

文章编号: 1001-0920(2012)01-0041-06

多目标强度 Pareto 混沌差分进化算法

章 萌, 章卫国, 孙 勇

(西北工业大学 自动化学院, 西安 710072)

摘 要: 提出一种多目标强度 Pareto 混沌差分进化算法 (SPCDE). 首先利用 Tent 映射进行种群的混沌初始化, 采用一种基于均匀排挤机制的截断排挤操作和混沌替换操作进行种群的环境选择操作; 然后基于一种变缩放因子的差分变异策略进行变异操作, 通过计算支配关系得到变异个体; 最后通过支配关系的计算和环境选择操作进行进化选择操作并得到子代个体. 以上操作不仅提高了算法的收敛性能, 而且保证了 Pareto 最优解的均匀分布性. 数值实验结果表明了该算法的有效性.

关键词: 多目标优化; 强度 Pareto; 差分进化; 混沌 Tent 映射; DE/current-to-best/1/bin 变异策略

中图分类号: TP18

文献标识码: A

Multi-objective strength Pareto chaotic differential evolution algorithm

ZHANG Meng, ZHANG Wei-guo, SUN Yong

(College of Automation, Northwestern Polytechnical University, Xi'an 710072, China. Correspondent: ZHANG Meng, E-mail: zmnwpu@126.com)

Abstract: A multi-objective strength Pareto chaotic differential evolution algorithm (SPCDE) is proposed. Firstly, the chaotic initialization based on the Tent map is adopted to initialize the population. A truncation crowding operation based on a uniform crowding mechanism and a chaotic substitution operation are introduced to the environmental selection operation of the population. Then, the differential mutation operation is operated based on a differential mutation strategy with changing scaling factor, and the mutation individuals are obtained by computing the dominance relation. Finally, the evolutionary selection operation is operated and the offspring individuals are obtained by the dominance relation computation and the environmental selection operation. The operations above mentioned not only enhance the convergence performance of the proposed algorithm, but also maintain the uniformity of the Pareto optimal solution. Numerical experiment results show the effectiveness of the proposed algorithm.

Key words: multi-objective optimization; strength Pareto; differential evolution; chaotic Tent map; DE/current-to-best/1/bin mutation strategy

1 引 言

科学计算和工程实践中的优化问题大多数是多目标优化问题 (MOPs). 进化算法在优化计算中能够较好地解决非连续、非凸、不可微、多峰值等问题. 因此, 在 MOPs 的求解中得到了广泛的应用, 产生了较为经典且常用的算法, 如 NSGA-II^[1], 改进的强度 Pareto 进化算法 (SPEA2)^[2] 等, 其中 SPEA2 是目前公认的具有较强优化性能的多目标优化算法.

差分进化 (DE) 算法^[3] 是一种随机并行搜索的优化算法, 该算法具有简单易用、鲁棒性和全局优化性能强等优点, 已在很多单目标优化问题 (SOPs) 中得到

了广泛应用并取得了较满意的效果^[4]. 近年来, 许多学者将差分进化算法扩展应用于解决 MOPs, 相继产生了比较成功的算法, 如: Abbass^[5] 将 Pareto 支配概念与 DE 算法相结合提出的 PDE 算法, Madavan^[6] 提出的 Pareto 差分进化算法 (PDEA) 和 Robic 等人^[7] 将非支配排序和拥挤距离的概念与 DE 算法相结合而提出的 DEMO 算法等.

文献 [8] 将 Logistic 混沌映射应用于多目标差分进化算法, 提出了多目标混沌差分进化算法 (CDEMO), 初步将混沌操作引入多目标差分进化算法中. 文献 [9] 研究了基于 Tent 映射的混沌优化算法,

收稿日期: 2010-10-18; 修回日期: 2010-12-15.

基金项目: 航空科学基金项目 (20090753008).

作者简介: 章萌 (1982-), 男, 博士生, 从事智能优化算法和飞行控制系统设计的研究; 章卫国 (1956-), 男, 教授, 博士生导师, 从事飞行控制系统等研究.

指出在优化问题中 Tent 映射比 Logistic 映射具有更好的遍历均匀性和更快的迭代速度, 因此具有广阔的应用前景. 但是, 目前 Tent 映射在多目标差分进化算法中的应用并不常见.

对于 MOPs 而言, 维持种群的多样性与提高群体的收敛速度同样重要, 在多目标差分进化算法中, 可以选择合适的差分变异策略来满足上述两方面的要求. 目前已有的多目标差分进化算法大都是基于 DE/rand/1/bin 差分变异策略的, 该策略虽然较好地保持了种群的多样性, 但并没有利用种群当前的最优信息来指导种群个体的进化以提高算法的收敛性能. 而 DE/current-to-best/1/bin 差分变异策略既能较好地保持种群的多样性, 又利用了种群当前的最优信息指导种群中个体朝着最优方向进化. 在 SOPs 中, 基于该差分变异策略的差分进化算法的收敛性能和收敛速度明显优于 DE/rand/1/bin 差分变异策略. 因此, 应将 DE/current-to-best/1/bin 差分变异策略应用于多目标差分进化算法中.

本文借鉴 SPEA2 的主要思想, 并结合差分进化算法的基本操作和混沌 Tent 映射, 提出了多目标强度 Pareto 混沌差分进化算法 (SPCDE). 该算法在个体适应度分配操作、环境选择操作和算法流程上借鉴了 SPEA2 的方法. 此外, 引入基于 Tent 映射的种群混沌初始化操作, 提高了初始种群的质量. 在环境选择操作上, 通过引入一种均匀排挤机制改进了 SPEA2 中的截断排挤操作, 同时改进了环境选择操作的适用范围, 并将基于 Tent 映射的混沌替换操作引入环境选择操作. 在种群进化方面, 将基于变缩放因子的 DE/current-to-best/1/bin 差分变异策略和含有支配关系计算的差分变异操作、交叉操作、含有支配关系计算以及环境选择操作的差分选择操作引入算法中, 以替代 SPEA2 中的进化操作.

2 多目标优化问题的描述

以最小化问题为例, MOPs 可以表述如下:

$$\min f(X) = [f_1(X), \dots, f_i(X), \dots, f_n(X)],$$

$$X = (x_1, \dots, x_j, \dots, x_m),$$

$$x_{j \min} \leq x_j \leq x_{j \max}, j = 1, 2, \dots, m. \quad (1)$$

其中: $f(X) \in R^n$ 为目标向量, $f_i(X) (i = 1, 2, \dots, n)$ 为第 i 个目标函数; X 为可行的决策向量, $x_j (j = 1, 2, \dots, m)$ 为 X 的第 j 个分量, 即第 j 个决策变量, $x_{j \min}$ 和 $x_{j \max}$ 为其最小和最大值.

下面给出 MOPs 中常用的几个基本定义.

定义 1 (Pareto 占优) 对于可行域内的决策向量 $X_1, X_2 \in X$, X_1 是 Pareto 占优 (非支配) 的或 X_1 支配 X_2 , 当且仅当下式成立并记为 $X_1 \succ X_2$:

$$(\forall i \in (1, 2, \dots, n) : f_i(X_1) \leq f_i(X_2)) \wedge$$

$$(\exists k \in (1, 2, \dots, n) : f_k(X_1) < f_k(X_2)). \quad (2)$$

定义 2 (Pareto 最优解) 若某一可行的决策向量 $X^* \in X$ 被称为 Pareto 最优解, 当且仅当满足如下条件:

$$\neg \exists \tilde{X} \in X : \tilde{X} \succ X^*, \quad (3)$$

其中 \tilde{X} 为可行域内的任意解. Pareto 最优解也称为非劣解或非支配解.

定义 3 (Pareto 最优解集) 所有 Pareto 最优解组成的矢量集合 P_S 称为 Pareto 最优解集, 其定义为

$$P_S = \{X^* | \neg \exists \tilde{X} \in X : \tilde{X} \succ X^*\}. \quad (4)$$

定义 4 (Pareto 前沿) Pareto 最优解集 P_S 中的解对应的目标向量组成的集合 P_F 称为 Pareto 前沿.

3 SPEA2 算法

3.1 SPEA2 的主要操作

SPEA2 由一个种群 P 和一个档案集 \bar{P} 构成. 主要采用了适应度分配操作和环境选择操作.

1) 适应度分配操作

对于种群 P 和档案集 \bar{P} 中的每一个个体都会分配一个强度 $S(i)$, 它表示第 i 个个体所支配的个体数量, 其定义为

$$S(i) = |\{j | j \in P_t \cup \bar{P}_t \wedge i \succ j\}|. \quad (5)$$

其中: $|\cdot|$ 为集合的规模, P_t 和 \bar{P}_t 分别为第 t 代种群和档案集.

在得到个体 i 的强度值 $S(i)$ 后, 个体 i 的原始适应度值 $R(i)$ 定义为

$$R(i) = \sum_{j \in (P_t \cup \bar{P}_t), j \succ i} S(j). \quad (6)$$

$R(i)$ 表示种群和档案集中优于第 i 个个体的所有个体的强度之和. $R(i)$ 越小, 表明第 i 个个体支配的个体越多; $R(i) = 0$ 表明个体 i 是一个非支配个体.

当大部分个体的原始适应度值相同时, 其 Pareto 支配关系将变得难以确定, 因此还需要引入个体密度的概念来表征个体间的拥挤程度, 从而进一步区分个体间的差异. 对每一个个体 i 到 P_t 和 \bar{P}_t 中其他个体的距离按升序排列, 有

$$d_1(i) \leq d_2(i) \leq \dots \leq d_n(i). \quad (7)$$

k 临近法所定义的个体 i 到其第 k 个邻近个体的距离 σ_i^k 为

$$\sigma_i^k = \frac{k}{2nd_k(i)}, \quad (8)$$

则个体 i 的密度估计函数 $D(i)$ 定义为

$$D(i) = 1/(\sigma_i^k + 2), \quad (9)$$

其中 $k = \sqrt{N + \bar{N}} (|P_t| = N, |\bar{P}_t| = \bar{N})$.

个体 i 的精确适应度值 $F(i)$ 由其原始适应度值 $R(i)$ 和其密度 $D(i)$ 组成, 即

$$F(i) = R(i) + D(i). \quad (10)$$

2) 环境选择操作

环境选择操作主要是用来维护和更新档案集合. 首先由式 (11) 将 P_t 和 \bar{P}_t 中所有非支配个体 (即适应度值小于 1 的个体) i 复制到 \bar{P}_{t+1} 中, 即

$$\bar{P}_{t+1} = \{i | i \in P_t \cup \bar{P}_t \wedge F(i) < 1\}. \quad (11)$$

因为档案集合的规模 \bar{N} 是固定的, 在将非支配个体复制到档案集合 \bar{P}_{t+1} 后, 接下来的操作可分为 3 种情况: 1) 若 $|\bar{P}_{t+1}| = \bar{N}$, 则环境选择操作结束; 2) 若 $|\bar{P}_{t+1}| < \bar{N}$, 则将 P_t 和 \bar{P}_t 中适应度值较小的 $\bar{N} - |\bar{P}_{t+1}|$ 个受支配个体复制到 \bar{P}_{t+1} 中; 3) 若 $|\bar{P}_{t+1}| > \bar{N}$, 则采用截断排挤操作. 在此定义一个截断排挤偏序关系为

$$i \succ_{dj} \Leftrightarrow \begin{cases} \forall 0 < k < |\bar{P}_{t+1}| : \sigma_i^k = \sigma_j^k \vee \exists 0 < k < |\bar{P}_{t+1}| : \\ [(\forall 0 < l < k : \sigma_i^l = \sigma_j^l) \wedge \sigma_i^k < \sigma_j^k]. \end{cases} \quad (12)$$

通过这种截断排挤将那些周围个体密度较大的个体从档案集合中排挤出去, 从而保证其解的分布均匀性, 直到其规模恢复到 \bar{N} 为止.

3.2 SPEA2 的算法步骤

Step 1: 初始化. 令 $t = 0$, 确定种群规模 N , 档案集合规模 \bar{N} , 最大进化代数 T 和随机初始化种群 P_0 , 并令档案集合 $\bar{P}_0 = \emptyset$.

Step 2: 适应度赋值. 由适应度分配操作计算 P_t 和 \bar{P}_t 中所有个体的适应度值.

Step 3: 环境选择. 进行环境选择操作.

Step 4: 终止条件判断. 如果 $t < T$, 则进行 Step 5; 否则, 终止算法, \bar{P}_{t+1} 中的个体即为 Pareto 最优解, \bar{P}_{t+1} 即为 Pareto 最优解集.

Step 5: 交配选择. 利用锦标赛选择方法选择 \bar{P}_{t+1} 中的个体进入交配池.

Step 6: 进化操作. 对交配池内个体进行重组和变异操作产生新的个体进入 P_{t+1} , 令 $t = t + 1$ 并返回 Step 2.

4 强度 Pareto 混沌差分进化算法 (SPCDE)

4.1 SPCDE 算法的主要思想

SPCDE 算法仍由种群 P_t 和档案集合 \bar{P}_t 构成, 在种群 P_t 初始化方面引入了基于 Tent 映射的混沌初始化, 从而提高了初始种群个体的质量和分布的均匀性.

SPCDE 算法在适应度分配上完全按照 SPEA2 的

适应度分配操作方法进行.

SPCDE 算法在环境选择操作中的截断排挤操作上, 首先通过一种均匀排挤机制改进了截断排挤操作, 同时, 改变了环境选择操作的适用范围, 在 $|\bar{P}_{t+1}| < \bar{N}$ 的情况下也对档案集合 \bar{P}_{t+1} 进行截断排挤操作, 将个体密度较大的非支配个体排挤出档案集合; 然后引入混沌替换操作, 把基于 Tent 映射产生的混沌替换个体补充到 \bar{P}_{t+1} 中, 使得档案集合规模恢复为 $|\bar{P}_{t+1}| = \bar{N}$.

在进化操作上, SPCDE 算法结合 MOPs 的要求和差分进化算法的主要操作, 基于一种变缩放因子的 DE/current-to-best/1/bin 差分变异策略计算支配关系并进行差分变异操作, 基于支配关系计算和环境选择操作进行差分选择操作. 通过上述措施有效提高了算法的收敛性能和速度.

4.2 SPCDE 算法的主要操作

4.2.1 种群混沌初始化操作

为提高初始种群的质量, 将基于 Tent 映射的混沌序列引入 SPCDE 算法的种群初始化中.

首先取初值分别介于 0 和 1 之间的随机数 r_{j1}^0 ($j = 1, 2, \dots, m$). 然后根据 Tent 映射的形式产生向量 $r_{i+1}^0 = (r_{1,i+1}^0, \dots, r_{j,i+1}^0, \dots, r_{m,i+1}^0)$ 的各个分量 $r_{j,i+1}^0$, 即

$$r_{j,i+1}^0 = \begin{cases} 2r_{ji}^0, & 0 \leq r_{ji}^0 \leq 0.5; \\ 2(1 - r_{ji}^0), & 0.5 \leq r_{ji}^0 \leq 1; \end{cases} \quad j = 1, 2, \dots, m; i = 1, 2, \dots, N - 1. \quad (13)$$

再将产生的混沌变量映射到决策变量空间 ($x_{j \min}, x_{j \max}$) 内, 得到初始种群的第 i 个个体 $X_i^0 = (x_{1i}^0, \dots, x_{ji}^0, \dots, x_{mi}^0)$ 的第 j 个分量 x_{ji}^0 为

$$x_{ji}^0 = x_{j \min} + (x_{j \max} - x_{j \min})r_{ji}^0, \quad j = 1, 2, \dots, m, i = 1, 2, \dots, N. \quad (14)$$

4.2.2 改进的环境选择操作

1) 基于均匀排挤机制的截断排挤操作

SPEA2 中的密度估计函数只关注与个体相邻的第 k 个最近距离. 如果周围存在着多于 k 个个体紧密相邻甚至是重合的, 这时算法的密度估计函数不能真实地反映出实际个体的聚集情况, 原有的截断排挤操作可能将存在微小间隔或重合的个体有效排挤出档案集合, 也可能将这些个体全部保留, 这显然不利于个体的均匀分布. 为此, 给出一种基于均匀排挤机制的截断排挤操作.

档案集合中的任意两个个体 \bar{p}_i 和 \bar{p}_{i+1} 之间的距离 $d(\bar{p}_i, \bar{p}_{i+1})$ 可定义为

$$d(\bar{p}_i, \bar{p}_{i+1}) = \sum_{k=1}^m |\bar{p}[i].f_k - \bar{p}[i+1].f_k|, \quad (15)$$

则个体 \bar{p}_1 和 $\bar{p}_{\bar{N}}$ 的距离为

$$\begin{aligned} d(\bar{p}_1, \bar{p}_{\bar{N}}) &= \sum_{k=1}^m |\bar{p}[1].f_k - \bar{p}[\bar{N}].f_k| = \\ &= \sum_{k=1}^m |\bar{p}[1].f_k - \bar{p}[2].f_k + \cdots + \\ &= \bar{p}[\bar{N}-1].f_k - \bar{p}[\bar{N}].f_k|. \end{aligned}$$

由不等式关系可得

$$\begin{aligned} d(\bar{p}_1, \bar{p}_{\bar{N}}) &\leq \sum_{k=1}^m |\bar{p}[1].f_k - \bar{p}[2].f_k| + \cdots + \\ &= \sum_{k=1}^m |\bar{p}[\bar{N}-1].f_k - \bar{p}[\bar{N}].f_k|. \end{aligned}$$

假设档案集中个体分布绝对均匀, 则有

$$d(\bar{p}_1, \bar{p}_{\bar{N}}) \leq (\bar{N}-1) \sum_{k=1}^m |\bar{p}[i].f_k - \bar{p}[i+1].f_k|,$$

即

$$d(\bar{p}_1, \bar{p}_{\bar{N}}) \leq (\bar{N}-1)d(\bar{p}_i, \bar{p}_{i+1}).$$

由上可得, 个体 i 与 $i+1$ 间的均匀距离应满足

$$d(\bar{p}_i, \bar{p}_{i+1}) \geq \frac{d(\bar{p}_1, \bar{p}_{\bar{N}})}{\bar{N}-1}. \quad (16)$$

因此, 可定义如下均匀排挤基准:

$$\sigma_{\text{crowd}} = \frac{d(\bar{p}_1, \bar{p}_{\bar{N}})}{\bar{N}-1}. \quad (17)$$

基于均匀排挤机制, 改进的截断排挤偏序关系定义为

$$\begin{aligned} i \succ_d j &\Leftrightarrow \\ \begin{cases} d(\bar{p}_i, \bar{p}_j) < (j-i)\sigma_{\text{crowd}}; \\ 0 < i < |\bar{P}(t+1)|-1, i+1 < j < |\bar{P}(t+1)|. \end{cases} \end{aligned} \quad (18)$$

如果 $i \succ_d j$, 则从档案集中将个体 j 排挤出去.

2) 混沌替换操作

SPEA2 的截断排挤操作只在 $|\bar{P}_{t+1}| > \bar{N}$ 时进行. 但如果 $|\bar{P}_{t+1}| < \bar{N}$ 时, \bar{P}_{t+1} 中的个体也可能分布不均匀, 个别个体周围的个体密度过大, 导致其个体密度较大, 而这些个体在经过进化操作后仍然可能分布在原父代个体周围. 这样不仅不能提高种群的多样性, 而且还会导致子代个体密度升高, 这显然不利于 Pareto 最优解的均匀分布. 因此, 应对 $|\bar{P}_{t+1}| < \bar{N}$ 时的档案集也进行改进的截断排挤操作, 将个体密度大的个体排挤出去.

在截断排挤操作完成之后, 新的档案集合的规模小于 \bar{N} , 因此需要根据合适策略选择合适的个体补充到档案集中使其规模达到 \bar{N} . 实验研究表明, 将使用随机方法产生的个体补充到待进化群体能够较好地保证种群的多样性^[10]. 利用混沌序列随机产生的

个体同时具有较好的均匀性和多样性, 因此, 完全可以进行混沌替换操作, 用基于混沌 Tent 映射产生的混沌个体替换基本种群 P_t 和档案集合 \bar{P}_t 中适应度值较小的支配个体, 并将其加入档案集合 \bar{P}_{t+1} 中. 混沌替换个体的产生方式与种群混沌初始化时个体的产生方式相同, 在此不再赘述.

4.2.3 引入变缩放因子和支配关系计算的差分变异操作

在单目标差分进化算法中, DE/current-to-best/1/bin 差分变异策略可表示为

$$V_i^{t+1} = X_i^t + F_1(X_{\text{best}}^t - X_i^t) + F_2(X_{r_1}^t - X_{r_2}^t). \quad (19)$$

其中: V_i^{t+1} 为差分变异个体, X_i^t 为父代个体, X_{best}^t 为种群当前的最优个体, $X_{r_1}^t$ 和 $X_{r_2}^t$ 为两互不相同的任意父代个体, F_1 和 F_2 为缩放因子 (通常 $F_1 = F_2$).

该策略利用 $(X_{\text{best}}^t - X_i^t)$ 项提高了算法的收敛速度, 利用 $(X_{r_1}^t - X_{r_2}^t)$ 项提高了种群的多样性, 较好地兼顾了算法的收敛速度和种群的多样性. 但是在算法进化初期, 希望算法收敛得更快一些, 因此, 这个阶段可以使得 $(X_{\text{best}}^t - X_i^t)$ 项的缩放因子 F_1 大一点以提高该权的权重, 从而更好地提高收敛速度. 到了算法进化后期算法易陷入局部最优, 这时应使得 $(X_{r_1}^t - X_{r_2}^t)$ 项的缩放因子 F_2 大一点以提高该权的权重, 从而更好地保证种群的多样性使算法跳出局部最优. 为此, 可根据进化代数, 线性实时地调整缩放因子 F_1 和 F_2 , 即

$$\begin{cases} F_1 = F_{\text{max}} - (F_{\text{max}} - F_{\text{min}})t/T, \\ F_2 = F_{\text{min}} + (F_{\text{max}} - F_{\text{min}})t/T, \end{cases} \quad (20)$$

其中 F_{max} 和 F_{min} 分别为最大、最小缩放因子.

在使用 DE/current-to-best/1/bin 差分变异策略时应找到当前种群中的最优个体 X_{best}^t , 在 MOPs 中多个目标函数对应着多个 X_{best}^t , 使用哪一个 X_{best}^t 是需要考虑的. 为此, 提出了如下一种引入支配关系计算的差分变异策略.

以双目标极小化问题为例, 首先分别找出两个目标函数值 f_1 和 f_2 最小时对应的最优个体 $X_{1\text{best}}^t$ 和 $X_{2\text{best}}^t$, 分别把这两个最优个体作为差分变异策略中的最优个体 X_{best}^t ; 然后基于上述变缩放因子的 DE/current-to-best/1/bin 差分变异策略进行两次差分变异操作, 分别得到两个变异个体 V_{1i}^{t+1} 和 V_{2i}^{t+1} . 对 V_{1i}^{t+1} 和 V_{2i}^{t+1} 进行支配关系计算, 将非支配的个体作为最终的差分变异个体 V_i^{t+1} , 即

$$V_i^{t+1} = \begin{cases} V_{1i}^{t+1}, & V_{1i}^{t+1} \succ V_{2i}^{t+1}; \\ V_{2i}^{t+1}, & \text{otherwise.} \end{cases} \quad (21)$$

其中 $V_i^{t+1} = (v_{1i}^{t+1}, v_{2i}^{t+1}, \dots, v_{ji}^{t+1}), j = 1, 2, \dots, m$.

4.2.4 交叉操作

将种群中第 i 个个体 X_i^t 与差分变异个体 V_i^{t+1} 进行交叉操作, 得到试验个体 $U_i^{t+1} = (u_{1i}^{t+1}, u_{2i}^{t+1}, \dots, u_{ji}^{t+1}, \dots, u_{mi}^{t+1})$. 其中

$$u_{ji}^{t+1} = \begin{cases} u_{ji}^t, & \text{rand}(0, 1) \leq \text{CR}; \\ x_{ji}^t, & \text{otherwise.} \end{cases} \quad (22)$$

其中: $\text{rand}(0, 1)$ 为 $[0, 1]$ 之间的均匀随机数, $\text{CR} \in (0, 1)$ 为交叉概率因子.

4.2.5 引入支配关系计算和环境选择操作的选择操作

对于 MOPs, 差分选择操作和 SOPs 有所不同, 需要对试验个体和父代个体进行支配关系计算来完成选择操作以得到子代个体.

1) 若一方 Pareto 支配另一方, 则非支配方个体作为选择操作后的个体被选出作为最终的子代个体, 即有

$$X_i^{t+1} = \begin{cases} U_i^{t+1}, U_i^{t+1} \succ X_i^t; \\ X_i^t, X_i^t \succ U_i^{t+1}. \end{cases} \quad (23)$$

2) 若双方个体互不支配, 则首先将双方个体合并生成中间档案集合 $mX_i^{t+1} = X_i^t \cup U_i^{t+1}$, 此时档案集合规模介于 \bar{N} 和 $2\bar{N}$ 之间; 然后进行环境选择操作, 把个体密度大的多余个体删除, 直到档案集合规模恢复为 \bar{N} .

4.3 SPCDE 的算法步骤

Step 1: 初始化.

Step 1.1: 算法参数初始化. 确定种群规模 N , 档案集合规模 \bar{N} , 最大进化代数 T , 最大和最小缩放因子 F_{\max} 和 F_{\min} , 交叉概率因子 CR , 并令进化代数 $t = 0$.

Step 1.2: 种群初始化. 基于 Tent 映射混沌初始化种群 P_0 , 并令档案集合 $\bar{P}_0 = \emptyset$.

Step 2: 适应度赋值. 进行适应度分配操作, 得到 P_t 和 \bar{P}_t 中个体的适应度值.

Step 3: 环境选择. 利用上述改进的环境选择操作进行环境选择.

Step 4: 终止条件判断. 如果 $t < T$, 则进行 Step 5; 否则, 终止算法, 输出结果.

Step 5: 档案集合中个体的差分进化操作.

Step 5.1: 基于引入支配关系计算和变缩放因子的 DE/current-to-best/1/bin 差分变异策略进行差分变异操作;

Step 5.2: 进行交叉操作;

Step 5.3: 进行引入支配关系计算和环境选择操作的选择操作, 得到子代个体并将其放入 P_{t+1} 中; 令

$t = t + 1$, 返回 Step 2.

5 数值实验和分析

5.1 算法性能评价指标

多目标进化算法的 Pareto 最优解集主要有两方面的要求: 1) 求得尽可能多的 Pareto 最优解并尽可能地逼近真实的 Pareto 前沿, 这是算法收敛性能的体现; 2) Pareto 最优解应尽可能地沿着 Pareto 前沿均匀分布, 这是解的分布均匀性的体现. 为此引入以下两个性能评价指标:

1) GD 指标. GD 为世代距离, 主要评价算法的收敛性能, 其定义为

$$\text{GD} = \left(\frac{1}{n_{PF}} \sum_{i=1}^{n_{PF}} d_i^2 \right)^{\frac{1}{2}}. \quad (24)$$

其中: n_{PF} 为 Pareto 最优解集中解的数目, d_i 为每个 Pareto 最优解到全局最优解的最小欧氏距离.

GD 越小, 说明求得的 Pareto 最优解集越逼近全局最优解集, 算法的收敛性也越好.

2) SP 指标. SP 指标描述非劣解在目标空间上的分布范围, 其定义为

$$\text{SP} = \sqrt{\frac{1}{n_{PF} - 1} \sum_{i=1}^{n_{PF}} (\bar{d} - d_i)^2}. \quad (25)$$

其中: n_{PF} 为 Pareto 最优解集中解的数目; $\bar{d} = \frac{1}{n_{PF}} \sum_{i=1}^{n_{PF}} d_i$; $d_i = \min_{j=1, 2, \dots, n_{PF}} \left(\sum_{k=1}^r |f_k^i - f_k^j| \right)$, f_k^i 为第 i 个 Pareto 最优解的第 k 维目标值.

SP 越小, 说明 Pareto 最优解集中解的分布性越好.

5.2 实验结果与分析

采用文献 [11] 中给出的测试函数 SCH, ZDT1, ZDT2 和 ZDT6 以及相应的 Pareto 前沿测试本文算法的性能. 第 1 个测试函数的维数 $m = 1$, 其余 3 个的维数 $m = 30$. 算法参数设置如下: $T = 200$, $N = 100$, $\bar{N} = 50$, $F_{\max} = 0.9$, $F_{\min} = 0.3$, $\text{CR} = 0.5$.

表 1~表 4 分别给出了 SPCDE 算法对 4 个测试函数分别运行 20 次, GD 指标和 SP 指标的均值 (Mean) 和方差 (Var) 的测试结果; 同时给出了文献 [12] 提供的 NSGA-II 和 SPEA2 算法的相应实验结果. 从表 1~表 4 中可以看出, 对于 GD 指标, 除 ZDT1 函数 SPCDE 算法得到的结果的方差略差于 SPEA2 算法外, 对其余测试函数 SPCDE 算法得到结果的平均值和方差都小于 NSGA-II 和 SPEA2 算法, 这表明 SPCDE 算法具有更好的收敛性能. 对于 SP 指标, SPCDE 算法对 NSGA-II 和 SPEA2 都具有绝对优势, 其结果一般能提高 1~2 个数量级, 表明 SPCDE 算法中的改进截断排挤操作能够很好地保证解的分布均匀性.

表 1 SCH 测试函数的测试结果

| 算法 | GD | | SP | |
|---------|------------|------------|------------|------------|
| | Mean | Var | Mean | Var |
| SPCDE | 1.278 7e-4 | 6.684 0e-6 | 4.325 3e-4 | 3.808 8e-4 |
| SPEA2 | 3.901 1e-4 | 1.934 0e-5 | 1.397 0e-2 | 1.088 5e-3 |
| NSGA-II | 4.257 0e-4 | 1.654 0e-5 | 3.313 2e-2 | 6.841 6e-3 |

表 2 ZDT1 测试函数的测试结果

| 算法 | GD | | SP | |
|---------|------------|------------|------------|------------|
| | Mean | Var | Mean | Var |
| SPCDE | 1.197 0e-4 | 2.843 7e-5 | 0 | 0 |
| SPEA2 | 2.639 8e-4 | 2.597 0e-5 | 3.028 2e-3 | 1.807 0e-4 |
| NSGA-II | 1.885 6e-4 | 5.222 0e-5 | 7.804 4e-3 | 7.836 0e-4 |

表 3 ZDT2 测试函数的测试结果

| 算法 | GD | | SP | |
|---------|------------|------------|------------|------------|
| | Mean | Var | Mean | Var |
| SPCDE | 9.669 0e-5 | 2.210 8e-6 | 4.319 3e-3 | 1.447 3e-3 |
| SPEA2 | 1.009 2e-4 | 3.715 0e-6 | 3.155 8e-3 | 2.961 0e-4 |
| NSGA-II | 1.071 4e-4 | 3.891 0e-5 | 7.705 6e-3 | 1.503 9e-3 |

表 4 ZDT6 测试函数的测试结果

| 算法 | GD | | SP | |
|---------|------------|------------|------------|------------|
| | Mean | Var | Mean | Var |
| SPCDE | 4.053 2e-4 | 1.675 0e-6 | 4.041 8e-4 | 8.589 6e-5 |
| SPEA2 | 5.630 5e-4 | 7.473 0e-6 | 1.827 4e-3 | 1.183 0e-4 |
| NSGA-II | 5.306 9e-2 | 3.495 1e-2 | 6.343 4e-3 | 2.102 1e-3 |

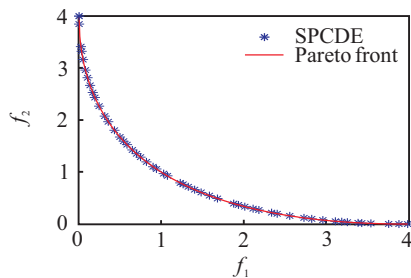


图 1 SCH 的仿真曲线

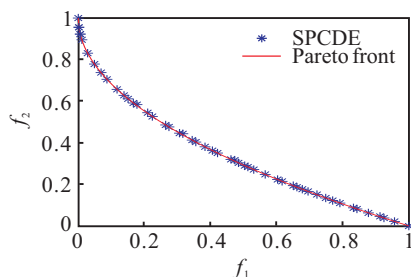


图 2 ZDT1 的仿真曲线

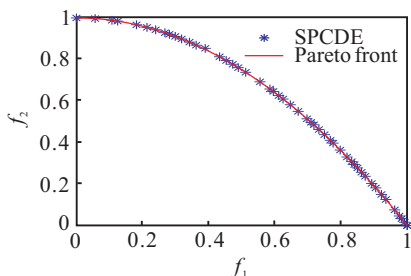


图 3 ZDT2 的仿真曲线

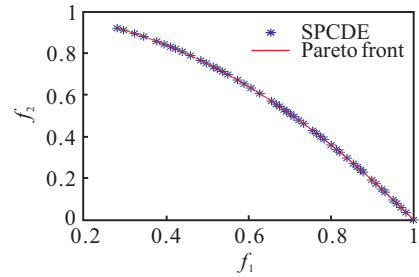


图 4 ZDT6 的仿真曲线

图 1~图 4 分别给出了从 20 次运行中随机选出的 1 组 SPCDE 算法求得的 4 个测试函数的 Pareto 最优解集及其相应的 Pareto 前沿的仿真曲线. 从图 1~图 4 中可以直观地看出, 利用 SPCDE 算法得到的 Pareto 最优解能够很好地逼近真实的 Pareto 前沿, 并且分布也较为均匀.

6 结 论

本文借鉴 SPEA2 算法和差分进化算法的思想并结合混沌 Tent 映射, 提出了多目标强度 Pareto 混沌差分进化算法. 通过基于 Tent 映射的种群混沌初始化, 提高了初始种群个体的质量和分布均匀性. 在环境选择操作上提出了一种引入均匀排挤机制的截断排挤操作和基于 Tent 映射的混沌替换操作. 在差分进化操作上, 提出了一种变缩放因子的 DE/current-to-best/1/bin 差分变异策略并由支配关系计算得到差分变异个体; 并提出了一种引入支配关系计算和环境选择操作的选择操作. 通过以上措施, 既保证了了解的多样性和分布均匀性, 又提高了算法的收敛性. 由 4 个经典测试函数的测试结果可知, 本文算法无论在 Pareto 最优解集的逼近性还是均匀性方面, 都优于有关文献中的多目标进化算法, 表明了本文算法在求解多目标优化问题中的有效性和高效性.

参考文献(References)

- [1] 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.
- [2] Zitzler E, Laumanns M, Thiele L. SPEA2: Improving the strength Pareto evolutionary algorithm[R]. Lausanne: Swiss Federal Institute of Technology Computer Engineering and Networks Laboratory, 2001.
- [3] Rainer S, Price K. Differential evolution: A simple and efficient heuristic for global optimization over continuous spaces[J]. J of Global Optimization, 1997, 11(4):341-359.
- [4] 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. Advance in differential evolution[J]. Control and Decision, 2007, 22(7): 721-729.)

(下转第 52 页)