

# 基于局部时空的多峰优化算法及其在 PID控制中的应用

赵宏 李珈瑞 刘静

(西安电子科技大学广州研究院 广州 510555)

**摘要** 多峰优化问题(MultiModal Optimization Problems, MMOPs)需要同时找到问题的多个高精度全局最优解,它需要算法具有较强的全局搜索能力且能很好地平衡种群的多样性和收敛性。当前在处理 MMOPs 时通常面临以下难点:(1)现有方法通常只考虑到进化过程中种群的当前状态(如常用的贪婪选择策略),容易导致种群陷入局部最优;(2)传统的随机搜索策略在复杂搜索空间内难以快速有效找到全局最优解;(3)当前设计的多峰优化算法往往需要手工设置参数(如变异因子和交叉因子等),而参数的大小将直接影响种群的多样性和收敛性。针对上述难点,本文提出了一种新的基于局部时空的多峰优化(Localized Time-Distance-based Multimodal Optimization, LTDMO)算法,主要包括三个贡献点:首先,提出了结合随机搜索和定向引导的变异(Random and Direction-based Mutation, RDM)策略,利用随机变异增加种群中个体的多样性,并通过划分邻域将整个种群分成不同的可重叠子种群,在局部搜索空间内进行变异操作来更好地定位全局最优解,从而避免个体陷入局部最优。其次,提出了基于时间局部性原理的拥挤选择(Locality-based Crowding Selection, LCS)策略,利用进化过程中的时间局部性记录对当前个体更有潜力的进化方向,并在此方向上生成新的子代,使种群进一步向全局最优解收敛。最后,提出了自适应参数控制(Self-adaptive Parameter Control, SPC)策略,基于个体进化信息自适应调整算法的参数值,降低算法在进化过程中对变异因子和交叉因子的参数敏感性。本文将 LTDMO 算法在 CEC'2013 测试集上进行实验,并将结果与其他 11 种多峰优化算法对比,表明 LTDMO 算法能有效处理较多的全局最优复杂多峰优化问题,具体地,在 F1~F5、F8 和 F10 问题上峰值率和成功率均达到 100%;在具有较多局部最优的多峰优化问题(F6 和 F7)上,LTDMO 算法的峰值率达到 86%以上,这优于 9 种其他对比算法的性能;在处理复合多峰优化问题时,LTDMO 算法在处理 F11、F12、F14、F16 问题上性能达到最优。同时将 LTDMO 算法在比例-积分-微分(Proportional-Integral-Derivative, PID)控制器上进行应用,结果表明 LTDMO 算法能为 PID 控制器找到多种最优控制参数,使系统达到稳定状态且误差更小。

**关键词** 多峰优化问题;邻域变异;时间局部性;自适应调整参数;PID控制

**中图法分类号** TP18 **DOI号** 10.11897/SP.J.1016.2024.01323

## Localized Time-Distance-Based Multimodal Optimization Algorithm and Its Application in PID Control

ZHAO Hong LI Jia-Rui LIU Jing

(Guangzhou Institute of Technology, Xidian University, Guangzhou 510555)

**Abstract** MultiModal Optimization Problems (MMOPs) require finding multiple high-precision global optima of a problem simultaneously, necessitating algorithms that have strong global search capabilities and can well balance the diversity and convergence of the population. Currently,

收稿日期:2023-06-02;在线发布日期:2024-04-08。本课题得到国家自然科学基金(62306225)、广东省基础与应用基础研究基金(2021A151511073,2022A1515011297,202201011236)、广东省高水平创新研究机构项目(2021B0909050008)、广州市重点研发计划(202206030003)资助。赵宏,博士,副教授,中国计算机学会(CCF)会员,主要研究方向为进化计算、群智优化以及相关应用的优化算法设计。E-mail: hongzhao@xidian.edu.cn。李珈瑞(通信作者),硕士研究生,主要研究方向为进化计算、群智优化以及相关应用的优化算法设计。E-mail: ljrdyx2@163.com。刘静,博士,教授,中国计算机学会(CCF)高级会员,主要研究领域为进化计算、智能优化、复杂网络和数据挖掘。

when dealing with MMOPs, there are typically the following difficulties: (1) Existing methods often only consider the current state of the population during the evolutionary process (such as the commonly used greedy selection strategy), which can easily lead to the population being trapped in local optima; (2) Traditional random search strategies have difficulty quickly and effectively finding global optima within complex search spaces; (3) Current multimodal optimization algorithm designs often require manual parameter setting (such as mutation and crossover factors), where the magnitude of these parameters directly impacts the population's diversity and convergence. To address these challenges, this paper introduces a new Localized Time-Distance-based Multimodal Optimization (LTDMO) algorithm, mainly contributing in three areas: First, a Random and Direction-based Mutation (RDM) strategy combining random search and directed guidance is proposed, using random mutation to increase the diversity of individuals within the population, and by dividing the population into different, possibly overlapping subpopulations for mutation operations in local search spaces, better locating global optima and thus avoiding individuals falling into local optima. Second, a Locality-based Crowding Selection (LCS) strategy is proposed, utilizing the principle of temporal locality in the evolutionary process to record more promising evolutionary directions for the current individual, generating new offspring in this direction to further converge the population towards global optima. Lastly, a Self-adaptive Parameter Control (SPC) strategy is proposed, which adaptively adjusts the algorithm's parameter values based on individual evolutionary information, reducing the algorithm's sensitivity to parameters like mutation and crossover factors during the evolutionary process. The LTDMO algorithm was tested on the CEC'2013 benchmark, and the results were compared with those of 11 other multimodal optimization algorithms, showing that the LTDMO algorithm can effectively handle complex multimodal optimization problems with many global optima. Specifically, on problems F1 – F5, F8, and F10, the peak rate and success rate both reached 100%; on multimodal optimization problems with many local optima (F6 and F7), the peak rate of the LTDMO algorithm exceeded 86%, surpassing the performance of nine comparison algorithms; in dealing with composite multimodal optimization problems, the LTDMO algorithm achieved optimal performance on problems F11, F12, F14, and F16. Furthermore, applying the LTDMO algorithm to the Proportional-Integral-Derivative (PID) controller shows that the LTDMO algorithm can find various optimal control parameters for the PID controller, allowing the system to reach a stable state with smaller error.

**Keywords** multimodal optimization problem; neighborhood mutation; temporal locality; adaptively adjust parameters; proportional integral derivative controller

## 1 引 言

多峰优化问题(MultiModal Optimization Problems, MMOPs)<sup>[1-4]</sup>在实际生活中较为常见,这类问题需要算法在一定精度下同时找到多个全局最优解.例如在水库发电的调度优化问题中,要根据发电站不同时段的水力情况、不同电站的发电能力等,对电站进行安排和调度,使得水库的发电量最大,而在不同情境下采用多种不同的发电策略.又如,对一个受温度影响的系统,温度为 10℃和 35℃时系统

的性能均达到最优,这时需要得到系统在两种温度下的最优执行方案,冬天使用 10℃对应的最优解,夏天使用 35℃对应的最优解.因此,设计的算法需要提供尽可能多的最优解以供决策者选择使用,即多峰优化算法.

进化算法(Evolutionary Algorithms, EAs)是一种成熟且广泛适用的全局优化方法,具有自适应性、高鲁棒性,能够有效处理多种类型的复杂优化问题<sup>[5-7]</sup>.因此,EAs也被用来处理 MMOPs,常用的方法有差分进化(Differential Evolution, DE)算法<sup>[8-9]</sup>、遗传算法(Genetic Algorithm, GA)<sup>[10-11]</sup>、蚁群优化

(Ant Colony Optimization, ACO)算法<sup>[12-14]</sup>、粒子群优化(Particle Swarm Optimization, PSO)算法<sup>[15-18]</sup>等. 其中 DE 算法通过模仿自然界中的生物进化现象, 在进化的过程中利用个体的差分信息进行变异, 并不断选择较优个体进入下一代, 从而使种群中个体的适应度越来越高. DE 算法具有全局搜索能力强、收敛速度快的优点, 因此本文将 DE 算法作为基准算法求解 MMOPs.

近年来, 基于 DE 解决 MMOPs 的算法越来越多, 常见的有拥挤差分进化(Crowding Differential Evolution, CDE)算法<sup>[19]</sup>、物种差分进化(Speciation Differential Evolution, SDE)算法<sup>[20]</sup>、邻域拥挤差分进化(Neighborhood Crowding Differential Evolution, NCDE)算法、邻域物种差分进化(Neighborhood Speciation Differential Evolution, NSDE)算法<sup>[21]</sup>等. 此外, 许多学者提出了基于不同进化策略的方法来处理 MMOPs, 例如: Zhang 等人<sup>[22]</sup>基于外部存档策略来进化种群中的个体, 这种策略结合外部历史存档和不同父代的差分向量, 充分利用存储的历史信息生成新的个体; Schmidt 等人<sup>[23]</sup>提出了一种与个体进化时间相关的算法, 在进化初期一些个体适应度较差, 但这些个体并不能被完全否定, 它们具有一定的潜力去搜索最优解, 此策略通过设置一定阈值保证表现较差的个体在进化初期也有一定生存概率, 从而有利于保持种群中个体的多样性; 另外, 对种群进行聚类也是一种被广泛应用的方法<sup>[24]</sup>, 这类方法将整个种群划分为多个子种群, 距离较近且具有相似属性的个体将共同进化, 同时在不同子种群中也可以应用不同的进化策略, 使个体朝着相应的峰值移动, 从而更好地定位多个全局最优解.

以上方法在进化过程中设计了不同进化策略来提高算法的全局搜索能力并取得了一定成效, 但目前在处理 MMOPs 上仍面临以下的问题:

(1) 如何避免种群中的个体陷入局部最优仍是求解 MMOPs 的一个挑战. 当局部最优解附近有较多个体时, 经过迭代进化后这些个体很可能会收敛到局部峰值, 导致进化停滞、个体无法继续寻找真正的全局峰值.

(2) 如何提高种群的收敛能力, 从而找到尽可能多的全局最优解也是求解 MMOPs 的一个挑战. 在收敛性好的策略中, 种群的收敛方向更有效、更接近峰值方向, 从而可以定位到更多的全局最优解, 则算法的性能越好.

(3) 参数设置对算法性能影响较大. 算法设计

中涉及各种各样的参数, 相同情况下不同的参数设置可能对算法的寻优能力产生较大的影响<sup>[25-26]</sup>, 例如: DE 算法中的变异因子  $F$  影响种群中个体的多样性, 交叉因子  $CR$  影响算法的收敛性.

鉴于以上问题和挑战, 本文提出了一种基于局部时空的多峰优化(Localized Time-Distance-based Multimodal Optimization, LTDMO)算法. 该算法主要包括以下三种策略: 结合随机搜索和定向引导的变异(Random and Direction-based Mutation, RDM)策略、基于时间局部性原理的拥挤选择(Locality-based Crowding Selection, LCS)策略和自适应参数控制(Self-adaptive Parameter Control, SPC)策略: 首先, RDM 策略利用随机变异增加种群中个体的多样性, 并通过划分邻域将整个种群分成不同的可重叠子种群, 在局部搜索空间内进行变异操作来更好地定位全局最优解, 从而避免个体陷入局部最优; 其次, LCS 策略借助时间局部性原理记录有潜力的进化方向, 利用这个方向继续产生子代并与当前父代进行比较, 选择更优的个体进入下一代, 使种群更好地向峰值收敛; 最后, SPC 策略结合种群中个体本身的特性与历史信息, 存储个体不同状态下的变异因子  $F$  值和交叉因子  $CR$  值, 并动态自适应地调整这两个参数, 增强算法鲁棒性, 降低算法对参数的敏感性.

此外, 在比例-积分-微分(Proportional-Integral-Derivative, PID)控制系统中, 需要找到 PID 控制器的最优参数从而使系统达到最优性能. 一般存在一组或多组最优参数都能使系统性能达到最优, 不同的最优参数组合能从不同角度满足 PID 控制的需求, 由此 PID 参数整定问题是一种 MMOPs. 本文将提出的 LTDMO 算法用于解决 PID 控制问题, 从而展示了 LTDMO 算法的有效性.

本文第 2 节介绍 DE 算法和当前 MMOPs 的研究现状, 并总结 PID 控制的相关理论背景; 第 3 节提出 LTDMO 算法并详细说明其设计思想和算法流程, 分析 LTDMO 算法的复杂度; 第 4 节将 LTDMO 算法在 CEC'2013 测试集上进行实验, 并在 PID 参数整定问题上进行应用来说明 LTDMO 算法的有效性; 最后, 第 5 节为本文总结.

## 2 相关工作

本节首先给出了经典 DE 算法的一般过程, 对初始化、变异、交叉、边界处理、选择这 5 个过程进行

了简要描述;其次综述了当前 MMOPs 的相关工作;最后介绍了 PID 控制的相关理论背景,概括了 PID 参数整定问题的原理,并以自动高炮随动系统为例说明了参数整定的重要性。

## 2.1 DE 算法

Storn 等人<sup>[27]</sup>提出了标准的 DE 算法,DE 是一种随机启发式算法,包括变异、交叉、选择等过程.其中用于进化的种群包含多个个体,这里“个体”是一个多维变量,评估个体优劣的标准是评价函数.对于最大化问题,评价函数值越大表示个体适应度越高,在进化过程中选择适应度更优的个体进入下一代,同时所有个体不断更新,最后使种群中个体的适应度达到最优.种群进化的过程如下:

(1) 初始化.在进化开始前,首先需要有一个原始种群,该种群一般是在给定的约束边界内随机生成,即生成一个规模为  $N \times D$  的向量矩阵,其中  $N$  表示个体数量, $D$  表示个体维度,种群中个体表示为  $\mathbf{x}_i = (x_{i,1}, x_{i,2}, \dots, x_{i,D}) (i=1, 2, \dots, N)$ .

(2) 变异.变异过程利用父代基向量、差分向量和缩放比例因子生成一个新向量,即 DE 算法中差分的概念发生在变异时,是通过向量和加权差值的相加实现的.经典的变异操作如式(1)和式(2),其中  $\mathbf{x}_{r_1}$ ,  $\mathbf{x}_{r_2}$  和  $\mathbf{x}_{r_3}$  表示随机选择的个体,  $\mathbf{x}_{\text{best}}$  表示种群中的最优个体,  $\mathbf{v}_i$  表示变异后生成的个体:

DE/rand/1:

$$\mathbf{v}_i = \mathbf{x}_{r_1} + F \cdot (\mathbf{x}_{r_2} - \mathbf{x}_{r_3}) \quad (1)$$

DE/best/1:

$$\mathbf{v}_i = \mathbf{x}_{\text{best}} + F \cdot (\mathbf{x}_{r_1} - \mathbf{x}_{r_2}) \quad (2)$$

(3) 交叉.交叉操作通过结合当前个体和变异后生成个体的不同基因<sup>[28]</sup>,用于生成新个体.它的主要思想是:以一定概率生成新个体不同位置上的基因,“一定概率”指的是  $CR$  的值, $CR$  越大表示个体间信息量的交换程度越大.本文采用均匀交叉算子,其中  $r$  表示 0 到 1 之间的随机数,即  $r \in (0, 1)$ ,  $j_r$  表示  $\{1, 2, \dots, D\}$  中的随机数,具体如式(3):

$$\mathbf{u}_{i,j} = \begin{cases} \mathbf{v}_{i,j}, & \text{if } r \leq CR \text{ or } j = j_r, \\ \mathbf{x}_{i,j}, & \text{其他} \end{cases} \quad (3)$$

(4) 边界处理.在经过变异、交叉操作后,由于个体基因的改变和交换,一些新生成的个体基因值会超出约束边界范围,因此要检查超出范围的个体基因并重新定义.一般情况下采用边界吸收的方法,即当变量值超出上界时定义为上界值,超出下界时定义为下界值.

(5) 选择.以上操作结束后,产生了一个新的子代个体  $\mathbf{u}_i$ ,但这个子代的优劣是未知的,能否成功进化用于下一代也是未知的.因此要进行选择操作,评估新生成的子代并根据适应度的优劣选择更好的个体进入下一代,如式(4):

$$\mathbf{x}'_i = \begin{cases} \mathbf{u}_i, & \text{if } f(\mathbf{u}_i) > f(\mathbf{x}_i) \\ \mathbf{x}_i, & \text{其他} \end{cases} \quad (4)$$

其中  $\mathbf{x}'_i$  表示选择的下一代个体,  $f(\cdot)$  表示适应度评估函数,用于计算个体的适应值.

本文中使用的 CEC'2013 测试集用于求解最大化问题,因此每次进化都选择适应值较大的个体进入下一代.

## 2.2 MMOPs 的相关工作

不同于传统单目标问题中只有一个全局最优解的特点,MMOPs 是一类具有多个全局最优解和多个局部最优解的问题,因此定位所有全局峰值更具挑战.在这一部分,本文从以下三个方面总结了 MMOPs 的相关工作.

### 2.2.1 基于小生境技术的多峰优化算法

人们提出了许多基于小生境技术的算法来处理 MMOPs,如基于拥挤(Crowding)<sup>[19]</sup>、物种形成(Speciation)<sup>[20]</sup>、聚类(Clustering)<sup>[29]</sup>、适应值共享(Fitness sharing)<sup>[30]</sup>等.这些方法将种群划分为包含许多独立子种群的小生境,旨在促进种群的多样性,防止种群过早收敛.

基于生成树的思想, Lin 等人<sup>[24]</sup>利用 Nearest-Better 聚类策略将整个种群划分为多个小生境,并采用平衡策略调整每个小生境的大小,以避免过多个体被归入同一个小生境;同时基于小生境中的关键个体提出了两种新的变异算子:“DE/keypoint/1”和“DE/keypoint/2”,能够有效处理峰值分布密集的 MMOPs.为了更好地平衡种群的探索能力和开发能力, Yang 等人<sup>[31]</sup>将基于估计分布的算法与动态调整聚类规模的策略相结合:在小生境范围内交替使用高斯分布和柯西分布生成后代,同时对聚类规模进行动态调整,以降低聚类结果对参数的敏感性.同样考虑到算法对小生境参数的敏感性, Wang 等人<sup>[32]</sup>提出了一种新的基于吸引力传播的聚类方法,并设计了一种新的自动小生境差分进化算法用于求解 MMOPs,在该算法中,使用吸引力传播聚类的方法无需预先确定小生境的数量或大小,从而实现无参数自动聚类,并将种群在搜索空间中分散化以定位到不同峰值区域. Jiang 等人<sup>[33]</sup>将小生境中心的

选择问题转化为 0/1 二元优化问题,并设计了一种双层优化算法用于解决 MMOPs: 在内层利用遗传算法从种群所有个体中选择出最优小生境中心,将一种新的适应度-熵值计算方法作为优化的目标函数;在外层综合利用局部小生境信息和全局种群信息,对个体执行协同变异策略以生成新的子代。

### 2.2.2 基于新算子的多峰优化算法

除上述研究外,许多学者提出基于新算子的多峰优化算法,包括改进的变异算子、交叉算子、选择算子等,旨在更有效地平衡种群的多样性和收敛性。相对于传统算法中的进化算子,这些新算子能够应对更加多样和复杂的 MMOPs。

在新的变异算子方面,Zhang 等人<sup>[22]</sup>引入“DE/current-to-pbest”和“DE/current-to-best”两种新的算子对个体进行变异操作,这两种算子利用了外部存档中的历史信息为当前个体的进化提供方向引导,在一定程度上能够平衡种群的多样性和收敛性。Zhao 等人<sup>[34]</sup>提出了一种自适应变异策略:根据问题的不同维度执行不同的变异算子和操作,以自适应的方式引导当前个体向最靠近自身的峰值移动,从而促进种群的收敛性,定位到更多全局峰值。根据当前个体的最近邻居和邻域内较优个体,Luo 等人<sup>[35]</sup>提出了一种新的粒子群算法,利用聚类结果和当前个体的最优邻居,对传统粒子群算法中的变异算子进行了改进,能够更好地平衡种群的多样性和收敛性。

在新的交叉算子方面,Tutkun<sup>[36]</sup>基于实数编码下的 GA 提出了一种新的交叉算子,该方法利用高斯分布的概率分布函数,对单点交叉方式进行了改进,从而增加种群中个体的多样性,能够有效解决连续 MMOPs。为了解决复杂 MMOPs 中种群收敛性较差的问题,Tuo 等人<sup>[37]</sup>提出了一种新的基于局部调整策略的交叉算子,该方法通过在进化过程中动态调整交叉率,能够使种群在进化前期具有较好的全局探索能力,并在进化后期增强种群的局部开发能力。

在新的选择算子方面,Chen 等人<sup>[38]</sup>在 GA 基础上提出了一种高效的选择算子,模拟了晋级竞争的规则和过程,以保护表现良好的个体,淘汰表现差的个体:该算子既不需要排序操作也不需要随机操作,因此具有较低的计算复杂度。为了平衡种群在不同进化阶段的探索能力和开发能力,Yuan 等人<sup>[39]</sup>根据拥挤策略中父代选择和替换的范围,将发生在单个小生境内部的拥挤策略称为局部选择,发生在

整个种群中的拥挤策略称为全局选择,通过结合这两种选择操作,实现对种群中个体的自适应动态选择。基于种群中的离群点个体,Zhao 等人<sup>[40]</sup>提出了一种新的选择算子,该方法综合利用个体适应值和个体分布信息来选择子代个体,从而增加种群中个体的多样性以定位到更多全局峰值。

### 2.2.3 基于新框架的多峰优化算法

随着研究的不断深入,越来越多的学者开始借鉴新框架或结合其他研究领域的方法来解决 MMOPs。学者们积极探索新思路和新技术,为解决现实世界中复杂的昂贵 MMOPs 提供了更合理可行的算法。

考虑到 MMOPs 和多目标优化问题的相似性,Wang 等人<sup>[41]</sup>将 MMOPs 转化为一种双目标问题进行求解,即 MMOPs 的最优解对应于多目标优化问题中的帕累托最优解,从而利用多目标优化算法同时定位到 MMOPs 的多个全局最优解。Chen 等人<sup>[42]</sup>提出了一种新的多峰分布式框架,该方法将每个个体作为一个分布式单元,并利用虚拟种群辅助变异操作,每个单元拥有一个虚拟种群用来辅助其进化,从而充分探索搜索空间,并维持种群的多样性以定位多个全局峰值。为解决昂贵 MMOPs,Gao 等人<sup>[43]</sup>提出了一种基于径向基函数的优化算法,通过采用子区域检测策略引导种群向全局最优解移动:在子区域检测策略中,采用 Mean-Shift 聚类方法来预测昂贵 MMOPs 的有潜力子区域;在局部搜索策略中,为每个具有潜力的子区域构建一个局部径向基函数代理模型,从而能够在每个子区域中更精准地定位全局最优值。考虑到昂贵 MMOPs 的复杂性, Ji 等人<sup>[44]</sup>将多代理模型与多任务优化相结合,同时使用克里金模型、径向基函数模型和多项式回归模型,降低了昂贵 MMOPs 的计算复杂度:将昂贵 MMOPs 转化为多任务优化问题,设计了基于技能因子和聚类的代理模型管理策略,有效平衡昂贵 MMOPs 中函数评估次数和预测精度之间的矛盾。

本文从小生境技术和新算子的角度出发,基于局部时空原理处理 MMOPs,主要包括以下内容:在 RDM 策略中基于邻域划分小生境,利用局部空间中的变异操作使得种群中的个体不断搜索靠近全局最优解;在 LCS 策略中,利用局部时间原理记录个体的进化方向,并根据该方向进一步设计了选择算子,从而提高种群的收敛能力。

## 2.3 PID 控制

PID 是自动化研究领域一种经典的、应用广

泛的闭环控制方式,是工业控制中常见的反馈回路<sup>[45]</sup>.在开环控制中,只需要着眼于输入和输出,特点是简单易维护,但系统的鲁棒性较差、动态响应速度不可控.对于闭环控制来说,关键问题是准确、快速地相应系统命令,因此闭环系统中加入了PID控制器,以改造系统的结构特性.PID控制由三部分组成:比例控制(Proportional)、积分控制(Integral)、微分控制(Derivative).图1为系统加入PID控制后的一般过程.

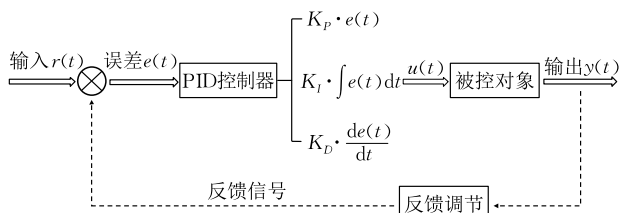


图1 PID闭环控制过程

从图1中可以看出,根据输入信号 $r(t)$ 和误差 $e(t)$ ,PID控制器输出新的信号 $u(t)$ ;结合 $r(t)$ 、 $u(t)$ 和被控对象自身,再计算出系统输出 $y(t)$ .之后继续执行反馈调节完成系统的闭环流程,计算得到新的误差 $e(t+1)$ ,如此循环往复,直到所有采样结束<sup>[46]</sup>.式(5)给出了PID控制的数学表达:

$$u(t) = K_P \cdot e(t) + K_I \cdot \int e(t) dt + K_D \cdot \frac{de(t)}{dt} \quad (5)$$

在一般的PID控制中,当微分作用过大时,可能导致控制器输出产生明显振荡;当积分作用过大或过小时,将直接影响系统稳态误差,因此需要优化调节PID控制器的三个参数 $K_P$ 、 $K_I$ 和 $K_D$ ,使系统更加稳定可靠.鉴于人工参数整定存在工作量大、效率低等问题,利用智能优化算法对PID参数进行整定成为一种趋势<sup>[47]</sup>.

以实际问题中的“自动高炮随动系统”为例,在该系统模型中,整个PID控制系统由位置环、速度环和电流环的三闭环结构形式组成<sup>[48]</sup>.在高炮行进过程中瞄准目标时,路面不平度会导致车体姿态变化,引起高低射角或方向的波动,进而影响高炮射击的精度,而且当车速变化较大或路面状况变化较大时,这种波动将更加频繁.因此,随动系统对PID控制的响应速度和精确度有更高的要求.由于PID控制中参数 $K_P$ 、 $K_I$ 和 $K_D$ 影响着系统控制的偏差和精度,如何优化控制参数成为一个重要问题.

常用的数字PID控制器可以分为位置式PID和增量型PID两种.位置式PID的控制器输出 $u(k)$

与过去所有采样时刻 $k$ 都相关,这种方式下的积分作用会对误差进行累积,当位置式PID的积分饱和后,将导致误差反向变化,此时积分项的存在不再有利,所以在位置式PID中,积分项一般为零.位置式PID表示如式(6):

$$u(k) = K_P \cdot e(k) + K_I \cdot \sum_{i=1}^k e(i) + K_D \cdot [e(k) - e(k-1)] \quad (6)$$

对于增量式PID来说,控制器输出的计算只需要前后三次误差的增量,且没有误差的累加.另一方面,系统发生问题时增量PID不会严重影响系统的工作,当超过值域时只需要输出限幅即可.增量式PID表示如式(7):

$$\begin{aligned} \Delta u(k) &= u(k) - u(k-1) \\ &= K_P \cdot [e(k) - e(k-1)] + K_I \cdot e(k) + K_D \cdot [e(k) - 2 \cdot e(k-1) + 2 \cdot e(k-2)] \quad (7) \end{aligned}$$

综上所述,在给定参数 $K_P$ 、 $K_I$ 和 $K_D$ 的情况下,PID闭环控制的方式通过反馈信号不断调节整个系统,根据目标量和实际输出量的偏差对系统进行合理的控制.这里的参数 $K_P$ 、 $K_I$ 和 $K_D$ 就是所需要的最优控制参数,且一般存在一组或多组最优参数都能使系统性能达到最优,每一组参数对应一个最优解,即PID参数整定问题是一种MMOPs.因此,本文使用多峰优化算法来解决PID控制中的参数整定问题.

### 3 LTDMO

本节将详细介绍本文提出的LTDMO算法,包括RDM策略、LCS策略和SPC策略,并对各个策略的设计思想进行分析说明,然后概括总结LTDMO算法的整体流程和优势,最后给出LTDMO算法的复杂度分析.

#### 3.1 RDM策略

在进化初始阶段,种群中的个体在搜索空间内由随机初始化生成,但这些初始化方法生成的个体并不能全面覆盖搜索空间,这时直接对种群中的个体向着峰值引导并不足够有效,而应该先增加物种的多样性再对个体进行定向引导的变异操作,这样更有利于寻找全局最优解.考虑到随机变异中进化方向的不确定性,这种变异操作能生成更多多样化的个体,因此本文在进化前期使用随机变异来增加物种的多样性,也能有效避免个体陷入局部最优.

基于随机变异的操作,一般从种群中随机选择三个个体进行变异,从而生成一个新的个体向量,这种操作的变异程度更大,但另一方面没有评估用来变异的个体,变异过程比较盲目、无方向性,在进化后期不利于种群在搜索空间中定位全局最优值.因此,本文提出一种 RDM 策略,在进化前期利用随机变异增加种群中个体的多样性,避免种群陷入局部最优;在进化后期划分一定范围的邻域,多个邻域对应多个小生境,多个全局峰值分布在这些小生境中,通过在这个局部邻域空间内的变异操作对个体进行引导,使其以更大概率定位到尽可能多的全局最优解.

在定向引导的邻域变异策略中,首先要计算当前个体和其他个体的欧式距离,并选择最近的  $m$  个个体,再从这  $m$  个个体中随机选择 3 个个体用于变异操作.一般来说,参数  $m$  可以被定义为种群大小的  $1/20 \sim 1/5^{[21]}$ .在这种操作下,种群被划分为多个小生境且变异发生在小生境中,不同的小生境又可能包含不同的峰值,因此种群中的个体能更有效地搜索全局最优解.以图 2 为例,展示了定向引导的变异过程.其中黑色五角星  $J$ 、 $Q$  表示两个全局峰值,红色实心圆  $A$ 、 $B$  分别表示峰值  $J$ 、 $Q$  附近的个体,蓝色实心圆表示种群中的其他个体.图中对个体  $A$ 、 $B$  划分了  $m=5$  的两个邻域小生境,可以看出,分别利用小生境内  $A$ 、 $B$  的邻居进行变异操作能以更大概率定位到当前个体附近的峰值  $J$ 、 $Q$ ,也可以阻止原本靠近峰值的个体在进化过程中发生大幅度转向偏离.

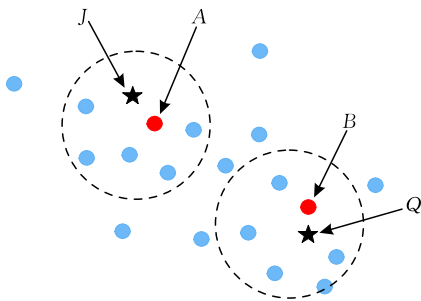


图 2 定向引导变异说明图

综上,为避免求解 MMOPs 时种群中个体陷入局部最优而导致进化停滞的问题,本文提出了 RDM 策略:该策略在进化前期进行基于随机变异的操作增加物种多样性,使种群中个体广泛分布在所有全局峰值的空间内;在进化后期利用邻域变异对个体进行局部空间内的定向引导,在小生境内部进一步

搜索,以定位到多个全局峰值.

### 3.2 LCS 策略

拥挤策略通过小生境技术维持种群中物种多样性,有利于全面搜索最优解.在基于拥挤的策略中,当子代替换了距离其最近的父代时,我们认为从父代到子代的这个方向是有利的、有潜力的,即这个方向上的向量是可以被再次利用的.

以图 3 为例,图中蓝色实心圆  $B$  表示当前子代,蓝色实心圆  $A$  表示距离子代  $B$  最近的父代个体,带箭头的向量  $T$  表示从  $A$  到  $B$  的进化方向,蓝色实心圆  $C$  表示根据向量  $T$  生成的新子代.从图 3 中可以看出, $B$  比  $A$  表现更好更靠近峰值,这也说明从  $A$  到  $B$  的进化方向  $T$  占据优势,能够帮助个体更好地向峰值方向移动.在此基础上产生新的子代  $C$ ,那么  $C$  大概率会朝着峰值移动,促进了种群的收敛,有利于全局寻优.

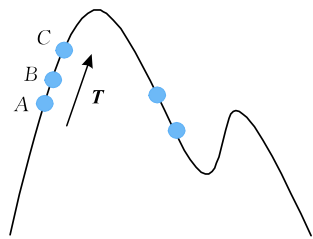


图 3 基于时间局部性的拥挤选择策略说明图

为了更好地利用这个方向向量,提高种群的收敛能力,本文提出基于时间局部性原理的个体进化方法,即 LCS 策略.在计算机系统中,局部性原理分为时间局部性和空间局部性,其中时间局部性的重要应用就是缓存 cache,计算机使用数据时,需要频繁访问主存来读取数据,访问主存是一个相对复杂又慢的过程,而且一段时间内刚刚使用过的数据有很大概率会被再次需要,于是 cache 被用来存储这些被多次需要的数据,以提高计算机的运行速度.同样地,本文基于时间局部性原理,存储了种群中产生的有潜力的进化方向,以便在后续进化过程中对该方向向量加以利用.

参考缓存 cache 的设置方式,我们引入变量  $\delta$  作为算法中时间局部性的体现.在 LCS 策略中, $\delta$  存储每个子代对应的方向向量, $\delta$  的规模和种群大小相同.若当前子代 (*Offspring*) 优于其最近父代 (*NearestParent*) 时, $\delta$  定义如式(8):

$$\text{Offspring}.\delta = \text{Offspring} - \text{NearestParent} + \alpha \cdot \text{NearestParent} \quad (8)$$

式(8)中  $\alpha$  的值小于 0.8 时,该参数值的改变基本不影响算法的性能,0.8 是一个转折点<sup>[49]</sup>,因此本文中参数  $\alpha$  设置为 0.8. 记录了方向向量后,LCS 策略将基于  $\delta$  产生新的子代(*NewOffspring*),定义如式(9):

$$\text{NewOffspring} = \text{Offspring} + \text{Offspring} \cdot \delta \quad (9)$$

由式(9)可以看出,新的子代在当前子代的基础上产生. 然而,新生成的子代并不一定朝峰值移动,也不一定优于当前子代,所以要比较当前子代和新生成子代的适应值,然后保留表现更好的个体进入下一代继续进化.

综上,为提高种群的收敛能力,本文提出了 LCS 策略:该策略在种群进化过程中记录具有潜力的进化方向,并进一步利用这些方向向量继续产生新的子代个体来进行比较和选择. 从而使种群中的个体更有效地靠近全局峰值,在 MMOPs 求解中有利于搜索到更多的高精度全局最优解.

### 3.3 SPC 策略

在处理 MMOPs 的算法中涉及许多参数设置,不同的参数设置会对算法产生不同的影响,本文中变异因子  $F$  和交叉因子  $CR$  就是两个关键参数. 为了更好地调控  $F$  和  $CR$ ,首先研究了它们具体如何影响算法的性能,并对此实施了调研实验. 根据得到的实验数据发现,在某些问题上不同的  $F$  和  $CR$  值对算法性能影响较大.

以图 4 为例,它表示 CEC'2013 测试集上 F8 函数在不同  $CR$  值下的峰值率(Peak Ratio,  $PR$ )变化. 从图 4 中可以看出,当  $CR$  取 0.1 时函数的峰值率可以达到 0.95 以上,表示基本可以找到该函数所有

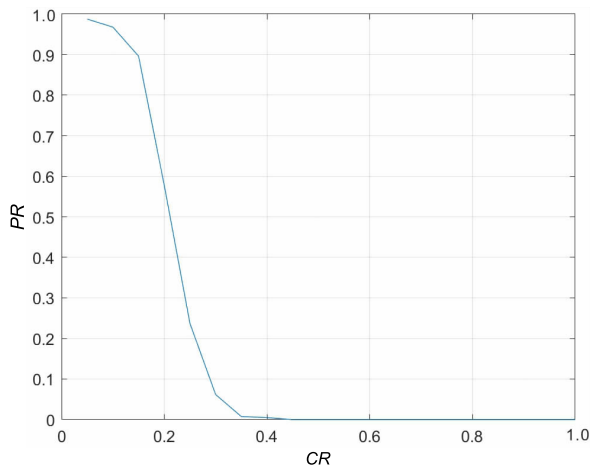


图 4 不同  $CR$  值对 F8 函数性能的影响  
(横坐标表示  $CR$ ,纵坐标表示  $PR$ )

的全局最优解;然而当  $CR$  取 0.5 或更大值时对应的峰值率为 0,即找不到该函数的全局最优解. 考虑到 F8 函数峰值较多且密集,当  $CR$  值较大时子代的信息交换度较大,由此导致个体进化时可能错过附近的峰值;而且过大的  $CR$  值也会导致种群收敛早熟的情况. 因此,较小的  $CR$  在实验中呈现出更好的全局寻优能力.

通过在测试集函数上进行大量实验与数据统计后得到以下结论:

(1)  $CR$  的最适值较小,分布在 0.1 附近; $F$  的最适值分布在 0.5 附近;

(2) 不同测试函数的  $CR$  最适值分布相对集中;而各函数的  $F$  最适值相对分散,可能差别较大.

基于以上论述和实验研究,为了解决关键参数设置对算法产生较大影响的问题,本文提出了一种 SPC 策略,该策略根据不同个体的不同进化阶段对  $CR$  和  $F$  进行动态调控,以降低算法对参数的敏感性,有效提高算法的鲁棒性.

首先,在进化过程的每一次迭代中,对于每个个体  $x_i$ ,分别根据正态分布、柯西分布生成相应的交叉因子  $CR_i$  和变异因子  $F_i$ ,如式(10)和式(11):

$$CR_i = \text{randn}_i(\mu CR, \sigma) \quad (10)$$

$$F_i = \text{randc}_i(\mu F, \gamma) \quad (11)$$

式(10)中  $\mu CR$  表示正态分布的均值, $\sigma$  表示正态分布的标准差;式(11)中  $\mu F$  表示柯西分布的位置参数, $\gamma$  表示柯西分布的尺度参数, $CR_i$  和  $F_i$  均为区间  $(0, 1)$  的值. 其中  $\mu CR$  和  $\mu F$  需要在进化开始之前初始化,并在之后的进化过程中对其进行自适应更新.

根据正态分布的规律可知,大部分数据的取值集中在期望附近,当标准差  $\sigma$  越小,相应的正态值分布越集中,这与  $CR$  最适值集中在 0.1 附近的结论(1)相符合. 在本文中,结合调研和实验结果将初始  $\mu CR$  设置为 0.1, $\sigma$  设置为 0.01,从而使  $CR_i$  取值尽可能靠近  $\mu CR$ . 对于柯西分布,其整体分布情况类似于正态分布,但在分布的两端更加平滑,即更有可能取到两端值,这与  $F$  最适值差别较大的结论(2)相符合. 类似地,将初始  $\mu F$  设置为 0.5, $\gamma$  设置为 0.01.

其次,SPC 策略定义了集合  $S_{CR}$  和  $S_F$  分别用来存储更优子代的  $CR_i$  值和  $F_i$  值,以便于参数  $\mu CR$  和  $\mu F$  的更新. 在选择过程中,若更优子代替换父代,则用集合  $S_{CR}$  和  $S_F$  存储这个子代对应的  $CR_i$  和  $F_i$ . 在每一次迭代结束前, $\mu CR$  和  $\mu F$  以式(12)和式(13)的方式进行更新:

$$\mu CR = (1 - c) \cdot \mu CR + \text{mean}_L(S_{CR}) \quad (12)$$



$$\mu F = (1-c) \cdot \mu F + \text{mean}_L(S_F) \quad (13)$$

式中  $c$  是  $(0, 1)$  区间的值;  $\text{mean}_L$  表示 Lehmer 平均值如式(14), 其中  $S$  表示集合  $S_{CR}$  或者集合  $S_F$ ,  $|S|$  表示集合中元素数量;  $\omega_j$  表示权重系数如式(15), 由个体  $x_j$  与其子代  $u_j$  之间的适应度差值计算得到:

$$\text{mean}_L(S) = \frac{\sum_{j=1}^{|S|} \omega_j S_j^2}{\sum_{j=1}^{|S|} \omega_j S_j} \quad (14)$$

$$\omega_j = \frac{|f(u_j) - f(x_j)|}{\sum_{k=1}^{|S|} |f(u_k) - f(x_k)|} \quad (15)$$

综上, 为降低参数设置对算法性能的影响, 本文提出了 SPC 策略: 该策略对种群中不同个体引入不同的  $CR$  和  $F$ , 并随着个体的不断进化记录历史参数信息, 对  $CR$  和  $F$  进行自适应调整更新, 以此降低算法对于参数设置的敏感性, 使得 LTDMO 算法在解决 MMOPs 时具有更好的鲁棒性。

### 3.4 LTDMO 算法总结

为了更好地理解 LTDMO 算法, 本文将算法整体流程阐述如算法 1 所示, 其中  $MaxFEs$  表示函数的最大评估次数;  $FEs$  表示当前评价次数, 初始值为 0;  $pt$  表示前期随机变异过程在进化总过程中的占比;  $x_i$  表示种群中的个体;  $f(x_i)$  表示个体  $x_i$  的适应值. 可以看到在每次循环迭代中 LTDMO 依次执行了变异、交叉和选择操作, 具体来说, LTDMO 算法首先根据当前进化程度执行基于随机搜索和定向引导的变异策略(步骤 5~10); 其次, 执行交叉操作生成新个体  $u_i$ (步骤 11); 最后, 基于时间局部性原理选择子代个体(步骤 12~20). 同时, LTDMO 算法在进化过程中对参数  $CR$  和  $F$  进行了自适应调整(步骤 4、14、22), 使算法具有更好的鲁棒性. 综上, 本文提出的 LTDMO 算法具有如下优势:

(1) LTDMO 算法中的随机变异操作能增加种群的多样性, 从而避免个体陷入局部最优; 邻域变异操作实现了局部空间中对个体的定向引导, 两种操作相结合能使种群以更好的定位到全局最优解。

(2) 根据时间局部性原理, 在进化过程中 LTDMO 算法记录了当前个体对应的进化方向, 并利用更有潜力的进化方向生成新的子代, 使种群进一步向全局最优解收敛。

(3) 在种群进化过程中, LTDMO 算法根据当前个体的状态信息, 动态调整变异因子  $CR$  和交叉因子  $F$  的参数值, 从而降低算法的参数敏感性、提升算法

的鲁棒性和稳定性。

### 算法 1. LTDMO.

输入: 算法参数  $(N, MaxFEs, \mu CR, \mu F, pt, m, \alpha, \sigma, \gamma, c)$

输出: 进化完成的种群  $P$

1. 随机生成规模为  $N$  的初始化种群  $P$ ;
2. WHILE  $FEs < MaxFEs$
3. FOR  $x_i \in P$
4. 根据式(10)和(11)生成  $CR_i$  和  $F_i$ .
5. IF  $FEs \leq MaxFEs \times pt$
6. 根据式(1)进行随机变异生成  $v_i$ ;
7. ELSE
8. 找到距离  $x_i$  最近的  $m$  个邻居个体;
9. 进行邻域变异生成  $v_i$ ;
10. END IF
11. 根据式(3)执行交叉操作生成  $u_i$ ;
12. 计算距离  $u_i$  最近的父代个体  $x_p$ ;
13. IF  $f(u_i) > f(x_p)$
14. 存储  $CR_i, F_i$  于集合  $S_{CR}, S_F$ ;
15. 根据式(8)计算  $u_i$  对应的  $\delta$ ;
16. 根据式(9)生成新的个体  $w_i$ ;
17. 选择  $u_i$  和  $w_i$  中更优者替换  $x_p$ ;
18. ELSE
19. 保持  $x_p$  不变;
20. END IF
21. END FOR
22. 根据式(12)和(13)更新  $\mu CR$  和  $\mu F$ ;
23. END WHILE

### 3.5 LTDMO 算法复杂度分析

为了方便分析 LTDMO 算法的复杂度, 我们将算法中种群  $P$  的规模表示为  $N$ , 种群的进化代数表示为  $G$ , 变量的维度表示为  $D$ .

由上述内容可知, LTDMO 算法的时间复杂度主要取决于 RDM 策略和 LCS 策略. 在 RDM 策略中, 对种群中每个个体  $x_i$  执行随机变异或者邻域变异; 随机变异操作只需要从种群中随机选择 3 个个体即可, 故时间复杂度为常数  $O(1)$ ; 邻域变异操作需要计算当前个体  $x_i$  与种群中其他个体的距离, 而后根据距离划分  $x_i$  的邻域并从中选择用来变异的个体, 故时间复杂度为  $O(N)$ . 考虑到种群中个体数量为  $N$ , 因此 RDM 策略的时间复杂度为  $O(N) \times O(N)$ , 即  $O(N^2)$ . 在 LCS 策略中, 对于个体  $x_i$  变异交叉后生成的个体  $u_i$ : 首先计算  $u_i$  与所有父代个体的距离, 并由此得到距离其最近的父代个体, 故时间复杂度为  $O(N)$ ; 之后通过比较适应值, 决定是否进一步利用时间局部性对  $u_i$  进行操作, 并由此选择最

优子代,故时间复杂度为常数级  $O(1)$ . 考虑到种群中个体数量为  $N$ ,因此 LCS 策略的时间复杂度为  $O(N) \times O(N+1)$ ,即  $O(N^2)$ . 在算法其他部分中:对每个个体变异后个体  $v_i$ ,在该个体的  $D$  个维度上进行交叉操作,故时间复杂度为  $O(N \times D)$ ;在 SPC 策略中,对种群中每个个体  $v_i$  自适应选择参数,故时间复杂度为  $O(N)$ ,并在每次迭代中更新参数,需要时间复杂度为常数级  $O(1)$ . 综上,考虑种群进化了  $G$  代,LTDMO 算法的时间复杂度为  $O(G) \times O(N^2 + N^2 + N \times D + N + 1)$ ,由于  $D < N$ ,则最终的时间复杂度为  $O(G \times N^2)$ .

对于空间复杂度,主要考虑种群  $P$ 、存储方向向量的数组  $\delta$  以及 SPC 策略中用于记录参数的数组:由于种群规模为  $N$ ,种群中个体维度为  $D$ ,考虑到进化时变异、交叉、选择过程中分别需要不同的数组来存储种群中的个体变量,因此空间复杂度为  $O(3 \times N \times D)$ ;利用时间局部性原理将方向向量存储到  $\delta$  数组时,由于方向向量的维度与个体维度一致,且每个个体的方向向量与其自身相关,因此  $\delta$  的空间复杂度为  $O(N \times D)$ ;在 SPC 策略中,需要记录每个个体的  $CR$  与  $F$ 、 $\mu CR$  与  $\mu F$ 、集合  $S_{CR}$  与  $S_F$ ,考虑到它们均为长度为  $N$  的一维数组,因此 SPC 策略所需的空间复杂度为  $O(6 \times N)$ . 综上,LTDMO 的空间复杂度为  $O(3 \times N \times D + N \times D + 6 \times N)$ ,则最终的空间复杂度为  $O(N \times D)$ .

## 4 实验分析

### 4.1 实验设置

本文中所有实验均在 MATLAB 中进行,并使用 CEC'2013 测试集<sup>[50]</sup>验证算法. CEC'2013 是针对 MMOPs 使用最普遍的测试集,它包括 20 个多模态测试函数,函数分别有不同的维度、不同的种群规模等,是一个多元化的函数测试集,用于求解最大化问题.

在此,CEC'2013 测试集中各个函数的基本信息如表 1 所示,包括种群规模( $N$ )、变量维度( $D$ )、最大评估次数( $MaxFEs$ )和全局最优解数量( $NPK$ ),并将算法中的主要参数在本实验中设置如下:RDM 策略中邻域的大小  $m$ (见 3.1 节)设置为  $NP/10$ ,前期随机变异在进化总过程中的占比  $pt$  设置为  $1/8$ . 用于更新  $\mu CR$  和  $\mu F$  的参数  $c$ (见 3.3 节)在本实验中设置为  $0.05$ .

表 1 CEC'2013 测试集中 20 个多峰函数的基本信息

函数	种群规模	维度	最大评估次数	全局最优解数量
F1	80	1	50 000	2
F2	80	1	50 000	5
F3	80	1	50 000	1
F4	80	2	50 000	4
F5	80	2	50 000	2
F6	100	2	200 000	18
F7	300	2	200 000	36
F8	300	3	400 000	81
F9	300	3	400 000	216
F10	100	2	200 000	12
F11	200	2	200 000	6
F12	200	2	200 000	8
F13	200	2	200 000	6
F14	200	3	400 000	6
F15	200	3	400 000	8
F16	200	5	400 000	6
F17	200	5	400 000	8
F18	200	10	400 000	6
F19	200	10	400 000	8
F20	200	20	400 000	8

此外,本文中使用了峰值率  $PR$  和成功率(Success Rate,  $SR$ )来评估算法的性能. 式(16)中  $PR$  表示找到全局最优解的平均百分比, $NPF_i$ 表示第  $i$  次测试时找到全局最优解的数量, $NR$ 表示总的测试次数,在此一次测试是指单独运行一次算法. 式(17)中  $SR$  表示成功运行的百分比,一次成功运行指该函数所有全局最优解均被找到,类似地, $NR$ 表示总测试次数, $NSR$ 表示所有测试中成功运行的次数:

$$PR = \frac{\sum_i^{NR} NPF_i}{NPK \times NR} \quad (16)$$

$$SR = \frac{NSR}{NR} \quad (17)$$

### 4.2 实验结果

#### 4.2.1 LTDMO 与其他算法的对比结果

本实验在 CEC'2013 测试集的每个函数上独立运行 51 次,取平均值作为最终结果,并和其他 11 个算法在  $10^{-4}$  精度下进行了对比,对比算法的参数设置与原文保持一致. 11 个对比算法的相关描述如表 2 所示,实验结果如表 3 所示.

表 2 关于 11 个对比算法的相关描述

算法名称	相关描述
CDE <sup>[19]</sup>	基于拥挤策略的 DE 算法
NCDE <sup>[21]</sup>	基于邻域的拥挤 DE 算法
SDE <sup>[20]</sup>	基于物种分化的 DE 算法
NSDE <sup>[21]</sup>	基于邻域的物种分化 DE 算法
PNPCDE <sup>[51]</sup>	基于父代个体为中心的归一化邻域 DE 算法
LoICDE <sup>[52]</sup>	基于局部信息共享的 DE 算法
Self_CCDE <sup>[29]</sup>	基于聚类方法的自适应 DE 算法
LIPS <sup>[53]</sup>	基于距离和局部信息共享的 PSO 算法
MOMMOP <sup>[41]</sup>	基于多目标优化的多峰优化算法
r2pso <sup>[54]</sup>	基于环状拓扑结构中右侧个体的 PSO 算法
r3pso <sup>[54]</sup>	基于环状拓扑结构及其左右邻居的 PSO 算法

表 3 LTDMO 算法与其他 MMOPs 算法的实验结果对比

函数	LTDMO		CDE		NCDE		SDE		NSDE		PNPCDE	
	PR	SR	PR	SR	PR	SR	PR	SR	PR	SR	PR	SR
F1	<b>1.000</b>	1.000	<b>1.000</b> (≈)	1.000	<b>1.000</b> (≈)	1.000	0.657(+)	0.373	<b>1.000</b> (≈)	1.000	<b>1.000</b> (≈)	1.000
F2	<b>1.000</b>	1.000	<b>1.000</b> (≈)	1.000	<b>1.000</b> (≈)	1.000	0.737(+)	0.529	0.776(+)	0.667	<b>1.000</b> (≈)	1.000
F3	<b>1.000</b>	1.000	<b>1.000</b> (≈)	1.000	<b>1.000</b> (≈)	1.000	<b>1.000</b> (≈)	1.000	<b>1.000</b> (≈)	1.000	<b>1.000</b> (≈)	1.000
F4	<b>1.000</b>	1.000	<b>1.000</b> (≈)	1.000	<b>1.000</b> (≈)	1.000	0.284(+)	0.000	0.240(+)	0.000	<b>1.000</b> (≈)	1.000
F5	<b>1.000</b>	1.000	<b>1.000</b> (≈)	1.000	<b>1.000</b> (≈)	1.000	0.922(+)	0.843	0.745(+)	0.490	<b>1.000</b> (≈)	1.000
F6	0.867	0.000	<b>1.000</b> (-)	1.000	0.305(+)	0.000	0.056(+)	0.000	0.056(+)	0.000	0.537(+)	0.000
F7	0.889	0.000	0.861(+)	0.000	0.873(+)	0.000	0.053(+)	0.000	0.053(+)	0.000	0.874(+)	0.000
F8	<b>1.000</b>	1.000	0.000(+)	0.000	0.002(+)	0.000	0.013(+)	0.000	0.013(+)	0.000	0.000(+)	0.000
F9	0.491	0.000	0.474(≈)	0.000	0.461(≈)	0.000	0.013(+)	0.000	0.006(+)	0.000	0.474(≈)	0.000
F10	<b>1.000</b>	1.000	<b>1.000</b> (≈)	1.000	0.988(≈)	0.863	0.147(+)	0.000	0.098(+)	0.000	<b>1.000</b> (≈)	1.000
F11	<b>0.833</b>	0.200	0.330(+)	0.000	0.727(+)	0.059	0.314(+)	0.000	0.248(+)	0.000	0.667(+)	0.000
F12	<b>0.400</b>	0.000	0.002(+)	0.000	0.253(+)	0.000	0.208(+)	0.000	0.135(+)	0.000	0.002(+)	0.000
F13	0.667	0.000	0.140(+)	0.000	0.667(≈)	0.000	0.297(+)	0.000	0.225(+)	0.000	0.461(+)	0.000
F14	<b>0.667</b>	0.000	0.024(+)	0.000	<b>0.667</b> (≈)	0.000	0.216(+)	0.000	0.190(+)	0.000	0.592(+)	0.000
F15	0.275	0.000	0.005(+)	0.000	0.319(≈)	0.000	0.108(+)	0.000	0.125(+)	0.000	0.258(≈)	0.000
F16	<b>0.667</b>	0.000	0.000(+)	0.000	<b>0.667</b> (≈)	0.000	0.108(+)	0.000	0.170(+)	0.000	0.000(+)	0.000
F17	0.250	0.000	0.000(+)	0.000	0.250(≈)	0.000	0.076(+)	0.000	0.108(+)	0.000	0.000(+)	0.000
F18	0.333	0.000	0.167(+)	0.000	<b>0.500</b> (-)	0.000	0.026(+)	0.000	0.163(+)	0.000	0.150(+)	0.000
F19	0.125	0.000	0.000(+)	0.000	0.148(≈)	0.000	0.105(+)	0.000	0.098(≈)	0.000	0.000(+)	0.000
F20	0.125	0.000	0.000(+)	0.000	<b>0.150</b> (≈)	0.000	0.000(+)	0.000	0.123(≈)	0.000	0.000(+)	0.000
+	(LTDMO 表现更优)		12		5		19		16		12	
-	(LTDMO 表现更差)		1		1		0		0		0	
≈	(无显著差异)		7		14		1		4		8	

函数	LoICDE		Self_CCDE		LIPS		MOMMOP		r2pso		r3pso	
	PR	SR	PR	SR	PR	SR	PR	SR	PR	SR	PR	SR
F1	<b>1.000</b> (≈)	1.000	<b>1.000</b> (≈)	1.000	0.833(+)	0.686	<b>1.000</b> (≈)	1.000	<b>1.000</b> (≈)	1.000	<b>1.000</b> (≈)	1.000
F2	<b>1.000</b> (≈)	1.000	<b>1.000</b> (≈)	1.000	<b>1.000</b> (≈)	1.000	<b>1.000</b> (≈)	1.000	<b>1.000</b> (≈)	1.000	<b>1.000</b> (≈)	1.000
F3	<b>1.000</b> (≈)	1.000	<b>1.000</b> (≈)	1.000	0.961(+)	0.961	<b>1.000</b> (≈)	1.000	<b>1.000</b> (≈)	1.000	<b>1.000</b> (≈)	1.000
F4	0.975(+)	0.902	<b>1.000</b> (≈)	1.000	0.990(+)	0.961	<b>1.000</b> (≈)	1.000	0.670(+)	0.784	0.966(+)	0.863
F5	<b>1.000</b> (≈)	1.000	<b>1.000</b> (≈)	1.000	<b>1.000</b> (≈)	1.000	<b>1.000</b> (≈)	1.000	<b>1.000</b> (≈)	1.000	<b>1.000</b> (≈)	1.000
F6	<b>1.000</b> (-)	1.000	0.942(-)	0.490	0.246(+)	0.000	<b>1.000</b> (-)	1.000	0.388(+)	0.000	0.687(+)	0.000
F7	0.705(+)	0.020	0.884(≈)	0.020	0.400(+)	0.000	<b>1.000</b> (-)	1.000	0.509(+)	0.000	0.434(+)	0.000
F8	0.000(+)	0.000	0.994(≈)	0.882	0.086(+)	0.000	<b>1.000</b> (≈)	1.000	0.000(+)	0.000	0.421(+)	0.000
F9	0.187(+)	0.000	0.459(≈)	0.000	0.108(+)	0.000	<b>1.000</b> (-)	1.000	0.091(+)	0.353	0.127(+)	0.000
F10	<b>1.000</b> (≈)	1.000	<b>1.000</b> (≈)	1.000	0.748(+)	0.000	<b>1.000</b> (≈)	1.000	0.788(+)	0.000	0.850(+)	0.157
F11	0.660(+)	0.000	0.778(+)	0.137	<b>0.974</b> (-)	0.843	0.710(+)	0.040	0.667(+)	0.000	0.650(+)	0.000
F12	0.495(-)	0.000	0.422(-)	0.000	0.574(-)	0.000	<b>0.955</b> (-)	0.600	0.448(≈)	0.000	0.537(-)	0.000
F13	0.510(+)	0.000	0.653(≈)	0.000	<b>0.794</b> (-)	0.176	0.667(≈)	0.000	0.660(≈)	0.000	0.647(≈)	0.000
F14	0.657(≈)	0.000	0.520(+)	0.000	0.644(≈)	0.000	<b>0.667</b> (≈)	0.000	0.403(+)	0.000	0.637(≈)	0.000
F15	0.299(≈)	0.000	0.343(-)	0.000	0.336(-)	0.000	<b>0.618</b> (-)	0.000	0.103(+)	0.000	0.213(+)	0.000
F16	0.556(+)	0.000	0.655(≈)	0.000	0.307(+)	0.000	0.630(≈)	0.000	0.000(+)	0.000	0.431(+)	0.000
F17	0.222(≈)	0.000	0.246(≈)	0.000	0.168(+)	0.000	<b>0.505</b> (-)	0.000	0.000(+)	0.000	0.096(+)	0.000
F18	0.219(+)	0.000	0.337(≈)	0.000	0.098(+)	0.000	<b>0.497</b> (-)	0.000	0.000(+)	0.000	0.100(+)	0.000
F19	0.032(+)	0.000	0.113(≈)	0.000	0.000(+)	0.000	<b>0.230</b> (-)	0.000	0.000(+)	0.000	0.032(+)	0.000
F20	0.126(≈)	0.000	0.024(+)	0.000	0.000(+)	0.000	0.125(≈)	0.000	0.000(+)	0.000	0.078(+)	0.000
+	9		3		13		1		14		13	
-	2		3		4		8		0		1	
≈	9		14		3		11		6		6	

注：“+”、“-”和“≈”分别表示 LTDMO 算法的实验结果比其他对比算法更优、更差和无显著差异。

从表 3 中数据可以看出, LTDMO 算法在大多数测试函数上表现出较好的性能, 在前 10 个函数上能找到大多数全局最优解, 部分函数的 PR 和 SR 达到了 1.000; 在后 10 个函数上, 除高维函数表现略差外, 其余可以找到半数以上的全局最优解。对于

测试集的 20 个函数, LTDMO 算法在大多数一维函数、二维函数和三维函数都有良好的表现, 其 PR 达到 60% 以上, 但 LTDMO 算法在五维及以上的函数中性能表现有些不足。

F1~F5 是五个低维函数, 峰值的数量相对较

少,根据表 3,本文提出的 LTDMO 算法能够找到这五个函数的所有峰值,即  $PR$  和  $SR$  均达到了 1.000,相对于 SDE 算法、NSDE 算法和 LIPS 算法具有较大的优势.  $F6 \sim F10$  这五个函数的全局峰值数量增多,尤其  $F9$  有 216 个全局峰值,导致算法很难定位到所有峰值. 对于这五个测试函数, LTDMO 算法在  $F6$  和  $F7$  上可以找到绝大部分全局峰值,  $PR$  值分别为 0.867、0.889,且在  $F9$  上的性能表现与 NCDE 算法、PNPCDE 算法和 Self\_CCDE 算法无明显差异;值得一提的是,在  $F8$  和  $F10$  上,本文提出的 LTDMO 算法表现最好,能够找到所有全局峰值,  $PR$  和  $SR$  达到了 1.000.

$F11 \sim F20$  由基础函数复合而得到,函数结构较复杂,导致算法在定位全局峰值时更具挑战性. 在  $F11 \sim F15$  这五个函数中, LTDMO 算法可以找到  $F11$ 、 $F13$ 、 $F14$  的大多数全局峰值,且在  $F11$ 、 $F12$ 、 $F14$  上相较于其他 11 个对比算法表现最好,  $PR$  值分别为 0.833、0.400 和 0.667,说明本文提出的算法具有较好的收敛性. 对于  $F15$ ,尽管 LTDMO 算法在该函数上的性能表现不是所有算法中最优,但相比于大多数对比算法具有一定优势,如 SDE 算法、

NSDE 算法、r2pso 算法.  $F16 \sim F20$  是五个高维复杂函数, LTDMO 算法在  $F16$  函数上相较于其他 11 个对比算法来说表现最优,  $PR$  值为 0.667;但对于  $F19$  和  $F20$ , LTDMO 算法的性能表现略有不足,高维且复杂的函数特性使种群中个体收敛性变差,只能找到一个全局最优解.

此外,表 3 中展示了 LTDMO 与其他 11 个对比算法的  $PR$  值比较结果,显著性水平设置为 0.05,说明 LTDMO 在该测试函数上更好、更差或者无显著差异,分别用符号 +、- 和  $\approx$  表示.

#### 4.2.2 进化过程中的种群分布

为了更加直观清楚地了解进化过程,以 CEC'2013 测试集的函数  $F1$ 、 $F10$  和  $F11$  为例,进化过程中不同迭代次数下的个体分布如图 5 所示,图中红色点表示个体.  $F1$  是一维函数且只有两个全局峰值,函数结构比较基础;  $F10$  是一个二维函数,维度低但全局峰值数量更多,且没有局部峰值;  $F11$  是一个二维的复合函数,结构相对复杂. 在图 5 中,  $Gen=0$  表示初始生成的种群分布情况;  $Gen=10$  表示第 10 次迭代后种群分布情况;  $Final\ generation$  表示最后一次迭代后种群分布情况,即消耗完所有评价次数.

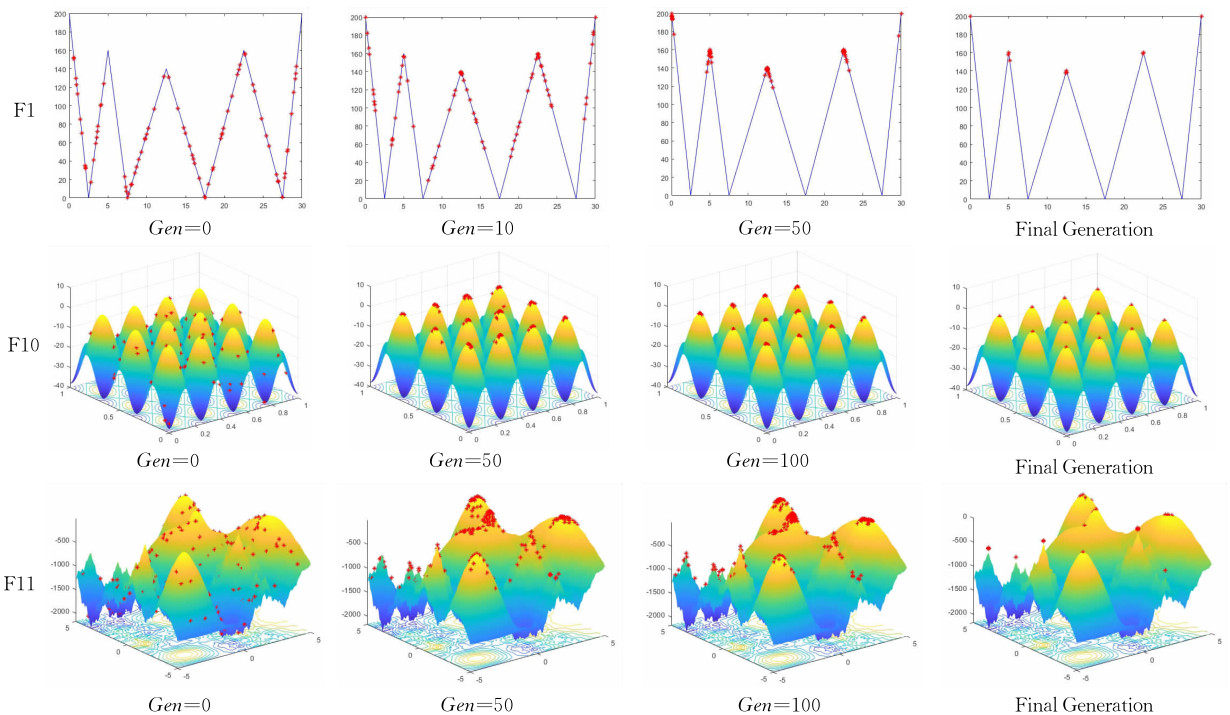


图 5 LTDMO 算法处理  $F1$ 、 $F10$ 、和  $F11$  时在不同迭代次数下的种群中的个体分布图

对于  $F1$ ,本实验记录了  $Gen=0$ 、10、50,  $Final\ generation$  的个体分布情况,当  $Gen=10$  时,  $F1$  的两个全局最优解均已找到,可以看出该寻优过程较为

迅速,种群收敛速度较快;在之后的迭代过程中,所有个体逐渐向全局峰值和局部峰值收敛,最后一次迭代后, LTDMO 算法能定位到所有峰值. 对于  $F10$ ,

本实验记录了  $Gen=0, 50, 100$ , Final generation 的个体分布情况, 当  $Gen=50$  时种群中多数个体已经集中分布在全局峰值附近, 这展示了 LTDMO 算法较好的收敛能力, 当  $Gen=Final\ generation$  时, LTDMO 算法能精确定位到所有全局最优解. 对于 F11, 本实验记录了  $Gen=0, 50, 100$ , Final generation 的个体分布情况, 考虑到该函数结构复杂且具有多个局部峰值, 则寻找全局最优解的过程更加困难,  $Gen=100$  时, 个体在峰值附近的分布趋势更加明显, 相对于 F10 其收敛速度更慢; 当最后一次迭代结束后可以有效定位到大多数全局最优解, 说明 LTDMO 算法能有效避免种群陷入局部最优.

### 4.3 组件分析

本实验部分将 LTDMO 算法、RDM 策略和 LCS 策略进行了对比, 来说明不同策略在 CEC'2013 测试集中 20 个函数上的性能表现. 实验中基本参数设置为  $CR=0.1, F=0.5$ , 其余同第 4.1 节保持一致. 如表 4 所示, LTDMO 算法在 18 个函数上表现最好, 即 LTDMO 算法在 CEC'2013 测试集上的性能明显优于 RDM 策略和 LCS 策略.

表 4 LTDMO 中不同策略的 PR 和 SR 对比

函数	LTDMO		RDM		LCS	
	PR	SR	PR	SR	PR	SR
F1	<b>1.000</b>	1.000	<b>1.000</b>	1.000	<b>1.000</b>	1.000
F2	<b>1.000</b>	1.000	<b>1.000</b>	1.000	<b>1.000</b>	1.000
F3	<b>1.000</b>	1.000	<b>1.000</b>	1.000	<b>1.000</b>	1.000
F4	<b>1.000</b>	1.000	<b>1.000</b>	1.000	0.200	0.000
F5	<b>1.000</b>	1.000	<b>1.000</b>	1.000	<b>1.000</b>	1.000
F6	0.867	0.000	0.389	0.000	<b>1.000</b>	1.000
F7	<b>0.889</b>	0.000	0.867	0.000	0.867	0.000
F8	<b>1.000</b>	1.000	0.825	0.000	0.017	0.000
F9	<b>0.491</b>	0.000	0.454	0.000	0.481	0.000
F10	<b>1.000</b>	1.000	<b>1.000</b>	1.000	<b>1.000</b>	1.000
F11	<b>0.833</b>	0.200	0.667	0.000	0.367	0.000
F12	<b>0.400</b>	0.000	0.100	0.000	0.000	0.000
F13	<b>0.667</b>	0.000	0.633	0.000	0.200	0.000
F14	<b>0.667</b>	0.000	0.667	0.000	0.033	0.000
F15	<b>0.275</b>	0.000	0.250	0.000	0.000	0.000
F16	<b>0.667</b>	0.000	0.633	0.000	0.000	0.000
F17	<b>0.250</b>	0.000	<b>0.250</b>	0.000	0.000	0.000
F18	<b>0.333</b>	0.000	0.300	0.000	0.000	0.000
F19	<b>0.125</b>	0.000	0.125	0.000	0.000	0.000
F20	0.125	0.000	<b>0.150</b>	0.000	0.000	0.000
#Best	18		8		6	

RDM 策略的实验性能与 LTDMO 算法相对接近, 对于 F1~F10, RDM 在前 5 个函数和 F10 上性能最优, 均能找到所有全局峰值, 即 PR 值为 1.000; 但 RDM 在 F6 上的表现是三者中最差的, PR 值仅为 0.389, 而 LTDMO 和 LCS 对应的 PR 值分别为

0.867、1.000. 在 F11~F20 上, RDM 整体表现明显好于 LCS, 这也意味着 RDM 在高维和复杂函数上的全局寻优能力更强. 对于 F12, RDM 相比于 LTDMO 明显较差, 但相比于 LCS 较好, PR 值为 0.100, 表示最多只能找到一个全局峰值. 同时, RDM 在 F17 和 F20 上的表现为三者中最优, PR 值为 0.250 和 0.150.

LCS 策略在 6 个函数上表现最优, 对于 F1~F10, 除 F4 和 F8 外, LCS 在其他函数上的性能表现与 LTDMO 算法基本一致. 值得说的是, LCS 在 F6 上的 PR 值和 SR 值达到了 1.000, 在三者中表现最优. 然而, 在 F4 和 F8 上, LCS 表现明显差于 LTDMO 和 RDM, 几乎找不到全局最优解. 对于 F11~F20, LCS 在 F11、F13 和 F14 上能找到少数全局最优解, 其他函数上的 PR 值为 0. 这说明单独使用 LCS 策略对于处理复杂函数具有局限性, 需要结合其他策略以提高算法在复杂函数上的性能.

因此, RDM 策略和 LCS 策略分别在不同类型的问题上起到了不同的作用, 通过结合 RDM 策略、LCS 策略和 SPC 策略, 构成了强健有效的 LTDMO 算法.

### 4.4 参数分析

结合表 3、表 4 可以看出在复杂高维函数上邻域变异的表现优于随机变异. RDM 策略在进化的前期使用随机变异增加种群多样性, 在后期的进化中使用邻域变异进行定向引导, 而前期和后期在进化总过程中的占比情况将影响 LTDMO 的寻优能力, 本文将前期时间在进化总时间中的占比定义为  $pt$ . 如果前期在进化过程中占比太多, 会导致在高维函数上的寻优能力不足; 相反地, 如果前期占比太少, 会导致算法在部分其他函数上的性能下降. 在此我们将前期时间在总进化过程中的占比(即  $pt$ )分别设置为  $1/8, 1/4$  和  $1/2$ , 并在  $10^{-4}$  精度下进行实验, 结果如表 5 所示.

根据表 5,  $pt$  设置为  $1/8$  时, LTDMO 算法在三者中表现最优.  $pt$  设置为  $1/4$  时, LTDMO 算法在 11 个函数上表现最好;  $pt$  设置为  $1/2$  时, LTDMO 在 10 个函数上表现最好. 在 F1~F5、F10 上, 以上三种方式都能找到所有的全局最优解, 即 PR 值为 1.000; 当前期时间越短时, LTDMO 算法在 F6 的表现越差, 但在 F8 的表现显著更好. 对于 F11~F20, 三种方式设置下 LTDMO 算法在 F14 和 F17 上的性能表现无差异. 同时可以看出当前期时间越长时, 本文提出的算法在复合函数 F11~F20 上的表现越

表 5 不同  $pt$  设置下的 PR 和 SR 对比

函数	$pt=1/8$		$pt=1/4$		$pt=1/2$	
	PR	SR	PR	SR	PR	SR
F1	<b>1.000</b>	1.000	<b>1.000</b>	1.000	<b>1.000</b>	1.000
F2	<b>1.000</b>	1.000	<b>1.000</b>	1.000	<b>1.000</b>	1.000
F3	<b>1.000</b>	1.000	<b>1.000</b>	1.000	<b>1.000</b>	1.000
F4	<b>1.000</b>	1.000	<b>1.000</b>	1.000	<b>1.000</b>	1.000
F5	<b>1.000</b>	1.000	<b>1.000</b>	1.000	<b>1.000</b>	1.000
F6	0.867	0.000	0.878	0.200	<b>0.889</b>	0.400
F7	<b>0.889</b>	0.000	0.872	0.000	0.878	0.000
F8	<b>1.000</b>	1.000	0.854	0.000	0.472	0.000
F9	<b>0.491</b>	0.000	0.469	0.000	0.469	0.000
F10	<b>1.000</b>	1.000	<b>1.000</b>	0.000	<b>1.000</b>	0.000
F11	<b>0.833</b>	0.000	0.700	0.000	0.667	0.000
F12	<b>0.400</b>	2.000	0.325	0.000	0.250	0.000
F13	<b>0.667</b>	0.000	<b>0.667</b>	0.000	0.633	0.000
F14	<b>0.667</b>	0.000	<b>0.667</b>	0.000	<b>0.667</b>	0.000
F15	<b>0.275</b>	0.000	0.250	0.000	<b>0.275</b>	0.000
F16	<b>0.667</b>	0.000	<b>0.667</b>	0.000	0.567	0.000
F17	<b>0.250</b>	0.000	<b>0.250</b>	0.000	<b>0.250</b>	0.000
F18	<b>0.333</b>	0.000	0.233	0.000	0.233	0.000
F19	<b>0.125</b>	0.000	<b>0.125</b>	0.000	0.100	0.000
F20	<b>0.125</b>	0.000	0.125	0.000	0.000	0.000
#Best	19		11		10	

差,尤其对于 F20,  $pt=1/2$  设置下算法的 PR 值为 0.000. 结合表 5 中的数据,  $pt$  设置为  $1/8$  时 LTDMO 算法的表现优于  $pt=1/4$  和  $pt=1/2$ , 因此, 综合考虑下本文将前期随机变异阶段设置为总进化过程的  $1/8$ , 即  $pt=1/8$ , 则后续过程为邻域变异阶段.

#### 4.5 LTDMO 在 PID 控制中的应用

结合第 2.3 节中 PID 控制的背景介绍, 本文选用增量式 PID 控制器进行实验, 实验的被控对象为三阶传递函数, 如式(18), 其中  $sig$  表示信号变量. 采样时间为  $1\text{ms}$ , 对离散系统的正弦信号进行响应并设计 PID 控制器. 因此, LTDMO 算法中对应个体为三维向量, 即  $x=[K_p, K_i, K_d]$ . 目标函数选择“时间乘绝对误差积分(Integrated Time and Absolute Error, ITAE)准则”, 即 ITAE 准则, 误差越小表示个体的适应度越好. 此准则的优点是实用性较强, 在误差积分评价下控制系统的瞬态响应振荡较小, ITAE 准则的数学表达如式(19), 其中  $t$  表示采样时间,  $e(t)$  表示  $t$  时刻的系统误差:

$$\frac{523500}{sig^3 + 87.35 \cdot sig^2 + 10470 \cdot sig} \quad (18)$$

$$\int_0^{\infty} t |e(t)| dt \quad (19)$$

##### 4.5.1 LTDMO 对 PID 参数整定的实验结果

本实验在 MATLAB 中进行, 设置种群规模  $N=300$ , 最大评价次数  $MaxFEs=100000$ , 个体的边

界条件为  $[0, 2]$ , 采样 1000 次, 算法基本参数设置与 4.1 节相同. 在此分别将 LTDMO 算法和 DE 算法用于参数优化和求解, 实验后可求得两个最优解:  $x_1=[0.94492003 \ 0.04765916 \ 2.00000000]$ ;  $x_2=[0.89396173 \ 0.04852951 \ 2.00000000]$ . 实验过程中目标函数值随进化代数的变化如图 6 所示, 其中红色线正方形图例表示 DE 算法对应的目标函数值, 蓝色线圆形图例表示 LTDMO 算法的目标函数值.

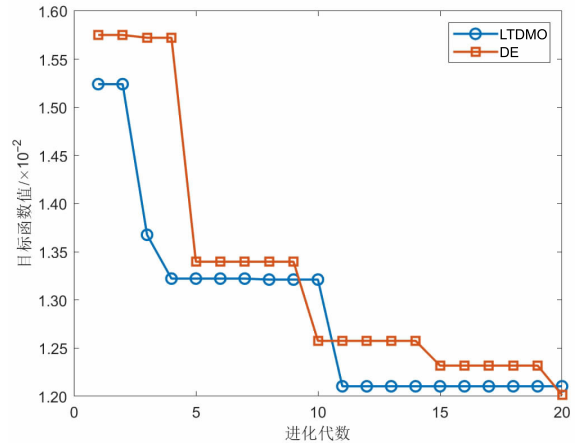
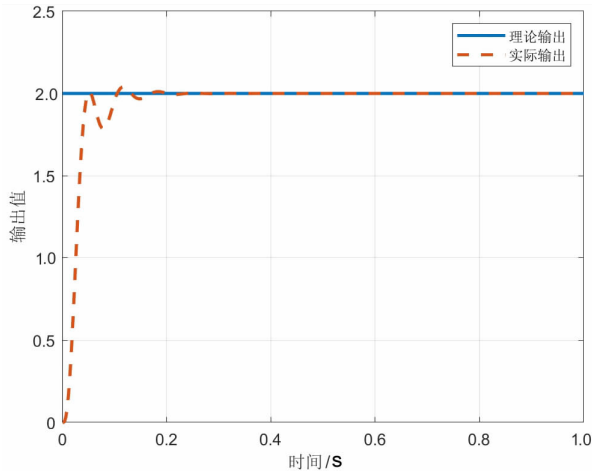


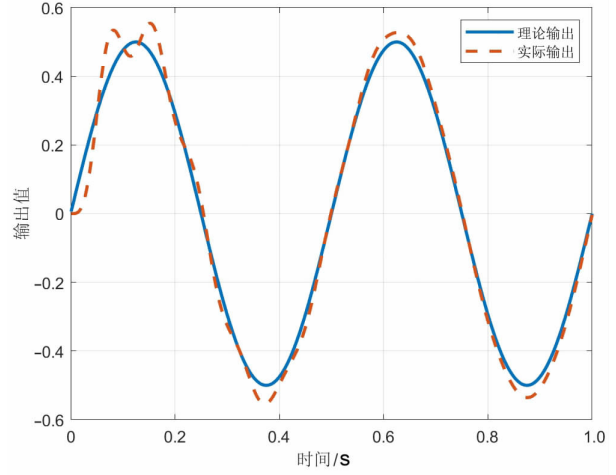
图 6 正弦信号下目标函数值随迭代变化

从图 6 可以看出, 随着进化代数增加, 目标函数值变化呈现先快后慢的变化规律. 在前几次迭代中目标函数值的变化较大, 而在之后的进化过程中目标函数值变化相对缓慢且幅度小, 并逐渐趋于稳定. 在前 20 代的优化过程中, LTDMO 算法对系统的优化速度快于 DE 算法; 整体来看, LTDMO 算法的收敛性好于 DE 算法, LTDMO 算法能够更快地趋于稳定状态; 随迭代次数增加, LTDMO 算法和 DE 算法性能趋于一致.

经过 LTDMO 算法对 PID 参数进行整定后, 可以得到 1000 次采样下控制系统的实际输出值, 我们将其记录下来并绘制了理论输出值与实际输出值的对比图, 如图 7 所示, 包括阶跃信号和正弦信号. 图中横坐标表示采样时间, 采样间隔为  $1\text{ms}$ , 共采样 1000 次, 纵坐标表示输出值; 蓝色实线表示理论输出值, 红色虚线表示系统经过 PID 控制后的实际输出值. 以图 7(b) 正弦信号为例, 开始时系统振荡明显, 系统误差较大; 经过 PID 闭环控制的不断调整后, 在后半部分采样过程中系统误差明显减小, 与标准正弦信号更加贴合, 系统逐渐趋于稳定. 这也反映了 PID 闭环控制的可靠性, 通过闭环反馈使系统响应更准确.



(a) 阶跃信号下输出值对比



(b) 正弦信号下输出值对比

图 7 理论输出值和实际输出值对比

### 4.5.2 进化过程中的种群分布

以正弦信号为例,为了更加直观地表示参数整定过程,我们在个体迭代进化过程中记录种群中的个体分布情况,如图 8 所示.图中  $x$ 、 $y$ 、 $z$  轴分别表示本文需要优化的三个参数  $K_P$ 、 $K_I$  和  $K_D$ ,图中各点表示种群中的个体,个体的不同颜色代表不同的适应度函数值(在此取目标函数的负值),颜色和适应值的对应关系如右侧色阶栏所示.

结合图 8 可以看出个体在进化时首先向系统误差较大的局部最优解移动,在第 30 次迭代时可以看出大致的收敛趋势;在进化到第 50 代时收敛趋势更加明显,且仍有一些较为分散的个体存在;在第 100 次迭代时进化基本完成,个体分布情况较为清晰.结合上文参数优化结果,实验中找到全局最优解位于图 8(f)中左侧褐色条状区域的上方,其余部分为局部峰值区域的个体聚集.

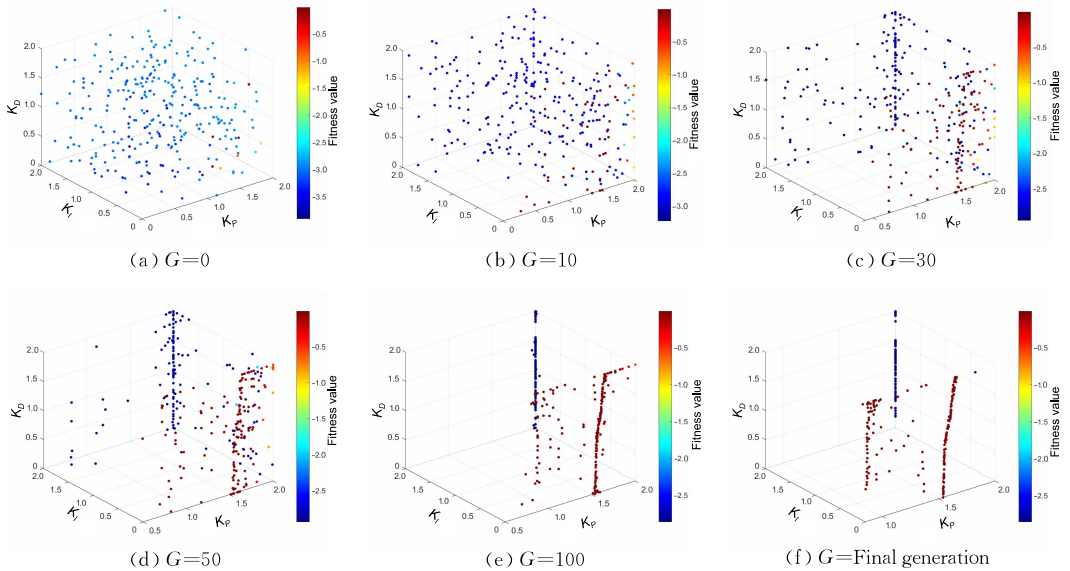


图 8 正弦信号下进化过程中的个体分布图

## 5 总 结

本文针对传统进化算法目前面临的挑战和问题,提出了 LTMDO 算法以更好地解决 MMOPs,并在 PID 闭环控制的参数整定问题中进行了实验验证.

首先,为了避免种群中的个体陷入局部最优,本文提出了一种 RDM 策略以增加种群的多样性,并对个体进行定向引导,从而更有效地寻找全局最优解.其次,为了使个体更好地朝着峰值收敛,本文提出了一种 LCS 策略,这种策略利用时间局部性记录有潜力的进化方向,进一步生成新的个体,通过比较

个体适应值并选择更优个体进入下一代。最后,本文对变异因子  $F$  和交叉因子  $CR$  进行了参数调研,结合调研结果和统计分析,提出了一种 SPC 策略,从而降低算法的参数敏感性,使算法具有更强的鲁棒性和稳定性。在 PID 参数整定问题上,通过实验分析可知 LTDMO 算法能够有效找到对应的最优控制参数,使系统达到稳定状态且误差更小。

同时也存在一些问题需要进一步研究,算法在高维复杂函数上的性能有待提升,需要针对高维空间的求解设计更有效的方法和策略。另一方面,在 PID 控制的参数整定实验中,可以设计更复杂的控制问题进行优化实验,从而更好地说明 LTDMO 算法在解决 PID 参数整定问题上的优势。

**致 谢** 感谢为本文提出意见和建议的各位专家,使本文的质量得到极大提升!

## 参 考 文 献

- [1] Hu Y, Qu B, Wang J, et al. Short-term load forecasting using multimodal evolutionary algorithm and random vector functional link network based ensemble learning. *Applied Energy*, 2021, 285: 116415
- [2] Li X, Eritropakis M G, Deb K, et al. Seeking multiple solutions: An updated survey on niching methods and their applications. *IEEE Transactions on Evolutionary Computation*, 2017, 21(4): 518-538
- [3] Chen D, Li Y. A development on multimodal optimization technique and its application in structural damage detection. *Applied Soft Computing*, 2020, 91(1): 106264
- [4] Bian Q, Nener B, Wang X. A quantum inspired genetic algorithm for multimodal optimization of wind disturbance alleviation flight control system. *Chinese Journal of Aeronautics*, 2019, 32(11): 2480-2488
- [5] Liang Zheng-Ping, Liu Cheng, Wang Zhi-Qiang, et al. Large-scale multi-objective optimization algorithm based on archive and weight extension. *Chinese Journal of Computers*, 2022, 45(5): 951-972(in Chinese)  
(梁正平, 刘程, 王志强等. 基于存档和权值扩展的大规模多目标优化算法. *计算机学报*, 2022, 45(5): 951-972)
- [6] Wang Zhu-Rong, Xue Wei, Hei Xin-Hong, et al. The multi-phase particle swarm optimization for solving the capacitated  $p$ -median problem. *Chinese Journal of Computers*, 2020, 43(6): 1139-1160(in Chinese)  
(王竹荣, 薛伟, 黑新宏等. 多阶段粒子群优化算法求解容量约束  $p$ -中位问题. *计算机学报*, 2020, 43(6): 1139-1160)
- [7] Li Jian-Yu, Zhan Zhi-Hui. Expensive multi-objective evolutionary algorithm with multi-objective data generation. *Chinese Journal of Computers*, 2023, 46(5): 896-908 (in Chinese)
- [8] Zhao H, Zhan Z H, Lin Y, et al. Local binary pattern-based adaptive differential evolution for multimodal optimization problems. *IEEE Transactions on Cybernetics*, 2019, 50(7): 3343-3357
- [9] Zhao H, Tang L, Li J R, et al. Strengthening evolution-based differential evolution with prediction strategy for multimodal optimization and its application in multi-robot task allocation. *Applied Soft Computing*, 2023, 139(1): 110218
- [10] Cheng R, Li M, Li K, et al. Evolutionary multiobjective optimization-based multimodal optimization: Fitness landscape approximation and peak detection. *IEEE Transactions on Evolutionary Computation*, 2018, 22(5): 692-706
- [11] Kieffer E, Danoy G, Brust M R, et al. Tackling large-scale and combinatorial bi-level problems with a genetic programming hyper-heuristic. *IEEE Transactions on Evolutionary Computation*, 2020, 24(1): 44-56
- [12] Dorigo M, Gambardella L M. Ant colony system: A cooperative learning approach to the traveling salesman problem. *IEEE Transactions on Evolutionary Computation*, 1997, 1(1): 53-66
- [13] Birattari M, Pellegrini P, Dorigo M. On the invariance of ant colony optimization. *IEEE Transactions on Evolutionary Computation*, 2007, 11(6): 732-742
- [14] Liang D, Zhan Z H, Zhang Y, et al. An efficient ant colony system approach for new energy vehicle dispatch problem. *IEEE Transactions on Intelligent Transportation Systems*, 2020, 21(11): 4784-4797
- [15] Kennedy J, Eberhart R. Particle swarm optimization// *Proceedings of the IEEE International Conference on Neural Networks*. Perth, Australia, 1995: 1942-1948
- [16] Wei H L, Isa N. An adaptive two-layer particle swarm optimization with elitist learning strategy. *Information Science*, 2014, 273(3): 49-72
- [17] Ji X, Zhang Y, Gong D, et al. Dual-surrogate assisted cooperative particle swarm optimization for expensive multimodal problems. *IEEE Transactions on Evolutionary Computation*, 2021, 25(4): 794-808
- [18] Wang Dong-Feng, Meng Li, Zhao Wen-Jie. Improved bare bones particle swarm optimization with adaptive search center. *Chinese Journal of Computers*, 2016, 39(12): 2652-2667(in Chinese)  
(王东风, 孟丽, 赵文杰. 基于自适应搜索中心的骨干粒子群算法. *计算机学报*, 2016, 39(12): 2652-2667)
- [19] Thomsen R. Multimodal optimization using crowding-based differential evolution// *Proceedings of the IEEE Congress on Evolutionary Computation*. Portland, USA, 2004: 1382-1389
- [20] Li X. Efficient differential evolution using speciation for multimodal function optimization// *Proceedings of the Genetic and Evolutionary Computation Conference*. Washington, USA, 2005: 873-880



- [21] Qu B Y, Suganthan P N, Liang J J. Differential evolution with neighborhood mutation for multimodal optimization. *IEEE Transactions on Evolutionary Computation*, 2012, 16(5): 601-614
- [22] Zhang J, Sanderson A C. JADE: Adaptive differential evolution with optional external archive. *IEEE Transactions on Evolutionary Computation*, 2009, 13(5): 945-958
- [23] Schmidt H, Thierauf G. A combined heuristic optimization technique. *Advances in Engineering Software*, 2005, 36(1): 11-19
- [24] Lin X, Luo W, Xu P. Differential evolution for multimodal optimization with species by nearest-better clustering. *IEEE Transactions on Cybernetics*, 2021, 51(2): 970-983
- [25] Gampeler R, Müller S D, Koumoutsakos P. A parameter study for differential evolution. *Advances in Intelligent Systems, Fuzzy Systems, Evolutionary Computation*, 2002, 10(10): 293-298
- [26] Zhang J, Sanderson A C. An approximate Gaussian model of differential evolution with spherical fitness functions// *Proceedings of the IEEE Congress on Evolutionary Computation*. Singapore, 2007: 2220-2228
- [27] Storn R, Price K. Differential evolution—A simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization*, 1997, 11(4): 341-359
- [28] Tassing R, Guo L, Liu J, et al. Gene sorting in differential evolution with cross-generation mutation. *Science China Information Sciences*, 2011, 54(2): 268-278
- [29] Gao W, Yen G G, Liu S. A cluster-based differential evolution with self-adaptive strategy for multimodal optimization. *IEEE Transactions on Cybernetics*, 2014, 44(8): 1314-1327
- [30] Goldberg D E, Richardson J. Genetic algorithms with sharing for multimodal function optimization// *Proceedings of the International Conference on Genetic Algorithms*. Hillsdale, USA: Lawrence Erlbaum, 1987: 41-49
- [31] Yang Q, Chen W N, Li Y, et al. Multimodal estimation of distribution algorithms. *IEEE Transactions on Cybernetics*, 2017, 47(3): 636-650
- [32] Wang Z J, Zhan Z H, Lin Y, et al. Automatic niching differential evolution with contour prediction approach for multimodal optimization problems. *IEEE Transactions on Evolutionary Computation*, 2019, 24(1): 114-128
- [33] Jiang Y, Zhan Z H, Tan K C, et al. Optimizing niche center for multimodal optimization problems. *IEEE Transactions on Cybernetics*, 2023, 53(4): 2544-2557
- [34] Zhao H, Zhan Z H, Zhang J. Adaptive guidance-based differential evolution with iterative feedback archive strategy for multimodal optimization problems// *Proceedings of the IEEE Congress on Evolutionary Computation*. Glasgow, UK, 2020: 1-8
- [35] Luo W, Qiao Y, Lin X, et al. Hybridizing niching, particle swarm optimization, and evolution strategy for multimodal optimization. *IEEE Transactions on Cybernetics*, 2020, 52(7): 6707-6720
- [36] Tutkun N. Optimization of multimodal continuous functions using a new crossover for the real-coded genetic algorithms. *Expert Systems with Applications*, 2009, 36(4): 8172-8177
- [37] Tuo S, Zhang J, Yuan X, et al. A new differential evolution algorithm for solving multimodal optimization problems with high dimensionality. *Soft Computing*, 2018, 22(1): 4361-4388
- [38] Chen J C, Cao M, Zhan Z H, et al. A new and efficient genetic algorithm with promotion selection operator// *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. Toronto, Canada, 2020: 1532-1537
- [39] Yuan S, Zhao H, Liu J, et al. Self-organizing map based differential evolution with dynamic selection strategy for multimodal optimization problems. *Mathematical Biosciences and Engineering*, 2022, 19(6): 5968-5997
- [40] Zhao H, Zhan Z H, Liu J. Outlier aware differential evolution for multimodal optimization problems. *Applied Soft Computing*, 2023, 140(1): 110264
- [41] Wang Y, Li H X, Yen G G, et al. MOMMOP: Multiobjective optimization for locating multiple optimal solutions of multimodal optimization problems. *IEEE Transactions on Cybernetics*, 2014, 45(4): 830-843
- [42] Chen Z G, Zhan Z H, Wang H, et al. Distributed individuals for multiple peaks: A novel differential evolution for multimodal optimization problems. *IEEE Transactions on Evolutionary Computation*, 2020, 24(4): 708-719
- [43] Gao W, Wei Z, Gong M, et al. Solving expensive multimodal optimization problem by a decomposition differential evolution algorithm. *IEEE Transactions on Cybernetics*, 2023, 53(4): 2236-2246
- [44] Ji X, Zhang Y, Gong D, et al. Multisurrogate-assisted multitasking particle swarm optimization for expensive multimodal problems. *IEEE Transactions on Cybernetics*, 2023, 53(4): 2516-2530
- [45] Zhao C, Guo L. PID controller design for second order nonlinear uncertain systems. *Science China Information Sciences*, 2017, 60(2): 5-17
- [46] Qiao D, Nankun M U, Liao X, et al. Improved evolutionary algorithm and its application in PID controller optimization. *Science China Information Sciences*, 2020, 63(9): 303-305
- [47] Liu Yan-Fei, Peng Zheng, Wang Yi-Hui, et al. PID parameter tuning of brushed direct-current motor based on improved genetic algorithm. *Journal of Computer Applications*, 2022, 42(5): 1634-1641 (in Chinese)  
(刘延飞, 彭征, 王艺辉等. 基于改进的遗传算法的有刷直流电机 PID 参数整定. *计算机应用*, 2022, 42(5): 1634-1641)
- [48] Sun Guo-Xuan, Gong Xin-Yu, Shi Yan, et al. PID parameter tuning of self-propelled antiaircraft gun servo system based on differential evolution algorithm. *Acta Armamentarii*, 2021, 42(5): 903-912 (in Chinese)  
(孙国轩, 宫新宇, 时岩等. 基于差分进化算法的自行高炮伺服系统 PID 参数整定. *兵工学报*, 2021, 42(5): 903-912)

- [49] Wong K C, Wu C H, Mok R, et al. Evolutionary multimodal optimization using the principle of locality. *Information Science*, 2012, 194(1): 138-170
- [50] Li X, Engelbrecht A, Epitropakis M G. Benchmark functions for CEC' 2013 special session and competition on niching methods for multimodal function optimization. Evolutionary Computation and Machine Learning Group, RMIT University, Australia; Technical Report, 2013. Available: <http://titan.csit.rmit.edu.au/~e46507/cec13-niching/competition/cec2013-niching-benchmark-techreport.pdf>
- [51] Biswas S, Kundu S, Das S. An improved parent-centric mutation with normalized neighborhoods for inducing niching behavior in differential evolution. *IEEE Transactions on Cybernetics*, 2014, 44(10): 1726-1737
- [52] Biswas S, Kundu S, Das S. Inducing niching behavior in differential evolution through local information sharing. *IEEE Transactions on Evolutionary Computation*, 2014, 19(2): 246-263
- [53] Qu B Y, Suganthan P N, Das S. A distance-based locally informed particle swarm model for multimodal optimization. *IEEE Transactions on Evolutionary Computation*, 2012, 17(3): 387-402
- [54] Li X. Niching without niching parameters: Particle swarm optimization using a ring topology. *IEEE Transactions on Evolutionary Computation*, 2009, 14(1): 150-169



**ZHAO Hong**, Ph. D., associate professor. Her current research interests include evolutionary computation, swarm intelligence, and their applications in design and optimization.

**LI Jia-Rui**, M.S. candidate. Her current research interests include evolutionary computation, swarm intelligence, and their applications in design and optimization.

**LIU Jing**, Ph.D., professor. Her research interests include evolutionary computation, intelligence optimization, complex networks, and data mining.

## Background

Multimodal optimization problems (MMOPs) have multiple global optima, which aims to locate the optima simultaneously in a single run as many as possible. In real-world, there are many instances involving MMOPs, such as the multirobot task allocation, the structural damage detection and the resource constrained project scheduling. And evolutionary algorithms (EAs) have been widely used to solve MMOPs. In recent years, many novel algorithms based on EAs have also been proposed including varieties of niching techniques, such as crowding, spanning tree clustering and adaptive niching strategy. Additionally, some new operators are designed to further enhance the performance in dealing with MMOPs, such as “DE/current-to-pbest” and “DE/keypoint”. Although these methods have achieved a success in dealing with MMOPs, they are faced with the problems of dropping into local optima and balancing the diversity and convergence of population.

In this paper, a localized time-distance-based multimodal optimization (LTDMO) algorithm is proposed to better deal with MMOPs. Specifically, the advantages of LTDMO include the following three points. (1) The random and direction-based mutation (RDM) strategy is introduced to avoid local optima. In the early stage of evolution, RDM performs a random-based mutation operation to diversify the population. Then in the later stage of evolution, RDM performs a neigh-

borhood-based mutation operation to guide the population explore the search space efficiently. (2) The locality-based crowding selection (LCS) strategy is proposed to accelerate the convergence of the population and approach the global optima by storing temporal locality information of individuals, which can record the evolution direction and make good use of the promising offspring. (3) A self-adaptive parameter control (SPC) strategy is adopted to reduce the sensitivity of parameters, which records the better parameters and changes the parameters dynamically during evolution.

Our team has been committing to the study of intelligence optimization, evolutionary computation and their applications in design and other real-world optimization problems. Up to now, we have published dozens of papers in international journals and conferences, including *IEEE Transactions*, *IEEE Congress on Evolutionary Computation*, *Information Sciences*, *Applied Soft Computing*, etc.

This research was supported in part by the National Natural Science Foundation of China (62306225), the Guangdong Basic and Applied Basic Research Foundation (2021A151511073, 2022A1515011297, 202201011236), in part by the Guangdong High-Level Innovation Research Institution Project (2021B09090-50008), in part by the Guangzhou Key Research and Development Program (202206030003).