

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使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.

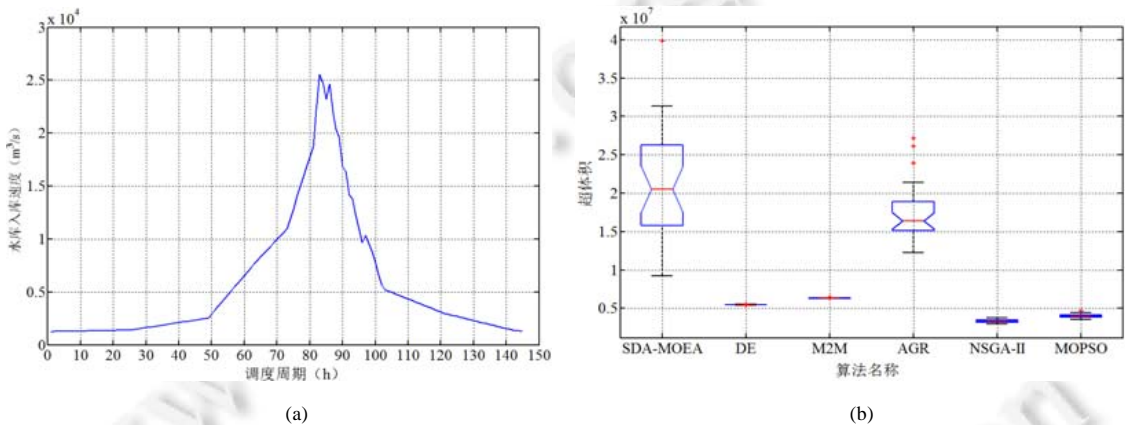


Fig.3 Data of inflow volume on July, 2010 and the box plots of different algorithms

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] Shahsavari 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]

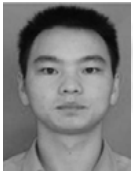
- [44] Chen H, Zhu X, Qiu D, Liu L. Uncertainty-Aware real-time workflow scheduling in the cloud. In: Proc. of the 9th IEEE Int'l Conf. on Cloud Computing. 2016. 577–584. [doi: 10.1109/CLOUD.2016.0082]

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.