

基于支配强度的 NSGA2 改进算法

赖文星 邓忠民

(北京航空航天大学宇航学院 北京 100191)

摘 要 NSGA2 是一种简单、高效且被广泛使用的多目标进化算法 (Multi-objective Evolutionary Algorithm, MoEA),但在求解实际工程领域中的高维、复杂非线性多目标优化问题 (Multi-objective Optimization Problems, MOP) 时,存在无法有效识别伪非支配解、计算效率低、解集收敛性和分布性较差等设计缺陷。对此,文中提出一种基于支配强度的 NSGA2 改进算法 (INSGA2-DS)。新算法采用快速支配强度排序法构造非支配集,引入了考虑方差的拥挤距离公式,并通过自适应精英保留策略动态调整精英保留规模。基于标准测试函数的仿真实验表明,INSGA2-DS 算法较好地改善了 NSGA2 算法的收敛性和分布性。

关键词 NSGA2,多目标进化算法,多目标优化问题,支配强度

中图分类号 TP301 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2018.06.033

Improved NSGA2 Algorithm Based on Dominant Strength

LAI Wen-xing DENG Zhong-min

(School of Astronautics, Beihang University, Beijing 100191, China)

Abstract NSGA2 algorithm is a simple, efficient and widely used multi-objective evolutionary algorithm. However, when solving high-dimensional and complex nonlinear multi-objective optimization problems in practical engineering field, NSGA2 has some obvious design defects, such as ineffective identification of pseudo non-dominated solutions, low computational efficiency, poor convergence and distribution. In order to remedy the above drawbacks, this paper proposed an improved NSGA2 algorithm based on dominant strength (INSGA2-DS). INSGA2-DS uses the fast dominant strength sorting method to construct non-dominated set, introduces a new crowding distance with considering variance to improve the distribution of solution sets, and adopts the adaptive elitist retention strategy to adjust elitist retention scale in evolutionary process automatically. The experimental results of INSGA2-DS and NSGA2 with standard test functions show that INSGA2-DS algorithm can improve the convergence and distribution of NSGA2 algorithm effectively.

Keywords NSGA2, Multi-objective evolutionary algorithm, Multi-objective optimization problems, Dominant strength

1 引言

多目标进化算法 (MOEA) 是一类通过模拟生物进化机制而形成的全局性概率优化搜索方法^[1],而 NSGA2 算法凭借简单、高效等优点,成为了最热门的 MOEA 算法之一。NSGA2 算法是目前求解多目标优化问题 (MOP) 的有效途径,被广泛应用于科学计算、工程设计等领域。

1989 年,Goldberg 提出采用 Pareto 优化求解多目标优化问题。基于 Goldberg 的思想,Srinivas 和 Deb 于 1994 年提出了第一代非支配排序遗传算法 (Non-dominated Sorting Genetic Algorithm, NSGA)^[2]。NSGA 通过非支配排序方法对种群个体进行分层排序,并利用适应度共享函数保持种群的多样性,取得了良好的效果;但其存在计算复杂度太高、需要指定共享参数、缺少精英策略等设计缺陷。鉴于此,Deb 等人

于 2002 年提出了第二代 NSGA2 算法^[3]。NSGA2 通过采用快速非支配排序方法、拥挤距离比较和精英保留策略,在处理低维 MOP 时计算效率较高、解集的质量较好;但是在求解高维复杂 MOP 时,因种群中非支配个体数量过于庞大,其无法有效获得 Pareto 解集。

为了进一步提高 NSGA2 算法的实用性,许多学者提出了改进方法。Kollat 等通过将 ϵ 支配引入 NSGA2 提出了 ϵ -NSGAII,以提高算法的效率和可靠性^[4],并用一组测试函数比较了两种算法的性能。2007 年,Zhang 等通过将数学规划的方法应用到 NSGA2 算法中^[5],提出了基于分解的多目标进化算法 (MOEA/D),将其高维多目标优化问题分解为若干组低维多目标优化问题,再进行优化求解;此外,他们于 2009 年又提出了基于差分进化算子 (DE) 的 MOEA/D^[6],与采用 GA 算子的 NSGA2 相比,该算法收敛速度快且解集质量高,

到稿日期:2017-05-03 返修日期:2017-08-26 本文受国家自然科学基金(10972019)资助。

赖文星(1993-),男,硕士生,主要研究方向为多目标优化算法,E-mail:laiwenxing@buaa.edu.cn;邓忠民(1968-),男,教授,主要研究方向为结构优化,E-mail:07011@buaa.edu.cn(通信作者)。

提高了求解具有复杂 Pareto 边界的 MOP 的性能。Tan 等^[7]提出了一个自适应的变化算子,该变化算子利用了基因的二元表示结构,且综合了交叉和变异的功能。Elhossini 等^[8]提出了粒子群算法(PSO)和 NSGA2 的混合算法,其有效改善了求解 MOP 的性能。谢承旺等提出了一种带差分局部搜索的改进型 NSGA2 算法^[9],该算法利用差分进化中变异算子的定向引导作用,与 NSGA2 算法相结合,改善了解集的分布性。Deb 等^[10]提出了一种基于 NSGA2 的高维多目标进化算法(NSGA III),其采用基于参考点的非支配排序方法,比 MOEA/D 和 NSGA2 具有更好的求解高维 MOP 的性能。Shim 等^[11]提出先将非支配排序和目标分解结合,再进入进化梯度局部搜索,提高了算法求解高维 MOP 的性能。Qiu 等^[12]提出了多目标自适应差分算法,其在考虑参数敏感性和算法收敛性的基础上,引入了参数自适应和跨代变异机制。

以上改进虽然在一定程度上改善了 NSGA2 算法的性能,但是在处理一些具有复杂 Pareto 边界、高维多变量、复杂非线性 MOP 时,还存在较多缺陷,如计算复杂度较高、无法识别伪非支配解、拥挤距离公式不合理等。

2 相关定义

2.1 多目标优化问题

多目标优化问题由决策变量、目标函数、约束条件 3 个部分组成,其数学描述如下:

$$\begin{aligned} \min f(x) &= [f_1(x), f_2(x), \dots, f_n(x)] \\ \text{s. t. } g(x) &= [g_1(x), g_2(x), \dots, g_k(x)] \leq 0 \\ h(x) &= [h_1(x), h_2(x), \dots, h_m(x)] = 0 \\ x &= [x_1, x_2, \dots, x_d, \dots, x_D] \\ x_{d_{\min}} &\leq x_d \leq x_{d_{\max}} \end{aligned} \quad (1)$$

其中, x 为 D 维决策变量, $f(x)$ 为第 n 个子目标函数, $g(x)$ 为不等式约束条件, $h(x)$ 为等式约束条件。

2.2 Pareto 最优化

对于个体 x^* 和 x , 如果满足条件 $\forall n, f_n(x^*) \leq f_n(x) \cap \exists n_0, f_{n_0}(x^*) < f_{n_0}(x)$, 则称 x^* 支配 x , x^* 为非支配解。

称所有非支配解组成的集合为 Pareto 解集, 称 Pareto 解集对应的目标所形成的区域为 Pareto 边界。

2.3 支配强度

支配强度 ξ 用来比较非支配集中个体的优劣性能, 计算公式如下:

$$\xi_i = \sum_{k=1}^M \left(\frac{f_k(i) - f_k^{\min}}{f_k^{\max} - f_k^{\min}} \right) \quad (2)$$

其中, 个体 i 属于非支配集, MOP 的子目标个数为 M , f_k^{\max} 和 f_k^{\min} 分别代表第 k 个子目标的最大值和最小值。

个体的非支配强度越小, 表示个体的优化结果总体上越好。在非支配集中, 首先比较支配强度, 然后比较拥挤距离。非支配集中任意个体 i 和个体 j 之间优劣性的比较算子为: 若 $\xi_i < \xi_j$ 或 $\xi_i = \xi_j \cap i_d > j_d$, 则个体 i 优于个体 j 。

2.4 进化操作

NSGA2 采用模拟二进制交叉和多项式变异产生新个体^[13]。从父辈种群中随机选取两个个体 P_1 和 P_2 , 若进入交

叉操作, 则产生两个子代个体 Q_1 和 Q_2 的过程如下:

$$\begin{cases} Q_1 = 0.5[(1+\beta)P_1 + (1-\beta)P_2] \\ Q_2 = 0.5[(1-\beta)P_1 + (1+\beta)P_2] \end{cases} \quad (3)$$

其中, β 与随机数 $\mu \in [0, 1]$ 有关, 由下列公式生成:

$$\beta = \begin{cases} (2\mu)^{\frac{1}{\eta_c+1}}, & \mu \leq 0.5 \\ (2(1-\mu))^{\frac{-1}{\eta_c+1}}, & \mu > 0.5 \end{cases} \quad (4)$$

其中, η_c 为非负交叉参数, η_c 越大, 交叉互换后产生的子代个体与父代个体就越相像。

若进入变异操作, 则从父辈种群中随机选取一个个体 P_3 , 产生变异子代个体 Q_3 的计算公式如下:

$$Q_3 = \begin{cases} P_3 + [(2r)^{\frac{1}{\eta_m+1}} - 1](x_{\max} - x_{\min}), & r \leq 0.5 \\ P_3 + [1 - (2-2r)^{\frac{1}{\eta_m+1}}](x_{\max} - x_{\min}), & r > 0.5 \end{cases} \quad (5)$$

其中, 随机数 $r \in [0, 1]$, η_m 是变异分布指数, x_{\max} 和 x_{\min} 分别是变量的上限和下限。 η_m 是自定义参数, 与交叉参数 η_c 一样, 可以根据进化的具体情况进行调整。

3 基于支配强度的 NSGA2 改进算法

本文提出一种基于支配强度的 NSGA2 改进算法——在非支配排序、分布性保持、精英保留策略 3 个方面对经典 NSGA2 算法进行了改进, 提高了算法的计算效率、收敛性和分布性。

3.1 快速支配强度排序法

经典 NSGA2 采用快速非支配排序法, 根据受支配次数, 将个体依次放入非支配集中, 从而完成对种群的分层排序。当求解高维、复杂的 MOP 时, 由于子目标数量较多, 或者目标函数变化范围较大, 导致排序耗时过长而进入慢速链状态, 无法高效完成种群的分层排序。此外, 根据传统的支配关系很难评判个体的优劣性, 造成非支配集中存在大量伪非支配解, 降低了算法的收敛效率和解集的质量。为此, 本文采用快速支配强度排序法。

1) 改进快速排序方法。为了尽可能避免算法进入慢速链状态, 从而降低算法的时间复杂度, 本文采用改进快速排序方法来构造非支配集^[14]。新方法将原来的一个比较个体改为两个比较个体, 减少了算法总的比较次数, 进而提高了算法的计算速度, 具体过程如下。

Step1 将种群中的所有个体放入比较集。

Step2 从比较集中选择第一个比较个体 C_1 , 然后将其与比较集中的其他个体逐一进行比较, 找到第一个与个体 C_1 互为非支配或者支配它的个体 C_2 , 并称其为第二个比较个体; 如果无法找到个体 C_2 , 则将个体 C_1 放入非支配集中, 重复步骤 Step2。

Step3 第一个比较个体 C_1 和第二个比较个体 C_2 分别与比较集中的其他个体进行比较, 被这两个比较个体之一所支配的个体均被清除; 若个体 C_1 或个体 C_2 不被比较集中的其他个体所支配, 则将其放入非支配集中; 原有比较集被清空, 将剩余不被个体 C_1 和 C_2 所支配的个体放入比较集中。

Step4 重复步骤 Step2,直至比较集为空集。至此,非支配集的构造完成。

2)引入支配强度。当种群中的所有个体完成分级排序后,本文引入一个新的优劣性能判断指标——支配强度 ξ 。非支配集中个体的支配强度越小,表示其大部分子目标的优化结果越好,进入下一代种群的概率就越大。因此,本文通过支配强度去除伪非支配解,保证性能更优异的个体优先获得遗传,提高了算法解集的质量。

3.2 分布性保持方法

NSGA2 拥挤距离公式仅仅考虑了相邻个体之间的距离,在不同子目标上拥挤距离差异程度较大的个体无法获得较多的遗传机会,不利于解集分布性的维护。本文提出一种考虑方差的新型拥挤距离,其计算公式如下:

$$i_D = \frac{i_d}{\left\{ \frac{1}{m} \sum_{j=1}^m [(|f_j^{i+1} - f_j^{i-1}|) - \frac{i_d}{m}]^2 \right\}^{-1} + 1} \quad (6)$$

传统距离公式 $i_d = \sum_{j=1}^m (|f_j^{i+1} - f_j^{i-1}|)$,其中 f_j^{i+1} 和 f_j^{i-1} 分别表示排序后个体 $i+1$ 和个体 $i-1$ 的第 j 个子目标的函数值。

经过上述快速支配强度排序法和新型拥挤距离计算,种群中每个个体 i 都具有 3 个属性:非支配集等级 $rank_i$ 、支配强度 ξ_i 和拥挤距离 i_D 。在非支配集中,首先比较非支配集等级,然后比较支配强度,最后比较拥挤距离。对于任意两个个体 i 和个体 j :若 $rank_i < rank_j$ 或 $rank_i = rank_j \cap \xi_i < \xi_j$ 或 $rank_i = rank_j \cap \xi_i = \xi_j \cap i_D > j_D$,则个体 i 优于个体 j 。

3.3 自适应精英保留策略

第二代多目标进化算法普遍采用精英保留策略来保留种群中优异的非支配个体^[15],从而使算法最大程度地逼近真实 Pareto 边界。NSGA2 算法的精英保留的规模通常是一个固定值,这显然是算法的设计缺陷。鉴于 NSGA2 算法中精英策略规模的选取较为困难,根据进化过程中非支配集的规模大小,本文采用了一种自适应精英保留的策略:在进化前期,限制种群中精英个体保留规模,增加下一代个体的多样性;在进化后期,逐步提高种群中精英个体保留的规模,以尽快收敛到 Pareto 边界,提升算法的性能。第 k 代进化过程中种群精英个体保留规模 $|Elitist|_k$ 的计算公式如下:

$$|Elitist|_k = |Pop| \times \alpha_k \quad (7)$$

其中, α_k 表示第 k 代进化过程中精英保留规模的影响因子, $|Pop|$ 为种群规模。 α_k 的自适应迭代计算公式如下:

$$\alpha_{k+1} = \alpha_k [1 + \ln(\rho + 1)] \quad (8)$$

其中, α_{k+1} 为第 $k+1$ 代进化过程的影响因子, ρ 表示种群中非支配集规模大小与种群规模大小的比值。在进化初期,非支配集规模较小,控制了精英个体保留的规模,算法可以搜索到更多新的随机个体;在进化后期,随着非支配集规模的逐代增大,精英个体保留的规模也不断增大,因此须使算法尽快收敛到 Pareto 边界。此外,为了防止精英个体保留的规模走向极端,当 $\alpha_k > 0.8$ 时,令 $\alpha_k = 0.8$;当 $\alpha_k < 0.2$ 时,令 $\alpha_k = 0.2$ 。

3.4 算法流程

基于支配强度的 NSGA2 改进算法的基本步骤如下:

Step1 随机初始化种群;

Step2 采用快速支配强度排序法对种群中的全部个体进行分级排序;

Step3 依据个体之间的非支配等级、支配强度和拥挤距离,根据自适应精英保留策略选择种群中的精英个体直接进入下一代种群;

Step4 采用模拟二进制交叉互换和多项式变异方法,由父代种群产生下一代的子代种群;

Step5 将父代种群中的精英个体与子代种群混合,组成下一代种群;

Step6 若达到进化终止条件,则停止循环,输出结果,否则转至 Step2。

4 算法测试及结果分析

为了验证基于支配强度的 NSGA2 改进算法 (INSGA2-DS) 的性能,将其与经典 NSGA2 进行对比数值实验仿真。本文选取了 11 个测试函数进行数值实验,包括 Deb 等于 1999 年提出的 ZDT 系列^[16]中的 ZDT1, ZDT2, ZDT3, ZDT6, 以及 Deb 等于 2005 年构造的 DTLZ 测试函数系列^[17]中的 DTLZ1—DTLZ7。

4.1 性能指标

为了更直观地比较算法在改进前后的性能差异,选取反向世代距离 (IGD) 作为算法的性能指标^[18]。该指标的计算公式如下:

$$IGD = \frac{\sum_{v \in P^*} d(v, P)}{|P^*|} \quad (9)$$

其中, P 为算法所求得 Pareto 边界的近似解集, P^* 是 MOP 真实 Pareto 边界上的个体集合, $|P^*|$ 表示 P^* 的个体数目, $d(v, P)$ 表示 P 中个体 v 与 P^* 的最小欧几里德距离。

IGD 是评价算法综合性能的指标,可以较为全面地反映算法的收敛性和分布性。IGD 越小,表示算法的计算结果与 MOP 真实的 Pareto 边界越接近,算法的收敛性和分布性越好。本文中,INSGA2-DS 和 NSGA2 对所有测试函数进行 30 次独立运算,取计算结果的 IGD 均值和标准差作为评价算法性能的指标。

4.2 实验参数

为比较两种算法在收敛性和分布性方面的差异,对 INSGA2-DS 和 NSGA2 均做如下设置:种群大小为 100,最大代数为 200,采用模拟二进制交叉和多项式变异的遗传操作,交叉参数为 20,交叉概率为 0.9,变异分布指数为 20,变异概率为 0.1,独立运行 30 次。值得说明的是,ZDT 系列测试函数的决策变量规模可任意改变,本文中 ZDT1 和 ZDT2 的变量规模为 30,ZDT3 和 ZDT6 的变量规模为 10;DTLZ 系列测试函数的目标个数都取为 3。

4.3 实验结果及分析

为了验证 INSGA2-DS 的性能,首先选择 NSGA2 和强度 Pareto 进化算法 (SPEA2) 与其进行比较,数值仿真结果如图 1—图 4 所示。

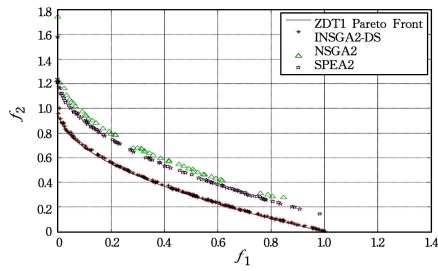


图1 ZDT1的测试结果

Fig. 1 Test results on ZDT1

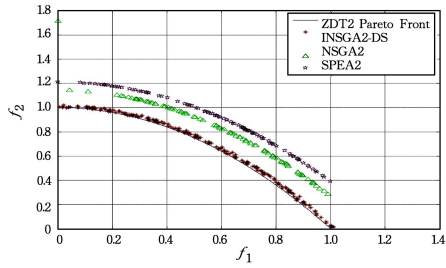


图2 ZDT2的测试结果

Fig. 2 Test results on ZDT2

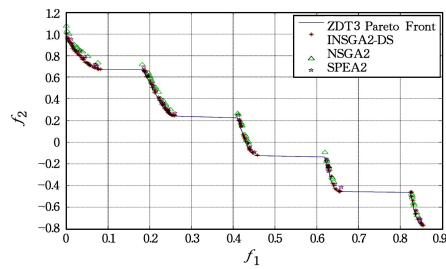


图3 ZDT3的测试结果

Fig. 3 Test results on ZDT3

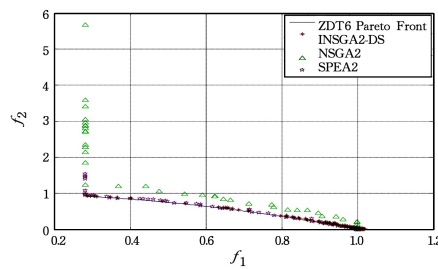


图4 ZDT6的测试结果

Fig. 4 Test results on ZDT6

ZDT系列中测试函数的特点是决策变量的数目较大,本文将其用于比较算法在求解多维决策变量时的MOP的优化性能,其中ZDT1为凸形Pareto边界,ZDT2和ZDT6为凹形Pareto边界,ZDT3为不连续Pareto边界^[19]。由图1—图4可知,求解ZDT3问题时,INSGA2-DS,NSGA2和SPEA2都可以较好地收敛到真实的Pareto边界;求解ZDT1和ZDT2问题时,INSGA2-DS所得结果明显优于NSGA2和SPEA2所得结果;求解ZDT6问题时,NSGA2获得最多的伪非支配解,其解集的收敛性和分布性均最差。总之,尽管INSGA2-DS最终收敛到真实的Pareto边界,但其在求解ZDT6问题时

所获得的解集的收敛性和分布性均不理想。

DTLZ系列是一组通过自底向上或对曲面进行约束的方法构造出规模可变的多元目标测试函数,由Del等等于2005年提出^[14]。本文采用函数DTLZ1—DTLZ7来检验INSGA2-DS求解Pareto边界为非凸、非线性、离散且分布不均匀时的MOP的全局优化能力和处理约束的能力。如图5—图11所示,图中浅色点构成的区域是测试函数的真实Pareto边界,INSGA2-DS所求结果(图中深点)较为均匀地分布在真实Pareto边界区域中(图中浅色区域);INSGA2-DS在求解DTLZ6函数时的收敛性和分布性均较差。

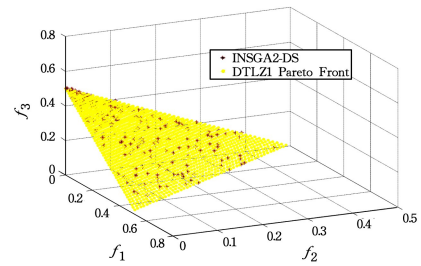


图5 DTLZ1的测试结果

Fig. 5 Test results on DTLZ1

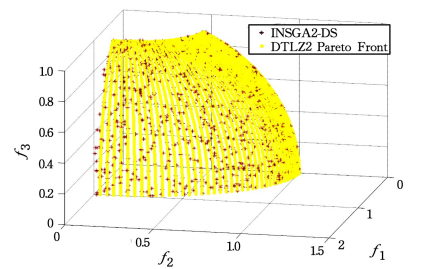


图6 DTLZ2的测试结果

Fig. 6 Test results on DTLZ2

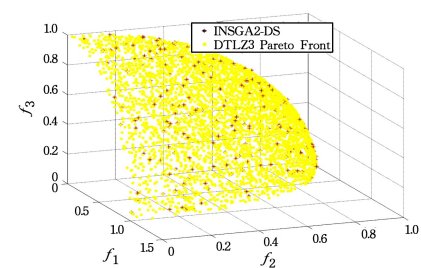


图7 DTLZ3的测试结果

Fig. 7 Test results on DTLZ3

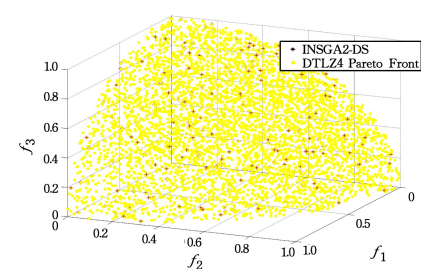


图8 DTLZ4的测试结果

Fig. 8 Test results on DTLZ4

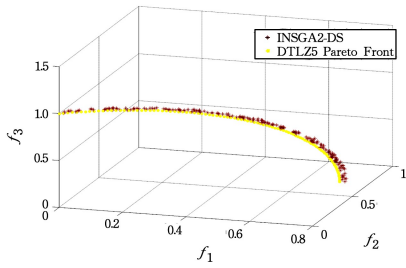


图 9 DTLZ5 的测试结果

Fig. 9 Test results on DTLZ5

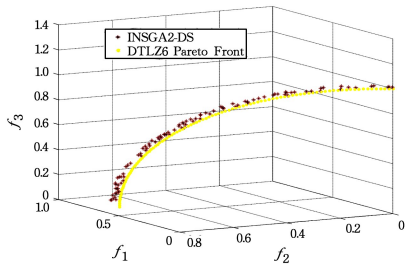


图 10 DTLZ6 的测试结果

Fig. 10 Test results on DTLZ6

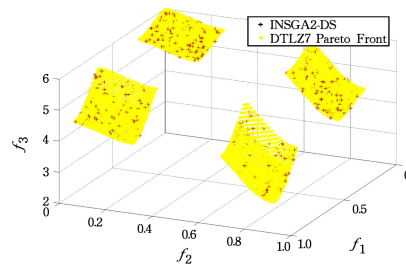


图 11 DTLZ7 的测试结果

Fig. 11 Test results on DTLZ7

为了进一步定量比较 INSGA2-DS 和 NSGA2 之间的性能差异,统计了两种算法在求解测试函数结果时的反向世代距离 IGD 。两种算法均对每个测试函数进行 30 次计算,统计其反向世代距离的平均值和标准差,结果如表 1 所列。表中较小值被加粗显示。此外,为更清晰地显示两种算法的计算结果,还利用了统计分析中常用的箱线图进行展示。由于篇幅限制,仅给出 ZDT1,ZDT3,ZDT6,DTLZ1,DTLZ2,DTLZ7 计算结果的 IGD 箱线图,其中中间水平线表示中位数,上下两条线分别表示四分位数。

表 1 算法的 IGD 的统计结果
Table 1 IGD statistical results

MOP	均值		标准差	
	NSGA2	INSGA2-DS	NSGA2	INSGA2-DS
ZDT1	1.67E-01	6.61E-05	7.64E-03	5.25E-04
ZDT2	8.16E-02	1.02E-03	3.21E-04	2.98E-05
ZDT3	5.51E-05	6.76E-05	4.12E-02	9.61E-03
ZDT6	6.70E-01	1.07E-02	1.07E-01	6.68E-02
DTLZ1	2.14E-02	9.94E-03	9.04E-02	4.42E-03
DTLZ2	3.37E-01	8.01E-02	6.73E-01	2.13E-02
DTLZ3	9.08E-02	3.40E-03	4.89E-02	1.65E-02
DTLZ4	6.66E-02	8.51E-02	5.27E-02	8.14E-02
DTLZ5	4.47E-01	9.11E-02	1.06E-01	4.56E-02
DTLZ6	1.01E-02	7.99E-01	1.28E-02	6.30E-02
DTLZ7	2.50E-01	4.31E-02	8.19E-01	9.74E-03

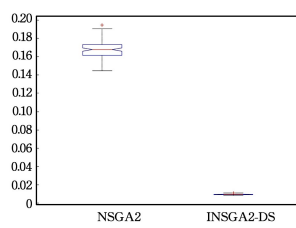


图 12 ZDT1 的 IGD 箱线图
Fig. 12 IGD boxplot of ZDT1

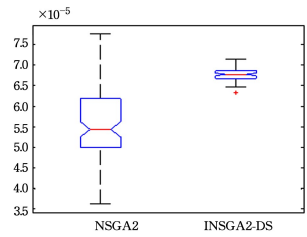


图 13 ZDT3 的 IGD 箱线图
Fig. 13 IGD boxplot of ZDT3

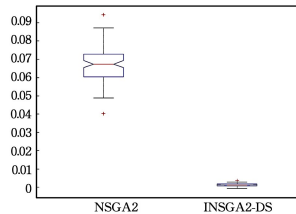


图 14 ZDT6 的 IGD 箱线图
Fig. 14 IGD boxplot of ZDT6

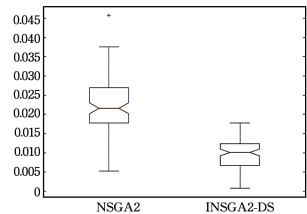


图 15 DTLZ1 的 IGD 箱线图
Fig. 15 IGD boxplot of DTLZ1

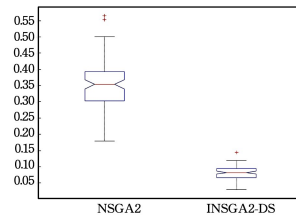


图 16 DTLZ2 的 IGD 箱线图
Fig. 16 IGD boxplot of DTLZ2

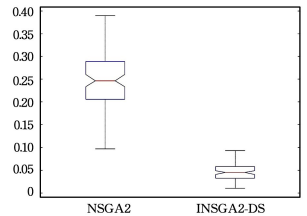


图 17 DTLZ7 的 IGD 箱线图
Fig. 17 IGD boxplot of DTLZ7

IGD 的均值反映了算法的收敛性和分布性,而标准差可以表征算法的鲁棒性。由表 1 可知,除了求解测试函数 ZDT3,DTLZ4,DTLZ6,INSGA2-DS 算法求解其他测试函数所得结果的 IGD 的均值均小于 NSGA2,即 INSGA2-DS 的收敛性和分布性总体优于 NSGA2;而对于 IGD 方差的统计结果,除了测试函数 DTLZ4,DTLZ6,INSGA2-DS 算法所求结果的方差均小于 NSGA2,因此 INSGA2-DS 具有较强的鲁棒性,算法稳定性明显优于 NSGA2。箱线图能显示测试函数 IGD 统计数据的分散情况,由图 12—图 17 可知,INSGA2-DS 的计算结果中位数较小,数据分布较为集中,其解集的质量总体上优于 NSGA2。INSGA2-DS 在测试函数 DTLZ4 和 DTLZ6 上的表现欠佳,主要由于自适应精英保留策略在求解某些问题时,进化初期非支配个体较多,每次进化过程中精英保留的规模较大,收敛过快,导致陷入局部优化,最终降低了算法的解集质量。但总体来讲,对于大部分测试函数,INSGA2-DS 的优化性能优于经典 NSGA2。

结束语 本文提出一种基于支配强度的 NSGA2 改进算法。INSGA2-DS 通过采用非支配强度来比较非支配集中个体的优劣性,可有效地识别和去除伪非支配解;改进后的快速支配强度排序方法所需的总比较次数较少,减少了分层排序的时间,提高了算法的计算速度;引入一个考虑方差的新型拥挤距离计算公式,提高了在各个子目标上差异程度较大的个体的存活概率,增加了算法的多样性和分布性;采用自适应精英保留策略,避免了精英保留规模选取比较困难的问题;根据

种群中非支配集的大小,自动调整父辈精英个体保留的规模,提高了算法的收敛性。数值实验结果表明,INSGA2-DS有效弥补了NSGA2的设计缺陷,显著提高了算法求解复杂、非线性多目标优化问题时的收敛性、分布性和计算效率。

未来将继续改善 INSGA2-DS 的性能,并将其应用于求解实际工程设计和科学计算领域的多目标优化问题的求解中,以进一步验证算法的实用性。

参考文献

- [1] ZITZLER E, DEB K, THIELE L. Comparison of multiobjective evolutionary algorithms: Empirical results [J]. *Evolutionary Computation*, 2000, 8(2): 173-195.
- [2] SRINIVAS N, DEB K. Multiobjective optimization using nondominated sorting in genetic algorithms[J]. *Evolutionary Computation*, 1994, 2(3): 221-248.
- [3] 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.
- [4] KOLLAT J B, REED P M. The value of online adaptive search: a performance comparison of NSGAII, ϵ -NSGAII and ϵ MOEA [C]// *International Conference on Evolutionary Multi-Criterion Optimization*. Springer Berlin Heidelberg, 2005: 386-398.
- [5] ZHANG Q, LI H. MOEA/D: A multiobjective evolutionary algorithm based on decomposition[J]. *IEEE Transactions on Evolutionary Computation*, 2007, 11(6): 712.
- [6] 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.
- [7] TAN K C, CHIAM S C, MAMUN A A, et al. Balancing exploration and exploitation with adaptive variation for evolutionary multi-objective optimization[J]. *European Journal of Operational Research*, 2009, 197(2): 701-713.
- [8] ELHOSSINI A, AREIBI S, DONY R. Strength Pareto particle swarm optimization and hybrid EA-PSO for multi-objective optimization[J]. *Evolutionary Computation*, 2010, 18(1): 127-156.
- [9] XIE C W, LI K, LIAO G Y. Improved NSGA2 Algorithm with Differential Evolutionary Local Search[J]. *Computer Science*, 2013, 40(10): 235-238, 273.
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, 273.
- [10] 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 Trans. Evolutionary Computation*, 2014, 18(4): 577-601.
- [11] SHIM V A, TAN K C, TANG H. Adaptive memetic computing for evolutionary multiobjective optimization[J]. *IEEE Transactions on Cybernetics*, 2015, 45(4): 610-621.
- [12] QIU X, XU J X, TAN K C, et al. Adaptive Cross-Generation Differential Evolution Operators for Multiobjective Optimization [J]. *IEEE Transactions on Evolutionary Computation*, 2016, 20(2): 232-244.
- [13] COELLO C A C, VAN VELDHIJZEN D A, Lamont G B. *Evolutionary algorithms for solving multi-objective problems*[M]. New York: Kluwer Academic, 2002.
- [14] 郑金华. 多目标进化算法及其应用[M]. 北京: 科学出版社, 2007: 109-112.
- [15] DEB K. *Multi-objective optimization using evolutionary algorithms*[M]. New York: John Wiley & Sons, 2001.
- [16] DEB K. Multi-objective genetic algorithms: Problem difficulties and construction of test problems[J]. *Evolutionary computation*, 1999, 7(3): 205-230.
- [17] DEB K, THIELE L, LAUMANN S M, et al. *Scalable test problems for evolutionary multi objective optimization*[M]. London: Springer, 2005.
- [18] 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.
- (上接第 175 页)
- [8] ZHU H J, WANG J J, WANG B, et al. Location privacy preserving obstructed nearest neighbor queries[J]. *Journal of Computer Research and Development*, 2014, 51(1): 115-125.
- [9] ZHU H, YANG X, WANG B, et al. Range-based obstructed nearest neighbor queries[C]// *2016 International Conference on Management of Data (SIGMOD'16)*, San Francisco, California, USA, 2016. New York, NY, USA: ACM, 2016: 2053-2068.
- [10] HAO Z X, WANG Y D, HE Y B. Line segment nearest neighbor query of spatial database[J]. *Journal of Computer Research and Development*, 2008, 45(9): 1539-1545. (in Chinese)
郝忠孝, 王玉东, 何云斌. 空间数据库平面线段最近邻查询问题研究[J]. *计算机研究与发展*, 2008, 45(9): 1539-1545.
- [11] LIU R T, HAO Z X. Fast algorithm of nearest neighbor query for line segments of spatial database[J]. *Journal of Computer Research and Development*, 2011, 48(12): 2379-2384. (in Chinese)
刘润涛, 郝忠孝. 空间数据库平面线段快速最近邻查询算法[J]. *计算机研究与发展*, 2011, 48(12): 2379-2384.
- [12] ZHOU Y, YANG Z X. Research on line segment kNN query in spatial database[J]. *Computer Engineering and Applications*, 2015, 51(18): 131-134. (in Chinese)
周屹, 杨泽雪. 空间数据库中的线段 k 近邻查询研究[J]. *计算机工程与应用*, 2015, 51(18): 131-134.
- [13] GU Y, ZHANG H, WANG Z, et al. Efficient moving k nearest neighbor queries over line segment objects[J]. *World Wide Web*, 2016, 19(4): 653-677.
- [14] PAPAPOPOULOU E, ZAVERSHYNSKYI M. The higher-order Voronoi diagram of line segments[J]. *Algorithmica*, 2014, 74(1): 1-25.
- [15] Dataset[EB/OL]. <http://konect.uni-koblenz.de/networks/road-Net-CA>, 2016.