

基于Kriging模型的改进型NSGA-III解决昂贵优化问题

耿焕同, 宋飞飞, 周征礼, 徐小涵

引用本文

耿焕同, 宋飞飞, 周征礼, 徐小涵. 基于Kriging模型的改进型NSGA-III解决昂贵优化问题[J]. 计算机科学, 2023, 50(7): 194-206.

GENG Huantong, SONG Feifei, ZHOU Zhengli, XU Xiaohan. [Improved NSGA-III Based on Kriging Model for Expensive Many-objective Optimization Problems](#) [J]. Computer Science, 2023, 50(7): 194-206.

相似文章推荐 (请使用火狐或 IE 浏览器查看文章)

Similar articles recommended (Please use Firefox or IE to view the article)

[一种结合标签分类和语义查询扩展的文本素材推荐方法](#)

Text Material Recommendation Method Combining Label Classification and Semantic QueryExpansion
计算机科学, 2023, 50(1): 76-86. <https://doi.org/10.11896/jsjcx.220100078>

[基于差分进化算法的字符对抗验证码生成方法](#)

Adversarial Character CAPTCHA Generation Method Based on Differential Evolution Algorithm
计算机科学, 2022, 49(11A): 211100074-5. <https://doi.org/10.11896/jsjcx.211100074>

[基于DE-lightGBM模型的上市公司高送转预测实证研究](#)

Empirical Study on the Forecast of Large Stock Dividends of Listed Companies Based on DE-lightGBM
计算机科学, 2022, 49(11A): 211000017-7. <https://doi.org/10.11896/jsjcx.211000017>

[基于DE-LSTM模型的教育统计数据预测研究](#)

Study on Prediction of Educational Statistical Data Based on DE-LSTM Model
计算机科学, 2022, 49(6A): 261-266. <https://doi.org/10.11896/jsjcx.220300120>

[一种基于切比雪夫距离的隐式偏好多目标进化算法](#)

Hidden Preference-based Multi-objective Evolutionary Algorithm Based on Chebyshev Distance
计算机科学, 2022, 49(6): 297-304. <https://doi.org/10.11896/jsjcx.210500095>

基于 Kriging 模型的改进型 NSGA-III 解决昂贵优化问题

耿焕同 宋飞飞 周征礼 徐小涵

南京信息工程大学计算机学院 南京 210044

摘要 在许多实际的优化问题中,为了进行适应度评估,其物理实验或数值仿真代价高昂,这给大多数现有的多目标进化算法(EAs)带来了巨大挑战。因此,文中提出了一种基于克里金模型辅助的改进参考点引导进化的优化算法,用于解决昂贵的超多目标优化问题。具体而言,根据种群的空间分布特征,借助关联点的熵差信息筛选参考点引导进化,以达到探索与开发的平衡。所提出的代理辅助进化算法(SAEA)使用克里金法来逼近每个目标函数,而无需进行原始昂贵的函数评估从而降低了计算成本。模型管理中采用一种纯指标填充采样准则,借助收敛性、多样性指标确定适当采样策略并使用昂贵目标函数对采样解进行真实评估以提升种群收敛与算法优化的效率。对具有 3 个以上目标的 80 个 DTLZ 与 WFG 基准测试问题进行了对比研究,证明了此算法的有效性和可行性。

关键词: 昂贵耗时问题;进化算法;代理辅助多目标优化;Kriging 模型;模型管理

中图法分类号 TP183

Improved NSGA-III Based on Kriging Model for Expensive Many-objective Optimization Problems

GENG Huantong, SONG Feifei, ZHOU Zhengli and XU Xiaohan

School of Computer Science, Nanjing University of Information Science & Technology, Nanjing 210044, China

Abstract In many real world multi-objective optimization problems(MOP), the cost of physical experiments or numerical simulations for fitness evaluation is very expensive, which poses a great challenge to most existing multi-objective evolutionary algorithms(EAs). Therefore, this paper proposes an improved reference point guided evolution optimization algorithm assisted by Kriging model to solve expensive many-objective optimization. Specifically, according to the distribution characteristics of the target spatial population, the reference points are selected to guide the evolution of the population to reach the balance of exploration and exploitation. The proposed surrogate-assisted evolution algorithm(SAEA) uses Kriging method to approximate each objective function without the need for the original expensive function evaluation to reduce computational cost. In model management, an infill sampling criterion is adopted to improve population convergence and algorithm optimization efficiency by evaluating convergence and diversity indexes to determine the appropriate sampling strategy for re-evaluation with expensive objective functions. The effectiveness and superiority of the proposed algorithm are proved by the empirical research on the benchmark problems with more than three objectives.

Keywords Expensive and time-consuming problems, Evolutionary algorithm, Surrogate-assisted multi-objective optimization, Kriging model, Model management

1 引言

许多实际优化问题都有多个相互冲突的目标需要同时优化,这被称为多目标优化问题(Multi-objective Optimization Problems, MOPs)。具有 4 个或更多目标的 MOP 可以进一步细分为超多目标优化问题(Many-objective Optimization Problem, MaOP),给优化方法带来了额外的挑战^[1]。在过去的几十年里,进化算法(Evolutionary Algorithm, EA)由于其自身的一些优点被广泛应用于多目标工程设计优化^[2]。

例如,EA 不需要分析目标函数或约束就能够在一次运行中获得一组非支配解^[3],各种多目标优化算法(Multi-objective Optimization Evolutionary Algorithm, MOEA)已被提出并证明在广泛的 MOPs 上表现良好^[4]。然而,由于选择压力较小,它们在处理具有 3 个以上目标的超多目标优化问题时性能急剧退化,主要归因于“维数灾难”,即随着目标数量的增加有限群体中非支配解呈指数级增长,使得传统的基于 Pareto 支配的 MOEAs 无法分辨这些解^[5]。大量的研究工作致力于为超多目标优化问题及其应用设计 MOEAs^[6],例如使用

到稿日期:2022-06-21 返修日期:2022-11-12

基金项目:国家自然科学基金(51977100)

This work was supported by the National Natural Science Foundation of China(51977100).

通信作者:耿焕同(htgeng@nuist.edu.cn)

基于参考点的非支配排序多目标进化优化算法 (NSGA-III)^[7]、基于超体积(HV)的进化算法^[8]、拐点驱动的进化算法^[9]、改进的多目标优化双档算法(Two_Arch2)^[10]以及基于参考向量的进化算法(RVEA)^[5]。值得注意的是,大多数现有的用于解决 MaOPs 的进化优化算法通常需要数以万计的适应度评价(Function Evaluations,FEs)。因此过多的函数求值评价显然使它们不适用于每次适应度评估都需要代价昂贵的模拟计算,例如熔炉优化^[2]、机翼设计^[11]和原油蒸馏装置的设计^[12]等问题,在这些设计中每进行一次实验评估都需要消耗代价高昂的金钱或时间成本,因此昂贵多目标优化问题只能承受少量的函数评估。由于评估预算有限,这种昂贵的 MOP 给 MOEA 带来了巨大的挑战。尽管近年来对超多目标优化的研究激增,但很少有人考虑到为了在合理时间内获得解,函数评估的数量将受到严格限制。因此,数据驱动的代理辅助进化算法 (Surrogate-assist Evolutionary Algorithm, SAEA)已成为 MOEAs 中减少昂贵函数评估的常用方法。通常,SAEA 构造代理模型近似原始昂贵的目标函数,然后使用代理模型评估部分候选解从而降低计算成本。为了提高预测性能,在评估过程中需采样新样本进行模型管理并通过在真实目标函数上的评估更新代理模型^[13-16]。

本文致力于开发一种高效的 SAEA,在不过多降低进化算法性能的前提下解决计算量大的昂贵超多目标优化问题。限制现有算法适用于多目标优化的主要原因之一是缺乏适合所用 MOEA 的有效代理管理方法。在 SAEA 管理代理模型时,个体的选择应同时考虑收敛性和多样性。为了选择这样的个体,代理模型需合理地嵌入到 MOEA 中。大多数现有的 SAEA 都是基于支配的,不适用于处理超多目标。因此,本文的主要贡献是提出了一种有效的算法来管理代理以处理大量的目标。为此,我们采用利用熵差信息筛选参考点引导种群进化的 ARNSGA-III 以及基于收敛性与多样性指标的填充采样准则进行模型管理,以达到种群进化过程探索与开发的平衡。为了限制克里金模型训练的计算时间,提出了一种选择训练样本的策略,以固定训练数据的最大数目。

2 相关研究及存在的问题

2.1 问题定义

在这项工作中,我们以最小化问题为例,考虑以下昂贵的高维多目标和多目标优化问题。

$$\min_x f(\mathbf{x})=(f_1(\mathbf{x}),f_2(\mathbf{x}),\cdots,f_m(\mathbf{x}))$$
$$\text{s. t. } \mathbf{x}\in X$$

(1)

其中, $\mathbf{x}=(x_1,x_2,\cdots,x_d)$ 是包含 d 个决策变量的决策向量, X 表示决策空间,目标向量由 $m(m>2)$ 个目标组成。决策空间中所有最优解的集合称为帕累托集合 (Pareto set,PS);目标空间中所有最优解的目标值集合称为帕累托前沿 (Pareto front,PF),也是 PS 在目标空间中映射的投影。对于超多目标优化问题,目标数 m 大于 3。此外,目标函数当作黑盒,可以通过耗时的数值模拟或昂贵的物理实验来进行评估。因此,必须根据由数值模拟实验收集的数据来建立代理模型以解决优化问题。

2.2 代理辅助进化算法

对于昂贵优化问题,一种流行的方法是引入计算效率高的代理模型来近似昂贵的原始目标函数^[17]。为了将代理模型嵌入到 MOEA 框架中以解决昂贵的 MOP,各类算法相继被提出,如表 1 所列。传统的多目标优化算法在代理模型辅助的优化中被视为优化器,用于提供选择压力,推动种群向帕累托前沿靠拢。这些优化器大致可分为 3 类:1)基于 Pareto 支配;2)基于指标;3)基于聚合的 MOEA^[18]。在第一类优化器中,根据帕累托支配选择候选解。例如,基于分类的预选 MOEA(CPS-MOEA)^[19]采用非支配排序方案,以选择下一代的解。第二类优化器使用一个指标作为其适应度函数来评估候选解^[20]。例如,基于 S 度量选择的高效全局优化(SMS-EGO)^[21]使用基于超容积(HV)的 S 度量来选择解,进行昂贵原始函数的评估。第三类优化器是 SAEA 中最常用的,它使用参考向量将多目标优化问题分解为多个单目标问题。例如基于帕累托的高效全局优化(ParEGO)^[22]、高效全局优化辅助的 MOEA/D(MOEA/DEGO)^[23]和克里金辅助的参考向量指导的进化算法(KRVEA)^[24]。此外,一些 SAEA 将上述优化器进行了融合,例如混合代理辅助多目标 EA(HSMEA)^[25]使用聚合优化器和基于指标的优化器进行选择。这些 SAEAs 在减少昂贵的超多目标优化函数评估数量方面被证明是有前景的^[26]。

表 1 相关 SAEA 特性的概括

Table 1 Summary of relevant characteristics of SAEA

算法	代理模型	优化器	填充采样准则
ParEGO	EGO(GP)	基于聚合的 MOEA	期待提升准则(EI)
MOEA/D-EGO	EGO(GP)	MOEA/D	期待提升准则(EI)
KRVEA	Kriging(GP)	RVEA	角度惩罚距离(APD)
KTA2	Kriging(GP)	Two_Arch2	基于 I_+ 指标
CSEA	FNN	RSEA	基于分类模型与参考点策略
EDN-ARMOEA	DropoutNN	ARMOEA	k-means 聚类与基于 I_+ 指标

在 SAEA 中,为了平衡搜索过程中的探索与开发,需要有效的模型管理。具体来说,模型管理表示为采样新解的过程,其中采样解的标准被称为填充采样标准或加点准则,不是直接优化昂贵的原始目标函数,而是在代理模型近似的廉价的辅助函数上进行优化即获取函数,它使用现有信息以便推荐下一个采样点,使用原始昂贵函数评估从而更新模型。有效的模型管理通常依赖于一个精心设计的获取函数和优化方法来有效识别其最佳状态。SAEA 在解决一个昂贵问题时获得的解包括所有使用真实昂贵函数评估的解。目前已经提出许多用于解决单目标问题的获取函数,如期待提升准则(EI)^[27]、置信下限(LCB)^[28]和概率改进准则(PoI)^[29]。然而,对于超多目标优化问题,加点准则的设计尤其具有挑战。这是因为,为获取一组在不同目标间权衡的解方案,获取函数应在收敛性和多样性之间取得良好的平衡。值得注意的是,在基于信息论的获取函数中,如用于多目标优化的预测熵搜索^[30]或最大值熵搜索^[31]中没有明确考虑目标之间的权衡,导致获得的解集收敛性和多样性之间不平衡。

2.3 存在的问题

尽管在 MOEA 中使用代理技术已经提出了许多算法,但仍然存在挑战。一个是代理模型的选择,SAEA 中可以使用各种机器学习的代理技术,如克里金模型(也称高斯过程回归)、人工神经网络、支持向量机、响应面法和径向基函数网络等。然而,不同的代理模型适用于不同类型的昂贵问题,没有简单的规则来确定合适类型的模型可以逼近给定的计算代价高昂的目标或约束函数。本文使用克里金模型作为代理近似目标函数,以满足超多目标问题对计算精度的需求。第二个挑战是构建代理模型的计算成本,这在 SAEA 中经常被忽略,在实践中训练代理可能需要大量时间。实验中,我们固定训练数据最大数目以提高算法效率。第三个挑战是如何更新代理模型,即如何在当前种群中采样合适解,使用原始函数进行重新评估。文献[32]中存在选择个体的不同方法,例如根据代理选择一组最佳解或非支配解,并选择一组代表性解。如果使用克里金模型,可以选择最佳预期改进^[27]、概率改进^[29]和超容改进^[33]的解,来重新评估个体的选择,其也被称为更新标准或填充标准。对于代理模型管理,填充采样标准将直接影响更新模型的近似精度,这也正是本文此项工作的努力方向。

此外,MOEA 在解决超多目标优化问题时遇到各种困难,如 Pareto 支配在处理高维目标空间时缺乏选择压力。在高维目标空间中,大量个体对其他个体无法形成支配关系,因此在第一支配层级中,总有更多非支配个体超过下次迭代的需求,这种情况会导致搜索陷入困境;随着目标维度的增加,在解决 MaOPs 时,实现收敛和多样性之间的良好平衡变得特别困难;最后大量解表示的最优权衡面将使得决策者难以理解和选择可利用的更优解。

如上文所述,处理昂贵超多目标优化问题的主要困难是需在搜索过程探索和开发之间取得平衡以及所获得解集收敛性与多样性之间的良好平衡。因此,代理辅助的超多目标优化算法的关键在于取得种群个体收敛性与多样性间的微妙平衡。对此,我们将根据收敛性或多样性指标,从两个优化过程得到的非支配解方案中选择最有希望的候选解进入模型管理,以更新模型并提升精度。

2.4 Kriging 模型

本文使用克里金模型,也称为高斯过程回归,用于近似每个目标函数。根据对计算成本高昂的超多目标优化问题的研究,相比 RBF 等人工神经网络,克里金模型常被用于代理技术,主要是因为其模型拟合精度较高且提供近似值的不确定信息可以在模型管理中展现良好效果^[17]。

克里金模型将单个目标函数值近似为:

$$y(x) = \mu(x) + \epsilon(x), \epsilon(x) \sim N(0, \sigma^2) \quad (2)$$

其中, μ 是回归模型 $F(\beta, x)$ 的预测,即 $\mu = F\beta$; $\epsilon(x)$ 是零均值和标准偏差 σ 的高斯分布。回归模型 $F(\beta, x) = \beta_1 g_1(x) + \dots + \beta_l g_l(x)$ 是 l 个选定函数与系数 β 的线性组合。

为降低昂贵评估代价,从式(2)中获取目标函数近似值,需要使用训练样本(SAEA 中的预评估个体)对克里金模型进行训练。让矩阵 $\mathbf{X} = [x^1, \dots, x^{N_I}]^T$ 表示决策空间中的训练

数据,其对应的目标向量 $\mathbf{y} = [y^1, \dots, y^{N_I}]^T$, 其中 $i = 1, 2, \dots, N_I$ 表示样本数的大小。此外,注意到 $\mathbf{X}$ 的大小是 $N_I \times n$, 其中 n 是决策变量的数目,即 $i = 1, 2, \dots, N_I$, 有 $x^i = [x_1, \dots, x_n]$ 。

对于任意两个输入 x^i 和 x^j , 两个随机过程 $\epsilon(x^i)$ 和 $\epsilon(x^j)$ 之间的协方差有:

$$\text{cov}[\epsilon(x^i), \epsilon(x^j)] = \sigma^2 \mathbf{R}([R(x^i, x^j)]) \quad (3)$$

其中, $\mathbf{R}$ 是 $N_I \times N_I$ 的相关矩阵。

$$\mathbf{R} = \begin{bmatrix} R(x^1, x^2) & \dots & R(x^1, x^{N_I}) \\ \vdots & \ddots & \vdots \\ R(x^{N_I}, x^1) & \dots & R(x^{N_I}, x^{N_I}) \end{bmatrix} \quad (4)$$

并且 $R(x^i, x^j)$ 是 $\epsilon(x^i)$ 和 $\epsilon(x^j)$ 之间的相关函数。通常,相关函数为:

$$R(x^i, x^j) = \exp\left(-\sum_{k=1}^n \theta_k |x_k^i - x_k^j|^2\right) \quad (5)$$

其中, θ_k 表示超参数,并且近似方差 σ^2 由式(6)给出, θ 值可由最大似然函数得到:

$$\begin{cases} \sigma^2 = \frac{1}{N_I} (\mathbf{y} - \mathbf{F}\beta)^T \mathbf{R}^{-1} (\mathbf{y} - \mathbf{F}\beta) \\ \phi(\theta) = -\frac{1}{2} (N_I \ln \sigma^2 + \ln \det(\mathbf{R})) \end{cases} \quad (6)$$

其中, $\det(\mathbf{R})$ 是矩阵 $\mathbf{R}$ 的行列式。在获得超参数 θ 后,分别计算系数 β 和方差 σ^2 , 进一步使用它们来近似目标函数值。

尽管克里金模型由于具有传递不确定信息的能力而成为一种使用广泛的替代模型,但代理模型训练的计算复杂性是其潜在的严重缺陷。如文献[17]训练克里金模型的计算复杂度为 $O(n^3)$, 其中 n 是训练样本数。如果超参数 θ 是使用优化算法最大似然函数来确定的,那么计算复杂度将变得更高,此过程通常是在克里金辅助 SAEA 中完成的。例如,MOEA/D-EGO 使用差分进化, SMS-EGO 使用协方差矩阵自适应来优化超参数。

本文中,由于使用基于种群的技术来优化超参数并不实际且模型需要经常重复训练使得计算时间非常长,因此我们使用了 K-RVEA 中 Kriging 模型的改进算法,算法在 DACE 工具箱中实现。

3 CDK-ARNSGA-III

3.1 K-ARNSGA-III

ARNSGA-III^[34] 是一种新型参考点选择策略的改进型 NSGA-III 算法^[7], 为解决原始 NSGA-III 算法参考点划分难以确定且算法在不规则 Pareto 前沿适应性差等问题,提出了利用决策空间中粒子的分布特征信息筛选超平面参考点的策略。其主要思想是运用统计学概念计算决策空间内粒子四分位差,用于判断进化状态,进而根据目标空间中粒子的分列特征信息筛选参考点,从而引导种群进化方向。算法 1 给出了 ARNSGA-III 的具体步骤。

ARNSGA-III 为种群提供个体选择压力,迫使种群更好地向 PF 收敛,此外改进的参考点选择策略有助于更好地引导种群的空间分布,为 Kriging 的模型管理提供收敛与多样的支撑。因此将 ARNSGA-III 作为优化器与 Kriging 模型

结合,采用形成的 K-ARNSGA-III 为基底以求不过多降低算法性能且大大减少算法原始昂贵评价次数,从而达到高效求解昂贵超多目标优化问题。

为了判断种群进化阶段,ARNSGA-III 采用每次迭代进化种群的熵值之差进行判定,其中熵值计算式如下:

$$e' = - \sum_{i=1}^n \inf_i \times \lg \inf_i - \sum_{i=1}^n \Delta mid_i' \times \lg \Delta mid_i' \quad (7)$$

则相邻两代种群熵差为:

$$\Delta e' = |e' - e^{t-1}| \quad (8)$$

当种群更新时, $\Delta e'$ 数值与决策空间中个体的变化幅度呈正相关。 $\Delta e'$ 越大说明种群变化越大,种群目前经历探索阶段并且算法将种群的探索方向引导向不确定区域;反之, $\Delta e'$ 越小说明个体变化越小,种群趋于稳定并不断收敛。因此, $\Delta e'$ 将判别种群经历何种进化状态 S' ,其中 t 为迭代代数, μ 为阈值。

(1)若 $|\Delta e'| \geq \mu$,则说明种群位于“探索”阶段,正经历潜在未知空间的探索;

(2)若 $|\Delta e'| < \mu$,则说明种群位于“探究”阶段,正不断向 Pareto 前沿收敛。

式(9)计算标准化四分位差以表示分布情况:

$$\overline{inf} = \frac{0.75E - 0.25E}{E} = \frac{1}{2} \quad (9)$$

选取 $\overline{inf}$ 的改变小于 $1/N$ 且种群没有偏移即 $\Delta mid \approx 0$ 时,通过上述计算 $|\Delta e'|$ 为阈值 μ 。在思考阈值 μ 和决策空间维数间的线性关系后,设计

$$\mu = -D \overline{inf} \lg \overline{inf} - D \left(\overline{inf} + \frac{1}{N} \right) \lg \left(\overline{inf} + \frac{1}{N} \right) \quad (10)$$

其中, N 是种群规模, D 是决策空间维数。

算法 1 ARNSGA-III

- Step 1 输入最大迭代次数 $G_{\max}$,种群规模 N ,设置约束边界 ub, lb ,设定当前进化代数 $t=0$ 。
- Step 2 种群 P 初始化,确定初始参考点集合 Z 并计算阈值 μ , S' 表示进化状态默认为“探索”,初始化关联参考点集数量 $Z_{\text{sum}}=0$ 。
- Step 3 判断若 S' 是“探索”状态,则种群中个体与参考点进行关联操作,并修改 Z_{sum} 。
- Step 4 判断 S' 若是“探究”状态,参考点数目 Z 与种群数目 N 不相等,则通过 Z_{sum} 中记录的各参考点关联数目删除 Z 中最少 $Z-N$ 个参考点。
- Step 5 依据参考点 Z 种群 P 迭代进化, $t=t+1$ 。
- Step 6 计算进化阶段。
- Step 7 如果 $t < G_{\max}$,则跳到 step3; 否则终止迭代,输出最终解。

3.2 基于收敛性与多样性双指标加点准则

模型管理对 SAEA 的成功至关重要,它主要关注如何使用和更新代理模型,包括使用原始目标函数选择需重新评估的个体。然后,这些重新评估的个体可以用作更新代理的训练数据。在选择要重新评估的个体时,必须同时考虑收敛性和多样性,这对于具有大量目标的问题来说变得更加困难。本节将重点放在采样点的选择上,在给定大量目标的情况下对收敛性和多样性进行管理,这也是本文的主要贡献之一。

基于帕累托的纯指标填充抽样标准的想法来源于进化

算法普遍采用的性能评价指标,即大多数现有的改进算法及抽样标准都是为了保持多样性和促进种群收敛而设计的。采样具有良好收敛性和多样性的有希望解可以提高代理模型开发与探索的性能。所提策略的主要思想是可以将填充采样任务描述为二目标问题,引入收敛性指标与多样性指标作为分别衡量实验解收敛与分布效益的两个目标,并依据目标引导采样符合标准的有希望解进入真实函数评估,以更新模型不断提高种群近似精确度。

此项工作面临的一个挑战是如何衡量与优化两个既定目标,首先需要考虑促进种群收敛到真实 Pareto 前沿。理想的衡量收敛的方法是计算实验解与真实 Pareto 前沿的距离以判定收敛程度,然而直接计算个体与超平面间的距离在实现上是非常困难的。因此,我们引入了平面内的理想点用于计算与实验解之间的欧几里得距离 (Euclidean Distance, ED),来更好地接近上述理想方法。理想点是由实验解和当前种群中可用的最佳目标函数值构成的一个阶段性最佳点,随着种群的进化,理想点也随之进一步优化。如果具有最大欧氏距离的解未被档案中的任何解所支配并且其至少支配了档案中的一个解,那么这个解将成为候选解,否则将选择档案中最接近理想点的非支配解。此外,收敛性衡量是一个最小化目标,距离值越小,实验解就越接近真实 Pareto 前沿。

受 KTA2^[35] 启发,算法使用实验解与其最近父解之间的距离作为多样性指标,以测量维持多样性的提升。多样性指标旨在提升种群在一个超平面内的均匀分布,考虑到实验解离父代解距离越远表示该区域越稀疏,因此设计算法将探索此潜在未知的稀疏解空间从而提升多样性。在进化过程前期,此设计不仅可以保持多样性,还可以加速种群的收敛。此外,由于预期多样性衡量指标将最大化,因此本文利用负值将问题统一转换为最小化指标。算法将双指标筛选出的实验解进行非支配排序,并根据种群集合的 Pareto 支配关系选择最高层级的非支配解进行原始昂贵的函数评估。

根据上述设计思想,基于收敛性与多样性的纯指标填充采样策略的详细步骤如下:

- (1) 将代理辅助评估的实验解进行归一化处理;
- (2) 将种群父代与子代决策变量中的最小值视为理想点;
- (3) 将与父代相距较远的个体 x^* 保存于多样性衡量 Dvs 档案中, x^* 满足:

$$Dvs(x^*) = \max_{p=1, N} \{ |x^*, x_p|^2 \} \quad (11)$$

- (4) 计算实验解与理想点间欧氏距离并将其保存于收敛性衡量 Cvg 档案中,将 Cvg 与 Dvs 中保存的个体合并为新集合 newObj;

- (5) 应用非支配排序筛选集合 newObj 中第一前沿的解;
- (6) 通过原始昂贵函数对筛选解进行评估以更新模型。

3.3 算法的总体流程

本文提出了一种基于 Kriging 模型辅助的进化算法 (CDK-ARNSGA-III),用于解决昂贵的超多目标优化问题。此设计中的关键驱动因素是假设每个函数评估的计算成本高昂,该假设的含义是与可能需要数小时或数天的真实函数评估相比,此算法“开销”可被认为是微不足道的,例如重组/

选择算子、代理模型的训练、代理模型的局部搜索等可能只需要数秒或几分钟,因此在搜索过程中可以进行的原始真实函数评估的数量设定是有限的。在 CDK-ARNSGA-III 中,我们采用 ARNSGA-III 作为优化器,进一步采用新的模型管理策略分别考虑收敛性和多样性的性能指标提升。CDK-ARNSGA-III 算法大致可分为以下 6 个主要阶段:

(1)初始化。这一阶段首先在决策空间中抽取初始训练数据,初始个体通过原始昂贵目标函数进行评估。如果对优化问题有一些先验知识,则可以对这个问题使用特定的初始化方法,否则采用拉丁超立方抽样^[36]方法来生成初始训练数据。此外,初始数据被用作进化的初始种群。

(2)建立替代模型。在使用代理模型的阶段,我们使用克里金模型代替原始昂贵函数来拟合目标函数,对于昂贵的优化任务, $f_1(x), f_2(x), \dots, f_m(x)$ 这 m 个目标的评估是昂贵的,因此我们为每个目标函数建立一个替代模型。使用的 Kriging 模型不仅会对个体做近似目标函数的评估,还会对应地给出此目标的不确定性信息,正因为此优势使 Kriging 模型成为了广泛流行的代理辅助模型。

(3)代理辅助评估。在这一阶段代理模型被视为嵌入式 MOEA 中的近似目标函数,以评估若干迭代的子代返回评估个体的近似目标值及目标方向不确定性信息以辅助新解采样,例如使用替代模型而不是昂贵的目标函数进行算法 1 中的循环操作。需要注意的是,除使用不同的评估函数外,其余步骤与 ARNSGA-III 均相同。

(4)子代生成。为了能在现有模型的迭代中不断生成更优秀的个体,在每一代的近似评估过程中使用多项式变异和模拟二进制交叉算子以相同的概率生成子代解。

(5)填充采样。通过上述阶段的进化,获得了 N 个由替代模型评估的实验解,填充采样进一步从 N 个实验解中筛选

出多个解,用于昂贵的函数评估。在这一阶段,我们采用上述收敛性与多样性双指标作为填充采样标准即加准则,在获得一组实验解后,先计算其相应的潜在收敛性和多样性,再应用非支配排序筛选第一个前沿面的解,并通过昂贵函数对筛选出的解进行评估以更新模型。

(6)环境选择。当通过昂贵函数评估的解数量增加时,执行环境选择,从当前种群和新评估的解中选择 N 个解。该阶段要求环境选择能够将种群从 $N + \lambda$ 减少到 N ,其中 λ 是新评估解的数量。

第(2)–(5)阶段将重复执行,直到达到昂贵函数评估的最大数量终止循环。

结合上述工作,基于 Kriging 模型的 ARNSGA-III 算法的具体步骤如算法 2 所示,算法流程如图 1 所示,左侧为所提算法的主体架构,右侧子框架为 ARNSGA-III 算法作为优化器近似优化种群。

算法 2 CDK-ARNSGA-III

- 输入:最大昂贵函数评估 FE_{max} ,种群规模 N
输出:种群 P_{FE} 中的非支配解
- Step 1 使用拉丁超立方采样生成初始训练数据 P_{FE} ,初始化函数评估; $FE=N$ 。
 - Step 2 为每个原始昂贵目标函数训练 Kriging 模型。
 - Step 3 当 $FE \leq FE_{max}$ 时,循环 Step4—Step9。
 - Step 4 记 $P_t = P_{FE}$; $t=1$ 。
 - Step 5 当 $t < t_{max}$ 时,循环 Step6—Step7。
 - Step 6 代理模型辅助评估实验解。
 - Step 7 选择解 P_{t+1} 作为父类个体进入下一次迭代。
 - Step 8 使用收敛性与多样性双指标填充采样准,则从 $P_{t_{max}}$ 中采样理想个体使用原始真实函数评估更新模型。
 - Step 9 环境选择 P_{FE} 。
 - Step 10 终止。

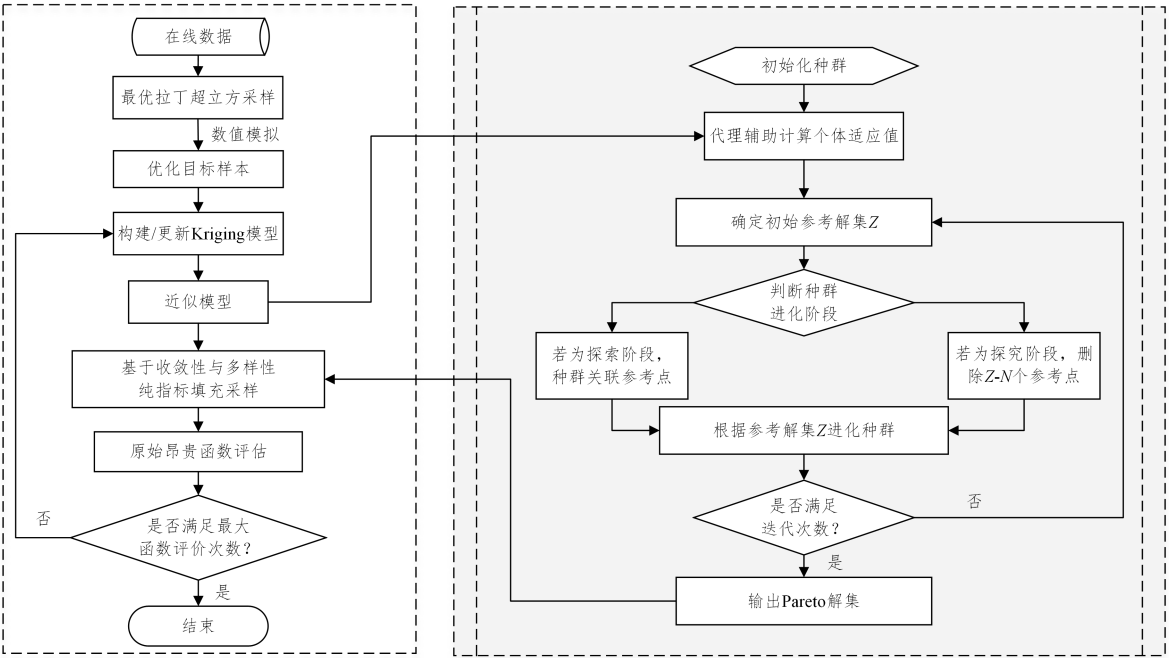


图 1 CDK-ARNSGA-III 算法的流程框架
Fig. 1 Framework of CDK-ARNSGA-III

3.4 时间复杂度分析

设目标数为 M , 决策变量数为 D , 种群规模为 N , 构建一个克里金模型的复杂度为 $O(N^3)$, NSGA-III 最坏时间复杂度为 $O(N^2 \log^{M-2} N)$ 或 $O(MN^2)$, 改进的 ARNSGA-III 增加熵差计算的时间复杂度为 $O(DN \lg N)$ 。此外, 优化器内关联操作与原始 NSGA-III 相同, 未增加多余算法开销。综上所述, CDK-ARNSGA-III 中每轮迭代最差时间复杂度为 $\max\{O(MN^3 + N^2 \log^{M-2} N), O(MN^3 + MN^2), O(MN^3 + DN \lg N)\}$, 总体计算复杂度仍为种群规模的立方级。

4 实验结果分析

本节将 CDK-ARNSGA-III 与一些代表性 SAEA 进行对比, 并将实验环境、参数设置及实验结果等进行简要阐述。在计算效率高的基准问题(如 DTLZ 和 WFG)上测试而不直接在昂贵的实际问题测试 SAEA 是一种常见做法, 与测试其他进化算法的一个主要区别是, 该算法仅允许使用目标函数非常有限数量的函数来进行评估, 以满足目标评估在计算上昂贵的假设。

4.1 实验环境及测试问题

为了验证所提 CDK-ARNSGA-III 的有效性, 将其与一些经典的以及近些年新提出的 SAEA 算法进行比较, 包括 ABSAEA^[37], KRVEA^[24], CSEA^[38], EDN-ARMOEA^[39] 和 HeE-MOEA^[40]。ABSAEA 使用一种自适应贝叶斯方法, 根据搜索动态调整获取函数中的超参数, 以确定使用原始昂贵函数评估的候选解。KRVEA 使用参考向量引导的 EA(RVEA) 作为优化器。它使用固定数量的训练数据为每个目标函数建立克里金模型, 当多样性退化时选择平均 STD 最大解作为填充样本, 否则选择具有最佳 APD 值的解进行重新评估。CSEA 是一种基于分类的 SAEA, 它使用一个单一的 FNN 模型来预先选择可能好的解进行重新评估。EDNARMOEA 使用计算高效的 Dropout 神经网络替代高斯过程, 以降低变量维数对模型构建的时间消耗。HeEMOEA 是一类多代理模型集成的 SAEA, 该算法提出了一种基于最小二乘支持向量机和两个径向基函数网络的异构集成方法, 并在原始决策变量的基础上选取决策变量的子集和一组转换后的变量作为异构集成的输入, 进一步提高了集成的多样性。

本文选取 DTLZ 和 WFG 测试函数集中不同目标数的 80 个标准测试实例, 可涵盖各种类型优化问题。测试目标数为

3, 4, 6, 8, 10, 对于 DTLZ 问题, 决策变量数设为 10; 而对于目标为 3, 4, 6, 8 和 10 的 WFG 问题, 决策变量数分别设为 10, 10, 9, 9 和 11。实验运行于一个开源多目标优化算法测试平台 PlatEMO^[41]。

4.2 参数设置

- (1) 算法在每个测试实例中独立运行 30 轮。
- (2) 初始数据大小和种群大小设置为 100。
- (3) 昂贵函数评估的最大数量设置为 300。
- (4) 用于建立克里金模型的计算机实验设计与分析工具箱(DACE)超参数范围设置为 $[10^{-5}, 10^5]$ 。
- (5) 种群使用多项式变异模拟二进制交叉算子, 分布指数设置为 20, 交叉变异概率分别设为 0.9 和 $1/d$, 其中 d 为决策变量维数, 相关参数设置与 ARNSGA-III 均相同。
- (6) 收敛性与多样性标准由欧氏距离衡量。

4.3 性能指标

为了评估比较算法在这些测试实例上的性能, 本文选择了反转世代距离(IGD)与超体积(HV)两个广泛使用的性能评价指标。IGD 与 HV 均可同时评估所得最终解集收敛性与分布性以衡量算法的有效性。

(1) 反转世代距离(IGD)。采用 Pareto 最优解集 PF_{true} 中的个体到算法所求得的非支配解集 PF^* 的平均距离表示。因此, 计算式为:

$$IGD(PF_{\text{true}}, PF^*) = \frac{\sum_{v \in PF_{\text{true}}} d(v, PF^*)}{|PF_{\text{true}}|}$$

(12)

其中, PF_{true} 是真实 Pareto 前沿中一组均匀分布的解, PF^* 是所求得非支配解集, v 是 PF_{true} 中的个体, $d(v, PF^*)$ 表示 v 到 PF^* 的最短距离。IGD 值的大小与算法性能成负相关。

(2) 超体积(HV)。通过计算非支配解集与参考点围成空间的超体积值实现对多目标优化算法性能的评价, 计算式为:

$$HV = \delta(U_{i=1}^{|S|} v_i)$$

(13)

其中, δ 表示贝格测度, v_i 表示参考点和非支配个体所构成的超体积, S 表示非支配集。HV 值与算法性能成正相关。

4.4 实验分析

算法在比较中获得的 IGD 值结果如表 2 所列, 其中以粗体突出显示最佳结果。为得到科学公平的对比结果, 我们使用显著性水平为 0.05 的非参数统计 Wilcoxon 秩和检验的方法与其他算法进行对比。表中“+”“-”“=”符号分别表示算法显著优于、显著劣于和与 CDK-ARNSGA-III 无明显差别。

表 2 在 DTLZ 测试集上各算法的 IGD 均值及标准差
Table 2 IGD mean and standard deviation of each algorithm on DTLZ test set

Problem	M	ABSAEA	CSEA	EDNARMOEA	KRVEA	HeEMOEA	CDKARNSGAIII
DTLZ1	3	8.3514×10 ⁺¹ (2.24×10 ⁺¹)—	5.5467×10 ⁺¹ (1.40×10 ⁺¹)=	1.0214×10 ⁺² (2.10×10 ⁺¹)—	9.2636×10 ⁺¹ (1.87×10 ⁺¹)—	1.0911×10 ⁺² (2.21×10 ⁺¹)—	5.4926×10⁺¹ (1.93×10⁺¹)
	4	5.7834×10 ⁺¹ (1.37×10 ⁺¹)—	4.2827×10⁺¹ (1.09×10 ⁺¹)+	7.4630×10 ⁺¹ (1.62×10 ⁺¹)—	6.4371×10 ⁺¹ (1.46×10 ⁺¹)—	7.4602×10 ⁺¹ (1.43×10 ⁺¹)—	4.9933×10 ⁺¹ (1.14×10 ⁺¹)
	6	2.6605×10 ⁺¹ (8.58)=	1.8800×10⁺¹ (4.29)+	3.8403×10 ⁺¹ (9.04)—	3.3797×10 ⁺¹ (7.49)—	3.4897×10 ⁺¹ (1.40×10 ⁺¹)—	2.8110×10 ⁺¹ (9.94)
	8	6.4587(3.51)=	4.4600(2.57)+	9.9094(3.37)—	1.0130×10 ⁺¹ (3.98)—	7.2775(3.62)—	5.9351(3.12)
	10	2.4806×10⁻¹ (6.78×10⁻²)+	3.1458×10 ⁻¹ (6.73×10 ⁻²)=	4.4499×10 ⁻¹ (1.62×10 ⁻¹)—	3.8193×10 ⁻¹ (1.13×10 ⁻¹)=	3.9258×10 ⁻¹ (1.10×10 ⁻¹)=	3.3446×10 ⁻¹ (1.20×10 ⁻¹)

(续表)							
Problem	M	ABSAEA	CSEA	EDNARMOEA	KRVEA	HeEMOEA	CDKARNSGAIII
DTLZ2	3	1.3239×10 ⁻¹ (3.08×10 ⁻²)—	2.2052×10 ⁻¹ (2.88×10 ⁻²)—	3.0474×10 ⁻¹ (3.19×10 ⁻²)—	1.6897×10 ⁻¹ (4.14×10 ⁻²)—	1.7427×10 ⁻¹ (1.09×10 ⁻²)—	5.9987×10 ⁻² (4.27×10 ⁻³)
	4	2.0715×10 ⁻¹ (3.10×10 ⁻²)—	2.9238×10 ⁻¹ (2.70×10 ⁻²)—	3.6322×10 ⁻¹ (1.57×10 ⁻²)—	2.4095×10 ⁻¹ (5.03×10 ⁻²)—	2.4042×10 ⁻¹ (1.91×10 ⁻²)—	1.4880×10 ⁻¹ (8.53×10 ⁻³)
	6	3.2674×10 ⁻¹ (1.65×10 ⁻²)—	4.5596×10 ⁻¹ (3.55×10 ⁻²)—	4.1975×10 ⁻¹ (1.32×10 ⁻²)—	3.4299×10 ⁻¹ (2.08×10 ⁻²)—	4.2875×10 ⁻¹ (1.81×10 ⁻²)—	3.0812×10 ⁻¹ (1.63×10 ⁻²)
	8	3.8762×10 ⁻¹ (9.58×10 ⁻³)—	5.8737×10 ⁻¹ (3.19×10 ⁻²)—	4.5530×10 ⁻¹ (1.54×10 ⁻²)—	4.1699×10 ⁻¹ (2.89×10 ⁻²)—	5.8345×10 ⁻¹ (2.27×10 ⁻²)—	3.7034×10 ⁻¹ (1.01×10 ⁻²)
	10	4.6641×10 ⁻¹ (1.04×10 ⁻²)—	6.7750×10 ⁻¹ (2.47×10 ⁻²)—	4.5798×10 ⁻¹ (1.18×10 ⁻²)—	5.6361×10 ⁻¹ (3.47×10 ⁻²)—	6.5220×10 ⁻¹ (1.98×10 ⁻²)—	4.3366×10 ⁻¹ (6.12×10 ⁻³)
	3	2.3844×10 ⁺² (6.28×10 ⁺¹)—	1.6413×10 ⁺² (4.01×10 ⁺¹)=	3.2257×10 ⁺² (5.32×10 ⁺¹)—	2.2734×10 ⁺² (4.23×10 ⁺¹)—	3.1346×10 ⁺² (5.49×10 ⁺¹)—	1.5026×10 ⁺² (5.38×10 ⁺¹)
	4	1.6922×10 ⁺² (4.78×10 ⁺¹)—	1.3538×10 ⁺² (3.53×10 ⁺¹)=	2.4745×10 ⁺² (6.26×10 ⁺¹)—	1.8432×10 ⁺² (2.87×10 ⁺¹)—	2.2613×10 ⁺² (5.29×10 ⁺¹)—	1.3272×10 ⁺² (5.50×10 ⁺¹)
	6	7.4078×10 ⁺¹ (2.71×10 ⁺¹)=	5.6883×10 ⁺¹ (2.10×10 ⁺¹)+	1.3290×10 ⁺² (3.59×10 ⁺¹)—	9.0161×10 ⁺¹ (2.86×10 ⁺¹)=	1.1056×10 ⁺² (3.14×10 ⁺¹)—	8.3195×10 ⁺¹ (3.59×10 ⁺¹)
	8	2.0592×10 ⁺¹ (1.20×10 ⁺¹)=	1.9038×10 ⁺¹ (7.88)=	3.5230×10 ⁺¹ (1.06×10 ⁺¹)—	3.1809×10 ⁺¹ (1.40×10 ⁺¹)—	2.8330×10 ⁺¹ (1.12×10 ⁺¹)—	1.8792×10 ⁺¹ (1.25×10 ⁺¹)
	10	9.7722×10 ⁻¹ (2.26×10 ⁻¹)+	1.0550 (2.84×10 ⁻¹)=	1.4065 (4.41×10 ⁻¹)=	1.3506 (4.09×10 ⁻¹)=	1.2137 (2.89×10 ⁻¹)+	1.2977 (5.94×10 ⁻¹)
DTLZ4	3	3.4284×10 ⁻¹ (8.90×10 ⁻²)=	4.7034×10 ⁻¹ (1.16×10 ⁻¹)—	2.2598×10 ⁻¹ (9.58×10 ⁻²)+	4.2161×10 ⁻¹ (9.05×10 ⁻²)—	8.0333×10 ⁻¹ (7.55×10 ⁻²)—	3.4928×10 ⁻¹ (9.92×10 ⁻²)
	4	4.1217×10 ⁻¹ (8.67×10 ⁻²)=	4.6648×10 ⁻¹ (1.07×10 ⁻¹)=	4.4148×10 ⁻¹ (1.09×10 ⁻¹)=	4.9060×10 ⁻¹ (7.60×10 ⁻²)—	8.7212×10 ⁻¹ (8.54×10 ⁻²)—	4.3569×10 ⁻¹ (9.15×10 ⁻²)
	6	5.7425×10 ⁻¹ (6.32×10 ⁻²)=	4.9326×10 ⁻¹ (5.38×10 ⁻²)+	4.9825×10 ⁻¹ (4.05×10 ⁻²)+	5.4277×10 ⁻¹ (4.79×10 ⁻²)=	8.6781×10 ⁻¹ (6.27×10 ⁻²)—	5.4434×10 ⁻¹ (5.60×10 ⁻²)
	8	5.7909×10 ⁻¹ (3.87×10 ⁻²)—	5.7918×10 ⁻¹ (2.56×10 ⁻²)—	5.1134×10 ⁻¹ (1.95×10 ⁻²)+	5.7581×10 ⁻¹ (2.99×10 ⁻²)—	8.1032×10 ⁻¹ (3.88×10 ⁻²)—	5.4654×10 ⁻¹ (5.04×10 ⁻²)
	10	6.3828×10 ⁻¹ (2.68×10 ⁻²)—	6.3460×10 ⁻¹ (2.94×10 ⁻²)—	5.3472×10 ⁻¹ (1.33×10 ⁻²)+	5.9091×10 ⁻¹ (2.73×10 ⁻²)—	7.8380×10 ⁻¹ (2.33×10 ⁻²)—	5.6309×10 ⁻¹ (2.95×10 ⁻²)
	3	8.6638×10 ⁻² (2.01×10 ⁻²)—	1.0438×10 ⁻¹ (2.08×10 ⁻²)—	1.6534×10 ⁻¹ (3.13×10 ⁻²)—	1.1510×10 ⁻¹ (4.42×10 ⁻²)—	1.1964×10 ⁻¹ (1.78×10 ⁻²)—	1.1967×10 ⁻² (1.60×10 ⁻³)
	4	8.4901×10 ⁻² (2.57×10 ⁻²)—	1.2250×10 ⁻¹ (1.94×10 ⁻²)—	1.6625×10 ⁻¹ (2.43×10 ⁻²)—	8.1133×10 ⁻² (3.29×10 ⁻²)—	1.0489×10 ⁻¹ (1.95×10 ⁻²)—	3.1097×10 ⁻² (6.50×10 ⁻³)
	6	4.2749×10 ⁻² (1.44×10 ⁻²)=	7.6261×10 ⁻² (2.05×10 ⁻²)—	1.1984×10 ⁻¹ (1.78×10 ⁻²)—	3.2379×10 ⁻² (7.25×10 ⁻³)+	7.5365×10 ⁻² (7.77×10 ⁻³)—	4.0825×10 ⁻² (8.39×10 ⁻³)
	8	1.8156×10 ⁻² (4.09×10 ⁻³)+	4.2363×10 ⁻² (8.85×10 ⁻³)=	5.7352×10 ⁻² (6.68×10 ⁻³)—	2.1309×10 ⁻² (6.18×10 ⁻³)+	4.4662×10 ⁻² (7.18×10 ⁻³)=	4.0774×10 ⁻² (8.77×10 ⁻³)
	10	9.4060×10 ⁻³ (1.77×10 ⁻³)+	1.4221×10 ⁻² (1.24×10 ⁻³)+	1.5290×10 ⁻² (1.96×10 ⁻³)+	1.0979×10 ⁻² (1.74×10 ⁻³)+	1.2613×10 ⁻² (1.89×10 ⁻³)+	1.9381×10 ⁻² (2.91×10 ⁻³)
DTLZ6	3	3.3802 (3.33×10 ⁻¹)—	5.0604 (7.13×10 ⁻¹)—	5.4167 (4.41×10 ¹)—	3.2697 (3.43×10 ⁻¹)=	6.7031 (1.06×10 ⁻¹)—	2.9900 (5.87×10 ⁻¹)
	4	2.5507 (4.61×10 ⁻¹)+	5.0279 (4.21×10 ⁻¹)—	4.9774 (3.05×10 ⁻¹)—	2.3055 (3.46×10 ⁻¹)+	5.8241 (8.82×10 ⁻²)—	2.8778 (4.43×10 ⁻¹)
	6	1.8545 (3.06×10 ⁻¹)+	3.1003 (5.74×10 ⁻¹)—	3.4121 (3.18×10 ⁻¹)—	1.2695 (3.61×10 ⁻¹)+	4.1092 (9.59×10 ⁻²)—	2.3914 (3.90×10 ⁻¹)
	8	8.2822×10 ⁻¹ (2.10×10 ⁻¹)+	1.5992 (3.64×10 ⁻¹)—	1.7996 (2.16×10 ⁻¹)—	5.3812×10 ⁻¹ (2.21×10 ⁻¹)+	2.3941 (7.84×10 ⁻²)—	1.1686 (2.97×10 ⁻¹)
	10	1.8617×10 ⁻¹ (9.63×10 ⁻²)—	1.0433×10 ⁻¹ (7.84×10 ⁻²)=	3.0928×10 ⁻¹ (1.21×10 ⁻¹)—	5.4949×10 ⁻² (2.24×10 ⁻²)+	7.1088×10 ⁻¹ (3.64×10 ⁻²)—	1.0536×10 ⁻¹ (3.84×10 ⁻²)
	3	2.7078×10 ⁻¹ (2.08×10 ⁻¹)=	1.6892 (5.95×10 ⁻¹)—	1.2316 (4.60×10 ⁻¹)—	1.3369×10 ⁻¹ (1.39×10 ⁻²)=	5.3946 (5.44×10 ⁻¹)—	3.0283×10 ⁻¹ (2.69×10 ⁻¹)
	4	7.3347×10 ⁻¹ (2.71×10 ⁻¹)—	2.8816 (6.81×10 ⁻¹)—	1.5282 (4.95×10 ⁻¹)—	3.0705×10 ⁻¹ (3.57×10 ⁻²)+	7.2109(1.08)—	5.9737×10 ⁻¹ (2.52×10 ⁻¹)
	6	1.3632 (3.36×10 ⁻¹)—	5.4606 (2.14×10 ⁺⁰)—	1.5263 (5.21×10 ⁻¹)—	6.2061×10 ⁻¹ (4.69×10 ⁻²)+	9.0065(1.51)—	1.1235 (4.91×10 ⁻¹)
	8	1.7990 (6.43×10 ⁻¹)=	6.6641 (2.50×10 ⁺⁰)—	1.7578 (5.43×10 ⁻¹)=	8.6816×10 ⁻¹ (7.36×10 ⁻²)+	9.0552(2.90)—	1.6525 (3.20×10 ⁻¹)
	10	1.4868 (2.31×10 ⁻¹)+	2.0881 (4.08×10 ⁻¹)—	1.3306 (1.25×10 ⁻¹)+	1.1518 (5.18×10 ⁻²)+	2.2584 (4.22×10 ⁻¹)—	1.7458 (2.77×10 ⁻¹)
+/-/=		8/17/10	6/20/9	6/26/3	11/18/6	2/31/2	

表 2 列出了各算法在 DTLZ 问题上的比较结果,从表中可以看出在 35 个 DTLZ 测试问题上,CDK-ARNSGAIII 取得了 13 次最佳结果,其次为 KRVEA,取得了 11 次最佳结果但总体差距不大。DTLZ1 和 DTLZ3 由于具有多峰特性,算法很难收敛,而 DTLZ2 算法很容易收敛,但需要保持多样性。可以观察到,本文算法在 DTLZ2 测试问题上几乎都取得了最佳结果。DTLZ4 是一个敏感问题,其很难获得分布均匀的初始训练数据。DTLZ5-7 具有不规则的 Pareto 前沿,这对多样

性维护构成了挑战。本文算法对 DTLZ5-7 问题优化效果较差的原因是,优化器 ARNSGA-III 中使用的熵差选择参考点策略虽然可以保证在不规则 Pareto 前沿问题上的良好性能,但仍存在不足。在 DTLZ 测试问题中,CDK-ARNSGA-III 与其他算法对比虽无明显压倒性优势,但实际性能展现出的效果仍可接受。

为了进一步分析算法的效果,我们选择 WFG 问题作为测试问题,在 WFG 问题中 WFG1 的变换函数使得 MOEA 很难实现良好的多样性;WFG2 是一个断开连接的问题,其 Pareto 前沿不规则;WFG3 将 WFG2 断开的 Pareto 前沿替换为连续前沿面;WFG4 是多式联合,对 MOEA 来说很难实现融合;WFG5 是一个欺骗问题;WFG6 和 WFG9 都是不可分离的简化问题;WFG7 是一个可分离的单峰问题;WFG8 是一个

不可分离问题,使算法很难保持多样性。因此,与 DTLZ 问题相比,在 WFG 问题上保持较优解更加困难。表 3 列出了各算法在 WFG 问题上的对比结果,从表 3 可以明显看出,本文算法在 WFG 上有良好效果且与其他算法对比优势更为明显。图 2 直观展示了本文算法与其他算法在 WFG8 问题上均匀分布的效果对比,图 3 为各算法在 300 次函数评估中向前沿面收敛的趋势图,灰色网格表示 WFG8 测试问题的真实 Pareto 前沿面,灰色点表示种群个体。从图 3 看出,与其他算法相比,CDK-ARNSGA-III 算法内较多解正不断向真实 Pareto 前沿靠拢,并在收敛时仍能保持前沿面的均匀分布。此外,本实验利用箱形图以分析各算法在可计算 HV 指标测试函数上的样本统计结果,如图 4 所示。其中,箱形上下线表示四分位数,内部中线为中位数,外部上下线表示最大最小值。

表 3 在 WFG 测试集上各算法的 IGD 均值及标准差
Table 3 IGD mean and standard deviation of each algorithm on WFG test set

Problem	M	ABSAEA	CSEA	EDNARMOEA	KRVEA	HeEMOEA	CDKARNSGAIII
WFG1	3	1.8069 (1.18×10 ⁻¹) +	1.7387 (7.79×10 ⁻²) +	1.9270 (5.59×10 ⁻²) =	1.8153 (1.23×10 ⁻¹) +	2.3688 (5.67×10 ⁻²) -	1.9347 (1.21×10 ⁻¹)
	4	2.0931 (9.81×10 ⁻²) +	1.9835 (8.19×10 ⁻²) +	2.1515 (4.73×10 ⁻²) =	2.1188 (9.00×10 ⁻²) =	2.5430 (4.33×10 ⁻²) -	2.1829 (1.48×10 ⁻¹)
	6	2.4701 (1.02×10 ⁻¹) =	2.4218 (1.25×10 ⁻¹) =	2.5147 (3.42×10 ⁻²) -	2.4896 (5.42×10 ⁻²) =	2.8634 (3.49×10 ⁻²) -	2.4188 (1.41×10 ⁻¹)
	8	2.8216 (1.43×10 ⁻¹) -	2.6609 (2.95×10 ⁻¹) =	2.8598 (6.19×10 ⁻²) -	2.8152 (8.25×10 ⁻²) -	3.2090 (5.61×10 ⁻²) -	2.6089 (2.25×10 ⁻¹)
	10	3.1467 (2.04×10 ⁻¹) -	2.9171 (3.40×10 ⁻¹) -	3.1956 (9.38×10 ⁻²) -	3.1005 (2.49×10 ⁻¹) -	3.4553 (3.78×10 ⁻²) -	2.7196 (2.89×10 ⁻¹)
	3	3.5848×10 ⁻¹ (6.09×10 ⁻²) -	5.7401×10 ⁻¹ (7.98×10 ⁻²) -	6.5641×10 ⁻¹ (5.22×10 ⁻²) -	3.8578×10 ⁻¹ (5.85×10 ⁻²) -	8.3097×10 ⁻¹ (6.57×10 ⁻²) -	2.5474×10 ⁻¹ (3.32×10 ⁻²)
	4	5.8000×10 ⁻¹ (1.03×10 ⁻¹) -	7.6951×10 ⁻¹ (1.17×10 ⁻¹) -	9.0539×10 ⁻¹ (7.74×10 ⁻²) -	5.5484×10 ⁻¹ (7.67×10 ⁻²) -	1.2691 (1.54×10 ⁻¹) -	4.1057×10 ⁻¹ (3.53×10 ⁻²)
	6	6.7135×10 ⁻¹ (5.94×10 ⁻²) -	1.2435 (4.20×10 ⁻¹) -	1.1156 (1.60×10 ⁻¹) -	6.8132×10 ⁻¹ (7.68×10 ⁻²) -	2.2679 (3.71×10 ⁻¹) -	6.4395×10 ⁻¹ (4.67×10 ⁻²)
	8	1.0519 (2.97×10 ⁻¹) =	1.9494 (7.18×10 ⁻¹) -	1.2944 (1.18×10 ⁻¹) -	9.9845×10 ⁻¹ (5.95×10 ⁻²) =	3.3762 (5.32×10 ⁻¹) -	1.0220 (2.97×10 ⁻¹)
	10	1.2497 (9.18×10 ⁻²) =	2.7433 (8.18×10 ⁻¹) -	1.7208 (2.00×10 ⁻¹) -	1.2486 (1.07×10 ⁻¹) =	4.5109 (7.28×10 ⁻¹) -	1.2486 (9.28×10 ⁻²)
WFG3	3	3.5656×10 ⁻¹ (5.39×10 ⁻²) -	4.9348×10 ⁻¹ (6.03×10 ⁻²) -	6.1465×10 ⁻¹ (2.91×10 ⁻²) -	5.2276×10 ⁻¹ (5.34×10 ⁻²) -	4.2887×10 ⁻¹ (2.19×10 ⁻²) -	2.5915×10 ⁻¹ (3.73×10 ⁻²)
	4	4.9490×10 ⁻¹ (8.42×10 ⁻²) -	6.0320×10 ⁻¹ (7.43×10 ⁻²) -	7.6285×10 ⁻¹ (3.19×10 ⁻²) -	6.5614×10 ⁻¹ (6.59×10 ⁻²) -	5.3664×10 ⁻¹ (2.28×10 ⁻²) -	4.3469×10 ⁻¹ (7.10×10 ⁻²)
	6	4.6520×10 ⁻¹ (7.73×10 ⁻²) -	5.6397×10 ⁻¹ (1.13×10 ⁻¹) -	8.6749×10 ⁻¹ (6.30×10 ⁻²) -	4.6934×10 ⁻¹ (1.17×10 ⁻¹) =	6.7894×10 ⁻¹ (4.27×10 ⁻²) -	4.2295×10 ⁻¹ (6.97×10 ⁻²)
	8	5.1386×10 ⁻¹ (8.92×10 ⁻²) -	4.5460×10 ⁻¹ (7.63×10 ⁻²) =	6.8226×10 ⁻¹ (7.39×10 ⁻²) -	4.7258×10 ⁻¹ (8.08×10 ⁻²) =	6.1867×10 ⁻¹ (7.31×10 ⁻²) -	4.4495×10 ⁻¹ (8.15×10 ⁻²)
	10	5.7920×10 ⁻¹ (8.84×10 ⁻²) -	5.9639×10 ⁻¹ (1.08×10 ⁻¹) -	8.4869×10 ⁻¹ (8.81×10 ⁻²) -	6.0680×10 ⁻¹ (1.04×10 ⁻¹) -	7.1309×10 ⁻¹ (9.42×10 ⁻²) -	4.8209×10 ⁻¹ (7.29×10 ⁻²)
	3	4.2304×10 ⁻¹ (2.03×10 ⁻²) +	4.6590×10 ⁻¹ (3.51×10 ⁻²) =	5.1317×10 ⁻¹ (1.72×10 ⁻²) -	4.9470×10 ⁻¹ (2.16×10 ⁻²) -	7.4110×10 ⁻¹ (7.16×10 ⁻²) -	4.5382×10 ⁻¹ (2.47×10 ⁻²)
	4	7.4921×10 ⁻¹ (2.53×10 ⁻²) -	9.1842×10 ⁻¹ (1.06×10 ⁻¹) -	8.7944×10 ⁻¹ (2.91×10 ⁻²) -	8.1599×10 ⁻¹ (2.87×10 ⁻²) -	1.5088 (9.31×10 ⁻²) -	7.1214×10 ⁻¹ (2.42×10 ⁻²)
	6	1.7861 (3.73×10 ⁻²) -	2.8886 (3.44×10 ⁻¹) -	1.8650 (5.19×10 ⁻²) -	1.8242 (1.31×10 ⁻¹) -	3.9968 (2.66×10 ⁻¹) -	1.6727 (7.39×10 ⁻²)
	8	3.4204 (2.89×10 ⁻¹) -	5.6953 (5.43×10 ⁻¹) -	3.2476 (1.26×10 ⁻¹) -	3.5033 (2.52×10 ⁻¹) -	7.2870 (3.30×10 ⁻¹) -	3.0871 (1.16×10 ⁻¹)
	10	5.3306 (4.64×10 ⁻¹) -	8.7587 (6.83×10 ⁻¹) -	5.1589 (2.06×10 ⁻¹) -	5.7120 (5.50×10 ⁻¹) -	1.0189e ⁺¹ (5.30×10 ⁻¹) -	4.7773 (2.76×10 ⁻¹)
WFG5	3	4.4653×10 ⁻¹ (7.25×10 ⁻²) -	5.3650×10 ⁻¹ (3.20×10 ⁻²) -	5.5014×10 ⁻¹ (3.30×10 ⁻²) -	3.6281×10 ⁻¹ (3.69×10 ⁻²) =	7.4901×10 ⁻¹ (1.55×10 ⁻¹) -	3.8424×10 ⁻¹ (1.51×10 ⁻¹)
	4	7.7334×10 ⁻¹ (4.93×10 ⁻²) =	8.7824×10 ⁻¹ (3.53×10 ⁻²) -	9.1788×10 ⁻¹ (2.31×10 ⁻²) -	7.3515×10 ⁻¹ (2.85×10 ⁻²) +	1.0978 (2.79×10 ⁻²) -	7.8938×10 ⁻¹ (1.15×10 ⁻¹)
	6	1.7516 (4.25×10 ⁻²) =	2.3851 (2.04×10 ⁻¹) -	1.9341 (7.98×10 ⁻²) -	1.7452 (9.21×10 ⁻²) =	2.2097 (7.48×10 ⁻²) -	1.7450 (1.13×10 ⁻¹)
	8	3.1478 (1.06×10 ⁻¹) -	4.6415 (3.30×10 ⁻¹) -	3.4094 (9.37×10 ⁻²) -	3.2744 (2.42×10 ⁻¹) -	3.8993 (1.27×10 ⁻¹) -	3.0971 (1.57×10 ⁻¹)
	10	4.7511 (3.79×10 ⁻¹) -	7.1052 (3.63×10 ⁻¹) -	5.2454 (2.13×10 ⁻¹) -	5.0256 (5.92×10 ⁻¹) -	5.6758 (2.24×10 ⁻¹) -	4.5004 (1.21×10 ⁻¹)

(续表)

Problem	M	ABSAEA	CSEA	EDNARMOEA	KRVEA	HeEMOEA	CDKARNSGAIII
WFG6	3	7.0231×10^{-1} $(4.53 \times 10^{-2})-$	7.0873×10^{-1} $(3.86 \times 10^{-2})-$	8.1339×10^{-1} $(2.97 \times 10^{-2})-$	7.1358×10^{-1} $(3.81 \times 10^{-2})-$	7.8912×10^{-1} $(2.44 \times 10^{-2})-$	5.8598×10^{-1} (9.89×10^{-2})
	4	1.1631 $(4.94 \times 10^{-2})-$	1.0821 $(5.51 \times 10^{-2})-$	1.1471 $(2.33 \times 10^{-2})-$	1.0396 $(2.13 \times 10^{-2})-$	1.3485 $(4.51 \times 10^{-2})-$	8.8007×10^{-1} (9.51×10^{-2})
	6	2.2727 $(1.28 \times 10^{-1})-$	2.4783 $(2.47 \times 10^{-1})-$	2.0849 $(4.08 \times 10^{-2})-$	2.1579 $(9.21 \times 10^{-2})-$	3.1116 $(1.43 \times 10^{-1})-$	1.7689 (4.74×10^{-2})
	8	3.7702 $(1.54 \times 10^{-1})-$	4.8605 $(5.02 \times 10^{-1})-$	3.4115 $(5.86 \times 10^{-2})-$	3.4066 $(9.65 \times 10^{-2})-$	5.7390 $(3.40 \times 10^{-1})-$	2.9903 (5.00×10^{-2})
	10	5.5258 $(3.54 \times 10^{-1})-$	7.0177 $(6.71 \times 10^{-1})-$	4.9551 $(1.28 \times 10^{-1})-$	4.9860 $(1.99 \times 10^{-1})-$	8.1555 $(3.68 \times 10^{-1})-$	4.5346 (7.56×10^{-2})
WFG7	3	5.5804×10^{-1} $(4.47 \times 10^{-2})+$	5.7243×10^{-1} $(4.17 \times 10^{-2})=$	6.5695×10^{-1} $(1.49 \times 10^{-2})-$	6.5023×10^{-1} $(2.56 \times 10^{-2})-$	6.0831×10^{-1} $(2.27 \times 10^{-2})-$	5.8533×10^{-1} (2.92×10^{-2})
	4	8.4403×10^{-1} $(3.95 \times 10^{-2})=$	9.8250×10^{-1} $(7.28 \times 10^{-2})-$	1.0006 $(1.53 \times 10^{-2})-$	9.8921×10^{-1} $(2.56 \times 10^{-2})-$	1.2022 $(6.41 \times 10^{-2})-$	8.3388×10^{-1} (3.55×10^{-2})
	6	2.1436 $(1.31 \times 10^{-1})-$	2.6091 $(3.13 \times 10^{-1})-$	2.0200 $(3.56 \times 10^{-2})-$	2.0698 $(9.98 \times 10^{-2})-$	3.1499 $(1.85 \times 10^{-1})-$	1.7251 (5.45×10^{-2})
	8	3.9912 $(2.81 \times 10^{-1})-$	5.3872 $(4.77 \times 10^{-1})-$	3.5164 $(6.56 \times 10^{-2})-$	3.8864 $(3.17 \times 10^{-1})-$	5.9492 $(3.29 \times 10^{-1})-$	3.2208 (7.83×10^{-2})
	10	6.3857 $(6.34 \times 10^{-1})-$	8.0520 $(4.95 \times 10^{-1})-$	5.1612 $(1.30 \times 10^{-1})-$	6.5564 $(6.95 \times 10^{-1})-$	8.6087 $(3.95 \times 10^{-1})-$	4.7337 (1.13×10^{-1})
WFG8	3	6.4333×10^{-1} $(5.61 \times 10^{-2})-$	7.5347×10^{-1} $(4.90 \times 10^{-2})-$	7.9913×10^{-1} $(2.68 \times 10^{-2})-$	6.6734×10^{-1} $(3.50 \times 10^{-2})-$	8.7565×10^{-1} $(3.43 \times 10^{-2})-$	5.0910×10^{-1} (4.49×10^{-2})
	4	1.2388 $(5.29 \times 10^{-2})-$	1.2348 $(4.33 \times 10^{-2})-$	1.2266 $(3.34 \times 10^{-2})-$	1.0980 $(3.64 \times 10^{-2})-$	1.4900 $(5.15 \times 10^{-2})-$	9.1259×10^{-1} (5.06×10^{-2})
	6	2.6327 $(1.70 \times 10^{-1})-$	2.9747 $(2.16 \times 10^{-1})-$	2.2914 $(4.67 \times 10^{-2})-$	2.4185 $(9.84 \times 10^{-2})-$	3.4388 $(1.73 \times 10^{-1})-$	2.0345 (3.25×10^{-2})
	8	4.5146 $(2.98 \times 10^{-1})-$	5.5159 $(3.87 \times 10^{-1})-$	3.7808 $(8.72 \times 10^{-2})-$	4.0391 $(2.11 \times 10^{-1})-$	6.1208 $(2.69 \times 10^{-1})-$	3.3952 (1.06×10^{-1})
	10	6.1334 $(3.96 \times 10^{-1})-$	8.1326 $(3.13 \times 10^{-1})-$	5.3184 $(9.05 \times 10^{-2})-$	6.1309 $(5.16 \times 10^{-1})-$	8.3978 $(3.58 \times 10^{-1})-$	4.8536 (1.61×10^{-1})
WFG9	3	5.9669×10^{-1} $(6.39 \times 10^{-2})+$	6.6025×10^{-1} $(8.81 \times 10^{-2})=$	7.6093×10^{-1} $(5.89 \times 10^{-2})-$	6.9212×10^{-1} $(5.56 \times 10^{-2})=$	7.0151×10^{-1} $(4.13 \times 10^{-2})-$	6.5122×10^{-1} (8.62×10^{-2})
	4	1.0121 $(1.00 \times 10^{-1})+$	1.1162 $(1.02 \times 10^{-1})=$	1.1192 $(5.76 \times 10^{-2})-$	1.1329 $(9.46 \times 10^{-2})-$	1.1610 $(6.15 \times 10^{-2})-$	1.0768 (9.24×10^{-2})
	6	2.0499 $(1.38 \times 10^{-1})+$	2.7962 $(2.10 \times 10^{-1})-$	2.2046 $(1.50 \times 10^{-1})=$	2.2401 $(1.73 \times 10^{-1})=$	2.6736 $(1.39 \times 10^{-1})-$	2.2087 (2.27×10^{-1})
	8	3.5869 $(3.96 \times 10^{-1})=$	5.2934 $(3.99 \times 10^{-1})-$	3.6474 $(2.62 \times 10^{-1})=$	4.2604 $(5.47 \times 10^{-1})-$	4.8051 $(2.71 \times 10^{-1})-$	3.7086 (3.65×10^{-1})
	10	5.8567 $(7.84 \times 10^{-1})=$	7.8619 $(4.85 \times 10^{-1})-$	5.8187 $(4.52 \times 10^{-1})-$	6.3387 $(5.90 \times 10^{-1})-$	6.9535 $(2.91 \times 10^{-1})-$	5.4647 (5.43×10^{-1})
+/-/=		7/30/8	2/36/7	0/41/4	2/33/10	0/45/0	

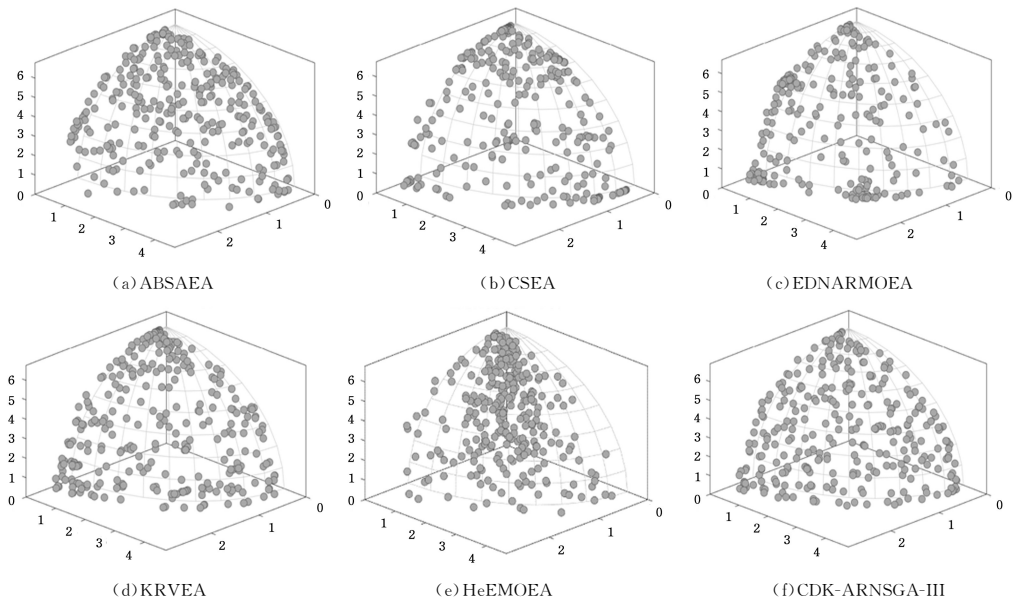


图 2 各算法在 WFG8 问题上的分布情况

Fig. 2 Distribution of each algorithm on WFG8 test problem

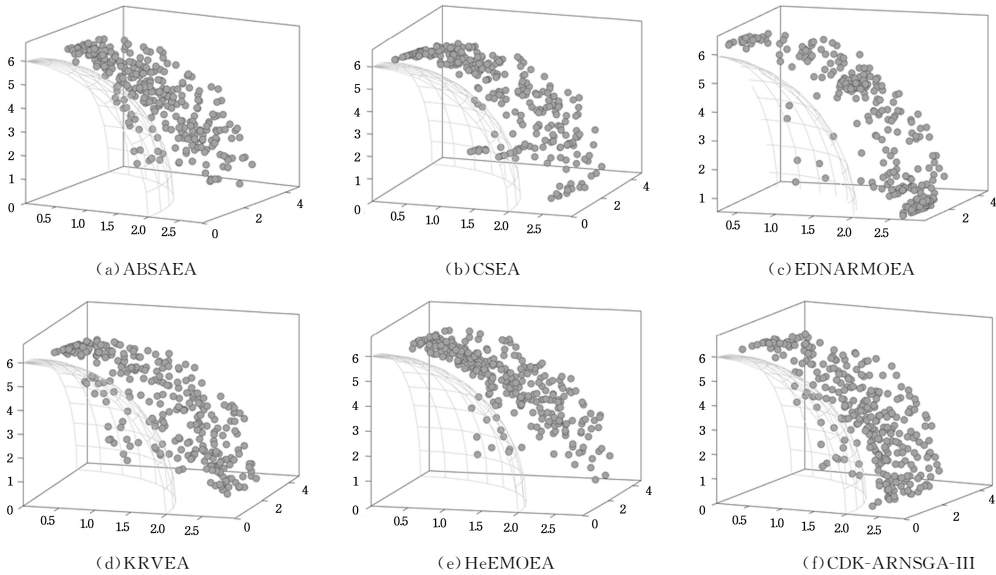


图 3 各算法在 WFG8 问题上的收敛情况

Fig. 3 Convergence of each algorithm on WFG8 test problem

其次,我们比较了 6 种算法的计算时间,表 4 列出了完成一次实验 $M=\{3,4,6,8,10\}$ 所消耗的平均计算时间。平均计算时间由表 2 和表 3 所列的所有实验结果计算。从表中可以看出,CSEA 执行速度最快,这是因为 CSEA 构建基于分类的模型直接预测解的支配关系能避免近似函数求值。HeE-MOEA 虽是多集合模型,但 SVM 与 RBFN 的训练时间短于 Kriging 模型,其运算效率更高。CDK-ARNSGA-III 运行

时间位于中间位置,与 KRVEA 虽同样使用 Kriging 模型但本文算法采用相对简洁的模型管理策略使得时间代价小于 KRVEA。ABSAEA 采用贝叶斯方法动态调整超参数,耗费了部分运算时间,EDN-ARMOEA 算法的运行时间相比其他算法上升了一个数量级,排在最末。作为一种普遍趋势,基于近似的 SAEA 往往会比基于分类的 SAEA 消耗更多的计算时间。

表 4 6 种算法的平均计算时间

Table 4 Average computational time of all six algorithms

(单位:s)

M	ABSAEA	CSEA	EDN-ARMOEA	KRVEA	HeE-MOEA	CDK-ARNSGAIII
3	$2.16\times10^{+1}$	$7.49\times10^{+1}$	$3.55\times10^{+2}$	$3.69\times10^{+1}$	$2.00\times10^{+1}$	$2.65\times10^{+1}$
4	$2.76\times10^{+1}$	$1.85\times10^{+1}$	$3.71\times10^{+2}$	$4.61\times10^{+1}$	$2.05\times10^{+1}$	$3.20\times10^{+1}$
6	$2.81\times10^{+1}$	7.32	$3.75\times10^{+2}$	$4.99\times10^{+1}$	$2.08\times10^{+1}$	$3.57\times10^{+1}$
8	$3.02\times10^{+1}$	5.53	$3.45\times10^{+2}$	$5.28\times10^{+1}$	$2.29\times10^{+1}$	$3.96\times10^{+1}$
10	$3.86\times10^{+1}$	5.23	$3.11\times10^{+2}$	$7.89\times10^{+1}$	$2.36\times10^{+1}$	$5.79\times10^{+1}$

最后,为综合了解以上不同 MOEAs 的性能,我们采用多问题分析法通过非参统计 Wilcoxon 秩和检验测试得到各算法的 IGD 指标、HV 指标以及 CPU 运行时间的平均排名,如表 5 所列,其中以粗体突出各项的最佳结果。

表 5 非参统计各算法 IGD 和 HV 指标以及计算时间平均排名

Table 5 Average rankings of HV,IGD and CPU time of each algorithm by nonparametric statistic Wilcoxon test

算法	IGD	HV	CPU time
	Ranking	Ranking	Ranking
ABSAEA	2.6445	2.5538	2.7428
CSEA	3.8318	3.7231	1.5428
EDNARMOEA	4.3889	4.5231	6.0000
K-RVEA	2.9841	2.8769	4.6571
HeEMOEA	5.3603	5.5384	1.9429
CDKARNSGAIII	1.8000	1.7846	4.0000

从表 5 可看出,CDK-ARNSGA-III 的 IGD 与 HV 指标在 6 种算法中取得了最好名次,虽然在 CPU 运行时间方面未

取得明显优势,但其良好的性能指标不容忽视。表 5 充分验证了本文算法在不过多增加计算开销的前提下仍能不断提升求解表现,以更好应对如实验模拟中复杂多样的昂贵优化问题。

从以上测试实例的实验结果可以看出,没有任何一个代理辅助进化算法能够成功处理所有类型的问题,从与上述算法的比较中看出 CDK-ARNSGA-III 取得了测试的最佳性能。在不同的问题上,优化状态会发生不同的变化。例如,在 WFG 问题的优化过程中经常出现收敛和分布需求状态,而 DTLZ2 经常出现分布需求状态,基于纯指标的采样实验解可以更好地满足问题的探索方向。因此,使用定制纯指标采样策略相比其他算法也更有竞争力。然而也可以观察到 CDK-ARNSGA-III 与其他算法相比在 DTLZ1 和 DTLZ6 以及 DTLZ7 上的性能并不理想,原因可能在于与 KRVEA 等利用参考向量保持收敛和分布不同,CDK-ARNSGA-III 的首要任务是提高收敛性,从而忽略了其他方面的平衡。

综上所述,CDK-ARNSGA-III 利用克里金模型近似目标

函数, 并采用双指标填充采样策略能够在不过多增加计算开销的情况下有效平衡算法的探索与开发。此外, 其通过种群

特征和对不规则前沿面的分析进行参考点筛选, 在应对各种测试问题中体现出了较好的鲁棒性。

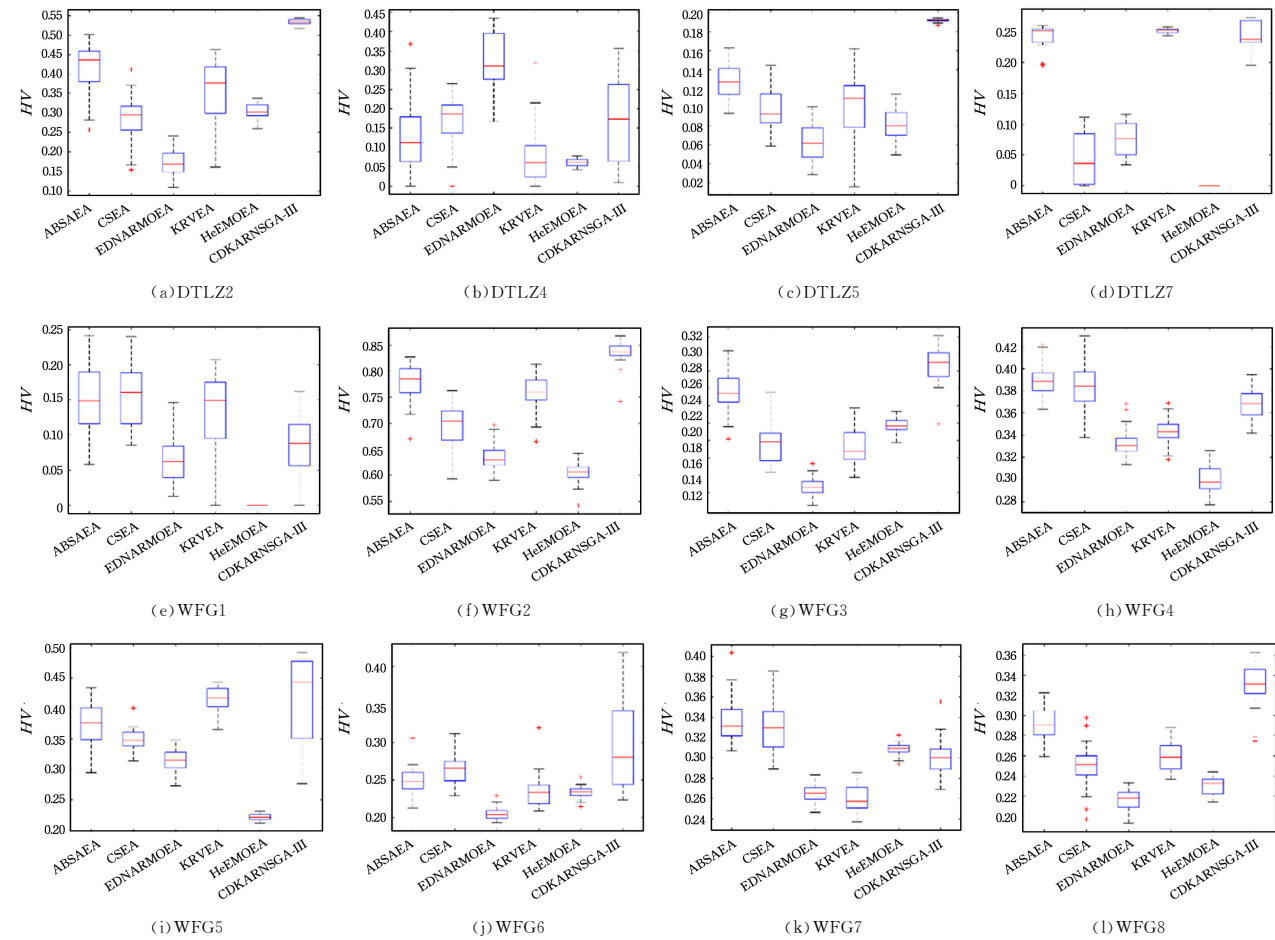


图 4 各算法的 HV 指标箱型图

Fig. 4 Box plots of HV metric values of each algorithm on 12 functions

结束语 本文提出了一种用于求解具有 3 个以上目标昂贵优化问题的 CDK-ARNSGA-III 算法, 其中克里金模型用于逼近每个目标函数。在选择原始昂贵函数重新评估的新采样个体时独立考虑收敛性、多样性作为填充采样准则, 分别促进种群的开发和探索。结果表明, 在本文研究的大多数测试实例上, CDK-ARNSGA-III 不仅优于上述对比算法而且计算效率更高, 同时熵差选择参考点使算法在处理不规则 Pareto 前沿问题中也显示出了良好效果。为了解决具有更多决策变量的优化问题, 将需要更多的训练数据, 这不仅需要更多的计算资源而且对基于克里金的代理技术提出了更严峻的挑战。尽管如此, 我们还是希望 SAEA 能够适用于具有更多决策变量的优化问题, 这将是我们的未来工作。解决此问题的一个可以继续探索的方法是使用其他回归模型代替克里金法, 如 RBFN 或 Dropout 神经网络。此外, 将代理模型的不确定性考虑到填充采样准则中也值得尝试。

参 考 文 献

[1] ISHIBUCHI H, TSUKAMOTO N, NOJIMA Y. Evolutionary manyobjective optimization: A short review[C] // 2008 IEEE Congress on Evolutionary Computation (IEEE World Congress

on Computational Intelligence). Hong Kong, China, 2008: 2419-2426.
[2] GUO D, CHAI T, DING J, et al. Small data- driven evolutionary multi-objective optimization of fused magnesium furnaces[C] // 2016 IEEE Symposium Series on Computational Intelligence (SSCI). 2016: 1-8.
[3] YANG S, LI M, LIU X, et al. A grid-based evolutionary algorithm for many-objective optimization[J]. IEEE Transactions on Evolutionary Computation, 2013, 17(5): 721-736.
[4] DEB K, PRATAP A, AGARWAL S, et al. A fast and elitist multiobjective genetic algorithm: NSGA-II[J]. IEEE Transactions on Evolutionary Computation, 2002, 6(2): 182-197.
[5] CHENG R, JIN Y, OLHOFFER M, et al. A reference vector guided evolutionary algorithm for many-objective optimization [J]. IEEE Transactions on Evolutionary Computation, 2016, 20(5): 773-791.
[6] LYGOE R J, CARY M, FLEMING P J. A real-world application of a many-objective optimization complexity reduction process [J]. Evolutionary Multi-Criterion Optimization, Berlin: Springer, 2013, 7811: 641-655.
[7] DEB K, JAIN H. An Evolutionary Many-Objective Optimization

- Algorithm Using Reference-Point-Based Nondominated Sorting Approach, Part I: Solving Problems with Box Constraints[J]. IEEE Transactions on Evolutionary Computation, 2014, 18(4): 577-601.
- [8] BADER J, ZITZLER E. HypE: An algorithm for fast hypervolume-based many-objective optimization[J]. Evolutionary Computation, 2011, 19(1): 45-76.
- [9] ZHANG X, TIAN Y, JIN Y. A knee point driven evolutionary algorithm for many-objective optimization[J]. IEEE Transactions on Evolutionary Computation, 2015, 19(6): 761-776.
- [10] WANG H, JIAO L, YAO X. Two_arch2: An improved two-archive algorithm for many-objective optimization[J]. IEEE Transactions on Evolutionary Computation, 2015, 19(4): 524-541.
- [11] JIN Y, SENDHOFF B. A systems approach to evolutionary multiobjective structural optimization and beyond[J]. IEEE Computational Intelligence Magazine, 2009, 4(3): 62-76.
- [12] OCHOA-ESTOPIERA L M, ENÍQUEZ-GUTIERREZA V M, CHEN L, et al. Industrial application of surrogate models to optimize crude oil distillation units[J]. Chemical Engineering Transactions, 2018, 69: 289-294.
- [13] JIN Y, WANG H, CHUGH T, et al. Data-driven evolutionary optimization: An overview and case studies[J]. IEEE Transactions on Evolutionary Computation, 2019, 23(3): 442-458.
- [14] SUN Z R, HUANG Y H, CHEN Z Y. Diversity based surrogate-assisted evolutionary algorithm for expensive multi-objective optimization problem[J]. Journal of Software, 2021, 32(12): 3814-3828.
- [15] ALAN D M, GREGORIO T, HUGO B, et al. A Review of Surrogate Assisted Multiobjective Evolutionary Algorithms[J]. Computational Intelligence and Neuroscience, 2016, 2016-6-12: 19.
- [16] LIN M M, ZHOU H, WANG L P. Research of Many-objective Evolutionary Algorithm Based on Alpha Dominance[J]. Computer Science, 2017, 44(1): 264-270.
- [17] JIN Y. Surrogate-assisted evolutionary computation: Recent advances and future challenges[J]. Swarm and Evolutionary Computation, 2011, 1(2): 61-70.
- [18] WAGNER T, BEUME N, NAUJOKS B. Pareto-, aggregation-, and indicator-based methods in many-objective optimization[J]. Evolutionary Multi-Criterion Optimization, 2007, 4403: 742-756.
- [19] ZHANG J, ZHOU A, ZHANG G. A classification and Pareto domination based multiobjective evolutionary algorithm[C]// 2015 IEEE Congress on Evolutionary Computation (CEC). 2015: 2883-2890.
- [20] SUN Y, YEN G, YI Z. IGD indicator-based evolutionary algorithm for many-objective optimization problems[J]. IEEE Transactions on Evolutionary Computation, 2019, 23(2): 173-187.
- [21] PONWEISER W, WAGNER T, BIERMANN D, et al. Multiobjective optimization on a limited budget of evaluations using model-assisted s-metric selection[C]// International Conference on Parallel Problem Solving from Nature. 2008: 784-794.
- [22] KNOWLES J. ParEGO: A hybrid algorithm with on-line landscape approximation for expensive multiobjective optimization problems[J]. IEEE Transactions on Evolutionary Computation, 2006, 10(1): 50-66.
- [23] ZHANG Q, LIU W, TSANG E, et al. Expensive multiobjective optimization by MOEA/D with Gaussian process model[J]. IEEE Transactions on Evolutionary Computation, 2010, 14(3): 456-474.
- [24] CHUGH T, JIN Y, MIETTINEN K, et al. A surrogate-assisted reference vector guided evolutionary algorithm for computationally expensive many-objective optimization[J]. IEEE Transactions on Evolutionary Computation, 2018, 22(1): 129-142.
- [25] HABIB A, KUMAR S H, CHUGH T, et al. A multiple surrogate assisted decomposition-based evolutionary algorithm for expensive multi-/many-objective optimization[J]. IEEE Transactions on Evolutionary Computation, 2019, 23(6): 1000-1014.
- [26] CHUGH T, SINDHYA K, MIETTINEN K, et al. On constraint handling in surrogate-assisted evolutionary many-objective optimization[C]// International Conference on Parallel Problem Solving from Nature. 2016: 214-224.
- [27] WAGNER T, EMMERICH M, DEUTZ A, et al. On Expected-Improvement Criteria for Model-based Multi-objective Optimization[C]// Proceedings of the 11th International Conference on Parallel Problem Solving From Nature: Part I. Berlin: Springer, 2010: 718-727.
- [28] LIU B, ZHANG Q, GIELEN G. A Gaussian process surrogate model assisted evolutionary algorithm for medium scale expensive optimization problems[J]. IEEE Transactions on Evolutionary Computation, 2014, 18(2): 180-192.
- [29] COUCKUYT I, DESCHRIJVER D, DHAENE T. Fast calculation of multiobjective probability of improvement and expected improvement criteria for Pareto optimization[J]. Journal of Global Optimization, 2014, 60(3): 575-594.
- [30] HERNANDEZ-LOBATOD, HERNANDEZ-LOBATAJ, SHAH A. Predictive entropy search for multi-objective Bayesian optimization[C]// Proceedings of the 33rd International Conference on International Conference on Machine Learning. 2016: 1492-1501.
- [31] BELAKARIA S, DESHWAL A. Max-value entropy search for multi-objective Bayesian optimization[C]// Neural Information Processing Systems. 2019: 1-30.
- [32] JIN Y, SENDHOFF B. Reducing fitness evaluations using clustering techniques and neural network ensembles[C]// Genetic and Evolutionary Computation Conference. Seattle, 2004: 688-699.
- [33] EMMERICH M, GIANNAKOGLU K, NAUJOKS B. Single- and multiobjective evolutionary optimization assisted by Gaussian random field metamodels[J]. IEEE Transactions on Evolutionary Computation, 2006, 10(4): 421-439.
- [34] GENG H T, DAI Z B, WANG T L, et al. Improved NSGA-III

Algorithm Based on Reference Point Selection Strategy[J]. Pattern Recognition and Artificial Intelligence, 2020, 33 (3): 191-201.

[35] SONG Z, WANG H, HE C, et al. A Kriging-Assisted Two-Archive Evolutionary Algorithm for Expensive Many-Objective Optimization[J]. IEEE Transactions on Evolutionary Computation, 2021, 25(6): 1013-1027.

[36] MCKAY M D, CONOVER R J B J. A comparison of three methods for selecting values of input variables in the analysis of output from a computer code[J]. Technometrics, 1979, 42(1): 55-61.

[37] WANG X L, JIN Y C, SCHMITT S, et al. An adaptive Bayesian approach to surrogate-assisted evolutionary multi-objective optimization[J]. Information Sciences, 2020, 519: 317-331.

[38] PAN L, HE C, TIAN Y, et al. A classification based surrogate-assisted evolutionary algorithm for expensive many-objective optimization [J]. IEEE Transactions on Evolutionary Computation, 2019, 23(1): 74-88.

[39] GUO D, WANG X, GAO K, et al. Evolutionary Optimization of High-Dimensional Multiobjective and Many-Objective Expensive Problems Assisted by a Dropout Neural Network[J]. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 2021, 52(4): 2084-2097.

[40] GUO D, JIN Y, DING J, et al. Heterogeneous ensemble-based infill criterion for evolutionary multiobjective optimization of expensive problems[J]. IEEE Transactions on Cybernetics, 2019, 49(3): 1012-1025.

[41] TIAN Y, CHENG R, ZHANG X, et al. PlatEMO: A MATLAB platform for evolutionary multi-objective optimization[J]. IEEE Computational Intelligence Magazine, 2017, 12(4): 73-87.



GENG Huantong, born in 1973, professor, Ph.D supervisor, is a senior member of China Computer Federation. His main research interests include computational intelligence, multi-objective optimization and meteorological data mining.

(责任编辑:喻黎)