

基于进化多任务的稀疏大规模多目标优化

梁正平¹⁾ 王侃¹⁾ 周倩¹⁾ 王继刚²⁾ 朱泽轩¹⁾

¹⁾(深圳大学计算机与软件学院 广东 深圳 518060)

²⁾(中兴通讯股份有限公司 广东 深圳 518057)

摘要 稀疏大规模多目标优化存在稀疏位置探测困难、搜索空间巨大等诸多挑战,现有为数不多的稀疏大规模多目标优化算法在稀疏位置的探测准确率和非零决策变量的优化程度方面尚存在较大提升空间.为进一步提升稀疏大规模多目标优化的性能,本文从辅助任务构建与优化、辅助任务重新初始化、知识迁移等三个方面,提出了基于进化多任务优化的稀疏大规模多目标优化算法(Evolutionary Multi-Task for Sparse Large-scale Multi-objective Optimization, SLMO-EMT).其中,辅助任务构建与优化方面,基于主任务精英解的稀疏分布,采用两种不同的方式对决策变量的搜索空间进行限定,构建分别用于对稀疏位置和非零决策变量进行降维优化的两个辅助任务.辅助任务重新初始化方面,根据辅助任务在历史迭代中的知识迁移效果,对其搜索空间和当前种群进行更新,以使辅助任务可持续促进主任务的进化.知识迁移方面,首先基于轮询方式和各辅助任务的知识迁移概率,挑选用于知识迁移的辅助任务,再基于相似度挑选适合的知识受体,最后在子代生成过程中采用迁移知识引导的局部交叉,借助辅助任务的知识促进主任务的进化.为验证SLMO-EMT的性能,将其与8个先进的稀疏大规模多目标优化算法在1000-10 000维的32个基准测试实例,以及8个应用测试实例上进行对比,实验结果表明SLMO-EMT对于稀疏大规模多目标优化问题的求解具有明显的竞争优势.SLMO-EMT的源代码已在Github上公开:<https://github.com/CIA-SZU/WK>.

关键词 稀疏大规模多目标优化;进化多任务;辅助任务;知识迁移

中图分类号 TP18 **DOI号** 10.11897/SP.J.1016.2025.00358

Evolutionary Multi-Task for Sparse Large-Scale Multi-Objective Optimization

LIANG Zheng-Ping¹⁾ WANG Kan¹⁾ ZHOU Qian¹⁾ WANG Ji-Gang²⁾ ZHU Ze-Xuan¹⁾

¹⁾(College of Computer Science and Software Engineering, Shenzhen University, Shenzhen, Guangdong 518060)

²⁾(ZTE Corporation, Shenzhen, Guangdong 518057)

Abstract Due to potential conflicts between different objectives, when solving sparse large-scale multi-objective problems (SLMOPs), it is usually necessary to use Pareto dominance relationships to balance potential global optimal solutions. Specially, the decision variables of SLMOPs are huge, and their Pareto optimal solutions have sparse properties, making the optimization of SLMOPs difficult to detect sparse positions and to explore huge search space. In recent years, some sparse large-scale multi-objective evolutionary algorithms (SLMOEAs) based on MOEAs have been proposed to address the challenges of SLMOPs, which can be divided into two categories: single-layer encoding approach and two-layer encoding approach. Although existing SLMOEAs have made some progress in optimizing SLMOPs, there still have significant room for improvement in terms of the detection accuracy of sparse positions and the optimization degree of

收稿日期:2024-05-29;在线发布日期:2024-12-04.本课题得到国家重点研发计划(2021YFB2900800)、深圳市基础研究计划(JCYJ20220531102617039)资助.梁正平,博士,副教授,中国计算机学会(CCF)会员,主要研究领域为计算智能、大数据分析与应用. E-mail: liangzp@szu.edu.cn.王侃,硕士,主要研究领域为计算智能.周倩,硕士研究生,主要研究领域为计算智能.王继刚,博士,主要研究领域为计算智能、网络安全.朱泽轩,博士,教授,主要研究领域为计算智能、生物信息学.

non-zero decision variables. To enhance the performance of sparse large-scale multi-objective optimization, this paper proposes an algorithm called Evolutionary Multi-Task for Sparse Large-scale Multi-objective Optimization (SLMO-EMT) from three aspects: auxiliary task construction and optimization, auxiliary task reinitialization, and knowledge transfer. In terms of auxiliary task construction and optimization, two auxiliary tasks are constructed to perform dimensionality reduction optimization for sparse positions and non-zero decision variables, respectively. Where two different methods are employed to reduce the search space of auxiliary tasks based on the sparse distribution of elite solutions of the main task. Regarding auxiliary task reinitialization, the search space and population of auxiliary task are updated based on the knowledge transfer effect in historical iterations, enabling auxiliary tasks to constantly promote the evolution of the main task. As for knowledge transfer, a suitable auxiliary task is selected based on a round-robin approach and the knowledge transfer probabilities of each auxiliary task firstly. A knowledge recipient is then selected based on similarity. After that, a new local crossover operator guided by transferred knowledge is utilized to facilitate the evolution of the main task with the help of auxiliary task. To validate the performance of SLMO-EMT, it is compared with eight advanced sparse large-scale multi-objective optimization algorithms on 32 benchmark test instances ranging from 1000 to 10 000 dimensions, as well as 8 application test instances. Experimental results demonstrate that SLMO-EMT has a significant competitive advantage in solving sparse large-scale multi-objective optimization problems. The source code of SLMO-EMT has been made public on Github: <https://github.com/CIA-SZU/WK>.

Keywords sparse large-scale multi-objective optimization; evolutionary multi-task; auxiliary task; knowledge transfer

1 引言

神经网络训练^[1]、稀疏信号重构^[2]、工业无源光网络智能运维^[3]等应用中存在各种类型的稀疏大规模多目标优化问题 (Sparse Large-scale Multi-objective Optimization Problems, SLMOPs)。由于不同目标之间可能相互冲突,在求解 SLMOPs 时,通常需要采用帕累托支配关系对潜在的全局最优解进行权衡。与此同时,SLMOPs 的决策变量庞大,存在“维度诅咒”,且其帕累托最优解具有稀疏性质,即帕累托最优解中大多数决策变量为零,使得 SLMOPs 的优化存在巨大挑战^[4-6]。

进化算法 (Evolutionary Algorithm, EAs)^[7-8]通过模仿生物进化过程对问题进行求解,可在多个冲突的目标之间取得较好平衡,并同时找到多个优异的帕累托非支配解,已广泛应用于各类多目标优化问题^[9-12]。然而,现有进化多目标优化算法 (Multi-Objective Evolutionary Algorithms, MOEAs) 难以很好地处理 SLMOPs 存在的“维度诅咒”和稀疏特性^[13-16]。

近年来,针对 SLMOPs 存在的挑战,学者们在 MOEAs 的基础上,陆续提出了一些稀疏大规模多目标进化算法 (Sparse Large-scale Multi-Objective Evolutionary Algorithms, SLMOEAs),可分为基于单层编码^[17-18]和基于双层编码^[19-22]两类。基于单层编码的 SLMOEAs 采用传统的实数向量形式对个体进行单层编码,通过设计具有稀疏位置探测能力和实数优化能力的搜索算子以获得较好的优化效果。基于双层编码的 SLMOEAs 则采用二进制向量和实数向量的组合对个体进行混合编码,又可分为基于高效搜索算子和基于降维策略两类。前者主要关注对二进制向量中稀疏位置的高效探测,可更准确地确定帕累托最优解中的稀疏位置。后者根据种群的稀疏分布对二进制向量和实数向量进行降维搜索,以降低帕累托最优解的搜索难度。

总体上,现有 SLMOEAs 在解决 SLMOPs 上已取得一定的进展,但仍存在较大的改进空间^[23-24]。其中,基于单层编码的 SLMOEAs 相对传统 MOEAs 能更好地优化 SLMOPs,但稀疏程度相关参数难以合理设置,且当决策空间维度较高时,最终所获稀疏解的质量不如基于双层编码的

SLMOEAs。基于双层编码的SLMOEAs采用并行方式独立搜索二进制向量和实数向量,不需对稀疏程度相关参数进行设定,但当二进制向量和实数向量的组合不合理时,反而会降低整体的搜索效率。与此同时,基于双层编码的SLMOEAs在搜索二进制向量的过程中,当稀疏位置探测不准确时,会导致实数向量的对应位置要么难以得到充分优化,要么浪费搜索资源。

进化多任务(Evolutionary Multi-Task, EMT)是进化计算领域的研究前沿,不仅可同时对多个任务进行优化,且通过相关任务间的正向知识迁移,可加速各任务的搜索进程,并获得比单任务形式更好的优化效果^[25-28]。近年来,EMT已在各类优化问题的求解上取得了较显著的进展^[29-32]。

受EMT的启发,并针对现有SLMOEAs存在的不足,本文提出了一种基于EMT的稀疏大规模多目标优化算法(Evolutionary Multi-Task for Sparse Large-scale Multi-objective Optimization, SLMO-EMT),以进一步提升SLMOPs的优化性能,获得质量更优的稀疏解。SLMO-EMT总体采用双层编码和EMT框架。首先,利用主任务种群的精英解,分别对二进制向量和实数向量的搜索空间进行限定,构建分别用于对稀疏位置和非零决策变量进行降维优化的两个辅助任务。其次,为使辅助任务可持续促进主任务的进化,根据辅助任务在历史迭代中的知识迁移效果,在优化过程中采用重新初始化方式更新辅助任务的搜索空间和当前种群。在知识迁移方面,首先基于轮询方式和各辅助任务的知识迁移概率,挑选用于知识迁移的辅助任务,再基于相似度挑选合适的知识受体,最后在子代生成过程中采用迁移知识引导的局部交叉,达到借助辅助任务的知识提升主任务进化效果的目的。为验证SLMO-EMT的性能,将其与8个先进的稀疏大规模多目标优化算法在1000-10 000维的32个基准测试实例,以及8个应用测试实例上进行对比,实验结果表明SLMO-EMT对于稀疏大规模多目标优化问题的求解具有明显的竞争优势。

本文的主要创新和贡献如下:

(1)针对现有SLMOEAs的不足,从辅助任务构建入手,提出了基于EMT的稀疏大规模多目标优化框架;

(2)为使辅助任务可持续促进主任务的进化,提出了基于知识迁移效果的辅助任务重新初始化策略;

(3)为提升知识迁移的有效性,提出了基于轮询和知识迁移概率的辅助任务选择策略、基于相似度的知识受体选择策略,以及基于局部交叉的知识利用策略。

本文的组织结构如下:第2节介绍预备知识和相关工作;第3节介绍所提出的算法;第4节介绍实验方案并对结果进行分析;最后总结全文并展望未来工作。

2 预备知识和相关工作

2.1 SLMOPs定义

多目标优化问题(Multi-objective Optimization Problems, MOPs)的数学定义如下^[33]:

$$\begin{aligned} \min_x f(x) &= (f_1(x), \dots, f_M(x)) \\ \text{s.t. } x &\in \Omega \end{aligned} \quad (1)$$

其中, $x=(x_1, x_2, \dots, x_D)$ 是决策空间 Ω 中的 D 维决策变量, $f(x)$ 为待求解的 M 个目标函数。若决策空间 Ω 中的其它任何个体均不能支配个体 x ,则称 x 为帕累托最优解^[34]。当决策变量个数 D 大于100,且帕累托最优解中非零元素的个数远小于 D 时,此类MOPs通常被称为SLMOPs^[15,22]。

2.2 进化多任务

受人脑可同时处理多个问题的启发,Gupta等人提出了EMT的思想,近年来已演变成进化计算领域的一种新型计算范式^[25,35-36]。

EMT中的每个任务对应一个优化问题,多个任务同时优化,并在优化过程中进行任务间的知识迁移。EMT可加速各任务的搜索进程,并在各个优化问题上获得比单任务形式更好的优化效果。EMT的数学定义如下:

$$\begin{aligned} \{x_1^*, \dots, x_i^*, \dots, x_k^*\} &= \{ \operatorname{argmin}_{f_1}(x_1), \dots, \\ &\operatorname{argmin}_{f_i}(x_i), \dots, \operatorname{argmin}_{f_k}(x_k) \} \end{aligned} \quad (2)$$

其中, k 代表任务个数, f_i 为第 i 个任务 T_i 的目标函数, x_i^* 表示 T_i 的全局最优解。在多目标优化场景中, x_i^* 为帕累托最优解集。

EMT的总体框架如图1所示。在进化过程的每轮迭代,每个任务基于知识迁移概率(Knowledge Transfer Probability, KTP),自适应地通过自身进化方式,基于各自的父代种群生成子代;或通过任务间的知识迁移,结合父代种群和源自其它任务的知识生成子代。

在EMT中,当源自其它任务的知识对当前任

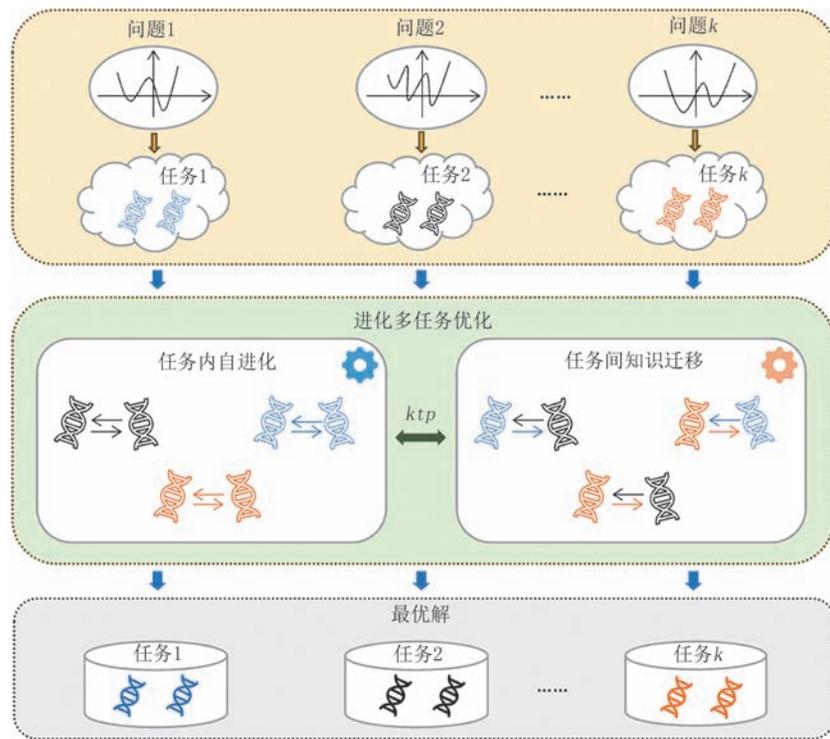


图1 EMT的总体框架图

务具有正向引导作用,即正迁移时,有利于提升当前任务的搜索性能,从而可促使当前任务输出更高质量的最优解。反之,当源自其它任务的知识对当前任务具有负向引导作用,即负迁移时,非但无法帮助当前任务更好地搜索,反而会使当前任务陷入局部最优。因此,在EMT中需要对知识来源、知识迁移时机、知识受体的选择、知识利用方式等进行合理设计。与此同时,在应用EMT解决特定应用问题时,首先需要为主任务合理构建辅助任务。

2.3 相关工作

现有SLMOEAs可分为基于单层编码和基于双层编码两类。

2.3.1 基于单层编码的SLMOEAs

基于单层编码的SLMOEAs采用单层实数向量形式对SLMOPs中的个体进行编码,重点关注高效搜索算子的设计,以实现SLMOPs稀疏位置的探测和非零决策变量的优化。SECSO^[17]在竞争粒子群优化算法中引入融合强凸稀疏策略和三类竞争机制的新型搜索算子,其中强凸稀疏策略用于对稀疏位置的优化,三类竞争机制用于非零决策变量探索和开发之间的平衡。为更好地利用种群进化过程中有用的历史信息,ST-CCPSO^[18]基于优化过程中的累积梯度值,采用稀疏截断策略以进一步提升稀疏位置探测的准确率,同时采用基于聚类的竞争粒

子群优化算子对非零决策变量进行优化,有助于种群跳出局部最优。基于单层编码的SLMOEAs明确考虑对稀疏位置的探测,相比传统MOEAs能更好地优化较大规模的SLMOPs。但稀疏程度相关参数难以有效设置,当面对维度更高的SLMOPs时,无法对相关参数的取值进行合理的自适应调整^[18],现有基于单层编码的SLMOEAs最终所获稀疏解的质量尚待提升。

2.3.2 基于双层编码的SLMOEAs

基于双层编码的SLMOEAs采用二进制向量和实数向量的组合对SLMOPs中的个体进行混合编码。其中,二进制向量用于稀疏位置的探测,实数向量用于非零决策变量的优化。SparseEA^[4]是首个基于双层编码的SLMOEA。在此基础上,学术界提出了多个性能增强的双层编码SLMOEAs,可分为基于高效搜索算子和基于降维策略两类。

现有基于高效搜索算子的双层编码SLMOEAs主要关注对二进制向量中稀疏位置的高效探测,以更准确地确定帕累托最优解中的稀疏位置。MSKEA^[16]引入三种动态稀疏性知识,并设计了基于知识融合的多阶段进化策略,可以更准确地估计每个决策变量为零的概率。SGECF^[19]提出稀疏引导的精英主义协同进化算子,用稀疏精英个体引导两个非精英子种群的进化。S-NSGA-II^[20]设计了

可变条纹稀疏种群采样策略,并提出了稀疏模拟二进制交叉算子和稀疏多项式变异算子,可生成更优的稀疏个体。SparseEA2^[21]将二进制向量的变异位置与实数向量的变异位置关联,可更合理地实现对稀疏位置的探测。SCEA^[37]则基于稀疏最优解的先验知识进行协同进化,有利于生成具有更好稀疏性的种群。AGSEA^[38]采用强化学习建立种群状态与引导向量预期累积奖励之间的映射模型,有利于在高维空间中更好地获得稀疏帕累托最优解。TELSO^[39]通过挖掘种群中精英个体二进制编码的频繁项集,以生成收敛程度更好的稀疏个体。

基于降维策略的双层编码SLMOEAs主要关注对二进制向量和实数向量的降维搜索,以降低帕累托最优解的搜索难度。根据所使用降维策略的不同,又可分为基于机器学习和基于决策变量分组两类。前者的代表性方法为MOEAPSL^[6],利用受限玻尔兹曼机和去噪自编码器分别生成二进制向量和实数向量的低维子空间。后者相关方法中,TS-SparseEA^[22]基于SparseEA的初始化策略对二进制向量进行静态分组。TS2-SparseEA^[40]对TS-SparseEA的向量匹配策略进行改进,提出了相互偏好计算方法和双向匹配策略。PMMOEA^[15]基于模式挖掘方式,将种群中个体的二进制向量分为最大和最小候选集。DSGEA^[23]基于决策变量的稀疏度对二进制向量进行平均分组,并在迭代过程中动态调整分组数量。SLMEA^[24]先归类稀疏度分别为1和0的决策变量,再采用快速聚类方式对剩余二进制向量进行动态分组。MGCEA^[41]则采用多粒度聚类方法估计每一代决策变量的稀疏分布,并将其划分为不同数量的层。

基于双层编码的SLMOEAs采用二进制向量显示设定个体的稀疏位置,并采用并行方式独立搜索二进制向量和实数向量,在优化过程中自动设置二进制向量中各变量的取值,不需对稀疏程度相关参数进行设定,克服了单层编码SLMOEAs中稀疏程度相关参数难以合理设置的问题。但在基于双层编码的SLMOEAs中,当二进制向量和实数向量的组合不合理,即一个较优向量和一个较差向量组成一个较差个体时,由于该个体在环境选择时容易被淘汰,导致所对应的较优向量亦容易被丢弃,从而降低整体的搜索效率。与此同时,基于双层编码的SLMOEAs在搜索二进制向量的过程中,若最优值为1的变量被错误地设置为0,将导致实数向量对应位置的变量难以得到充分优化。反之,若最优值为0的变量被错误地设置为1,将导致对实数向量对应

位置变量的无效搜索,浪费搜索资源。

2.4 研究动机

针对现有双层编码SLMOEAs的不足,本文通过降维方式构建稀疏优化和实数优化两个辅助任务,提出了一种采用双层编码、基于EMT的稀疏大规模多目标优化算法SLMO-EMT,以更好地提升SLMOPs中稀疏位置的探测准确率和非零决策变量的优化程度。其中,对于二进制向量和实数向量的组合问题,在辅助任务的构建/重新初始化过程中,从主任务当前帕累托非支配解集中选取二进制向量和实数向量,分别作为实数优化辅助任务和稀疏优化辅助任务中的初始二进制向量和实数向量,并通过对辅助任务的优化,搜索对应的最优实数向量或最优二进制向量,从而实现对二进制向量和实数向量的合理组合。对于二进制向量中稀疏位置探测不准确带来的问题,一方面在实数优化辅助任务构建/重新初始化时,将初始二进制向量中少量0翻转为1,以缓解主任务中实数向量对应位置难以充分优化的问题。另一方面基于稀疏优化辅助任务的知识合理提升主任务中二进制向量的稀疏度,以减少不必要的搜索资源浪费。

3 算法设计

本节首先介绍SLMO-EMT的整体框架,然后分别介绍辅助任务构建与优化策略、辅助任务重新初始化策略、知识迁移策略,最后分析SLMO-EMT的时间复杂度。

3.1 SLMO-EMT整体框架

SLMO-EMT的整体框架如图2所示,包括一个直接优化待求解SLMOP的主任务 T_{main} ,两个分别关注二进制向量优化和实数向量优化的辅助任务,分别命名为稀疏优化辅助任务 T_{sparse} 和实数优化辅助任务 T_{real} 。

首先,采用双层编码方式,在待优化SLMOP的原始决策空间随机初始化种群 P_{main} ,作为 T_{main} 的种群。接着,基于 P_{main} 中的精英解,通过辅助任务构建策略,分别生成 T_{sparse} 和 T_{real} 。 T_{sparse} 和 T_{real} 采用与 T_{main} 相同的双层编码方式和评价函数,但仅对各自所关注的决策变量进行降维搜索。在优化过程中, T_{sparse} 和 T_{real} 通过自进化方式生成子代。具体而言,针对所关注决策变量,对二进制向量执行单点交叉和按位变异^[42],对实数向量执行模拟二进制交叉^[43]和多项式变异^[44]。 T_{main} 则基于 T_{sparse} 和 T_{real} 的KTP

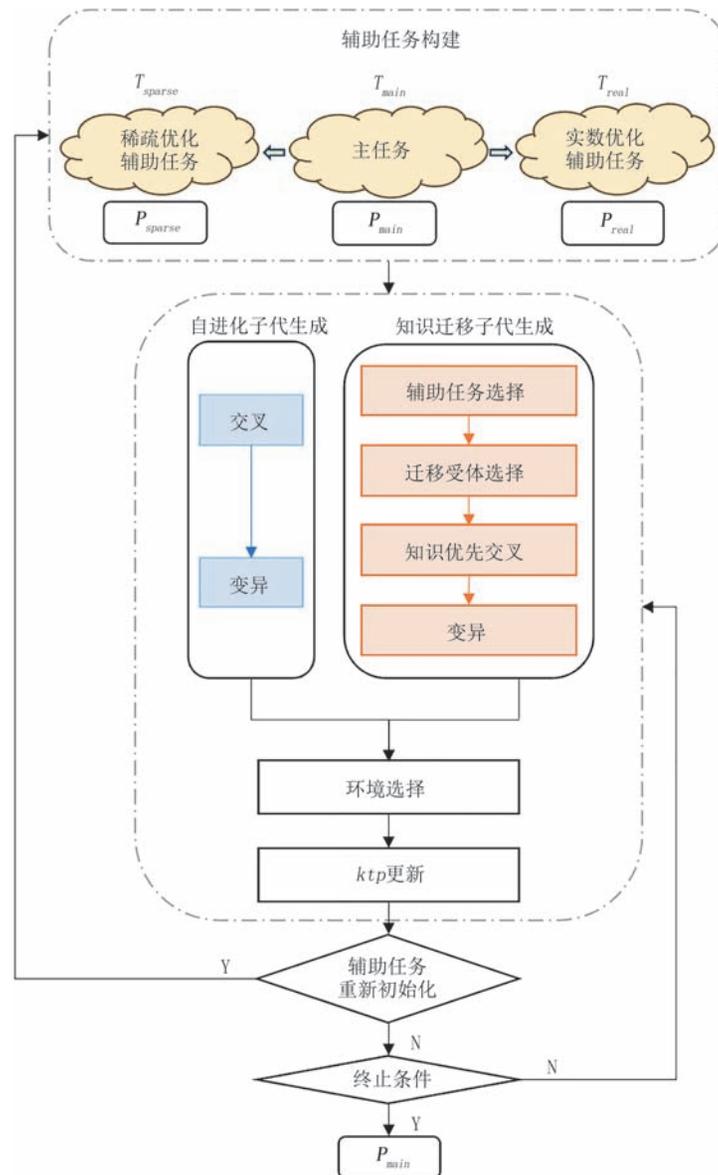


图2 SLMO-EMT的整体框架

值,针对所有决策变量,自适应地通过自进化方式或知识迁移方式生成子代。基于知识迁移的子代生成首先选择辅助任务,并从所选辅助任务中选取个体作为知识,然后在 T_{main} 中选择合适的知识迁移受体,再采用本文提出的局部交叉算子和常规变异算子生成子代。之后,对各任务进行环境选择^[45],并更新各辅助任务的KTP值。在每轮迭代结束时,根据各辅助任务在历史迭代中的知识迁移效果,确定是否需要对各辅助任务进行重新初始化。最后,当满足终止条件时,输出 T_{main} 的种群 P_{main} 。

T_{main} 在原始搜索空间中同时搜索二进制向量和实数向量,理论上具有搜索到全局最优解的可能性,但由于搜索空间巨大,难以在规定的迭代次数内获得全局最优解。而 T_{sparse} 和 T_{real} 基于 T_{main} 的精英解

构建,且仅在部分决策变量上进行降维搜索,具有快速收敛到最优解的能力,因而基于 T_{sparse} 和 T_{real} 的引导,有利于提升 T_{main} 的优化性能。

SLMO-EMT 的伪代码如算法 1 所示。首先,通过随机初始化,生成由 N 个 $(mask, dec)$ 元组构成的主任务种群 P_{main} ,并对所有个体进行评价(第 1 行)。其中, $mask = (mask_1, mask_2, \dots, mask_D)$ 为二进制向量, $dec = (dec_1, dec_2, \dots, dec_D)$ 为实数向量,对于 P_{main} 中的个体 $x = (x_1, x_2, \dots, x_D)$,其决策变量 $x_i = mask_i * dec_i, i \in \{1, 2, \dots, D\}$, D 为决策变量维度。然后,分别采用算法 2 和算法 3 构建初始的 T_{sparse} 和 T_{real} ,并对所生成种群 P_{sparse} 和 P_{real} 中的所有个体进行评价(第 2-3 行)。接着,分别初始化 T_{sparse} 和 T_{real} 的KTP值 ktp_s 和 ktp_r (第 4 行),并设置当前已

用评价次数(第5行)。当累计评价次数 FE 小于最大评价次数 MFE 时,进行迭代优化(第6-21行)。

在每轮迭代中,首先 T_{sparse} 和 T_{real} 进行任务内自进化(第7行),同时更新 FE (第8行)。然后 T_{main} 采用算法4生成子代(第9行),更新 FE 后进行环境选择(第10-11行)。接着更新 ktp_s 和 ktp_r (第12行)。最后,当 T_{sparse} 和 T_{real} 需要进行重新初始化时,基于 T_{main} 的当前种群 P_{main} ,分别采用算法2和算法3进行重新初始化,并重新评价初始生成的所有个体(第13-20行)。

算法1. SLMO-EMT

输入: N (种群大小), MFE (最大评价次数)

输出: P_{main} (T_{main} 种群)

1. 初始化 P_{main} ;
2. $[P_{sparse}, Space_s] \leftarrow InitializeTaskS(N, P_{main}); //$ 算法2
3. $[P_{real}, Space_r] \leftarrow InitializeTaskR(N, P_{main}); //$ 算法3
4. 初始化 ktp_s, ktp_r ;
5. $FE \leftarrow 3N$;
6. WHILE $FE < MFE$
7. T_{sparse} 和 T_{real} 自进化;
8. $FE \leftarrow FE + 2N$;
9. $O_{main} \leftarrow GenerateOffspring(N, P_{main}, P_{sparse}, P_{real}, ktp_s, ktp_r, Space_s, Space_r); //$ 算法4
10. $FE \leftarrow FE + N$;
11. 对 $P_{main} \cup O_{main}$ 进行环境选择;
12. 更新 ktp_s, ktp_r ; $//$ 公式(4)
13. IF 满足 T_{sparse} 的重新初始化条件 THEN
14. $[P_{sparse}, Space_s] \leftarrow InitializeTaskS(N, P); //$ 算法2
15. $FE \leftarrow FE + N$;
16. END IF
17. IF 满足 T_{real} 的重新初始化条件 THEN
18. $[P_{real}, Space_r] \leftarrow InitializeTaskR(N, P); //$ 算法3
19. $FE \leftarrow FE + N$;
20. END IF
21. END WHILE

3.2 辅助任务构建与优化

SLMO-EMT在初始环节构建 T_{sparse} 和 T_{real} 两个辅助任务,分别用于对二进制向量和实数向量的优化。 T_{sparse} 和 T_{real} 采用与 T_{main} 相同的双层编码方式和评价函数。为使 T_{sparse} 和 T_{real} 可更好的正向引导 T_{main} , T_{sparse} 和 T_{real} 的初始种群均基于 P_{main} 的帕累托非支配解集 F_1 生成。与此同时,为提升SLMO-EMT的整体收敛速度, T_{sparse} 和 T_{real} 在优化过程中采用降维形式,仅对部分决策变量进行搜索。

T_{sparse} 和 T_{real} 初始种群的生成过程如图3所示。首先,对当前 P_{main} 进行非支配排序,得到第1层帕累托非支配解集 F_1 。然后,各从 F_1 中随机选择一个个体,分别取其对应的实数向量和二进制向量,作为 T_{sparse} 的恒定实数向量 dec^* 和 T_{real} 的初始二进制向量 $mask^\#$ 。在此基础上,再基于 F_1 中各个体的二进制向量和实数向量,分别为 T_{sparse} 生成 N 个二进制向量,为 T_{real} 生成恒定二进制向量 $mask^*$ 和 N 个实数向量,最终组合得到 T_{sparse} 和 T_{real} 的初始种群 P_{sparse} 和 P_{real} 。

(1) T_{sparse} 的构建与优化

对于 P_{sparse} 中各个体的初始二进制向量,采用如下方式生成:

首先,基于 F_1 中各个体的二进制向量,计算各维决策变量的稀疏度 $score_i$:

$$score_i = \frac{1}{|F_1|} \sum_{j=1}^{|F_1|} (1 - mask_{j,i}) \quad i \in \{1, \dots, D\} \quad (3)$$

其中, $|F_1|$ 表示 F_1 中的个体数量, $mask_{j,i}$ 表示 F_1 中个体 j 所对应二进制向量中第 i 维的值。 $score_i$ 的取值范围为 $[0-1]$, $score_i$ 越大,说明 F_1 中有越多个体的二进制向量在第 i 维上的值为0;反之,说明越多个体的二进制向量在第 i 维上的值为1。进一步可知, $score_i$ 的值越靠近0或1,说明 F_1 所对应各二进制向量第 i 维已正确取值的概率越大;反之, $score_i$ 的值越靠近0.5,说明在二进制向量中第 i 维已正确取值的概率越小,也即 T_{main} 对 P_{main} 所对应二进制向量中第 i 维的搜索越不充分。

接着,基于 $score_i$ 逐一生成 P_{sparse} 中各初始二进制向量第 i 维变量的值。具体而言,对于第 j 个二进制向量的第 i 维变量 $mask_{j,i}$,生成一个随机数 $rand$,如果 $score_i$ 小于 $rand$,则将 $mask_{j,i}$ 置为1,反之则置为0。采用此种方式初始化 P_{sparse} 中各二进制向量的原因,旨在充分感知并利用 P_{main} 中精英个体的稀疏分布,以使 P_{sparse} 在初始化时就具有较理想的稀疏分布,从而为后续的快速收敛奠定良好基础。

至此,将所获得二进制向量集合与 F_1 中随机获取的恒定实数向量 dec^* 逐一组合,生成 T_{sparse} 的初始种群 P_{sparse} 。

为降低 T_{sparse} 的优化难度,并使其关注 T_{main} 中尚未充分优化的维度,以提升后续知识迁移的有效性,在初始化过程中,进一步对 P_{sparse} 的决策变量进行降维限定,且使 T_{sparse} 后续仅在限定维度上搜索。具体做法如下:

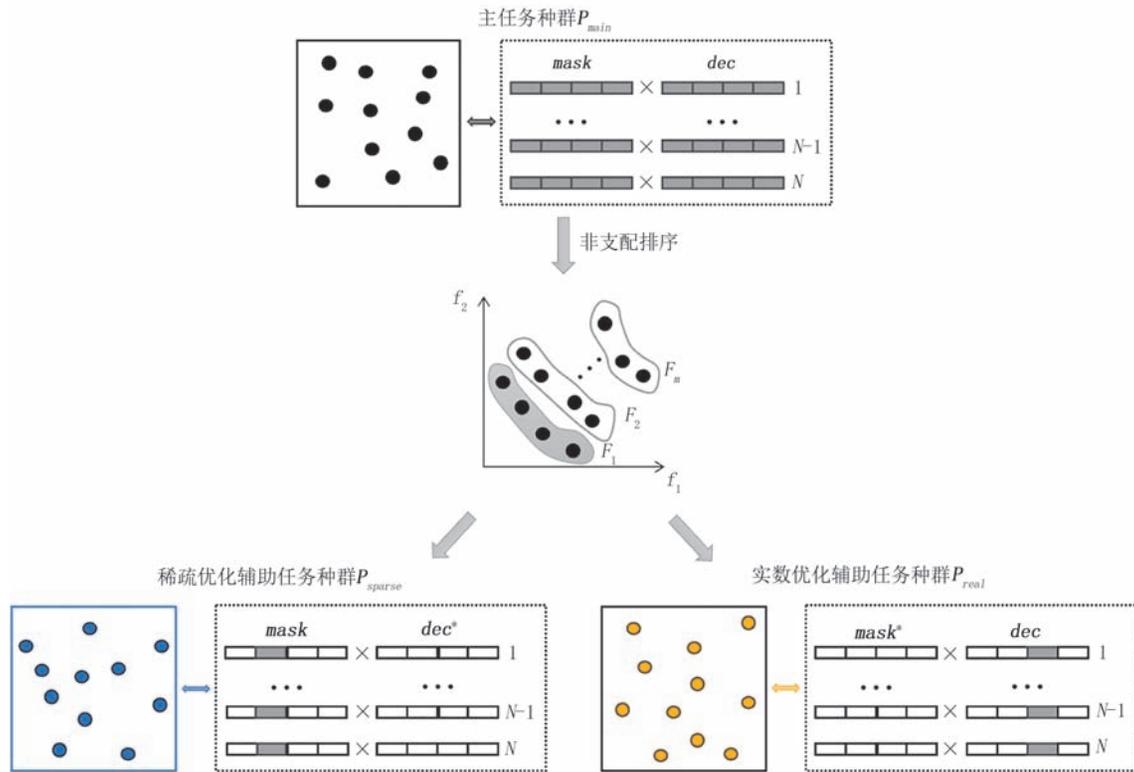


图3 辅助任务初始种群生成示意

基于公式(3)的计算结果,根据设定的阈值 α 和参数 d_1 ,选取所有稀疏度在 $[0.5 - \alpha, 0.5 + \alpha]$ 中的决策变量。若所选取决策变量占总决策变量的比例大于等于 d_1 ,则对所选取决策变量的位置进行标记。若小于 d_1 ,则选取稀疏度值最靠近0.5的 $d_1 D$ 个决策变量,并对其位置进行标记。所有标记位置的集合记为 $Space_s$ 。

构建 T_{sparse} 的伪代码如算法2所示。首先,从 P_{main} 的第1层帕累托非支配解集 F_1 中随机选取一个个体的实数向量作为 T_{sparse} 的恒定实数向量 dec^* (第1行)。接着,基于公式(3)计算各维决策变量的稀疏度 $score_i$ (第2行)。之后,对 T_{sparse} 的种群 P_{sparse} 进行初始化(第3-12行)。最后,设定 T_{sparse} 的搜索空间 $Space_s$ (第13-17行)。

算法2. InitializeTaskS

输入: N (种群大小), P_{main} (T_{main} 种群)

输出: P_{sparse} (T_{sparse} 初始种群), $Space_s$ (T_{sparse} 搜索空间)

1. 从 P_{main} 的 F_1 中随机选取一个个体,将其实数向量记为 dec^* ;
2. 计算 F_1 中每维决策变量的稀疏度 $score_i$; //公式(3)
3. FOR $j=1$ to N
4. $P_{sparse\ j}.dec \leftarrow dec^*$;
5. FOR $i=1$ to D
6. IF $score_i < rand$ THEN

7. $P_{sparse\ j}.mask_i \leftarrow 1$;

8. ELSE

9. $P_{sparse\ j}.mask_i \leftarrow 0$;

10. END IF

11. END FOR

12. END FOR

13. IF 稀疏度属于 $[0.5 - \alpha, 0.5 + \alpha]$ 的决策变量不少于 $d_1 D$ THEN

14. $Space_s \leftarrow$ 稀疏度属于 $[0.5 - \alpha, 0.5 + \alpha]$ 的决策变量位置;

15. ELSE

16. $Space_s \leftarrow$ 稀疏度值最靠近0.5的 $d_1 D$ 个决策变量位置;

17. END IF

在 T_{sparse} 的优化过程中,仅采用自进化方式对二进制向量中位置编号属于 $Space_s$ 的维度进行搜索。具体而言,首先从 P_{sparse} 中随机选择两个父代个体,然后在这两个个体所对应二进制向量中,位置编号属于 $Space_s$ 的维度执行单点交叉和按位变异,生成两个子代个体。最后基于 T_{main} 的评价函数,采用非支配排序和拥挤度距离,进行环境选择。

由于 dec^* 为 P_{main} 中精英解的实数向量,在 T_{sparse} 的构建与优化过程中,将 P_{sparse} 中的实数向量固定为 dec^* ,一方面是为了使 P_{sparse} 中的实数向量具有较高质量,另一方面旨在为该 dec^* 搜索到与之最为匹配的二进制向量,从而实现二进制向量和实数向量

的合理组合,并在后续作为潜在知识迁移给 T_{main} 。

(2) T_{real} 的构建与优化

对于 P_{real} 中各个体的初始实数向量,采用如下方式生成:

首先,基于 F_1 中各个体的实数向量,获取在各维决策变量上的最小值 min_i 和最大值 max_i 。然后,在各 $[min_i, max_i]$ 区间,采用随机方式,逐一生成各实数向量的第 i 维变量值。

类似 P_{sparse} 的初始化,采用上述方式初始化 P_{real} 中各实数向量的原因,旨在使 P_{real} 在初始生成时即具有较理想的取值,从而为后续的快速收敛奠定良好基础。

由于 T_{main} 在优化过程中存在稀疏位置探测不准确问题,可能使 F_1 所对应各二进制向量中,部分应该取值为1的变量,被错误地设置为0。此种情况下,若从 F_1 中选取一个二进制向量 $mask^\#$ 直接作为 T_{real} 的恒定二进制向量,会导致 T_{real} 无法对实数向量中对应位置的变量进行有效搜索。

为此,对于 $mask^\#$ 中部分取值为0的变量,采用如下方式将其取值翻转为1,并将新得到的二进制向量 $mask^*$ 作为 T_{real} 的恒定二进制向量:

首先,通过公式(3)获得各决策变量的稀疏度,并根据设定的参数 d_2 ,选取前 d_2D 个稀疏度值对应的决策变量。然后,再从所选取决策变量中,随机选取 d_2d_2D 个决策变量。最后,将 $mask^\#$ 中与所选取 d_2d_2D 个决策变量对应位置的变量值设置为1。

至此,将所获得实数向量集合与 $mask^*$ 逐一组合,生成 T_{real} 的初始种群 P_{real} 。

构建 T_{real} 的伪代码如算法3所示。首先,从 P_{main} 的第1层帕累托非支配解集 F_1 中随机选取一个个体的二进制向量作为 T_{real} 的初始二进制向量 $mask^\#$ (第1行)。接着,基于公式(3)计算各维决策变量的稀疏度 $score_i$ (第2行)。之后,基于稀疏度将 $mask^\#$ 中的部分决策变量翻转为1,得到 T_{real} 的恒定二进制向量 $mask^*$ (第3-5行),同时,设定 T_{real} 的搜索空间 $Space_r$ (第6行)。最后,再基于 F_1 计算每维决策变量最小值和最大值(第7行),逐一初始化 T_{real} 的每个实数向量,并与 $mask^*$ 组合得到 P_{real} (第8-13行)。

算法3. InitializeTaskR

输入: N (种群大小), P_{main} (T_{main} 种群)

输出: P_{real} (T_{real} 初始种群), $Space_r$ (T_{real} 搜索空间)

1. 从 P_{main} 的 F_1 中随机选取一个个体,将其二进制向量记为 $mask^\#$;
2. 计算 F_1 中每维决策变量的稀疏度 $score_i$; //公式(3)

3. 选取稀疏度最大的 d_2D 个决策变量;
4. 从上述 d_2D 个决策变量中随机选取 d_2d_2D 个,并记录其位置;
5. 将 $mask^\#$ 中与上述对应位置的变量值设置为1,得到 $mask^*$;
6. $Space_r \leftarrow mask^*$ 中所有值为1的变量位置;
7. 计算 F_1 中每维决策变量的最小值 min_i 和最大值 max_i ;
8. FOR $j = 1$ to N
9. FOR $i = 1$ to D
10. $P_{real.j}.dec_i \leftarrow [min_i, max_i]$ 间随机生成一个值;
11. END FOR
12. $P_{real.j}.dec \leftarrow mask^*$;
13. END FOR

在 T_{real} 的优化过程中,仅采用自进化方式对实数向量中位置编号属于 $Space_r$ 的维度进行降维搜索。具体而言,首先从 P_{real} 中随机选择两个父代个体,然后在这两个个体所对应实数向量中,位置编号属于 $Space_r$ 的维度执行模拟二进制交叉和多项式变异,生成两个子代个体。最后类似 T_{sparse} ,基于 T_{main} 的评价函数,采用非支配排序和拥挤度距离进行环境选择。值得说明的是,对于 $mask^*$ 中取值为0的变量,无论实数向量中对应位置的变量如何优化,组合后所得个体在该位置的决策变量值始终为0。为提升 T_{real} 的搜索速度,在算法3中将 $Space_r$ 设置为 $mask^*$ 中所有值为1的变量所在位置的集合。

由于 $mask^*$ 对 $mask^\#$ 中取值为1的变量进行了完整保留,且 $mask^\#$ 为 P_{main} 中精英解的实数向量,在 T_{real} 的构建与优化过程中,将 P_{real} 中的二进制向量固定为 $mask^*$ 。对应 T_{sparse} ,一方面是为了使 P_{real} 中的二进制向量具有较高的质量,另一方面旨在为该 $mask^*$ 搜索到与之最为匹配的实数向量,从而实现实数向量和二进制向量的合理组合,并在后续作为潜在知识迁移给 T_{main} 。

此外,由于 T_{real} 仅在 $Space_r$ 所关联维度上进行搜索,使得 P_{real} 能快速收敛。因此,即使将 $mask^*$ 中少量应当为0的变量错误翻转为1, P_{real} 在对应位置的实数向量值仍然可以快速收敛到最优值0附近。而对于 $mask^*$ 中那些被正确翻转为1的变量, T_{real} 则能帮助对应位置的实数变量快速逼近到其最优值,从而有效处理由于稀疏位置探测不准确带来的实数向量对应位置难以充分优化的问题。

3.3 辅助任务重新初始化

从基于 T_{sparse} 和 T_{real} 的构建与优化方式中,可以

看出,一方面由于 T_{sparse} 和 T_{real} 的初始种群基于 T_{main} 的精英解构建,另一方面由于 T_{sparse} 和 T_{real} 仅在限定的搜索空间进行降维搜索,因而其具有快速收敛能力,且所获得的搜索结果可为 T_{main} 的进化提供理想的正向引导作用。

但与此同时,随着迭代次数的增加, T_{sparse} 和 T_{real} 的上述优点,亦可能对 T_{main} 形成负向引导作用,原因在于:

首先, P_{main} 的质量会在进化过程中不断提升, T_{sparse} 和 T_{real} 初始化时所设定恒定向量 dec^* 和 $mask^*$ 的质量可能会越来越差于当前 P_{main} 中精英解所对应的实数向量和二进制向量,导致基于 T_{sparse} 和 T_{real} 所生成种群对 T_{main} 进行引导时,反倒起负作用。

其次,同样随着 P_{main} 质量的提升, T_{sparse} 和 T_{real} 初始化时基于 P_{main} 的精英解所设定的搜索空间可能已经不合理。若仍限定在上述空间进行搜索,导致 T_{sparse} 和 T_{real} 容易陷入局部最优,进而导致基于 T_{sparse} 和 T_{real} 所生成种群对 T_{main} 进行引导时,亦起负作用。

为避免上述问题,在每轮迭代中,采用连续窗口机制,衡量 T_{sparse} 和 T_{real} 对 T_{main} 的引导效果。若 T_{sparse} 或 T_{real} 对 T_{main} 的引导效果不够理想,则基于当前 P_{main} 对 T_{sparse} 或 T_{real} 进行重新初始化,以使 T_{sparse} 和 T_{real} 可持续正向引导 T_{main} 。具体方式如下:

首先,在每轮迭代中,采用如下方式更新 T_{sparse} 和 T_{real} 的KTP值:

$$ktp^{g+1} = \rho * sur^g + (1 - \rho) * ktp^g \quad (4)$$

其中, ktp^g 为 T_{sparse}/T_{real} 第 g 代的KTP值。 sur^g 表示 T_{main} 中,基于 T_{sparse}/T_{real} 的知识所生成个体在第 g 代的存活率。 ρ 为KTP更新系数, ρ 越大,表示KTP的更新越由当前知识迁移所生成子代的存活率决定,反之则越由历史知识迁移效果决定。

其次,为 T_{sparse} 和 T_{real} 分别设置初始值为0的参数 w_s 和 w_r 。若某轮迭代中 T_{sparse} 或 T_{real} 的KTP值小于阈值 θ ,则将 w_s 或 w_r 加1;反之将 w_s 或 w_r 置0。若 w_s 或 w_r 的值大于等于连续窗口阈值 λ ,则调用算法2或算法3重新初始化 T_{sparse} 或 T_{real} ,并将对应的 w_s 或 w_r 置0。

采用连续窗口机制判断 T_{sparse} 和 T_{real} 是否需要重新初始化的原因,一方面是在 T_{sparse} 和 T_{real} 自进化早期,当 T_{sparse} 或 T_{real} 对 T_{main} 的某次或连续较少次引导效果不是非常理想时,并不宜就此认为 T_{sparse} 或 T_{real} 无法继续引导 T_{main} ,原因在于随着迭代次数的

增加, P_{sparse} 和 P_{real} 的质量仍会持续提升, T_{sparse} 和 T_{real} 仍具有继续正向引导 T_{main} 的潜力,过早的重新初始化既无法充分发挥 T_{sparse} 和 T_{real} 的潜力,亦增加不必要的成本。另一方面,在 T_{sparse} 和 T_{real} 自进化晚期, P_{sparse} 和 P_{real} 的质量已趋稳定,若持续较多次仍无法理想地引导 T_{main} ,说明 P_{sparse} 或 P_{real} 已陷入局部最优,已无更大潜力来引导 T_{main} ,此时,则应该进行重新初始化。

3.4 知识迁移

在SLMO-EMT中, T_{main} 自适应地采用自进化或知识迁移方式生成子代。其中,采用自进化方式时,对二进制向量执行单点交叉和按位变异,对实数向量执行模拟二进制交叉和多项式变异。采用知识迁移方式时,则引入了基于轮询和KTP的辅助任务选择策略、基于相似度的知识受体选择策略,以及基于局部交叉的知识利用策略。

具体而言,当从 T_{sparse} 向 T_{main} 进行知识迁移时,从 P_{sparse} 中选取二进制向量作为知识;当从 T_{real} 向 T_{main} 进行知识迁移时,从 P_{real} 中选取的实数向量作为知识。知识迁移的过程如下:首先,基于轮询和KTP,从 T_{sparse} 或 T_{real} 中选取一个作为 T_{main} 的辅助任务,并从所选辅助任务的种群中随机挑选知识。接着,基于相似度在 P_{main} 中选择一个与所选知识最为相关的受体,以确保知识迁移能产生理想的正迁移效果。最后,基于局部交叉的知识利用策略进行知识迁移,以更好地提升 T_{main} 中稀疏位置的探测准确率和非零决策变量的优化程度。

(1) 基于轮询和KTP的辅助任务选择

由于SLMO-EMT包含两个辅助任务,在知识迁移过程中,首先需要考虑如何合理地选择哪个辅助任务。为此,在每轮迭代中,首先将优先标识 $flag$ 随机初始化为0或1,再采用如下方式逐一选择用于生成子代的辅助任务:

若 $flag$ 为0,优先从 T_{sparse} 向 T_{main} 迁移知识。具体而言,若 ktp_s 大于随机数 $rand_1$ ($rand_1 \in [0, 1]$),则选择从 T_{sparse} 向 T_{main} 迁移知识,且将 $flag$ 置1;否则,若 ktp_r 大于随机数 $rand_2$ ($rand_2 \in [0, 1]$),则选择从 T_{real} 向 T_{main} 迁移知识,但不改变 $flag$ 的值;否则, T_{sparse} 和 T_{real} 此次均不向 T_{main} 迁移知识,且不改变 $flag$ 的值。

若 $flag$ 为1,则优先从 T_{real} 向 T_{main} 迁移知识。类似上面做法,但先判断 ktp_r ,并在满足条件时将 $flag$ 置0,再判断 ktp_s 。

采用上述轮询方式选择 T_{sparse} 和 T_{real} 的目的,是为了平衡 T_{sparse} 和 T_{real} 参与知识迁移的频率,避免某个辅助任务相关的知识迁移过于频繁而导致另一个辅助任务难以获得足够的机会。

(2) 基于相似度的知识受体选择

SLMO-EMT 采用双层编码方式,需要合理组合二进制向量和实数向量,才能生成高质量的个体。由于 T_{sparse} 和 T_{real} 分别用于引导 T_{main} 中二进制向量和实数向量的优化,为使基于知识迁移所生成子代亦是相关二进制向量和实数向量的合理组合,在基于 T_{sparse} 或 T_{real} 向 T_{main} 迁移知识生成子代之前,需在 T_{main} 中选择合适的知识受体。

当从 T_{sparse} 向 T_{main} 迁移知识时,首先,采用公式(5)逐一计算 P_{main} 中所有个体的实数向量与 P_{sparse} 的恒定向量 dec^* 之间相似度 sim_{real} :

$$sim_{real_i} = 1 - \frac{\sqrt{\sum_{j=1}^D (dec_j^i - dec_j^*)^2}}{\sqrt{\sum_{j=1}^D (dec_j^u - dec_j^l)^2}} \quad (5)$$

其中, D 为决策变量维度, dec_j^i 表示 P_{main} 中个体 i 所对应实数向量第 j 维的值, dec_j^* 表示 dec^* 第 j 维的值, dec_j^u 和 dec_j^l 分别表示实数向量中第 j 维变量的取值上限和下限。可以看出,两个实数向量间的相似度越大, sim_{real} 的值亦越大。

接着,采用轮盘赌方式,根据 sim_{real} 从 P_{main} 中选择知识迁移受体。 sim_{real_i} 的值越大,则 P_{main} 中个体 i 被选中作为知识迁移受体的概率越大。

由于 T_{sparse} 中的恒定实数向量 dec^* 在初始化时即具有较高质量,且通过搜索已获得与其合理关联的更高质量二进制向量,因此选择 P_{main} 中与 dec^* 具有高相似性的个体作为知识受体,既有利于所生成个体具有高质量的二进制向量,又能确保其二进制向量和实数向量的组合具有合理性。与此同时,未选择 P_{main} 中与 dec^* 最相似的个体作为知识迁移受体,而采用轮盘赌方式,目的在于为基于知识迁移的子代生成引入一定的不确定性,以提升子代种群的多样性,从而降低陷入局部最优的可能。

当从 T_{real} 向 T_{main} 迁移知识时,首先,采用公式(6)逐一计算 P_{main} 中所有个体的二进制向量与 P_{real} 的恒定向量 $mask^*$ 之间相似度 sim_{sparse} :

$$sim_{sparse_i} = \frac{\sum_{j=1}^D (1 - mask_j^i \oplus mask_j^*)}{D} \quad (6)$$

其中, \oplus 为异或运算符, D 为决策变量维度, $mask_j^i$ 表示 P_{main} 中个体 i 所对应二进制向量第 j 维的值, $mask_j^*$

表示 $mask^*$ 第 j 维的值。可以看出,两个二进制向量间的相似度越大, sim_{sparse} 的值亦越大。

接着,采用轮盘赌方式,根据 sim_{sparse} 从 P_{main} 中选择知识迁移受体。 sim_{sparse_i} 的值越大,则 P_{main} 中个体 i 被选中作为知识迁移受体的概率越大。

对应 T_{sparse} , 由于 T_{real} 中的恒定二进制 $mask^*$ 在初始化时亦具有较高质量,且通过搜索已获得与其合理关联的更高质量实数向量,因此选择 P_{main} 中与 $mask^*$ 具有高相似性的个体作为知识受体,既有利于所生成个体具有高质量的实数向量,又能确保其二进制向量和实数向量的组合具有合理性。同样,采用轮盘赌方式选择知识迁移受体,也是为了降低陷入局部最优的可能。

(3) 基于局部交叉的知识利用策略

为利用 T_{sparse} 和 T_{real} 的知识来引导 T_{main} 的进化,在 T_{main} 的子代生成过程中,采用了如下迁移知识引导的局部交叉算子:

当采用 T_{sparse} 引导 T_{main} 进化时,首先,以 P_{sparse} 中随机选择的非支配个体 p^s , 以及基于相似度从 P_{main} 中选择的知识迁移受体 p^m 作为父代。接着,提取 p^s 的二进制向量中所处位置属于 $Space_s$ 的各个变量,将其作为知识 k^s 。最后,采用“与”运算,在对应位置将 k^s 作用于 p^m 的二进制向量,生成 T_{main} 交叉操作后的子代个体 o^s 。

上述交叉过程如图4所示,其中 $Space_s$ 包含的位置用阴影表示。

由于从 p^s 提取的 k^s 已在 T_{sparse} 中进行了相对充分的搜索,采用上述交叉方式,可合理地提升 T_{main} 中个体的稀疏度,从而减少不必要的搜索资源浪费,并加速 T_{main} 的进化。

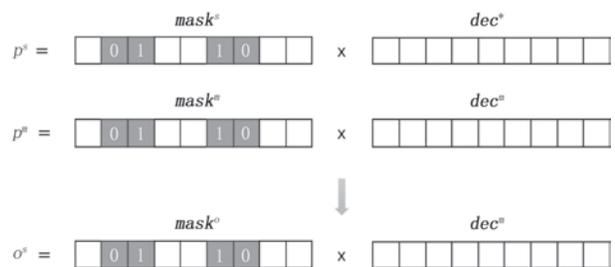


图4 基于“与”运算的交叉

当采用 T_{real} 引导 T_{main} 进化时,首先,以 P_{real} 中随机选择的非支配个体 p^r , 以及基于相似度从 P_{main} 中选择的知识迁移受体 p^m 作为父代。接着,提取 p^r 的实数向量中所处位置属于 $Space_r$ 的各个变量,将其

作为知识 k^r 。然后,采用公式(7),在对应位置将 k^r 作用于 p^m 的实数向量,生成 T_{main} 交叉操作后子代个体 o^r 的实数向量 dec^o :

$$dec_i^o = \left(1 - \frac{|dec_i^m - dec_i^r|}{dec_i^u - dec_i^l}\right) dec_i^m + \frac{|dec_i^m - dec_i^r|}{dec_i^u - dec_i^l} dec_i^r, i \in Space_r \quad (7)$$

其中, dec_i^r 和 dec_i^m 分别表示 p^r 和 p^m 的实数向量中第 i 维变量的值, dec_i^u 和 dec_i^l 分别表示实数向量中第 i 维变量的取值上限和下限, dec_i^o 表示 o^r 的实数向量中第 i 维变量的值。若 $i \notin Space_r$, 则直接将 dec_i^o 的设置为 dec_i^m 。最后,将 p^m 的二进制向量作为 o^r 的二进制向量,但其中位置属于 $Space_r$ 的变量,全部设置为 1。

从公式(7)可以看出, dec_i^r 和 dec_i^m 相差越大,则 dec_i^o 越接近 dec_i^r , 反之越接近 dec_i^m 。由于 dec_i^r 已在 T_{real} 中获得相对较理想的搜索结果,因此基于公式(7)可较好地引导 T_{main} 的进化。值得注意的是, o^r 未直接采用 p^m 的二进制向量,原因在于避免 p^m 二进制向量中原有的稀疏位置遮蔽知识迁移效果。

基于上述知识迁移策略的自适应子代生成伪代码如算法4所示。首先随机初始化优先标识 $flag$ (第1行),然后逐一生成 N 个子代个体(第2-24行)。其中,当 $flag$ 为 0 时,优先从 T_{sparse} 向 T_{main} 迁移知识(第4-11行),当 $flag$ 为 1 时,优先从 T_{real} 向 T_{main} 迁移知识(第13-20行)。每当优先使用一次 T_{sparse} 或 T_{real} 后,下次则优先使用 T_{real} 或 T_{sparse} , 分别通过第6行和第15行设置。第5、8、14、17行先从 P_{sparse} 或 P_{real} 中随机选取一个非支配个体,再基于相似度从 P_{main} 中选择一个知识受体,然后再执行本文所提的局部交叉算子。第10、19行则采用与辅助任务自进化相同的交叉算子。第22行的变异算子亦采用与辅助任务自进化相同的变异算子。

算法4. GenerateOffspring

输入: N (种群大小), P_{main} (T_{main} 种群), P_{sparse} (T_{sparse} 种群), P_{real} (T_{real} 种群), P_{sparse} (T_{sparse} 种群), ktp_s (T_{sparse} 的KTP), ktp_r (T_{real} 的KTP), $Space_s$ (T_{sparse} 搜索空间), $Space_r$ (T_{real} 搜索空间)

输出: O (子代种群)

1. 初始化 $flag$;
2. FOR $i = 1$ to N
3. IF $flag == 0$ THEN
4. IF $rand_1 < ktp_s$ THEN
5. $o \leftarrow$ 对从 P_{sparse} 和 P_{main} 中选择的个体进行交叉;

6. $flag = 1$;
7. ELSE IF $rand_2 < ktp_r$ THEN
8. $o \leftarrow$ 对从 P_{real} 和 P_{main} 中选择的个体进行交叉;
9. ELSE
10. $o \leftarrow$ 从 P_{main} 中随机选择两个个体进行交叉;
11. END IF
12. ELSE
13. IF $rand_3 < ktp_r$ THEN
14. $o \leftarrow$ 对从 P_{real} 和 P_{main} 中选择的个体进行交叉;
15. $flag = 0$;
16. ELSE IF $rand_4 < ktp_s$ THEN
17. $o \leftarrow$ 对从 P_{sparse} 和 P_{main} 中选择的个体进行交叉;
18. ELSE
19. $o \leftarrow$ 从 P_{main} 中随机选择两个个体进行交叉;
20. END IF
21. END IF
22. 对 o 进行变异;
23. $O \leftarrow O \cup \{o\}$;
24. END FOR

3.5 时间复杂度分析

SLMO-EMT 的主体包括 T_{sparse} 和 T_{real} 的自进化、 T_{main} 的自适应进化,以及 T_{sparse} 和 T_{real} 的若干次初始化。在每轮迭代中, T_{sparse} 和 T_{real} 自进化子代生成所需时间为 $O(DN)$, 环境选择所需时间为 $O(MN^2)$, 其中, D 为决策变量维度, N 为种群大小, M 为目标个数。 T_{main} 自适应进化过程中,辅助任务选择所需时间为 $O(N)$, 知识迁移受体选择所需时间为 $O(DN)$, 包含局部交叉算子的子代生成所需时间为 $O(DN)$, 环境选择所需时间为 $O(MN^2)$ 。 T_{sparse} 和 T_{real} 初始化所需时间主要用于种群初始化和搜索空间的限定,分别为 $O(DN)$ 和 $O(Dn)$, 其中, n 为当前 P_{main} 中第一层帕累托非支配解集 F_1 中的个体数。因此,在每轮迭代中,SLMO-EMT 的时间复杂度为 $\max(O(DN), O(MN^2))$ 。

4 实验结果与分析

本节在稀疏大规模多目标优化基准测试问题集 SMOP^[4]和源自实际应用的神经网络训练问题集 (Neural Network Training Problems, NNTP)^[46]上,对 SLMO-EMT 的性能进行综合分析,并与 8 个先进的 SLMOEAs 进行对比。

4.1 实验设计

- (1) 测试问题集

SMOP 包含 8 个测试问题 SMOP1-SMOP8, 每

个问题存在多模性、欺骗性、上位性、低固有维数等挑战中的一个或多个。各测试问题的决策变量维度 D ，目标数量 M ，稀疏度 K 均可自行设定。参照本领域的通用做法，将各测试问题的目标数量 M 设置为2，稀疏度 K 设置为0.1，决策变量维度 D 则分别设置为1000、2000、5000和10 000，共构成32个测试实例。

NNTP为实际应用中的SLMOPs，旨在寻找一个兼具高效与精度的分类神经网络，即在确保分类错误率最低的同时实现网络结构的稀疏性^[47]。NNTP共包含665个数据集，参照本领域的通用做法，从中选取8个构建测试实例。在每个测试实例中，将需训练的神经网络结构设定为包含20个神经元的前馈神经网络，各测试实例的参数如表1所示。

表1 神经网络训练问题集

测试实例	数据集	维度	样本数量	特征数量	类别数量
NNTP1	Spambase	1181	4597	57	2
NNTP2	Connectionist Bench (Sonar)	1241	208	60	2
NNTP3	Hill valley	2041	606	100	2
NNTP4	Movementlibras	2135	360	90	15
NNTP5	Semeion Handwritten Digit	5161	1593	256	10
NNTP6	LSVT Voice Rehabilitation	6241	126	310	2
NNTP7	DARWIN	9041	450	174	2
NNTP8	Madelon	10 041	2600	500	2

(2) 对比算法与参数设置

本文将SLMO-EMT与8个先进的代表性SLMOEAs进行对比，分别为SECSO^[17]、ST-CCPSO^[18]、SparseEA^[4]、MOEAPSL^[6]、PMMOEA^[15]、MSKEA^[16]、S-NSGA-II^[20]和DSGEA^[23]。选取这8个算法进行对比的原因在于SECSO和ST-CCPSO为现有研究中仅有的2个单层编码SLMOEAs，SparseEA为首次采用双层编码的经典SLMOEAs，MSKEA和S-NSGA-II为基于高效搜索算子的代表性双层编码SLMOEAs，MOEAPSL、PMMOEA和DSGEA为基于降维策略的代表性双层编码SLMOEAs。

参照本领域的通用做法^[4]，所有算法使用统一的种群大小和评价次数。在SMOP基准问题集上，种群大小设置为100，评价次数设置为 $100 \times D$ ， D 为各测试实例的决策变量维度。在NNTP上，种群大小设置为50，对于决策变量维度分别在1000、2000、5000、10 000附近的测试实例，评价次数分别设置为20 000、40 000、100 000和200 000。

在所有算法中，交叉概率设置为1，变异概率设置为 $1/D$ ，模拟二进制交叉的分布指标 η_c 和多项式变异的分布指标 η_m 均设置为20。SLMO-EMT相关的特定参数设置如表2所示，其它算法各自特定的参数，参照原始论文。

表2 SLMO-EMT相关的特定参数设置

参数名称	参数值	参数名称	参数值
d_1	0.25	α	0.1
d_2	0.1	ρ	0.6
θ	0.2	λ	5

各算法在版本4.2 PlatEMO^[48]平台运行，硬件环境为intel i7-6700CPU，64GRAM，软件环境为MATLAB2020b。

(3) 评价指标

采用多目标优化领域常用的综合评价指标反转世代距离(Inverted Generational Distance, IGD)^[49]和超体积(Hypervolume, HV)^[50]评估各算法的性能。

IGD的计算公式如下：

$$IGD(P^*, P) = \frac{\sum_{v \in P^*} dis(v, P)}{|P^*|} \quad (8)$$

其中， P^* 代表真实帕累托前沿上一组均匀分布的参考点， P 是算法最终获得的非支配解集，在本文中 P 为任务最终获得的非支配解集。 $dis(v, P)$ 是 P^* 中个体 v 与 P 中个体的最小欧几里得距离。IGD综合评价种群的收敛性和多样性，IGD值越小，说明算法的综合性能越好。

HV的计算公式如下：

$$HV = \delta(\cup_{i=1}^n v_i) \quad (9)$$

其中， δ 表示勒贝格测度， $|P|$ 表示最终所获得非支配解集 P 中个体的数量， v_i 表示 P 中个体 p_i 与参考点 z_i 构成的超体积。HV同样综合评价种群的收敛性和多样性，HV值越大，算法的综合性能越好。

在各测试实例上，每个算法均运行20次，采用置信水平为95%的Wilcoxon秩和检验^[51]，对SLMO-EMT与各对比算法的性能进行衡量，符号“+”、“-”和“ \approx ”分别表示对比算法的性能明显优于、劣于和近似于SLMO-EMT。

4.2 对比实验

(1) SMOP上各算法的实验结果与分析

表3和表4分别为SLMO-EMT与8个对比算法在32个SMOP测试实例上，独立运行20次的平均IGD值和平均HV值，以及对应标准差的统计结

果,在每个测试实例上的最佳结果用粗体表示。

可以看出,SLMO-EMT在绝大多数测试实例上均获得了最佳的IGD均值和HV均值,说明基于所采用的多任务优化框架和优化策略,总体上可明显提升SLMO-EMT在各类SLMOPs上的综合优化性能。

MSKEA在5000维的SMOP4上获得了最佳的IGD均值,在1000维的SMOP5上获得了最佳的HV均值,且同时在2000维、5000维、10000维的SMOP5上获得了最佳的IGD均值和HV均值。原因在于MSKEA引入了三种动态稀疏性知识,并设计了基于知识融合的多阶段进化策略,对于帕累托前沿形状为凸,且非零决策变量优化难度不随所处位置改变的部分高维SMOP4和SMOP5测试实例,

可以更准确地估计二进制向量中每个决策变量为零的概率,从而获得了更好的性能。

SESCO在10000维的SMOP4上获得了最佳的IGD均值,并同时在2000维的SMOP4上获得了最佳的IGD均值和HV均值。原因在于SESCO引入了融合强凸稀疏策略和三类竞争机制的新型搜索算子,对于低固有维数的SMOP4,有利于更准确地定位其决策变量中的稀疏位置。

S-NSGA-II在10000维的SMOP3、5000维和10000维的SMOP4上获得了最佳的HV均值。原因在于S-NSGA-II提出的稀疏模拟二进制交叉算子和稀疏多项式变异算子,在解决具有高欺骗性的SMOP3和低固有性的SMOP4上,维度越高,相对其它算法越具有优异的竞争力。

表3 SLMO-EMT与8个对比算法在32个SMOP测试实例上独立运行20次获得的IGD均值和标准差

测试实例	维度	SECSO	ST-CCPSO	SparseEA	MOEAPSL	PMMOEA	MSKEA	S-NSGA-II	DSGEA	SLMO-EMT
SMOP1	1000	6.2010e-2 (2.03e-3) -	2.7310e-2 (3.86e-3) -	2.5186e-2 (2.71e-3) -	2.2612e-2 (1.49e-3) -	2.1101e-2 (2.30e-3) -	1.3698e-2 (2.35e-3) -	1.4507e-2 (4.96e-3) -	6.1885e-3 (8.09e-4) -	5.6512e-3 (6.15e-4)
	2000	6.2788e-2 (1.94e-3) -	2.8123e-2 (2.28e-3) -	3.2714e-2 (2.25e-3) -	2.7407e-2 (2.83e-3) -	2.6697e-2 (2.15e-3) -	1.9725e-2 (2.38e-3) -	1.6063e-2 (5.92e-3) -	8.4877e-3 (1.61e-3) -	8.1221e-3 (3.21e-3)
	5000	6.3598e-2 (1.12e-3) -	2.9708e-2 (4.52e-3) -	3.9199e-2 (3.99e-3) -	3.9143e-2 (2.75e-3) -	3.3983e-2 (1.47e-3) -	2.7581e-2 (2.08e-3) -	1.9196e-2 (3.51e-3) -	3.3558e-2 (2.96e-3) -	1.4292e-2 (3.68e-3)
	10000	6.3879e-2 (1.96e-3) -	3.2614e-2 (2.46e-3) -	4.7795e-2 (7.77e-3) -	5.7853e-2 (6.27e-3) -	3.5787e-2 (3.36e-3) -	3.3376e-2 (1.47e-3) -	2.9757e-2 (1.12e-3) -	6.4290e-2 (2.94e-3) -	2.1583e-2 (5.12e-3)
SMOP2	1000	1.3186e-1 (5.60e-3) -	7.1724e-2 (5.44e-2) -	6.6328e-2 (5.98e-3) -	5.0128e-2 (3.86e-3) -	5.1845e-2 (4.53e-3) -	4.4298e-2 (5.28e-3) -	4.9822e-2 (1.51e-2) -	1.8399e-2 (2.20e-4) -	1.1768e-2 (3.45e-3)
	2000	1.3466e-1 (3.82e-3) -	7.6242e-2 (4.22e-2) -	8.8935e-2 (3.42e-3) -	6.1287e-2 (5.16e-3) -	6.4177e-2 (4.79e-3) -	5.5172e-2 (3.99e-3) -	6.3977e-2 (1.68e-2) -	2.1639e-2 (2.81e-3) -	1.4621e-2 (5.11e-3)
	5000	1.3575e-1 (2.02e-3) -	8.0658e-2 (4.81e-3) -	1.0310e-1 (3.85e-3) -	8.1989e-2 (7.73e-3) -	8.2830e-2 (2.16e-3) -	7.3770e-2 (2.70e-3) -	8.5698e-2 (1.44e-2) -	3.8712e-2 (6.45e-3) -	2.7651e-2 (5.89e-3)
	10000	1.3615e-1 (6.59e-3) -	8.2465e-2 (3.87e-3) ≈	2.1450e-1 (3.13e-3) -	1.1157e-1 (1.81e-3) -	9.3118e-2 (2.67e-3) -	8.3406e-2 (3.09e-3) -	1.0921e-1 (4.86e-2) -	1.3930e-1 (3.89e-3) -	8.1868e-2 (6.73e-3)
SMOP3	1000	6.1651e-2 (2.93e-3) -	2.3596e-2 (8.39e-3) -	2.8992e-2 (2.91e-3) -	2.3601e-2 (1.96e-3) -	2.2586e-2 (2.21e-3) -	9.6286e-3 (1.80e-3) -	1.8363e-2 (4.00e-3) -	5.9380e-3 (1.37e-3) -	5.2312e-3 (2.31e-3)
	2000	6.3951e-2 (1.78e-3) -	2.4419e-2 (8.06e-3) -	3.5154e-2 (2.79e-3) -	2.5421e-2 (5.45e-3) -	2.7410e-2 (1.70e-3) -	1.4411e-2 (2.32e-3) -	2.2059e-2 (4.13e-3) -	6.2776e-3 (9.32e-3) -	5.7239e-3 (3.19e-3)
	5000	6.4101e-2 (3.65e-3) -	2.6813e-2 (5.54e-3) -	4.3775e-2 (7.59e-3) -	3.6860e-2 (2.24e-3) -	3.3230e-2 (6.08e-3) -	2.2641e-2 (1.29e-3) -	2.3859e-2 (4.71e-3) -	2.0522e-2 (4.94e-3) -	1.7642e-2 (4.91e-3)
	10000	6.5339e-2 (2.90e-4) -	2.9827e-2 (5.34e-4) -	4.9652e-2 (6.62e-3) -	4.1267e-2 (5.79e-3) -	3.9215e-2 (5.79e-3) -	2.5902e-2 (3.21e-3) ≈	2.5003e-2 (3.21e-3) ≈	5.6102e-2 (2.59e-3) -	2.3475e-2 (5.37e-3)
SMOP4	1000	3.9912e-3 (7.35e-5) ≈	4.1377e-3 (6.13e-4) -	4.7891e-3 (2.10e-4) -	5.1048e-3 (7.67e-4) -	4.0763e-3 (7.55e-5) ≈	4.1390e-3 (6.12e-5) -	4.8010e-3 (2.89e-4) -	4.7176e-3 (1.77e-4) -	3.9719e-3 (3.75e-5)
	2000	4.0051e-3 (7.06e-5) +	4.0956e-3 (1.89e-4) -	4.7554e-3 (1.85e-4) -	5.3205e-3 (3.62e-4) -	4.0821e-3 (5.56e-5) -	4.1352e-3 (5.36e-5) -	4.8284e-3 (2.15e-4) -	4.7389e-3 (3.11e-4) -	4.1236e-3 (3.71e-4)
	5000	4.0161e-3 (3.36e-6) -	4.2022e-3 (7.74e-4) -	4.8474e-3 (1.66e-4) -	5.3484e-3 (4.19e-5) -	4.1292e-3 (6.86e-5) -	3.9990e-3 (5.21e-5) +	4.8370e-3 (2.61e-4) -	4.7563e-3 (2.78e-4) -	4.1471e-3 (4.12e-5)
	10000	4.1233e-3 (2.45e-4) +	4.2137e-3 (7.45e-5) ≈	5.0535e-3 (3.72e-4) -	5.3424e-3 (4.79e-4) -	4.1765e-3 (4.56e-5) +	4.2567e-3 (3.97e-5) -	4.8693e-3 (4.77e-4) -	4.7405e-3 (2.69e-4) -	4.2389e-3 (5.51e-4)

续表

测试实例	维度	SECSO	ST-CCPSO	SparseEA	MOEAPSL	PMMOEA	MSKEA	S-NSGA-II	DSGEA	SLMO-EMT	
SMOP5	1000	5.1565e-3	6.2063e-3	5.8628e-3	9.0587e-3	4.5243e-3	4.2072e-3	4.8453e-3	4.8413e-3	4.1471e-3	
		(7.20e-4) -	(4.21e-4) -	(3.30e-4) -	(4.57e-4) -	(9.83e-5) -	(7.68e-4)≈	(2.14e-4) -	(4.74e-4) -	(3.12e-5)	
	2000	6.2707e-3	6.2806e-3	5.9531e-3	9.1766e-3	4.5692e-3	4.2492e-3	7.8179e-3	4.8765e-3	4.3212e-3	
		(1.65e-3) -	(3.71e-4) -	(1.88e-4) -	(3.99e-4) -	(9.92e-5) -	(7.43e-4)≈	(3.25e-4) -	(1.54e-4) -	(7.93e-4)	
	5000	8.3843e-3	7.4672e-3	8.6021e-3	9.1737e-3	4.7921e-3	4.3241e-3	2.1217e-2	5.5788e-3	5.1137e-3	
		(3.45e-3) -	(7.34e-4) -	(7.69e-4) -	(1.09e-4) -	(4.12e-4)≈	(1.50e-4) +	(3.65e-4) -	(8.15e-4) -	(6.31e-4)	
	10 000	8.7586e-3	8.0782e-3	9.1049e-3	9.1989e-3	5.2257e-3	4.4245e-3	2.6957e-2	2.7043e-2	6.2874e-3	
		(1.43e-3) -	(8.45e-4) -	(6.39e-4) -	(2.95e-4) -	(6.12e-4) -	(8.31e-4) +	(6.08e-4) -	(7.81e-4) -	(9.11e-4)	
	SMOP6	1000	4.0324e-3	6.3173e-3	7.2246e-3	1.1130e-2	4.7750e-3	4.4990e-3	4.2176e-3	5.2113e-3	3.5215e-3
			(1.37e-3) -	(4.15e-4) -	(3.92e-4) -	(8.01e-4) -	(1.42e-4) -	(9.52e-5) -	(2.37e-4) -	(2.64e-4) -	(2.11e-5)
		2000	8.6465e-3	6.8280e-3	7.4876e-3	1.2120e-2	4.8499e-3	4.7005e-3	4.7798e-3	5.6377e-3	4.1742e-3
			(1.03e-3) -	(4.28e-4) -	(3.36e-4) -	(6.04e-4) -	(1.14e-4) -	(1.00e-4) -	(3.62e-4) -	(2.62e-4) -	(6.72e-4)
5000		8.8186e-3	7.3691e-3	7.9474e-3	1.3685e-2	5.3821e-3	4.9772e-3	7.2087e-3	6.3128e-3	4.6936e-3	
		(1.23e-3) -	(5.17e-4) -	(3.04e-4) -	(5.18e-4) -	(4.30e-4) -	(9.11e-4) -	(3.04e-4) -	(2.44e-4) -	(6.25e-4)	
10 000		9.1429e-3	1.3217e-2	1.8046e-2	1.5675e-2	6.2315e-3	5.8323e-3	1.5612e-2	1.8323e-2	5.7221e-3	
		(1.29e-3) -	(5.84e-4) -	(9.62e-4) -	(1.04e-4) -	(7.18e-4) -	(4.18e-3)≈	(3.19e-4) -	(4.18e-4) -	(7.67e-4)	
SMOP7		1000	1.8892e-1	8.4268e-2	8.7057e-2	1.1856e-1	7.0207e-2	5.3941e-2	7.1952e-2	2.2305e-2	1.8282e-2
			(5.49e-3) -	(2.72e-2) -	(7.14e-3) -	(1.01e-3) -	(8.87e-3) -	(6.44e-3) -	(2.07e-2) -	(2.98e-3) -	(4.13e-3)
		2000	1.9691e-1	9.5730e-2	1.1277e-1	1.4165e-1	8.9446e-2	7.8836e-2	1.0437e-1	2.6693e-2	2.3152e-2
			(4.01e-3) -	(2.24e-2) -	(5.69e-3) -	(2.91e-2) -	(5.86e-3) -	(5.78e-3) -	(2.97e-2) -	(1.90e-3) -	(6.81e-3)
	5000	1.9790e-1	1.0612e-1	1.2635e-1	1.5323e-1	1.1247e-1	9.2124e-2	1.1996e-1	3.1760e-2	2.8179e-2	
		(5.18e-3) -	(3.37e-2) -	(2.65e-3) -	(4.57e-3) -	(8.84e-2) -	(2.65e-3) -	(2.28e-2) -	(3.52e-3) -	(6.11e-3)	
	10 000	2.1875e-1	1.1328e-1	1.7665e-1	1.9321e-1	1.5341e-1	2.1134e-1	1.6538e-1	4.2826e-2	3.6988e-2	
		(6.71e-3) -	(3.21e-2) -	(2.94e-2) -	(7.11e-2) -	(7.91e-2) -	(9.81e-3) -	(4.36e-2) -	(2.28e-3) -	(6.72e-3)	
	SMOP8	1000	4.1717e-1	2.6158e-1	2.4411e-1	3.7858e-1	1.9987e-1	1.8154e-1	1.7098e-1	1.6419e-1	1.2298e-1
			(1.20e-2) -	(2.29e-2) -	(1.80e-3) -	(2.28e-2) -	(1.92e-2) -	(1.22e-2) -	(2.60e-3) -	(1.19e-2) -	(3.21e-3)
		2000	4.2475e-1	2.7316e-1	2.8319e-1	4.2062e-1	2.3554e-1	2.1281e-1	2.0732e-1	1.7331e-1	1.6397e-1
			(2.52e-2) -	(7.07e-3) -	(1.59e-3) -	(2.10e-2) -	(1.24e-2) -	(1.74e-2) -	(1.93e-3) -	(2.35e-2) -	(6.21e-3)
5000		4.2715e-1	2.8854e-1	3.2483e-1	4.8013e-1	2.7633e-1	2.2626e-1	2.1170e-1	1.7807e-1	1.7252e-1	
		(1.71e-2) -	(3.89e-2) -	(7.14e-3) -	(1.18e-2) -	(9.86e-3) -	(8.78e-2) -	(1.44e-3) -	(4.81e-3)≈	(5.82e-3)	
10 000		4.3158e-1	2.9109e-1	3.6426e-1	4.8915e-1	2.8912e-1	2.3192e-1	2.4314e-1	1.8688e-1	1.8026e-1	
		(4.31e-2) -	(8.31e-3) -	(4.90e-3) -	(6.89e-3) -	(4.21e-3) -	(3.17e-2) -	(1.04e-3) -	(5.70e-3)≈	(6.98e-3)	
+/-/≈		2/29/1	0/30/2	0/32/0	0/32/0	1/29/2	3/25/4	0/31/1	0/30/2		

DSGEA 在 10 000 维的 SMOP2 和 5000 维的 SMOP3 上获得了最佳的 HV 均值。原因在于 DSGEA 基于决策变量的稀疏度对二进制向量进行平均分组,并在迭代过程中动态调整分组数量,有利于处理带欺骗性,且帕累托前沿为线性的 SMOP2 和 SMOP3 测试实例。

为更直观地展示各算法的对比性能,图 5 给出

了 SLMO-EMT 与 8 个对比算法在 2 目标 5000 维的 SMOP3 和 SMOP7 测试实例上,非支配解集在目标空间的分布情况。图中所示非支配解集是各算法在 20 次独立运行中所获得的最接近于 IGD 均值的非支配解集。可以看出,SLMO-EMT 在这两个测试实例的目标空间上都具有更好的收敛性和多样性。

表 4 SLMO-EMT 与 8 个对比算法在 32 个 SMOP 测试实例上独立运行 20 次获得的 HV 均值和标准差

测试实例	维度	SECSO	ST-CCPSO	SparseEA	MOEAPSL	PMMOEA	MSKEA	S-NSGA-II	DSGEA	SLMO-EMT
SMOP1	1000	5.0945e-1	5.6282e-1	5.5132e-1	5.5516e-1	5.6271e-1	5.6863e-1	5.6314e-1	5.7819e-1	5.8284e-1
		(2.58e-3) -	(3.54e-2) -	(7.56e-3) -	(3.35e-3) -	(2.17e-3) -	(1.62e-3) -	(1.26e-2) -	(2.13e-4) -	(1.57e-4)
	2000	5.0449e-1	5.5269e-1	5.3986e-1	5.5052e-1	5.5441e-1	5.5728e-1	5.5435e-1	5.7365e-1	5.7831e-1
		(1.15e-3) -	(5.27e-3) -	(4.03e-3) -	(1.23e-3) -	(1.47e-3) -	(6.81e-4) -	(3.44e-3) -	(2.34e-3) -	(2.29e-4)
	5000	5.0314e-1	5.4816e-1	5.3547e-1	5.3536e-1	5.4376e-1	5.4940e-1	5.4026e-1	5.4156e-1	5.5269e-1
		(2.18e-3) -	(5.19e-3) -	(2.33e-3) -	(1.66e-3) -	(2.38e-4) -	(1.16e-3) -	(4.79e-3) -	(3.77e-3) -	(2.63e-4)

续表

测试实例	维度	SECSO	ST-CCPSO	SparseEA	MOEAPSL	PMMOEA	MSKEA	S-NSGA-II	DSGEA	SLMO-EMT
SMOP2	10 000	5.0376e-1 (1.47e-3) -	5.3793e-1 (4.87e-3)≈	4.8736e-1 (8.26e-3) -	5.2592e-1 (4.15e-3) -	5.3768e-1 (1.98e-3) -	5.4296e-1 (5.06e-4)≈	5.3474e-1 (5.13e-3) -	5.0218e-1 (4.23e-3) -	5.4595e-1 (4.29e-4)
	1000	4.1011e-1 (2.67e-3) -	5.2759e-1 (4.92e-2) -	5.0257e-1 (1.89e-3) -	5.1529e-1 (6.93e-4) -	5.1064e-1 (7.79e-3) -	5.3199e-1 (5.88e-3) -	5.3513e-1 (2.10e-2) -	5.5933e-1 (8.51e-4)≈	5.6282e-1 (6.26e-3)
	2000	4.0997e-1 (3.69e-4) -	5.0274e-1 (4.07e-2) -	4.7105e-1 (7.52e-3) -	5.1244e-1 (5.13e-3) -	5.0056e-1 (3.55e-3) -	5.1349e-1 (6.35e-3) -	5.0283e-1 (9.84e-3) -	5.4257e-1 (3.58e-3)≈	5.4918e-1 (4.26e-3)
	5000	4.0998e-1 (3.53e-3) -	4.8618e-1 (5.26e-2) -	4.5485e-1 (2.40e-3) -	4.7429e-1 (7.13e-3) -	4.8116e-1 (3.18e-3) -	4.9348e-1 (6.93e-3) -	4.7201e-1 (6.14e-3) -	5.3514e-1 (8.25e-4)≈	5.3991e-1 (2.69e-3)
	10 000	4.1121e-1 (3.09e-3) -	4.1869e-1 (8.76e-2) -	3.8275e-1 (4.27e-3) -	4.2975e-1 (5.21e-3) -	4.6613e-1 (1.14e-3) -	4.7951e-1 (7.44e-4) -	4.5068e-1 (1.32e-2) -	4.9879e-1 (5.45e-3)≈	4.9793e-1 (9.25e-4)
SMOP3	1000	5.0656e-1 (3.98e-3) -	5.7821e-1 (6.95e-3)≈	5.4566e-1 (1.69e-3) -	5.5869e-1 (1.80e-3) -	5.5300e-1 (3.92e-4) -	5.7090e-1 (6.00e-4) -	5.6388e-1 (2.62e-3) -	5.7780e-1 (2.52e-3) -	5.8126e-1 (5.83e-4)
	2000	5.0273e-1 (4.29e-3) -	5.3189e-1 (6.21e-3) -	5.3891e-1 (4.13e-4) -	5.5483e-1 (1.73e-3) -	5.5060e-1 (1.88e-3) -	5.6671e-1 (2.45e-3) -	5.5330e-1 (8.45e-3) -	5.7291e-1 (1.66e-3)≈	5.7652e-1 (1.55e-3)
	5000	5.0506e-1 (2.17e-3) -	5.2879e-1 (5.29e-3) -	5.1643e-1 (7.26e-3) -	5.4731e-1 (7.27e-3) -	5.4488e-1 (3.12e-3) -	5.5568e-1 (1.77e-3)≈	5.5186e-1 (3.60e-3)≈	5.5988e-1 (6.12e-3) +	5.5872e-1 (4.87e-3)
	10 000	5.0366e-1 (3.32e-3) -	5.1748e-1 (8.26e-3) -	5.0286e-1 (9.32e-3) -	5.3421e-1 (6.12e-2) -	5.3707e-1 (3.88e-4) -	5.4852e-1 (2.67e-3)≈	5.5261e-1 (3.39e-3) +	5.1276e-1 (3.32e-3) -	5.4949e-1 (3.16e-3)
SMOP4	1000	8.1859e-1 (9.50e-7) -	8.1849e-1 (6.63e-5)≈	8.1871e-1 (2.10e-4)≈	8.1851e-1 (2.98e-4) -	8.1868e-1 (3.29e-5)≈	8.1869e-1 (9.88e-5)≈	8.1819e-1 (2.96e-5) -	8.1809e-1 (9.23e-5) -	8.1887e-1 (9.88e-5)
	2000	8.1867e-1 (8.32e-5)≈	8.1812e-1 (7.21e-5)≈	8.1826e-1 (2.22e-5)≈	8.1805e-1 (3.62e-4) -	8.1802e-1 (1.58e-5) -	8.1834e-1 (6.54e-5)≈	8.1826e-1 (2.57e-5)≈	8.1797e-1 (2.93e-4) -	8.1824e-1 (9.25e-5)
	5000	8.1765e-1 (7.92e-5)≈	8.1791e-1 (6.95e-5) +	8.1815e-1 (4.79e-5) +	8.1776e-1 (7.28e-4)≈	8.1784e-1 (6.22e-5)≈	8.1816e-1 (7.16e-5)≈	8.1819e-1 (1.32e-5) +	8.1749e-1 (2.26e-4)≈	8.1757e-1 (7.16e-5)
	10 000	8.1749e-1 (5.45e-5)≈	8.1726e-1 (4.21e-4)≈	8.1801e-1 (5.11e-5) +	8.1749e-1 (6.59e-4)≈	8.1751e-1 (8.95e-5)≈	8.1805e-1 (8.63e-5) +	8.1809e-1 (1.97e-5) +	8.1718e-1 (2.03e-4)≈	8.1705e-1 (8.63e-5)
SMOP5	1000	8.0839e-1 (2.35e-3) -	8.1722e-1 (3.64e-5)≈	8.1477e-1 (7.53e-5) -	8.1109e-1 (4.27e-4) -	8.1717e-1 (1.20e-4)≈	8.1822e-1 (1.41e-4) +	8.1754e-1 (2.78e-5)≈	8.1670e-1 (5.38e-4) -	8.1785e-1 (3.76e-4)
	2000	8.0680e-1 (2.65e-4) -	8.1692e-1 (4.91e-4)≈	8.1532e-1 (1.09e-4) -	8.1084e-1 (3.04e-4) -	8.1691e-1 (2.67e-4)≈	8.1812e-1 (7.78e-5) +	8.1490e-1 (2.11e-3) -	8.1708e-1 (2.46e-4)≈	8.1695e-1 (3.57e-4)
	5000	8.1004e-1 (4.27e-3)≈	8.1297e-1 (1.67e-3) +	8.1482e-1 (2.60e-4) +	8.0246e-1 (5.56e-4) -	8.1674e-1 (2.15e-4) +	8.1795e-1 (1.00e-4) +	7.9349e-1 (8.39e-4) -	8.1573e-1 (2.81e-4) +	8.1095e-1 (1.00e-4)
	10 000	8.1104e-1 (1.67e-3) +	8.0039e-1 (4.86e-3) +	8.1356e-1 (6.29e-4) +	7.8937e-1 (2.58e-3)≈	8.1587e-1 (5.27e-4) +	8.1769e-1 (1.39e-5) +	7.8963e-1 (1.45e-3)≈	7.9028e-1 (8.54e-4)≈	7.8572e-1 (3.26e-4)
SMOP6	1000	8.1759e-1 (2.28e-3) -	8.1973e-1 (6.93e-4) -	8.1315e-1 (9.38e-4) -	8.0952e-1 (1.23e-3) -	8.1647e-1 (2.03e-4) -	8.1742e-1 (1.80e-4) -	8.1761e-1 (1.17e-4) -	8.2573e-1 (3.26e-4) -	8.3193e-1 (5.26e-4)
	2000	8.1499e-1 (1.83e-3) -	8.1742e-1 (3.16e-4) -	8.1248e-1 (9.90e-4) -	8.0796e-1 (5.72e-4) -	8.1558e-1 (2.87e-5) -	8.1651e-1 (3.28e-5) -	8.1627e-1 (1.52e-4) -	8.1982e-1 (5.98e-4)≈	8.2182e-1 (6.85e-4)
	5000	8.1295e-1 (9.56e-4) -	8.1489e-1 (4.84e-3) -	8.1177e-1 (6.95e-4) -	8.0268e-1 (3.19e-4) -	8.1393e-1 (2.76e-4) -	8.1532e-1 (1.60e-4) -	8.1359e-1 (4.68e-3) -	8.1521e-1 (6.29e-3) -	8.1796e-1 (7.20e-4)
	10 000	8.0052e-1 (5.69e-4) -	8.0417e-1 (3.64e-3) -	8.1082e-1 (4.12e-4) -	8.0136e-1 (5.24e-4) -	8.0929e-1 (4.21e-4) -	8.1171e-1 (5.12e-4) -	8.0281e-1 (1.98e-3) -	8.0694e-1 (5.21e-3) -	8.1336e-1 (8.28e-4)
SMOP7	1000	1.1725e-1 (1.34e-4) -	2.9126e-1 (7.86e-3) -	2.4599e-1 (1.64e-3) -	1.6876e-1 (1.50e-4) -	2.6393e-1 (8.82e-3) -	2.8061e-1 (8.56e-3) -	2.7320e-1 (3.26e-3) -	3.1413e-1 (4.26e-3) -	3.1747e-1 (3.26e-3)
	2000	1.1599e-1 (3.94e-3) -	2.0825e-1 (5.21e-3) -	1.9395e-1 (1.87e-3) -	9.5158e-2 (7.06e-3) -	2.3332e-1 (5.05e-4) -	2.3653e-1 (3.06e-3) -	2.0307e-1 (1.49e-2) -	3.1269e-1 (2.48e-3) -	3.1575e-1 (4.29e-3)
	5000	1.1411e-1 (3.15e-3) -	1.6847e-1 (5.97e-3) -	1.8136e-1 (4.26e-3) -	8.2736e-2 (6.73e-3) -	1.9828e-1 (1.23e-3) -	2.2210e-1 (3.53e-3) -	1.9476e-1 (4.83e-2) -	3.0621e-1 (5.09e-3) -	3.1152e-1 (5.91e-3)
	10 000	1.1382e-1 (5.29e-3) -	1.3674e-1 (6.23e-3) -	1.6735e-1 (8.14e-3) -	8.1692e-2 (8.27e-3) -	1.8973e-1 (3.91e-3) -	2.1837e-1 (5.29e-3) -	1.4202e-1 (1.80e-3) -	2.8470e-1 (3.18e-3)≈	2.9821e-1 (6.43e-3)

续表

测试实例	维度	SECSO	ST-CCPSO	SparseEA	MOEAPSL	PMMOEA	MSKEA	S-NSGA-II	DSGEA	SLMO-EMT
SMOP8	1000	1.0768e-1 (3.33e-4)	1.2296e-1 (2.63e-2)	9.0006e-2 (1.15e-2)	3.5305e-2 (2.50e-3)	1.1673e-1 (7.11e-3)	1.3386e-1 (5.80e-3)	1.3731e-1 (2.00e-3)	1.4357e-1 (1.05e-2)	1.6317e-1 (3.36e-3)
	2000	8.2054e-2 (3.26e-2)	1.1539e-1 (4.75e-2)	6.2863e-2 (4.98e-3)	1.5439e-2 (1.18e-2)	8.8098e-2 (1.40e-2)	1.0724e-1 (2.92e-3)	1.1940e-1 (1.28e-3)	1.2782e-1 (1.87e-2)	1.4793e-1 (6.45e-3)
	5000	6.1268e-2 (3.29e-3)	1.0927e-1 (5.29e-2)	4.2555e-2 (3.25e-3)	4.8800e-3 (3.27e-2)	6.4568e-2 (2.60e-3)	8.5842e-2 (1.33e-3)	1.0748e-1 (4.03e-3)	1.1493e-1 (4.28e-3)	1.2895e-1 (7.95e-3)
	10 000	4.4596e-2 (6.84e-3)	9.1936e-2 (5.93e-2)	2.7136e-2 (5.91e-3)	2.6643e-3 (6.45e-2)	4.2936e-2 (2.60e-3)	6.3796e-2 (4.27e-3)	8.4937e-2 (5.69e-3)	1.0462e-1 (3.98e-3)	1.0979e-1 (8.84e-3)
+/-/≈		1/27/4	3/22/7	4/26/2	0/29/3	2/25/5	5/21/6	3/25/4	2/18/12	

(2) NNTP上各算法的实验结果与分析

由于NNTP各问题的真实帕累托前沿未知,无法计算各测试实例的IGD值。采用本领域的通用做法,在实验过程中通过设置参考点(1,1)来计算各测试实例的HV值。

表5为SLMO-EMT与8个对比算法在8个

NNTP测试实例上,独立运行20次的平均HV值和标准差统计结果,在每个测试实例上的最佳结果同样用粗体表示。

可以看出,SLMO-EMT在NNTP1-NNTP7上均获得了最佳的HV均值。在NNTP8上,则与最佳HV均值在置信水平为95%的Wilcoxon秩和检验

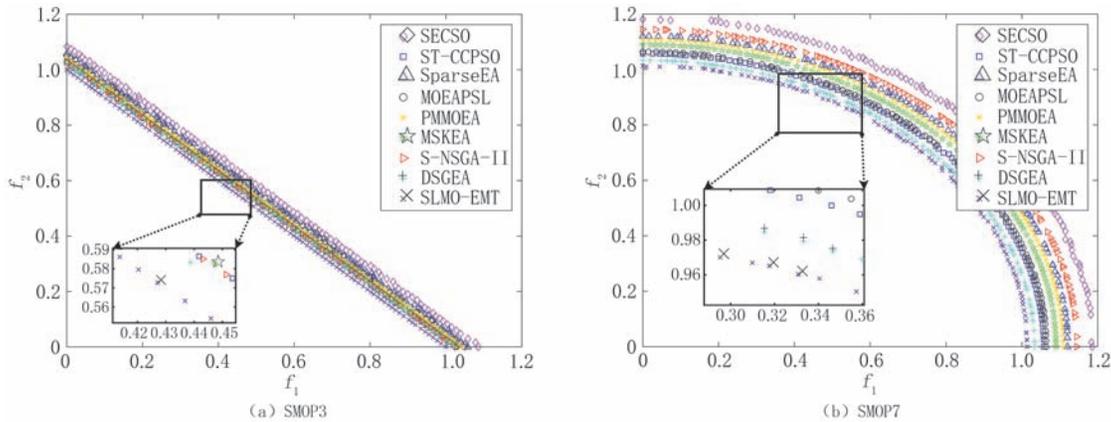


图5 SLMO-EMT与8个对比算法在部分SMOP测试实例目标空间上的种群分布

表5 SLMO-EMT与8个对比算法在8个NNTP测试实例上独立运行20次获得的HV均值和标准差

测试实例	维度	SECSO	ST-CCPSO	SparseEA	MOEAPSL	PMMOEA	MSKEA	S-NSGA-II	DSGEA	SLMO-EMT
NNTP1	1181	9.2829e-1 (6.11e-2)	9.2813e-1 (7.21e-2)	8.8121e-1 (3.82e-2)	9.1219e-1 (4.12e-2)	9.2212e-1 (9.19e-2)	9.2431e-1 (6.32e-2)	9.3192e-1 (6.41e-2)	9.3223e-1 (7.21e-2)	9.4929e-1 (4.21e-2)
NNTP2	1241	8.1625e-1 (4.02e-2)	8.6862e-1 (3.12e-2)	8.0297e-1 (1.18e-2)	8.4569e-1 (2.08e-2)	8.2197e-1 (8.91e-3)	8.3582e-1 (1.83e-2)	8.7797e-1 (8.91e-3)≈	8.7115e-1 (9.81e-3)≈	8.8271e-1 (1.18e-2)
NNTP3	2041	6.5113e-1 (8.42e-2)	6.8553e-1 (6.12e-2)	7.5020e-1 (9.24e-2)	7.7553e-1 (7.84e-2)	6.8387e-1 (3.49e-2)	7.3223e-1 (8.42e-2)	7.2531e-1 (5.84e-2)	8.0953e-1 (7.64e-2)≈	8.1682e-1 (6.21e-2)
NNTP4	2135	6.2686e-1 (1.13e-1)	6.3431e-1 (6.32e-2)	6.4643e-1 (6.64e-2)	6.3369e-1 (7.77e-2)	4.6557e-1 (1.02e-1)	6.9121e-1 (3.82e-2)	5.3192e-1 (3.57e-2)	7.2672e-1 (6.12e-2)	7.6722e-1 (5.64e-2)
NNTP5	5161	8.9216e-1 (9.19e-2)	9.2129e-1 (6.32e-2)	9.1765e-1 (1.69e-2)	9.3562e-1 (8.99e-3)	9.5383e-1 (8.54e-3)	9.7624e-1 (1.47e-2)≈	9.6231e-1 (5.84e-2)	9.7862e-1 (3.12e-2)≈	9.8723e-1 (4.42e-3)
NNTP6	6241	9.4229e-1 (2.13e-2)	9.4121e-1 (9.11e-3)	9.6177e-1 (9.13e-3)≈	9.2939e-1 (1.43e-2)	9.2095e-1 (1.76e-2)	9.6241e-1 (1.02e-2)≈	9.5917e-1 (1.13e-2)	9.6746e-1 (2.48e-2)≈	9.8162e-1 (9.42e-3)
NNTP7	9041	9.2187e-1 (7.26e-2)	9.1286e-1 (5.21e-2)	9.6872e-1 (4.69e-2)	9.6572e-1 (6.31e-2)	9.1864e-1 (3.97e-2)	9.5031e-1 (8.27e-3)	9.8528e-1 (4.21e-2)≈	9.6391e-1 (2.57e-2)	9.9129e-1 (3.12e-2)
NNTP8	10 041	6.5687e-1 (2.36e-2)	6.8121e-1 (3.82e-2)	6.2468e-1 (4.29e-2)	6.6986e-1 (1.65e-2)	6.7113e-1 (8.42e-2)	7.1816e-1 (2.45e-2)	7.8717e-1 (8.72e-3)≈	8.0231e-1 (1.84e-2)≈	7.8857e-1 (7.29e-3)
+/-/≈		0/8/0	0/8/0	0/7/1	0/8/0	0/8/0	0/6/2	0/5/3	0/3/5	

上近似。说明基于所采用的多任务优化框架和优化策略, SLMO-EMT 在实际应用问题上亦具有优异的综合优化性能。

4.3 消融实验

为进一步分析 SLMO-EMT 中各组成部分所起的作用, 本节对 SLMO-EMT 进行消融实验。

仅使用实数优化辅助任务 T_{real} , 不使用稀疏优化辅助任务 T_{sparse} 的变种算法称为 SLMO-EMT/S。仅使用稀疏优化辅助任务 T_{sparse} , 不使用实数优化辅助任务 T_{real} 的变种算法称为 SLMO-EMT/R。保留 ktp 更新机制, 但取消辅助任务重新初始化策略的变

种算法称为 SLMO-EMT/RE。同时取消 ktp 更新机制和辅助任务重新初始化策略的变种算法称为 SLMO-EMT/KR。用随机选择方式取代基于轮询和 KTP 的辅助任务选择策略的变种算法称为 SLMO-EMT/P。用随机选择方式取代基于相似度的知识受体选择策略的变种算法称为 SLMO-EMT/KN。使用模拟二进制交叉和单点交叉取代基于局部交叉的知识利用策略的变种算法称为 SLMO-EMT/KU。

表 6 为 SLMO-EMT 与 7 个变种算法在 16 个 SMOP 测试实例上, 独立运行 20 次的平均 IGD 值和

表 6 SLMO-EMT 与 7 个变种算法在部分 SMOP 测试实例上独立运行 20 次获得的 IGD 均值与标准差

测试实例	维度	SLMO-EMT/S	SLMO-EMT/R	SLMO-EMT/RE	SLMO-EMT/KR	SLMO-EMT/P	SLMO-EMT/KN	SLMO-EMT/KU	SLMO-EMT
SMOP1	1000	1.4128e-2 (8.94e-4) -	7.3466e-3 (7.15e-4) -	1.8084e-2 (5.08e-4) -	2.1701e-2 (9.14e-4) -	1.1221e-2 (1.54e-3) -	1.2541e-2 (1.14e-3) -	1.3614e-2 (7.26e-4) -	5.6512e-3 (6.15e-4)
	2000	2.0305e-2 (1.70e-3) -	1.0559e-2 (1.36e-3) -	2.5991e-2 (9.64e-4) -	3.1189e-2 (1.74e-3) -	1.6127e-2 (2.93e-3) -	1.8025e-2 (2.16e-3) -	1.9566e-2 (1.38e-3) -	8.1221e-3 (3.21e-3)
	5000	3.5730e-2 (1.65e-3) -	1.8580e-2 (1.32e-3) -	4.5734e-2 (9.37e-4) -	5.4881e-2 (1.69e-3) -	2.8378e-2 (2.85e-3) -	3.1717e-2 (2.10e-3) -	3.4429e-2 (1.34e-3) -	1.4292e-2 (3.68e-3)
	10 000	5.3958e-2 (3.76e-3) -	2.8058e-2 (3.01e-3) -	6.9066e-2 (2.14e-3) -	8.2879e-2 (3.85e-3) -	4.2855e-2 (6.50e-3) -	4.7897e-2 (4.78e-3) -	5.1993e-2 (3.06e-3) -	2.1583e-2 (5.12e-3)
SMOP3	1000	2.9217e-2 (3.26e-3) -	5.4153e-3 (3.26e-3) ≈	4.2153e-2 (6.32e-3) -	3.5421e-2 (4.28e-3) -	6.0537e-3 (5.37e-3) -	6.5079e-3 (3.82e-3) -	6.3693e-3 (6.72e-3) -	5.2312e-3 (2.31e-3)
	2000	3.1572e-2 (4.18e-3) -	7.1824e-3 (4.41e-3) -	5.7891e-2 (6.51e-3) -	4.568e-2 (7.04e-3) -	6.6748e-3 (6.83e-3) -	7.4636e-3 (4.37e-3) -	6.8307e-3 (5.89e-3) -	5.7239e-3 (3.19e-3)
	5000	4.2893e-2 (5.32e-3) -	2.4286e-2 (4.89e-3) -	6.9276e-2 (7.16e-3) -	6.2638e-2 (5.47e-3) -	2.2584e-2 (6.37e-3) -	2.6269e-2 (5.82e-3) -	2.5196e-2 (6.54e-3) -	1.7642e-2 (4.91e-3)
	10 000	5.8327e-2 (6.21e-3) -	3.1573e-2 (5.32e-3) -	8.3531e-2 (6.84e-3) -	7.6029e-2 (7.26e-3) -	2.8685e-2 (6.82e-3) -	3.0534e-2 (4.79e-3) -	2.9274e-2 (6.92e-3) -	2.3475e-2 (5.37e-3)
SMOP5	1000	1.0368e-2 (2.74e-4) -	5.3912e-3 (2.19e-4) -	1.3271e-2 (1.56e-4) -	1.5925e-2 (2.80e-4) -	8.2345e-3 (4.74e-4) -	9.2032e-3 (3.48e-4) -	9.9904e-3 (2.23e-4) -	4.1471e-3 (3.12e-5)
	2000	1.0803e-2 (2.39e-4) -	5.6176e-3 (1.92e-4) -	1.3828e-2 (1.36e-4) -	1.6593e-2 (2.45e-4) -	8.5802e-3 (4.14e-4) -	9.5896e-3 (3.04e-4) -	1.0410e-2 (1.94e-4) -	4.3212e-3 (7.93e-4)
	5000	1.2784e-2 (6.54e-5) -	6.6478e-3 (5.23e-5) -	1.6364e-2 (3.71e-5) -	1.9637e-2 (6.69e-5) -	1.0154e-2 (1.13e-4) -	1.1348e-2 (8.31e-5) -	1.2319e-2 (5.31e-5) -	5.1137e-3 (6.31e-4)
	10 000	1.5719e-2 (1.77e-4) -	8.1736e-3 (1.42e-4) -	2.0120e-2 (1.01e-4) -	2.4144e-2 (1.81e-4) -	1.2484e-2 (3.06e-4) -	1.3953e-2 (2.25e-4) -	1.5146e-2 (1.44e-4) -	6.2874e-3 (9.11e-4)
SMOP7	1000	1.9432e-1 (6.62e-3) -	4.2196e-2 (5.32e-3) -	3.2739e-1 (5.28e-3) -	3.9361e-1 (7.51e-3) -	2.0938e-2 (5.61e-3) ≈	2.6547e-2 (6.29e-3) -	2.2762e-2 (6.38e-3) -	1.8282e-2 (4.13e-3)
	2000	2.7261e-1 (7.24e-3) -	6.4359e-2 (5.62e-3) -	4.5738e-1 (6.53e-3) -	5.2672e-1 (8.19e-3) -	2.4935e-2 (6.49e-3) ≈	3.1392e-2 (6.74e-3) -	2.6381e-2 (6.78e-3) -	2.3152e-2 (6.81e-3)
	5000	3.5658e-1 (6.62e-3) -	8.2821e-2 (6.31e-3) -	5.8375e-1 (7.29e-3) -	6.4729e-1 (6.65e-3) -	2.7462e-2 (7.63e-3) ≈	3.6705e-2 (7.38e-3) -	2.9275e-2 (7.29e-3) ≈	2.8179e-2 (6.11e-3)
	10 000	4.2526e-1 (7.63e-3) -	9.8675e-2 (5.87e-3) -	6.5318e-1 (8.23e-3) -	7.2795e-1 (7.81e-3) -	3.2892e-2 (6.26e-3) +	4.2842e-2 (6.82e-3) -	3.7684e-2 (6.85e-3) ≈	3.6988e-2 (6.72e-3)
+/-/≈		0/16/0	0/15/1	0/16/0	0/16/0	1/12/3	0/16/0	0/14/2	

标准差统计结果,在每个测试实例上的最佳结果用粗体表示。

可以看出,除 SLMO-EMT/P 在 5000 维和 10 000 维 SMOP7 上获得了最佳的 IGD 均值,SLMO-EMT 在其它 14 个测试实例上均获得了最佳的 IGD 均值,且 SLMO-EMT 在 5000 维 SMOP7 上的结果与 SLMO-EMT/P 在置信水平为 95% 的 Wilcoxon 秩和检验上近似。说明 SLMO-EMT 所采用辅助任务构建策略、辅助任务重新初始化策略和各知识迁移策略,对于 SLMO-EMT 的性能提升均具有积极作用。

4.4 参数敏感性实验

SLMO-EMT 包括六个独有的参数,分别为用于设置 T_{sparse} 搜索空间的 d_1 和 α 、用于设置 T_{real} 搜索空间的 d_2 、用于 KTP 更新的 ρ ,以及辅助任务重新初始化相关的 θ 和 λ 。

为直观展示在不同测试实例上各参数的敏感程度,在决策变量维数为 5000,最终 IGD 均值在同一量级的 4 个 SMOP 测试实例上,对上述 6 个参数进行敏感性实验。每次仅改变单个参数的取值,且每个参数以表 2 中的取值作为中间值。在每个测试实例上独立运行 20 次,各参数相关的 IGD 均值变化曲线如图 6 所示。

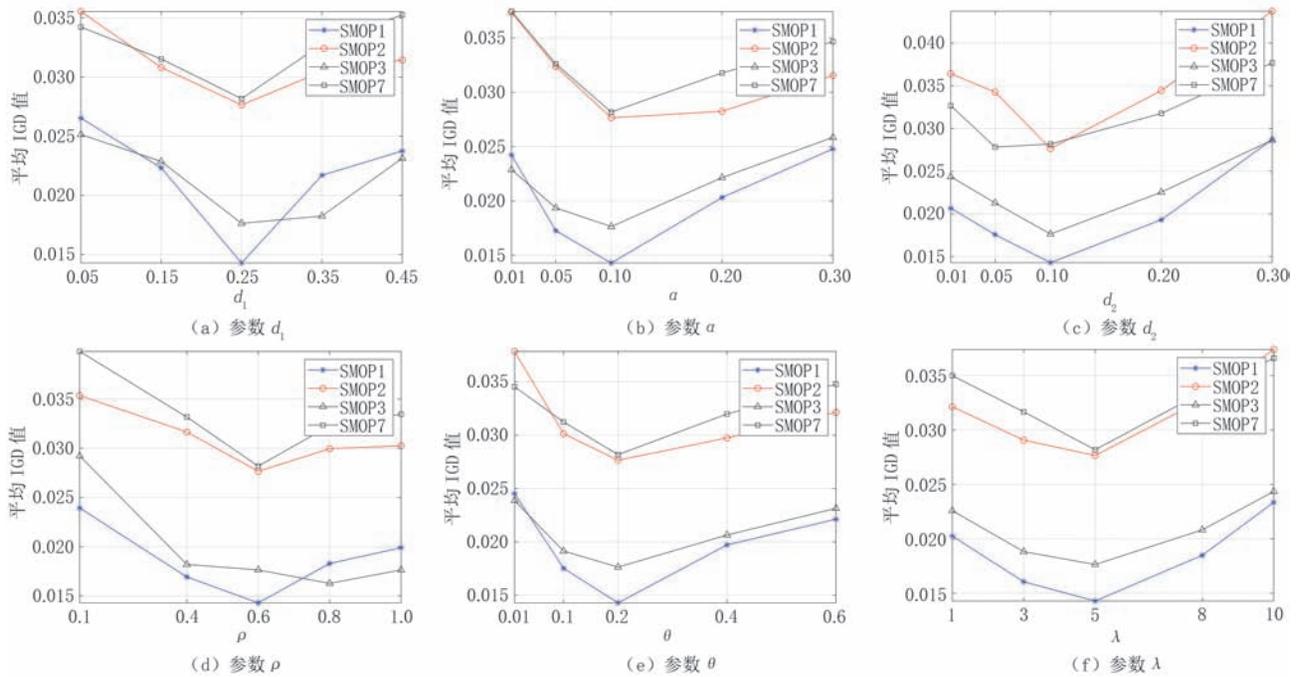


图6 SLMO-EMT 的参数敏感性

总体上看,各参数均对取值敏感,且都在取中间值时,在各测试实例上的平均 IGD 值最小,越往两端,平均 IGD 值越大。此种变化趋势与 SLMO-EMT 中各参数的预期影响一致。

其中,对于 d_1 和 α ,取值过小时,会导致 T_{sparse} 的搜索空间过小,无法对二进制向量中所有潜在的稀疏位置进行充分探测,难以为 T_{main} 提供足够的知识,会导致总体性能的下降。当取值过大时,会导致 T_{sparse} 的搜索空间过大,无法快速探测出二进制向量中的稀疏位置,难以及时为 T_{main} 提供有用的知识,因而总体性能也会下降。

对于 d_2 ,取值过小时,会导致 T_{real} 无法对实数向量中部分对应稀疏度值高但实际非稀疏的位置进行

有效搜索,从而无法为 T_{main} 提供足够的知识,因而总体性能下降;当取值过大时,会导致 T_{real} 的搜索空间包括一些无需搜索的维度,不仅难以快速获得高质量的实数向量,及时为 T_{main} 提供有用的知识,甚至会因 T_{real} 搜索空间的不合理导致潜在的负迁移,因而带来总体性能下降。

对于 ρ ,取值过小时,会导致在更新 KTP 时未充分体现当前知识迁移效果的影响;取值过大时,又容易忽略历史知识迁移效果的影响。因而需要在两者间进行合理平衡,才能使 SLMO-EMT 获得理想的性能。

对于 θ ,取值过小时,会导致辅助任务难以及时进行重新初始化,从而增加负迁移的概率;取值过大

时,则容易导致辅助任务过于频繁地重新初始化,浪费计算资源。与 θ 相关, λ 过小会导致辅助任务过于频繁地重新初始化,过大则会导致辅助任务难以及时进行重新初始化。因此,对于 θ 和 λ ,也需进行合理取值,才能使SLMO-EMT获得理想的性能。

4.5 进一步讨论

(1) 策略有效性实验

SLMO-EMT的核心是采用EMT框架,基于辅助任务引导主任务的进化。为进一步分析EMT框架及相关策略的有效性,将本文所提各策略整体嵌入三个代表性的稀疏大规模多目标优化算法 SparseEA、MOEAPSL、MSKEA,对应的多任务形式变种算法分别称为 SparseEA-EMT、MOEAPSL-EMT 和 MSKEA-EMT。具体而言,在各变种算法中,主任务的自进化采用与各原始算法完全相同的方式,但利用

SLMO-EMT中提出的辅助任务构建与优化策略,分别为每个原始算法构建一个稀疏优化辅助任务和一个实数优化辅助任务,并采用和SLMO-EMT完全一样的辅助任务重新初始化策略和知识迁移策略。

表7为各原始算法与对应变种算法在16个SMOP测试实例上,独立运行20次的平均IGD值和标准差统计结果的两两比较,在每个测试实例上的最佳结果用粗体表示。可以看出,在嵌入了SLMO-EMT所提出策略后,三个变种算法在16个测试实例上均获得了比原始算法更好的性能。进一步说明SLMO-EMT所采用的EMT框架及相关策略对于各类稀疏大规模多目标优化问题具有理想的泛化性。

(2) NNTP上与基于梯度算法的对比

NNTP为源自实际应用的神经网络训练问题,基于梯度的算法是神经网络训练最常用的方法。为

表7 SparseEA、MOEAPSL、MSKEA与对应变种算法在部分SMOP测试实例上独立运行20次获得的IGD均值与标准差

问题	维度	SparseEA	SparseEA-EMT	MOEAPSL	MOEAPSL-EMT	MSKEA	MSKEA-EMT
SMOP1	1000	2.5186e-2	1.0578e-2	2.2612e-2	1.2663e-2	1.3698e-2	8.4928e-3
		(2.71e-3) -	(2.17e-3) -	(1.49e-3) -	(1.95e-3) -	(2.35e-3) -	(1.70e-3) -
	2000	3.2714e-2	1.3740e-2	2.7407e-2	1.5348e-2	1.9725e-2	1.2230e-2
		(2.25e-3) -	(2.31e-3) -	(2.83e-3) -	(2.08e-3) -	(2.38e-3) -	(1.81e-3) -
	5000	3.9199e-2	1.6464e-2	3.9143e-2	2.1920e-2	2.7581e-2	1.7100e-2
		(3.99e-3) -	(2.65e-3) -	(2.75e-3) -	(2.39e-3) -	(2.08e-3) -	(2.07e-3) -
10 000	4.7795e-2	2.0074e-2	5.7853e-2	3.2398e-2	3.3376e-2	2.0693e-2	
SMOP3	1000	(7.77e-3) -	(3.23e-3) -	(6.27e-3) -	(2.91e-3) -	(1.47e-3) -	(2.53e-3) -
		2.8992e-2	1.2177e-2	2.3601e-2	1.3217e-2	9.6286e-3	5.9697e-3
	2000	(2.91e-3) -	(3.28e-3) -	(1.96e-3) -	(2.96e-3) -	(1.80e-3) -	(2.57e-3) -
		3.5154e-2	1.4765e-2	2.5421e-2	1.4236e-2	1.4411e-2	8.9348e-3
	5000	(2.79e-3) -	(5.24e-3) -	(5.45e-3) -	(4.72e-3) -	(2.32e-3) -	(4.11e-3) -
		4.3775e-2	1.8386e-2	3.6860e-2	2.0642e-2	2.2641e-2	1.4037e-2
10 000	(7.59e-3) -	(4.53e-3) -	(2.24e-3) -	(4.08e-3) -	(1.29e-3) -	(3.55e-3) -	
SMOP5	1000	4.9652e-2	2.0854e-2	4.1267e-2	2.3110e-2	2.5902e-2	2.3059e-2
		(6.62e-3) -	(4.88e-3) -	(5.79e-3) -	(4.39e-3) -	(3.21e-3) =	(3.82e-3) -
	2000	5.8628e-3	5.4624e-3	9.0587e-3	8.0729e-3	4.2072e-3	3.6085e-3
		(3.30e-4) -	(7.15e-5) -	(4.57e-4) -	(6.44e-5) -	(7.68e-4) -	(5.60e-5) -
	5000	5.9531e-3	5.5003e-3	9.1766e-3	8.1389e-3	4.2492e-3	3.6345e-3
		(1.88e-4) -	(1.07e-4) -	(3.99e-4) -	(9.63e-5) -	(7.43e-4) -	(8.38e-5) -
10 000	8.6021e-3	7.6129e-3	9.1737e-3	8.1373e-3	4.3241e-3	3.6809e-3	
SMOP7	1000	(7.69e-4) -	(1.64e-4) -	(1.09e-4) -	(1.48e-4) -	(1.50e-4) -	(1.29e-4) -
		9.1049e-3	8.8241e-3	9.1989e-3	8.1514e-3	4.4245e-3	2.7432e-3
	2000	(6.39e-4) \approx	(1.60e-4) -	(2.95e-4) -	(1.44e-4) -	(8.31e-4) -	(1.26e-4) -
		8.7057e-2	3.6564e-2	1.1856e-1	6.6394e-2	5.3941e-2	3.3443e-2
	5000	(7.14e-3) -	(1.36e-2) -	(1.01e-3) -	(1.22e-2) -	(6.44e-3) -	(1.06e-2) -
		1.1277e-1	4.7363e-2	1.4165e-1	7.9324e-2	7.8836e-2	4.8878e-2
10 000	(5.69e-3) -	(1.55e-2) -	(2.91e-2) -	(1.39e-2) -	(5.78e-3) -	(1.21e-2) -	
SMOP5	1000	1.2635e-1	5.3067e-2	1.5323e-1	8.5809e-2	9.2124e-2	7.7117e-2
		(2.65e-3) -	(1.82e-2) -	(4.57e-3) -	(1.64e-2) -	(2.65e-3) -	(1.43e-2) -
	2000	1.7665e-1	7.4193e-2	1.9321e-1	1.0820e-1	2.1134e-1	1.8103e-1
		(2.94e-2) -	(1.94e-2) -	(7.11e-2) -	(1.75e-2) -	(9.81e-3) =	(1.52e-2) -
	5000	1.7665e-1	7.4193e-2	1.9321e-1	1.0820e-1	2.1134e-1	1.8103e-1
		(2.94e-2) -	(1.94e-2) -	(7.11e-2) -	(1.75e-2) -	(9.81e-3) =	(1.52e-2) -
10 000	1.7665e-1	7.4193e-2	1.9321e-1	1.0820e-1	2.1134e-1	1.8103e-1	
+/-/≈	0/15/1		0/16/0		0/14/2		

进一步分析 SLMO-EMT 相比基于梯度算法在性能上的优势,采用经典的随机梯度下降算法 SGD^[52]对 NNTP 的 8 个测试实例进行训练,并基于独立运行 20 次所获得 HV 均值和标准差,与 SLMO-EMT 进行对比,如表 8 所示。

可以看出,相比 SGD, SLMO-EMT 在全部

8 个 NNTP 测试实例上,均具有性能优势。与此同时,结合表 8 和表 5 可以看出,其它 LSMOEAs 也均比 SGD 具有竞争优势。其中, SLMO-EMT 在 NNTP1-NNTP7 上均获得了最佳的 HV 均值,同时在 NNTP8 上,则与最佳 HV 均值在置信水平为 95% 的 Wilcoxon 秩和检验上近似。

表 8 SLMO-EMT 与 SGD 在 8 个 NNTP 测试实例上独立运行 20 次获得的 HV 均值和标准差

算法	NNTP1	NNTP2	NNTP3	NNTP4	NNTP5	NNTP6	NNTP7	NNTP8
SGD	3.4378e-1 (5.38e-2) -	3.5860e-1 (2.65e-2) --	2.5796e-1 (4.16e-2) -	1.5251e-1 (3.78e-2) -	3.8472e-1 (4.73e-2) -	4.1215e-1 (3.54e-2) -	3.3294e-1 (3.43e-2) -	2.8536e-1 (1.84e-2) -
SLMO-EMT	9.4929e-1 (4.21e-2)	8.8271e-1 (1.18e-2)	8.1682e-1 (6.21e-2)	7.6722e-1 (5.64e-2)	9.8723e-1 (4.42e-3)	9.8162e-1 (9.42e-3)	9.9129e-1 (3.12e-2)	7.8857e-1 (7.29e-3)

但值得说明的是,尽管 SLMO-EMT 和现有各类 LSMOEAs 相比基于梯度的算法在万维左右的 NNTP 上获得了更好的性能,但它们尚难以有效处理参数量动辄几百万甚至若干亿的超大规模神经网络训练问题,基于梯度的算法相比而言更适合此类超大规模神经网络训练问题的处理。

5 结 论

针对现有稀疏大规模多目标优化算法在稀疏位置探测准确率和非零决策变量优化程度方面存在的不足,本文提出了一个基于进化多任务优化的稀疏大规模多目标优化算法 SLMO-EMT。通过降维方式构建两个分别用于稀疏位置探测和非零决策变量优化的辅助任务,并针对性设计知识迁移策略,同时适时对辅助任务进行重新初始化,可有效引导主任务持续高质量进化。实验结果表明,相比其它稀疏大规模多目标优化算法,SLMO-EMT 在综合优化性能上具有明显的竞争优势。

未来研究中,我们将进一步研究如何自适应地设置 SLMO-EMT 中各参数的取值,同时将 SLMO-EMT 应用于更多、更复杂、更大规模的实际稀疏大规模多目标优化问题的求解。此外,如何为稀疏大规模多目标优化设计针对性的环境选择策略,从而更好地提升优化过程中的搜索性能,具有重要价值。

参 考 文 献

- [1] Atashgahi Z, Pieterse J, Liu S, Mocanu D, Veldhuis R, Pechenizkiy M. A brain-inspired algorithm for training highly sparse neural networks. *Machine Learning*, 2022, 111(12): 4411-4452
- [2] Han W, Li H, Gong M, Li J, Liu Y, Wang Z. Multi-swarm particle swarm optimization based on CUDA for sparse reconstruction. *Swarm and Evolutionary Computation*, 2022, 75: 101153
- [3] Bonk R. The future of passive optical networks//*Proceedings of the 2021 International Conference on Optical Network Design and Modeling*. Gothenburg, Sweden, 2021: 1-3
- [4] Tian Y, Zhang X, Wang C, Jin Y. An evolutionary algorithm for large-scale sparse multiobjective optimization problems. *IEEE Transactions on Evolutionary Computation*, 2020, 24(2): 380-393
- [5] Li H, Zhang Q, Deng J, Xu Z. A preference-based multiobjective evolutionary approach for sparse optimization. *IEEE Transactions on Neural Networks and Learning Systems*, 2017, 29(5): 1716-1731
- [6] Tian Y, Lu C, Zhang X, Tan K C, Jin Y. Solving large-scale multiobjective optimization problems with sparse optimal solutions via unsupervised neural networks. *IEEE Transactions on Cybernetics*, 2021, 51(6): 3115-3128
- [7] Bäck T H W, Kononova AV, Stein B, Wang H, Antonov K A, Kalkreuth R T, Nobel J, Vermetten D, Winter R, Ye F. Evolutionary algorithms for parameter optimization-thirty years later. *Evolutionary Computation*, 2023, 31(2): 81-122
- [8] Aishwaryaprajna, Rowe J. Evolutionary and estimation of distribution algorithms for unconstrained, constrained, and multiobjective noisy combinatorial optimisation problems. *Evolutionary Computation*, 2023, 31(3): 259-285
- [9] Zhong X, Yao X, Gong D, Qiao K, Gan X, Li Z. A dual-population-based evolutionary algorithm for multi-objective optimization problems with irregular Pareto fronts. *Swarm and Evolutionary Computation*, 2024, 87: 101566
- [10] Han H, Zhou H, Huang Y, Hou Y. Robust multiobjective particle swarm optimization with feedback compensation strategy. *IEEE Transactions on Cybernetics*, 2023, 53(2): 1062-1074
- [11] XIE Cheng-Wang, PAN Jia-Min, Guo Hua, Wang Dong-Mei, FU Shi-Wei. A large scale multi-objective evolutionary algorithm adopting hybrid strategies. *Chinese Journal of Computers*, 2024, 47(1):69-89 (in Chinese)

(谢承旺,潘嘉敏,郭华,王冬梅,付世伟.一种采用混合策略的大

- 规模多目标进化算法. 计算机学报, 2024, 47(1): 69-89)
- [12] Shi J, Sun J, Zhang Q, Zhang H, Fan Y. Improving pareto local search using cooperative parallelism strategies for multiobjective combinatorial optimization. *IEEE Transactions on Cybernetics*, 2024, 50(4): 2369-2382
- [13] He Z, Liu H. Offspring regeneration driven by finite element mapping for large-scale evolutionary multiobjective optimization. *Swarm and Evolutionary Computation*, 2023, 83: 101393
- [14] Hong H, Jiang M, Yen G. Boosting scalability for large-scale multiobjective optimization via transfer weights. *Information Sciences*, 2024, 670: 120607
- [15] Tian Y, Lu C, Zhang X, Cheng F, Jin Y. A pattern mining-based evolutionary algorithm for large-scale sparse multiobjective optimization problems. *IEEE Transactions on Cybernetics*, 2022, 52(7): 6784-6797
- [16] Ding Z, Chen L, Sun D, Zhang X. A multi-stage knowledge-guided evolutionary algorithm for large-scale sparse multi-objective optimization problems. *Swarm and Evolutionary Computation*, 2022, 73: 101119
- [17] Wang X, Zhang K, Wang J, Jin Y. An enhanced competitive swarm optimizer with strongly convex sparse operator for large-scale multiobjective optimization. *IEEE Transactions on Evolutionary Computation*, 2022, 26(5): 859-871
- [18] Wang X, Zhang B, Wang J, Zhang K, Jin Y. A cluster-based competitive particle swarm optimizer with a sparse truncation operator for multi-objective optimization. *Swarm and Evolutionary Computation*, 2022, 71: 101083
- [19] Wu C, Tian Y, Zhang Y, Jiang H, Zhang X. A sparsity-guided elitism co-evolutionary framework for sparse large-scale multi-objective optimization//*Proceedings of the 2023 IEEE Congress on Evolutionary Computation*. Chicago, USA, 2023: 1-8
- [20] Kropp I, Nejadhashemi A, Deb K. Improved evolutionary operators for sparse large-scale multiobjective optimization problems. *IEEE Transactions on Evolutionary Computation*, 2024, 28(2): 460-473
- [21] Zhang Y, Tian Y, Zhang X. Improved SparseEA for sparse large-scale multi-objective optimization problems. *Complex & Intelligent Systems*, 2023, 9(2): 1127-1142
- [22] Jiang J, Han F, Wang J, Ling Q, Han H, Wang Y. A two-stage evolutionary algorithm for large-scale sparse multiobjective optimization problems. *Swarm and Evolutionary Computation*, 2022, 72: 101093
- [23] Zou Y, Liu Y, Zou J, Yang S, Zheng J. An evolutionary algorithm based on dynamic sparse grouping for sparse large scale multiobjective optimization. *Information Sciences*, 2023, 631: 449-467
- [24] Tian Y, Feng Y, Zhang X, Sun C. A fast clustering based evolutionary algorithm for super-large-scale sparse multi-objective optimization. *IEEE/CAA Journal of Automatica Sinica*, 2022, 10(4): 1048-1063
- [25] Gupta A, Ong Y S, Feng L, Tan K C. Multiobjective multifactorial optimization in evolutionary multitasking. *IEEE Transactions on Cybernetics*, 2017, 47(7): 1652-1665
- [26] Liang Z, Zhu Y, Wang X, Li Z, Zhu Z. Evolutionary multitasking for multi-objective optimization based on generative strategies. *IEEE Transactions on Evolutionary Computation*, 2023, 27(4): 1042-1056
- [27] Qiao K, Liang J, Yu K, Wang M, Qu B, Yue C, Guo Y. A self-adaptive evolutionary multi-task based constrained multi-objective evolutionary algorithm. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 2023, 7(4): 1098-1112
- [28] Jiang Y, Zhan Z H, Tan K C, Zhang J. Block-level knowledge transfer for evolutionary multitask optimization. *IEEE Transactions on Cybernetics*, 2024, 54(1): 558-571
- [29] Huang Y, Feng L, Li M, Wang Y, Zhu Z, Tan K C. Fast vehicle routing via knowledge transfer in a reproducing Kernel Hilbert space. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2023, 53(9): 5404-5416
- [30] Gupta A, Zhou L, Ong Y S, Chen Z, Hou Y. Half a dozen real-world applications of evolutionary multitasking, and more. *IEEE Computational Intelligence Magazine*, 2022, 17(2): 49-66
- [31] Chen K, Xue B, Zhang M, Zhou F. An evolutionary multitasking-based feature selection method for high-dimensional classification. *IEEE Transactions on Cybernetics*, 2020, 52(7): 7172-7186
- [32] Li H, Li D, Gong M, Li J, Qin A K, Xing L, Xie F. Sparse hyperspectral unmixing with preference-based evolutionary multiobjective multitasking optimization. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 2024, 8(2): 1922-1937
- [33] Gong Mao-Guo, Jiao Li-Cheng, Yang Dong-Dong, Ma Wen-Ping. Research on evolutionary multi-objective optimization algorithms. *Journal of Software*, 2009, 20(2): 271-289 (in Chinese) (公茂果, 焦李成, 杨咚咚, 马文萍. 进化多目标优化算法研究. 软件学报, 2009, 20(2): 271-289)
- [34] Li M M, Yang S, Liu X. Pareto or Non-Pareto: Bi-criterion evolution in multiobjective optimization. *IEEE Transactions on Evolutionary Computation*, 2016, 20(5): 645-665
- [35] Wei T, Wang S, Zhong J, Liu D, Zhang J. A review on evolutionary multitask optimization: Trends and challenges. *IEEE Transactions on Evolutionary Computation*, 2021, 26(5): 941-960
- [36] Li Hao, Wang Lei, Zhang Yuan-Qiao, Wu Yue, Gong Maoguo. Survey of evolutionary multitasking optimization. *Journal of Software*, 2023, 34(2): 509-538 (in Chinese) (李豪, 汪磊, 张元侨, 武越, 公茂果. 演化多任务优化研究综述. 软件学报, 2023, 34(2): 509-538)
- [37] Zhang Y, Wu C, Tian Y, Zhang X. A co-evolutionary algorithm based on sparsity clustering for sparse large-scale multi-objective optimization. *Engineering Applications of Artificial Intelligence*, 2024, 133: 108194
- [38] Shao S, Tian Y, Zhang X. Deep reinforcement learning assisted automated guiding vector selection for large-scale sparse multi-objective optimization. *Swarm and Evolutionary Computation*, 2024, 88: 101606
- [39] Qi S, Wang R, Zhang T, Yang X, Sun R, Wang L. A two-layer encoding learning swarm optimizer based on frequent itemsets for sparse large-scale multi-objective optimization. *IEEE/CAA Journal of Automatica Sinica*, 2024, 11(6): 1342-1357

- [40] Jiang J, Wang H, Hong J, Liu Z, Han F. Improving two-layer encoding of evolutionary algorithms for sparse large-scale multiobjective optimization problems. *Complex & Intelligent Systems*, 2024: 1-19
- [41] Tian Y, Shao S, Xie G, Zhang X. A multi-granularity clustering based evolutionary algorithm for large-scale sparse multi-objective optimization. *Swarm and Evolutionary Computation*, 2024, 84: 101453
- [42] Vose M, Wright A. The simple genetic algorithm and the walshtransform: Part I, theory. *Evolutionary Computation*, 1998, 6 (3): 253-273
- [43] Deb K, Agrawal R B. Simulated binary crossover for continuous search space. *Complex Systems*, 1995, 9(2): 115-148
- [44] Deb K, Goyal M. A combined genetic adaptive search (GeneAS) for engineering design. *Computer Science and Informatics*, 1996, 26: 30-45
- [45] Deb K, Pratap A, Agarwal S, et al. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 2002, 6(2): 182-197
- [46] Dheeru and ED.. KarraTaniskidou. UCI Machine Learning Repository, <https://archive.ics.uci.edu/datasets>, 2024.3.10
- [47] Wu T, Shi J, Zhou D, Lei Y, Gong M. A multi-objective particle swarm optimization for neural networks pruning// *Proceedings of the 2019 IEEE Congress on Evolutionary Computation*. Wellington, New Zealand, 2019: 570-577
- [48] Tian Y, Cheng R, Zhang X, Jin Y. PlatEMO: A MATLAB platform for evolutionary multi-objective optimization. *IEEE Computational Intelligence Magazine*, 2017, 12(4): 73-87
- [49] Bosman P A N, Thierens D. The balance between proximity and diversity in multiobjective evolutionary algorithms. *IEEE Transactions on Evolutionary Computation*, 2003, 7(2): 174-188
- [50] While L, Hingston P, Barone L, Huband S. A faster algorithm for calculating hypervolume. *IEEE Transactions on Evolutionary Computation*, 2006, 10(1): 29-38
- [51] Haynes W. Wilcoxon rank sum test. *Encyclopedia of Systems Biology*, 2013, 3(1): 2354-2355
- [52] LeCun Y, Bengio Y, Hinton G. Deep learning. *Nature*, 2015, 521(7553): 436-444



LIANG Zheng-Ping, Ph. D., associate professor, His research interests include computational intelligence and big data analysis.

WANG Kan, M. S. His research interest is computational intelligence.

ZHOU Qian, M. S. candidate. Her research interest is computational intelligence.

WANG Ji-Gang, Ph. D. His research interests include computational intelligence and network security.

ZHU Ze-Xuan, Ph. D., professor. His research interests include computational intelligence and bioinformatics.

Background

There are various types of sparse large-scale multi-objective optimization problems in the real world. For example, the neural network training problem aims to prevent overfitting by training neural networks with fewer weights. The signal reconstruction problem aims to minimize model complexity errors by sparsely reconstructing signals.

Sparse large-scale multi-objective optimization faces many challenges, such as difficulties in detecting sparse positions and explosion of search space. Existing sparse large-scale multi-objective optimization algorithms have significant room for improvement in terms of the detection accuracy of sparse positions and the optimization degree of non-zero decision variables. To enhance the performance of sparse large-scale multi-objective optimization, this paper proposes an algorithm called Evolutionary Multi-Task for Sparse Large-

scale Multi-objective Optimization (SLMO-EMT) inspired by Evolutionary Multi-Task. SLMO-EMT include threenovel strategies: auxiliary task construction and optimization, auxiliary task reinitialization, and knowledge transfer. Experimental results demonstrate that SLMO-EMT has a significant competitive advantage in solving sparse large-scale multi-objective optimization problems.

Our research team has been working in the field of intelligent optimization for more than ten years, and has published a series of algorithms in the top-ranked journals, such as IEEE TEVC and IEEE TCYB, on multi-objective optimization, large-scale optimization, multi-task optimization, etc. This paper is partly supported by the National Key Research and Development Program of China (2021YFB2900800), and the Shenzhen Science and Technology Program, China (JCYJ20220531102617039).