基于 Tent 映射的自适应混沌混合多目标遗传算法

聂 瑞 章卫国 李广文 刘小雄

(西北工业大学 自动化学院, 西安 710072)

摘 要: 提出一种 Tent 映射在计算机上实现的改进算法,有效解决了受计算机字长影响,Tent 映射存在不动点和小周期的情况. 将改进的 Tent 映射应用于混沌优化算法中,对基本 NSGA-Ⅱ算法进行改进. 使用混沌序列对初始种群赋值,提高算法收敛能力;使用改进的混沌搜索增强种群多样性;分别使用基本算法和改进算法对标准测试函数进行数值仿真. 统计结果显示:改进的算法可以在保持高效率求解的同时,得到的非劣解在收敛性和多样性指标上均优于基本 NSGA-Ⅱ算法.

关 键 词: 多目标优化; 遗传算法; 混沌; Tent 映射; 世代距离指标

中图分类号: TP 3

文献标识码: A **文 章 编 号**: 1001-5965(2012)08-1010-07

Adaptive chaos hybrid multi-objective genetic algorithm based on the Tent map

Nie Rui Zhang Weiguo Li Guangwen Liu Xiaoxiong

(School of Automation, Northwestern Polytechnical University, Xi'an 710072, China)

Abstract: A modified algorithm which realized the tent map on the computer was proposed to deal effectively with the problem of fixed point and the small periodic of the tent map, which affected by the finite word-length of computer. The chaos research algorithm based on the modified Tent map was introduced to the multi-objective genetic algorithm. Firstly, the chaos sequence was applied to assign the initial population value to enhance the convergence ability. Then, the chaos search optimization algorithm was adopted to improve the diversity of the population. The benchmark problems were tested by the basic and the modified algorithm respectively. The statistical results show that the improved algorithm could seek the solution more efficiently, and achieve better convergence and diversity than basic NSGA-Ⅱ.

Key words: multi-objective optimization; genetic algorithm; chaos; Tent map; generation gap distance index

多目标进化算法以其在一次运行中可以得到多个 Pareto 非劣解的优点,成为多目标优化领域的一个研究热点. 涌现出许多优秀的算法,其应用也取得了较好的效果[1-2]. 文献[3]对其算法 NSGA 的改进,是目前最优秀的多目标进化算法之一. NSGA-Ⅱ是基于 Pareto 最优概念的多目标进化算法[3-5],具有精英保留策略、密度值估计和快速非支配排序等特点,成为目前研究最多且应用最广的一种多目标进化算法. 但是,NSGA-Ⅱ采用基本模拟二进制交叉算子和多项式变异算子的搜索能力相对较弱[6],一定程度上限制了该算法的搜索性能,使算法在收敛性和多样性方面尚不能令人满意.

混沌是非线性系统中的普遍现象,混沌运动具有遍历性、随机性和对初始条件敏感性等特点[7-8]. 混沌搜索可以对陷入局部极值的个体产

收稿日期: 2011-04-02; **网络出版时间**: 2012-08-23 14:55

网络出版地址: www.cnki.net/kcms/detail/11.2625.V.20120823.1455.006.html

基金项目: 航空科学基金资助项目(20090753008)

作者简介: 聂 瑞(1985-),男,湖北麻城人,博士生,raynr@yahoo.cn.

生混沌扰动变异,有效地跳出局部最优点[9]. 文献[10]将混沌序列替代随机数发生器改善进化算法性能. 文献[11]将混沌搜索引入多目标优化算法以产生更多的非劣解. 但是在混沌优化算法中,一般使用的映射函数都是 Logistic 函数,但受到该映射函数遍历不均匀性的影响,寻优的效率并不高.

基于 Tent 函数映射的混沌序列,结构简单,迭代过程适合计算机运行[12-13]. 但是,Tent 映射受计算机字长有限的影响,小数部分的二进制序列经过一定次数的无符号左移运算将趋向全零,趋向 Tent 映射的不动点[14-15]. 在仿真中还发现,计算机迭代的序列中还存在小周期的情况,严重影响了混沌优化算法的寻优效率. 因此,本文改进了 Tent 映射在计算机上实现的算法,克服了以上存在的问题,并具有更好的遍历均匀性.

将本文改进的 Tent 映射的混沌优化算法与经典的 NSGA-Ⅱ相结合,使用混沌序列对初始种群赋值,使其在全决策空间有更加均匀的分布,提高求解效率和求解精度. 使用改进的混沌搜索增强种群多样性. 另外,使用基于本文改进的世代距离(GD, Generation Distance)指标作为混沌搜索次数自适应因子的判断条件,提出一种基于混沌优化算法的自适应混沌混合多目标遗传算法 ACHNSGA-Ⅱ(Adaptive Chaos Hybrid NSGA-Ⅱ). 最后,通过对 FON 以及 ZDT1 ~ ZDT3 标准测试函数进行仿真,验证了改进算法的有效性.

1　Tent 映射性能分析及其改进

混沌现象是非线性动态系统中出现的确定性的伪随机过程. 混沌系统[15-16]的非周期性、整体稳定而局部具有扩张性,以及对初始条件及参数异常敏感等特性都显示了混沌系统具有优良的随机性. 常用的混沌映射有 Logistic 映射和 Tent 映射.

一般将确定性方程得到的具有随机性的运动状态称为混沌. 呈现混沌状态的变量称为混沌变量. Logistic 映射是一个典型的混沌系统,其数学方程描述如下:

$$x_{n+1} = f(u, x_n) = ux_n(1 - x_n) \quad n = 0,1,2,\cdots \tag{1}$$

式中,u 为控制参数;初值 $x_0 \in [0,1]$. 当 u 确定后,由任意 x_0 可迭代出一个时间序列,当 $u = 4$ 时,式(1)将呈现完全混沌状态,所得到的时间序列是区间上[0,1]的一个满映射. Logistic 映射在混沌状态下,n 从零到无穷变化时,轨道点$\{x_n\}$在[-1,1]区间上服从切比雪夫分布[17]:

$$\rho(x_n) = \frac{1}{\pi\sqrt{1 - x_n^2}} \tag{2}$$

以初始值为 0.12345 和 0.12346 二维向量为例,迭代 5000 次的 Logistic 映射变量如图 1 所示. 从图 1 可以得出,基于 Logistic 映射的混沌变量的分布概率并不均匀. 需要通过增加迭代步数达到点集在全决策空间中较广的分布,否则,许多解集密集的分布在一部分区域,若最优点恰好在这些很难到达的区域,优化算法只能通过交叉和变异算子进化寻优,造成求解效率低,收敛速度慢. 因此,需要使用一种拥有更好遍历均匀性的映射.

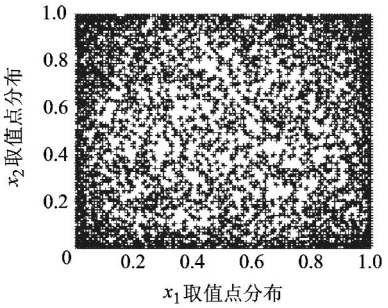


图 1　Logistic 映射变量分布

Tent 映射又称帐篷映射,一维单帐篷映射是分段线性的函数,有均匀的概率密度和功率谱密度以及理想的相关特性,其数学表达式为

$$x_{n+1} = \mu(1 - 2 \mid x_n - 0.5 \mid) \quad \mu \in (0,1] \tag{3}$$

式(3)的 Lyapunov 指数为

$$\gamma = \lim_{x\to\infty} \frac{1}{n}\sum_{i=1}^{n-1} \ln\left|\frac{\mathrm{d}x_{n+1}}{\mathrm{d}x_n}\right| = \ln(2\mu) \tag{4}$$

当$\mu > 0.5$,$\lambda > 0$ 时,式(3)是混沌的. 当$\mu = 1$时,式(3)可以写为

$$x_{k+1} = \begin{cases} 2x_k & 0 \leqslant x_k \leqslant 0.5 \\ 2(1 - x_k) & 0.5 \leqslant x_k \leqslant 1 \end{cases} \tag{5}$$

即

$$x_{k+1} = (2x_k) \bmod 1 \tag{6}$$

二进制序列的移位变换可由 Bernoulli 平移得到. 在该序列使用中,受计算机字长有限影响,小数部分的二进制序列经过一定次数的无符号左移运算将趋向全零,趋向 Tent 映射$\{0, 0.25, 0.5, 0.75\}$的不动点,如式(6)所示. 另外,分析式(5),该映射还会出现 5 以内的小周期$\{0.2, 0.4, 0.6, 0.8\}$. 尽管 Tent 映射可以得到比较均匀的解集,但以上这些情况导致计算机仿真的伪随机序列很不理想. 以初始值为 0.12345 和 0.12346 二维向量为例,迭代 5000 次的 Tent 映射变量见图 2.

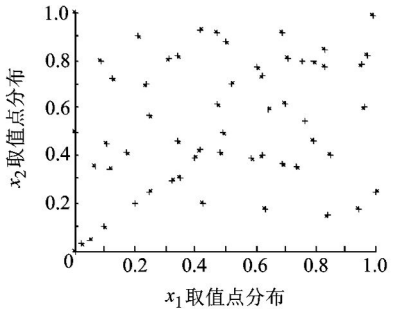


图 2 Tent 映射变量分布

尽管迭代了 5000 次,图 2 的结果得到解集的数量远比迭代次数少得多. 从得到的数据分析出,从第 56 次迭代进入不动点后,所有后面点的取值均为[0,0]. 文献[18]所附带的程序中,使用分数的表示方法进行函数构造,即每一代的计算均使用分数表示,结果如图 3 所示. 这种方法虽然解决了计算机字长带来的影响,但由于需要用到 Matlab 软件中的 sym 函数,大大降低了计算速度,影响优化算法的寻优效率. 另外,这种方法会出现规律性的迭代数值,即只要初始值不变化,使用这种方法迭代得到的每一步的变量是一个常值,及在某些确定点处的值可能永远都不会得到. 这种情况的出现会使得优化算法的多样性受损. 还有些文献使用式(7)的方式改出不动点或小周期情况.

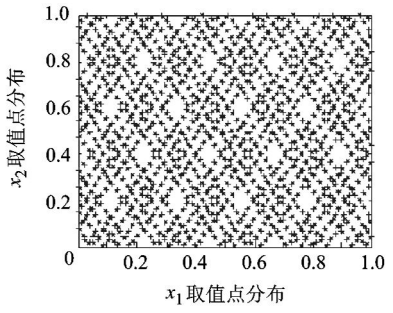


图 3 使用符号函数实现 Tent 映射变量分布

$$x_{k+1} = x_{k+1} + \xi \quad \xi \in [0,1] \tag{7}$$

混沌变量 x_{k+1}是一个取值范围有限的变量,考虑式(7)中 $0 \leqslant x_k \leqslant 1$,又已知 ξ 为随机数且 $\xi \in [0,1]$,则

$$0 \leqslant x_k + \xi \leqslant 2 \tag{8}$$

按式(7)的方法,在原有变量加上任意随机数,会导致出现跳出取值范围的情况,这是不希望出现的. 因此,本文考虑改进该算法在计算机上实现. 在每次迭代中增加一次判断,如果变量进入不动点或者 5 以内的小周期,则使用式(9)重新赋值:

$$x_{k+1} = x_{k+1}\text{rand}(0,1) \quad x \in [0,1] \tag{9}$$

式(9)的计算不会增加过多的计算复杂度,更不需要使用 sym 函数计算,也不会对算法的寻优效率造成过大的影响. 与式(7)不同,原变量与区间为[0,1]之间的随机数相乘,不会出现跳出取值范围. 由于加入了随机数,即使初值相同,在遇到小周期或不动点时,会通过式(9)生成同一区间的随机数,不会出现文献[18]的常值情况. 将其应用于优化算法中,每次迭代的结果都不相同,使优化算法可以搜索到更多的区域,增强了算法的多样性.

以初始值为 0.12345 和 0.12346 二维向量为例,改进 Tent 映射算法迭代 5 000 次的运行结果如图 4 所示. 图中点集的分布明显比图 1 和图 2 的点集密集且分布均匀,达到了预期的效果.

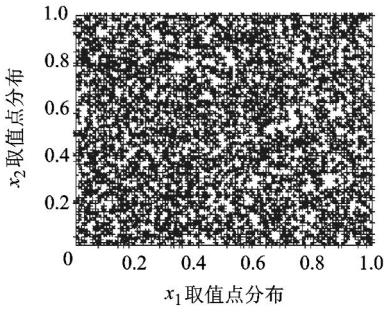


图 4 改进的 Tent 映射变量分布

将本文改进的 Tent 映射的算法步骤总结如下:

1) 参数初始化. 令 $i=1$,给定初始值 x_0,并令中间变量 $z_i = x_0$;

2) 使用式(5)对变量 z_i 进行迭代,令 $i = i+1$ 并判断是否达到最大迭代次数,若未达到则进入步骤 3);否则,进入步骤 5);

3) 判断得到的中间变量是否在小周期内或者进入不动点,即判断 $z_i = \{0, 0.25, 0.5, 0.75\}$ 或者 $z_i = z_{i-k}, k = \{1,2,3,4\}$. 判断为假则返回步骤 2),否则进入步骤 4);

4) 按式(9)形式对 z_i 进行更新,并给 x_i 赋值,返回步骤 2);

5) 终止运行,保存结果.

2 改进的 ACHNSGA-Ⅱ关键技术

2.1 种群初始值赋值

在全决策空间中,更加均匀的种群初始值分布有利于提高算法的收敛效率和求解精度. 混沌系统的基本特点就是系统对初始值的极端敏感性. 两个相差无几的初值所产生的轨迹,随着时间的推移将按指数方式分离,如图 5 所示. 两个变量的初始值设置为 0.12345 和 0.12346,经过 15 次迭代,两个相差很小的变量几乎变得完全不同了.

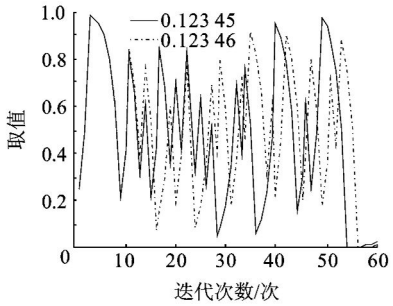


图 5　改进的 Tent 映射对于初始值的敏感性

因此,使用本文第 3 节的方法进行种群混沌初始化赋值,步骤如下:

1) 随机产生一个 D 维的随机数向量,$z_1 = (z_{11}, z_{12}, \cdots z_{1j} \cdots, z_{zD})$,$z_{1j} \in [0,1]$,其中 D 为待优化参数的个数,N 为种群规模;

2) 根据本文第 1 节改进方法,按式(10)的形式计算其余分量:

$$z_{i+1,j} = \begin{cases} 2z_{ij} & 0 \leqslant z_{ij} \leqslant 0.5 \\ 2(1 - z_{ij}) & 0.5 < z_{ij} \leqslant 1 \end{cases}$$
$$i = 1,2,\cdots,N; j = 1,2,\cdots,D \qquad (10)$$

3) 将式(10)得到的各混沌分量,按式(11)载入到各优化变量的取值范围:

$$x_{ij} = a_j + z_{ij}(b_j - a_i)$$
$$i = 1,2,\cdots,N; j = 1,2,\cdots,D \qquad (11)$$

式中,$x_{ij} \in [a_j, b_j]$ 为第 j 维变量的取值范围.

2.2　GD 指标的改进

GD 指标是用来描述算法所获得的非劣解与真实 Pareto 前端之间的距离[19],计算公式如下:

$$\mathrm{GD} = \frac{\sqrt{\sum_{i=1}^{n} d_i^2}}{n} \qquad (12)$$

式中,d_i 为第 i 个非劣解与真实 Pareto 前端 PFtrue 之间的最短距离. 如果得到的非劣解不能完全覆盖 PFtrue 或者只能得到真实前沿中的一部分解,但优化得到的这一部分解可以很好地逼近真实前沿. 这就意味着真实前沿的其它一部分解并没有通过优化得到. 图 6 所示是基本 NSGA-Ⅱ优化 ZDT3 的结果,用式(12)求取这组值的世代距离指标 $\mathrm{GD} = 1.92 \times 10^{-4}$,$f_1$,$f_2$ 为标准测试函数. 图 7 所示是本文改进算法求得的结果,不仅很好地逼近了 Pareto 真实前沿,而且其多样性也很好. 如果按式(12)的计算结果为:$\mathrm{GD} = 6.65 \times 10^{-4}$,比图 6 的结果差. 这个结果显然不能综合反映两个算法的优劣. 仅就解的逼近程度而言,在 10^{-4} 这个数量级上都可以接受. 通常还可以通过加大迭代次数的方法增加求解精度. 从图中直观地可以看出,图 6 显然比图 7 的求解效果差,尤其反映在其多样性上. 这说明该指标不能用于对算法求解多样性进行评价.

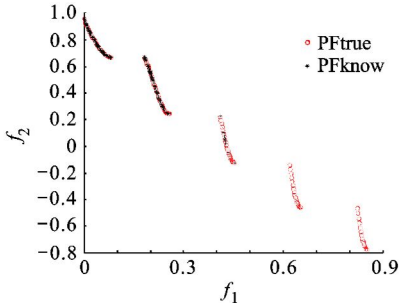


图 6　基本 NSGA-Ⅱ优化 ZDT3 结果

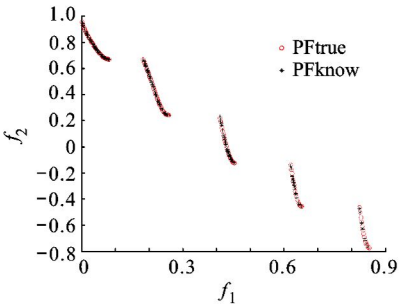


图 7　ZDT2 的 Pareto 前沿

为了使 GD 指标能够反映算法多样性性能,在式中,令 d_i 表示第 i 个真实 Pareto 前端的解与算法运行的所有 Pareto 非劣解之间的最短距离. 这样,在多样性不足的情况下,没有被优化算法求得,但在 PFtrue 上的点集与优化求得的点集之间的距离也会加入该指标,使得指标值变大,意味着优化效果变差.

改进后,再次计算图 6 的 GD 指标结果为 0.021 4,而图 7 的 GD 指标结果为 6.93×10^{-4}. 这个结果能够客观的反应两种算法在该测试函数上的求解效果. 说明改进的 GD 指标不仅可以表示解收敛性,同时也能在一定程度上反应求得的非劣解的多样性.

2.3　自适应混沌搜索算子

将混沌搜索算法加入基本的 NSGA-Ⅱ中,增强算法的搜索能力. 将当前代种群值载入混沌变量进行混沌搜索,在最大迭代次数内,若迭代得到的解支配所有当前种群,则结束迭代并记录当前非劣解;若始终没有得到非劣解,则迭代至最大迭代次数后停止迭代. 通常,最大迭代次数是根据经验定义的. 最大迭代次数过大会增加算法复杂性;最大迭代次数过小会降低混沌载波变量得到非劣解的概率.

另外,算法迭代开始,全部种群由于使用了混沌初始赋值而均匀的分布于整个决策空间,种群收敛速度比较快,不需要过多的混沌搜索;但随着

迭代次数的增加,一部分解陷入局部最优或者解的多样性较差的时候,种群的变化很小,进化效率降低,这时就需要使用混沌优化进行深度搜索. 因此,本文考虑使用与迭代次数、收敛性和多样性都相关的自适应算子为当前最大迭代次数赋值. 本文改进的 GD 指标可以综合评价种群的多样性和收敛性,这里使用该指标进行最大迭代次数自适应增减的判断,令算法向着该指标值不断减小的方向进化.

GD 指标需要已知真实 Pareto 前沿,但在实际应用中很难得到. 这里借用这个指标的概念,在每次迭代中,动态的计算两次迭代之间 Pareto 前沿的距离 GD^*. 假设有 3 次迭代的 Pareto 前沿数据为 P_1, P_2, P_3,分别计算 $GD_{P_2P_1}$ 和 $GD_{P_3P_2}$. 按下式计算 d_{GD}:

$$d_{GD} = \frac{GD_{P_3P_2}}{GD_{P_2P_1}} \tag{13}$$

当 $d_{GD} < 1$,说明算法收敛速度很快,减小混沌搜索的最大迭代次数,提高算法运行效率;当 $d_{GD} > 1$,说明算法收敛速度变慢,需要增大混沌搜索的最大迭代次数,进行深度搜索得到更多的非劣解,增强种群的收敛性和增加种群的多样性.

设混沌搜索的最大和最小迭代次数分别为 $[C_{max}, C_{min}]$,当前种群进化迭代代数为 t,种群最大进化代数为 t_{max},则当前代混沌搜索最大迭代次数 M 的值进行更新:

$$M(t) = \begin{cases} C_{min} + \dfrac{C_{max} - C_{min}}{t_{max}} t & d_{GD(t)} > 1 \\ C_{min} & d_{GD(t)} = 0 \\ C_{max} - \dfrac{C_{max} - C_{min}}{t_{max}} t & d_{GD(t)} < 1 \end{cases} \tag{14}$$

这里给出算法进行局部混沌搜索的步骤如下:

1) 令 $k = 1$(当前混沌搜索的迭代代数),初始化混沌搜索最大最小步数. 由于从第 3 代才能得出 $GD_{P_2P_1}$ 和 $GD_{P_3P_2}$,又已知算法开始的收敛效率很高,因此,前两代的 M 值按最小混沌迭代次数 C_{min} 计算. 从第 3 代开始,按式计算 d_{GD} 的值并按式(14)动态更新 M 的值;

2) 从当前种群中选取随机选取 20% 的 N 个解作为待搜索的决策变量的初值 $x_{i,j}^k$, $i = 1, 2, \cdots, N$, $j = 1, 2, \cdots, D$, D 为变量维数, k 为混沌最大迭代次数.

3) 随机选择 N 组解 x_i, $i = 1, 2, \cdots, N$,将其按下式映射为[0,1]之间的混沌变量:

$$s_j^k = \frac{x_j^k - x_{min,j}}{x_{max,j} - x_{min,j}} \tag{15}$$

其中, $x_{max,j}$ 和 $x_{min,j}$ 为第 j 维变量的搜索上下界;

4) 按式(10)的形式迭代混沌变量,并按本文第 1 节的方法进行混沌小周期和不动点的判断及改进小周期的方法;

5) 按式(16)将混沌变量转化为决策变量:

$$x_j^{k+1} = x_{min,j} + s_j^{k+1}(x_{max,j} - x_{min,j}) \tag{16}$$

6) 根据决策变量 x_j^{k+1} 对新解进行评价,若新解可以支配当前种群其它解或搜索次数 k 已达最大步数,则记录新解并结束当前混沌搜索,转到步骤 3)进行下一组解 $x_{i+1,j}^k$ 的混沌优化. 否则,置 $k = k + 1$,并转到步骤 4).

2.4 ACHNSGA-Ⅱ算法步骤

将本文提出的 ACHNSGA-Ⅱ算法步骤整理如下:

1) 按 2.1 节的方法产生初始种群 P_0;

2) 对 P_0 进行非劣排序,每个个体被赋予 Pareto 秩;再对种群执行二元锦标赛选择、交叉和变异,得到新的种群 Q_0,令当前进化代数 $t = 0$;

3) 在第 t 代形成新的群体 $R_t = Q_t \cup P_t$,对种群 R_t 进行非劣排序,得到非劣前端 $F_1, F_2, \cdots$;

4) 计算所有 F_i(F_i 为第 i 个非劣前端)的拥挤距离;

5) 进行偏序选择,选择其中最好的 L 个个体形成种群 P_{t+1};

6) 对种群 P_{t+1} 执行复制、交叉和变异,形成种群 Q_{t+1}^*;

7) 按本文 2.3 节的方法对 Q_{t+1}^* 中的 N 个种群值进行自适应混沌搜索,并将得到的非劣解与步骤 6)中的解 Q_{t+1}^* 合并形成新的种群 Q_{t+1};

8) 如果终止条件成立,则结束;否则, $t = t + 1$,转到步骤 2).

3 仿真结果及分析

将本文改进的算法与文献[3]中基本 NSGA-Ⅱ进行比较. NSGA-Ⅱ初始种群 150,迭代 500 次. 结果输出所有 Pareto 秩为 1 的个体. ACHNSGA-Ⅱ算法其它参数设置均与基本算法相同. 另外,设置混沌迭代最大最小次数为 50 次和 30 次,每代选取 20% 的种群进行混沌搜索,总的进化迭代次数为 50 次.

对标准测试函数 FON 和 ZDT1 ~ ZDT3 进行测试. 测试函数的具体描述详见文献[2]. 每种测试函数分别使用两种方法优化 30 次,使用本文改

进的 GD 指标,Pareto 前沿覆盖范围 MS 和多样性指标 SP 综合评价算法的结果. 除本文改进的 GD 指标,其它各指标的具体描述见文献[11]. 算法运行的计算机配置为双核 2.33 GHz,2G 的 Hp PC 机上完成.

通过 ACHNSGA-Ⅱ优化后的各测试函数的 Pareto 前沿 PFknow 和真实 Pareto 前沿 PFtrue 的仿真图如图 8 ~ 图 10 所示.

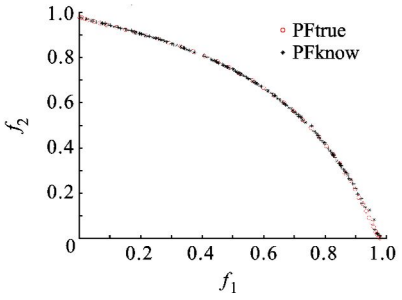


图 8　FON 的 Pareto 前沿

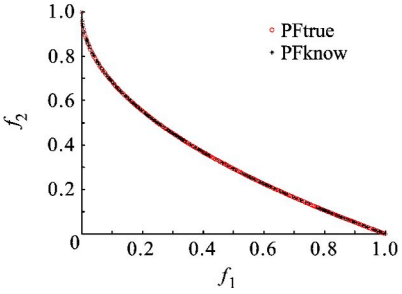


图 9　ZDT1 的 Pareto 前沿

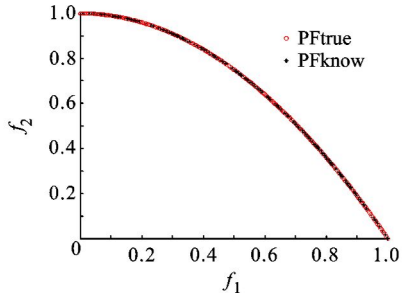


图 10　ZDT2 的 Pareto 前沿

从表 1 得出,ACHNSGA-Ⅱ算法比 NSGA-Ⅱ算法在优化 ZDT3 上的改进效果最为明显,从图 7 可以得出,ZDT3 的真实前沿是分段连续的,说明 ACHNSGA-Ⅱ算法在保持算法多样性方面取得了预期的效果. 表 2 说明改进算法获得的非劣解集在 Pareto 前沿拥有更好的覆盖范围. 表 3 中的 ZDT1-3 函数的改进效果也较明显. 相比较而言,表 1 中改进的 GD 指标的改进效果最为明显. 这与将 GD 指标作为优化算法的自适应参数的判断条件有关,使得算法不断的向着 GD 减小的方向优化. 从表 4 也可以看出,ACHNSGA-Ⅱ算法的运行时间均比 NSGA-Ⅱ算法短.

表 1　改进的 GD 指标比较

算法		FON	ZDT1	ZDT2	ZDT3
NSGA-Ⅱ	均值	1.68×10^{-3}	2.54×10^{-4}	2.16×10^{-4}	2.52×10^{-2}
	方差	4.17×10^{-4}	3.21×10^{-5}	1.80×10^{-5}	9.82×10^{-3}
ACHN SGA-Ⅱ	均值	1.26×10^{-3}	1.51×10^{-4}	1.48×10^{-4}	7.37×10^{-4}
	方差	6.08×10^{-4}	9.23×10^{-6}	6.99×10^{-6}	1.63×10^{-4}
均值对比效果/%		24.97	40.37	47.72	97.07

表 2　Pareto 前沿覆盖范围 MS 指标比较

算法		FON	ZDT1	ZDT2	ZDT3
NSGA-Ⅱ	均值	9.85×10^{-1}	1	1	7.38×10^{-1}
	方差	6.32×10^{-3}	0	0	8.80×10^{-2}
ACHN SGA-Ⅱ	均值	9.91×10^{-1}	1	1	1
	方差	8.53×10^{-3}	0	0	7.45×10^{-5}
均值对比效果/%		0.52	0	0	35.58

表 3　多样性 SP 指标比较

算法		FON	ZDT1	ZDT2	ZDT3
NSGA-Ⅱ	均值	5.94×10^{-3}	4.90×10^{-3}	4.30×10^{-3}	9.52×10^{-3}
	方差	7.00×10^{-4}	7.40×10^{-4}	2.28×10^{-4}	5.12×10^{-3}
ACHN SGA-Ⅱ	均值	5.93×10^{-3}	3.53×10^{-3}	3.48×10^{-3}	7.60×10^{-3}
	方差	6.29×10^{-4}	4.50×10^{-4}	3.44×10^{-4}	1.71×10^{-3}
均值对比效果/%		0.11	27.89	18.99	20.17

表 4　两种优化算法运行时间　　s

算法		FON	ZDT1	ZDT2	ZDT3
NSGA-Ⅱ	均值	34.5	25.83	26.5	32.8
	方差	0.222	0.0929	0.250	0.379
ACHNSGA-Ⅱ	均值	19.7	13.3	13.8	19.2
	方差	0.298	0.0368	0.230	0.241
均值对比效果/%		43.05	48.4	47.72	41.45

4　结　论

改进的基于 Tent 映射的混沌算法比 Logistic 映射有更加均匀的遍历性. 使用基于改进 Tent 映射的混沌序列对种群初始值赋值,使初始种群可以在全决策空间更加均匀地分布. 通过在基本算法中加入基于改进 Tent 映射的混沌序列的混沌搜索,能够令种群得到较快的收敛性的同时,获得满意的多样性. 改进的世代距离指标能够更加准确的反应种群收敛性的同时,在一定程度上反应种群的多样性.

虽然使用混沌优化算法增加了算法单次运行的复杂度,但减少了改进算法总的运行次数. 而且使用改进的世代距离指标作为混沌最大迭代次数

自适应判断条件,可以进一步加快求解效率,减少优化算法求解复杂度.

参考文献(References)

[1] 公茂果,焦李成,杨咚咚,等. 进化多目标优化算法研究[J]. 软件学报,2009(2):271-289
Gong Maoguo, Jiao Licheng, Yang Dongdong, et al. Evolutionary multi-objective optimization algorithms[J]. Journal of Software, 2009,20(2):271-289(in Chinese)

[2] Li H, Zhang Q. Multiobjective optimization problems with complicated Pareto sets, MOEA/D and NSGA-Ⅱ[J]. Evolutionary Computation, 2009,13(2):284-302

[3] Deb K, Pratap A, Agarwal S, et al. A fast and elitist multiobjective genetic algorithm: NSGA-Ⅱ[J]. Evolutionary Computation, 2002,6(2):182-197

[4] Köppen M, Yoshida K. Substitute distance assignments in NSGA-Ⅱ for handling many-objective optimization problems[C]// Shigeru Obayashi, Kalyanmoy Deb. Evolutionary Multi-Criterion Optimization. Berlin: Springer, 2007:727-741

[5] Deb K, Rao N U, Karthik S. Dynamic multi-objective optimization and decision-making using modified NSGA-Ⅱ: a case study on hydro-thermal power scheduling[C]// Shigeru Obayashi, Kalyanmoy Deb. Evolutionary Multi-Criterion Optimization. Berlin: Springer, 2007:803-817

[6] Coello C A C, Lamont G B, Van Veldhuizen D A. Evolutionary algorithms for solving multi-objective problems[M]. New York: Kluwer Academic Publishers, 2002:35-39

[7] 郝柏林. 从抛物线谈起[M]. 上海:上海科技教育出版社, 1993:26-45
Hao Bolin. Speaking of parabola[M]. Shanghai: Shanghai Scientific & Technological Education Publishing House, 1993:26-45(in Chinese)

[8] 汪仲清,薛皓,李婧,等. 基于双帐篷映射的伪混沌编码器及其解码方法[J]. 重庆邮电大学学报:自然科学版,2010, 22(2):200-204
Wang Zhongqing, Xue Hao, Li Jing, et al. Realization scheme of pseudo-chaotic encoder and decoder based on double-tentmapping[J]. Journal of Chongqing University of Posts and Telecommunications: Natural Science Edition, 2010, 22(2): 200-204 (in Chinese)

[9] Alatas B, Akin E, Ozer A B. Chaos embedded particle swarm optimization algorithms[J]. Chaos, Solitons & Fractals, 2009, 40(4):1715-1734

[10] Caponetto R, Fortuna L, Fazzino S, et al. Chaotic sequences to improve the performance of evolutionary algorithms[J]. Evolutionary Computation, 2003, 7(3):289-304

[11] 雷德明,严新平,吴智铭. 多目标混沌进化算法[J]. 电子学报,2006,34(6):1142-1145
Lei Deming, Yan Xinping, Wu Zhiming. Multi-objective chaotic evolutionary algorithm[J]. Acta Electronica Sinica, 2006, 34(6):1142-1145(in Chinese)

[12] 单梁,强浩,李军,等. 基于Tent映射的混沌优化算法[J]. 控制与决策,2005,20(2):179-182
Shan Liang, Qiang Hao, Li Jun, et al. Chaotic optimization algorithm based on Tent map[J]. Control and Decision, 2005, 20(2):179-182(in Chinese)

[13] Tavazoei M S, Haeri M. Comparison of different one-dimensional maps as chaotic search pattern in chaos optimization algorithms[J]. Applied Mathematics and Computation, 2007, 187(2): 1076-1085

[14] 张浩,张铁男,沈继红,等. Tent混沌粒子群算法及其在结构优化决策中的应用[J]. 控制与决策,2008,23(8):857-862
Zhang Hao, Zhang Tienan, Shen Jihong, et al. Research on decision-makings of structure optimization based on improved Tent PSO[J]. Control and Decision, 2008, 23(8): 857-862(in Chinese)

[15] 程志刚,张立庆,李小林,等. 基于Tent映射的混沌混合粒子群优化算法[J]. 系统工程与电子技术,2007,29(1): 103-106
Cheng Zhigang, Zhang Liqing, Li Xiaolin, et al. Chaotic hybrid particle swarm optimization algorithm based on Tent map[J]. Systems Engineering and Elect ronics, 2007, 29(1): 103-106 (in Chinese)

[16] 高鹰,谢胜利. 混沌粒子群优化算法[J]. 计算机科学,2004, 31(8):13-15
Gao Ying, Xie Shengli. Chaos particle swarm optimization algorithm[J]. Computer Science, 2004, 31(8): 13-15 (in Chinese)

[17] 盛煜翔,潘海天,夏陆岳,等. 混合混沌粒子群算法在苯与甲苯闪蒸过程优化中的应用[J]. 浙江工业大学学报,2010, 38(3):318-321
Sheng Yuxiang, Pan Haitian, Xia Luyue, et al. Hybrid chaos particle swarm optimization algorithm and application in benzene-toluene flash vaporization[J]. Journal of Zhejiang University of Technology, 2010, 38(3): 318-321(in Chinese)

[18] Lynch S. Dynamical systems with applications using MATLAB[M]. Boston: Birkhäuser Boston Product, 2004:69-83

[19] 雷德明,严新平. 多目标智能优化算法及其应用[M]. 北京:科学出版社,2009:389-390
Lei Deming, Yan Xinping. Multi-objective evolutionary optimization and application[M]. Beijing: Science Press, 2009:389-390(in Chinese)

(编 辑:刘登敏)