

快速提高 NSGA-II 算法双目标优化效率的方法

刘敏¹, 陈宝兴¹, 郑金华²

LIU Min¹, CHEN Bao-xing¹, ZHENG Jin-hua²

1.漳州师范学院 计算机科学与工程系, 福建 漳州 363000

2.湘潭大学 信息工程学院, 湖南 湘潭 411105

1.Department of Computer Science and Engineering, Zhangzhou Normal College, Zhangzhou, Fujian 363000, China

2.Institute of Information Engineering, Xiangtan University, Xiangtan, Hunan 411105, China

E-mail: liumin_xt@163.com

LIU Min, CHEN Bao-xing, ZHENG Jin-hua. Approach to improve bi-objective optimization efficiency of NSGA-II. Computer Engineering and Applications, 2009, 45(34): 47-49.

Abstract: NSGA-II is a multi-objective evolutionary algorithm, and its performance is so good that it has become very popular in the last few years. To improve its bi-objective optimization efficiency, in this paper, a layering strategy according to need is adopted and so a new algorithm to construct the set of non-dominated fronts is proposed to replace the original method of NSGA-II. Compared with the NSGA-II's computational complexity ($O(N^2)$), the new algorithm's computational complexity is reduced to $O(kN + N \log N)$, k is the number of fronts, and $k \ll N$. The experiment results also show that there are fewer layers of non-dominated fronts, counts of dominate compare and much less running-time in the new approach compared with NSGA-II.

Key words: multi-objective evolution; non-dominated front; layering strategy according to need

摘要: NSGA-II 是一种性能优良的多目标进化算法, 近年来非常流行。为了进一步改进 NSGA-II 在双目标优化时的效率, 采取了按需分层的策略, 提出了一种新的非支配前沿集分层方法以替代 NSGA-II 原有的分层方法。与 NSGA-II 的时间复杂度 $O(N^2)$ 相比, 新方法的时间复杂度减少为 $O(kN + N \log N)$, k 为所分前沿层数 ($k \ll N$)。实验结果也表明, 新方法比 NSGA-II 相比具有更少的非支配前沿层数, 支配比较次数和运行时间。

关键词: 多目标进化; 非支配前沿; 按需分层

DOI: 10.3778/j.issn.1002-8331.2009.34.015 文章编号: 1002-8331(2009)34-0047-03 文献标识码: A 中图分类号: TP18

1 引言

近年来, 多目标进化算法 (Multi-Objective Evolutionary Algorithm, MOEA) 在科学研究和实际应用领域引起了广大学者浓厚的研究兴趣。一方面在于现实世界中的许多实际问题一般都是多属性的, 通常需要对多个目标进行同时优化。另一方面, 这些问题通常又是高度复杂的, 非线性的。而 MOEA 恰好非常适合于求解该类问题, MOEA 同时能提供一组折衷解 (trade-off solutions) 供用户选择, 这组折衷解是针对不同的子目标的一个折衷, 因此 MOEA 通过一次运行便可以搜索到多个解, 同时具有比单目标进化算法更强的求解问题的能力。

目前, 在 MOEA 的众多研究方法中, 基于 Pareto 的方法最为活跃。其中比较典型的 MOEA 有 Knowles 和 Corne 提出的 PAES (the Pareto Archived Evolutionary Strategy)^[1], Zitzler 和 Thiler 提出的 SPEA (Strength Pareto Evolutionary Algorithm)^[2], Deb 提出的 NSGA-II (Non-dominated Sorting Genetic Algorithm

II)^[3]等。这类方法的共同特点是, 在进化过程中, 通过构造当前进化种群的非支配集 (non-dominated set, Nds), 并使之不断逼近 Pareto 最优解来实现。这些算法构造非支配集的时间复杂度均不低于 $O(N^2)$, 其中, N 为组合种群规模。目前 NSGA-II 是被应用得最为广泛的一种快速的多目标进化算法, 该算法在每一代种群进化过程中采用了一种快速非支配排序的方法, 将组合种群分成多层非支配前沿集 F (the set of non-dominated fronts), 其时间复杂度为 $O(N^2)$ ^[3]。

双目标优化是多目标优化当中维数最少的一种方式, 也是一种常见的优化形式。在给出双目标快速排序的定义后, 分析论证了双目标快速排序后的线性表满足相关性性质和定理。利用这些性质和定理, 对 NSGA-II 算法的双目标优化效率进行改进。采取了按需分层的策略, 提出了一种新的构造非支配前沿集的方法来替代 NSGA-II 原来的构造方法, 将其原来的时间复杂度由 $O(N^2)$ 降至 $O(kN + N \log N)$, 一般 $k \ll N$ 。极大地减少了

基金项目: 福建省科技三项经费资助项目 (the Three Items of Sci-tech Foundation of Fujian Province under Grant No.2006F5068); 漳州师范学院科研资助项目 (the Natural Science Foundation Program of Zhangzhou Normal College under Grant No.SK08006)。

作者简介: 刘敏 (1974-), 男, 助教, 主要研究领域为进化计算、多目标优化; 陈宝兴 (1961-), 男, 博士, 教授, 主要研究领域为互联网的拓扑结构、最优设计与算法; 郑金华 (1963-), 男, 教授, 博士生导师, 主要研究领域为进化计算、智能科学等。

收稿日期: 2008-07-08 **修回日期:** 2009-01-08

非支配比较次数、分层数目和分层时间,使得改进后的 NSGA-II 具有更高的双目标优化效率。

2 相关的定义、性质与定理

定义 1 设 P 为一个集合,其大小为 n , P 中每个个体均有 r 个属性, $f_k(\cdot)$ 是每个属性的评价函数 ($k=1, 2, \dots, r$), P 中个体之间的关系定义为:

(1) Pareto 支配关系: $\forall x, y \in P$, 若 $f_k(x) \leq f_k(y)$, ($k=1, 2, \dots, r$); 且 $\exists l \in \{1, 2, \dots, r\}$, 使 $f_l(x) < f_l(y)$, 则称 x 支配 y , 表示为 $x > y$ 。此时称 x 为非支配的 (non-dominated), y 为被支配的 (dominated), 其中“ $>$ ”是支配关系。

(2) 不相关: $\forall x, y \in P$, 若 x 和 y 之间不存在支配关系, 则称 x 和 y 不相关或无关。

定义 2 非支配集: 对于给定个体 $x \in P$, 若 $\exists y \in P$, 使 $y > x$, 则 x 称之为集合 P 的非支配个体。由所有 P 的非支配个体组成的集合, 称之为 P 的非支配集。

定义 3 非支配前沿: 对于组合种群 R_t (由父种群 P_t 和子种群 Q_t 合并而成), 称其非支配集为 R_t 的第一层非支配前沿 F_1 ; 称集合 $R_t - F_1$ 的非支配集为 R_t 的第二层非支配前沿 F_2 ; 称集合 $R_t - (F_1 \cup F_2)$ 的非支配集为 R_t 的第三层非支配前沿 F_3 。

定义 4 双目标快速排序: $\forall x, y \in R_t$, 对组合种群 R_t 进行快速排序, 这里的排序比较定义为 $x < y \rightarrow (f_1(x) < f_1(y)) \vee (f_1(x) = f_1(y) \wedge f_2(x) < f_2(y))$ 。即先按第一个目标值进行升序排序, 当第一个目标值相同时, 再按照第二个目标值进行升序排序。

将 R_t 快速排序后的结果保存在线性表 F_i 中 (初始时 $i=1$), 则 F_i 具有如下的性质和定理:

性质 1 经双目标快速排序后, 线性表 F_i 中所有的非支配个体在第一个目标上是按照升序排列的。由于非支配个体之间彼此互不支配, 那么这些非支配个体在第二个目标上是降序排列的。

定理 1 在线性表 F_i 中, 任意一个元素 (下文有时为了表述方便将元素称为个体) 一定不被它后面的元素支配。

证明 $\forall a, b \in F_i$, 且 b 是 a 后面的某一元素。由于 F_i 已经过快速排序, 所以有 $f_1(a) < f_1(b) \vee (f_1(a) = f_1(b) \wedge f_2(a) < f_2(b))$ 。由支配关系的定义可知, a 不被 b 支配。

推论 1 线性表 F_i 中的第一个元素一定是非支配个体。

定理 2 假设 F_i 中某元素的前驱是非支配个体, 如果前驱元素不支配后继元素, 则该后继元素也是非支配个体。

证明 $\forall a, b \in F_i$, 假设 a 是 b 的前驱, 根据双目标快速排序有 $f_1(a) < f_1(b) \vee (f_1(a) = f_1(b) \wedge f_2(a) < f_2(b))$ 。如果 $f_1(a) = f_1(b) \wedge f_2(a) < f_2(b)$ 成立, 则有 a 支配 b , 与已知条件矛盾, 故 $f_1(a) = f_1(b) \wedge f_2(a) < f_2(b)$ 不成立, 因而必定有 $f_1(a) < f_1(b)$ 。再假设 $f_2(a) \leq f_2(b)$, 由于已有 $f_1(a) < f_1(b)$, 则有 a 支配 b , 又与已知条件矛盾, 所有 $f_2(b) < f_2(a)$ 。又因为 a 是非支配个体, 根据性质 1 可知, a 的第二目标值 $f_2(a)$ 比它前面所有的非支配个体的第二目标值都要小。所以个体 b 和它前面所有的个体相比, $f_2(b)$ 最小, 故 b 不被它前面的非支配个体支配。再由定理 1 可知, b 也不会被它后面的个体支配, 所以后继 b 是非支配个体。

需要说明的是, 在使用提出的构造方法后, F_i 中的所有被支配个体将依次被删除掉, 再依次插入至线性表 F_{i+1} 的末尾。最后 F_i 中保留的全部是非支配个体, 此时的 F_i 才是最终的第 i 层非支配前沿。此外, 由于是顺序地插入, F_{i+1} 同样满足以上的性质和定理。

3 按需构造非支配前沿集的新算法

3.1 按需分层策略

NSGA-II 算法需要将组合种群 R_t 分成多层的非支配前沿的集合 ($F = F_1 \cup F_2 \cup \dots \cup F_k$, k 为非支配前沿的层数)。首先, 利用构造非支配集的方法求出 R_t 的非支配集, 得到第一层非支配前沿 F_1 。然后, 从种群 R_t 中删除掉 F_1 中的个体, 重复以上过程可分别得到前沿 F_2, F_3, F_4 等等, 直至 R_t 为空为止。分层的过 程如图 1 所示。当 NSGA-II 算法分层完毕后, 需要一个称为截断的过程, 将两倍种群规模的非支配前沿集 F 减半, 即删除掉 F 下半部区域内的个体, 取其上半部作为下一代的进化种群 P_{t+1} 。

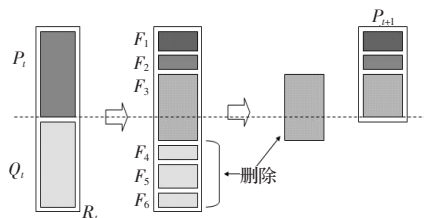


图 1 NSGA-II 的分层方法

然而以上的分层方法还有不足之处, 如图 1 所示, F_3 分层完毕后, 前沿 F_1, F_2, F_3 中的累计个体数目已经超出了单个种群规模, 再对 F_4, F_5, F_6 进行分层已是多余。因此提出了按需分层的策略。在分层的过程中, 不是以集合 R_t 为空作为停止分层的标准, 而是当已分层的非支配前沿中的累计个体数超过种群规模时便停止非支配前沿分层。如图 2 所示, 前三层非支配前沿构造完后, 便停止分层。由此可见, 按需分层的方法比 NSGA-II 的方法不仅能有效的减少非支配集的分层数目, 而且依然能得到与 NSGA-II 相同的下一代的进化种群 P_{t+1} 。

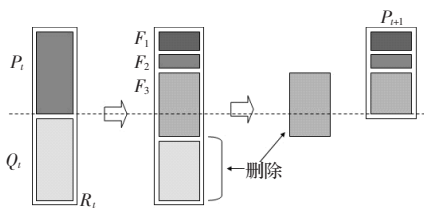


图 2 按需分层的方法

3.2 新算法的基本思想

步骤 1 双目标排序预处理: 令 $i=1$; 对组合种群 R_t 先进行双目标快速排序, 排序后的结果保存在线性表 F_i 当中;

步骤 2 前向非支配比较遍历线性表 F_i : 从第二个元素一直到最后元素顺序地遍历线性表 F_i , 进行前向非支配比较, 即将当前个体与其前驱进行非支配比较。若当前个体不被前驱支配, 则该个体是非支配的, 将继续保留在 F_i 中。否则, 该个体是被支配的, 将转移到 F_{i+1} 末尾保存;

步骤 3 按需分层: 设 k 为已分层的非支配前沿集的层数, $POPSIZE$ 为单种群大小, $|F_i|$ 表示非支配前沿 F_i 中的元素个数,

如果 $\sum_{i=1}^k |F_i| < POPSIZE$, 则 $i=i+1$ 并跳转至步骤 2 继续执行, 否则

算法分层完毕。

3.3 新算法的详细描述

FUNCTION Generate_Non-dominated_Fronts(R_t)

{ 对组合种群 R_t 进行非支配排序, 分成 k 层非支配前沿集 $\{F_i | 1 \leq i \leq k\}$, 初始时 $F_i = \Phi$ }

$i=1$; count=0; {count 为已分层的非支配前沿中元素的个数累计值}

```

 $F_i = \text{quicksort}(R_i)$ ; {对  $R_i$  进行双目标快速排序预处理}
WHILE (count < POPSIZE) DO {按需分层}
  [  $q = F_i.\text{begin}()$ ; {令  $q$  为  $F_i$  中的第一个元素}
  count = count + 1; { $q$  作为第一个元素一定是非支配, 非支配前沿中元素个数加 1}
  FOR  $p = q \rightarrow \text{next}$  TO  $F_i.\text{end}()$  DO {前向非支配比较遍历线性表  $F_i$ }
    [ IF ( $q > p$ )
      THEN { $q$  支配  $p$ }
        [  $F_{i+1}.\text{push\_back}(p)$  {将被支配元素  $p$  插入到下一层非支配前沿末尾}
           $r = p$ ;  $p = p \rightarrow \text{next}$ ;  $q \rightarrow \text{next} = p$ ;  $\text{dispose}(r)$ ; {删除被支配个体}
        ]
      ELSE [  $q = p$ ;  $p = p \rightarrow \text{next}$ ; count = count + 1; ] { $p$  不受  $q$  支配,  $p$  属于当前非支配前沿, 非支配前沿中元素个数加 1}
    ] [end for 结束  $F_i$  层非支配前沿的分层]
     $i = i + 1$ ; {进入下一层非支配前沿的非支配比较}
  ] [end while 非支配前沿分层完毕]
   $k = i - 1$ ; { $k$  为最后一层前沿数}
  RETURN  $F_1, F_2, \dots, F_k$ ;
ENDF; {Generate_Non-dominated_Fronts}

```

3.4 算法的时间复杂性分析

设组合种群 R_t 的规模大小为 N (这里 $N = 2 \times \text{POPSIZE}$, POPSIZE 为单种群规模大小), 非支配排序分层的最后一前沿层数为 k 。下面对该算法时间复杂度进行分析:

首先, 快速排序的平均时间复杂度为 $O(N \log N)$ 。

其次, 计算 WHILE 循环体的时间复杂性: 生成第一层非支配前沿最多要进行 $N-1$ 次非支配比较; 生成第二层非支配集最多要进行 $N-2$ 次非支配比较; ……; 生成第 k 层非支配集最多要进行 $N-k$ 次非支配比较, 所以该 WHILE 循环体最坏的时间复杂度为:

$$T(N) = \sum_{i=1}^k \text{第 } i \text{ 层的最多比较次数} = \sum_{i=1}^k N - i = N - 1 + N - 2 + \dots + N - k < kN$$

综合这两部分时间复杂度, 该算法的时间复杂度为 $O(kN + N \log N)$ 。这里的 k 满足 $1 \leq k \leq N$ 。最坏情况下, 当每层非支配前沿恰好仅有一个元素时, 才有 $k = N$, 而这种情形出现的概率非常小。最好情况下, 当组合种群 R_t 中有一半以上的个体为非支配个体时, 这时 $|F_1| \geq \text{POPSIZE} / 2$ ($\text{POPSIZE} = N/2$), 算法仅分一层非支配前沿, 故有 $k = 1$ 。由于进化算法是一种随机概率算法, k 的取值将随算法运行代数的不同而取不同的值, 因而通过实验方法来计算 k 的平均值, 一般情况下有 $k \ll N$ 。当 $\text{POPSIZE} \geq 100$, 最大进化代数 $\text{Maxgen} \geq 200$ 时, 一般还有 $k < \log N$, 则此时算法的时间复杂度是 $O(N \log N)$, 好于 NSGA-II 算法的时间复杂度 $O(N^2)$ 。相关数据可参见下文的实例和实验部分来验证。

3.5 构造非支配前沿集实例

下面给出一个具体实例^[4], 说明该算法是如何构造非支配前沿集的。

例 1 考虑一个具有两个目标, 20 个个体的进化群体 (这里 $N = 20$, $\text{POPSIZE} = 10$):

$C_1 = (9, 1)$, $C_2 = (7, 2)$, $C_3 = (5, 4)$, $C_4 = (4, 5)$, $C_5 = (3, 6)$, $C_6 = (2, 7)$, $C_7 = (1, 9)$, $C_8 = (10, 1)$, $C_9 = (8, 5)$, $C_{10} = (7, 6)$, $C_{11} = (5, 7)$, $C_{12} = (4, 8)$, $C_{13} = (3, 9)$, $C_{14} = (10, 5)$, $C_{15} = (9, 6)$, $C_{16} = (8, 7)$, $C_{17} = (7, 9)$, $C_{18} = (10, 6)$, $C_{19} = (9, 7)$, $C_{20} = (8, 9)$

这里, 个体 $C_i = (f_1, f_2)$ 的两个目标值分别为 f_1 和 f_2 , $i = 1, 2, \dots, 20$ 。

给定初始集 $R_t = \{C_{15}, C_{16}, C_{11}, C_6, C_8, C_{13}, C_1, C_9, C_{17}, C_{10}, C_7, C_3, C_{12}, C_2, C_{14}, C_{18}, C_4, C_{20}, C_5, C_{19}\}$ 。

构造其多层非支配前沿集的过程描述如下:

快速排序后, $F_1 = \{C_7, C_6, C_5, C_{13}, C_4, C_{12}, C_3, C_{11}, C_2, C_{10}, C_{17}, C_9, C_{16}, C_{20}, C_1, C_{15}, C_{19}, C_8, C_{14}, C_{18}\}$ 。

(1) 执行完第一次的内层 FOR 循环后, 已分层的 $F_1 = \{C_7, C_6, C_5, C_4, C_3, C_2, C_1\}$, 待分层的 $F_2 = \{C_{13}, C_{12}, C_{11}, C_{10}, C_{17}, C_9, C_{16}, C_{20}, C_{15}, C_{19}, C_8, C_{14}, C_{18}\}$, 进行了 19 次前向非支配比较 (没有计算快速排序的比较)。count = 7 < POPSIZE。

(2) 执行完第二次的内层 FOR 循环后, 已分层的 $F_2 = \{C_{13}, C_{12}, C_{11}, C_{10}, C_9, C_8\}$, 待分层的 $F_3 = \{C_{17}, C_{16}, C_{20}, C_{15}, C_{19}, C_{14}, C_{18}\}$, 进行了 12 次比较, 此时 count = 13 > POPSIZE, 停止分层。返回非支配前沿 F_1, F_2 。

以上两次 FOR 循环共计比较了 31 次, 小于 $kN (2 \times 20 = 40)$, 更小于 $N \ln N (> 80)$ 。验证了时间复杂度分析中提到的 $k < \log N$ 。

4 实验结果

主要针对双目标函数优化进行实验, 对 NSGA-II 和新方法分别计算它们分层过程中的支配比较次数, 非支配前沿集层数 (k) 和分层耗费的 CPU 时间。实验时采用了两套标准测试函数。其中的 ZDT 系列标准测试函数^[5]是 Zitzler 等人针对双目标优化问题提出的, DTLZ 系列标准测试函数^[6]是 Deb 等人针对多目标优化问题提出的, 该文实验的 DTLZ 系列测试函数的目标数也固定为两个目标。实验时使用 P4-1.7 G CPU, 512 M 内存的 PC 机, 操作系统为 WinXP, 编程环境为 VC++ 6.0。实验参数设置: 种群规模 $\text{POPSIZE} = 100$, 最大进化代数 $\text{Maxgen} = 300$ 。采用二进制的染色体编码, 单点交叉, 交叉率 $P_c = 0.9$, 点位变异, 变异率 $P_m = 1/l$, l 为染色体串的长度。实验时, 对于每一个测试函数, 都重复 20 次实验, 并取 20 次实验数据的平均值。NSGA-II 和新方法在每次构造非支配前沿集时的实验对比结果如表 1 所示。

表 1 NSGA-II 和新方法的实验对比数据

测试函数	NSGA-II			新方法		
	比较次数	分层数目	分层时间/ms	比较次数	分层数目	分层时间/ms
ZDT1	12 596.42	9.18	24.07	2 278.17	2.78	3.43
ZDT2	11 692.21	14.30	24.70	2 582.98	4.85	4.62
ZDT3	12 701.83	9.21	24.00	2 270.91	2.73	3.41
DTLZ1	11 956.45	12.68	22.90	2 137.35	1.93	2.49
DTLZ2	14 012.42	7.52	23.91	2 141.04	1.49	2.46
DTLZ3	11 128.52	15.82	23.24	2 328.16	3.36	3.62

从表 1 的数据来看, 新方法与 NSGA-II 相比, 无论是非支配比较次数, 分层数目, 还是分层时间均有很大的减少。究其原因, 主要是因为采用新算法替代了 NSGA-II 的构造非支配前沿集的方法, 并且采用了按需分层的策略。新算法的时间复杂度为 $O(kN + N \log N)$, $N = 2 \times \text{POPSIZE}$, 分层数 k 满足 $1 \leq k \leq N$, k 的取值将随算法运行代数的不同而取不同的值, 如图 3、4 所示。在这两张图中, 新方法的分层数明显少于 NSGA-II, 并且随着进化代数的增加, 分层数基本上呈现递减的趋势。当进化代数 ≥ 200 时, 新方法的分层数目 k 降至 1。而表 1 中的分层数是 300 代的平均值, 新方法的 k 值最大不超过 5, 且有 $k < \log N$ 。

(下转 126 页)