

引用格式:柯永斌,周红标. 基于自适应惩罚策略的 MOEA/D 算法设计及应用[J]. 微电子学与计算机, 2020, 37(7): 59-65.
[KE Y B, ZHOU H B. Design and Application of MOEA/D with Adaptive Penalty Strategy [J]. Microelectronics & Computer, 2020, 37(7): 59-65.]

基于自适应惩罚策略的 MOEA/D 算法设计及应用

柯永斌¹, 周红标²

(1 淮阴工学院 电子信息工程学院, 江苏 淮安 223001; 2 淮阴工学院 自动化学院, 江苏 淮安 223001)

摘 要: 基于分解的多目标进化算法(Multiobjective evolutionary algorithm based on decomposition, MOEA/D)将一个多目标优化问题(multiobjective optimization problem, MOP)分解成一系列的单目标优化子问题, 然后利用相互协作的进化方式优化这些子问题. MOEA/D 利用独特的分解机制促进种群逼近 Pareto 最优前端(Pareto optimal front, POF), 同时利用均匀分布的权重向量维护种群的多样性, 在解决 MOPs 时具有较大的优势. 但是, 在实际工程中, 大多数 MOPs 的 POF 具有复杂的特性. 比如说 POF 可能具有长尾和顶点, 这极大地降低了 MOEA/D 算法的性能. 基于惩罚的边界交集法(penalty-based boundary intersection, PBI)是 MOEA/D 常用的分解方法之一. 在 PBI 法中, 惩罚因子起着平衡算法收敛性和多样性的关键作用. 本文提出了一种自适应惩罚策略(adaptive penalty strategy, APS), 能够在进化过程中自适应调整每个权重向量对应的惩罚因子值, 有效地增强了近似 Pareto 前端的多样性. 最后, 利用六个具有复杂 POF 的基准 MOPs 和空间桁架结构多目标优化实验验证了所提 MOEA/D-APS 算法的有效性.

关键词: 多目标优化; 分解; 基于惩罚的边界交集法; 自适应惩罚策略; 复杂 Pareto 最优前沿; 空间桁架结构优化

中图分类号: TP274

文献标识码: A

文章编号: 1000-7180(2020)07-0059-07

Design and application of MOEA/D with adaptive penalty strategy

KE Yong-bin¹, ZHOU Hong-biao²

(1 Faculty of Electronic Information Engineering, Huaiyin Institute of Technology, Huai'an 223001, China

2 Faculty of Automation, Huaiyin Institute of Technology, Huai'an 223001, China)

Abstract: Multiobjective evolutionary algorithm based on decomposition (MOEA/D) decomposes a multiobjective optimization problem (MOP) into a number of scalar optimization subproblems and optimizes them in a collaborative manner. In MOEA/D, decomposition mechanisms are used to push the population to approach the Pareto optimal front (POF), whilst a set of uniformly distributed weight vectors are applied to maintain the diversity of the population. MOEA/D has been shown to be very efficient in solving MOPs. However, the POF of many MOPs has complex characteristics, which significantly degrades the performance of MOEA/D. Penalty-based boundary intersection (PBI) is one of the frequently used decomposition approaches. In PBI approach, penalty factor plays a crucial role in balancing convergence and diversity. This paper proposes an adaptive penalty strategy (APS) which can dynamically adjust the value of the penalty factor for each weight vector during the evolutionary process. The diversity of approximated Pareto front can be enhanced by using this APS approach. The performance of the proposed MOEA/D-APS algorithm is investigated on six newly designed benchmark MOPs with complex POF shapes and the multiobjective optimization of space truss structure.

Key words: multiobjective optimization; decomposition; penalty-based boundary intersection; adaptive penalty strategy; complex Pareto optimal front; space truss structure optimization

收稿日期: 2019-10-19; 修回日期: 2019-11-22

基金项目: 江苏省产学研前瞻性项目(BY2016061-02)

1 引言

在日常生产生活中,人们遇到的问题大多是遵循“最大化效益,最小化成本”这一基本优化原则的多目标优化问题(multiobjective optimization problems, MOPs)^[1]. 由于这些目标之间一般都是相互冲突的,即没有一个最优解能够同时优化所有的目标. 对于这类问题, Pareto 最优解对于决策者来说是一个很好的选择,由于多目标进化算法(multiobjective evolutionary algorithms, MOEAs)一次运行能够提供多个 Pareto 最优解,且不受目标函数数学性质的影响,因此利用 MOEAs 来求解多目标优化问题受到研究者的广泛关注^[2]. 多目标优化根据选择机制的不同,可以将当前的 MOEAs 分成三类^[5-11].

基于分解的多目标进化算法(Multiobjective evolutionary algorithm based on decomposition, MOEA/D)将一个多目标优化问题分解成一系列的单目标优化子问题,然后利用进化算法同时求解这些单目标子问题^[10-12]. MOEA/D 常用的分解方法有加权法和、切比雪夫法和基于惩罚的边界交集法(PBI)^[13]. PBI 法利用惩罚因子 θ 构建由解到当前搜索方向的垂直距离和解到参考点的距离组成的单目标优化子问题,然后通过优化该子问题推动解逼近真实的 POF. 因此,在 PBI 法中, θ 的取值对 MOEA/D 的性能有着至关重要的影响^[14].

然而,原始的 PBI 法采用固定的 θ 值,在搜索过程中保持不变,对于具有复杂 POF 的 MOPs 以及实际工程优化问题,难以获得分布性较好的解. 但是,由于缺乏具有复杂特性的测试问题(经常用于 MOEAs 性能检验的 ZDT、DTLZ 和 WFG 基准测试集不具备复杂的 POF),对 PBI 的研究相对较少. 在文献[15]中, Sato 采用反转的 PBI 法提高了算法的搜索能力. 但是,该方法仍未考虑惩罚因子的影响,在解决具有不规则 POF 的 MOPs 时难以取得收敛性和多样性的平衡. 在文献[16]中, Yang 等人提出一种随迭代次数线性递增的 θ 值调整策略,命名为 MOEA/D-LPS 算法. 在 LPS 策略中,进化初期采用较小的 θ 值以期加快算法的收敛速度;反之,进化末期采用较大的 θ 值以期获得分布更均匀的解. 实验结果表明,该方法在解决具有复杂特性的 MOPs 时能够获得分布性更好的近似 Pareto 前端. 但是,该方法并未考虑每一个子问题的进化状态,统一的 θ 值不利于边界解多样性的

维护^[17-18].

针对上述问题,本文利用进化过程中每个权重向量周围解的浓度信息设计了惩罚因子自适应调整策略(Adaptive penalty strategy, APS),并将其集成到 MOEA/D 中,提出了 MOEA/D-APS 算法.

2 PBI 分解法

本文 MODA/D 采用的分解方法是 PBI 法^[1]. 在 PBI 法中,单目标优化子问题定义为:

$$\begin{aligned} \text{minimize } g^{pbi}(x|w, z^*) &= d_1 + \theta d_2 \\ \text{subject to } x &\in \Omega \end{aligned} \quad (1)$$

式中,

$$d_1 = \frac{\| (F(x) - z^*)^T w \|}{\| w \|},$$

$$d_2 = \| F(x) - (z^* + d_1 \frac{w}{\| w \|}) \|.$$

$w = (w_1, w_2, \dots, w_m)^T$ 为该子问题的搜索方向, $\theta > 0$ 为惩罚因子; $z^* = (z_{1*}, z_{2*}, \dots, z_{m*})$ 为目标空间中的理想点, $z_{i*} = \min\{f_i(x) | x \in \Omega\}$. 在搜索之前, z^* 一般是未知的,算法利用搜索过程发现的最小的 $f_i(x)$ 值来替代 z_{i*} . 图 1 给出了 PBI 法的示意图, L 为通过 z^* 与 w 方向一致的直线, p 为解 $F(x)$ 在 L 上的投影, d_1 为 z^* 和 p 之间的距离, d_2 为 $F(x)$ 和 L 之间的垂直距离. 该方法的目的是通过最小化 g^{pbi} 函数来推动 $F(x)$ 尽可能地低,从而使得其能够逼近可用目标解集的边缘. 通过图 1 不难发现, θ 起到了平衡收敛性(由 d_1 表示)和多样性(由 d_2 表示)的作用. 一个较小的 θ 值能够加快算法的收敛速度,但是容易导致种群陷入局部 Pareto 前沿;而一个较大的 θ 值有利于维护种群的多样性,但是会减慢算法的收敛速度.

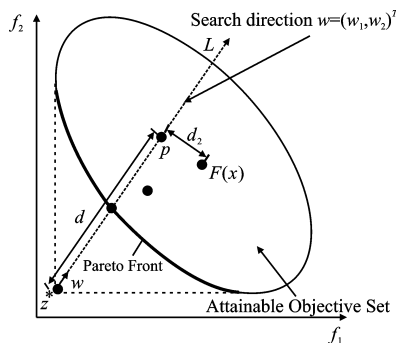


图 1 PBI 分解法示意图

3 自适应惩罚策略

本节进一步阐述 θ 取值对算法搜索性能的影响,

从而总结规律,设计本文的自适应惩罚策略. 图2给出了惩罚因子对算法搜索性能影响的示意图,其中图(a) θ 取1.0,图(b) θ 取5.0. 在图2中, $A \sim F$ 为近似 Pareto 前沿上分布的解, w^1 和 w^2 为两个相邻的权重向量. G 和 H 为两个目标空间中的解,从图2(a)可以发现,当 θ 取1.0时,对于权重向量 w^1 ,由于 $g^{thi}(C | w^1, z^*) < g^{thi}(B | w^1, z^*)$,因此在替换操作时解 C 将会取代解 B ,从而使得解向中间区域聚集,降低了种群的多样性. 从图2(b)可以发现,当 θ 取5.0时,对于权重向量 w^1 ,由于 $g^{thi}(H | w^1, z^*) > g^{thi}(G | w^1, z^*)$,因此在替换操作时解 H 无法取代解 G ,从而降低了种群的收敛速度.

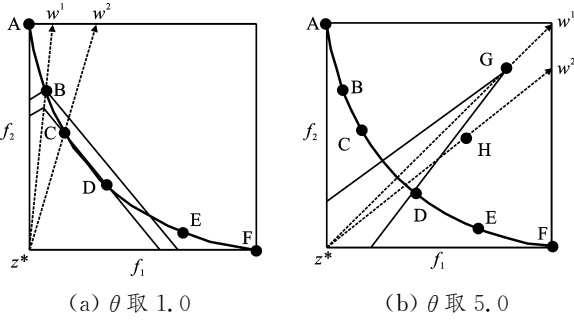


图2 惩罚因子对搜索性能影响的示意图

为了能够在进化过程中根据进化状态自适应调整每个子问题对应的 θ 值,首先定义权重向量(即子问题)周围解的浓度 con, 具体公式如下:

$$\text{con}^i = \frac{nc^i - nc_{\min} + \epsilon}{nc_{\max} - nc_{\min} + \epsilon}, i = 1, 2, \dots, N \quad (2)$$

式中, $\text{con}^i \in [0, 1]$ 为权重向量 i 的浓度, nc^i 为权重向量 i 周围聚集的解的个数, $nc_{\min}$ 为 $nc(W)$ 的最小值, $nc_{\max}$ 为 $nc(W)$ 的最大值, ϵ 取 10^{-6} 以防止分母为0. 对于权重向量 w^i , 如果种群中某个解 x^j 到该权重向量的距离小于自身对应解 x^i 到该权重向量的距离, 就认为 x^j 聚集在该权重向量周围. nc^i 的计算公式如下:

$$nc^i = \{x^j | d(x^j, w^i) < d(x^i, w^i)\} \quad (3)$$

$$i = 1, 2, \dots, N, j = 1, 2, \dots, N$$

式中, $d(x^j, w^i)$ 为解 x^j 和权重向量 w^i 之间的垂直距离, 它的计算公式如下:

$$d(x^j, w^i) = \left\| \bar{F}(x^j) - \frac{w^{iT} \bar{F}(x^j)}{w^{iT} w^i} w^i \right\| \quad (4)$$

式中, $\|\cdot\|$ 为 L_2 范数. $\bar{F}(x^j)$ 为解 x^j 的规范化目标向量, 它的第 k 个目标函数通过如下公式进行规范化

$$\bar{f}_k(x^j) = \frac{f_k(x^j) - z_k^*}{z_k^{\text{nad}} - z_k^*}, k = 1, 2, \dots, m \quad (5)$$

式中, $z_k^{\text{nad}} = \max\{f_k(x) | x \in \text{POS}\}$ 为搜索过程发现的近似 Pareto 前端的顶点.

因为子问题的最优解应该位于它的方向向量周围, 如果某些方向向量周围聚集的解的个数大于其他的方向向量, 则表明当前解的分布不均匀. 在进化过程中, 当惩罚参数过小时, 边界子问题的解容易被邻域内收敛性更好的解替换, 导致多样性的丧失; 当惩罚参数过大时, 种群的收敛速度会减慢, 导致无法逼近真实的 Pareto 前沿. 因此, 可根据每个子问题的浓度对惩罚因子进行动态调整. 本文利用 Sigmoid 函数映射方式来表征子问题惩罚因子 θ 与其浓度 con 之间的关系, 具体公式为:

$$\theta(\text{con}) = \frac{1}{1 + ae^{-b \times \text{con}}} \quad (6)$$

设置当 con 从 0 变化到 1 时, θ 从 50 变化到 2, 即当 con = 0 时, $\theta = 50$, 当 con = 1 时, $\theta = 2$ 代入公式(11)进行求解, 可以推出 $a = -0.98, b = 0.6729$, 因此, 子问题 i 的 θ 值与其浓度 con 之间的表达式为:

$$\theta^i = \frac{1}{1 - 0.98e^{-0.6729 \text{con}^i}} \quad (7)$$

根据公式(12), 在进化过程中动态调整每个子问题的惩罚参数, 可以取得收敛性和多样性的平衡. 对于具有复杂 Pareto 前沿的测试问题, 边界解容易向中间子问题聚集, 导致边界解的多样性丧失. 因此, 对于边界子问题, 当其浓度较小时, 赋予其一个较大的惩罚参数, 有利于增强解的多样性; 对于中间子问题, 当其浓度较大时, 赋予其一个较小的惩罚参数, 有利于保持解的收敛性.

4 MOEA/D-APS 算法

首先简要介绍初始化和进化操作, 然后给出 MOEA/D-APS 算法的流程.

4.1 初始化

MOEA/D 采用 Scheffé 于 1958 年提出的单纯形格子点设计方法设置权重向量, 具体如下^[2]:

$$w_j^i \in \left\{ \frac{0}{H}, \frac{1}{H}, \frac{2}{H}, \dots, \frac{H}{H} \right\} \quad (8)$$

$$\sum_{j=1}^m w_j^i = 1 \quad (9)$$

式中, 每一个子问题 i ($i = 1, 2, \dots, N$) 对应一个权重向量 $w^i = (w_{1i}, w_{2i}, \dots, w_{mi})$, $w_{ji} \geq 0, j = 1, 2, \dots, m$, 权重向量集为 $W = (w^1, w^2, \dots, w^N)$, $N = C_{H+m-1}^{m-1}$ 为权重向量的总数.

4.2 进化操作

MOEA/D-APS 也采用差分进化操作和多项式变

异产生子代种群 $Q_t = \{x^1, x^2, \dots, x^N\}$. 其中, 子代解 $x^i = \{x_1^i, x_2^i, \dots, x_n^i\}$ 采用如下公式产生

$$u_j^i = \begin{cases} x_j^{r_2} + F \times (x_j^{r_2} - x_j^{r_3}) & \text{if rand} < CR \\ x_j^i & \text{otherwise} \end{cases}$$

(10)

式中, CR 和 F 为差分进化操作的两个控制参数, rand 为 $[0, 1]$ 之间的随机数. x_{jr1} , x_{jr2} 和 x_{jr3} 为从交配池中随机选取的 3 个父代解. 多项式变异的操作如下:

$$x_j^i = \begin{cases} u_j^i + \sigma_j \times (b_j - a_j) & \text{if rand} < p_m \\ u_j^i & \text{otherwise} \end{cases}$$

(11)

$$\sigma_j = \begin{cases} (2 \times \text{rand})^{\frac{1}{\eta+1}} - 1 & \text{if rand} < 0.5 \\ 1 - (2 - 2 \times \text{rand})^{\frac{1}{\eta+1}} & \text{otherwise} \end{cases}$$

(12)

式中, $j \in \{1, 2, \dots, n\}$, p_m 为变异概率, η 为分布性指数, a_j 和 b_j 是第 j 维决策变量的下限和上限.

4.3 MOEA/D-APS 算法流程

表 1 给出 MOEA/D-APS 算法的流程.

表 1 MOEA/D-APS 算法流程

Algorithm: MOEA/D-APS	
	Initialize the population $P \leftarrow \{x^1, x^2, \dots, x^N\}$, a set of weight
1	vector $W \leftarrow \{w^1, w^2, \dots, w^N\}$, the ideal and nadir objective
	vectors z^*, z^{nad} ;
2	Set iteration $\leftarrow 0$;
3	for $i \leftarrow 1$ to N do
4	$B(i) \leftarrow \{i_1, i_2, \dots, i_T\}$ where $w^1, w^2, \dots, w^T$ are the T
	closest weight vectors to w^i ;
5	end
6	while Stopping criterion is not satisfied do
7	Normalize objective vector by Eq. (5);
8	Calculate the distance between the solution and the
	weight vector by Eq. (4);
9	Calculate the number of solutions around each weight
	vector by Eq. (3);
10	Calculate the concentration of solutions for each weight
	vector by Eq. (2);
11	Determine the value of penalty factor for each weight
	vector by Eq. (7);
12	for $i \leftarrow 1$ to N do
13	if $\text{uniform}(0, 1) < \delta$ then
14	$E \leftarrow B(i)$;
15	else
16	$E \leftarrow P$;
17	end
18	Randomly select three solutions x^{r1} , x^{r2} , and x^{r3} from
	E ;
19	Generate a candidate x' by using the method described
	in section 4.2;
20	Evaluate the F-function value FV of x' ;
21	Update the current ideal objective vector z^* ;
22	Update the current nadir objective vector z^{nad} ;
23	Set $c = 0$;

24	while $c < n_r$ & & E is not null do
25	Randomly pick an index j from E ;
26	Calculate the aggregation function value g using PBI
	method;
27	if $g(x' w^j, z^*) \leq g(x^j w^j, z^*)$ then
28	Set $x^j = x'$, $\text{FV}j = \text{FV}(x')$;
29	$c = c + 1$;
30	end
31	Delete j from E ;
32	end
33	iteration++;
34	end
35	return P ;

5 仿真研究

5.1 性能指标

(1) 反转世代距离 IGD

令 S 为均匀采样于真实 POF 的一组解, S' 为进化算法获取的近似 POF, 则 S' 的 IGD 值定义为

$$\text{IGD}(S, S') = \frac{\sum_{i=1}^{|S|} d(S_i, S')}{|S|}$$

(13)

式中, $d(S_i, S')$ 为 S_i 与近似 POF 的最小欧氏距离, $|S|$ 为 S 的数目. IGD 值越小, 表明算法获得的近似 Pareto 前端 S' 的质量越好.

(2) 超体积 HV

令 $z^r = (z_{1r}, z_{2r}, \dots, z_{mr})^T$ 为目标空间中的参考点, S' 为进化算法获取的近似 POF, 则 S' 的 HV 值定义为:

$$\text{HV}(S') = \text{VOL}(\bigcup_{x \in S'} [f_1(x), z_{1r}] \times \dots [f_m(x), z_{mr}])$$

(14)

式中, VOL 表示 Pareto 前沿 S' 中的解 x 与参考向量 z^r 所围成的区域的超体积. HV 值越大, 表明算法获得的解的质量越好.

5.2 参数设置

具体参数设置如下: $\delta = 0.9$, $CR = 0.5$, $F = 0.5$, $p_m = 1/n$, $\eta = 20$, $n_r = 2$, $N = 100$, $T = 20$, $\text{maxIteration} = 100$.

5.3 实验结果

5.3.1 F1-F6 测试结果

表 2 和表 3 分别给出了 MOEA/D-DE、MOEA/D-LPS 和 MOEA/D-APS 在 F1-F6 上 IGD 和 HV 的最好值、平均值和最差值, 其中粗体表示对应算法在该测试问题上取得了最好的结果.

从表 2 可以看出, MOEA/D-APS 能够有效提高算法的性能. 对于 F1、F3-F5, 与 MOEA/D-LPS 相比, MOEA/D-APS 的 IGD 值有所降低, 表明所提 APS 策略更加有利于发现边界解, 从而提高算

法的性能. 对于 F2, MOEA/D-DE 和 MOEA/D-LPS 对 POF 的覆盖性不足,致使最差的 IGD 值较大,而 MOEA/D-APS 的性能稳定,能够逼近整个 Pareto 前沿. 对表 3 进行分析,也能够得到类似结论.

表 2 IGD 对比结果				
Prob.		MOEA/D-DE	MOEA/D-LPS	MOEA/D-APS
F1	Best	0.118 0	0.073 2	0.025 8
	mean	0.122 1	0.082 6	0.029 7
	worst	0.125 8	0.090 3	0.038 3
F2	Best	0.007 4	0.008 4	0.004 7
	mean	0.011 6	0.010 4	0.005 8
	worst	0.783 8	0.039 1	0.007 2
F3	Best	0.147 1	0.125 9	0.039 7
	mean	0.321 1	0.279 4	0.266 2
	worst	0.342 3	0.342 8	0.347 1
F4	Best	0.082 5	0.070 7	0.019 7
	mean	0.083 6	0.074 4	0.024 4
	worst	0.084 5	0.079 1	0.028 7
F5	Best	0.111 5	0.036 3	0.026 8
	mean	0.116 0	0.102 4	0.037 3
	worst	0.120 3	0.110 8	0.056 2
F6	Best	0.048 9	0.037 1	0.036 6
	mean	0.053 5	0.038 7	0.042 7
	worst	0.060 9	0.042 0	0.049 0

表 3 HV 对比结果				
Prob.		MOEA/D-DE	MOEA/D-LPS	MOEA/D-APS
F1	Best	1.088 2	1.100 6	1.106 1
	mean	1.087 4	1.099 4	1.105 9
	worst	1.086 5	1.098 2	1.105 6
F2	Best	0.338 9	0.340 0	0.340 5
	mean	0.192 2	0.270 4	0.323 2
	worst	0.110 0	0.229 1	0.309 0
F3	Best	1.108 2	1.114 5	1.117 0
	mean	1.080 5	1.090 8	1.087 4
	worst	1.076 9	1.081 2	1.085 9
F4	Best	1.027 9	1.029 3	1.034 7
	mean	1.027 4	1.028 0	1.032 2
	worst	1.026 9	1.027 7	1.030 0
F5	Best	0.527 7	0.531 0	0.532 7
	mean	0.526 8	0.527 4	0.531 5
	worst	0.525 1	0.524 1	0.528 1
F6	Best	1.274 8	1.278 3	1.278 1
	mean	1.272 0	1.276 5	1.275 8
	worst	1.268 9	1.274 9	1.271 0

为了更加清晰地阐明本文所提自适应惩罚策略的优势,图 3 和图 4 分别给出了 MOEA/D-DE、

MOEA/D-LPS 和 MOEA/D-APS 在 F1 和 F6 上 30 次运行取得最低 IGD 值时近似 Pareto 前端和 IGD 指标的进化曲线.

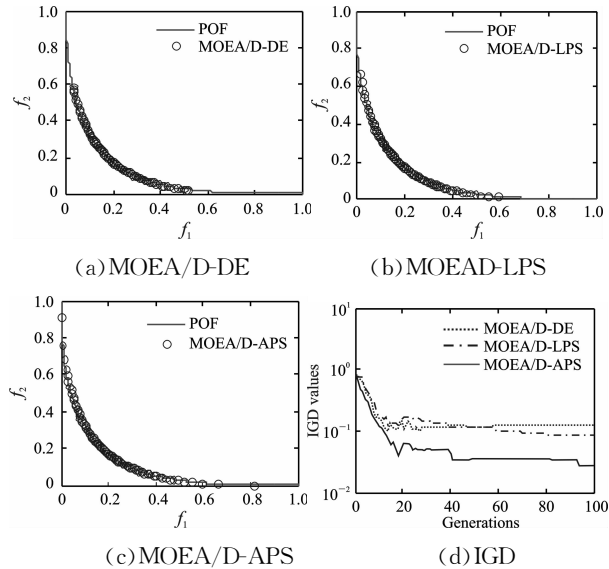


图 3 近似 Pareto 前端和 IGD 进化曲线

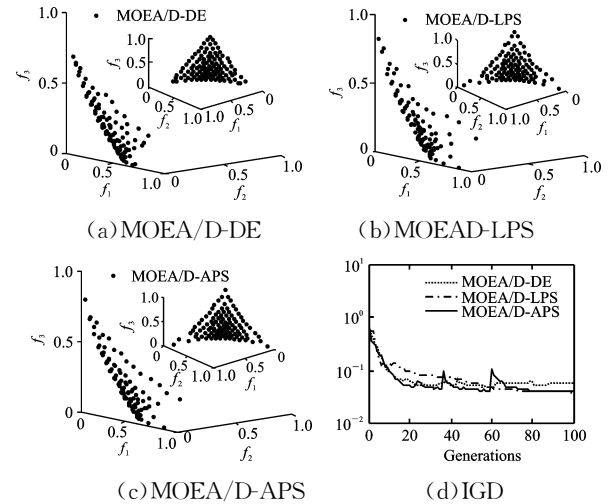


图 4 近似 Pareto 前端和 IGD 进化曲线

从图 3 可以看出,对于 F1 测试问题,MOEA/D-LPS 和 MOEA/D-APS 更容易发现边界解,尤其是 IGD 指标进化曲线表明 MOEA/D-APS 的性能有较大程度提升. 从图 4 可以看出,对于 F6 测试问题,MOEA/D-LPS 和 MOEA/D-APS 获取解的宽广性比原始算法更好.

5.3.2 空间桁架结构多目标优化

本文选择的 空间桁架结构具有 72 个杆件,16 个输入变量^[19]. 选取桁架结构质量和最大节点位移作为优化目标. 在仿真实验时,分别采用 MOEA/D-DE、MOEA/D-STM、MOEA/D-ACD 和 MOEA/D-APS 处理该问题. 图 5 给出了优化结果.

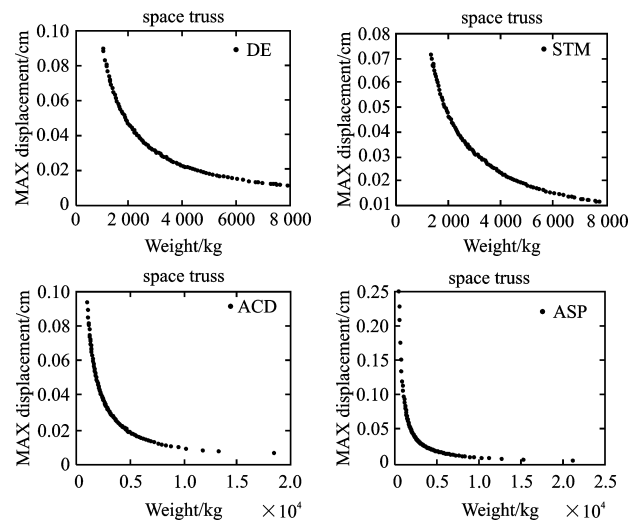


图 5 空间桁架结构优化结果

从图 5 可以看出,MOEA/D-APS 能够找到更多的边界解,这些解对应着更低的质量和更小的节点位移,对桁架结构优化具有重要的意义。

表 4 给出了四种对比算法的 HV 值。可以看出,MOEA/D-APS 算法的优化结果最好。

表 4 空间桁架 HV 数据比较表

空间桁架结构	DE	STM	ACD	APS
最优值	0.897 9	0.891 9	0.908 5	0.915 9
平均值	0.896 9	0.888 7	0.907 5	0.914 6
最差值	0.895 4	0.884 2	0.906 1	0.912 2

6 结束语

针对传统的 MOEA/D 算法在具有复杂 Pareto 前沿的多目标测试问题上容易引起多样性丧失的问题,本文提出一种带有自适应惩罚策略的改进 MOEA/D 算法,即 MOEA/D-APS。在进化过程中,利用每个权重向量周围解的浓度信息,结合 Sigmoid 映射函数,设计了惩罚因子自适应调整策略,达到了平衡算法收敛性和多样性的目的。利用六个带有复杂 Pareto 前沿的基准测试函数和空间桁架结构多目标优化问题对所提 MOEA/D-APS 算法的性能进行了验证,实证结果表明所提的自适应惩罚策略能够在一定程度上提高解的宽广性和多样性,尤其是在边界子问题上能够获取更多的解。

参考文献:

[1] TRIVEDI A, SRINIVASAN D, SANYALK, et al. A survey of multiobjective evolutionary algorithms based on decomposition[J]. IEEE Transactions on Evolutionary Computation, 2017, 21(3): 440-462.

[2] LI K, ZHANG Q F, KWONG S, et al. Stable matching-based selection in evolutionary multiobjective optimization[J]. IEEE Transactions on Evolutionary Computation, 2014, 18(6): 909-923.

[3] LI K, KWONG S, ZHANG Q F, et al. Interrelationship-based selection for decomposition multiobjective optimization[J]. IEEE Transactions on Cybernetics, 2015, 45(10): 2076-2088.

[4] WANG Z K, ZHANG Q F, ZHOU A, et al. Adaptive replacement strategies for MOEA/D[J]. IEEE Transactions on Cybernetics, 2016, 46(2): 474-486.

[5] COELLO C A C, PULIDO G T, Lechuga M S. Handling multiple objectives with particle swarm optimization[J]. IEEE Transactions on Evolutionary Computation, 2004, 8(3): 256-279.

[6] DEB K, PRATAP A, AGARWAL S, et al. A fast and elitist multiobjective genetic algorithm: NSGA-II[J]. IEEE Transactions on Evolutionary Computation, 2002, 6(2): 182-197.

[7] ZITZLER E, LAUMANN S, THIELE L. SPEA2: Improving the strength Pareto evolutionary algorithm[EB/OL]. [2019-09-10]. https://link.springer.com/chapter/10.1007%2F978-3-540-30217-9_75 2001.

[8] BEUME N, NAUJOKS B, EMMERICH M. SMS-EMOA: Multiobjective selection based on dominated hypervolume[J]. European Journal of Operational Research, 2007, 181(3): 1653-1669.

[9] BADER J, ZITZLER E. HypE: An algorithm for fast hypervolume-based many-objective optimization[J]. Evolutionary Computation, 2011, 19(1): 45-76.

[10] ZHANG Q F, LI H. MOEA/D: A multiobjective evolutionary algorithm based on decomposition[J]. IEEE Transactions on Evolutionary Computation, 2007, 11(6): 712-731.

[11] JASZKIEWICZ A. On the performance of multiple-objective genetic local search on the 0/1 knapsack problem-a comparative experiment[J]. IEEE Transactions on Evolutionary Computation, 2002, 6(4): 402-412.

[12] Qi Y, Ma X, Liu F, et al. MOEA/D with adaptive weight adjustment[J]. Evolutionary Computation, 2014, 22(2): 231-264.

[13] JIANG S Y, YANG S X. An improved multiobjective optimization evolutionary algorithm based on decomposition for complex Pareto fronts[J]. IEEE Transactions on Cybernetics, 2016, 46(2): 421-437.

[14] WU M, LI K, KWONG S, et al. Matching-based selection with incomplete lists for decomposition multi-objective optimization[J]. IEEE Transactions on Evo-

- lutionary Computation, 2017, 21(4): 554-568.
- [15] SATO H. Inverted PBI in MOEA/D and its impact on the search performance on multi and many-objective optimization [C]//Proceedings of the 2014 Annual Conference on Genetic and Evolutionary Computation. ACM, 2014: 645-652.
- [16] YANG S X, JIANG S Y, JIANG Y. Improving the multiobjective evolutionary algorithm based on decomposition with new penalty schemes[J]. Soft Computing, 2017, 21(16): 4677-4691.
- [17] LI H, ZHANG Q F. Multiobjective optimization problems with complicated Pareto sets, MOEA/D and NSGA-II [J]. IEEE Transactions on Evolutionary Computation, 2009, 13(2): 284-302.
- [18] ZHOU H, QIAO J F. Multiobjective optimal control for wastewater treatment process using adaptive MOEA/D[J]. Applied Intelligence, 2019, 49(3): 1098-1126.
- [19] HO-HUU V, HARTJES S, VISSER H G, et al. An improved MOEA/D algorithm for bi-objective optimization problems with complex Pareto fronts and its application to structural optimization[J]. Expert Systems with Applications, 2018(92): 430-446.

作者简介:

柯永斌 男,(1984-),硕士,工程师.研究方向为多目标优化算法. E-mail:282321661@qq.com.

周红标 男,(1980-),博士,讲师.研究方向为复杂系统的建模和优化控制.