

一种基于新型邻域更新策略的 MOEA/D 算法

耿焕同 韩伟民 周山胜 丁洋洋

(南京信息工程大学计算机与软件学院 南京 210044)

摘要 针对 MOEA/D 算法求解复杂优化问题时,邻域更新策略的无限制替换易造成种群多样性缺失的问题,提出了一种基于新型邻域更新策略的 MOEA/D 算法(MOEA/D-ENU)。该算法在进化过程中对解的信息进行充分挖掘,按照邻域更新能力对产生的新解进行分类,并针对不同类型的新解,自适应地采取不同的邻域更新策略,在保证种群收敛速度的同时,又兼顾了种群的多样性。实验中,选取 ZDT,UF,CF 等 9 个函数作为标准测试集,将改进后的算法 MOEA/D-ENU 与其他 5 种算法进行对比实验,并以 IGD 和 HV 为评估指标。实验结果表明新算法具有更好的收敛性和分布性。

关键词 基于分解的多目标进化算法,挖掘解,分类,邻域更新策略

中图分类号 TP183 文献标识码 A DOI 10.11896/j.issn.1002-137X.2019.05.029

MOEA/D Algorithm Based on New Neighborhood Updating Strategy

GENG Huan-tong HAN Wei-min ZHOU Shan-sheng DING Yang-yang

(College of Computer & Software, Nanjing University of Information Science & Technology, Nanjing 210044, China)

Abstract To solve the problem of the lack of population diversity caused by the unrestricted replacement of neighbourhood updating strategy when the MOEA/D algorithm solves the complex optimization problem, a new MOEA/D algorithm based on new Neighbourhood Updating Strategy (MOEA/D-ENU) was proposed. In the process of evolution, the algorithm fully excavates the information of the solution, classifies the new solution generated according to the capacity of neighbourhood updating, and adopts different neighbourhood updating strategies adaptively to different types of new solutions to ensure the population convergence rate. At the same time, it takes into account the diversity of the population. The proposed algorithm was compared with five other algorithms on 9 benchmarks including ZDT, UF and CF. The values of IGD and HV show that MOEA/D-ENU has certain advantages than other algorithms in terms of convergence and distribution.

Keywords Multiobjective evolutionary algorithm based on decomposition, Excavating solution, Classification, Neighbourhood updating strategy

1 引言

在实际应用中,通常将有多个优化目标且目标之间相互冲突的问题称为多目标优化问题(Multi-objective Optimization Problems, MOPs)^[1]。进化算法作为一种启发式的搜索算法,由于不涉及复杂的数学运算,又能得到一组或多组最优解,常常被用于求解 MOPs。

在过去的十几年,有许多优秀的多目标进化算法被提出。这些多目标进化算法可以被分为三大类:基于 Pareto 支配的多目标进化算法^[2-3]、基于指标的多目标进化算法^[4-5]和基于分解的多目标进化算法^[6-7]。其中,基于分解的多目标进化算法由于引进了数学规划中的分解策略,将多目标问题分解为

若干个单目标优化子问题,并同时对这些子问题进行优化。分解策略的引入使该类算法在处理复杂和高维目标的优化问题时有着明显的优势。

基于分解的多目标进化算法(MOEA/D)^[8]在 2007 年被 Zhang 首次提出,该算法因简单、高效而受到学者们的广泛关注研究。2009 年, Zhang 提出了 MOEA/D-DE^[9]算法,即采用拥有更强探索和勘探能力的 DE 算子替换原先的 SBX 算子;2014 年,侯薇提出了 LMS-MOEA/D 算法,即利用算子的混合交叉策略使不同算子的优势得到充分发挥^[10];马庆于 2016 年提出一种使用多目标进化算法产生任意多个均匀分布的权重向量的方法^[11]。在邻域方面,2009 年, Zhang^[10]指出无限制的邻域更新策略在进化过程中会使种群多样性缺

到稿日期:2018-04-04 返修日期:2018-07-27 本文受国家重点研发计划(2017YFC1502104),江苏省自然科学基金(BK20151458),江苏省“青蓝工程”(2016)资助。

耿焕同(1973—),男,教授,博士生导师,CCF 高级会员,主要研究方向为气象数据挖掘、计算智能、多目标优化, E-mail: htgent@nuist.edu.cn (通信作者);韩伟民(1992—),男,硕士生,主要研究方向为多目标优化;周山胜(1992—),男,硕士生,主要研究方向为多目标优化;丁洋洋(1992—),男,硕士生,主要研究方向为高维多目标优化。

失;2013年,郑金华教授更是指出了无限制的邻域更新策略具有随机指配、重复更新的缺陷。针对邻域更新策略存在的不足,2015年,Wang提出了基于自适应全局替换策略的算法MOEA/D-AGR^[12],即将自适应全局替换策略引入了邻域更新,很好地弥补了邻域更新在全局搜索方面的不足;2017年,Zhang提出MOEA/D-GR^[13]算法,即根据种群的分布性与收敛性动态调整邻域大小,通过控制替换邻域的大小来实现精英解的全局替换,最终达到维持种群的多样性分布的目的。

上述算法虽然在一定程度上改善了MOEA/D算法的性能,但对于邻域更新策略中无限制替换存在的缺陷,并未给出较好的解决方法。因此,本文提出了一种新型的邻域更新策略。在进化过程中,充分挖掘解的信息,将新解按照其更新邻域能力的大小归为4类;针对这4种类型的新解,提出了与之匹配的4种邻域更新策略,新型邻域更新策略在加速种群收敛速度的同时,又兼顾了种群的多样性。基于此,本文提出了一种基于新型邻域更新策略的MOEA/D算法。

2 相关研究与问题提出

2.1 基于分解的多目标进化算法

传统的MOEA/D算法使用聚合函数将MOPs分解为N个子问题 $P = \{p^1, \dots, p^N\}$,并同时对这些子问题进行优化。通过计算权重向量之间的Euclidean距离来为每个子问题划分邻域,并且通过邻域内的子问题完成协同进化,MOEA/D算法的主要进化过程如下。

Step1 初始化

Step1.1 初始化均匀权重向量 $\lambda = \{\lambda_1, \lambda_2, \dots, \lambda_N\}$,并为均匀权重划分对应邻域;对于个体 $x_i (i = 1, \dots, N)$,设置 $B(i) = \{i_1, \dots, i_T\}$ 为第i个权重向量的邻域。

Step1.2 初始进化种群 $x = \{x^1, \dots, x^N\}$,并计算目标值 $FV^i = F(x^i)$ 。

Step1.3 计算理想点,记为 $z = \{z_1, \dots, z_m\}^T$,其中m为目标维数, z_i 为当前代中 f_i 的最优值。

Step2 更新

对于 $i = 1, \dots, N$,有:

Step2.1 进化:从邻域 $B(i)$ 中随机选择两个个体 x^j 和 x^k ,差分进化产生新个体 y ,并通过相应的问题约束改进为 y' 。

Step2.2 新理想点 z :对于 $j = 1, \dots, m$,若 $z_j > f_j(y')$,则 $z_j = f_j(y')$ 。

Step2.3 更新邻域:对于 $j \in B(i)$,若 y' 的聚合函数值不大于 x^j ,则 $x^j = y'$,并替换相应的目标值 $FV^j = F(y')$ 。

Step3 停止迭代

若满足停止条件,则停止迭代并输出最优解集;否则,继续执行Step2。

在传统MOEA/D中一般使用切比雪夫方法(Tchebycheff)或基于惩罚的边界交叉方法(PBI)将MOPs分解为N个子问题。本文使用的聚合函数是Tchebycheff,其数学描述如式(1)所示:

$$\min g^w(x|\lambda, z^*) = \max_{1 \leq i \leq m} \{\lambda_i | f_i(x) - z_i^* | \} \quad (1)$$

2.2 MOEA/D的邻域更新策略

传统的MOEA/D邻域更新策略的思想是:通过使用若干个邻近子问题的信息优化每个子问题,使子问题朝有利方向进化,最终收敛到前沿。因此,MOEA/D算法具有计算复杂度低和搜索能力强的优点。

聚合函数值可以较好地评价解的优劣程度,因此MOEA/D的邻域更新策略一般以解和子问题的聚合函数值作为适应度值,通过比较新解与其邻域内的其他解的适应度值大小,来逐一择优替换,更新掉邻域内的所有“劣解”。如图1所示,在一次邻域更新操作中,计算新解 y' 的聚合函数值 $g_{(y', \lambda_i)}^w$,并且和与邻域内 x_3, x_4, x_5, x_6 这4个解对应的聚合函数值 $g_{(x_i, \lambda_i)}^w$ 分别进行比较,发现 $g_{(y', \lambda_i)}^w < g_{(x_4, \lambda_i)}^w$,且 $g_{(y', \lambda_5)}^w < g_{(x_5, \lambda_5)}^w$,这说明相比于 x_4 和 x_5 , y' 在子问题 p^4 和 p^5 下的适应度值更高,因此新解 y' 会替换掉邻域中的劣解 x_4 和 x_5 。邻域更新策略通过在每一代选取更优秀的子代个体,使种群最终进化收敛到Pareto前沿。

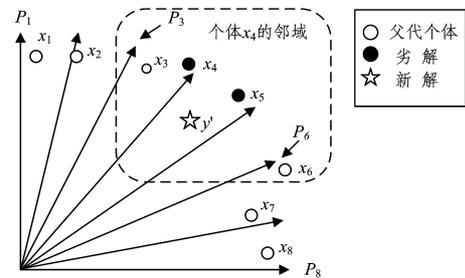


图1 邻域更新策略示意图

Fig.1 Schematic diagram of neighbouring updating

虽然邻域更新可以加快种群的收敛速度,但是从上述的更新过程不难看出,邻域更新策略的无限制替换会造成种群多样性的缺失。这种缺陷会随着新解更新能力的加强而越发明显:当新解 y' 更新掉邻域内劣解的数目越多时,在MOEA/D的邻域协同进化模式的影响下,其他邻域内有效解的数目会急剧变少,邻域内可供选择的父代样本也随之减少,最终造成种群进化减缓,甚至停滞。如图1所示,若子问题 P_4 对应的新解 y' 有能力更新掉邻域内的解 x_3, x_4, x_5, x_6 ,当新解 y' 完成邻域更新操作之后,此时解 x_4 对应邻域内的有效解只有 y' ,解 x_5 邻域内的有效解由原来的 x_4, x_5, x_6, x_7 变为 y', x_7 ,而且其他邻域内有效解的数目也会随之减少,部分子问题在差分进化时因为可供选择的父代样本变少,影响了种群进化效率,大量的解被替换的同时也造成种群多样性缺失,使算法极易陷入局部最优。

3 一种基于新型邻域更新策略的MOEA/D算法

针对MOEA/D算法邻域更新策略的无限制替换会使种群多样性缺失,最终导致算法陷入局部最优的缺陷,本文提出了一种新型的邻域更新策略。根据新解更新邻域能力的不同,自适应地采用相应邻域更新策略,保证算法的收敛性和分布性。最终形成一种基于新型邻域更新策略的MOEA/D算法,记为MOEA/D-ENU。

3.1 挖掘解的信息

从2.2节的分析可知,邻域更新策略的无限制替换会造

成种群多样性缺失。首先,这种缺陷会随着新解更新掉邻域中劣解数目 N_r 的增大而越发严重。其次,邻域中的劣解并不都是“劣解”,对于某个子问题,可能有在分布性或收敛性其中一方面比新解更优的劣解,邻域更新策略替换掉这部分解会使种群多样性缺失的情况更为严重。因此可以保留这部分劣解,以达到维护种群的多样性的目的。综上,在进化过程中需要对新解和其邻域内的劣解分别进行解挖掘操作,获取解的相关信息,为新型邻域更新策略做好前期的信息挖掘工作。

对新解进行解挖掘操作:首先统计出新解更新邻域中劣解的数目 N_r ,并设置 $T/2, 2T/5, 0$ 这 3 个阈值。 N_r 值可以较好地反映出新解更新邻域能力的大小。 N_r 值越大,新解更新邻域能力越强,因此通过比较新解的 N_r 值与设定阈值的大小,可以将新解归为超强解、较强解、普通解、次解这 4 类中的一类,再针对不同类型的新解,自适应地采取不同的邻域更新策略,使种群在快速收敛的同时,又能兼顾种群的多样性。

对劣解进行解挖掘操作:根据劣解的质量将其分成一等劣解和二等劣解 2 类,保留二等劣解,更新掉一等劣解。为了评估劣解的质量,首先根据式(2)计算出新解 y' 和劣解 x_j 所有维的切比雪夫值,并进行降序排序获得 2 组解集 $\{g_1^e, \dots, g_m^e\}$ 和 $\{g_1^{e1}, \dots, g_m^{e1}\}$ 。按照 2.2 节所述经典的邻域更新策略,只要新解的 g_1^e 小于解 x_j 的 g_1^{e1} ,则新解 y' 即可替换掉 x_j ,此时新解 y' 位于图 2 中的区域 1 或区域 2 中。通过 Pareto 支配关系可以明显地看出,位于区域 1 内的新解 y' 的质量相对更好,此时邻域中劣解 x_j 的质量与新解 y' 差距较大。为了更好地评估劣解的质量,可以对这 2 组解集 $\{g_1^e, \dots, g_m^e\}$ 和 $\{g_1^{e1}, \dots, g_m^{e1}\}$ 中所有切比雪夫值进行 m 次比较,若 g_{jj}^e 全部都小于 g^{e1}_{jj} ,则新解 y' 位于图 2 中的区域 1 处,此时将劣解 x_j 归为一等劣解;否则新解 y' 就位于图 2 中的区域 2 处,此时劣解 x_j 被归为二等劣解。通过上述分析可知,用新解更新掉所有一等劣解,保留二等劣解,有利于保持种群多样性。

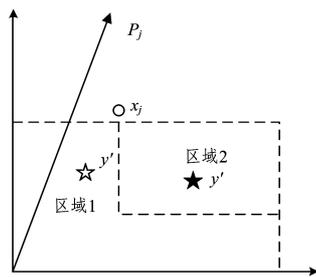


图 2 邻域中劣解的分类

Fig. 2 Categories of inferior solutions in neighbourhood

在进化过程中,通过上述方法对新解和邻域内的劣解信息进行挖掘,为新型邻域更新策略做好前期的信息挖掘工作。

$$g^e(x | \lambda, z^*) = \{\lambda_{jj} | f_{jj}(x) - z_{jj}^* | 1 \leq j \leq m\} \quad (2)$$

解挖掘操作的具体执行步骤如下:

Step1 新解类型判断。差分进化生成新解 y' ,统计出新解 y' 更新邻域内劣解的数目 N_r ,通过比较新解的 N_r 值与设定阈值的大小判断出新解的类型。

Step2 判断劣解的类型。

Step2.1 新解 y' 对应邻域中被替换的劣解的解集为 $C(i) = \{i_1, \dots, i_{N_r}\}$ 。

Step2.2 判断邻域内劣解类型。对于 $j = 1, \dots, N_r$,通过式(2)计算新解 y' 和劣解 x_j 所有维的切比雪夫值,并进行降序排序,分别得到 2 组解集 $\{g_1^e, \dots, g_m^e\}$ 和 $\{g_1^{e1}, \dots, g_m^{e1}\}$ 。

对于 $jj = 1, \dots, m$,若全部都满足 $g_{jj}^e < g_{jj}^{e1}$,则劣解 x_j 为一等劣解,将其并放入一等劣解解集 IS_1 ;否则劣解 x_j 为二等劣解,并将其放入二等劣解解集 IS_2 。

3.2 一种新型的邻域更新策略

通过 3.1 节所述的方法在进化过程中对新解和邻域内劣解信息进行挖掘,获得新解的类型和邻域内劣解的类型。针对不同类型的新解,本文提出了与之匹配的邻域更新策略,使新算法可以做到在加速种群收敛速度的同时,又兼顾种群的多样性。

3.2.1 超强解的邻域滞后重更新策略

超强解在执行邻域更新操作的无限制替换时,会更新掉邻域中的大部分劣解,在 MOEA/D 的邻域协同进化模式的影响下,会严重影响种群的多样性。因此,本文针对超强解提出了邻域滞后重更新策略,其思想是对超强解交替使用邻域更新的限制性替代和邻域更新的无限制替代,综合 2 种邻域更新策略的优势来保持种群的多样性。在每一代进化过程中,当超强解第一次执行邻域更新操作时,设定超强解最大替换阈值 T_r ,对超强解进行邻域更新的限制性替代,使邻域内其他解在之后的进化过程中,有更大的概率进化生成更优的解,以达到维持种群多样性的目的。当所有的新解都完成了一次邻域更新后,再对当代所有超强解进行第二次邻域更新操作,此时邻域内的劣解都是在之前进化过程中的“淘汰者”,未能进化成为更优的个体,因此取消阈值 T_r ,对超强解进行邻域更新的无限制替换,替换掉邻域内所有的劣解。综上所述,在每一代进化过程中,邻域滞后重更新策略既能使邻域内质量不是很差的劣解有机会生成更优秀的新解,又能更新掉质量很差的劣解,极大地保证了种群的多样性。

3.2.2 较强解的邻域择优更新策略

较强解在执行邻域更新操作的无限制替换时,会更新掉邻域中较多的解,从而造成种群多样性缺失。因此,本文针对较强解提出邻域择优更新策略。其思想是对邻域中的劣解进行分类,只更新掉部分质量相对较差的劣解,以达到维持种群多样性的目的。根据 3.1 节所述的方法将邻域中的劣解分成一等劣解和二等劣解 2 类,再用较强解更新掉所有一等劣解,保留下二等劣解。该策略既可以更新邻域中质量相对较差的“一等劣解”,又可以使邻域中质量相对较优的“二等劣解”有机会进化产生更优的子代,有利于维护种群的多样性。

3.2.3 普通解的无限制邻域更新策略

普通解在执行邻域更新操作的无限制替换时,替换掉邻域内劣解的数目比较少,并不会影响种群多样性的维持,还能使种群的收敛速度加快,因此对于普通解继续使用邻域更新策略的无限制替换。

3.2.4 次解的全局适配策略

经典 MOEA/D 算法中的邻域更新策略对于次解直接舍去,不做任何操作。但可能存在邻域外的某一子问题下适应度值很高的次解,简单地舍去次解对于种群多样性的保持

有一定的影响。如图3中次解 xx 在邻域外的子问题 P_4 下的适应度值很高,若对次解进行全局适配,为次解寻找最匹配的子问题 P ,则有利于种群多样性的保持。在算法前期出现上述次解的频数远远高于算法后期,为了避免计算资源的浪费,只在算法前期执行次解全局适配操作。

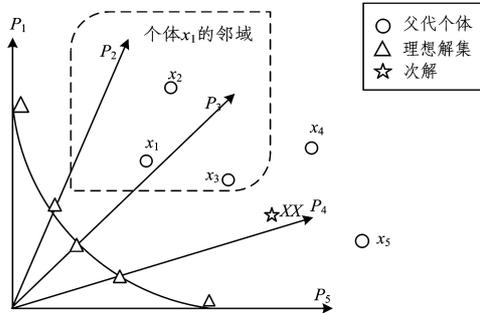


图3 次解的全局适配

Fig. 3 Global adaptation of bad solution

为了能更加准确地判断出算法的时期,本文提出了一种基于邻域更新能力的算法时期判断机制。在每一代都统计有效邻域更新次数 L ,同类型新解出现的频次可以准确地反映出当前种群的进化状态,从而准确地判断出算法的时期。设置阈值 E_r 和 E_l ,为不同类型的新解赋不同的权值,并计算出每一代新解的总值 L_r ,通过比较 L 与 E_r 、 L_r 与 E_l 的值来判断算法的时期。

综上,根据解挖掘操作获得解的一些相关信息,并针对不同类型的新解,自适应地使用不同的邻域更新策略。新型邻域更新策略的具体执行步骤如下。

Step1 判断算法的时期 M_{dis} 。

Step1.1 通过式(3)计算每一代中有效更新邻域($\mu_i = 1$)的次数 L 。

Step1.2 根据新解的类型赋不同权值 $\mu\mu_i$: 超强解 $\mu\mu_i = 3$,较强解 $\mu\mu_i = 2$,普通解 $\mu\mu_i = 1$ 。通过式(4)统计每代邻域更新策略的总值 L_r 。

Step1.3 通过式(5)判断算法的时期,若 $M_{dis} = 0$,则算法处于前期;否则算法处于后期。

Step2 按照3.1节的机制进行解挖掘操作,获得 y' 新解类型,对应邻域的一等劣解解集 IS_1 和二等劣解解集 IS_2 。

Step3 基于解挖掘的邻域更新策略,针对不同类型的新解使用不同的邻域更新策略。

对于 $i = 1, \dots, N$,

若新解为超强解则执行 Step3.1;若新解为较强解执行 Step3.2;若新解为较弱解执行 Step3.3;若新解为次解且 $M_{dis} = 0$ 执行 Step3.4。

Step3.1 执行超强解的邻域滞后重更新操作。用新解 y' 随机更新一等劣解解集 IS_1 中的2个一等劣解,并将超强解 y' 放入超强解集 P_i 中。

Step3.2 执行较强解的邻域择优更新操作。

新解 y' 更新掉一等劣解解集 IS_1 中的所有劣解。

Step3.3 执行普通解的无限制邻域更新操作。

新解 y' 更新一等劣解解集 IS_1 和二等劣解解集 IS_2 中所有的劣解。

Step3.4 执行次解的全局适配操作。

计算并找到与新解 y' 欧几里得距离最近的权重向量 λ_i ,对新解 y' 和 λ_i 对应的解进行择优替换。

Step4 当所有解都完成一次邻域更新操作,对超强解集 P_i 中所有解执行邻域更新操作,操作完毕后将超强解集 P_i 置空。

$$L = \sum_{i=1}^N \mu_i \quad (3)$$

$$L_r = \sum_{i=1}^N \mu\mu_i \quad (4)$$

$$M_{dis} = \begin{cases} 1, & L < E_r \text{ and } L_r < E_l \\ 0, & \text{else} \end{cases} \quad (5)$$

3.3 MOEA/D-ENU 算法的流程

参照 MOEA/D 的进化框架,结合新型的邻域更新策略,给出 MOEA/D-ENU 的主要流程。

Step1 初始化

Step1.1 初始化种群 $x = \{x^1, \dots, x^N\}$ 和均匀权重向量 $\lambda = \{\lambda_1, \lambda_2, \dots, \lambda_N\}$ 。

Step1.2 计算理想点为 $z = \{z_1, \dots, z_m\}^T$ 。

Step2 进化操作

对于 $i = 1, \dots, N$, 有:

Step2.1 进化。差分进化生成新解 y' , 并更新理想点。

Step2.2 对解的信息进行挖掘。按照3.1节介绍的方法对解的信息进行挖掘,获得新解 y' 类型,以及邻域内的一等劣解解集 IS_1 和二等劣解解集 IS_2 。

Step2.3 基于解挖掘的邻域更新。按照3.2介绍的方法,针对不同类型的新解 y' , 使用与之匹配的邻域更新策略。

Step3 停止迭代

若满足停止条件,则停止迭代并输出最优解集 $'$ 否则继续执行 Step2。

4 实验结果与分析

4.1 测试问题与对比算法

为验证文中提出的基于新型邻域更新策略的 MOEA/D 算法的有效性,选取 CEC09 测试集中的 UF1, UF8, CF2 和 CF9 函数^[14],文献[9]中提到的 F2, F6, F8 函数, ZDT 系列测试函数中的 ZDT4^[15],以及 WFG 系列测试函数中的 WFG9 函数^[16]进行试验。对比算法则选择了以下5种进化算法: MOEA/D^[8], NSGA-II^[17], MOEA/DD^[18], RVEA^[19], IM-MOEA^[20]。

为保证各算法的可比性,实验中对于所有2目标测试问题,设置种群规模均为100,最高评价次数为100000次;对于3目标测试问题,设置种群规模为300,最高评价次数为300000。进化算法交叉率为0.9,变异率为 $1/n$ (n 为决策变量维数),邻域规模 T 为20。

4.2 性能指标

为了评估算法获得解集的收敛性和多样性,本实验将采用广泛用于多目标进化算法评估的超体积 HV、反转世代距

离 IGD 作为性能评估指标。

1)超体积 HV 性能指标:通过计算在集合 P 中被 z^r 支配的解与 z^r 形成的空间大小来综合度量算法的收敛性和解的分布性,其定义如式(6)所示。HV 值越大,说明所得的 Pareto 解集能够越广泛地分布在其前沿面,此时算法的综合性能越好。

$$HV(P, z^r) = Vol(\bigcup_{X \in P} [f_1(X), z_1^r] \times \dots \times [f_m(X), z_m^r]) \quad (6)$$

2)反转世代距离 IGD 性能指标:使用一组真实 Pareto 最优前沿 PF^* , 计算其中每个点到算法获得的解集 PF 之间的最近距离的平均值。其定义如式(7)所示,其中 N^* 是 PF^* 的规模, d_i 是第 i 个真实前沿点到 PF 的最近欧氏距离。IGD 值越低,说明算法获得的近似 Pareto 前端的收敛性和多样性

越好,越接近真实 Pareto 前沿。

$$IGD = \frac{\sum_{i=1}^{N^*} d_i}{N^*} \quad (7)$$

4.3 实验结果与分析

表 1 给出了 MOEA/D, NSGA-II, MOEA/DD, RVEA, IMMOEA 这 5 种进化算法和本文提出的 MOEA/D-ENU 算法在分别独立运行 30 次后,IGD 和 HV 的平均值。图 4 和图 5 给出了这 6 种算法对应的性能指标 IGD 值和 HV 值统计的盒图集合。为了说明实验结果的有效性和实验对比结果的优劣性,本文还使用 Wilcoxon's Sign Rank Test 实验检验方法对各算法的实验数据进行显著性分析。表 1 中,对各算法中显著性较强的数据进行加黑操作,并且在数据后面加了“+”。

表 1 IGD 和 HV 性能指标平均值统计表

Table 1 Statistic results of IGD and HV metric values

性能指标	测试函数	算法					
		MOEAD	MOEADD	NSGA-II	RVEA	IMMOEA	MOEA/D-ENU
IGD	ZDT4	4.15×10^{-3}	4.10×10^{-3}	4.58×10^{-3}	5.40×10^{-3}	4.88×10^{-3}	$3.98 \times 10^{-3} +$
	UF1	1.05×10^{-1}	7.05×10^{-2}	1.11×10^{-1}	1.02×10^{-1}	6.53×10^{-2}	$5.84 \times 10^{-2} +$
	UF8	2.81×10^{-1}	2.29×10^{-1}	2.48×10^{-1}	2.84×10^{-1}	2.08×10^{-1}	$9.04 \times 10^{-2} +$
	CF2	5.25×10^{-2}	4.37×10^{-2}	4.72×10^{-2}	6.23×10^{-2}	4.99×10^{-2}	$1.38 \times 10^{-2} +$
	CF9	4.59×10^{-2}	4.57×10^{-2}	6.97×10^{-2}	1.13×10^{-1}	7.63×10^{-2}	$4.50 \times 10^{-2} +$
	F2	1.11×10^{-1}	7.41×10^{-2}	1.21×10^{-1}	1.02×10^{-1}	6.42×10^{-2}	$6.38 \times 10^{-2} +$
	F6	1.19×10^{-1}	7.76×10^{-2}	9.00×10^{-2}	1.21×10^{-1}	9.12×10^{-2}	$4.50 \times 10^{-2} +$
	F8	2.59×10^{-1}	1.45×10^{-1}	2.00×10^{-1}	2.25×10^{-1}	1.46×10^{-1}	$6.69 \times 10^{-2} +$
	WFG9	3.33×10^{-2}	2.01×10^{-2}	1.99×10^{-2}	2.78×10^{-2}	2.64×10^{-2}	$1.63 \times 10^{-2} +$
HV	ZDT4	8.70×10^{-1}	8.70×10^{-1}	$8.71 \times 10^{-1} +$	8.67×10^{-1}	8.69×10^{-1}	$8.71 \times 10^{-1} +$
	UF1	7.17×10^{-1}	7.44×10^{-1}	7.13×10^{-1}	6.89×10^{-1}	7.65×10^{-1}	$7.80 \times 10^{-1} +$
	UF8	4.12×10^{-1}	4.09×10^{-1}	4.46×10^{-1}	4.23×10^{-1}	4.81×10^{-1}	$6.15 \times 10^{-1} +$
	CF2	7.64×10^{-1}	7.42×10^{-1}	7.42×10^{-1}	7.05×10^{-1}	7.47×10^{-1}	$8.01 \times 10^{-1} +$
	CF9	6.65×10^{-1}	6.53×10^{-1}	5.99×10^{-1}	5.22×10^{-1}	5.69×10^{-1}	$6.70 \times 10^{-1} +$
	F2	6.95×10^{-1}	7.36×10^{-1}	7.00×10^{-1}	7.00×10^{-1}	7.60×10^{-1}	$7.81 \times 10^{-1} +$
	F6	5.90×10^{-1}	6.29×10^{-1}	4.07×10^{-1}	5.79×10^{-1}	6.07×10^{-1}	$7.16 \times 10^{-1} +$
	F8	4.56×10^{-1}	5.94×10^{-1}	5.31×10^{-1}	5.31×10^{-1}	6.20×10^{-1}	$7.54 \times 10^{-1} +$
	WFG9	3.21	3.26	3.27	3.23	3.22	3.28+

由表 1 中各算法的 IGD 指标值可以看出:MOEA/D-ENU 算法在测试函数 UF1, F8, CF2, F2, F6, F8, WFG9 上相较于其他 5 个对比算法有着较为明显的优势,尤其是与对比算法中 MOEA/D 相比,MOEA/D-ENU 算法的 IGD 值要小很多。MOEA/D-ENU 算法在 ZDT4 和 CF9 测试函数上,相比于对比算法 MOEA/D 和 MOEA/DD 仍有着些微优势,相比于另外 3 种对比算法优势比较明显。由表 1 中各算法的 HV 指标值可以看出:MOEA/D-ENU 算法在测试函数 UF1, F8, CF2, F2, F6, F8 上相较于其他 5 个对比算法有着较为明显的优势,但 MOEA/D-ENU 算法在 ZDT4 和 WFG9 测试函数上优势较弱。从上述分析中不难看出,在解决相对复杂的测试问题(如 UF1, UF8, F6, F8)时,MOEA/D-ENU 算法的 IGD 和 HV 指标值优势比较明显,若测试问题比较简单(如 ZDT4, WFG9),MOEA/D-ENU 算法的优势不明显。这说明在解决相对复杂的测试问题时,新型邻域更新策略能够在加快种群收敛速度的同时,很好地兼顾种群的分布性。另外通过 Wilcoxon's Sign Rank Test 检验发现 MOEA/D-ENU 算法的指标值显著性较高。

由图 4 中的盒图可以看出,在 IGD 方面,MOEA/D-ENU 较其他几种算法有着不错的优势,特别是图 4(a)、图 4(c)一图 4(g)所示的盒图,其对应的测试问题都比较复杂,但 MOEA/D-ENU 算法在均值、最小值、中位数上均优于其他算法,这说明新型邻域更新策略对于保持种群多样性有着明显效果。而在处理难度较小的 ZDT4 和 WFG9 的测试问题时,如图 4(h)和图 4(i)所示,MOEA/D-ENU 算法在最小值/中位数只有略微提升。由图 5 的盒图可以看出,在 HV 值方面,MOEA/D-ENU 较其他几种算法总体有着一定的优势,尤其是在解决复杂测试问题(如 UF1, UF8, F6, F8)时优势更明显,其 HV 值远高于其他算法的 HV 值,而在解决其他测试问题时 HV 值也有着些微的优势。从上述盒图分析中可以发现,MOEA/D-ENU 算法的稳定性远远高于其他对比算法,其指标值都集中分布在一个区域。

综上,在本文实验中,提出的 MOEA/D-ENU 算法在解决复杂的优化问题时,既保证了收敛性又很好地兼顾了分布性,而且在处理不同问题的鲁棒性上也明显优于其他几种算法。

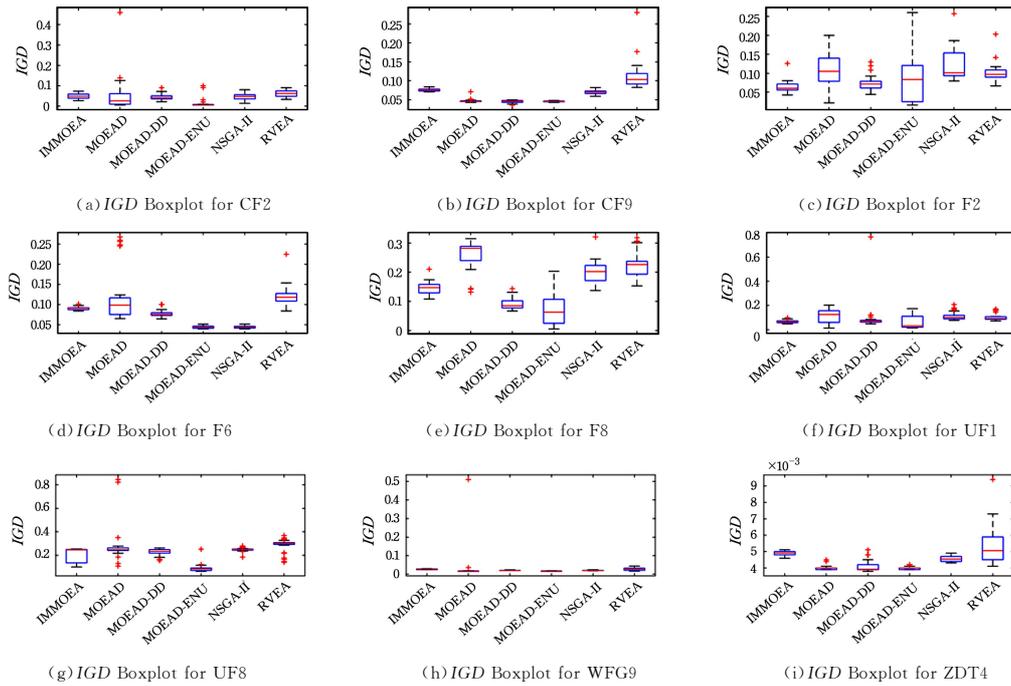


图 4 各算法在不同测试函数上的性能指标 IGD 盒图集合

Fig. 4 Box plots of the IGD metric values obtained by IMMOEA, MOEAD, MOEADD, MOEAD-ENU, NSGA-II and RVEA

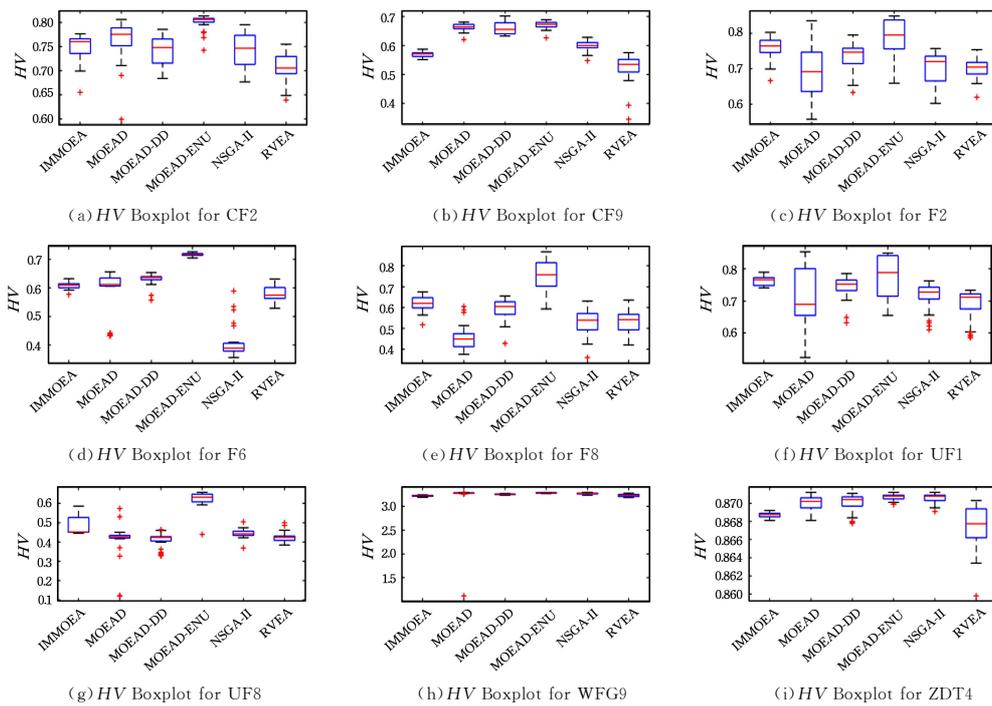


图 5 各算法在不同测试函数上的性能指标 HV 盒图集合

Fig. 5 Box plots of the HV metric values obtained by IMMOEA, MOEAD, MOEADD, MOEAD-ENU, NSGA-II and RVEA

结束语 为了避免经典 MOEA/D 算法邻域更新策略的无限制替换所造成的种群多样性缺失,本文通过在进化过程中充分挖掘解的信息,对新解和对应邻域内的劣解进行了更细致的分类,同时提出了新型邻域更新策略,即针对不同类型的新解,使用不同的更新策略,在加速种群收敛速度的同时,又兼顾种群的多样性,最终形成了一种基于新型邻域更新策略的 MOEA/D 算法。实验结果表明,所提出的算法体现了更优的分布性和收敛性。接下来,将对邻域更新策略扩展,通

过研究算法在不同时期的邻域更新策略的效用,使用更合理的邻域更新策略。

参考文献

[1] SBALZARINI I F, MÜLLER S, KOUMOUTSAKOS P. Multiobjective optimization using evolutionary algorithms [C] // Summer Program. CiteSeer, 2001: 63-74.
 [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, LAUMANN S M, THIELE L. SPEA2: Improving the Strength Pareto Evolutionary Algorithm [J]. IEEE Evolutionary Computation, 2002, 10(3): 263-282.
- [4] JIANG S, ZHANG J, ONG Y S, et al. A Simple and Fast Hypervolume Indicator-Based Multiobjective Evolutionary Algorithm [J]. IEEE Transactions on Cybernetics, 2015, 45(10): 2202-2213.
- [5] BEUME N, NAUJOKS B, EMMERICH M. SMS-EMOA: Multiobjective selection based on dominated hypervolume [J]. European Journal of Operational Research, 2007, 181(3): 1653-1669.
- [6] LI Y L, ZHOU Y R, ZHAN Z H, et al. A primary theoretical study on decomposition-based multiobjective evolutionary algorithms [J]. IEEE Transactions on Evolutionary Computation, 2016, 20(4): 563-576.
- [7] WANG R, ZHANG Q, ZHANG T. Decomposition-based algorithms using pareto adaptive scalarizing methods [J]. IEEE Transactions on Evolutionary Computation, 2016, 20(6): 821-837.
- [8] ZHANG Q, LI H. MOEA/D: A Multiobjective Evolutionary Algorithm Based on Decomposition [J]. IEEE Transactions on Evolutionary Computation, 2007, 11(6): 712-731.
- [9] LI H, ZHANG Q. Multiobjective optimization problems with complicated Pareto sets, MOEA/D and NSGA-II [J]. IEEE Transactions on Evolutionary Computation, 2009, 13(2): 284-302.
- [10] HOU W, DONG H B, YIN G S. Enhanced Multi-objective Evolutionary Algorithm Based on Decomposition [J]. Computer Science, 2014, 41(2): 114-118. (in Chinese)
侯薇,董红斌,印桂生.一种改进的基于分解的多目标进化算法[J].计算机科学,2014,41(2):114-118.
- [11] MA Q. Multi-objective Evolutionary Algorithm Based Weight Vectors Generation Method of MOEA/D [J]. Computer Science, 2016, 43(s2): 117-122. (in Chinese)
马庆.基于多目标进化算法的 MOEA/D 权重向量产生方法[J].计算机科学,2016,43(S2):117-122.
- [12] WANG Z, ZHANG Q, GONG M, et al. A replacement strategy for balancing convergence and diversity in MOEA/D [C] // Evolutionary Computation. IEEE, 2014: 2012-2139.
- [13] WANG Z, ZHANG Q, ZHOU A, et al. Adaptive Replacement Strategies for MOEA/D [J]. IEEE Transactions on Cybernetics, 2017, 46(2): 474-486.
- [14] ZHANG Q, LIU W, LI H. The performance of a new version of MOEA/D on CEC09 unconstrained MOP test instances [C] // IEEE Congress on Evolutionary Computation. IEEE, 2009: 203-208.
- [15] ZITZLER E, DEB K, THIELE L. Comparison of Multiobjective Evolutionary Algorithms: Empirical Results [J]. Evolutionary Computation, 2000, 8(2): 173.
- [16] HUBAND S, HINGSTON P, BARONE L, et al. A review of multiobjective test problems and a scalable test problem toolkit [J]. IEEE Transactions on Evolutionary Computation, 2006, 10(5): 477-506.
- [17] 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.
- [18] LI K, DEB K, ZHANG Q, et al. Combining Dominance and Decomposition in Evolutionary Many-Objective Optimization [J]. International Journal of Behavioral Development, 2006, 30(3): 1.
- [19] CHENG R, JIN Y, OLHOFFER M, et al. A Reference Vector Guided Evolutionary Algorithm for Many-Objective Optimization [J]. IEEE Transactions on Evolutionary Computation, 2016, 20(5): 773-791.
- [20] CHENG R, JIN Y, NARUKAWA K, et al. A Multiobjective Evolutionary Algorithm Using Gaussian Process-Based Inverse Modeling [J]. IEEE Transactions on Evolutionary Computation, 2015, 19(6): 838-856.