



基于多种群分解预测的动态多目标引力搜索算法

刁鹏飞, 李树森, 姜雪松

引用本文:

刁鹏飞, 李树森, 姜雪松. 基于多种群分解预测的动态多目标引力搜索算法[J]. 控制与决策, 2021, 36(12): 2910–2918.

在线阅读 View online: <https://doi.org/10.13195/j.kzyjc.2020.1002>

您可能感兴趣的其他文章

Articles you may be interested in

基于弱关联的自适应高维多目标进化算法

A weak association-based adaptive evolutionary algorithm for many-objective optimization

控制与决策. 2021, 36(8): 1804–1814 <https://doi.org/10.13195/j.kzyjc.2019.1723>

基于向量角分解的高维多目标进化算法

Many-objective evolutionary algorithm based on vector angle decomposition

控制与决策. 2021, 36(3): 761–768 <https://doi.org/10.13195/j.kzyjc.2019.0925>

基于分解的多目标多因子进化算法

A multiobjective multifactorial evolutionary algorithm based on decomposition

控制与决策. 2021, 36(3): 637–644 <https://doi.org/10.13195/j.kzyjc.2019.0525>

基于双种群模糊引力搜索算法的舰载机甲板作业调度

Flight deck operations scheduling based on dual population fuzzy gravitational search algorithm

控制与决策. 2021, 36(11): 2751–2759 <https://doi.org/10.13195/j.kzyjc.2020.0523>

一种求解约束多目标问题的协作进化算法

A collaborative evolutionary algorithm for solving constrained multi-objective problems

控制与决策. 2021, 36(11): 2656–2664 <https://doi.org/10.13195/j.kzyjc.2020.0791>

基于多种群分解预测的动态多目标引力搜索算法

刁鹏飞^{1,3†}, 李树森², 姜雪松¹

(1. 东北林业大学 工程技术学院, 哈尔滨 150001; 2. 东北林业大学 机电工程学院, 哈尔滨 150001;
3. 江苏省金象传动设备股份有限公司, 江苏 淮安 223001)

摘要: 为提高算法求解动态多目标问题的寻优性能, 提出一种多种群分解预测动态多目标算法. 首先, 提出进化向量生成策略, 即基于偏好目标的解生成一组均匀分布的平行向量, 并采用引力搜索算法优化每个子问题, 保证其对应解的精度和分布的均匀性; 其次, 设计插值生成策略, 即根据进化向量子问题的解在目标空间中的取值, 通过线性插值的方式生成更多非支配解, 保证解集的多样性和均匀性; 再次, 在环境变化后, 根据相邻子问题的解存在相近性预测生成搜索种群, 提高算法的寻优速度. 与 5 个对比算法在 10 个标准动态测试函数上进行对比分析, 实验结果表明采用所提出算法求解动态多目标问题具有较好的分布性和收敛性.

关键词: 动态多目标; 基于分解技术的多目标算法; 多种群预测; 引力搜索算法

中图分类号: TP301.6

文献标志码: A

DOI: 10.13195/j.kzyjc.2020.1002

开放科学(资源服务)标识码(OSID):



引用格式: 刁鹏飞, 李树森, 姜雪松. 基于多种群分解预测的动态多目标引力搜索算法[J]. 控制与决策, 2021, 36(12): 2910-2918.

Dynamic multi-objective gravitational searching algorithm based on multi-population decomposition prediction

DIAO Peng-fei^{1,3†}, LI Shu-sen², JIANG Xue-song¹

(1. College of Engineering and Technology, Northeast Forestry University, Harbin 150001, China; 2. College of Mechanic and Electronic Engineering, Northeast Forestry University, Harbin 150001, China; 3. Jiangsu Province Jinxiang Transmission Equipment Co. Ltd, Huai'an 223001, China)

Abstract: In order to improve the non-dominant solution set with better convergence and distributivity of dynamic multi-objective problems, a multi-population decomposition prediction algorithm is proposed. Firstly, an evolutionary vector adaptive generation strategy is proposed, which generates a set of uniform evolutionary vectors based on the solutions of preference sub-problems to ensure the convergence and distribution of the Pareto set. Then, the new non-dominant solution is obtained based on the location of the solution of the subproblem in the target space. Thirdly, a predictive model is adopted to initialize the subpopulation to improve the optimum speed and performance of the algorithm. The experimental results show that compared with five existing algorithms, the proposed algorithm has obvious advantages in convergence and distribution over ten standard dynamic multi-objective problems..

Keywords: dynamic multi-objective; the multi-objective algorithm based on decomposition; multiple population prediction; gravitational search algorithm

0 引言

在实际生产和生活中存在着许多复杂的优化问题, 如生产调度问题^[1-2]、资源分配问题^[3-4]、网络路由优化问题^[5-6]. 当优化问题随时间动态改变时, 可将其称为动态多目标优化问题(dynamic multi-objective optimization problems, DMOPs). 由于该类问题的 Pareto 解集(Pareto set, PS)和 Pareto 前沿(Pareto

front, PF)都可能随时间变化, 虽然静态多目标算法发展较为成熟, 但在求解 DMOPs 时, 环境变化导致的种群多样性下降问题将影响算法获得解集的质量.

鉴于此, 有学者从提高种群多样性的角度设计相应策略, 如 Deb 等^[7]采用基于最优解变异的策略, 在环境变化后重新初始化种群, 但当环境变化较快时, 求得的解集收敛性和分布性距理想 PS 仍有一定

收稿日期: 2020-07-20; 修回日期: 2020-09-27.

基金项目: 中央高校基本科研业务费专项资金项目(2572018BL04).

责任编辑: 巩敦卫.

[†]通讯作者. E-mail: 398317196@qq.com.

差距. 考虑到环境变化前后, 待优化问题不是完全盲目的随机变化, 越来越多的学者通过建立预测模型的方式提高算法的求解性能^[8-9]. 代表性算法如Zhou等^[10]提出了基于种群预测的动态多目标优化算法, 根据历史最优解建立对未来解集的线性预测模型, 但仅在待优化问题随时间变化规律性较强时才能展现较好效果, 且需要较长的模型训练周期. 张世文等^[11]提出基于生态策略的动态多目标优化算法, 通过对进化种群和历史最优解执行变异策略, 实现环境变化后提高种群进化质量的目的, 但由于变异策略存在一定的盲目性, 不能有效缩小种群寻优空间. 刘若辰等^[12]提出了一种基于预测策略的动态多目标免疫优化算法, 利用历史非支配解集建立对当前时刻最优解集的预测模型, 并根据预测结果初始化种群, 但算法在求解非周期性变化的问题时, 所得解集的收敛性和分布性有待提高. 郑金华等^[13]通过记录环境变化初始时以及种群自主进化一段时间后种群聚类中心点位置的前后变化, 预测当前时刻最优解集所在方向, 从而实现对新环境下理想解集的预测, 若种群不完全收敛, 则会增大对聚类中心变化预测的误差. 丁进良等^[14]对关联到相同参考点的个体建立时间序列, 一方面对这些时间序列通过线性回归模型预测新环境下种群, 另一方面将历史时刻的预测误差反馈到当前预测中以提高预测的准确性, 从而加快算法在新环境下的收敛速度. 陈美蓉等^[15]根据 Pareto 较优解可以以一定稳定性阈值逼近未来多个连续动态环境下的真实前沿, 从而直接作为这些环境下的 Pareto 解集, 引入基于分解的多目标进化优化方法和无惩罚约束处理方法, 构建了动态多目标分解鲁棒进化优化方法, 但该算法中的参数较难根据不同的优化问题加以确定. Ruan等^[16]提出了基于预测及多样性保持的混合模型, 通过逐步搜索策略使种群快速收敛, 采用随机多样性保持策略提高和增强种群的多样性, 但有限的计算资源使得算法在前期环境变化中所得解集的质量有待于提高. 刁鹏飞等^[17]提出了基于分解技术的动态多目标引力搜索算法, 基于相邻权重在前后代对应子问题的相近性以及同一权重在前后代对应子问题的相近性初始化种群, 但没有考虑 DMOPs 发生平移、拉伸等形变情况导致的解集预测分布不均匀问题. Liu等^[18]提出了基于多种群的动态多目标粒子群算法, 首先采用粒子群算法优化每个子目标, 再基于得到的解初始化优化其他子问题的种群, 在环境变化后基于历史解建立对解集中心点和形状的预测模型, 虽然取得了较好的效果, 但忽略了不同子目标对

应决策变量的差异性, 且建立的预测模型需要较长时间的积累. 耿焕同等^[19]提出了一种基于分解的预测型动态多目标粒子群优化算法, 将多目标问题分解为多个不同子问题, 并提出一种群体预测策略, 根据同一方向上的历史解建立时间序列分析模型, 预测未来最优解的变化, 但当 PF 的形状随时间出现拉伸变化时, 采用固定的进化向量将导致解集分布不均匀. Rong等^[20]提出了一种多模型预测方法, 从 PS 的不同变化情况出发设计对应的种群初始化策略, 但该方法在判断 PS 变换类型时容易出现误判.

基于以上分析, 为有效求解具有不同变化特征的 DMOPs, 本文在以往研究的基础上, 提出一种基于多种群分解预测的动态多目标算法 (dynamic multi-objective gravitational searching algorithm based on multi-population decomposition prediction, MDP-DMGSA). 在环境变化前, 首先基于偏好目标的解生成一组与 PF 交点均匀分布在 PF 上的进化向量; 在优化每个进化向量子问题时, 基于已寻进化向量子问题的解初始化种群, 采用插值生成策略得到更多非支配解; 在环境变化后, 基于相似进化向量对应解的相近性以及相邻进化子问题对应解具有相近性初始化种群, 提高算法寻优效率. 实验结果表明, 采用所提出算法求解动态多目标问题具有较好的分布性和收敛性.

1 基本理论

动态多目标优化问题是指目标函数或相关参数随时间变化的一类优化问题. 以最小化问题为例, 其数学模型^[13]为

$$\begin{aligned} \min_{x \in \Omega} F(x, t) &= (f_1(x, t), f_2(x, t), \dots, f_m(x, t))^T; \\ g_i(x, t) &\leq 0, \quad i = 1, 2, \dots, p; \\ h_j(x, t) &= 0, \quad j = 1, 2, \dots, q. \end{aligned} \quad (1)$$

其中: $t = 0, 1, \dots, t \in T$ 为时间变量; $x = (x_1, x_2, \dots, x_n)$ 为 n 维决策变量, 其定义域为 Ω ; $F = (f_1, f_2, \dots, f_m)$ 为 m 维最小化问题的目标向量; $g_i \leq 0$ ($i = 1, 2, \dots, p$) 为 p 个不等式约束; $h_j = 0$ ($j = 1, 2, \dots, q$) 为 q 个等式约束.

定义 1 (Pareto) 支配对种群中任意两个个体 q 和 p , 如果对目标 i 有 $f_i(p) \leq f_i(q)$, $\forall i = \{1, 2, \dots, M\}$ 且 $\exists j \in \{1, 2, \dots, M\}$ 有 $f_j(p) < f_j(q)$, 则称 p 支配 q , 表示为 $f_j(p) \prec f_j(q)$.

定义 2 (Pareto 最优前沿) 设 x 为决策变量, F 为目标函数, 则 PF 定义描述为

$$PF := \{y = F(x) | x \in PS\}.$$

定义 3 (Pareto 最优解集) 设 x 为决策变量, Ω 为

决策空间, F 为目标函数, 则PS定义为

$$PS := \{x \in \Omega | \exists x^* \in \Omega, F(x^*) \prec F(x)\}.$$

2 基于多种群分解预测的动态多目标算法

由于DMOPs的PF和PS都可能随时间变化, 求解算法要在环境变化前获得收敛性和分布性更优的解集, 又要在环境变化后基于历史进化信息保证非支配解集的质量. 本文在文献[17]的基础上, 采用分解多目标算法作为多目标优化框架, 采用引力搜索算法^[21]作为种群搜索的进化策略, 设计子进化向量的生成方式、插值生成策略和解集预测策略. 下面分别从环境变化前和变化后具体阐述.

2.1 环境变化前

2.1.1 权重自适应生成策略

分解多目标算法^[22]通过一组权重向量将多目标优化问题分解为多个单目标子问题, 这种将待优化问题降维处理的方式能够极大地降低求解难度. 然而, 权重向量的设计将影响子种群的搜索方向, 进而影响解集的分布性. 由于分解多目标算法权重向量是基于理想PF分布均匀得到的, 在面对实际PF为非均匀的情况时, 会出现实际解分布不均的现象. 如图1所示, 当理想PF呈凹凸形状时, 进化向量与PF的交点在PF上的分布不均.

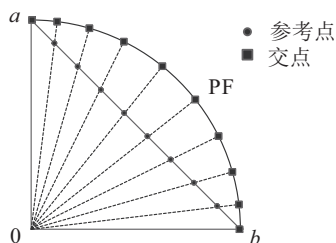


图1 原有进化向量的生成

为使权重向量与实际前沿面的交点在前沿面上分布得更均匀, 本文根据理想解之间的互不支配性以及偏好子目标的解在目标空间的分布情况, 生成子种群进化向量, 核心思想如图2所示. 由图2可见, 一组平行的进化向量与PF的交点更均匀地分布在PF上.

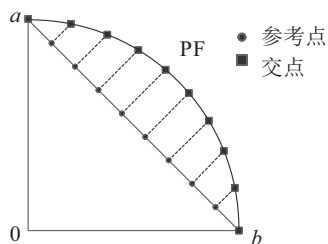


图2 改进进化向量生成

针对二目标动态问题和三目标动态问题的求解展开研究, 下面具体介绍进化向量的产生方式. 对于

二目标问题, 在算法求解DMOPs时, 首先采用引力搜索算法优化偏好于不同目标的子问题, 具体为

$$F = \min(\max(l_{i1} \times f_1, l_{i2} \times f_2)), \quad (2)$$

其中 l_{i1} 和 l_{i2} 分别为偏好目标 f_1 和 f_2 的权值. 当优化得到偏好于不同目标函数的解时, 将其对应于目标空间中的位置作为边界点, 并以一定的斜率得到一组间距相等的平行向量, 即待优化子问题的进化向量. 设进化向量斜率为 k , 对于侧重不同目标的子问题, 优化得到目标值分别为 (a_1, b_1) 和 (a_2, b_2) , 偏好子问题进化向量可表示为

$$y_1 = kx + b_1 - a_1k, \quad (3)$$

$$y_2 = kx + b_2 - a_2k. \quad (4)$$

根据几何关系在两条平行线间生成平行且间距相等的一组向量, 表示为

$$y_i = kx + m, \quad (5)$$

$$m = |b_1 - a_1k| - \frac{i}{l} \times |b_1 - a_1k - b_2 + a_2k|. \quad (6)$$

其中: $A(a_1, b_1)$ 和 $B(a_2, b_2)$ 分别为优化偏好子问题的解在目标空间的值; l 为生成的进化向量数目; i 为第 i 个进化向量, $i \in p$. 进化向量对应子问题的解在两目标函数上的差值为 m , 取进化向量斜率 k 为 1, 则式(7)可表示为进化向量对应的子问题

$$F = \min(\max(f_1 + m, f_2)). \quad (7)$$

根据 m 的不同取值, 采用引力搜索算法优化其最优解便可得到当前进化向量的最优解. 多个平行的进化向量子问题相当于对 f_1 和 f_2 之间的冲突性增加了惩罚值, 根据惩罚值的不同, 算法会得到不同的非支配解.

对于三目标问题, 首先优化得到分别偏好于 3 个子目标的解, 其对应目标值分别为 $A(a_1, b_1, c_1)$ 、 $B(a_2, b_2, c_2)$ 、 $C(a_3, b_3, c_3)$, 方向向量为 $(1, 1, 1)$, 利用几何知识寻找一组方向向量为 $(1, 1, 1)$, 与平面 ABC 的交点均匀地分布在三角形 ABC 上的一组向量, 交点选取如图3所示. 即首先将三角形按照两腰长等分成 4 段, 再在每条线段上得到间距相等的点, 基于方向向量和选取的交点即可得到进化向量. 引力搜索算法按照选取的交点依次优化每个进化向量子问题.

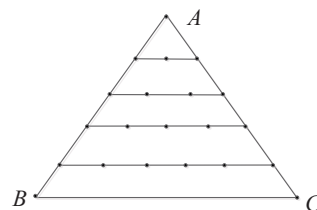


图3 进化向量与PF交点的选取

2.1.2 子种群初始化

生成进化向量后,优化每个进化向量子问题,由于相邻子问题对应的解具有相近性,根据已寻解初始化种群,能够减小算法的寻优范围,提高进化算法的求解速度. 子种群初始化如下所示:

$$X(o+1) = X(o) + (X(o) - X(o-1)) \times (1+r). \quad (8)$$

其中: $X(o+1)$ 为第 $o+1$ 个进化向量子问题的预测解; $X(o)$ 为第 o 个进化向量子问题的最优解; r 为区间 $[-0.5, 0.5]$ 之间的随机数. 根据相邻子问题的解以及已寻相邻子问题解的变化趋势完成种群初始化,能够缩小种群对当前子问题的求解空间,提高种群寻优速度和精度.

2.1.3 插值生成策略

通过上一节的寻优可以得到一组分布较均匀且收敛性较好的非支配解集,但由于 DMOPs 随时间动态变化,只能优化有限的进化向量子问题,为提高解集多样性并保持解集分布均匀性,根据已寻子问题最优解在目标空间上的分布情况,通过插值的方式得到更多非支配解,再根据其在目标空间上的取值保留非支配解. 具体过程描述如下.

对已获得的 l 个进化向量子问题对应的解,按其在目标空间中的位置进行排序. 设 l 个向量对应子问题的目标矩阵为 M , 计算得到目标点依次相连的长度,对任意两个相邻的目标点 l_o 和 l_{o+1} , 其距离为 $D_{o,o+1}$, 若此时需要再得到 num 个解, 则对于 l_o 与 l_{o+1} 这两个点之间,通过插值得到的数量如下所示:

$$\text{num}_{o,o+1} = \left[D_{o,o+1} / \sum_{o=1}^{l-1} D_{o,o+1} \times \text{num} \right], \quad (9)$$

其中: $\text{num}_{o,o+1}$ 为进化向量 o 与 $o+1$ 之间插值生成解的数值. 在任意两个相邻子问题解之间,通过插值得到的解可表示为

$$X_i = \frac{X(o) + X(o+1)}{\text{num}_{o,o+1}} \times i. \quad (10)$$

其中: i 为当前生成的第 i 个解, $i \in (0, \text{num}_{o,o+1})$; X_i 为在解 $X(o)$ 与 $X(o+1)$ 之间插值得到的第 i 个解. 对于三目标问题,有

$$\text{num}_{A'B'C'} = S_{A'B'C'} / S_{ABC} \times \text{num}. \quad (11)$$

其中: $S_{A'B'C'}$ 为相邻点 A' 、 B' 、 C' 构成三角形的面积; S_{ABC} 为空间任意相邻3个点构成三角形面积的累积和; $\text{num}_{A'B'C'}$ 为在相邻点 A' 、 B' 、 C' 插值生成点的数目. 按照图3所示的方式在每个三角形中插值得到更多点.

基于已寻解在目标空间的位置生成更多非支配

解,能够使算法求解 PF 发生平移、拉伸、凹凸性等变化的问题时,在提高解集多样性的同时,更好地保持解集分布的均匀性.

2.2 环境变化后

在环境变化后,首先优化侧重于不同目标的子问题,此时为加快种群的寻优速度,节省计算资源,将根据相邻进化向量对应子问题的解具有相近性完成种群初始化. 若 $t < 2$, 此时可参考的历史信息较少,则基于相邻进化向量对应子问题的解具有相近性初始化种群,有

$$X_t = X_{t-1} + r_1. \quad (12)$$

若 $t \geq 2$, 则根据前两次环境变化下偏好子问题对应解的变化趋势预测当前子问题解的区域,有

$$X_t = X_{t-1} + (X_{t-1} - X_{t-2}) \times r_2. \quad (13)$$

其中: $x_{\max}$ 和 $x_{\min}$ 分别为 X_{t-1} 各维度变量的上界和下界; r_1 为区间 $\left(-\frac{x_{\max} - x_{\min}}{10}, \frac{x_{\max} - x_{\min}}{10} \right)$ 内的随机数; r_2 为区间 $[-0.5, 0.5]$ 之间的随机数; X_t 为第 t 个环境下偏好子问题的预测解; X_{t-1} 为第 $t-1$ 次环境下偏好子问题的最优解.

对于其他进化向量子问题,根据相邻进化向量子问题的已寻解,预测生成待优化问题的初始种群^[17]. 具体计算过程如下所示:

$$x_{ij}^{t+1}(o+1) = x_{ij}^{t+1}(o) + (x_j^t(o+1) - x_j^t(o)) \times r. \quad (14)$$

其中: $r \in (-0.5, 0.5)$; $x_{ij}^{t+1}(o+1)$ 为第 $t+1$ 环境下第 $o+1$ 个进化向量子问题,种群粒子 i 在第 j 维上的初始值; $x_j^t(o)$ 为第 t 个环境下第 o 个向量子问题的解在第 j 维上的值.

2.3 实现流程

step 1: 在第0次环境变化下,根据待优化问题的目标数目,产生种群规模为 N_1 、迭代次数为 K_1 的种群,并采用引力搜索算法优化各个偏好子问题.

step 2: 根据偏好子问题的解,生成1个进化向量,对于每个进化向量子问题,通过式(8)得到规模为 N_2 的初始种群,设置迭代次数为 K_2 ,并采用引力搜索算法求解.

step 3: 对于优化得到的 l 个子问题的解,根据其在目标空间上的坐标,得到总距离 $\sum_{o=1}^{l-1} D_{o,o+1}$ 和任意相邻两点的距离 $D_{o,o+1}$,再根据待求插值解的数量 num 及目标数得到更多非支配解.

step 4: 在环境变化后,若此时是第 l 次环境变化,则采用式(12)初始化种群,若环境变化次数大于1,则

采用式(13)初始化种群,生成规模为 N_3 的初始种群,并采用引力搜索算法迭代 K_3 次,得到偏好目标较优解。

step 5: 根据偏好目标的解得到1个进化向量,对于每个进化向量子问题,由式(14)生成规模为 N_4 的初始种群,并采用引力搜索算法迭代 K_4 获得进化子问题较优解。

step 6: 采用插值策略生成并保留非支配解。

step 7: 若满足环境变化停止条件,则终止算法,否则转至step 4。

3 实验仿真与分析

为验证本文算法求解 DMOPs 的有效性和先进性,选取具有代表性且优化效果较好的 DMS^[16]、PPS^[10]、MDMOA/D^[17]、FPS^[23]、EGS^[24]作为对比算法。测试函数有 FDA1-FDA4^[25]、DMOP1-DMOP3^[26]、F5-F7^[10]等10个问题。所有实验在硬件配置为 Intel[®] Pentium[®] CPU G620@ 2.60 GHz、4.00 G 内存、2.60 GHz 主频的计算机上进行,开发环境为 Matlab R2014a。

3.1 参数设置

为了保证各算法在第0时刻的收敛性,在有关二目标问题的实验中,设置0时刻的函数评价次数为5000,其余时刻函数评价次数为3000;对于三目标问题,第0时刻的函数评价次数为10000,其余时刻为6000,环境变化次数为120。对比算法的种群规模设置为100,对于每个测试问题,各算法独立运行20次; n_T 取值10, τ_T 取值30。

本文算法中,设置引力搜索算法的引力常量 G_0 为5,在第0次环境变化,对偏好子问题的种群规模 N_1 设置为15,迭代次数 K_1 为100;对于其他进化量子问题,种群规模 N_2 为5,迭代次数 K_2 为50。在环境变化后,偏好子问题的种群规模 N_3 为10,迭代次数 K_3 为50,其余子问题的种群规模 N_4 为5,迭代次数 K_4 为50。根据每轮环境变化设定的函数评价次数以及子种群规模,设置二目标问题进化向量数量为8,三目标为18;对于偏好权重,为使进化策略既能得到偏好目标的较优解,又能兼顾其他非偏好目标解的质量,二目标问题的偏好权重设置为(0.99, 0.01)和(0.01, 0.99),三目标问题的权重设置为(0.99, 0.05, 0.05)、(0.05, 0.99, 0.05)、(0.05, 0.05, 0.99)。

3.2 性能评价指标

为对比本文算法的有效性,采用平均反向世代距离 $\overline{IGD}$ ^[10]和平均间距指标 $\overline{SP}$ ^[27]评价算法的性能,具体表示为

$$\overline{IGD} = \frac{1}{T} \sum_{t \in T} \frac{\sum_{v \in PF(t)^*} d(v, PF(t))}{|PF(t)^*|}, \quad (15)$$

$$\overline{SP} = \frac{1}{T} \sum_{t \in T} \sqrt{\frac{1}{N-1} \sum_{i=1}^N (\overline{d(t)} - d_i(t))^2}. \quad (16)$$

其中: T 为动态多目标问题的变化次数; $PF(t)^*$ 为待优化问题的理想非支配解集; $PF(t)$ 为经算法优化得到的实际非支配解集; $d(v, PF(t))$ 为理想解 v 到 $PF(t)$ 的最小欧几里得距离; $d_i(t)$ 为 t 时刻解集中第 i 个解的目标矢量到其他解目标矢量的最小欧几里得距离; $\overline{d(t)}$ 为所有非支配解的平均值; N 为 $PF(t)$ 中解的数量。 $\overline{IGD}$ 的值越小,表明算法随时间变化过程中获得解集的综合质量越好, $\overline{SP}$ 的值越小,表明算法随时间变化过程中求得解集的分布性越好。

3.3 测试结果与分析

6个算法在10个测试问题上 $\overline{SP}$ 和 $\overline{IGD}$ 平均值和方差的统计结果如表1和表2所示。120次环境变化被划分为3个时间段:问题变化的前40次作为第1阶段,中间40次作为第2阶段,后40次作为第3阶段。其中5个获得最好指标值的算法数据被标记为黑体。

由表1可见,本文算法在10个测试问题上都得到较好的评价结果,经本文算法优化后得到的非支配解集的分布性较好。这主要是因为本文算法一方面通过生成一组能均匀分布在 PF 上的进化向量,另一方面基于已寻解在目标空间的距离,插值得到一组解,使其对应在目标空间上的值更均匀。为进一步对比5个算法的求解能力, $\overline{IGD}$ 具体统计结果如表2所示。

由表2可见,本文算法在 FDA_1 、 FDA_2 、 FDA_3 、 FDA_4 、 $DMOP_3$ 、 F_6 六个测试问题上表现出更优的综合性能。对于 $DMOP_1$,FPS在第2阶段和第3阶段得到了更好的评价指标,但该算法在第1阶段的性能指标较差,综合而言MDP-DMGSA在该测试问题上的优化性能更优。对于 $DMOP_2$,PPS在第2阶段和第3阶段的综合评价指标更优,但在第1阶段表现出的指标较差,综合而言本文算法更优。对于 F_5 ,PPS在第2和第3阶段表现出较好的解集质量,但第1阶段的解集质量较差,本文算法在这一阶段表现更优,综合而言,DMS在该测试问题上的综合性能更好,本文算法次之。对于 F_7 ,虽然在第2阶段本文算法不如DMS,但在第1阶段和第3阶段本文算法都表现出了更好的解集质量,综合而言本文算法更优。综合表1和表2,本文算法在求解10个测试问题上,从收敛性和分布性角度评价都得到了更好的寻优效果。究其原因,一方面是因为设计的权重生成策略保证了进化策略

表1 6个算法在10个测试问题上 $\overline{SP}$ 的均值和方差

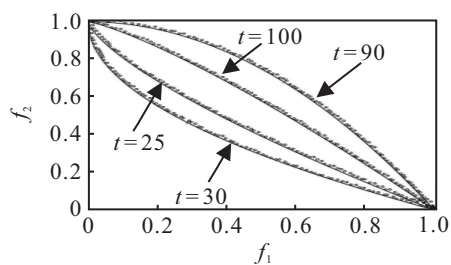
问题	MDP-DMGSA	PPS	FPS	DMS	EGS	MDMOA/D
FDA ₁	0.004 9 (0.000 5)	0.015 4 (0.009 4)	0.019 9 (0.024 1)	0.010 9 (0.007 9)	0.021 9 (0.018 5)	0.037 1 (0.014 5)
FDA ₂	0.008 1 (0.000 6)	0.010 3 (0.006 6)	0.009 0 (0.002 0)	0.008 7 (0.002 0)	0.011 3 (0.009 5)	0.010 1 (0.004 7)
FDA ₃	0.032 6 (0.000 5)	0.013 7 (0.009 1)	0.015 9 (0.011 4)	0.011 6 (0.008 2)	0.020 2 (0.018 5)	0.024 5 (0.010 5)
FDA ₄	0.089 4 (0.001 4)	0.178 0 (0.069 7)	0.277 6 (0.073 1)	0.112 3 (0.046 3)	0.218 4 (0.177 7)	0.123 1 (0.097 7)
DMOP ₁	0.004 1 (0.000 6)	0.013 4 (0.010 7)	0.023 0 (0.042 7)	0.009 4 (0.004 8)	0.024 2 (0.018 8)	0.004 2 (0.000 4)
DMOP ₂	0.004 2 (0.000 5)	0.012 5 (0.009 9)	0.012 1 (0.009 7)	0.005 3 (0.011 8)	0.029 1 (0.027 2)	0.004 5 (0.000 5)
DMOP ₃	0.006 0 (0.000 4)	0.016 0 (0.008 0)	0.018 0 (0.014 8)	0.013 3 (0.013 7)	0.014 1 (0.009 2)	0.007 1 (0.001 0)
F ₅	0.009 8 (0.000 9)	0.021 2 (0.010 5)	0.034 7 (0.010 9)	0.013 9 (0.009 1)	0.023 2 (0.013 4)	0.012 4 (0.009 7)
F ₆	0.002 6 (0.000 3)	0.032 5 (0.021 4)	0.066 5 (0.020 5)	0.005 4 (0.104 7)	0.046 3 (0.022 1)	0.033 9 (0.012 7)
F ₇	0.003 5 (0.000 7)	0.065 3 (0.014 4)	0.101 5 (0.085 7)	0.017 1 (0.008 6)	0.072 4 (0.018 9)	0.015 4 (0.008 9)

表2 6个算法在10个测试问题上 $\overline{IGD}$ 的均值和方差

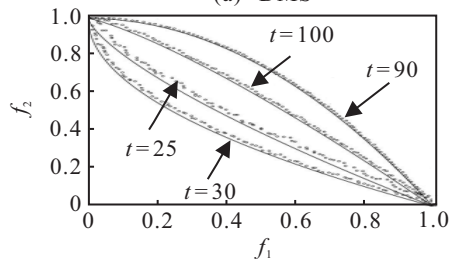
问题	统计	MDP-DMGSA	PPS	FPS	DMS	EGS	MDMOA/D
FDA ₁	total	0.0041 (0.000 5)	0.064 9 (0.028 9)	0.025 4 (0.006 0)	0.007 2 (0.000 6)	0.014 1 (0.001 8)	0.070 5 (0.012 5)
	1st stage	0.004 3 (0.000 6)	0.186 4 (0.087 9)	0.065 3 (0.010 7)	0.008 9 (0.001 0)	0.023 9 (0.004 8)	0.077 0 (0.012 9)
	2nd stage	0.004 0 (0.000 4)	0.004 2 (0.000 4)	0.010 0 (0.000 5)	0.007 0 (0.000 5)	0.009 0 (0.000 8)	0.069 4 (0.010 1)
	3rd stage	0.004 0 (0.000 4)	0.004 1 (0.000 4)	0.000 9 (0.000 6)	0.005 6 (0.000 4)	0.009 1 (0.000 7)	0.065 3 (0.009 7)
FDA ₂	total	0.006 2 (0.000 6)	0.013 5 (0.000 7)	0.013 1 (0.000 6)	0.016 1 (0.000 6)	0.013 0 (0.000 5)	0.011 0 (0.001 1)
	1st stage	0.005 4 (0.000 4)	0.016 9 (0.001 5)	0.017 0 (0.001 4)	0.017 9 (0.000 7)	0.016 0 (0.001 1)	0.012 7 (0.001 2)
	2nd stage	0.006 3 (0.000 7)	0.011 9 (0.000 2)	0.011 2 (0.000 3)	0.015 0 (0.000 5)	0.011 4 (0.000 2)	0.010 4 (0.000 9)
	3rd stage	0.006 9 (0.000 6)	0.011 8 (0.000 2)	0.011 1 (0.000 3)	0.014 9 (0.000 8)	0.011 3 (0.000 2)	0.010 1 (0.001 1)
FDA ₃	total	0.027 9 (0.004 0)	0.172 1 (0.065 0)	0.124 1 (0.007 7)	0.028 0 (0.009 0)	0.078 0 (0.006 2)	0.030 9 (0.007 3)
	1st stage	0.027 8 (0.003 8)	0.361 4 (0.196 6)	0.145 1 (0.009 1)	0.029 5 (0.001 2)	0.084 0 (0.008 0)	0.032 4 (0.006 9)
	2nd stage	0.027 5 (0.001 2)	0.081 0 (0.015 8)	0.115 0 (0.007 0)	0.027 6 (0.001 3)	0.072 9 (0.004 9)	0.030 5 (0.007 4)
	3rd stage	0.025 8 (0.001 4)	0.074 0 (0.014 9)	0.112 2 (0.007 0)	0.027 1 (0.000 7)	0.0772 (0.006 3)	0.029 7 (0.007 1)
FDA ₄	total	0.069 7 (0.005 7)	0.146 6 (0.006 4)	0.209 4 (0.013 5)	0.114 9 (0.005 9)	0.160 1 (0.015 7)	0.012 7 (0.008 5)
	1st stage	0.069 9 (0.006 1)	0.183 9 (0.012 7)	0.227 0 (0.015 7)	0.131 2 (0.008 8)	0.194 7 (0.019 4)	0.013 8 (0.008 9)
	2nd stage	0.069 6 (0.005 3)	0.128 6 (0.006 0)	0.203 2 (0.012 9)	0.106 7 (0.006 9)	0.142 8 (0.015 0)	0.012 4 (0.008 1)
	3rd stage	0.069 6 (0.005 2)	0.127 4 (0.004 9)	0.198 0 (0.012 9)	0.106 7 (0.005 1)	0.142 7 (0.014 6)	0.011 9 (0.008 0)
DMOP ₁	total	0.006 2 (0.000 4)	0.024 7 (0.068 0)	0.009 0 (0.003 7)	0.009 0 (0.003 1)	0.017 9 (0.007 2)	0.023 6 (0.009 6)
	1st stage	0.006 4 (0.000 6)	0.066 1 (0.202 9)	0.023 0 (0.008 9)	0.020 9 (0.008 9)	0.048 8 (0.020 9)	0.026 2 (0.010 0)
	2nd stage	0.005 8 (0.000 4)	0.004 0 (0.000 5)	0.002 0 (0.000 3)	0.003 1 (0.000 6)	0.002 5 (0.000 6)	0.022 7 (0.009 0)
	3rd stage	0.006 3 (0.000 3)	0.003 9 (0.000 5)	0.002 0 (0.000 4)	0.003 0 (0.000 6)	0.002 5 (0.000 7)	0.0220 (0.009 2)
DMOP ₂	total	0.004 6 (0.000 8)	0.114 6 (0.062 1)	0.011 3 (0.010 3)	0.008 1 (0.000 8)	0.018 7 (0.001 9)	0.012 9 (0.007 7)
	1st stage	0.004 7 (0.000 9)	0.335 9 (0.187 5)	0.015 7 (0.002 8)	0.012 1 (0.001 5)	0.031 2 (0.004 7)	0.013 5 (0.008 5)
	2nd stage	0.004 6 (0.000 8)	0.003 9 (0.000 8)	0.009 7 (0.001 4)	0.006 3 (0.000 6)	0.012 4 (0.000 8)	0.012 7 (0.007 4)
	3rd stage	0.004 5 (0.000 7)	0.003 9 (0.000 7)	0.009 6 (0.001 3)	0.006 0 (0.000 5)	0.012 3 (0.009 0)	0.012 5 (0.007 0)
DMOP ₃	total	0.004 3 (0.000 3)	0.040 3 (0.019 0)	0.026 6 (0.004 7)	0.007 4 (0.000 5)	0.013 4 (0.001 4)	0.018 4 (0.002 5)
	1st stage	0.004 4 (0.000 4)	0.110 0 (0.056 9)	0.057 9 (0.015 1)	0.008 8 (0.000 8)	0.020 9 (0.002 9)	0.019 1 (0.003 6)
	2nd stage	0.004 3 (0.000 3)	0.005 5 (0.000 8)	0.010 9 (0.000 5)	0.006 7 (0.000 3)	0.009 6 (0.000 6)	0.018 2 (0.002 1)
	3rd stage	0.004 3 (0.000 3)	0.005 4 (0.000 7)	0.010 8 (0.000 6)	0.006 6 (0.000 3)	0.009 7 (0.000 9)	0.018 0 (0.002 0)
F ₅	total	0.015 3 (0.001 2)	0.106 2 (0.076 9)	0.095 0 (0.051 2)	0.015 2 (0.001 4)	0.051 3 (0.007 8)	0.075 4 (0.010 2)
	1st stage	0.012 0 (0.000 9)	0.294 0 (0.227 9)	0.197 9 (0.158 1)	0.017 9 (0.002 4)	0.059 1 (0.011 4)	0.076 3 (0.012 1)
	2nd stage	0.017 7 (0.001 6)	0.013 1 (0.001 3)	0.041 1 (0.010 9)	0.014 1 (0.002 2)	0.047 6 (0.017 9)	0.075 1 (0.009 1)
	3rd stage	0.016 3 (0.001 5)	0.011 4 (0.001 2)	0.046 0 (0.012 7)	0.013 7 (0.001 4)	0.047 3 (0.009 0)	0.074 9 (0.009 3)
F ₆	total	0.004 8 (0.000 7)	0.120 2 (0.028 0)	0.125 9 (0.027 9)	0.023 4 (0.004 9)	0.048 4 (0.014 1)	0.165 4 (0.015 7)
	1st stage	0.005 1 (0.000 9)	0.337 9 (0.083 9)	0.325 8 (0.084 1)	0.051 2 (0.015 1)	0.083 9 (0.039 5)	0.171 4 (0.014 7)
	2nd stage	0.004 7 (0.000 7)	0.011 5 (0.000 7)	0.026 3 (0.004 2)	0.009 6 (0.000 6)	0.031 1 (0.004 7)	0.164 9 (0.012 1)
	3rd stage	0.004 6 (0.000 6)	0.011 1 (0.000 5)	0.025 7 (0.006 2)	0.009 6 (0.000 6)	0.030 1 (0.004 2)	0.160 1 (0.010 6)
F ₇	total	0.009 9 (0.005 8)	0.052 2 (0.024 0)	0.055 7 (0.012 5)	0.016 5 (0.002 4)	0.034 6 (0.005 7)	0.019 4 (0.008 4)
	1st stage	0.011 8 (0.000 6)	0.134 8 (0.072 0)	0.099 1 (0.031 0)	0.034 0 (0.005 4)	0.048 0 (0.013 9)	0.020 1 (0.009 7)
	2nd stage	0.010 4 (0.001 0)	0.011 1 (0.000 5)	0.031 9 (0.010 7)	0.007 8 (0.000 6)	0.027 9 (0.005 9)	0.019 5 (0.008 6)
	3rd stage	0.007 6 (0.000 4)	0.010 8 (0.000 6)	0.036 1 (0.023 9)	0.007 9 (0.000 6)	0.027 9 (0.004 7)	0.018 7 (0.008 2)

在优化子问题最优解时,其对应目标空间的均匀性,插值生成策略的设计丰富了解集的多样性;另一方面,有效解决了PF形状发生变化时,对非支配解在目

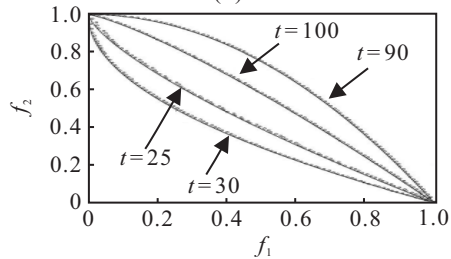
标空间分布性的影响.最后基于相邻子问题最优解的相近性以及同一权重向量对应子问题最优解的相近性预测未来子问题最优解,提高了算法在环境变



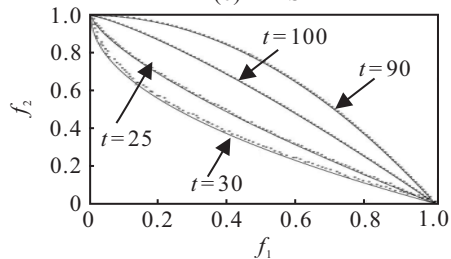
(a) DMS



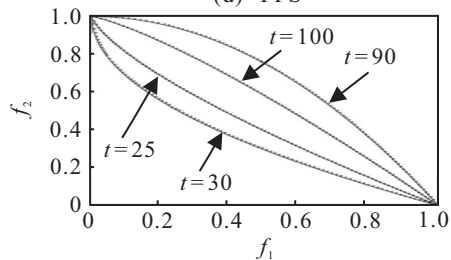
(b) EGS



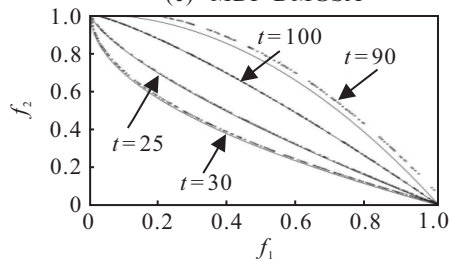
(c) EPS



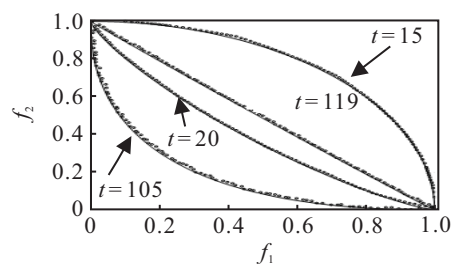
(d) PPS



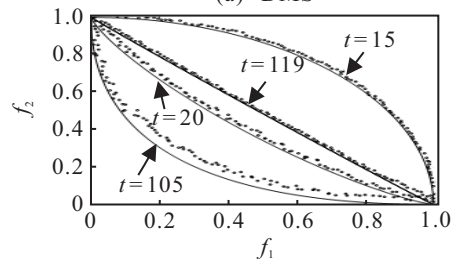
(e) MDP-DMGSA



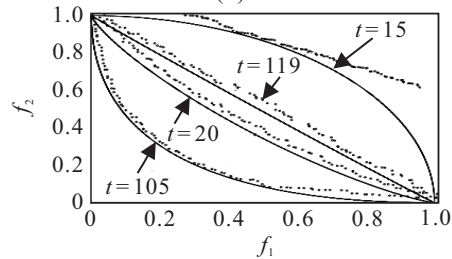
(f) MDMOA/D

图4 6个算法对DMOP₂得到的非支配解集

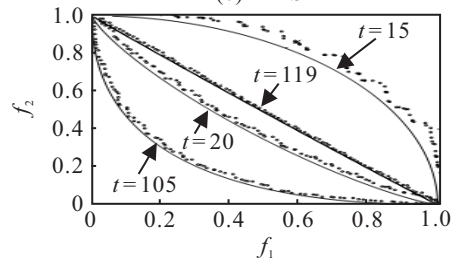
(a) DMS



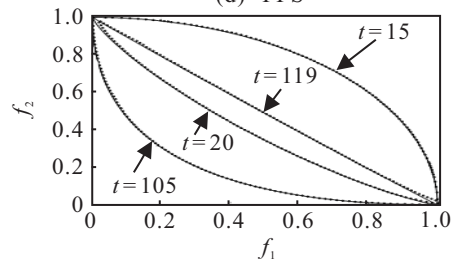
(b) EGS



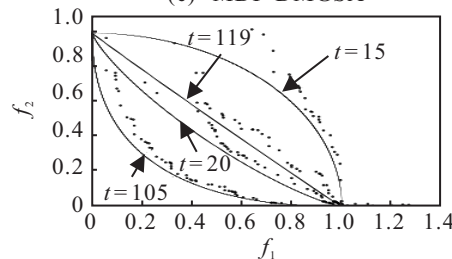
(c) EPS



(d) PPS



(e) MDP-DMGSA



(f) MDMOA/D

图5 6个算法对F₆得到的非支配解集

化后的寻优效率,因此本文算法取得了更好的寻优效果。

为了更直观地比较6个算法的寻优能力,图4和图5绘制了算法对DMOP₂和F₆在不同时期的非支配解集分布图。

由图4可见,EGS和PPS在 $t = 25$ 和 $t = 30$ 次环境变化时收敛性略差,但随着模型对历史解信息的积累,在 $t = 90$ 和 $t = 100$ 时收敛性和分布性有所提高。综合而言,FPS和DMS的分布性优于PPS和EGS,但其收敛性仍然不如本文算法。从图4可以看出,本文算法优化DMOP₂,在4个阶段均展现了更好的收敛性和分布性。

由图5可见:FPS和PPS在 $t = 15$ 时的分布性和收敛性较差,随着变化次数的增多,虽然收敛性和分布性有所提高,但仍有较大的提升空间;EGS虽然在 $t = 15$ 时收敛性和分布性优于FPS和PPS,但在问题变化后期,解集收敛性有所下降,表明该算法在求解变化规律较强的问题时能展现较好的性能,当问题的变化趋势与预测模型不符时,收敛性和分布性都将下降。DMS和MDP-DMGSA在不同阶段的收敛性都相对更优,但本文算法在优化问题初期的解集收敛性和分布性优于DMS,综合而言,本文算法的分布性和收敛性更优。

综合图4和图5,本文算法在问题变化的不同阶段,收敛性和分布性都能保持较好的效果。分析其原因,主要是因为本文算法采用多种群串行搜索的寻优机制,在环境变化前基于偏好子问题的最优目标值生成一组彼此平行且等间距的进化向量,使进化向量与PF的交点均匀分布在PF上,且插值策略又进一步提升了解集多样性和分布性。在环境变化后,基于相邻进化向量对应解的相近性以及相邻进化向量在环境变化前后对应最优解的相近性预测当前子问题的解,能够有效降低种群寻优空间,提高求解效率。

4 结 论

为在求解DMOPs时获得收敛性和分布性都更优的非支配解集,本文提出了多种群分解预测动态多目标算法。首先,基于偏好子问题的最优解,生成一组互相平行且相邻距离相等的方向向量,使其对应的子问题的解均匀地分布在PF上,且基于相邻子问题的解初始化种群,提高进化策略寻优速度;其次,为提高PS的多样性并保持分布均匀性,根据进化向量子问题的解在目标空间上的位置,插值生成更多非支配解,提高解集的多样性并保证解集分布的均匀性;再次,在环境变化后,为快速得到更优解,基于相邻子

问题对应最优解的相近性以及环境变化前后相似进化向量对应子问题的解存在相近性,预测生成新的种群;最后,通过在10个标准测试函数问题上的实验表明,采用本文算法求解DMOPs相比其他算法具有更好的解集收敛性和分布性。

参考文献(References)

- [1] Eaton J, Yang S X, Gongora M. Ant colony optimization for simulated dynamic multi-objective railway junction rescheduling[J]. IEEE Transactions on Intelligent Transportation Systems, 2017, 18(11): 2980-2992.
- [2] Chen L W, Hu T Y. Flow equilibrium under dynamic traffic assignment and signal control — An illustration of pretimed and actuated signal control policies [J]. IEEE Transactions on Intelligent Transportation Systems, 2012, 13(3): 1266-1276.
- [3] Wu X L, Ma Z, Wang Y. Joint user grouping and resource allocation for multi-user dual layer beamforming in LTE-A[J]. IEEE Communications Letters, 2015, 19(10): 1822-1825.
- [4] Navratil P A, Childs H, Fussell D S, et al. Exploring the spectrum of dynamic scheduling algorithms for scalable distributed-memory ray tracing[J]. IEEE Transactions on Visualization and Computer Graphics, 2014, 20(6): 893-906.
- [5] 刁鹏飞, 王艳娇. 基于节点休眠的水下无线传感器网络覆盖保持分簇算法[J]. 电子与信息学报, 2018, 40(5): 1101-1107.
(Diao P F, Wang Y J. Coverage-preserving clustering algorithm for underwater sensor networks based on the sleeping mechanism[J]. Journal of Electronics & Information Technology, 2018, 40(5): 1101-1107.)
- [6] 毕晓君, 刁鹏飞. 基于引力搜索算法的异构无线传感器网络路由分簇算法[J]. 控制与决策, 2017, 32(3): 563-569.
(Bi X J, Diao P F. Routing and clustering algorithm heterogeneous wireless sensor networks based on gravitational search algorithm[J]. Control and Decision, 2017, 32(3): 563-569.)
- [7] Deb K, Rao N U B, Bhaskara U N. Dynamic multi-objective optimization and decision-making using modified NSGA-II: A case study on hydro-thermal power scheduling[C]. Lecture Notes in Computer Science. Berlin: Springer, 2007: 803-817.
- [8] Guo Y N, Yang H, Chen M R, et al. Ensemble prediction-based dynamic robust multi-objective optimization methods[J]. Swarm and Evolutionary Computation, 2019, 48: 156-171.
- [9] Wang Y J, Du T L, Liu T T, et al. Dynamic multiobjective squirrel search algorithm based on decomposition with evolutionary direction prediction and bidirectional memory populations[J]. IEEE Access, 2019, 7: 115997-116013.

- [10] Zhou A M, Jin Y C, Zhang Q F. A population prediction strategy for evolutionary dynamic multiobjective optimization[J]. IEEE Transactions on Cybernetics, 2014, 44(1): 40-53.
- [11] 张世文, 李智勇, 陈少森, 等. 基于生态策略的动态多目标优化算法[J]. 计算机研究与发展, 2014, 51(6): 1313-1330.
(Zhang S W, Li Z Y, Chen S M, et al. Dynamic multi-objective optimization algorithm based on ecological strategy[J]. Journal of Computer Research and Development, 2014, 51(6): 1313-1330.)
- [12] 刘若辰, 马亚娟, 张浪, 等. 基于预测策略的动态多目标免疫优化算法[J]. 计算机学报, 2015, 38(8): 1544-1560.
(Liu R C, Ma Y J, Zhang L, et al. Dynamic multi-objective immune optimization algorithm based on prediction strategy[J]. Chinese Journal of Computers, 2015, 38(8): 1544-1560.)
- [13] 郑金华, 彭舟, 邹娟, 等. 基于引导个体的预测策略求解动态多目标优化问题[J]. 电子学报, 2015, 43(9): 1816-1825.
(Zheng J H, Peng Z, Zou J, et al. A prediction strategy based on guide-individual for dynamic multi-objective optimization[J]. Acta Electronica Sinica, 2015, 43(9): 1816-1825.)
- [14] 丁进良, 杨翠娥, 陈立鹏, 等. 基于参考点预测的动态多目标优化算法[J]. 自动化学报, 2017, 43(2): 313-320.
(Ding J L, Yang C E, Chen L P, et al. Dynamic multi-objective optimization algorithm based on reference point prediction[J]. Acta Automatica Sinica, 2017, 43(2): 313-320.)
- [15] 陈美蓉, 郭一楠, 巩敦卫, 等. 一类新型动态多目标鲁棒进化优化方法[J]. 自动化学报, 2017, 43(11): 2014-2032.
(Chen M R, Guo Y N, Gong D W, et al. A novel dynamic multi-objective robust evolutionary optimization method[J]. Acta Automatica Sinica, 2017, 43(11): 2014-2032.)
- [16] Ruan G, Yu G, Zheng J H, et al. The effect of diversity maintenance on prediction in dynamic multi-objective optimization[J]. Applied Soft Computing, 2017, 58(6): 631-647.
- [17] 刁鹏飞, 毕晓君, 王艳娇. 基于分解技术的动态多目标引力搜索算法[J]. 系统工程理论与实践, 2018, 38(5): 1300-1309.
(Diao P F, Bi X J, Wang Y J. Dynamic multi-objective gravitational searching algorithm based on decomposition[J]. System Engineering—Theory & Practice, 2018, 38(5): 1300-1309.)
- [18] Liu R C, Li J X, Fan J, et al. A dynamic multiple populations particle swarm optimization algorithm based on decomposition and prediction[J]. Applied Soft Computing, 2018, 73(8): 434-459.
- [19] 耿焕同, 周山胜, 陈哲, 等. 基于分解的预测型动态多目标粒子群优化算法[J]. 控制与决策, 2019, 34(6): 1307-1318.
(Geng H T, Zhou S S, Chen Z, et al. A decomposition-based predictive particle swarm optimization for dynamic multi-objective optimization research[J]. Control and Decision, 2019, 34(6): 1307-1318.)
- [20] Rong M, Gong D W, Pedrycz W, et al. A multi-model prediction method for dynamic multi-objective evolutionary optimization[J]. IEEE Transactions on Evolutionary Computation, 2020, 24(2): 290-304.
- [21] Rashedi E, Nezamabadi-Pour H, Saryazdi S. GSA: A gravitational search algorithm[J]. Information Sciences, 2009, 179(13): 2232-2248.
- [22] Zhang Q F, Li H. MOEA/D: A multi objective evolutionary algorithm based on decomposition[J]. IEEE Transactions on Evolutionary Computation, 2007, 11(6): 712-731.
- [23] Iason H, David W. Dynamic multi-objective optimization with evolutionary algorithms: A forward-looking approach[C]. Conference on Genetic & Evolutionary Computation. New York: ACM, 2006: 1201-1208.
- [24] Koo W T, Goh C K, Tan K C. A predictive gradient strategy for multiobjective evolutionary algorithms in a fast changing environment[J]. Memetic Computing, 2010, 2(2): 87-110.
- [25] Farina M, Deb K, Amato P, et al. Dynamic multiobjective optimization problems: Test cases, approximations, and applications[J]. IEEE Transactions on Evolutionary Computation, 2004, 8(5): 425-442.
- [26] Goh C K, Tan K C. A competitive-cooperative coevolutionary paradigm for dynamic multiobjective optimization[J]. IEEE Transactions on Evolutionary Computation, 2009, 13(1): 103-127.
- [27] Schott J R. Fault tolerant design using single and multi-criteria genetic algorithms[J]. Master's Thesis, Massachusetts Institute of Technology, 1995, 37(1): 1-13.

作者简介

刁鹏飞(1987—), 男, 讲师, 博士, 从事智能优化、无线传感器网络优化等研究, E-mail: 398317196@qq.com;

李树森(1963—), 男, 教授, 博士, 从事精密机械设计、气体轴承润滑和起重运输机械等研究, E-mail: 601559001@qq.com;

姜雪松(1979—), 男, 副教授, 博士, 从事运筹优化、复杂系统优化等研究, E-mail: 120084556@qq.com.

(责任编辑: 郑晓蕾)