

基于机器学习的演化多任务优化框架

麦伟杰¹⁾ 刘伟莉²⁾ 钟竞辉¹⁾

¹⁾(华南理工大学计算机科学与工程学院 广州 510006)

²⁾(广东技术师范大学计算机科学学院 广州 510665)

摘 要 演化多任务优化是近年来计算智能领域的研究热点之一,其原理是通过任务间的知识转移提高演化算法同时求解多个任务的效率.由于任务间相似性对促进任务之间的正向知识转移具有重要的影响,因此,如何度量任务间的相似性成为了重点研究方向之一.目前,演化多任务优化在处理两个任务时,辅助任务的选取仅限于两者之一,且在处理超多任务时对任务间知识的转移缺乏灵活性.为此,本文提出一个基于机器学习的演化多任务优化框架,命名为 MaTML.该框架联合所有任务关联的子种群形成一个统一的初始化种群,利用目标任务的技能因子及其对应的种群个体分别构建标签和训练集,应用十折交叉法拟合模型,并运用模型预测与目标任务相似的个体以组成辅助种群,从而促进演化优化中的正向知识转移.本文提出的算法能够在动态的种群个体中找到目标任务的辅助种群,不仅可以为三个或以上的多任务优化灵活地选取相似辅助任务,而且解决了当任务数量为两个时有效地选择辅助任务的问题.通过与现阶段的多任务算法和超多任务算法分别在 CEC2017 问题测试集和 WCCI2020SO 问题测试集进行比较,实验结果证实 MaTML 在优化多任务问题时具有更优或竞争性的性能.此外,文中还详细研究了 MaTML 的计算资源、模型性能、模型稳定性以及相关组件.最后,本文还基于真实问题的测试进一步验证了 MaTML 的有效性.

关键词 演化多任务优化;机器学习;任务间相似性;知识转移;辅助任务

中图法分类号 TP18

DOI号 10.11897/SP.J.1016.2024.00029

Evolutionary Many-Task Optimization Framework Based on Machine Learning

MAI Wei-Jie¹⁾ LIU Wei-Li²⁾ ZHONG Jing-Hui¹⁾

¹⁾(School of Computer Science and Engineering, South China University of Technology, Guangzhou 510006)

²⁾(School of Computer Science, Guangdong Polytechnic Normal University, Guangzhou 510665)

Abstract Evolutionary multitasking optimization is one of the research hotspots in the field of computational intelligence in recent years. Its principle is to enhance the efficiency of evolutionary algorithms to solve multiple tasks simultaneously through knowledge transfer between tasks. Since inter-task similarity plays an important role in promoting the positive knowledge transfer between tasks, how to measure the similarity between tasks has become one of the key research directions. At present, the existing evolutionary multitasking algorithms can be divided into evolutionary multi-task algorithms and evolutionary many-task algorithms according to the number of tasks processed. However, these algorithms are not efficient enough in handling both multi task and multi task problems simultaneously. For example, when evolutionary multitasking optimization tackles two tasks, the selection of auxiliary task is limited to one of them, and can lead to the lack of flexibility of knowledge transfer between tasks in the aspect of dealing with many-task. In addition, the dynamic nature of task knowledge transfer should occur across tasks,

rather than limited to a certain task or several tasks, which requires the algorithm to have the ability to adaptively find other individuals similar to the target task individuals in iteration. Based on machine learning, this paper proposes a framework for solving evolutionary multitasking optimization, named as MaTML, which combines all task-associated subpopulations to form a unified initial populations. MaTML employs the skill factors of the target task and its corresponding population individuals to construct labels and training sets respectively, utilizes the 10-fold cross-validation to fit the model, and then applies the model to predict that population individuals similar to the target task compose auxiliary populations, so as to achieve more positive knowledge transfer in evolutionary optimization. Specifically, in each iteration, the skill factor of the target task is taken as the training label, and all the individuals corresponding to the label, that is, the target task individuals, are combined to form the training sample. Then the model is trained based on the machine learning algorithm, and the model is applied to predict the individuals in the population except the target task individuals. Since the characteristics of the individuals correctly predicted by the algorithm are essentially similar to those of the target task individuals, these individuals can be labeled as auxiliary individuals of the target task. The proposed algorithm can find the auxiliary populations of the target task in the dynamic population individuals, so it can not only flexibly select similar auxiliary tasks for three or more tasks, but also solve the problem of effectively selecting auxiliary tasks at that time the number of tasks is two. What's more, the selection of population individuals for auxiliary tasks of the target task is variable and dynamically adaptive in MaTML, and there is no need to associate a subpopulation with the target task. Compared with state-of-the-art multi-task algorithms and many-task algorithms in CEC2017 test suite and WCCI2020SO test suite respectively, the experimental results show that MaTML has superior or competitive performance in optimizing multi-task or many-task problems. We also conducted a detailed study on MaTML's computing resources, model performance, model stability, and related components. Finally, an optimization problem of real-world was served as test to further verify the effectiveness of MaTML.

Keywords evolutionary multitasking optimization; machine learning; inter-task similarity; knowledge transfer; auxiliary task

1 引 言

演化算法 (Evolutionary Algorithms, EAs) 是一种基于自然选择与遗传学原理的随机搜索优化算法^[1], 由于其搜索机制具有隐式并行性, 且能够自动并行地在多个区域进行优化, 因此可以通过一次性运行同时获得目标问题的多个解^[2-3]. 与传统优化方法相比, 演化算法具有出色的优化性能, 被广泛应用在多目标优化问题^[4-5]、组合优化问题^[6-7]、无人机搜索问题^[8]及限制性优化问题^[9-10]等科学理论研究和工程实践应用问题.

尽管 EAs 具有强大的隐式并行计算潜力, 但现有的 EAs 主要专注于一次性优化同一个问题, 导致

隐式并行的优点没有被充分利用. 另一方面, 实际应用中的优化问题通常与其他问题或历史解决过的问题有关联, 而现有的 EAs 都是基于零先验知识从头开始解决一个问题, 未能充分利用问题间的相关性来提升求解效率. 此外, 近年来, 随着云计算和物联网等技术的快速发展, 系统在某个时刻可能会收到来自不同用户的多个请求需要同时处理, 传统的 EAs 不能有效地应对这种挑战.

受人类大脑能够同时处理多个任务的情景启发, Gupta 等人^[11]提出多因子优化算法 (Multifactorial Evolutionary Algorithm, MFEA) 来同时优化多个任务, 该算法目前已成为演化多任务 (Evolutionary Multitask, EMT) 优化算法的范式. 相比较传统的 EAs, MFEA 的特点在于能够通过一次性运行同时

得到多个任务的所有解,并通过利用多个任务间的共享知识加快各个任务收敛速度^[12-13].自从 MFEA 被提出以后,演化多任务优化成为演化计算方向一个新的研究分支,在计算智能领域引起了众多学者的关注,且越来越多的 MFEAs 算法被相继提出并赋之于实践^[14-15].

然而,现有的 MFEAs 仍然存在一些不足之处.一方面,它们有些仅适用于处理少量任务(例如两个)^[11,16-18],当任务数量增加时就会在优化性能上大打折扣,这是因为这类 MFEAs 中的知识转移策略具有盲目性,不能自适应地在复杂的多任务环境中自我反馈,当正向与负向的知识转移混合在一起时,负向知识转移会阻碍任务的收敛速度.为了解决任务数量较大(三个以上)的多任务优化问题,一些 MFEAs 变种也被提出^[19-20],但这类 MFEAs 不擅长处理两个任务问题,这是因为它们在设计时采用了每个任务都有一个子种群与之对应的策略,且每个任务的子种群大小固定,因此在每次迭代时子种群的更新及任务之间的知识共享都被限制在预先设置的子种群内进行,导致任务间知识转移失去了灵活性.例如,假设所有任务的子种群大小均为 $|P|$,那么,当目标任务 Target 利用辅助任务 Assist 进行优化的过程中,每次迭代它们之间的知识迁移就只能在大小为 $|P|$ 的空间交流,忽略了剩余种群中对任务优化有用的知识,即知识迁移失去灵活性和动态性.深入研究发现,由于设定了固定的子种群,也会导致目标任务对应的个体丢失多样性,因为擅长于目标任务的个体不仅存在于设定的子种群中,而且可能存在其它子种群里面.这种策略在进化中通过寻找与目标任务相似的辅助任务来协助优化,必须要求在任务数量较多时候才奏效,当任务数量变成两个时就会面临无法选取与目标任务相似的辅助任务进行优化的困境(一个是目标任务,另一个则必然成为辅助任务).

为了解决上述问题,本文提出一个基于机器学习(Machine Learning, ML)的演化多任务优化框架(MaTML),该框架不仅可以用于两个任务(Multi-task)的优化,而且同样适用于超多任务(Many-task)的优化. MaTML 的主要思想是在演化多任务环境中利用机器学习强大的预测能力^[21-22]动态找到与目标任务相似的个体,并把相似的个体组成一个临时的子种群,构造出目标任务的辅助种群.具体来讲,在每次迭代中以目标任务的技能因子作为训练

标签,将标签对应的所有个体一起组成训练样本(即目标任务个体),之后基于机器学习算法训练模型,应用模型对种群中除了目标任务个体之外的个体进行预测.由于被算法正确预测的个体在特征本质上与目标任务个体相近或相似,因此,可把这些个体标记为目标任务的辅助个体.在进化过程中,当算法执行目标任务利用辅助种群进行知识转移这一步骤时,由于目标任务与辅助种群之间的相似性较大,正向知识转移发生的概率相对就会增大,从而加速算法收敛速度以及提高解的质量.在 MaTML 中,每次迭代组成目标任务的个体及其辅助种群均具有动态性,这种动态性体现在每次迭代中目标任务的技能因子数量均不同,即每代中组成目标任务子种群大小会变化,但不论如何变化, MaTML 总能找出与目标任务相似的个体,这些个体即为构成辅助子种群,故辅助种群的个体也具有动态性.换句话说, MaTML 在每次迭代过程能够自适应地调节日标任务子种群与辅助子种群的大小.正是这种动态性使得 MaTML 可以解决当任务数量为两个时无法选择辅助任务的问题,并为任务间的知识转移提高了灵活性及种群个体的多样性.本文的贡献主要有如下三方面:

(1) 提出一个基于机器学习的新型演化多任务优化框架.该框架主要特点可以动态训练目标任务模型和预测与目标任务相关的辅助任务个体,在优化过程直接促进目标任务与辅助任务的正向知识交流.

(2) 框架只需要一个数量较大的个体组成统一种群,目标任务的辅助任务的种群个体选取是可变、动态自适应的,无需为目标任务关联一个子种群,解决由于在设计上采用了每个任务关联一个大小相同的子种群而限制了任务间知识转移的灵活性这一缺陷,同时解决当任务变为两个时在寻找目标任务的辅助任务面临选取的困境,统一了在处理两个任务与三个以上任务时的操作.

(3) 应用最近提出的 9 个问题 2 个任务 CEC17^[23]测试集、10 个问题 50 个任务 WCCI2020SO^[24]测试集和一个真实的工程优化问题对本文提出的 MaTML 进行性能测试,证实所提出算法的有效性.

本文第 2 节介绍演化多任务的相关知识及有关文献回顾;第 3 节详细描述本文提出的 MaTML 架构;第 4 节通过实验对 MaTML 进行详细测试,并对实验结果深入分析;最后,第 5 节进行总结并指出下一步需要研究的地方.

2 相关工作

2.1 多任务优化问题

EMT 优化问题旨在利用一个单一的种群同时获得多个任务的最优解,其充分地利用种群搜索的隐式并行性发掘不同任务间潜在的互补特性,使每一任务都在多任务环境下尽可能得到高效优化^[25].不失一般性,考虑 K 个在优先权上均相等的最小值优化问题,假设 T_j 为第 j 个任务,它所对应的搜索空间的一个候选解为 \mathbf{X}_j ,目标问题为 f_j .那么,拥有 K 个任务的 EMT 优化问题在数学上可以表示为

$$\begin{aligned} (\mathbf{X}_1^*, \mathbf{X}_2^*, \dots, \mathbf{X}_K^*) = \arg \min (f_1(\mathbf{X}_1), f_2(\mathbf{X}_2), \dots, f_K(\mathbf{X}_K)) \\ \text{subject to: } \mathbf{X}_j \in R^D, j=1, 2, \dots, K \end{aligned} \quad (1)$$

其中, \mathbf{X}_j^* 表示第 j 个任务的最优解, $j=1, 2, \dots, K$, K 代表任务数量, $\mathbf{X}_j = (x_{j,1}, x_{j,2}, \dots, x_{j,D})$ 代表第 j 个任务对应的搜索空间的一个候选解, D 为候选解的维度.通常, $K=2$ 时称之为 multi-task, $K \geq 3$ 时称之为 many-task.

EMT 优化问题与演化多目标(Evolutionary Multi-Objective, EMO)优化问题在表达形式上相近,但在本质上存在很大的差别:在多任务优化中,各个任务在不同搜索空间进行进化,多目标优化要求所有的优化问题均在统一的搜索空间中进行.另一个显著不同之处是,在多任务优化中,各个任务之间一般不具约束关系,但多目标优化中的各个问题往往存在彼此间约束.此外,多任务优化目标是同时找到所有任务的最优解,多目标优化则是寻找在所有优化目标上具有不同权衡的帕累托解集.

2.2 MFEA

MFEA 首次由 Gupta 等人^[11]基于生物遗传学的基因和文化特征提出,旨在利用演化算法一次运行同时求解多个优化任务.在 MFEA 中,所有种群个体在统一搜索空间 $\Omega \in [0, 1]^D$ 中进行基因型(genotype)编码.在优化某个任务时,个体都可以被解码成与任务相对应的表型(phenotype)编码表示.这种编码方式可以使得算法在求解不同的任务过程中共享遗传物质. MFEA 中,种群中的每个个体拥有如下特征:

(1) 因子代价(factorial cost). 种群中个体 \mathbf{X}_i 在任务 T_j 上的因子代价通过 $\Psi_j^i = f_j^i + \lambda \cdot \delta_j^i$ 表示.其中, f_j^i 和 δ_j^i 分别代表个体 \mathbf{X}_i 在任务 T_j 中的目标函数值与约束违反值,特别地,当 \mathbf{X}_i 为可行解时, $\Psi_j^i = f_j^i$.

(2) 因子等级(factorial rank). 个体 \mathbf{X}_i 在任务 T_j 上的因子等级 r_j^i 为该个体在种群中按因子代价 Ψ_j^i 从小到大排序后的索引.

(3) 标量适应度(scalar fitness). 个体 \mathbf{X}_i 的标量适应度 φ_i 通过该个体在所有任务下的因子排名构成的列表 $\{r_1^i, r_2^i, \dots, r_K^i\}$ 简化的标量值表示,即 $\varphi_i = 1/\min\{r_j^i\}, j=1, 2, \dots, K$.

(4) 技能因子(skill factor). 个体 \mathbf{X}_i 的技能因子 τ_i 为该个体在所有任务中表现最好的任务的索引,即 $\tau_i = \arg \min\{r_j^i\}, j=1, 2, \dots, K$. 技能因子用于将种群划分为具有不同文化特征的子群体.

MFEA 基本结构的伪代码如算法 1 所示. 算法的流程从随机初始化 N 个个体开始,并将它们表示在统一搜索空间中. 该空间维度为所有任务中维度的最大值,即 $D_{\text{multitask}} = \max\{D_j\}, j=1, 2, \dots, K$. 在处理第 j 个任务 T_j 时,通过式(2)将该任务相对应的个体基因型 \mathbf{Y}_j 映射到实际问题变量中:

$$\mathbf{X}_j = L_j + (U_j - L_j) \cdot \mathbf{Y}_j \quad (2)$$

其中, \mathbf{X}_j 表示个体基因型 \mathbf{Y}_j 对应的实际问题变量值, L_j 与 U_j 分别表示实际问题变量的下限和上限.

算法 1. MFEA 算法.

输入: 种群大小 N , 任务大小 K

输出: 所有任务最优解

1. 在统一搜索空间 $\Omega \in [0, 1]^D$ 随机产生 N 个个体,形成初始种群 \mathbf{P} ;
2. 在每一任务中评价每一个体,并计算出每一个体的技能因子;
3. WHILE (不满足终止条件) DO
4. 通过选择性交配(Assortative Mating, AM)触发后代种群 \mathbf{O} ;
5. 针对当前的任务,根据垂直文化传播(Vertical Cultural Transmission, VCT)在种群 \mathbf{O} 评价个体;
6. 合并 \mathbf{P} 与 \mathbf{O} 获得中间种群 \mathbf{C} ;
7. 更新种群 \mathbf{C} 中每一个体的标量适应度和技能因子;
8. 从种群 \mathbf{C} 中选出 N 个最优个体构成下一代种群 \mathbf{P}_{g+1} ;
9. END WHILE

此外, MFEA 与传统的 EAs 非常相似,它们的最大区别在于选型性交配(Assortative Mating, AM)和垂直文化传播(Vertical Cultural Transmission, VCT). 根据 AM 原则,个体更倾向于与具有相同文化背景(技能因子)的其它个体进行交配,而对于不同文化背景的个体,它们之间交配的概率由随机交配概率(Random Mating Probability, RMP)决定,

RMP 控制任务间的知识转移. VCT 决定对后代个体的评价仅在其继承的技能因子这一任务上进行, 即只需要关心个体最擅长的那个任务, 而无需在所有任务中进行评价, 大大节约 MFEA 的计算时间.

2.3 有关文献回顾

演化多任务优化是知识迁移在优化领域的应用. 自从 Gupta 等人^[11]提出 MFEA 之后, 产生了众多基于 MFEA 结构改进的演化多任务优化算法. 不同的 EAs 搜索机制与 MFEA 结合首先得到研究. Feng 等人首次将粒子群算法与差分进化算法应用到多任务优化中, 分别提出了多因子粒子群优化 (Multi-factorial Particle Swarm Optimization, MFPSO) 算法^[26]与多因子差分进化 (Multifactorial Differential Evolution, MFDE) 算法^[26], MFPSO 和 MFDE 分别基于 PSO 算法和 DE 算法更新个体的选型交配策略, 但同时保留了 MFEA 中的其它结构. 迁移过程的自适应控制策略也成为热点研究方向之一. 为了减少多任务优化过程中负向知识迁移带来的消极影响、提高任务优化的收敛速度, Bali 等人^[27]提出一种基于数据驱动的演化多任务优化算法——MFEA-II, MFEA-II 通过在线学习方式和利用多任务环境中不同任务之间的相似性以增强优化过程. 鉴于现有的演化多任务优化算法通常基于一种常见的知识转移模式, 即通过染色体交叉的隐性遗传转移, 而这种模式不能有效利用不同演化搜索算子中包含的多种形式的搜索偏差, 因此文献^[28]提出了一种跨任务显式遗传转移的 EMT 算法, 即通过自动编码的 EMT, 该算法允许在 EMT 范式中结合不同偏差的多个搜索算子来提升多任务优化问题的求解效率. 文献^[16]提出了一种线性化自适应 EMT 算法 (LDA-MFEA), 该算法将简单任务的搜索空间转换为与其组成的复杂任务相似的搜索空间中, 解决了优化过程任务之间的相似性降低而引起的负向知识转移. Zhong 等人^[29]将 MFEA 与 GP (Genetic Programming) 结合, 提出一种新的多因素 GP (MFGP) 算法, MFGP 由一种新的可扩展染色体编码方案和基于自学习基因表达编程的多任务演化机制组成, 所提出的算法对多任务场景进行了全面的实验研究, 实验结果证实了 MFGP 的有效性. 文献^[17]指出大多数现有的 MFEAs 仅采用单一交叉进行知识转移, 而忽略了不同的交叉算子在产生子代时具有独特的偏好性, 因此, Zhou 等人^[17]提出一种具有自适应知识迁移的演化多任务优化算法 (MFEA-

AKT), 在 MFEA-AKT 中, 用于知识转移的交叉算子沿着演化搜索过程不断地收集自适应反馈信息以实现稳健和高效的多任务优化性能. 近年来, 由数据驱动的优化算法成为了智能优化算法研究的新方向. Jin 等人^[30]指出现实中许多优化问题的目标函数可能不存在, 必须通过计算上昂贵的数值模拟来进行适配性评价, 并讨论了数据驱动演化优化在数据性质和数量方面的主要挑战. Yang 等人^[31]提出了一种由两个代理 (一个粗略模型和一个精细模型) 辅助的进化算法, 粗代理 (CS) 旨在引导算法在搜索空间中快速找到有希望的子区域, 而精细代理则专注于利用好从 CS 传递过来的个体进行知识迁移. 文献^[32]提出了一个基于批量数据驱动的演化多目标优化框架, 该框架通过一次并行评估多个样本来减少数据驱动的演化优化过程的计算时间. 此外, 文献^[33]提出了一种新的两阶段数据驱动演化优化 (TS-DDEO), 在第一阶段, 使用代理辅助分层的粒子群优化方法从整个搜索空间中找到有希望的区域, 而在第二阶段提出了一种具有强大开发能力的的数据驱动优化 (BDDO) 方法以加快优化过程.

演化多任务优化在工程实践中也陆续取得一些成功应用. Gupta 等人^[34]将 MFEA 应用到双层优化工程中, 以演化多任务思维扩展了基本嵌套双层进化算法 (NBLEA), 并提出了一种新的双层进化算法 (M-BLEA), M-BLEA 充分利用不同内层规划任务间的潜在相似信息来加速算法收敛, 使得同时处理多个内层规划任务成为可能. Chandra 等人^[35]将 MFEA 应用到优化神经网络问题上, 提出一种神经网络演化多任务学习算法 (EMTL), 算法将具有不同数量隐藏层神经元的网络拓扑结构视为不同的任务, 通过多任务学习的方法同时进行训练优化. Zhou 等人^[36]提出了一种核化自动编码搜索 (KAES) 的演化多任务范式, 该范式在搜索过程中自适应地选择线性和核化的自动编码, 并将其应用在真实车辆耐撞性设计问题中. 计算机视觉与模式识别经常涉及到稀疏重建问题, Li 等人^[37]扩展了 MFEA 框架, 并在此基础上提出一种演化多任务稀疏重建 (MTSR) 框架, 并将其应用在高光谱图像的稀疏重建问题上, 证明了所提出方法用以解决稀疏重建问题的高效性. 此外, 基于动态资源分配的 EMT 算法在文献^[38]中得到体现, 应用于云计算的 EMT 算法也在文献^[39]进行了描述.

虽然在 EMT 领域出现了许多具有良好性能的

算法,但研究演化超多任务(Evolutionary Many-Task, EMaT)优化算法相对较少. Liaw 等人^[40]基于共生生物群落提出了一种 EMaT 框架(EBS),EBS 根据知识转移的程度自适应地调整任务间信息转移的频率,并通过分享子集合来实现知识转移. Chen 等人^[19]设计一种基于自适应存档的演化多任务优化框架(MaTEA),该框架通过任务之间的相似性和演化过程中知识转移的累积回报为目标任务选择合适的辅助任务. 文献^[41]提出了一种高效的代理辅助多任务进化框架(SaEF-AKT),在 SaEF-AKT 中,计算昂贵的任务在每一代中通过联合在一起来解决,任务的历史搜索信息被用来构建代理模型以减少适应度的评价次数,并使用 KLD(Kullback Leibler Divergence)和基于信息素的方法选择任务进行知识转移,以更好地处理计算成本高昂的任务. Liang 等人^[20]提出了一种基于多源知识转移机制的演化多任务优化算法(EMaTO-MKT),EMaTO-MKT 算法由自适应匹配概率(AMP)策略、基于最大平均差异的任务选择(MTS)策略以及基于局部分布估计的知识转移(LEKT)策略组成,AMP 通过学习演化过程中的经验用来平衡任务内的自我演化趋势和任务间的知识转移程度,MTS 用来计算不同任务子种群中决策变量分布之间的差异,并选择适当数量的任务参与知识转移和缓解负转移,LEKT 用于构建每个任务子种群的概率模型以支持多源知识转移.

现有的 EMaT 优化算法仅适用于处理 Many-task 优化问题,而面对两个任务问题无法有效处理. 此外,虽然上述文献都采取了不同的相似性准则来寻找辅助任务,但它们在迭代中,任务间的知识迁移受限于预先设置的子种群空间,致使任务间知识转移失去了灵活性与动态性,同时任务个体丢失多样性. 我们认为任务间的知识迁移应具有动态、灵活、高效三种特性,使用某种准则粗略地估计空间受限的两个任务之间的相似性,而忽略了种群中真正对任务优化有用的其它个体,这样在任务优化过程中会出现效率低的知识转移,同时导致计算资源的浪费. 另一方面,通过粗略的方式来实现知识转移,只有在两个任务本质相似时才高效. 然而,现实中的应用场景是复杂的,以至于估计出来的任务中的个体只有部分相似. 此外,任务知识迁移的动态性应该发生在跨任务间,而不仅仅限制在某一任务或几个任务中,这就需要算法在迭代中具有自适应地寻找与目标任务个体相似的其它个体的能力. 本文提出基于机器学

习的演化多任务优化框架(MaTML)来解决上述缺点,下一节对所提出算法进行详细描述.

3 提出的方法

3.1 MaTML 架构

在机器学习(Machine Learning, ML)中^[42],多任务学习旨在利用任务(模型)之间的相似性同时解决多个不同任务,如果相关任务的知识可以重用,即可以节省机器学习过程中的工作量. 借鉴 ML 中的多任务学习原理,本文提出基于机器学习的演化多任务优化框架.

图 1 展示了 MaTML 的基本框架,由任务空间、动态子种群、动态寻找辅助任务及优化过程四部分组成. 在整个流程中,待优化的任务首先按特定方式被储存在任务空间中;然后通过任务的技能因子与

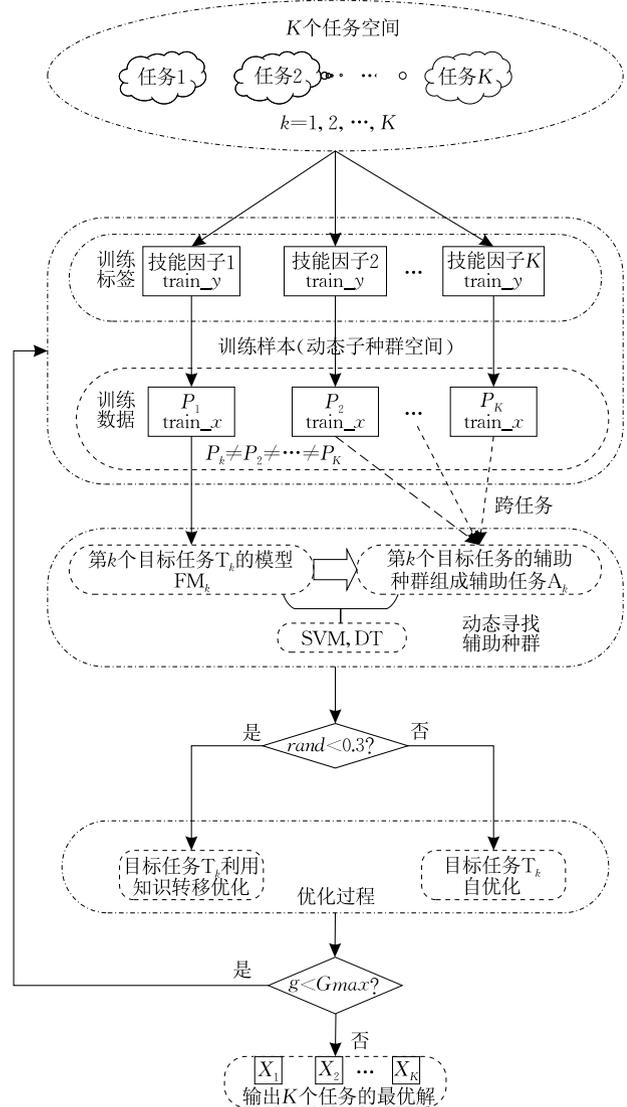


图 1 MaTML 框架

对应的个体形成任务的动态子种群;紧接着,应用机器学习算法寻找目标任务相似的个体构成辅助种群;最后,运用 DE 对目标任务进行优化,促进目标任务与辅助种群正向知识交流. 框架各部分功能具体描述如下:

(1) 任务空间储存 K 个待优化的任务, 各个任务都拥有自己的搜索空间. 为了实现任务间的知识共享, 任务在初始化时, 它们被编码在范围为 $[0, 1]^D$ 的统一搜索空间, 而在优化过程中, 可以解码为任务所对应的实际搜索空间.

(2) 动态子种群(即训练样本)由每个任务对应的个体组成. 注意的是 MaTML 中的任务子种群不像上述 EMaT 文献那样在初始化时为每个任务关联一个大小固定的子种群, 这些文献中任务的子种群大小在整个迭代中均保持不变, 而 MaTML 的动态子种群体现在每代迭代中, 框架能够在种群个体中自适应地根据目标任务关联的技能因子形成训练标签 train_y , 从而找到标签相应的训练数据 train_x . 因此, 无论任务个数为多少, MaTML 均能组成与任务个数一样数量的子种群, 为解决两任务优化情况奠定基础, 这就是 MaTML 具有动态性、灵活性的原因之一. 由于每次迭代后个体擅长的任务可能都不同, 故 MaTML 任务子种群中包含的个体数量也会随迭代变动, 而上述 EMaT 文献任务子种群个体总是限制在设定的空间中迭代. 从这方面讲, MaTML 不仅具有动态性, 同时具有多样性.

(3) 寻找目标任务的辅助种群是通过机器学习算法对目标任务的样本进行训练完成, 然后基于模型预测目标任务的辅助种群, 其过程与上述 EMaT 文献有很大的区别. MaTML 寻找辅助种群中的个体是由模型预测得到, 它的优点有: 一方面, 这种方法可以保证辅助种群中的个体与目标任务子种群中的个体在特征上相似, 使得在优化期间任务间的知识转移具有高效性; 另一方面, 这种方法得到的相似个体是通过跨任务动态获得, 如图 1 所示, 第 k 个任务的辅助种群的来源不限于某一任务或几个任务的子种群中, 而是可以除了目标任务外所有的任务中获取, 这也是 MaTML 具有多样性的原因之一.

(4) MaTML 的优化过程通过 DE 优化器实现. 第(2)、(3)部分已充分准备好了有利于任务间的优化工作, 根据知识转移率 rpm , 为了实现目标任务利用辅助种群的知识转移进行高效优化, 我们这里对 DE 算法的变异算子 $\mathbf{X}_i + F \cdot (\mathbf{X}_{r_1} - \mathbf{X}_i) + F \cdot (\mathbf{X}_{r_2} - \mathbf{X}_{r_3})$ 进行改进, 使得算子中的最后一个差分 $(F \cdot (\mathbf{X}_{r_2} - \mathbf{X}_{r_3}))$ 的个体分别来源于目标任务 (\mathbf{X}_{r_3})

与辅助种群 (\mathbf{X}_{r_2}) 中的个体, 从而促进任务间直接交流, 即目标任务与辅助任务的正向知识得到高效转移.

3.2 基于 MaTML 的 MaTSVM 算法

本小节, 我们将机器学习算法库中的支持向量机(Support Vector Machines, SVM)与 MaTML 框架结合, 提出 MaTSVM 演化多任务算法. 为了更加清楚阐明 MaTML 如何与 SVM 结合, 这里简要介绍一下 SVM. SVM 是求解二分类问题的一种重要机器学习算法, 它可以通过一个超平面(间隔最大化)将二分类样本正确开分, 且 SVM 中的核函数可以度量数据空间样本点的相似性. 如图 2 所示.

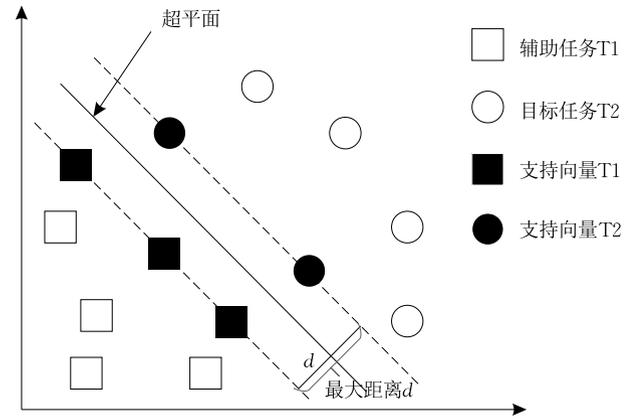


图 2 SVM 原理图

图 2 中, 超平面由法向量 \mathbf{W} 与截距系数 b 决定, 其方程为 $\mathbf{X}^T \mathbf{W} + b = 0$, 位于超平面左下方的样本(本文为辅助任务对应的个体)为辅助任务 T1, 右上方样本(本文为目标种群对应的个体)为目标任务 T2, 样本点中与分离超平面距离最近的样本点即为支持向量(Support Vector), SVM 工作原理就是要最大化支持向量之间的最大距离 d . 令

$$f(x) = \mathbf{X}^T \mathbf{W} + b \quad (3)$$

若 $f(x) = 0$, x 位于超平面上; 对于所有满足 $f(x) < 0$ 的点, 设对应的 $y = -1$; 而所有满足 $f(x) > 0$ 的点, 设对应的 $y = 1$, 即式(3)转化为求式(4)的最大值优化问题:

$$\begin{aligned} \max(d) &= \frac{1}{\|\mathbf{W}\|} \\ \text{subject to: } & y_i (\mathbf{X}_i^T \mathbf{W} + b) \geq 1 \end{aligned} \quad (4)$$

其中, $\|\mathbf{W}\|$ 表示法向量 \mathbf{W} 的模, y_i 表示样本的目标函数值. 在 MaTSVM 中, 我们将种群个体看作样本集, 目标任务所对应的个体看作训练集, 与目标任务对应的技能因子作为机器学习标签, 通过 SVM 算法学习, 寻找与目标任务相似的辅助任务.

算法 2 描述了 MaTSVM 的伪代码, 其过程主要

分为技能因子计算、训练集及标签选取、辅助种群选取及生成下一代种群四部分组成。

算法 2. MaTSVM 算法.

输入: 种群大小 N , 任务大小 K , 知识转移率 rpm , 最大迭代次数 $Gmax$, $g=1$

输出: 所有任务最优解

1. 在 D 维的统一搜索空间 $\Omega \in [0, 1]^D$ 中随机产生 N 个个体, 形成初始种群 \mathbf{P} ;
2. 对种群中的个体进行评价;
3. WHILE $g \leq Gmax$ DO
4. 计算出个体的技能因子;
5. FOR $k=1$ to K
6. 针对当前任务 k , 找出对应的所有技能因子 $train_y$ 及其个体 $train_x$;
7. IF $K=2$ THEN
8. IF $length(train_y) > N/2$ THEN
9. 随机选取 $train_x$ 的一半组成当前任务 k 的种群个体;
10. END IF
11. END IF
12. 通过 SVM 对 $train_x$ 与 $train_y$ 进行训练, 拟合学习模型 $fitsvm$;
13. 用模型 $fitsvm$ 对不属于任务 k 的种群个体进行预测, 得到与目标任务 k 大小相等的辅助任务个体 $assist_train_x$;
14. FOR $i=1$ to $length(train_x)$ DO
15. IF $rand(0, 1) < rpm$ THEN
16. 执行式(5)产生变异个体;
17. ELSE
18. 执行式(6)产生变异个体;
19. END
20. 执行式(7)进行交叉操作;
21. 重新评价个体;
22. 执行式(8)进行选择操作产生下一代个体;
23. 更新种群 \mathbf{P} ;
24. END FOR
25. END FOR
26. $g = g + 1$;
27. END WHILE

Part1. 技能因子计算. 算法在 D 维空间 $[0, 1]^D$ 中初始化大小为 N 的统一种群 \mathbf{P} , 并计算种群中的个体在各个任务中的适应度, 计算各个任务的技能因子, 亦即每个个体所擅长的任务编号, 伪代码中由第 1 行到第 4 行完成。

Part2. 训练样本, 即目标任务子种群选取. 从第 5 行到第 11 行是算法为目标任务选取训练样本的过程, 如图 3. 首先, 基于 Part1 中的任务编号将所有个体划分到 K 个任务中形成任务子种群, 假设任务

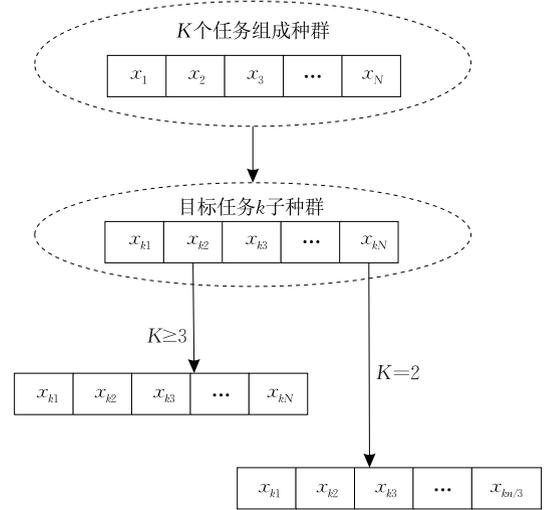


图 3 阐明目标任务训练样本的选取示意图

k 对应的编号个体有 n 个, 那么, 将这些个体的技能因子标签 $train_y$ 设置为 1, n 个体组成的子种群即为任务 k 的训练数据 $train_x$, $train_x$ 与 $train_y$ 合在一起形成任务 k 的训练样本. 若任务量为两个时 ($K=2$), 即随机从训练样本中选取一半的个体用于模型训练; 若任务数量大于或等于三个时 ($K \geq 3$), 即把任务 k 的训练样本全部用于模型训练. 这是因为当 $K=2$ 时为了防止任何一个任务的技能因子数目超过 $N/2$ (即该任务的子种群超过 $N/2$) 而另一个任务的技能因子数目小于 $N/2$, 从而导致得到的辅助任务子种群大小 $|A_k|$ 与目标任务 $|T_k|$ 的大小不一致, 致使 A_k 与 T_k 无法等量交流 (伪代码第 14 行~24 行). 就这点而言, MaTSVM 解决了在优化两个任务时无法选择辅助任务的问题, 同时使得任务间知识转移不失去灵活性。

Part3. 辅助种群选取. 第 12 行到第 13 行是算法为任务 k 选取辅助任务过程. 首先应用 SVM 对 $train_x$ 与 $train_y$ 合并的数据 (训练样本) 采用十折交叉法对第 k 个任务进行学习, 考虑到数据特征 (任务维度) 较多, 且超平面为非线性, 本文在训练中应用高斯核 (Gaussian Kernel) 函数^[43] 将原始空间数据映射到高维空间, 解决在原始空间中线性不可分的问题, 并采取序列最小最优^[44] (Sequential Minimal Optimization, SMO) 优化式 (3) 中的约束参数. 假设第 k 任务经学习得到的模型为 FM_k , 然后用模型对不属于任务 k 的种群个体进行预测, 得到预测集 $temp$. $temp$ 代表与任务 k 相似的个体集合, 由 $|P_k|$ 个个体组成. 最后在 $temp$ 中选取与 $train_x$ 大小相等的辅助种群 $assist_train_x$. 在训练中, 我们采用高斯内核为核函数, 且对其主要参数进行设置. 其中,

惩罚参数 $C=20$, 核系数 $\sigma=0.05$, 精确度(容忍度) $tol=1 \times 10^{-3}$, 交叉折数 $kflod=10$.

Part4. 生成下一代种群. 第 14 行到第 24 行为算法生成后代个体过程. 在此过程中, 本文采取 DE 算法触发后代个体. 考虑相似任务彼此之间拥有更多的共享知识来促进各自的优化进程, 且兼顾到 DE 算法的全局与局部搜索平衡. 因此, 算法在执行第 16 行时, 此时为算法利用任务间的知识转移对目标任务进行优化阶段, 为了充分利用相似任务具有正向知识迁移的优势, 从而加速任务收敛速度, 图 4 所示, 我们应用式(5)中的变异算子触发新个体:

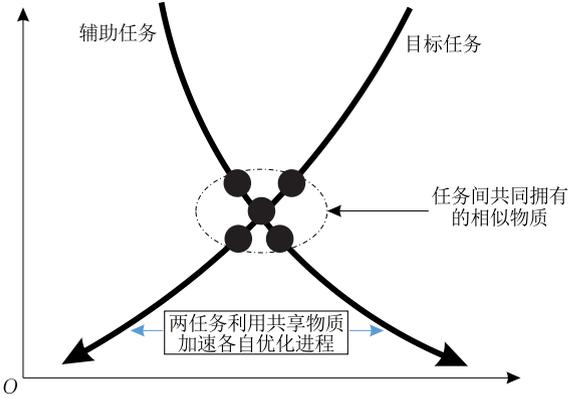


图 4 阐明相似任务间的优化进程

$$\mathbf{V}_{i,g} = \mathbf{X}_{i,g} + F \cdot (\mathbf{X}_{pbest,g} - \mathbf{X}_{i,g}) + F \cdot (\mathbf{X}_{r1,g}^a - \mathbf{X}_{r2,g}) \quad (5)$$

其中, $\mathbf{V}_{i,g}$ 为变异个体, $\mathbf{X}_{i,g}$ 为目标个体, $\mathbf{X}_{pbest,g}$ 代表从当代前 $p\%$ 最优的个体中随机选择的个体, $p=5$, $\mathbf{X}_{r1,g}^a$ 从目标任务的辅助种群 `assist_train_x` 中选取, $\mathbf{X}_{r2,g}$ 从目标任务 `train_x` 选取. 这种选取个体的方式从意义上实现了目标任务与辅助种群之间的直接交流, 提高知识转移效率. 当算法执行第 18 行时, 此时任务 k 处于自优化阶段, 我们应用式(6)中的变异算子触发新个体:

$$\mathbf{V}_{i,g} = \mathbf{X}_{i,g} + F \cdot (\mathbf{X}_{pbest,g} - \mathbf{X}_{i,g}) + F \cdot (\mathbf{X}_{r1,g} - \mathbf{X}_{r2,g}) \quad (6)$$

其中, $\mathbf{V}_{i,g}$ 为变异个体, $\mathbf{X}_{i,g}$ 为目标个体, $\mathbf{X}_{pbest,g}$ 代表从当代前 $p\%$ 最优的个体中随机选择的个体, $p=5$, $\mathbf{X}_{r1,g}$ 与 $\mathbf{X}_{r2,g}$ 均为从任务 k 选取的个体, 这种选取方式实现了利用任务自身特点进行优化. 在第 20 行算法按式(7)执行交叉操作:

$$\mathbf{U}_{i,j,g} = \begin{cases} \mathbf{V}_{i,j,g}, & \text{rand}(0,1) \leq CR \text{ or } j == l \\ \mathbf{X}_{i,j,g}, & \text{其他} \end{cases} \quad (7)$$

其中, $\mathbf{U}_{i,g}$ 代表实验个体, j 表示个体维度索引, $l \in [1, D]$ 的随机整数 (D 表示维度), 确保 $\mathbf{U}_{i,g}$ 至少有

一个分量来自 $\mathbf{V}_{i,g}$. 最后算法执行式(8)产生下一代个体:

$$\mathbf{P}_{i,g+1} = \begin{cases} \mathbf{U}_{i,g}, & \text{if } f(\mathbf{U}_{i,g}) < \text{if } f(\mathbf{X}_{i,g}) \\ \mathbf{X}_{i,g}, & \text{其他} \end{cases} \quad (8)$$

其中, $\mathbf{P}_{i,g+1}$ 表示下一代个体.

算法会从 WHILE 循环中不断迭代执行 Part1~Part4, 直到满足停止条件为止结束, 最终输出 K 个任务的所有最优解. 从上述步骤看到, MaTSVM 算法在实现任务间的知识迁移具有动态性, 这种动态性体现在算法选择辅助种群时, 辅助种群的选取不限于预先分配的固定大小子种群进行, 而是在整个种群 \mathbf{P} 中动态地根据模型的预测结果中选择, 所以提高了任务间知识转移的灵活性.

3.3 时间复杂度分析

与 MFEA 相比, MaTSVM 在时间复杂度计算方面增加的主要部分为 SVM 模型训练. 具体分析如下. 参考文献[45], 假设样本大小为 N (种群 \mathbf{P} 大小), 特征数为 D (个体维度), N_s 为支持向量 (见图 2) 数目; 当支持向量数目远小于样本数量时, 即 $N_s \ll N$, 此时 SVM 训练的时间复杂度为 $O(N_s^3 + N_s^2 \cdot N + N_s \cdot D \cdot N)$; 当支持向量数目与样本数量大约相等时, 即 $N_s \approx N$, 此时 SVM 训练的时间复杂度为 $O(N_s^3 + N_s \cdot N + N_s \cdot D \cdot N)$; 在极端情况 ($N_s \approx N$, 且所有的支持向量均在边界上), SVM 训练的时间复杂度达到最大值 $O(D \cdot N^2)$. 因此, SVM 训练的时间复杂度为 $O(N_s^3 + N_s \cdot N + N_s \cdot D \cdot N) \sim O(D \cdot N^2)$ 之间. 算法在一次迭代中, 假设每个个体评价的时间复杂度 $O(f)$, 算法在种群评价时 (算法第 2 行) 的时间复杂度为 $O(N \cdot f)$, 在迭代循环 (算法第 3 行)、优化任务循环 (算法第 5 行) 及生成下一代种群 (算法第 14 行) 的时间复杂度为 $O(K \cdot n)$, n 为目标任务 `train_x` 大小, 重新评价任务子种群的时间复杂度 (第 21 行) 为 $O(n \cdot f)$. MaTSVM 总的时间复杂度为 $O(N \cdot f + K \cdot n^2 \cdot f) \cdot O(\text{SVM})$.

本文中, 由于参与模型训练的样本数 (`trin_x` 的大小 n) 一般情况下都远小于种群 N , 任务维度 $D=50$, 因此, 即使最坏情况下 SVM 的时间复杂度为 $O(D \cdot n^2)$, 算法也在有限的时间内完成所有的步骤.

4 实验与分析

在本节中, MaTSVM 与现阶段的 EMT 算法、EMaT 算法分别在 multi-task 测试集 CEC17 和 multi-task 测试集 WCCI2020SO 进行比较以验证其性能.

除此之外,有关 MaTSVM 中组件的必要性测试分析、模型性能分析以及模型稳定分析也在实验中进行.最后,传感器覆盖率真实优化问题也用来检验 MaTSVM 的有效性.

4.1 测试问题集和比较算法

CEC17 测试集共有 9 个测试问题,每个问题包含 2 个最小化任务.根据全局最优解的相交程度,CEC17 测试集可分为三类:完全相交(Complete Intersection, CI)、部分相交(Partial Intersection, PI)和无相交(No Intersection, NI).其次,根据函数适应度外观之间的相似性,这些问题也可以分为三组:高相似性(High

Similarity, HS)、中等相似性(Medium Similarity, MS)和低相似性(Low Similarity, LS).有关这些基准问题的更多详细信息在文献[23]中能找到.

WCCI2020SO 测试集共有 10 个测试问题,每个问题包含 50 个最小化任务,这 500 个任务由 Sphere、Rosenbrock、Ackley、Rastrigin、Griewank、Weierstrass 和 Schwefel 共 7 个函数基于转移(shifted)、旋转(rotated)操作生成,如表 1.其中,P1、P2、P3 分别由对应的单独函数 Sphere、Rosenbrock 和 Rastrigin 组成,它们都包含 50 个同质任务;P4~P10 由多个函数混合组成,它们都包含 50 个异质任务.

表 1 WCCI2020SO 测试集组成结构表

函数	问题名称									
	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10
Sphere	✓			✓						
Rosenbrock		✓		✓		✓		✓	✓	
Rastrigin			✓		✓		✓	✓	✓	✓
Ackley				✓			✓	✓	✓	✓
Griewank					✓	✓		✓	✓	✓
Weierstrass					✓		✓	✓	✓	✓
Schwefel						✓			✓	✓
问题特征	shifted rotated									

所比较的算法包含 EMT 优化算法和 EMaT 优化算法.具体讲,对于多任务(multi-task),MaTSVM 分别与 MFEA^[11]、LDA-MFEA^[16]、MFEA-AKT^[17]和 MTEA-SaO^[46]比较;对于超多任务(many-task),MaTSVM 分别与 MFEA、MaTEA^[19]和 EMaTO-MKT^[20]比较.其中,MFEA 为经典的 EMT 优化算法,LDA-MFEA、MFEA-AKT 和 MTEA-SaO 擅长于处理少量任务的 EMT 优化算法,而 MaTEA 和 EMaTO-MKT 被设计为专门处理超多任务的 EMT 优化算法.这些算法详细的介绍见 2.3 小节.

4.2 参数设置

在多任务测试集 CEC17 实验中,MaTSVM 的种群大小设置为 $N=300$,DE 的缩放因子 $F=0.5$,交叉因子 CR 根据式(9)在 $[0.4, 0.6]$ 范围中取值,知识转移率 $rpm=0.3$;MFEA、LDA-MFEA、MFEA-AKT 及 MTEA-SaO 有关参数设置同原文献[11]、[16]、[17]和文献[46]保持一致.每种算法最大迭代次数为 $Gmax=1000$ 代,实验使用不同的随机种子进行 25 次独立运行.

在超多任务测试集 WCCI2020SO 实验中,MaTSVM 的种群大小设置为 $N=7500$,DE 的缩放因子 $F=0.5$,交叉因子 CR 根据式(9)在 $[0.4, 0.6]$ 范围中取值,知识转移率 $rpm=0.3$;根据文献[19],

MFEA 的种群大小设置为 $N=5000$,MaTEA 的种群大小 $N=5000$;根据文献[20],EMaTO-MKT 种群大小设置为 $N=5000$.其余特有的参数设置同原文献[11]、[19]和[20]保持一致.每种算法最大迭代次数为 $Gmax=1000$ 代,实验使用不同的随机种子进行 25 次独立运行.

$$CR=0.4+(0.6-0.4)^{(g/Gmax)^2} \quad (9)$$

4.3 在 CEC17 问题集的测试结果

所有 MFEAs 的实验测试结果如表 2 所示.在实验中,目标函数适应度的平均值被用来评价算法的性能,非参数统计测试(具有 5% 显著水平的 Wilcoxon 符号秩检验)用来测试 MaTSVM 算法与其余 4 个 MFEAs 之间的显著差异.其中,加粗部分表示最好的结果,符号“+”、“-”、“ \approx ”表示所比较的算法比 MaTSVM 算法在结果上“好”、“差”、“大约相等”.

从表 2 中可知,总共 18 个任务中 MaTSVM 在性能上有 9 个优于其它 MFEAs,达到了任务总数的一半,特别在问题 P5 和 P8 的所有任务中 MaTSVM 获得全部胜出.注意到 MTEA-SaO 与 MaTSVM 具有竞争性,且获得 7 个任务的最优解,这可能是由于 MTEA-SaO 具有自适应求解器选择策略.MFEA-

表 2 MaTSVM 与 MFEA、LDA-MFEA、MFEA-AKT 和 MFEA-SaO 在 CEC17 的实验结果 ($D=50$)

问题	任务	MFEA	LDA-MFEA	MFEA-AKT	MTEA-SaO	MaTSVM
CI+HS(P1)	T1	3.97e-01-	7.62e-01-	3.99e-01-	1.02e-02-	2.88e-04
	T2	2.34e+02+	1.61e+02+	2.22e+02+	3.47e+01+	3.64e+02
CI+MS(P2)	T1	8.84e+00-	7.34e+00-	1.02e+01-	1.57e+00-	3.46e-01
	T2	5.21e+02-	3.84e+02≈	6.64e+02-	3.05e+02+	3.82e+02
CI+LS(P3)	T1	2.01e+01+	2.09e+01+	2.01e+01+	2.12e+01≈	2.12e+01
	T2	4.87e+03+	4.77e+03+	4.32e+03+	3.61e+03+	1.15e+04
PI+HS(P4)	T1	9.04e+02-	9.85e+02-	7.53e+02-	1.88e+02+	3.86e+02
	T2	7.72e+00-	2.27e+01-	6.02e+00-	3.13e+00-	2.35e-06
PI+MS(P5)	T1	3.66e+00-	4.46e+00-	4.27e+00-	1.44e+00-	9.47e-01
	T2	1.12e+03-	2.11e+03-	5.47e+02-	3.50e+02-	1.20e+02
PI+LS(P6)	T1	2.02e+01-	2.04e+01-	7.10e+00-	1.81e-01-	6.57e-02
	T2	1.73e+01-	1.95e+01-	8.96e+00-	2.08e-01+	4.87e-01
NI+HS(P7)	T1	2.20e+03-	2.48e+03-	7.04e+02-	7.71e+01-	5.52e+01
	T2	4.35e+02-	3.58e+02+	3.62e+02+	1.03e+02+	4.05e+02
NI+MS(P8)	T1	3.99e-01-	6.39e-01-	3.44e-01-	1.02e-01-	1.90e-04
	T2	2.96e+01-	2.51e+01-	2.54e+01-	3.10e+01-	5.55e+00
NI+LS(P9)	T1	7.46e+02-	1.55e+03-	6.65e+02-	2.41e+02+	3.80e+02
	T2	3.26e+03+	5.20e+03+	3.65e+03+	3.46e+03+	1.09e+04
Friedman rank	/	3.6389	4.0000	3.1944	1.9267	2.0500

AKT、MFEA 仅分别在 P3 的任务 T1、P9 的任务 T2 击败 MaTSVM, 而 LDA-MFEA 仅在 P2 的任务 T2 处获得与 MaTSVM 大约相同的结果. 因此, 从任务数量上分析, MaTSVM 在获得的最优解数目上比 MFEA-AKT、LDA-MFEA、MFEA 和 MTEA-SaO 表现更佳.

进一步分析发现, 根据统计学知识, 基于 Friedman test 结果得出的平均排名 MTEA-SaO 第一, MaTSVM 紧跟其后, 其次分别是 MFEA-AKT、MFEA 和 LDA-MFEA. 总之, 在 CEC17 问题测试集上, MaTSVM 的综合性性能显著优于 MFEA-AKT、LDA-MFEA 和 MFEA, 且与 MTEA-SaO 形成竞争性. MaTSVM 其成功取决于两个因素:

(1) 在寻找目标任务的辅助种群过程中, 通过机器学习算法 SVM 可以在动态中准确地预测辅助种群中的个体. 与其它个体相比, 辅助种群个体与目标任务种群个体更加接近或相似, 为任务间的正向知识转移奠定了基础;

(2) 在任务优化中, MaTSVM 采用了式(5)的方式实现相似任务间直接交流, 这种方式使得目标任务与辅助种群的正向知识转移可以更积极、更加高效, 同时具有更好的开发与探索权衡, 从而加速任务收敛速度.

此外, 从图 5 中的收敛曲线看到 MaTSVM 的收敛速度在所比较的 MFEAs 中总体上也是最好的, 具体地讲:

(1) 对于 CI+HS(P1)问题, 注意到 P1 为完全相交的高度相似问题, 该问题为 9 个问题中最容易

优化, MaTSVM 与 MTEA-SaO 分别在任务 T1 和 T2 上获得最好的收敛性. MTEA-SaO 在任务 T2 上优于 MaTSVM 的原因是其具有自适应求解器选择策略, 该策略可以利用任务之间的隐式相似性实现知识转移.

(2) 对于 CI+LS(P3)问题, MFEA-AKT 与 MTEA-SaO 分别在任务 T1 和 T2 上获得最好的收敛性, 归结于 MFEA-AKT 在优化过程中按照实际需要动态地选择不同的交叉算子, 提高了算法在低度相似问题的搜索能力和任务间的知识迁移效果. 注意到, MaTSVM 在优化 T1 时获得与 MTEA-SaO 大约一致的性能.

(3) P5 为部分相交的中度相似问题, 此时在统一搜索空间中, 每个任务的全局最优值彼此较远, 因此对其优化的难度会有所增大, 而 MaTSVM 在该问题的所有任务中均获得最好的收敛性能, 说明 MaTSVM 在机器学习算法 SVM 及任务之间直接交流(式(5))帮助下成功地处理 P5 问题.

(4) P7 与 P8 为不相交问题. 在 P7 上, MaTSVM 与 MTEA-SaO 分别在任务 T1 和 T2 上获得最好的收敛性, 进一步分线发现, MaTSVM 在 P8 上击败了其它算法获得最佳收敛性能, 进一步说明 MaTSVM 所提出的机制优越性.

综合上述, 从表 2 的实验结果与图 5 的收敛曲线图可知, 无论在任务数量上或任务优化过程中的收敛趋势上分析, 实验结果均证实了 MaTSVM 具有更优秀的性能, 换句话说, MaTSVM 在所有 4 种所比较的 MFEAs 中表现更出色.

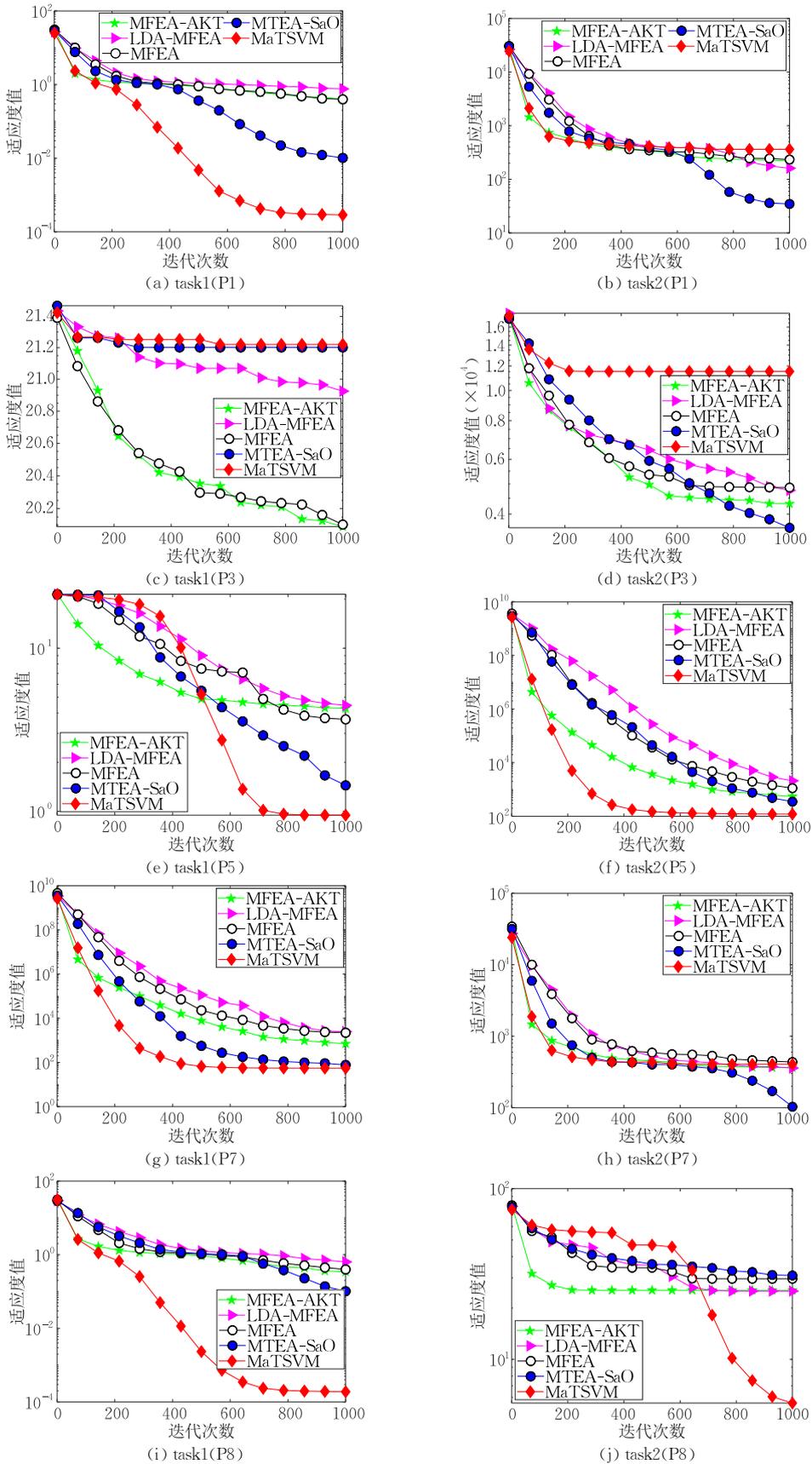


图5 MaTSVM与4个MFEAs算法在P1、P3、P5、P7、P8中的收敛曲线图

4.4 在 WCCI2020SO 问题集的测试结果

所有的 MFEAs 的实验测试结果如表 3 所示. 在实验中, 目标函数适应度的平均值被用来评价算法的性能, 非参数统计测试(具有 5% 显著水平的 Wilcoxon

符号秩检验)用来测试 MaTSVM 算法与其余 3 个 MFEAs 之间的显著差异. 其中, 加粗部分表示最好的结果, 符号“+”、“-”、“ \approx ”表示所比较的算法比 MaTSVM 算法在结果上“好”、“差”、“大约相等”.

表 3 MaTSVM 与 MFEA、MaTEA、EMaTO-MKT 在 WCCI2020SO 每个问题前 5 个任务实验结果 ($D=50$)

问题	任务	MFEA	MaTEA	EMaTO-MKT	MaTSVM
P1	T1	1.85e+01-	1.80e-04-	1.31e-25+	2.58e-20
	T2	2.72e+01-	5.76e-04-	3.36e-25+	1.38e-18
	T3	1.34e+01-	1.05e-03-	2.64e-23+	5.76e-23
	T4	1.998e+01-	2.72e-04-	1.18e-25+	1.52e-22
	T5	1.82e+01-	1.93e-04-	2.44e-25+	1.57e-18
P2	T1	1.67e+04-	2.92e+02-	5.44e+01-	4.72e+01
	T2	2.63e+03-	2.82e+02-	1.04e+02+	1.18e+02
	T3	2.24e+04-	6.82e+02-	2.81e+02-	4.91e+01
	T4	1.01e+04-	2.83e+02-	1.25e+02-	1.15e+02
	T5	1.40e+05-	7.19e+02-	4.86e+01 \approx	4.81e+01
P3	T1	5.39e+02-	4.54e+02-	1.83e+02+	3.60e+02
	T2	5.64e+02-	5.03e+02-	1.26e+02+	3.69e+02
	T3	5.56e+02-	4.74e+02-	9.25e+01+	3.86e+02
	T4	6.53e+02-	4.69e+02-	2.48e+02+	3.77e+02
	T5	6.59e+02-	4.59e+02-	1.48e+02+	3.58e+02
P4	T1	2.69e+01-	2.00e+00-	1.02e-03-	1.65e-13
	T2	9.51e+04-	3.65e+04-	7.25e+01-	4.77e+01
	T3	6.08e+00-	2.46e+00-	2.93e-02-	6.24e-07
	T4	2.06e+01-	2.36e+00-	4.81e-09-	3.94e-13
	T5	1.34e+05-	4.02e+02-	1.19e+02-	4.83e+01
P5	T1	5.99e+02-	4.88e+02-	9.85e+01+	3.88e+02
	T2	6.80e-01-	3.16e-01-	2.05e-02-	3.40e-07
	T3	3.85e+01-	2.37e+01-	2.03e+01-	5.42e-02
	T4	1.12e+03-	4.65e+02-	1.24e+02+	3.44e+02
	T5	5.56e-01-	3.67e-01-	1.01e-02-	5.20e-06
P6	T1	3.16e+05-	1.92e+03-	1.06e+02+	3.77e+02
	T2	6.39e-01-	3.13e-01-	3.42e-03-	4.34e-06
	T3	7.50e+03+	1.45e+04-	5.27e+03+	1.26e+04
	T4	2.96e+03-	1.58e+03-	2.85e+02-	4.82e+01
	T5	6.10e-01-	3.07e-01-	4.27e-03-	3.99e-06
P7	T1	6.74e+00-	2.43e+00-	7.82e-02-	1.90e-06
	T2	4.65e+02-	4.62e+02-	8.90e+01+	3.79e+02
	T3	4.41e+01-	1.99e+01-	2.89e+01-	6.43e-01
	T4	8.56e+00-	2.37e+00-	3.58e-04-	2.57e-06
	T5	5.93e+02-	4.59e+02-	1.68e+02+	3.94e+02
P8	T1	5.39e+03-	3.09e+03-	1.50e+02+	1.58e+02
	T2	2.02e+01-	8.37e+00-	1.95e+00-	1.70e-04
	T3	8.91e+02-	4.48e+02-	1.24e+02+	3.71e+02
	T4	6.76e-01-	4.86e-01-	1.78e-03-	3.42e-08
	T5	5.99e+01-	5.62e+01-	2.33e+01-	2.25e+00
P9	T1	1.09e+05-	1.29e+03-	4.95e+03-	1.15e+02
	T2	2.03e+01-	8.31e+00-	1.08e-01-	3.10e-04
	T3	5.43e+02-	4.53e+02-	1.45e+02+	3.87e+02
	T4	5.93e-01-	3.01e-01-	4.28e-03-	8.61e-07
	T5	5.56e+01-	5.62e+01-	3.27e+01-	2.67e+00
P10	T1	2.02e+01-	7.83e+00-	7.53e-01-	1.78e-03
	T2	8.45e+02-	4.24e+02-	1.42e+02+	3.74e+02
	T3	6.18e-01-	2.43e-01-	1.29e-03-	2.82e-05
	T4	5.57e+01-	6.21e+01-	3.20e+01-	2.95e+00
	T5	4.31e+03+	1.40e+04 \approx	7.16e+03+	1.35e+04
Friedman rank	/	3.8600	3.0400	1.6200	1.4800

注:鉴于篇幅有限,表 3 显示的是每个问题中前 5 个任务的实验结果.表 4 为 MaTSVM、MFEA、MaTEA 和 EMaTO-MKT 在 10 个问题的所有任务中获得的最好解总数.下面结合实验数据进行详细分析.

表 4 MaTSVM 与 MFEA、MaTEA、EMaTO-MKT 在 WCCI20SO 每个问题的所有任务中获得的最好解数目

问题	算法			
	MFEA	MaTEA	EMaTO-MKT	MaTSVM
P1	0	0	47	3
P2	0	0	17	33
P3	0	0	50	0
P4	0	0	5	45
P5	0	0	17	33
P6	3	1	16	30
P7	0	0	17	33
P8	0	0	15	35
P9	4	0	16	30
P10	2	0	19	29

(1)从表 1 可知,P1~P3 各自的 50 个任务分别由函数 Sphere、Rosenbrock 和 Rastrigin 基于不同的转移向量(shifted vector)与旋转矩阵(rotated matrix)生成,问题中所包含的任务均为同质任务,故算法对问题中任务的优化会获得稳定的结果.其中,Sphere 为单峰函数、Rosenbrock 和 Rastrigin 为多峰函数,因此 P1 为最简单的问题,其所对应的 50 个任务也相对容易优化.由表 4 得到, MaTSVM、MFEA、MaTEA 和 EMaTO-MKT 在问题 P1 的所有任务中获得的最好解数目分别是 3、0、0、47;在问题 P2 的所有任务中获得的最好解数目分别是 33、0、0、17;在问题 P3 的所有任务中获得的最好解数目分别是 0、0、0、50.由此知道对于同质任务组成的问题 P1~P3,除了 P2 之外,EMaTO-MKT 表现最佳,但在问题 P2 上, MaTSVM 性能最好.同时注意到,对于 P1,虽然 MaTSVM 在解的总数上被 EMaTO-MKT 击败,但从表 3 中看到两个算法在处理 P1 问题性能比较接近.因此,就问题 P1 而言, MaTSVM 相对于 EMaTO-MKT 具有竞争性.图 6(P1~P2)为 MaTSVM、MFEA、MaTEA 和 EMaTO-MKT 在 P1~P2 前 5 个任务获得的平均值收敛曲线图(对应表 3).

(2)问题 P4~P7 均由 3 个不同的函数基于不同的转移向量与旋转矩阵生成,它们各自对应的 50 个任务为异质的,因此算法对这些任务优化会有所差异,且比 P1~P3 复杂.具体地讲,问题 P4 由函数 Sphere、Rosenbrock、Ackley 组成,由表 4 可知, MaTSVM、MFEA、MaTEA 和 EMaTO-MKT 在该问题的 50 个任务中获得的最好解数目分别是 45、0、0、5;问题 P5 由函数 Rastrigin、Griewank、

Weierstrass 组成, MaTSVM 在该问题的 50 个任务中得到 33 个最优解;同理分析得到, MaTSVM 在问题 P6、P7 的 50 个任务中分别以总数 30、33 胜出.就 P4~P7 而言, MaTSVM 以绝对的优势击败了 MFEA、MaTEA 和 EMaTO-MKT,说明了 MaTSVM 中所提出的策略具有合理性,能够动态地为上述异质任务找到相似的辅助任务,从而实现了任务间的正向知识转移.此外,对于 P4,由函数 Rosenbrock 构成的所有任务具有“不可分离、多峰”等特性,且从局部最优到全局最优存在非常窄的低谷,故对其优化难度较大. P4 的前 5 个任务中 T2 和 T5 由 Rosenbrock 生成,从表 3 中的实验结果可以印证 T2 和 T5 的优化难度比 T1、T2 及 T4 大,图 6(P4)为所有算法在 P4 中任务 T2 和 T5 的平均值收敛曲线图(对应表 3),从图中看到 MaTSVM 的收敛速度最快. P5 的 3 个函数中 Rastrigin 具有多个局部最优解,因此其对应的任务在优化过程中容易使得算法陷入局部最优, P5 的前 5 个任务中 T1 和 T4 由 Rastrigin 生成,图 6(P5)为所有算法在 P5 中任务 T1 和 T4 的平均值收敛曲线图(对应表 3),从图中看到 MaTSVM 的收敛速度仅次于 EMaTO-MKT. 问题 P7 由函数 Rastrigin、Ackley、Weierstrass 组成,其中,由 Ackley 生成的任务最为复杂,因为 Ackley 函数具有更多的局部最优值,从函数外观图看,它的全局最优值位于狭窄的山谷中,因此有利于测试算法的搜索能力. P7 的前 5 个任务中 T1 和 T4 由 Ackley 生成,图 6(P7)为所比较的 MFEAs 在 P7 中任务 T1 和 T4 的平均值收敛曲线图(对应表 3),图中直率地表明了 MaTSVM 拥有良好的收敛性,且在 4 个 MFEAs 中收敛性能最佳,具备处理复杂任务的能力.

(3)问题 P8、P10 均由 5 个不同的函数基于不同的转移向量与旋转矩阵生成,它们所对应为 50 个异质的任务,情况比 P4~P7 更复杂.从表 4 得到 MaTSVM 在 P8、P10 所对应的 50 个任务中最好解数目分别占了 35 个、29 个,占比均超过 50%,证实 MaTSVM 在面对复杂的问题时更有优势,这归功于 MaTSVM 自身拥有的策略.对于 P8 的 50 个任务中,由函数 Weierstrass 生成的任务具有“多峰、不可分离、仅在一组解上可微”等特性,同时存在多个谷底,只有最低的谷底为全局最优解,很考验算法的搜索能力.表 3 中相应的数据为所有 MFEAs 在 P8 上前 5 个任务的实验数据,其中任务 T5 为函数 Weierstrass 生成,图 6(P8)展示了所有算法在 T5

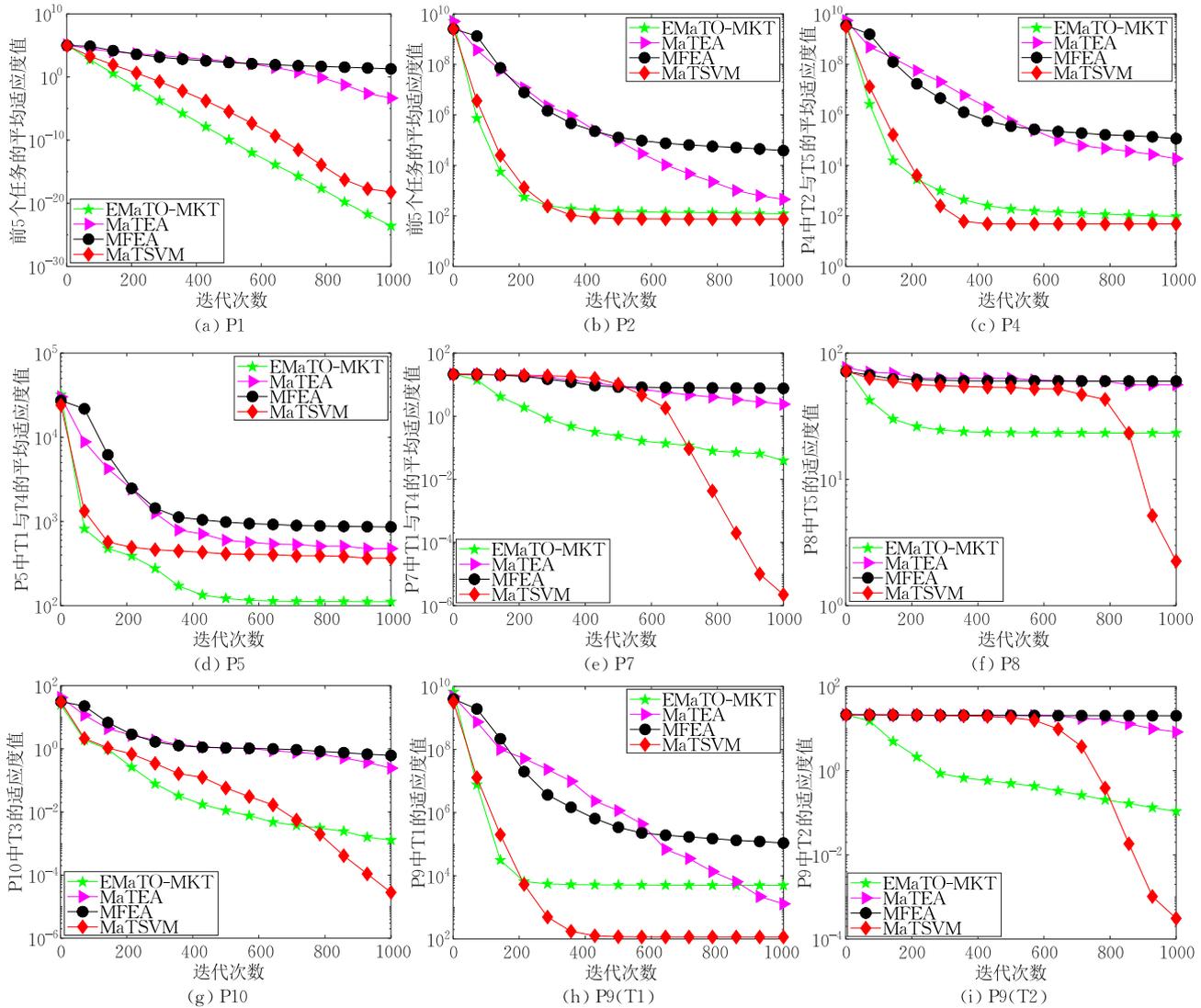


图 6 MaTSVM 与 3 个 MFEAs 算法在不同问题中的收敛曲线图

处的收敛情况(对应表 3),从收敛曲线图再次肯定了 MaTSVM 比其它 3 种算法更适合优化复杂的任务. 在 P10 中,用于组成 P10 的函数 Griewank 为“不可分离”的多峰函数,且它的局部最小值的数量随函数维数呈指数增长,故对其生成的任务具有重要研究意义. P10 的前 5 个任务中 T3 为函数 Griewank 生成,实验测试结果如表 3 所示,图 6(P10)绘出了所有算法在 T3 处的收敛曲线(对应表 3),就解的质量和收敛趋势而言,表 3 和图 6 均证实了 MaTSVM 更好的性能.

(4)除了函数 Sphere 外,其它 6 个函数一起构成了问题 P9,因此 P9 为所有 10 个问题中最为复杂的一个. 从表 4 得出 MaTSVM 在该问题的 50 个任务中获得 30 个最好解,说明 MaTSVM 在 SVM 与知识直接交流方式(式(5))的共同作用下击败了其它 3 个 MFEAs,对 P9 的优化取得了良好的效果.

通过上述分析可知,由函数 Rosenbrock 与函数 Ackley 生成的任务较为复杂,P9 的前 5 个任务中 T1、T2 分别由 Rosenbrock 和 Ackley 生成,其对应任务的优化结果如表 3 所示. 图 6 中的 P9(T1)与 P9(T2)分别为所有 MFEAs 在相应任务的收敛曲线图(对应表 3),图中形象地表明了 MaTSVM 在两个任务的优化过程中收敛速度更胜一筹,这得益于本文所提出的策略.

为了分析算法的计算资源,表 5 列出了 4 种 MFEAs 算法分别在 WCCI2020SO 问题集上独立运行 25 次所需要的平均时间(总时间/次数). 所有计时单位用秒量度,且保留 2 位有效数字. 从表 5 看到,所有的 MFEAs 算法在处理 P1~P10 问题时所需要的运行平均时间都呈逐渐增长的趋势,且 EMaTO-MKT 耗消的时间最多,其次为 MaTSVM、MaTEA 和 MFEA. EMaTO-MKT 平均运行时间

最多,我们推测这是因为它里面包含的最大平均差异选择策略(MTS)及局部分布估计的知识转移策略(LEKT)分别在计算任务间的相似性和执行知识转移时消耗了大量的计算资源. 因为 MTS 是应用机器学习方法 MMD(最大平均差异)通过高斯核函数将原始种群空间的数据映射到高维空间中,而 LEKT 通过遍历目标任务个体,然后运用高斯分布实现目标任务与辅助任务知识迁移,这两个过程都会增加算法的运行时间. 对于 MaTSVM,虽然每次迭代都要对模型进行更新,但每次训练模型时不需要全部的种群,只需要用到与目标任务关联的子种群即可,这会减低了算法的运行时间. 此外,所有算法在处理 P1~P3、P4~P7、P8 和 P10、P9 这 4 个分段的问题上的运行时间也是呈递增趋势,深入分析我们得到如下的原因: P1~P3 为同质任务的问题,它们各自所包含的 50 个任务均是一个函数组成(见 4.1 节); P4~P7 它们各自所包含的 50 个任务由 3 个不同的函数组成; P8 和 P10 各自所包含的 50 个任务由 5 个不同的函数组成;而 P9 为最复杂的问题,它包含的 50 个任务由 6 个不同的函数组成. 因此,算法对这些分段的问题进行适宜度评价时, P1~P3 所需要的时间最少,其次为 P4~P7、P8 和 P10、P9,且在同一段问题中所消耗的计算资源相差不大. 综合分析可知,4 种 MFEA 算法在处理 WCCI2020SO 问题的平均运行时间从大到小排序为 $T(\text{EMaTO-MKT}) > T(\text{MaTSVM}) > T(\text{MaTEA}) > T(\text{MFEA})$.

表 5 MaTSVM 与 MFEA、MaTEA 和 EMaTO-MKT 在 WCCI20SO 问题运行的平均时间

问题	算法			
	MFEA	MaTEA	EMaTO-MKT	MaTSVM
P1	529.43	641.31	2426.73	856.97
P2	541.06	662.77	2478.19	870.08
P3	558.71	678.02	2520.68	910.42
P4	631.55	710.92	2528.40	942.08
P5	630.74	715.84	2527.16	1069.33
P6	635.97	721.49	2531.08	1031.99
P7	642.68	746.28	2550.84	1054.27
P8	681.15	801.12	2839.89	1469.23
P9	807.33	948.87	2964.45	1728.60
P10	695.64	829.51	2872.03	1487.92

4.5 算法组件分析

为了探讨 MaTSVM 中每种策略的影响,本节对 MaTSVM 的主要组成部分进行消融(Ablation Study)研究以证明这些策略提出的合理性. 由前面的分析可知, WCCI20SO 测试集比 CEC17 测试集更复杂,因此本节在 WCCI20SO 测试集进行实验.

我们对 MaTSVM 的两种简化形式进行了实验,它们分别是:用随机抽样代替预测相似的个体,记此形式为 MaTRM;用预测不相似的个体代替预测相似的个体,记此形式为 MaTRP. 参数设置与上述实验保持一致. 这三个简化形式均在每个问题上独立执行 25 次,非参数统计测试(具有 5% 显著水平的 Wilcoxon 符号秩检验)用来测试 MaTSVM 与 2 个简化形式之间的显著差异. 实验结果如表 6 所示.

表 6 MaTSVM 与 MaTRP、MaTRM 在 WCCI20SO 每个问题的所有任务中获得的最好解数目

问题	算法		
	MaTRP	MaTRM	MaTSVM
P1	0	0	50
P2	11	15	24
P3	5	15	30
P4	2	5	43
P5	1	13	36
P6	3	10	37
P7	5	8	37
P8	4	7	39
P9	3	8	39
P10	4	7	39

从表 6 的实验结果可知,与其它 2 种简化形式相比, MaTSVM 在 P1~P10 的各自 50 个任务中获得最好解的数目分别为: 50、24、30、43、36、37、37、39、39 和 39,除了在问题 P2 之外, MaTSVM 最好解数目所占的比例均超过一半,特别在 P1 上,获得最好解总数达到 100%,整体上以绝对的优势击败 MaTRP 与 MaTRM. 其原因分析如下:

(1) 对于 MaTRP,它在优化任务过程中辅助种群个体的选择由机器学习预测与目标任务不相似的个体构成,因此,目标任务与辅助种群存在更多的负向知识迁移,导致正向知识迁移的概率减少,从而阻碍任务间的优化进程.

(2) 对于 MaTRM,它的辅助种群通过随机选择个体组成,就相似或相近而言, MaTRM 辅助种群中的个体比 MaTSVM 辅助种群中的个体更远离目标任务个体,这使得在 MaTSVM 中有更多的正向知识迁移发生,加速优化任务的收敛速度.

由 4.4 节分析知道, P9 为在 P1~P10 中是最复杂的问题,其中,由函数 Rastrigin 与 Ackley 生成的任务更具有比较意义. 在图 7 中的 (a) P9 是所有由函数 Rastrigin 生成的任务平均适应度收敛曲线图, (b) P9 为所有由函数 Ackley 生成的任务平均适应度收敛曲线图. 图中显示 MaTSVM 在提出的策略帮助下对复杂任务的优化卓有成效,收敛速度更快,意味着获得的解精度更高.

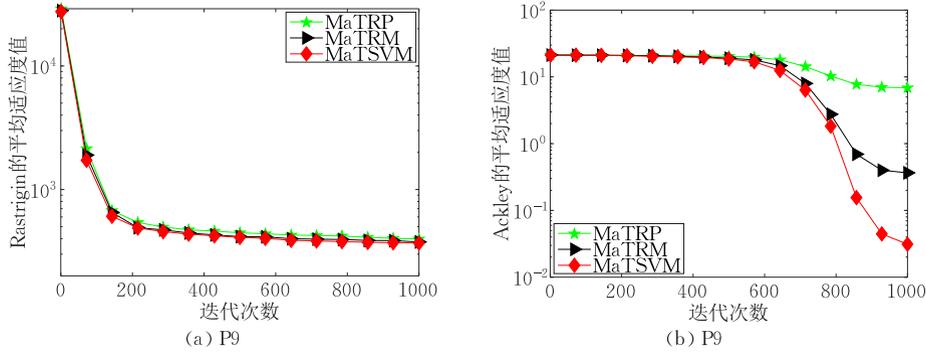


图7 MaTSVM与2个简化形式在P9中的收敛曲线图

4.6 模型性能分析

为了研究样本量与模型性能的关系,我们分别取种群 $N=3500$ 、 4500 、 5500 、 6500 、 7500 及 8500 测试模型的预测准确度和错误预测的个体对问题优化的影响. 由前面章节分析可知, WCCI2020SO 问题集比 CEC17 问题集更难优化, 而 P9 为 WCCI2020SO 问题集中最复杂的问题, 因而选取 P9 进行实验. 在参数 $C=20$ 和 $\sigma=0.05$ 前提下采用十折交叉法对每个取值均独立运行 25 次, 表 7 显示不同的 N 值在问题 P9 所有任务中获得的预测准确度、优化结果 $f1$ (预测为不属于目标任务的个体优化结果)、优化结果 $f2$ (预测为属于目标任务的个体优化结果) 以及 $f1-f2$ 差值, 所有数据都为执行 25 次的平均值 (保留 2 位有效小数).

表 7 不同的 N 值在 P9 中独立执行 25 次的调研结果

N	结果			
	准确度/%	优化结果($f1$)	优化结果($f2$)	优化差值
3500	77.16	$2.44e+03$	$2.39e+03$	$5.59e+01$
4500	79.92	$2.48e+03$	$2.38e+03$	$9.94e+01$
5500	83.38	$2.49e+03$	$2.36e+03$	$1.30e+02$
6500	85.12	$2.61e+03$	$2.28e+03$	$3.28e+02$
7500	88.71	$2.73e+03$	$2.13e+03$	$5.99e+02$
8500	89.93	$2.82e+03$	$2.10e+03$	$7.16e+02$

从表 7 看到, 随着样本 (种群) 数量增大, 模型的准确度有所提高, 同时注意到对应的优化结果 $f2$ 逐渐变小, 优化结果 $f1$ 逐渐变大, 它们之间的优化差值亦逐渐增大, 说明模型的拟合能力随样本增大而变强. 在辅助种群大小与目标任务种群大小一致的情况下 (算法 2 的第 14 行~24 行), 上述现象我们作出如下解释: (1) 当样本量较小时 (例如 $N=3500$ 、 4500), 预测为属于目标任务的个体组成辅助种群时, 由于此时模型的预测准确度相对较低, 从而造成较多的不属于目标任务的个体混进辅助种群. 同理, 若预测为不属于目标任务的个体组成辅助种群时, 即辅助种群中亦含有更多属于目标任务的个

体; (2) 当样本量较大时 (例如 $N=7500$ 、 8500), 模型的预测准确度相对提高, 这时模型对样本的预测能力更加精确, 预测为属于目标任务的个体组成辅助种群时, 因此会产生较少不属于目标任务的个体混进辅助种群. 另一方面, 若预测为不属于目标任务的个体组成辅助种群时, 即辅助种群中也含有更少属于目标任务的个体. 上面两点都在说明同一个问题, 即增加样本量使得模型对样本的预测产生更少错误预测的个体, 从而减少错误预测的个体混进辅助种群. 当错误预测的个体与正确预测的个体一起组成目标任务的辅助种群时, 由于演化多任务优化中的知识迁移具有两面性, 即正向的知识迁移能够促进优化进程, 而负向的知识迁移会阻碍优化进程, 因此, 随着样本数量越来越大, 辅助种群中包含错误预测的个体对优化结果的影响变小 (即负向知识迁移相对减少). 图 8(a)、图 8(b) 分别为取不同的样本量获得的优化结果 $f2$ 和优化结果 $f1$ 的收敛曲线图, $f1$ (图 8(b)) 随样本量增加而变大, 是因为正如上述分析那样辅助种群混进了更多的错误预测样本而造成了较多的负向知识迁移, 相反, $f2$ (图 8(a)) 是在辅助种群包含更少的错误预测的个体情况进行优化, 故优化效果越来越好. 因此, 随着样本量增加 $f1$ 与 $f2$ 的差值也会变大.

图 8(c) 中, 模型的准确度曲线图表明随着样本量的增加模型的准确度提升 (即模型性能变强). 同时留意到, 尽管不同的样本量导致模型的准确度不一样, 但在演化初始阶段, 模型获得的准确度均相对较低, 随着迭代次数越来越大, 模型的准确度也越来越高, 但到了后期阶段, 所有准确度的变化率趋向于水平状态. 出现上述现象我们作出如下解释: 由于在演化初期种群的所有个体具有较大的多样性, 导致模型对这些个体的预测具有更多的不确定性, 故准确度较低. 随着演化进行, 模型不断被更新以及越来越多的个体被正确划分到属于自己擅长的任务中,

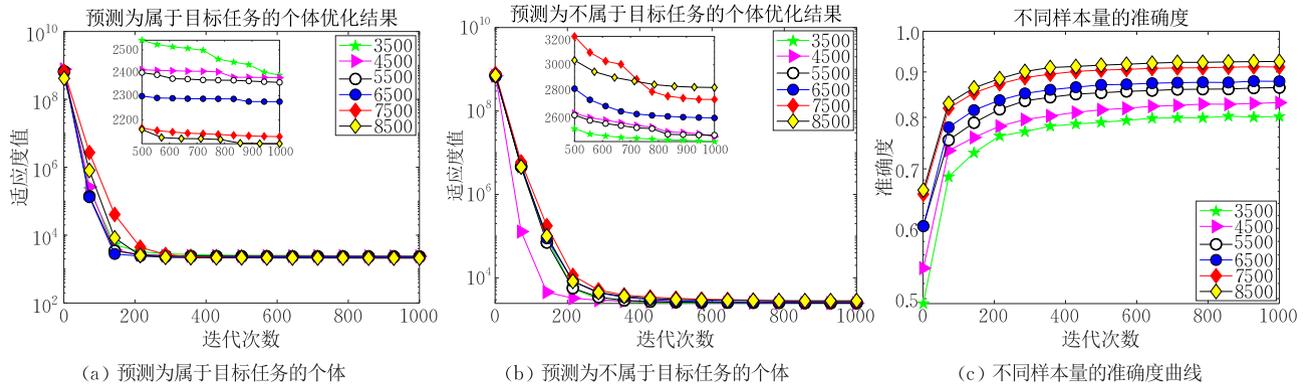


图 8 不同样本量在 P9 中的优化结果及对应的准确度

同时种群中的多样性也相对减少,此时训练出来的模型对某一任务的个体预测拥有更大的概率,因而准确度会提升.但在演化后期,当所有任务种群的个体被模型识别后,模型的准确度也到了极限,故而图 8(c)的曲线在末端时趋向水平状态.

综合上述分析,我们一方面考虑算法的运行时间,另一方面考虑模型复杂度及其准确度情况下,本文设置种群 $N=7500$.

4.7 模型稳定性分析

为了研究模型的稳定性和泛化能力,本文对影响 SVM 性能的主要参数进行敏感性分析,即惩罚参数 C 及核系数 σ . 惩罚参数 C 主要用于平衡模型的复杂度和误分率之间的关系,可以理解为正则化系数. C 取值过大时会容易造成模型过拟合,但较小的 C 也会导致模型性能降低.而核系数 σ 定义了单个样本对整个分类超平面的影响, σ 越大,模型越容易过拟合,反之模型会欠拟合.总之, C 与 σ 都会影响模型的稳定性和泛化能力.由前面章节分析可知,WCCI2020SO 问题集比 CEC17 问题集更难优化,而 P9 为 WCCI2020SO 问题集中最复杂的问题,因而选取 P9 进行实验.在种群 $N=7500$ 前提下,我们分别取 $C=5,7,10,15,20,25$ 与 $\sigma=0.01,0.03,0.05,0.07,0.09$ 进行网格调试,同时采用十折交叉法进行检验模型的样本外误分率,所有的参数对均独立运行 25 次.表 8 显示不

同的 C 与 σ 值在问题 P9 所有任务上获得的优化结果及其样本外误分率的平均值(每个网格),表中的最后两行为不同的 C 值对应列的适应度与误分率的平均值.从表 8 中观察得到, $C=20$ 且 $\sigma=0.05$ 在 30 个参数对中获得最好的结果,此时应适应度和样本外误分率分别为 $2.13e+03$ 、11.29%.此外 $C=20$ 与所有的 σ 组合($C=20, \sigma=0.01$ 、 $C=20, \sigma=0.03$ 、 $C=20, \sigma=0.05$ 、 $C=20, \sigma=0.07$ 、 $C=20, \sigma=0.09$)获得的适应度与误分率的分别为 $2.41e+03$ 、15.73%、 $2.30e+03$ 、13.51%、 $2.13e+03$ 、11.29%、 $2.18e+03$ 、12.48%、 $2.25e+03$ 、13.35%,且对应列的适应度与误分率的平均值分别为 $2.25e+03$ 、13.27%最低,总体上比 C 取其值与 σ 组合获得的结果更好,故 $C=20$ 时训练样本得到的模型的稳定性和泛化能力更佳.同时注意到 $C=25$ 和 $\sigma=0.05$ 、 $C=20$ 和 $\sigma=0.07$ 、 $C=5$ 和 $\sigma=0.03$ 获得模型的稳定性接近,这与 SVM 的调参规则^[45]相吻合,即同时增大 C 与 σ 可能获得更好模型,而小的 C 值一般对应着小的 σ .然而, $C=20$ 和 $\sigma=0.05$ 时样本外的误分率最低(11.29%),图 9 的收敛曲线图则进一步表明 $C=20$ 和 $\sigma=0.05$ 具有更出色的优化性能,说明此时模型的泛化能力更强.由于 C 取值过大会增加模型的复杂性,而过小的 σ 也会导致模型欠拟合,综合上述分析,本文推荐 $C=20$ 和 $\sigma=0.05$.

表 8 不同的 C 与 σ 独立执行 25 次的调研结果

C	Sigma/结果										不同 C 的 适应度	不同 C 的 误分率/%
	0.01		0.03		0.05		0.07		0.09			
	适应度	误分率/%	适应度	误分率/%	适应度	误分率/%	适应度	误分率/%	适应度	误分率/%		
5	2.28e+03	14.27	2.17e+03	13.05	2.38e+03	15.82	2.35e+03	15.77	2.70e+03	18.63	2.38e+03	15.50
7	2.32e+03	15.33	2.26e+03	14.09	2.30e+03	14.48	2.33e+03	15.42	2.36e+03	15.38	2.31e+03	14.94
10	2.75e+03	19.22	2.30e+03	14.87	2.32e+03	15.29	2.29e+03	15.36	2.35e+03	15.71	2.40e+03	16.09
15	2.62e+03	18.41	2.24e+03	13.58	2.37e+03	15.42	2.27e+03	14.18	2.26e+03	13.73	2.35e+03	15.06
20	2.41e+03	15.73	2.30e+03	13.51	2.13e+03	11.29	2.18e+03	12.48	2.25e+03	13.35	2.25e+03	13.27
25	2.60e+03	17.61	2.33e+03	14.92	2.18e+03	12.44	2.29e+03	14.30	2.34e+03	15.20	2.35e+03	14.89

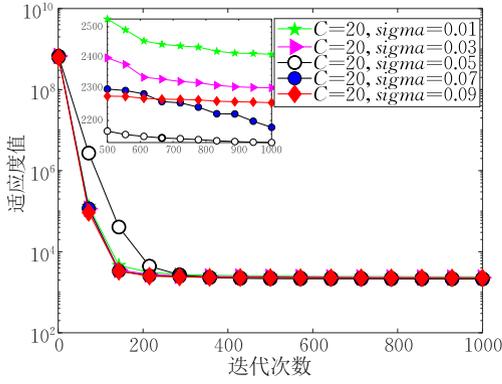
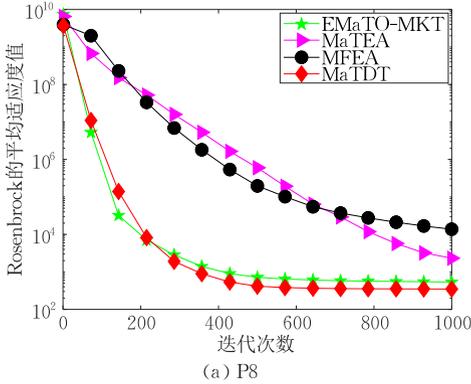


图 9 $C=20$ 与不同的 σ 组合在 P9 中的收敛曲线图

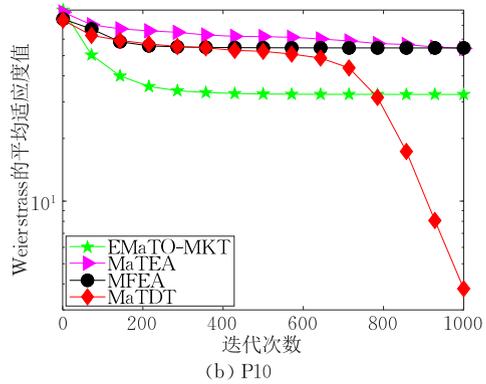
4.8 MaTDT

为了研究 MaTML 框架的通用性,我们将其与机器学习的另一种算法——决策树(Decision Tree, DT)进一步融合,简称为 MaTDT. 所比较的算法为 MFEA、MaTEA 及 EMaTO-MKT,所有的参数设置与 4.4 节保持一致. 同理,由于 WCCI20SO 测试集比 CEC17 测试集更复杂,故本节实验在 WCCI20SO 测试集上进行. 每种算法均在每个问题上独立执行 25 次,非参数统计测试(具有 5%显著水平的 Wilcoxon 符号秩检验)用来测试 MaTDT 与其它 3 种 MFEAs 之间的显著差异. 实验结果如表 9 所示.

从表 9 看到 MaTDT 在实验中获得最好解的总数与表 4 相比存在微小波动,例如,在相对容易处理的问题 P1~P2、P4 上相对于 MaTSVM 有很小的



(a) P8



(b) P10

图 10 MaTDT 与其它 3 种 MFEAs 在 P8、P10 中的收敛曲线图

4.9 传感器覆盖问题

传感器覆盖问题(Sensor Coverage Problem, SCP)由其线圈数与覆盖面积共同关联而成,理想的传感器模型在给定区域内提供 100%的覆盖,换句话说,传感器覆盖优化^[47]旨在以经济高效的方式最小化给定域的未覆盖区域. 在数学上,传感器覆盖问题可通过式(10)进行描述.

表 9 MaTDT 与 MFEA、MaTEA、EMaTO-MKT 在 WCCI20SO 每个问题的所有任务中获得的最好解数目

问题	算法			
	MFEA	MaTEA	EMaTO-MKT	MaTDT
P1	0	0	48	2
P2	0	0	19	31
P3	0	0	50	0
P4	0	0	8	42
P5	0	0	17	33
P6	3	0	18	29
P7	0	0	17	33
P8	0	0	16	34
P9	4	1	15	30
P10	2	0	20	28

下降,但在复杂的问题 P8~P10 上几乎没有变动,特别是在最复杂的问题 P9 中, MaTDT 得到的结果与 MaTSVM 保持一致,这充分体现了 MaTML 框架的易拓展性及稳定性. 另一方面也反映了基于 MaTML 架构实现的 MFEAs 更有助于正向知识转移在目标任务与辅助任务之间进行,同时更有效地避免负向知识转移,这完全归功于 MaTML 自身的策略.

鉴于 P8、P10 为复杂问题,图 10 分别绘出所有基于函数 Rosenbrock(P8)与函数 Weierstrass(P10)生成的任务的平均适应度收敛过程. 两幅收敛曲线图都表明 MaTDT 在处理复杂任务上比其它 3 种 MFEAs 效果更佳,再次验证基于 MaTML 框架实现的演化多任务算法具有很好的稳定性及高效性.

$$\min f(x) = 1000 \cdot \left[1 - \frac{A \cap \bigcup_{i=1}^K \pi r_i^2(x_i, y_i)}{A} \right] + \sum_{i=1}^K (1 + 10 \cdot r_i^2) \quad (10)$$

subject to $-1 < x_i < 1, i = 1, 2, \dots, K$
 $-1 < y_i < 1, i = 1, 2, \dots, K$
 $0.1 < r_i < 0.25, i = 1, 2, \dots, K$

其中, $x = (x_1, y_1, r_1, \dots, x_k, y_k, r_k)$ 决定了传感器在区域中的位置, x_i, y_i, r_i 分别表示传感器的横坐标、纵坐标及半径大小, A 代表要覆盖的区域, 且 $A \in [-1, 1] \times [-1, 1]$, K 为传感器数. 式(10)的意义在于在给定区域前提下, 寻找最佳的传感器数目以尽可能地覆盖 A , 即最小化目标函数 $f(x)$.

已有许多技术用来优化无线传感器网络覆盖问题, 如文献[48-49], 但它们每次都是以固定传感器数目的方式通过多次尝试来找到最佳的 K 值, 这需要花费大量的时间才能得到一个合适的解. 最近, Li 等人^[50] 提出一种竞争性多任务算法 (DEORA), 它的解为所有任务解中的最小值, 并以此方式来解决传感器覆盖问题. 本文基于 MaTML 框架下对传感器覆盖问题进行优化, 通过一次运行获得目标函数的最优解.

由于最优传感器数未知, 我们考虑 7 个待优化的任务, 每个任务都是基于式(10)衍生得到, 它们的 K 值分别设置为 26、27、28、29、30、31 和 32, 目的是通过 MaTSVM 同时对这 7 个任务进行最小优化来求解 SCP. 通过 4.4 节的分析考虑可知, 上述 4 种 MFEA 算法的性能大小排名为 $\text{MaTSVM} > \text{EMaTO-MKT} > \text{MaTDE} > \text{MFEA}$, 且 EMaTO-MKT 生成后代个体是通过 GA 优化器 (Genetic Algorithm) 触发得到,

而 MaTSVM、MaTDE 以及 DEORA 均是通过 DE 优化器 (Differential Evolution Algorithm) 触发得到, 因此我们在实验中加入 EMaTO-MKT 进行比较. MaTSVM、EMaTO-MKT 的参数设置与上述保持一致, DEORA 参数设置参照文献[50], 所有的算法都独立执行 25 次, 最大迭代次数设置为 $Gmax = 10\,000$. 表 10 列出了三种算法在实验中的最优值及对应的传感器数目.

表 10 MaTSVM 与 DEORA、EMaTO-MKT 在 SCP 问题的实验结果

算法	最好值	传感器数
DEORA	51.84	31
EMaTO-MKT	49.97	30
MaTSVM	48.49	30

从表 10 看到 MaTSVM、EMaTO-MKT 及 DEORA 优化目标函数得到的最小值分别为 48.49、49.97 和 51.84, 相应的传感器数目分别为 30、30 和 31, 图 11 (a)、(b) 及 (c) 分别为三种算法覆盖指定区域所需要的传感器数量. 表 10 和图 11 表明, 与 EMaTO-MKT 及 DEORA 相比, MaTSVM 需要更少的传感器数即可覆盖指定区域, 且获得最好的值. 更多的是, 从图 11(d) 收敛曲线趋势看到, 在大约 7000 代之前, EMaTO-MKT 的收敛速度最快, 这可能是因为使用了 GA 作为优化器而导致不同的搜索路径, 从

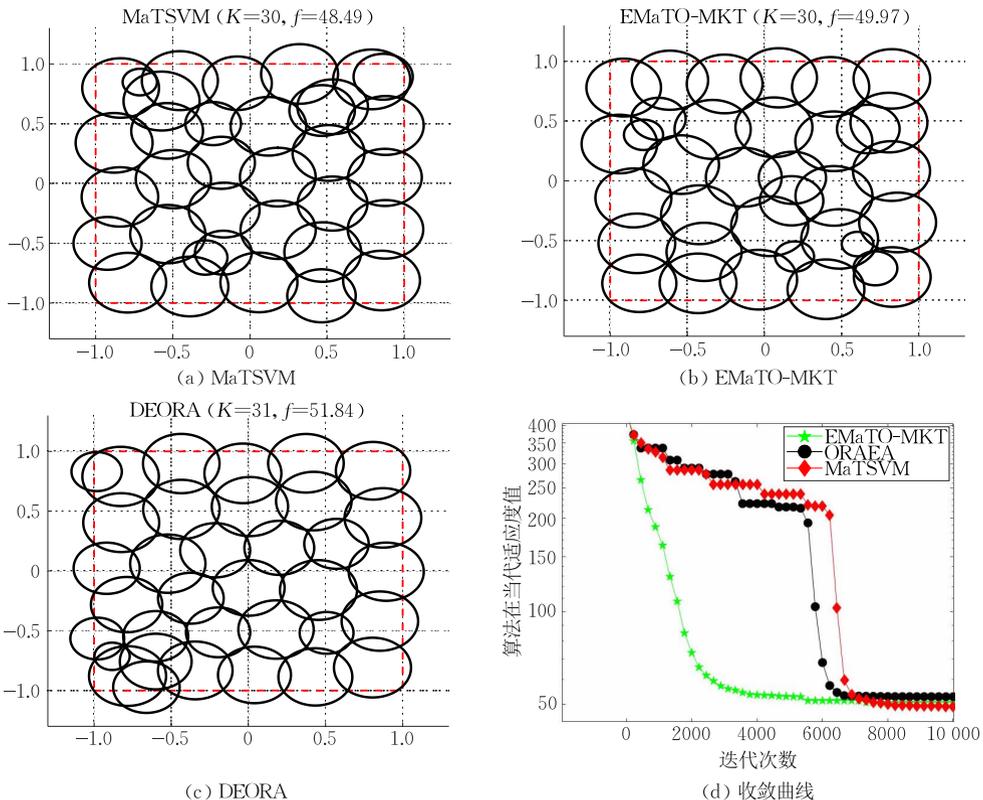


图 11 MaTSVM 与 EMaTO-MKT、DEORA 获得的传感器最佳覆盖数及其收敛曲线

而在这段迭代中搜索到问题的解的质量更好,但在 7000 代之后, MaTSVM 超越了 EMaTO-MKT, 直到 10000 代获得了问题的最好解, 验证了 MaTSVM 具有更好的收敛性能。

5 总结与展望

本文提出了一个基于机器学习的演化多任务优化框架 MaTML, 其核心是利用目标任务的技能因子及对应的种群个体分别构建标签和训练集, 并基于机器学习算法训练模型, 应用模型预测个体组成辅助种群。提出的框架不仅可以在任务的优化中动态地找到目标任务的辅助种群, 而且可以在优化中实现目标任务与辅助任务直接交流, 提高任务间正向知识转移效率。进一步, 文中将 SVM 集成到框架中形成 MaTSVM 算法, 通过在 CEC2017 问题测试集和 WCCI2020SO 问题测试集与相应的现阶段 MFEAs 算法比较, 实验结果证实 MaTSVM 在优化多任务问题时具有优越或竞争性的性能。此外, MaTSVM 也用于优化传感器覆盖的真实问题, 实验结果与比较分析同样检验了 MaTSVM 的有效性。

在将来的工作中存在几个方面的研究, 一方面, 当任务的维度增大及问题复杂性加大时, 设计更高效的演化多任务算法。另一方面考虑将人工智能领域中的深度学习技术融合到框架中建立新型演化多任务架构。最后, 将所提出的框架应用于优化分类、离散的问题也是未来重点的研究课题。

参 考 文 献

- [1] Back T, Hammel T, Schwefel H P. Evolutionary computation: Comments on the history and current state. *IEEE Transactions on Evolutionary Computation*, 1997, 1(1): 3-17
- [2] Pourhassan M, Neumann F. Theoretical analysis of local search and simple evolutionary algorithms for the generalized travelling salesperson problem. *Evolutionary Computation*, 2018, 27(3): 525-558
- [3] Tanabe R, Ishibuchi H. A review of evolutionary multimodal multiobjective optimization. *IEEE Transactions on Evolutionary Computation*, 2020, 24(1): 193-200
- [4] Tian Y, Zheng X T, Zhang X Y, et al. Efficient large-scale multiobjective optimization based on a competitive swarm optimizer. *IEEE Transactions on Cybernetics*, 2020, 50(8): 3696-3708
- [5] Qi S, Zou J, Yang S X, et al. A self-exploratory competitive swarm optimization algorithm for large-scale multiobjective optimization. *Information Sciences*, 2022, 609(2): 1601-1620
- [6] Cai X Y, Sun H R, Zhang Q F, et al. A grid weighted sum pareto local search for combinatorial multi and many-objective optimization. *IEEE Transactions on Cybernetics*, 2019, 49(9): 3586-3598
- [7] Gu Q H, Wang Q, Li X X, et al. A surrogate-assisted multi-objective particle swarm optimization of expensive constrained combinatorial optimization problems. *Knowledge-Based Systems*, 2021, 223(1): 107049
- [8] Wang Feng, Zhang Heng, Han Meng-Chen, et al. Co-evolution based mixed-variable multi-objective particle swarm optimization for UAV cooperative multi-task allocation problem. *Chinese Journal of Computers*, 2021, 44(10): 1967-1983(in Chinese) (王峰, 张衡, 韩孟臣等. 基于协同进化的混合变量多目标粒子群优化算法求解无人机协同多任务分配问题. *计算机学报*, 2021, 44(10): 1967-1983)
- [9] Dang Q L. Multiple dynamic penalties based on decomposition for constrained optimization. *Expert Systems with Applications*, 2022, 206(3): 117820
- [10] Wang B C, Li H X, Zhang Q F, et al. Decomposition-based multiobjective optimization for constrained evolutionary optimization. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2021, 51(1): 574-587
- [11] Gupta A, Ong Y S, Feng L. Multifactorial evolution: Toward evolutionary multitasking. *IEEE Transactions on Evolutionary Computation*, 2016, 20(3): 343-357
- [12] Ong Y S, Gupta A. Evolutionary multitasking: A computer science view of cognitive multitasking. *Cognitive Computation*, 2016, 8(2): 125-142
- [13] Gupta A, Ong Y S, Feng L. Insights on transfer optimization: Because experience is the best teacher. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 2017, 2(1): 51-64
- [14] Gupta A, Zhou L, Ong Y S, et al. Half a dozen real-world applications of evolutionary multitasking, and more. *IEEE Computational Intelligence Magazine*, 2022, 17(2): 49-66
- [15] Feng L, Huang Y X, Zhou L, et al. Explicit evolutionary multitasking for combinatorial optimization: A case study on capacitated vehicle routing problem. *IEEE Transactions on Cybernetics*, 2021, 51(6): 3143-3156
- [16] Bali K K, Gupta A, Feng L, et al. Linearized domain adaptation in evolutionary multitasking//*Proceedings of the IEEE Congress on Evolutionary Computation*. Donostia, Spain, 2017: 1295-1302
- [17] Zhou L, Feng L, Tan K C, et al. Toward adaptive knowledge transfer in multifactorial evolutionary computation. *IEEE Transactions on Cybernetics*, 2021, 51(5): 2563-2576
- [18] Gupta A, Ong Y S, Feng L, et al. Multiobjective multifactorial optimization in evolutionary multitasking. *IEEE Transactions on Cybernetics*, 2017, 47(7): 1652-1665
- [19] Chen Y L, Zhong J H, Feng L, et al. An adaptive archive-based evolutionary framework for many-task optimization. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 2020, 4(3): 369-384
- [20] Liang Z P, Xu X J, Liu L, et al. Evolutionary many-task optimization based on multisource knowledge transfer. *IEEE*

- Transactions on Evolutionary Computation, 2022, 26(2): 319-333
- [21] Bernardini M, Romeo L, Frontoni E, et al. A semi-supervised multi-task learning approach for predicting short-term kidney disease evolution. *IEEE Journal of Biomedical and Health Informatics*, 2021, 25(10): 3983-3994
- [22] Li Y M, Yang M, Qi Z G, et al. Bayesian multi-task relationship learning with link structure. *IEEE Transactions on Knowledge and Data Engineering*, 2016, 28(4): 873-887
- [23] Da B S, Ong Y S, Feng L, et al. Evolutionary multitasking for single-objective continuous optimization: Benchmark problems, performance metric, and baseline results. Nanyang Technological University, Singapore. <https://arxiv.org/abs/1706.03470>[doi: 10.48550/arXiv.1706.03470], 2017
- [24] Feng L, Qin K, Gupta A, et al. IEEE WCCI2020 Competition on Evolutionary Multi-task Optimization. http://www.bdsc.site/websites/MTO_competition_2020/MTO_Competition_WCCI_2020.html, 2020
- [25] Li Hao, Wang Lei, Zhang Yuan-Qiao, et al. Survey of evolutionary multitasking optimization. *Journal of Software*, 2023, 34(2): 509-538(in Chinese)
(李豪, 汪磊, 张元侨等. 演化多任务优化研究综述. *软件学报*, 2023, 34(2): 509-538)
- [26] Feng L, Zhou W, Zhou L, et al. An empirical study of multifactorial PSO and multifactorial DE//Proceedings of the IEEE Congress on Evolutionary Computation. Donostia, Spain, 2017: 921-928
- [27] Bali K K, Ong Y S, Gupta A, et al. Multifactorial evolutionary algorithm with online transfer parameter estimation: MFEA-II. *IEEE Transactions on Evolutionary Computation*, 2020, 24(1): 69-83
- [28] Feng L, Zhou L, Zhong J H, et al. Evolutionary multitasking via explicit autoencoding. *IEEE Transactions on Cybernetics*, 2019, 49(9): 3457-3470
- [29] Zhong J H, Feng L, Cai W T, et al. Multifactorial genetic programming for symbolic regression problems. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2020, 50(11): 4492-4505
- [30] Jin Y C, Wang H D, Chugh T, et al. Data-driven evolutionary optimization: An overview and case studies. *IEEE Transactions on Evolutionary Computation*, 2019, 23(3): 442-458
- [31] Yang C E, Ding J L, Jin Y C, et al. Offline data-driven multiobjective optimization: Knowledge transfer between surrogates and generation of final solutions. *IEEE Transactions on Evolutionary Computation*, 2020, 24(3): 409-423
- [32] Li K, Chen R Z. Batched data-driven evolutionary multiobjective optimization based on manifold interpolation. *IEEE Transactions on Evolutionary Computation*, 2023, 27(1): 126-140
- [33] Zhen H X, Gong W Y, Wang L. Two-stage data-driven evolutionary optimization for high-dimensional expensive problems. *IEEE Transactions on Cybernetics*, 2023, 53(4): 2368-2379
- [34] Gupta A, Mandziuk J, Ong Y S. Evolutionary multitasking in bi-level optimization. *Complex & Intelligent Systems*, 2015, 1(1-4): 83-95
- [35] Chandra R, Gupta A, Ong Y S, et al. Evolutionary multi-task learning for modular training of feedforward neural networks//Proceedings of the International Conference on Neural Information Processing. Kyoto, Japan, 2016: 37-46
- [36] Zhou L, Feng L, Gupta A, et al. Learnable evolutionary search across heterogeneous problems via kernelized autoencoding. *IEEE Transactions on Evolutionary Computation*, 2021, 25(3): 567-581
- [37] Li H, Ong Y S, Gong M, et al. Evolutionary multitasking sparse reconstruction: Framework and case study. *IEEE Transactions on Evolutionary Computation*, 2018, 23(5): 733-747
- [38] Gong M G, Tang Z G, Li H, et al. Evolutionary multitasking with dynamic resource allocating strategy. *IEEE Transactions on Evolutionary Computation*, 2019, 23(5): 858-869
- [39] Wei T Y, Wang S B, Zhong J H, et al. A review on evolutionary multitask optimization: Trends and challenges. *IEEE Transactions on Evolutionary Computation*, 2022, 26(5): 941-960
- [40] Liaw R T, Ting C K. Evolutionary many-tasking based on biocoenosis through symbiosis: A framework and benchmark problems//Proceedings of the IEEE Congress on Evolutionary Computation. Donostia, Spain, 2017: 2266-2273
- [41] Huang S J, Zhong J H, Yu W J. Surrogate-assisted evolutionary framework with adaptive knowledge transfer for multi-task optimization. *IEEE Transactions on Emerging Topics in Computing*, 2021, 9(4): 1930-1944
- [42] Zhang Y, Yang Q. A survey on multi-task learning. *IEEE Transactions on Knowledge and Data Engineering*, 2022, 34(12): 5586-5609
- [43] Schlkopf B, Platt J C, Taylor J S, et al. Estimating the support of a high-dimensional distribution. *Neural Computation*, 2001, 13(7): 1443-1471
- [44] Fan R E, Chen P H, Lin C J. Working set selection using second order information for training support vector machines. *Journal of Machine Learning Research*, 2005, 6: 1889-1918
- [45] Burges C J C. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 1998, 2(2): 121-167
- [46] Li Y C, Gong W Y, Li S J. Multitasking optimization via an adaptive solver multitasking evolutionary framework. *Information Sciences*, 2023, 630(1): 688-712
- [47] Ryerkerk M L, Averill R C, Deb K, et al. Solving metameric variable-length optimization problems using genetic algorithms. *Genetic Programming & Evolvable Machines*, 2017, 18(2): 247-277
- [48] Ting C K, Lee C N, Chang H C, et al. Wireless heterogeneous transmitter placement using multiobjective variable-length genetic algorithm. *IEEE Transactions on Systems, Man, and Cybernetics*, 2009, 39(4): 945-948
- [49] Wu C H, Lee K C, Chung Y C. A Delaunay triangulation based method for wireless sensor network deployment. *Computer Communications*, 2007, 30(14-15): 2744-2752
- [50] Li G H, Zhang Q F, Wang Z K. Evolutionary competitive multitasking optimization. *IEEE Transactions on Evolutionary Computation*, 2022, 26(2): 278-289



MAI Wei-Jie, Ph.D. candidate. His research interests include evolutionary computation and machine learning.

LIU Wei-Li, Ph.D., lecturer. Her research interests include evolutionary computation and swarm intelligence.

ZHONG Jing-Hui, Ph.D., professor. His research interests include evolutionary computation, machine learning, and multi-agent modeling and simulation.

Background

Evolutionary multitasking optimization belongs to the research branch of computational intelligence, aims to solve two or more optimization tasks simultaneously by leveraging intertask knowledge transfer. In evolutionary multitasking optimization, finding auxiliary tasks for target tasks is crucial, which has become a research hotspot for many scholars. However, as we have pointed out, although existing literature adopts different similarity criteria to search for auxiliary tasks, their knowledge transfer between tasks is limited by pre-set subpopulation spaces during iteration, resulting in a loss of flexibility and dynamism in knowledge transfer between tasks. We believe that knowledge transfer between tasks should have three characteristics: dynamic, flexible and efficient. Some criteria are used to roughly estimate the similarity between two tasks with limited space, while other individuals in the population that are really useful for task optimization are ignored. In this way, inefficient knowledge transfer will occur in the process of task optimization and lead to the waste of computing resources. This paper proposes an evolutionary multitasking framework (MaTML) based on machine learning to address the above shortcomings. The

main contributions include: (1) a novel evolutionary multitasking optimization framework based on machine learning is proposed. The main features of the framework can dynamically train the model and predict the auxiliary task individuals related to the target task; (2) the framework only requires a large number of individuals to form a unified population, and the selection of population individuals for the auxiliary tasks of the target task is variable and dynamically adaptive. There is no need to associate a subpopulation with the target task, which solves the limitation of flexibility in knowledge transfer between tasks due to the use of a subpopulation of the same size associated with each task in the design. The experimental results on the two representative test problems and practical engineering problem show that, compared with state-of-the-art multi-task algorithms and many-task algorithms respectively, MaTML has superior or competitive performance on convergence efficiency and accuracy of the solution.

This work is supported by the National Natural Science Foundation of China (No. 62076098), and the Guangdong Basic and Applied Basic Research Foundation (Nos. 2021A1515110072 and 2023A1515012291).