基于树形结构无界存档的多目标粒子群算法

纪昌明†, 马皓宇, 李宁宁, 吴嘉杰, 彭 杨, 王丽萍

(华北电力大学 可再生能源学院，北京 102206)

摘 要: 多目标优化算法大多采用基于线性链表结构的有界Pareto存档策略,其存在迭代过程中Pareto前沿震荡衰退等弊端以及相关参数难以预先确定等技术难题. 为此,构造一种适用于大规模存档集合的树形结构,并利用其取代线性结构以保证存档维护与管理的高效性,进而提出基于树形结构的无界存档策略. 在此基础上,将基于正交设计的种群初始化、基于树形结构的存档更新以及基于树形结构的最优个体选择引入多目标粒子群优化,提出基于树形结构无界存档的多目标粒子群算法. 最后,通过测试函数上的仿真实验验证了所提出策略与算法的科学性和有效性.

关键词: 多目标优化；树形结构；无界存档；粒子群优化；正交设计

中图分类号: TP301.6 **文献标志码:** A

DOI: 10.13195/j.kzyjc.2019.0276 **开放科学(资源服务)标识码(OSID):**

引用格式: 纪昌明,马皓宇,李宁宁,等. 基于树形结构无界存档的多目标粒子群算法[J]. 控制与决策, 2020, 35(11): 2675-2686.

Multi-objective particle swarm optimization algorithm based on tree-structured unbounded archive

JI Chang-ming†, MA Hao-yu, LI Ning-ning, WU Jia-jie, PENG Yang, WANG Li-ping

(College of Renewable Energy，North China Electric Power University，Beijing 102206，China)

Abstract: Most of the current multi-objective optimization algorithms adopt the bounded Pareto archive strategy based on the linear list structure, which has the drawbacks of Pareto fronts' oscillation, shrinking and other technical difficulties such as pre-determining relevant parameters. Therefore, this paper constructs a tree structure suitable for the storage and update of large-scale archive, replacing the linear structure to ensure high efficiency of archive maintenance and management. And a tree-structured unbounded archive strategy is proposed. We introduce population initialization based on orthogonal design, archive updating and optimal individual selection based on the tree structure into multi-objective particle swarm optimization, and propose a multi-objective particle swarm optimization algorithm based on tree-structured unbounded archive. Finally, simulation experiments on test functions verify the feasibility and effectiveness of the improved strategies and algorithm.

Keywords: multi-objective optimization；tree structure；unbounded archive；particle swarm optimization；orthogonal design

0 引 言

多目标优化问题(MOP)广泛存在于科学研究与工程应用中,要求在给定条件下同时对多个相互冲突的目标进行优化[1-5]. 不同于单目标优化问题(SOP), MOP要求寻找高质量解的集合而非单个最优解. 在多目标优化问题的研究初期,研究人员通常利用加权等方式将MOP转化为SOP,然后利用成熟的数学规划方法进行求解. 这种求解方式不仅效率低,而且对目标权重和目标次序极为敏感,无法满足实际应用要求. 多目标进化算法因每次运行可以找到一组非支配解,且不易受问题的Pareto前沿的形状或连续性影响,因而受到学者们的广泛关注与应用[6].

自Zitzler于1999年提出SPEA后,大部分的多目标进化算法均引入精英策略以提升算法性能,其中最具代表性的有NSGA-II、SPEA2以及PESA-II[7-9]. 这些方法以Pareto支配关系指引搜索进程,通过Pareto存档存储进化过程中搜寻到的优秀个体,最终输出存档集合所对应的目标向量集作为Pareto近似前沿. 国外学者已经证明了Pareto存档不仅是改善算法性能的重要手段,更是保证算法收敛的关键措施,故该策

收稿日期: 2019-03-11；**修回日期:** 2019-07-09.

基金项目: 国家自然科学基金项目(51679088, 51279062)；“十三五”国家重点研发计划课题(2016YFC0402208).

责任编委: 巩敦卫.

†通讯作者. E-mail: cmji@ncepu.edu.cn.

略已成为多目标优化算法的重要组成部分[10].

存档更新是存档的关键操作,其时间复杂度通常与所处理问题的目标维数以及迭代过程中Pareto存档的规模成正比. 为保证计算速度一般采用有界存档,即当存档个体数超出预设上限时将一部分个体舍弃,这种方法虽能提高存档更新速度,但存在诸多弊端: 当存档规模越限时,部分潜在有效解的舍弃势必会造成存档质量的下降,导致Pareto近似前沿出现衰退与震荡[11]; 有界存档需额外引入一系列策略以保证输出结果的质量,这将增加算法的复杂度且与策略相关的参数较难确定与调整[12]; 近年提出的多目标决策方法如理想均变率法[13]要求提供具有良好分布特性的高密度Pareto前沿,而有界存档难以满足. 上述问题均是因算法对存档容量设限造成的,最简单的处理方式是取消存档的规模限制. 但是,当所处理问题的目标维数较大时,如何实现存档的高效维护与管理成为主要难题[14].

为解决这一难题,本文提出基于树形结构的无界存档(tree-structured unbounded archive, TUA),使用规模不受限的存档收集非支配个体,并利用树形结构作为存档的数据存储结构以减少存档更新中存在的冗余支配关系比较,进而提升存档的更新效率. 在此基础上,将基于正交设计的种群初始化策略、基于树形结构的最优个体选择与更新策略引入多目标粒子群算法中,提出基于树形结构无界存档的多目标粒子群算法(MOPSO/TUA),并通过测试函数检验所提出策略的效果以及新算法的有效性.

1 多目标优化基本概念

MOP问题可表述为

$$\begin{aligned}&\min \boldsymbol{y}=\boldsymbol{f}(\boldsymbol{x})=(f_1(\boldsymbol{x}),f_2(\boldsymbol{x}),\dots,f_m(\boldsymbol{x}));\\&\text{s.t. }\boldsymbol{g}(\boldsymbol{x})=(g_1(\boldsymbol{x}),g_2(\boldsymbol{x}),\dots,g_p(\boldsymbol{x}))\geqslant 0.\end{aligned}\tag{1}$$

其中: $\boldsymbol{x}=(x_1,x_2,\dots,x_n)\in\boldsymbol{X}\subset\boldsymbol{R}^n$为$n$维决策向量, $\boldsymbol{y}=(y_1,y_2,\dots,y_m)\in\boldsymbol{Y}\subset\boldsymbol{R}^m$为$m$维目标向量, $\boldsymbol{f}$为将n维决策空间映射至m维目标空间的目标函数集合, $\boldsymbol{g}(\boldsymbol{x})\geqslant 0$为$p$个不等式约束. 在式(1)的基础上给出与多目标优化相关的如下重要概念[15].

定义1 Pareto支配: $\boldsymbol{x}_A,\boldsymbol{x}_B\in\boldsymbol{X}_f$为问题的两个可行解, $\boldsymbol{X}_f$为可行域, $\boldsymbol{x}_A$支配$\boldsymbol{x}_B(\boldsymbol{x}_A\succ\boldsymbol{x}_B)$当且仅当

$$\begin{aligned}&\forall k\in\{1,2,\dots,m\},\ f_k(\boldsymbol{x}_A)\leqslant f_k(\boldsymbol{x}_B)\bigwedge\\&\exists k\in\{1,2,\dots,m\},\ f_k(\boldsymbol{x}_A)<f_k(\boldsymbol{x}_B).\end{aligned}\tag{2}$$

定义2 Pareto弱支配: $\boldsymbol{x}_A,\boldsymbol{x}_B\in\boldsymbol{X}_f$为两个可行解, $\boldsymbol{x}_A$弱支配$\boldsymbol{x}_B(\boldsymbol{x}_A\succeq\boldsymbol{x}_B)$当且仅当

$$\boldsymbol{x}_A\succ\boldsymbol{x}_B\bigvee\boldsymbol{x}_A=\boldsymbol{x}_B.\tag{3}$$

定义3 Pareto最优解: $\boldsymbol{x}^*\in\boldsymbol{X}_f$为Pareto最优解当且仅当

$$\neg\exists\boldsymbol{x}\in\boldsymbol{X}_f:\boldsymbol{x}\succ\boldsymbol{x}^*.\tag{4}$$

定义4 Pareto最优解集: Pareto最优解集是所有Pareto最优解的集合,定义如下:

$$\mathrm{PS}\triangleq\{\boldsymbol{x}^*|\neg\exists\boldsymbol{x}\in\boldsymbol{X}_f,\boldsymbol{x}\succ\boldsymbol{x}^*\}.\tag{5}$$

定义5 Pareto前沿: Pareto最优解集中所有解对应的目标向量所组成的曲线或曲面称为Pareto前沿,即

$$\mathrm{PF}\triangleq\{\boldsymbol{f}(\boldsymbol{x}^*)|\boldsymbol{x}^*\in\mathrm{PS}\}.\tag{6}$$

定义6 Pareto存档: Pareto存档是至今为止搜索到的非支配解的集合,集合中的任意两个个体满足相互非支配的关系,即

$$F_t=\{\boldsymbol{x}_t^*|\neg\exists\boldsymbol{x}\in F_t,\boldsymbol{x}\succ\boldsymbol{x}_t^*\}.\tag{7}$$

多目标进化算法旨在获得逼近真实Pareto前沿的非支配解集,同时要求集合所对应的近似前沿的分布具有均匀性和广泛性.

2 基于树形结构的无界存档

2.1 传统有界存档策略

目前使用的Pareto存档通常以线性链表作为其数据存储结构,按是否设置容量上限分为有界存档与无界存档. 线性结构的存档更新流程为: 假设第t代新生成个体$\boldsymbol{x}$,将$\boldsymbol{x}$与存档F_t存储的集合$W=\{\boldsymbol{w}_1,\boldsymbol{w}_2,\dots,\boldsymbol{w}_{Nf}\}$中的所有成员依次进行比较,直至发现个体弱支配$\boldsymbol{x}$或所有个体均完成比较. 若发现被$\boldsymbol{x}$支配的存档个体,则将其从所在链表位置删除; 若比较完成后发现$\boldsymbol{x}$不被集合中的任意个体弱支配,则认为其是非支配个体并插至链表末尾[16]. 上述操作的时间复杂度为$O(mN_f)$, N_f为当前存档规模.

若存档的规模设限,则需在$\boldsymbol{x}$加入后判断集合的基数是否超出预设上限$N_f_\max$; 若超过则执行删除操作. 由于集合中的个体无法通过Pareto支配关系区分优劣,需额外引入性能指标,通过移除就该指标而言的低质量个体使存档满足容量约束. 决定存档个体去留的删除策略对输出结果的影响很大,设计不当将产生一系列问题. 以经典算法为例[17]: NSGA-II采取一次性移除$(N_f-N_f_\max)$个拥挤距离最小个体的删除策略,因最近邻密度值只需计算一次,故计算效率较高,但可能导致存档出现断层现象; SPEA2采取k-th近邻规则迭代移除$(N_f-N_f_\max)$个个体的删除策略,所得解的分布均匀性是很多算法无法超越的,但每次仅移除单个个体且需对剩余个体的密度值重新计算,导致其计算复杂度高达$O((N+N_f)^3)$; PESA-II采取迭代删除$(N_f-N_f_\max)$个位

于最拥挤超格内的随机选择个体的删除策略,网格机制的引入提高了计算效率,但网格构建所需设置的参数对结果影响很大,在缺乏所处理问题的目标空间先验信息的情况下较难确定.

上述问题可归结为有界存档很难在保证存档集合质量的条件下降低计算复杂度,而且删除策略可能导致存档所对应的近似前沿在迭代过程中出现衰退与震荡现象[18]. 无界存档虽可避免上述问题,但其更新耗时与内存占用是实际应用中的一大障碍,故本文构建适用于大规模存档数据存储的树形结构以提高存档的数据存储与更新效率.

2.2　基于树形结构的无界存档

2.2.1　定义与原理介绍

为便于说明,首先定义决策空间中的点与点、点与整个集合间的支配关系.

定义 7　$\boldsymbol{x}_1, \boldsymbol{x}_2 \in \boldsymbol{X}$ 为MOP的两个解,两者间的所有可能支配关系如下:

$$\boldsymbol{x}_1.\mathrm{dominace}(\boldsymbol{x}_2) = \begin{cases} 1: \boldsymbol{x}_1 \succ \boldsymbol{x}_2, \\ 2: \boldsymbol{x}_1 \prec \boldsymbol{x}_2, \\ 0: \boldsymbol{x}_1 \sim \boldsymbol{x}_2, \\ -1: \boldsymbol{f}(\boldsymbol{x}_1) = \boldsymbol{f}(\boldsymbol{x}_2). \end{cases} \tag{8}$$

其中: 值为 1 表示 $\boldsymbol{x}_1$ 支配 $\boldsymbol{x}_2$,值为 2 表示 $\boldsymbol{x}_1$ 被 $\boldsymbol{x}_2$ 支配,值为 0 表示 $\boldsymbol{x}_1$ 与 $\boldsymbol{x}_2$ 相互非支配,值为 -1 表示 $\boldsymbol{x}_1$ 与 $\boldsymbol{x}_2$ 的目标向量相等.

定义 8　$\boldsymbol{x} \in \boldsymbol{X}$ 为MOP的解,$W = \{\boldsymbol{w}_i | \boldsymbol{w}_i \in \boldsymbol{X}, i \in \{1,2,\ldots,k\}\}$ 为MOP的解集,两者间的所有可能支配关系如下:

$$\boldsymbol{x}.\mathrm{dominace}(W) = \begin{cases} 1: \forall i \in \{1,2,\ldots,k\},\ \boldsymbol{x} \succ \boldsymbol{w}_i; \\ 2: \forall i \in \{1,2,\ldots,k\},\ \boldsymbol{x} \prec \boldsymbol{w}_i; \\ 0: \forall i \in \{1,2,\ldots,k\},\ \boldsymbol{x} \sim \boldsymbol{w}_i. \end{cases} \tag{9}$$

其中: 值为 1 表示 $\boldsymbol{x}$ 支配 W 中的任意个体,值为 2 表示 $\boldsymbol{x}$ 被 W 中的任意个体支配,值为 0 表示 $\boldsymbol{x}$ 与 W 中的任意个体相互非支配.

存档 F_t 存储的集合记为 $W=\{\boldsymbol{w}_1, \boldsymbol{w}_2, \ldots, \boldsymbol{w}_{N_f}\}$,对应的目标向量集记为 $Z = \{\boldsymbol{z}_1, \boldsymbol{z}_2, \ldots, \boldsymbol{z}_{N_f}\}$,包含 Z 中所有点的超格为 $C_m = \{(y_1, y_2, \ldots, y_m) | \forall i \in \{1,2,\ldots,m\}, l_i \leqslant y_i \leqslant u_i\}$,超格顶点为 $\boldsymbol{P} = \{(y_1, y_2, \ldots, y_m) | \forall i \in \{1,2,\ldots,m\}, y_i \in \{l_i, u_i\}\}$,各维分量最小值组成的顶点称为最小顶点 $\boldsymbol{P}_{\min}$,各维分量最大值组成的顶点称为最大顶点 $\boldsymbol{P}_{\max}$. $\boldsymbol{P}_{\min}$ 的各维度值均小于 Z 中任意点的对应值,故 $\boldsymbol{P}_{\min}$ 支配集合 Z; $\boldsymbol{P}_{\max}$ 的各维度值均大于 Z 中任意点的对应值,故 $\boldsymbol{P}_{\max}$ 被集合 Z 支配.

如图 1 所示,将点 $\boldsymbol{x}$ 的目标向量 $\boldsymbol{y}$ 与 $\boldsymbol{P}_{\min}$、$\boldsymbol{P}_{\max}$ 进行比较,当结果满足某些条件时可直接推导出 $\boldsymbol{x}$ 与集合 W 的支配关系. 首先比较目标向量 $\boldsymbol{y}$ 与 $\boldsymbol{P}_{\max}$. 若 $\boldsymbol{y}$ 被 $\boldsymbol{P}_{\max}$ 弱支配,则由Pareto支配传递性可得 $\boldsymbol{x}$ 被 W 支配; 否则比较 $\boldsymbol{y}$ 与 $\boldsymbol{P}_{\min}$. 若 $\boldsymbol{y}$ 弱支配 $\boldsymbol{P}_{\min}$,则同样由支配传递性可得 $\boldsymbol{x}$ 支配 W; 否则检查两次比较的结果. 若均为相互非支配,则可知 $\boldsymbol{y}$ 中至少存在一个分量其值小于 Z 中所对应维度的最小值,至少存在一个分量其值大于 Z 中所对应维度的最大值,故 $\boldsymbol{x}$ 与 W 相互非支配; 若上述条件均不满足,则无法仅通过两次比较得出点与集合间的关系.

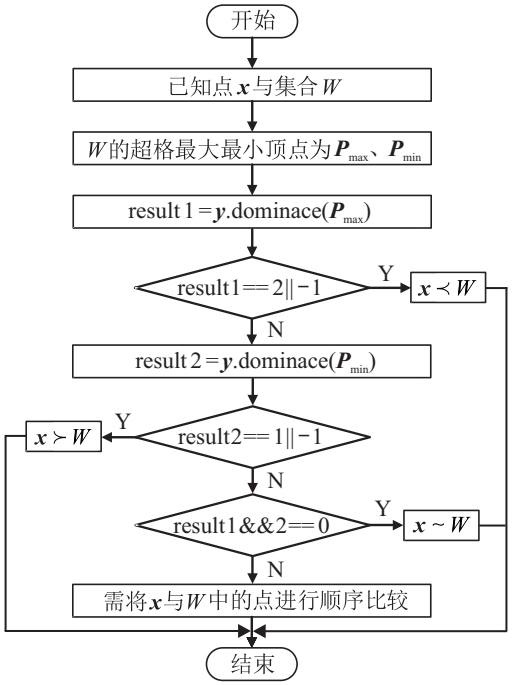


图 1　点与集合间支配关系的快速判别

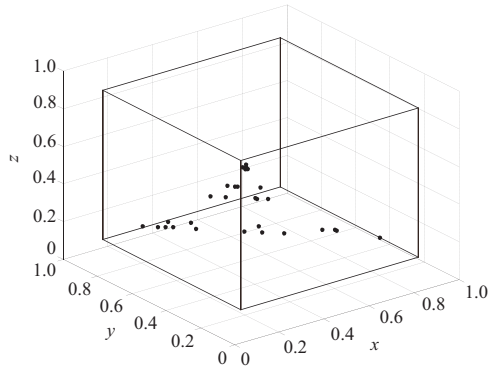
本文将可实现与集合关系快速判定的点所在的目标空间区域称为该集合的快速判别区域,若 $\boldsymbol{x}$ 位于存档集合的快速判别区域,则可通过与超格顶点的比较直接得出其与集合间的关系,进而判断 $\boldsymbol{x}$ 能否进入存档,本质是通过Pareto支配性质减少冗余比较. 然而,随着算法迭代搜索的进行,存档集合逐渐扩大,覆盖目标向量集的超格随之增大,快速判别区域减少,致使实现快速判别的几率降低. 本文从以下两点入手解决该问题:

1) 超格最小化. 利用集合 Z 各维分量的准确范围确定超格所对应维度的范围,即将超格定义为 $C_m = \{(y_1, y_2, \ldots, y_m) | \forall i \in \{1,2,\ldots,m\}, \min(Z_i) \leqslant y_i \leqslant \max(Z_i)\}$ 以最小化超格占据的区域.

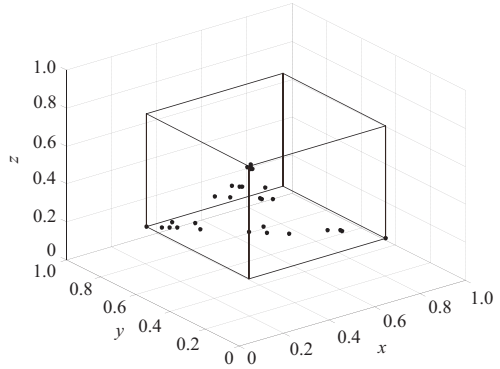
2) 集合递归划分. 当集合达到一定规模时,将其划分为若干个目标空间中相互接近的子集,对各子集分别建立超格,当子集达到相应规模时再对其进行相

同操作.

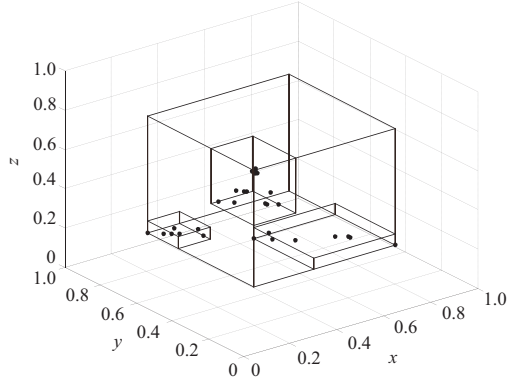
下面通过一个简单的实例验证以上两点的效果. 在图2所示的$[0,1]^3$目标空间中,图2(a)的黑点集合是基数为30的非支配解集所对应的目标向量集Z,Z各维分量的准确范围均为$0.2 \sim 0.8$,随机生成该目标空间内基数为10 000的点集$\boldsymbol{R}$,将$\boldsymbol{R}$中各点依次与Z进行比较以判断其能否加入集合,通过平均比较次数评价效率. 首先采用传统的顺序比较方式,测得平均比较次数为21.15;然后构建图2(a)所示的超格$C_m = \{(y_1,y_2,y_3)|0.1 \leqslant y_1,y_2,y_3 \leqslant 0.9\}$,采用快速判别方式进行比较,测得平均比较次数为21.53. 为提高判别效率,首先实现图2(b)所示的超格最小化,即$C_m = \{(y_1,y_2,y_3)|0.2 \leqslant y_1,y_2,y_3 \leqslant 0.8\}$,



(a) 实例展示1



(b) 实例展示2



(c) 实例展示3

图 2 三维目标空间实例

可将平均比较次数由21.53降至17.15;然后,通过集合递归划分将Z划分为图2(c)所示的目标空间中相互接近的子集$\{\text{Child_}Z(1),\text{Child_}Z(2),\text{Child_}Z(3)\}$,对子集构建超格. 比较时先判断$R$中的点是否位于$Z$的快速判别区域,若否,则依次检查其是否位于$\text{Child_}Z(i)$的快速判别区域,若位于,则直接得出其与$\text{Child_}Z(i)$的关系;否则需要与$\text{Child_}Z(i)$中的点依次进行比较. 若该点不被$\text{Child_}Z(i)$中的任意点弱支配,则继续与$\text{Child_}Z(i+1)$进行比较. 此时的平均比较次数为10.19,较传统方法的21.15减少幅度明显,由此表明了改进方法的有效性.

树形结构作为经典的非线性层次嵌套结构,适用于递归形式数据的存储与查找[19]. 本文将该结构作为存档的数据存储结构,定义如下:树中每个结点包含W、$P_{\max}$、$P_{\min}$、Parent、Child这5个域,其中W表示父结点处划分所得的存档子集,$P_{\max}$和$P_{\min}$表示覆盖W目标向量集的超格的最大最小顶点,Parent表示该结点的父结点,Child表示该结点的子结点集合. 树形结构的使用将加重计算机的内存负担,为此,本文提供两种解决方案:方案1通过消除集合信息的重复存储以降低内存占用,方案2将集合数据转至外存存储以减轻内存负担. 因方案2将增加数据传输时间,故推荐使用方案1. 方案具体措施如下:

1) 树中叶结点存储所分配点集的完整信息,内部结点的点集是其子树下所有叶结点点集的并集,故无需重复存储点集信息.

2) 将树中各结点的集合信息存于数据库中,当相关操作需读取或修改数据时,从数据库中进行实时存取.

2.2.2 基于树形结构的无界存档更新

将2.2.1节中的树形结构作为无界存档的数据存储结构以提高计算效率,相应的存档更新流程与线性结构的有较大差异. TUA的更新操作记为$F_t.\text{ArchiveUpdate}(\boldsymbol{x})$,通过新点$\boldsymbol{x}$更新存档$F_t$. 具体步骤为:先判断$\boldsymbol{x}$能否加入存档,再将$\boldsymbol{x}$与树的根结点root处的点集(根结点对应完整的存档集合)进行比较. 若$\boldsymbol{x}$不被集合中的任意个体弱支配,则将$\boldsymbol{x}$插至树中;否则拒绝$\boldsymbol{x}$加入.

存档更新有两个主要操作:支配关系比较与插入操作. 前者记作$\text{flag} = \boldsymbol{x}.\text{dominace}(\text{node})$,如图3所示,将点$\boldsymbol{x}$与结点node所对应的点集进行比较,并返回bool值以表明$\boldsymbol{x}$是否被集合node.W中个体弱支配. 该操作的具体步骤为:先检测点$\boldsymbol{x}$是否位于node.W的快速判别区域,即将$\boldsymbol{x}$与结点对应的超格

顶点 node.$\boldsymbol{P}_{\max}$、node.$\boldsymbol{P}_{\min}$ 进行比较,若 $\boldsymbol{x}$ 被 $\boldsymbol{P}_{\max}$ 弱支配,则 $\boldsymbol{x}$ 被 node.W 支配,立即返回 flag = false;若 $\boldsymbol{x}$ 弱支配 $\boldsymbol{P}_{\min}$,则 $\boldsymbol{x}$ 支配 node.W 及其子集,删除该结点及其子树以保证存档集合的非支配性,并返回 flag = true. 若两次比较的结果均为不相关,则 $\boldsymbol{x}$ 与 node.W 中的任意个体相互非支配,返回 flag = true;若上述条件均不满足,则检查该结点是否位于底层. 若是,则无法实现快速判别,将 $\boldsymbol{x}$ 与 node.W 中的个体依次进行比较;若否,则将 $\boldsymbol{x}$ 与子结点集合 node.Child 中的各结点依次进行比较,比较方式同上. 上述操作中若发现被 $\boldsymbol{x}$ 支配的存档个体,则需将其从树中删除.

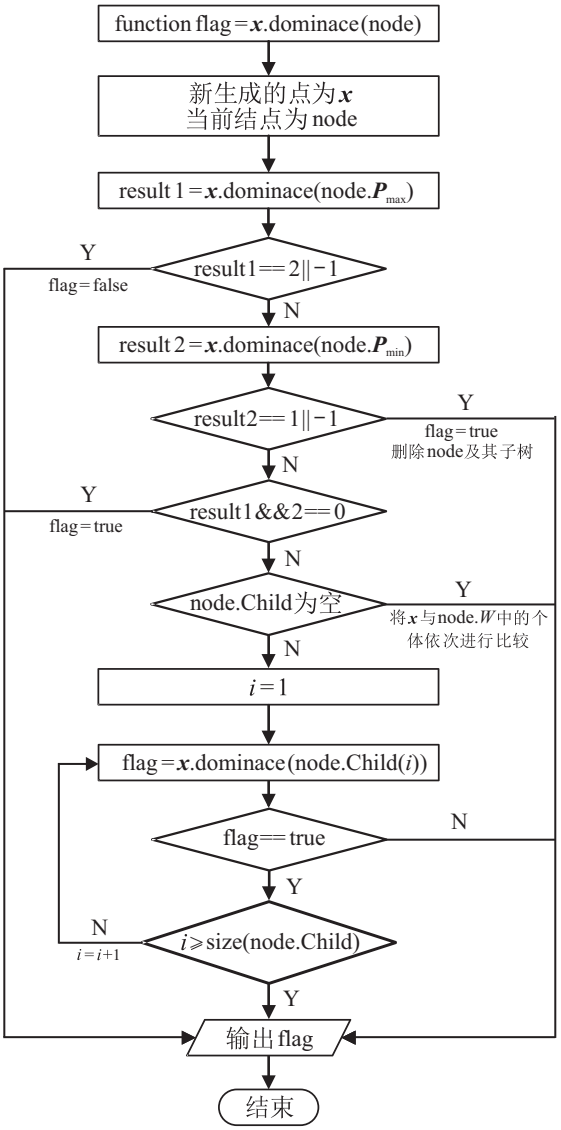


图 3　TUA 支配关系比较流程

插入操作记为 node.insertpoint($\boldsymbol{x}$),将点 $\boldsymbol{x}$ 插入树中的结点 node. 具体步骤为:先判断 node 在树中的位置,若位于树的底层,则将 $\boldsymbol{x}$ 直接加入 node.W 即可;否则,将 $\boldsymbol{x}$ 插入该结点下距其最近的子结点并重复上述操作. 该操作需注意以下 3 点:

1) 当 $\boldsymbol{x}$ 并入叶结点的集合后,若集合规模超出预设值 leafsize,则需通过目标空间中的聚类对集合进行划分,产生预设数量 childsize 个的子集,并建立相同数量的子结点以存储子集数据;

2) 当 $\boldsymbol{x}$ 并入叶结点的集合后,若相应的超格发生改变,则需及时更新;

3) 当目标各维分量的量纲不同或尺度存在明显差异时,为防止其对操作中涉及的目标空间内的距离计算产生影响,需利用各维分量的范围或树中根结点的超格范围进行距离的标准化计算.

3　基于树形结构无界存档的多目标粒子群算法

粒子群算法是基于种群的单目标元启发式优化算法,其具有形式简洁、收敛速度快以及参数调整灵活等优点而被认为是求解多目标优化问题的最具潜

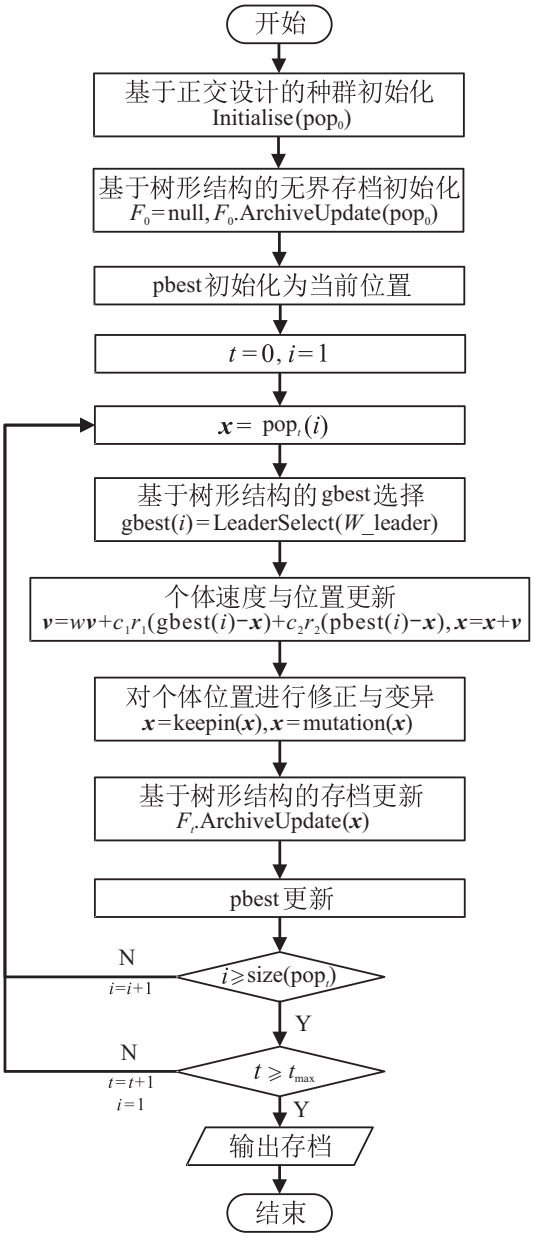


图 4　MOPSO/TUA 算法流程

力的方法之一[20-21]. Coello等[22]基于自适应网格与变异策略提出的MOPSO是最为经典的多目标优化算法之一. 本文以此算法为基础提出MOPSO/TUA,采用TUA作为算法的精英策略,针对种群初始化、存档维护与个体选择进行改进以提升算法性能. MOPSO/TUA的算法流程如图4所示,下面对算法中的重要操作进行介绍.

3.1 基于正交设计的种群初始化

初始种群的质量对启发式算法的输出结果有较大影响,实际处理的多目标优化问题,其最优解在决策空间中的分布往往是未知的,故要求算法的初始种群尽可能均匀散布在整个决策空间以充分发掘最优解信息. 由于常用的随机均匀初始化方法无法保证初始解在搜索空间内均匀分布,为此,本文利用正交试验设计进行种群的初始化工作.

将目标向量的各维分量视为试验指标,将决策变量的可行域视为试验区域,对决策变量的各维分量x_i进行离散,离散水平为Q. x_i及其离散水平序号$a_{j,i} \in \{1,2,\ldots,Q\}$可视为试验因素及相应的试验水平,由此将多目标优化问题看作等水平正交试验[23]. 接下来构建$L_M(Q^n) = [a_{j,i}]_{M\times n}$的正交矩阵并基于此生成初始种群. M为选取的试验组合数,满足$M = Q^J$; J为正整数,满足$(Q^J-1)/(Q-1) \geqslant n$. 种群初始化的具体步骤如下:

1) 令$J=2, N=(Q^2-1)/(Q-1)=Q+1$,选择适当的离散水平Q使得$Q+1 \geqslant n$,构造正交矩阵$L_M(Q^N)$;

2) 选择$L_M(Q^N)$矩阵的前n列组成正交矩阵$L_M(Q^n)$;

3) 基于$L_M(Q^n)$矩阵每行的试验组合数生成初代种群的个体.

3.2 基于树形结构的最优个体选择

算法通过TUA存储迭代过程中搜索到的非支配个体,具体操作流程见2.2.2节,此处介绍种群速度更新中的最优个体选择策略. 在多目标粒子群算法中,速度更新是种群粒子开展搜索的关键,要求为各粒子选择合适的个体最优粒子pbest以及全局最优粒子gbest. pbest用于引导局部搜索,其选取方式较为简单:开始时,将各粒子所对应的pbest初始化为当前位置,位置更新后,若当前粒子支配pbest,则将pbest更新为当前位置;若相互非支配,则以0.5的概率进行更新. 而gbest用于引导全局搜索,其选取方式较为复杂:为提高输出结果的质量,应优先勘探存档所对应的近似前沿的稀疏区域与边界区域,即优先选择稀疏区域与边界区域的个体作为gbest,以吸引种群粒子向该区域搜索. 个体是否位于边界可通过目标向量各维分量的大小比较进行标识,而个体是否位于稀疏区域则需利用密度指标进行标识. 指标计算的复杂度通常与目标向量维数及存档规模成正比,当采用无界存档处理较高维度的问题时,密度指标的计算成本将十分昂贵. TUA采用树形结构进行数据存储,树中各结点超格的建立实质上属于一种动态网格划分,已暗含密度信息,因此,无需额外引入指标进行计算.

图5为全局最优个体gbest的选取流程,将引导种群个体的gbest的候选集记为W_leader,存档中边际个体的集合标记为W_boundary. 每次迭代进行gbest选取前先更新W_leader集合,将原集合清空并将当前存档中的W_boundary直接加入;然后,选择存档中稀疏区域的个体加入集合,该操作记为F_t.LeaderUpdate(W_leader). 具体的步骤为:从树的根结点层开始,查看该层各结点的点集大小,若小于等于2,则将对应的点集直接并入W_leader;否则,将超格大小(超格最大最小顶点的间距)与点集规模的比值作为该结点的拥挤系数. 每次有个体加入时需检查W_leader的规模是否超过预设上限,若超过则

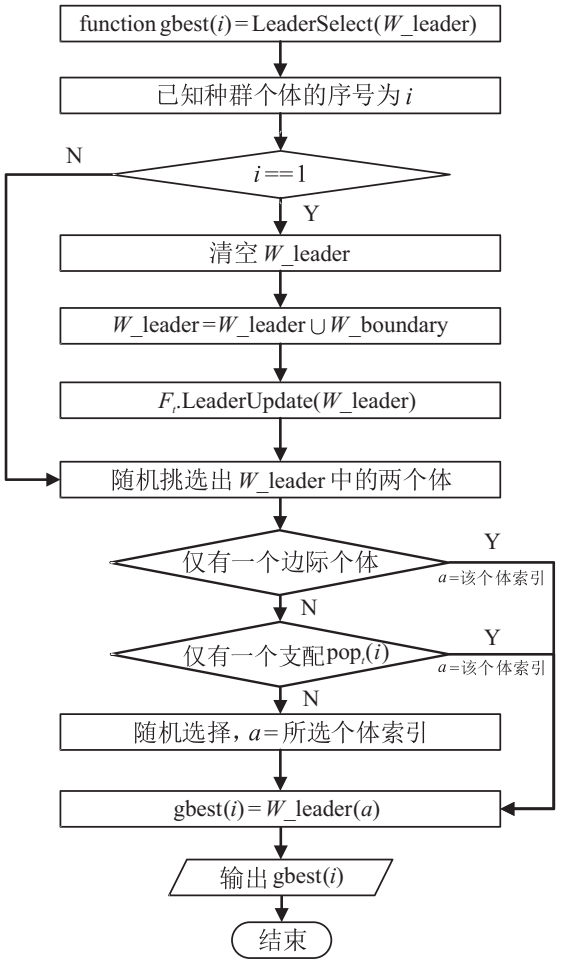


图5 全局最优个体选择流程

结束更新. 对该层的操作完成后若集合大小仍未满足要求,则将未执行并入操作的结点按拥挤系数降序排列并依次进行判断. 若当前结点为叶结点,则将点集直接并入W_leader;否则,浏览其下层子结点并重复上述操作.

W_leader 的更新完成后,通过二进制锦标赛从中选择个体作为种群粒子的gbest. 锦标赛的规则如下:首先以个体是否位于近似前沿的边界为获胜条件,目的是加强对边界区域的搜索以拓宽前沿范围;若无法判定,则以个体是否支配当前种群粒子为获胜条件,目的是加快粒子向前沿方向的移动;若仍无法判定,则随机选择.

4　实验与结果分析

4.1　实验设计

为检验所提出策略的效果与算法性能,选择目前应用广泛且目标空间能拓展至任意维度的DTLZ1 ~ 4[24]和WFG1~ 4[25]为测试函数,目标维数m取3 ~ 5. 对于DTLZ1令$|\boldsymbol{x}_m| = 5$,对于DTLZ2 ~ 4令$|\boldsymbol{x}_m| = 10$,对于WFG系列函数令$|\boldsymbol{x}_m| = m - 1, l = 10$. 按侧重点的不同进行分组实验,第1组实验和第2组实验检验改进策略以及策略组合使用的效果,第3组实验检验MOPSO/TUA较经典算法NSGA-II、 MOPSO以及近年提出的高性能算法NSLS[26]、pccsAMOPSO[27]的性能优劣,实验具体设计如下.

1) 以MOPSO为基础,先取消存档的规模限制,标记为MOPSO1,再用树形结构取代线性结构,标记为MOPSO2,将三者在不同测试函数上的计算结果进行比较.

2) 以MOPSO2为基础,先采用基于树形结构的最优个体选择取代原个体选择策略,标记为MOPSO3,再利用基于正交设计的种群初始化取代原初始化策略,此即为MOPSO/TUA,将三者在不同测试函数的计算结果进行比较.

3) 将MOPSO/TUA与NSGA-II、MOPSO、NSLS、pccsAMOPSO这4种算法在不同测试函数的计算结果进行比较.

实验的参数设置如下:种群规模设为$N = 400$,有界存档的容量设为$N_f = 200$,速度更新的参数设为$w = 0.4, r_1 = r_2 = 2$;MOPSO/TUA的参数设置为$\text{leafsize} = 50, \text{childsize} = m + 2, |W_\text{leader}| = 50$;实验3中4种对比算法的参数设置参照原文献. 为减少随机因素对结果的影响,每种算法在每个测试函数上均独立运行50次,最大迭代次数$T_{\max} = 5\,000$. 利用SPSS中的Wilcoxon符号秩检验(双边检验,显著性水平为0.05)对实验结果进行显著性分析.

4.2　评价指标

多目标进化算法的性能评价一般分为两方面:一是对算法时空复杂度进行评价;二是对算法求解结果的质量进行评价. 对于前者,本文利用计算时间t度量算法的时间复杂度,因空间复杂度并非本文的研究重点,故未对算法运行过程中的内存占用量进行监测;对于后者,本文选择二元超体积指标$\nu(A, B)$和间距指标SP对不同算法的求解性能进行对比分析,$\nu(A, B)$可评价存档集合所对应的近似前沿的收敛性与分布广泛性,SP可评价集合所对应的近似前沿的分布均匀性,两者的定义分别如下:

$$\nu(A, B) = \lambda(D_{A\cup B}(A) \backslash D_{A\cup B}(B)). \tag{10}$$

其中:A和B表示进行比较的两个解集,$D_{A\cup B}(A)$表示包含两集合目标向量集的最小超格中被A中个体支配的区域,$D_{A\cup B}(B)$表示包含两集合目标向量集的最小超格中被B中个体支配的区域,λ表示勒贝格测度,$\nu(A, B)$表示超格中被A而非B中的个体所支配的区域大小. 而

$$\text{SP} \triangleq \sqrt{\frac{1}{|A| - 1}\sum_{i=1}^{|A|}(\bar{d} - d_i)^2}, \tag{11}$$

$$d_i = \min_j \Big(\sum_{k=1}^{m} |f_k(\boldsymbol{x}_i) - f_k(\boldsymbol{x}_j)|\Big). \tag{12}$$

其中:A为待评价的非支配解集,d_i为目标空间中距i最近的集合A中个体与i的间距,$\bar{d}$为d_i的均值.

4.3　实验结果分析

1) 第1组实验.

表1给出了第1组实验的各指标统计结果. 其中第3、第4列为50次运行的MOPSO1与MOPSO最终输出结果间的二元超体积指标平均值,记为ν(M1, M)和ν(M, M1),加粗值表示对应列算法在对应行测试函数上50次运行结果的ν指标值,经Wilcoxon符号秩检验可知其显著高于对比算法的相应指标值. 可以看出,在19个测试函数上MOPSO1的输出结果在前沿收敛性与分布延展性上均显著优于MOPSO. 表1中第5、第6列为50次运行两种算法最终输出结果的间距指标平均值,记为SP(M1)和SP(M),加粗值表示对应列算法在对应行测试函数上50次运行结果的SP指标值,经Wilcoxon符号秩检验可知其显著低于对比算法的相应指标值. 可以看出,在6个测试函数上MOPSO1的输出结果在前沿分布均匀性上显著优于MOPSO,在4个测试函数上劣于MOPSO,而在其余测试函数上两者无显著差异. 由此可知,无界存档

表1 第1组实验的指标对比

function	m	ν(M1, M)/%	ν(M, M1)/%	SP(M1)	SP(M)	t(M2)	t(M1)
DTLZ1	3	**3.931**	2.016	2.351e-1	2.159e-1	17.58	16.38
	4	**5.305**	1.336	**1.327e-1**	1.783e-1	**98.77**	130.43
	5	**8.146**	0.407	1.279e-1	1.318e-1	**173.23**	271.64
DTLZ2	3	**4.820**	0.745	1.906e-2	2.389e-2	**18.46**	23.78
	4	**6.349**	0.249	3.458e-2	3.879e-2	**178.03**	283.51
	5	**9.305**	0.001	5.401e-2	5.262e-2	**224.55**	392.11
DTLZ3	3	**4.485**	1.834	2.028e-2	2.502e-2	**16.75**	18.83
	4	**5.401**	2.107	6.764e-2	6.988e-2	**132.80**	185.67
	5	**7.574**	0.846	**8.140e-2**	1.571e-1	**164.32**	262.17
DTLZ4	3	2.114	1.720	3.314e-2	3.658e-2	34.86	**30.22**
	4	**3.517**	2.049	5.968e-2	**3.531e-2**	**255.49**	368.14
	5	**6.703**	2.546	6.839e-1	6.441e-1	**356.62**	611.70
WFG1	3	2.777	2.832	4.891e-2	4.115e-2	**20.81**	24.93
	4	**6.475**	0.279	1.303e-1	1.574e-1	**76.11**	92.37
	5	**4.060**	2.592	7.015e-1	**5.917e-1**	**368.84**	525.41
WFG2	3	1.488	1.580	5.674e-2	5.931e-2	15.14	14.92
	4	1.831	1.617	6.143e-1	**4.812e-1**	**236.51**	346.28
	5	**5.394**	2.469	**2.152e0**	4.169e0	**262.41**	401.53
WFG3	3	**0.836**	0.447	3.833e-1	4.371e-1	**33.51**	38.18
	4	**1.162**	0.687	2.127e0	**8.214e-1**	**153.07**	185.76
	5	**3.234**	0.733	3.307e0	3.196e0	**382.11**	746.31
WFG4	3	0.700	0.689	**5.783e-2**	1.914e-1	**42.21**	56.04
	4	**2.655**	0.183	**9.503e-1**	2.516e0	**289.95**	459.51
	5	**3.094**	1.723	**2.254e0**	4.845e0	**344.76**	513.04

替代有界存档能显著提升结果质量,尤其是在收敛性与延展性这两个方面.

表1中第7、 第8列为50次运行MOPSO2与MOPSO1在存档更新上的耗时平均值,记为t(M2)和t(M1),加粗值表示对应列算法在对应行测试函数上50次运行的t指标值,经Wilcoxon符号秩检验可知其显著低于对比算法的相应指标值. 两种算法的区别在于存档的数据存储结构与更新方式不同,每代搜索得到的非支配解是相同的,因此,每次迭代所处理的存档更新问题是严格相同的. 比较更新耗时即是比较不同结构的存档更新效率. 可以看出,在3维DTLZ1和3维WFG2上两者耗时无显著差异,而且在3维DTLZ4上MOPSO2的耗时高于MOPSO1,原因在于在这些测试函数上算法计算中的存档规模较小,树形结构较线性结构在存档更新上的时间削减被其数据结构的维护耗时所补偿. 其余函数上MOPSO2的更新耗时均显著低于MOPSO1,且两者的时间差值随目标维数的增大而增大,说明树形结构较线性结构是更为高效的数据存储结构,尤其在处理高维多目标问题上.

2) 第2组实验.

表2给出了MOPSO3与MOPSO2各指标统计结果. 其中第3、第4列为50次运行的两种算法最终输出结果间的二元超体积指标平均值,标记为ν(M3, M2)和ν(M2, M3). 可以看出,MOPSO3在8个测试函数上显著优于MOPSO2,在5维DTLZ2和3维WFG4上劣于MOPSO2,在其余函数上两者无显著差异. 说明基于树形结构的个体选择能提升算法收敛性与分布广泛性,原因在于该策略能积极引导种群粒子向前沿方向与边界区域探索. 表2中第5、第6列为50次运行两种算法最终输出结果的间距指标平均值,记为SP(M3)和SP(M2). 可以看出,MOPSO3在5个测试函数上显著优于MOPSO2,在另外8个测试函数上劣于MOPSO2,在其余测试函数上两者无显著差异,说明改进策略未能改善前沿分布的均匀性. 表2中第7、第8列为50次运行两种算法在存档更新与个体选择上的耗时平均值,记为t(M3)和t(M2). 可以看出,在所有函数上MOPSO3的耗时均显著低于MOPSO2,原因在于改进策略使得算法无需计算复杂的密度指标,故降低了计算复杂度.

表 2　MOPSO3 与 MOPSO2 的指标对比

function	m	ν(M3, M2) / %	ν(M2, M3) / %	SP (M3)	SP (M2)	t(M3)	t(M2)
DTLZ1	3	0.934	0.721	2.008e-1	2.351e-1	**23.05**	42.01
	4	2.330	2.784	1.655e-1	**1.327e-1**	**138.81**	257.61
	5	**3.810**	0.580	2.070e-1	1.279e-1	**255.19**	393.95
DTLZ2	3	**1.130**	0.007	**2.597e-3**	1.906e-2	**39.76**	54.05
	4	**2.457**	0.234	**2.566e-2**	3.458e-2	**284.60**	450.47
	5	0.688	**0.981**	7.026e-2	5.401e-2	**348.54**	557.65
DTLZ3	3	2.607	1.937	2.669e-2	2.028e-2	**30.94**	42.49
	4	**3.529**	2.119	6.065e-2	6.764e-2	**207.59**	305.14
	5	1.193	1.354	1.052e-1	**8.140e-2**	**260.85**	487.5
DTLZ4	3	**2.998**	2.262	3.135e-2	3.314e-2	**41.82**	63.73
	4	0.715	0.686	5.667e-2	5.968e-2	**329.71**	545.90
	5	**5.912**	3.821	7.203e-1	6.839e-1	**438.52**	744.51
WFG1	3	1.220	0.723	6.021e-2	**4.891e-2**	**34.63**	48.57
	4	**4.767**	2.365	1.443e-1	1.303e-1	**111.76**	196.68
	5	0.063	0.069	**5.269e-1**	7.015e-1	**531.08**	943.32
WFG2	3	0.158	0.193	6.735e-2	**5.674e-2**	**22.53**	30.77
	4	**2.318**	0.904	6.507e-1	6.143e-1	**338.92**	594.25
	5	1.385	1.416	3.459e0	**2.152e0**	**430.16**	715.01
WFG3	3	3.983	4.047	4.618e-1	**3.833e-1**	**40.48**	57.49
	4	0.985	1.282	**7.401e-1**	2.127e0	**220.38**	323.62
	5	3.021	2.811	3.165e0	3.307e0	**543.88**	856.75
WFG4	3	0.057	**1.796**	7.690e-2	**5.783e-2**	**56.34**	85.79
	4	3.218	3.584	**8.236e-1**	9.503e-1	**413.25**	643.46
	5	0.578	0.461	4.814e0	**2.254e0**	**620.01**	941.63

表 3 给出了 MOPSO/TUA 与 MOPSO3 各指标统计结果. 其中第 3、第 4 列为 50 次运行的两种算法最终输出结果间的二元超体积指标平均值, 记为 ν(M/T, M3) 和 ν(M3, M/T). MOPSO/TUA 在 7 个测试函数上显著优于 MOPSO3, 在 4 维 DTLZ4 和 5 维 WFG1 上劣于 MOPSO3, 在其余函数上两者无显著差异. 表 3 中第 5、第 6 列为 50 次运行的两种算法最终输出结果的间距指标平均值, 记为 SP(M/T) 和 SP(M3). 可以看出, MOPSO/TUA 在 6 个测试函数上显著优于 MOPSO3, 在 4 个测试函数上劣于 MOPSO3, 在其余测试函数上无显著差异. 两种算法的区别在于种群的初始化方式不同, 由此可知, 基于正交设计的种群初始化可通过改善初始点集在决策空间中的分布来提高算法的寻优能力.

表 3　MOPSO/TUA 与 MOPSO3 的指标对比

function	m	ν(M/T,M3) / %	ν(M3,M/T) / %	SP(M/T)	SP(M3)
DTLZ1	3	**4.259**	2.609	2.172e-1	2.008e-1
	4	1.354	1.165	1.355e-1	1.655e-1
	5	**2.915**	0.475	2.028e-1	2.070e-1
DTLZ2	3	0.676	0.588	2.404e-3	2.597e-3
	4	3.708	3.978	**1.740e-2**	2.566e-2
	5	**3.098**	1.715	6.592e-2	7.026e-2
DTLZ3	3	**2.180**	1.435	**2.061e-2**	2.669e-2
	4	4.669	3.945	5.802e-2	6.065e-2
	5	2.235	2.950	**7.350e-2**	1.052e-1
DTLZ4	3	0.038	0.041	3.194e-2	3.135e-2
	4	1.745	**3.869**	8.169e-2	**5.667e-2**
	5	5.147	5.362	6.793e-1	7.203e-1
WFG1	3	**3.182**	1.710	**3.881e-2**	6.021e-2
	4	6.360	5.959	**9.728e-2**	1.443e-1
	5	0.614	**1.251**	5.012e-1	5.269e-1
WFG2	3	1.286	0.994	6.821e-2	6.735e-2
	4	1.227	1.695	8.048e-1	**6.507e-1**
	5	**2.576**	0.784	3.694e0	3.459e0
WFG3	3	6.425	6.146	4.343e-1	4.618e-1
	4	3.762	3.550	7.416e-1	7.401e-1
	5	2.895	2.836	**2.870e0**	3.165e0
WFG4	3	**6.021**	1.917	8.893e-2	**7.690e-2**
	4	3.818	3.025	7.852e-1	8.236e-1
	5	0.729	0.813	7.695e0	**4.814e0**

3) 第 3 组实验.

表 4 为 MOPSO/TUA 与 MOPSO、NSGA-II 的各指标统计结果. 可以看出: MOPSO/TUA 的 ν 指标值在 20 个测试函数上显著优于 MOPSO, 在 4 维 DTLZ4 上劣于 MOPSO, 在其余函数上无显著差异; SP 指标值在 13 个测试函数上显著优于 MOPSO, 在 6 个测试函数上劣于 MOPSO, 在其余函数上无显著差异. 由此可知, MOPSO/TUA 较 MOPSO 显著提升了多目标

表 4 MMOPSO/TUA与MOPSO及NSGA-II的指标对比

function	m	ν(M/T,M)/%	ν(M,M/T)/%	SP(M/T)	SP(M)	ν(M/T,N)/%	ν(N,M/T)/%	SP(M/T)	SP(N)
DTLZ1	3	**4.630**	0.945	2.172e-1	2.159e-1	1.495	**2.632**	2.172e-1	**1.121e-2**
	4	**3.048**	0.251	**1.355e-1**	1.783e-1	**5.743**	1.427	1.355e-1	**1.690e-2**
	5	**9.373**	0.004	2.028e-1	**1.318e-1**	**7.525**	0.000	**2.028e-1**	1.727e0
DTLZ2	3	**4.211**	0.033	**2.404e-3**	2.389e-2	**4.200**	0.024	**2.404e-3**	2.703e-2
	4	**5.978**	0.145	**1.740e-2**	3.879e-2	**8.389**	0.059	**1.740e-2**	6.365e-2
	5	**8.113**	0.000	6.592**e-2**	**5.262e-2**	**10.751**	0.367	**6.592e-2**	1.180e-1
DTLZ3	3	**4.669**	1.473	**2.061e-2**	2.502e-2	2.449	2.177	2.061e-2	**1.741e-2**
	4	**5.808**	0.283	**5.802e-2**	6.988e-2	**6.235**	3.067	5.802e-2	5.911e-2
	5	**5.387**	0.020	**7.350e-2**	1.571e-1	**11.172**	0.293	**7.350e-2**	2.134e0
DTLZ4	3	2.731	2.589	3.194e-2	3.658e-2	0.044	0.047	**3.194e-2**	4.387e-2
	4	1.973	**2.370**	8.169e-2	**3.531e-2**	**3.130**	0.984	**8.169e-2**	1.557e-1
	5	**4.539**	2.378	6.793e-1	6.441e-1	**6.770**	1.012	**6.793e-1**	3.301e0
WFG1	3	**1.163**	0.308	3.881e-2	4.115e-2	**2.475**	0.891	**3.881e-2**	5.752e-2
	4	**7.143**	1.392	**9.728e-2**	1.574e-1	**1.459**	0.082	**9.728e-2**	1.359e-1
	5	0.735	0.688	**5.012e-1**	5.917e-1	**2.386**	1.329	**5.012e-1**	7.750e0
WFG2	3	0.011	0.015	6.821e-2	**5.931e-2**	0.217	**3.226**	6.821e-2	**3.937e-2**
	4	**2.464**	1.437	8.048e-1	**4.812e-1**	**2.598**	1.372	8.048e-1	**7.116e-1**
	5	**9.863**	0.667	**3.694e0**	4.169e0	**0.329**	0.004	3.694e0	**1.034e0**
WFG3	3	**3.976**	3.092	4.343e-1	4.371e-1	0.146	**0.218**	**4.343e-1**	6.183e-1
	4	**1.756**	0.236	**7.416e-1**	8.214e-1	0.386	**2.095**	**7.416e-1**	2.173e0
	5	**4.176**	3.275	**2.870e0**	3.196e0	3.850	4.232	2.870e0	**2.383e0**
WFG4	3	**2.820**	0.032	**8.893e-2**	1.914e-1	6.293	5.710	**8.893e-2**	3.642e-1
	4	**5.051**	2.080	**7.852e-1**	2.516e0	**3.281**	1.066	7.852e-1	**6.483e-1**
	5	**1.313**	0.109	7.695e0	**4.845e0**	**9.631**	0.563	7.695e0	**5.999e0**

粒子群算法的寻优能力. MOPSO/TUA的ν指标值在16个测试函数上显著优于NSGA-II, 在4个测试函数上劣于NSGA-II; SP指标值在14个测试函数上显著优于NSGA-II, 在9个测试函数上劣于NSGA-II. 由此可知, MOPSO/TUA较NSGA-II能得出更高质量的Pareto近似前沿.

表5为MOPSO/TUA与NSLS、pccsAMOPSO的各指标统计结果. 可以看出: 在ν指标值上, MOPSO/TUA在15个测试函数上显著优于NSLS, 在8个测试函数上劣于NSLS, 在16个测试函数上显著优于pccsAMOPSO, 在6个测试函数上劣于pccsAMOPSO; 在SP指标值上, MOPSO/TUA在10个测试函数上显著优于NSLS, 在14个测试函数上劣于NSLS, 在11个测试函数上显著优于pccsAMOPSO, 在12个函数上劣于pccsAMOPSO. 由上述分析可知, MOPSO/TUA相较于对比算法NSLS和pccsAMOPSO有可观的性能表现, 尤其在前沿收敛性和分布延展性方面.

5 结 论

本文采用无界存档存储迭代过程中搜寻到的非支配解以规避存档规模设限所带来的弊端, 并构造出适合大规模存档存储与更新的树形结构, 由此提出了基于树形结构的无界存档. 在此基础上设计出一种新的多目标粒子群优化算法MOPSO/TUA, 此算法基于正交实验设计初始化外部种群, 使种群个体在决策空间内均匀分布, 利用树形结构实现了高效的存档更新与个体选择. 最后, 通过3组对比实验的结果表明了所提出策略能够明显改善算法性能, 算法MOPSO/TUA较经典算法的表现具有显著提升, 较近年提出的高性能算法具有明显的竞争力. MOPSO/TUA是一个整体性能更优、颇具发展前景的多目标优化方法, 探索其在水库多目标优化调度等实际工程问题上的应用将是下一步的研究方向.

表 5　MOPSO/TUA与NSLS及pccsAMOPSO的指标对比

function	m	ν(M/T, NS) / %	ν(NS, M/T) / %	SP(M/T)	SP(NS)	ν(M/T, pA) / %	ν(pA, M/T) / %	SP(M/T)	SP(pA)
	3	3.120	**4.769**	2.172e-1	**6.811e-2**	3.031	3.796	**2.172e-1**	3.464e-1
DTLZ1	4	0.078	**1.269**	**1.355e-1**	3.265e-1	**5.744**	4.892	**1.355e-1**	2.035e0
	5	**5.625**	3.096	**2.028e-1**	1.296e0	**7.413**	5.303	**2.028e-1**	5.129e0
	3	0.487	**0.913**	**2.404e-3**	5.992e-3	**4.433**	2.738	**2.404e-3**	1.382e-2
DTLZ2	4	**6.325**	4.381	1.740e-2	**8.691e-3**	0.097	**0.381**	**1.740e-2**	6.384e-2
	5	**0.605**	0.012	6.592e-2	**4.061e-2**	**1.969**	0.481	**6.592e-2**	3.590e-1
	3	2.450	**3.136**	**2.061e-2**	4.172e-2	**5.706**	2.018	2.061e-2	**9.182e-3**
DTLZ3	4	0.568	**1.647**	**5.802e-2**	8.145e-1	**3.012**	2.873	5.802e-2	**2.319e-2**
	5	**3.741**	2.982	**7.350e-2**	3.024e0	**4.740**	0.449	7.350e-2	**6.541e-2**
	3	1.324	**2.671**	3.194e-2	**2.145e-2**	2.689	**3.823**	**3.194e-2**	5.339e-2
DTLZ4	4	**0.417**	0.189	8.169e-2	**7.266e-2**	0.064	**2.953**	**8.169e-2**	1.306e-1
	5	**4.336**	0.992	**6.793e-1**	4.073e0	**4.034**	1.028	6.793e-1	**6.234e-1**
	3	**5.168**	4.473	3.881e-2	**2.089e-2**	0.048	**2.363**	**3.881e-2**	4.438e-2
WFG1	4	**1.385**	0.746	9.728e-2	**8.125e-2**	**2.846**	2.132	**9.728e-2**	6.810e-1
	5	**2.267**	0.098	5.012e-1	**4.261e-1**	2.983	**4.267**	**5.012e-1**	4.128e0
	3	2.258	**3.904**	**6.821e-2**	9.281e-2	**3.858**	0.589	6.821e-2	**5.721e-2**
WFG2	4	**5.731**	1.597	8.048e-1	**3.714e-1**	**5.747**	2.352	8.048e-1	**4.019e-1**
	5	**2.126**	0.059	3.694e0	**2.018e0**	**3.577**	0.100	3.694e0	**1.816e0**
	3	**1.836**	0.541	4.343e-1	**2.704e-1**	1.837	**6.171**	4.343e-1	**2.706e-1**
WFG3	4	2.956	2.542	7.416e-1	**6.962e-1**	**2.956**	0.175	7.416e-1	**6.748e-1**
	5	**3.594**	1.327	**2.870e0**	5.763e0	**1.093**	0.109	4.870e0	5.006e0
	3	2.871	**6.479**	8.893e-2	**4.709e-2**	**4.347**	0.057	8.893e-2	**6.225e-2**
WFG4	4	**1.285**	0.674	**7.852e-1**	9.336e-1	1.545	1.708	7.852e-1	**4.075e-1**
	5	**5.083**	4.731	7.695e0	**5.901e0**	**2.621**	0.254	7.695e0	**3.792e0**

参考文献(References)

[1] Coello C A C. Evolutionary multi-objective optimization: A historical view of the field[J]. IEEE Computational Intelligence Magazine, 2006, 1(1): 28-36.

[2] 韩玉艳, 李俊青, 桑红燕, 等. 离散NSGA-II求解带有限缓冲区的多目标批量流水线调度问题[J]. 聊城大学学报: 自然科学版, 2018, 31(1): 89-96.
(Han Y Y, Li J Q, Sang H Y, et al. Discrete NSGA-II for multi-objective lot-streaming flow shop scheduling problem with limited buffers[J]. Journal of Liaocheng University: Natural Science Edition, 2018, 31 (1): 89-96.)

[3] Han Y, Li Y Q, Gong D. Multi-objective migrating birds optimization algorithm for stochastic lot-streaming flow shop scheduling with blocking[J]. IEEE Access, 2019, 7: 5946-5962.

[4] 陈南祥, 李跃鹏, 徐晨光. 基于多目标遗传算法的水资源优化配置[J]. 水利学报, 2006, 37(3): 308-313.
(Chen N X, Li Y P, Xu C G. Optimal deployment of water resources based on multi-objective genetic algorithm[J]. Journal of Hydraulic Engineering, 2006, 37(3): 308-313.)

[5] Sivasubramani S, Swarup K S. Multi-objective harmony search algorithm for optimal power flow problem[J]. Electrical Power and Energy Systems, 2011, 33(3): 745-752.

[6] 公茂果, 焦李成, 杨咚咚, 等. 进化多目标优化算法研究[J]. 软件学报, 2009, 20(2): 271-289.
(Gong M G, Jiao L C, Yang D D, et al. Research on evolutionary multi-objective optimization algorithms[J]. Journal of Software, 2009, 20(2): 271-289.)

[7] 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.

[8] Zitzler E, Laumanns M, Thiele L. SPEA2: Improving the strength Pareto evolutionary algorithm[R]. ETH Zurich: Computer Engineering and Networks Laboratory, 2001: 1-21.

[9] Corne D, Jerram N, Knowles J, et al. PESA-II: Region-based selection in evolutionary multiobjective optimization[C]. Proceedings of the 3rd Annual

Conference on Genetic and Evolutionary Computation. San Francisco: Morgan Kaufmann Publishers Inc, 2001: 283-290.

[10] Rudolph G. Convergence properties of some multi-objective evolutionary algorithm[C]. Proceedings of the 2000 Congress on Evolutionary Computation. La Jolla: IEEE, 2000: 1010-1016.

[11] Fieldsend J E, Everson R M, Singh S. Using unconstrained elite archives for multi-objective optimization[J]. IEEE Transactions on Evolutionary Computation, 2003, 7(3): 305-323.

[12] 石川, 李清勇, 史忠植. 一种快速的基于占优树的多目标进化算法[J]. 软件学报, 2007, 18(3): 505-516.
(Shi C, Li Q Y, Shi Z Z. A quick multi-objective evolutionary algorithm based on dominating tree[J]. Journal of Software, 2007, 18(3): 505-516.)

[13] 纪昌明, 张培, 苏阳悦, 等. 理想均变率法及其在水库群多目标调度决策中的应用[J]. 水力发电学报, 2017, 36(12): 1-9.
(Ji C M, Zhang P, Su Y Y, et al. Ideal mean rate method and its application to multi-objective reservoir operation decision making[J]. Journal of Hydroelectric Engineering, 2017, 36(12): 1-9.)

[14] 汪勇, 程姣, 高娜, 等. 基于初集排序的Pareto非支配解集构造算法[J]. 系统工程理论与实践, 2018, 38(4): 960-970.
(Wang Y, Cheng J, Gao N, et al. A construction algorithm of Pareto non-dominated solution set based on initial set sorting[J]. Systems Engineering — Theory & Practice, 2018, 38(4): 960-970.)

[15] 耿焕同, 陈哲, 陈正鹏, 等. 一种基于群体分布特征的自适应多目标粒子群优化算法[J]. 控制与决策, 2017, 32(8): 1386-1394.
(Geng H T, Chen Z, Chen Z P. A self-adaptive multi-objective particle swarm optimization algorithm based on swarm distribution characteristic[J]. Control and Decision, 2017, 32(8): 1386-1394.)

[16] 杨景明, 侯新培, 崔慧慧, 等. 基于融合多策略改进的多目标粒子群优化算法[J]. 控制与决策, 2018, 33(2): 226-234.
(Yang J M, Hou X P, Cui H H, et al. Improved multi-objective particle swarm optimization algorithm based on integrating multiply strategies[J]. Control and Decision, 2018, 33(2): 226-234.)

[17] 公茂果, 程刚, 焦李成, 等. 基于自适应划分的进化多目标优化非支配个体选择策略[J]. 计算机研究与发展, 2011, 48(4): 545-557.
(Gong M G, Cheng G, Jiao L C, et al. Nondominated individual selection strategy based on adaptive partition for evolutionary multi-objective optimization[J]. Journal of Computer Research and Development, 2011, 48(4): 545-557.)

[18] Wang S, Ali S, Yue T, et al. Integrating weight assignment strategies with NSGA-II for supporting user preference multi-objective optimization[J]. IEEE Transactions on Evolutionary Computation, 2018, 22(3): 378-393.

[19] Drozdik M, Akimoto Y, Aguirre H. Computational cost reduction of nondominated sorting using the m-front[J]. IEEE Transactions on Evolutionary Computation, 2015, 19(5): 659-678.

[20] Reyes-Sierra M, Coello Coello C A. Multi-objective particle swarm optimizers: A survey of the state-of-the-art[J]. International Journal of Computational Intelligence Research, 2006, 2(3): 287-308.

[21] 胡旺, Yen G G, 张鑫. 基于Pareto熵的多目标粒子群优化算法[J]. 软件学报, 2014, 25(5): 1025-1050.
(Hu W, Yen G G, Zhang X. Multi-objective particle swarm optimization based on pareto entropy[J]. Journal of Software, 2014, 25(5): 1025-1050.)

[22] Coello C A, Pulido G T. Handling multiple objectives with particle swarm optimization[J]. IEEE Transactions on Evolutionary Computation, 2004, 8(3): 256-279.

[23] Wang Y P, Dang C, Li H C. A clustering multi-objective evolutionary algorithm based on orthogonal and uniform design[C]. 2009 IEEE Congress on Evolutionary Computation. Trondheim: IEEE, 2009: 2927-2933.

[24] Deb K, Thiele L, Laumanns M. Scalable multi-objective optimization test problems[C]. Proceedings of the 2002 Congress on Evolutionary Computation. Honolulu: IEEE, 2002: 825-830.

[25] Huband S, Hingston P. A review of multiobjective test problems and a scalable test problem toolkit[J]. IEEE Transactions on Evolutionary Computation, 2006, 10(5): 477-506.

[26] Chen B, Zeng W H, Lin B. A new local search-based multiobjective optimization algorithm[J]. IEEE Transactions on Evolutionary Computation, 2015, 19(1): 50-73.

[27] Hu W, Yen G G. Adaptive multiobjective particle swarm optimization based on parallel cell coordinate system[J]. IEEE Transactions on Evolutionary Computation, 2015, 19(1): 1-18.

作者简介

纪昌明(1956–), 男, 教授, 博士生导师, 从事水文水资源等研究, E-mail: cmji@ncepu.edu.cn;

马皓宇(1995–), 男, 博士生, 从事多目标决策与水库调度的研究, E-mail: 940467366@qq.com;

李宁宁(1994–), 女, 博士生, 从事水资源与能源科学、风险管理与决策的研究, E-mail: 465956539@qq.com;

吴嘉杰(1992–), 男, 博士生, 从事水资源与能源科学、风险管理与决策的研究, E-mail: 1014640449@qq.com;

彭杨(1975–), 女, 教授, 博士生导师, 从事水库(群)水沙联合优化调度、水沙运动模拟技术、多目标决策、水资源系统风险分析等研究, E-mail: pengyang@ncepu.edu.cn;

王丽萍(1956–), 女, 教授, 博士生导师, 从事水资源系统工程、水电能源系统规划与管理、水资源优化配置与应用、水资源经济与能源经济等研究; E-mail: lqwang@ncepu.edu.cn.

(责任编辑: 李君玲)