文章编号:1001 - 9081(2008)06 - 1570 - 05

一种基于邻域的多目标进化算法

李密青,郑金华,罗 彪,伍 军,文诗华
(湘潭大学 信息工程学院, 湖南 湘潭 411105)
(limit1008@126.com)

摘 要:种群维护是多目标进化算法的重要组成部分。针对维护方法和运行效率的矛盾,提出一种基于邻域的多目标进化算法(NMOEA)。定义了一个反映个体之间邻近程度的指标——邻域包含关系,利用此关系对个体进行分布适应度分级的赋值,并用动态方法快速地对种群进行维护。通过7个测试问题和3个方面的测试标准,结果表明新算法在较快速地接近真实的最优面的同时,拥有良好的分布性。

关键词:多目标进化算法;多目标优化问题;种群维护;分布适应度;邻域
中图分类号: TP18 **文献标志码**:A

Multi-objective evolutionary algorithm based on neighborhood

LI Mi-qing, ZHENG Jin-hua, LUO Biao, WU Jun, WEN Shi-hua
(*Institute of Information Engineering, Xiangtan University, Xiangtan Hunan* 411105, *China*)

Abstract: Population maintenance is an important issue in multi-objective evolutionary algorithms. For the deficiency that the maintenance methods of good distribution usually have a high time complexity, a multi-objective evolutionary algorithm based on neighborhood (named NMOEA) was proposed. This measure defined a criterion-neighborhood containing relation, which represented the close degree of individuals. And it was used to assign diversity fitness in a dynamic method that maintained the population rapidly. By examining three performance metrics on seven test problems, the new algorithm can approach the true Pareto front fast, and has good distribution.

Key words: multi-objective evolutionary algorithm; multi-objective optimization problem; population maintenance; diversity rank; neighborhood

0 引言

进化算法(Evolutionary Algorithm, EA)是一类模拟生物自然选择与自然进化的随机搜索算法,因其适用于求解高度复杂的非线形问题而得到了非常广泛的应用。在解决只有单个目标的复杂系统优化问题时,进化算法的优势得到了充分展现。然而,现实世界中的优化问题通常是多属性的,一般是对多个目标的同时优化,且被同时优化的多个目标之间通常是相互冲突的,如在企业生产活动中,产品质量与生产成本是两个相互冲突的目标。为了达到总目标的最优化,通常需要对相互冲突的子目标进行综合考虑,即对各个子目标进行折中。由此,针对多目标优化问题(Multi-objective Optimization Problem, MOP),出现了多目标进化算法(Multi-Objective Evolutionary Algorithm, MOEA)[1]。

近年来,进化计算界相继提出了大量有效的多目标进化算法。最有代表性的主要有:NSGA-II[2]和SPEA2[3]。NSGA-II有很快的运行效率,但解集的分布性并不理想;SPEA2能很好地维持解集的分布性,但速度较慢。

多目标进化算法的研究目标主要是使算法种群快速收敛,并且广泛而均匀分布于问题的非劣最优域。其中算法的时间效率是一个重要的因素。国内学者在这方面进行了大量的研究:文献[4]提出了基于排序集合快速求解算法;文献[5-6]利用快速排序和擂台赛法构造非支配集;文献[7]用占优树来减少个体的比较次数,提出了一种快速的基于占优树的多目标进化算法。以上这些方法都是围绕怎样构造非支配集的,算法虽然在时间上得到了较大提高,但收敛性和分布性的改进并不明显。

为了在拥有较快速度的同时,取得好的收敛性和分布性结果,本文提出一种基于邻域的多目标进化算法(NMOEA)。该方法首先定义了一个反映个体邻近程度的指标——邻域包含关系,然后利用邻域包含关系对个体进行分布适应度分级的赋值,并且用动态方法快速地对种群进行维护,最后通过进行多个函数实验,测试和讨论了算法的性能。实验结果表明算法具有良好的收敛性、分布性和时间效率。

1 基本概念

最小化与最大化问题可以相互转化,因此,仅以最小化多目标问题为研究对象。多目标问题的一般描述为:

给定决策向量 $X = (x_1, x_2, \cdots, x_n)$,它满足下列约束:

$$g_i(X) \geqslant 0;\ i = 1, 2, \cdots, k \tag{1}$$

$$h_i(X) = 0;\ i = 1, 2, \cdots l \tag{2}$$

设有 r 个优化目标,且这 r 个优化目标是相互冲突的,优化目标可表示为:

收稿日期:2007 - 12 - 21;**修回日期**:2008 - 02 - 20。 **基金项目**:国家自然科学基金资助项目(60773047);国家863计划项目(2001AA114060);留学回国人员科研启动基金资助项目(教外司留[2005]546号);湖南省自科基金资助项目(05JJ30125);湖南省教育厅重点科研资助项目(06A074)。

作者简介:李密青(1981 -),男,湖南常德人,硕士研究生,主要研究方向:进化计算; 郑金华(1963 -),男,湖南邵阳人,教授,博士生导师,主要研究方向:进化计算、智能科学; 罗彪(1984 -),男,湖南邵阳人,硕士研究生,主要研究方向:进化计算; 伍军(1982 -),男,湖南邵阳人,硕士研究生,主要研究方向:进化计算; 文诗华(1981 -),男,湖南衡阳人,硕士研究生,主要研究方向:进化计算。

$\vec{f}(X) = (f_1(X), f_2(X), \cdots, f(X))$

寻求 $X^* = (x_1^*, x_2^*, \cdots, x_n^*)$,使 $f(X^*)$ 在满足约束(1)和(2)的同时达到最小。MOEA 经常用到如下几个基本概念:

定义 1　个体的 Pareto 支配关系。设 p 和 q 是进化群体 Pop 中的任意两个不同的个体,我们称 p 支配 q,则必须满足下列 2 个条件:

1) 对所有的子目标,p 不比 q 差,即 $f_k(p) \leqslant f_k(q)(k = 1, 2, \cdots, r)$;

2) 至少存在一个子目标,使 p 比 q 好。即 $\exists l \in \{1,2,\cdots, r\}$,使 $f_l(p) < f_l(q)$。

其中 r 为子目标的数量。此时称 p 为非支配的,q 为被支配的。表示为 $p \succ\!\!- q$,其中"$\succ\!\!-$" 是支配关系。

定义 2　Pareto 最优解。给定一个多目标优化问题 $\min \vec{f}(X)$,称 $X^* \in \Omega$ 是最优解,若 $\forall X \in \Omega$,满足下列条件之一:

1) $\wedge (f_i(X) = f_i(X^*))$;

2) 至少存在一个 $j \in I, I = \{1,2,\cdots,r\}$,使: $f_j(X) > f_j(X^*)$。

其中 Ω 是满足式(1) 和式(2) 的可行解集,即:

$$\Omega = \{X \in \mathrm{R}^n \mid g_i(x) \geqslant 0, h_i(x) = 0; (i = 1,2,\cdots,k; j = 1,2,\cdots,l)\}$$

定义 3　Pareto 最优面(边界)。给定一个多目标优化问题 $\min \vec{f}(X)$ 和它的最优解集 $\{X^*\}$,它的 Pareto 最优面定义为:

$$PF^* = \{\vec{f}(X) = (f_1(X), f_2(X), \cdots, f_r(X)) \mid X \in \{X^*\}\} \tag{3}$$

定义 4　非支配集。设有解集 P,P 中的个体 q 不被任何其他个体支配,则 q 是 P 中的非支配个体;P 的非支配个体构成的子集称为 P 的非支配集 $NDset$。即 $NDset = \{q \mid q \in P$ 且 $\overline{\exists} p \in P$,使 $p \succ\!\!- q\}$。

2　相关工作

在进化过程中,种群中的非支配个体数可能大于外部种群的大小,此时需要对外部种群进行维护,通常的做法是根据非支配个体的密度信息移出部分个体。维护方法的好坏不仅决定着最终解集的分布情况,还很大程度上影响算法的搜索能力[8]。对此,学者们提出了很多行之有效的方法,如 NPGA[9] 采用了适应度共享机制来对种群进行修剪,它根据小生境数目这一参数连续地更新适应度共享函数,以便获得非支配边界上均匀分布的结果,但当共享适应度与锦标赛选择相结合时,算法在执行过程中可能会出现混沌行为;PESA[10] 采用了一种超网格技术对种群进行维护,它将目标空间分成若干个 hyper-box,每个个体就与某个网格相关联,网格里的个体数目称为挤压因子,在进入归档集的过程中,挤压因子大的个体将被清除掉;NSGA-II 利用个体的聚集距离来对种群进行维护,通过计算进化群体中每个个体的聚集距离,构造了一个偏序集,对聚集距离较小的个体进行淘汰,其时间复杂度为 $O(rN\mathrm{lb}N)$。但随着维数的增大,构造的这种聚集距离很难精确地反映个体的密度信息,并且由于构造方法的静态性,没有考虑已经移出个体对种群分布情况的影响。算法虽然有很好的时效性,但分布结果并不理想。SPEA2 用剪切方法对外部种群进行维护。选择第 k 个最近邻的个体进行删除,算法时间复杂度为 $O(rN^3)$。该方法具有动态性,每移出一个个体,外部种群中个体的分布情况都会发生变化,这样算法虽能很好地维持解集的分布性,但具有很高的时间复杂度。

3　基于邻域的多目标进化算法

首先我们定义一个反映个体邻近程度的指标——邻域包含关系。

定义 5　个体的邻域包含关系。设 p 和 q 是进化群体 Pop 中的任意两个不同的个体,对一距离 r,若 q 在以 p 为圆心,以 r 为半径的区域内,则称 p 邻域包含 q,记为 $p \rhd_r q$。

容易发现邻域包含关系具有以下几个性质:

性质 1　关系 $\rhd_r$ 具有反自反性。

性质 2　关系 $\rhd_r$ 具有对称性。

性质 3　关系 $\rhd_r$ 不具备传递性。

我们依据性质 1,2,3 就对个体进行分布适应度赋值,为了能精确地表示个体的分布情况,我们把个体的分布适应度分为两级(D_{rank1},D_{rank2})。具体方法如下:

算法 1　计算个体的分布适应度

```
for each i
{ i.D_rank1 = 0;
  i.D_rank2 = 0;
}                                          // 个体分布适应度初始化
for each i
{ temp = ind[i];    // ind[i] 为一存储个体邻域内信息的链表
  for each j
  { if (i !=j && distance(i,j) < r)
                    // 若 j 在 i 的邻域内,则调整 i 的分布适应度
    { insert(temp, j);
      i.D_rank1 = i.D_rank1 + 1;                          // 为整型
      i.D_rank2 = i.D_rank2 + distance(i,j);              // 为浮点型
    temp = temp -> child;
    }
  }
}
```

从算法 1 中容易看出,$i.D_{\mathrm{rank1}}$ 为邻域内个体数,值小表明 i 周围个体比较稀少;$i.D_{\mathrm{rank2}}$ 为邻域内个体与 i 的距离之和,对于 D_{rank1} 相同的个体,其 D_{rank2} 值大表明其距离周围个体相对较远。在图 1 中,对于个体 B 与 E,$B.D_{\mathrm{rank1}} = 3$,$E.D_{\mathrm{rank2}} = 2$,E 周围个体密度要低于 B 周围个体密度;对于个体 C 与 F,它们有相同的 D_{rank1} 值,$C.D_{\mathrm{rank1}} = F.D_{\mathrm{rank2}} = 3$,但 $C.D_{\mathrm{rank2}} < F.D_{\mathrm{rank2}}$,表明在其邻域内距离 F 的个体要远于距离 C 的个体。

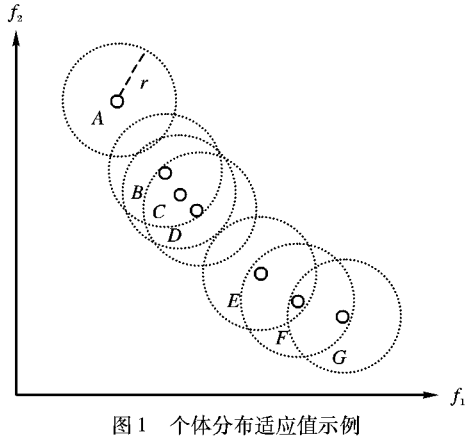

图 1　个体分布适应值示例

算法 1 对空间中的非支配个体进行了分布适应度赋值,下面我们根据个体的分布适应值对种群进行维护,其具体方

法如下：

```
算法 2  种群维护
while (| Q | > M)            // 当非支配个体数目大于归档集时
{ x = findmax( ), y = findsecmax( );
                             // 分别找出 D_rank1 最大,次大的个体
  if (x. D_rank1 > y. D_rank1) delete (x), z = x;
                                        // 淘汰 D_rank1 大的个体
  else if (x. D_rank1 = y. D_rank1 && x. D_rank2 < y. D_rank2)
    delete (x), z = x;    // 若 D_rank1 相同,淘汰 D_rank2 小的个体
  else delete (y), z = y;
  temp = ind [z];
  while(temp -> child ! = NULL)
                        // 调整 z 邻域内其他个体的分布度适应值
  { (temp -> index). D_rank1 = (temp -> index). D_rank1 - 1
    (temp -> index). D_rank2 = (temp -> index). D_rank2 -
      distance( z, temp -> index)
    temp = temp -> child;
  }
| Q | = | Q | - 1;
}
```

从算法 2 中可以看出，维护方法总是淘汰 D_{rank1} 最大的个体，若 D_{rank1} 相同，则淘汰 D_{rank2} 小的个体。对于图 1 中的情况，若归档集大小为 5，种群维护过程如图 2 所示，个体 C 与 F 有相同的 D_{rank1}，但 $C.D_{rank2}$ 小于 $F.D_{rank2}$，所以淘汰 C，并调整 B，D 的分布适应值，得到图 2(a) 的情况，B、D 的 D_{rank1} 降为 2，此时整个群体中 F 具有最大的 D_{rank1}，F 被淘汰，得到图 2(b) 的情况。

算法 2 采用动态的方法维护种群，虽然增加一定的时间开销，但种群分布性得到了较好的保持。算法 1 的时间复杂度为 $O(rN^2)$，算法 2 在最坏情况下时间复杂度为 $O(rN^2)$，所以种群维护的总时间复杂度为 $O(rN^2)$，逊于 NSGA-II，好于 SPEA2。下面我们给出整个算法的具体流程。

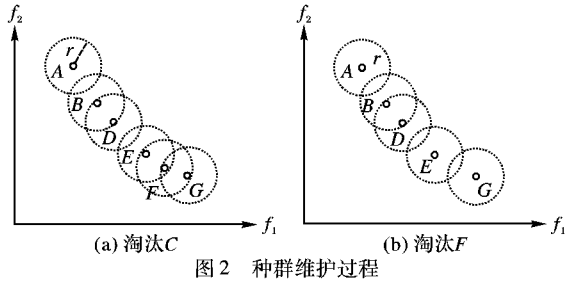

图 2　种群维护过程

NMOEA 借鉴 NSGA-II 采用了分层构造非支配集的方法[2]。

1) 算法参数设定。内部种群 R 规模为 N，外部种群 Q 的最大规模 M，中间种群 P 的规模 $M+N$，交叉概率 P_c，变异概率 P_m，算法搜索的最大代数 gen_{max}，令进化代数 $gen=1$。

2) 随机产生一个初始种群 R，同时初始化外部种群 Q，并使之为空。

3) 合并 R_{gen}、Q_{gen} 到 P_{gen}，将 P_{gen} 中的个体按支配的关系逐层加入到 Q_{gen+1} 中。当加入第 i 层非支配集 S 后的总个体数大于 M 时，按上面给出的方法对外部种群进行维护。若 $gen+1=gen_{max}$，将 Q_{gen+1} 中的个体作为返回结果，程序结束。

4) 对 Q_{gen+1} 执行锦标赛选择操作，对选中的个体执行交叉和变异操作。并将结果保存到 R_{gen+1} 中，$gen=gen+1$，转 3)。

4　实验与分析

为了检验所提出 NMOEA 的有效性，将其与另外 3 个著名的 MOEA(SPEA2[3]，NSGA-II[2]，EPS-MOEA[11]) 相比较。实验运行在 1.7 GHz CPU 256 MB 内存 Windows XP 环境下，选取了 7 个测试函数。SCH[12] 被认为是在 MOP 领域具有代表性的测试函数，FON[13] 是一个可变决策变量维数的测试函数，ZDT 是由文献[14]提出来的一系列测试函数，这些函数是具有代表性的测试函数，其中 ZDT1、ZDT2 较易收敛，ZDT3 为非连续的，ZDT4、ZDT6 较难收敛且 ZDT6 为非均匀分布。在实验中，四种算法都采用实数编码，交叉概率 0.8，变异概率 0.01，种群规模为 100，被评价个体的数目为 20000，算法的运行代数为评价个体的数目除以种群规模。每种算法对各个测试函数独立运行 10 次，结果取平均值。

多目标优化的三个基本目标为收敛性，分布性和运行时间。在多目标优化中解集的评价方法也是一个重要而复杂的问题[17]，研究者们提出了很多有效的方法[15-16,18-19]。在这里采用 Generational Disdance(GD)[15] 来估计算法的最终边界与全局非劣最优区域的趋近程度，计算如下：

$$GD=\sqrt{\sum_{i=1}^{n}d_i^2}\Big/n \tag{4}$$

n 是解集中个体的数目，d_i 是每个个体到全局非劣最优解的最小欧几里得距离。GD 的值越小就说明解集越靠近全局非劣最优区域，如果 $GD=0$ 就说明算法的解都在全局非劣最优区域上，这是最理想的情况。分布性评价采用 *Schott* 提出的方法[16]，其函数定义如下：

$$SP=\sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(\bar{d}-d_i)^2} \tag{5}$$

$d_i=\min_j(|f_1^i(\vec{x})-f_1^i(\vec{x})|+f_2^i(\vec{x})-f_2^i(\vec{x})|)(i,j=1,2,\cdots,n)$，$\bar{d}$ 是所有 d_i 的平均值。当算法获得的非劣解完全均匀地分布在目标空间时，$SP=0$。

表 1 显示了四种算法在 7 个测试函数上的收敛度，分布度，运行时间结果。在每个评价指标中，第一列为均值，第二列为标准差，加粗数据为最优值。从表 1 中可以发现，在收敛性方面，对较易收敛的测试函数 SCH，FON，ZDT1 和 ZDT2 四种算法性能相差不大，在 SCH，FON 上 NMOEA 要略优于其他三种算法，在 ZDT1，ZDT2 上 EPS-MOEA，SPEA2 分别拥有最好的收敛度。对非连续或较难收敛的测试函数 ZDT3，ZDT4 和 ZDT6，NMOEA 拥有较稳定的收敛度，特别对于其他两种算法很难收敛的测试函数 ZDT4，NMOEA 能收敛到真实的最优面。

从表 1 中可以发现，在分布性方面，NMOEA 和 SPEA2 比较接近，好于其他两种算法，NMOEA 在 SCH，ZDT3，ZDT4 上有最好的分布性，SPEA2 在 FON，ZDT1，ZDT2，ZDT6 上有最好的分布性，但 SPEA2 对非连续函数 ZDT3 很难收敛真实的最优面，这是由其基于密度的修剪方法决定的。图 3 ~ 图 5 为四种算法在部分测试函数上最终边界分布。

四种算法在运行时间方面，EPS-MOEA 具有最快的速度，NSGA-II、NMOEA 次之，SPEA2 最差。

通过分析和比较发现：在趋近最优前沿方面和运行效率方面，四种算法性能相差不大，NMOEA 的稳定性要略好于其他三种算法；在保持解集的分布性方面，NMOEA 与 SPEA2 相当，优于其他两种算法；在运行时间方面 EPS-MOEA 最快，NSGA-II 次之，HOMEA 要好于 SPEA2。

表 1　算法性能比较

测试函数	算法	收敛性 (*GD*)		分布性 (*SP*)		时间/s	
		均值	标准差	均值	标准差	均值	标准差
SCH	NSGA-II	0.00042635	6.449E-006	0.0352146	0.0047235	1.8814	0.0989
	SPEA2	0.00043522	2.772E-005	0.0153041	0.0017333	23.2523	0.1921
	EPS-MOEA	0.00041651	8.528E-006	0.3806914	0.0034641	**0.4060**	0.0024
	NMOEA	**0.00041042**	1.101E-005	**0.0148636**	0.0041327	5.8624	0.3228
FON	NSGA-II	0.00087117	0.00013394	0.0076202	0.0010584	1.9657	0.2241
	SPEA2	0.00085775	2.654E-005	**0.0028152**	0.0002455	22.1666	0.6524
	EPS-MOEA	0.00084571	0.00014574	0.0227531	0.0063714	**0.7344**	0.0865
	NMOEA	**0.00084273**	0.00010997	0.0049798	0.0014077	5.8534	0.1064
ZDT1	NSGA-II	0.00036451	0.00028417	0.0072641	0.0098571	2.1188	0.1209
	SPEA2	0.00034259	2.461E-005	**0.0031796**	0.0003471	24.4002	0.2013
	EPS-MOEA	**0.00029911**	2.844E-005	0.0059046	0.0002801	**1.1565**	0.1288
	NMOEA	0.00036046	0.00011021	0.0034302	0.0001034	5.5656	0.1404
ZDT2	NSGA-II	0.00037354	0.00028647	0.0082369	0.0009915	2.1406	0.1041
	SPEA2	**9.142E-005**	3.326E-005	**0.0029657**	0.0002985	24.8564	0.8242
	EPS-MOEA	0.00025649	3.717E-005	0.0086154	0.0008812	**1.1257**	0.0874
	NMOEA	0.00032841	0.00010372	0.0048987	0.0002161	4.1468	0.6355
ZDT3	NSGA-II	**0.00057591**	8.031E-006	0.0087924	0.0011501	3.1002	0.1027
	SPEA2	0.00182973	8.411E-005	0.0146478	0.0090537	19.3032	1.2483
	EPS-MOEA	0.00245679	0.00011954	0.0116454	0.0005162	**1.0931**	0.0247
	NMOEA	0.00072975	2.248E-005	**0.0044042**	0.0010774	5.3966	0.0307
ZDT4	NSGA-II	0.03612438	0.00913627	0.0272427	0.0014793	1.8344	0.0599
	SPEA2	0.06463929	0.02982384	0.0041184	0.0004551	20.1066	0.9807
	EPS-MOEA	0.07168257	0.03654741	0.0078693	0.0005784	**0.7658**	0.0166
	NMOEA	**0.00043619**	0.00030905	**0.0040764**	0.0006364	4.9094	0.2357
ZDT6	NSGA-II	0.05244596	0.01802994	0.0079248	0.0032411	1.8312	0.0467
	SPEA2	**0.00055196**	1.352E-005	**0.0018444**	0.0001988	23.6721	0.1603
	EPS-MOEA	0.00087653	0.00046027	0.0036385	0.0004201	**0.7814**	0.0355
	NMOEA	0.00302169	0.00010797	0.0068088	0.0014987	3.1532	0.1817

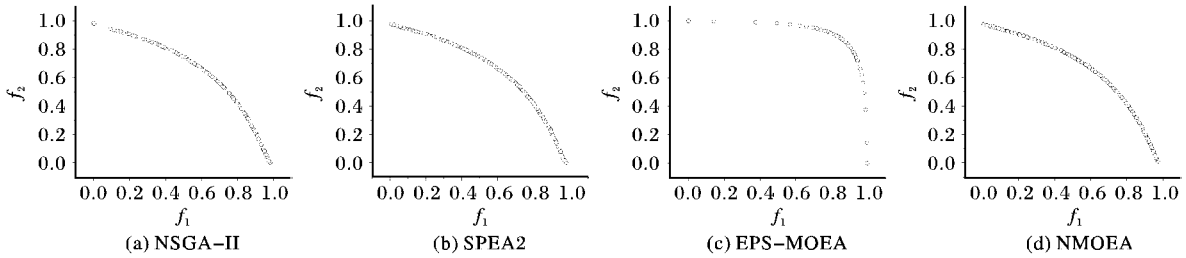

图 3　4 种算法在 FON 上的最终边界

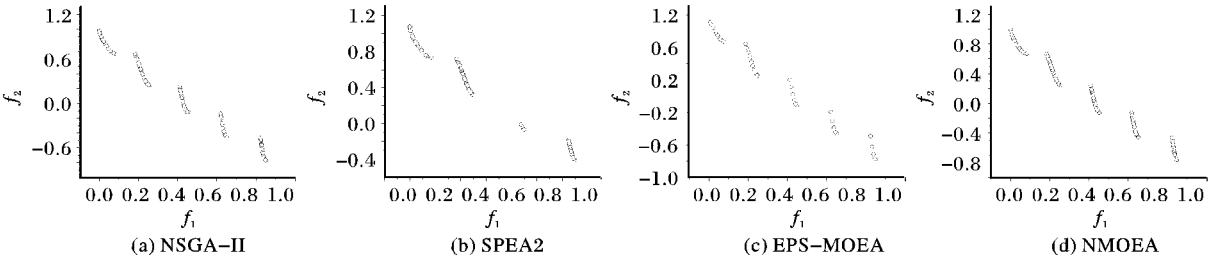

图 4　4 种算法在 ZDT3 上的最终边界

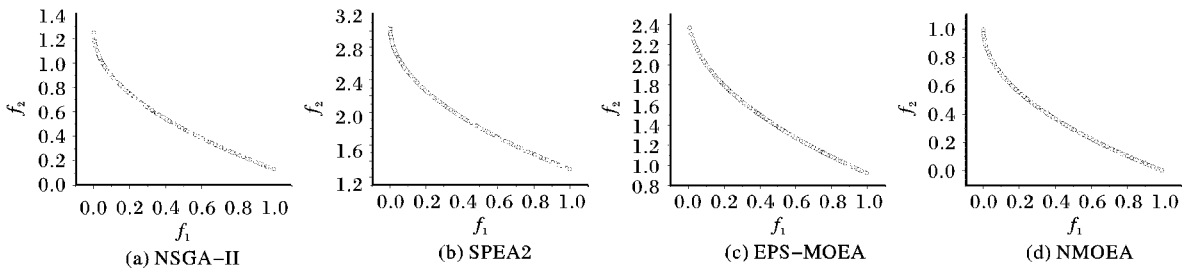

图 5　4 种算法在 ZDT4 上的最终边界

5 结语

种群维护是多目标进化算法的重要组成部分,维护方法的好坏决定着最终解集的分布状况。针对维护方法和运行效率的矛盾,本文设计了一种基于邻域的种群维护方法NMOEA。定义了一个反映个体邻近程度的指标——邻域包含关系,利用邻域包含关系对个体进行分布适应度分级的赋值,并且用动态方法快速地对种群进行维护。将其与NSGA-II,SPEA2 和 EPS-MOEA 进行比较实验,结果表明NMOEA 具有良好的搜索性能。

参考文献:

[1] 郑金华. 多目标进化算法及其应用[M]. 北京: 科学出版社, 2007.

[2] 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.

[3] ZITZLER E, LAUMANNS M, THIELE L. SPEA2: Improving the strength Pareto evolutionary algorithm, TIK-Report 103 [R]. 2001.

[4] 曾三友, 李晖, 丁立新, 等. 基于排序的非劣集合快速求解算法[J]. 计算机研究与发展, 2004, 41(9): 1565 - 1571.

[5] 郑金华, 史忠植, 谢勇. 基于聚类的快速多目标遗传算法[J]. 计算机研究与发展, 2004, 41(7): 1081 - 1087.

[6] 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[J]. 软件学报, 2007, 18(6): 1287 - 1297.

[7] 石川, 李清勇, 史忠植. 一种快速的基于占优树的多目标进化算法[J]. 软件学报, 2007, 18(3): 505 - 516.

[8] LAUMANNS M, THIELE L, DEB K, *et al.* Combining convergence and diversity in evolutionary multiobjective optimization [J]. Evolutionary computation, 2002, 10(3): 263 - 282.

[9] HORN J, NAFPLIOTIS N, GOLDBERG D E. A niched pareto genetic algorithm for multiobjective optimization [C]// Proceedings of the First IEEE Conference on Evolutionary Computation, IEEE World Congress on Computational Intelligence. Piscataway, New Jersey: IEEE Service Center, 1994, 1: 82 - 87.

[10] CORNE D W, KNOWLES J D, OATES M J. The pareto envelope-based selection algorithm for multiobjective optimization [C]// Proceedings of the Parallel Problem Solving from Nature VI Conference. Berlin: Springer-Verlag, 2000: 839 - 848.

[11] DEB K, MOHAN M, MISHRA S. A fast multi-objective evolutionary algorithm for finding well-spread pareto-optimal solutions, KanGAL Report No. 2003002 [R]. 2003.

[12] Van VELDHUIZEN D A, LAMONT G B. Multiobjective evolutionary algorithm test suites [C]// Proceedings of the 1999 ACM Symposium on Applied Computing. New York: ACM Press, 1999: 351 - 357.

[13] FONSECA C M, FLEMING P J. Multiobjective genetic algorithms made easy: Selection, sharing, and mating restriction [C]// Proceedings of the First International Conference on Genetic Algorithms in Engineering Systems: Innovations and Applications. Washington: IEEE Press, 1995: 42 - 52.

[14] DEB K. Multi - objective genetic algorithms: Problem difficulties and construction of test problems [J]. Evolutionary Computation, 1999, 7(3): 205 - 230.

[15] Van VELDHUIZEN D A, LAMONT G B. Multiobjective evolutionary algorithm test suites [C]// Proceedings of the 1999 ACM Symposium on Applied Computing. New York: ACM Press, 1999: 351 - 357.

[16] DEB K. Multi-objective optimization using evolutionary algorithms [M]. Chichester, England: John Wiley & Sons, 2001.

[17] ZITZLER E, LAUMANNS M, THIELE L, *et al.* Why quality assessment of multiobjective optimizers is difficult [C]// Proceedings of the Genetic and Evolutionary Computation Conference (GECCO'2002). San Francisco, California: Morgan Kaufmann Publishers, 2002: 666 - 673.

[18] FARHANG-MEHR A, AZARM S. An information-theoretic entropy metric for assessing multi-objective optimization solution set quality [J]. Journal of Mechanical Design, 2003, 125(4): 655 - 663.

[19] ZHENG JIN-HUA, LI MI-QING. A diversity metric for MOEAs [C]// 7th International Conference on Optimization: Techniques and Applications (ICOTA7). KobeJapan: Universal Academy Press, 2007: 451 - 452.

(上接第 1569 页)

表 1 LSI, PLSI, LPI 和 FLPI 的计算时间 s

训练集百分比/%	LSI	PLSI	LPI	FLPI
5	3.84	2.33	2.97	9.560
10	4.08	86.05	73.83	12.010
20	5.43	131.70	*	16.370
30	6.35	173.40	*	20.970
40	7.51	214.50	*	26.220
50	9.02	254.30	*	32.380
60 (原始分割)	9.75	294.10	*	39.520

注:*表示由于内存限制 LPI 不适用

4 结语

本文提出了一种基于 LPI 的优化的局部保留索引算法 FLPI,其显著减少了 LPI 的计算复杂性。理论分析显示,当 α 减小到 0 时,FLPI 能给出和 LPI 相同的解。对监督学习,当训练集小时,通常倾向于选择一个更平滑的基函数,带有 $\alpha(>0)$ 的 FLPI 能获得比 LPI 更优的性能。然而,如何自动评价一个最优的 α 将是今后进一步研究的关键问题。

参考文献:

[1] XU W, LIU X, GONG Y. Document clustering based on nonnegative matrix factorization [C]// Proceedings of 2003 International Conference on Research and Development in Information Retrieval (SIGIR'03). 2003: 267 - 273.

[2] HOFMANN T. Probabilistic latent semantic indexing [C]// Proceedings of 1999 International Conference on Research and Development in Information Retrieval (SIGIR'99). 1999: 50 - 57.

[3] CHUNG F R K. Spectral Graph Theory, volume 92 of Regional Conference Series in Mathematics. AMS, 1997.

[4] GOLUB G H, LOAN C F V. Matrix computations [M]. 3rd ed. Johns Hopkins University Press, 1996.

[5] HASTIE T, TIBSHIRANI R, FRIEDMAN J. The elements of statistical learning: Data mining, inference, and prediction [M]. New York: Springer-Verlag, 2001.

[6] PENROSE R. A generalized inverse for matrices [C]// Proceedings of the Cambridge Philosophical Society. 1955, 51: 406 - 413.

[7] CHANG C-C, LIN C J. LIBSVM: A library for support vector machines [EB/OL]. http://www.csie.ntu.edu.tw/~cjlin/libsvm.