

基于混合差分进化和 α 约束支配处理的多目标优化算法

徐 斌, 祁荣宾, 钱 锋

(化工过程先进控制和优化技术教育部重点实验室, 上海 200237; 华东理工大学 信息科学与工程学院, 上海 200237)

摘要: 针对约束多目标优化问题, 提出了一种基于混合差分进化和 α 约束支配处理的优化算法. 算法在用约束水平度对个体满足约束条件的程度进行定量化表达的同时融入支配关系. 在初期放宽约束水平度, 利用不可行解所携带的有用信息, 增加种群多样性, 在后期紧缩约束水平度, 控制不可行解的比例, 朝可行域方向进化. 同时, 将动态单纯形交叉算子和差分进化结合起来构成一种混合差分进化算法, 提高算法的探索 and 开发能力. 对 6 个典型测试函数求解的结果显示, 本文算法无论是在收敛性方面还是解集分散性方面, 与其它算法相比具有很大的优势.

关键词: 差分进化; 多目标; α 约束支配; 单纯形交叉

中图分类号: TP18 **文献标识码:** A

Constrained multi-objective optimization with hybrid differential evolution and α constrained domination technique

XU bin, QI Rong-bin, QIAN Feng

(Key Laboratory of Advanced Control and Optimization for Chemical Processes, Ministry of Education, Shanghai 200237, China;
School of Information Science and Engineering, East China University of Science and Technology, Shanghai 200237, China)

Abstract: To solve the constrained multi-objective optimization problems, we present a hybrid differential evolution algorithm with α constrained domination technique. In this approach, the constraint level, which measures how well an individual satisfies the constraints, is incorporated with the domination principle to solve multi-objective problems. At the early stage, the constraint level is relaxed in order to utilize the useful information carried by some infeasible individuals, so this relaxation increases the diversity of the population. At the later stage, the constraint level is tightened to make the evolution process searching for the feasible area. At the same time, a new dynamic simplex crossover operator is incorporated into differential evolution to improve the abilities of exploration and exploitation. The proposed algorithm is tested on 6 typical benchmarks and compared with other algorithms. Comparison results indicate that the proposed algorithm has advantages in converging to Pareto front and maintaining the evenly-distributed optima along the Pareto front.

Key words: differential evolution; multi-objective; α constrained domination; simplex crossover

1 引言(Introduction)

在科学研究和工程实践中, 很多的具体问题都可以归纳为一个带有约束条件的优化问题. 而这些优化问题往往有多个优化目标, 它们互相联系, 彼此制约, 一个目标的性能优化通常会导致另外至少一个目标性能的退化, 即多个目标很难同时得到最优结果^[1]. 因此, 研究多目标优化问题特别是有约束条件的多目标优化问题是非常有必要的. 进化算法(evolutionary algorithm, EA)在处理多目标优化问题已经得到了较好的应用, 如 NSGA-II(nondominated sorting genetic algorithms-II)算法^[2], SPEA(strength pareto evolutionary algorithm)算法等^[3].

差分进化(differential evolution, DE)是一种基于

种群的全局优化进化算法^[4], 具有容易理解、收敛速度快、鲁棒性高以及可调参数少等优秀特性. 许多学者将DE算法进行扩展并用它来处理多目标优化问题, 得到很好的结果^[5]. 约束优化算法通常是在无约束优化算法的基础之上增加约束处理技术来实现的^[6]. 常用的约束处理技术有死亡罚函数法、罚函数法以及多目标法等^[7-10]. 但是这些方法存在一些不足之处. 如死亡罚函数法, 它不考虑不可行解, 没有充分利用一些不可行解所携带的重要信息. 罚函数法通过将约束优化问题转化为一个无约束优化问题的方式来挖掘不可行解携带的信息, 但是设计一个比较适合的、通用的罚系数是比较困难的. 多目标法将个体违反约束条件的程度看成是优化目标,

收稿日期: 2010-10-25; 收修改稿日期: 2011-05-03.

基金项目: 国家自然科学基金重点基金资助项目(U1162202); 国家重点基础研究发展计划资助项目(2009CB320603); 中央高校基本科研业务费专项资金资助项目; 高等学校博士学科点专项科研基金新教师基金资助项目(200802511011); 上海市科技攻关资助项目(09DZ1120400); 上海市基础研究重点资助项目(10JC1403500); 上海市重点学科建设资助项目(B504).

但是这样处理会增加目标函数的个数,增加计算量。

为了更好的调节算法在可行域和不可行域中的搜索,同时提高算法的探索和开发能力,防止陷入局部最优或早熟收敛,本文结合 α 约束处理技术和支配关系,提出一种 α 约束支配方法(alpha constrained domination technique, α CD),并将它融合到混合DE算法中,提出一种新的基于混合DE和 α CD的多目标优化算法(简称为HDE- α CD),用于处理带有约束条件的多目标优化问题。

2 Alpha约束支配处理和混合差分进化(α CD and hybrid DE)

不失一般性,一个带约束条件的多目标优化问题可以表示为

$$\begin{cases} \min_{x \in S} F(x) = [f_1(x) \ f_2(x) \ \cdots \ f_m(x)]^T, \\ \text{s.t. } G_i(x) \leq 0 (i = 1, 2, \cdots, p), \\ H_j(x) = 0 (j = 1, 2, \cdots, q), \end{cases} \quad (1)$$

这里 m 表示需要被同时优化的目标问题个数, $x = [x_1 \ x_2 \ \cdots \ x_n]$ 是 n 维决策变量. S 是搜索空间,它由决策变量的上边界约束 $x^{\max}$ 和下边界约束 $x^{\min}$ 决定, $G(x)$ 和 $H(x)$ 分别是 p 个和 q 个线性或非线性的不等式和等式约束,它们决定了搜索空间的可行域 Ω , 一般有 $\Omega \subseteq S$.

2.1 处理约束多目标问题的 α 约束方法(alpha constrained technique for constrained multi-objective problems)

借鉴模糊逻辑中隶属度的概念, α 约束方法对种群中的每个个体 x 都定义了一个衡量满足约束条件程度的量——约束水平度 $\mu(x)$ ^[7]. 如果 $\mu(x) = 1$, 表明 x 属于可行解, 如果 $\mu(x) \neq 1$, 则 x 不属于可行解, 且 $\mu(x)$ 越大, 其隶属于可行解的程度就越大. 为了计算 $\mu(x)$, 引入一个正常数 b , 将 x 的每一个约束条件转化为一个处于 $[0, 1]$ 之间的模糊量, 然后根据最小隶属度取小值的原则计算 $\mu(x)$. 这里参数 b 通常被限制在一个约定范围, 且 b 的大小与求解的问题有关.

根据模糊逻辑中水平截集的概念和多目标中的支配关系, 提出一种新的 α 约束支配概念. 假设 F_1 , F_2 和 μ_1 , μ_2 分别是个体 x_1 与 x_2 对应的目标函数值和约束水平度. 对于 $0 \leq \alpha \leq 1$, 可以定义这两个个体的 α 约束支配关系(表示为 $\succ_{\alpha\text{CD}}$):

$$x_1 \succ_{\alpha\text{CD}} x_2 \Leftrightarrow \begin{cases} F_1 \succ F_2, \mu_1, \mu_2 \geq \alpha, \\ F_1 \succ F_2, \mu_1 = \mu_2, \\ \mu_1 \geq \mu_2, \text{ 其他.} \end{cases} \quad (2)$$

可以看出, 如果两个个体满足 α 约束, 则它们的支配关系仅由目标函数值来决定, 相反, 则是由约束水平度的大小来决定的. 特别的, 如果 $\alpha = 0$, 则该方法退

化为无约束优化问题, 即不考虑约束条件. 如果 $\alpha = 1$, 则可以认为是和Deb提出的约束条件处理方法是相似的^[2].

假设在初始阶段 $\alpha(0) = \alpha_0$, 则在 t 时刻 α 的取值可以表示为

$$\alpha(t) = (1 - \beta)\alpha(t-1) + \beta, \quad (3)$$

其中 β 是 α 的控制参数, 根据文献[7], 本文中取 $\beta = 0.03$.

2.2 混合差分进化算子(Hybrid differential evolution operation)

对存档集的个体进行拥挤度评价^[2], 以竞争选择的方式选择具有较大拥挤度的个体(这里表示为 best)参与变异操作:

$$u_i^t = x_{r_1} + SF_1(\text{best} - x_{r_1}) + SF_2(x_{r_2} - x_{r_3}), \quad (4)$$

这里 u_i^t 是变异后的个体, x_{r_1} , x_{r_2} , x_{r_3} 是从当前种群中随机选取出来的3个互异个体. 变异缩放因子 SF 是一个在0和1之间的数. 变异后的个体经过交叉操作之后产生子代个体^[4].

单纯形交叉算子是一个多父代组合操作^[11]. 在搜索空间 $\mathbb{R}^n$ 中, 有 $n+1$ 个独立的个体形成一个单纯形. 设有扩张系数 $1 + \varepsilon$, 则 $n+1$ 个个体经过扩张后可形成一个新的单纯形, 然后可以在新的单纯形中随机选择一个个体作为子代个体输出. 在单纯形交叉过程当中, 扩张系数决定了新单纯形的大小. 可以分成两种情况:

1) 扩张阶段: 如果扩张系数 $1 + \varepsilon$ 大于1, 即 $\varepsilon > 0$ 时, 新的单纯形(由 y 构成)在原来单纯形的外面, 总体来说是外延的, 这时该交叉算子有更好的探索能力, 如图1左图所示.

2) 收缩阶段: 如果扩张系数 $1 + \varepsilon$ 小于1, 即 $\varepsilon < 0$ 时, 新的单纯形(由 y 构成)在原来单纯形的里面, 总体来说是内敛的, 这时该交叉算子有更好的开发能力. 如图1右图所示.

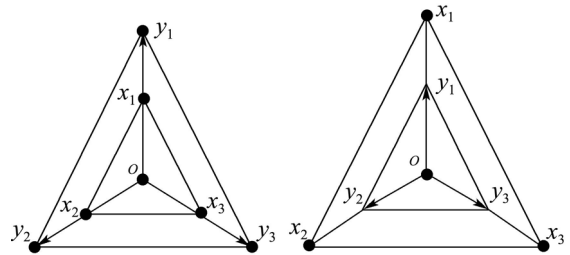


图1 2维情况3个体单纯形交叉算子示例(左图为扩张阶段, 右图为收缩阶段)

Fig. 1 Illustration of simplex crossover operator with three-parent in two-dimensional space

通过上面的分析, 可以得到这样的结论: 单纯形交叉算子可以很好地对探索和开发能力进行平

衡. 在早期, 可以设定较大的扩张系数, 这样的话就可以使得算法在更广阔的区域进行搜索; 在后期, 可以设定较小的扩张系数, 使得算法具有较好的开发能力, 提高解的质量. 为此, 本文中设计了一种动态时变的扩张系数, 假设在 t 时刻, 有

$$\varepsilon(t) = \frac{t-1}{T-1}(\varepsilon_T - \varepsilon_0) + \varepsilon_0, \quad (5)$$

这里 T 代表总的进化代数, ε_0 和 ε_T 分别是初始和最终的扩张系数. 显然, ε 从 ε_0 逐渐线性减少到 ε_T , 也从探索阶段慢慢过渡到开发阶段.

2.3 多目标选择和存档集更新操作(Selection operator and archive updating method for MOPs)

基于 α CD约束支配关系和拥挤度评价方式对父代个体和子代个体进行选择算法. 其实现过程如下: 首先把子代和父代个体的集合分成满足 α 约束的子集 X_f 和不满足 α 约束的子集 X_{inf} , 对于 X_f 中的个体依据支配关系进行非支配排序分级, 且对同一等级的个体进行拥挤度的评价并排序^[2]. 对 X_{inf} 中的个体依据个体的约束水平度进行排序. 然后从 X_f 中依次选择最好等级的个体加入到下一代种群. 如果加入某一等级的所有个体超过预定的种群大小, 则依次选择该等级个体中拥挤度较大的个体, 直至填满下一代种群. 如果 X_f 中的个体数目比下一代种群规模要小, 则需从 X_{inf} 中依次选择约束水平度较大的个体加入到子代种群, 直到达到预定的种群大小.

此外, 用 X_f 中的最优个体(这里表示为 U)对最优存档集 ACH 进行更新. 对于 U 中的每个个体 u_i , 如果 u_i 被 ACH 中的某个个体 α 约束支配, 则直接舍弃 u_i ; 如果 u_i 能 α 约束支配 ACH 中的部分个体(这里表示为 $D(u_i)$), 则首先令 $ACH = ACH \setminus D(u_i)$, 然后取 $ACH = ACH \cup u_i$; 如果 ACH 和 u_i 是 α 约束非支配的, 则 $ACH = ACH \cup u_i$. 然后考察当前存档集的规模, 如果超过预定的大小, 则进行拥挤度评价, 舍弃拥挤度较小的个体以维持 ACH 的规模不会超过预定大小.

2.4 本文算法的流程(The flowchart of proposed algorithm)

综合上面所述, 基于混合DE和 α CD处理的多目标优化算法的步骤可以表示如下:

Step 1 初始化. 令进化代数 $t = 0$, 存档集 $ACH = \emptyset$, 随机初始化初始种群 X^0 , 评价初始种群中所有个体的目标值 F 和约束值 $[G \ H]^T$, 根据约束值先计算 b , 然后计算种群中每个个体的约束水平度 μ . 将满足 α 约束的非支配最优个体加入到存档集中;

Step 2 生成子代个体. 对于当前种群 X^t 中的每个个体, 以 $p = 0.5$ 的选中概率从种群中选出, 并对它实行DE变异操作和交叉操作, 生成新的子代种

群 Q_1 . 对于其它未被选中的个体, 实行动态单纯形交叉操作, 生成新的子代种群 Q_2 ;

Step 3 评价. 评价种群 Q_1 和 Q_2 的目标函数值 F 以及约束值 $[G \ H]^T$, 并计算每个个体的约束水平度 μ ;

Step 4 选择. 按照第2.3节中选择方法形成下一代种群 X^{t+1} , 同时更新最优存档集 ACH ;

Step 5 令 $t = t + 1$;

Step 6 如果满足停止条件, 则算法停止, 并且输出存档集中的个体为最终的Pareto最优解, 否则返回Step 2.

3 仿真实验设计(Experimental study)

3.1 测试函数(Test functions)

为验证算法的性能, 选取6个优化问题作为测试函数, 即TP1^[12], TNK^[12], CTP1^[13], CTP2^[13], CTP4^[13]和CTP6^[13]. 对CTP系列函数, 取 $0 \leq x_i \leq 1$, $i = 1, \dots, 10$, $g(x) = 1 + 9 \sum_{i=1}^n \frac{x_i}{n-1}$. 这里CTP1和CTP6有连续的最优边界, 而其它函数的最优边界是离散的.

3.2 性能评价指标(Performance assessment metric)

针对多目标优化问题, 用超体积法(hypervolume indicator, I_H)来评价解集逼近真实解的程度^[14], 用分散度指标法(diversity metric, I_{DM})来评价解集的分布状况^[2]. 超体积表示近似最优解集在目标空间相对于参考点(reference point)所支配的空间的比例^[14], 该指标越大, 该解集越接近真实最优解, 收敛性越好. 对于参考点的选取, 则选择所有目标函数在每一目标分量上的最大值所组成的序列作为参考点. 分散度指标法则能体现近似最优解集的分布情况, 该指标越小, 则表明解集的分布性越均匀.

3.3 仿真结果(Experimental results)

为了验证本文算法的有效性, 与NSGA-II^[2], Woldesenbet提出的方法(表示为Woldesenbet)^[15]、Ray提出的方法(表示为Ray)^[16]进行比较. 对于本文的算法, 设定种群大小 $N = 100$, 存档集 $NA = 100$, SF 取 $[0, 1]$ 之间的随机数, 交叉参数 $CR = 0.8$, $\alpha_0 = 0.85$, 对于 b 值, 设定取值范围为 $[0.01, 1e+4]$, 即当初始种群中不可行解对应的约束条件的平均值在取值范围之外时, 可取值为该范围的边界值. 动态单纯形交叉算子中扩张系数 $\varepsilon_0 = 10$, $\varepsilon_T = -0.5$. 另外, NSGA-II, Woldesenbet和Ray算法的参数设置与对应文献保持一致. 4种算法的停止条件是目标函数评价次数为25000次. 对于每个算法分别独立运行10次, 然后对所得数据进行统计分析.

图2给出了6个测试函数分别运用4种算法运算后得到的Pareto最优边界。图中左边第1列是采用HDE- α CD, 第2列是采用NSGA-II方法, 第3列是采用Woldesenbet的方法, 第4列是采用Ray算法。从上到下依次是TP1, TNK, CTP1, CTP2, CTP4, CTP6系列函数。从图中可以看出, HDE- α CD算法对于所有的测试函数, 都能在一定误差范围内逼近最优边界, 且最优解的分散性比较好。NSGA-II算法对CTP4的求解能力比较弱, 虽然

最优解能很好的逼近最优边界, 但是, 这些最优解没有很好的分散性, 即不能最大程度的沿着整个最优边界均匀分布, 而是出现了大量的解聚集到某一个很小的区域范围之内。Woldesenbet算法对于CTP4求解能力表现不足, 最优解不能沿着整个最优边界分布, 即只能求解得到一部分最优边界。Ray算法除了对CTP6有较高的收敛性外, 对于其他的几个问题都不能很好的收敛到最优边界。

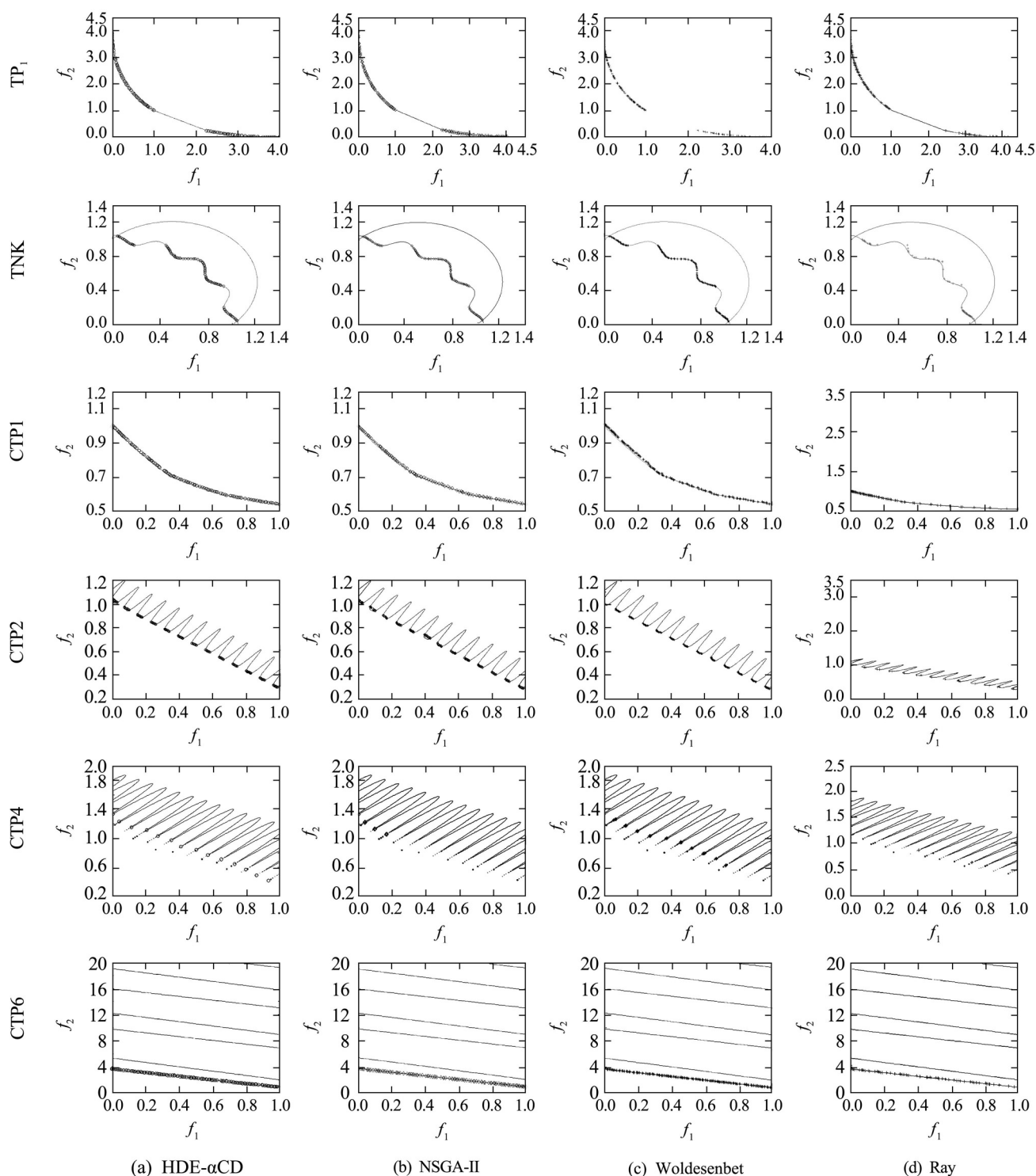


图2 4种算法求解6个测试函数所得最优边界

Fig. 2 The obtained Pareto fronts of 6 test functions by 4 algorithms

为了定量的分析4种算法的性能,表1给出了4种算法6个测试函数对于 I_H 指标的平均值和标准方差.从表1中的数据可以看出(黑色粗体代表最好统计值),本文算法所得到的结果对于其他3种算法来说,平均值较大,这也就说明了本文算法在收敛性方面比其它几种算法要好,另外,对应的标准方差也较小,即算法比较稳定,鲁棒性高.

图3展示了10次运行结果的 I_H 指标盒须图(1代表HDE- α CD, 2代表NSGA-II, 3代表Wolfsenbet, 4代表Ray),它能直观明了地识别数据中的异常

值,也能判断数据的偏态情况.可以看出,HDE- α CD对于大多数的测试问题,不仅可以得到较好的平均值,得到的最小值也比对应的方法得到的最小值要好,特别是对于CTP4问题,HDE- α CD得到的数据中的最小值都明显好于其他算法的到的数据的最小值.另外,HDE- α CD得到的统计数据的异常值都比较少,表明算法鲁棒性较强,虽然对于TP1问题,出现了部分异常值,但是这些异常值都比其他数据值要大,即是一些表现比较优良的异常值.

表1 4种算法对6个测试函数分别进行10次独立运行的 I_H 指标统计分析值
Table 1 Statistical results of the I_H indicator based on 10 independent runs by 4 algorithms in solving the 6 test problems

测试函数	I_H			
	HDE- α CD	NSGA-II	Wolfsenbet	Ray
TP1	0.8396$\pm$0.0010	0.8391 $\pm$ 0.0012	0.8392 $\pm$ 0.8316	0.8316 $\pm$ 0.0039
TNK	0.5886$\pm$0.0015	0.5782 $\pm$ 0.0313	0.5550 $\pm$ 0.0884	0.5757 $\pm$ 0.0038
CTP1	0.4436$\pm$0.0014	0.4413 $\pm$ 0.0029	0.4388 $\pm$ 0.0011	0.4377 $\pm$ 0.0029
CTP2	0.6725$\pm$0.0017	0.6706 $\pm$ 0.0021	0.6702 $\pm$ 0.0064	0.6654 $\pm$ 0.0019
CTP4	0.5784$\pm$0.0122	0.5143 $\pm$ 0.0422	0.5335 $\pm$ 0.0185	0.5205 $\pm$ 0.0148
CTP6	0.5408$\pm$0.0049	0.5384 $\pm$ 0.0050	0.5374 $\pm$ 0.0039	0.5339 $\pm$ 0.0054

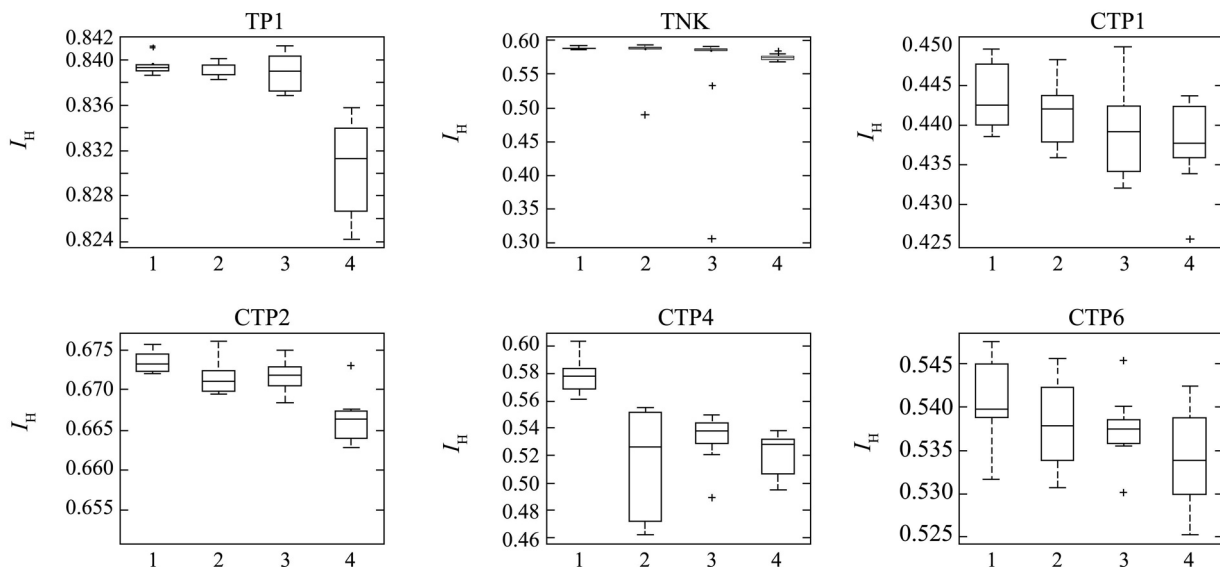


图3 4种算法对6个测试函数分别进行10次独立运行结果的 I_H 指标的盒须图

Fig. 3 Box plots of the I_H indicator based on 10 independent runs by 4 algorithms in solving the 6 test problems

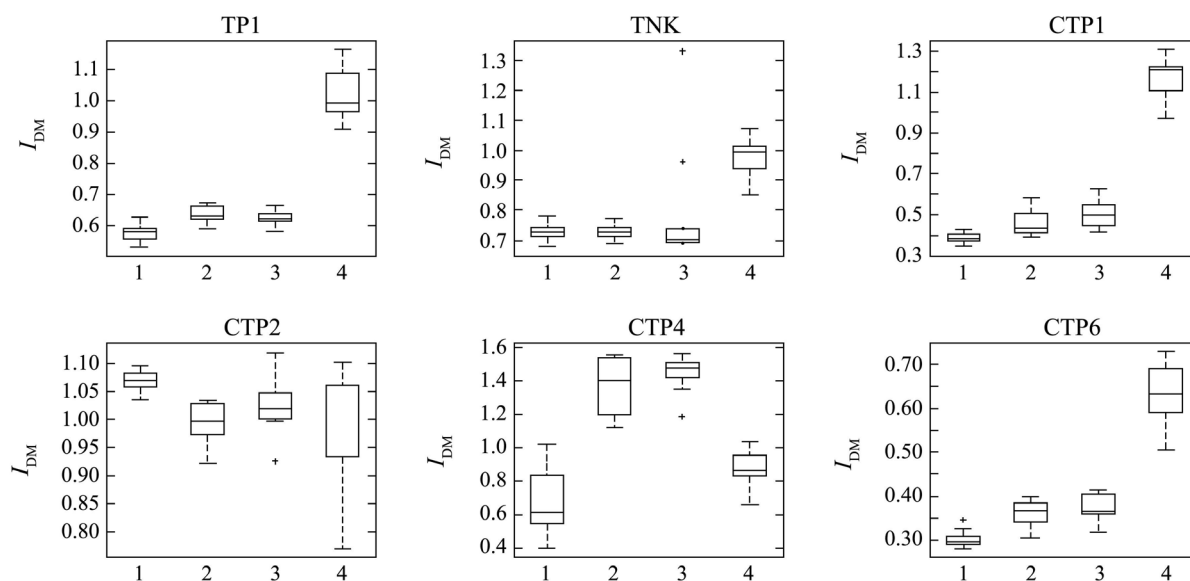
表2给出了4种算法运行10次得到的 I_{DM} 指标的平均值和标准方差.从表格中的数据可以看出,HDE- α CD求解TP1, CTP1, CTP4和CTP6时有最小的平均值,也就表明通过HDE- α CD求解得到的最优Pareto最优解沿着整个Pareto边界分布最均匀,比其它方法要好.另外采用HDE- α CD方法得到的结果的标准方差都相对要小,也就是该算法比较

稳定.

图4显示 I_{DM} 指标的盒须图(1代表HDE- α CD, 2代表NSGA-II, 3代表Wolfsenbet, 4代表Ray).从图中可看出,HDE- α CD对于CTP2测试函数表现不足,而对其他几个测试问题,具有很大的优势,特别是TP1, TNK, CTP1及CTP6都得到比较小的分散度指标值,且统计数据的方差也很小、稳定性高.

表2 4种算法对6个测试函数分别进行10次独立运行结果的 I_{DM} 指标统计分析值Table 2 Statistical results of the I_{DM} indicator based on 10 independent runs by 4 algorithms in solving the 6 test problems

测试函数	I_{DM}			
	HDE- α CD	NSGA-II	Wolfsenbet	Ray
TP1	0.5810 ± 0.0269	0.6348 ± 0.0290	0.6227 ± 0.0265	1.0157 ± 0.0812
TNK	0.7292 ± 0.0275	0.7275 ± 0.0243	0.7938 ± 0.2062	0.9806 ± 0.0659
CTP1	0.3838 ± 0.0259	0.4581 ± 0.0703	0.5032 ± 0.0660	1.1761 ± 0.1000
CTP2	1.0693 ± 0.0192	0.9927 ± 0.0353	1.0230 ± 0.0499	0.9839 ± 0.1092
CTP4	0.6654 ± 0.2021	1.3693 ± 0.1737	1.4405 ± 0.1082	0.8666 ± 0.1277
CTP6	0.3033 ± 0.0193	0.3592 ± 0.0305	0.3756 ± 0.0307	0.6326 ± 0.0681

图4 4种算法对6个测试函数分别进行10次独立运行的 I_{DM} 指标的盒须图Fig. 4 Box plots of the I_{DM} indicator based on 10 independent runs by 4 algorithms in solving the 6 test problems

3.4 Alpha约束支配处理技术有效性的分析 (Effectiveness study of the proposed α CD)

为了分析 α CD的有效性,以TP1为例进行分析. TP1的最优边界的表达式为

$$f_2 = (\sqrt{f_1} - 2)^2,$$

其中: $0 \leq f_1 \leq 1$, $2.25 \leq f_1 \leq 4$, 试验中每隔5代保存存档集中的解并对这些解进行分析. 图5给出了前30代运行过程中存档集中解集对应的最优边界. 图中黑色实线代表真实可行区域与不可行区域的分界边界.

$t = 5$ 时,由于 α 比较小,故存档集中既包含了可行解也包含了部分满足 α 约束关系的不可行解. 所以此时的最优边界由两部分组成:一部分是位

于可行域的,一部分是位于不可行域的. 到 $t = 10$ 时,最优边界仍由位于可行域部分和位于不可行域部分组成,但是,此时位于不可行域部分的解的数量变得更少,出现了缺口,使得原本是一条连续的最优边界变得不太连续. 到 $t = 15$ 时,原来连续的最优边界被彻底分成了两部分,中间的不连续部分变得更大,也就是说,存档集中的不可行解变得越来越少,且这些不可行解都是违反约束条件比较小的,是比较靠近可行解的. 到 $t = 20$ 以及以后的进化过程中,存档集中不可行解基本已经很少了,此时的最优边界接近真实最优边界. 可以看出, α CD通过调整约束水平截集参数来挖掘一些不可行解所携带的信息,引导算法朝着真实的最优边界进化,最终收敛到真实最优边界.

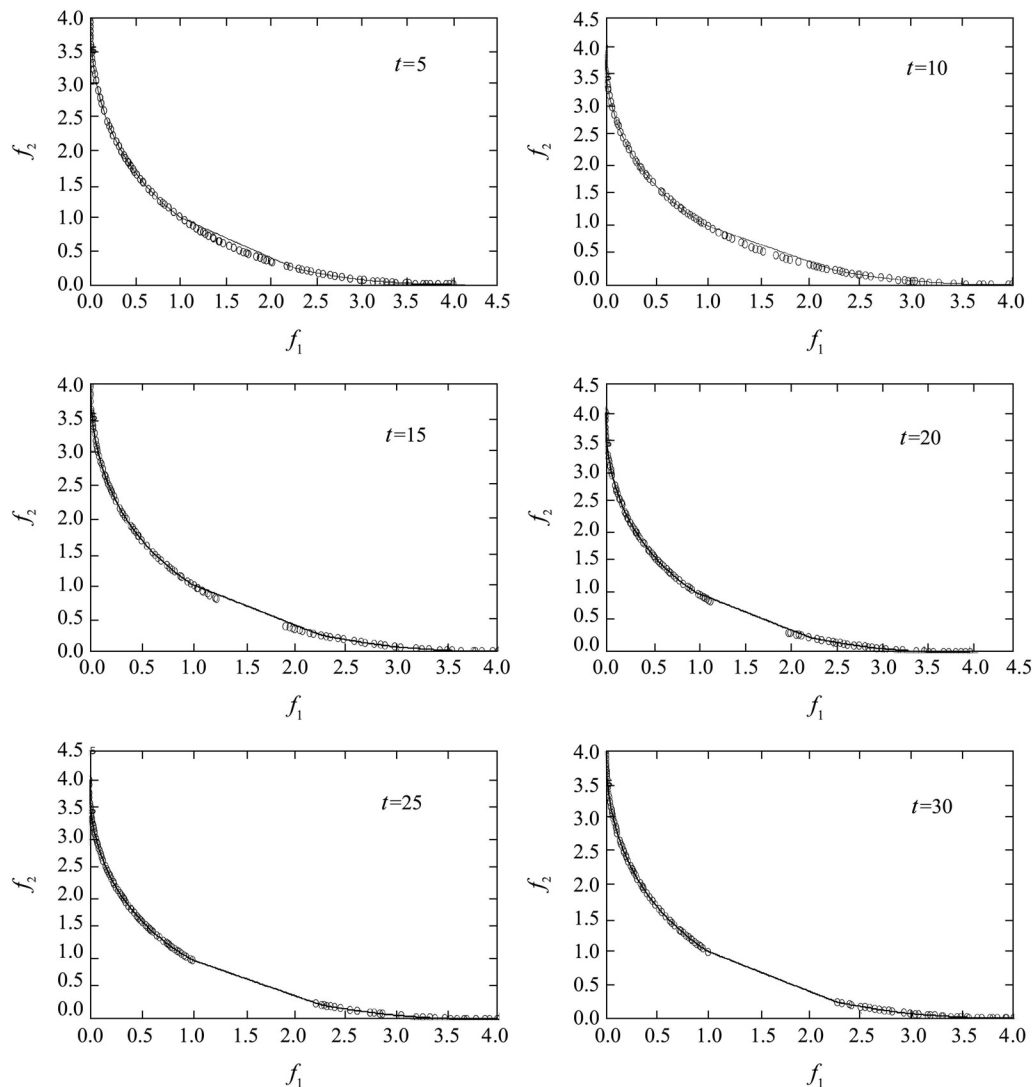


图5 不同进化时刻TP1函数的最优边界图

Fig. 5 The optimal Pareto front of TP1 at different time

4 结论(Conclusion)

针对多目标优化问题的约束条件,提出了一种alpha约束支配处理技术,该方法对每个个体定义一个约束水平度,在进化过程中通过有效调约束水平度来挖掘不可行解所携带的信息,引导算法朝着可行域收敛.针对多目标问题,对基本差分进化算法的变异和选择操作进行修改的同时,融入了一种动态单纯形交叉算子,形成一种混合差分进化算法,提高了算法的探索 and 开发能力.与其他3种多目标约束优化算法进行比较,结果显示,在收敛性方面,本文的算法明显优于其他算法,在分散性方面,本文算法具有一定的优势,是一种处理约束多目标优化问题的有效方法.

参考文献(References):

- [1] COELLO COELLO C A. Evolutionary multi-objective optimization: a historical view of the field[J]. *IEEE Computational Intelligence Magazine*, 2006, 1(1): 28 – 36.
- [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, 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.
- [4] STORN R, PRICE K. Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces[J]. *Journal of Global Optimization*, 1997, 11(4): 341 – 359.
- [5] EFREN M M, MARGARITA R S, COELLO C A. Multi-objective optimization using differential evolution: a survey of the state-of-the-art[M] //Uday K. Chakraborty. *Advances in Differential Evolution*. Berlin: Springer. Berlin: Springer, 2008: 173 – 196.
- [6] COELLO COELLO C A. Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: a survey of the state of the art[J]. *Computer Methods in Applied Mechanics and Engineering*, 2002, 191(11/12): 1245 – 1287.
- [7] TAKAHAMA T, SAKAI S. Constrained optimization by applying the alpha constrained method to the nonlinear simplex method with mutations[J]. *IEEE Transactions on Evolutionary Computation*, 2005, 9(5): 437 – 451.

- [8] WANG Y, CAI Z X, ZHOU Y R, et al. An adaptive tradeoff model for constrained evolutionary optimization[J]. *IEEE Transactions on Evolutionary Computation*, 2008, 12(1): 80 – 92.
- [9] VIEIRA D A G, ADRIANO R L S, VASCONCELOS J A, et al. Treating constraints as objectives in multiobjective optimization problems using niched pareto genetic algorithm[J]. *IEEE Transactions on Magnetics*, 2004, 40(2): 1188 – 1191.
- [10] RUNARSSON T P, YAO X. Search biases in constrained evolutionary optimization[J]. *IEEE Transactions on Systems Man and Cybernetics, Part C—Applications and Reviews*, 2005, 35(2): 233 – 243.
- [11] SHIGEYOSHI T, MASAYUKI Y, TAKAHIDE H. Multi-parent recombination with simplex crossover in real coded genetic algorithms[C] // *Proceedings of the Genetic and Evolutionary Computation Conference*. San Francisco: Morgan Kaufmann, 1999: 395 – 413.
- [12] 王跃宣, 刘连臣, 牟盛静, 等. 处理带约束的多目标优化进化算法[J]. 清华大学学报(自然科学版), 2005, 45(1): 103 – 106. (WANG Yuexuan, LIU Liancheng, MU Shengjing, et al. Constrained multi-objective optimization evolutionary algorithm[J]. *Journal of Tsinghua University(Science and Technology)*, 2005, 45(1): 103 – 106.)
- [13] DEB K, PRATAP A, MEYARIVAN T. Constrained test problems for multi-objective evolutionary optimization[C] // *The 1st International Conference on Evolutionary Multi-Criterion Optimization*. Berlin: Springer-Verlag, 2001: 284 – 298.
- [14] ZITZLER E, THIELE L, LAUMANN M, et al. Performance assessment of multiobjective optimizers: an analysis and review[J]. *IEEE Transactions on Evolutionary Computation*, 2003, 7(2): 117 – 132.
- [15] WOLDESENBET Y G, YEN G G, TESSEMA B G. Constraint handling in multiobjective evolutionary optimization[J]. *IEEE Transactions on Evolutionary Computation*, 2009, 13(3): 514 – 525.
- [16] RAY T, TAI K, SEOW K C. Multiobjective design optimization by an evolutionary algorithm[J]. *Engineering Optimization*, 2001, 33(4): 399 – 424.

作者简介:

徐 斌 (1984—), 男, 博士研究生, 主要从事智能计算、多目标优化的研究, E-mail: xubin@mail.ecust.edu.cn;

祁荣宾 (1974—), 女, 副研究员, 主要从事智能优化、多目标优化的研究, E-mail: qirongbin@ecust.edu.cn;

钱 锋 (1961—), 男, 教授, 博士生导师, 主要从事复杂工业过程的建模、控制与优化的研究, E-mail: fqian@ecust.edu.cn.