

基于相似历史信息迁移学习的进化优化框架

张勇¹ 杨康¹ 郝国生² 巩敦卫¹

摘要 现有进化算法大都从问题的零初始信息开始搜索最优解, 没有利用先前解决相似问题时获得的历史信息, 在一定程度上浪费了计算资源. 将迁移学习的思想扩展到进化优化领域, 本文研究一种基于相似历史信息迁移学习的进化优化框架. 从已解决问题的模型库中找到与新问题匹配的历史问题, 将历史问题对应的知识迁移到新问题的求解过程中, 以提高种群的搜索效率. 首先, 定义一种基于多分布估计的最大均值差异指标, 用来评价新问题与历史模型之间的匹配程度; 接着, 将相匹配的历史问题的知识迁移到新问题中, 给出一种基于模型匹配程度的进化种群初始化策略, 以加快算法的搜索速度; 然后, 给出一种基于迭代聚类的代表个体保存策略, 保留求解过程中产生的优势信息, 用于更新历史模型库; 最后, 将自适应骨干粒子群优化算法嵌入到所提框架, 给出一种基于相似历史信息迁移学习的骨干粒子群优化算法. 针对多个改进的典型测试函数, 实验结果表明, 所提迁移策略可以加速粒子群的搜索过程, 显著提高算法的收敛速度和搜索效率.

关键词 进化优化, 迁移学习, 粒子群优化, 模型匹配

引用格式 张勇, 杨康, 郝国生, 巩敦卫. 基于相似历史信息迁移学习的进化优化框架. 自动化学报, 2021, 47(3): 652–665

DOI 10.16383/j.aas.c180515

Evolutionary Optimization Framework Based on Transfer Learning of Similar Historical Information

ZHANG Yong¹ YANG Kang¹ HAO Guo-Sheng² GONG Dun-Wei¹

Abstract Most of existing evolutionary algorithms search optimal solutions from zero initial information of a given problem. Because of the lacking a mechanism to use historical information, this must waste computing resources to some extent when solving a problem similar to old ones. This paper extends the idea of transfer learning to the field of evolutionary optimization, and studies a new evolutionary optimization framework based the transfer learning of similar historical information. To improve the search efficiency of the population, the proposed framework finds out the history problem from the model library, which matches current new problem, and transfers the knowledge of the historical problem into optimization process of the new problem. First, a maximum mean discrepancy indicator based on multi-distribution estimation is defined to evaluate the matching degree between a new problem and historical models. Secondly, the knowledge of matched historical problem is transferred into the new problem, and a new initialization strategy of the population based on matching degree is given to accelerate the search speed of the algorithm. Next, a preservation strategy based on iterative clustering is presented to save good information generated during the evolutionary process, for updating the model library. Finally, embedding an adaptive bare-bones particle swarm optimization (PSO) into the proposed framework, a bare-bones PSO algorithm based on the transfer learning of similar historical information is presented. Testing on several improved typical functions, experimental results show that the proposed transfer strategy accelerates the search process of the particle swarm, and significantly improve the convergence speed and the search efficiency of the proposed PSO algorithm.

Key words Evolutionary optimization, transfer learning, particle swarm optimization (PSO), model matching

Citation Zhang Yong, Yang Kang, Hao Guo-Sheng, Gong Dun-Wei. Evolutionary optimization framework based on transfer learning of similar historical information. *Acta Automatica Sinica*, 2021, 47(3): 652–665

收稿日期 2018-07-27 录用日期 2019-03-25
Manuscript received July 27, 2018; accepted March 25, 2019
国家自然科学基金 (61876185, 61573364, 61573362) 资助
Supported by National Natural Science Foundation of China (61876185, 61573364, 61573362)

本文责任编辑 魏庆来

Recommended by Associate Editor WEI Qing-Lai

1. 中国矿业大学信息与控制工程学院 徐州 221116 2. 江苏师范大学
计算机科学与技术学院 徐州 221116

1. School of Information and Control Engineering, China University of Mining and Technology, Xuzhou 221116 2. School of Computer Science and Technology, Jiangsu Normal University, Xuzhou 221116

进化优化是模拟生物进化行为和机制产生的一类迭代搜索算法, 目前已广泛应用于模式识别、经济管理、电气工程和生物学等众多领域^[1]. 典型进化优化技术包含遗传算法 (Genetic algorithm, GA)、进化策略 (Evolution strategy, ES) 和进化规划 (Evolutionary programming, EP) 等. 由于同样启发于生物进化行为, 部分学者也将粒子群优化和蚁群优化等群体智能优化算法归为进化优化技术. 该类算法都是从选定的初始种群出发, 通过不断迭代更新

种群中个体位置,直至搜索到最适合问题或任务的解.通常,这些算法所依赖的初始种群都是在问题可行域中生成的随机位置.换句话说,现有进化优化算法,如文献[2-8],都是从问题的零初始信息开始搜索,并没有考虑是否以前优化过类似问题,是否可以从历史信息中获得解决当前相似问题的能力.例如,在车间调度问题中,当前月份的生成任务可能与往年某一月份的任务相同或相似.在利用进化优化寻找当前月份最佳调度序列时,如果能够充分利用相似任务的历史信息,势必可以提高其搜索能力.

迁移学习(Transfer learning)是一种人性化的机器学习方法,其目的是把一个领域(即源领域)的知识迁移到另外一个领域(即目标领域),使得目标领域能够取得更好的学习效果,即使用以前解决相似任务时获得的信息来帮助解决新问题^[9].目前迁移学习已广泛应用于图形识别、文本分类、网页分类等诸多问题^[10-15].但是,将迁移学习应用于进化优化中,相关研究仍然很少.Dinh等^[16]和Koçer等^[17]在使用遗传算法处理源域任务时,保存得到的最优、中等和最差等个体;当处理新的目标任务时,迁移这些个体并随机代替初始种群中的部分个体;Feng等^[18]将迁移学习融合到进化优化中,提出了一种新的文化基因进化框架;最近,Jiang等^[19]成功地将迁移学习用于解决动态多目标优化问题,显著提高了进化优化算法对环境变化的响应速度.

上述方法的成功应用充分说明了利用迁移学习提高进化算法性能的可行性.然而,相关研究起步较晚,仍然存在如下不足或可改进之处:1)在构造迁移学习的源域样本时,已有工作仅保存种群中最优、中等或者最差的个体,没有考虑所保存个体的多样性.受益于进化优化的优胜劣汰机制,种群中的所有个体往往会收敛到搜索空间中的一个点或一个很小的区域.此时,种群中的最优、中等和最差可能十分相似.在源域中保存诸多相似的个体,不免造成样本空间分布的过拟合现象.2)现有工作大都假设源域和目标域已经相似或匹配,并未给出源域和目标域相似性或匹配程度的判断标准.通常决策者可以获得的相似历史问题或任务往往较多,如何从众多的历史问题或任务中选出最为相似的一个,将直接影响到迁移学习的效果;相反,当源域问题和目标域问题完全不同时,强制迁移可能会污染目标域,产生负面的影响^[20].3)由源域到目标域的知识迁移方法研究不够.文献[16-17]将源域中保存的个体样本直接替代目标问题的初始种群,没有考虑源问题与目标问题之间的差异性.尽管文献[18-19]所设计的知识迁移方法考虑了历史问题与目标问题之间的差异性(或映射关系),但其所得成果皆立足于车辆路径规划和动态多目标优化问题的自身特点,难以用

于本文考虑的静态全局优化问题.

鉴于此,本文研究一种基于历史相似信息迁移学习的进化优化框架,用以解决复杂优化问题.针对某一待处理的新问题,该方法首先从模型库中找到与其最为匹配的历史模型;接着,从历史模型对应的源域知识中提取有用信息,构建源域到目标域的映射关系,并基于该映射关系产生新问题下进化算法的初始种群.本文主要工作如下:1)提出一种基于多分布估计的最大均值差异指标,用来评价新问题与历史问题之间的相似程度;2)基于模型的相似程度,给出一种(历史问题对应的)源域到(待优化问题对应的)目标域的自适应知识迁移策略,初始化当前进化种群中的部分个体,加快进化算法的搜索速度;3)为不断更新和丰富历史模型库,给出一种基于迭代聚类的代表个体保存策略,形成源域的样本集合,同时保证个体样本的多样性和质量;4)将自适应骨干粒子群优化算法嵌入到所提框架,给出一种基于相似历史信息迁移学习的骨干粒子群优化算法.

本文结构如下:第1节介绍相关工作,包括迁移学习和粒子群优化的基本理论;第2节给出所提基于历史相似信息迁移学习的进化优化框架,以及基于迁移学习的改进粒子群优化算法;第3节给出相应的对比实验,验证算法的有效性;第4节对本文工作进行总结.

1 相关工作

1.1 粒子群优化算法

粒子群优化(Particle swarm optimization, PSO)^[21]算法是一种仿生物群体觅食行为的全局随机搜索算法.在求解优化问题时,每个粒子通过记忆和追随两个最优位置,来不断更新自身的位置^[22].一个是该粒子到目前为止自己发现的最好位置 $Pbest = (pbest_{i,1}, pbest_{i,2}, \dots, pbest_{i,N})$,即通常所说的粒子个体最优点或局部引导者;另一个是目前为止邻域粒子发现的最好位置 $Gbest_i = (gbest_{i,1}, gbest_{i,2}, \dots, gbest_{i,N})$,即通常所说的粒子的全局最优点或全局引导者.设粒子 i 在 N 维空间的位置为矢量 $\mathbf{X}_i = (x_{i,1}, x_{i,2}, \dots, x_{i,N})$,飞行速度为矢量 $\mathbf{V}_i = (v_{i,1}, v_{i,2}, \dots, v_{i,N})$,该粒子位置更新式为

$$v_{i,j}(k+1) = \omega v_{i,j}(k) + c_1 r_1 (pbest_{i,j}(k) - x_{i,j}(k)) + c_2 r_2 (gbest_j(k) - x_{i,j}(k)) \quad (1)$$

$$x_{i,j}(k+1) = x_{i,j}(k) + v_{i,j}(k+1) \quad (2)$$

其中, k 为算法迭代次数, ω 为惯性权重, c_1 和 c_2 为学习因子, r_1 和 r_2 为服从均匀分布 $U(0,1)$ 的随机数.

1.2 迁移学习

迁移学习主要包含域和任务两个概念^[23-24]. 通常, 域可以表示为 $D = \{Z, P_{ob}(Z)\}$, 即特征空间 Z 和特征空间的边缘分布 $P_{ob}(Z)$. 在给定一个域 $D = \{Z, P_{ob}(Z)\}$ 的情况下, 一个任务可以表示为 $T = \{Y, f(\cdot)\}$, 即标签空间 Y 和一个目标预测函数 $f(\cdot)$. 考虑存在一个源域 D_s 和一个目标域 D_t 的情况, 不妨设源域 $D_s = \{(z_{s,1}, y_{s,1}), (z_{s,2}, y_{s,2}), \dots, (z_{s,ns}, y_{s,ns})\}$, $z_{s,j} \in Z_s$ 表示源域样本, $y_{s,j} \in Y_s$ 表示源域样本 $z_{s,j}$ 对应的标签. 目标域 $D_t = \{(z_{t,1}, y_{t,1}), (z_{t,2}, y_{t,2}), \dots, (z_{t,nt}, y_{t,nt})\}$, $z_{t,j} \in Z_t$ 表示目标域样本, $y_{t,j} \in Y_t$ 表示目标域样本 $z_{t,j}$ 对应的输出. 基于以上的符号定义, 迁移学习的目的是: 在给定源域 D_s 和源域学习任务 T_s 、目标域 D_t 和目标域任务 T_t , 且满足 $D_t \neq D_s$ 和 $T_t \neq T_s$ 的情况下, 通过迁移学习使用源域 D_s 和 T_s 中的知识, 提升或优化目标域 D_t 中目标预测函数 $f(\cdot)$ 的学习效果^[9].

2 基于历史相似信息迁移学习的进化优化框架

现有进化优化算法大都从给定问题的零初始信息开始, 随机产生初始种群, 并通过迭代搜索找到问题的最优解. 为充分利用相似问题的历史信息, 本节提出一种基于迁移学习的进化优化框架, 如图 1 所示. 在图 1 中, 模型库 (Model library, ML) 保存着进化优化算法所解决过的相似历史信息. 该模型库保存的信息单元是 $ml_i = (f_i, P'_i)$, 其中, f_i 和 P'_i 分别为所保存的第 i 个历史问题及其最优解集信息. 由于需要利用历史问题的解信息来指导新问题的求解, 因此, 模型库中保存的历史问题应该与待求解问题属于同一种 (类) 问题, 两者应该具有相似的编解码策略和问题特征.

当决策者获得一个新的优化问题时, 首先, 对模型库进行预处理, 分析并判断每个历史问题是否与新问题具有相似的编解码策略和特征, 并删除不相似和相似程度低的历史问题; 接着, 利用模型匹配方法匹配模型库中的信息, 计算新问题与模型库中已有历史问题的相似度; 找到匹配程度最高的历史优化问题, 获取其历史数据, 并将历史问题的高质量解迁移到新问题的进化过程中, 用来初始化新问题进化种群中的部分个体; 若没有与之匹配的历史优化问题, 则直接使用进化算法求解优化问题. 在优化当前新问题时, 将问题求解过程中得到的代表信息存储到模型库中, 如此循环, 随着所解决问题数目的不断加入, 模型库不断增大, 进而有助于解决更多的优化问题. 分析上述进化优化框架可知, 该框架存在如下三部分直接影响其最终性能, 即模型匹配、相似历

史数据的利用和模型库的更新. 针对上述三个关键算子, 下面分别给出详细解决方案.

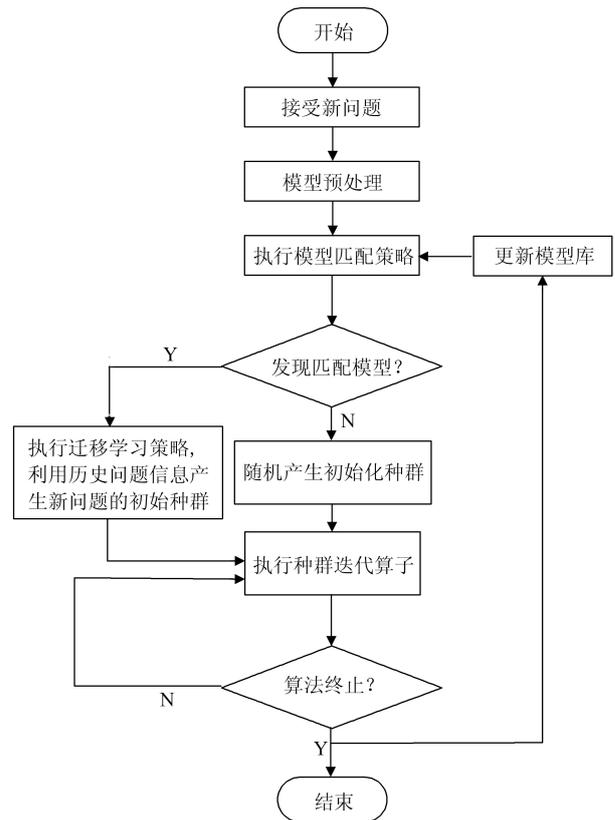


图 1 所提基于迁移学习的进化算法框架

Fig. 1 Evolutionary algorithm framework based on transfer learning

2.1 基于多分布估计的模型匹配策略

将模型库中保存的历史问题信息作为源域知识, 待解决的新问题看作目标域任务, 本文旨在利用已知源域知识去求解目标域中的新问题. 通常源域模型库中存有的历史问题或任务较多, 如果历史问题模型和目标域中新问题的相似度较低或完全不同时, 强制迁移可能会污染目标域, 产生负迁移现象. 因此, 如何准确判断源域和目标域中问题的相似程度, 从众多的历史问题或任务中选出最为相似的一个, 将直接影响到目标域优化问题的求解效率.

最大均值差异 (Maximum mean discrepancy, MMD) 是一种用来判断两个数据分布是否相同的指标, 最初主要用于双样本的检测问题. 近些年, 学者们开始将其用于迁移学习中, 利用源域数据集和目标域数据集的均值差来表示源域和目标域样本之间的分布差异^[25]. MMD 的基本原理如下: 假设有一个满足 Q_1 分布的源域数据集 $X^s = \{x_1^s, x_2^s, \dots, x_n^s\}$ 和一个满足 Q_2 分布的目标域数据集 $X^t = \{x_1^t, x_2^t, \dots, x_m^t\}$; 令 H 为再生希尔伯特空间 (Re-

producing kernel Hilbert space, RKHS), 并且存在一个从原始空间到希尔伯特空间的映射函数 $F(\cdot) : X \rightarrow H$, 那么, 当 n 和 m 趋于无穷时, X^s 和 X^t 在 RKHS 上的最大均值差异可以表示为

$$MMD(X^S, X^T) = \left\| \frac{1}{n} \sum_{i=1}^n F(x_i^s) - \frac{1}{m} \sum_{i=1}^m F(x_i^t) \right\| \quad (3)$$

然而, 不同于传统意义上的采样数据 (如传感器数据、图像数据等), 对于进化种群产生的源域或目标域样本而言, 由于所优化问题通常包含若干个峰值, 此时将源域或目标域中所有样本看成服从同一分布的整体, 并采用 MMD 判断源域和目标域问题的相似性, 将很难准确评价出它们的相似程度.

鉴于此, 本文提出一种基于多分布估计的最大均值差异指标, 用来评价新问题与历史问题之间的匹配程度, 思想如下: 首先, 采用 K-means 算法对源域和目标域中样本分别进行聚类, 并假设聚类后每一类中的样本服从同一分布 (注: 传统方法假设源域 (目标域) 中所有样本服从同一分布); 随后, 针对目标域中的每一类, 从源域中找到与其最为匹配的类, 即 MMD 值最小的类; 最后, 利用目标域中所有样本类的 MMD 值的平均值, 作为源域和目标域样本的整体相似性程度.

算法 1 给出了基于多分布估计的模型匹配策略的具体步骤. 对于一个新给定的优化问题, 首先, 步骤 1 随机初始化 n 个个体, 运行少量代数 T_{low} 的进化优化算法, 并利用进化后的种群个体作为目标域样本. 具体地, 不妨设 f_t 为目标域中待求解的优化问题, P_t 为运行少量代数后进化算法得到的种群 (或解集), $P_t = \{p_{t,1}, p_{t,2}, \dots, p_{t,n}\}$; 将 P_t 代入优化问题 f_t 中, 即可得到目标域样本集 $T_S = \{(p_{t,1}, f_t(p_{t,1})), (p_{t,2}, f_t(p_{t,2})), \dots, (p_{t,n}, f_t(p_{t,n}))\}$. 其次, 步骤 2 从历史模式库中按照顺序选定一个历史问题, 从其保存的信息中提取与目标域信息相对应的样本, 作为源域样本. 具体地, 不妨设 f_s 为源域中指定的历史优化问题, P'_s 为模型库中历史优化问题 f_s 的解信息, $P'_s = \{p'_{s,1}, p'_{s,2}, \dots, p'_{s,m}\}$; 将 P'_s 代入优化问题 f_s 中, 得到源域样本集 $O_S = \{(p'_{s,1}, f_s(p'_{s,1})), (p'_{s,2}, f_s(p'_{s,2})), \dots, (p'_{s,m}, f_s(p'_{s,m}))\}$. 接着, 步骤 3 使用 K-means 算法对源域和目标域中样本分别进行聚类, 得到源域的样本类为 $O_S = \{O_{s_1}, O_{s_2}, \dots, O_{s_{ck}}\}$, 目标域的样本类为 $T_S = \{T_{s_1}, T_{s_2}, \dots, T_{s_{ck}}\}$, ck 为样本聚类个数. 随后, 步骤 4 评价源域历史问题和目标域问题的相似程度值 S_{de} . 具体地, 步骤 4.1 针对目标域中的每个样本类, 不妨以第 q 的样本类 $T_{s_q} = (t_{s_q,1}, t_{s_q,2}, \dots, t_{s_q,|T_{s_q}|})$, $q = 1, 2, \dots, ck$ 为例, 从源域所有样本类 $O_S = \{O_{s_1}, O_{s_2}, \dots,$

$O_{s_{ck}}\}$ 中寻找与其差异最小的样本类 $O_{s_j} = (o_{s_j,1}, o_{s_j,2}, \dots, o_{s_j,|O_{s_j}|})$, $j = 1, 2, \dots, ck$, 计算出类 T_{s_q} 相对 O_S 所有类的最大均值差异, 即

$$MMD(T_{s_q}, O_{s_j}) = \min_{j=1,2,\dots,ck} \left(\left\| \frac{1}{|O_{s_j}|} \sum_{h=1}^{|O_{s_j}|} F(o_{s_j,h}) - \frac{1}{|T_{s_q}|} \sum_{h=1}^{|T_{s_q}|} F(t_{s_q,h}) \right\| \right) \quad (4)$$

算法 1. 模型匹配策略

输入. 历史模型库 $ML = \{(f_1, P'_1), (f_2, P'_2), \dots, (f_{pn}, P'_{pn})\}$, 待优化问题 F_t , 聚类数目 ck , 进化代数 T_{low} .

输出. 所得相似问题及其解集信息.

步骤 1. 运行少量代数 T_{low} 的进化算法, 产生目标域样本 $T_S = \{P_t, F(P_t)\}$.

步骤 2. $i = 1$, 从模型库中选定第 i 个历史问题 (F_i, P'_i) , 确定源域样本 $O_S = \{P'_s, F(P'_s)\}$.

步骤 3. 使用 K-means 算法对源域和目标域中样本分别进行聚类, 得到源域的样本类为 $O_S = \{O_{s_1}, O_{s_2}, \dots, O_{s_{ck}}\}$, 目标域的样本类为 $T_S = \{T_{s_1}, T_{s_2}, \dots, T_{s_{ck}}\}$.

步骤 4. 评价源域历史问题 F_i 和待优化问题 F_t 的相似程度值, 方法如下:

步骤 4.1. 针对目标域中的每个样本类 T_{s_q} , $q = 1, 2, \dots, ck$, 利用式 (4) 计算 T_{s_q} 与源域样本的最大均值差异值 $MMD(T_{s_q}, O_S)$;

步骤 4.2. 计算目标域中所有 ck 个样本类的最大均值差异的平均值, 并取其倒数为历史问题 F_i 和待优化问题 F_t 的相似程度值 $S_{de}(F_i, F_t)$;

步骤 4.3. $i = i + 1$, 返回步骤 2, 直到求得所有历史问题和待优化问题 F_t 的相似程度值 $S_{de}(F_i, F_t)$, $i = 1, 2, \dots, pn$.

步骤 5. 确定相似程度值最大的历史问题, 作为匹配结果.

步骤 4.2 利用目标域中所有样本类的最大均值差异的平均值, 表示源域和目标域的整体相似性程度. 该平均值越小, 源域和目标域中样本的分布越相似, 进而历史问题和新问题越相似. 循环执行步骤 2 ~ 4, 直到计算出新问题与所有历史问题相似程度值. 最后, 步骤 5 选择相似程度值最高 (即最大均值差异的平均值最小) 的历史问题, 作为最终匹配结果. 注意: 当源域和目标域样本数据具有不同的量纲时, 需要对这些样本进行归一化. 式 (4) 中映射函数选择最为简单的单位映射.

本文通过运行一定迭代次数 (T_{low}) 的智能优化

算法产生目标样本。一方面,对于不同规模、不同操作流程的算法来说,它们通常具有不同的求解速度;另一方面,求解问题的复杂程度也对算法的迭代效率产生较大影响。因此,很难准确估计出可以产生高质量样本(即可以高精度刻画出问题解分布特性的个体)的迭代次数。算法 1 利用少量迭代后的种群进化结果作为目标域样本,具有如下特点: 1) 执行少量迭代次数后,种群中的部分个体会接近多个最优区域,而不会集中收敛到某个最优区域,此时的种群样本不仅具有较好的适应值,而且多样性相对较高;相反,如果迭代次数设置过大,在很大程度上种群中的大部分个体会集中收敛到某个最优区域,此时的种群样本虽然具有较好的适应值,但其多样性相对较差。2) 尽管通过加大种群的迭代次数,可以找到更为接近问题最优解的样本,但是,需要付出的计算代价也会随之增加。更重要的是,在很多情况下,只需要找到较为接近问题最优解的样本,即可判断出两个问题的相似性。因此,在满足样本基本需要的基础上,所提目标域样本产生方法还可以显著减少样本的产生代价。

关于算法 1 中迭代次数 T_{low} ,一方面,由于无法事先判断不同智能算法处理不同问题时的迭代速度与精度,很难给出适合所有算法的固定 T_{low} 值;另一方面,如前所述,在很多情况下只需要找到较为接近问题最优解的样本,即可估计出两个问题的相似性。因此,在一定程度上,可以放松对迭代次数 T_{low} 取值的要求,本文建议 T_{low} 取值为最大迭代次数的 [1%, 5%]。具体而言,对于收敛速度较快的智能算法,可以取值为上述区间的下限值;相反,针对收敛速度较慢的智能算法,可以取值为上述区间的上限值。

2.2 源域匹配信息的利用

为加快进化算法的搜索速度,给出一种基于源域匹配信息的种群初始化策略。当从源域中找到与目标域问题相匹配的历史问题后,迁移源域中相似历史问题的信息(解集信息),用其生成新个体,并利用这些新个体替换目标域中进化种群的部分个体;为了增加种群多样性、防止出现过迁移和欠迁移现象,根据历史问题与新问题的相似度自动设置替换初始种群的比例,即种群中被初始化个体的比例,具体计算式为

$$ratio = (r_{max} - r_{min}) \times S_{de} + r_{min} \quad (5)$$

其中, S_{de} 为历史问题与新问题的相似度值, r_{max} 表示利用迁移信息初始化种群个体的比例上限, r_{min} 表示利用迁移信息初始化种群个体的比例下限。为保证种群多样性,本文利用迁移信息来初始化种群

中的部分个体,并通过 r_{max} 和 r_{min} 值来限制初始化个体的比例 $ratio$ 。可以看出, $ratio$ 取值不能过大,迁移比例设置过大会引起负迁移现象;而 $ratio$ 取值过小时,由于被初始化个体数目过少,迁移效果不明显,进而出现弱迁移现象。鉴于此,本文设置 $r_{max} = 0.5$ 和 $r_{min} = 0.1$ 。

不妨设第 2.1 节从源域中得到的匹配解集为 $P_s = \{p_{s,1}, p_{s,2}, \dots, p_{s,m}\}$, 运行进化算法后得到的目标域解集为 $P_t = \{p_{t,1}, p_{t,2}, \dots, p_{t,N}\}$, 需要借助历史问题信息生成的新个体数目为 $ratio \times N$, 其中, N 为进化种群的规模。算法 2 给出了基于源域匹配信息的种群初始化策略的具体步骤。首先,步骤 1 从种群 P_t (最优个体除外) 中随机选出 $N' = ratio \times N$ 个个体,组成待初始化个体集合 $P'_t = \{p'_{t,1}, p'_{t,2}, \dots, p'_{t,N'}\}$ 。注意:为防止丢失最优进化信息,种群中最优个体不在选择范围之列。步骤 2 针对集合 P'_t 中每个个体,从源域匹配解集 P_s 中找到距离最近的元素;按照与 P'_t 中个体顺序对应的方式,对这些元素进行排序,得到一个源域匹配集合 $P'_s = \{p'_{s,1}, p'_{s,2}, \dots, p'_{s,N'}\}$ 。步骤 3 基于源域匹配集合 P'_s , 采用式 (6) 更新 P'_t 中所有个体,产生 N' 个新个体,即

$$\begin{pmatrix} \tilde{p}_{t,1} \\ \tilde{p}_{t,2} \\ \vdots \\ \tilde{p}_{t,N'} \end{pmatrix} = \begin{pmatrix} p'_{t,1} \\ p'_{t,2} \\ \vdots \\ p'_{t,N'} \end{pmatrix} + \begin{pmatrix} p'_{t,1} - p'_{s,1} & 0 & 0 & 0 \\ 0 & p'_{t,2} - p'_{s,2} & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & p'_{t,N'} - p'_{s,N'} \end{pmatrix} \times \Delta V \quad (6)$$

其中, ΔV 是元素服从标准正态分布的随机向量,即 $\Delta V = [N(0,1), N(0,1), \dots, N(0,1)]^T$ 。特别地,当仅考虑 P'_t 中第 i 个待初始化个体时,式 (6) 可以简化为

$$\tilde{P}_{t,i} = P'_{t,i} + (P'_{t,i} - P'_{s,i}) \times N(0,1) \quad (7)$$

其中, $\tilde{P}_{t,i}$ 表示产生的新个体, $P'_{t,i}$ 表示从原先种群中选中的待初始化个体, $P'_{s,i}$ 表示从历史解信息中得到的距离 $P'_{t,i}$ 最近的匹配个体, $N(0,1)$ 为服从标准正态分布的随机数。最后,步骤 4 将 N' 个新个体 $\{\tilde{p}_{t,1}, \tilde{p}_{t,2}, \dots, \tilde{p}_{t,N'}\}$ 插入种群 P_t , 替换原先对应的个体。

算法 2. 基于源域匹配信息的种群初始化策略

输入. 源域匹配解集 $P_s = \{p_{s,1}, p_{s,2}, \dots, p_{s,m}\}$, 目标域种群 $P_t = \{p_{t,1}, p_{t,2}, \dots, p_{t,N}\}$, 源域问题与新问题相似程度 S_{de} .

输出. 初始化后的种群 P_t .

步骤 1. 从种群 P_t 中随机选出部分个体, 组成待初始化个体集合 $P'_t = \{p'_{t,1}, p'_{t,2}, \dots, p'_{t,N'}\}$.

步骤 2. 针对集合 P'_t , 利用 P_s 中解产生源域匹配集合 $P'_s = \{p'_{s,1}, p'_{s,2}, \dots, p'_{s,N'}\}$.

步骤 3. 基于源域匹配集合 P'_s , 采用式 (6) 产生 N' 个新个体.

步骤 4. 将 N' 个新个体插入种群 P_t , 替换原先对应的个体, 输出更新后的种群 P_t .

2.3 历史模型库的更新

对于已经解决的历史问题, 保留求解问题过程中产生的历史解信息, 用于模型库的更新. 由于种群的进化过程通常产生大量个体信息, 保存所有个体信息代价较大. 更为重要的是, 这些个体信息中可能存在大量低价值甚至冗余的信息.

为此, 本文给出一种基于迭代聚类的代表个体保存策略. 该策略采用一种简化的迭代 K-means 算法, 选择种群迭代过程中产生的适应值高且多样性好的代表解集. 进一步, 为减少模型库更新的计算代价, 设定一个更新频率 $0 \leq \mu \leq 1$, 每间隔 $[\mu \times T_{\max}]$ 代采样一次种群进化信息, 用来更新所保存的代表个体. 针对当前优化问题, 基于迭代聚类的代表个体保存策略的步骤如下.

步骤 1. 初始化代表个体保存集合, 以及初始类半径. 对第 T_{low} 次迭代产生的种群, 使用 K-means 算法 (在决策变量空间中) 对其个体进行聚类, 将其划分到 ck 个不同类中; 从每一个类中选出适应值最好的个体 (即目标函数值最小的个体), 保存到代表个体保存集合. 计算集合中个体之间的两两距离, 取距离最小值为初始类半径.

步骤 2. 每间隔 $[\mu \times T_{\max}]$ 代采样一次种群进化信息, 更新代表个体保存集合中的元素 (解) 及类半径, 直到算法结束. 以某次更新为例, 设现有代表个体集合为 $A = \{a_1, a_2, \dots, a_{ck}\}$, 类半径为 r' , 当前种群为 $P = \{p_1, p_2, \dots, p_N\}$, 从 P 中第 1 个个体起, 依次检查它与集合 A 中元素的位置和优劣关系, 由下述策略更新 A 中元素, 直到检查完种群中所有 N 个个体. 以 P 中第 i 个个体 p_i 为例, 更新策略如下: 先计算个体 p_i 与集合 A 中每个元素之间的距离, 从 A 中找到距离 p_i 最近的元素, 不妨设为 a_{\min} , 记两者距离为 $d(p_i, a_{\min})$; 接着, 判断两者之间的位置和优劣关系: 1) 如果 $d(p_i, a_{\min}) \leq r'$ 且 p_i 的适应值优于 a_{\min} 的适应值 (即 p_i 的目标函数值小于 a_{\min} 的目标函数值), 那么利用 p_i 代替 A 中元素

a_{\min} ; 2) $d(p_i, a_{\min}) \leq r'$, 但 p_i 的适应值劣于 a_{\min} 的适应值 (即 p_i 的目标函数值大于 a_{\min} 的目标函数值), 保持 A 中元素 a_{\min} ; 3) 如果 $d(p_i, a_{\min}) > r'$, 且 p_i 的适应值优于 A 中最差元素的适应值 (即 p_i 的目标函数值小于 A 中最差元素的目标函数值), 那么, 利用 p_i 替换 A 中最差元素; 4) 否则, 删除 p_i , 保持 A 中元素不变. 最后, 计算集合 A 中个体之间的两两距离, 取距离最小值为新的类半径.

一方面, 上述方法采用 K-means 思想来更新代表解集, 可以保持所选代表个体的多样性; 另一方面, 当新个体与 A 中某个元素同属一个类时, 只保留适应值大的个体或元素, 这样可以保证所选代表个体具有高的适应值. 另外, 除在开始阶段采用传统 K-means 算法来产生初始化代表个体外, 在后续更新 A 中元素时, 上述方法采用简化的 K-means 算法, 无需反复更新类中心, 可以有效减少模型库更新的计算复杂度.

2.4 基于相似历史信息迁移的骨干粒子群优化算法

为了进一步说明所提进化优化框架的有效性, 将一种改进的骨干粒子群优化算法 (Bare-bone PSO, BBPSO)^[26] 嵌入到所提框架, 给出一种基于相似历史信息迁移的骨干粒子群优化算法 (Adaptive bare-bone particle swarm optimization based on transfer learning, TL-ABPSO). 骨干粒子群优化算法 (BBPSO) 是一种改进版本的粒子群优化算法^[27]. 与采用更新式 (1) 和式 (2) 的传统粒子群优化算法相比, BBPSO 删除了粒子速度项、加速系数、惯性权值等控制参数, 是一种少控制参数的新型粒子群优化算法, 算法结构更为简单, 更加易于操作. 然而, 由于删除了粒子速度项, BBPSO 算法存在易于陷入局部收敛的不足^[27-28]. 为此, 在先前工作中作者提出了一种改进 BBPSO 算法, 即自适应骨干粒子群算法 (Adaptive bare-bone PSO, ABPSO)^[26]. 该算法利用当前粒子 X_i 与最佳粒子 $Gbest$ 之间适应值的差值, 自适应扰动新生粒子位置, 显著提高了算法的全局搜索能力. 以粒子 $X_i = (x_{i,1}, x_{i,2}, \dots, x_{i,N})$ 为例, ABPSO 所提粒子位置更新式为

$$x_{i,j}(t+1) = \begin{cases} g_{i,j}(t) + \sigma_{i,j}(t)N(0,1), & rand < prob \\ x_{i,j}(t), & \text{否则} \end{cases}$$

$$g_{i,j}(t) = 0.5(pbest_{i,j}(t) + gbest_j(t))$$

$$\sigma_{i,j}(t) = |pbest_{i,j}(t) - gbest_j(t)| + \Delta$$

$$\Delta = rand \times |pbest_{n1,j}(t) - pbest_{n2,j}(t)| \times$$

$$\exp(f_t(Gbest(t)) - f_t(X_i(t))) \quad (8)$$

其中, $rand$ 为 $[0, 1]$ 之间的随机数, $Pbest_{n1}(t)$ 和 $Pbest_{n2}(t)$ 为随机选择的两个粒子的局部引导者; 同上, $Pbest_i(t)$ 和 $Gbest(t)$ 分别为当前粒子的局部和全局引导者, f_t 为目标域中被优化的新问题.

鉴于 ABPSO 具有控制参数少、全局搜索能力强的优点, 本文将将其嵌入到上述所提进化框架, 算法 3 给出了基于相似历史信息迁移的改进骨干粒子群优化算法的执行步骤. 对于一个新的优化问题, 首先随机初始化粒子群, 运行 T_{low} 代 ABPSO, 产生一个目标域样本; 接着, 利用第 2.1 节方法从模型库中寻找相匹配的历史问题. 如果发现匹配模型, 利用第 2.2 节方法迁移源域中相似历史信息的信息, 初始化第 T_{low} 代粒子群中部分粒子的位置, 继续执行 ABPSO 相关算子, 更新粒子位置; 否则, 直接执行 ABPSO 相关算子. 在算法迭代过程中, 每间隔 $[\mu \times T_{max}]$ 代运行一次第 2.3 节方法, 产生问题的代表个体集合. 所提算法结束后, 将被优化问题及算法输出的最终代表个体集合保存到模型库中.

算法 3. 基于相似历史信息迁移的骨干粒子群优化算法

输入. 模型库 ML , 粒子群规模 N , 代表个体集合的更新频率 μ , 进化代数 T_{low} , 算法终止代数 T_{max} , 聚类数目 ck .

输出. 问题最优解, 模型库.

步骤 1. 初始化. 随机初始化规模 N 的粒子群, 初始化每个粒子的局部引导者为其自身; 评价每个粒子的适应值, 设置所有粒子的全局引导者为种群最优粒子的位置; 初始化迭代次数为 $k = 1$.

步骤 2. 产生目标域样本. 运行 T_{low} 代 ABPSO, 得到粒子群 $P(T_{low})$, 利用 $P(T_{low})$ 中粒子产生目标域样本.

步骤 3. 利用第 2.1 节方法从模型库 ML 中寻找相匹配的历史问题. 如果发现相匹配的历史问题, 执行步骤 4; 否则, 执行步骤 5.

步骤 4. 执行第 2.2 节方法, 初始化 $P(T_{low})$ 中 $[ratio \times N]$ 个随机粒子的位置.

步骤 5. $k = T_{low}$, 循环执行 ABPSO 中相关粒子更新算子, 不断更新粒子群, 方法如下:

步骤 5.1. 评价粒子群 $P(k)$ 中所有粒子的适应值;

步骤 5.2. 判断 $T_{max}/[\mu \times T_{max}]$ 是否为整数. 如果是, 利用第 2.3 节方法更新代表个体保存集合; 否则, 执行步骤 5.3;

步骤 5.3. 采用常规方法更新粒子的局部引导者和全局引导者, 具体更新方法可参加文献 [29];

步骤 5.4. 利用式 (8) 更新粒子的位置;

步骤 5.5. 判断算法是否得到终止代数 T_{max} .

如果是, 终止算法; 否则, $k = k + 1$, 返回步骤 5.2.

步骤 6. 输出问题的最优解, 以及代表个体保存集合.

2.5 进一步分析

将迁移学习的思想融入到进化优化框架, 从已解决的相似历史问题中提取有价值的历史信息, 用来指导新问题的进化求解, 上述进化框架可以加速种群的进化过程, 提高算法的搜索效率. 具体地, 1) 在利用相似历史问题的迁移信息来初始化种群个体时, 本文根据历史问题与新问题的相似程度, 自适应确定迁移信息的使用程度, 即种群中被迁移信息初始化的个体比例, 可以有效防止出现负迁移或过迁移的现象, 避免算法性能的退化; 2) 在更新和丰富历史模型库时, 所给出的基于迭代聚类的代表样本保存策略, 在保证历史样本质量的同时, 还可以增强它们的多样性, 进而能够提高历史迁移信息的多样性, 防止对单一历史信息的过度学习; 3) 由于历史迁移信息仅用来初始化种群中的部分个体, 且初始化比例受新旧问题相似程度的严格限制, 因此, 在第 3.4 节中自适应变异因子 Δ , 以及粒子全局引导者更新策略的作用下, 本文所提 TL-ABPSO 算法仍然能够以概率 1 收敛. 具体收敛性证明, 可参考我们先前的工作 [26].

3 实验分析

3.1 实验环境及测试函数

本节将所提 TL-ABPSO 算法用于 10 个改进的标准测试函数, 与包括 ABPSO 在内的 4 种代表算法进行比较, 验证其性能. 所用对比算法包括: 标准的 PSO 算法 (Standard particle swarm optimization, SPSO)、基于突变和交叉的改进骨干 PSO 算法 (Bare-bone particle swarm optimization with mutation and crossover, BBPSO-MC)^[30]、基于跳跃策略的骨干 PSO 算法 (Bare-bone particle swarm optimization with jump, BBJ)^[31] 以及自适应骨干 PSO 算法 (ABPSO)^[26]. 所用实验采用相同环境, 即 Intel Core(TM)2 Duo、2.80 GHz 的 CPU、2.00 GB RAM 存储.

采用 MATLAB 2014b 实现所提算法, 并构建模型库. 表 1 展示了模型库中存储的 5 个经典历史测试函数. 首先, 针对表 1 给出的基本测试函数, 运行文献 [31] 所提 BBJ 算法, 并采用第 2.3 节方法保存迭代过程中适应值高且多样性好的代表解, 产生初始模型库信息. 表 2 给出了需要求解的 10 个新的问题. 这些新问题皆是表 1 中函数的变形, 在处理这些新优化问题时, 同样采用第 2.3 节方法将新问题的求解信息保存到模型库中. 表 2 中测试函数可分为

表 1 源域中保存的历史优化函数

Table 1 Historical optimization functions saved in the source domain

函数名称	表达式	定义域	最优解	维数
Sphere	$f(x) = \sum_{i=1}^n x_i^2$	$[-100, 100]$	0	30
Rastrigin	$f(x) = \sum_{i=1}^n (x_i^2 - 10 \cos(2\pi x_i) + 10)$	$[-5.12, 5.12]$	0	30
Ackley	$f(x) = -20 \exp(-0.2 \sqrt{\frac{1}{30} \sum_{i=1}^n x_i^2}) - \exp(\frac{1}{30} \sum_{i=1}^n \cos 2\pi x_i) + 20 + e$	$[-32, 32]$	0	30
Griewank	$f(x) = \frac{1}{4000} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos(\frac{x_i}{\sqrt{i}}) + 1$	$[-600, 600]$	0	30
Rosenbrock	$f(x) = \sum_{i=1}^n (100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2)$	$[-30, 30]$	0	30

表 2 目标域中新的优化函数

Table 2 New optimization functions in the target domain

函数名称	表达式	定义域	最优解	维数
F1: 修改后 Sphere	$f(x) = \sum_{i=1}^n x_i^{10}$	$[-100, 100]$	0	30
F2: 修改后 Rastrigin	$f(x) = \sum_{i=1}^n (x_i^{10} - 10 \sin(2\pi x_i))$	$[-5.12, 5.12]$	0	30
F3: 修改后 Ackley	$f(x) = -20 \exp(-0.2 \times \sqrt{10 \sum_{i=1}^n x_i^{10}}) - \exp(\frac{1}{30} \sum_{i=1}^n \sin 2\pi x_i) + 20 + 1$	$[-32, 32]$	0	30
F4: 修改后 Griewank	$f(x) = \frac{1}{4000} \sum_{i=1}^n x_i^{10} - \prod_{i=1}^n \sin(\frac{x_i}{\sqrt{i}})$	$[-600, 600]$	0	30
F5: 修改 Rosenbrock	$f(x) = \sum_{i=1}^n (100(x_{i+1} - x_i^{10})^2 + (x_i - 1)^{10})$	$[-30, 30]$	0	30
F6: Shifted Sphere	文献 [32]	$[-100, 100]$	-450	30
F7: Shifted rotated Rastrigin	文献 [32]	$[-5, 5]$	-330	30
F8: Shifted rotated Ackley	文献 [32]	$[-32, 32]$	-140	30
F9: Shifted rotated Griewank	文献 [32]	$[-600, 600]$	-180	300
F10: Shifted Rosenbrock	文献 [32]	$[-100, 100]$	390	30

两组, 其中, F1~F5 为第 1 组函数, 该组函数在不改变测试函数的特点基础上, 对函数的参数进行调整, 使其全局最优值发生变化. F6~F10 为第 2 组函数, 该组函数直接选自 CEC2005 测试集合^[32], 在整个搜索空间内, 其全局最优值被任意移动, 具有旋转、漂移和多模态等特点.

3.2 评价标准及参数设置

本文采用三种不同的评价指标, 分别从成功率、收敛速度和精度方面来评价算法的有效性.

1) 成功率 (SR). 即算法找到全局最优解的次数与实验总次数的比值. 当算法所得最优解与真实

最优解误差达到 ε 时, 即认为算法成功找到全局最优解, 其具体计算式为

$$SR = \frac{Sn'}{Sn} \times 100\% \quad (9)$$

其中, Sn' 为算法找到全局最优解的次数, Sn 为实验总次数, 本文取 30 次.

2) 收敛速度 (Time). 算法的收敛速度由两部分组成: a) 算法找到问题全局最优解时所需要的平均评价次数 F_{ave} ; b) 算法找到问题全局最优解时所需要的平均时间 (算法达到最大迭代次数时的平均运行时间) $Time$.

算法找到问题全局最优解时所需要的平均评价次数, 计算式为

$$F_{ave} = \frac{1}{S_n} \sum_{i=1}^{S_n} k_i \quad (10)$$

其中, S_n 为实验总次数, k_i 为第 i 次运行算法时找到全局最优解的代数.

3) 精度 (AC). 算法所得全局最优解与函数真实最优解之间的误差, 计算式为

$$AC = |f(X_{best}) - f(opt)| \quad (11)$$

其中, X_{best} 为算法找到的全局最优解, opt 为优化函数的真实最优解. 当两者差值的绝对值小于误差值 ε 时, 表示算法已经找到函数的真实最优值.

本文的参数设置如下: 实验中的所有算法, 其种群或粒子群的规模为 30, 算法的最大评价次数为 30 000 次. 其他对比算法的参数设置参照相应文献, 其中, SPSO 算法的惯性权重从 0.9 线性递减 0.4; BBJ 算法采用全局邻域, 其尺度参数设为 $\alpha = 0.75$; BBPSO-MC 算法中, 其邻域大小为 2; ABPSO 算法中, 突变概率为 0.7; 本文算法 TL-ABPSO 中, 聚类数目 ck 取为 5. 为便于比较算法的收敛速度, 对于函数 F8 和 F9, 设置成功误差 ε 分别为 10^0 和 10^{-3} , 其他函数皆设置误差 ε 为 10^{-8} .

3.3 实验结果分析

3.3.1 第 1 组测试函数的实验结果分析

1) 对于比较简单的单模态优化问题 F1, SPSO、ABPSO 和 TL-ABPSO 算法均能以 100% 的成功率找到其全局最优解, BBPSO-MC 和 BBJ 算法的误差精度也在很小的范围之内. 但是, 通过比较 F_{ave} 指标可以看出, 得益于所迁移相似历史问题信息的帮助, 本文算法 TL-ABPSO 的收敛速度更快, 其平均收敛代数为 1.574×10^3 , 其他四种比较算法的最小迭代次数为 2.635×10^3 .

2) F2~F4 函数是多模态问题, 随着决策变量维数的增加, 函数局部最优解的数量呈指数增加. 由表 3 可知, 对于函数 F2, 只有本文算法 TL-ABPSO 以 100% 的成功率收敛到全局最优点; 对于函数 F3 和 F4, 只有 ABPSO 和 TL-ABPSO 算法的成功率是 100%; 与其他四种比较算法的 F_{ave} 指标相比, TL-ABPSO 算法明显优于其他四种算法.

3) F5 函数是一个单峰函数, 决策变量之间具有很强的相关性, 且梯度信息经常误导算法的搜索方向. 如表 3 所示, 所有算法均没有成功找到全局最优解, 但是, 比较算法的精度误差指标 AC 可知, ABPSO 和 TL-ABPSO 算法明显好于其他三种算

法. 这其中得益于本文采用的历史问题迁移学习策略, TL-ABPSO 算法精度又好于 ABPSO 算法.

4) 在算法运行时间方面, 由于增加了模型匹配等测试, 在相同迭代次数下, 本文算法 TL-ABPSO 的运算时间要高于其他四种比较算法, 如函数 F5; 然而, 通过剩余 4 个函数的运行结果可以看出, 由于显著缩短了种群找到最优解的迭代次数, TL-ABPSO 找到最优解的时间花费并不比其他算法高; 对于函数 F3 和 F4, TL-ABPSO 甚至取得了最小的时间花费值. 好于其他三种算法, 其中得益于本文所采用历史迁移学习策略, TL-ABPSO 算法的精度又好于 ABPSO 算法.

3.3.2 第 2 组测试函数实验结果分析

针对表 2 中 F6~F10 等 5 个复杂测试函数, 分别运行每种比较算法 30 次, 表 4 给出了本文算法 TL-ABPSO、SPSO、ABPSO、BBPSO-MC 和 BBJ 等所得评价指标的平均结果. 可以看出:

1) 对于相对简单的函数 F6, SPSO、ABPSO、TL-ABPSO 算法都能以 100% 的成功率找到它们的全局最优解; 但是, 通过比较 F_{ave} 指标可以看出, 得益于所迁移相似历史问题信息的帮助, 本文算法 TL-ABPSO 的收敛速度更快, 其平均收敛代数为 1.421×10^2 , 其他四种比较算法的最小迭代次数为 1.065×10^2 .

2) 对于函数 F7, ABPSO 和 TL-ABPSO 的成功率分别为 72% 和 100%, 高于 BBPSO-MC 和 BBJ 的成功率 (33.33% 和 50%); 进一步, 得益于本文所采用的历史问题迁移学习策略, TL-ABPSO 算法的收敛速度明显好于 ABPSO 的收敛速度, TL-ABPSO 算法找到问题全局最优解时所需要的平均评价次数 (即 F_{ave} 指标) 为 7.467×10^3 .

3) 对于函数 F8 而言, 五种算法均没有 100% 地找到其全局最优解, 其中, TL-ABPSO 算法的成功率最高, 为 63.33%; 进一步, 比较 ABPSO 和 TL-ABPSO 算法的结果, 类似于函数 F7, TL-ABPSO 算法的收敛速度好于 ABPSO 的收敛速度, 且 TL-ABPSO 算法的误差指标 AC 高于 ABPSO 算法.

4) 对于具有多模、旋转和不可分离等特点的复杂函数 F9 和 F10, 五种算法均没有成功地找到其全局最优解; 五种算法的成功率都是 0; 但是, 比较其精度误差指标 AC 可知, ABPSO 和 TL-ABPSO 算法明显好于其他三种算法, 其中, 得益于本文所采用历史迁移学习策略, TL-ABPSO 算法的精度又好于 ABPSO 算法.

5) 对于复杂函数 F8~F10, 在相同迭代次数下, 尽管本文算法 TL-ABPSO 的运算时间要高于其他四种比较算法, 但是, 其结果的精度皆优于其他四种

表 3 比较算法优化第 1 组测试函数所得实验结果

Table 3 Experimental results obtained by comparison algorithm for the first set of test functions

优化函数	比较算法	指标 SR (%)	指标 F_{ave}	指标 $Time$ (s)	指标 AC
函数 F1	SPSO	100	2.635×10^3	1.539	0
	BBPSO-MC	66.67	—	2.960	1.8241×10^{-5}
	BBJ	63.33	—	7.614	8.6800×10^{-6}
	ABPSO	100	3.730×10^3	3.150	0
	TL-ABPSO	100	1.574×10^3	1.860	0
函数 F2	SPSO	0	—	9.722	5.169×10^1
	BBPSO-MC	26.67	—	12.761	1.824×10^{-0}
	BBJ	0	—	20.169	8.680×10^{-6}
	ABPSO	72	—	21.083	7.067×10^{-5}
	TL-ABPSO	100	1.283×10^4	20.183	0
函数 F3	SPSO	0	—	12.769	1.494×10^0
	BBPSO-MC	100	2.815×10^4	13.185	0
	BBJ	0	—	22.356	1.350×10^{-2}
	ABPSO	100	3.568×10^3	17.514	0
	TL-ABPSO	100	2.214×10^3	7.9190	0
函数 F4	SPSO	0	—	8.726	3.312×10^{-4}
	BBPSO-MC	100	1.163×10^3	3.820	0
	BBJ	0	—	9.947	2.366×10^{-5}
	ABPSO	100	8.110×10^2	4.706	0
	TL-ABPSO	100	2.310×10^2	3.370	0
函数 F5	SPSO	0	—	10.154	2.908×10^1
	BBPSO-MC	0	—	15.651	2.277×10^1
	BBJ	0	—	25.425	1.861×10^1
	ABPSO	0	—	31.817	1.281×10^1
	TL-ABPSO	0	—	49.476	1.253×10^1

比较算法; SPSO 算法的运算时间最短, 但其运行结果的精度较差. 对于函数 F6 和 F7, 由于显著缩短了种群找到最优解的迭代次数, 本文算法 TL-ABPSO 找到最优解的时间花费要小于其他四种比较算法.

3.3.3 ABPSO 和 TL-ABPSO 算法比较

本小节单独比较 ABPSO 和 TL-ABPSO 的收敛效果. 如前所述, 本文所提 TL-ABPSO 是 ABPSO 算法和历史问题迁移学习策略的结合体. 通过比较两种算法的收敛效果, 可以反映出所提历史问题迁移学习策略的有效性. 图 2 展示了优化第 1 组测试函数时两种算法的收敛曲线. 可以看出, TL-ABPSO 算法的进化趋势和 ABPSO 算法大致相同, 加入迁移策略后并没有影响算法本身的搜索机理; 但是, 得益于所迁移历史问题最优解信息的帮助, TL-ABPSO 算法的收敛速度始终明显好于 ABPSO 算法.

3.4 实验拓展及补充

3.4.1 迁移学习策略在差分进化算法中的应用

上述工作成功地将迁移学习策略加入到了骨干粒子群优化算法中, 实验结果也说明了迁移学习策略的有效性. 本小节尝试将所提迁移学习策略加入到另一种典型的优化算法, 即自适应差分进化算法^[33]中, 以验证迁移学习策略在不同进化算法中的效果. 由此, 给出一种基于相似历史信息迁移学习的自适应差分进化算法, 简称 TL-DE 算法.

为了验证 TL-DE 算法的有效性, 选择在第 1 组测试函数上进行实验, 算法参数设置为种群规模 50, 使用“DE/rand/1”变异策略, 缩放因子为 0.5, 交叉概率为 0.9, 最大进化代数数为 3000. 图 3 展示了优化第 1 组测试函数时两种算法的收敛曲线. 可以看出, 在进化初期, TL-DE 算法的收敛速度明显好于 DE 算法; 在进化后期, 对于大部分函数, TL-DE 算法所得结果也明显好于 DE 算法. 这进一步说明, 加入迁

表 4 比较算法优化第 2 组测试函数所得实验结果

Table 4 Experimental results obtained by comparison algorithm for the second set of test functions

优化函数	比较算法	指标 SR (%)	指标 F_{ave}	指标 $Time$ (s)	指标 AC
函数 F6	SPSO	100	3.525×10^3	10.105	0
	BBPSO-MC	66.67	—	12.132	0
	BBJ	0	—	14.835	0
	ABPSO	100	1.066×10^3	7.451	0
	TL-ABPSO	100	1.421×10^2	7.160	0
函数 F7	SPSO	0	—	36.853	4.279×10^1
	BBPSO-MC	33.33	—	62.168	3.800×10^{-3}
	BBJ	50	—	57.679	1.893×10^0
	ABPSO	72	—	39.707	0
	TL-ABPSO	100	7.467×10^3	22.704	0
函数 F8	SPSO	0	—	33.796	1.329×10^2
	BBPSO-MC	26.67	—	61.905	2.095×10^1
	BBJ	0	—	62.813	2.094×10^1
	ABPSO	44.33	—	166.905	8.231×10^{-2}
	TL-ABPSO	63.33	—	214.023	1.709×10^{-4}
函数 F9	SPSO	0	—	39.021	5.483×10^2
	BBPSO-MC	0	—	62.722	7.901×10^{-3}
	BBJ	0	—	54.329	1.040×10^1
	ABPSO	0	—	125.621	5.286×10^1
	TL-ABPSO	0	—	183.422	9.567×10^{-4}
函数 F10	SPSO	0	—	36.927	2.908×10^1
	BBPSO-MC	0	—	61.933	2.674×10^1
	BBJ	0	—	51.946	3.597×10^1
	ABPSO	0	—	122.359	1.922×10^1
	TL-ABPSO	0	—	180.816	1.347×10^1

移学习的进化算法不仅可以节省进化成本, 加速种群的进化过程, 而且可以提高算法的求解精度. 因此, 本文所提于历史信息迁移的进化优化框架同样适用于差分进化算法.

3.4.2 聚类数目 ck 的敏感性分析

由于影响历史问题与新问题的匹配精度, 聚类数目 ck 的取值非常关键. 聚类数目 ck 设置过小, 聚类结果不能有效反映解分布的多样性; 而聚类数目过大, 每一类中保存的样本较少, 利用少量样本学习得到的解分布模型通常不精确. 基于优化问题的复杂程度和解分布特点, 可以设置合理的 ck 值, 但是准确获取这些信息往往需要决策者具有一定的问题先验知识, 这在很多实际情况下是不可能或者需要花费太多代价的. 本节通过实验分析聚类数目 ck 值对算法性能的影响, 旨在给决策者提供一个较为合理的 ck 取值范围.

鉴于此, 选择第 1 组测试函数为例, 参数 ck 分别取值为 3, 5, 7, 10, 运行 TL-ABPSO 算法 30 次,

图 4 给出了在不同参数 ck 取值下, TL-ABPSO 找到问题最优解所需平均迭代次数. 可以看出, 聚类数目 ck 取 3 或者 10 时, TL-ABPSO 算法可以找到问题的全局最优解, 但其所需迭代次数要大于 ck 取 5 和 7 的情况; 当聚类数目 ck 取 5 或者 7 时, TL-ABPSO 算法能够以较快的速度找到问题的全局最优解. 这在一定程度上归功于由聚类数目 ck 间接决定的种群初始化比例 $ratio$, 合适的种群初始化比例明显提高了算法的搜索速度.

4 结论

本文研究了一种基于相似历史信息迁移学习的进化优化框架. 针对框架中历史模型匹配、历史信息的迁移学习和模型库更新等关键算子, 分别提出了基于多分布估计最大均值差异的历史模型匹配策略、基于模型匹配程度的进化种群初始化策略, 以及于基于迭代聚类的模型库更新策略. 通过从模型库中找到与新问题匹配的历史问题, 并将其知识迁移

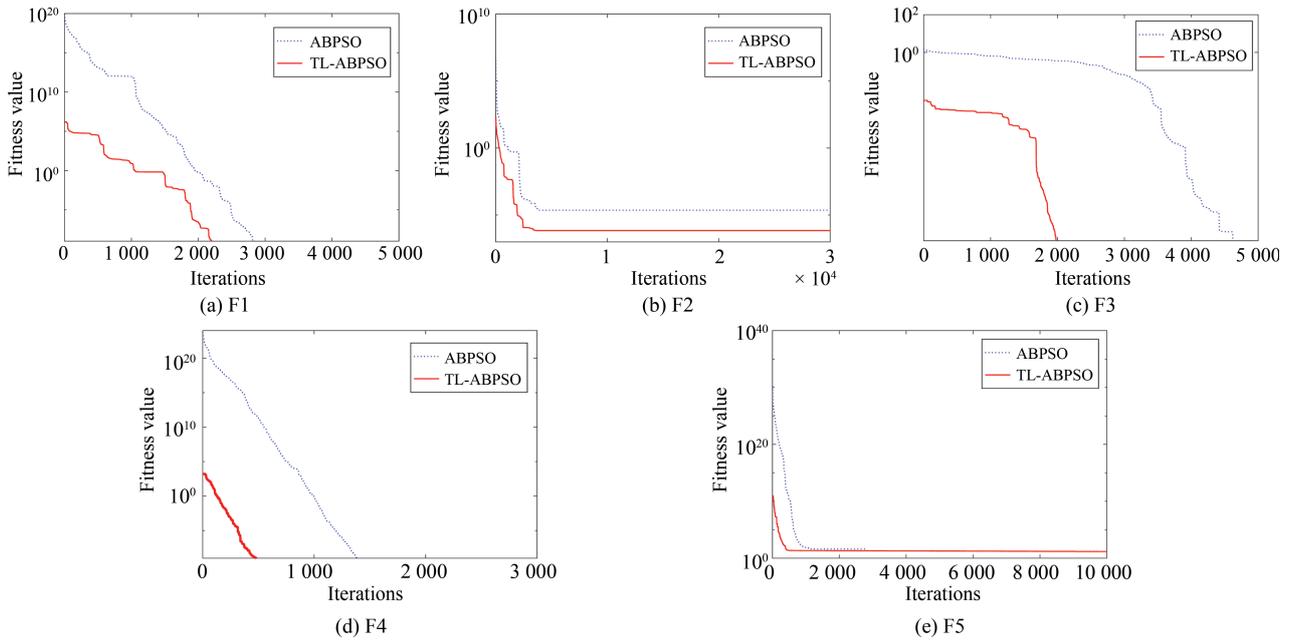


图 2 优化第 1 组测试函数时 ABPSO 和 TL-ABPSO 算法的收敛曲线

Fig. 2 Convergence curves of ABPSO and TL-ABPSO algorithms for the first set of test functions

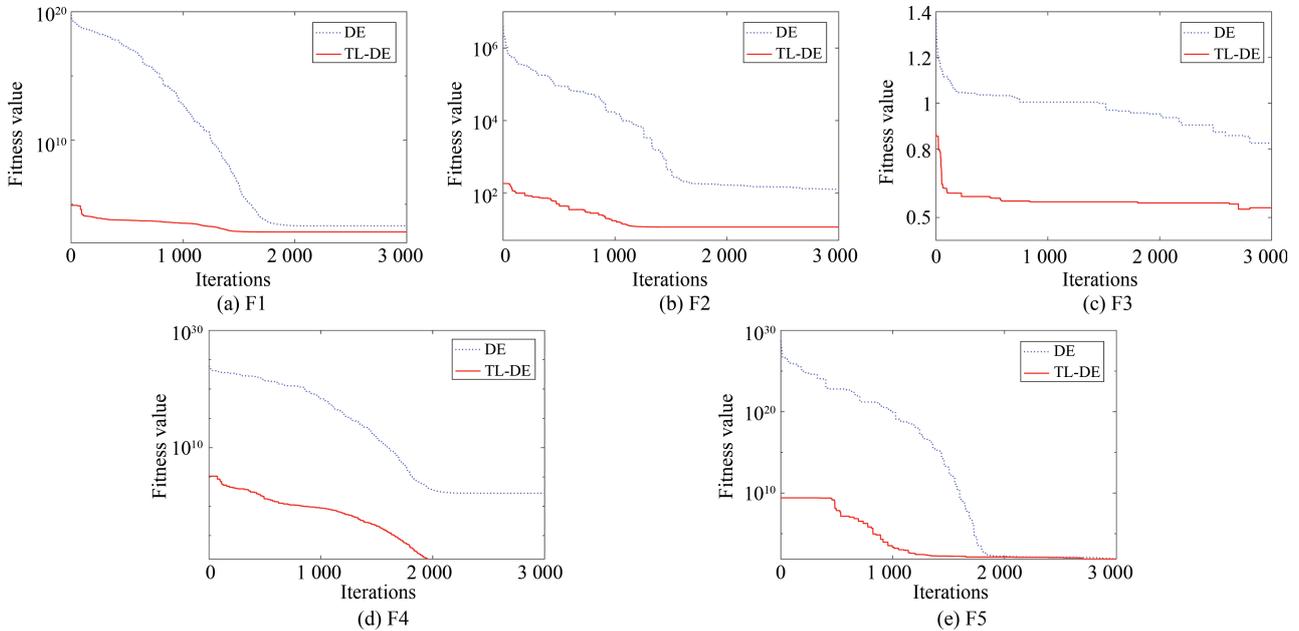


图 3 优化第 1 组测试函数时 DE 和 TL-DE 算法的收敛曲线

Fig. 3 Convergence curves of DE and TL-DE algorithms for the first set of test functions

到新问题的进化优化过程中, 该框架明显提高了算法的搜索效率. 将已有的粒子群优化算法和差分进化算法分别嵌入到所提进化优化框架, 给出了基于相似历史信息迁移学习的骨干粒子群优化算法和基于相似历史信息迁移学习的自适应差分进化算法. 两种算法在 10 个典型测试问题上的应用表明, 本文所提基于相似历史问题的迁移学习机制, 不仅可以明显加速种群进化过程, 而且能够提高算法的求解

质量.

然而, 不可否认, 本文所提策略需要历史和新问题同属一种问题, 而且它们应该具有相似的编解码策略和特性, 这限制了所提进化框架的应用范围. 是否可以在两种不同问题 (如背包问题和设施布局问题等) 之间进行求解信息或求解规则的迁移, 将是我们今后研究的重点; 此外, 如何降低模型匹配等策略带来的计算复杂度问题, 也是今后需要研究的内容.

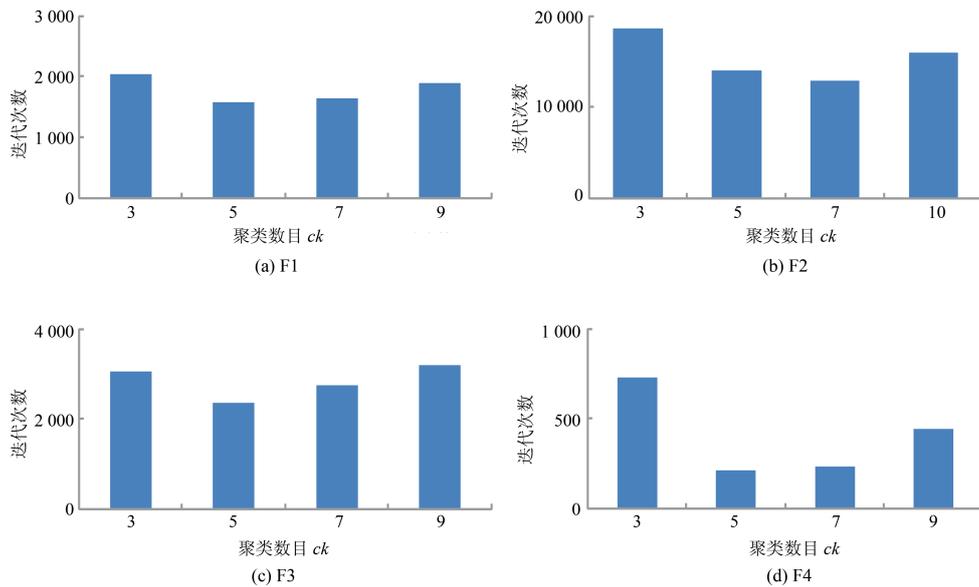


图 4 在不同参数 ck 取值下, TL-ABPSO 找到问题最优解所需平均迭代次数

Fig. 4 The average iteration time required by TL-ABPSO to find the optimal solution under different ck value

References

- Li Yuan-Cheng, Li Zong-Pu, Yang Li-Qun, Wang Bei. Reactive power optimization of distribution network with distributed power supply based on improved quantum difference evolution. *Acta Automatica Sinica*, 2017, **43**(7): 1280–1288
(李元诚, 李宗圃, 杨立群, 王蓓. 基于改进量子差分进化的含分布式电源的配电网无功优化. *自动化学报*, 2017, **43**(7): 1280–1288)
- Yu Wei-Wei, Xie Cheng-Wang, Bi Ying-Zhou, Xia Xue-Wen, Li Xiong, Ren Ke-Yan, Zhao Huai-Rui, Wang Shao-Feng. Many-objective particle swarm optimization based on adaptive fuzzy dominance. *Acta Automatica Sinica*, 2018, **44**(12): 2278–2289
(余伟伟, 谢承旺, 闭应洲, 夏学文, 李雄, 任柯燕, 赵怀瑞, 王少峰. 一种基于自适应模糊支配的高维多目标粒子群算. *自动化学报*, 2018, **44**(12): 2278–2289)
- Wang Dong-Feng, Meng Li. Performance analysis and parameter selection of PSO algorithms. *Acta Automatica Sinica*, 2016, **42**(10): 1552–1561
(王东风, 孟丽. 粒子群优化算法的性能分析和参数选择. *自动化学报*, 2016, **42**(10): 1552–1561)
- Tang L, Wang X. A hybrid multiobjective evolutionary algorithm for multiobjective optimization problems. *IEEE Transactions on Evolutionary Computation*, 2013, **17**(1): 20–45
- Tian Meng-Chu, Bo Yu-Ming, Chen Zhi-Min, Wu Pan-Long, Zhao Gao-Peng. Firefly algorithm intelligence optimized particle filter. *Acta Automatica Sinica*, 2016, **42**(1): 89–97
(田梦楚, 薄煜明, 陈志敏, 吴盘龙, 赵高鹏. 萤火虫算法智能优化粒子滤波. *自动化学报*, 2016, **42**(1): 89–97)
- Xu Mao-Xin, Zhang Xiao-Shun, Yu Tao. Transfer bees optimizer and its application on reactive power optimization. *Acta Automatica Sinica*, 2017, **43**(1): 83–93
(徐茂鑫, 张孝顺, 余涛. 迁移蜂群优化算法及其在无功优化中的应用. *自动化学报*, 2017, **43**(1): 83–93)
- Zhang X Y, Tian Y, Cheng R, Jin Y C. An efficient approach to nondominated sorting for evolutionary multiobjective optimization. *IEEE Transactions on Evolutionary Computation*, 2015, **19**(2): 201–213
- Patvardhan C, Bansal S, Srivastav A. Quantum-inspired evolutionary algorithm for difficult knapsack problems. *Memetic Computing*, 2015, **7**(2): 135–155
- Pan S J, Yang Q. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 2010, **22**(10): 1345–1359
- Murre J M J. Transfer of learning in backpropagation and in related neural network models. *Connectionist Models of Memory and Language*, London: UCL Press, 1995. 73–93
- Pratt L Y. Discriminability-based transfer between neural networks. *Advances in Neural Information Processing Systems*, 1992, **5**: 204–211
- Mihalkova L, Huynh T, Mooney R J. Mapping and revising Markov logic networks for transfer learning. In: Proceedings of the 22nd Conference on Artificial Intelligence (AAAI-07). Vancouver, Canada: AAAI, 2007. 608–614
- Mihalkova L, Mooney R J. Transfer learning with Markov logic networks. In: Proceedings of the ICML-06 Workshop on Structural Knowledge Transfer for Machine Learning, Pittsburgh, PA, USA: 2006.
- Gupta R, Ratnoff L. Text categorization with knowledge transfer from heterogeneous data sources. In: Proceedings of the 23rd AAAI Conference on Artificial Intelligence, Chicago, Illinois, USA: AAAI, 2008. 842–847
- Ling X, Xue G R, Dai W Y, Jiang Y, Yang Q, Yu Y. Can Chinese web pages be classified with English data source? In: Proceedings of the 17th International Conference on World Wide Web, Beijing, China: WWW, 2008. 969–978
- Dinh T T H, Chu T H, Nguyen Q U. Transfer learning in genetic programming. In: Proceedings of the 2015 IEEE Congress on Evolutionary Computation (CEC), Sendai, Japan: IEEE, 2015. 1145–1151

- 17 Koçer B, Arslan A. Genetic transfer learning. *Expert Systems with Applications*, 2010, **37**(10): 6997–7002
- 18 Feng L, Ong Y S, Tan A H, Tsang Z W. Memes as building blocks: A case study on evolutionary optimization + transfer learning for routing problems. *Memetic Computing*, 2015, **7**(3): 159–180
- 19 Jiang M, Huang Z Q, Qiu L M, Huang W Z, Yen G G. Transfer learning based dynamic multiobjective optimization algorithms. *IEEE Transactions on Evolutionary Computation*, 2018, **22**(4): 501–514
- 20 Rosenstein M T, Marx Z, Pack Kaelbling L, Dietterich T G. To transfer or not to transfer. In: Proceedings of the NIPS 2005 Workshop on Transfer Learning, 2005.
- 21 Shi Y H, Eberhart R C. A modified particle swarm optimizer. In: Proceedings of the 1998 IEEE International Conference on Evolutionary Computation, Anchorage, AK, USA: IEEE, 1998. 69–73
- 22 Han Hong-Gui, Lu Wei, Qiao Jun-Fei. A multi-objective particle swarm optimization algorithm based on diversity information and convergence. *Chinese Journal of Electronics*, 2018, **46**(2): 315–324
(韩红桂, 卢薇, 乔俊飞. 一种基于多样性信息和收敛度的多目标粒子群优化算. 电子学报, 2018, **46**(2): 315–324)
- 23 Zhang Qian, Li Ming, Wang Xue-Song, Cheng Yu-Hu, Zhu Mei-Qiang. An example of migration learning for multi-sources. *Acta Automatica Sinica*, 2014, **40**(6): 1176–1183
(张倩, 李明, 王雪松, 程玉虎, 朱美强. 一种面向多源领域的实例迁移学习. 自动化学报, 2014, **40**(6): 1176–1183)
- 24 Wang Jun, Li Shi-Jun, Yang Sha, Jin Hong, Yu Wei. A new migration learning model for cross-domain recommendation. *Chinese Journal of Computers*, 2017, **40**(10): 2367–2380
(王俊, 李石君, 杨莎, 金红, 余伟. 一种新的用于跨领域推荐的迁移学习模型. 计算机学报, 2017, **40**(10): 2367–2380)
- 25 Jiang Hai-Yan, Liu Hao-Tian, Shu Xin, Xu Yan, Wu Yan-Lian, Guo Xiao-Qing. Multi-label migration learning algorithm based on large mean difference. *Information and Control*, 2016, **45**(4): 463–478
(姜海燕, 刘昊天, 舒欣, 徐彦, 伍艳莲, 郭小清. 基于最大均值差异的多标记迁移学习算法. 信息与控制, 2016, **45**(4): 463–478)
- 26 Zhang Y, Gong D W, Sun X Y, Geng N. Adaptive bare-bones particle swarm optimization algorithm and its convergence analysis. *Soft Computing*, 2014, **18**(7): 1337–1352
- 27 Clerc M, Kennedy J. The particle swarm-explosion, stability, and convergence in a multidimensional complex space. *IEEE Transactions on Evolutionary Computation*, 2002, **6**(1): 58–73
- 28 Omran M G H, Engelbrecht A P, Salman A. Bare bones differential evolution. *European Journal of Operational Research*, 2009, **196**(1): 128–139
- 29 Shi Y H, Eberhart R C. Empirical study of particle swarm optimization. In: Proceedings of the 2002 IEEE Conference on Evolutionary Computation, Washington DC, USA: IEEE, 2002. 320–324
- 30 Zhang H, Kennedy D D, Rangaiah G P, Bonilla-Petriciolet A. Novel bare-bones particle swarm optimization and its performance for modeling vapor — liquid equilibrium data. *Fluid Phase Equilibria*, 2011, **301**(1): 33–45
- 31 Blackwell T. A study of collapse in bare bones particle swarm optimization. *IEEE Transactions on Evolutionary Computation*, 2012, **16**(3): 354–372
- 32 Suganthan P N, Hansen N, Liang J J, Deb K, Chen Y P, Auger A, Tiwari S. Problem Definitions and Evaluation Criteria for the CEC 2005 Special Session on Real-parameter Optimization. Technical Report for CEC 2005 Special Session, 2005. [Online], available: <http://www3.ntu.edu.sg/home/EPNSugan>, April 1, 2018
- 33 Qin A K, Huang V L, Suganthan P N. Differential evolution algorithm with strategy adaptation for global numerical optimization. *IEEE Transactions on Evolutionary Computation*, 2009, **13**(2): 398–417



张勇 中国矿业大学信息与控制工程学院教授. 2009年获控制理论与控制工程专业博士学位. 主要研究方向为群体智能和机器学习.

E-mail: yongzh401@126.com

(**ZHANG Yong** Professor at the School of Information and Control Engineering, China University of Mining and Technology. He received his Ph.D. degree of control theory and control engineering from China University of Mining and Technology in 2009. His research interest covers swarm intelligence and machine learning.)



杨康 中国矿业大学硕士研究生. 主要研究方向为群体智能和机器学习.

E-mail: zgkydxyk@sina.com

(**YANG Kang** Master student of information and control engineering at China University of Mining and Technology. His research interest covers swarm intelligence and machine learning.)

ing.)



郝国生 江苏师范大学计算机学院教授. 2009年获得控制理论与控制工程专业博士学位. 主要研究方向为进化计算和大数据应用. 本文通信作者.

E-mail: hgskd@jsnu.edu.cn

(**HAO Guo-Sheng** Professor at the School of Computer Science and Technology, Jiangsu Normal University. He received his Ph.D. degree of control theory and control engineering from China University of Mining and Technology in 2009. His research interest covers evolutionary algorithms and the application of big data. Corresponding author of this paper.)



巩敦卫 中国矿业大学信息与控制工程学院教授. 1999年在中国矿业大学获博士学位. 主要研究方向为进化计算与应用. E-mail: dwgong@vip.163.com

(**GONG Dun-Wei** Professor at the School of Information and Control Engineering, China University of Mining and Technology. He received his Ph.D.

degree from China University of Mining and Technology in 1999. His research interest covers evolutionary computation and its applications.)