

文章编号: 1001-0920(2016)01-0045-07

DOI: 10.13195/j.kzyjc.2014.1885

基于环境 Pareto 支配选择策略的有约束多目标差分进化算法

孙浩¹, 杨景明¹, 刘醒², 唐永超³, 车海军^{1,4}

(1. 燕山大学 工业计算机控制工程河北省重点实验室, 河北 秦皇岛 066004; 2. 首钢京唐钢铁联合有限责任公司, 河北 唐山 063200; 3. 秦皇岛凯鸿科技有限公司, 河北 秦皇岛 066004; 4. 国家冷轧板带装备及工艺工程技术研究中心, 河北 秦皇岛 066004)

摘要: 在处理有约束多目标问题的进化算法中, 目前普遍采用 Deb 教授提出的约束占优的直接支配选择策略. 在约束处理中, 优秀不可行解与优秀可行解同样重要, 但在直接支配选择策略中, 不可行解被选择的几率很小. 针对此问题, 设计一种环境 Pareto 支配的选择策略, 并基于此提出用于解决有约束多目标问题的差分进化算法. 对经典测试函数进行仿真计算, 结果表明, 与其他算法相比, 所提出的算法具有更高的收敛性和稳定性.

关键词: 约束多目标; 选择策略; 差分进化

中图分类号: TP18

文献标志码: A

Differential evolution algorithm based on environment Pareto dominated selection strategy in constrained multi-objective optimization problem

SUN Hao¹, YANG Jing-ming¹, LIU Xing², TANG Yong-chao³, CHE Hai-jun^{1,4}

(1. Key Laboratory of Industrial Computer Control Engineering of Hebei Province, Yanshan University, Qinhuangdao 066004, China; 2. Shougang Jingtang United Iron and Steel Co. Ltd, Tangshan 063200, China; 3. Qinhuangdao Kai Hong Technology Co. Ltd, Qinhuangdao 066004, China; 4. National Engineering Research Center for Equipment and Technology of C.S.R, Qinhuangdao 066004, China. Correspondent: SUN Hao, E-mail: sunhaoyan603@163.com)

Abstract: In the evolutionary algorithm with constrained multi-objective problems, the selection strategy of constrained-dominate proposed by Professor Deb is widely used. The excellent infeasible solution is equally important as the feasible solution in the constraint treatment method. The infeasible solution has a small chance of winning in the selection strategy of constrained-dominate. Therefore, a differential evolution algorithm based on the environment Pareto dominated selection strategy in the constrained multi-objective optimization problem is proposed. Benchmark functions are simulated, and the results show that, compared with other algorithms, the proposed algorithm has better convergence and stability.

Keywords: constrained multi-objective optimization; selection strategy; differential evolution

0 引言

在人工智能领域中, 基于达尔文的进化理论, 涌现了很多优秀的智能算法, 其中差分进化算法以其较少的设定参数和良好的寻优能力得到很多人的青睐. 国内外对无约束多目标问题的研究成果很多, 但在实际的科研与工程当中, 约束条件必不可少. 因此, 对于多目标优化问题, 约束处理技术的研究至关重要^[1-6].

文献 [7] 对当前基于进化算法的约束处理技术进行了较为详尽的概括. 对于多目标约束处理技术大致分为 3 类: 基于罚函数的约束处理技术、基于多目标

算法的约束处理技术和基于排序的约束处理技术. 罚函数法主要是通过对各目标函数增加惩罚因子构造惩罚适应值函数, 将有约束多目标优化问题转换为无约束多目标优化问题, 虽然罚函数法构造简单易于实现, 但合理的惩罚系数难以确定; 基于多目标的约束处理技术是将约束条件当作额外的目标函数来处理, 但是随着目标函数与约束条件的增加, 综合目标函数的维数变得更高, 求解难度进一步增加; 基于排序的方法不进行无约束化处理, 而是通过对目标函数和约束条件的综合考虑对候选解进行对比, 选择出更加适

收稿日期: 2014-12-11; 修回日期: 2015-04-05.

基金项目: 国家自然科学基金委员会与宝钢集团有限公司联合项目(U1260203); 河北省高等学校创新团队领军人才培养计划项目(LJRC013); 国家冷轧板带装备及工艺工程技术研究中心开放课题(2012005).

作者简介: 孙浩(1985-), 男, 博士生, 从事智能优化算法及冶金自动化的研究; 杨景明(1957-), 男, 教授, 博士生导师, 从事冶金机械综合自动化及人工智能等研究.

合的最优解^[8-10].

进化算法中的选择策略是算法前进的动力,在有约束多目标问题中,大多数采用 Deb 提出的约束占优直接支配选择策略^[11],这种方法过于偏重可行解的开发,在计算过程中不利于不可行解的保留,甚至在进化后期不可行解将被全部淘汰.然而,对于有约束多目标优化问题,优秀不可行解同样重要.因此,为了有效利用这部分优秀不可行解,首先要增加优秀不可行解被选择的概率.

1 有约束多目标优化问题模型

以最小多目标优化问题为例,多目标有约束优化问题(MOCOP)可以定义为^[12]

$$\min_{x \in R^n} F(x) = (f_1(x), f_2(x), \dots, f_k(x)).$$

$$\text{s.t. } g_i(x) \leq 0, i = 1, 2, \dots, p;$$

$$h_j(x) = 0, j = 1, 2, \dots, q. \quad (1)$$

其中: $F(x)$ 为目标函数, $g_i(x)$ 和 $h_j(x)$ 分别为不等式约束和等式约束条件, $x = (x_1, x_2, \dots, x_n) \in R^n$ 为 n 维决策向量. 将满足所有约束条件的解空间 S 称为式(1)的可行域. 对于等式约束 $h_j(x) = 0$, 可以通过设定容许误差 ($\delta > 0$), 将它转变为一个不等式约束

$$|h_j(x)| \leq \delta, \quad (2)$$

故在以后讨论的问题中,仅考虑带有不等式约束的优化问题.

2 基于环境 Pareto 支配的选择策略

在有约束多目标问题中,目前应用最广泛的是 Deb 等^[13]提出的约束占优直接支配选择策略:

1) 两个个体相比较,如果一个个体为可行解,而另一个个体为不可行解,则选择可行解作为较优个体;

2) 两个个体相比较,当两个个体均为可行解时,选择非支配个体作为较优解,如果两个个体互不支配,则选择拥挤度较大的个体作为较优解;

3) 两个个体相比较,如果两个个体均为不可行解,则选择违反约束程度较小的个体作为较优个体.

通过以上描述可知,3个规则最终的选择结果更加倾向于可行解.当可行解与不可行解相比较时,可行解均会胜出,在有约束多目标问题中,判断解的优劣不应只以约束作为唯一标准,应结合更多的环境信息综合比较.以进化种群作为计算整体,其中每一个个体均具有以下环境信息:违反约束程度、与约束边界的距离、拥挤度和支配关系(以无约束 Pareto 支配关系计算).

违反约束程度及与约束边界距离的计算方法为

$$C(i) = \sum_{j=1}^p \max(0, g_j(x_i)), \quad (3)$$

$$D(i) = \sum_{j=1}^p |g_j(x_i)|. \quad (4)$$

其中: p 为不等式约束个数, $g_j(x_i)$ 为第 i 个个体在不等式约束 j 下的值.

拥挤度计算方法为

$$W(i) = \begin{cases} \infty, & i \text{ is in the boundary;} \\ \sum_{j=1}^k \left| \frac{f_j(i+1) - f_j(i-1)}{f_j(N) - f_j(1)} \right|, & \text{else.} \end{cases} \quad (5)$$

其中: k 为目标函数个数, N 为种群个体数, $f_j(1), f_j(2), \dots, f_j(N)$ 为全部个体在目标函数 j 上从小到大的排列.

基于上述分析,提出一种环境 Pareto 支配选择策略:

1) 当两个个体相比较时,一个个体为可行解,另一个个体为不可行解,如果可行解与约束边界的距离小于不可行解与约束边界的距离,则选择可行解作为较优个体,否则选择两个个体中的非支配个体作为较优个体,如果两个个体互不支配,则选择拥挤度较大的个体;

2) 当两个比较的个体均为可行解时,选择非支配个体,如果两个个体互不支配,则选择拥挤度较大的个体;

3) 当两个比较的个体均为不可行解时,选择非支配个体作为较优解,如果两个个体互不支配,则选择违反约束程度较低的个体.

以上选择规则可用图 1~图 3 表示.

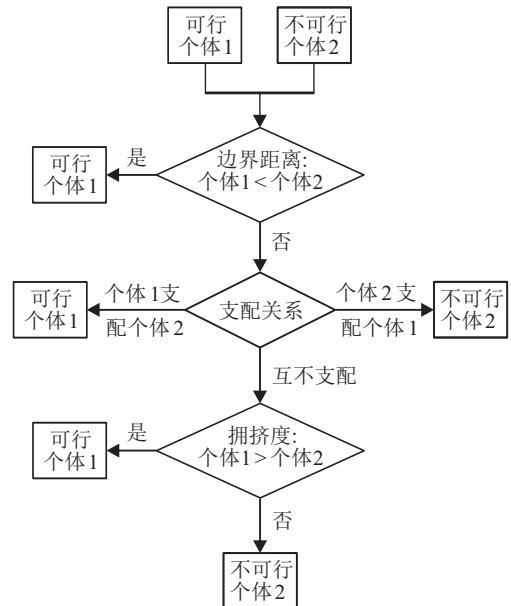


图 1 环境 Pareto 支配选择策略 1

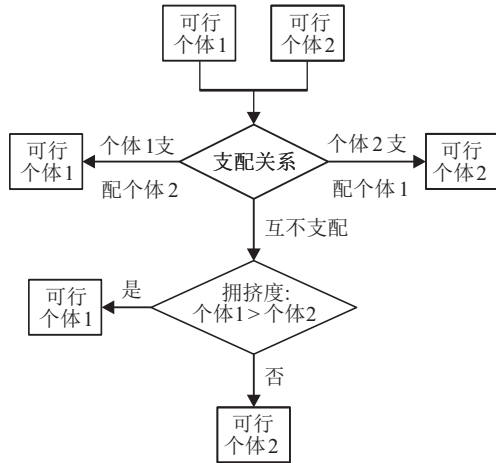


图 2 环境 Pareto 支配选择策略 2

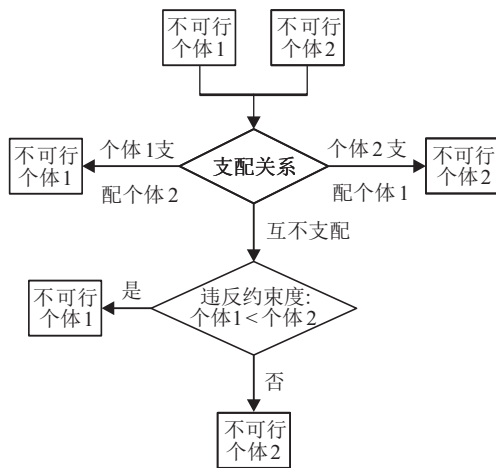


图 3 环境 Pareto 支配选择策略 3

通过比较两种不同的选择策略可知，环境 Pareto 支配选择策略对规则 1 进行了较大的改动，不再以是否可行作为选择的唯一准则。由图 1 可知，规则 1 结合了其他环境因素，更加客观地评价了两个解之间的优劣，使得可行解与不可行解竞争时，优秀的不可行解得以胜出。对于有约束问题，这一选择机制更加合理，更加有利于优秀不可行解的保留，从而为变异、交叉等操作提供更有利的信息；规则 2 两者相同，仍然以支配关系作为选择标准；规则 3 增加了支配关系的比较，将其作为首要判断标准，使得不可行解也在一定程度上向着 Pareto 前沿进化，其次将约束程度作为最终判定指标，选择出违反约束程度较低的个体。

选择操作决定着算法的进化方向，由环境 Pareto 支配选择策略可知，新的选择策略为可行解和不可行解设计了两个进化方向。可行解将向着可行域内的 Pareto 前沿前进，不可行解将向着不可行域内约束边界的 Pareto 前沿前进。

环境 Pareto 支配选择策略结合违反约束程度、与约束边界的距离、拥挤度和支配关系 4 方面环境信息，更加客观地评价了两个个体之间的优劣程度，使优秀不可行解有更大的机会保留下来。

3 其他操作算子的改进

3.1 变异策略

有约束多目标问题有它自身的特点，最优 Pareto 前沿往往位于可行域与不可行域之间的边界，不可行解在进化计算过程中常被视为较差解而予以丢弃，但它们往往是开发孤立可行解的桥梁，更有利于搜索到较好的可行解。所以在进化计算中，优秀的不可行解同样不容忽视，它也可以为进化过程提供有益的方向信息。

在进化过程中，外部不可行解集可能为空，所以定义以下两种情况。

如果不可行解集中个体数小于 $N'_2 (N'_2 < N_2)$ ，则变异操作按照下式进行：

$$v^{t+1} = \begin{cases} G_{\text{best}} + F(x_{r1}^t - x_{r2}^t), & \text{rand} \leq \beta_1; \\ x_{r1}^t + F(x_{r2}^t - x_{r3}^t), & \text{rand} > \beta_1. \end{cases} \quad (6)$$

如果不可行解集中个体数大于 $N'_2 (N'_2 > N_2)$ ，则变异操作按照下式进行：

$$v^{t+1} = \begin{cases} G_{\text{best}} + F(H_{\text{best}1} - x_{r1}^t) + F(H_{\text{best}2} - x_{r2}^t), & \text{rand} \leq \beta_1; \\ x_{r1}^t + F(x_{r2}^t - x_{r3}^t), & \text{rand} > \beta_1. \end{cases} \quad (7)$$

其中： F 为缩放因子， G_{best} 为可行解集中的随机个体， H_{best} 为不可行解集中的随机个体， x_{r1}^t 、 x_{r2}^t 和 x_{r3}^t 为在 t 代中不同的 3 个随机个体， v^{t+1} 为更新之后的新个体， β_1 为策略选择概率， N'_2 为变异策略选择阈值， N_2 为不可行解集最大规模。

3.2 自适应缩放因子

差分进化算法中变异操作的高效性一方面取决于变异方式的选取，一方面取决于缩放因子的合理选择。基本差分进化算法为固定缩放因子，合适的缩放因子往往是多次实验后的结果，不利于进一步的应用。为进一步提高算法的自适应能力，本文提出一种缩放因子的自适应策略。

首先，在初始化阶段，为每一个个体随机初始化一个缩放因子

$$F = F_{\text{min}} + \text{rand}(F_{\text{max}} - F_{\text{min}}). \quad (8)$$

其中： F_{max} 和 F_{min} 分别为最大和最小缩放因子， rand 为 $[0, 1]$ 之间的随机数。

因为差分进化算法采取的是“贪婪”选择策略，所以新父代个体均是进化种群的个体历史最优解。基于此种策略，在每一代进化过程中，父代个体与子代个体均有优劣之分，将产生较优子代个体所采用的缩放因子认为是优秀缩放因子，将其保存至 F_G 中。在下次迭代进化中，上代产生优秀子代个体中的缩放因子

不变,产生劣质子代个体中的缩放因子按下式更新:

$$F_{\text{new}} = (1 - \alpha)F_I + \alpha F_G. \quad (9)$$

其中: F_I 为较差缩放因子, F_G 为随机选择的较优缩放因子, α 为学习效率系数.

每一次自适应前首先计算所有缩放因子的标准差,当标准差小于阈值 σ 时,说明所有缩放因子之间的差异变小,需要按下式重新分配缩放因子:

$$F = F_{\min} + \text{rand}(F'_{\max} - F_{\min}), \quad (10)$$

其中 F'_{\max} 为所有缩放因子中的最大值.

3.3 自适应交叉概率

在进化前期,一般都能满足多样性的要求,应尽快提高算法的收敛速度,对于较差个体应更多地保留变异个体部分;在算法后期,为了避免算法陷入局部最优,提高种群的多样性,需设置较大的交叉概率.对于多目标算法而言,交叉概率是保持种群多样性和分布性的有效参数.所以当种群多样性较好时,应设置较小的交叉概率以加快搜索速度;在种群多样性下降后,要提高交叉概率使种群保持多样性.根据这一特性,提出以下交叉概率自适应策略:

$$\alpha_t = \frac{\sum_{j=1}^k \sqrt{\frac{\sum_{i=1}^{\text{NP}} (f_i^j - \bar{f}^j)^2}{\text{NP}}}}{k}, \quad (11)$$

$$\text{CR} = \text{CR}_{\max} - \frac{\alpha_t}{\max(\alpha_T)} (\text{CR}_{\max} - \text{CR}_{\min}). \quad (12)$$

其中: f_i^j 为第 i 个个体的第 j 个目标函数值, \bar{f}^j 为第 j 个目标函数的平均值, k 为目标函数个数, α_t 为第 t 代交叉概率自适应系数, α_T 为第 1 代到第 t 代自适应系数的集合.

由式(1)和(2)可知, α_t 可以反映出种群目标函数的分散情况.当种群较分散时,其值较大,使得交叉概率变小;当种群过于集中时,其值变小,使得交叉概率变大.

3.4 外部种群更新策略

外部种群分为可行解集和不可行解集,设其最大容量分别为 N_1 和 N_2 ($N_1 > N_2$),具体更新方式如下:

可行解集的更新.将本代新产生的可行个体与可行解集结合为一个新的种群,选出其中的非支配个体,删除其他个体.如果个体数目小于 N_1 ,则直接构成新的可行解集;如果数目大于 N_1 ,则采用循环删除的方法删除拥挤度较小的个体^[14].

不可行解集的更新.设置平衡系数 β_2 ($0 < \beta_2 < 1$),将本代新产生的不可行个体与不可行解集结合为一个新的种群,选出其中的非支配个体,删除其他个体.如果个体数目小于 N_2 ,则直接构成新的不可行解

集;如果个体数目介于 N_2 与 $N_2(1 + \beta_2)$ 之间,则采用循环删除的方法删除拥挤度较小的个体;如果个体数目大于 $N_2(1 + \beta_2)$,则首先删除违反约束度较大的个体,使个体数目为 $N_2(1 + \beta_2)$,然后再采用循环删除的方法删除拥挤度较小的个体.

4 算法实现流程

基于上述改进,本文提出基于环境 Pareto 支配选择策略的有约束多目标差分进化算法 (EPDSDE),其具体实现步骤如下.

1) 初始化,设定相关参数.种群大小 NP,可行解集大小 N_1 ,不可行解集大小 N_2 ,最大和最小缩放因子 F_{\max} 、 F_{\min} ,最大和最小变异概率 CR_{\max} 、 CR_{\min} ,变异策略选择概率 β_1 ,平衡系数 β_2 ,最大迭代次数 Gen.

2) 随机产生大小为 NP 的初始种群,计算其目标函数值和约束程度.

3) 依据 3.4 节生成可行解集和不可行解集.

4) 依据 3.1 节对本代个体进行变异及交叉操作.

5) 依据 3.2 节和 3.3 节对缩放因子和交叉概率进行更新.

6) 运用基于环境 Pareto 支配的选择策略选择出下一代个体.

7) 更新可行解集和不可行解集.

8) 判断是否满足终止条件,若不满足转到 Step 4,否则输出可行解集,即 Pareto 前沿.

5 实验仿真与分析

本文采用 CTP3、CTP4 和 CF1 三个经典测试函数来检验算法的性能,CTP3、CTP4 表达式如下.

$$\begin{cases} \min F(\bar{x}) = \min \begin{cases} f_1(\bar{x}) \\ f_2(\bar{x}) \end{cases} = \begin{cases} x_1 \\ c(\bar{x}) \left(1 - \frac{f_1(\bar{x})}{c(\bar{x})}\right) \end{cases}, \\ g(\bar{x}) = \\ a|\sin\{b\pi[\sin(\theta)[f_2(\bar{x}) - e] + \cos(\theta)f_1(\bar{x})]^c\}|^d - \\ \cos(\theta)[f_2(\bar{x}) - e] + \sin(\theta)f_1(\bar{x}) \leq 0. \end{cases}$$

其中

$$c(x) = 41 + \sum_{i=2}^5 [x_i^2 - 10 \cos(2\pi x_i)],$$

$$0 < x_1 < 1, \quad -5 < x_i < 5, \quad i = 2, 3, \dots, 5.$$

其他参数如表 1 所示.

表 1 CTP3、CTP4 参数列表

	θ	a	b	c	d	e
CTP3	-0.2π	0.1	10	1	0.5	1
CTP4	-0.2π	0.75	10	1	0.5	1

CF1 表达式如下:

$$\begin{cases} \min F(\bar{x}) = \min \begin{cases} f_1(\bar{x}) \\ f_2(\bar{x}) \end{cases} = \\ \begin{cases} x_1 + \frac{2}{|J_1|} \sum_{j \in J_1} \left(x_j - x_1^{0.5(1+\frac{3(j-2)}{n-2})}\right)^2 \\ 1 - x_1 + \frac{2}{|J_2|} \sum_{j \in J_2} \left(x_j - x_1^{0.5(1+\frac{3(j-2)}{n-2})}\right)^2 \end{cases}, \\ g(\bar{x}) = 1 - f_1 - f_2 + a|\sin[N\pi(f_1 - f_2 + 1)]| \leq 0. \end{cases}$$

其中: $0 \leq x_j \leq 1, j = 1, 2, \dots, n, n = 10, J_1 = \{j|j = 3, 5, \dots, n\}, J_2 = \{j|j = 2, 4, \dots, n\}, N = 10, a = 1$.

在相同测试函数及硬件下, 本文算法 (EPDSDE) 与 NSGA-II^[15]、CMODE^[16]和 DPDE^[17]分别独立运行 30 次. 以收敛性指标 GD、IGD^[18]和附加二元指标 TSC^[19]分别对 4 种算法所得结果进行统计比较, 以此来衡量算法的优劣. 本文算法参数设置如表 2 所示.

表 2 本文算法参数列表

参数	数值
种群大小	200
可行域规模	100
最小交叉概率	0.2
进化代数	1000
最大缩放因子	1
标准差阈值	0.01
缩放因子学习效率	0.1
不可行域规模	50
最大交叉概率	0.6
不可行解集平衡系数	0.2
最小缩放因子	0.2
策略选择概率	0.75

图 4~图 6 为从 30 次运行中随机选择的一次运行结果, 其中 PF 为真实 Pareto 前沿. 从图中可以看出, 4 种算法都具有良好的分布广度, 均能找到 Pareto 前沿的大致位置.

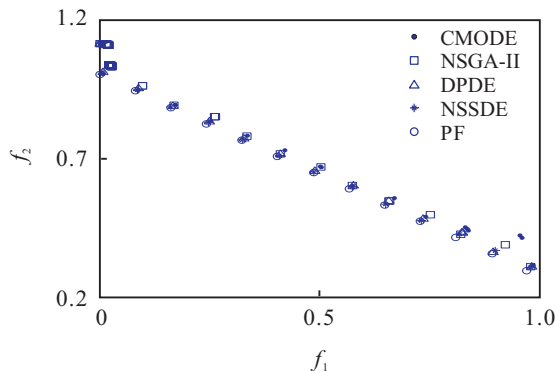


图 4 测试函数 CTP3 的 Pareto 前沿

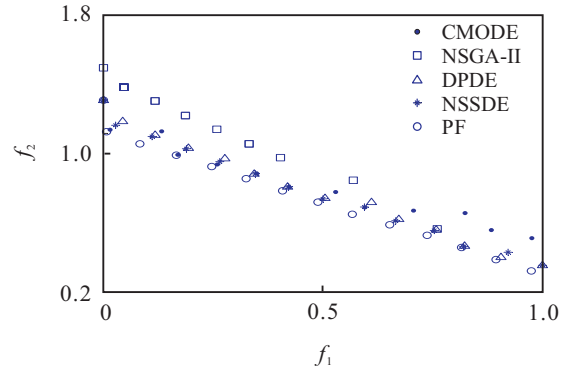


图 5 测试函数 CTP4 的 Pareto 前沿

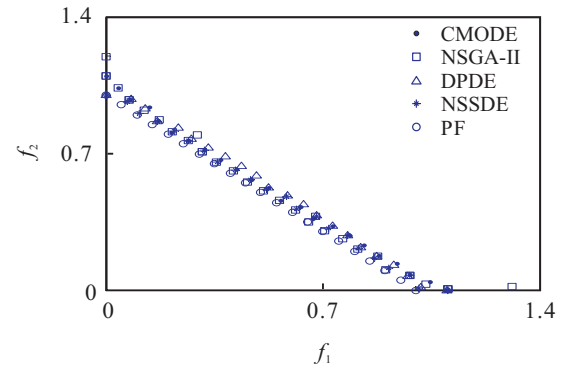


图 6 测试函数 CF1 的 Pareto 前沿

表 3 给出了 4 种算法的平均运行时间, 由于本文算法在外部解集只保留非支配解, 不必对整个种群进行非支配排序, 所以计算时间大大降低.

表 3 4 种算法的平均运行时间 s

	CMODE	NSGA-II	DPDE	EPDSDE
CTP3	111	245	153	82
CTP3	112	250	141	83
CF1	116	204	142	85

为了进一步定量评价 4 种算法的性能, 表 4~表 7 给出了 4 种算法对以上测试函数收敛性及附加二元指标的统计结果. A 代表本文算法, B 代表 NSGA-II 算法, C 代表 CMODE 算法, D 代表 DPDE 算法.

从表 4~表 6 可以看出, 对于 CTP3 和 CTP4, 本文算法在收敛性和反向收敛性指标上均优于其他 3 种算法, 且具有最小的标准差. 这说明本文算法所得非劣解集更加接近真实的 Pareto 前沿, 且算法更加稳定. 对于 CF1, 本文算法具有最小的收敛及反向收敛性指标和次优的标准差, 说明本文所得非劣解集最接近真实 Pareto 前沿, 但稳定性稍差于 DPDE 算法.

表 4 测试函数 CTP3 评价统计结果

	CMODE		NSGA-II		DPDE		EPDSDE	
	GD	IGD	GD	IGD	GD	IGD	GD	IGD
最差解	1.48 e-3	2.61 e-3	4.00 e-3	1.37 e-1	3.40 e-4	3.56 e-4	7.34 e-5	7.33 e-5
最优解	2.78 e-4	2.16 e-4	3.82 e-5	3.03 e-4	6.63 e-5	6.78 e-5	3.40 e-5	3.52 e-5
平均值	7.69 e-4	8.05 e-4	6.01 e-4	1.57 e-2	1.77 e-4	1.66 e-4	5.36 e-5	5.16 e-5
标准差	2.99 e-4	4.66 e-4	9.63 e-4	3.03 e-2	6.58 e-5	6.21 e-5	1.16 e-5	1.06 e-5

表5 测试函数 CTP4 评价统计结果

	CMODE		NSGA-II		DPDE		EPDSDE	
	GD	IGD	GD	IGD	GD	IGD	GD	IGD
最大值	6.31 e-3	2.43 e-2	9.24 e-1	2.55	6.96 e-3	7.36 e-3	1.25 e-3	1.62 e-3
最小值	9.85 e-4	1.68 e-3	2.30 e-4	8.11 e-3	1.59 e-3	1.59 e-3	6.34 e-4	6.34 e-4
平均值	3.02 e-3	9.33 e-3	7.26 e-2	3.42 e-1	2.65 e-3	3.02 e-3	9.15 e-4	9.38 e-4
标准差	1.27 e-3	6.11 e-3	1.81 e-1	6.32 e-1	1.00 e-3	1.35 e-3	1.83 e-4	2.19 e-4

表6 测试函数 CF1 评价统计结果

	CMODE		NSGA-II		DPDE		EPDSDE	
	GD	IGD	GD	IGD	GD	IGD	GD	IGD
最大值	2.26 e-2	2.71 e-2	2.35 e-2	5.85 e-2	5.24 e-3	3.08 e-3	7.64 e-3	2.97 e-3
最小值	1.55 e-3	1.09 e-3	2.34 e-3	1.68 e-3	1.76 e-3	1.43 e-3	1.07 e-3	7.46 e-4
平均值	5.14 e-3	4.55 e-3	7.79 e-3	1.31 e-2	2.62 e-3	2.08 e-3	2.02 e-3	1.22 e-3
标准差	3.99 e-3	4.94 e-3	4.76 e-3	1.13 e-2	7.71 e-4	3.87 e-4	1.36 e-3	5.27 e-4

表7 测试函数 CTP2、CTP4、CF1 附加二元指标统计结果

函数	指标	TSC(A,B)	TSC(B,A)	TSC(A,C)	TSC(C,A)	TSC(A,D)	TSC(D,A)
CTP3	最大值	1.00	0.21	0.95	0.33	1.00	0.53
	最小值	0.11	0.00	0.69	0.00	0.50	0.00
	平均值	0.71	0.06	0.85	0.15	0.81	0.21
	标准差	0.35	0.08	0.06	0.07	0.10	0.12
CTP4	最大值	1.00	0.27	0.92	0.43	1.00	0.43
	最小值	0.05	0.00	0.45	0.07	0.54	0.00
	平均值	0.80	0.04	0.73	0.22	0.80	0.20
	标准差	0.33	0.07	0.13	0.10	0.10	0.10
CF1	最大值	0.74	0.80	0.81	0.65	0.91	0.50
	最小值	0.22	0.42	0.43	0.17	0.56	0.08
	平均值	0.51	0.60	0.62	0.43	0.73	0.29
	标准差	0.12	0.10	0.09	0.10	0.09	0.10

从表7可以看出:对于测试函数CTP3和CTP4, TSC(A, B)、TSC(A, C)和TSC(A, D)均大于TSC(B, A)、TSC(C, A)和TSC(D, A),这说明本文算法所得非劣解多于其他3种算法。对于CF1, TSC(A, C)和TSC(A, D)均大于TSC(C, A)和TSC(D, A),这说明本文算法所得非劣解多于CMODE和DPDE两种算法;但TSC(A, B)小于TSC(B, A),这说明NSGA-II所得非劣解更多,而由于本文算法收敛性优于NSGA-II,说明本文算法虽然能找到更优的Pareto前沿,但是算法在最优解附近的开发性还有待进一步的改进。

6 结论

本文综合考虑了可行解和不可行解在算法寻优过程中起到的积极作用。基于环境Pareto支配的选择策略,同时考虑Pareto支配关系、违反约束程度、与约束边界的距离和拥挤度4方面环境因素,使优秀不可行个体在进化过程中得以保存,并参与到变异操作中;采用了缩放因子和交叉概率的自适应策略,共同构成了一种新的有约束多目标差分进化算法。综合分析本文算法与另外3种优秀算法在经典测试函数中的比较结果,表明本文所提算法具有更高的收敛精度。

参考文献(References)

- [1] Cai Z X, Wang Y. A multi objective optimization based evolutionary algorithm for constrained optimization[J]. IEEE Trans on Evolutionary Computation, 2006, 10(6): 658-675.
- [2] Zhang H, Zhu Y L, Yan X H. Multi-hive artificial bee colony algorithm for constrained multi-objective optimization[C]. 2012 IEEE Congress on Evolutionary Computation(CEC). IEEE, 2012: 1-8.
- [3] 张勇, 巩敦卫, 任永强, 等. 用于约束优化的简洁多目标微粒群优化算法[J]. 电子学报, 2011, 39(6): 1436-1440. (Zhang Y, Gong D W, Ren Y Q, et al. Barebones multi-objective particle swarm optimizer for constrained optimization problems[J]. Acta Electronica Sinica, 2011, 39(6): 1436-1440.)
- [4] Zhang H, Zhu Y L, Zhang D Y. Self-adaptive root growth model for constrained multi-objective optimization[C]. 2013 IEEE Congress on Evolutionary Computation(CEC). IEEE, 2013: 2360-2367.
- [5] 俞国燕, 李鹏, 何真, 等. 一种用于多目标约束优化的

- 改进进化算法[J]. 计算机集成制造系统, 2009, 15(6): 1172-1178.
(Yu G Y, Li P, He Z, et al. Advanced evolutionary algorithm used in multi-objective constrained optimization problem[J]. Computer Integrated Manufacturing Systems, 2009, 15(6): 1172-1178.)
- [6] 刘若辰, 焦李成, 雷七峰, 等. 一种新的差分进化约束优化算法[J]. 西安电子科技大学学报, 2011, 38(1): 47-53.
(Liu R C, Jiao L C, Lei Q F, et al. new differential evolution constrained optimization algorithm[J]. J of Xidian University, 2011, 38(1): 47-53.)
- [7] 王勇, 蔡自兴, 周育人, 等. 约束优化进化算法[J]. 软件学报, 2009, 20(1): 11-29.
(Wang Y, Cai Z X, Zhou Y R, et al. Constrained optimization evolutionary algorithms[J]. J of Software, 2009, 20(1): 11-29.)
- [8] 王建林, 吴佳欢, 张超然, 等. 基于自适应进化学习的约束多目标粒子群优化算法[J]. 控制与决策, 2014, 29(10): 1765-1770.
(Wang J L, Wu J H, Zhang C R, et al. Constrained multi-objective particle swarm optimization algorithm based on self-adaptive evolutionary learning[J]. Control and Decision, 2014, 29(10): 1765-1770.)
- [9] 蔡自兴, 江中央, 王勇, 等. 一种新的基于正交实验设计的约束优化进化算法[J]. 计算机学报, 2010, 33(5): 855-864.
(Cai Z X, Jiang Z Y, Wang Y, et al. A Novel Constrained Optimization Evolutionary Algorithm Based on Orthogonal Experimental Design[J]. Chinese J of Computers, 2010, 33(5): 855-864.)
- [10] 尚荣华, 焦李成, 胡朝旭, 等. 修正免疫克隆约束多目标优化算法[J]. 软件学报, 2012, 23(7): 1773-1786.
(Shang R H, Jiao L C, Hu C X, et al. Modified immune clonal constrained multi-objective optimization algorithm[J]. J of Software, 2012, 23(7): 1773-1786.)
- [11] 邹秀芬, 刘敏忠, 吴志健, 等. 解约束多目标优化问题的一种鲁棒的进化算法[J]. 计算机研究与发展, 2004, 41(6): 985-990.
(Zou X F, Liu M Z, Wu Z J, et al. A robust evolutionary algorithm for constrained multi-objective optimization problems[J]. J of Computer Research and Development, 2004, 41(6): 985-990.)
- [12] Ling H F, Xiao Y H, Zhou X Z, et al. An improved PSO algorithm for constrained multi-objective optimization problems[C]. 2011 Int Conf on Computer Science and Service System(CSSS). IEEE, 2011: 3859-3863.
- [13] Deb K, Pratap A, Meyarivan T. Constrained test problems for multi-objective evolutionary optimization[C]. Evolutionary Multi-Criterion Optimization. Berlin Heidelberg: Springer, 2001: 284-298.
- [14] 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.
(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.)
- [15] Deb K, Pratap A, Agarwal S, et al. A fast and elitist multi-objective genetic algorithm: NSGA-II[J]. IEEE Trans on Evolutionary Computation, 2002, 6(2): 182-197.
- [16] 毕晓君, 刘国安. 基于云差分进化算法的约束多目标优化实现[J]. 哈尔滨工程大学学报, 2012, 33(8): 1022-1031.
(Bi X J, Liu G A. A cloud differential evolutionary algorithm for constrained multi-objective optimization[J]. J of Harbin Engineering University, 2012, 33(8): 1022-1031.)
- [17] 孟红云, 张小华, 刘三阳. 用于约束多目标优化问题的双群体差分进化算法[J]. 计算机学报, 2008, 31(2): 228-235.
(Meng H Y, Zhang X H, Liu S Y. A differential evolution based on double populations for constrained multi-objective optimization problem[J]. Chinese J of Computers, 2008, 31(2): 228-235.)
- [18] Venske S M, Goncalves R A, Delgado M R. ADEMO/D: Multiobjective optimization by an adaptive differential evolution algorithm[J]. Neurocomputing, 2014, 127: 65-77.
- [19] 尚荣华, 焦李成, 马文萍, 等. 用于约束多目标优化的免疫记忆克隆算法[J]. 电子学报, 2009, 37(6): 1289-1294.
(Shang R H, Jiao L C, Ma W P, et al. An Immune Memory Clone Algorithm for Constrained Multi-Objective Optimization[J]. Acta Electronica Sinica, 2009, 37(6): 1289-1294.)

(责任编辑: 齐 霁)