

一种基于“探测”与“开采”的多目标进化算法^①

罗彪^{②*} 郑金华^{*} 朱云飞^{***} 蔡自兴^{**}

(* 湘潭大学进化计算与智能系统研究中心 湘潭 411105)

(** 中南大学信息科学与工程学院 长沙 410083)

摘要 针对实践中多目标优化问题(MOPs)的 Pareto 解集(PS)未知且比较复杂的特性,提出了一种基于“探测”(Exploration)与“开采”(Exploitation)的多目标进化算法(MOEA)——MOEA/2E。该算法在进化过程中采用“探测”与“开采”相结合的方法,用进化操作不断地探测新的搜索区域,用局部搜索充分开采优秀的解区域,并用隐最优个体保留机制保存每一代的最优个体。与目前最流行且有效的多目标进化算法 NSGA-II 及 SPEA-II 进行的比较实验结果表明,MOEA/2E 获得的 Pareto 最优解集具有更好的收敛性与分布性。

关键词 多目标进化算法,多目标优化问题(MOPs),复杂 Pareto 解集,探测,开采

0 引言

实践中的许多优化问题均为多目标优化问题(multi-objective optimization problems, MOPs),多个目标之间一般是相互冲突的,因而,MOPs 的最优解是一个解的集合,这个集合称为 Pareto 解集(Pareto set, PS)。多目标进化算法(multi-objective evolutionary algorithms, MOEAs)是目前最为流行且有效的解决 MOPs 的智能算法,在过去的 20 年里有了快速的发展。作为一种全局搜索算法,MOEA 通过模拟生物进化来解决多目标优化问题,代表性的算法有 Deb 等提出的 NSGA-II^[1],Zitzler 等提出的 SPEA-II^[2],等等。MOEAs 适用于处理复杂的多目标优化问题,因而被广泛地应用于函数优化、智能控制、机器人路径规划等^[3]众多方面。无论是在科研领域,还是在工程领域,对 MOEAs 的研究引起了广泛的关注。

但是,多数研究在测试 MOEAs 性能时,使用的测试函数的 PS 均比较简单,如 Deb 等^[4]提出的 ZDT 系列测试函数的 PS 只在第一维变量上取值,在其余的各变量均为 0。这样的 MOPs 测试函数很难反映出实际优化问题的情况,也很难测试出 MOEAs 的性能。而且,随着 MOEAs 在工程领域的应用,各种 MOPs 对 MOEAs 提出了更高的要求。在实践中,PS 往往是未知且比较复杂的,所以,用 MOEAs 优化具

有更为复杂 PS 的 MOPs 有着重要的现实意义。为此,本文提出了一种基于“探测”和“开采”的 MOEA (MOEA based on exploration and exploitation, MOEA/2E)。MOEA/2E 利用“探测”的方法寻找新的搜索区域,用“开采”的方法搜索目前较优秀的解区域,从而能更好地优化复杂 PS 的 MOPs。

1 有关概念

本节讨论 MOPs 的相关概念。

定义 1 (MOP:多目标优化问题) MOP 的一般描述为

$$\text{Min } F(X) = (f_1(X), \dots, f_r(X)) \quad (1)$$

$$g_i(X) \geq 0; (i = 1, 2, \dots, k) \quad (2)$$

$$h_i(X) = 0; (i = 1, 2, \dots, l) \quad (3)$$

其中, $F(X)$ 为目标函数向量, r 为维数; $X = (x_1, \dots, x_n)$ 为决策向量, n 为维数;若 $F: \Omega \rightarrow \Pi$, 则, $\Omega \subseteq R^n$, $\Pi \subseteq R^r$, Ω 为称决策空间,也称为可行解空间; Π 是目标空间;式(2)和式(3)分别为不等式约束和等式约束;多目标优化的目的就寻找最优解 $X^* = (x_1^*, x_2^*, \dots, x_n^*)$, 使 $F(X^*)$ 在满足约束式(2)和式(3)的条件下同时达到 Pareto 最优。

定义 2 (解的支配关系) 对于定义 1 的 MOP, $\forall p, q \in \Omega$, 如果 p 支配 q , 必需满足下面两

① 国家自然科学基金(60773047), 863 计划(2001AA114060), 湖南省教育厅重点科研项目(06A074)和湖南省研究生科研创新项目(x2008yjcx18)资助项目。

② 男, 1984 年生, 硕士; 研究方向: 计算智能; 联系人, E-mail: biao.luo@hotmail.com
(收稿日期: 2008-11-17)

个条件:

(1) 对 $\forall k = 1, 2, \dots, r$, 有 $f_k(p) \leq f_k(q)$ 。

(2) $\exists l \in \{1, 2, \dots, r\}$, 使 $f_l(p) < f_l(q)$ 。

此时称 p 为非支配解, q 为支配解。MOPs 的最优解通常称为 Pareto 最优解。

定义 3 (Pareto 最优解) MOPs 的全局非支配解称为 Pareto 最优解。

由于 Pareto 最优解往往不止一个, 而是一个解的集合, 这里用 PS 表示。

定义 4 (Pareto 最优解集) 对于定义 1 的 MOP, 它的 Pareto 最优解集定义为

$$PS = \{X \in \Omega \mid \neg \exists X' \in \Omega, f_j(X') \leq f_j(X), (j = 1, \dots, r)\} \quad (5)$$

在本文中, 种群的 Pareto 最优解集也称为非支配解集(non-dominated set, NDS), 非支配解集的构造是根据个体间的支配关系得到的。MOEAs 的任务就是在进化过程中构造出当前种群的非支配解集, 使非支配集逐渐逼近真实的 Pareto 最优解集。

定义 5 (Pareto 最优边界) 对于定义 1 的 MOP, Pareto 最优边界(Pareto front, PF)定义为

$$PF = \{F(X) \mid X \in PS\} \quad (6)$$

Pareto 最优解集与 Pareto 最优边界是不同的, 前者属于决策空间, 而后者属于目标空间。在下文中, 用 PF 表示算法得到的非支配集, PF_{True} 表示真实的 PF。

2 提高 PS 的复杂程度——变量关联

过于简单的 PS 无法反映实际优化问题, 也无法反映 MOEAs 的真实性能。

本节采用变量关联^[5](variable linkages)方法来改变 PS 的形状, 以提高 PS 的复杂程度。在此之前, 我们先介绍 ZDT 系列测试函数的构造方法。

ZDT 系列测试函数均满足表达式

$$\begin{aligned} \text{Min } f_1(X) &= f_1(X_I) \\ f_2(X) &= g(X_{II}) \cdot h(f_1(X_I), g(X_{II})) \end{aligned} \quad (7)$$

其中, 决策向量 X 被拆成两个没有重叠的分向量 X_I 与 X_{II} , 即: $X = (X_I, X_{II})$; f_1 、 g 与 h 在测试函数的构造中, 起着不同的作用:

f_1 : 用于刻画 PF_{True} 上的解集的均匀性与分布性;

g : 用于定义 PS 的形状;

h : 用于定义 PF_{True} 的形状。

这样, 通过设计合理的函数 f_1 、 g 与 h , 就构造出了 ZDT 系列测试函数, 表 1 为 ZDT1 与 ZDT2 测试

函数。

表 1 ZDT1 与 ZDT2 测试函数

测试问题	目标函数	特征
ZDT1	$f_1(x_1) = x_1; f_2(X) = g(1 - \sqrt{(f_1/g)})$ $g(X) = 1 + 9 \sum_{i=2}^n x_i / (n-1)$	PF_{True} 为凸 $X \in [0, 1]^n$
ZDT2	$f_1(x_1) = x_1; f_2(X) = g(1 - (f_1/g)^2)$ $g(X) = 1 + 9 \sum_{i=2}^n x_i / (n-1)$	PF_{True} 为凹 $X \in [0, 1]^n$

ZDT 系列测试函数具有以下特点: (1) $g(X)$ 取最小值时决策变量的取值即为 PS, 即: $PS = \{X^* \mid \min g(X^*)\}$ 。(2) 改变 $g(X)$ 可以改变 PS, 但不会改变相应的 PF_{True} ; ZDT1 与 ZDT2 的 PF_{True} 分别为

$$f_2(X) = 1 - \sqrt{f_1(X)} \quad (8)$$

$$f_2(X) = 1 - [f_1(X)]^2 \quad (9)$$

变量关联通过设计 $g(X)$ 来得到不同形状的 PS, 以提高 PS 的复杂程度, 由 ZDT 系列测试函数的第(2)个特点, 变量关联后测试函数的 PF_{True} 不变, 仍为式(8)和式(9)。

本文将用线性变量关联^[6](linear variable linkages)与非线性变量关联^[6](nonlinear variable linkages)来改变测试函数的 PS。

(1) 线性变量关联

设计 $g(X) = 1 + 9 \sum_{i=2}^n (x_i - x_1)^2 / (n-1)$, 即保持 x_1 不变, 用 $x_i - x_1$ 代替 x_i 。

(2) 非线性变量关联

设计 $g(X) = 1 + 9 \sum_{i=2}^n (x_i^2 - x_1)^2 / (n-1)$, 即保持 x_1 不变, 用 $x_i^2 - x_1$ 代替 x_i 。

采用线性变量关联与非线性变量关联就构造出了本文的测试函数集(见表 2)

表 2 测试函数

测试问题	目标函数	特征
F1	$f_1(x_1) = x_1$ $f_2(X) = g(1 - \sqrt{(f_1/g)})$ $g(X) = 1 + 9 \sum_{i=2}^n ((x_i - x_1)^2 / (n-1))$	PF_{True} 为凸 $X \in [0, 1]^n$ $n = 20$ 线性变换
F2	$f_1(x_1) = x_1$ $f_2(X) = g(1 - (f_1/g)^2)$ $g(X) = 1 + 9 \sum_{i=2}^n ((x_i - x_1)^2 / (n-1))$	PF_{True} 为凹 $X \in [0, 1]^n$ $n = 20$ 线性变换

F3	$f_1(x_1) = x_1$	PF _{Test} 为凸 $X \in [0, 1]^n$ $n = 20$ 非线性变换
	$f_2(X) = g(1 - \sqrt{(f_1/g)})$	
	$g(X) = 1 + 9 \sum_{i=2}^n ((x_i^2 - x_i) / (n - 1))$	
F4	$f_1(x_1) = x_1$	PF _{Test} 为凹 $X \in [0, 1]^n$ $n = 20$ 非线性变换
	$f_2(X) = g(1 - (f_1/g)^2)$	
	$g(X) = 1 + 9 \sum_{i=2}^n ((x_i^2 - x_i) / (n - 1))$	
F5	$f_1(x_1) = x_1 + \frac{2}{ J_1 }$	PF _{Test} 为凸 $X \in [0, 1] \times [-1, 1]^{n-1}$ $n = 20$
	$\sum_{j \in J_1} [x_j - \sin(6\pi x_1 + \frac{j\pi}{n})]^2$	
	$f_2(X) = 1 - \sqrt{x_1} + \frac{2}{ J_2 }$	
	$\sum_{j \in J_2} [x_j - \sin(6\pi x_1 + \frac{j\pi}{n})]^2$	
	$J_1 = \{j j \text{ 是奇数且 } 2 \leq j \leq n\}$	
	$J_2 = \{j j \text{ 是偶数且 } 2 \leq j \leq n\}$	

3 基于“探测”与“开采”的 MOEA (MOEA/2E)

在本节中,我们先引入“探测”与“开采”的思想,再提出 MOEA/2E。

3.1 “探测”与“开采”

MOEA 作为一种群体搜索算法,在算法的执行过程中,存在较大的随机性。MOEAs 在搜索最优解时,也不是完全朝着有利的方向搜索,甚至会将好的解区域丢弃,从而产生退化现象。当 MOPs 的 PS 较为复杂时,这种现象更加突出。为此,本文提出“探测”与“开采”结合起来搜索解空间,目的在于减少退化现象的产生,进而提高算法的收敛性及解集的分布性。

本文提出的 MOEA/2E,将进化过程分成两个过程:一是“探测”过程,二是“开采”过程(图 1,图 2)。用“探测”方法定位优秀的解区域,用“开采”方法在这些优秀的解区域中充分搜索更优秀的解。

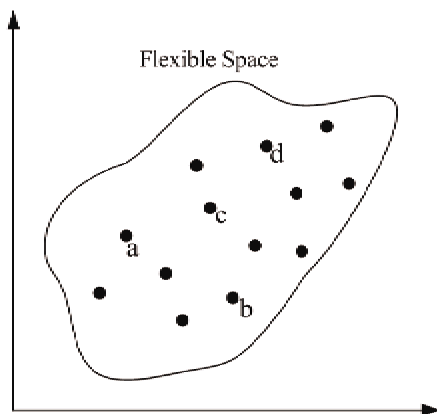


图 1 “探测”过程

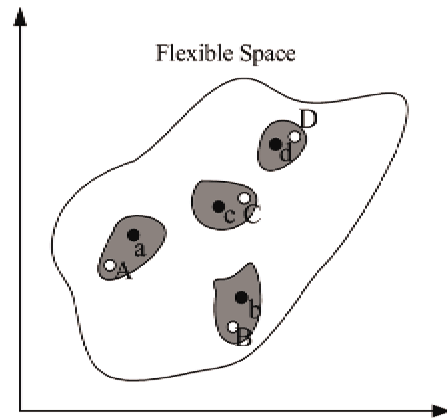


图 2 “开采”过程

在“探测”过程中,在整个可行解空间内搜索,使用进化操作(选择、交叉与变异)产生个体,通过非支配集构造方法寻找当前种群的非支配个体,使用“隐最优个体保留机制”(elitism)^[1](见 3.2 节)保存当前种群的优秀个体。如图 1,图中实心黑点为当前种群中通过进化操作产生的个体,经过“探测”过程后,可定位当前的优秀个体为 a、b、c 和 d。因此,“探测”过程具有产生和定位优秀个体的功能。

在“开采”过程中,在当前优秀个体的邻域内进行局部搜索。通常,在个体的周围往往还存在更优秀的个体,通过对优秀个体邻域的开采,可以实现更充分的搜索。如图 2,对于已探测到的优秀个体 a、b、c 和 d,通过对其邻域进行开采,可以得到更优秀的个体 A、B、C 和 D。因此,“开采”过程具有有效利用当前优秀个体信息,进行充分搜索的能力。

在具体的 MOEA/2E 实现中,我们对进化过程中的每一代种群,采用“开采”与“探测”相结合的方法产生。(1) 在“开采”过程中,用局部搜索在当前种群中的非支配个体的邻域内产生一部分下一代种群。这样可以开采目前较优秀的搜索区域。(2) 在“探测”过程中,用进化操作产生另一部分下一代种群。这样可以探测新的搜索区域,用于保持种群的多样性,防止算法陷入局部收敛。(3) 用“隐最优个体保留机制”保存进化过程中的优秀个体,用擂台赛法则^[7](arena's principle, AP)构造种群的非支配集。

3.2 算法设计

设种群 $Pop(t)$ 大小为 N ; 混合种群为 $Merge_Pop$, 其大小为 $2N$; 临时种群为 Pop' , 其大小为 M , 即为当前种群中非支配集的大小; 进化代数 G 。

算法 1 MOEA/2E

步骤 1 $t = 0$; 初始化种群 $Pop(0)$; $Merge_Pop = \Phi$;
步骤 2 用 AP 构造非支配集; 对 $Pop(0)$ 中的非支

配个体,标记为 $rank = 1$;

步骤 3 Exploit ($Pop(t)$);

步骤 4 Explore ($Pop(t)$);

步骤 5 $t = t + 1$;

步骤 6 若 $t \geq G$,则结束 MOEA/2E,将 $Pop(t)$ 作为输出;否则返回步骤 3 继续执行。

算法 2 “开采”Exploit ($Pop(t)$)

步骤 1 $Pop' = \Phi$;将 $Pop(t)$ 中所有 $rank = 1$ 的个体放入 Pop' 中;设此时 $|Pop'| = M$;

步骤 2 $Q = \Phi$; $i = 1$;

步骤 3 设 X_i 为 Pop' 中的第 i 个个体,在 X_i 的 δ -邻域 B_δ 内随机产生一个新的个体 X' ;

步骤 4 $Q = Q \cup X'$;

步骤 5 $i = i + 1$;

步骤 6 若 $i \leq M$,则返回步骤 3;

步骤 7 对 Q 中的每一个个体以概率 p'_m 进行微变异(p'_m 要远小于进化操作中的变异概率 p_m);

步骤 8 $Merge_Pop = Merge_Pop \cup Pop' \cup Q$;此时, $|Merge_Pop| = 2M$ 。

算法 2 中 X_i 的 δ -邻域是一个以 X_i 为中心,以 δ 为半径的超立方体,为 $B_\delta = [x_i^j - \delta, x_i^j + \delta]$, ($j = 1, \dots, n$)。

算法 3 “探测”Explore ($Pop(t)$)

步骤 1 在 $Pop(t)$ 中利用进化操作产生 $2(N - M)$ 个体。将这些个体并入 $Merge_Pop$ 中;此时, $|Merge_Pop| = 2N$;

步骤 2 $i = 1$; $Pop(t + 1) = \Phi$;

步骤 3 用 AP 对 $Merge_Pop$ 构造非支配集并进行分层排序^[1],使第 i 层非支配个体的 $rank = i$, N_i 为第 i 层非支配集的大小;

步骤 4 分两种情况:1)若 $|Pop(t + 1)| + N_i \leq N$,则将 $Merge_Pop$ 中 $rank$ 为 i 的所有个体并入 $Pop(t + 1)$ 中;2)若 $|Pop(t + 1)| + N_i > N$,用动态聚集距离^[8] (dynamic crowding distance, DCD)来对 $Merge_Pop$ 中 $rank$ 为 i 的个体来进行种群维护,逐步删除分布性差的个体,直到 $|Pop(t + 1)| = N$;结束算法 3,将 $Pop(t + 1)$ 返回;(条件 2 同时也为算法 3 的结束条件);

步骤 6 将 $Merge_Pop$ 中 $rank = i$ 的所有个体从 $Merge_Pop$ 中除去;

步骤 7 $i = i + 1$;返回 Step3。

在 MOEA/2E 中,将父代种群中的非支配集 $Pop'(t)$ 与新产生的个体一起并入混合种群 $Merge_Pop$ 中,再对 $Merge_Pop$ 构造非支配集,这样就可以逐步将进化过程中非支配个体保留下来,称这种机制为“隐最优个体留机制”。

4 实验仿真

参与比较实验的 MOEAs,除了本文提出的 MOEA/2E 外,还有目前最流行且效果很好的两个 MOEAs: NSGA-II 和 SPEA-II。算法的参数设置见表 3。

表 3 算法的参数设置

进化代数	种群规模	邻域半径 $\delta_i (i = 1, \dots, n)$	交叉概率 P_c	变异概率 P_m	微变异概率 P'_m
500	200	0.05	0.9	$1/n$	0.0001

说明:表中 n 为决策变量的维数。

4.1 性能评价指标

本文使用的性能评价指标有反向世代距离^[9] (inverted generational distance, IGD) 和最大分布广度^[10] (maximum spread, MS)。

(1) IGD 定义为

$$IGD = \frac{\sum_{v \in P^*} d(v, P)}{|P^*|} \quad (10)$$

其中, $d(v, P)$ 表示个体 v 到解集 P 中个体的最小欧几里德距离; P^* 为在 PF_{True} 上均匀取的一定数目的个体,在本文的实验中,我们取 $|P^*| = 500$; P 为使用 MOEAs 得到的非支配集。IGD 越小越好,可以看出,只有当解集的收敛性与分布性都好时,IGD 才会很小。因此,如果 $|P^*|$ 足够大,IGD 既能反映解集的收敛性,同时也能反映分布性。正如 IGD 名字所反映的一样,IGD 为我们熟悉的 GD^[11] 的一种反向思想。由于 GD 表示种群 P 到 PF_{True} 的平均距离,它只能反映 P 对 PF_{True} 的近似程度,而无法反映 P 的分布性。

(2) MS 定义为

$$MS = \sqrt{\frac{1}{r} \sum_{i=1}^r \left(\frac{\min(f_i^{\max}, F_i^{\max}) - \max(f_i^{\min}, F_i^{\min})}{F_i^{\max} - F_i^{\min}} \right)^2} \quad (11)$$

其中, r 为目标维数, $F_i^{\max}, F_i^{\min}$ 分别为理论上的第 i 维目标的最大值与最小值; $f_i^{\max}, f_i^{\min}$ 分别为算法得

到解集的第 i 维目标的最大值与最小值。MS 用于衡量解集的分布广度,MS 的值越大,说明解集的跨度越大,最大值为 $MS = 1$ 。

4.2 测试函数

本文使用的测试函数见表 2,由表 1 中的 ZDT1、ZDT2 通过线性变量关联分别得到 F1 和 F2;通过非

线性变量关联分别得到 F3 和 F4;F5 选自于 Zhang 等人^[12]的报告。

4.3 实验及数据分析

在表 3 所设置的参数环境下,对表 2 中的测试函数,分别利用 MOEA/2E、NSGA-II 与 SPEA-II 进行优化,得到的非支配解集(PF)分别见于图 3-图 7。

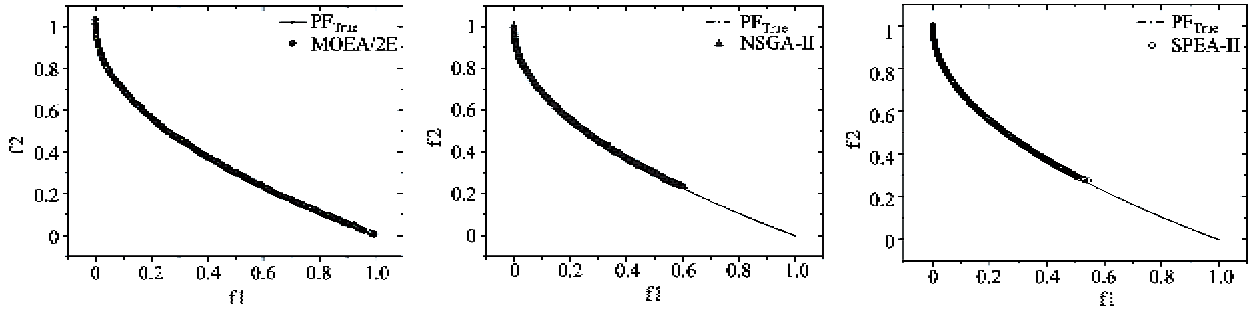


图 3 三算法(依次为: MOEA/2E, NSGA-II 和 SPEA-II)优化 F1 得到的 PF

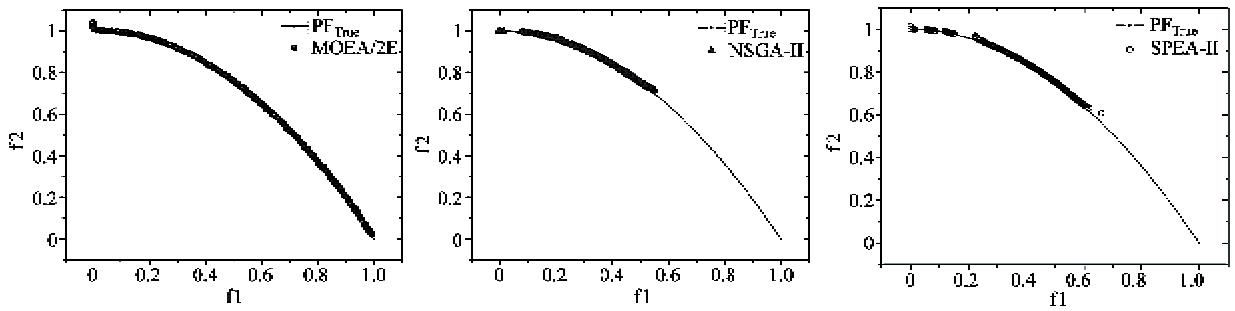


图 4 三算法(依次为: MOEA/2E, NSGA-II 和 SPEA-II)优化 F2 得到的 PF

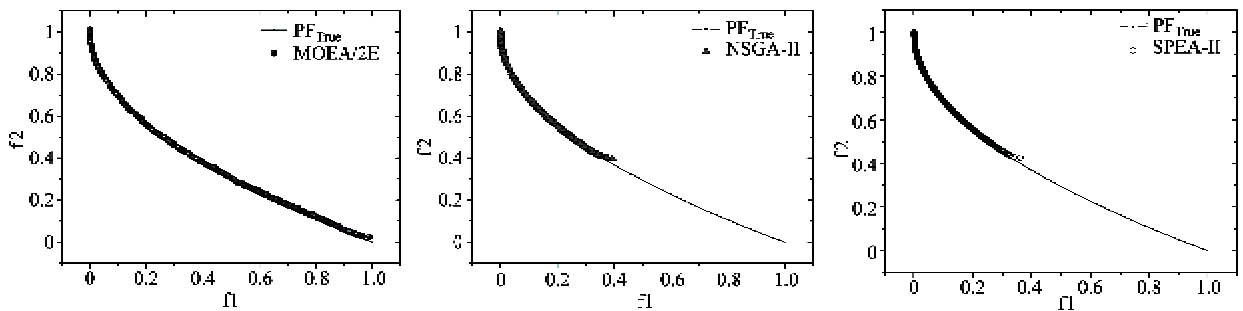


图 5 三算法(依次为: MOEA/2E, NSGA-II 和 SPEA-II)优化 F3 得到的 PF

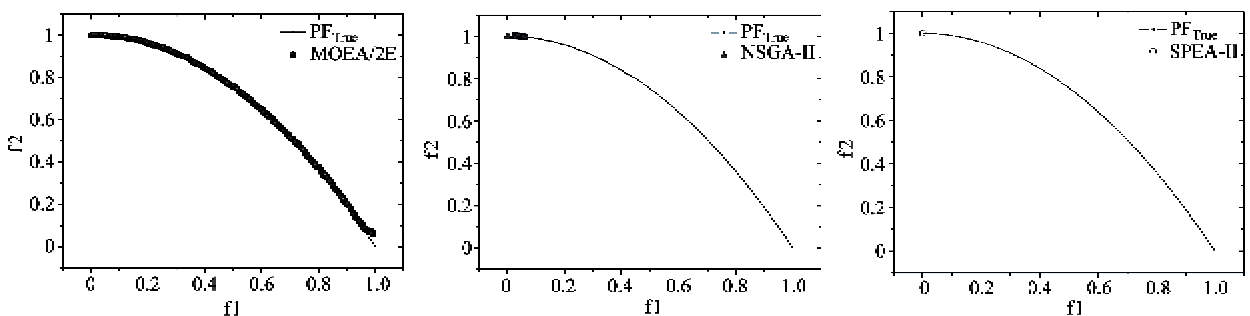


图 6 三算法(依次为: MOEA/2E, NSGA-II 和 SPEA-II)优化 F4 得到的 PF

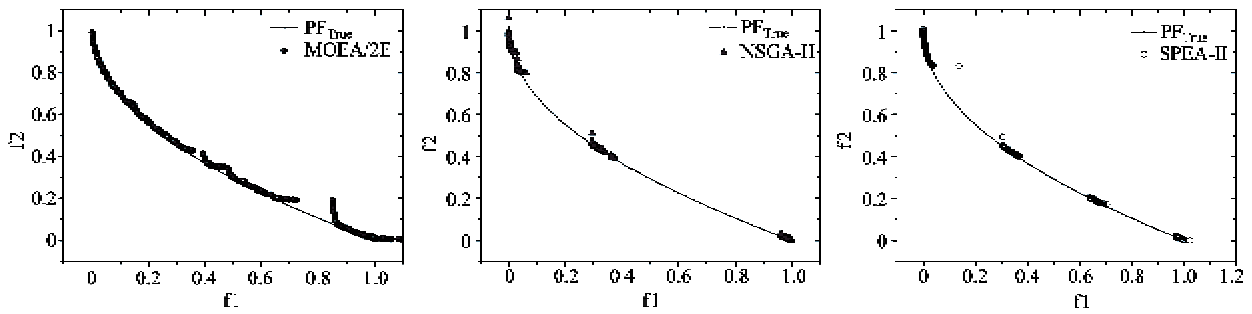


图 7 三算法(依次为: MOEA/2E, NSGA-II 和 SPEA-II)优化 F5 得到的 PF

对于测试函数 F1-F4, NSGA-II 与 SPEA-II 均只能找到 PF_{True} 上的一部分解集;而 MOEA/2E 得到的 PF 几乎覆盖了整个 PF_{True} 区域。对于 F5, MOEA/2E 在 PF_{True} 找到的解也比 NSGA-II 与 SPEA-II 要多。这从直观上说明了 MOEA/2E 得到的解集有更好的收敛性及分布广度。

为了定量地比较这三个算法,我们在表 3 的参数环境下,使用 MOEA/2E、NSGA-II 与 SPEA-II 优化 F1-F5 各 20 次,将 20 次的 IGD 与 MS 的平均值分别列于表 4 与表 5 中。从表 4 中可以看出,MOEA/2E 得到的 IGD 均明显小于 NSGA-II 与 SPEA-II 所得到的 IGD,这说明 MOEA/2E 得到的解集的收敛性与分布性优于 NSGA-II 与 SPEA-II。从表 5 可知,对于 F1-F4, MOEA/2E 的 MS 值远大于 NSGA-II 与 SPEA-II 的 MS 值,对于 F5,三算法的 MS 值相差不大。

表 4 用 MOEA/2E、NSGA-II 与 SPEA-II 分别优化 F1-F5 各 20 次,得到的 IGD 平均值

测试函数	F1	F2	F3	F4	F5
MOEA/2E	0.006669	0.003925	0.024764	0.356825	0.023645
NSGA-II	0.085448	0.213004	0.287394	0.609671	0.084904
SPEA-II	0.088317	0.140576	0.246571	0.605310	0.068340

表 5 用 MOEA/2E、NSGA-II 与 SPEA-II 分别优化 F1-F5 各 20 次,得到的 MS 平均值

测试函数	F1	F2	F4	F4	F5
MOEA/2E	0.995883	0.974688	0.987041	0.877401	0.999290
NSGA-II	0.690385	0.462657	0.511628	0.025148	0.999164
SPEA-II	0.410460	0.543048	0.480144	0.000264	0.998863

解集的收敛性和分布性是衡量 MOEAs 性能的两个最重要指标。通过上述比较分析可知, NSGA-II 与 SPEA-II 在优化 PS 较为复杂的 MOPs 时,只能得到 PF_{True} 一部分,解集的分布广度较差。同时,通过 IGD 与 MS 的比较,发现 MOEA/2E 得到的解集比

NSGA-II 与 SPEA-II 具有更好的收敛性与分布性。

5 结论

目前在测试 MOEAs 时,使用的测试函数的 PS 较为简单,很难反映 MOEAs 的性能。同时,较为复杂的 PS 更能反映实际优化问题的特征,因为实践中 MOPs 的 PS 往往是未知且比较复杂的。因此,提高 MOEAs 优化此类 MOPs 的能力,具有很重要的现实意义。

本文提出的 MOEA/2E,在进化过程中采用“探测”与“开采”相结合的方法,用进化操作不断地探测新的搜索区域,用局部搜索充分开采优秀的解区域,并用隐最优个体保留机制保存每一代的最优个体。与目前最流行且有效的多目标进化方法 NSGA-II 和 SPEA-II 进行实验比较的结果表明, MOEA/2E 优化 PS 较为复杂的 MOPs 得到的解集具有更好的收敛性与分布性。

参考文献

- [1] Deb K, Pratap A, Agarwal S, Meyarivan T. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 2002, 6(2):182-197
- [2] Zitzler E, Thiele L. Multiobjective evolutionary algorithms: A comparative case study and the strength pareto approach. *IEEE Transactions on Evolutionary Computation*, 1999, 3(4):257-271
- [3] 郑金华. 多目标进化算法及其应用. 北京: 科学出版社, 2007
- [4] Deb K. Multi-objective genetic algorithms: Problem difficulties and construction of test problems. *Evolutionary Computation*, 1999, 7(3), 205-230
- [5] Deb K, Sinha A, Kukkonen S. Multi-objective test problems, linkages, and evolutionary methodologies. In: *Proceedings of Genetic and Evolutionary Computation Conference*, Seattle, Washington D.C., USA, 2006. 2. 1141-1148

- [6] Zhang Q, Zhou A, Jin Y. RM-MEDA: A regularity model based multiobjective estimation of distribution algorithm. *IEEE Transactions on Evolutionary Computation*, 2008, 12 (1): 41-63
- [7] 郑金华, 蒋浩, 邝达等. 擂台赛法则构造多目标 Pareto 最优解集的方法研究. *软件学报*, 2007, 18(6): 1287-1297
- [8] Luo B, Zheng J, Xie J, et al. Dynamic crowding distance a new diversity maintenance strategy for MOEAs. In: *Proceedings of the 4 th International Conference on Natural Computation*, Jinan, China, 2008, 1: 580-585
- [9] Van Veldhuizen D A. Multiobjective Evolutionary Algorithms: Classifications, Analyses, and New Innovations: [Ph. D dissertation]. Wright-Patterson AFB, Ohio: Air Force Institute of Technology, 1999
- [10] Zitzler E, Deb K, Thiele L. Comparison of multiobjective evolutionary algorithms: Empirical results. *Evolutionary Computation*, 2003, 8(2): 117-132
- [11] Van Veldhuizen D A, Lamont G B. Evolutionary computation and convergence to a Pareto front. In: *Proceedings of the Genetic Programming Conference*, Madison, WI, USA, 1998. 221-228
- [12] Zhang Q, Zhou A, Zhao S, et al. Multiobjective optimization Test Instances for the CEC 2009 Special Session and Competition. <http://www.cec.2009.org>; IEEE, 2008

A multi-objective evolutionary algorithm based on “exploration” and “exploitation”

Luo Biao^{*}, Zheng Jinhua^{*}, Zhu Yunfei^{**}, Cai Zixing^{**}

(^{*} Research Center of Evolutionary Computation and Intelligent System, Xiangtan University, Xiangtan 411105)

(^{**} College of Information Science and Engineering, Central South University, Changsha, 410083)

Abstract

In view of the fact that Pareto Set (PS) of multi-objective optimization problems (MOPs) is often unknown and complex in practice. This paper proposes a multi-objective evolutionary algorithm (MOEA) based on “Exploration” and “Exploitation”, named MOEA/2E. This algorithm combines “Exploration” and “Exploitation” in the evolutionary process. It explores new searching areas with evolutionary operators, exploits promising areas effectively with local search and stores optimal individual of a population with elitism. Compared with two popular and efficient MOEAs—NSGA-II and SPEA-II, the experimental results demonstrate that MOEA/2E can obtain Pareto optimal solutions set with better convergence and diversity.

Key words: multi-objective evolutionary algorithms, multi-objective optimization problems, complex Pareto set, exploration, exploitation