

一种基于密度的局部搜索NSGA2算法

栗三一[†], 李文静, 乔俊飞

(1. 北京工业大学 自动化学院, 北京 100124; 2. 计算智能与智能系统北京市重点实验室, 北京 100124)

摘要: 针对局部搜索类NSGA2算法计算量大的问题, 提出一种基于密度的局部搜索NSGA2算法(NSGA2-DLS). 使用解的密度衡量解的稀疏度, 并将当前非支配解中稀疏度最小的解定义为稀疏解, 每次遗传过程在稀疏解周围进行局部搜索. 在局部搜索过程中, 同时采用极限优化策略和随机搜索策略以提高解的质量和收敛速度. 对ZDT系列函数和DTLZ系列函数进行仿真实验并与标准NSGA2算法、一种局部随机搜索算法和一种定向搜索算法进行比较, 结果表明, NSGA2-DLS在消耗计算量和优化效果方面均优于对比方法.

关键词: NSGA2; 密度; 局部搜索; 多目标优化

中图分类号: TP301

文献标志码: A

A local search strategy based on density for NSGA2 algorithm

LI San-yi[†], LI Wen-jing, QIAO Jun-fei

(1. College of Automation, Beijing University of Technology, Beijing 100124, China; 2. Beijing Key Laboratory of Computational Intelligence and Intelligent System, Beijing 100124, China)

Abstract: In order to reduce the amount of calculation and keep the advantage of local search strategy simultaneously, this paper proposes a kind of local search method based on density for the NSGA2(NSGA2-DLS) algorithm. Firstly, the sparse degree of each solution is evaluated with the density of each solution in the solution space. Then the non-dominated solution with the smallest sparse degree is defined as the sparse solution, and the sparse solution is searched around locally during every genetic process. The NSGA2-DLS algorithm adopts extreme optimization strategy and random search strategy simultaneously to improve the quality of solutions and convergence rate. The performance of the NSGA2-DLS algorithm is compared with the performance of the baseline NSGA2 algorithm and two other reported local search NSGA2 algorithms for multi-objective test problems, including ZDT and DTLZ functions. The simulation results show that the solution quality and the calculation amount of the NSGA2-DLS algorithm are much better than that of other methods.

Keywords: NSGA2; density; local search; multi-objective optimization

0 引言

实际工程中通常存在一些相互冲突的工程目标, 这类工程问题可以被视为一个多目标优化问题(MOP). 由于很多实际多目标优化问题的理想最优解集(帕累托解集)是凹的或非连续的, 传统的数学规划方法无法解^[1], 在一次学习中可以得到一组解的多目标进化算法受到广泛关注.

在解决MOP问题的过程中, 人们提出很多经典进化算法, 其中Deb等^[2]提出了NSGA2算法是目前最优秀的多目标进化算法之一. NSGA2算法引入精

英策略, 采用快速非支配排序方法, 并结合拥挤度比较算子选择胜出解. 但NSGA2算法作为一种类随机搜索算法, 存在收敛速度慢(遗传操作次数多)和解分布特性(广泛性和均匀性)较差的问题^[3].

局部搜索策略可以有效提高NSGA2算法的收敛速度和分布特性, 研究者已提出多种基于局部搜索策略的改进型NSGA2算法. Deb等^[4]首次针对多目标进化算法提出爬山局部搜索方法, 使用加权求和的方式形成复合目标函数, 使解向帕累托前沿快速靠近, 并且有更好的分布性. Lara等^[5]提出一种混

收稿日期: 2016-10-16; 修回日期: 2016-12-19.

基金项目: 国家自然科学基金重点项目(61533002); 国家杰出青年科学基金项目(61225016); 国家自然科学基金青年基金项目(61603009); 中国博士后科学基金项目(2015M570910); 北京工业大学基础研究基金项目(002000514315501); 朝阳区博士后工作经费项目(2015ZZ-6).

作者简介: 栗三一(1988-), 男, 博士生, 从事优化控制的研究; 乔俊飞(1968-), 男, 教授, 博士生导师, 从事智能优化控制等研究.

[†]通讯作者. E-mail: wslisanyi@126.com

合局部搜索算法,该方法使用爬山法进行局部搜索以提高收敛速度和精度,使用偏移法进行横向搜索,从而增加种群分布的广泛性. Ishibuchi等^[6]针对组合优化问题提出多目标遗传局部搜索算法(MOGLS),该方法对每一次遗传操作的每一个子代都使用复合目标函数进行评价,复合函数的权值随机选取,但每一次遗传操作复合函数的权值是固定的. Jaskiewicz^[7]改进了MOGLS方法,在原方法的基础上加入父代选择约束,提高了子代质量. 与此类似, Sindhya等^[8]提出了一种标量函数法,该方法首先定义一个参考点,然后建立一个复合函数,在优化过程中使得该函数值最小,其目的是使当前种群尽量远离参考点,向Pareto前沿靠近,提高了算法的收敛精度和收敛速度. 在此基础上, Palar等^[9]将标准化的上界作为参考点,从而扩大了搜索空间,提高了解分布的广泛性. Harada等^[10]首次将梯度信息用于局部搜索,提出一种基于梯度信息的局部搜索算法,根据梯度信息找到向Pareto前沿收敛的方向,从而提高收敛速度和精度. 但是,梯度的计算耗费大量计算资源,因此Kim等^[11]从邻近解获得的信息推导出局部搜索的方向,避免了梯度计算,且设定自适应参数来平衡局部搜索与全局搜索之间的关系,是目前最好的定向搜索算法之一. 以上这些局部搜索算法在提高NSGA2算法的收敛速度和种群分布特性方面取得了一定成果. 然而,目前所有局部搜索算法存在着共同的缺点^[9],在局部搜索过程中产生大量局部解,使得遗传过程的计算量成倍增加.

为解决当前基于局部搜索的NSGA2算法计算量大的问题,本文受文献[12]的启发,提出一种基于密度的局部搜索NSGA2算法(NSGA2-DLS). 文献[12]针对单目标动态优化问题提出基于记忆的精英移民策略,每次学习过程中在当前精英解(最优解)周围形成精英移民,参与到下代竞争中,该机制有效提高了单目标优化算法的收敛速度和种群多样性. 将精英移民思想应用于多目标优化算法,每次遗传操作只在一个解周围进行局部搜索,可以大量减少局部解的数量,从而减少计算量. 然而,对多目标优化问题而言,最优解不只是一个,而且需要确保解的分布特性,因此本文提出的方法使用目标空间中解的密度衡量一个解的稀疏度,将当前非支配解中密度最小的解作为稀疏解,使用两种变异策略在稀疏解周围进行局部搜索: 1) 使用文献[1]中的极限优化变异策略产生新种群,从而增加解的精度; 2) 使用局部随机搜索策略产生新种群从而加快算法的收敛速度. 实验结果表

明,该方法有效提高了NSGA2算法的收敛速度和解的分布特性,并且NSGA2-DLS在有限函数调用次数(计算量评价指标)内得到的解的质量明显优于其他算法,评价指标达到设定值所消耗的函数调用次数明显少于其他算法.

1 基本概念

1.1 MOP及相关定义

定义1 (MOP问题) 不受约束的多目标极小化问题可表示为

$$\begin{aligned} \min F(X) &= (f_1(X), f_2(X), \dots, f_m(X)), \\ X &= (x_1, x_2, \dots, x_n), \\ l_i &\leq x_i \leq u_i, \quad i = 1, 2, \dots, n. \end{aligned} \quad (1)$$

其中: $X = (x_1, x_2, \dots, x_n) \in \Omega$ 为 n 维决策向量, Ω 为决策空间; l_i 和 u_i 分别为第 i 个决策变量的下界和上界; $F(X) \in R^m$ 为 m 维目标向量; $f_j(X)$ ($j = 1, 2, \dots, m$) 为第 j 个目标函数; R^m 为目标函数空间.

定义2 (Pareto支配或Pareto占优) 对于任意决策向量 $u, v \in \Omega$, 若满足以下条件:

$$\begin{cases} f_i(u) \leq f_i(v), \quad \forall i \in 1, 2, \dots, m; \\ f_j(u) < f_j(v), \quad \exists j \in 1, 2, \dots, m. \end{cases} \quad (2)$$

则称 u 支配 v , 或称 u 较 v 占优, 记为 $u \prec v$.

定义3 (Pareto最优解) 在决策空间 Ω 中, 若对于任意解 x , 不存在 $x' \in \Omega$ 使得 $F(x') = (f_1(x'), f_2(x'), \dots, f_m(x'))$ 占优于 $F(x) = (f_1(x), f_2(x), \dots, f_m(x))$, 则称 x 为Pareto最优解或非劣解.

定义4 (Pareto最优解集) 对于给定的MOP, 所有Pareto最优解的集合称为Pareto最优解集.

定义5 (Pareto最优前沿) 对于给定的MOP, 所有Pareto最优解在目标空间中对应的目标向量的集合称为Pareto最优前沿.

1.2 局部搜索NSGA2算法

NSGA2算法与局部搜索NSGA2算法的种群操作流程如图1所示. NSGA2算法的主要框架为, 父代进行交叉、变异操作形成子代, 然后通过非支配排序和拥挤距离排序从父代和子代中挑选优秀个体保留. 局部搜索NSGA2算法在挑选优秀个体之前对每个父代都进行局部搜索操作, 产生局部解集, 然后在父代、子代和局部解集中挑选优秀个体保留. 不同的局部搜索NSGA2算法仅在具体局部搜索操作上有所不同, 因为每次局部搜索操作都在每个父代个体周围产生局部解, 局部解个数数倍于父代个体数, 所有局部解都要调用目标函数进行计算, 所以局部搜索类NSGA2算法存在计算量大的问题.

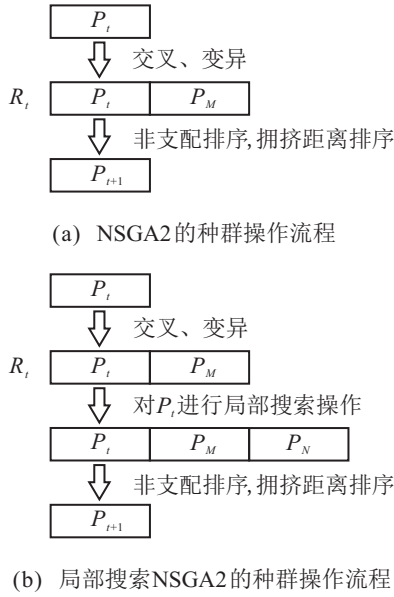


图1 种群操作流程

2 NSGA2-DLS算法

本文将单目标优化中的精英移民策略用于解决MOP问题,提出一种基于密度的局部搜索多目标优化算法NSGA2-DLS. NSGA2-DLS使用解的密度衡量种群稀疏度,在稀疏解周围进行局部搜索以保证解的均匀性和广泛性;采用极限优化方法进行局部搜索,以提高算法的收敛精度;采用随机搜索方法进行局部搜索,以提高算法的收敛速度;局部搜索过程中只在稀疏解周围产生新种群,较目前的局部搜索算法计算量明显减少.

2.1 求取稀疏度

传统NSGA2算法使用的拥挤距离方法并不能完全保证种群的均匀性,因此NSGA2-DLS使用解的密度衡量一个解的稀疏度(SP). 设种群总数为 N ,即有 N 个解,首先对目标函数值进行归一化. 设第 i 个解 X^i 的目标向量为 $(f_1(X^i), \dots, f_m(X^i))$,归一化公式^[13]为

$$\bar{f}_j(X^i) = \frac{f_j(X^i) - f_{j \min}}{f_{j \max} - f_{j \min}}, \quad (3)$$

其中 $f_{j \min}$ 和 $f_{j \max}$ 分别为当前所有解对应的第 j 个目标函数值的最大值和最小值. 归一化后第 i 个解 X^i 的稀疏度计算公式如下:

$$SP(X^i) = \frac{n_i}{N}, \quad (4)$$

其中 n_i 为目标函数空间中与目标向量 $F(X^i)$ 欧氏距离小于 r 的其他目标向量的个数, r 的取值范围为 $0 < r < 1$,本文所有实验取0.1.

2.2 局部搜索

NSGA2-DLS将当前非支配解中稀疏度最小的解作为稀疏解. 该稀疏解的设定使NSGA2-DLS具有

以下优势:当解分布不均匀时,该方法有效提高了稀疏解周围空间的探索,从而使得解分布均匀;当非支配解分布均匀时,解集两端的解稀疏度最小,算法开始向边缘以外探索,从而增加解的广泛性.

稀疏解设定后,本文使用两种变异策略同时产生局部解. 首先使用文献[1]中的极限优化策略在稀疏解周围产生新种群. 极限优化变异方法在产生局部解时,每个局部解只变动稀疏解的一个决策变量. 极限优化策略提高了种群接近Pareto最优解时的局部搜索能力,有效提高了解的精度. 具体方法如下:设当前稀疏解为 $X = (x_1, x_2, \dots, x_n)$, n 为决策变量个数,种群总数为 N ,则产生局部解个数为 n ,产生局部解的变异公式为

$$X_i = (x_1, \dots, x'_i, \dots, x_n), \quad 0 < i \leq n; \quad (5)$$

$$x'_i = x_i + \alpha \cdot \beta_{\max}(x_i), \quad 0 < i \leq n; \quad (6)$$

$$\alpha = \begin{cases} (2h)^{(1/(q+1))} - 1, \\ 1 - [2(1-h)]^{(1/(q+1))}; \end{cases} \quad (7)$$

$$\beta_{\max}(x_i) = \max[x_i - l_i, u_i - x_i], \quad 0 < i \leq n. \quad (8)$$

其中: x_i 为决策变量; h 为 $0 \sim 1$ 之间的随机数; q 为正实数,称为形状参数,本文将 q 设为11; $\beta_{\max}(x_i)$ 为当前决策变量 x_i 可变动的最大值. 极限优化变异方法每次只改变一个决策变量,具有很强的微调能力,但搜索范围较小. 为了增加算法的收敛速度,同时采用第2种变异策略,设当前稀疏解为 $X = (x_1, x_2, \dots, x_n)$, n 为决策变量个数,种群总数为 N ,则产生局部解个数为种群总数的20%^[12](如不能整除则取整),变异公式为

$$X_k = (x'_1, \dots, x'_i, \dots, x'_n), \quad k = 1, 2, \dots, [0.2N]; \quad (9)$$

$$x'_i = \gamma x_i, \quad i = 1, 2, \dots, n, \quad 0 < \gamma < 1.2, \quad (10)$$

其中 γ 为 $0 \sim 1.2$ 之间的随机数. 以上两种变异策略共产生 $n + [0.2N]$ 个局部解.

2.3 NSGA2-DLS算法流程

根据实际MOP问题设定算法参数:初始种群数量 N ,最大迭代次数或最大函数调用次数 I_{\max} ,决策向量维数 n ,决策变量取值上界 $u = (u_1, u_2, \dots, u_n)$ 和下界 $l = (l_1, l_2, \dots, l_n)$,目标函数个数 m ,形状参数 q ,交叉概率 p_c 和变异概率 p_m ,交叉参数 η_c 和变异参数 η_m . NSGA2-DLS的具体算法流程如下.

Step 1: 在取值范围内随机初始化种群 $P_T = \{X^1, X^2, \dots, X^N\}$,其中 $X^i = (x'_1, x'_2, \dots, x'_n)$, $i = 1, 2, \dots, n$.

Step 2: 计算 P_I 中所有解的适应度值和拥挤距离 (与标准 NSGA2 相同, 本文不介绍详细过程), 对 P_I 中的解进行非支配排序, 当前种群中所有非支配解记为 P_C .

Step 3: 按照标准 NSGA2 算法对 P_I 中的种群进行交叉变异, 形成子代 P_M .

Step 4: 在目标函数空间内, 按式 (3) 和 (4) 计算 P_C 中所有解的稀疏度, 选择稀疏度最小的解作为稀疏解 $X = (x_1, x_2, \dots, x_n)$.

Step 5: 按式 (5) ~ (10) 形成 $n + 0.2N$ 个局部解, 所有局部解组成种群 P_N , P_N 参与到下一代的竞争中.

Step 6: 将 P_I 、 P_N 和 P_M 合并, 并对所有解进行非支配排序和拥挤距离计算, 从中选取最优的 N 个解形成下一代种群 P_O , 并设 $P_I = P_O$. P_O 的选取规则为: 优先选取非支配解, 当非支配解数量不足 N 时, 选取次优解, 以此类推, 直到解的数量达到 N 为止. 在同一等级的解中, 拥挤距离大的解优先选取.

Step 7: 重复 Step 2 ~ Step 6, 当达到预设目标或最大迭代步数 I_{\max} 时进行下一步.

Step 8: 当前 P_I 中的非支配解即为得到的最优解.

与其他局部搜索 NSGA2 算法相比, NSGA2-DLS 算法仅在稀疏解周围进行局部搜索, 单步计算量明显减少; 极限优化策略确保了解的精度; 局部搜索策略有效保证了算法的收敛速度; 在稀疏解周围进行局部搜索有效保持了种群的分布特性.

3 仿真实验

实验 1 与实验 2 使用双目标 ZDT 系列函数 (ZDT1 ~ ZDT4, ZDT6), 实验 3 使用三目标 DTLZ 系列函数 (DTLZ1, DTLZ2) 进行实验, ZDT 系列函数和 DTLZ 系列函数均被广泛应用于验证多目标优化算法^[3-9]. 实验结果与标准 NSGA2^[2]、随机局部搜索算法 HMOEA/D^[8] 和定向搜索算法 NSGA2-cls^[11] 进行对比, 测试函数如表 1 所示. 本文采用 Matlab 10.0 b 软件进行仿真实验.

表 1 测试函数

测试问题	目标函数	约束条件和特征
ZDT1	$f_1(x_1) = x_1$ $f_2(x) = g(1 - \sqrt{(f_1/g)})$ $g = 1 + \sum_{i=2}^n x_i / (n - 1)$	$n = 30, 0 \leq x_i \leq 1$ 真实Pareto凸
ZDT2	$f_1(x_1) = x_1$ $f_2(x) = g(1 - (f_1/g)^2)$ $g = 1 + \sum_{i=2}^n x_i / (n - 1)$	$n = 30, 0 \leq x_i \leq 1$ 真实Pareto凹
ZDT3	$f_1(x_1) = x_1$ $f_2(x) = g(1 - \sqrt{(f_1/g)} - (f_1/g) \sin(10\pi f_1))$ $g = 1 + \sum_{i=2}^n x_i / (n - 1)$	$n = 30, 0 \leq x_i \leq 1$ 真实Pareto凸且非连续
ZDT4	$f_1(x_1) = x_1$ $f_2(x) = g(1 - \sqrt{(f_1/g)})$ $g = 1 + 10(n - 1) + \sum_{i=2}^n (x_1^2 - 10 \cos(4\pi x_i))$	$n = 10, 0 \leq x_1 \leq 1$ $-5 \leq x_i \leq 5, i = 2, 3, \dots, n$ 真实Pareto凸
ZDT6	$f_1(x_1) = 1 - \exp(-4x_1) \sin^6(6\pi x_1)$ $f_2(x) = g(1/(f_1/g)^2)$ $g = 1 + 9 \left(\sum_{i=2}^n x_i / (n - 1) \right)^{0.25}$	$n = 10, 0 \leq x_i \leq 1$ 真实Pareto凹
DTLZ1	$f_1(x) = 0.5(1 + g)x_1x_2$ $f_2(x) = 0.5(1 + g)x_1(1 - x_2)$ $f_3(x) = 0.5(1 + g)(1 - x_1)$ $g = 100 \left[x_m + \sum_{x_i \in x_m} ((x_i - 0.5)^2 - \cos(20\pi(x_i - 0.5))) \right]$	$n = 10, 0 \leq x_i \leq 1$ $i = 1, 2, \dots, n$ $x_m, m = 3, 4, \dots, n$
DTLZ2	$f_1(x) = (1 + g) \cos(x_1\pi/2) \cos(x_2\pi/2)$ $f_2(x) = (1 + g) \cos(x_1\pi/2) \sin(x_2\pi/2)$ $f_3(x) = (1 + g) \sin(x_1\pi/2)$ $g = \sum_{x_i \in x_m} (x_i - 0.5)^2$	$n = 10, 0 \leq x_i \leq 1$ $i = 1, 2, \dots, n$ $x_m, m = 3, 4, \dots, n$

对实验1和实验2设初始种群规模 N 为100,实验3的种群规模与所对比原文献相同;决策变量个数 n 和决策变量取值范围如表1所示;所有实验均采用模拟二进制交叉变异方法,交叉参数 η_c 和变异参数 η_m 均设为20,交叉和变异概率分别为0.9和 $1/n$;形状参数 q 设为11. 评价函数采用综合评价指标(IGD)^[11]和种群多样性指标 I ^[2],IGD值越小则解的质量越好, I 越小则解的分布特性越好. IGD计算公式如下:

$$\text{IGD}(P^*, P) = \frac{\sum_{v \in P^*} d(v, P)}{|P^*|}. \quad (11)$$

其中: P^* 为帕累托解集, P 为近似解, $d(v, P)$ 为向量 v 与解集 P 中的向量的欧氏距离最小值, $|P^*|$ 为帕累托解集中解的个数. 指标 I 计算公式如下:

$$I = \frac{d_f + d_l + \sum_{i=1}^{N-1} |d_i - d_m|}{d_f + d_l + (N-1)d_m}. \quad (12)$$

其中: d_f 和 d_l 为帕累托末端解和所得解集边界解之间的欧氏距离, $d_i (i = 1, 2, \dots, N-1)$ 为所获得的连续非支配解之间的欧氏距离, d_m 为所有 d_i 的平均值.

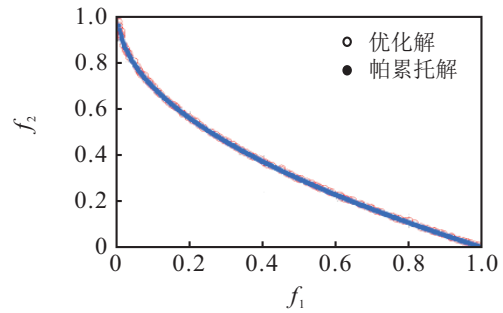
3.1 实验1

本实验对ZDT系列函数进行实验,最大计算量设为 $I_{\max} = 5000$ (NSGA2-DLS算法在5000次函数调用时已经达到与对比论文相当的实验结果,而对比文献中函数调用次数分别为20000和30000),即对于种群规模为100的标准NSGA2而言,一次遗传操作通过交叉变异产生50个子代,由于父代不需要重复计算函数值,一次遗传操作的计算量为50,共进行100次遗传操作. 实验目的是检验在计算量相同时各算法的优化效果.

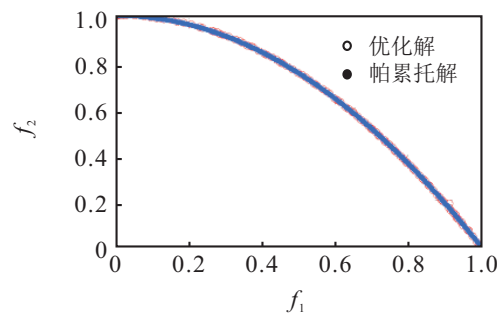
采用NSGA2-DLS算法的仿真实验结果如图2所示. 由图2可见,对Pareto前沿凹、凸和非连续的双目标函数,NSGA2-DLS都可以很好地逼近Pareto前沿,而且解的分布十分均匀.

NSGA2-DLS的实验结果与其他算法的结果对比如表2和表3所示. 表2和表3分别列出了函数调用次数为5000时各算法所得种群的IGD值和 I 值,实验结果包括连续30次实验的最大值、最小值和平均值. 由表2可见,在函数调用次数为5000时,对于5个ZDT测试函数,NSGA2-DLS算法解得的IGD值的最大值、最小值和平均值均小于对比方法. 在ZDT1~ZDT4的实验结果中,NSGA2-DLS的平均IGD值比对比算法的结果小1到2个数量级,在ZDT6中,NSGA2-DLS的IGD值也是最小的. 因此NSGA2-

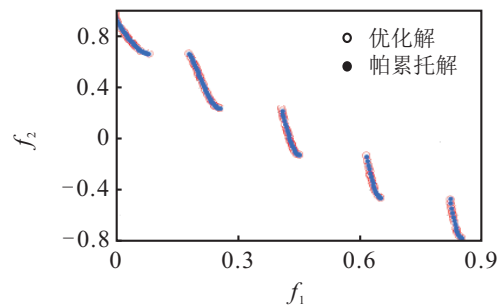
DLS算法在有限函数调用次数内得到的解的综合质量更高.



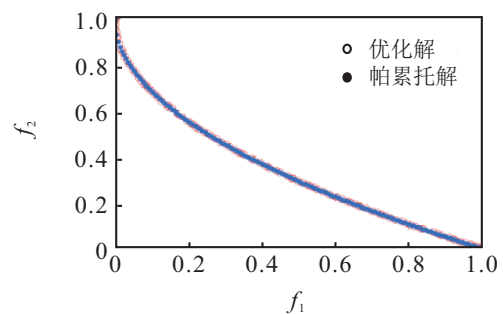
(a) ZDT1优化效果



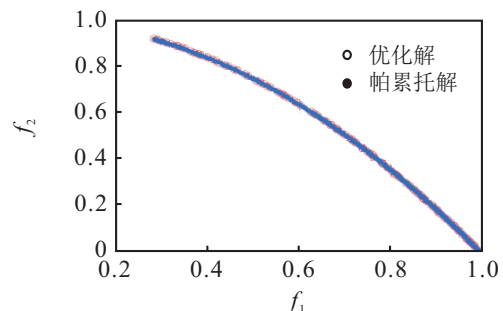
(b) ZDT2优化效果



(c) ZDT3优化效果



(d) ZDT4优化效果



(e) ZDT6优化效果

图2 函数调用5000次时NSGA2-DLS算法的优化结果

表2 NSGA2-DLS与其他局部搜索算法的ZDT系列实验IGD结果(连续30次实验)

	算法	函数调用次数	最大值	最小值	平均值
ZDT1	NSGA2-DLS	5 000	0.009 7	0.004 2	0.006 4
	NSGA2 ^[2]	5 000	1.025 5	0.318 2	0.655 5
	HMOEA/D ^[8]	5 000	0.131 8	0.084 8	0.106 7
	NSGA2-cls ^[11]	5 000	0.578 5	0.006 9	0.195 1
ZDT2	NSGA2-DLS	5 000	0.007 0	0.004 2	0.005 0
	NSGA2 ^[2]	5 000	0.457 8	0.030 3	0.179 2
	HMOEA/D ^[8]	5 000	0.132 2	0.038 4	0.079 7
	NSGA2-cls ^[11]	5 000	0.345 2	0.005 7	0.104 3
ZDT3	NSGA2-DLS	5 000	0.009 1	0.005 2	0.006 3
	NSGA2 ^[2]	5 000	0.969 7	0.531 7	0.746 3
	HMOEA/D ^[8]	5 000	0.145 7	0.106 5	0.130 8
	NSGA2-cls ^[11]	5 000	0.592 7	0.017 3	0.257 7
ZDT4	NSGA2-DLS	5 000	0.004 6	0.004 1	0.004 3
	NSGA2 ^[2]	5 000	0.815 9	0.243 7	0.569 3
	HMOEA/D ^[8]	5 000	0.521 2	0.005 8	0.177 3
	NSGA2-cls ^[11]	5 000	0.868 3	0.125 8	0.498 7
ZDT6	NSGA2-DLS	5 000	0.005 3	0.002 7	0.003 4
	NSGA2 ^[2]	5 000	0.019 4	0.008 8	0.011 4
	HMOEA/D ^[8]	5 000	0.100 1	0.006 0	0.017 7
	NSGA2-cls ^[11]	5 000	0.005 8	0.003 2	0.004 4

表3 NSGA2-DLS与其他局部搜索算法的ZDT系列实验I结果(连续30次实验)

	算法	函数调用次数	最大值	最小值	平均值
ZDT1	NSGA2-DLS	5 000	0.710 6	0.552 0	0.401 3
	NSGA2 ^[2]	5 000	1.359 3	0.765 2	0.918 8
	HMOEA/D ^[8]	5 000	0.906 6	0.681 6	0.747 6
	NSGA2-cls ^[11]	5 000	0.942 2	0.049 7	0.657 6
ZDT2	NSGA2-DLS	5 000	0.441 5	0.241 5	0.325 2
	NSGA2 ^[2]	5 000	1.084 0	0.791 3	0.944 0
	HMOEA/D ^[8]	5 000	0.924 9	0.684 0	0.846 6
	NSGA2-cls ^[11]	5 000	1.232 2	0.459 5	0.719 9
ZDT3	NSGA2-DLS	5 000	0.682 0	0.548 6	0.628 4
	NSGA2 ^[2]	5 000	1.186 7	0.674 4	0.913 9
	HMOEA/D ^[8]	5 000	1.003 4	0.779 8	0.888 4
	NSGA2-cls ^[11]	5 000	1.261 6	0.721 0	1.036 2
ZDT4	NSGA2-DLS	5 000	1.161 7	0.296 9	0.408 3
	NSGA2 ^[2]	5 000	1.175 4	0.132 0	0.558 6
	HMOEA/D ^[8]	5 000	1.392 3	0.881 4	1.162 9
	NSGA2-cls ^[11]	5 000	1.911 9	0.449 6	1.070 4
ZDT6	NSGA2-DLS	5 000	0.386 6	0.319 3	0.345 4
	NSGA2 ^[2]	5 000	1.539 2	1.123 5	1.357 0
	HMOEA/D ^[8]	5 000	1.735 0	1.467 0	1.601 6
	NSGA2-cls ^[11]	5 000	1.684 3	1.542 1	1.619 2

由表3可知,NSGA2-DLS在5个ZDT函数实验中取得的I值的最大值、最小值和平均值均小于对比方法,因此NSGA2-DLS在有限函数调用次数内得到的解的分布性更好。

3.2 实验2

参照文献[9]的实验,设定当IGD值达到0.01(对比文献中达到的最好值在0.005左右)时停止优化,记

录函数计算次数.实验目的在于验证算法达到指定性能标准所用函数调用次数,实验结果如表4所示。

表4 IGD值达到0.01时NSGA2-DLS与其他局部搜索算法的函数计算次数(10次实验求平均)

	算法	函数调用次数	最大值	最小值	平均值
NSGA2-DLS	3 010	2 240	3 500	2 030	2 100
NSGA2 ^[2]	20 000+	20 000+	20 000+	20 000+	5 150
HMOEA/D ^[8]	13 200	9 600	10 500	7 050	6 900
NSGA2-cls ^[11]	13 350	8 550	17 250	13 200	3 000

由表4可见:在ZDT1~ZDT4的实验中,当IGD达到设定值时,NSGA2-DLS所用函数调用次数远小于对比方法;在ZDT6的实验中,NSGA2-DLS的函数调用次数也是最少的.因此,NSGA2-DLS达到指定标准消耗的计算量更少。

3.3 实验3

本实验对三目标函数DTLZ1和DTLZ2进行优化.实验分别按照文献[8]和文献[11]对种群数、停止条件进行设定,交叉概率均为0.9,变异概率均为1/n,仅局部搜索策略不同,对两个DTLZ函数进行测试,实验目的在于验证NSGA2-DLS算法对三目标函数的优化效果.由于HMOEA/D^[8]和NSGA2-cls^[11]的实验设定不同,本文实验作两组对比.实验结果如表5和表6所示(*表示与原文相同).表5为按照HMOEA/D^[8]实验进行参数设置所得到的实验结果,表6为按照NSGA2-cls^[11]实验进行参数设置所得到的实验结果。

由表5可见,在初始种群数、停止条件、交叉变异概率均相同的情况下,NSGA2-DLS所得解的IGD值的最大值、平均值均小于HMOEA/D^[8],仅在DTLZ2实验和计算量为30 000时的DTLZ1实验中,NSGA2-DLS的IGD最小值略大于HMOEA/D^[8]。

由表6可见,在初始种群数、停止条件、交叉变异概率均相同的情况下,NSGA2-DLS所得解的IGD值的最大值和平均值均小于NSGA2-cls^[11],仅在DTLZ1实验中,当函数调用次数为20 000时NSGA2-DLS的IGD最小值略大于NSGA2-cls^[11]。

由以上分析可知,对于三目标函数,NSGA2-DLS也可以达到较满意的结果.当实验条件完全相同时,NSGA2-DLS的实验结果优于所对比的两种方法。

通过以上3个实验可知:在较少的函数调用次数内,NSGA2-DLS可以得到更优质的解,同时可以保证解的分布性;达到指定目标,NSGA2-DLS消耗的计算量远小于对比的局部搜索算法;在实验设定完全相同的情况下,NSGA2-DLS对双目标优化问题和三目标优化问题都有更好的优化效果。

表5 按文献[8]实验设置参数进行实验的IGD结果对比(连续10次实验)

	算法	种群个数	函数调用次数	最大值	最小值	平均值
DTLZ1	NSGA2-DLS	200	20 000	1.233 4	0.024 3	0.268 6
	NSGA2 ^[2]	200	20 000	1.369 3	0.129 1	0.652 8
	HMOEA/D ^[8]	200	20 000*	1.328 0*	0.082 3*	0.374 1*
	NSGA2-DLS	200	30 000	0.115 1	0.024 7	0.035 6
	NSGA2 ^[2]	200	30 000	0.320 8	0.056 1	0.160 6
	HMOEA/D ^[8]	200	30 000*	0.783 3*	0.018 5*	0.054 6*
DTLZ2	NSGA2-DLS	200	5 000	0.064 4	0.049 6	0.051 9
	NSGA2 ^[2]	200	5 000	0.121 9	0.095 4	0.111 5
	HMOEA/D ^[8]	200	5 000*	0.064 9*	0.048 2*	0.055 4*

表6 按文献[11]实验设置参数进行实验的IGD结果对比(连续10次实验)

	算法	种群个数	函数调用次数	最大值	最小值	平均值
DTLZ1	NSGA2-DLS	100	20 000	0.655 6	0.038 9	0.365 1
	NSGA2 ^[2]	100	20 000	4.680 2	1.307 1	2.751 3
	NSGA2-els ^[11]	100	20 000*	3.125 0*	0.023 5*	0.466 7*
DTLZ2	NSGA2-DLS	100	20 000	0.076 8	0.066 4	0.071 7
	NSGA2 ^[2]	100	20 000	0.169 7	0.121 3	0.141 0
	HMOEA/D ^[8]	100	20 000*	0.185 7*	0.091 4*	0.102 9*

4 结 论

为了解决目前局部搜索NSGA2算法计算量大的问题,本文提出了一种基于密度的局部搜索算法NSGA2-DLS. NSGA2-DLS在一次遗传操作内仅对一个稀疏解进行局部搜索操作,因此单次遗传操作的计算量远小于其他局部搜索算法. 由于使用密度进行稀疏解的定义,在稀疏解周围进行局部搜索,NSGA2-DLS可以确保解分布的均匀性和广泛性;极限优化策略与随机搜索策略的结合使得NSGA2-DLS向Pareto前沿的收敛速度更快,解得质量更高. 通过与标准NSGA2^[2]、HMOEA/D^[8]和NSGA2-els^[11]的对比可以看出,NSGA2-DLS在保证解的质量和分布性的同时,有效减少了计算量,使得局部搜索类优化算法可以应用于解决计算量限制较强的优化问题.

参考文献(References)

- [1] Zeng G Q, Chen J, Li L M, et al. An improved multi-objective population-based extremal optimization algorithm with polynomial mutation[J]. Information Sciences, 2016, 330: 49-73.
- [2] 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.
- [3] Ahmadi A. Memory-based adaptive partitioning(MAP) of search space for the enhancement of convergence in pareto-based multi-objective evolutionary algorithms[J]. Applied Soft Computing, 2016, 41: 400-417.
- [4] Deb K, Goel T. A hybrid multi-objective evolutionary approach to engineering shape design[C]. Int Conf on Evolutionary Multi-criterion Optimization. Berlin:

Springer Heidelberg, 2001: 385-399.

- [5] Lara A, Sanchez G, Coello C A C, et al. HCS: A new local search strategy for memetic multiobjective evolutionary algorithms[J]. IEEE Trans on Evolutionary Computation, 2010, 14(1): 112-132.
- [6] Ishibuchi H, Murata T. A multi-objective genetic local search algorithm and its application to flowshop scheduling[J]. IEEE Trans on Systems, Man, and Cybernetics, Part C: Applications and Reviews, 1998, 28(3): 392-403.
- [7] Jaskiewicz A. Genetic local search for multi-objective combinatorial optimization[J]. European J of Operational Research, 2002, 137(1): 50-71.
- [8] Sindhya K, Miettinen K, Deb K. A hybrid framework for evolutionary multi-objective optimization[J]. IEEE Trans on Evolutionary Computation, 2013, 17(4): 495-511.
- [9] Palar P S, Tsuchiya T, Parks G T. A comparative study of local search within a surrogate-assisted multi-objective memetic algorithm framework for expensive problems[J]. Applied Soft Computing, 2016, 43: 1-19.
- [10] Harada K, Sakuma J, Kobayashi S. Local search for multiobjective function optimization: Pareto descent method[J]. Trans of the Japanese Society for Artificial Intelligence, 2006, 21(4): 659-666.
- [11] Kim H, Liou M S. Adaptive directional local search strategy for hybrid evolutionary multiobjective optimization[J]. Applied Soft Computing, 2014, 19(6): 290-311.
- [12] Yang S. Genetic algorithms with memory and elitism-based immigrants in dynamic environments[J]. Evolutionary Computation, 2008, 16(3): 385-416.
- [13] Messac A. The normalized normal constraint method for generating the pareto frontier[J]. Structural & Multidisciplinary Optimization, 2003, 25(2): 86-98.

(责任编辑: 郑晓蕾)