

基于败者组与混合编码策略的 NSGA-II 改进算法

刘鑫平 顾春华 罗 飞 丁炜超

(华东理工大学信息科学与工程学院 上海 200237)

摘要 在精英选择中 NSGA-II 的拥挤系数算子对局部拥挤区域的分布性优化效果不佳,并且会使某些更接近 Pareto 最优解集的个体被淘汰。针对拥挤系数算子存在优秀个体不被保留的缺陷,提出了一种基于败者组与混合编码策略的改进算法(LGHC-NSGA-II)。参照棋类比赛中的双败淘汰制,构建了败者组外部归档集,在迭代结束后将归档集与末代父代种群合并,并采用循环拥挤系数排序策略优化分布性。同时,针对传统编码方式在全局或局部空间上搜索能力较差的缺陷,提出了一种混合编码策略,有效地提高了算法的收敛性。基于 ZDT 系列问题上的测试结果表明,改进算法与 8 种多目标进化算法相比,在算法的收敛性、分布性与鲁棒性上均具有较高的优越性。

关键词 多目标进化算法,NSGA-II,败者组,循环拥挤系数排序,混合编码

中图分类号 TP301 **文献标识码** A **DOI** 10.11896/jsjcx.181001852

Improved NSGA-II Algorithm Based on Loser Group and Hybrid Coding Strategy

LIU Xin-ping GU Chun-hua LUO Fei DING Wei-chao

(School of Information Science and Engineering, East China University of Science and Technology, Shanghai 200237, China)

Abstract The congestion coefficient operator of NSGA-II in elite selection can not optimize the distribution of local congestion area effectively, and some individuals closer to Pareto optimal solution set will be eliminated. An improved algorithm based on loser group and hybrid encoding strategy (LGHC-NSGA-II) was proposed to overcome the shortcoming that excellent individuals are not retained in the congestion coefficient operator. Referring to the double-losing elimination system in chess games, an external archive set of loser group is constructed. After the iteration, the archive set is merged with the last generation parent population, and the distribution coefficient is optimized by cyclic congestion coefficient ranking strategy. At the same time, a hybrid coding strategy was proposed to overcome the shortcomings of traditional coding methods in global or local space, which effectively improves the convergence of the algorithm. The test results on ZDT series problems show that the improved algorithm is superior to eight multi-objective evolutionary algorithms in convergence, distribution and robustness.

Keywords Multi-objective evolutionary algorithm, NSGA-II, Loser group, Cyclic congestion coefficient ranking, Hybrid coding

1 引言

多目标优化问题(MOP)是优化领域的主要研究方向之一,在实际工程中这些待优化目标之间往往相互冲突。不同于单目标优化问题, MOP 的最优解为一组折中解,即 Pareto 最优解集。学术界参考了许多模拟生物自然选择与自然进化的随机搜索算法来求解 MOP,并根据这些启发式算法或元启发式算法(例遗传算法(GA)、蚁群算法、粒子群算法、免疫算法等)提出了一系列多目标进化算法(MOEA),其中具有代表性的 MOEA 包括但不限于 Zitzler 等提出的 SPEA^[1]、Corne 等提出的 PESA^[2]、Srinivas 等提出的 NSGA^[3]、Coello 等提出的 MOPSO^[4]、Deb 等提出的 NSGA-II^[5]、Zhang 等提出的

MOEA/D^[6]。而 NSGA-II 作为其中的标杆,已经成为其他 MOEA 性能比较的基准^[7],NSGA-II 在 NSGA 的基础上采用了快速非支配排序和精英选择策略,通过拥挤系数算子筛选适应度更高的个体。这些策略同样被应用于其他 MOEA,例如 MOPSO 在 NSGA-II 的基础上采用了分解与差分思想的 MOEA/D 以及基于参考点的 NSGA-III^[8]。

为了提高 NSGA-II 的收敛性与分布性,相关学者提出了许多改进算法。文献[9]采用最小生成树来表示个体分布情况,使算法的分布性得到提高;文献[10]采用循环拥挤系数排序的方法提高算法的分布性,但该方法的时间代价较高,增加了算法的总体时间复杂度;文献[11]提出了分布式处理以及非支配解精度缺失的补偿方式,缩短了获得 Pareto 最优解集

收稿日期:2018-10-07 返修日期:2019-03-07 本文受国家自然科学基金项目(61472139)资助。

刘鑫平(1993—),男,硕士生,主要研究方向为多目标优化、云计算;顾春华(1970—),男,博士,教授,主要研究方向为云计算、物联网,E-mail:lxp_ecust@163.com(通信作者);罗 飞(1978—),男,博士,副教授,主要研究方向为云计算、分布式计算;丁炜超(1989—),男,博士生,主要研究方向为云计算、数据中心资源管理与优化、大数据应用。

的执行时间;文献[12]将生物遗传学中的跳跃基因融入 NSGA-II,提高了种群规模较少情况下的算法性能;文献[13]通过差分进化中变异算子的定向引导作用改善了种群的分布性,但该文所采用的 SP^[14]评价指标未计算解集与真实帕累托前沿(PF)的边界值偏差;文献[15]结合 SPEA 中凝聚层次的聚类方法与极值解的保存方法,使算法的解集能保持良好的多样性,但这种方法在更高维度的 MOP 上表现不佳;文献[16]引入了计算方差的拥挤距离公式,通过自适应的精英保留策略提高了算法的收敛性与分布性。

可以发现,大部分研究者对 NSGA-II 种群分布性的改进策略会涉及父子代种群的进化过程。本文针对拥挤系数算子存在优秀个体不被保留的缺陷,引入了执行双败淘汰制的败者组外部归档集,即不给予父子代种群额外的选择压力,而是在外部归档集内执行相应的归档与剪枝策略。在迭代完成后,对于末代父代与归档集的合并种群,采用文献[10]设计的循环拥挤系数排序策略来筛选个体,以获取最终解集。在算法收敛性的改进上,本文提出了一种将两种传统编码方式结合的混合编码策略,本文将基于以上改进方法的算法称为 LGHC-NSGA-II(Loser Group and Hybrid Coding NSGA-II)。

2 问题分析与算法构造

2.1 败者组策略

NSGA-II 的遗传操作能在迭代过程中逐渐将种群收敛到 Pareto 最优解集,然而使用拥挤系数算子作为合并种群的筛选算子有时会淘汰部分更接近真实 PF 的个体。

如图 1 所示,该图表示 NSGA-II 对某一问题迭代到某次时,合并种群内非支配个体的分布情况。假设当前合并种群中存在 7 个非支配个体并需从中选取 6 个,根据 NSGA-II 在精英选择中的拥挤系数计算公式:

$$L_i = \sum_{m=1}^M (f_m(L_{i+1}) - f_m(L_{i-1})) / (f_m^{\max} - f_m^{\min})$$

可知,虽然非支配个体 A 比合并种群中的另外 3 个非支配个体(B,C,D)更接近真实 PF(红线),但拥挤系数的计算公式并不考虑个体是否已经接近或处于真实 PF,只根据计算得到的适应度值(拥挤系数值)淘汰合并种群中适应度更低(拥挤系数值更小)的个体。虽然这种策略在一定程度上能维持解在全局空间上的分布性,但也导致了某些接近或已经处于 Pareto 最优解集的非支配个体被淘汰,而这些非支配个体在随后的迭代过程(遗传操作)中不一定会再次产生。

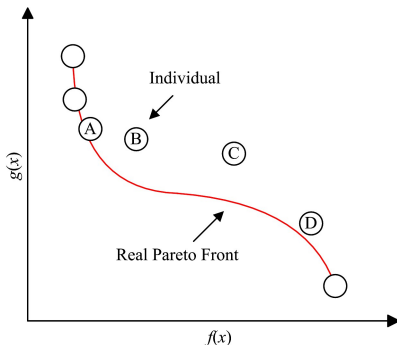


图 1 拥挤系数算子的筛选缺陷(D)(电子版为彩色)

Fig. 1 Screening defects of crowding coefficient operators(I)

在实际应用场景中(例如云资源的调度问题),无法提前

得知所构造的目标函数的真实 PF(除非遍历所有可能解),因此难以判断某次迭代中产生的非支配个体是否接近或已处于 Pareto 最优解集。对此,本文参考了在棋类比赛中常用的双败淘汰制,构造了一种外部归档集,称其为败者组。

败者组双败淘汰制的执行过程如图 2 所示。在每一次迭代过程中的精英选择操作后,收集所有被淘汰的非支配个体,将其放入败者组,并计算个体的 Pareto 支配关系,仅保留非支配个体。若当前败者组内非支配个体的数量超过预设大小,则对败者组执行剪枝操作,剪枝操作由拥挤系数算子完成,在败者组内适应度仍然较差的个体将被彻底遗弃。

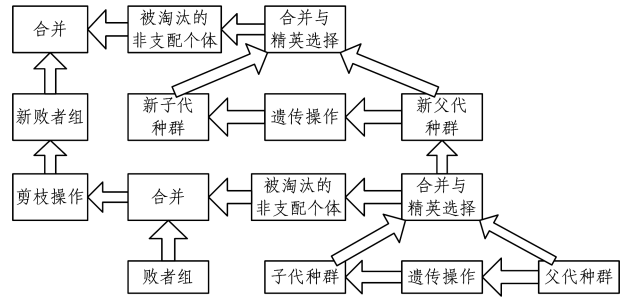


图 2 种群与败者组之间的协作流程

Fig. 2 Collaboration process between population and loser group

在算法迭代结束后,将得到末代父代种群与败者组归档集,将败者组与末代父代种群合并,通过 Pareto 支配关系优化父代种群。图 3 给出改进算法在迭代结束后,败者组与父代种群的个体分布情况。败者组内存在一个个体 A,而父代种群在某个局部区域上的收敛效果较差,即个体 B 和 C,都明显未收敛于 Pareto 最优解集。通过判断合并种群内的 Pareto 支配关系,个体 A 将淘汰个体 B 和 C,并做为最终解集的一部分。该例子说明,双败淘汰制可以有效缓解优秀个体的流失,优秀个体即使在精英选择中被淘汰,依然能在败者组中得到保留。

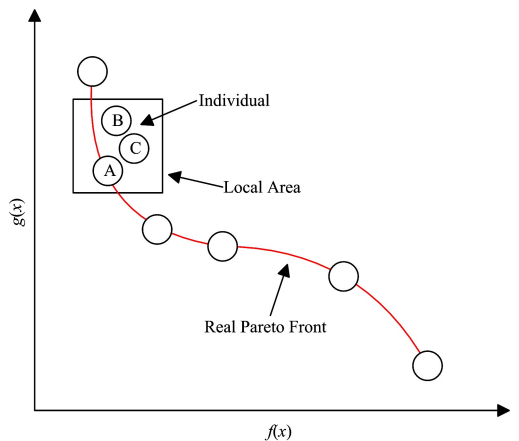


图 3 败者组归档集对子代种群的优化(电子版为彩色)

Fig. 3 Optimization of progeny population of loser group archive set

此外,考虑到算法的时间代价,将精英选择中被淘汰的非支配个体的归档初始时刻设为当大部分个体都接近真实 PF 的时刻。其判断依据为当代父代种群与前代父代种群之间的 Pareto 支配关系,若当代父代种群与前代父代种群之间满足强支配或弱支配关系的个体数量不超过 $k(k < n, n$ 为种群大小),则认为当代父代种群已经收敛于 Pareto 最优解集。

2.2 合并种群的筛选策略

NSGA-II 通过拥挤系数算子对合并种群执行精英选择操作。然而,拥挤系数算子对大量个体执行筛选时有可能出现局部区域分布性优化效果较差的情况,如图 4 所示。

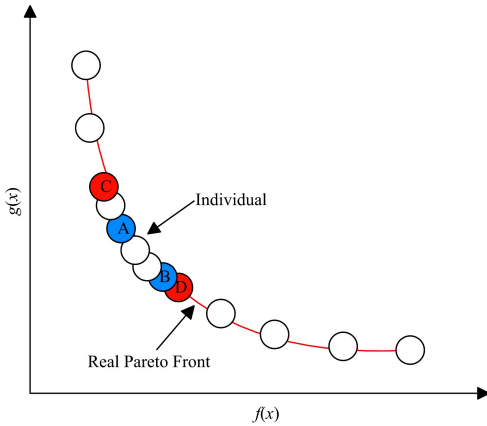


图 4 拥挤系数算子的筛选缺陷(II)(电子版为彩色)

Fig. 4 Screening defects of crowding coefficient operators(II)

假设图 4 中的所有个体都已经收敛于 Pareto 最优解集,现在需要从中选取分布尽可能均匀的个体。根据拥挤系数算子的公式,从拥挤区域内选择了个体 C 和 D(红点)作为算法的解,但显而易见的是个体 A 和 B(蓝点)与拥挤区域外的其他个体形成的 PF 分布更加均匀。

以往研究者针对拥挤系数算子提出了许多不同的改进方法。对此,本文采用了文献[10]的循环排序方法对子代种群和败者组的合并种群执行筛选操作。

循环拥挤系数排序的伪代码如算法 1 所示。

算法 1 Cycled-congestion-coefficient-ranking

```
while  $a_s > p_s$  do // 当个体数量大于初始种群大小
  Crowding-distance-assignment; // 拥挤系数计算
  Fast-sorting; // 按拥挤系数值快速排序的个体
   $a_s = a_s - 1$ ; // 淘汰拥挤系数值最小的个体
end while
```

在采用循环拥挤系数排序策略后,图 4 中个体的筛选过程如图 5 所示。

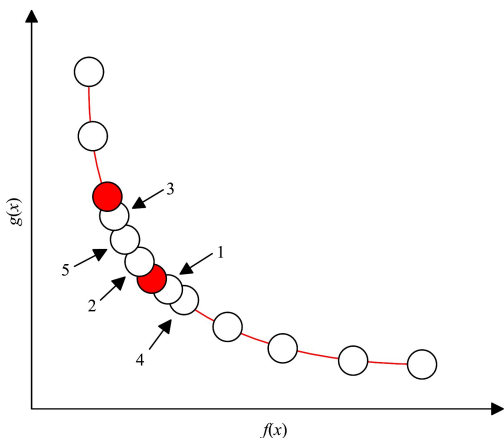


图 5 循环拥挤系数排序策略(电子版为彩色)

Fig. 5 Sorting strategy for cyclic congestion coefficient

循环排序策略实际上每次只淘汰一个拥挤系数值最小的个体,若当前种群大小为 a_s 、初始种群为 p_s ($a_s > p_s$),则需要

重复执行 $a_s - p_s$ 次。图 5 中的红点表示最终选取的个体,数字表示个体被淘汰的顺序。对比图 4 与图 5 的筛选结果发现,相比单次排序,循环拥挤系数排序策略对种群分布性的优化效果有较高的提升。

然而,循环排序方式的时间复杂度较高,若重复执行 T 次,则该操作的时间复杂度为 $O(TMN \log N)$ 。文献[10]的算法在每次迭代时都会对合并种群执行循环排序,导致整体的时间复杂度较高,这也是该算法的缺陷。因此,本文仅在迭代结束后对败者组与父代种群的合并种群执行循环拥挤系数排序,而每次迭代时对败者组的剪枝操作不适用于该策略。

此外,本文在该节做出了如下假设:合并种群内的个体数目越多,即败者组归档集越大,筛选得到的种群在 PF 上的分布性就越好,消耗的时间也越长。本文将在实验部分对该假设进行验证。

2.3 混合编码策略

编码方式是 GA 的一个关键问题,编码方式有多种,如二进制编码、格雷码编码、实数编码、序号编码。对于某些特定问题(如 ZTD 系列问题中的 ZDT4 问题^[17]),在基于一般的交叉和变异操作(如二进制编码中的单点交叉和按位变异、实数编码中的模拟二进制交叉和多项式变异)下,无论是实数编码方式还是二进制编码方式,NSGA-II 都难以将种群收敛到 Pareto 最优解集。

二进制编码由 Holland^[18] 提出,其简单且易于实现,但二进制编码存在 Hamming 悬崖,即表现型空间中的距离(欧氏距离)很小的个体对可能在基因型空间中的距离(Hamming 距离)很大,格雷码编码^[19]的提出就是为了解决该问题。在一组数的编码中,若任意两个相邻代码只有一位二进制数不同,则称这种编码为格雷码编码。二进制编码和格雷码编码之间可互相转换。若一个二进制码为 $B = b_m b_{m-1} \dots b_2 b_1$, 对应的格雷码为 $G = g_m g_{m-1} \dots g_2 g_1$ 。则二进制码到格雷码的转换公式和格雷码到二进制码的转换公式分别如式(1)、式(2)所示:

$$g_m = b_m, g_i = b_i \oplus b_{i+1}, m-1 \geq i \geq 1 \quad (1)$$

$$b_m = g_m, b_i = g_i \oplus b_{i+1}, m-1 \geq i \geq 1 \quad (2)$$

格雷码编码虽然提高了 GA 对全局空间的搜索能力,但同时也降低了 GA 对局部空间的搜索能力。格雷码编码与二进制编码之间存在着良好的互通性,本文提出了一种混合编码策略,即将二进制编码与格雷码编码相结合。

在迭代前期使用格雷码编码来提高算法的全局搜索能力,以使种群快速收敛于 Pareto 最优解集,在迭代后期使用二进制编码来提高算法的局部搜索能力,以使种群精确地收敛于 Pareto 最优解集。对于编码方式转换的界限,本文引入常用于评估在未知真实 PF 情况下算法的收敛性与分布性的 HV^[20]评价指标。二目标 MOP 的 HV 评价指标的计算方式如图 6 所示。图 6 中,虚线围成的所有方形的面积之和(扩展到 M 目标则是所有超体积之和)即为 HV 值, HV 值越大则表示算法获得解的收敛性与分布性越好。HV 可以衡量在格雷码编码下算法是否已经收敛于真实 PF。本文设定每经过 l 次迭代,若当代父代种群的 HV 值 c_{ho} 与最近 l 次迭代时的平均父代种群的 HV 值 a_{ho} 之间的偏差不超过 $\pm c\%$, 则当代父代已经收敛于真实 PF,此时需要采用二进制编码提高

算法的局部搜索能力。这种基于 HV 评价指标的判断方法相对于 2.1 节中基于 Pareto 支配关系的判断方法能更准确地鉴别当前基于格雷码编码下的算法是否已经无法继续提升其收敛性。同时,该混合编码策略在提高算法收敛性上的有效性将会在实验部分进行验证。

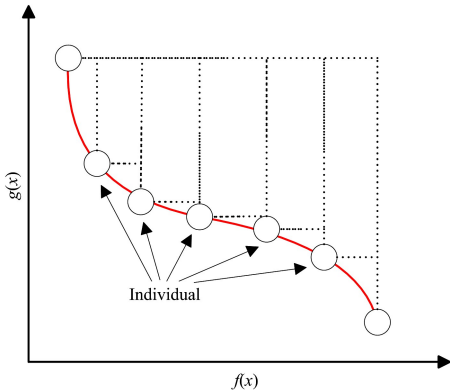


图 6 HV 指标计算二目标 MOP

Fig. 6 HV indicator calculating two-objective MOP

2.4 改进算法的整体框架

本文将基于败者组与混合编码策略的 NSGA-II 改进算法称为 LGHC-NSGA-II,该算法的详细流程如算法 2 所示。

算法 2 改进算法 LGHC-NSGA-II

1. 初始化种群并产生第一代父代和子代种群。
2. 迭代开始,合并父子代种群并计算拥挤系数。
3. 执行精英选择并产生父代种群。
4. 比较当代父代种群与前代父代种群之间个体的 Pareto 支配关系,若满足强支配或弱支配关系的个体数量不超过 $k(k < n, n$ 为种群大小),则将所有在精英选择中被淘汰的非支配个体放入败者组。
5. 对败者组内所有个体判断 Pareto 支配关系,仅保留非支配个体。若归档集内个体数量大于预设的归档集大小,则计算拥挤系数并淘汰拥挤系数较小的个体。
6. 对父代种群执行遗传操作中的选择操作。
7. 计算当代父代种群的 HV 值 c_{hv} 与每 1 次迭代内平均父代种群的 HV 值 a_{hv} ,若 c_{hv} 与 a_{hv} 之间的偏差不超过 $\pm c\%$,则使用二进制编码替代格雷码编码(若迭代少于 1 次则不用执行该步骤)。
8. 对父代种群执行遗传操作中的交叉和变异操作并产生子代种群。
9. 本轮迭代结束,转到步骤 2 开始新一轮的迭代操作。
10. 当达到预定迭代次数后,合并父代种群与败者组归档集,并对合并种群执行循环拥挤系数排序策略,直到合并种群与初始种群的大小一致,将该种群作为算法的最终解集。

从时间复杂度来看,与原算法相比,新产生的步骤 4、步骤 5、步骤 7、步骤 10 的时间复杂度分别为 $O(MN^2)$, $O(MN^2)$, $O(MN \log N)$, $O(MN^2)$,因此改进算法并没有增加额外的时间复杂度,总的时间复杂度和原算法保持一致,为 $O(MN^2)$ 。

3 实验结果与分析

3.1 测试问题及对比文献

本文选择以往研究者常采用的标准测试问题 ZDT 系列函数^[17],以便将本文算法与以往算法在同种群规模 and 同重复实验次数下进行对比。

本文选取了 8 种不同的多目标算法,除了 NSGA-II 和文献^[10]的算法(INSAGA-II)以外,还包括了两种基于分解的多目标算法(分解多目标算法的典型代表 MOEA/D 和基于动态邻域的分解多目标进化算法 MOEA/D-DNS^[21])、两种基于粒子群的多目标算法(速度约束的多目标粒子群算法 SMP-SO^[22]和基于 Pareto 熵的多目标粒子群算法 peMOPSO^[23]),以及两种基于人工免疫的多目标算法(混合多目标免疫算法 HEIA^[24]和基于动态拥挤距离的混合多目标免疫优化算法 DHMOP^[25])。

3.2 实验参数设置

本文中的所有对比实验将直接引用文献中的实验数据(不包括 NSGA-II),因此不对其做额外的参数设置。NSGA-II 在使用二进制编码和格雷码编码时,其遗传算子分别为随机选择、单点交叉和逐位变异,编码字符串长度 $l=15$,交叉概率 $p_m=0.9$,变异概率 $p_c=0.1/l$;在使用实数编码时,遗传算子分别为随机选择、模拟二进制交叉和多项式变异,交叉参数 $\mu_c=20$,变异参数 $\mu_m=20$ 。

本文的改进算法 LGHC-NSGA-II 采用混合编码,遗传算子分别为随机选择、单点交叉和逐位变异,编码字符串长度 $l=15$,交叉概率 $p_m=0.9$,格雷码编码和二进制编码的变异概率分别为 $0.1/l$ 与 $0.05/l$,败者组归档集的大小为 200,判断败者组机制参与执行的参数 $k=5$,编码方式转换评价周期 $l=10$,偏差参数 $c=5$ 。所有实验重复计算 30 次,初始种群规模为 100,函数评估次数为 20 000。实验平台为 Unity3D;编程语言为 C#;CPU 为 Intel(R) Core(TM) i5-4590 3.30 GHz;内存为 16GB;操作系统为 Windows 7。

3.3 性能评价指标

本文选取了 3 种评价指标,包括一种综合性能评价指标 IGD^[26]、一种分布性评价指标 Δ ^[5] 和一种收敛性评价指标 GD^[27]。这 3 种评价指标的值越接近 0,说明解的质量越高。

IGD 是一种综合性能评价指标,能衡量算法的收敛性与分布性,其计算方法如式(3)、式(4)所示:

$$IGD = \frac{1}{|P|} \sum_{i=1}^{|P|} Dist_i \quad (3)$$

$$Dist_i = \min_{j=1}^{|A|} \sqrt{\sum_{m=1}^M \left(\frac{f_m(p_i) - f_m(a_j)}{f_m^{\max} - f_m^{\min}} \right)^2} \quad (4)$$

其中, P 为均匀分布在真实 PF 上的解集(本文在实验中取 500 个点), $Dist_i$ 表示 P 中点 i 与算法获得解 j 的最小归一化欧氏距离, M 为目标个数, f_m^{\max} 与 f_m^{\min} 表示 P 在第 m 个函数上的最大值和最小值。

Δ 是一种包含计算真实 PF 边界值与实际个体间距离的分布性评价指标,计算方式如式(5)所示:

$$\Delta = \frac{d_f + d_t + \sum_{i=1}^{N-1} |d_i - \bar{d}|}{d_f + d_t + (N-1)\bar{d}} \quad (5)$$

其中, d_f 和 d_t 表示真实 PF 的边界值与获得解的边界值之间的欧氏距离, d_i 为相邻个体之间的欧氏距离。

GD 是一种度量算法收敛性的评价指标,通过计算算法获得解映射的 PF 与真实 PF 之间的距离,计算方法如式(6)所示:

$$GD = \frac{1}{n} \sqrt{\sum_{i=1}^n d_i^2} \quad (6)$$

其中, N 表示个体数量, d_i 为个体与真实 PF 的最小欧氏距离。

3.4 优化效果

本实验将改进算法 LGHC-NSGA-II 与基于二进制编码的 NSGA-II 对 ZDT 系列问题的优化效果进行了对比, 结果分别如图 7—图 11 所示。

从优化效果对比图可以看到, 本文的改进算法在 ZDT 系列函数上的分布性与收敛性都明显优于 NSGA-II。在分布性方面, 改进算法的个体在 PF 上的分布较均匀, 而 NSGA-II 的个体在 PF 上存在大量局部拥挤区域; 在收敛性方面, NSGA-II 在 ZDT1—ZDT3 问题上的收敛性劣于改进算法, 且在 ZDT4 问题上无法收敛。这是因为 ZDT4 问题的搜索空间有 2^{19} 个不同的局部 PF, 但其中只有一个与全局 PF 相关。文献[5]采用基于实数编码的 NSGA-II, 其也无法将种群收敛到 Pareto 最优解集, 说明二进制编码和实数编码对于存在大量局部 PF 的 MOP 处理效果均不理想。为了得到进一步的定量分析, 下文将分别基于 3 种评价指标以及归档集的大小进行 4 组对比实验。

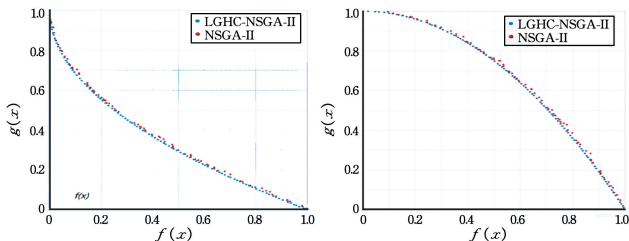


图 7 ZDT1 测试函数
Fig. 7 ZDT1 test function

图 8 ZDT2 测试函数
Fig. 8 ZDT2 test function

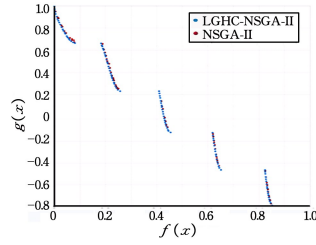


图 9 ZDT3 测试函数
Fig. 9 ZDT3 test function

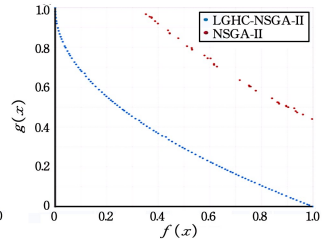


图 10 ZDT4 测试函数
Fig. 10 ZDT4 test function

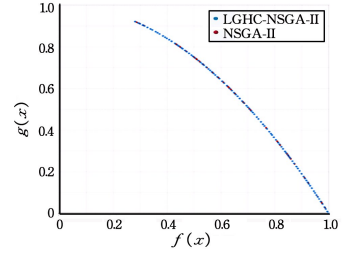


图 11 ZDT6 测试函数
Fig. 11 ZDT6 test function

3.5 基于 IGD 评价指标的对比实验

本实验将使用 IGD 评价指标在 ZDT 系列问题上将本文算法与其他 7 种算法 (NSGA-II, peMOPSO, SMPSO, MOEA/D-DNS, MOEA/D, HEIA, DHMOP) 进行对比, 其目的是验证本文改进算法相对于其他各改进 MOEA 的优越性。除了 NSGA-II, 其他对比算法在各自文献中的实验部分的函数评估次数分别为 30 000, 25 000, 30 000, 25 000, 25 000, 25 000, 实验结果如表 1、表 2 所列。本文表中的加粗字体均表示最优值。

表 1 不同算法的 IGD 平均值的对比数据

Table 1 Comparison of IGD mean values of different algorithms

	NSGA-II	peMOPSO	SMPSO	MOEA/D-DNS	MOEA/D	HEIA	DHMOP	LGHC-NSGA-II
ZDT1	1.09×10^{-2}	4.08×10^{-3}	5.93×10^{-3}	5.70×10^{-3}	5.50×10^{-3}	3.90×10^{-3}	4.60×10^{-3}	3.98×10^{-3}
ZDT2	1.25×10^{-2}	4.19×10^{-3}	5.33×10^{-3}	4.10×10^{-3}	7.90×10^{-3}	3.96×10^{-3}	4.80×10^{-3}	3.94×10^{-3}
ZDT3	6.74×10^{-3}	3.39×10^{-3}	5.10×10^{-3}	3.00×10^{-3}	1.43×10^{-2}	4.43×10^{-3}	1.05×10^{-2}	2.88×10^{-3}
ZDT4	2.32×10^{-1}	1.84×10^{-2}	6.02×10^{-3}	5.21×10^{-2}	7.60×10^{-3}	3.84×10^{-3}	7.50×10^{-3}	3.71×10^{-3}
ZDT6	1.03×10^{-2}	3.67×10^{-3}	4.43×10^{-3}	2.80×10^{-3}	4.20×10^{-3}	3.02×10^{-3}	1.90×10^{-3}	3.78×10^{-3}

表 2 不同算法的 IGD 方差的对比数据

Table 2 Comparison of IGD variances for different algorithms

	NSGA-II	peMOPSO	SMPSO	MOEA/D-DNS	MOEA/D	HEIA	DHMOP	LGHC-NSGA-II
ZDT1	9.99×10^{-4}	2.41×10^{-4}	2.60×10^{-4}	6.00×10^{-4}	2.00×10^{-4}	6.75×10^{-5}	3.00×10^{-4}	3.90×10^{-5}
ZDT2	1.19×10^{-3}	1.77×10^{-4}	1.70×10^{-4}	4.00×10^{-4}	1.09×10^{-2}	5.23×10^{-5}	6.00×10^{-4}	2.87×10^{-5}
ZDT3	4.12×10^{-4}	1.04×10^{-4}	7.30×10^{-4}	3.00×10^{-4}	9.10×10^{-3}	5.41×10^{-5}	9.40×10^{-3}	4.03×10^{-5}
ZDT4	1.67×10^{-1}	2.26×10^{-2}	4.30×10^{-4}	4.79×10^{-2}	2.30×10^{-3}	2.00×10^{-4}	2.50×10^{-3}	4.66×10^{-5}
ZDT6	1.04×10^{-3}	4.00×10^{-4}	3.00×10^{-4}	0.00	3.00×10^{-4}	1.29×10^{-4}	1.00×10^{-4}	3.39×10^{-5}

实验结果表明, 本文改进算法在 ZDT1 上的综合性能稍差于 HEIA, 但好于其他算法, 在 ZDT6 上的表现差于其他 4 种算法, 而在 ZDT2—ZDT4 的收敛性与分布性上均好于所有算法。还可以发现, 除了本文算法、HEIA 和 SMPSO, 其他算法在 ZDT4 上的表现明显差于其在其他 ZDT 系列问题上的表现, 这说明局部 PF 的数量对算法的性能产生了较大的影响。在算法的鲁棒性方面 (见表 2), 本文改进算法除了在 ZDT6 的方差上差于 MOEA/D-DNS, 其他数据均好于其他算法。实验数据说明, 本文改进算法的收敛性、分布性与鲁棒性

在更少的函数评估次数下优于所对比的其他算法。

3.6 基于 Δ 评价指标的对比实验

该实验将使用 Δ 评价指标在 ZDT 系列问题上将所提算法与其他两种算法 (NSGA-II, INSGA-II) 进行对比, 目的是测试在借鉴了文献[10]的循环拥挤系数排序策略后, 所提算法与原算法 NSGA-II 和文献[10]的算法 INSGA-II 相比在分布性上的改善情况, 函数评估次数均为 20 000, 实验结果如表 3 所列 (由于文献[10]缺少 ZDT6 的实验数据, 因此不对该问题做对比实验)。

表 3 不同算法的 Δ 平均值与方差的对比数据

Table 3 Comparison of Δ mean and variance of different algorithm

	NSGA-II		INSGA-II		LGHC-NSGA-II	
	$\Delta(\text{avg})$	$\Delta(\text{var})$	$\Delta(\text{avg})$	$\Delta(\text{var})$	$\Delta(\text{avg})$	$\Delta(\text{var})$
ZDT1	7.35×10^{-1}	4.99×10^{-2}	2.41×10^{-1}	1.70×10^{-4}	2.04×10^{-1}	8.08×10^{-3}
ZDT2	8.09×10^{-1}	3.02×10^{-2}	4.01×10^{-1}	1.59×10^{-1}	2.09×10^{-1}	6.74×10^{-3}
ZDT3	9.07×10^{-1}	3.69×10^{-2}	5.70×10^{-1}	1.36×10^{-3}	4.56×10^{-1}	7.17×10^{-3}
ZDT4	9.45×10^{-1}	2.80×10^{-2}	5.36×10^{-1}	1.20×10^{-1}	2.07×10^{-1}	4.81×10^{-3}

通过实验结果发现,本文改进算法在 4 个测试问题上的分布性均明显优于其他两种算法,且时间复杂度少于文献[10]的算法 INSGA-II。在算法的鲁棒性方面,INSGA-II 虽然在 ZDT1 和 ZDT3 上的方差略优于本文改进算法,但在 ZDT2 和 ZDT4 上的表现较差,而本文改进算法在 4 个问题上的方差均较好。实验数据表明,这本文改进算法的分布性与

鲁棒性优于 NSGA-II 与 INSGA-II。

3.7 4 种编码方式的对比实验

本实验将使用 GD 评价指标在 ZDT 系列问题上测试 NSGA-II 在不同编码下的收敛性表现。同时,为了验证本文提出的混合编码策略的有效性,函数评估次数均为 20 000,实验结果如表 4 所列。

表 4 4 种编码方式下的 GD 平均值与方差的对比数据

Table 4 Comparison of GD mean value and variance under four coding modes

	NSGA-II(二进制编码)		NSGA-II(格雷码编码)		NSGA-II(实数编码)		NSGA-II(混合编码)	
	GD(avg)	GD(var)	GD(avg)	GD(var)	GD(avg)	GD(var)	GD(avg)	GD(var)
ZDT1	6.93×10^{-4}	7.08×10^{-5}	3.29×10^{-4}	2.04×10^{-5}	3.25×10^{-4}	3.25×10^{-5}	6.11×10^{-5}	7.37×10^{-6}
ZDT2	7.03×10^{-4}	7.25×10^{-5}	3.01×10^{-4}	5.92×10^{-5}	2.53×10^{-4}	4.80×10^{-5}	4.77×10^{-5}	9.42×10^{-6}
ZDT3	3.24×10^{-4}	2.52×10^{-5}	2.18×10^{-4}	1.30×10^{-5}	1.74×10^{-4}	9.82×10^{-5}	5.67×10^{-5}	2.63×10^{-6}
ZDT4	2.68×10^{-2}	2.07×10^{-2}	8.72×10^{-5}	8.03×10^{-6}	9.36×10^{-3}	6.04×10^{-6}	3.89×10^{-5}	3.87×10^{-6}
ZDT6	8.03×10^{-5}	7.19×10^{-6}	1.55×10^{-5}	7.95×10^{-6}	9.87×10^{-6}	5.10×10^{-6}	7.93×10^{-6}	1.37×10^{-6}

实验结果表明,本文提出的混合编码在 ZDT 系列问题的收敛性与鲁棒性上的表现均明显优于其他 3 种编码方式,说明混合编码策略是可行有效的。

此外,对典型的 3 种编码方式进行比较,二进制编码与实数编码的 NSGA-II 在 ZDT4 问题上的收敛性明显差于格雷码编码的 NSGA-II,而实数编码的 NSGA-II 在其他问题上的表现均好于二进制编码与格雷码编码的 NSGA-II。这说明格雷码编码虽然在一定程度上加强了全局搜索的能力,使种群

更容易跳出局部 PF,但也牺牲了其在局部空间上的搜索能力。因此,从实验数据来看,综合搜索能力最强的是实数编码,最差的是二进制编码(不考虑不同遗传算子带来的差异)。

3.8 归档集大小的对比实验

本实验将使用 IGD 评价指标在 ZDT 系列问题上测试改进算法在不同归档集大小下的综合性能表现与时间消耗。实验结果如表 5、表 6 所列。

表 5 不同归档集大小下的 IGD 平均值的对比数据

Table 5 Comparison of IGD mean values under different archives sizes

	25	50	75	100	150	200	400	800
ZDT1	5.66×10^{-3}	5.51×10^{-3}	5.13×10^{-3}	4.88×10^{-3}	4.28×10^{-3}	3.98×10^{-3}	3.97×10^{-3}	3.97×10^{-3}
ZDT2	5.67×10^{-3}	5.23×10^{-3}	4.97×10^{-3}	4.85×10^{-3}	4.05×10^{-3}	3.94×10^{-3}	3.94×10^{-3}	3.93×10^{-3}
ZDT3	4.96×10^{-3}	3.81×10^{-3}	3.42×10^{-3}	3.59×10^{-3}	3.10×10^{-3}	2.88×10^{-3}	2.88×10^{-3}	2.87×10^{-3}
ZDT4	6.74×10^{-3}	6.09×10^{-3}	4.84×10^{-3}	4.27×10^{-3}	3.97×10^{-3}	3.71×10^{-3}	3.68×10^{-3}	3.67×10^{-3}
ZDT6	6.01×10^{-3}	4.80×10^{-3}	4.34×10^{-3}	3.96×10^{-3}	3.84×10^{-3}	3.78×10^{-3}	3.76×10^{-3}	3.76×10^{-3}

表 6 不同归档集大小下的时间平均值的对比数据

Table 6 Comparison data of time averages under different archive set sizes

	(单位:s)							
	25	50	75	100	150	200	400	800
ZDT1	5.74	6.33	6.66	6.84	7.32	7.70	12.2	27.2
ZDT2	5.76	5.90	6.47	6.58	7.01	7.85	12.1	26.8
ZDT3	5.67	6.20	6.31	6.89	7.59	8.11	13.4	28.9
ZDT4	3.22	3.33	3.48	3.66	4.07	5.17	10.4	24.2
ZDT6	3.05	3.26	3.72	4.16	4.36	4.90	8.88	20.5

实验结果表明,随着败者组归档集大小的逐渐增大,最终获得解的质量也越来越好,时间消耗也随之增加,但本文所采用的循环拥挤系数排序策略存在一个优化极限。从表 5 可以看到,当归档集大小分别为 200,400 和 800 时,IGD 值优化幅度的提升非常小,在 ZDT1 和 ZDT6 上保持不变,这说明当归档集大小达到一定程度时,归档集的增大和当前采用的循环

排序策略对算法性能难以进行进一步的优化,且归档集规模为 200 时算法的优化效果较为接近最佳优化效果,因此从运行时间和优化结果来看,归档集规模选择 200 时算法的表现最佳。该实验结果也验证了 2.2 节中提出的假设的正确性。

结束语 本文针对 NSGA-II 的拥挤系数算子在精英选择中的两种缺陷,构造了败者组外部归档集,制定了相应的归档与剪枝策略,并采用了文献[10]中的循环拥挤系数排序方法作为对合并种群的筛选策略。同时,本文提出了一种将二进制编码与格雷码编码相结合的混合编码策略,设计了基于 NSGA-II 的改进算法 LGHC-NSGA-II。在 5 个 ZDT 系列测试问题上该所提算法与 8 种算法做对比,实验结果表明本文的改进算法在收敛性、分布性与鲁棒性上均有较为显著的优势。本文还测试了 4 种编码方式的优劣性,结果表明本文提出的混合编码策略明显优于其他 3 种传统编码方式。未来的研究重点为合并种群的筛选策略,本文在最后一个实验中研

究了归档集大小对算法优化性能的影响,然而当归档集规模达到一定程度时,算法的分布性无法得到进一步提升,这可能是因为本文所采用的循环拥挤系数排序算子在处理大规模数量个体时依然存在问题。尝试选用其他或构造新的筛选算子以进一步提升算法性能是未来的研究目标。

参考文献

- [1] ZITZLER E, THIELE L. Multi-objective evolutionary algorithms: A comparative case study and the strength Pareto approach[J]. *IEEE Transactions on Evolutionary Computation*, 1999, 3(4): 257-271.
- [2] CORNE D W, KNOWLES J D, OATES M J. The Pareto Envelope-Based Selection Algorithm for Multiobjective Optimization [C] // *International Conference on Parallel Problem Solving From Nature*. Springer-Verlag, 2000: 839-884.
- [3] SRINIVAS N, DEB K. Multi-objective optimization using non-dominated sorting in genetic algorithms[J]. *Evolutionary Computation*, 1994, 2(3): 221-248.
- [4] COELLO C A C, PULIDO G T, LECHUGA M S. Handling multiple objectives with particle swarm optimization[J]. *IEEE Transactions on Evolutionary Computation*, 2004, 8(3): 256-279.
- [5] DEB K, AGRAWAL S, PRATAP A, et al. A Fast Elitist Non-dominated Sorting Genetic Algorithm for Multi-objective Optimization: NSGA-II [C] // *International Conference on Parallel Problem Solving from Nature*. Springer, 2002.
- [6] ZHANG Q, LI H. MOEA/D: A Multiobjective Evolutionary Algorithm Based on Decomposition[J]. *IEEE Transactions on Evolutionary Computation*, 2007, 11(6): 712-731.
- [7] COELLO C A C. Evolutionary multi-objective optimization: a historical view of the field[J]. *IEEE Computational Intelligence Magazine*, 2006, 1(1): 28-36.
- [8] DEB K, JAIN H. An Evolutionary Many-Objective Optimization Algorithm Using Reference-Point-Based Nondominated Sorting Approach, Part I: Solving Problems With Box Constraints[J]. *IEEE Transactions on Evolutionary Computation*, 2014, 18(4): 577-601.
- [9] LI M, ZHENG J, WU J. Improving NSGA-II Algorithm Based on Minimum Spanning Tree [C] // *Asia-Pacific Conference on Simulated Evolution and Learning*. Springer Berlin Heidelberg, 2008: 170-179.
- [10] LUO C Y, CHEN M Y, ZHANG C Y. Improved NSGA-II algorithm with circular crowded sorting[J]. *Control and Decision*, 2010, 25(2): 227-231. (in Chinese)
罗辞勇, 陈民铀, 张聪誉. 采用循环拥挤排序策略的改进 NSGA-II 算法[J]. *控制与决策*, 2010, 25(2): 227-231.
- [11] SATO Y, SATO M, MIYAKAWA M. Distributed NSGA-II with migration using compensation on many-core processors for improving performance and accuracy [C] // *the Genetic and Evolutionary Computation Conference Companion*. 2017: 161-162.
- [12] KUMAR M, GURIA C. The elitist non-dominated sorting genetic algorithm with inheritance (i-NSGA-II) and its jumping gene adaptations for multi-objective optimization[J]. *Information Sciences*, 2017, 382-383: 15-37.
- [13] XIE C W, LI K, LIAO G Y. Improved NSGA2 Algorithm with Differential Evolution Local Search[J]. *Computer Science*, 2013, 40(10): 235-238. (in Chinese)
谢承旺, 李凯, 廖国勇. 一种带差分局部搜索的改进型 NSGA2 算法[J]. *计算机科学*, 2013, 40(10): 235-238.
- [14] SCHOTT J R. Fault Tolerant Design Using Single and Multicriteria Genetic Algorithm Optimization [J]. *Cellular Immunology*, 1995, 37(1): 1-13.
- [15] VACHHANI V L, DABHI V K, PRAJAPATI H B. Improving NSGA-II for solving multi objective function optimization problems [C] // *International Conference on Computer Communication and Informatics*. IEEE, 2016: 1-6.
- [16] LAI W X, DENG Z M. Improved NSGA2 Algorithm Based on Dominant Strength [J]. *Computer Science*, 2018, 45(6): 187-192. (in Chinese)
赖文星, 邓忠民. 基于支配强度的 NSGA2 改进算法[J]. *计算机科学*, 2018, 45(6): 187-192.
- [17] ZITZLER E, DEB K, THIELE L. Comparison of Multiobjective Evolutionary Algorithms: Empirical Results [J]. *Evolutionary Computation*, 2000, 8(2): 173-195.
- [18] HOLLAND J H. Adaptation in natural and artificial systems [J]. *Ann Arbor*, 1992, 6(2): 126-137.
- [19] GRAY F. Pulse code communication [P]. U. s. patent 2,1953.
- [20] ZITZLER E, THIELE L. Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach [J]. *IEEE Transactions on Evolutionary Computation*, 1999, 3(4): 257-271.
- [21] ZHOU H, MENG L M, WANG L P, et al. Multi-objective Evolutionary Algorithm Based on Decomposition Using Dynamic Neighbor [J]. *Mini-micro Systems*, 2017, 38(9): 2039-2044. (in Chinese)
周欢, 孟利民, 王丽萍, 等. 动态邻域的分解多目标进化算法[J]. *小型微型计算机系统*, 2017, 38(9): 2039-2044.
- [22] NEBRO A J, DURILLO J J, GARCIA-NIETO J, et al. SMPSO: A new PSO-based metaheuristic for multi-objective optimization [C] // *Computational Intelligence in Multicriteria Decision-making*, IEEE, 2009: 66-73.
- [23] HU W, YEN G G, ZHANG X. Multiobjective Particle Swarm Optimization Based on Pareto Entropy [J]. *Journal of Software*, 2014, 25(5): 1025-1050. (in Chinese)
胡旺, YEN G G, 张鑫. 基于 Pareto 熵的多目标粒子群优化算法 [J]. *软件学报*, 2014(5): 1025-1050.
- [24] LIN Q, CHEN J, ZHAN Z H, et al. A Hybrid Evolutionary Immune Algorithm for Multiobjective Optimization Problems [J]. *IEEE Transactions on Evolutionary Computation*, 2016, 20(5): 711-729.
- [25] MA Y F, LI A R, YU H M, et al. Dynamic Crowding Distance-based Hybrid Immune Algorithm for Multi-objective Optimization Problem [J]. *Computer Science*, 2018, 45(6): 63-68. (in Chinese)
马元锋, 李昂儒, 余慧敏, 等. 基于动态拥挤距离的混合多目标免疫优化算法[J]. *计算机科学*, 2018, 45(6): 63-68.
- [26] VAN D A, GARY V, LAMONT B. Multiobjective Evolutionary Algorithm Research: A History and Analysis [J]. *Evolutionary Computation*, 1998, 8(2): 125-147.
- [27] VAN VELDHUIZEN D A, LAMONT G B. Evolutionary computation and convergence to a Pareto front [C] // *Koza J R. Late Breaking Papers at the Genetic Programming 1998 Conference*. Stanford University, California, 1998: 221-228.