

基于自适应分解的多任务协作型昂贵多目标优化算法

蔡昕焯¹⁾ 马中雨¹⁾ 张 峰¹⁾ 李 楠¹⁾ 程会林¹⁾ 孙 祺¹⁾
肖禹舜¹⁾ 李小平²⁾

¹⁾(南京航空航天大学计算机科学与技术学院 南京 211106)

²⁾(东南大学计算机科学与工程学院 南京 211189)

摘 要 现实世界的工程优化问题通常需要同时优化多个冲突的目标,且这些目标函数的评估由于依赖仿真、物理实验而十分昂贵,这类问题被称为昂贵多目标优化问题.使用机器学习方法建立代理模型用于估计候选解的目标函数值是求解此类问题的一种有效手段.高斯代理模型适用于训练样本数较少的中小规模问题,且能提供评估的不确定性,因此常作为代理模型被应用于昂贵优化.分解是处理多目标优化问题的一种有效手段.一个多目标优化问题可被分解为多个单目标优化子问题,且多个子问题可被进一步划分为代理模型学习的一个目标任务.现有基于分解的昂贵多目标优化算法大多将固定数量的子问题静态地划分到同一任务,从而构造多个固定任务并对其建立多任务高斯代理模型进行求解.这未能充分利用数据的相关信息动态反映出任务间的相关性,限制了多任务高斯过程模型的预测精度以及优化算法的最终性能.为此,本文提出了一种自适应多任务多种群协作搜索算法(AMMCS).AMMCS使用相似性指标实时度量已评估的解集,获得子问题间的相关性,从而自适应地划分任务,提升多任务模型的预测质量.此外,AMMCS使用一个解集(种群)优化一个任务,并通过多种群的协作搜索实现多任务高斯模型的批量优化,提高了采样效率,提升了算法的收敛效率.通过 AMMCS 与六个代理辅助进化算法进行多组实验对比和分析,显示了 AMMCS 具有良好的性能.我们同时也设计实验验证了算法中自适应分解以及多种群协作搜索的有效性.

关键词 代理辅助进化算法;昂贵优化;多目标优化;多任务高斯过程模型;多种群协作搜索

中图分类号 TP18 DOI号 10.11897/SP.J.1016.2021.01934

Adaptive Multitask with Multipopulation-Based Cooperative Search for Expensive Multiobjective Optimization Problems

CAI Xin-Ye¹⁾ MA Zhong-Yu¹⁾ ZHANG Feng¹⁾ LI Nan¹⁾ CHENG Hui-Lin¹⁾ SUN Qi¹⁾
XIAO Yu-Shun¹⁾ LI Xiao-Ping²⁾

¹⁾(College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing 211106)

²⁾(School of Computer Science and Engineering, Southeast University, Nanjing 211189)

Abstract Real-world engineering optimization problems usually need to optimize multiple conflicting objectives at the same time, and the objective function evaluation is very expensive because it depends on simulation and physical experiments. They need to be solved within the minimum number of objective function evaluation times. This type of problem is called expensive

收稿日期:2020-09-08;在线发布日期:2021-03-23. 本课题得到国家自然科学基金(62072234,61300159,61732006,61876075,61832004)、江苏省自然科学基金(BK20181288)、中国博士后基金(2015M571751)、航空科学基金会(20175552042)资助.蔡昕焯,博士,副教授,中国计算机学会(CCF)会员,主要研究方向为优化、机器学习及其应用. E-mail: xinye@nuaa.edu.cn. 马中雨,硕士研究生,主要研究方向为 DIMA 建模与算法优化. 张 峰,硕士研究生,主要研究方向为昂贵多目标优化和机器学习. 李 楠,硕士研究生,主要研究方向为昂贵多目标优化和机器学习. 程会林,硕士研究生,主要研究方向为对抗样本和优化. 孙 祺,硕士研究生,主要研究方向为多目标优化和双层优化. 肖禹舜,硕士研究生,主要研究方向为多目标优化度量指标和机器学习. 李小平,博士,教授,中国计算机学会(CCF)会员,主要研究领域为云计算、资源调度优化、服务计算、大数据、云制造、复杂生产调度优化.

multiobjective optimization problem. Using machine learning to build a surrogate model to assist algorithms in evaluating candidate solutions is an effective way to solve such problems. Among all of the surrogate models, the Gaussian process model can handle small and medium-scale problems with a small number of training samples, not to mention its ability to provide the uncertainty estimation for each predicted point. Decomposition is an effective way to handle multiobjective optimization problems. A multiobjective optimization problem can be decomposed into a number of single objective subproblems. Several subproblems can be further divided into a task targeted by training the surrogate model. However the existing expensive multiobjective optimization based on decomposition usually divides a fixed number of subproblems into a same task. By constructing multiple fixed tasks, a multi-task Gaussian surrogate model is establish for the expensive multiobjective optimization problems. However, this method fails to make full use of the data to dynamically reflect the correlation between tasks, which limits the prediction accuracy of the multi-task Gaussian process model and the final performance of the optimization algorithm. Therefore, we propose an adaptive multitask with multipopulation-based cooperative search algorithm (AMMCS) to handle expensive multiobjective optimization problems in this paper. The Pearson coefficient is used to measure the evaluated solution set for obtaining the correlation between subproblems, and adaptively assign subproblems into the related tasks for improving the quality of model prediction. In addition, as it is difficult to use one population for optimizing multiple related tasks at the same time, AMMCS adopts one solution set (subpopulation) for one task. The batched optimization of the multi-task Gaussian process model is accomplished through the collaborative search of multiple subpopulations for high efficiency. We compare the performance of AMMCS with six surrogate-assisted evolutionary algorithms, including GCS-MOE, MOEA/D-EGO, K-RVEA, ParEGO, CSEA and SMS-EGO. Among all of the compared algorithms, AMMCS is significantly better than others on most selected test problems. In addition, we design experiments to verify the effectiveness of adaptive decomposition and multi-group collaborative search. The sensitivity test of the task number parameter Q used in AMMCS has also been conducted for selecting its optimal value. The experimental result shows the algorithm achieves the best performance when the value of Q is 5. The effects of different utility functions on the performance of AMMCS have also been verified. The experimental results show the hybrid utility functions with both UBC and EI are able to achieve best performance. The lack of ability for handling irregular Pareto fronts have also been observed in the experimental studies. Therefore, one of the future works is the considerations of adding direction vectors adaptations into AMMCS for further enhancing its performance on expensive multiobjective optimization problems with irregular Pareto fronts.

Keywords surrogate-assisted evolutionary algorithm; expensive multiobjective optimization; multiobjective optimization problem; multitask Gaussian process model; multi-population collaborative search

1 引言

现实工程优化问题经常存在多个互相冲突的优化目标,它们通常被称为多目标优化问题^[1-3]

(Multiobjective Optimization Problem, MOP). 近年来多目标进化算法(Multiobjective Evolutionary Algorithm, MOEA)成为了求解 MOPs 的主流方法.

此外,许多工程优化问题在评估目标函数时需要消耗大量计算时间,或其通过物理实验进行评

估的代价十分昂贵,这类多目标优化问题被称为昂贵多目标优化问题^[4] (Expensive Multiobjective Optimization Problem, EMOP).

如何在尽可能少的函数评估次数下快速收敛得到满意的解,是设计算法求解 EMOPs 的关键. 使用机器学习技术建立代理模型进行辅助评估,成为求解昂贵优化问题的有效手段. 这些代理模型包括高斯过程^[5]、神经网络^[6]和随机森林^[7]等. 其中,由于高斯代理模型能够很好地处理训练样本数较少的中小规模问题,并且能够为每个预测点提供模型的不确定性估计^[8-9],因此作为代理模型被广泛应用于昂贵优化问题.

在过去的十多年里,已涌现了一批代理模型辅助的多目标进化算法(Surrogate-Assisted Evolutionary Algorithm, SAEA),用于求解昂贵多目标优化问题. 通常,使用聚合函数可将目标空间进行有效分解,即将一个多目标优化问题分解为多个单目标优化子问题,此类典型算法是 MOEA/D^[10]. 多个子问题间可建立多任务高斯过程代理模型,任务间有效的信息迁移可提高优化的收敛效率^[11].

然而,目前此类算法存在两点不足:一是基于多任务高斯过程的 SAEA 通常通过预设参数来指定每个任务包含的子问题数量,并根据子问题权重的欧式距离来指定子问题的邻域,进而构造多个相关的优化任务. 由于子问题间的相关性结构在优化过程中是动态变化的,这种静态任务划分方法未能实时地反应出任务间的相关性,并极可能影响多任务高斯过程模型的预测精度和最终的优化算法性能;二是目前大多代理辅助多目标进化算法均使用单点采样,然而如使用批量采样评估可大量节省计算时间^[12].

因此,本文提出了一种基于自适应分解的多任务协作型昂贵多目标优化算法(AMMCS). 主要贡献总结如下:

AMMCS 通过使用优化过程中产生的数据实时地度量子问题间的相关性,从而实现多任务的自适应划分,进而提高了多任务高斯模型预测精度.

AMMCS 使用一个解集(种群)优化一个任务,并通过多种群的协作搜索实现多任务高斯模型的批量优化,提高了采样效率,提升了算法的收敛效率.

与六个代理辅助进化算法进行多组实验对比和分析,显示了其具有良好的性能. 另外,我们设计实验验证了算法中自适应分解以及多种群协作搜索

的有效性.

本文第 2 节概括 EMOP 的相关工作;第 3 节介绍算法的实现细节;第 4 节展示算法的实验设计;第 5 节对提出的 AMMCS 进行验证、对比与分析;第 6 节对本文进行总结.

2 相关工作

由于在对非线性问题近似时,高斯过程模型表现出较高精度和鲁棒性,并且能够同时预测待测点的目标值和相应的方差,因此被广泛应用于 SAEAs. 例如,Knowles 提出了一个基于聚合函数的经典代理辅助进化算法 ParEGO^[13],其主要思想是随机选择一个聚合函数子问题并使用高斯过程来近似聚合函数,然后通过优化效用准则函数来获取下一个候选解进行评估. Ponweiser 等人提出了 SMS-EGO^[14],SMS-EGO 使用高斯过程来建立相应的代理模型,然后通过协方差矩阵自适应进化策略优化解集的超体积指标,决定下一个待评估的候选解. 而刘建昌等人根据高斯过程的输出,计算个体的基于 R2 指标的效用函数来选择评估点,能够保持种群个体的收敛性和多样性^[15-16]. Zhang 等人在 MOEA/D 的基础上拓展提出了用于求解昂贵多目标优化的 MOEA/D-EGO^[17],对每个子问题建立了一个高斯过程模型,并通过 MOEA/D-DE^[18] 优化最大改进效用准则,获取一个待评估的候选解集. Chugh 等人提出了 K-RVEA^[9],在 RVEA 算法的基础上采用高斯模型为代理模型,根据近似目标值的不确定性信息来保持解多样性和收敛性,并提出了一种有效的模型管理方案,以处理多个目标. 为了进一步解决昂贵的超多目标优化问题, Pan 和 He 等人在使用神经网络来建立代理模型的基础上提出了一个代理辅助进化算法 CSEA^[20] 来解决问题, CSEA 根据神经网络预测的候选解与参考解的支配关系来挑选相应的候选解进行评估. 随着多任务学习领域的发展,多种多任务高斯过程模型^[21-25] 被相继提出. 近年也出现了使用多任务高斯过程求解昂贵多目标优化的相关工作. 例如, Luo 等人提出的 GCS-MOE^[26] 是一种基于多任务高斯过程的代理辅助进化算法, GCS-MOE 在 MOEA/D 框架的基础上提出了任务划分方法及其邻居任务的定义,将对子问题的优化转为对任务的优化,并使用多任务高斯过程对多个邻居任务的聚合函数进行预测来提升模型

的预测质量.

由于高斯过程模型的计算复杂度为 $O(N^3)$, 计算量随着训练样本的增多而急剧增长, 其并不适合求解中大规模的昂贵多目标优化问题. 针对此类问题, 神经网络、随机森林等非高斯过程的机器学习技术被用于建立代理模型. 例如, Lin 等人在 MOEA/D 基础上结合贝叶斯神经网络提出了 BS-MOBO^[12] 来处理不同类型的大规模优化问题. 除了连续的昂贵多目标优化算法外, Wang 等人提出的 RF-CMOCO (FS)^[27] 开始使用随机森林和径向基函数建立代理模型分别近似目标函数和约束函数来解决组合的昂贵多目标优化问题.

3 背景介绍

3.1 昂贵多目标优化问题

一个连续的最小化 MOP 可定义为如下:

$$\begin{aligned} \text{minimize } F(\mathbf{x}) &= (f_1(\mathbf{x}), \dots, f_m(\mathbf{x}))^T \in R^m \\ \text{subject to } \mathbf{x} &\in \Omega \end{aligned} \quad (1)$$

其中 $\Omega \subset R^n$ 是决策空间, \mathbf{x} 是一个有着 n 个决策变量的解向量, m 是目标个数. $F(\mathbf{x}): \Omega \rightarrow R^m$ 表示解 \mathbf{x} 到 m 维目标函数空间 R^m 的投影. 当其函数评估代价昂贵时, 该问题被称为昂贵多目标优化问题.

帕累托支配 (Pareto dominance)^[28]. x^a 和 x^b 是决策空间 Ω 中的两个解, 定义 x^a 帕累托 (Pareto) 支配 x^b 并记为 $x^a < x^b$, 成立的条件是当且仅当对 $\forall i \in \{1, \dots, m\}$, 都有 $f_i(x^a) \leq f_i(x^b)$ 并且 $\exists j \in \{1, \dots, m\}$, $f_j(x^a) < f_j(x^b)$.

帕累托最优解 (Pareto optimal solution). 若不存在一个解 $\mathbf{x} \in \Omega$, 使得 $\mathbf{x} < \mathbf{x}^*$. 则 \mathbf{x}^* 就被称为帕累托最优解, 全部 Pareto 最优解构成了 Pareto 最优解集 (Pareto optimal Set, PS), 而 PS 在目标空间的投影被称为帕累托前沿 (Pareto Front, PF).

理想点 z^* (Ideal point) 可用于刻画 PF 的最小值, 其定义如下:

$$z_i^* = \min_{\mathbf{x} \in \Omega} f_i(\mathbf{x}), \quad i = 1, \dots, m \quad (2)$$

3.2 分解方法

常用的分解方法包括加权和 (Weighted Sum, WS) 和切比雪夫 (Tchebycheff Approach, TCH)^[10]. 加权和只适用于形状为凸的 PF, 切比雪夫则无此限制. 虽然它的聚合函数对于连续 MOP 来说是不平滑的, 但由于在本文中不需对聚合函数求导, 所以本

文使用了切比雪夫分解方法. 其定义如下:

$$\begin{aligned} \text{minimize } \mathbf{g}^{\text{TCH}}(\mathbf{x} | \boldsymbol{\lambda}^i, z^*) &= \max_{1 \leq j \leq m} \lambda_j^i |f_j(\mathbf{x}) - z_j^*| \\ \text{subject to } \mathbf{x} &\in \Omega \end{aligned} \quad (3)$$

其中, $\boldsymbol{\lambda} = \{\lambda^1, \dots, \lambda^m\}$ 是一组方向向量, λ^i 是其中任意一个方向向量, λ_j^i 是 λ^i 的第 j 个分量, 并且满足

$$\sum_{j=1}^m \lambda_j^i = 1, \quad z_i^* \text{ 是其理想点.}$$

3.3 多任务高斯过程

相比单任务高斯过程, 多任务高斯过程具有更好的泛化性, 并且能够利用样本在多个相关目标下的有用信息来提升预测质量^[29-30].

本文中采用 Titsias 等人提出的 spike and slab^[31] 重新参数化方法来近似训练 $\text{Model}(\tilde{f}_q(\mathbf{x}), \sigma_q^2)$, 其中 $\tilde{f}_q(\mathbf{x}), \sigma_q^2$ 分别为一个待测点 \mathbf{x} 在第 q 个任务上的预测均值和方差, 假设 $\tilde{w}_{ql} \sim N(\tilde{w}_{ql} | 0, \sigma_w^2)$, 并且 $s_{ql} \sim \pi^{s_{ql}} (1 - \pi)^{1 - s_{ql}}$, 则模型表达式如下所示:

$$\tilde{f}_q(\mathbf{x}) = \sum_{l=1}^M s_{ql} \tilde{w}_{ql} \phi(\mathbf{x}) \quad (4)$$

$$\begin{aligned} \sigma_q^2 &= \frac{1}{N} \text{tr} \left[\mathbf{y}_q \mathbf{y}_q^T - \mathbf{y}_q \sum_{l=1}^M \langle s_{ql} \tilde{w}_{ql} \rangle \langle \phi_l \rangle^T + \right. \\ &\quad \left. \sum_{l=1}^M \langle s_{ql} \tilde{w}_{ql}^2 \rangle \langle \phi_l \phi_l^T \rangle + \right. \\ &\quad \left. 2 \sum_{l > l'} \langle s_{ql} \tilde{w}_{ql} \rangle \langle s_{ql'} \tilde{w}_{ql'} \rangle \langle \phi_l \rangle \langle \phi_{l'} \rangle \right] \end{aligned} \quad (5)$$

其中, N_D 为训练样本个数, 训练样本 $D_i = \{\mathbf{X}, \mathbf{Y}\}$, $\mathbf{X} \in R^{N_D \times n}$, $\mathbf{Y} \in R^{N_D \times Q}$, \mathbf{X} 和 \mathbf{Y} 的第 i 行分别表示第 i 个训练样本的输入向量和输出向量, y_q 表示 \mathbf{Y} 的第 q 列, $\mathbf{w}_q = [w_{q1}, \dots, w_{qM}]^T$ 是一组方向向量, $\phi(\mathbf{x}) = [\phi_1(\mathbf{x}), \dots, \phi_M(\mathbf{x})]^T$ 是相应的一组高斯过程, s 是一个二进制变量. 括号 $\langle \cdot \rangle$ 中的任何内容都是根据变分分布的当前值计算的, 并假定为固定的. $\text{tr}(\cdot)$ 为矩阵的迹. 通常情况下高斯过程个数 $M < Q$, 本文设置 $M = Q$, 除了方便设置参数 M 以外, 模型的预测效果也会有所提升.

4 基于自适应分解的多任务协作型昂贵多目标优化算法

4.1 算法流程

算法框架如算法 1 所示. 下面就算法各个步骤分别介绍.

① 目标函数为最大化时, 不等式符号与占优符号应反转.

算法 1. AMMCS 主要流程.

输入: 子问题数 N , 决策变量个数 n , 相关任务数 Q

输出: 非支配解集 A^*

1. FOR EACH $i = 1, \dots, N, B(i) = \{i_1, \dots, i_T\}, \lambda^{i_1}, \dots, \lambda^{i_T}$ 均匀分布的权重向量 $\lambda = (\lambda^1, \dots, \lambda^N)$ DO
2. $A = \text{Latinhypercube}(11n-1)$
3. $P = A; z^* = \min_{x \in A} f_i(x), i = 1, \dots, m;$
4. $i = \text{rand}() * N$
5. $[\mathbf{TW}, \mathbf{TS}] = \text{GRT}(Q, P, A, \lambda^i, \lambda);$ // 获取相关任务
6. $\text{training_data} \leftarrow A;$ // 训练模型
7. $\text{Model} \leftarrow \text{multi-taskGPLearning}(\text{training_data})$
// 基于多种群协作搜索的批量处理
8. $[P_B, \lambda_B] = \text{BO}(\text{Model}, N, Q, P, A, \mathbf{TW}, \mathbf{TS});$
9. $F(P_B)$, 并用 P_B 来更新 z^* 和 λ_B 邻居的解;
10. $A = A \cup P_B;$
11. END FOR
12. $A^* = \text{NDSort}(A);$ // A 为非支配解集
13. RETURN A^*

4.2 初始化

如算法 1 所示, AMMCS 首先产生一组均匀分布的方向向量 $\lambda^1, \dots, \lambda^N$, 并使用切比雪夫分解方法(式(2))把一个 MOP 分解成 N 个子问题. 接着使用 Latin hypercube sampling^[32] 来产生并评估 $11n-1$ 个初始解并保存到一个外部集 A 中, 同时使用 A 来初始化一个工作种群 P 和理想点 z^* .

4.3 获取相关任务

正如算法 2 所示, GRT 函数首先计算 A 中的解集在子问题 λ^i 下的聚合函数向量 $\mathbf{g}^{\text{TCH}}(A|\lambda^i, z^*)$, 对任一方向向量 $\lambda^j \in \lambda$, 计算 A 中的解集在 λ^j 下的聚合函数向量 $\mathbf{g}^{\text{TCH}}(A|\lambda^j, z^*)$. 计算皮尔逊相关系数判断 λ^j 是否和 λ^i 相关, 计算公式如下:

$$\rho(\mathbf{g}^{\text{TCH}}(A|\lambda^i, z^*), \mathbf{g}^{\text{TCH}}(A|\lambda^j, z^*)) = \frac{\text{cov}(\mathbf{g}^{\text{TCH}}(A|\lambda^i, z^*), \mathbf{g}^{\text{TCH}}(A|\lambda^j, z^*))}{\sigma_i \sigma_j} \quad (6)$$

其中, $\text{cov}(\mathbf{g}^{\text{TCH}}(A|\lambda^i, z^*), \mathbf{g}^{\text{TCH}}(A|\lambda^j, z^*))$ 表示 $\mathbf{g}^{\text{TCH}}(A|\lambda^i, z^*)$ 和 $\mathbf{g}^{\text{TCH}}(A|\lambda^j, z^*)$ 的协方差, σ_i 和 σ_j 分别是 $\mathbf{g}^{\text{TCH}}(A|\lambda^i, z^*)$ 和 $\mathbf{g}^{\text{TCH}}(A|\lambda^j, z^*)$ 的标准差. 皮尔逊相关系数标准为: 相关系数 r 的绝对值一般在 0.8 以上, 认为 A 和 B 有强的相关性. 0.3 到 0.8 之间, 可以认为有弱的相关性. 0.3 以下, 认为没有相关性.

算法 2. 获取相关任务(GRT).

输入: 相关任务数 Q , 当前种群 P , 所有评估过的解集 A , 被选中的权重向量 λ_i , 所有权重向量 λ

输出: 任务的代表性权重向量集 \mathbf{TW} , 任务包含子问题关联的解集 \mathbf{TS}

1. $W_i = \emptyset$
2. FOR EACH $\lambda^j \in \lambda$ DO
3. $G_p = |\rho(\mathbf{g}^{\text{TCH}}(A|\lambda^i, z^*), \mathbf{g}^{\text{TCH}}(A|\lambda^j, z^*))|$
4. IF $G_p \geq 0.8$ THEN
5. $W_i = W_i \cup \lambda^j$
6. END IF
7. END FOR
8. $W_i^1, \dots, W_i^Q = \text{cluster}(W_i)$
9. $\mathbf{TW} = \mathbf{TS} = \emptyset$
10. FOR EACH $j = 1:Q$ DO
11. $\mathbf{TW}_j = \text{random}(W_i^j)$
12. 获取在 P 中 W_i^j 对应的解集并保存到 \mathbf{TS}_j ;
13. END FOR
14. RETURN $\mathbf{TW}, \mathbf{TS};$

将由强相关性的 λ^j 保存到 W_i . 然后使用 K 均值聚类算法把 W_i 分成 Q 个聚类(任务). 最后从每个任务中随机选择一个子问题作为该任务的代表子问题, 将其保存到 \mathbf{TW} 中, 其关联解集保存到 \mathbf{TS} 中.

GCS-MOE^[26] 通过一个预设参数来指定每个任务包含的子问题数量, 其根据代表性子问题和其它子问题间的欧式距离排序来分配子问题到相关任务中, 这种静态的分配方式无法动态地反映任务间的相关性. 本文提出了利用皮尔逊相关系数指标动态地计算子问题的相关性, 以实现多任务自适应的划分, 从而提升多任务高斯代理模型的预测精度和优化算法的最终性能.

如图 1 所示, GRT 函数首先随机选择一个子问题 λ^i , 然后计算所有子问题与 λ^i 的相关性, 选出与 λ^i 相关的子问题集合. 最后使用聚类算法把这些相关子问题划分成多个任务, 并获取这些任务的代表性方向向量集 \mathbf{TW} 以及任务所包含子问题的当前解集 \mathbf{TS} .

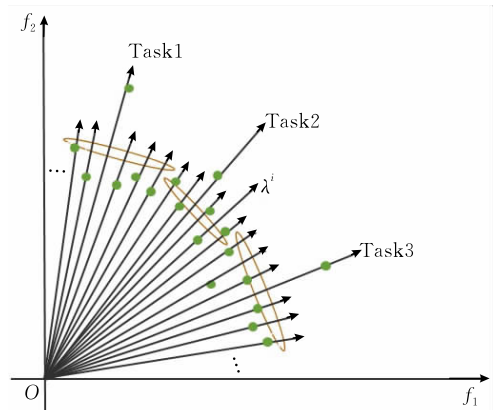


图 1 自适应划分任务示意图

4.4 模型管理

在获取了 Q 个相关任务后, 算法将会使用一个多任务高斯过程来近似每个任务代表性问题的聚合函数. 本文开发了一种简单的模型管理方法, 从所有评估过的解中挑选部分解作为样本, 来训练代理模型, 这种方法能够同时考虑 Q 个任务的情况.

如算法 3 所示, 首先计算所有评估过的解在这 Q 个任务代表性方向向量下的聚合函数向量, 并在对应任务代表性方向向量下对每个解的聚合函数值进行排序, 然后再对每个解在 Q 个任务下的序号按从小到大进行排序调整, 最后使用字典序排序从这些解中挑选部分解作为训练样本.

算法 3. 模型管理.

输入: 所有评估过的解集 A , 任务的代表性权重向量集

TW , 训练样本容量 m

输出: 训练样本 $training_data$

1. FOR EACH $A^i \in A$ DO
2. FOR EACH $j=1$ to Q DO
3. $g_{A^i}^j = g^{TCH}(A^i | TW_j, z^*)$
4. END FOR
5. END FOR
6. FOR EACH $j=1$ to Q DO
7. $[B, R^j] = sort(g^j)$
8. END FOR
9. FOR EACH $A^i \in A$ DO
10. $B, r_{A^i} = sort(R_{A^i})$
11. END FOR
12. $Index = argsort(r(A^i))$
13. $training_data \leftarrow A[Index(1, \dots, m)]$

获得样本后, 把这些解的决策变量组成一个矩阵作为输入, 将这些解在对应任务下的聚合函数值向量组成一个矩阵作为输出来训练多任务高斯过程模型.

4.5 基于多种群协作搜索的批量优化

针对只使用一个种群难以维持多个任务的情况, 本文使用一个解集(种群)优化一个任务, 并通过多种群的协作搜索实现多任务高斯模型的批量优化, 提高了采样效率, 提升算法的收敛效率.

算法使用多个规模为 $\lceil N/Q \rceil$ 的小种群来维持每个任务, 多个种群之间进行协作搜索来同时优化多个任务的代表性子问题; 此外, 每个任务的初始子代使用 Differential Evolution (DE) 算子^[31]生成, 优化过程中使用 Simulated Binary crossover (SBX) 算子^[33]生成子代.

在多种群协作搜索开始前, BO 会初始化一些

相关参数. 对任意任务 $i (i \in \{1, \dots, Q\})$, 从 A 中找出在第 i 个任务代表性方向向量下的最佳聚合函数值, 并保存到 TF_i 中. 然后根据 DE 算子, 使用第 i 个任务的相关解集 TS_i 和当前种群 P 来产生一个规模为 $\lceil N/Q \rceil$ 的初始子代 TC_i , 并把当前种群 P 的非支配解集作为该任务的初始父代 TP_i , Q 个任务的初始父代和子代分别保存在 TP 和 TC . CS_i 用来保存第 i 个任务当前找到的最佳候选解, 并且 CS_i 初始为空, CS 保存着 Q 个任务的最佳候选解. 第 i 个任务的当前最佳效用准则值 TU_i 初始为一个很大的常数, TU 用来保存 Q 个任务的当前最佳效用准则值. 最后, 使用训练好的多任务高斯过程模型来评估初始父代和初始子代, 并计算初始父代和初始子代的效用准则值.

在多种群协作搜索时, 需要多次迭代来找到每个任务的最佳候选解. 当优化第 i 个任务的最佳候选解时 ($i \in \{1, \dots, Q\}$), 根据效用准则值, 从每个任务 ($i \in \{1, \dots, Q\}$) 的当前子代 TC_j 中挑选出最佳的 $\lceil N/Q^2 \rceil$ 个解来组成一个规模为 $\lceil N/Q \rceil$ 的种群 P_T . 接下来使用代理模型预测 P_T 的聚合函数并计算相应的效用准则值. 根据效用准则值, 从 P_T 中挑选出一个最佳个体 I_B , 若优于 TU_i , 则使用 I_B 来更新 CS_i 和 TU_i . 然后, 算法合并 P_T 和第 i 个任务的当前父代 TP_i 组成一个合并种群 P_U , 并根据效用准则值来挑选出最佳的 $\lceil N/Q \rceil$ 个解来更新 TP_i , 利用 SBX 算子来更新 TC_i . 最后, 使用代理模型来评估 TC_i 并计算相应的效用准则值, 具体的算法步骤可以参考算法 4.

算法 4. 基于种群协作搜索的批量优化 (BO).

输入: 代理模型 $Model$, 子问题数目 N , 相关任务数 Q , 当前种群 P , 所有评估过的解集 A , 相关任务的代表性权重向量集 TW , 相关任务包含子问题关联的解集 TS

输出: 优化多个任务最佳候选解集 P_B , 最佳候选解集对应的权重向量 λ_B

//初始化相关参数

1. WHILE 未达到终止条件 DO
2. FOR EACH $i=1:Q$ DO
3. $sort(F(TS_i))$;
4. $PT_j = TS_i[1:\lceil N/Q \rceil]$
5. $Model(P_T)$;
6. $I_B = \arg \min(Utility(PT))$;
7. IF $Utility(I_B) \leq TU_i$ THEN
8. $CS_i = I_B$;
9. $TU_i = Utility(I_B)$;

10. END IF
11. $P_U = P_T \cup TP_i$;
12. $Index = \text{argsort}(F(P_U))$;
13. $TP_i = P_u[index(1, \dots, \lceil N/Q \rceil)]$
14. $TC_i = \text{SBX}(TP_i)$
15. $Model(TC_i)$;
16. $Utility(TC_i)$;
17. END FOR
18. END WHILE

图 2 是优化第二个任务时的大致流程. 假设任务数为 3, 首先从 3 个任务的当前子代 TC 中获取最佳的 $\lceil N/Q \rceil$ 个解组成 P_T , 然后根据效用准则选出最佳个体来更新第 2 个任务的最佳候选解 CS_2 , 并产生新的父代来替换 TP_2 .

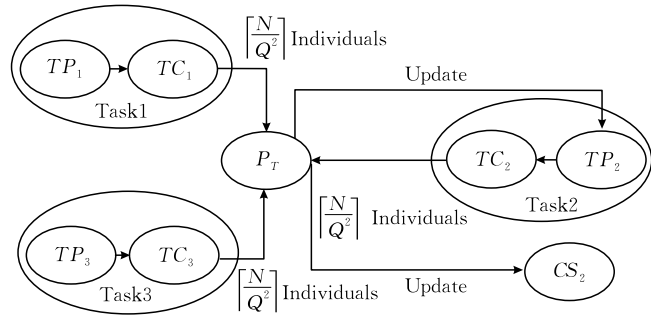


图 2 多种群协作搜索示意图

AMMCS 采用了 Upper Confidence Bound utility criterion (UCB)^[34] 以及 UCB 和 Expected Improvement utility criterion (EI) 的组合效用准则^[24]; 对于第 q 个任务, $q \in \{1, \dots, Q\}$

$$Utility_q(\mathbf{x}) = \begin{cases} f_q(\mathbf{x}) + \alpha\sigma_q, & f_q(\mathbf{x}) < TF_q \\ -[TF_q - \tilde{f}_q(\mathbf{x})] \Phi_N\left(\frac{TF_q - f_q(\mathbf{x})}{\sigma_q}\right) - \sigma_q \phi_N\left(\frac{TF_q - f_q(\mathbf{x})}{\sigma_q}\right) + C_F, & \text{其他} \end{cases} \quad (7)$$

当 $\tilde{f}_q(\mathbf{x}) < TF_q$ 时, 则选用 UCB 效用准则, 否则使用 EI 效用准则的负值再加上一个惩罚系数 C_F , 其中 TF_q 表示在第 q 个任务代表性子问题下真实计算出来的最佳聚合函数值。

下面介绍过滤 CS 中相似解的步骤. FSS 函数设置一个最短欧式距离阈值 $d_u = 0.01$, 计算两个解在决策空间的欧式距离, 如果小于这个阈值, 则认为这个两个解是相似的. 具体的算法流程可以参考算法 5.

算法 5. 过滤相似解 (FSS).

输入: 相关任务数 Q , 相关任务优化的所有最佳候选解

CS , 相关任务的代表性权重向量 TW

输出: 优化多个任务最佳候选解集 P_B 和最佳候选解集对应的权重向量 λ_B

1. $P_B = CS_1$
2. $\lambda_B = TW_1$
3. FOR EACH $i = 2 : Q$ DO
4. $d_{\min} = \text{argmin}(\text{distances}(CS_i, P_B))$;
5. IF $d_{\min} > d_u$ THEN
6. $P_B = P_B \cup CS_i$;
7. $\lambda_B = \lambda_B \cup TW_i$;
8. END IF
9. END FOR
10. RETURN P_B, λ_B

4.6 时间复杂度分析

AMMCS 一次迭代时间消耗主要取决于算法 1

中的步骤 5(算法 2)、步骤 7(算法 3)和步骤 8(算法 4). 对于算法 2 的步骤 3 到步骤 8, 从最坏情况考虑, 假设已评估的解集达到最大评估次数 N_E , 计算皮尔逊相关系数中时需计算 N_E^2 次, 又因为子问题数量为 N , 得到相应的时间复杂度 $O(NN_E^2)$. 对于算法 2 的步骤 9, 因为 K 均值聚类的迭代次数、任务数 Q 和每个子问题的方向向量维度 m 通常都是不大的常数, 所以 N 个子问题划分成 Q 个任务的时间复杂度可简化为 $O(N)$. 因此, 算法 2 的时间复杂度主要为 $O(NN_E^2)$. 算法 1 中步骤 7 的多任务高斯过程训练时间消耗相对较多, 但由于限制了训练样本数, 所以模型在训练和预测的时间中消耗会相对少很多; 此外, 由于本文所选的多任务高斯过程工作原理过于复杂, 导致计算模型训练和预测的时间复杂度较为困难, 为了方便计算算法 3 的时间复杂度, 忽略了模型预测的时间消耗. 算法 3 的主要时间消耗在步骤 1 和 10. 在步骤 1 中, 一个任务包含的解集规模为 N/Q , 其排序的时间复杂度为 $O((N/Q)\log(N/Q))$, Q 个任务排序的时间复杂度可简化为 $O(N\log(N))$. 步骤 10 同为 $O(N\log(N))$. 算法 3 的时间复杂度为 $O(N\log(N))$.

由于最大评估次数 N_E 通常是种群数 N 的倍数, AMMCS 时间消耗主要在步骤 5(算法 2), 所以, AMMCS 最终时间复杂度为 $O(NN_E^2)$.

5 实验结果与分析

5.1 对比算法

在这章小节中,我们将 AMMCS 和六个代理辅助进化算法(GCS-MOE^[26]、MOEA/D-EGO^[17]、K-RVEA^[19]、ParEGO^[15]、CSEA^[20]和 SMS-EGO^[6]) 在 ZDT 和 DTLZ 测试问题集上进行对比实验. 其中 AMMCS 使用 Matlab 实现, GCS-MOE 是用 C++ 实现, 其它的对比算法则由 Matlab 实现的 PlatEMO 平台提供^[35]. 在这些测试问题上, 上述七个代理辅助进化算法都独立地运行 20 次.

5.2 实验设置

(1) 所有测试问题的决策变量数 n 均设置为 8, DTLZ 测试问题集的目标数 m 均设置为 3.

(2) 种群规模或子问题数目 N 在 ZDT 测试问题集上设置为 150, 在 DTLZ 测试问题集上设置为 230.

(3) 所有测试问题的最大目标函数评估次数 N_E 都统一设置为 200 次.

(4) 初始生成的解集数目设置为 $11n-1$.

(5) $C_F=100\,000$, $\alpha=5$.

(6) 这些参数在所有对比算法上都是相同的.

AMMCS 在 ZDT 测试集上的相关任务数 Q 设置为 3, 在 DTLZ 测试集上的 Q 设置为 5, $K_E=Q$. 对比算法中 $K_E=5$. AMMCS、GCS-MOE 和 MOEA/D-EGO 的邻居规模设置为 20. 由于训练多任务高斯过程会比较耗时, 因此, 本文对 AMMCS 和 GCS-MOE 的训练样本数和模型求解超参数的迭代次数都进行了限制: AMMCS 和 GCS-MOE 的最大训练样本数都设置为 100, 求解多任务高斯过程超参数的迭代次数也设置为 100. 除上述参数外, 所

有算法的其它实验设置均和原论文保持一致.

5.3 评价指标

Inverted Generational Distance (IGD)^[36] 和 Hypervolume (HV)^[35] 指标被用于度量多目标算法性能. IGD 需要真实 PF 作为参考来进行指标计算. 本文从真实 PF 中均匀采样作为参考 PF. 对于 ZDT 测试集的参考 PF, 除 ZDT3 从真实 PF 上均匀地采集 2658 个样本点, 其它 ZDT 测试问题从真实 PF 上均匀采集 10 000 个样本点作为参考 PF. 对于 DTLZ 测试集, 除 DTLZ1 到 DTLZ4 是从真实 PF 上均匀采集 9870 个样本点外, 其它问题从真实 PF 上均匀采集 10 000 个样本点作为参考 PF. 对于 HV 指标, 真实 PF 中各目标的最大值作为参考点.

此外, 本文还使用显著性水平为 0.05 的 Wilcoxon 秩和检验来度量 AMMCS 和其它对比算法的性能显著性. 符号 =, + 和 - 则分别表示 AMMCS 性能相较于其它对比算法, 没有显著差异、显著好以及显著差.

5.4 AMMCS 和五个算法的对比实验结果

表 1 和表 2 为 AMMCS 和六个对比算法在 ZDT、DTLZ 问题集上的 IGD 和 HV 指标, 加粗数据均表示相应算法在某个测试问题上的最佳指标值. AMMCS 的 IGD 和 HV 指标在 ZDT 和 DTLZ 中大部分测试问题效果都是最好的. 在 DTLZ2 和 DTLZ4 测试问题上, 尽管 AMMCS 的 IGD 指标和次优的 K-RVEA 的 IGD 指标在数值上比较相似, 但在 HV 指标上, AMMCS 明显要优于 K-RVEA. 在 ZDT3、DTLZ6 和 DTLZ7 这些具有不规则 PF 的测试问题上, 在 ZDT3、DTLZ6 和 DTLZ7 这些具有不规则 PF 的测试问题上, 由于算法中使用的方向向量集合是预先设定的, 未进行动态地调整, 因此对具有不规则 PF 的多目标优化问题算法的效果不佳.

表 1 AMMCS 和六个对比算法在 ZDT 和 DTLZ 问题集上的 IGD 指标值

Problem	AMMCS	MOEA/D-EGO	K-RVEA	ParEGO	SMS-EGO	GCS-MOE	CSEA
ZDT1	0.0136 (0.0077)	0.0419 (0.0143)+	0.0367 (0.0104)+	0.0327 (0.0085)+	0.7433 (0.2047)+	0.0203 (0.0019)+	3.5084 (3.0204)+
ZDT2	0.0112 (0.0018)	0.0528 (0.0186)+	0.0623 (0.0162)+	0.0347 (0.0133)+	0.5963 (0.0418)+	0.0241 (0.0075)+	4.9950 (3.0581)+
ZDT3	0.0994 (0.0968)	0.1636 (0.0777)+	0.0594 (0.0210)=	0.0847 (0.0190)=	0.8203 (3.2306e-06)+	0.0534 (0.0098)=	3.2939 (3.3410)+
ZDT4	25.7984 (9.7550)	55.9804 (8.5000)+	30.2408 (12.6291)=	43.5971 (12.4912)+	42.9459 (12.1274)+	1.1970e+03 (468.5990)+	39.3909 (14.7987)=

(续 表)

Problem	AMMCS	MOEA/D-EGO	K-RVEA	ParEGO	SMS-EGO	GCS-MOE	CSEA
ZDT6	0.0859 (0.0565)	0.3656 (0.1388)+	1.6806 (0.4361)+	0.2837 (0.0647)+	1.5663 (1.7791)+	0.4098 (0.1406)+	7.8408 (1.3328)+
DTLZ1	29.9376 (22.2089)	63.9595 (10.3352)+	64.5656 (18.2218)+	45.7748 (5.1928)+	45.8687 (5.4631)+	42.3877 (12.3031)=	58.4731 (14.0905)+
DTLZ2	0.1285 (0.0202)	0.2638 (0.0202)+	0.1333 (0.0075)=	0.2872 (0.0187)+	0.1894 (0.0282)+	0.2051 (0.0105)+	0.1914 (0.0218)+
DTLZ3	78.6525 (39.9586)	149.9160 (49.5795)+	122.4429 (54.4351)+	123.6749 (8.0531)+	119.4050 (7.3474)+	137.4559 (19.5956)+	116.8795 (43.4190)=
DTLZ4	0.3390 (0.2248)	0.5306 (0.0710)+	0.3472 (0.0730)=	0.5732 (0.0688)+	0.6277 (0.1049)+	0.5326 (0.0545)+	0.5878 (0.1695)+
DTLZ5	0.0267 (0.0045)	0.1870 (0.0192)+	0.0564 (0.0109)+	0.1992 (0.0262)+	0.0811 (0.0191)+	0.0637 (0.0073)+	0.0917 (0.0239)+
DTLZ6	0.8905 (0.1200)	1.5763 (0.7033)+	2.5223 (0.2291)+	0.5000 (0.1619)-	2.4205 (0.5776)+	0.0869 (0.0777)-	3.8567 (0.4099)+
DTLZ7	0.3524 (0.0636)	0.2068 (0.0535)-	0.3031 (0.0987)=	0.1864 (0.0238)-	1.5379 (3.7928e-09)+	0.2215 (0.0256)-	1.9983 (0.6665)+

表 2 AMMCS 和六个对比算法在 ZDT 和 DTLZ 问题集上的 HV 指标值

Problem	AMMCS	MOEA/D-EGO	K-RVEA	ParEGO	SMS-EGO	GCS-MOE	CSEA
ZDT1	0.7088 (0.0057)	0.6625 (0.0231)+	0.6825 (0.0075)+	0.6847 (0.0110)+	0.1714 (0.1697)+	0.7020 (0.0027)+	0 (0)+
ZDT2	0.4364 (0.0022)	0.3746 (0.0252)+	0.3607 (0.0147)+	0.4080 (0.0174)+	0.0929 (0.0063)+	0.4221 (0.0098)+	0 (0)+
ZDT3	0.5467 (0.0932)	0.6131 (0.0800)=	0.6124 (0.0551)=	0.5238 (0.0309)+	0.0909 (3.1362e-06)+	0.5618 (0.0108)=	0.0024 (0.0077)+
ZDT4	0 (0)	0 (0)=	0 (0)=	0 (0)=	0 (0)=	0 (0)=	0 (0)=
ZDT6	0.2948 (0.0573)	0.1726 (0.0638)+	0 (0)+	0.1231 (0.0315)+	0.0647 (0.0946)+	0.1823 (0.0458)+	0 (0)+
DTLZ1	0 (0)	0 (0)=	0 (0)=	0 (0)=	0 (0)=	0 (0)=	0 (0)=
DTLZ2	0.4519 (0.0213)	0.2492 (0.0303)+	0.4252 (0.0188)+	0.2159 (0.0433)+	0.4396 (0.0430)=	0.3494 (0.0132)+	0.3297 (0.0380)+
DTLZ3	0 (0)	0 (0)=	0 (0)=	0 (0)=	0 (0)=	0 (0)=	0 (0)=
DTLZ4	0.3327 (0.1041)	0.0644 (0.0483)+	0.1676 (0.0852)+	0.0306 (0.0410)+	0.0421 (0.0605)+	0.0781 (0.0619)+	0.1382 (0.0622)+
DTLZ5	0.1751 (0.0062)	0.0633 (0.0269)+	0.1557 (0.0073)+	0.0432 (0.0350)+	0.1503 (0.0191)+	0.1404 (0.0107)+	0.1144 (0.0265)+
DTLZ6	0 (0)	0.0189 (0.0400)=	0 (0)=	0.0262 (0.0378)-	0 (0)=	0.1485 (0.0470)-	0 (0)=
DTLZ7	0.1953 (0.0364)	0.2105 (0.0233)=	0.2309 (0.0111)-	0.2446 (0.0064)-	0.0909 (5.6117e-10)+	0.2306 (0.0106)-	0.0307 (0.0345)+

图 3 和图 4 展示了 ZDT2 和 DTLZ5 的真实 PF 以及算法在其上获得的非支配解集(ND solutions). 在 ZDT2 和 DTLZ5 测试问题上, AMMCS 得到的 PF 近似解集大多能落到真实 PF 上, 其多样性保持也较好. GCS-MOE、MOEA/D-EGO、K-RVEA 和 ParEGO 得到的部分解集也能收敛到真实 PF 上, 但其多样性比 AMMCS 差. CSEA 和 SMS-EGO 得到的解集几乎未能落在真实 PF 上且其多样性较差. 这些观察也进一步表明了 AMMCS 算法性能在

ZDT 和 DTLZ 测试集上比其它比较算法具有更好的性能.

此外, 七个算法运行时间比较如表 3 所示. 使用多任务高斯过程作为代理模型的算法(AMMCS 和 GCS-MOE)比使用其它模型的算法在运行时间要慢. 这是由于多任务高斯过程在有效提升模型预测精度的同时, 也增加了算法的运行时间. 但是值得指出的是 AMMCS 比其它基于多任务高斯过程模型的算法(如 GCS-MOE)要快一个数量级.

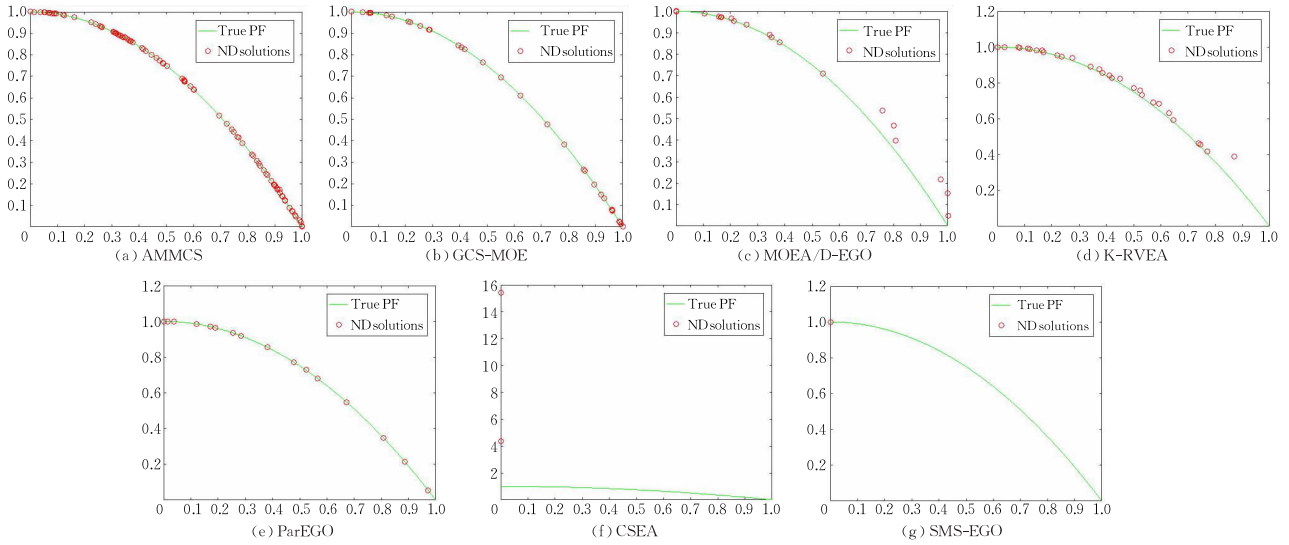


图 3 所有算法在 ZDT2 问题的 IGD 值中位数运行次数下的非支配解集分布图

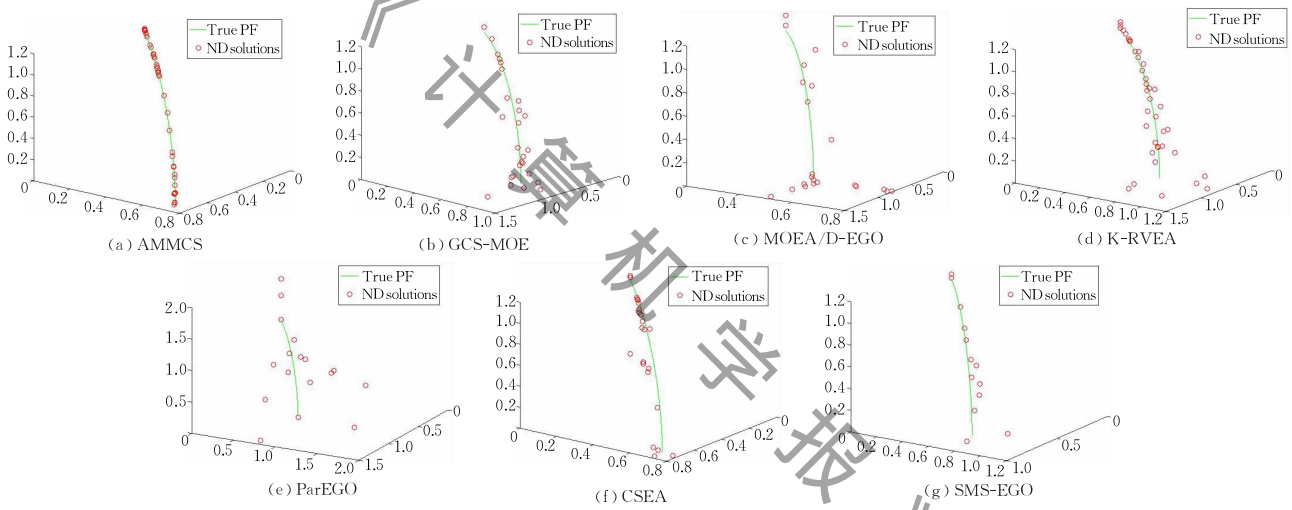


图 4 所有算法在 DTLZ5 问题的 IGD 值中位数运行次数下的非支配解集分布图

表 3 AMMCS 和六个对比算法在 ZDT 和 DTLZ 问题集上的运行时间

(单位:s)

Problem	AMMCS	GCS-MOE	MOEA/D-EGO	K-RVEA	ParEGO	CSEA	SMS-EGO
ZDT1	1.2120e+03	1.8540e+04	25.0107	22.1496	197.5482	43.5753	546.2811
ZDT2	1.5240e+03	4.8030e+04	28.4244	23.8856	167.6976	40.9840	490.7907
ZDT3	1.5120e+03	2.0910e+04	21.7444	43.2417	133.8873	29.8067	451.5278
ZDT4	3.5280e+03	2.0028e+04	31.1026	25.0240	127.9629	119.8566	480.1318
ZDT6	2.1060e+03	2.2164e+04	29.4664	25.0141	125.5214	41.5211	446.2368
DTLZ1	1.9620e+03	3.3132e+04	45.9058	47.9588	122.6998	160.2450	1.3251e+04
DTLZ2	1.4640e+03	2.1894e+04	48.5002	53.8045	126.1783	45.6651	1.2425e+04
DTLZ3	1.7580e+03	4.1514e+04	41.8672	38.4986	121.6028	159.0521	1.0157e+04
DTLZ4	2.0040e+03	1.3626e+04	33.4986	28.9384	129.3153	73.5848	1.1997e+04
DTLZ5	1.4280e+03	2.0598e+04	42.2752	23.2398	126.1103	49.1969	4.7360e+03
DTLZ6	1.1040e+03	2.0700e+04	40.9013	39.3512	126.2922	21.7920	2.7504e+03
DTLZ7	1.1820e+03	3.0102e+04	39.0987	40.2931	126.7335	17.2450	3.6179e+03

5.5 自适应任务划分的有效性

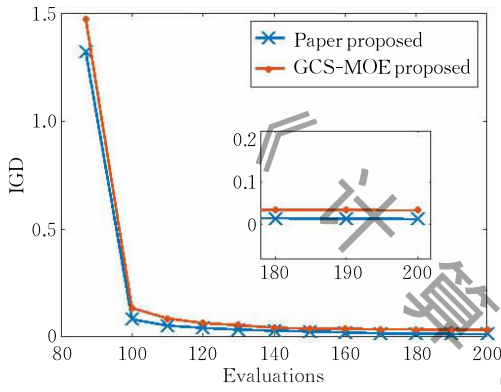
为了验证自适应任务划分方法的有效性,我们对原始 AMMCS 算法和使用固定任务划分方法的 AMMCS 算法(AMMCS+固定任务划分)。

如表 4 所示,原始 AMMCS 在大多数测试问题

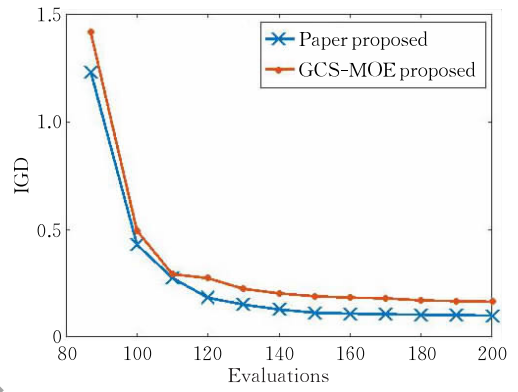
上的 IGD 和 HV 指标都比(AMMCS+固定任务划分)要显著更好.图 5 的收敛曲线也进一步直观展示了自适应多任务划分方法比固定多任务划分方法在收敛速度和最终性能上效果更好.这些对比实验都验证了自适应任务划分方法的有效性。

表 4 原始 AMMCS 和(AMMCS+固定任务划分)算法的 IGD 和 HV 指标值

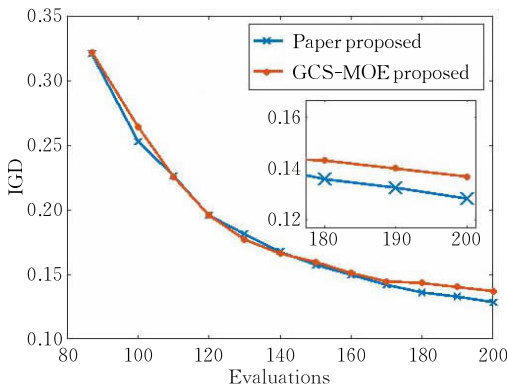
Metric		IGD		HV	
Problem	m	原始 AMMCS	AMMCS+固定任务划分	原始 AMMCS	AMMCS+固定任务划分
ZDT1	2	0.0136(0.0077)	0.0340(0.0204)	0.7088(0.0057)	0.6967(0.0117)
ZDT2	2	0.0112(0.0018)	0.0111(0.0018)	0.4364(0.0022)	0.4352(0.0038)
ZDT3	2	0.0994(0.0968)	0.1639(0.0919)	0.5467(0.0932)	0.6459(0.0897)
ZDT4	2	25.7984(9.7550)	26.2871(7.5732)	0(0)	0(0)
ZDT6	2	0.0859(0.0565)	0.1033(0.0439)	0.2948(0.0573)	0.2839(0.0303)
DTLZ1	3	29.9376(22.2089)	23.4454(9.3589)	0(0)	0(0)
DTLZ2	3	0.1285(0.0202)	0.1371(0.0120)	0.4519(0.0213)	0.4216(0.0213)
DTLZ3	3	78.6525(39.9586)	63.7645(0.2166)	0(0)	0(0)
DTLZ4	3	0.3390(0.2248)	0.6707(0.2166)	0.3327(0.1041)	0.1769(0.0815)
DTLZ5	3	0.0267(0.0045)	0.0311(0.0081)	0.1751(0.0062)	0.1756(0.0075)
DTLZ6	3	0.8905(0.1200)	0.9207(0.2165)	0(0)	0(0)
DTLZ7	3	0.3524(0.0636)	0.7076(0.1900)	0.1953(0.0364)	0.2050(0.0134)



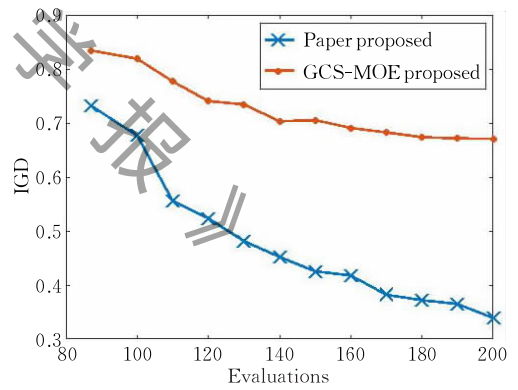
(a) ZDT1



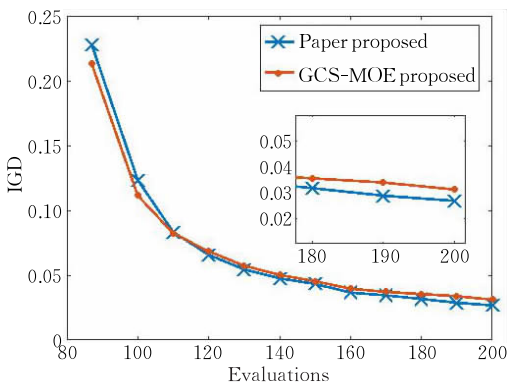
(b) ZDT3



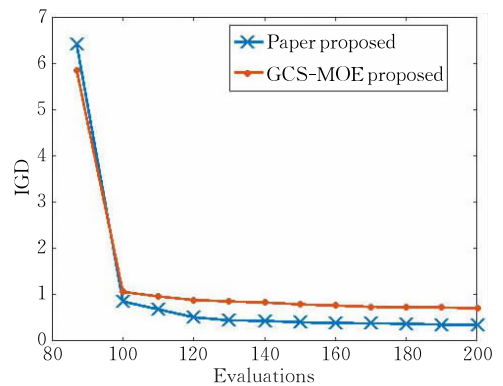
(c) DTLZ2



(d) DTLZ4



(e) DTLZ5



(f) DTLZ7

图 5 原始 AMMCS 和(AMMCS+固定任务划分)算法运行 10 次的 IGD 均值收敛曲线图

5.6 多种群协作搜索的有效性

AMMCS 采用了多种群协作搜索方法来同时优化多个任务, 获取多个待评估的候选解. 在这个小节中, 我们对比原始 AMMCS 和 (AMMCS+单种群搜索), 结果如表 5 所示.

正如表 5 所展示的一样, 原始 AMMCS 算法在大部分测试问题上的 IGD 以及 HV 指标值都比使用 (AMMCS+单种群搜索) 的显著性好. 这组对比实验显示了多种群协作搜索的有效性.

表 5 原始 AMMCS 和 (AMMCS+单种群搜索) 算法的 IGD 和 HV 指标值

Metric		IGD		HV	
Problem	m	原始 AMMCS	AMMCS+单种群搜索	原始 AMMCS	AMMCS+单种群搜索
ZDT1	2	0.0136(0.0077)	0.0938(0.0407)	0.7088(0.0057)	0.6564(0.0207)
ZDT2	2	0.0112(0.0018)	0.1489(0.0922)	0.4364(0.0022)	0.2977(0.0744)
ZDT3	2	0.0994(0.0968)	0.2215(0.1682)	0.5467(0.0932)	0.6217(0.1371)
ZDT4	2	25.7984(9.7550)	23.0679(8.0165)	0(0)	0(0)
ZDT6	2	0.0859(0.0565)	0.3365(0.1833)	0.2948(0.0573)	0.1472(0.0843)
DTLZ1	3	29.9376(22.2089)	25.6019(8.7114)	0(0)	0(0)
DTLZ2	3	0.1285(0.0202)	0.1323(0.0155)	0.4519(0.0213)	0.4191(0.0270)
DTLZ3	3	78.6525(39.9586)	61.7915(22.3280)	0(0)	0(0)
DTLZ4	3	0.3390(0.2248)	0.4107(0.0871)	0.3327(0.1041)	0.2469(0.0611)
DTLZ5	3	0.0267(0.0045)	0.0328(0.0084)	0.1751(0.0062)	0.1692(0.0079)
DTLZ6	3	0.8905(0.1200)	1.1236(0.2769)	0(0)	0(0)
DTLZ7	3	0.3524(0.0636)	0.8075(0.2447)	0.1953(0.0364)	0.1731(0.0172)

5.7 相关任务数量参数的敏感性测试

AMMCS 含有相关任务数 Q 这个参数. 我们在本节中对其进行了敏感度测试: 即分析不同 Q 值 (1, 3 和 5) 对算法性能的影响, 结果如表 6 所示.

当相关任务数 $Q=1$ 时, 多任务高斯过程会变成只有单个核函数的高斯过程, 模型的训练时间会

大幅降低, 但此时未能利用训练样本在多个相关任务下的有用信息, 模型预测精度不高, 且批量采用退化为单点采样, 降低了优化算法的最终性能. 随着 Q 值增加 ($Q=3$), 模型训练的复杂度会上升, 但模型能利用训练样本在相关任务中的隐含相关信息来提升模型的预测精度, 并可通过多种群协作搜索实现

表 6 AMMCS 在不同任务数量参数 Q 下的 IGD 和 HV 指标值

Metric	IGD			HV		
	Q=1	Q=3	Q=5	Q=1	Q=3	Q=5
ZDT1	0.0147 (0.0074)	0.0136 (0.0077)	0.0359 (0.0335)	0.7088 (0.0050)	0.7088 (0.0057)	0.6939 (0.0204)
ZDT2	0.0169 (0.0164)	0.0112 (0.0018)	0.0123 (0.0069)	0.4290 (0.0201)	0.4364 (0.0022)	0.4331 (0.0121)
ZDT3	0.0531 (0.0317)	0.0994 (0.0968)	0.1687 (0.1404)	0.5970 (0.0683)	0.5467 (0.0932)	0.5567 (0.1265)
ZDT4	41.0998 (7.3467)	25.7984 (9.7550)	24.6718 (8.2773)	0 (0)	0 (0)	0 (0)
ZDT6	0.0754 (0.0551)	0.0859 (0.0565)	0.0903 (0.0374)	0.3083 (0.0535)	0.2948 (0.0573)	0.2784 (0.0457)
DTLZ1	59.1262 (6.7684)	34.1900 (12.9851)	29.9376 (22.2089)	0 (0)	0 (0)	0 (0)
DTLZ2	0.2673 (0.0282)	0.1375 (0.0132)	0.1285 (0.0202)	0.1997 (0.0395)	0.4359 (0.0215)	0.4519 (0.0213)
DTLZ3	142.2761 (12.6293)	83.3694 (69.3625)	78.6525 (39.9586)	0 (0)	0 (0)	0 (0)
DTLZ4	0.6354 (0.1370)	0.3356 (0.0938)	0.3390 (0.2248)	0.0497 (0.0580)	0.2924 (0.0845)	0.3327 (0.1041)
DTLZ5	0.1641 (0.0378)	0.0246 (0.0032)	0.0267 (0.0045)	0.0705 (0.0275)	0.1800 (0.0050)	0.1751 (0.0062)
DTLZ6	1.0736 (0.4973)	0.8877 (0.1881)	0.8905 (0.1200)	0 (0)	0 (0)	0 (0)
DTLZ7	0.2671 (0.0598)	0.3701 (0.1859)	0.3524 (0.0636)	0.2191 (0.0160)	0.2202 (0.0226)	0.1953 (0.0364)

批量候选解的评估,提高优化算法的性能.然而,由于评估次数有限,随着 Q 的继续上升($Q=5$),由于训练样本数量并没有得到相应的增长,这会影响模型的预测精度以及优化算法的最终性能.因此在本文中,相关任务数 Q 在解决两目标测试问题时设置为3,在解决三目标测试问题时,设置为5.

5.8 效用准则对算法的影响

效用准则主要是用来度量候选解重要性的指标,具体来说就是在一个预测目标均值比较有前途的区域和一个预测不确定性大(通常用方差度量)的

未知空间进行权衡的准则.根据没有免费的午餐准则,一个效用准则很难在所有测试问题上都取得最好的效果.

本节对比 AMMCS 使用 UCB 效用准则和一种由 UCB 和 EI 组合的效用准则(记为 UCB+EI)的性能,如表 7 所示.在大部分测试问题上,AMMCS 在 UCB 效用准则上的算法性能明显优于或者接近于 UCB+EI 效用准则上的算法性能.在具有不规则 PF 的 ZDT3 和 DTLZ7 上,AMMCS 在 UCB 效用准则上的算法性能明显差于在 UCB+EI 上的算法性能.

表 7 AMMCS 算法在不同效用准则下的指标值

Metric		IGD		HV	
Problem	m	UCB	UCB+EI	UCB	UCB+EI
ZDT1	2	0.0136(0.0077)	0.0117(0.0018)	0.7088(0.0057)	0.7105(0.0023)
ZDT2	2	0.0112(0.0018)	0.0154(0.0029)	0.4364(0.0022)	0.4303(0.0038)
ZDT3	2	0.0994(0.0968)	0.0483(0.0082)	0.5467(0.0932)	0.5646(0.0082)
ZDT4	2	25.7934(9.7550)	20.8063(9.5802)	0(0)	0(0)
ZDT6	2	0.0859(0.0565)	0.2068(0.1722)	0.2948(0.0573)	0.2139(0.0543)
DTLZ1	3	29.9376(22.2089)	48.9643(20.8589)	0(0)	0(0)
DTLZ2	3	0.1285(0.0202)	0.1494(0.0190)	0.4519(0.0213)	0.4317(0.0107)
DTLZ3	3	78.6525(39.9586)	119.4373(50.5480)	0(0)	0(0)
DTLZ4	3	0.3390(0.2248)	0.4473(0.1868)	0.3327(0.1041)	0.2468(0.0755)
DTLZ5	3	0.0267(0.0045)	0.0405(0.0065)	0.1751(0.0062)	0.1631(0.0073)
DTLZ6	3	0.8905(0.1200)	1.0218(0.2902)	0(0)	0(0)
DTLZ7	3	0.3524(0.0636)	0.2322(0.0764)	0.1953(0.0364)	0.2342(0.0106)

造成这种现象的一种可能原因是在简单常规的问题上,多任务高斯过程能够较好地近似多个任务代表性子问题的聚合函数,使得 UCB 效用准则能够很好地辨别候选解的优劣来有效地指导搜索.但在较为复杂的测试问题上,多任务高斯过程未能较好地近似,影响了模型对候选解聚合函数值的预测精度,继续使用 UCB 效用准则进行指导搜索会造成很多较差的候选解被误识别为较好的候选解.然而使用一种由 UCB 和 EI 混合的效用准则可以从一定程度上减轻较差的候选解被误识别为较好候选解的概率.

6 总结与展望

本文提出了一种基于自适应分解的多任务协作型昂贵多目标优化算法,主要针对固定任务划分方法的不足,提出使用一种基于相似性指标的自适应任务划分方法,使代理模型能够充分利用样本在多个相关任务下的信息来提升预测精度.此外,进一步提出了一种基于多种群协作搜索方法批量获得多个任务的最佳候选解.在与六个代理辅助多目标进化算法在多组测试问题的实验对比表明,AMMCS 具

有更好的性能.

在算法中,因方向向量为预先设置的集合,未能进行自适应调整以适应不规则 PF,因此本文提出的方法在此类问题上实验效果不理想.在今后的研究中,将继续研究自适应方向向量调整并嵌入到算法中.

参 考 文 献

- [1] Deb K. Multi-Objective Optimization Using Evolutionary Algorithms. New Jersey, USA: John Wiley & Sons Inc, 2001
- [2] Coello C A C, Lamont G B, Van Veldhuizen D A. Evolutionary Algorithms for Solving Multi-Objective Problems. New York, USA: Springer, 2007
- [3] Kang Meng, Xu Feng. Multi-evolution strategy adaptive immune multi-objective evolution algorithm. Journal of Anhui University of Science and Technology(Natural Science Edition), 2019, 39(5): 43-47(in Chinese)
(康猛,许峰.多进化策略自适应免疫多目标进化算法.安徽理工大学学报(自然科学版),2019,39(5):43-47)
- [4] Li H, Zhang Q, Deng J. Biased multiobjective optimization and decomposition algorithm. IEEE Transactions on Cybernetics, 2016, 47(1): 52-66

- [5] Rasmussen C E. Gaussian Processes in Machine Learning. Berlin, Germany: Springer, 2003: 63-71
- [6] Haykin S. Neural networks: A comprehensive foundation. Prentice Hall PTR, 1994
- [7] Liaw A, Wiener M. Classification and regression by random-forest. *R News*, 2002, 2(3): 18-22
- [8] Ackermann E R, de Villiers J P, Cilliers P J. Nonlinear dynamic systems modeling using Gaussian processes: Predicting ionospheric total electron content over South Africa. *Journal of Geophysical Research Space Physics*, 2011, 116(A10): A10303: 1-13
- [9] Buche D, Schraudolph N N, Koumoutsakos P. Accelerating Evolutionary Algorithms with Gaussian Process Fitness Function Models. New Jersey, USA: IEEE Press, 2005
- [10] Zhang Q, Li H. MOEA/D: A multiobjective evolutionary algorithm based on decomposition. *IEEE Transactions on Evolutionary Computation*, 2007, 11(6): 712-731
- [11] Le M N, Ong Y S, Menzel S, et al. Multi co-objective evolutionary optimization: Cross surrogate augmentation for computationally expensive problems//*Proceedings of the 2012 IEEE Congress on Evolutionary Computation*. Brisbane, Australia, 2012: 1-8
- [12] Lin X, Zhen H L, Li Z, et al. A batched scalable multi-objective Bayesian optimization algorithm. *arXiv: Neural and Evolutionary Computing*, 2018
- [13] Knowles J. ParEGO: A hybrid algorithm with on-line landscape approximation for expensive multiobjective optimization problems. *IEEE Transactions on Evolutionary Computation*, 2006, 10(1): 50-66
- [14] Ponweiser W, Wagner T, Biermann D, et al. Multiobjective optimization on a limited budget of evaluations using model-assisted \mathcal{S} -metric selection//*Proceedings of the International Conference on Parallel Problem Solving from Nature*. Berlin, Germany: Springer, 2008: 784-794
- [15] Li Fei, Wu Zi-Heng, Liu Kan-Rong, Ge Er-Qian. High dimensional multi-objective particle swarm optimization algorithm based on R2 index and target space decomposition. *Control and Decision*, 2020: Online Publishing. <https://doi.org/10.13195/j.kzyjc.2020.0113>(in Chinese)
(李飞, 吴紫恒, 刘闾蓉, 葛二千. 基于 R2 指标和目标空间分解的高维多目标粒子群优化算法. *控制与决策*, 2020: 在线发表阶段. <https://doi.org/10.13195/j.kzyjc.2020.0113>)
- [16] Liu Jian-Chang, Zhao Yang-Jie, Li Fei, Son Yue-Xi. Expensive multi objective evolutionary algorithm based on R2 index. *Control and Decision*, 2020, 35(4): 823-832(in Chinese)
(刘建昌, 赵阳杰, 李飞, 宋悦熙. 基于 R2 指标的昂贵多目标进化算法. *控制与决策*, 2020, 35(4): 823-832)
- [17] Zhang Q, Liu W, Tsang E, et al. Expensive multiobjective optimization by MOEA/D with Gaussian process model. *IEEE Transactions on Evolutionary Computation*, 2010, 14(3): 456-474
- [18] Li H, Zhang Q. Multiobjective optimization problems with complicated Pareto sets, MOEA/D and NSGA-II. *IEEE Transactions on Evolutionary Computation*, 2009, 13(2): 284-302
- [19] Chugh T, Jin Y, Miettinen K, et al. A surrogate-assisted reference vector guided evolutionary algorithm for computationally expensive many-objective optimization. *IEEE Transactions on Evolutionary Computation*, 2016, 22(1): 129-142
- [20] Pan Linqiang, He Cheng, Tian Ye, et al. A classification based surrogate-assisted evolutionary algorithm for expensive many-objective optimization. *IEEE Transactions on Evolutionary Computation*, 2019, 23(1): 74-88
- [21] Bonilla E V, Chai K M, Williams C. Multi-task Gaussian process prediction//Cortes C, Lee D D, Lawrence N D, Sugiyama M, Garnett R eds. *Advances in Neural Information Processing Systems*. Cambridge, MA: MIT Press, 2008: 153-160
- [22] Sener O, Koltun V. Multi-task learning as multi-objective optimization//Cortes C, Lee D D, Lawrence N D, Sugiyama M, Garnett R eds. *Advances in Neural Information Processing Systems*. Montreal, Canada, 2018: 527-538
- [23] Titsias M K, Lázaro-Gredilla M. Spike and slab variational inference for multi-task and multiple kernel learning//Cortes C, Lee D D, Lawrence N D, Sugiyama M, Garnett R eds. *Advances in Neural Information Processing Systems*. Granada, Spain, 2011: 2339-2347
- [24] Yu S, Yu K, Tresp V, et al. Collaborative ordinal regression // *Proceedings of the 23rd International Conference on Machine Learning*. Pennsylvania, USA, 2006: 1089-1096
- [25] Yu K, Tresp V, Schwaighofer A. Learning Gaussian processes from multiple tasks//*Proceedings of the 22nd International Conference on Machine Learning (ICML-05)*. Bonn, Germany, 2005: 1012-1019
- [26] Luo J, Gupta A, Ong Y S, et al. Evolutionary optimization of expensive multiobjective problems with co-sub-Pareto front Gaussian process surrogates. *IEEE Transactions on Cybernetics*, 2018, 49(5): 1708-1721
- [27] Wang H, Jin Y. A random forest-assisted evolutionary algorithm for data-driven constrained multiobjective combinatorial optimization of trauma systems. *IEEE Transactions on Cybernetics*, 2020, 50(2): 536-549
- [28] Yeung S, Man K. Multiobjective optimization. *IEEE Microwave Magazine*, 2011, 12(6): 120-133
- [29] Zhang Y, Yeung D Y. Multi-task warped Gaussian process for personalized age estimation//*Proceedings of the 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. San Francisco, USA, 2010: 2622-2629
- [30] Zhang Y, Yeung D Y. Multi-task warped Gaussian process for personalized age estimation//*Proceedings of the 2010 IEEE Computer Society Conference on Computer Vision & Pattern Recognition*. San Francisco, USA, 2010: 2622-2629

- [31] Agrawal R B, Deb K, Agrawal R B. Simulated Binary Crossover for Continuous Search Space. *Complex Systems*, 2000, 9(3): 115-148
- [32] Press W H, Flannery B P, Teukolsky S A, et al. *Numerical Recipes in C: The Art of Scientific Computing*. Cambridge, UK: Cambridge University Press, 1993
- [33] Agrawal R B, Deb K. Simulated binary crossover for continuous search space. *Complex Systems*, 2000, 9(3): 115-148
- [34] Storn R, Price K. Differential evolution—A simple and efficient

heuristic for global optimization over continuous space. *Journal of Global Optimization*, 1997, 11(4): 341-359

- [35] Tian Y, Cheng R, Zhang X, et al. PlatEMO: A Matlab platform for evolutionary multi-objective optimization. *IEEE Computational Intelligence Magazine*, 2017, 12(4): 73-87
- [36] Zitzler E, Thiele L, Laumanns M, et al. Performance assessment of multiobjective optimizers: An analysis and review. *IEEE Transactions on Evolutionary Computation*, 2003, 7(2): 117-132



CAI Xin-Ye, Ph. D. , associate professor. His research interests include optimization, machine learning and their applications.

MA Zhong-Yu, M. S. candidate. His research interests include and DIMA modeling and algorithm optimization.

ZHANG Feng, M. S. candidate. His research interests include expensive multi-objective optimization and machine learning.

LI Nan, M. S. candidate. Her research interests include

expensive multi-objective optimization and machine learning.

CHENG Hui-Lin, M. S. candidate. Her research interests include adversarial examples and optimization.

SUN Qi, M. S. candidate. His research interests include multi-objective optimization and bilevel optimization.

XIAO Yu-Shun, M. S. candidate. His research interests include metrics of multi-objective optimization and machine learning.

LI Xiao-Ping, Ph. D. , professor. His research interests include cloud computing, resource scheduling optimization, service computing, big data, cloud manufacturing and machine learning.

Background

Expensive multi-objective optimization problem is a common type of problem in engineering. Based on the multi-objective optimization algorithm framework, it is a very popular method to use machine learning to establish an agent model to solve expensive multi-objective optimization problems. These surrogate models include Gaussian processes, neural networks and random forests. The Gaussian process model is widely used to solve expensive optimization problems because it is relatively simple to construct the model and can provide the uncertainty estimation of the model for each prediction point.

This paper proposes an Adaptive Multi-task with Multipopulation-based Cooperative Search algorithm (AMMCS) for solving expensive multi-objective optimization problems. Compared with other existing algorithms, the algorithm has been improved in two aspects: first, AMMCS measures the correlation among sub-problems by using a similarity indicator on the evaluated solution set, and then dynamically clusters

the sub-problems into multiple related tasks. Based on the dynamically divided tasks, we establish a multi-task Gaussian process model. This improvement makes AMMCS have better model prediction quality and algorithm performance. Second, this paper proposes a Multipopulation-based collaborative search method to optimize these tasks at the same time and achieve batched sampling. As a result, the overall computational clock time is reduced. The examination results reveal that the algorithm has better diversity and convergence than the other six algorithms on most test problems in ZDT and DTLZ.

This work was supported in part by the National Natural Science Foundation of China under Grant Nos. 62072234, 61300159, 61732006, 61876075 and 61832004, by the Natural Science Foundation of Jiangsu Province of China under Grant No. BK20181288, by the China Postdoctoral Science Foundation under Grant No. 2015M571751 and by the Aeronautical Science Foundation of China under Grant No. 20175552042.