

JDA_TrDE₁ 算法对初始种群采用迁移技术。考虑到种群初始化的随机性,该算法在搜索之初就利用源问题优化后的种群,在初始时刻利用已学知识给目标问题一个指向性的引导,尽可能地避免种群的无效搜索。在算法中需要对源数据设置标签,本文采用的方法是将种群个体的适应值进行归一化处理后,按照大小划分成 3 类,分别赋予个体 0、1、2 共 3 类标签。

JDA_TrDE₁ 算法流程如下。

输入:种群规模 NP 、比例因子 F 和交叉概率 P_{CR} 、迭代次数 $generation$ 、源问题优化种群 P_s ;

输出:目标优化问题的最优解 y 和最优个体 x 。

Step 1 从 P_s 中抽取个体的基因形成源域数据 \mathbf{X}_s , 个体的适应值转化为源域数据的标签 y_s , 形成有标注的源域 \mathcal{D}_s ;

Step 2 在目标问题下生成随机的初始化种群 P_t , 从 P_t 中抽取个体的基因形成目标域数据 \mathbf{X}_t , 转化为无标注的目标域 \mathcal{D}_t ;

Step 3 将 \mathcal{D}_s 和 \mathcal{D}_t 作为 JDA 算法的输入,利用式(6)得到的变换矩阵 \mathbf{A} 将 \mathbf{X}_t 进行变换,得到新的矩阵 \mathbf{X}_{t_new} ;

Step 4 令 \mathbf{X}_{t_new} 作为个体的基因构建新种群,替换目标问题的初始种群;

Step 5 使用 DE 算法的变异、交叉、选择操作进行迭代;

Step 6 迭代次数达到设定的代数 $generation$, 算法终止;

Step 7 得到最优解 y 和最优个体 x , 结束迭代过程。

2.2 过程迁移的 DE 算法 JDA_TrDE₂

JDA_TrDE₂ 算法在迭代过程中进行种群迁移操作。目标问题求解之初,采用 DE 算法进行随机搜索一定的次数,使其充分搜索可能的区域。而后加入从源问题中所学到的知识,引导种群在剩余迭代次数中尽可能快地收敛。

JDA_TrDE₂ 算法流程如下。

输入:种群规模 NP 、比例因子 F 和交叉概率 P_{CR} 、迭代次数 $generation_1$ 、迭代次数 $generation_2$ 、源问题优化种群 P_s ;

输出:单目标优化问题的最优解 y 和最优个体 x 。

Step 1 初始化种群 P_t ;

Step 2 使用 DE 算法求解最优解 $generation_1$ 代;

Step 3 得到运行 $generation_1$ 代的种群 P_{t_1} ;

Step 4 从 P_s 中抽取个体的基因形成源域数据 \mathbf{X}_s , 个体的适应值转化为源域数据的标签 y_s , 形成有标注的源域 \mathcal{D}_s ;

Step 5 从 P_{t_1} 中抽取个体的基因形成目标域数据 \mathbf{X}_t , 转化为无标注的目标域 \mathcal{D}_t ;

Step 6 将 \mathcal{D}_s 和 \mathcal{D}_t 作为 JDA 算法的输入,利用式(6)得到的变换矩阵 \mathbf{A} 将 \mathbf{X}_t 进行变换,得到新的矩阵 \mathbf{X}_{t_new} ;

Step 7 令 \mathbf{X}_{t_new} 作为个体的基因构建新种群,替换目标问题的 P_{t_1} , 应用 DE 算法进行求解

generation_2 代;

Step 8 得到最优解 y 和最优个体 x ,结束迭代过程。

3 实验及结果分析

3.1 实验设置

本文采用文献[16]中的测试问题对所提算法的性能进行评估,每组测试问题由 2 个常见的单目标优化函数组成。一般地,每组测试问题中,任务 1 (T_1) 为简单函数,任务 2 (T_2) 为复杂函数。对于每组函数给出函数类型和函数间的相交程度与相似度,相似度值越高表明函数间相似程度越高(基于 Pearson 相关性^[21]对任务间相似度进行定义),于表 1 中总结了实验中涉及的 9 组测试问题, d 为维度。为了验证本文所提出的改进差分进化算法的有效性,在对比实验中增加无联合分布适配直接迁移算法,包含 2 种模式:一是在初始阶段直接将 T_1 问题的最优解完全迁移到 T_2 问题中的初始迁移 TrDE₁;二是 generation_1 代迭代过后直接将 T_1 问题的解完全迁移到 T_2 问题中的过程迁移 TrDE₂。实验具体参数设置如表 2 所示。

3.2 实验结果与分析

3.2.1 解的质量

本文所提算法与对比算法在测试问题实验中得到的平均目标值和标准偏差如表 3 所示,与文献[16]中进化多任务算法中对 T_2 问题的求解结果进

行对比,采用置信水平为 95%的 Wilcoxon 秩和检验法对实验结果进行检测。

表 3 中记录每个问题求解出的平均目标值和标准偏差。在 9 组测试函数中,有 7 组测试函数是利用迁移学习的方法可以求出更优解的。在这 7 组求出更优解的实验中,通过本文提出的两种迁移学习方法求出最优解的测试函数一共有 4 组, JDA_TrDE₁ 和 JDA_TrDE₂ 方法求出的解为这 5 种算法中最优的 2 种解;测试问题 1 和 8,虽然使用 JDA 方法求出来的解不如使用直接迁移方法求出来的解更优,但都比不使用迁移学习方法直接求解的效果更优。测试问题 6 为文献[16]中所提迁移学习算法求解结果更优,本文提出的 JDA_TrDE₁ 和 JDA_TrDE₂ 算法为第 2、第 3 优解,且均比不使用迁移学习方法直接求解的效果更优。通过较好的实验结果证明,使用单向迁移学习优化过后的差分进化算法是可以提高算法的优化性能的,并且实验结果显示,初始迁移的 JDA_TrDE₁ 方法比过程迁移的 JDA_TrDE₂ 方法求得的目标值更优。

采用联合分布适配方法将源问题的先验知识迁移到目标问题上,对帮助求解目标问题是有效的。将源域数据和目标域数据映射到同一高维空间中,分别进行数据间边缘分布适配与条件分布适配,减小随机变量的分布距离,使源问题数据中的有效信息更完整、更精确地帮助求解目标问题。但联合分布适配也存在一些问题,例如,在求解复杂多峰函数

表 1 测试问题属性
Table 1 Test problem attributes

测试问题	优化函数	任务性质	搜索空间	相交程度	相似度
1	Griewank (T_1)	多峰,不可分	$x \in [-100, 100]^d$	完全相交	1.000 0
	Rastrigin (T_2)	多峰,不可分	$x \in [-50, 50]^d$		
2	Ackley (T_1)	多峰,不可分	$x \in [-50, 50]^d$	完全相交	0.226 1
	Rastrigin (T_2)	多峰,不可分	$x \in [-50, 50]^d$		
3	Ackley (T_1)	多峰,不可分	$x \in [-50, 50]^d$	完全相交	0.000 2
	Schwefel (T_2)	多峰,可分	$x \in [-500, 500]^d$		
4	Rastrigin (T_1)	多峰,不可分	$x \in [-500, 500]^d$	部分相交	0.867 0
	Sphere (T_2)	单峰,可分	$x \in [-50, 50]^d$		
5	Ackley (T_1)	多峰,不可分	$x \in [-50, 50]^d$	部分相交	0.215 4
	Rosenbrock (T_2)	多峰,不可分	$x \in [-50, 50]^d$		
6	Ackley (T_1)	多峰,不可分	$x \in [-50, 50]^d$	部分相交	0.072 5
	Weierstrass (T_2)	多峰,不可分	$x \in [-0.5, 0.5]^d$		
7	Rosenbrock (T_1)	多峰,不可分	$x \in [-50, 50]^d$	不相交	0.943 4
	Rastrigin (T_2)	多峰,不可分	$x \in [-50, 50]^d$		
8	Griewank (T_1)	多峰,不可分	$x \in [-100, 100]^d$	不相交	0.366 9
	Weierstrass (T_2)	多峰,不可分	$x \in [-0.5, 0.5]^d$		
9	Rastrigin (T_1)	多峰,不可分	$x \in [-50, 50]^d$	不相交	0.001 6
	Schwefel (T_2)	多峰,可分	$x \in [-500, 500]^d$		

问题时,当源域数据与目标域数据之间的距离没有缩减到合适的范围,源域数据没有提供很好的知识帮助,就可能使目标问题陷入局部最优解,例如测试问题 3 和 9;当源问题与目标问题之间完全相交且相似程度极高时,例如测试问题 1,源问题与目标问题的最优解可能极为相似,因此进行联合分布适配以后的迁移效果略差于直接迁移。

3.2.2 收敛速度

为了进一步说明使用单向迁移技术可以加快求解单目标优化问题的收敛速度,本节对原始差分进化算法(DE)、无联合分布适配技术的直接迁移差分进化算法的 2 种模式即初始迁移 TrDE₁ 算法和过程迁移 TrDE₂ 算法以及基于联合分布适配的迁移差分进化算法 2 种模式即初始迁移 JDA_TrDE₁ 算法和过程迁移 JDA_TrDE₂ 算法每次迭代结束后的最小值变化过程进行轨迹追踪分析。

图 1 描绘了采用 5 种算法求解测试问题中任务

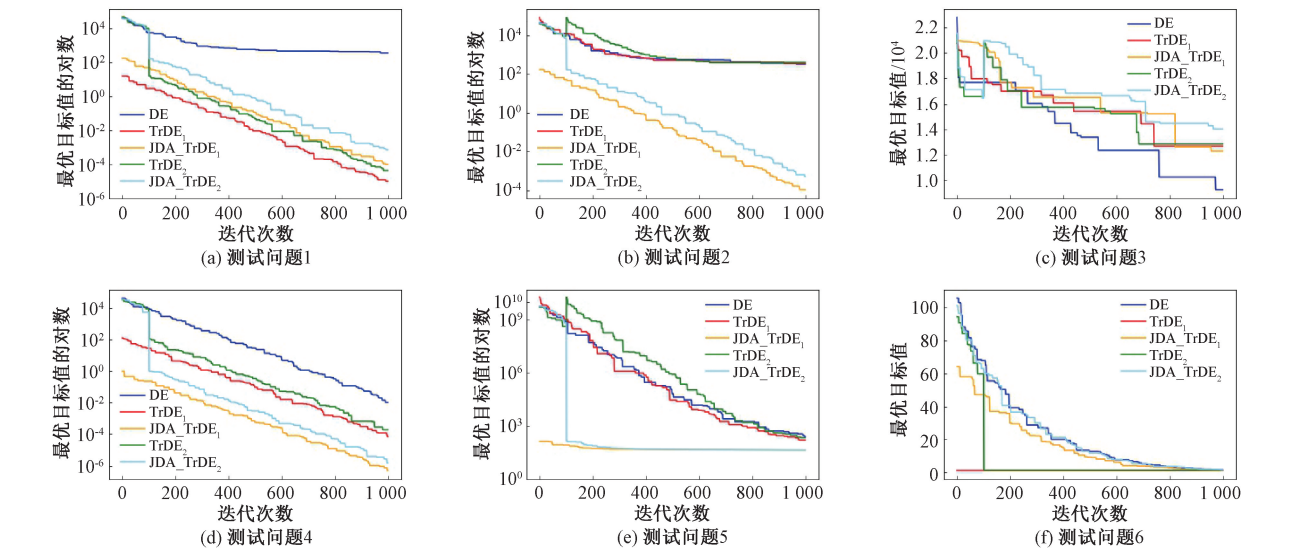
T₂ 的每次迭代结束后的最小值变化过程。其中,横坐标为迭代次数,图 1(a)、1(b)、1(d)、1(e)、1(g)的纵坐标为目标值的对数,图 1(c)、(f)、(h)、(i)的纵坐标为目标值。

表 2 实验具体参数设置

Table 2 Experiment-specific parameter settings			
项目	参数名称	参数含义	取值
DE 算法	<i>NP</i>	种群大小	100
	—	算子类型	DE/rand/1
	<i>generation</i>	迭代次数	1 000
	<i>F</i>	缩放因子	0.6
JDA 算法	<i>P_{CR}</i>	交叉概率	0.7
	<i>kernel_type</i>	内核类型	primal
	<i>K</i>	迭代次数	1
JDA_TrDE ₂	<i>generation₁</i>	迁移前迭代次数	100
	<i>generation₂</i>	迁移后迭代次数	900
测试问题	<i>n</i>	独立运行次数	20
	<i>d</i>	问题维度	60

表 3 不同算法在单目标问题上的平均目标值和标准偏差

Table 3 Average objective values and standard deviations of Different algorithms on the single objective problem													
测试 问题	测试函数结果												
	DE		文献[16]		TrDE ₁		TrDE ₂		JDA_TrDE ₁		JDA_TrDE ₂		
	均值	标准差	均值	标准差	均值	标准差	均值	标准差	均值	标准差	均值	标准差	
1	3.71E+2	3.38E+1	2.16E-2	8.54E-2	8.21E-6	4.73E-6	3.39E-5	1.99E-5	1.33E-4	8.28E-5	5.09E-4	1.32E-4	
2	3.59E+2	3.88E+1	1.37E+1	2.46E+1	3.59E+2	3.29E+1	3.95E+2	4.12E+1	1.28E-4	3.98E-5	4.37E-4	1.62E-4	
3	1.12E+4	7.96E+2	6.73E+3	9.17E+2	1.19E+4	6.87E+2	1.23E+4	8.93E+2	1.31E+4	8.18E+2	1.34E+4	1.06E+3	
4	2.01E-2	1.13E-2	1.74E+2	1.99E+2	5.65E-5	1.95E-5	2.23E-4	7.06E-5	5.57E-7	2.91E-7	2.50E-6	7.76E-7	
5	1.46E+2	1.25E+2	1.57E+2	5.40E+1	1.62E+2	1.52E+2	3.79E+2	2.25E+2	4.38E+1	2.59E-1	4.43E+1	2.37E-1	
6	1.83E+0	2.19E-1	2.35E-3	3.93E-3	3.34E+0	1.75E+0	3.34E+0	1.75E+0	1.17E+0	1.10E-1	1.80E+0	2.35E-1	
7	3.45E+2	2.88E+1	2.84E+1	2.25E+1	4.55E+1	8.19E+1	5.19E+1	8.33E+1	1.14E-4	4.50E-5	4.70E-4	1.47E-4	
8	1.96E+0	2.46E-1	2.63E+0	5.15E-1	5.41E-1	8.98E-2	8.03E-1	1.42E-1	1.31E+0	1.68E-1	1.83E+0	1.80E-1	
9	1.11E+4	1.04E+3	7.08E+3	9.85E+2	1.29E+4	9.64E+2	1.34E+4	1.06E+3	1.31E+4	9.05E+2	1.34E+4	9.12E+2	



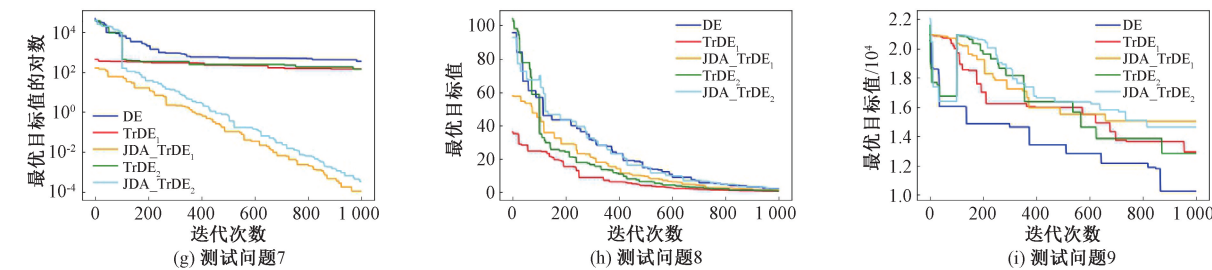


图1 优化问题迭代求解过程中的最小值变化轨迹

Figure 1 Minimax change trajectory during iterative solution of optimization problems

从图1可以看出,使用单向迁移技术后的差分进化算法在搜索过程中明显快于原始的差分进化算法。其中图1(b)、(d)、(e)和(g)中,JDA_TrDE₁算法和JDA_TrDE₂算法的2条曲线不仅在收敛精度上优于其他3种算法,且收敛速度也远快于其他3种算法,效果最佳。JDA_TrDE₂算法在100代使用联合分布适配技术后,下一代的最小适应度值均有大幅减小,并且以后的收敛速度有明显的加快。在图1(e)中,使用联合分布适配的两种算法在400代之前就已经收敛到最优值,且最优值比其他算法迭代1000代的收敛精度更好。在图1(a)、(f)和(h)中,收敛曲线证明单向迁移技术对求解问题最优值的有效性是存在的,同样具有较高的收敛速度和收敛精度。因此,JDA_TrDE₁算法和JDA_TrDE₂算法同样有较好的收敛速度和收敛精度。

3.3 对联合适配分布不同参数的对比实验

在联合分布适配算法中,有5个重要参数,即内核类型、迁移后的维度、lambda值、rbf内核带宽gamma以及迭代次数。本文对其中的内核类型 *kernel_type* 及迭代次数 *K*,选取不同的值进行对比实验。对每种不同参数的算法独立运行10次。

表4为对联合分布适配选取不同内核的实验结

果,记录每个问题求解出的平均目标值和标准偏差。对于验证的9个测试问题,其中内核为 *primal* 时有1个问题求得最优解;内核为 *linear* 时有6个问题求得最优解;内核为 *rbf* 时有2个问题求得最优解。对于内核为 *rbf*,出现在测试问题1、2和7中求出来的解的收敛精度远小于内核为 *primal* 和内核为 *linear* 的情况,所以在作为迁移技术求解单目标问题时,不适合作为JDA算法的内核。内核为 *primal* 和内核为 *linear* 时分别求出的最优解在比较后认为精度相差不大,均可作为JDA算法在求解单目标问题时的内核。

表5为对联合分布适配选取不同迭代次数的实验结果,记录每个问题求解出的平均目标值和标准偏差。对于验证的9个测试问题,*K*=2时有2个问题求得最优解,*K*=3时有4个问题求得最优解,*K*=5时有3个问题求得最优解。理论上,使用实验中预测的标签作为伪目标标签并迭代运行JDA,就可以提高标签质量,直到收敛。实验结果在一定程度上也证明了这一点。不过分析9个实验的收敛精度发现,大多数问题对于迭代次数这个参数的敏感程度并不高,最优解与次优解,甚至第三优解之间的差距很小,因此在实验过程中可酌情选择JDA算法的迭代次数。

表4 联合分布适配选取不同内核的对比实验

Table 4 Comparison experiments of different kernels selected for joint distribution adaptation

测试 问题	内核为 <i>primal</i>				内核为 <i>linear</i>				内核为 <i>rbf</i>			
	JDA_TrDE ₁		JDA_TrDE ₂		JDA_TrDE ₁		JDA_TrDE ₂		JDA_TrDE ₁		JDA_TrDE ₂	
	均值	标准差	均值	标准差	均值	标准差	均值	标准差	均值	标准差	均值	标准差
1	1.33E-4	8.28E-5	5.09E-4	1.32E-4	1.01E-4	3.05E-5	6.19E-4	4.05E-4	9.90E-1	1.46E-5	9.90E-1	1.26E-4
2	1.28E-4	3.98E-5	4.37E-4	1.62E-4	1.26E-4	6.21E-5	5.10E-4	2.15E-4	6.03E-1	5.17E-1	2.02E-1	4.21E-1
3	1.31E+4	8.18E+2	1.34E+4	1.06E+3	1.34E+4	1.05E+3	1.34E+4	1.03E+3	1.33E+4	6.49E+2	1.36E+4	8.39E+2
4	5.57E-7	2.91E-7	2.50E-6	7.76E-7	4.92E-7	1.87E-7	1.53E-6	4.94E-7	5.74E-7	1.91E-7	2.45E-6	4.59E-7
5	4.38E+1	2.59E-1	4.43E+1	2.37E-1	4.38E+1	2.31E-1	4.42E+1	1.77E-1	4.47E+1	1.95E+0	4.42E+1	3.09E-1
6	1.17E+0	1.10E-1	1.80E+0	2.35E-1	1.05E+0	1.34E-1	1.47E+0	1.45E-1	1.09E+0	1.60E-1	1.73E+0	1.40E-1
7	1.14E-4	4.50E-5	4.70E-4	1.47E-4	1.01E-4	3.53E-5	6.12E-4	3.19E-4	1.99E-1	4.19E-1	2.00E-1	4.19E-1
8	1.31E+0	1.68E-1	1.83E+0	1.80E-1	1.01E+0	1.75E-1	1.52E+0	1.51E-1	9.23E-1	1.09E-1	1.46E+0	2.00E-1
9	1.31E+4	9.05E+2	1.34E+4	9.11E+2	1.32E+4	8.96E+2	1.37E+4	7.17E+2	1.30E+1	9.97E+2	1.36E+4	6.20E+2

表 5 联合分布适配选取不同迭代次数的对比实验

Table 5 Comparison experiments of different iterations of joint distribution adaptation selection

测试 问题	K=2				K=3				K=5			
	JDA_TrDE ₁		JDA_TrDE ₂		JDA_TrDE ₁		JDA_TrDE ₂		JDA_TrDE ₁		JDA_TrDE ₂	
	均值	标准差	均值	标准差	均值	标准差	均值	标准差	均值	标准差	均值	标准差
1	9.94E-5	2.64E-5	5.10E-4	1.24E-4	1.32E-4	3.97E-5	4.43E-4	1.77E-4	1.73E-4	7.59E-5	3.70E-4	1.42E-4
2	1.14E-4	3.97E-5	4.93E-4	1.74E-4	1.17E-4	4.64E-5	4.20E-4	1.42E-4	1.16E-4	4.61E-5	5.00E-4	1.42E-4
3	1.29E+4	7.14E+2	1.36E+4	1.06E+3	1.26E+4	4.15E+2	1.36E+4	8.58E+2	1.32E+4	7.49E+2	1.36E+4	6.68E+2
4	5.67E-7	1.32E-7	2.21E-6	7.02E-7	6.14E-7	2.53E-7	2.26E-6	6.56E-7	5.56E-7	1.69E-7	2.46E-6	1.23E-6
5	4.38E+1	2.17E-1	4.43E+1	1.52E-1	4.38E+1	1.49E-1	4.42E+1	2.91E-1	4.37E+1	2.25E-1	4.44E+1	1.64E-1
6	1.24E+0	1.50E-1	1.70E+0	1.86E-1	1.16E+0	1.64E-1	1.86E+0	2.43E-1	1.18E+0	1.74E-1	1.86E+0	1.87E-1
7	1.13E-4	5.40E-5	5.10E-4	2.71E-4	9.35E-5	2.42E-5	4.30E-4	1.08E-4	1.28E-4	5.18E-5	4.12E-4	1.10E-4
8	1.28E+0	1.95E-1	1.74E+0	2.31E-1	1.21E+0	1.58E-1	1.93E+0	2.18E-1	1.25E+0	1.78E-1	1.76E+0	2.50E-1
9	1.30E+4	1.02E+3	1.36E+4	5.90E+2	1.30E+4	7.54E+2	1.30E+4	5.64E+2	1.28E+4	9.65E+2	1.39E+4	1.36E+3

4 结论

本文提出了基于联合分布适配的单向迁移差分进化算法求解单目标优化问题,实现了 2 种迁移模式的算法即初始迁移的 JDA_TrDE₁ 和过程迁移的 JDA_TrDE₂。采用 9 组函数进行测试,并与原始差分算法(DE)、无适配技术的直接迁移差分算法(TrDE)进行比较,实验结果表明:引入 JDA 后的算法不仅能更快地跳出局部最优,提高算法的求解精度,而且具有较好的收敛速度。此外实验结果还得出问题间的关系对算法性能有一定的影响,迁移模式和 JDA 的参数对算法性能影响不大。本文仅仅对单目标优化问题进行了测试,下一步考虑将基于 JDA 的迁移优化应用到约束优化问题 and 多目标优化问题,以全面评价所提方法的性能。另外,本文采用的测试函数具有一定的先验知识,源问题和目标问题的相交程度和相似度是已知的,如果是在任意的 2 个问题中迁移,需要使用 JDA 评测问题间的关系,并根据相似关系进行自适应迁移,这些也是后续的研究内容。

参考文献:

[1] DAS S, SUGANTHAN P N. Differential evolution: a survey of the state-of-the-art[J]. IEEE Transactions on Evolutionary Computation, 2011, 15(1): 4-31.

[2] 程适,王锐,伍国华,等. 群体智能优化算法[J]. 郑州大学学报(工学版), 2018,39(6): 1-2.

CHENG S, WANG R, WU G H, et al. Swarm intelligence optimization algorithm[J]. Journal of Zhengzhou University (Engineering Science), 2018,39(6): 1-2.

[3] 刘倩,冯艳红,陈巍瑛. 基于混沌初始化和高斯变异的飞蛾火焰优化算法[J]. 郑州大学学报(工学版), 2021, 42(3): 53-58.

LIU Q, FENG Y H, CHEN Y Y. Moth-flame optimization algorithm based on chaotic initialization and Gaussian mutation[J]. Journal of Zhengzhou University (Engineering Science), 2021, 42(3): 53-58.

[4] 汪慎文,张佳星,褚晓凯,等. 两阶段搜索的多模态多目标差分进化算法[J]. 郑州大学学报(工学版), 2021, 42(1): 9-14, 110.

WANG S W, ZHANG J X, CHU X K, et al. Multimodal multi-objective differential evolution algorithm based on two-stage search[J]. Journal of Zhengzhou University (Engineering Science), 2021, 42(1): 9-14, 110.

[5] TAN Z P, LI K S, WANG Y. Differential evolution with adaptive mutation strategy based on fitness landscape analysis[J]. Information Sciences, 2021, 549: 142-163.

[6] PAN S J, YANG Q. A survey on transfer learning[J]. IEEE Transactions on Knowledge and Data Engineering, 2010, 22(10): 1345-1359.

[7] XU H P, YANG L Z, LONG X Y. Underwater sonar image classification with small samples based on parameter-based transfer learning and deep learning[C]//2022 Global Conference on Robotics, Artificial Intelligence and Information Technology (GCRAIT). Piscataway: IEEE, 2022: 304-307.

[8] CHIBA S, SASAOKA H. Basic study for transfer learning for autonomous driving in car race of model car[C]//2021 6th International Conference on Business and Industrial Research (ICBIR). Piscataway: IEEE, 2021: 138-141.

[9] SINGH N K, KUMAR A. Image based recommender system using transfer learning[C]//2022 2nd International Conference on Emerging Frontiers in Electrical and Electronic Technologies (ICEFEET). Piscataway: IEEE, 2022: 1-4.

[10] TAN K C, FENG L, JIANG M. Evolutionary transfer optimization-a new frontier in evolutionary computation re-

search[J]. IEEE Computational Intelligence Magazine, 2021, 16(1): 22–33.

[11] LIM R, GUPTA A, ONG Y S, et al. Non-linear domain adaptation in transfer evolutionary optimization[J]. Cognitive Computation, 2021, 13(2): 290–307.

[12] JIANG M, HUANG Z Q, QIU L M, et al. Transfer learning-based dynamic multiobjective optimization algorithms[J]. IEEE Transactions on Evolutionary Computation, 2018, 22(4): 501–514.

[13] JIANG M, QIU L M, HUANG Z Q, et al. Dynamic multi-objective estimation of distribution algorithm based on domain adaptation and nonparametric estimation[J]. Information Sciences, 2018, 435: 203–223.

[14] 张勇, 杨康, 郝国生, 等. 基于相似历史信息迁移学习的进化优化框架[J]. 自动化学报, 2021, 47(3): 652–665.

ZHANG Y, YANG K, HAO G S, et al. Evolutionary optimization framework based on transfer learning of similar historical information[J]. Acta Automatica Sinica, 2021, 47(3): 652–665.

[15] 刘璐, 蒋艳. 基于迁移学习的 NSGA II 算法[J]. 软件导刊, 2021, 20(3): 134–138.

LIU L, JIANG Y. NSGA II algorithm based on transfer learning[J]. Software Guide, 2021, 20(3): 134–138.

[16] 尚青霞, 周磊, 冯亮. 基于降噪自动编码器的多任务优化算法[J]. 大连理工大学学报, 2019, 59(4): 417–426.

SHANG Q X, ZHOU L, FENG L. Multi-task optimization algorithm based on denoising auto-encoder[J]. Journal of Dalian University of Technology, 2019, 59(4): 417–426.

[17] FENG L, ZHOU L, ZHONG J H, et al. Evolutionary multitasking via explicit autoencoding[J]. IEEE Transactions on Cybernetics, 2019, 49(9): 3457–3470.

[18] PAN S J, TSANG I W, KWOK J T, et al. Domain adaptation via transfer component analysis[J]. IEEE Transactions on Neural Networks, 2011, 22(2): 199–210.

[19] GONG B Q, SHI Y, SHA F, et al. Geodesic flow kernel for unsupervised domain adaptation[C]//IEEE Conference on Computer Vision and Pattern Recognition. Piscataway: IEEE, 2012: 2066–2073.

[20] LONG M S, WANG J M, DING G G, et al. Transfer feature learning with joint distribution adaptation[C]//2013 IEEE International Conference on Computer Vision. Piscataway: IEEE, 2014: 2200–2207.

[21] DA B S, ONG Y S, FENG L, et al. Evolutionary multitasking for single-objective continuous optimization; benchmark problems, performance metric and baseline results[EB/OL]. (2017–06–12) [2022–10–26]. <https://doi.org/10.48550/arXiv.1706.03470>.

Unidirectional Transfer Differential Evolution Algorithm Based on Joint Distribution Adaptation

LI Xi^{1,2}, LI Shuai^{1,2}, FENG Yanhong¹, LI Mingliang¹

(1. School of Information Engineering, Hebei GEO University, Shijiazhuang 050031, China; 2. Laboratory of AI and Machine Learning, Hebei GEO University, Shijiazhuang 050031, China)

Abstract: In traditional differential evolution algorithm solves optimization problems generally starting from zero-knowledge and searching independently without using the information of similar problems that have been solved. In this paper, the transfer learning technique was introduced into DE. Firstly, the extracted key information from the optimized population of the source problem and the current population of the target problem was mapped to the high-dimensional Hilbert space by the joint distribution adaptation method. Then a new population was constructed from the mapped matrix to replace the population of the target problem. Lastly, the subsequent evolution was completed. Two transfer modes were implemented: transferring the effective information of the source problem to guide the search direction of the algorithm during the initialization; utilizing the transferred effective information after a certain number of iterations to accelerate the population convergence. The proposed algorithm was tested with nine sets of multi-task test functions, and compared with the DE without transfer and transferred DE without adaptation technique. The results showed that, the proposed algorithm outperforms the traditional DE on seven sets functions in term of solution quality. Moreover, the proposed algorithm converged faster than DE. Therefore, the transfer learning-based differential evolution algorithm was effective in improving the solution accuracy and convergence speed of the objective optimization problem.

Keywords: optimization algorithms; transfer learning; joint distribution adaptation; unidirectional transfer; differential evolutionary