文章编号:1671-6833(2026)02-0051-08

基于竞争与合作多任务的约束多目标进化算法

张 萌[1], 梁 静[2], 乔康加[2], 岳彩通[2], 王曦璐[3]

(1. 河南牧业经济学院 能源与智能工程学院,河南 郑州 450046;2. 郑州大学 电气与信息工程学院,河南 郑州 450001;3. 萨里大学 计算机科学与电子工程学院,英国 萨里 GU2 7XH)

摘 要:基于多任务的约束多目标进化算法在资源分配和协同优化方面存在不足,导致低有效性种群浪费计算资源以及优质解信息未被充分利用。因此,设计了一种基于竞争与合作多任务的约束多目标进化算法,包含两个主要的策略:第一,提出一种基于竞争的资源分配策略,基于各任务种群的历史表现实现计算资源的自适应分配;第二,设计了基于父代聚合和子代扩散的协同优化策略,通过跨任务合作生成高质量的子代,并使子代扩散到各任务种群中,实现有效信息的高效利用。提出的算法与其他 5 种先进算法(CMOEA_MS、cDPEA、EMCMO、MTCMO、CMOEMT)在 38 个测试函数上进行对比实验。结果表明:所提算法在 *IGD* 和 *HV* 指标上分别在 25 个和 26 个函数上取得了最优结果,且分别至少在 23 个和 24 个函数上优于对比算法;所提算法在所有函数上的可行率达到 100%,能够有效求解约束多目标优化问题。

关键词:约束多目标优化; 进化算法; 多任务; 资源分配; 协同优化

中图分类号:TP301 **文献标志码**:A **doi**:10.13705/j.issn.1671-6833.2025.05.021

约束多目标优化问题[1](constrained multi-objective optimization problem, CMOP)广泛存在于科学研究和工程应用中,不仅涉及多个冲突目标函数的优化,还需满足多种约束条件,是一类复杂的优化问题,对现有求解方法提出了严峻挑战。

为了求解 CMOP,学术界通过结合多目标进化算法和约束处理技术设计了多种约束多目标进化算法(constrained multi-objective evolutionary algorithm, CMOEA)[2-3],其中主流的算法包括基于多阶段的 CMOEA[4-5]、基于多种群的 CMOEA[6-7]和基于多任务的 CMOEA[8-9]。基于多任务的 CMOEA 通过设计简化的辅助任务,降低了约束或目标的求解难度[8,10]。进一步,算法从辅助任务向主任务(即原 CMOP)迁移有效的知识,帮助求解主任务。例如,Qiao 等[8]提出了第一个基于多任务的 CMOEA,创建了任意数量的辅助种群,并通过知识迁移提升每个种群的性能。进一步,Qiao 等[9]提出了基于动态辅助任务的方法,该辅助种群动态缩放约束边界,提升了和主种群的相似性,从而提升了知识迁移的有效性。近期,多位学者提出了更多类型的辅助任务和知识定义方法[11-12],丰富了该类方法的理论,提升了该类方法的有效性。

然而,现有基于多任务的方法忽视了调节种群间的竞争关系,采用均质化计算资源分配策略,导致进化后期低效种群仍产生冗余个体,显著降低了收敛效率。此外,种群间信息交互局限于子代分享,子代生成过程缺乏跨种群协作,导致高质量个体的信息未能被充分利用。

为了克服以上缺陷,本文提出了一种基于竞争与合作多任务的约束多目标进化算法(competition and cooperation multitasking based constrained multi-objective evolutionary algorithm, CCMT),其包含 3 种任务及其对应的进化种群。主任务即原 CMOP,其种群优先保存可行帕累托非支配解;辅助任务 1 为

收稿日期:2025-10-21;**修订日期**:2025-12-03

基金项目:国家自然科学基金资助项目(61922072, U23A20340);河南省自然科学基金优秀青年科学基金项目(242300421168)

作者简介:张萌(1990—),女,河南郑州人,河南牧业经济学院讲师,主要从事约束多目标进化优化理论及应用研究,E-mail: 13569393509@163.com。

通信作者:乔康加(1996—),男,河南焦作人,郑州大学讲师,博士,主要从事约束多目标进化优化理论及应用研究,E-mail: qiaokangjia@yeah.net。

引用本文:张萌,梁静,乔康加,等. 基于竞争与合作多任务的约束多目标进化算法[J]. 郑州大学学报(工学版),2026,47(2):51-58,76. (ZHANG M, LIANG J, QIAO K J, et al. A constrained multi-objective evolutionary algorithm based on competition and cooperation multi-tasking [J]. Journal of Zhengzhou University (Engineering Science), 2026,47(2):51-58,76.)

删除约束的无约束多目标优化问题,其种群忽视约束以加强目标信息的利用,从而加速算法的收敛速度;辅助任务 2 为约束边界动态缩放的 CMOP,其种群动态利用有效的不可行解信息,以维持算法的多样性。此外,本文设计了一种基于竞争的资源分配策略,通过测评各任务种群的历史表现,动态调整其在随后进化过程中所使用的计算资源,以适合当前地形的任务发挥更大的价值。进一步,本文还设计了基于父代聚合和子代扩散的协同优化策略,3 个种群在子代产生和环境选择过程中通过合作来提高信息利用效率。在 38 个标准测试函数上的实验结果表明,所提算法明显优于 5 种现有的较先进的算法。

1 约束多目标优化问题

不失一般性,一个 CMOP 可以用以下公式表示[13]:

$$\begin{cases}\min \boldsymbol{F}(\boldsymbol{x}) = (f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), \cdots, f_M(\boldsymbol{x})), \\ \quad \boldsymbol{x} = (x_1, x_2, \cdots, x_D)^{\mathrm{T}} \in \boldsymbol{S}^D; \\ \text{s.t. } g_i(\boldsymbol{x}) \leqslant 0, i = 1, 2, \cdots, p; \\ \quad h_i(\boldsymbol{x}) = 0, i = p+1, p+2, \cdots, n。\end{cases} \tag{1}$$

式中:$f(\boldsymbol{x})$ 代表单个优化目标;$\boldsymbol{F}(\boldsymbol{x})$ 代表一个 M 维的优化目标向量;$\boldsymbol{x}$ 代表一个 D 维决策向量;$\boldsymbol{S}^{\mathrm{D}}$ 代表决策空间;$g(\boldsymbol{x})$ 和 $h(\boldsymbol{x})$ 分别代表不等式约束和等式约束;p 和 $(n-p)$ 分别代表不等式约束和等式约束的数量。

为了计算等式约束的违反度,首先,一般通过松弛参数 δ 将等式约束转换为不等式约束[14]:

$$|h_i(\boldsymbol{x})| - \delta \leqslant 0, i = p+1, p+2, \cdots, n。 \tag{2}$$

其次,每个约束的违反度可以由下式计算:

$$G_i(\boldsymbol{x}) = \begin{cases}\max(0, g_i(\boldsymbol{x})), 1 \leqslant i \leqslant p; \\ \max(0, |h_i(\boldsymbol{x})| - \delta), p+1 \leqslant i \leqslant n。\end{cases} \tag{3}$$

最后,$\boldsymbol{x}$ 的总约束违反度[14]可由下式计算:

$$G(\boldsymbol{x}) = \sum_{i=1}^{n} G_i(\boldsymbol{x})。 \tag{4}$$

当 $G(\boldsymbol{x})$ 等于 0 时,$\boldsymbol{x}$ 满足所有约束,被称为可行解,否则被称为不可行解。给定两个可行解 $\boldsymbol{x}_1$ 和 $\boldsymbol{x}_2$,对任意 $j \in (1, 2, \cdots, M)$,$f_j(\boldsymbol{x}_1) \leqslant f_j(\boldsymbol{x}_2)$,并且存在 $k \in (1, 2, \cdots, M)$,使得 $f_k(\boldsymbol{x}_1) < f_k(\boldsymbol{x}_2)$,则称解 $\boldsymbol{x}_1$ 帕累托支配解 $\boldsymbol{x}_2$。若 $\boldsymbol{x}^*$ 不被任何其他解支配,则称其为帕累托最优解[15]。求解 CMOP 的目的是获得一组可行的帕累托最优解集(constrained Pareto set, CPS),其在目标空间中的映射向量被称为约束帕累托前沿(constrained Pareto front, CPF)。若不考虑约束,CPF 将变为无约束帕累托前沿(unconstrained PF, UPF)[16]。

2 算法描述

2.1 算法框架

图 1 为算法 CCMT 的框架。如图 1 所示,算法 CCMT 框架中包含一个主任务(表示为 T_1)和两个辅助任务(分别表示为 T_2 和 T_3),并创建 3 个种群分别求解 3 个任务(分别表示为 P_1、P_2 和 P_3)。第一个任务 T_1 为用式(1)表示的原 CMOP,该任务被称为主任务,需要算法找到可行帕累托最优解。因此,种群 P_1 使用约束支配准则(constrained dominance principle, CDP)[15]更新种群,优先保存约束值小的解,使得 P_1 可以快速定位到可行域并保存可行解。

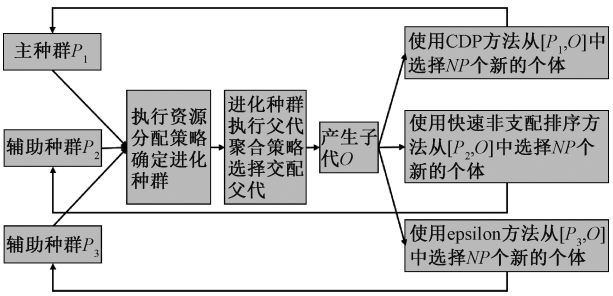

图 1 算法 CCMT 的框架

Figure 1 Framework of the algorithm CCMT

第二个任务 T_2 为无约束辅助任务[6],其公式为

$$\min \boldsymbol{F}(\boldsymbol{x}) = (f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), \cdots, f_M(\boldsymbol{x}))。 \tag{5}$$

该辅助任务删除了所有约束,最优解为 UPF。通过舍弃约束,该辅助任务能够帮助主任务种群快速探索整个目标空间,从而快速找到有潜力的搜索区域。因为 T_2 为无约束多目标任务,种群 P_2 使用无约束非支配排序方法[15]更新种群。

第 3 个任务 T_3 为约束缩放任务[11],公式为

$$\begin{cases}\min \boldsymbol{F}(\boldsymbol{x}) = (f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), \cdots, f_M(\boldsymbol{x})); \\ \text{s.t. } G(\boldsymbol{x}) \leqslant \varepsilon。\end{cases} \tag{6}$$

式中:ε 代表约束缩放边界,是一个动态变化的值,在本文中,ε 等于后代种群中所有不可行解的平均约束值。为了更有效地探索约束边界内的区域,种群 P_3 使用 epsilon 方法[17]更新种群,其中约束值小于 ε 的解使用基于多目标的约束处理方法[18]进行排序;约束值大于 ε 的解使用 CDP 方法进行排序。

这 3 个任务具有不同的最优解,使得 3 个种群有不同的进化轨迹,因此,执行知识迁移可以提高种群多样性。此外,两个辅助任务分别利用全局

解和缩放区域中的解,对主任务具有互补的辅助效果。

CCMT 算法提供了详细的伪代码,如算法 1 所示。首先,为每个任务初始化一个大小为 NP 的特定种群。此外,初始化奖励矩阵 $\boldsymbol{R}$ 和每个任务的选择概率 s_1、s_2 和 s_3。随后,算法进入主循环。第 5 行,执行轮盘赌操作确定选择第 $k \in \{1,2,3\}$ 个任务作为进化任务。第 6 行,第 k 个任务执行父代聚合操作产生一个杂交父代种群 HP。第 7 行,HP 使用差分进化操作[9](DE/current-to-rand/1 和 DE/current-to-$p$best/1)产生后代种群 O。第 8 行,执行后代扩散策略完成环境选择过程。最后,更新 3 个任务的选择概率。

算法 1: CCMT 算法。

输入: 种群规模 NP、父代聚合概率 ρ;

输出: P_1。

① 初始化种群 P_1、P_2、P_3,每个种群大小为 NP;

② 初始化奖励矩阵 $\boldsymbol{R}$ 和选择概率 (s_1,s_2,s_3);

③ $t=0$;

④ While 没有达到最大迭代次数

⑤　$k \leftarrow$ 轮盘赌 (s_1,s_2,s_3);

⑥　$HP \leftarrow$ 父代聚合$(P_1,P_2,P_3,k,\rho) \rightarrow$ 算法 2;

⑦　$O \leftarrow HP$ 使用差分进化操作产生 NP 个后代个体;

⑧　$P_1,P_2,P_3 \leftarrow$ 后代扩散$(P_1,P_2,P_3,O) \rightarrow$ 算法 3;

⑨　$t=t+1$;

⑩　For $k=1:3$

⑪　$R_{k,t}^b$ 和 $R_{k,t}^p \leftarrow$ 根据式(8)、式(9)计算第 k 个任务的全局奖励和种群奖励;

⑫　$R_{k,t} \leftarrow$ 根据式(7)计算第 k 个任务的奖励;

⑬　End

⑭ 根据式(10)更新 (s_1,s_2,s_3);

⑮ End While

2.2　基于竞争的资源分配策略

现有基于多任务的算法中,每个种群被分配相同的计算资源,忽略了种群在不同进化阶段适应性的动态变化,因此,有必要调整种群间的资源分配[11]。图 2 提供了一个小可行域问题[19]上种群在进化后期的分布。P_2 通过忽视约束定位在无约束帕累托前沿上,P_3 通过约束缩放定位在约束搜索边界内部,而 P_1 通过优化原问题定位在局部约束帕累托前沿。如图 2 所示,无约束帕累托前沿和约束帕累托前沿相距较远,导致 P_2 的更新无法继续为寻找约束帕累托前沿提供有效的信息。因此,有必要实时调整种群间的计算资源。基于此,本节提出了一种基于竞争的资源分配策略。

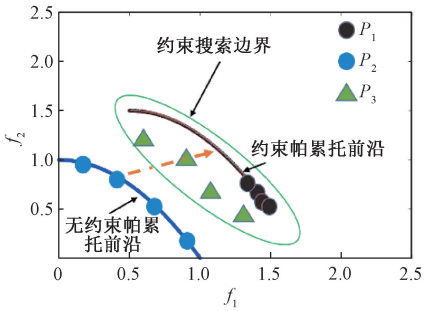

图 2　小可行域问题上 3 个种群在进化后期的分布

Figure 2　Distribution of three populations at the late evolutionary phase on a problem with small feasible regions

如算法 1 所示,CCMT 基于任务的选择概率使用轮盘赌方法选择有效的进化种群,其中选择概率由任务种群在过去 LR 代的累计奖励计算得来。

第 t 代中第 k 个任务的奖励 $R_{k,t}$ 为

$$R_{k,t} = \alpha \cdot R_{k,t}^{\mathrm{b}} + (1-\alpha) \cdot R_{k,t}^{\mathrm{p}}。\tag{7}$$

式中: $R_{k,t}^{\mathrm{b}}$ 为全局奖励,反映任务对全局最优的贡献程度; $R_{k,t}^{\mathrm{p}}$ 为种群奖励,反映第 k 个种群的多样性和收敛性的改进速度;参数 α 控制两种奖励的权重。如果一个任务有助于更频繁地更新全局最优值,或者促进其种群内个体的更快进化,它将会获得更高的奖励。$R_{k,t}^{\mathrm{b}}$ 和 $R_{k,t}^{\mathrm{p}}$ 的具体计算过程为

$$R_{k,t}^{\mathrm{b}} = \mathrm{sign}(z_t^* < z_{t-1}^*);\tag{8}$$

$$R_{k,t}^{\mathrm{p}} = \frac{num\_O}{N}。\tag{9}$$

式中: z_t^* 表示直到第 t 代所有可行解在每个目标上的最小值; $z_t^* < z_{t-1}^*$ 表示前者支配后者。如果 sign 函数中的布尔值为真,则返回值为 1,否则返回值为 0;num_O 代表在环境选择中成功进入下一代的后代个体数量,num_O 越大,代表对应种群进化得越快。

基于上述定义,每个任务的选择概率可用式(10)更新:

$$s_k = \begin{cases} 1/3, t \leqslant \beta \cdot T; \\ \sum_{i=1}^{LR} R_{k,i} \Big/ \left(\sum_{j=1}^{3} \sum_{i=1}^{LR} R_{j,i} \right), \text{其他}。 \end{cases}\tag{10}$$

式中: T 为最大进化代数;参数 β 控制选择概率的更新方式。在进化前期,每个任务有相同的选择概率,使得每个任务都能被选中以评估其表现;在进化后期,种群在历史过程中的表现奖励被用于更新选择概率,使得适合当前环境的任务能被更频繁地选中,从而发挥此任务种群的效果,避免低效种群浪费计算资源。

2.3 基于合作的种群协同优化

种群协同优化包含两个步骤:基于父代聚合的交配父代选择和基于后代扩散的环境选择。

2.3.1 基于父代聚合的交配父代选择

现有方法仅在环境选择过程中分享子代个体完成信息分享,然而,这种方式未能充分利用不同种群的信息产生高质量的杂交子代[20]。如图 2 所示,小可行域导致 P_1 收敛至局部约束帕累托前沿上。虚线箭头是由 P_2 的个体和 P_3 的个体产生的搜索方向,由该搜索方向产生的子代个体可以定位于小可行域上。因此,基于种群合作的子代产生方式能提供更多的搜索多样性,有助于种群跳出局部最优。基于此,本节设计了基于父代聚合的交配父代选择策略。

算法 2 提供了交配父代选择的伪代码,其目的是从 3 个种群中选择包含 NP 个父代个体的交配池 HP。首先,当前进化种群 P_k 由算法 1 的第 5 行决定。对于种群 P_k 的第 i 个个体 $\boldsymbol{x}_{i,k}$,若随机数 $rand$ 小于 ρ,则它与其他种群的个体进行比较,从而确定能够进入交配池的个体。具体步骤为从另外两个种群中随机选择一个种群 P_h,并从中随机选择一个个体 $\boldsymbol{x}_{r,h}$。然后使用 CDP 方法比较 $\boldsymbol{x}_{i,k}$ 和 $\boldsymbol{x}_{r,h}$,其中较好的个体被添加进集合 HP 中。若随机数 $rand$ 大于等于 ρ,直接将 $\boldsymbol{x}_{i,k}$ 添加进 HP 中。

此方法中,基于概率的方法能从另外两个种群中选择有效的个体参与后代产生,这有助于信息交流。此外,基于 CDP 的比较方法使得另外两个种群中约束违反度较低的解参与后代产生,能够帮助种群更快收敛至可行域。因此,基于父代聚合的交配父代选择方法能够维持种群间的合作并且加快收敛。

算法 2:父代聚合算法。

输入:种群规模 NP、父代聚合概率 ρ、种群 P_1、种群 P_2、种群 P_3、被选中种群的索引 k、聚合概率 β;

输出:交配父代种群 HP。

① 设置 HP 等于空集;
② for $i=1$: NP
③ 选择 P_k 中第 i 个个体,记作 $\boldsymbol{x}_{i,k}$;
④ if $rand<\rho$
⑤ 随机选择一个任务 T_h;
⑥ 从 P_h 中随机选择一个个体 $\boldsymbol{x}_{r,h}$;
⑦ if $\boldsymbol{x}_{r,h} \prec_{\mathrm{CDP}} \boldsymbol{x}_{i,k}$
⑧ $\boldsymbol{x}^* \leftarrow \boldsymbol{x}_{r,h}$;
⑨ elseif $\boldsymbol{x}_{i,k} \prec_{\mathrm{CDP}} \boldsymbol{x}_{r,h}$
⑩ $\boldsymbol{x}^* \leftarrow \boldsymbol{x}_{i,k}$;
⑪ else
⑫ $\boldsymbol{x}^* \leftarrow$ 从 $\boldsymbol{x}_{r,h}$ 和 $\boldsymbol{x}_{i,k}$ 中随机选择一个个体;
⑬ End if
⑭ else
⑮ $\boldsymbol{x}^* \leftarrow \boldsymbol{x}_{i,k}$;
⑯ End if
⑰ $HP \leftarrow x^*$;
⑱ End for

2.3.2 基于后代扩散的环境选择

算法 3 提供了该方法的伪代码,主要目的是使用杂交后代种群 O 更新 3 个任务种群。3 个任务种群分别通过 CDP 方法、无约束非支配排序方法和 epsilon 方法完成环境选择过程。

算法 3:后代扩散算法。

输入:种群规模 NP、种群 P_1、种群 P_2、种群 P_3、后代种群 O;

输出:P_1、P_2、P_3。

① $TP_1 \leftarrow P_1 \cup O$;
② $P_1 \leftarrow$ 使用 CDP 方法从 TP_1 中选择 NP 个个体;
③ $TP_2 \leftarrow P_2 \cup O$;
④ $P_2 \leftarrow$ 使用无约束非支配排序方法从 TP_2 中选择 NP 个个体;
⑤ $TP_3 \leftarrow P_3 \cup O$;
⑥ $P_3 \leftarrow$ 使用 epsilon 方法从 TP_3 中选择 NP 个个体。

2.4 算法复杂度分析

CCMT 的时间复杂度主要来源于父代聚合操作、子代产生操作和后代扩散操作。前两种操作的时间复杂度分别为 $O(NP)$ 和 $O(NP \cdot D)$。后代扩散操作中,CDP、无约束非支配选择和改进 epsilon 方法的时间复杂度均为 $O(M \cdot NP^2)$。因此,CCMT 的时间复杂度为 $O(NP)+O(NP \cdot D)+O(M \cdot NP^2)=O(M \cdot NP^2)$,与大多数约束多目标进化算法的时间复杂度相同[6-7]。

3 实验设置和结果

3.1 实验设置

对比方法。本文选择了几个最新的 CMOEA 作为对比方法,包括 CMOEA_MS[5]、cDPEA[7]、EMCMO[8]、MTCMO[9] 和 CMOEMT[11]。CMOEA_MS 为基于多阶段策略的方法,学习约束的优先级,并逐渐合并有效的约束条件以利用多样化的不可行信息。cDPEA 是基于多种群策略的算法,同时使用 CDP 和惩罚函数方法平衡算法的收敛性和多样性。最后 3 个为基于多任务的算法,但他们设计了不同的辅助任务和知识迁移策略,因此适用于求解不同类型的问题。这 5 个对比算法从单约束信息利用角度、

约束处理技术角度、知识迁移策略角度和辅助任务角度设计了不同的策略,因此具有代表性并且被选为对比算法。

测试集。本文选择 DASCMOP[21]、LIRCMOP[19]和 SDC[22]作为测试集,它们分别包含 9,14,15 个测试函数,这 3 个测试集共包含 38 个测试函数。前两个测试集在目标空间包含了离散可行域和小可行域。SDC 在决策空间包含了大量局部不可行域。这 3 个测试集具有不同的特性,以评估算法的综合性能。

实验参数设置。对于两目标和三目标问题,所有算法种群大小分别设置为 100 和 120,最大函数评价次数设置为 20 万,每个算法的运行次数设置为 30[23]。对于使用的差分进化算法操作,比例因子 F 和杂交率 CR 分别从集合{0.6, 0.8, 1.0}和集合{0.1, 0.2, 1.0}中随机选择。此外,参数 β 设置为 0.1,参数 α 设置为 0.5,参数 ρ 设置为 0.3,参数 LR 设置为 10。对比算法的参数设置和原文保持一致。本文所有实验均在服务器上运行,其硬件环境配置为 Inter (R) Xeon (R) Gold 6230 CPU@2.10 GHz 640 GB RAM。

评价指标。使用的评价指标为反向世代距离 *IGD* 和超体积 *HV*。*IGD* 计算最优解和算法输出可行非支配解集的平均距离,而 *HV* 计算参考点和算法输出可行非支配解集围成的目标空间中区域的体积。*IGD* 越小或者 *HV* 越大,代表算法的收敛性和多样性越好。

3.2　对比结果和分析

表 1 提供了 6 种算法在 LIRCMOP 测试集上的 *IGD* 结果,其中每个问题上的最优结果用灰色表示。表 2 提供了所有算法在 DASCMOP 和 SDC 测试集上的统计 *IGD* 结果。图 3 提供了所有算法在 3 个测试集上的 *HV* 的对比情况。

表 1　6 种算法在 LIRCMOP 测试集上的 *IGD*

Table 1　*IGD* of six algorithms on LIRCMOP benchmark

问题	*IGD* 均值 (*IGD* 标准差)					
	CMOEA_MS	cDPEA	EMCMO	MTCMO	CMOEMT	CCMT
LIRCMOP1	3.0385e-1 (1.26e-1) -	2.6626e-2 (8.99e-3) +	1.2591e-1 (7.62e-2) -	2.4149e-2 (9.16e-3) +	2.5455e-2 (2.68e-2) +	4.4518e-2 (1.98e-2)
LIRCMOP2	1.9266e-1 (9.40e-2) -	6.3464e-3 (4.91e-4) +	1.1472e-1 (6.42e-2) -	1.2825e-2 (5.81e-3) =	3.6101e-2 (4.79e-2) =	1.6459e-2 (1.14e-2)
LIRCMOP3	3.2548e-1 (1.68e-1) -	7.9127e-3 (5.33e-3) +	1.3209e-1 (7.42e-2) -	2.7255e-2 (2.05e-2) +	4.6272e-2 (6.69e-2) -	3.7920e-2 (1.54e-2)
LIRCMOP4	2.7296e-1 (7.33e-2) -	6.1471e-3 (3.78e-3) +	1.5981e-1 (5.45e-2) -	2.7872e-2 (2.07e-2) +	6.5701e-2 (8.80e-2) =	3.4114e-2 (1.53e-2)
LIRCMOP5	4.2658e-1 (5.74e-1) -	1.2335e-2 (5.50e-3) -	8.6057e-3 (8.96e-4) -	7.6624e-2 (2.58e-1) -	1.3204e-2 (8.77e-3) -	6.8212e-3 (3.97e-4)
LIRCMOP6	3.9039e-1 (5.84e-1) =	7.3729e-3 (5.45e-4) -	7.7869e-3 (1.03e-3) -	8.8336e-2 (3.02e-1) -	1.1301e-2 (4.94e-3) -	6.1947e-3 (2.36e-4)
LIRCMOP7	3.3850e-2 (5.63e-2) =	8.1932e-3 (1.88e-3) -	7.5581e-3 (2.17e-4) -	1.3309e-2 (1.39e-2) -	1.0820e-2 (5.08e-3) -	7.3008e-3 (3.64e-4)
LIRCMOP8	1.6450e-2 (2.80e-2) =	1.3456e-2 (2.62e-2) =	7.5481e-3 (2.54e-4) =	7.6846e-3 (4.73e-4) =	8.6779e-3 (7.31e-4) -	7.4540e-3 (4.60e-4)
LIRCMOP9	4.9078e-1 (8.74e-2) -	1.9781e-1 (4.67e-2) -	2.3729e-2 (2.63e-2) -	3.2604e-1 (8.23e-2) -	8.4322e-2 (5.59e-2) -	8.6783e-3 (2.96e-3)
LIRCMOP10	1.7763e-1 (9.98e-2) -	1.3828e-2 (2.02e-2) -	7.9222e-3 (7.62e-4) -	8.7950e-3 (1.46e-3) -	1.1255e-2 (3.28e-3) -	5.6205e-3 (2.64e-4)
LIRCMOP11	1.6427e-1 (1.11e-1) -	5.2423e-2 (3.90e-2) -	4.3364e-3 (2.97e-3) -	1.6867e-1 (1.43e-1) -	5.2631e-3 (1.85e-3) -	2.4632e-3 (4.92e-5)
LIRCMOP12	2.2497e-1 (8.64e-2) -	1.3314e-1 (3.04e-2) -	4.8508e-3 (1.14e-3) -	1.8514e-1 (4.58e-2) -	3.2491e-2 (3.17e-2) -	3.1923e-3 (5.53e-5)
LIRCMOP13	1.0913e-1 (1.91e-3) -	1.0982e-1 (2.00e-3) -	1.0438e-1 (1.84e-3) -	8.7360e-1 (3.36e-1) -	1.0222e-1 (6.49e-3) -	9.3323e-2 (1.70e-3)
LIRCMOP14	9.2574e-2 (9.16e-4) -	9.4087e-2 (1.36e-3) -	9.1582e-2 (1.16e-3) -	8.6899e-1 (3.80e-1) -	9.1673e-2 (1.56e-3) -	8.7767e-2 (5.98e-4)

注:括号后的符号“+”、“-”及“ =”分别表示对比算法显著好于、明显差于和相似于 CCMT,下同。

LIRCMOP 和 DASCMOP 包含单模态特性的目标空间约束,这需要算法首先利用 UPF 信息快速逼近可行域,然后利用 UPF 和 CPF 中间的不可行解维持多样性并找到更多的可行域。表 1 显示 CCMT 在大部分 LIRCMOP 函数上取得了最优结果,这得益于 CCMT 中两个特性互补的辅助任务。对比算法中,只有 cDPEA 和 MTCMO 分别在 3 个和 1 个函数上取得了最优结果。LIRCMOP1 ~ LIRCMOP4 包含了非常小的可行域,需要算法利用多样化的不可行解,从而推动种群从不可行域逼近可行域。EMCMO 仅包含了无约束辅助任务,因此在大多数问题上取得了较差的表现。MTCMO 仅包含了使用 epsilon 方法的辅助任务以维持种群多样性,因此在部分问题上取得了较好的表现。虽然 CMOEMT 同时包含了无约束辅助任务和约束缩放辅助任务,但是种群独立产生后代,不充分的信息迁移使得算法产生了较差的结果。表 2 显示 CCMT 在 DASCMOP 测试集中的至少 6 个问题上显著优于 5 个对比算法,表明了所提算法的优异性。

表 2 在 DASCMOP 和 SDC 测试集上,基于 *IGD* 指标对比算法优于、差于或等于 CCMT 的函数数量

Table 2 On the DASCMOP and SDC benchmarks, the number of functions with algorithm better than, worse than, or equal to CCMT based on the *IGD* metric

对比算法	函数数量					
	测试集 DASCMOP			测试集 SDC		
	+	−	=	+	−	=
CMOEA_MS	0	8	1	2	9	4
cDPEA	2	6	1	2	9	4
EMCMO	1	8	0	2	9	4
MTCMO	1	7	1	4	7	4
CMOEMT	1	8	0	5	9	1

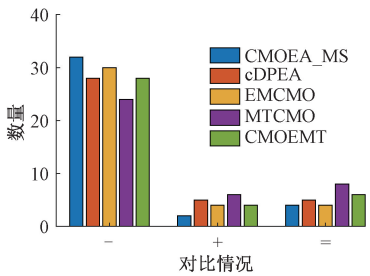

图 3 CCMT 与其他算法在 3 个测试集上的 *HV* 的对比情况

Figure 3 Comparison of *HV* between CCMT and other algorithms on three test sets

SDC 的决策空间包含了大量局部不可行域,导致算法容易陷入局部最优。所有方法中,CMOEA_MS、EMCMO 和 CMOEMT 在 3 个、1 个和 1 个函数上未能取得 100% 的可行率。CCMT 和 MTCMO 在 8 个和 5 个函数上取得了最优结果,且表 2 的统计结果显示 CCMT 至少在 7 个问题上显著优于 5 个对比算法,表明了所提方法的优异性。

经 3 个测试集共 38 个测试函数上的 Wilcoxon 秩和检验,在 *IGD* 指标上,CCMT 优于对比算法 CMOEA_MS、cDPEA、EMCMO、MTCMO、CMOEMT 的函数数量分别为 28,24,30,23,28 个,而差于 5 个对比算法的函数数量分别为 2,7,3,8,7 个。此外,图 3 显示 CCMT 在 *HV* 指标上依然取得了优异的表现,它优于 5 个对比算法的函数数量分别为 32,28,30,24,28 个,而差于 5 个对比算法的函数数量分别为 2,5,4,6,4 个。总之,*IGD* 和 *HV* 结果显示了所提方法在 3 个测试集上显著优于对比算法。

图 4 提供了所有算法在所有测试函数上的统计对比结果,其中 R^+ 和 R^- 分别代表 CCMT 和对比算法的表现,符号“ * ”代表两个算法之间存在显著差异。对于每个对比算法,R^+ 明显大于 R^-,并且除了 MTCMO 在 *IGD* 指标上与 CCMT 未存在显著差异,其余所有算法在两个指标上均与 CCMT 存在显著差异,证明 CCMT 显著优于对比算法。图 5 提供了每个算法基于 Friedman 检验的平均排名,CCMT 取得了最优的排名。

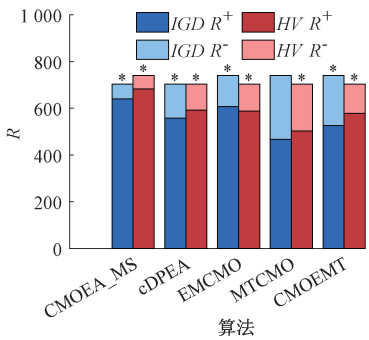

图 4 统计性检验结果

Figure 4 Statistical test results

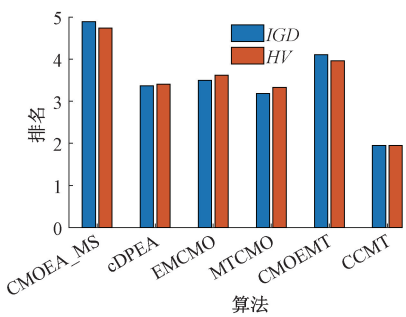

图 5 不同算法的 *IGD* 和 *HV* 排名

Figure 5 *IGD* and *HV* rankings of different algorithms

3.3　参数分析

本节进行了参数分析实验,对于 4 个参数,每个参数选取了 5 个不同的值,参数敏感性分析如表 3 所示。

表 3　参数敏感性分析

Table 3　Parameter sensitive analysis

LR	排名	α	排名	β	排名	ρ	排名
1	2.6	0	2.6	0	3.2	0	3.2
5	2.8	0.3	2.5	0.1	2.2	0.1	2.9
10	2.2	0.5	2.1	0.2	2.6	0.3	2.4
50	3.1	0.8	3.5	0.3	3.4	0.5	2.7
100	3.4	1.0	3.6	0.5	3.6	0.7	2.8

参数 *LR* 决定了利用历史信息的代数,*LR* 越大,种群利用历史信息的程度越深,从而使得种群偏向于全局探索。参数 α 控制全局奖励和种群奖励的比例,且 α 越小使得种群越偏向于全局探索。因此,*LR* 和 α 共同控制了算法全局探索和局部开发的平衡。表 3 显示两个参数的最优值均为 5 个参数的中间值,表明算法赋予了全局探索和局部开发相似的权重,使得种群既能寻找更优的解,同时也能维持多样性。

参数 β 控制何时开始竞争阶段。太小的 β 值使得奖励矩阵未能记录有效信息;而较大的 β 值使得资源分配未能被充分执行。因此,由表 3 可知,最优 β 值是一个较小的值。

参数 ρ 控制父代聚合概率。较大的值使得其他种群深度参与子代产生,提供的多样性可能会影响种群的进化方向;而太小的值则会抑制任务间的信息交流,导致有效信息无法被利用。因此,由表 3 可知,最优 ρ 值为较小的 0.3。

3.4　消融实验

CCMT 中包括两个主要创新点:资源分配和种群协同进化。为了验证资源分配的有效性,设计了无资源分配的算法变种 CCMT-noRA,其中每个任务的选择概率在进化过程中保持相同。为了验证种群协同进化的有效性,设计了两个算法变种:第一个是无父代聚合的算法变种 CCMT-noPA,第二个是无子代聚合的算法变种 CCMT-noOD。算法变种和 CCMT 的统计对比 *IGD* 结果如表 4 所示。在 *IGD* 指标上,CCMT 优于对比变种算法 CCMT-noRA、CCMT-noPA、CCMT-noOD 的函数数量为 21,25,33 个,证明了所提策略的有效性。CCMT-noRA 优于 CCMT 的函数数量为 5 个,这表明资源分配策略能够在不同的问题和不同的进化阶段为更有效的任务分配更多的计算资源。CCMT-noPA 优于 CCMT 的函数数量为 3 个,CCMT-noOD 优于 CCMT 的函数数量为 0 个。上述结果表明在子代产生和环境选择过程中的知识迁移均是有效的。

表 4　基于 *IGD* 指标,变种算法优于、差于或等于 CCMT 的函数数量

Table 4　The number of functions with the variant algorithm better than, worse than, or equal to CCMT based on the *IGD* metric

变种算法	函数数量		
	+	−	=
CCMT-noRA	5	21	12
CCMT-noPA	3	25	10
CCMT-noOD	0	33	5

4　结论

本文提出了一种基于竞争与合作多任务的约束多目标进化算法 CCMT。该算法中设置了一个主任务和两个互补的辅助任务,同时设计了基于竞争的资源分配策略和基于合作的种群协同进化策略。所提方法在 3 个不同特性的测试集上和 5 个最新的算法进行了对比,实验结果表明所提算法寻找可行解的能力较强,且多样性和收敛性明显优于对比算法,能够为决策者提供多样化的可行解。

虽然所提方法相比于 5 个最新的对比方法取得了显著优势,但该方法在具有非常小可行域的函数上未能取得最佳结果,因此未来将继续研究如何提升算法在该类问题上的效果,例如可以通过适应度地形分析技术提取小可行域的地形特征,从而指导种群在小可行域进行更为细致的搜寻。此外,基于多任务的约束多目标优化仍然是一个比较新的研究方向,目前研究主要关注于辅助任务的设计,在知识定义、知识迁移、任务内与任务间信息利用平衡等方面还需要进行深入研究。

参考文献:

[1]　李二超, 李进. 两阶段三存档集约束优化算法(TSDA)[J]. 郑州大学学报(工学版), 2018, 39(6): 23-29.
LI E C, LI J. Constraint optimization algorithm with two-stage and three-archive (TSDA)[J]. Journal of Zhengzhou University (Engineering Science), 2018, 39(6): 23-29.

[2]　陈少淼, 陈瑞, 梁伟, 等. 面向复杂约束优化问题的进化算法综述[J]. 软件学报, 2023, 34(2): 565-581.
CHEN S M, CHEN R, LIANG W, et al. Overview of ev-

olutionary algorithms for complex constrained optimization problems [J]. Journal of Software, 2023, 34(2): 565-581.

[3] 王林锋, 揭丽琳, 黎明, 等. 基于自适应双阶段分级均衡的约束高维多目标进化算法 [J]. 控制与决策, 2024, 40(5): 1512-1522.
WANG L F, JIE L L, LI M, et al. Constrained many-objective evolutionary algorithm based on adaptive two-stage hierarchical equilibrium [J]. Control and Decision, 2024, 40(5): 1512-1522.

[4] FAN Z, LI W, CAI X, et al. Push and pull search for solving constrained multi-objective optimization problems [J]. Swarm and Evolutionary Computation, 2019, 44: 665-679.

[5] TIAN Y, ZHANG Y, SU Y, et al. Balancing objective optimization and constraint satisfaction in constrained evolutionary multiobjective optimization [J]. IEEE Transactions on Cybernetics, 2021, 52(9): 9559-9572.

[6] TIAN Y, ZHANG T, XIAO J, et al. A coevolutionary framework for constrained multiobjective optimization problems [J]. IEEE Transactions on Evolutionary Computation, 2020, 25(1): 102-116.

[7] MING M, TRIVEDI A, WANG R, et al. A dual-population-based evolutionary algorithm for constrained multiobjective optimization [J]. IEEE Transactions on Evolutionary Computation, 2021, 25(4): 739-753.

[8] QIAO K J, YU K J, QU B Y, et al. An evolutionary multitasking optimization framework for constrained multiobjective optimization problems [J]. IEEE Transactions on Evolutionary Computation, 2022, 26(2): 263-277.

[9] QIAO K J, YU K J, QU B Y, et al. Dynamic auxiliary task-based evolutionary multitasking for constrained multiobjective optimization [J]. IEEE Transactions on Evolutionary Computation, 2022, 27(3): 642-656.

[10] LI G, WANG Z, GAO W, et al. Decoupling constraint: task clone-based multi-tasking optimization for constrained multi-objective optimization [J]. IEEE Transactions on Evolutionary Computation, 2025, 29(2): 404-417.

[11] MING F, GONG W, WANG L, et al. Constrained multiobjective optimization via multitasking and knowledge transfer [J]. IEEE Transactions on Evolutionary Computation, 2022, 28(1): 77-89.

[12] YE Q, WANG W, LI G, et al. A self-organizing assisted multi-task algorithm for constrained multi-objective optimization problems [J]. Information Sciences, 2024, 664: 120339-120360.

[13] WANG J, LI Y, ZHANG Q, et al. Cooperative multiobjective evolutionary algorithm with propulsive population for constrained multiobjective optimization [J]. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 2021, 52(6): 3476-3491.

[14] GARCíA J L L, MONROY R, HERNáNDEZ V A S, et al. COARSE-EMOA: An indicator-based evolutionary algorithm for solving equality constrained multi-objective optimization problems [J]. Swarm and Evolutionary Computation, 2021, 67: 100983-100997.

[15] 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.

[16] WANG C, LIU Z, QIU J, et al. Adaptive constraint handling technique selection for constrained multi-objective optimization [J]. Swarm and Evolutionary Computation, 2024, 86: 101488-101503.

[17] SONG S, ZHANG K, ZHANG L, et al. A dual-population algorithm based on self-adaptive epsilon method for constrained multi-objective optimization [J]. Information Sciences, 2024, 655: 119906-119923.

[18] LIU Z Z, QIN Y, SONG W, et al. Multiobjective-based constraint-handling technique for evolutionary constrained multiobjective optimization: a new perspective [J]. IEEE Transactions on Evolutionary Computation, 2022, 27(5): 1370-1384.

[19] FAN Z, LI W, CAI X, et al. An improved epsilon constraint-handling method in MOEA/D for CMOPs with large infeasible regions [J]. Soft Computing, 2019, 23: 12491-12510.

[20] MING M, WANG R, ISHIBUCHI H, et al. A novel dual-stage dual-population evolutionary algorithm for constrained multiobjective optimization [J]. IEEE Transactions on evolutionary Computation, 2021, 26(5): 1129-1143.

[21] FAN Z, LI W, CAI X, et al. Difficulty adjustable and scalable constrained multiobjective test problem toolkit [J]. Evolutionary Computation, 2020, 28(3): 339-378.

[22] QIAO K, LIANG J, YU K, et al. Evolutionary constrained multiobjective optimization: Scalable high-dimensional constraint benchmarks and algorithm [J]. IEEE Transactions on Evolutionary Computation, 2023, 28(4): 965-979.

[23] ZHONG X, YAO X, QIAO K, et al. A multitasking-based constrained multi-objective evolutionary algorithm with forward and backward stages [J]. IEEE Transactions on Emerging Topics in Computational Intelligence, 2024, 8(5): 3474-3488.

(下转第76页)

Optimized Configuration of Two Layers of Smart Energy Storage Soft Switches for Distribution Networks Considering Demand Response

CHENG Zixia[1], TANG Xing[1], CHAI Xuzheng[2], GUO Zichan[1], YAO Wenbo[1]

(1. School of Electrical and Information Engineering, Zhengzhou University, Zhengzhou 450001, China; 2. State Grid Xuchang Power Supply Company, Xuchang 461000, China)

Abstract: Aiming at the problems of increasing network losses and voltage overruns faced by power systems containing a high proportion of new energy, a two-layer planning strategy of soft open point integrated with energy storage system (E-SOP) for distribution networks considering demand response was proposed. Firstly, the typical scenarios of wind power output were generated based on Frank-Copula function considering the relevant characteristics of wind power output. Secondly, a two-layer planning model of E-SOP was established. The upper layer took the objective of the lowest annual comprehensive operating cost of the distribution network for the siting and capacity setting of E-SOP, The lower layer took the demand response participation into consideration. The operation optimization was carried out with the objective of the minimum operating cost of each scenario, and the multi-strategy improved whale optimization algorithm (MIWOA) and Second-order Conic Programming (SOCP) were used to solve the model. Finally, the IEEE33-node systems were used for example analysis, and the simulation results showed that the annual integrated costs of the systems were reduced by 7.94%, respectively, which verified that the proposed scheme could effectively improve the stability and economy of distribution network operation.

Keywords: Soft Open Point integrated with energy storage system; Frank-Copula function; two-layer planning; whale optimization algorithms; second-order cone planning

(上接第 58 页)

A Constrained Multi-objective Evolutionary Algorithm Based on Competition and Cooperation Multi-tasking

ZHANG Meng[1], LIANG Jing[2], QIAO Kangjia[2], YUE Caitong[2], WANG Xilu[3]

(1. School of Energy and Intelligent Engineering, Henan College of Animal Husbandry and Economics, Zhengzhou 450046, China; 2. School of Electrical and Information Engineering, Zhengzhou University, Zhengzhou 450001, China; 3. School of Computer Science and Electronic Engineering, University of Surrey, Surrey GU2 7XH, U. K.)

Abstract: Constrained multi-objective evolutionary algorithm based on multi-tasking competition and cooperation has problems in resource allocation and collaborative optimization, resulting in low effectiveness populations wasting computational resources, and underutilized high-quality solution information. Therefore, in this study, a constrained multi-objective evolutionary algorithm based on competitive and cooperation multitasking was proposed, which included two main strategies. Firstly, a competition-based resource allocation strategy was proposed to achieve adaptive allocation of computing resources based on the historical performance of each task population. Secondly, a collaborative optimization strategy based on parent aggregation and offspring diffusion was designed to generate high-quality offspring through cross-task cooperation and spread them to various task populations, achieving efficient utilization of effective information. The proposed algorithm was compared with five other advanced algorithms (CMOEA_MS, cDPEA, EMCMO, MTCMO, and CMOEMT) on 38 test functions, and the results showed that the proposed algorithm achieved optimal results on 25 and 26 functions with *IGD* and *HV* indicators, respectively, and was superior to the compared algorithms on at least 23 and 24 functions, respectively. The proposed algorithm had a feasibility rate of 100% on all functions and can effectively solve constrained multi-objective optimization problems.

Keywords: constrained multi-objective optimization; evolutionary algorithm; multi-tasking; resource allocation; collaborative optimization