

利用种群扩张与稀疏化策略改进NSGA-II-DE算法

蒋永华^{1†}, 许妙忠², 成 刚²

(1. 武汉大学 遥感信息工程学院, 武汉 430070; 2. 武汉大学 测绘遥感信息工程国家重点实验室, 武汉 430070)

摘 要: NSGA-II-DE 算法是在 NSGA-II 算法的基础上利用 DE 算法的收敛速度快、鲁棒性高的特性得到的改进算法, 该算法提高了原算法的收敛速度, 同时也降低了原算法对参数的依赖性. 然而, 原算法的解群分布性却没有得到提高. 鉴于此, 提出一种基于种群扩张与稀疏化策略的改进型 NSGA-II-DE 算法. 该算法利用种群扩张增加候选解的数量, 再利用稀疏化策略从候选解中选出使得整体分布尽可能均匀的最优解. 种群扩张通过在进化最后的若干代保留每代中的第一非支配面上的个体来实现. 在迭代结束后, 对种群进行非支配排序, 去除第一非支配面以外的个体, 以提高解群质量. 进行稀疏化处理, 即对扩张后的全部个体按目标向量的某一维度排序, 再筛选出相邻间距最接近期望距离的个体, 以达到改善解群分布性的目的. 仿真实验表明, 所提出的算法在改善原算法的解群分布性上表现优异, 但算法的时间和空间复杂度较原算法有所增加.

关键词: 种群扩张; 稀疏化; NSGA-II-DE; 非支配排序; 种群规模; 分布性

中图分类号: TP18 **文献标志码:** A

Using population expansion and sparsity strategy to improve NSGA-II-DE algorithm

JIANG Yong-hua^{1†}, XU Miao-zhong², CHENG Gang²

(1. School of Remote Sensing and Information Engineering, Wuhan University, Wuhan 430070, China; 2. State Key Laboratory of Information Engineering in Surveying, Mapping and Remote Sensing, Wuhan University, Wuhan 430070, China)

Abstract: The NSGA-II-DE algorithm improves the convergence speed and reduces the dependency on the parameters of NSGA-II by taking advantage of the fast convergence and high robustness of the DE algorithm. However, the solutions distribution of the original algorithm is not improved. In this study, an advanced NSGA-II-DE algorithm based on the population expansion and sparsity strategy is proposed. This algorithm increases the number of candidate solutions by using population expansion operation, and the sparsity strategy is used to get optimum solutions from candidates that make the global distribution as uniform as possible. Population expansion is achieved by preserving individuals on the first non-dominated surface of each generation in the last few generations of evolution. At the end of the iteration, the population is non-dominated sorted, and the individuals outside the first non-dominating surface are removed to improve the quality of the solutions. Then, the sparseness is processed, that is, all the individuals after expansion are sorted according to a certain dimension of the target vector, and the individuals whose adjacent distance closest to the expected distance are selected, so as to improve the distribution of the solutions. The simulation results show that the proposed algorithm is superior to the original algorithm in terms of the distribution of solutions, but inferior to the original algorithm in terms of the time and space complexity.

Keywords: population expansion; sparsity; NSGA-II-DE; non-dominated sorting; population size; distribution

0 引 言

多目标优化问题 (Multi-objective optimization problem, MOP) 是一类求解同时使多个目标最优化的解集问题, 相对于单目标优化问题 (Single-objective

optimization problem, SOP) 而言, MOP 一般不存在一个使得其在所有目标维度都最优的最优解. 这是因为, MOP 问题中各个目标函数通常是相关而且互相矛盾的, 即调整决策向量, 使目标向量的某一维度的

收稿日期: 2017-07-20; 修回日期: 2017-11-09.

责任编委: 巩敦卫.

作者简介: 蒋永华 (1987-), 男, 副教授, 博士, 从事光学/视频卫星几何指标论证、几何定位精度提升、高精度产品生产等研究; 许妙忠 (1963-), 男, 教授, 博士, 从事摄影测量与遥感等研究.

[†]通讯作者. E-mail: jiangyh@whu.edu.cn.

值减小,往往意味着另外一个或多个维度的值会增大.如目标权重法这一类传统的多目标优化方法,是通过将MOP转化成SOP以达到简化求解的目的,但这类方法需要较多的先验知识,往往需要针对不同的问题及需求调整参数,且容易导致搜索向某一特定目标进行,产生遗传漂移现象,在实际运用中效果不好^[1].由此,人们引入Pareto最优概念来解决MOP.

所谓Pareto最优解,是指不受决策空间中其他任何解支配的解,一个解如果是Pareto最优解,则意味着没有任何一个其他解能够在全部目标维度上都比这个解更优.MOP的解就是Pareto最优解集.为了解决MOP,研究者们提出了诸多经典算法^[2],如VEGA(Vector evaluated genetic algorithms)和MOGA(Multi-objective genetic algorithm),Corne等^[3]提出的PESA(Pareto envelope-based selection algorithm),Zitzler等^[4]提出的SPEA(Strength Pareto evolutionary algorithm)以及在其基础上提出的改进SPEA2^[5],Sendhoff等^[6]提出的PAES(Pareto archived evolution strategy),Hom等^[7]提出的NPGA(Niched Pareto genetic algorithm),Srinivas等^[8]提出的NSGA(Nondominated sorting in genetic algorithms)以及改进后的NSGA-II^[9]等,这些算法统称为多目标进化算法(Multi-objective evolutionary algorithms, MOEAs).这些算法原理不同,但共同的目标包括以下几点:1)进化结果尽可能地接近Pareto最优前沿;2)收敛速度尽可能快;3)进化结果尽可能分布均匀;4)进化结果尽可能分布在整个Pareto最优前沿上,而不是集中在某一块或多块区域^[10].

在上述诸多算法中,NSGA-II算法以其较好的收敛性、分布性,较低的时间复杂度,成为基于非支配排序算法中的经典算法^[11].Li等^[12]提出的NSGA-II-DE算法是NSGA-II算法的改进版本,该算法使用DE(Differential evolution)算法的变异和交叉算子替换原NSGA-II算法中的模拟二进制杂交^[13](The simulated binary crossover, SBX)算子,降低了交叉算子对参数的敏感程度,提高了算法的收敛性和收敛速度.然而,该算法在提高NSGA-II算法的解群分布性方面却没有太大的进步.因此本文基于NSGA-II-DE算法进行改进,提出一种基于种群扩张与稀疏化策略的改进型NSGA-II-DE算法,即NSGA-II-DEES算法.实验表明,该算法在解群分布性方面,相对于NSGA-II-DE算法具有很大的提高.

1 NSGA-II与NSGA-II-DE算法

NSGA-II和NSGA-II-DE都是解决MOP的算法,后者对前者的改进之处主要在于结合了差分进化

(DE)算法^[14].DE算法由Rainer等^[15]于1995年提出,具有较强的全局收敛能力和鲁棒性,是一种高效的并行搜索算法^[16].由于其在进化问题中具有优良表现,DE算法得到了学术界和工程界的广泛关注.下面先对MOP进行数学描述,再介绍NSGA-II,然后介绍NSGA-II-DE对前者的改进部分.

1.1 问题描述

不失一般性,MOP可以描述成以下形式:

$$\begin{cases} \min f(x) = (f_1(x), f_2(x), \dots, f_k(x)), \\ e(x) = (e_1(x), e_2(x), \dots, e_k(x)) \leq 0. \end{cases} \quad (1)$$

其中: $x = (x_1, x_2, \dots, x_n) \in X \subset R^n$, x 为决策向量, X 为决策空间; $y = (y_1, y_2, \dots, y_k) \in Y$, y 为目标向量, Y 为目标空间; $e(x) \leq 0$ 为约束条件.本文仅讨论无约束条件的MOP,即求解使得目标向量 $f(x)$ 整体最优的解集.

1.2 NSGA-II

NSGA-II由NSGA改进而来,其采用快速非支配排序及精英策略,以较低的时间复杂度提高了原算法的收敛性,同时采用拥挤距离(Crowding-distance)排序机制,提高了解群的分布性^[17].NSGA-II的整体算法流程如下^[9].

Step 1: 初始化.随机生成初始种群 R_t , R_t 的种群规模为 $2N$,其中 $t = 0$ 表示代数.

Step 2: 快速非支配排序.根据 R_t 中各个体的目标向量,将种群按照个体之间的支配情况进行快速排序,排序完成后个体即被分配到不同的非支配面 F_i (i 为非支配序).

Step 3: 拥挤距离计算.定义一个空的集合 P_{t+1} ,按 i 值从小到大的顺序计算各 F_i 内个体的拥挤距离,并入集合 P_{t+1} ,直到 $|P_{t+1}| + |F_i| > N$ ($|P|$ 表示集合 P 中元素个数).

Step 4: 对 F_i 中的个体,按照拥挤距离从大到小进行排序,最后取前 $N - |P_{t+1}|$ 个个体,并入到集合 P_{t+1} 中.若 $t + 1 \geq \text{MaxGen}$ (最大迭代代数)或满足算法退出条件,则返回 P_{t+1} ,否则继续Step 5.

Step 5: 对 P_{t+1} 中的个体使用锦标赛选择算法、模拟二进制杂交算法和多项式变异算法,得到新的大小为 N 的种群 Q_{t+1} .

Step 6: $t = t + 1$, $R_t = P_t \cup Q_t$,转到Step 2.

NSGA-II由于使用快速非支配排序和拥挤距离排序机制,其解的收敛性和分布性都比较好.但由于使用了效率较低的模拟二进制杂交算法,收敛速度较慢,对于一些复杂问题,如论文中提到的ZTD4,在收

敛速度和收敛性方面表现都不是很理想.

1.3 NSGA-II-DE

针对NSGA-II在复杂问题中收敛速度和收敛性问题,NSGA-II-DE使用DE算法的变异、杂交和选择算子代替原算法中的模拟二进制杂交算子,对算法作出了改进.由于利用了DE算法在收敛速度上的优异表现,改进后的算法提高了原算法的收敛速度和收敛性.NSGA-II-DE算法与NSGA-II算法在其他方面基本相同,这里着重介绍NSGA-II-DE中结合DE算法改进的部分.

从种群的 $N(N \geq 4)$ 个个体中随机选择4个不同的个体,记为 $x_i, x_{p1}, x_{p2}, x_{p3}$.首先进行变异,令

$$h_i = x_{p1} + F \cdot (x_{p2} - x_{p3}).$$

其中: $x_{p2} - x_{p3}$ 为差分向量; F 为缩放因子,调整 F 的大小可以控制差分向量的影响力.然后进行交叉,交叉操作可以增加种群的多样性,方法如下:

$$v_i = \begin{cases} h_i, & \text{rand}(0, 1) \leq \text{cr}; \\ x_i, & \text{else.} \end{cases} \quad (2)$$

其中 $\text{cr} \in [0, 1]$ 为交叉概率, $\text{rand}(0, 1)$ 为 $[0, 1]$ 上服从均匀分布的随机数.最后进行选择,即根据评价函数选择 v_i 或是 x_i 作为 x_{i+1} :

$$x_{i+1} = \begin{cases} v_i, & f(v_i) < f(x_i); \\ x_i, & \text{else.} \end{cases} \quad (3)$$

反复进行变异、交叉和选择操作,直至获得 N 个下一代个体.

2 NSGA-II-DEES

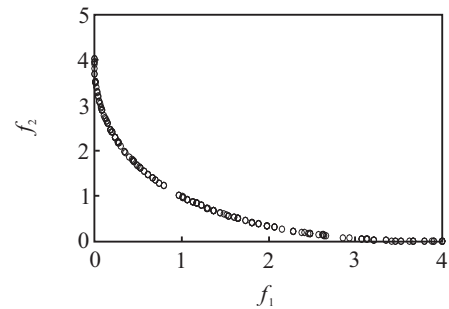
虽然NSGA-II-DE对原算法的收敛速度和收敛性的提高做出了很大贡献,但在解群分布均匀性上没有很大提高,因此本文在NSGA-II-DE的基础上继续改进,采用种群扩张(Expansion)与稀疏化(Sparsity)策略,提出一种改进的NSGA-II-DEES算法.

2.1 算法思想

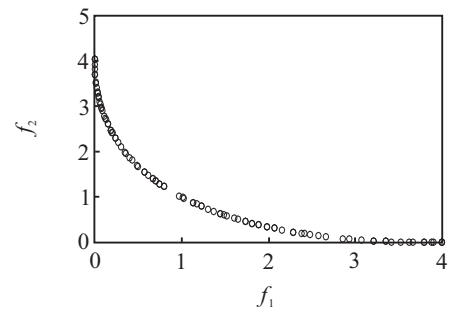
无论是NSGA-II还是NSGA-II-DE,在维持种群分布的均匀性上都只是依赖于拥挤距离排序算子.以NSGA-II为例,在种群进化后期,快速非支配排序使得几乎全部的 $2N$ 个解都集中在了第一非支配面 F_1 上,因此 F_1 中的个体将会直接进行拥挤距离排序,即计算全部个体的拥挤距离,然后按照排序算法,对 F_1 中的全部个体按拥挤距离从大到小排序,取前 N 个个体并入到 P_{t+1} 中,若满足退出条件(如达到最大迭代次数),则返回 P_{t+1} 作为最终解集.由上面的分析

过程可知,最终解集的均匀程度取决于最后一次从 F_1 中取出个体的方法优劣.

一般而言,最后一次从 F_1 中取值时, F_1 中个体的数量 $|F_1|$ 大于 N 而接近或等于 $2N$,若小于 N ,则说明进化代数可能不够,算法还未收敛.要从 $|F_1|$ 个个体中选出 N 个个体,并使这些个体在目标空间中分布尽可能均匀,首先可以考虑提高候选解的数量,即 F_1 的种群规模.若 F_1 中的个体数量较少,比如略大于 N ,则最终解集的均匀程度就基本取决于 F_1 中个体的均匀程度,这样最终解集的均匀性就带有一定的随机性.而且随着迭代的继续,均匀性的提高是有限度的,当 $|F_1|$ 等于 $2N$ 时均匀性在理论上达到上限.图1表示在SCH问题中,采用NSGA-II-DE算法迭代到最大代数100代时, F_1 和 P_{t+1} 中个体分布情况($N = 100$).



(a) F_1 个体分布情况(个体数量为199)



(b) P_{t+1} 个体分布情况(个体数量为100)

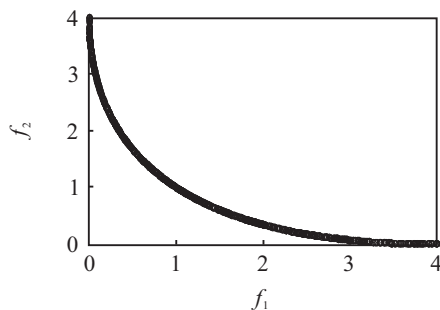
图1 F_1 和 P_{t+1} 中个体分布情况

图1(a)是对SCH问题的某次实验得到的 F_1 在目标空间的分布情况.从图1(a)中可以看出, F_1 中的个体分布并不均匀,存在着有些地方过于密集而有些地方过于稀疏的情况.实验表明,随着进化代数的增加,这种不均匀的现象并不会消失或减弱,而是呈现出一种随机变化状态.

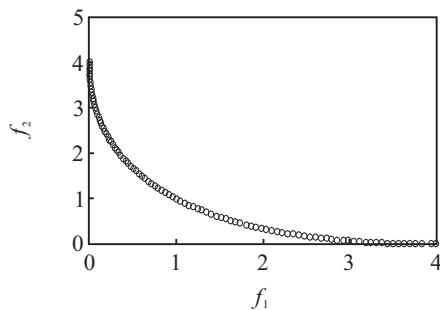
为了得到较为均匀的最终解集 P_{t+1} , F_1 中过于密集的地方可以通过减少一些个体来使其均匀,但过于稀疏的地方则无法“插入”个体,只能任由这种稀疏的情况保留到提取后的种群 P_{t+1} 中,这样无疑会影响 P_{t+1} 的均匀性.图1(b)为提取出来的 N 个个体,

可以看出,这些个体虽然大致上分布均匀,但在一些细节地方分布情况还不够理想,依然存在着部分区域密集部分区域稀疏的问题。

$|F_1|$ 并不是一定要限制在 $2N$ 以下,之所以会有此限制,是因为参与非支配排序的种群规模为 $2N$.可以考虑在种群进化到超过最大代数后继续迭代GenExp代,在这一段进化过程中将每一代 P_t 中的个体全部保存到 P_Exp 中,进化结束后,对 P_Exp 中的 $Gen_Exp \times N$ 个个体进行去重、快速非支配排序操作,再取其中的第一非支配面中的个体作为新的 F_1 ,此时的 F_1 中的个体数量上限为 $Gen_Exp \times N$,这样就达到了扩充 F_1 的种群规模的目的.图2(a)为继续迭代50代后扩充 F_1 的分布情况,此时 F_1 中的个体数量为973,分布密度大大增加。



(a) 扩充50代后 F_1 中的个体分布(种群扩张结果)



(b) 对扩张种群稀疏化得到最终结果(稀疏化结果)

图2 种群扩张与稀疏化结果

在扩充种群后的 F_1 中选择 N 个个体,使其在目标空间中的分布尽可能均匀,这一过程本文称之为稀疏化.对于稀疏化,如果采用原NSGA-II中的先进行拥挤距离排序、再取拥挤距离最大的前 N 个个体的方法,则得到的 N 个个体的分布均匀程度并不高.因此本文采取另一种更直接的方法:将 F_1 中的个体按各目标向量值的大小排序,计算相邻个体间的欧氏距离,对这些距离求和,然后计算将要抽取出来的 N 个个体的预期个体距离的均值 \bar{d}_{expect} ,再从排好序的全部个体一端开始,按照优先抽取端点和使抽取出来个体相邻距离最接近 \bar{d}_{expect} 的原则,逐个计算需要抽取出来的个体.图2(b)为从 F_1 中抽取出来的 N 个个体,

从图2(b)可看出,最终结果分布的均匀性很好。

因为种群扩张与稀疏化的策略直接以解群分布的均匀性为目标,思想简单直接,方法易于操作,因此在提高均匀性方面具有很好的效果.但是此方法也有一些不足之处,具体包括增加了时间与空间复杂度和难以适应高于二维的目标空间等问题。

2.2 算法实现与分析

按照算法的核心思想,NSGA-II-DEES对NSGA-II-DE的改进包括两方面:种群扩张和稀疏化.种群扩张策略的实现体现在算法主流程上的改进.下面给出改进后的算法流程。

Step 1: 初始化.随机生成初始种群 R_t, R_t 的种群规模为 $2N$,其中 $t = 0$ 表示代数,定义集合 S 为扩充种群, S 初始值为空集,MaxGen、ExpGen分别为最大迭代代数和种群扩张的代数。

Step 2: 快速非支配排序.根据 R_t 中各个体的目标向量,将种群按照个体之间的支配情况进行快速排序,排序完成后个体即被分配到不同的非支配面 F_i (i 为非支配序),若 $t > \text{MaxGen}$,则将 F_1 并入 S 中(从 $\text{MaxGen} + 1$ 代到迭代结束,每次将生成的 F_1 中的全部个体并入 S ,即种群扩张)。

Step 3: 若 $t + 1 \geq \text{MaxGen} + \text{ExpGen}$,则将 S 中的个体去重,再进行快速非支配排序,剔除 S 中第一非支配面 F_1 以外的个体,然后使用稀疏化操作对 S 中的个体进行稀疏化,返回稀疏化结果,否则继续。

Step 4: 拥挤距离计算.定义一个空的集合 P_{t+1} ,按 i 值从小到大的顺序计算各 F_i 内个体的拥挤距离,并入集合 P_{t+1} ,直到 $|P_{t+1}| + |F_i| > N$ ($|P|$ 表示集合 P 中元素个数)。

Step 5: 对于 F_i 中的个体,按照拥挤距离从大到小进行排序,最后取前 $N - |P_{t+1}|$ 个个体,并入到集合 P_{t+1} 中。

Step 6: 对 P_{t+1} 中的个体使用差分进化算子进行杂交变异,得到新的大小为 N 的种群 Q_{t+1} 。

Step 7: $t = t + 1, R_t = P_t \cup Q_t$,返回Step 2。

改进后的算法主流程与原NSGA-II-DE非常相似,主要区别在Step 2和Step 3.在Step 2中,使用集合 S 记录种群在最后MaxGen到 $\text{MaxGen} + \text{ExpGen}$ 代的进化过程中产生的第一非支配面的个体,即当前代中的最优个体. Step 3中,迭代结束后首先对扩充种群 S 进行去重,这是因为前面各次迭代中可能会存在一些个体被保留至下一代,被重复记录.另外, S 中的个体虽然是各代中的非支配解,但实验发现,这些

个体放在一起后会出现新的支配关系,因此在去重后需对 S 进行非支配排序,剔除第一非支配面以外的个体,之后对 S 进行稀疏化操作. 图3是改进算法的流程图,两处虚线框部分即相对NSGA-II-DE的主要改进部分.

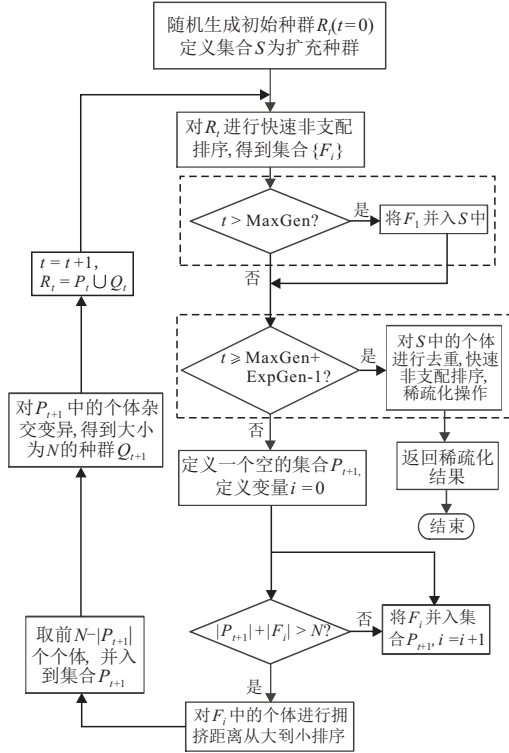


图3 改进后算法主流程

假设需要进行稀疏化的种群为 S , 稀疏化后的个体数量为 M (M 为稀疏化函数的形参, 若要将个体数量稀疏化成 N , 则将 N 作为实参传入), 稀疏化操作的流程图见图4.

下面是稀疏化操作的具体步骤.

Step 1: 判断 S 中的个体数是否大于 M , 若是则继续, 否则返回 S .

Step 2: 对集合 S 中的个体按照目标向量的任一维度进行排序, 升序或者降序都可以.

Step 3: 计算相邻个体的目标空间欧氏距离 d_i ($i = 1, 2, \dots, |S| - 1$), 记 D 为 d_i 组成的集合.

Step 4: 对全部 d_i 求均值 d_{mean} 和标准差 d_{std} .

Step 5: 查找 d_i 中是否存在明显过大的异常值 $d_{abnormal}$ 满足 $d_{abnormal} > d_{mean} + 12 \times d_{std}$. 若存在, 则定义 $m = 3$, 否则定义 $m = 9$, m 为标准差系数, 这里的 12、3 和 9 为多次实验得到的较合理的值, 后两个数分别对应目标连续和目标分段两种情况.

Step 6: 定义空集 D_{sel} , 将全部 d_i 中满足条件 $d_i < d_{mean} + m \times d_{std}$ 的数并入集合 D_{sel} , 定义 $k = |D| - |D_{sel}|$, 代表未被选入的数的个数, 定义 $flag = 0$, $flag$

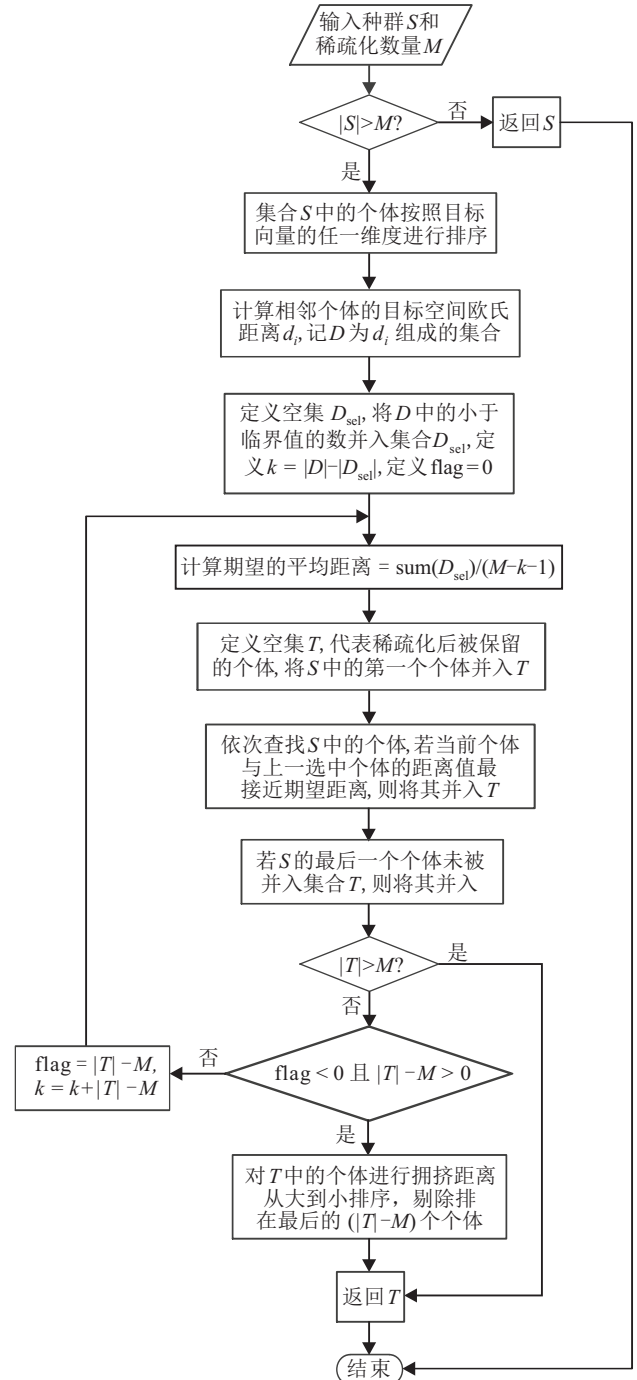


图4 稀疏化操作流程

代表前一次迭代过程中得到的解集个体数 $|T|$ 与期望值 M 的差.

Step 7: 计算期望的平均距离

$$\bar{d}_{expect} = \text{sum}(D_{sel}) / (M - k - 1). \quad (4)$$

Step 8: 定义空集 T , T 中的元素为种群个体, 代表稀疏化后被保留的个体, 将 S 中的第一个个体并入 T .

Step 9: 从上一次在 S 中选出个体的位置 i 开始, 往后依次查找 S 中的个体, 直到找到位置 j , 使得个体 S_i 到个体 S_j 之间的距离段之和 $d_{i,j}$ 与 S_i 到 S_{j+1} 之间的距离段之和 $d_{i,j+1}$ 满足关系 $d_{i,j} \leq \bar{d}_{expect} \leq d_{i,j+1}$.

Step 10: 计算 $dd_{i,j} = \bar{d}_{\text{expect}} - d_{i,j}$, $dd_{i,j+1} = d_{i,j+1} - \bar{d}_{\text{expect}}$, 若 $dd_{i,j} < dd_{i,j+1}$, 则将 S_j 并入集合 T , 否则将 S_{j+1} 并入集合 T .

Step 11: 重复 Step 9 和 Step 10, 直到 S 中的个体已全部找完.

Step 12: 检查 S 的最后一个个体是否被并入集合 T , 若没有并入则将其并入.

Step 13: 检查集合 T 的大小 $|T|$, 若 $|T|$ 等于 M , 则返回集合 T , 算法结束, 否则继续.

Step 14: 若 $\text{flag} < 0$ 且 $|T| - M > 0$, 则表示上一次迭代得到的个体数少于 M 而本次多于 M , 若继续迭代则会产生振荡且无法终止, 此时对 T 中的个体进行拥挤距离从大到小排序, 将排在最后的 $|T| - M$ 个个体从 T 中去除, 然后返回 T , 算法结束, 否则继续.

Step 15: 用 $|T| - M$ 更新 flag 的值, 用 $k + |T| - M$ 更新 k 的值, 转至 Step 7.

算法 Step 5 中首先使用 12 倍标准差来检测是否存在异常值, 这里检测到的异常值并不一定是全部的异常值. 因此定义标准差系数 m , 当 D 中存在异常值, 即过大的值时, 为了尽可能多地剔除过大的值, 将 m 设置为较小数 ($m = 3$), 即“筛选”较为严格; 而当 D 中未检测出异常值时, 说明 D 中的数比较平均, 此时标准差 d_{std} 会比较小, 为了防止把过多的正常值当异常值剔除掉, 将 m 设置为较大值 ($m = 9$), 即“筛选”较宽松. Step 6 中的 $|D| - |D_{\text{sel}}|$ 表示从原个体间距离集合 D 中剔除的过大距离的数量, 这里考虑了目标分段情况. 若目标分段, 则断开处的“缺口”会使得集合 D 中存在较一般值大很多的值, 这样的值如果不剔除的话, 会使得 Step 7 计算期望的平均距离偏大,

从而导致最终结果 T 的个体数相对于 N 偏小. 图 5 为 KUR 问题的种群扩张后 F_1 中个体分布情况, KUR 问题就是一个目标分段问题.

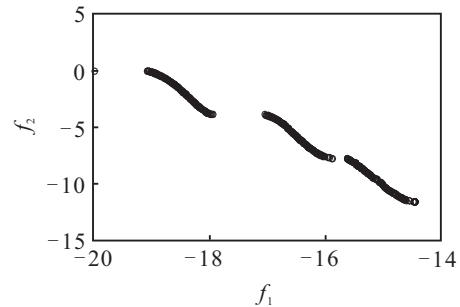


图 5 KUR 问题的种群扩张后 F_1 个体分布情况

算法的 Step 13 ~ Step 15 保证了输出集合 T 的大小一定为 N . Step 14 中, $\text{flag} < 0$ 代表上一次计算出来的 T 的个体数量小于 M , 而 $|T| - M > 0$ 代表本次计算出来的 T 的个体数量大于 M , 这两个条件同时成立, 说明无法通过调整 k 值的大小, 使得 T 中的个体数量刚好等于 M . 在这种情况下采用拥挤距离排序的方法, 对 T 中的个体排序, 然后去除拥挤距离过小的多出来的个体. 若 Step 14 中的条件不满足, 则进入到 Step 15 中, 对 flag 的值和 k 的值进行更新, 因为期望的平均距离与 k 的值有关, 所以调整 k 能影响到 T 中的个体数量, 更新 k 的值后转至 Step 7, 可以使得 T 中的个体数量快速往 M 靠近.

3 仿真实验分析

选取 9 个测试函数^[9, 18]: SCH, FON, POL, KUR, ZDT1, ZDT2, ZDT3, ZDT4, ZDT6, 对 NSGA-II-DE 与 NSGA-II-DEES 两种算法进行实验对比. 测试函数信息见表 1.

表 1 测试函数信息

测试函数	SCH	FON	POL	KUR	ZDT1	ZDT2	ZDT3	ZDT4	ZDT6
决策向量维度	1	3	2	3	30	30	30	10	10
目标向量维度	2	2	2	2	2	2	2	2	2
目标是否连续	是	是	否	否	是	是	否	是	是

对于所有的测试函数, 初始种群大小 N 统一设为 100, DE 缩放因子 F 统一设为 0.5, DE 交叉概率 cr 统一设为 0.3, 种群扩张的进化代数 ExpGen 统一设为 50 代, 由于最大迭代代数 MaxGen 与问题复杂程度有关, 对于不同的问题在大小设置上有些区别: 对于 SCH, FON, POL, KUR 问题, 将 MaxGen 设为 100; 对于 ZDT1, ZDT2, ZDT6 问题, 将 MaxGen 设为 500; 对于 ZDT3 问题, 将 MaxGen 设为 400; 对于 ZDT4 问题, 将 MaxGen 设为 600. 事实上, 由于 NSGA-II-DE 算法本身具有较好的保持种群多样性的能力, 将 MaxGen 设

置的大一些 (例如所有测试函数的 MaxGen 都设为 1000) 也不会使种群收敛过度, 从而出现聚集现象. NSGA-II-DE 算法的进化代数为 $\text{MaxGen} + \text{ExpGen}$.

表 2 显示了各问题解群的分布度的均值与方差结果. 由实验结果可以看出, 整体上 NSGA-II-DEES 在维持解群的分布性方面相比于 NSGA-II-DE 有很大提升, 而且对结果分段情况的效果一样好.

表 2 中的分布度 Δ 由 Deb^[9] 等提出, 是衡量解群分布性的一个指标, 其均值越小, 说明种群的分布性越好.

表2 分布度 Δ 的均值与方差

测试函数		SCH	FON	POL	KUR	ZDT1	ZDT2	ZDT3	ZDT4	ZDT6
NSGA-II-DE	均值	1.159 8	0.834 1	1.426 8	1.179 3	0.809 9	0.820 6	1.185 0	1.256 3	1.495 8
	方差	0.003 2	0.001 8	0.041 5	0.000 9	0.002 4	0.011 5	0.003 2	0.171 5	0.022 1
NSGA-II-DEES	均值	0.068 9	0.084 8	0.078 8	0.243 0	0.062 5	0.260 1	0.431 3	0.643 6	0.618 6
	方差	0.000 0	0.000 0	0.000 0	0.000 0	0.000 3	0.211 1	0.001 2	0.381 3	0.239 0

在某些情况下,类似图2(b)的目标分布图上会有一个较为明显的“缺口”(非目标分段处,两相邻个体间距离大概等于平均值的两倍),这种局部性分布不均匀的现象,是由稀疏化过程的最后剔除多余点的操作造成的.在稀疏化过程中,有时通过调整 k 值的大小不能使解群大小正好变为 N (会在 N 的附近上下浮动).出现这种情况时,为了使解群大小严格等于 N ,这里首先将解群大小调整到略大于 N ,再按照拥挤距离从解中剔除多余的个体,这就会导致“缺口”的出现.事实上,如果对解群的大小要求不是十分严格,即只要在 N 左右或略大于 N ,就不需要剔除操作,这种局部性分布不均匀的现象就能消除.要完全消除缺口,一种可能的解决方法是:当无法通过调整 k 值来使解群大小成为 N 时,就不再调整 k 值,而是使用二分法调整 \bar{d}_{expect} 的值,由于 \bar{d}_{expect} 为实数,理论上总能找到合适的 \bar{d}_{expect} ,使得解群的大小刚好为 N ,但可能对于某些特殊的情况,需要调整很多次才能找到,因此本文没有使用这种方法.

从理论上分析,算法有两点不足之处:

1) 对于目标维数超过2维的情况不太容易实现稀疏化.在稀疏化过程中,需要预先估计各个目标点之间的欧氏距离,再根据此距离进行目标点的选取,当目标维数超过2维时,很难再采用2维情况时先计算总距离再预测各个目标点之间的欧氏距离的方法,因为高维情况下总距离不容易计算.

2) 算法时间复杂度与空间复杂度都有所增加.由于算法要求在原NSGA-II-DE进化结束后继续进化若干代,而且需要将后面进化过程中新产生的个体记录下来,算法时间和空间复杂度都有所增加,增加的幅度与后续进化的代数成正比.

在实际使用算法时,为了尽可能将算法的劣势减小,发挥算法的优势,应该尽量减小种群扩张的代数,降低算法运行过程中对时间和空间的消耗,使算法的时间和空间复杂度上的劣势尽量避免.因此选取不同的种群扩张代数,对SCH和KUR问题进行实验,实验结果见表3.

表3 不同种群扩张代数下分布度 Δ 的均值与方差

算法		SCH			KUR		
		10	20	50	10	20	50
NSGA-II-DE	均值	1.100 7	1.107 9	1.159 5	1.143 0	1.165 7	1.265 7
	方差	0.000 1	0.002 5	0.002 5	0.001 8	0.013 1	0.004 4
NSGA-II-DEES	均值	0.186 9	0.116 7	0.064 3	0.312 8	0.261 9	0.239 7
	方差	0.000 4	0.000 0	0.000 0	0.000 4	0.000 0	0.000 0

从表3可以看出,实验结果与预期相符:使用种群扩张与稀疏化策略后,解群的分布均匀性明显得到提高,而种群扩张代数越多,解群的分布均匀性提高得越显著.从结果可以看到,对原NSGA-II-DE在MaxGen的基础上多进化50代,其解群分布性甚至不如NSGA-II-DEES扩张10代的结果,说明新算法在提高解群分布性上效果显著.新算法的时间和空间消耗与种群扩张代数相关,因此使用此算法时,需要使用者根据自己需要进行权衡,是选择较多的种群扩张代数以提高解群分布性,还是选择较少的种群扩张代

数以降低时间和空间消耗.

4 结 论

本文对NSGA-II及NSGA-II-DE的分布性保持策略进行了深入研究,针对其不足之处,提出了一种基于种群扩张与稀疏化策略的改进NSGA-II-DEES算法.仿真结果表明,相比于原算法,所提出的算法大大提高了最终解群的分布均匀性,在分布性上具有明显的优势.但是算法也存在不足之处:1) 需要针对高维情况对稀疏化方法进行改进;2) 需要尽量降低算法的时间与空间复杂度.未来将针对这两方面进行

改进和优化.

参考文献(References)

- [1] 王向慧, 连志春, 徐志英, 等. 基于 Pareto 最优概念的多目标进化算法研究[J]. 计算机工程与应用, 2008, 44(27): 58-61.
(Wang X H, Lian Z C, Xu Z Y, et al. Research on Pareto optimal-based multiobjective evolutionary algorithms[J]. Computer Engineering and Applications, 2008, 44(27): 58-61.)
- [2] 杨景明, 侯宇浩, 孙浩, 等. 采用数量级阈值与二维信息排序策略的 NSGA-II-DE 算法[J]. 控制与决策, 2016, 31(9): 1577-1584.
(Yang J M, Hou Y H, Sun H, et al. Modified NSGA-II-DE with two-dimensional information ordering strategy and magnitude threshold[J]. Control and Decision, 2016, 31(9): 1577-1584.)
- [3] Corne D W, Knowles J D, Oates M J. The Pareto envelope-based selection algorithm for multiobjective optimization[C]. Int Conf on Parallel Problem Solving From Nature. London: Springer-Verlag, 2000: 839-848.
- [4] Zitzler E, Thiele L. Multiobjective evolutionary algorithms: A comparative case study and the strength Pareto approach[J]. IEEE Trans on Evolutionary Computation, 1999, 3(4): 257-271.
- [5] Zitzler E, Laumanns M, Thiele L. SPEA2: Improving the strength Pareto evolutionary algorithm[Z]. Zurich: Swiss Federal Institute of Technology, 2001.
- [6] Joshua Knowles, David Corne. The Pareto archived evolution strategy: A new baseline algorithm for Pareto multiobjective optimisation[C]. Proc of the Congress on Evolutionary Computation. Washington: IEEE, 1999: 98-105.
- [7] Horn J, Nafpliotis N, Goldberg D E. A niched Pareto genetic algorithm for multiobjective optimization[C]. Proc of the 1st IEEE Conf on Evolutionary Computation. Piscataway: IEEE Service Center, 2002: 82-87.
- [8] Srinivas N, Deb K. Multiobjective optimization using nondominated sorting in genetic algorithms[J]. Evolutionary Computation, 1994, 2(3): 221-248.
- [9] Deb K, Pratap A, Agarwal S, et al. A fast and elitist multiobjective genetic algorithm: NSGA-II[J]. IEEE Trans on Evolutionary Computation, 2002, 6(2): 182-197.
- [10] 张福威, 李军, 孟品超, 等. 多目标进化算法综述[J]. 长春光学精密机械学院学报, 2012, 35(3): 102-105.
(Zhang F W, Li J, Meng P C, et al. Survey of multi-objective evolutionary algorithms[J]. J of Changchun Institute of Optics and Fine Mechanics, 2012, 35(3): 102-105.)
- [11] Vachhani V L, Dabhi V K, Prajapati H B. Improving NSGA-II for solving multi objective function optimization problems[Z]. Piscataway: IEEE, 2016.
- [12] Li H, Zhang Q F. Multiobjective optimization problems with complicated Pareto sets, MOEA/D and NSGA-II[J]. IEEE Trans on Evolutionary Computation, 2009, 13(2): 284-302.
- [13] Deb K, Agrawal R B. Simulated binary crossover for continuous search space[J]. Complex Systems, 1994, 9(3): 115-148.
- [14] Pan X Y, Zhu J, Chen H, et al. A differential evolution-based hybrid NSGA-II for multi-objective optimization[Z]. Piscataway: IEEE, 2015.
- [15] Rainer S, Price K. Differential evolution — A simple and efficient heuristic for global optimization over continuous spaces[M]. Hingham: Kluwer Academic Publishers, 1997: 341-359.
- [16] 刘波, 王凌, 金以慧. 差分进化算法研究进展[J]. 控制与决策, 2007, 22(7): 721-729.
(Liu B, Wang L, Jin Y H. Advances in differential evolution[J]. Control and Decision, 2007, 22(7): 721-729.)
- [17] Fu Y, Huang M, Wang H, et al. An improved NSGA-II to solve multi-objective optimization problem[Z]. Piscataway: IEEE, 2014.
- [18] Zitzler E, Deb K, Thiele L. Comparison of multiobjective evolutionary algorithms: Empirical results[J]. Evolutionary Computation, 2000, 8(2): 173-195.

(责任编辑: 闫妍)