

改进多目标进化算法 NSGA-II 运行效率的方法

刘敏

(漳州师范学院计算机科学与工程系 福建漳州 363000)

【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 的运行效率,采用了擂台赛法则作为构造非支配集的方法,提出了按需分层的策略。实验结果表明,提出的方法与 NSGA-II 相比具有更少的非支配集层数,支配比较次数和运行时间。

【关键词】 多目标进化;非支配集;擂台赛法则;按需分层

0. 引言

近年来,多目标进化算法(MOEA: Multi-Objective Evolutionary Algorithm)引起了许多研究学者的兴趣,主要是因为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。由于 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,同时具有比标准的遗传算法(CGA: Canonical Genetic Algorithm)更强的求解问题的能力。此外,MOEA 的搜索是一个不断逼近 pareto 最优前沿(Pareto-Optimal Front)的并行过程,因此,它具有比 CGA 更好的并行特性。目前,基于 pareto 的比较典型的 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,这些算法构造非支配集的时间复杂度均不低于 $O(rN^2)$,其中, r 为目标数, N 为种群规模。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,其时间复杂度为 $O(rN^2)$ ^[3]。

由于基于 pareto 的多目标进化算法每一代都需要构造进化种群的非支配集,因此,构造非支配集的效率直接影响 MOEA 的运行效率。本文针对 NSGA-II 算法,提出了非支配集按需分层的策略,并采用擂台赛法则作为构造非支配集的方法,以求进一步提高 NSGA-II 的运行效率。

1. 用擂台赛法则构造非支配集

为了描述方便起见,下面先给出... 关定义。

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 $f_k()$ 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 $x \succ y$ 。此时称 x 为非支配的 (non-dominated), y 为被支配的 (dominated), 其中 " \succ " 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 对于给定个体 $x \in P$, 若 $\forall y \in P$, 使 $y \succ x$, 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。

在 Deb 提出的 NSGA-II 算法中,构造非支配集是一项非常重要的工作,需要选出构造集中的所有非支配个体构成非支配集,其时间复杂度为 $O(rN^2)$ 。

为了进一步提高运行效率,本文采纳国内学者郑金华提出的擂台赛法则^[4]来构造非支配集。该算法的基本思想是:设 P 为进化群体, Q 为构造集,初始时 $Q=P$; Nds 为非支配集,初始时为

空。从 Q 中任取一个个体 x , 依次与 Q 中所有其他个体 y 比较, 如果 x 支配 y , 则将个体 y 从 Q 中清除; 如果 y 支配 x , 则用 y 代替 x (即产生了新的擂主), 并继续进行比赛。一轮比较后,形成族 $Cluster(x)=\{y|x \succ y, \text{且 } x, y \in P\}$, x 为族长, 将 x 并入非支配集 Nds 中, 而 $Cluster(x)$ 中个体是被 x 所支配的, 必须从构造集 Q 中清除。依此类推, 直至 Q 为空。

由于 Q 中的个体与擂主进行比较时, 如果被擂主支配的话就会从 Q 中清除出去, 从而不参与下一轮比较。当被清除的个体越多, 接下来算法执行的次数就会越少。文献^[4]中分析其时间复杂度为 $O(rmN)$, 其中, r 为优化目标的数目, m 为非支配个体的数目。在一般情形下有 $m < N$, 因此擂台赛法则具有比 NSGA-II 的 $O(rN^2)$ 更好的运行效率。

2. 依据按需分层策略将组合种群分成多层非支配集

在 NSGA-II 算法中需要将组合种群 R_t (由父种群 P_t 和子种群 Q_t 合并而成) 分成多层的非支配集的集合 $F(F = F_1 \cup F_2 \cup \dots \cup F_e, e$ 为非支配集的层数)。首先, 利用构造非支配集的方法求出 R_t 的非支配集, 并将该集称为第一层非支配集 F_1 。然后, 从种群 R_t 中删除掉 F_1 中的个体, 即令 $R_t = R_t \setminus F_1$ 。重复以上过程可分别得到 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集的集合 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} 。

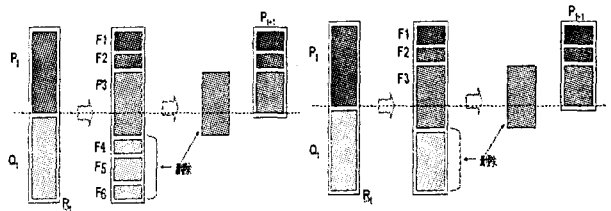


图 1 NSGA-II 的分层方法

图 2 改进的分层方法

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, F_1, F_2, F_3 中的累计个体数目已经超出了一个种群规模, 再对 F_4, F_5, F_6 进行分层已是多余的。因此本文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集中的累计个体数大于或等于种群规模时便停止非支配分层。如图 2 所示, 前三层非支配集构造完后, $F_1 \cup F_2 \cup F_3$ 中个体数目已超过了一个种群规模, 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} 。

3. 实验结果

本文主要针对双目标和三目标函数优化进行实验, 对 NSGA-II 和改进方法分别计算它们的支配比较次数, 非支配集层

数和分层耗费的 CPU 时间。双目标采用 Zitzler 等人提出的 ZDT 系列标准测试函数^[9],三目标采用 Deb、Thiele 和 Zitzler 提出的 DTLZ 系列标准测试函数^[10]。实验时使用 P4-1.7G CPU,512M 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 $N=100$,最大进化代数 $Maxgen=300$ 。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,都重复五次实验,并取五次实验数据的平均值。NSGA-II 和改进方法的实验对比结果如表一所示。表中偏差这一列是指改进的方法比 NSGA-II 所节约的分层时间比值,即(改进方法的分层时间-NSGA-II 的分层时间)/NSGA-II 的分层时间。

测试函数	NSGA-II			改进的方法			偏差
	比较次数	分层数目	分层时间(s)	比较次数	分层数目	分层时间(s)	
ZDT1	12958.08	9.38	4.72	8502.37	2.63	3.20	-32.2%
ZDT2	12102.52	14.10	5.02	8311.58	5.16	3.46	-31.1%
ZDT3	12936.54	9.59	4.77	8779.10	2.66	3.20	-32.9%
DTLZ1	13303.46	8.48	4.42	9352.93	1.20	2.80	-36.6%
DTLZ2	14990.21	5.89	4.83	10950.91	1.11	3.17	-34.4%
DTLZ3	12467.58	9.70	4.36	9557.09	1.69	3.00	-31.2%

表 1 NSGA-II 和改进方法的对比数据

从表 1 的数据来看,改进的方法与 NSGA-II 相比,无论是比较次数,分层数目,还是分层时间均较大的减少。究其原因,比较次数的减少主要是因为采用擂台赛法则以替代了 NSGA-II 的构造非支配集的方法,而擂台赛法则的时间复杂度为 $O(rmN)$,好于 NSGA-II 的 $O(rN^2)$ 。此外,本文采用按需分层的策略,因而大大的减少了分层数目。综合以上两点,改进的方法使得分层所耗费的 CPU 时间有了较大的减少,如从偏差列可看出,平均节约时间在 30%以上。

改进的方法提高了算法的运行效率,那获得的解结果会不会变差呢?答案是否定的,这两种方法获得的解结果是相同的。这里的相同并非绝对,应为大家知道进化算法是一种近似算法,因而每次算法的运行结果可能会有所不同,但是会得到近似的结果。如图 3 所示,双目标优化测试问题 ZDT1 的最优解集是在第一象限中介与 $[0,1]$ 间的一条凸曲线。NSGA-II 和改进方法的结果都收敛于该曲线。而图 4 显示的是三目标测试问题 DTLZ2 的解结果,该问题的最优解集是在第一象限中介与 $[0,1]$ 间的一个曲面。NSGA-II 和改进方法的结果都近似收敛于该曲

面,它们的结果是近似相同的。

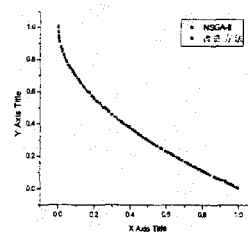


图 3 ZDT1 的解结果

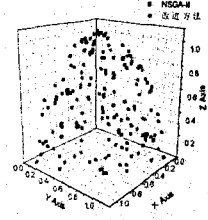


图 4 DTLZ2 的解结果

4. 结束语

多目标进化算法的一个重要研究方面是如何有效地对进化群体进行非支配分层,本文在 Deb 提出的 NSGA-II 算法研究基础上,针对算法的运行效率,提出了进化种群按需分层的策略,并采用擂台赛法则作为构造非支配集的方法。最后以六个函数优化问题为例,验证了改进的方法比 NSGA-II 具有更好的运行效率。

参考文献:

1. J. D. Knowles, D. W. Corne. Approximating the Non-dominated Front Using the Pareto Archived Evolution Strategy [J]. IEEE Transactions on Evolutionary Computation, 2000,8(2):149-172.
2. E. Zitzler, L. Thiele. Multiobjective Evolutionary Algorithms: A Comparative Case Study and the Strength Pareto Approach [J]. IEEE Transactions on Evolutionary Computation, 1999,3(4):257-271.
3. K. Deb, A. Pratap, S. Agrawal, T. Meyrivan. A Fast and Elitist Multi-objective Genetic Algorithm: NSGA-II [J]. IEEE Transactions on Evolutionary Computation, 2002,6(2):182-197.
4. 郑金华, 蒋浩, 邝达, 史忠植. 用擂台赛法则构造多目标 Pareto 最优解集的方法[J]. 软件学报,2007,18(6):1287-1297.
5. E. Zitzler, K. Deb, and L. Thiele. Comparison of multiobjective evolutionary algorithm: Empirical results [J]. Evolutionary Computation, 2000,8(2):173-195.
6. K. Deb, L. Thiele, M. Laumanns, E. Zitzler. Scalable test problems for evolutionary multi-objective optimization [C]. In Yao X, ed. Proc. of the 2002 Congress on Evolutionary Computation (CEC 2002), New Jersey: IEEE Service Center,2002: 825-830.

(上接第 130 页)

运行时间有变化,那么任务 B 将出现"抖动"。当两个任务重叠时,则不能在正确时间调用。如果以如下方式调度这些任务:

Add_Task (TaskA,0,1000);

Add_Task (TaskB,5,3000);

则任务 A 的调度总是在任务 B 之前 5ms 开始,任务 B 将总能准时运行。在应用中使用这种方法,将能有效避免任务重叠。

4.2 确定所需的时标间隔

使用合作式调度器的关键要求是:对于所有任务,无论在何种情况下,任务的运行时间 D_{task} 都必须满足以下条件:

$D_{task} < \text{时标间隔}$

所以,时标间隔要设置合适。所有任务运行的时间都应小于调度时标间隔。可以为任务设置"超时",以使它们不会阻塞调度器。同时为了精确调度,调度器的时标间隔应该设置为所有任务

间隔的"最大公因数"。

5. 结束语

基于时间触发模式的合作式调度器提供了一种简单而可预测性非常高的平台,它可以满足许多嵌入式系统的应用。本文介绍了它的具体实现方法,对于在嵌入式系统中建立基于时间触发模式的合作式调度器具有指导作用。

参考文献:

1. [英]Michael J. Pont 著,周敏 译. 时间触发嵌入式系统设计模式 [M]. 中国电力出版社,2001
2. [美]Jean. J. Labrosse. 邵贝贝译. 嵌入式实时操作系统 $\mu C/OS-II$ (第二版)[M]. 中国电力出版社,2003