

一种双链结构的多目标进化算法 DCMOEA

谢承旺^{1a}, 王志杰^{1a}, 魏波^{1a}, 徐君^{1b}, 汪慎文²

(1. 华东交通大学 a. 软件学院, b. 信息工程学院, 南昌 330013; 2. 石家庄经济学院 信息工程学院, 石家庄 050031)

摘要: 提出一种双链结构的多目标进化算法(DCMOEA). 该算法采用双链结构表示个体, 执行过程中无需设置外部归档集合, 并采用 ϵ 支配策略保持解群的多样性. DCMOEA与MOEA/D、NSGA-II、SPEA2和PAES一同在4个2-目标ZDT函数和4个3-目标DTLZ问题上进行实验, 并从算法所获解集的收敛性、分布均匀性和宽广性3个方面进行比较, 仿真实验结果表明了DCMOEA的综合性能最好, 是一种颇具竞争力的多目标进化算法.

关键词: 双链个体; 自重组; ϵ 支配; 多目标进化算法

中图分类号: TP18

文献标志码: A

A multi-objective evolutionary algorithm based on double chains

XIE Cheng-wang^{1a}, WANG Zhi-jie^{1a}, WEI Bo^{1a}, XU Jun^{1b}, WANG Shen-wen²

(1a. School of Software, 1b. School of Information Engineering, East China Jiaotong University, Nanchang 330013, China; 2. School of Information Engineering, Shijiazhuang University of Economics, Shijiazhuang 050031, China. Correspondent: XIE Cheng-wang, E-mail: chengwangxie@163.com)

Abstract: A multi-objective evolutionary algorithm based on double chains(DCMOEA) is proposed, which is characterized with populated individuals based on double chains, no external archive is needed, and the ϵ dominance mechanism is adopted for preserving population diversity. The DCMOEA is compared with MOEA/D, NSGA-II, SPEA2 and PAES simultaneously on the platform employing four 2-objective ZDT test functions and four 3-objective DTLZ instances from three aspects including convergence, spacing and maximum spread. Experimental results show that the DCMOEA has the best comprehensive performance among five multi-objective evolutionary algorithms, and is a promising multi-objective evolutionary algorithm.

Keywords: individual based on double chains; self-recombination; ϵ dominance; multi-objective evolutionary algorithm

0 引言

科学研究与工程实践中大量的优化问题需要对多个目标同时优化, 一般而言, 多目标优化问题(MOP)中各目标之间通过决策变量相互制约, 对其中一个目标优化必须以其他目标作为代价, 因而总体目标没有单个的最优解, 而是一组均衡各目标的折衷解. 进化算法(EA)是一种基于群体的搜索方法, 其隐含的并行性可以搜索多个目标, 运行一次可产生一组解. 鉴于EA的这些优势, 近年来已被广泛用于求解多目标优化问题.

第1个真正意义上的多目标进化算法(MOEA)当属 Schaffer^[1]提出的向量评估遗传算法(VEGA). 此

后研究者基于不同的应用背景和实际问题, 提出各种不同的MOEAs, 代表性的有: Zitzler等^[2]提出的强度Pareto进化算法(SPEA); Kim等^[3]在SPEA基础上提出的改进型强度Pareto进化算法SPEA2; Srinivas等^[4]提出的非劣分类遗传算法(NSGA); Deb等^[5]对原始的NSGA进行改进, 提出了改进型非劣分类遗传算法(NSGA-II); Knowles等^[6]提出的Pareto存档进化策略(PAES)等. 从2003年至今, 多目标进化算法前沿领域的研究呈现出新的特点, 一些新型占优机制和新的进化机制被引入多目标进化优化领域, 如Zhang等^[7]将传统的数学规划方法与进化算法相结合提出的MOEA/D等. 多目标进化算法的研究呈现出了蓬

收稿日期: 2013-12-13; 修回日期: 2014-04-18.

基金项目: 国家自然科学基金项目(61165004); 江西省自然科学基金项目(20114BAB201025); 教育部人文社会科学青年基金项目(14YJCZH172); 江西省教育厅科技基金项目(GJJ14373, GJJ14396, GJJ12307); 河北省教育厅科研基金项目(QN20131053); 河北省青年拔尖人才支持计划项目(冀字[2013]17号).

作者简介: 谢承旺(1974—), 男, 副教授, 博士, 从事智能计算等研究; 王志杰(1989—), 男, 硕士生, 从事群智能算法的研究.

勃发展的态势。

然而,已有的多目标进化算法基本上是采用单链结构实现,即种群中的个体用一条单独的链表示其基因结构,这种表示方法简单,进化操作执行方便,但也存在明显不足,即个体携带的遗传信息量少,如不采用一定的机制,群体易出现退化。主流的多目标进化算法一般直接或间接地使用外部档案来保留精英个体,防止种群退化,促进算法收敛。因此,单链模式便成为影响MOEAs收敛的重要潜在因素。

Goldberg等^[8]对单链模式进行过改进,提出了一种变长染色体遗传算法(MGA),Van等^[9]提出将积木块假说和MGA的思想扩充到多目标进化领域,提出了多目标变长染色体遗传算法(MOMGA),Zydallis等^[10]针对MOMGA在PEI阶段耗时严重的缺陷作了改进,提出了MOMGA-2算法。Matayoshi^[11]用一种带角联接的双染色体遗传算法解决二维条带包装的单目标优化问题,取得了较好的效果。Guo等^[12]用一种改进的双倍体遗传算法解决单目标的车间作业调度问题,在全局搜索能力和收敛性方面均有较大改善。

与目前已有的研究不同,本文提出一种双链结构的多目标进化算法(DCMOEA),它是一种双链模式的多目标进化算法,在多目标进化优化的研究中尚未发现类似的工作。DCMOEA的特点在于:1)它用两条相对独立的基因链表示种群个体,使个体能携带大量的遗传信息,有利于抑制种群退化,促进算法收敛;2)进化过程中搜索到的优良个体均保留在种群中,无需设置外部归档集;3)利用 ε 支配方法保持解群的多样性,比较实验表明该策略具有明显的优势。

1 多目标优化问题及相关概念

下面给出与本文密切相关的几个概念。

定义1 多目标优化问题(不失一般性,以最小化问题为例):

$$\begin{aligned} \min y &= f(x) = (f_1(x), f_2(x), \dots, f_k(x)), \\ \text{s.t. } e(x) &= (e_1(x), e_2(x), \dots, e_m(x)) \leq 0. \end{aligned} \quad (1)$$

其中: $x = (x_1, x_2, \dots, x_n) \in X \subset \mathbf{R}^n$, x 为决策向量, X 为由决策向量 x 形成的决策空间; $y = (y_1, y_2, \dots, y_k) \in Y \subset \mathbf{R}^k$, y 为目标向量, Y 为目标向量 y 形成的目标空间; $e(x) \leq 0$ 确定决策变量 x 的可行范围。

定义2 Pareto 支配关系: 如果决策空间中两个向量 x, y 满足以下条件:

$$\forall i \in [1 : k]: f_i(x) \leq f_i(y) \wedge \exists j \in [1 : k], \quad (2)$$

$$f_j(x) < f_j(y),$$

则称 x 支配 y , 记为 $x \prec y$ 。

定义3 Pareto 最优解: 对于决策空间 X 中的

任一点 x^* , 若决策空间 X 中不存在支配 x^* 的向量, 则称 x^* 为 Pareto 最优解, 即 $\neg \exists x \in X: x \prec x^*$ 。

定义4 Pareto 最优集: 决策空间 X 中所有的 Pareto 最优解构成的集合称为 Pareto 最优集 (POS), 即

$$\text{POS} = \{x^* \in X | \neg \exists x \in X: x \prec x^*\}. \quad (3)$$

定义5 Pareto 前沿: 集合 POS 对应的目标向量的集合记为 Pareto 前沿 (POF), 即

$$\text{POF} = \{f(x^*) | x^* \in \text{POS}\}. \quad (4)$$

定义6 ε 支配: 设 $x, y \in X$, $\varepsilon_i > 0, \forall i \in \{1, 2, \dots, k\}$, $f_i(x) - \varepsilon_i \leq f_i(y)$ 且存在 i 使得 $f_i(x) - \varepsilon_i < f_i(y)$, 则称 $x \varepsilon$ 支配 y , 记为 $x \prec_\varepsilon y$ 。

2 相关工作

2.1 双链结构的个体

主流多目标进化算法一般采用单链结构表示个体, 即若 x 为 MOEA 种群中任一个体, 则其编码后的基因链可表示为 (x_1, x_2, \dots, x_n) , 其中 $x_i (i \in [1 : n])$ 为某基因位。单链结构个体携带的遗传信息较少, 这种表达方式与生物界中低等生物(例如细菌和病毒等)的 DNA 或 RNA 结构类似, 而低等生物的进化存在不可控因素, 一般很难稳定寻优^[13]。高等生物的 DNA 一般是双链的, 与低等生物的单链相比, 高等生物的进化相对稳定, 种群一般会朝着更适应环境的方向进化^[14]。本文提出一种拟合高等生物遗传进化的双链模式, 与生物双螺旋 DNA 结构不同, 本文设计的双链彼此之间独立, 并无严格的配对关系。图1为个体的双链结构示意图。

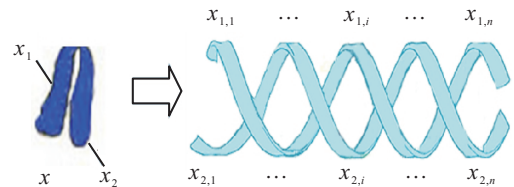


图1 双链结构个体的基因示意图

图1中 $x_1 = (x_{1,1}, x_{1,2}, \dots, x_{1,n})$ 表示 x 的第1条链, $x_2 = (x_{2,1}, x_{2,2}, \dots, x_{2,n})$ 表示 x 的第2条链, x_1 与 x_2 相互独立, 进化过程中 x_1 代表个体 x 参与适应值评价和群体选择, x_2 用于保留潜在的较优基因, 并不参与群体选择。

2.2 个体自重组

种群中的个体在参与选择之前, 先在个体内部进行自重组, 其目的是用两条链中较优的一条链代表个体参与群体选择, 并用重组后的一条链保留个体潜在的优良基因。该过程可以在更大的范围内(通常是种群规模的两倍)选择较优个体参与繁殖, 并将潜在的优良基因保留下来, 不致丢失, 有利于促进种群较快收敛。

下面给出个体自重组过程的描述.

算法 1 个体的自重组过程.

输入: $x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$;

输出: $x^{SR} // x^{SR}$ 为重组后的个体.

Step 1: 输入个体 x .

Step 2: $x_{\text{Better}} = \text{Evaluation_chain}(x)$; // 评估 x 较优链 x_{Better} .

Step 3: $x_r = \text{Recombination}(x_1, x_2)$; // 重组链 x_1 和 x_2 .

Step 4: 用 x_{Better} 链替换链 x_1 , 用 x_r 替换链 x_2 .

Step 5: 输出自重组后的个体 $x^{SR} = \begin{bmatrix} x_{\text{Better}} \\ x_r \end{bmatrix}$.

算法 1 中 `Evaluation_chain` 函数用于评估双链个体中较优的一条链. 通常个体中两条链可能存在以下两种情况: 1) x_1 与 x_2 之间存在支配关系; 2) x_1 与 x_2 相互非支配. 如果出现第 1 种情况则选择支配链为较优链; 如果是第 2 种情形则需进一步计算两条链分别对应的目标向量的胜出数. 胜出数的计算规则如下: 对于两个维度相同的目标向量, 累计各自的目标分量优于对方相应的目标分量的目标数目, 该累计数即为胜出数, 胜出数较大的链胜出.

下面给出函数 `Evaluation_chain` 的描述.

算法 2 评估个体的较优链.

输入: $x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$;

输出: $x_{\text{Better}} // x_{\text{Better}}$ 为 x 中较优的一条链.

Step 1: 考察个体 x 的双链 x_1 和 x_2 对应的目标向量之间的支配关系.

Step 2: 如果 $f(x_1)$ 支配 $f(x_2)$, 则 x_1 为较优链, $x_{\text{Better}} = x_1$.

Step 3: 如果 $f(x_2)$ 支配 $f(x_1)$, 则 x_2 为较优链, $x_{\text{Better}} = x_2$.

Step 4: 如果 $f(x_1)$ 与 $f(x_2)$ 相互非支配, 则计算两条链的胜出数, 胜出数较大者为较优链; 如果两条链的胜出数相同, 则取 x_1 为较优链.

Step 5: 输出 x_{Better} .

算法 2 中出现两条链胜出数相同时选择第一条链为较优链的原因在于, 个体的第 1 条链一般是以前世代中较优基因的延续, 具有更大的保留价值.

算法 1 中的个体 x 双链之间的重组分两种情况进行:

1) 如果 x 是连续型基因, 则按下式进行杂交:

$$x_{r,i} = \frac{1}{2}[(1 - \alpha)x_{1,i} + (1 + \alpha)x_{2,i}], i \in \{1 : n\}. \quad (5)$$

其中: $x_{1,i}, x_{2,i}$ 分别为个体 x 的第 1 和第 2 条链上的

第 i 个基因值, $x_{r,i}$ 为重组后的链 x_r 上的第 i 个基因值, α_i 为 $[0,1]$ 区间内的均匀随机数.

2) 如果 x 是离散型基因, 则重组后的链 x_r 上的第 i 个基因值按下式产生:

$$x_{r,i} = \begin{cases} x_{1,i}, & k = 0; \\ x_{2,i}, & k = 1. \end{cases} \quad (6)$$

其中 k 为 $\{0, 1\}$ 中的均匀随机数.

2.3 群体多样性策略

Pareto 最优性定义中存在固有的缺陷, 对于 3 个或 3 个以上目标的优化问题, 进化种群极易被非支配个体充斥, 从而阻滞群体向真正的 Pareto 前沿逼近, 因此, 有必要对种群进行修剪. 有很多修剪策略, 如 PAES 中的自适应网格^[6], NSGA-II 中的 `Crowding-Distance`^[5], SPEA 中的聚类^[2]和 SPEA2 中的 k -最近邻方法^[3]等都在各自算法中发挥了很好的作用. 然而, 正如文献 [15] 所述, 这些修剪策略很可能造成 Pareto 前沿的退化, 进而影响到最终种群的收敛. ϵ 支配^[16]采用一种弱支配方式控制种群的多样性, 通过用户自定义的参数 ϵ 确定网格大小, Pareto 前沿上每个网格内最多只有一个个体, 保持了种群的多样性, 能够很好地防止进化过程中种群的退化. 图 2 以两个目标的优化问题为例说明了 ϵ 支配的原理.

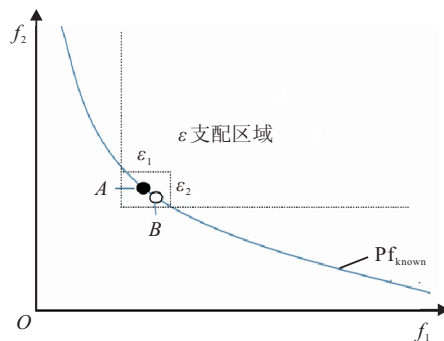


图 2 ϵ 支配示意图

图 2 中的 A 和 B 是 Pareto 前沿上的两个解, 它们相互非支配, 但 B 点位于 A 点的 ϵ 支配区域内, 因而会被淘汰. ϵ 支配的优点在于其简单、快速和高效, 并能在短时间内获得一个具有良好分布性和收敛性的解集.

DCMOEA 将较优的个体保留在种群中, 不另设归档集. 当种群中非支配个体数目未达到种群最大规模时不执行 ϵ 支配更新操作; 否则, 执行 ϵ 支配更新种群, 维持群体的多样性, 具体的更新过程如下.

算法 3 ϵ 支配更新种群.

输入: 非支配种群 ND_{Set} ;

输出: ϵ 支配更新后的种群.

Step 1: 输入非支配种群 ND_{Set} ;

Step 2: 对 ND_{Set} 中每个个体 i 相应的目标向量增加一个 ε 分量, 形成个体 i 的 ε 支配区域;

Step 3: 如果在个体 i 的 ε 支配区域内存在其他个体, 则剔除这些个体, 重复执行 Step 2 和 Step 3, 直至种群规模达到要求;

Step 4: 输出 ε 支配更新后的种群.

2.4 种群规模维护

DCMOEA 算法没有设置归档集, 所有较优个体均保留在种群中, 对种群进行环境选择后, 一些非支配个体被淘汰, 种群规模变小. 为维持每一代种群规模大小不变, 需要及时补充一定数目的新个体. 设种群规模为 N , 其中非支配个体数目为 $|ND_{Set}|$, 则删除被支配的个体后补充新个体的方法如下.

算法 4 种群规模维持.

输入: 大小为 $|ND_{Set}|$ 的非支配个体集合;

输出: 规模为 N 的新种群.

Step 1: 若 $|ND_{Set}| < N/2$, 则执行下面两步:

Step 1.1: 将 $|ND_{Set}|$ 个个体的双链拆开成 $(2 \times |ND_{Set}|)$ 条单链, 然后再给每条单链补齐一条由自身与种群中任一条其他链重组后的单链;

Step 1.2: 剩余 $(N - 2 \times |ND_{Set}|)$ 个个体由原非支配个体变异得到.

Step 2: 若 $|ND_{Set}| = N/2$, 则执行 Step 1.1.

Step 3: 若 $|ND_{Set}| > N/2$, 则随机选择 $(N - |ND_{Set}|)$ 个非支配个体并执行 Step 1.1, 不足的 $(N - 2 \times (N - |ND_{Set}|))$ 个个体分别由未选中的非支配个体的第 1 条链与随机产生的一条链组成 (随机产生的链用作新个体的第 2 条链).

Step 4: 输出规模为 N 的新种群.

3 DCMOEA 算法

在第 2 节的基础上给出 DCMOEA 算法的流程.

算法 5 一种双链结构的多目标进化算法 DCMOEA.

Step 1: 初始化. 随机生成规模为 N 的初始种群, 设置算法最大迭代次数 T , 并设进化代数 $t = 0$.

Step 2: 对种群执行个体自重组过程 (见算法 1).

Step 3: 环境选择. 选择种群中非支配个体构成非支配集, 被支配个体淘汰.

Step 4: 保持群体多样性. 利用 ε 支配更新非支配集合 (参见算法 3).

Step 5: 群体规模维护. 利用算法 4 对种群进行修剪.

Step 6: 如果 $t < T$, 则 $t = t + 1$, 并转至 Step 2; 否则, 将所获群体的非支配解集视为算法最终的求解结

果输出.

需要说明的是, 本文在实验中若连续 10 代以上获得的非支配解集大小均达到种群规模的 95% 以上时, 算法终止.

若用 M 表示优化目标数目, N 表示群体规模, 则 DCMOEA 执行个体自重组操作的时间复杂度为 $O(MN)$, 进行环境选择时获得非支配解集的时间为 $O(MN^2)$, 利用 ε 支配保持群体多样性的时间为 $O(MN^2)$, 维护种群规模的时间不超过 $O(N)$, 因此 DCMOEA 总的时间复杂度为 $O(MN^2)$.

4 实验及结果分析

将 DCMOEA 与 4 个多目标进化算法 MOEA/D、NSGA-II、SPEA2 和 PAES 在 8 个测试问题上进行仿真实验, 分别考察算法在收敛性、分布均匀性和宽广性 3 个方面的性能, 并对实验结果进行分析.

为公平比较起见, 所有实验均在 Think Pad X200 个人笔记本电脑上进行, 电脑配置 5 G 内存和 2.4 GHz 双核 CPU, 安装 Windows 7 X64 操作系统和 Matlab 2010a 编程环境.

4.1 测试问题

ZDT 系列问题^[17]和 DTLZ 一族函数^[18]是 MOEA 研究领域广泛使用的代表性测试问题. ZDT 问题由 6 个具有不同性质的 2-目标优化问题组成, 本实验选用了其中的 4 个问题: ZDT1、ZDT2、ZDT3、ZDT6. DTLZ 系列问题是一族高维多目标优化问题的测试函数, 它们具有可伸缩性, 即优化目标的数目可任意设置. 这里选用 4 个 3-目标的测试函数 DTLZ1、DTLZ2、DTLZ4 和 DTLZ7. 更为重要的是, 本实验中所用测试问题的 Pareto 前沿均为已知, 这为算法的性能评估提供了极大的便利.

4.2 性能度量指标

多目标进化算法所获解集一般要从 3 个方面进行评价: 1) 解集逼近真实 Pareto 前沿的程度, 即收敛性度量; 2) 解集在 Pareto 前沿上分布的均匀性; 3) 解集在 Pareto 前沿上分布的宽广性, 即覆盖程度的度量. 同时考虑这 3 个方面才能较全面反映出 MOEA 的性能.

本文采用文献 [19] 提出的 Generational Distance (GD) 评价算法的收敛性, 其定义如下:

$$GD = \frac{1}{n} \sqrt{\sum_{i=1}^n d_i^2}. \quad (7)$$

其中: n 为解集的大小, d_i 为算法获得的解距离真实 Pareto 前沿的最小欧式距离. GD 值越小说明算法的解集越靠近 Pareto 前沿, 即收敛性越好.

采用 Schott^[20]提出的 Spacing(SP) 空间评价方法来评价解集分布的均匀性,其定义如下:

$$SP = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (\bar{d} - d_i)^2} \quad (8)$$

其中: $d_i = \min_j (|f_1^j(x) - f_1^i(x)| + |f_2^j(x) - f_2^i(x)|)$, $i, j \in \{1, 2, \dots, n\}$, n 为解集中个体的数目, \bar{d} 为所有 d_i 的平均值, SP 越小说明解集分布越均匀。

采用 Maximum Spread(MS)^[21]来衡量所得到的解集对真实 Pareto 前沿的覆盖程度,其定义如下:

$$MS = \sqrt{\frac{1}{n} \sum_{i=1}^n \left\{ \frac{\min(f_i^{\max}, F_i^{\max}) - \min(f_i^{\min}, F_i^{\min})}{F_i^{\max} - F_i^{\min}} \right\}} \quad (9)$$

其中: n 为优化目标函数的数目; f_i^{\max} 和 f_i^{\min} 分别为解集中第 i 个目标的最大值和最小值; F_i^{\max} 和 F_i^{\min} 分别表示真实 Pareto 前沿上的第 i 个目标的最大值和最小值。MS 越大说明解集对真实 Pareto 前沿的覆盖

程度越好。

4.3 实验与分析

仿真实验中 5 个算法的基本参数如表 1 所示。为合理选取 ε 支配参数,多次运行 DCMOEA 算法,并最终确定了该算法在 4 个 2-目标的 ZDT 问题和 4 个 3-目标的 DTLZ 函数中 ε 的取值,它们分别是: 1) ZDT1 和 ZDT2 取相同的 ε 参数,且 $\varepsilon_1 = 0.008, \varepsilon_2 = 0.008$; 2) ZDT3 取 $\varepsilon = (0.0028, 0.016)$; 3) ZDT6 取 $\varepsilon = (0.0072, 0.0085)$; 4) DTLZ1 取 $\varepsilon = (0.25, 0.25, 0.25)$; 5) DTLZ2 和 DTLZ4 取相同的 $\varepsilon = (0.042, 0.042, 0.042)$; 6) DTLZ7 取 $\varepsilon = (0.042, 0.042, 0.048)$ 。

最后, 5 个算法 DCMOEA、MOEA/D、NSGA-II、SPEA2 和 PAES 在 2-目标的测试问题中各执行函数评估 20 000 次,在 3-目标的测试函数中分别执行函数评估 100 000 次。

为了直观、清晰地表示出各算法在求解多目标 benchmark 测试问题中所获解集的均匀性及其分布的宽广性,图 3 和图 4 给出了 5 个算法分别在 2-目标测

表 1 5 个算法的基本参数

算法	种群规模	归档集合	杂交算子	变异算子	用户自定义参数
DCMOEA	100	无	SBX(分布指数 $\mu = 20$)	多项式变异(分布指数 $\eta = 20$)	与目标函数相关的 ε 向量
MOEA/D	100	100	SBX(分布指数 $\mu = 20$)	多项式变异(分布指数 $\eta = 20$)	邻域大小 $T = 20$
NSGA-II	100	100	SBX(分布指数 $\mu = 20$)	多项式变异(分布指数 $\eta = 20$)	无
SPEA2	100	100	SBX(分布指数 $\mu = 20$)	多项式变异(分布指数 $\eta = 20$)	$k = 2$
PAES	100	100	无	多项式变异(分布指数 $\eta = 20$)	无

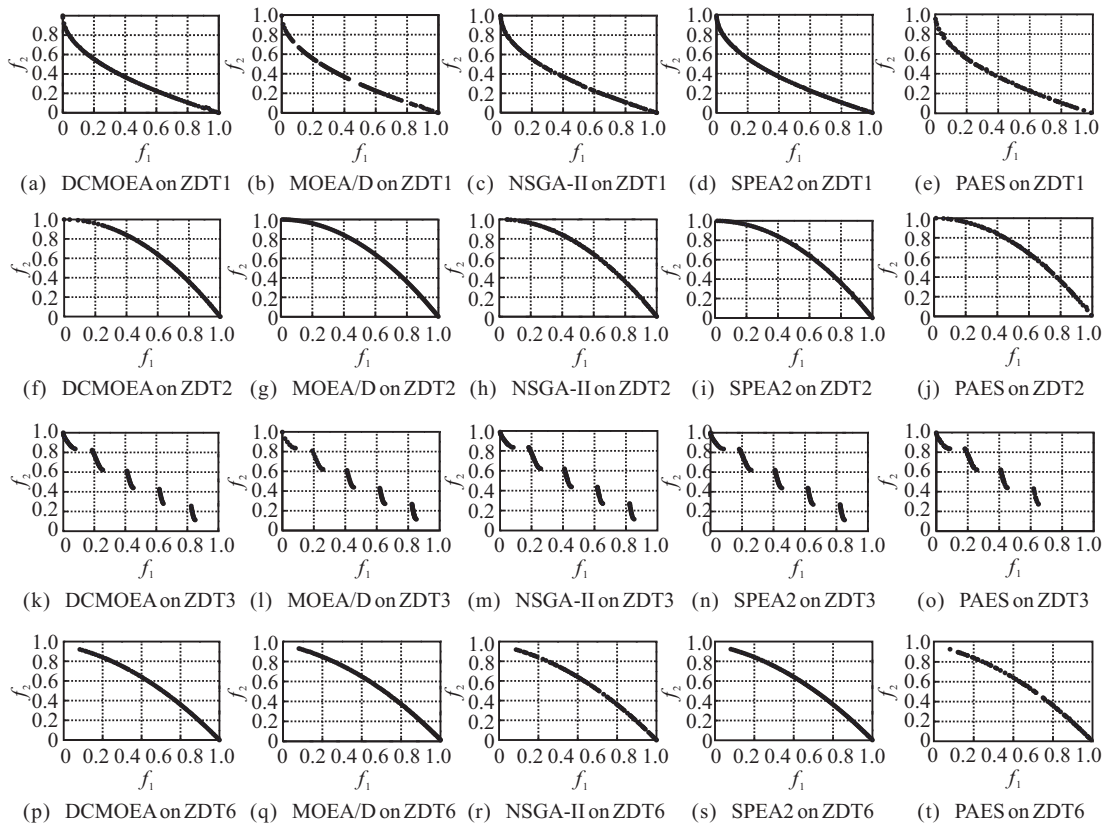


图 3 5 种算法在 2-目标的 ZDT 系列测试函数上获得的近似 Pareto 前沿

试问题(ZDT1、ZDT2、ZDT3、ZDT6)和3-目标测试函数(DTLZ1、DTLZ2、DTLZ4、DTLZ7)上获得的近似Pareto前沿,它们是根据各算法第1次执行后所获得的近似Pareto前沿生成的.表2~表4分别给出了

5个算法在4个2-目标测试问题和4个3-目标测试函数上获得的GD、SP和MS等指标的均值与方差.需要说明的是,表2~表4中的数据是统计各算法独立执行20次所获得的均值与方差.

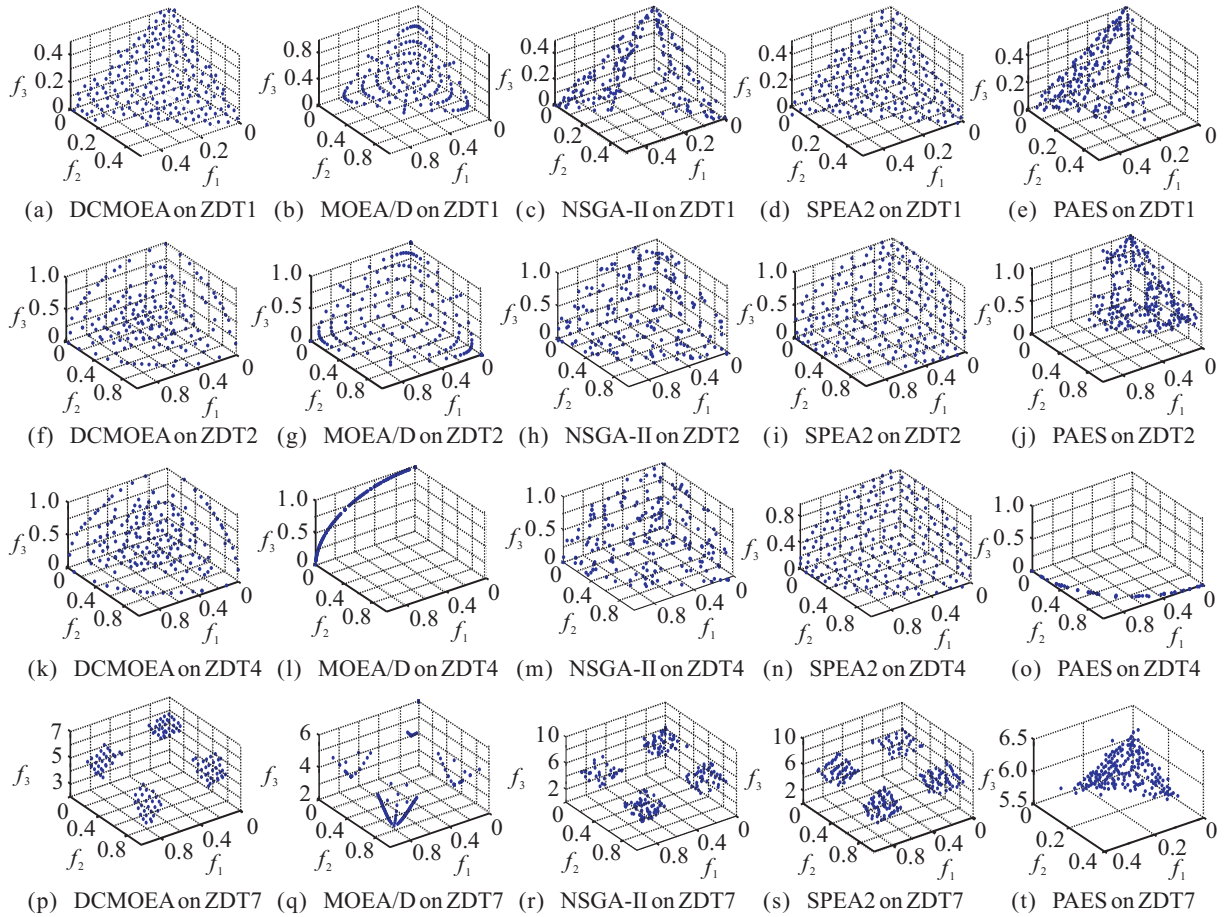


图4 5种算法在3-目标的DTLZ系列测试函数上获得的近似Pareto前沿

表2 各算法在基准多目标测试问题中的GD均值与方差

Problem	DCMOEA		MOEA/D		NSGA-II		SPEA2		PAES	
	GD(avg)	GD(σ)	GD(avg)	GD(σ)	GD(avg)	GD(σ)	GD(avg)	GD(σ)	GD(avg)	GD(σ)
ZDT1	1.93e-4	1.32e-4	1.36e-4	3.16e-5	2.59e-4	1.58e-4	2.44e-4	2.01e-4	2.44e-4	2.01e-4
ZDT2	1.10e-4	6.56e-5	1.52e-4	1.27e-4	2.12e-4	1.63e-4	1.36e-4	7.88e-5	1.36e-4	7.88e-5
ZDT3	3.26e-4	2.97e-5	8.9e-4	1.8e-3	6.12e-4	6.93e-5	3.89e-4	2.44e-5	.89e-4	2.44e-5
ZDT6	6.01e-4	3.25e-5	7.89e-4	1.24e-4	6.77e-4	7.50e-5	6.33e-4	2.98e-5	6.33e-4	2.98e-5
DTLZ1	1.40e-3	6.70e-4	2.3e-2	4.11e-4	2.1e-3	1.8e-3	1.6e-3	3.61e-2	1.6e-3	3.61e-2
DTLZ2	1.10e-3	1.70e-4	6.60e-4	1.37e-5	1.1e-3	7.4e-5	1.1e-3	1.26e-4	1.1e-3	1.26e-4
DTLZ4	8.60e-4	6.7e-4	6.33e-4	8.25e-5	1.1e-3	8.8e-4	9.3e-4	2.9e-4	9.3e-4	2.9e-4
DTLZ7	1.60e-3	2.7e-4	7.24e-4	1.25e-5	1.8e-3	1.8e-5	2.3e-3	3.6e-4	2.3e-3	3.6e-4

表3 各算法在基准多目标测试问题中的SP均值与方差

Problem	DCMOEA		MOEA/D		NSGA-II		SPEA2		PAES	
	GD(avg)	GD(σ)	GD(avg)	GD(σ)	GD(avg)	GD(σ)	GD(avg)	GD(σ)	GD(avg)	GD(σ)
ZDT1	7.2e-3	1.5e-3	1.02e-2	9.14e-4	7.9e-3	1.3e-3	3.2e-3	2.4e-4	9.9e-3	1.4e-3
ZDT2	7.3e-3	1.3e-3	6.0e-3	1.7e-3	8.4e-3	3.1e-3	3.1e-3	3.7e-4	1.17e-2	3e-3
ZDT3	7.9e-3	4.4e-4	2.32e-2	5.45e-4	8.0e-3	8.2e-4	3.9e-3	5.9e-4	1.66e-2	1.54e-2
ZDT6	6.4e-3	1.1e-3	4.3e-3	5.24e-4	7.2e-3	5.8e-3	2.4e-3	2.1e-4	3.18e-2	3.58e-2
DTLZ1	1.86e-2	1.5e-3	2.58e-2	8.35e-4	2e-2	8.1e-3	4.86e-1	5.16e-1	6.61e-1	1.27e0
DTLZ2	2.08e-2	2.4e-3	3.86e-2	6.92e-4	3.9e-2	2.1e-3	1.68e-2	6.9e-4	2.49e-2	5.3e-3
DTLZ4	3.06e-2	4.4e-4	3.17e-2	1.43e-2	4.05e-2	2.5e-3	1.41e-2	6.7e-3	INF	INF
DTLZ7	2.1e-2	4.5e-3	2.21e-2	3.1e-3	4.79e-2	4.1e-3	2.25e-2	3.7e-3	1.39e-2	2.6e-3

表4 各算法在基准多目标测试问题中的MS均值与方差

Problem	DCMOEA		MOEA/D		NSGA-II		SPEA2		PAES	
	GD(avg)	GD(σ)	GD(avg)	GD(σ)	GD(avg)	GD(σ)	GD(avg)	GD(σ)	GD(avg)	GD(σ)
ZDT1	1.0e0	0.0e0	9.90e-1	1.66e-2	1.0e0	0.0e0	1.0e0	2.6e-4	9.79e-1	2.6e-4
ZDT2	1.0e0	0.0e0	9.97e-1	7.4e-3	1.0e0	0.0e0	1.0e0	2.2e-5	9.94e-1	5.6e-3
ZDT3	1.17e0	6.4e-4	1.17e0	2.7e-3	1.15e0	3.43e-2	1.16e0	3.9e-4	1.11e0	4.7e-2
ZDT6	7.81e-1	6.70e-5	7.81e-1	3.02e-5	7.80e-1	6.2e-4	7.81e-1	3.1e-6	7.79e-1	1.5e-3
DTLZ1	6.12e-1	1.2e-16	5.0e-1	0.0e0	6.12e-1	1.2e-16	6.12e-1	1.2e-16	6.11e-1	5.4e-3
DTLZ2	1.22e0	6.30e-5	1.0e0	0.0e0	1.22e0	2.3e-16	1.22e0	2.3e-16	1.10e0	6.85e-2
DTLZ4	1.16e0	1.0e-5	9.71e-1	9.26e-2	1.16e0	2.1e-16	1.16e0	9.4e-3	7.66e-1	1.2e-2
DTLZ7	3.80e0	2.0e-4	7.94e-1	1.09e-1	3.80e0	1.2e-3	3.80e0	1.7e-4	3.70e0	7.4e-3

图3和图4直观上反映出了DCMOEA在2-目标和3-目标的测试问题上具有较好的分布均匀性和宽广性.表2给出了各算法在8个测试问题中所获得的收敛性指标GD的均值和方差.从表2的数据可以看出,DCMOEA的收敛性指标要明显优于SPEA2、NSGA-II和PAES算法(如表2中粗体数据所示),而它与MOEA/D的收敛性相差不大.其中DCMOEA在ZDT2、ZDT3、ZDT6和DTLZ1这4个测试问题上获得最好的GD均值,而MOEA/D则在ZDT1、DTLZ2、DTLZ4和DTLZ7这4个测试函数上获得最好的GD均值.但结合图4来看,MOEA/D在DTLZ4和DTLZ7问题上并未收敛到真实的Pareto前沿.如果从收敛性指标来看,SPEA2要略优于NSGA-II,PAES的表现则是几个算法中最差的.

表3给出了各算法分布均匀性指标SP的统计值,明显地反映出SPEA2的分布均匀性在所有5个算法中是最好的,这一点在图3和图4中得到了印证.而DCMOEA算法的均匀性要好于MOEA/D、NSGA-II和PAES,究其原因,DCMOEA中的支配易于在Pareto前沿的极端位置形成较大的网格,偌大的网格内仅能保留一个非支配解,从而影响了整个解集的SP值.值得一提的是,DCMOEA在DTLZ1和DTLZ7这两个测试问题上的分布均匀性要优于SPEA2算法.表3中的数据还反映出,在2-目标测试问题中MOEA/D的分布均匀性要好于DCMOEA,而在3-目标测试函数中DCMOEA的分布均匀性要优于MOEA/D.

表4给出了各算法在8个测试问题上获得的解集对真实Pareto前沿覆盖程度的度量.从表4可看出,DCMOEA的宽广性能总体上是最好的,DCMOEA在ZDT1和ZDT2问题上与NSGA-II的宽广性能相当,而它在ZDT3问题上的表现好于MOEA/D、NSGA-II、SPEA2和PAES.在ZDT6问题上DCMOEA的MS均值好于NSGA-II和PAES,而与MOEA/D和SPEA2的MS均值相当.在DTLZ1、DTLZ2、DTLZ4和DTLZ7这4个测试问题上,DCMOEA、NSGA-II和SPEA2这3个算法均获得最好的MS均值,这一点也可从图4中反映出来.

另外,从5个算法的时间复杂性来看,MOEA/D算法在每一代的时间复杂度为 $O(nTN)^{[7]}$ (这里的 T 为算法自定义的邻域大小),由于 T 通常小于 N ,DCMOEA的时间复杂性 $O(MN^2)$ 不如MOEA/D算法好,但二者处于相同的数量级.而DCMOEA的时间复杂性优于SPEA2的时间复杂性 $O(MN \log N)^{[3]}$,它与PAES算法的时间复杂性 $O(\alpha MN)^{[6]}$ (α 为档案大小)相比,若档案 α 的规模取与种群规模大小相同的值,则DCMOEA算法的复杂度与PAES算法相当.DCMOEA与NSGA-II的时间复杂度 $O(MN^2)^{[5]}$ 相比,它们的复杂度处于相同水平.

综合5个算法在8个基准多目标测试函数中的性能表现,并从各算法时间复杂性的比较来看,DCMOEA的整体性能是最好的.由此可见,本文提出的算法在多目标进化算法中颇具竞争力.

5 结 论

主流多目标进化算法一般采用单链模式,这种基因结构使个体携带的遗传信息少,容易导致种群的退化.本文提出了一种双链结构的多目标进化算法DCMOEA以克服单链模式存在的不足.DCMOEA与MOEA/D、NSGA-II、SPEA2和PAES一同在4个2-目标的ZDT问题和4个3-目标的DTLZ问题上进行比较实验,通过考察算法在收敛性、分布均匀性和解集宽广性3个方面的性能比较了它们的时间复杂性,得出的结论是:DCMOEA算法的综合性能最好,MOEA/D、NSGA-II和SPEA2三个算法的性能各有优劣,其中综合性能最差的则是PAES算法.另外,本文还分析了影响算法性能的潜在因素,以利于将来对算法作进一步改进.未来将用更多的测试问题更全面地考察DCMOEA的性能,还要将DCMOEA用于各类工程优化问题的应用中,在实践中不断完善DCMOEA算法.

参考文献(References)

- [1] Schaffer J D. Multi objective optimization with vector evaluated genetic algorithms[C]. Proc of the Int Conf on Genetic Algorithms and Their Applications. Hillsdal: L Erlbaum Associates, Inc, 1985: 93-100.

- [2] Zitzler E, Thiele L, Lingle. Multi-objective evolutionary algorithms: A comparative case study and the strength Pareto approach[J]. IEEE Trans on Evolutionary Computation, 1999, 3(4): 257-271.
- [3] Zitzler E, Laumanns M, Thiele L. SPEA2: Improving the strength Pareto evolutionary algorithm[C]. Proc of Int Conf on Evolutionary Method for Design, Optimization and Control with Applications to Industrial Problems. Berlin: Springer, 2002: 95-100.
- [4] Srinivas N, Deb K. Multiobjective optimization using non-dominated sorting in genetic algorithms[J]. Evolutionary Computation, 1994, 2(3): 221-248.
- [5] Deb K, Pratab A, Agarwal S, et al. A fast and elitist multi-objective genetic algorithm: NSGA-II[J]. IEEE Trans on Evolutionary Computation, 2002, 6(2): 182-197.
- [6] Knowles J D, Corne D W. Approximating the non-dominated front using the Pareto archived evolution strategy[J]. Evolutionary Computation, 2000, 8(2): 149-172.
- [7] Qingfu Zhang, Hui Li. MOEA/D: A multiobjective evolutionary algorithm based on decomposition[J]. IEEE Trans on Evolutionary Computation, 2007, 11(6): 712-731.
- [8] Goldberg, David E, Korb B, et al. Messy genetic algorithms: Motivation, analysis, and first results[J]. Complex Systems, 1989, 3(5): 493-530.
- [9] David A, Van Veldhuizen. Multiobjective optimization with messy genetic algorithms[C]. Proc of 2000 ACM Symposium On Applied Computing. Villa Olmo, 2000, 3: 470-476.
- [10] Jesse B, Zydallis, David A, et al. A statistical comparison of multiobjective evolutionary algorithms including the MOMOEA-II[C]. Proc of the 1st Int Conf on Evolutionary Multi-Criterion Optimization. 2001: 226-240.
- [11] Mitsukuni Matayoshi. Double chromosome GA with corner junction for solving the 2D strip packing problem[C]. IECON the 36th Annual Conf on IEEE Industrial Electronics Society. Glendale, 2010: 1110-1116.
- [12] Chen Guo, Ming Huang, Xu Liang. An improvement diploid genetic algorithm for job-shop scheduling problem[C]. The 18th Int Conf on Industrial Engineering and Engineering Management. Singapore, 2011, 9: 36-38.
- [13] Schopf J W. The evolution of the earliest cells[M]. Berlin: Scientific American, 1978: 110-138.
- [14] Fischer M, Hock M, Paschke M. Low genetic variation reduces cross compatibility and offspring fitness in populations of a narrow endemic plant with a self-incompatibility system[C]. Conservation Genetics. Netherlands: Kluwer Academic, 2011: 325-336.
- [15] Eckart Zitzler, Marco Laumanns, Stefan Bleule. A tutorial on evolutionary multiobjective optimization[C]. Workshop on Multiple Objective Metaheuristics. Berlin: Springer, 2004: 1-32.
- [16] Deb K, Mohan M, Mishra S. Evaluating the ϵ domination based multi-objective evolutionary algorithm for a quick computation of Pareto optimal solutions[J]. Evolutionary Computation, 2005, 13(4): 501-525.
- [17] Deb Kalyanmoy. Multi-objective genetic algorithm: Problem difficulties and construction of test problems[J]. Evolutionary Computation, 1999, 7(3): 205-230.
- [18] Deb K, Lothar T, Morco L, et al. Scalable test problems for evolutionary multi-objective optimization[C]. Evolutionary Multi-objective Optimization. Berlin: Springer Verlag, 2005: 105-145.
- [19] van Veldhuizen D A, Lamont G B. On measuring multi-objective evolutionary algorithm performance[C]. Congress on Evolutionary Computation. Piscataway: IEEE Press, 2000: 204-211.
- [20] Schoot J R. Fault tolerant design using single and multicriteria genetic algorithm optimization[D]. Cambridge: Massachusetts Institute of Technology, 1995.
- [21] Zitzler E, Deb K, Thiele L. Comparison of multiobjective evolutionary algorithms: Empirical results[J]. Evolutionary Computation, 2000, 8(2): 173-195.

(责任编辑: 孙艺红)