

基于目标空间划分的自适应多目标进化算法*

陈黄科¹, 伍国华¹, 霍离俗¹, 戚玉涛²

¹(国防科技大学 系统工程学院, 湖南 长沙 410073)

²(西安电子科技大学 计算机学院, 陕西 西安 710071)

通讯作者: 伍国华, E-mail: guohuawu.nudt@gmail.com



摘要: 目前,多目标进化算法在众多领域具有极高的应用价值,是优化领域的研究热点之一.分析已有多目标进化算法在保持种群多样性方面的不足并提出一种基于解空间划分的自适应多目标进化算法(space division based adaptive multiobjective evolutionary algorithm,简称 SDA-MOEA)来解决多目标优化问题.该方法首先将多目标优化问题的解空间划分为大量子空间,在算法进化过程中,每个子空间都保留一个非支配解集,以保证种群的多样性.另外,该方法根据每个子空间推进种群前进的距离,自适应地为每个子空间分配进化机会,以提高种群的进化速度.最后,利用 3 组共 14 个多目标优化问题检验 SDA-MOEA 的性能,并将 SDA-MOEA 与其他 5 个已有多目标进化算法进行对比分析.实验结果表明:在 10 个问题上,算法 SDA-MOEA 显著优于其他对比算法.

关键词: 空间划分;自适应;多目标优化;进化算法;防洪调度

中图法分类号: TP301

中文引用格式: 陈黄科,伍国华,霍离俗,戚玉涛.基于目标空间划分的自适应多目标进化算法.软件学报,2018,29(9): 2649–2663. <http://www.jos.org.cn/1000-9825/5278.htm>

英文引用格式: Chen HK, Wu GH, Huo LS, Qi YT. Objective space division based adaptive multiobjective optimization algorithm. Ruan Jian Xue Bao/Journal of Software, 2018, 29(9): 2649–2663 (in Chinese). <http://www.jos.org.cn/1000-9825/5278.htm>

Objective Space Division Based Adaptive Multiobjective Optimization Algorithm

CHEN Huang-Ke¹, WU Guo-Hua¹, HUO Li-Su¹, QI Yu-Tao²

¹(College of Systems Engineering, National University of Defense Technology, Changsha 410073, China)

²(School of Computer Science and Technology, Xidian University, Xi'an 710071, China)

Abstract: Currently, multiobjective evolutionary algorithm has been applied widely in various fields, and become one of the most attractive topics in the optimization area. This paper analyzes the deficiency of traditional multiobjective evolutionary algorithms in maintaining population diversity, and further proposes an objective space division based adaptive multiobjective evolutionary algorithm (SDA-MOEA) to solve multiobjective optimization problems. The proposed algorithm divides the objective space of a multiobjective optimization problem into a series of subspaces. During the evolution process, each subspace in SDA-MOEA can maintain a set of non-dominated solutions to guarantee the population diversity. Besides, SDA-MOEA self-adaptively distributes the evolutionary opportunities for each subspace according to its forward distance, which can promote the population convergence. Finally, 14 multiobjective problems of three groups are selected to measure the performance of SDA-MOEA. By comparing with five existing multiobjective evolutionary algorithms, the experimental results demonstrate that SDA-MOEA shows obvious superiority over these existing algorithms on 10 problems.

* 基金项目: 国家自然科学基金(61603404, 61572511); 国防科技大学科研计划(ZK16-03-30)

Foundation item: National Natural Science Foundation of China (61603404, 61572511); Scientific Research Project of National University of Defense Technology (ZK16-03-30)

收稿时间: 2016-10-12; 修改时间: 2016-11-21, 2017-02-14; 采用时间: 2017-03-07; jos 在线出版时间: 2017-03-31

CNKI 网络优先出版: 2017-03-31 21:54:57, <http://kns.cnki.net/kcms/detail/11.2560.TP.20170331.2154.011.html>

Key words: space division; adaptive; multi objective optimization; evolutionary algorithm; flood control scheduling

实际优化问题往往需要处理多个相互冲突的目标,比如水库防洪调度问题^[1]。为了保障上游堤坝的安全,需要最小化上游的水位;而为了下游堤坝的安全,则需要最小化排水量,这两个就是相互冲突的调度目标。云平台中的工作流调度也是典型的多目标优化问题:为了缩短工作流的完成时间,可以租用更多的虚拟机来并行运行工作流任务,这就会增加运行工作流的费用;相反,减少虚拟机的数量可达到减少费用的目的,但会延后工作流的完成时间。很明显,工作流的完成时间与费用是两个相互冲突的优化目标。我们将这类问题称为多目标优化问题(multiobjective optimization problem,简称 MOP)^[2]。根据变量的取值特征,多目标优化问题可粗略分为连续多目标优化问题^[3]和离散多目标优化问题^[4]。本文侧重研究连续多目标优化问题,一般模型如公式(1):

$$\begin{cases} \text{Minimize } F(x) = (f_1(x), f_2(x), \dots, f_m(x)), \\ \text{s.t. } x \in \Omega, \end{cases} \quad (1)$$

其中, $x=(x_1, x_2, \dots, x_n) \in R^n$ 表示一个决策向量, Ω 为决策空间, $F: \Omega \rightarrow R^m$ 由 m 个目标函数 f_1, f_2, \dots, f_m 组成的目标空间。对于连续多目标优化问题, Ω 是空间 R^n 的相连接闭区间;所有的目标函数都是决策向量的连续函数^[5]。

由于多目标优化问题的不同目标相互冲突,一个解在优化某个目标的同时往往是以劣化其他目标为代价,所以不存在单个能够同时最优所有目标的解。一般情况下, MOP 的最优解不是一个解,而是一个被称为 Pareto 最优解的集合。对于两个解 $x, y \in \Omega$, 当且仅当 $f_i(x) \leq f_i(y) \wedge F(x) \neq F(y), \forall i=1, 2, \dots, m$, 我们称 x Pareto 支配 y (记为 $x \prec y$)。如果一个解没有被其他任何解 Pareto 支配,那么该解就是 Pareto 最优解。所有 Pareto 最优解组成的集合被定义为 Pareto 最优解集(Pareto set, 简称 PS), 即 $PS = \{x | \neg y \in \Omega, y \prec x\}$ 。另外, Pareto 最优解集在目标空间的映射被定义为 Pareto 最优前沿(Pareto front, 简称 PF), 即 $PF = \{F(x) | x \in PS\}$ 。

多目标进化算法(multi objective evolutionary algorithm, 简称 MOEA)简单、有效且具有良好的通用性,在求解多目标优化问题时极具吸引力。以 NSGA-II, MOEA/D, MOPSO 等为代表的多目标进化算法在诸多领域得到广泛应用,比如数据挖掘^[6]、云计算中的资源管理和任务调度^[7]、工程项目的调度优化^[8,9]、对地观测卫星的任务规划^[10]。在实际需求和应用的推动下,在过去的 20 多年里,多目标进化算法的研究获得大量关注且取得丰硕的成果,大量相关的算法被提出和改进^[11,12]。根据已有算法的基本思想,这些算法大致可分为 3 大类^[13]: (1) 基于评估指标的 MOEA, 比如算法 IBEA^[14] 和 HypE^[15] 等, 其核心思想是使用一个评估指标来间接评估 Pareto 最优解, 其中, 超体积(hypervolume)是最普遍使用的指标之一; (2) 基于 Pareto 支配关系的 MOEA, 比如 NSGA-II 和 MOPSO 等, 其思想是根据解的 Pareto 支配关系对种群进行分类, 然后优先保留被 Pareto 支配数量少的解, 对于同样被支配数量的解, 再根据它们的密度进行筛选; (3) 基于分解的 MOEA, 比如 MOEA/D, MOEA/D-M2M 和 MOEA/D-ARs 等, 其核心思想是将一个多目标优化问题分解为多个单目标子问题, 然后通过邻域将不同子问题有机组织起来求解多目标优化问题。

对于多目标进化算法,如何在进化的过程中保持种群的多样性极其关键,这将决定种群最终是否能逼近真正的 PF。根据文献[2],连续多目标优化问题的 Pareto 最优解在目标空间是一个分段连续的 $m-1$ 维流形。在进化的过程中,如果种群缺乏多样性,算法可能会忽略一些关键的搜索区域和削弱自身的搜索能力,使得算法的收敛能力变弱甚至无法收敛到真正的 PF。已有的多目标进化算法在种群多样性和收敛性方面都做了大量探讨,并取得丰硕的成果,但是这些算法在保证种群多样性方面仍然存在不足。接下来,我们使用图 1 中的例子分别分析已有的 3 类 MOEA 在保持种群多样性方面存在的不足。图 1 中的例子是根据这几类算法在求解 MOP1-MOP6 系列问题^[16]的进化特征总结而来。

图 1(a)中的 Pareto 前沿是一类常见的前沿轮廓,并不是特例。接下来,使用这样的前沿来分析现有的种群选择方法很难保持种群多样性的原理。对于基于评估指标(超体积)的种群筛选方法,如图 1(b)所示,不失一般性,假设点 A 为参考点,那么由 A-B-C-D-E 围成区域具有最大的超体积。那么,使用这类方法筛选出的种群就只剩点 C 和点 D,如图 1(c)所示。更多关于基于评估指标的种群筛选方法的内容请查阅文献[14]。另外,基于 Pareto 占优关系的种群筛选方法,如图 1(e)所示,根据占优关系将种群划分为多个类,如果前个 K 类(即 F_1, F_2, \dots, F_k)中的个体总

和达到种群大小,那么将舍弃其他类中的个体,如图 1(e)中圆角矩形包含部分.最后筛选出的种群在目标空间的分布如图 1(f)所示.更多关于基于 Pareto 占优关系的种群筛选方法的内容请见文献[17].最后,对于 MOEA/D 中的种群更新策略,如图 1(d)所示.假设子问题 g^4 的邻域为 $B^4=\{2,3,4,5,6\}$,如图 1(d)中的虚线椭圆所示.在进化过程中,如果子问题 g^4 与邻域 B^4 进化出新解 G ,很明显,点 G 目标向量的加权和优于点 F .那么,点 G 将会替代点 F 作为子问题的当前解.随着进化的推进,这样的替代不断发生,筛选出的种群在目标空间的分布也将如图 1(d)所示.文献[16]已经注意到该问题,并对算法 MOEA/D 进行了改进,但仍然无法彻底消除算法在保持种群多样性方面的缺陷.

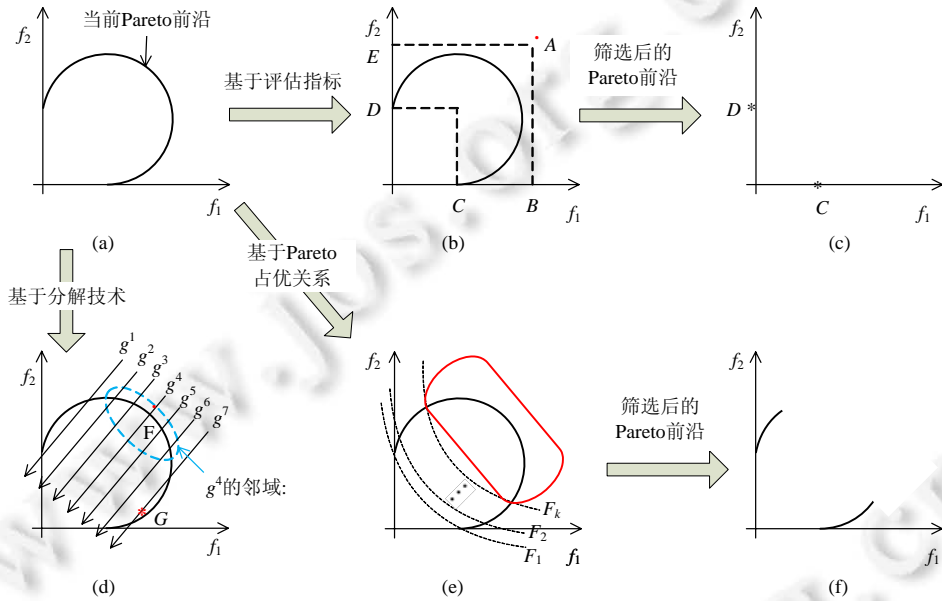


Fig.1 Process of filtering Pareto solution set by three kinds of algorithms

图 1 3 类算法筛选 Pareto 解集的过程

根据以上分析,传统的种群筛选方法在保持种群多样性方面依然存在不足.而大量的研究表明,保持种群的多样性对多目标优化问题的求解起到极其关键的作用.因此,本文将重点研究在 MOEA 进化的过程中如何保持种群的多样性.Cheng 等人的研究^[19]表明:一个适应度差的粒子以适应度好的粒子为引导,也能获得适应度更好的新粒子.该研究成果给我们的启发是:被支配的解与其他支配它的解进行进化操作也可能得到更好的解.基于以上启发,为了保持种群的多样性,适当保留一部分支配解,将有助于种群的进化.

另外,Cai 等人的研究^[4]发现:面向组合优化问题,基于分解的多目标进化算法(MOEA/D)中,不同的子问题更新外部档案库的能力相差很大.然后,Cai 等人根据每个子问题更新外部档案的数量来决定每个子问题进化的概率.该研究成果启发我们:在 MOEA 的进化过程中,种群中的不同个体在推进种群靠近真正 Pareto 前沿方面的能力存在差异.因此,本文研究如何衡量种群中不同个体的能力,并自适应地为种群中的个体分配进化机会.

本文的主要贡献有如下 4 点.

- (1) 提出目标空间划分策略,将目标空间划分为多个子空间,在算法进化过程中,每个子空间都保留一个非支配解集合,以保证种群的多样性;
- (2) 定义了种群的前进距离,并提出一个自适应子空间进化策略,根据每个子空间推进种群前进的距离,自适应地为子空间分配进化机会,以提高种群的收敛速度;
- (3) 基于以上两个策略,提出一个新的多目标进化算法 SDA-MOEA,用于解决多目标优化问题;
- (4) 使用普遍使用的多目标测试问题,对比算法 SDA-MOEA 与其他已有算法的性能.实验结果表明:在 14

个测试问题的 10 个中,算法 SDA-MOEA 的 IGD 指标显著优于其他对比算法。

本文第 1 节通过分析已有多目标进化算法在保持种群多样性方面的缺陷,导出本文的研究动机和内容。第 2 节对已有的多目标进化方法进行总结。第 3 节首先介绍本文提出的目标空间分解策略,并在此基础上介绍本文提出的算法。第 4 节通过对比实验分析本文所提出方法的有效性。最后总结全文,并探讨下一步的研究方向。

1 相关工作

实际优化问题涉及同时处理多个相互冲突的目标,这些问题可建模为多目标优化问题,且被证明是 NP 难问题。传统的精确算法,比如分支定界、分支定价等,很难在可接受的时间内找出问题的最优解。然而,多目标进化算法在解决这些实际应用中展现出良好的性能。在实际需求和应用的推动下,多目标进化算法在过去的 20 多年里得到迅猛的发展,大量的算法被提出或者改进。根据更新种群的思想,这些算法可粗略分为以下 3 类:(1) 基于 Pareto 支配关系;(2) 基于评估指标;(3) 基于分解。

基于支配关系的多目标进化算法,在更新种群的过程中,将当前种群和新产生的种群混合成一个大种群,然后根据种群中个体的支配关系对个体进行排序,然后筛选出新的种群用于下一代的进化。比如,Deb 等人^[17]最先提出非支配快速排序方法来将种群划分为一系列非支配前沿,然后根据前沿的排序顺序筛选出精英解,对于同一个前沿内的解,则根据解的拥挤距离对解进行排序。McClymont 等人^[20]针对已有非支配排序方法的高计算复杂度,提出了两种新非支配排序方法:演绎排序和爬山排序。Zhang 等人^[21]提出一个新的非支配排序方法 ENS 来减少解之间的比较,从而减少排序的时间开销。Coello 等人^[18]使用档案库扩展粒子群算法来解决多目标优化问题。在档案库的更新中,对于新解,只有不被支配才能进入外部档案库;对于档案库中解,如果被新解支配,就被移出档案库。另外,如果档案库中的非支配解超过设定的容量,Coello 等人还应用超网格来筛选非支配解,以保持外部档案库的多样性。戚玉涛等人^[22]提出了一种混合优化算法 HIAEDA,利用 EDA 的模型采样算子学习多目标优化问题决策变量之间的相关性,以提高算法求解问题的能力。胡旺等人^[23,24]使用 Pareto 熵及其差熵监控种群的状态,并根据种群的反馈信息自适应调节粒子群的飞行参数。另外,胡旺等人还引入了格占优和格距离密度来更新外部档案,以提高非支配解的多样性。Cheng 等人^[25]针对外部档案库记录非支配解的有限性,提出了基于高斯过程的逆模型,将目标空间的采样点映射到决策空间,以提高算法的进化性能。Zhang 等人^[26]提出一种基于决策变量聚类的多目标进化算法,并提出一种快速的非支配排序方法,以进一步提高算法的计算效率。李等人^[27]提出一种基于最小生成树的多目标进化算法,根据个体间的支配关系,用树的度数和树聚集密度修剪超出种群规模的解。

基于评估指标的多目标进化算法,使用一个标量指标(比如超体积、逼近性或均匀性)来衡量近似 Pareto 前沿的质量,以引导种群进行搜索和种群的筛选。例如,Zitzler 等人^[14]首先定义了评价指标用于种群的筛选,并提出一种基于通用指标的进化算法(IBE),该算法可根据决策者的偏好信息使用任意评价指标。Beume 等人^[28]混合超体积指标和非支配排序来筛选种群,并提出算法 SMS-EMOA。Emmerich 等人^[29]使用超体积的评估指标来比较进化算法的结果。针对超体积的计算量较大,Brockhoff 等人^[30]提出多个目标缩减策略来降低超体积的计算时间。在后续工作中,Bader 等人^[15]提出使用蒙特卡洛仿真来逼近真实的超体积。Menchaca-Mendez 等人^[31]提出了一种基于逼近性指标和基于欧氏距离的选择机制,以提高种群的多样性。

目前,在多目标进化算法的研究中,基于分解的方法吸引了大量关注。这类算法通过目标向量加权将多目标优化问题转化为大量单目标子优化问题,然后同时优化所有单目标子问题^[32]。比如,周爱民等人^[2]提出了多目标优化算法 MOEA/D-MG,该算法采用混合高斯模型来拟合种群的分布,然后通过采样产生新个体,并集成一个贪婪策略来更新种群。Liu 等人^[16]提出新的基于分解的多目标进化算法 MOEA/D-M2M 来将多目标优化问题转化为一系列简单的多目标优化问题,然后以协同的方式解决这些简单的问题。Wang 等人^[5]发现种群更新策略显著影响 MOEA/D 的多样性和收敛性,然后提出一个全局更新策略,并集成到 MOEA/D 中形成新的算法 MOEA/D-ARs。Cai 等人^[4]将基于支配排序策略集成到基于分解的多目标进化算法中,形成新的算法 EAG-MOEA/D 来解决多目标组合优化问题。Wang 等人^[33]针对如何平衡 MOEA/D 的多样性和收敛性,并提出了一个

约束分解方法,来为每个子问题施加约束.Qi 等人^[1]提出一种新的算法 SaMOEA/D,该算法能够学习种群进化的状态,来自适应调整邻域大小和选择遗传算子.Cai 等人^[34]集成排序选择策略到基于分解的多目标进化算法中,以平衡算法的收敛性和多样性.Ma 等人^[35]提出了基于决策变量分析的多目标进化算法,通过学习变量之间的关联性,将变量划分为多个低维变量集合,然后依次优化每个变量集合.Zhang 等人^[36]提出一种自组织的多目标进化算法,该方法在每一代中建立当前解之间的相邻关系,为个体的进化找出合适的邻域.刘等人^[37]提出一种权重设计方法来辅助 MOEA/D 算法求解一般的多目标优化问题.

这些已有的多目标进化算法在保持种群多样性方面仍然存在不足,具体的分析如图 1 所示.针对该问题,本文提出基于目标空间分解的策略,将目标空间划分为一系列子空间,在算法进化的过程中,每个子空间都保存一个解集,以保持种群的多样性.另外,基于文献[4]的启发,本文将区分不同子空间对种群进化的贡献,并根据每个子空间的贡献,为它们自适应分配进化机会,以提高种群的收敛能力.

2 SDA-MOEA 算法

本节将先介绍基于目标空间分解的策略.在此基础上,详细阐述本文的算法 SDA-MOEA.

2.1 目标空间分解策略

种群的多样性在多目标进化算法的性能上扮演着极其重要的角色.如果在算法进化的过程中种群失去多样性,种群很难最终逼近真实的 Pareto 前沿.从图 1 的总结和分析中可以看出,传统多目标进化算法中的种群更新策略在保持种群多样性方面还存在不足.针对此问题,本文提出一种基于目标空间划分的个体选择策略,将目标空间均匀划分为一系列子空间,在算法进化过程中每个子空间都保留一定数量的解,从而提高种群的多样性.

为了方便理解,我们给出一个两维目标空间的划分实例,如图 2(a)所示.假设多目标优化问题的每个目标 $f_i(x) \in F(x)$ 都满足 $f_i(x) \geq 0$,对于不满足以上条件的目标向量,将其减去理想点的坐标,就可以满足以上条件.本文提出的目标空间划分策略使用从原点发出的 $K-1$ 条射线和坐标轴将目标空间划分为 K 个均匀的子空间 S_1, S_2, \dots, S_K .在算法进化过程中,每个子空间都保留一定数量的解,如图 2(a)中的红星所示.对于一个子空间,其中保留的解都是非支配,如果一个子空间的某些解被该子空间的其他解支配,那么被支配的解将被移除.如图 2(a)中的子空间 S_3 ,其中的解 B 被解 A 支配,那么解 B 将被移除出该子空间.

另外,为了保持种群的多样性,即使一个子空间的所有解都被其他子空间的解支配,这个子空间的解集依然被保留到下一代.如图 2(a)所示,即使子空间 S_{K-1} 的两个解(如红星所示)都被子空间 S_K 的解支配,子空间 S_{K-1} 中的解依然保留在种群中.

对于图 1(a)中的 Pareto 前沿,使用本文提出的目标空间划分策略,种群更新后的目标向量分布将如图 2(b)所示.从图 2(b)中可以看出:目标空间划分策略能够有效保持种群目标向量分布的轮廓,从而有益于保持种群的多样性,并为提高种群的收敛能力做好铺垫.

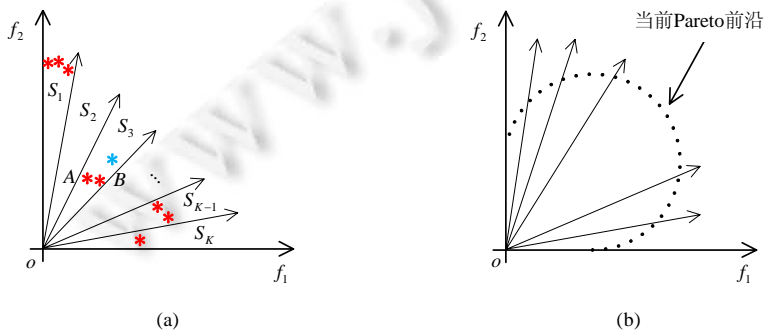


Fig.2 An example of objective space division

图 2 目标空间分解实例

定义 1(子空间). 假设 $u^1, u^2, \dots, u^K \in R_+^m$ 为 K 个单位向量, 那么 m 维目标空间划为 K 个子空间 S_1, S_2, \dots, S_K . 任意一个子空间 $S_i (i=1, 2, \dots, K)$ 定义为 $S_i = \{o \in R_+^m \mid \langle o, u^i \rangle \leq \langle o, u^j \rangle, \forall j=1, 2, \dots, K\}$, 其中 $\langle o, u^i \rangle$ 表示向量 o 与单位向量 u^i 之间的锐角.

根据定义 1, 如果一个目标向量 o 与单位向量 u^i 之间有最小的锐角, 那么该目标向量属于子空间 S_i . 由于一些目标向量可能同时与多个单位向量之间存在最小的锐角, 这样的目标向量就同时属于多个子空间. 另外, 为了均匀划分目标空间, 划分目标空间的单位向量也需要均匀分布. 文献[16]对设置均匀的权重向量深有研究, 并得到广泛的使用. 本文的单位向量设置借鉴这篇文献的权重向量设置方法. 参考 MOEA/D 中邻域子空间的概念, 本文将邻域子空间定义如下:

定义 2(邻域子空间). 一个子空间 $S_i (i=1, 2, \dots, K)$ 的邻域子空间 $B(i)$ 定义一组单位向量与其单位向量锐角最小的 L 个子空间, 即 $B(i) = (i_1, i_2, \dots, i_L)$, 其中 $\langle u^i, u^{i_l} \rangle < \langle u^i, u^j \rangle, \forall l \in B(i), \forall j \in \{1, 2, \dots, K\} \setminus B(i)$.

2.2 算法设计

基于以上目标空间划分策略, 我们提出新的多目标进化算法 SDA-MOEA. 在介绍算法的详细步骤之前, 我们需要定义解的子空间索引来确定一个解与子空间的映射关系以及子空间推进 Pareto 前沿的距离.

定义 3(解的子空间索引). 对于目标向量为 obj 的解, 其子空间的索引 $spaceIndex$ 定义如下:

$$spaceIndex = \arg \max_{1 \leq i \leq K} \frac{(obj - z) \times (u^i)^T}{\|obj - z\|_2 \times \|u^i\|_2} \quad (2)$$

其中, $z = (z_1, z_2, \dots, z_m)$ 为理想点, $z_i \in z$ 表示到目前为止, 第 i 个目标 f_i 的最小值.

实际上, 一个解可能存在多个子空间索引, 本文将从这些子空间索引中为该解随机选择一个子空间索引.

在进化算法的实际应用中, 我们发现在种群进化的不同阶段, 不同子空间对种群进化的效用存在差异. 在算法进化的某些阶段, 一些子空间与其邻域子空间进化出的解频繁推动 Pareto 前沿向真实 Pareto 前沿逼近, 而一些子空间对 Pareto 前沿的推进没有明显的作用. 那么, 如何区分不同子空间对种群的效用也是一个值得研究的问题. 本文使用子空间进化出来的解推动 Pareto 前进的距离来作为子空间对种群的效用. 具体的定义如下:

定义 4(推进距离). 在种群的第 G 代中, 假设子空间 S_i 与其邻域子空间在第 k 次进化后, 得到新解的目标向量为 o_i ; 新解被加入到子空间 S_j 中, 令 S_j 的当前解集的目标向量集合为 $objSet$. 那么, 子空间 S_j 在该次进化中推进前沿的距离 FD_i 定义为

$$FD_i^k = \min_{obj \in objSet} \|obj - z\| - \|o_i - z\| \quad (3)$$

另外, 如果子空间进化出来的新解不能进入任何一个子空间, 那么该子空间推进前沿的距离设为 0.

由于本文提出的算法根据子空间推进前沿的距离, 自适应地为每个子空间分配进化机会, 那么某些子空间在种群的一次迭代中可能获得多次进化机会, 这些子空间推进前沿的距离就是多次推进前沿距离的总和. 如果在种群的第 G 代中, 子空间 S_i 获得 N 次进化机会, 那么子空间 S_i 推进 Pareto 前沿的总距离为 $FD_{i,G} = \sum_{k=1}^N FD_i^k$. 相反, 在种群的一代进化中, 也存在未获得进化机会的子空间, 这些子空间的推进前沿的总距离设为 0.

根据每个子空间的推进距离, 我们接下来定义每个子空间在种群一次迭代中获得进化的概率.

定义 5(子空间进化概率). 对于子空间 S_i 在种群的第 $G+1$ 代进化中获得进化机会的概率 $p_{i,G+1}$ 定义如下:

$$p_{i,G+1} = \frac{\sum_{g=G-R+1}^G (FD_{i,g} + \Delta)}{\sum_{i=1}^K \sum_{g=G-R+1}^G (FD_{i,g} + \Delta)} \quad (4)$$

其中, R 表示算法最大记忆代数, K 表示子空间的总量, $\sum_{g=G-R+1}^G (FD_{i,g} + \Delta)$ 表示子空间在种群最近 R 代中推进 Pareto 前沿的总距离. 为了避免某些子空间因为在种群的最近 R 代进化中没有推进 Pareto 前沿而永远得不到进化机会, 我们使用参数 Δ 来保证每个子空间都有可能获得进化的机会, 参数 Δ 定义如下:

$$\Delta = \frac{1}{10K} \sum_{i=1}^K \sum_{g=G-R+1}^G FD_{i,g} \quad (5)$$

其中, $\frac{1}{K} \sum_{i=1}^K \sum_{g=G-R+1}^G FD_{i,g}$ 表示所有子空间在种群的最近 R 代进化中推进 Pareto 前沿的平均距离。

基于以上定义和分析,算法 SDA-MOEA 的具体步骤见算法 1。

算法 1. SDA-MOEA 算法流程。

输入:多目标优化问题,包括变量维度 n ,变量的定义域;目标维度 m ,和目标函数;最大评估次数 $\max FE$;

输出:一个解矩阵 $PosM$ 及对应的目标矩阵 $ObjM$,其中, $PosM$ 的每行表示一个解, $ObjM$ 中的第 i 行对应 $PosM$ 中第 i 个解的目标向量。

1. 初始化 K 个均匀分布的 m 维单位向量 u^1, u^2, \dots, u^K ,并将目标空间均匀划分为 K 个子空间;
2. 初始化每个子空间非支配解的最大数量为 $\max SoluNum$;
3. 随机生成 $initSize$ 个解向量,记为矩阵 $initPopM_{initSize \times n}$,矩阵的每行表示一个解向量;
4. 获取 $initPopM_{initSize \times n}$ 中每个解的目标向量,记为矩阵 $initObjM_{initSize \times m}$;
5. 初始化评估次数 FES 为 $initSize$,即 $FES=initSize$;并初始化进化代数 $gen=0$;
6. 根据定义 3,将初始解 $initPopM_{initSize \times n}$ 及对应的目标向量分配到对应的目标子空间,以初始化目标子空间;
7. 初始化每个子空间推进前沿的距离都为 1;初始化算法的迭代代次为 $gen=1$;
8. **while (true)** //算法进行迭代
9. **if** $FES \geq \max FE$ **then** //如果评估次数大于或等于最大评估次数
10. 从每个目标子空间中选出一个目标向量及对应的解,构成矩阵 $ObjM$ 和 $PosM$;
11. 输出 $ObjM$ 和 $PosM$,并停止算法 SDA-MOEA;
12. **end if**
13. 初始化一个零矩阵 $newSoluM_{d \times n}=0$,用来记录进化出来的新解; // d 为每代进化获取新解的数量
14. **for** $k=1:d$ **do**
15. 根据每个子空间进化概率,见定义 5,以轮盘赌的方式选出一个子空间 S_i ;
16. 从子空间 S_i 中随机选出一个解,坐标记为 p ;
17. 从子空间 S_i 的邻域子空间随机选择出两个解,分别记为 p_1 和 p_2 ;
18. 进化出的新解为: $newSolu=p+F \cdot (p_1-p_2)$;
19. 随机变异 $newSolu$ 的一个坐标点;
20. 将新解 $newSolu$ 加入到 $newSoluM_{n \times d}$ 中,即 $newSoluM(k,:)=newSolu$;
21. **end for**
22. 评估 $newSoluM$ 中每个新解的目标向量,所有目标向量组成的矩阵记为 $newObjM_{d \times m}$;
23. 更新评估次数为 $FES=FES+d$;
24. 根据定义 3,将新解归类到不同的目标子空间中;
25. 更新每个子空间的解集,并在每个子空间中最多保留 $\max SoluNum$ 非支配解;
26. 根据定义 4,计算每个子空间在 gen 代中推进 Pareto 前沿的距离;
27. 更新进化代数为 $gen=gen+1$;
28. **end while**

算法 1 中首先初始化进化参数(见第 1 行 1~第 7 行);接着,算法不断迭代(见第 8 行 1~第 28 行),直到评估次数达到设定的最大评估次数(见第 9 行 1~第 12 行),在算法的每一代迭代中,根据每个子空间在之前 R 代中推进的 Pareto 距离,有选择地选择 d 个子空间进行进化,得到 d 个新解(见第 14 行 1~第 21 行),值得注意的是:由于子空间的选取是使用轮盘赌的方式选取,在算法一次迭代中,某些子空间可能得到多次迭代机会,而一些子空间可能没得到迭代。

2.3 算法时间复杂度分析

算法 SDA-MOEA 通过轮盘赌的方式来选择一个进化子空间的时间复杂度为 $O(K)$ (见第 15 行),其中, K 为子空间的数量.这是因为产生一个随机数后,遍历所有子空间,就能选出一个子空间.接着,为一个子空间选择两个非支配解的时间复杂度为 $O(L)$ (见第 17 行),其中, L 为子空间的邻域大小.这可解释为该操作需要遍历所有邻域子空间.那么,算法 SDA-MOEA 每次迭代中,产生一个新解集的时间复杂度为 $O(d \times K + d \times L) = O(d \times K)$ (见第 14 行~第 21 行),由于 $K \geq L$.

评估新解集目标向量的时间复杂度为 $O(d)$ (见第 22 行).同样,分配所有新解到对应子空间的时间复杂度也是为 $O(d)$ (见第 24 行).因为以上两个操作都需要遍历所有的新解.算法 SDA-MOEA 中,时间复杂度最高的操作是更新每个子空间的新解集,该操作的时间复杂度是 $O(d^2)$ (见第 25 行).假设 d_i 为分配到子空间 u^i 的新解数量,那么选出子空间 u^i 新解集中非支配解的时间复杂度为 $O(d_i^2 + d_i \times \max \text{SoluNum})$.这是由于新解集中所有新解需要相互进行比较,时间复杂度为 $O(d_i^2)$;另外,所有新解还需要与该子空间的已有解进行相互比较,时间复杂度为 $O(d_i \times \max \text{SoluNum})$.那么,更新所有子空间的解集的时间复杂度为

$$O\left(\sum_{i=1}^K d_i^2 + \max \text{SoluNum} \times \sum_{i=1}^K d_i\right) = O\left(\sum_{i=1}^K d_i^2 + \max \text{SoluNum} \times d\right) \leq O(d^2 + \max \text{SoluNum} \times d) = O(d^2),$$

(见第 25 行).这由于 $O\left(\sum_{i=1}^K d_i^2\right) \leq O\left(\left(\sum_{i=1}^K d_i\right)^2\right) = O(d^2)$ 和 $\max \text{SoluNum} < d$.

综上所述,算法 SDA-MOEA 迭代一次的时间复杂度为 $O(d \times K + d^2)$ (见第 9 行~第 27 行).

3 实验分析

为了检验算法 SDA-MOEA 的性能,本文选择 3 组广泛使用的测试问题:(1) ZDT1-ZDT4 和 ZDT6^[3];(2) DTLZ1 和 DTLZ2^[3,5];(3) MOP1~MOP7^[16].一共 14 个测试问题,其中有 10 个两目标优化问题,4 个三目标优化问题.表 1 描述了这些测试问题的名称、变量维度和目标个数.

Table 1 Test instances

表 1 测试集

问题名称	变量维度	目标个数	问题名称	变量维度	目标个数
ZDT1	50	2	MOP1	10	2
ZDT2	50	2	MOP2	10	2
ZDT3	50	2	MOP3	10	2
ZDT4	10	2	MOP4	10	2
ZDT6	50	2	MOP5	10	2
DTLZ1	10	3	MOP6	10	3
DTLZ2	10	3	MOP7	10	3

本文将算法 SDA-MOEA 与 5 个代表性的多目标进化算法进行对比.这 5 种对比算法分别是:MOEA/D (<http://dces.essex.ac.uk/staff/zhang/web/moea.html>)^[32],MOEA/D-M2M(<http://www.cs.cityu.edu.hk/~qzhang/publications.html>)^[16],MOEA/D-AGR(<http://www.cs.cityu.edu.hk/~qzhang/publications.html>)^[5],NSGA-II(http://www.surrey.ac.uk/cs/people/yaochu_jin/index.htm#publications)^[21]和 MOPSO(<http://delta.cs.cinvestav.mx/~ccoello/EMOO/EMOOsoftware.html>)^[18].这些算法都采用 Matlab 编写,且可以从作者的主页下载.为了方便对比,对于两目标优化问题,种群的规模都设置为 300;对于 3 个目标的优化问题,种群规模设置为 600.由于算法 MOEA/D-AGR 对种群规模比较敏感,其在本文中的实验结果与文献[5]存在一定的差异.算法 SDA-MOEA 的源代码可以通过邮件联系(hkchen@nudt.edu.cn)获取.为了公平起见,对于每个测试问题,每个算法都是以相同的最大评估次数为停止条件.在实验中,本文根据测试问题的难易程度,为每个问题都设置一个最大评估次数.对于测试问题 ZDT1-ZDT3 和 ZDT6,最大评估次数设置为 60 000;对于测试问题 ZDT4,DTLZ1-DTLZ2 和 MOP1-MOP5,最大评估次数设置为 300 000;而对于比较复杂的优化问题 MOP6 和 MOP7,最大评估次数设置为 600 000.除了种群规模和停止条件,对比算法的其他参数直接使用公开代码中的参数设置,具体可参考对比算

法的源代码。

对于两目标优化问题,算法 SDA-MOEA 的子空间数量设置为 300;对于 3 个目标的优化问题,子空间数量设置为 600。每个子空间非支配解的最大数量设置为 5。每个子空间的最大记忆代数 R 设置为 10。为了公平起见,在 SDA-MOEA 达到停止条件时,从每个子空间任意挑选最多一个解来作为输出种群。类似算法 MOEA/D 和 MOEA/D-M2M,算法 SDA-MOEA 的邻域大小设置为 30;DE 操作算子(见算法 1 第 18 行)中参数 F 设置为 0.5,并使用 Polymutation^[2,5]来对每个新解进行变异。

类似于文献[1],本文采用 IGD^[38]作为评价指标。假设 PF 为算法求解出来的目标向量集合, PF^* 为真实 Pareto 前沿均匀分布的目标向量集合。那么,IGD(PF, PF^*)定义如下:

$$\begin{cases} IGD(PF, PF^*) = \frac{\sum_{v \in PF} d(v, PF^*)}{|PF^*|} \\ d(v, PF^*) = \min_{y \in PF^*} \|v - y\| \end{cases} \quad (6)$$

其中, $| \cdot |$ 表示集合中元素的数量, $\| \cdot \|$ 表示向量的模。当算法求解出的 Pareto 前沿越靠近真实 Pareto 前沿,那么 $d(v, PF^*)$ 的值就越小。因此,算法求解出来的 Pareto 前沿,如果其 IGD 值越小,说明算法的性能越好。在实验中:对于 2 个目标问题,我们从 PF^* 中均匀选择 500 个目标向量代表 PF^* ;而对于 3 目标问题,则选择 1 000 个目标向量。

在实验中,对于每个测试问题,每个算法都独立运行 30 次,并以评价指标 IGD 的均值(mean)和方差(std.)作为实验结果。另外,使用 t 检验方法,在可信度为 0.95 的条件下,比较算法的性能,其中符号+, -和 \approx 分别表示对比算法的性能显著优于、劣于和相近于算法 SDA-MOEA。 t 检验方法的主要计算公式如下所示^[41]:

假设 μ_1 和 s_1 分别为算法 1 重复 n_1 次获得的 IGD 均值和标准方差, μ_2 和 s_2 分别为算法 2 重复 n_2 次获得的 IGD 均值和标准方差,那么建立检验假设如下:

$$\begin{aligned} H_0: \mu_1 - \mu_2 &\leq 0, \\ H_1: \mu_1 - \mu_2 &> 0. \end{aligned} \quad (7)$$

假设检验的 t 检验统计量可描述如下:

$$t = \frac{\mu_1 - \mu_2}{\sqrt{s_1^2/n_1 + s_2^2/n_2}} \quad (8)$$

对于 t 分布的自由度,其计算公式如下所示:

$$df = \frac{\left(\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}\right)^2}{\frac{1}{n_1 - 1} \left(\frac{s_1^2}{n_1}\right)^2 + \frac{1}{n_2 - 1} \left(\frac{s_2^2}{n_2}\right)^2} \quad (9)$$

从 t 检验的计算公式可以看出:根据算法 SDA-MOEA 与对比算法的重复次数、IGD 均值和标准方差, t 检验统计量和自由度就可以获取。然后,通过对比 t 分布的临界值表,算法之间的优劣关系就能被确定。

3.1 面向 ZDT 和 DTLZ 系列问题的对比实验结果

表 2 展示了算法 SDA-MOEA 和其他 5 个对比算法求解 ZDT 和 DTLZ 系列测试问题的实验结果。

如表 2 所示:在 7 个测试问题中,本文提出的算法 SDA-MOEA 在 6 个测试问题上具有最好的 IGD;算法 MOEA/D 在 ZDT3 上取得最好的结果;另外,对于测试问题 ZDT3,算法 NSGA-II 的性能仅次于算法 MOEA/D,仍然优于本文提出的算法。值得注意的是:对于测试问题 ZDT1,ZDT2,ZDT4,ZDT6,DTLZ1 和 DTLZ2,算法 SDA-MOEA 的 IGD 均值分别比排序第二的算法低 18.70%,35.41%,123.07%,339.59%,17.94%和 518.85%。表 2 的实验结果表明,算法 SDA-MOEA 在解决 ZDT 和 DTLZ 两组测试问题上具有显著优势。

ZDT 和 DTLZ 系列测试问题的求解难度不高,本文的算法 SDA-MOEA 以及 5 个对比算法都能获得较低的 IGD 值。但是表 2 中的结果显示:对于这 7 个测试问题,算法 SDA-MOEA 在其中 6 个问题取得的 IGD 指标明显优于其他 5 个算法。这可解释为以下两方面:1) 算法 SDA-MOEA 中的目标空间划分策略能够提高种群的多样

性,使得种群在目标空间的分布能够均匀地靠近真正的 Pareto 前沿,从而降低 IGD 值;2) 算法 SDA-MOEA 中自适应策略能够为进化能力更强的子空间分配更多的进化机会,减少无效的进化,从而提高了整个种群推进近似 Pareto 前沿向真正 Pareto 前沿靠近的能力,即提高种群的全局收敛能力。

Table 2 Comparison result on test set ZDT and DTLZ

表 2 面向测试集 ZDT 和 DTLZ 的对比结果

Problems	IGD	SDA-MOEA	MOEA/D	MOEA/D-M2M	MOEA/D-AGR	NSGA-II	MOPSO
ZDT1	Mean	2.209e-03	2.622e-03	3.139e-02	2.028e-01	2.767e-03	2.412e-01
	Std.	8.868e-04	1.194e-04	1.935e-02	6.172e-02	9.805e-05	1.154e-01
	Rank	1	2-	4-	5-	3-	6-
ZDT2	Mean	1.618e-03	2.191e-03	4.962e-02	4.093e-01	2.684e-03	4.215e-01
	Std.	5.385e-04	1.764e-04	3.300e-02	2.051e-01	1.194e-04	2.971e-01
	Rank	1	2-	4-	5-	3-	6-
ZDT3	Mean	5.867e-03	3.838e-03	4.554e-02	3.397e-01	4.042e-03	6.398e-01
	Std.	1.079e-02	6.947e-05	2.744e-02	9.864e-02	1.886e-04	1.826e-01
	Rank	3	1+	4-	5-	2+	6
ZDT4	Mean	6.191e-04	1.381e-03	2.921e-03	1.600e+01	1.589e-03	3.561e+00
	Std.	3.182e-05	9.092e-07	1.719e-03	3.352e+00	2.630e-05	2.147e+00
	Rank	1	2-	4-	6-	3-	5-
ZDT6	Mean	1.725e-02	7.583e-02	1.126e+00	5.672e+00	1.391e+00	6.794e-01
	Std.	1.507e-02	1.072e-02	8.407e-01	2.532e-01	6.900e-02	1.780e+00
	Rank	1	2-	4-	6-	5-	3-
DTLZ1	Mean	8.151e-03	9.613e-03	1.017e+00	7.278e-01	3.122e+01	9.774e+00
	Std.	2.632e-03	2.179e-05	1.196e+00	8.627e-01	3.319e+00	5.027e+00
	Rank	1	2-	4-	3-	6-	5-
DTLZ2	Mean	4.043e-03	2.502e-02	2.946e-02	2.164e-02	2.740e-02	5.588e-02
	Std.	7.015e-05	6.648e-05	1.319e-03	7.685e-05	4.782e-04	9.512e-03
	Rank	1	3-	5-	2-	4-	6-

3.2 面向MOP系列问题的对比实验结果

为了进一步检验算法 SDA-MOEA 的性能,本组实验使用一组 PS 形状复杂的测试集^[16]来对比不同算法的性能.6 种算法的实验结果见表 3.

Table 3 Comparison result on test set MOP

表 3 面向测试集 MOP 的对比结果

Problems	IGD	SDA-MOEA	MOEA/D-DE	MOEA/D-M2M	MOEA/D-AGR	NSGA-II	MOPSO
MOP1	Mean	1.552e-02	3.492e-01	4.497e-02	1.775e-02	3.609e-01	3.342e-01
	Std.	8.248e-04	1.338e-02	1.753e-03	2.888e-03	9.377e-04	5.620e-02
	Rank	1	5-	3-	2-	6-	4-
MOP2	Mean	2.449e-02	3.378e-01	2.102e-02	3.563e-02	3.256e-01	3.480e-01
	Std.	5.190e-02	3.514e-02	3.082e-03	5.137e-02	1.850e-02	2.991e-02
	Rank	2	5-	1≈	3-	4-	6-
MOP3	Mean	3.059e-02	4.900e-01	2.639e-02	4.676e-02	4.119e-01	4.853e-01
	Std.	5.809e-02	4.849e-02	5.062e-03	6.024e-02	6.546e-03	2.138e-01
	Rank	2	6-	1≈	3-	4-	5-
MOP4	Mean	1.0359e-01	3.310e-01	1.080e-01	1.612e-01	4.172e-01	3.122e-01
	Std.	2.488e-03	6.111e-02	1.311e-03	9.047e-03	4.956e-03	1.105e-01
	Rank	1	5-	2	3-	6-	4-
MOP5	Mean	1.326e-02	2.952e-01	3.773e-02	1.752e-02	2.114e-01	3.091e-01
	Std.	9.186e-04	2.110e-02	1.205e-03	3.807e-03	1.993e-02	2.911e-02
	Rank	1	5-	3-	2-	4-	6-
MOP6	Mean	5.590e-02	3.047e-01	7.322e-02	5.158e-02	3.085e-01	3.089e-01
	Std.	1.311e-03	1.956e-02	2.651e-03	2.742e-03	1.577e-06	4.728e-04
	Rank	2	4-	3-	1+	5-	6-
MOP7	Mean	8.639e-02	3.492e-01	1.104e-01	1.764e-01	3.559e-01	3.559e-01
	Std.	2.518e-03	2.565e-02	5.353e-03	4.069e-02	1.454e-06	1.222e-05
	Rank	1	4-	2-	3-	5-	6-

从总体上看:对于 7 个测试问题,在 MOP1,MOP4,MOP5 和 MOP6 这 4 个测试问题上,本文提出的算法 SDA-MOEA 显著优于其他对比算法;在 MOP2 和 MOP3 这两个测试问题上,算法 SDA-MOEA 的 IGD 排序第二,性能稍微逊色于算法 MOEA/D-M2M.根据 Wilcoxon 秩和检验方法,在 0.95 可信度下,算法 SDA-MOEA 和 MOEA/D-M2M 的 IGD 值没有明显的差别.对于测试问题 MOP6,算法 MOEA/D-AGR 展示出最好的性能,而本文的算法 SDA-MOEA 的 IGD 明显大于算法 MOEA/D-AGR.综上所述,在 MOP1~MOP7 系列测试问题上,算法 SDA-MOEA 的性能明显优于另外 5 个对比算法.

测试问题 MOP1~MOP7 很难求解,且在算法进化过程中,种群在目标空间呈现出发散的特征.在求解这类问题的过程中,为了保持种群的多样性,本文提出的目标空间分解策略将目标空间划分为一系列子空间,并在每个子空间中都保留一定数量的非支配解.即使一个子空间的某些解被其他子空间的解支配,这些支配解依然保留在子空间中,这样就能够较好地提高种群的多样性.算法在迭代的过程中,不断更新每个子空间的解,并将整个种群在目标空间的分布不断向真正的 Pareto 前沿推进.而算法 MOEA/D-DE,NSGA-II,MOPSO 在解决这类问题过程中,支配解将被舍弃,导致种群的多样性下降,甚至造成种群出现趋同的情况.以上分析可用来解释本文算法 SDA-MOEA 求解这组测试问题的性能都优于算法 MOEA/D-DE, NSGA-II 和 MOPSO.算法 MOEA/D-M2M 和 MOEA/D-AGR 注意到传统 MOEA/D 在保持种群多样性方面的不足,并提出了种群更新的约束策略来提高种群的多样性和避免种群出现趋同.与以上两个算法相比,算法 SDA-MOEA 中的目标空间划分策略能够更好地保持种群的多样性.另外,算法 SDA-MOEA 根据每个子空间在最近 R 代迭代中推进种群前进的距离,自适应地为每个子空间分配进化机会.那么,推进种群前进能力越强的子空间将获得更多的进化机会,从而提高种群的收敛性.这可以解释算法 SDA-MOEA 在 5 个测试问题上的性能明显优于算法 MOEA/D-M2M,只有在 2 个测试问题中稍微逊色于算法 MOEA/D-M2M.算法 SDA-MOEA 中的自适应策略也可以解释算法 SDA-MOEA 性能明显优于算法 MOEA/D-AGR.

3.3 实际优化问题测试

为了检验本文提出的算法在实际优化问题中的性能,本小节选用水库防洪调度问题来比较 6 种算法性能.水库防洪调度涉及确定每个调度周期内水库的排水速度,并同时优化上游的水位高度和下游的泄洪峰值,以保障水库上下游堤坝的安全.另外,排水速度将受到水库枢纽的最大下泄能力的限制.水库防洪调度问题的数学模型如下所示^[1,22]:

$$\begin{aligned} & \text{Minimize } F(x) = (f_1(x), f_2(x)); \\ & \quad f_1(x) = \max(Z_t); \\ & \quad f_2(x) = \max(x_t); \\ \text{s.t. } & \quad 0 \leq x_t \leq x_{\max}; \\ & \quad Z_{\min} \leq Z_t \leq Z_{\max}; \\ & \quad V_t = V_{t-1} + I_t - x_t; \\ & \quad t = 1, 2, \dots, T; \end{aligned}$$

其中, $x = \{x_1, x_2, \dots, x_T\}$ 为决策变量, $x_t \in x$ 表示第 t 个调度周期水库的下泄流量; T 是总的调度时段数,即决策变量的数量; Z_t 为开始第 t 个调度周期时水库的水位; x_{\max} 指枢纽的最大下泄速度; Z_{\min} 和 Z_{\max} 分别表示水库水位的上下界; V_{t-1} 和 V_t 分别是第 t 个调度周期开始和结束时水库的蓄水量; I_t 是第 t 个调度时段上游的入库流量.由于受地理因素影响,水库底边形状错综复杂,造成水库蓄水量与水位之间的关系极其复杂,很难通过表达式来描述.然而,水库蓄水量与水位之间的关系可通过长期统计数据的分析来获得.

本节选取陕西省安康水库 2010 年 7 月一场洪水的调度问题作为对比案例.该水库的初始水位是 325m,即 $Z_1=325\text{m}$; 水库的水位约束为 300m~330m,即 $Z_{\min}=300\text{m}$ 和 $Z_{\max}=330\text{m}$; 水库排洪系统的最大排水速度为 $37474\text{m}^3/\text{s}$,即 $x_{\max}=37474\text{m}^3/\text{s}$; 整个排洪过程一直延续了 145 个小时,即 $T=145$; 这段时间的洪水入库流量 ($I_t, t=1, 2, \dots, 145$) 曲线如图 3(a) 所示.

由于安康水库 2010 年 7 月的洪水调度问题变量维度高(高达 145 维),每维变量的取值范围较大(在 0~

37474 区间),该问题的真实 Pareto 前沿无法获取,导致无法计算 IGD 指标.在本小节,我们使用另一个广泛用于评估多目标进化算法性能的指标,即超体积.所有算法的最大评估次数为 300 000,其他参数与前面两组实验的设置一样.

为了避免算法名称相互覆盖,在图 3(b)中,我们使用符号 DE,M2M 和 AGR 来分别表示算法 MOEA/D-DE, MOEA/D-M2M 和 MOEA/D-AGR.从图 3(b)中可以看出:在求解 2010 年 7 月安康水库防洪调度问题中,本文算法 SDA-MOEA 获得的超体积值远远大于算法 MOEA/D-DE,MOEA/D-M2M,NSGA-II 和 MOPSO.虽然算法 MOEA/D-AGR 在 30 实验中获得超体积值与算法 SDA-MOEA 存在重叠,但是算法 SDA-MOEA 超体积的平均值比算法 MOEA/D-AGR 高 28.83%.通过实际案例的测试,实验结果表明,本文算法在提高种群多样性和收敛性方面具有良好的性能.

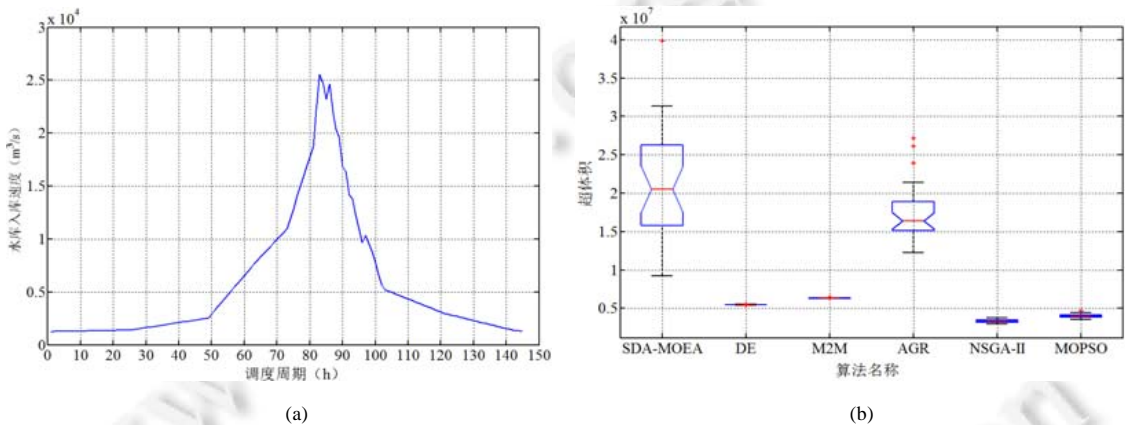


Fig.3 Data of inflow volume on July, 2010 and the box plots of different algorithms

图 3 2010 年 7 月的洪水数据和不同算法超体积的盒图

4 结论和展望

本文针对已有多目标进化算法在保持种群多样性方面的不足,提出了基于目标空间划分策略,将目标空间均匀分为一系列子空间.在算法进化过程中,每个子空间都保留一个解集,以保持种群的多样性.另外,本文根据每个子空间推进 Pareto 前沿靠近真实 Pareto 前沿的能力,自适应地为每个子空间分配进化机会.在以上两个策略的基础上,本文提出多目标进化算法 SDA-MOEA 来解决多目标优化问题.最后,本文选用 14 个广泛使用的测试问题,将算法 SDA-MOEA 与 5 个已有对比算法进行比较.实验结果表明:在 10 个测试问题上,本文提出的算法 SDA-MOEA 明显优于其他对比算法;在两个测试问题上,根据 Wilcoxon 秩和检验结果显示,算法 SDA-MOEA 与最好的对比算法没有明显的差别;只有在另外两个测试问题上,算法 SDA-MOEA 的性能明显劣于已有算法.

在实际的优化问题中,变量的维度高达上千甚至上万,比如云计算中的工作流和任务调度^[42-44].目前,针对如此大规模的多目标优化问题的研究成果还很少.在实际需求推动下,大规模多目标优化的研究必将成为一个热点问题.对于大规模多目标优化问题,保持种群的多样性显得更加重要.在下一步工作中,我们计划改进空间划分策略来解决大规模优化问题.

References:

- [1] Qi YT, Bao L, Ma XL, Miao QG, Li XD. Self-Adaptive multi-objective evolutionary algorithm based on decomposition for large-scale problems: A case study on reservoir flood control operation. *Information Sciences*, 2016,367:529-549. [doi: 10.1016/j.ins.2016.06.005]

- [2] Zhou AM, Zhang QF, Zhang GX. Multiobjective evolutionary algorithm based on mixture Gaussian models. *Ruan Jian Xue Bao/ Journal of Software*, 2014,25(5):913–928 (in Chinese with English abstract). <http://www.jos.org.cn/1000-9825/4514.htm> [doi: 10.13328/j.cnki.jos.004514]
- [3] Li H, Zhang QF. Multiobjective optimization problems with complicated Pareto sets, MOEA/D and NSGA-II. *IEEE Trans. on Evolutionary Computation*, 2009,13(2):284–302. [doi: 10.1109/TEVC.2008.925798]
- [4] Cai XY, Li YX, Fan Z, Zhang QF. An external archive guided multiobjective evolutionary algorithm based on decomposition for combinatorial optimization. *IEEE Trans. on Evolutionary Computation*, 2015,19(4):508–523. [doi: 10.1109/TEVC.2014.2350995]
- [5] Wang ZK, Zhang QF, Zhou AM, Gong MG, Jiao LC. Adaptive replacement strategies for MOEA/D. *IEEE Trans. on Cybernetics*, 2016,46(2):474–486. [doi: 10.1109/TCYB.2015.2403849]
- [6] Mukhopadhyay A, Maulik U, Bandyopadhyay S, Coello CA. A survey of multiobjective evolutionary algorithms for data mining: Part I. *IEEE Trans. on Evolutionary Computation*, 2014,18(1):4–19. [doi: 10.1109/TEVC.2013.2290086]
- [7] Zhan ZH, Liu XF, Gong YJ, Zhang J, Chung H, Li Y. Cloud computing resource scheduling and a survey of its evolutionary approaches. *ACM Computing Surveys (CSUR)*, 2015,47(4):1–63. [doi: 10.1145/2788397]
- [8] Shahsavar A, Najafi AA, Niaki STA. Three self-adaptive multi-objective evolutionary algorithms for a triple-objective project scheduling problem. *Computers & Industrial Engineering*, 2015,87:4–15. [doi: 10.1016/j.cie.2015.04.027]
- [9] Xiong J, Leus R, Yang ZY, Abbass HA. Evolutionary multi-objective resource allocation and scheduling in the Chinese navigation satellite system project. *European Journal of Operational Research*, 2016,251(2):662–675. [doi: 10.1016/j.ejor.2015.11.031]
- [10] Tangpattanakul P, Jozefowicz N, Lopez P. A multi-objective local search heuristic for scheduling earth observations taken by an agile satellite. *European Journal of Operational Research*, 2015,245(2):542–554. [doi: 10.1016/j.ejor.2015.03.011]
- [11] Gutjahr WJ, Pichler A. Stochastic multi-objective optimization: A survey on non-scalarizing methods. *Annals of Operations Research*, 2016,236(2):475–499. [doi: 10.1007/s10479-013-1369-5]
- [12] Mukhopadhyay A, Maulik U, Bandyopadhyay S. A survey of multiobjective evolutionary clustering. *ACM Computing Surveys (CSUR)*, 2015,47(4):1–61. [doi: 10.1145/2742642]
- [13] Zhou A, Qu BY, Li H, Zhao SZ, Suganthan PN, Zhang QF. Multiobjective evolutionary algorithms: A survey of the state of the art. *Swarm and Evolutionary Computation*, 2011,1(1):32–49. [doi: 10.1016/j.swevo.2011.03.001]
- [14] Zitzler E, Künzli S. Indicator-Based selection in multiobjective search. In: *Proc. of the Int'l Conf. on Parallel Problem Solving from Nature*. LNCS 3242. Berlin, Heidelberg: Springer-Verlag, 2004. 832–842. [doi: 10.1007/978-3-540-30217-9_84]
- [15] Bader J, Zitzler E. HypE: An algorithm for fast hypervolume-based many-objective optimization. *Evolutionary Computation*, 2011, 19(1):45–76. [doi: 10.1162/EVCO_a_00009]
- [16] Liu HL, Gu F, Zhang Q. Decomposition of a multiobjective optimization problem into a number of simple multiobjective subproblems. *IEEE Trans. on Evolutionary Computation*, 2014,18(3):450–455. [doi: 10.1109/TEVC.2013.2281533]
- [17] Deb K, Pratap A, Agarwal S, Meyarivan T. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Trans. on Evolutionary Computation*, 2002,6(2):182–197. [doi: 10.1109/4235.996017]
- [18] Coello CAC, Pulido GT, Lechuga MS. Handling multiple objectives with particle swarm optimization. *IEEE Trans. on Evolutionary Computation*, 2004,8(3):256–279. [doi: 10.1109/TEVC.2004.826067]
- [19] Cheng R, Jin Y. A competitive swarm optimizer for large scale optimization. *IEEE Trans. on Cybernetics*, 2015,45(2):191–204. [doi: 10.1109/TCYB.2014.2322602]
- [20] McClymont K, Keedwell E. Deductive sort and climbing sort: New methods for non-dominated sorting. *Evolutionary Computation*, 2012,20(1):1–26. [doi: 10.1162/EVCO_a_00041]
- [21] Zhang X, Tian Y, Cheng R, Jin YC. An efficient approach to nondominated sorting for evolutionary multiobjective optimization. *IEEE Trans. on Evolutionary Computation*, 2015,19(2):201–213. [doi: 10.1109/TEVC.2014.2308305]
- [22] Qi YT, Liu F, Liu JL, Ren Y, Jiao LC. Hybrid immune algorithm with EDA for multi-objective optimization. *Ruan Jian Xue Bao/ Journal of Software*, 2013,24(10):2251–2266 (in Chinese with English abstract). <http://www.jos.org.cn/1000-9825/4314.htm> [doi: 10.13328/j.cnki.jos.004314]
- [23] Hu W, Yen GG. Adaptive multiobjective particle swarm optimization based on parallel cell coordinate system. *IEEE Trans. on Evolutionary Computation*, 2015,19(1):1–18. [doi: 10.1109/TEVC.2013.2296151]

- [24] Hu W, Yen GG, Zhang X. Multiobjective particle swarm optimization based on Pareto entropy. *Ruan Jian Xue Bao/Journal of Software*, 2014,25(5):1025–1050 (in Chinese with English abstract). <http://www.jos.org.cn/1000-9825/4496.htm> [doi: 10.13328/j.cnki.jos.004496]
- [25] Cheng R, Jin Y, Narukawa K, Sendhoff B. A multiobjective evolutionary algorithm using Gaussian process-based inverse modeling. *IEEE Trans. on Evolutionary Computation*, 2015,19(6):838–856. [doi: 10.1109/TEVC.2015.2395073]
- [26] Zhang X, Tian Y, Jin Y, Cheng, R. A decision variable clustering-based evolutionary algorithm for large-scale many-objective optimization. *IEEE Trans. on Evolutionary Computation*, 2016. [doi: 10.1109/TEVC.2016.2600642]
- [27] Li MQ, Zheng JH, Luo B. A multi-objective evolutionary algorithm based on minimum spanning tree. *Journal of Computer Research and Development*, 2009,46(5):803–813 (in Chinese with English abstract). <http://crad.ict.ac.cn/CN/article/searchArticleResult.do>
- [28] Beume N, Naujoks B, Emmerich M. SMS-EMOA: Multiobjective selection based on dominated hypervolume. *European Journal of Operational Research*, 2007,181(3):1653–1669. [doi: 10.1016/j.ejor.2006.08.008]
- [29] Emmerich M, Beume N, Naujoks B. An EMO algorithm using the hypervolume measure as selection criterion. In: *Proc. of the Int'l Conf. on Evolutionary Multi-Criterion Optimization*. Berlin, Heidelberg: Springer-Verlag, 2005. 62–76. [doi: 10.1007/978-3-540-31880-4_5]
- [30] Brockhoff D, Zitzler E. Improving hypervolume-based multiobjective evolutionary algorithms by using objective reduction methods. In: *Proc. of the IEEE Congress on Evolutionary Computation*. 2007. 2086–2093. [doi: 10.1109/CEC.2007.4424730]
- [31] Menchaca-Mendez A, Coello CAC. GD-MOEA: A new multi-objective evolutionary algorithm based on the generational distance indicator. In: *Proc. of the Int'l Conf. on Evolutionary Multi-Criterion Optimization*. Berlin, Heidelberg: Springer-Verlag, 2015. 156–170. [doi: 10.1007/978-3-319-15934-8_11]
- [32] Zhang Q, Li H. MOEA/D: A multiobjective evolutionary algorithm based on decomposition. *IEEE Trans. on Evolutionary Computation*, 2007,11(6):712–731. [doi: 10.1109/TEVC.2007.892759]
- [33] Wang L, Zhang Q, Zhou A, Gong MG, Jiao LC. Constrained subproblems in a decomposition-based multiobjective evolutionary algorithm. *IEEE Trans. on Evolutionary Computation*, 2016,20(3):475–480. [doi: 10.1109/TEVC.2015.2457616]
- [34] Cai X, Yang Z, Fan Z, Zhang Q. Decomposition-Based-Sorting and angle-based-selection for evolutionary multiobjective and many-objective optimization. *IEEE Trans. on Cybernetics*, 2016. [doi: 10.1109/TCYB.2016.2586191]
- [35] Ma XL, Liu F, Qi YT, Wang XD, Li LL, Jiao LC, Yin ML, Gong MG. A multiobjective evolutionary algorithm based on decision variable analyses for multiobjective optimization problems with large-scale variables. *IEEE Trans. on Evolutionary Computation*, 2016,20(2):275–298. [doi: 10.1109/TEVC.2015.2455812]
- [36] Zhang H, Zhou AM, Song SM, Zhang QF, Gao XZ, Zhang J. A self-organizing multiobjective evolutionary algorithm. *IEEE Trans. on Evolutionary Computation*, 2016,20(5):792–806. [doi: 10.1109/TEVC.2016.2521868]
- [37] Liu HL, Gu FQ, Cheung XM. A weight design method based on power transformation for multi-objective evolutionary algorithm MOEA/D. *Journal of Computer Research and Development*, 2012,49(6):1264–1271 (in Chinese with English abstract). <http://crad.ict.ac.cn/CN/abstract/abstract112.shtml>
- [38] Zhang Q, Zhou A, Jin Y. RM-MEDA: A regularity model-based multiobjective estimation of distribution algorithm. *IEEE Trans. on Evolutionary Computation*, 2008,12(1):41–63. [doi: 10.1109/TEVC.2007.894202]
- [39] Qi Y, Yu J, Li X, Wei Y, Miao Q. Reservoir flood control operation using multi-objective evolutionary algorithm with decomposition and preferences. *Applied Soft Computing*, 2017,50:21–33.
- [40] Qi Y, Bao L, Sun Y, Luo J, Miao Q. A memetic multi-objective immune algorithm for reservoir flood control operation. *Water Resources Management*, 2016,30(9):2957–2977.
- [41] Anderson DR, Sweeney DJ, Williams TA. *Statistics for Business and Economics IIE*. South-Western Cengage Learning, 2008.
- [42] Chen H, Zhu X, Guo H, Zhu J, Qin X, Wu J. Towards energy-efficient scheduling for real-time tasks under uncertain cloud computing environment. *Journal of Systems and Software*, 2015,9:20–35. [doi: 10.1016/j.jss.2014.08.065]
- [43] Chen H, Zhu X, Qiu D, Liu L, Du Z. Scheduling for workflows with security-sensitive intermediate data by selective tasks duplication in clouds. *IEEE Trans. on Parallel and Distributed Systems*, 2018,28(9):2674–2688. [doi: 10.1109/TPDS.2017.2678507]

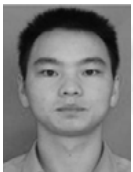
- [44] Chen H, Zhu X, Qiu D, Liu L. Uncertainty-Aware real-time workflow scheduling in the cloud. In: Proc. of the 9th IEEE Int'l Conf. on Cloud Computing. 2016. 577–584. [doi: 10.1109/CLOUD.2016.0082]

附中文参考文献:

- [2] 周爱民,张青富,张桂戌.一种基于混合高斯模型的多目标进化算法.软件学报,2014,25(5):913–928. <http://www.jos.org.cn/1000-9825/4514.htm> [doi: 10.13328/j.cnki.jos.004514]
- [22] 戚玉涛,刘芳,刘静乐,任元,焦李成.基于免疫算法和 EDA 的混合多目标优化算法.软件学报,2013,24(10):2251–2266. <http://www.jos.org.cn/1000-9825/4314.htm> [doi: 10.13328/j.cnki.jos.004314]
- [24] 胡旺,Yen GG,张鑫.基于 Pareto 熵的多目标粒子群优化算法.软件学报,2014,25(5):1025–1050. <http://www.jos.org.cn/1000-9825/4496.htm> [doi: 10.13328/j.cnki.jos.004496]
- [27] 李密青,郑金华,罗彪.一种基于最小生成树的多目标进化算法.计算机研究与发展,2009,46(5):803–813. <http://crad.ict.ac.cn/CN/article/searchArticleResult.do>
- [37] 刘海林,辜方清,张晓明.基于幂变换的多目标进化算法 MOEA/D 权重设计方法.计算机研究与发展,2012,49(6):1264–1271. <http://crad.ict.ac.cn/CN/abstract/abstract112.shtml>



陈黄科(1990—),男,博士生,主要研究领域为计算智能,大规模优化,云计算,任务调度,资源管理.



伍国华(1987—),男,博士,讲师,CCF 专业会员,主要研究领域为计算智能,进化计算,调度,机器学习.



霍离俗(1991—),男,博士生,主要研究领域为计算智能,大规模优化,调度.



戚玉涛(1981—),男,博士,副教授,CCF 专业会员,主要研究领域为演化计算,多目标优化.