

求解有约束优化问题的实数遗传算法改进研究

王吉权[†], 程志文, 张攀利, 代伟婷

(东北农业大学 工程学院, 哈尔滨 150030)

摘要: 提出一种求解有约束优化问题的改进实数遗传算法. 首先, 提出一种排序分组选择法, 该方法具有种群多样性好、易于实现的优点. 其次, 给出一种基于方向的启发式交叉算子(DBHX), DBHX能够产生无数个交叉方向, 且有可能产生一个引导参与交叉的染色体向最优解移动的方向 \vec{D} , 即使交叉方向与 \vec{D} 不一致, 也有很大的可能性非常接近方向 \vec{D} , 可以保证有很大的机会产生更好的子代染色体. 最后, 针对单一的变异算子无法兼顾局部搜索能力和全局搜索能力的缺点, 提出一种组合变异方法, 使得变异操作既能保证算法的局部搜索能力, 又能兼顾全局搜索能力. 10个实例的计算结果表明, 所提出的改进实数遗传算法具有较快的收敛速度, 从而验证了所提出算法的可行性和有效性.

关键词: 约束优化问题; 实数遗传算法; 罚函数法; 启发式交叉算子; 组合变异

中图分类号: TP18

文献标志码: A

Research on improvement of real-coded genetic algorithm for solving constrained optimization problems

WANG Ji-quan[†], CHENG Zhi-wen, ZHANG Pan-li, DAI Wei-ting

(College of Engineering, Northeast Agricultural University, Harbin 150030, China)

Abstract: An improved real-coded genetic algorithm (IRCGA) is proposed to solve constrained optimization problems. Firstly, a sorting group selection method is proposed, which has good population diversity and is easy to realize. Then, a direction-based heuristic crossover operator (DBHX) is proposed, which can generate numerous crossover directions. and it also enables a great possibility to generate a crossover direction \vec{D} to guide the chromosomes of participation crossover to move towards the optimal solution direction. Even if the crossover direction is inconsistent with \vec{D} , it is also very closed to the direction \vec{D} to the utmost so as to ensure that there is a great chance to produce better offspring chromosomes. Finally, aiming at the shortcoming that a single mutation operator cannot both take into account the local search ability and the global search ability, a combined mutation method is proposed, which makes the mutation operation not only take into account the local search ability, but also the global search ability. The computing results of ten examples show that the proposed IRCGA has a fast convergence speed, and also verify its effectiveness and feasibility.

Keywords: constrained optimization problems; real-coded genetic algorithm; penalty function method; heuristic crossover operator; combinational mutation

0 引言

遗传算法(GA)^[1-4]是由Holland首次提出的. 随后, De首次提出了精英保留的进化策略, 后来人们对其作了进一步研究, 提出了多种精英保留和用选择代替复制的进化策略^[5-8], 目前的遗传算法基本都是按照这种进化策略进行计算的. 实数遗传算法具有精度高、便于大空间搜索、运算简单、收敛速度快、不易陷入局部极值等特点^[6-7], 已广泛地应用于自动控制、

组合优化、机器学习、图像处理、自适应控制、规划设计、工业工程、智能制造系统、生物工程、系统工程、人工智能、智能机器系统、人工生命等领域^[6-13], 尤其适用于处理传统搜索方法难以解决的复杂和非线性问题, 是21世纪有关智能计算的关键技术之一^[6].

遗传算法中应用较多的编码方案有二进制编码和实数编码. 对于一些多维、高精度连续函数优化问题, 二进制编码遗传算法的随机特性使得其局部搜

收稿日期: 2017-10-26; 修回日期: 2018-03-09.

基金项目: 黑龙江省社科基金项目(16JYB06).

责任编委: 刘德荣.

作者简介: 王吉权(1970—), 男, 副教授, 博士, 从事遗传算法理论及应用、神经网络理论及应用等研究; 程志文(1993—), 男, 硕士生, 从事遗传算法理论与应用的研究.

[†]通讯作者. E-mail: wang-jiquan@163.com.

索能力较差,而相邻整数的二进制编码可能具有较大的Hamming距离,会降低遗传算子的搜索效率并影响计算精度.另外,使用二进制编码需频繁地编码和解码,既增加了算法的计算量,又存在转换误差;产生有限的离散点阵,导致接近极值的个体漏掉,造成算法早熟收敛或寻优速度慢,算法效率随着变量增多和计算精度提高急剧下降.同时,由于解的精度受编码长度控制,采用二进制编码会带来编码过长的问题,从而导致计算量和存储量太大,运算速度降低.Michalewicz等^[14]提出了一种实数编码的遗传算法,基于实数编码的遗传算法不存在编码和解码过程,能够大大提高解的精度和运算速度.近年来,实数遗传算法以其独特而卓越的性能逐渐引起了人们的关注,很多学者对实数遗传算法进行了深入研究.根据使用的机理和技术,实数遗传算法所取得的进展和尝试可以被分成以下几类:1)最优种群规模的确定^[15-18];2)初始种群的产生^[19-21];3)现有交叉算子的改进^[22-25];4)现有变异算子的改进^[26-29];5)进化策略的改进^[30-32];6)算子参数的自动调整^[33-36].这些改进算法的提出,不仅提高了遗传算法的运算速度,也推动了遗传算法理论和应用的发展.

交叉算子通过模拟自然界生物的杂交过程对个体进行交叉操作,不断产生新的个体,从而使得遗传算法具有较强的搜索能力,可以说交叉算子在扩展求解空间、获得全局最优解的过程中发挥着至关重要的作用.变异算子是增加种群多样性的重要算子,能够使GA扩大搜索空间,避免陷入局部最优.因此,许多学者将研究的重点放在了交叉算子和变异算子的改进方面.其中,对交叉算子的改进研究最多,其次是对变异算子的改进研究.Chuang等^[22]提出了一种基于方向的交叉(DBX)算子和一种动态随机变异的变异算子.DBX算子产生的交叉方向数量有限,虽然有可能产生一个引导参与交叉的染色体向最优解移动的交叉方向,但产生该交叉方向的可能性较小,并且当种群中个体的分量即维数较少时,容易产生零向量,即不能产生交叉方向.在动态随机变异算子中,变异算子的步长计算公式与文中对步长公式的解释相矛盾.另外,由于在排序选择中用种群中一定数量的较好的染色体替换较差的染色体,当经过多次迭代后,种群的多样性变差,最终导致种群中绝大多数个体都相同或所有的个体都相同,致使GA收敛到全局最优解的时间大大增加.汪定伟等^[37]提出了一种向目标函数的梯度方向变异的变异算子,如果目标函

数比较复杂,即使目标函数可导,也很难求得目标函数的梯度,若目标函数不可导,则无法使用该变异算子,但对于目标函数比较简单且可导时,该变异算子可使GA快速收敛到最优解点.Peltokangas等^[38-43]对均匀变异(Uniform mutation)算子、非均匀变异(Non-uniform mutation)算子、幂变异(Power Mutation)算子和边界变异(Boundary mutation)算子进行了总结.其中:非均匀变异算子和幂变异算子计算量较大,边界变异只适用于最优解位于可行域边界上的优化问题,均匀变异算子产生的子代与要变异的个体没有关系.

综上所述,一个好的交叉和变异算子应该具备3个条件:一是具有启发性,即引导参与交叉的染色体向好的方向移动,从而产生有潜力的子代;二是计算量越小越好;三是编程时能够避免使用循环语句,即程序运行时完成相应操作所需的时间少.针对这3个方面,本文提出一种求解有约束优化问题的改进实数遗传算法.10个测试函数的计算结果表明,改进实数遗传算法在求解约束优化问题时是可行、有效的.

1 约束优化问题的处理方法——罚函数法

约束优化问题数学模型的一般形式可用如下公式表示:

$$\begin{aligned} \min f(X), X &= [X_1, X_2, \dots, X_k, \dots, X_n] \in R. \\ \text{s.t. } h_i(X) &= 0, i = 1, 2, \dots, p; \\ g_j(X) &\geq 0, j = 1, 2, \dots, q. \end{aligned} \quad (1)$$

其中: n 表示种群规模, $h_i(X) = 0$ 表示第 i 个等式约束, p 表示等式约束的个数, $g_j(X) \geq 0$ 表示第 j 个不等式约束, q 表示不等式约束的个数, X 表示一个 n 维向量, $X_k = (x_{k1}, x_{k2}, \dots, x_{km})$.

式(1)也可以表示为

$$\begin{aligned} \min f(X), X &= [X_1, X_2, \dots, X_k, \dots, X_n] \in R. \\ R &= \{X | h_i(X) = 0, i = 1, 2, \dots, p; \\ &g_j(X) \geq 0, j = 1, 2, \dots, q\}. \end{aligned} \quad (2)$$

令 X^* 为有约束优化问题的最优解,于是有 $\forall X \in R: f(X^*) \leq f(X)$.此外,如果 $g_j(X^*) = 0$,则不等式约束被称为有效约束.在这个概念下,在 X^* 点,所有的等式约束 $h_i(X) = 0 (i = 1, 2, \dots, p)$ 都是有效的.

罚函数法是一种将有约束优化问题转化为无约束优化问题的方法.该方法通过构造两个制约函数,并引入惩罚因子,然后将它们加到约束优化问题的目标函数上,构成惩罚函数,其罚函数^[44]为

$$P(X, M) = f(X) + M_1 \sum_{i=1}^p [h_i(X)]^2 + M_2 \sum_{j=1}^q [\min(0, g_j(X))]^2. \quad (3)$$

其中: M_1 和 M_2 是惩罚因子, 一般设为足够大的正常数; 第2项和第3项是惩罚项; $P(X, M)$ 是惩罚函数.

由式(3)容易看出: 当 $X \in R$ 时, 对于可行点没有也不应有任何惩罚作用, 有 $P(X, M) = f(X)$; 当 $X \notin R$ 时, 由于 M_1 和 M_2 很大, 将使式(3)中的第2项和第3项的值也很大, 这相当于对非可行点的“惩罚”, 且 X 点离可行域越远, 惩罚越严厉. 可以想见, 当 M_1 和 M_2 变得足够大时, 相应于这样的 M_1 和 M_2 值, 式(3)的无约束优化问题的极小点 $X(M)$ 就会与原来的约束优化问题的极小点足够接近. 而当 $X(M) \in R$ 时, 它将成为原约束问题的极小点.

求式(3)的最小值, 即求

$$\min P(X, M), \quad (4)$$

等价于求式(1)的最小值.

2 实数遗传算法的改进

2.1 排序分组选择

设种群大小 n 为偶数, 将种群中的个体按照目标函数值 $P(X, M)$ 从大到小的顺序排序 (目标函数为极大值). 如果目标函数为极小值, 则根据 $\max P(X, M) = -\min[-P(X, M)]$, 将目标函数 $P(X, M)$ 变为求极大值. 排序之前的种群为 $X = (X_1, X_2, \dots, X_n)$, 排序之后的种群为 $X' = (X'_1, X'_2, \dots, X'_n)$, 且满足 $P(X'_1, M) \geq P(X'_2, M) \geq \dots \geq P(X'_n, M)$. 选择 X'_1 和 $X'_{n/2+1}$ 配对参与交叉, X'_2 和 $X'_{n/2+2}$ 配对参与交叉, 依此类推, 按照这样的方法可得到 $n/2$ 对参与交叉的个体.

与轮盘赌和锦标赛选择相比, 排序分组选择方法能够扩大配对父代个体之间的距离, 增大配对父代个体之间的差异性, 有助于保持种群的多样性. 另外, 排序分组选择只是根据目标函数值的大小来选择参与交叉的个体, 选择参与交叉的父代个体容易实现、选择速度快.

2.2 基于方向的启发式交叉算子 (Direction-based heuristic crossover, DBHX)

因为目标函数值越大的个体越有可能接近最优解, 所以认为最优个体有可能在目标函数值大的个体附近^[45]. 受此启发, 本文提出一种基于方向的启发式交叉算子, 该交叉算子直接使用目标函数值和交叉方向信息获得有潜力的子代, 从而提高收敛的速度.

设参与交叉的两个父代个体分别为 X'_i ($i =$

$1, 2, \dots, n/2$) 和 X'_j ($j = n/2 + 1, n/2 + 2, \dots, n$), 且 X'_j 优于 X'_i , 可按下式产生两个子代个体 \bar{X}_i 和 \bar{X}_j , 即

$$\begin{cases} \bar{X}_i = X'_i + \lambda R_1 * \vec{D}, & i = 1, 2, \dots, n/2; \\ \bar{X}_j = X'_j + \lambda R_2 * \vec{D}, & j = n/2 + 1, n/2 + 2, \dots, n; \\ R_1 = (r_{11}, r_{12}, \dots, r_{1m}); \\ R_2 = (r_{21}, r_{22}, \dots, r_{2m}); \\ \vec{D} = X'_j - X'_i, & i = 1, 2, \dots, n/2, \\ & j = n/2 + 1, n/2 + 2, \dots, n. \end{cases} \quad (5)$$

其中: $R_1 * \vec{D}$ 和 $R_2 * \vec{D}$ 是交叉方向, “*” 与 Matlab 中的点乘 (即 “.”) 相同, $R_1 * \vec{D}$ 是 R_1 和 \vec{D} 中相同位置的元素对应相乘, $R_2 * \vec{D}$ 是 R_2 和 \vec{D} 中相同位置的元素对应相乘; $r_{11}, r_{12}, \dots, r_{1m}$ 和 $r_{21}, r_{22}, \dots, r_{2m}$ 是 $[0, 1]$ 之间均匀分布的随机数; λ 是步长, 通常取 $\lambda = 1$.

为了分析基于方向的启发式交叉操作产生的子代所在的空间位置, 以变量维数 $m = 2$ 为例, 设参与交叉的两个父代个体分别为 X'_1 和 X'_2 , 且 X'_2 优于 X'_1 , 分析基于方向的启发式交叉算子产生的可能交叉方向. 由于 $m = 2$, 式(5)中 $R_1 = (r_{11}, r_{12})$, $R_2 = (r_{21}, r_{22})$, $r_{11}, r_{12}, r_{21}, r_{22}$ 是 $[0, 1]$ 之间均匀分布的随机数. 对于 X'_1 点, 当 $r_{11} = 0$ 时, 交叉方向 $R_1 * (X'_2 - X'_1)$ 为 \vec{D}_2 , 如图1所示. 同理, 当 $r_{12} = 0$ 时, 交叉方向 $R_1 * (X'_2 - X'_1)$ 为 \vec{D}_1 ; 当 $r_{11} = r_{12}$ 时, 交叉方向 $R_1 * (X'_2 - X'_1)$ 为 \vec{D} . 对于 X'_2 点, 当 $r_{21} = 0$ 时, 交叉方向 $R_2 * (X'_2 - X'_1)$ 为 \vec{D}_4 ; 当 $r_{22} = 0$ 时, 交叉方向 $R_2 * (X'_2 - X'_1)$ 为 \vec{D}_3 ; 当 $r_{21} = r_{22}$ 时, 交叉方向 $R_2 * (X'_2 - X'_1)$ 为 \vec{D} . 当 r_{11} 和 r_{12} 为 $[0, 1]$ 之间均匀分布的随机数时, 交叉方向 $R_1 * (X'_2 - X'_1)$ 可为 \vec{D}_1 和 \vec{D}_2 之间的任意一个方向. 当 r_{21} 和 r_{22} 为 $[0, 1]$ 之间均匀分布的随机数时, 交叉方向 $R_2 * (X'_2 - X'_1)$ 可为 \vec{D}_3 和 \vec{D}_4 之间的任意一个方向. 因此, 基于方向的启发式交叉算子产生的可能交叉方向有无数个. 另外, 由于式(5)中的步长 $\lambda = 1$, $\lambda R_1 * (X'_2 - X'_1)$ 为图1中

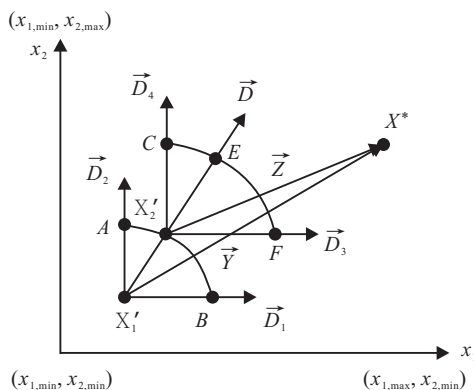


图1 X'_1 和 X'_2 的 DBHX 操作示意图

$AX'_2BX'_1$ 中的任意一点, $\lambda R_2 * (X'_2 - X'_1)$ 为 $CEFX'_2$ 中的任意一点. 图1中: $CX'_2 = FX'_2 = X'_2E = |X'_2 - X'_1|$, $AX'_1 = BX'_1 = |X'_2 - X'_1|$.

设 X^* 为待求问题的最优解, 对于父代个体 X'_1 而言, 最优的交叉方向为 \vec{Y} , 对于父代个体 X'_2 而言, 最优的交叉方向为 \vec{Z} . 基于方向的启发式算子能够产生一个引导 X'_1 向最优解移动的方向 \vec{Y} , 即使不能与方向 \vec{Y} 完全一致, 也有很大的可能性产生一个与方向 \vec{Y} 非常靠近的方向; 另外, 基于方向的启发式交叉算子也能够产生一个引导 X'_2 向最优解移动的方向 \vec{Z} , 即使不能与方向 \vec{Z} 完全一致, 也有很大的可能性产生一个与方向 \vec{Z} 非常靠近的方向. 综上所述, 基于方向的启发式算子可以保证有很大的机会产生更好的子代染色体, 能够显著提高算法的收敛速度.

与文献[22]中的DBX交叉操作相比, 基于方向的启发式交叉算子能够产生无数个交叉方向, 而DBX只能产生 $2^m - 1$ 个交叉方向, 且基于方向的启发式交叉算子产生的引导父代个体 X'_1 和 X'_2 向最优解方向移动的可能性大大高于DBX. 因此, 基于方向的启发式交叉算子可以保证有很大的机会产生更好的子代染色体.

2.3 替代操作

对于具有多极值的优化问题, 当算法找寻到一个存在极值的区域时(不管是局部极值还是全局极值), 种群中的个体会不断向这一区域靠拢, 种群中有可能出现相同或相似的个体. 随着迭代次数的增加, 种群中相同的个体数量逐渐增加, 甚至有可能使得种群中所有的个体都相同, 使种群的多样性变差, 从而影响算法遗传操作的效率和探索其他极值区域的能力. 文献[22]中给出的改进遗传算法对于多极值的待优化问题, 有可能出现种群中的绝大多数个体相同或者所有个体都相同的现象, 使得算法无法收敛到全局最优解. 为了避免上述现象的出现, 并在迭代过程中保持种群的多样性, 本文给出的遗传算法中增加了一个替代操作. 替代操作的方法是: 交叉后的种群中若有两个或两个以上相同的染色体存在, 只保留多个相同染色体中的一个, 去掉其余相同的染色体, 令此时种群规模为 n_1 , 为了保持种群规模 n 不变, 随机产生 $n - n_1$ 个个体, 用随机产生的 $n - n_1$ 个个体替代去掉的 $n - n_1$ 个相同的个体.

2.4 组合变异

现有文献中给出的变异算子, 有的局部搜索能力较强^[47-48], 如高斯变异算子, 有的算子全局搜索能力较强^[48], 如柯西变异算子. 对于极值点较少的待优化

问题, 应采用局部搜索能力较强的变异算子. 对于极值点较多的待优化问题, 如果采用局部搜索能力强的变异算子, 则容易收敛到局部极值点, 而无法收敛到全局极值点; 如果采用全局搜索能力较强的变异算子, 且对于待求解问题最优解的精度要求较高, 则算法的收敛速度会变慢. 有的文献给出的变异算子虽然在迭代初期全局搜索能力强, 但随着迭代次数的增加, 局部搜索能力增加, 这需要给定最大迭代次数. 对于不同的待优化问题, 很难事先给出一个较合适的最大迭代次数, 这种方法虽然理论上可行, 但实际上可操作性较差^[22]. 综上所述, 单一的变异算子在解决优化问题时很难兼顾全局探索和局部搜索能力. 为此, 本文提出一种组合变异的方法. 该方法使用3个变异算子: 第1个变异算子随着迭代次数的增加, 局部搜索能力加强; 第2个变异算子的全局搜索能力较强; 第3个变异算子的局部搜索能力较强. 具体做法是: 当迭代次数除以3的余数为1时, 采用全局搜索能力较强的变异算子; 当迭代次数除以3的余数为2时, 采用局部搜索能力较强的变异算子; 当迭代次数能够被3整除时, 采用第3个变异算子. 组合变异的优点是, 在算法的迭代过程中, 既考虑到了变异算子的局部搜索能力, 又能兼顾变异算子的全局搜索能力. 3个变异算子如下:

1)

$$\bar{X}'_i = \bar{X}_i + \lambda_1 * (b^T - a^T) / \text{gen}. \quad (6)$$

其中: gen 是迭代次数; \bar{X}_i 是交叉产生的子代个体; \bar{X}'_i 是变异得到的子代个体; 向量 a 和 b 是变量的取值下限和上限; $\lambda_1 = (\lambda_{11}, \lambda_{12}, \dots, \lambda_{1m}), \lambda_{11}, \lambda_{12}, \dots, \lambda_{1m}$ 是 $[-1, 1]$ 之间均匀分布的随机数.

2)

$$\bar{X}'_i = \bar{X}_i + \text{Cauchy}(0, 1). \quad (7)$$

其中: $\text{Cauchy}(0, 1)$ 为标准柯西分布, \bar{X}_i 为待变异的个体, \bar{X}'_i 为变异后的个体.

3)

$$\bar{X}'_i = \bar{X}_i + N(\mu, \delta^2). \quad (8)$$

其中: μ 为正态分布的均值, 通常取 $\mu = \bar{X}_i$; δ^2 为正态分布的方差.

式(7)中, δ^2 表示为

$$\delta^2 = \frac{|X_{\text{best}} - \bar{X}_i|}{6}, \quad (9)$$

其中 X_{best} 为种群中的最优个体.

第1个变异算子在迭代初期的全局搜索能力较强, 随着迭代次数的增加, 局部搜索能力增加, 但当迭代次数增加到某一值时, 其几乎失去了变异的功能,

而当迭代次数不是很大时,其效果比较明显.第2个变异算子为柯西变异,柯西变异算子相比于正态变异算子会产生较大的变异步长,因此会使算法具有较好的全局搜索能力.第3个变异算子为正态变异,正态变异算子重点搜索原个体附近的某个局部区域,局部搜索能力较好,但是引导个体跳出局部较优解的能力较弱,不利于全局收敛.因此,组合变异算子既兼顾了全局探索能力,又兼顾了局部搜索能力,可使GA能够快速收敛到全局最优解.

2.5 算法的伪代码

结合上面对本文所提出改进实数遗传算法的描述,以算法是否达到给定的精度要求为迭代终止条件,给出的改进实数遗传算法的伪代码如下所示.

算法1 改进实数遗传算法.

输入: 种群规模 n , 变异概率 P_m , 交叉概率 P_c , 计算精度 ε , 保留精英个数 s , 罚因子 M_1 和 M_2 , 初始迭代次数 $t = 0$;

输出: 最优解 x_b , 最优值 $f(x_b)$.

1) 种群的初始化 $X = (X_1, X_2, \dots, X_n)$, $X_i = (x_{i1}, x_{i2}, \dots, x_{im}), i = 1, 2, \dots, n$.

2) 计算种群中个体的目标函数值, $f(X_i), i = 1, 2, \dots, n$.

3) 将种群中的个体 $X = (X_1, X_2, \dots, X_n)$ 按目标函数值从大到小排序, 排序后为 $X' = (X'_1, X'_2, \dots, X'_n)$.

4) $t = 0$.

5) while

判断是否满足迭代终止条件,如满足,则输出最优解和最优值,否则,执行下一步操作;

根据种群中个体的排序结果,运用排序分组选择法,选择参与交叉的个体,保留 s 个精英个体;

按照 DBHX, 即式(5)进行交叉操作;

从交叉产生的 n 个个体和 n 个父代个体中保留 s 个精英个体,选出 n 个较好的个体进行变异操作;

用 s 个精英个体替换变异后的 n 个个体中最差的 s 个个体, $t \leftarrow t + 1$.

6) end while.

3 算法流程

本文所提实数遗传算法的流程如下:首先,进行参数的初始化,给出种群的规模 n 、变异概率 P_m 、交叉概率 P_c 、保留精英个体的数量 s 、计算精度要求 ε , 并产生初始种群.然后,计算种群中个体的目标函数值,判断是否满足迭代终止条件,若满足迭代终止条件,则输出结果,停止计算;若不满足迭代终止条件,

则将种群中的所有个体按目标函数值从大到小排序(目标函数求极大值),根据目标函数值的排序结果进行选择、交叉,并从交叉产生的 n 个个体和 n 个父代个体中保留 s 个精英个体,之后将交叉产生的 n 个子代个体中的 nP_m 个个体进行变异,并用 s 个精英个体替换变异后的 n 个个体中最差的 s 个个体.重复上述步骤,直至满足迭代终止条件为止.本文提出的改进实数遗传算法的流程如图2所示.

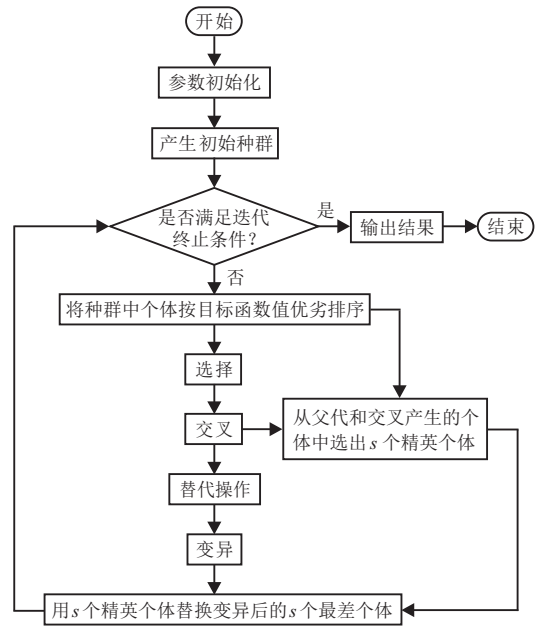


图2 算法流程

4 算法的测试与分析

为了验证本文所提出改进实数遗传算法的有效性和可行性,本文所提出的改进实数遗传算法需要与现有的实数遗传算法进行比较.在现有关于实数遗传算法的研究文献中,由于文献[22]是近几年发表的,且文中对其所提出的实数遗传算法进行了大量的比较研究,所以本文所提出的改进实数遗传算法将与文献[22]中所提出的实数遗传算法进行比较.另外,还将本文所提出的改进实数遗传算法与文献[49]和文献[50]进行比较,如果本文所提出实数遗传算法的性能优于文献[22]、文献[49]和文献[50]中的实数遗传算法,则说明本文所提出的改进实数遗传算法是有效、可行的.

文献[22]中提出的改进实数遗传算法记为IRCGA-1,本文所提出的改进实数遗传算法记为IRCGA-2,文献[49]中的遗传算法记为RCQGA,文献[50]中的实数遗传算法记为IRCGA-3.

4.1 迭代终止条件

定义实数遗传算法的迭代终止条件为

$$|f_i - f_i^*| \leq \varepsilon_i, i = 1, 2, \dots, p. \quad (10)$$

其中: f_i^* 为第 i 个测试函数理论上的全局最优值, f_i 为实数遗传算法求得的第 i 个测试函数的最优值, ε_i 为第 i 个测试函数的精度.

4.2 测试函数

为了验证改进实数遗传算法的有效性和适用性, 选取10个具有相当复杂度的常用测试函数.

第1个测试函数:

$$f_1(x, y) = 0.5 - \frac{\sin^2 \sqrt{x^2 + y^2} - 0.5}{(1 + 0.001(x^2 + y^2))^2},$$

$$-10 < x, y < 10. \quad (11)$$

函数 f_1 有无数个局部极大点, 但只有一个 $(0, 0)$ 为全局最大点, 最大值为1. 此函数的最大峰周围有两个圈脊, 它们的取值分别为0.990 284和0.962 776, 因此优化过程中很容易停滞在这些局部极大点.

第2个测试函数:

$$f_2(x, y) =$$

$$\left(4 - 2.1x^2 + \frac{x^4}{3}\right)x^2 + xy + (-4 + 4y^2)y^2,$$

$$-10 < x, y < 10. \quad (12)$$

函数 f_2 有6个局部极小点, 有两个点 $(-0.0898, 0.7126)$ 和 $(0.0898, -0.7126)$ 为全局最小点, 最小值为 -1.031628 .

第3个测试函数:

$$f_3(x, y) =$$

$$-(x^2 + 2y^2 - 0.4\cos(3\pi x) - 0.6\cos(4\pi y)),$$

$$-10 \leq x, y < 10. \quad (13)$$

函数 f_3 是一个多峰函数, 但只有一个全局最大点 $(0, 0)$, 最大值为1.

第4个测试函数:

$$\min f_4(x, y) = 100(y - x^2)^2 + (x - 1)^2,$$

$$-10 \leq x, y \leq 10. \quad (14)$$

函数 f_4 为 Rosenbrock 函数, 也称为香蕉函数, 是很难极小化的典型病态函数. 该函数在 $-10 \leq x, y \leq 10$ 范围内虽然只有一个全局极小点, 但它在全局极小点临近的狭长区域内取值变化极为缓慢, 所以常用该函数来评价算法的搜索性能. 函数 f_4 的全局最优解为 $(1, 1)$, 最小值为0.

第5个测试函数:

$$\min f_5(x, y) =$$

$$-\sin x \left[\sin \left(\frac{x^2}{\pi} \right) \right]^{20} - \sin y \left[\sin \left(\frac{2y^2}{\pi} \right) \right]^{20},$$

$$0 \leq x, y \leq \pi. \quad (15)$$

函数 f_5 为 Michalewicz 函数, 该函数具有多个局部最小值, 其全局最小值为 -1.80130 . 全局最小值在一个较小的区域, 而其他大部分区域都与最小值不相关.

第6个测试函数:

$$\min f_6(x, y) = -\frac{\sin^3(2\pi x) \sin(2\pi y)}{x^3(x+y)}.$$

$$\text{s.t. } g_1(x, y) = x^2 - y + 1 \leq 0;$$

$$g_2(x, y) = 1 - x + (y - 4)^2 \leq 0;$$

$$0 \leq x, y \leq 10. \quad (16)$$

函数 f_6 为有约束非线性函数, 并有大量局部最优点的复杂多峰函数, 此函数很容易使算法陷入局部最优, 而得不到全局最优解. 函数 f_6 的全局最优解为 $(1.2279723, 4.2453733)$, 最小值为 -0.095825 .

第7个测试函数:

$$\min f_7(x) = (x_1^2 + x_2 - 11)^2 + (x_1 + x_2^2 - 7)^2.$$

$$\text{s.t. } (x_1 - 0.05)^2 + (x_2 - 2.5)^2 - 4.84 \leq 0;$$

$$4.84 - x_1^2 - (x_2 - 2.5^2) \leq 0;$$

$$0 \leq x_1, x_2 \leq 6.$$

函数 f_7 的全局最优解为 $(2.246826, 2.381865)$, 最小值为13.59085.

第8个测试函数:

$$\min f_8(x) = (x_1 - x_2)^2 + (x_2 + x_3 - 2)^2 -$$

$$(x_4 - 1)^2 + (x_5 - 1)^2.$$

$$\text{s.t. } x_1 + 3x_2 = 0;$$

$$x_3 + x_4 - 2x_5 = 0;$$

$$x_2 - x_5 = 0;$$

$$-10 \leq x_i \leq 10, i = 1, 2, \dots, 5.$$

函数 f_8 的全局最优解为 $(-33, 11, 27, -5, 11)/43$, 最小值为176/43.

第9个测试函数:

$$\min f_9(x) = 3x_1 + 10^{-6}x_1^3 + 2x_2 + \frac{2 \times 10^{-6}}{3}x_2^3.$$

$$\text{s.t. } x_3 - x_4 - 0.48 \leq 0;$$

$$x_4 - x_3 - 0.48 \leq 0;$$

$$1000(\sin(-x_3 - 0.25) + \sin(-x_4 -$$

$$0.25)) - x_1 + 894.8 = 0;$$

$$1000(\sin(x_3 - 0.25) + \sin(x_3 - x_4 -$$

$$0.25)) - x_2 + 894.8 = 0;$$

$$1000(\sin(x_4 - 0.25) + \sin(x_4 - x_3 -$$

$$0.25)) + 1294.8 = 0;$$

$$0 \leq x_i \leq 1200, i = 1, 2;$$

$$-0.48 \leq x_j \leq 0.48, j = 3, 4.$$

函数 f_9 的全局最优解为 (679.945 3, 1 026.067, 0.118 876 4, -0.396 233 6), 最小值为 5 126.498 1.

第10个测试函数:

$$\begin{aligned} \min f_{10}(x) &= x_1 + x_2 + x_3. \\ \text{s.t. } &0.0025(x_4 + x_6) - 1 \leq 0; \\ &0.0025(x_5 + x_7 - x_4) - 1 \leq 0; \\ &0.01(x_8 - x_5) - 1 \leq 0; \\ &-x_1x_6 + 833.33252x_4 + \\ &100x_1 - 83333.333 \leq 0; \\ &-x_2x_7 + 1250x_5 + x_2x_4 - 1250x_4 \leq 0; \\ &-x_3x_8 + x_3x_5 - 2500x_5 + 125000 \leq 0; \\ &100 \leq x_1 \leq 10000; \\ &1000 \leq x_2, x_3 \leq 10000; \\ &10 \leq x_i \leq 1000, i = 4, 5, \dots, 8. \end{aligned}$$

函数 f_{10} 的全局最优解为 (579.316 7, 1 359.943, 5 110.071, 182.017 4, 295.598 5, 217.979 9, 286.416 2,

395.597 9), 最小值为 7 049.248 025.

4.3 参数设置

为了获得一个公平的性能比较, 本文提出的改进实数遗传算法执行以下配置: 1) 各测试函数的计算精度为 10^{-4} . 2) 种群规模为 $n = 100$. 3) 罚函数中 $M_1 = 10^7, M_2 = 10^6$. 4) IRCGA-2 中, 变异概率 $p_m = 0.5$, 交叉概率 $p_c = 1$; IRCGA-1 中, 变异概率 $p_m = 0.05$, 交叉概率 $p_c = 0.9$. 5) IRCGA-2 中保留 $p_e = 50$ 个精英个体. 6) IRCGA-2 的进化策略与文献 [46] 中的进化策略相同. 7) RCQGA 的参数设置与文献 [49] 相同. 8) IRCGA-3 的参数设置与文献 [50] 相同.

4.4 文献 [22] 中改进实数遗传算法的局限性

IRCGA-1 对于局部极值点较多的优化问题, 算法有很大的可能性陷入局部极值点. 以 4.2 节中的第 1 个测试函数为例, IRCGA-1 在不同迭代次数时种群中个体的分布情况如图 3 所示, IRCGA-2 在不同迭代次数时种群中的个体分布情况如图 4 所示.

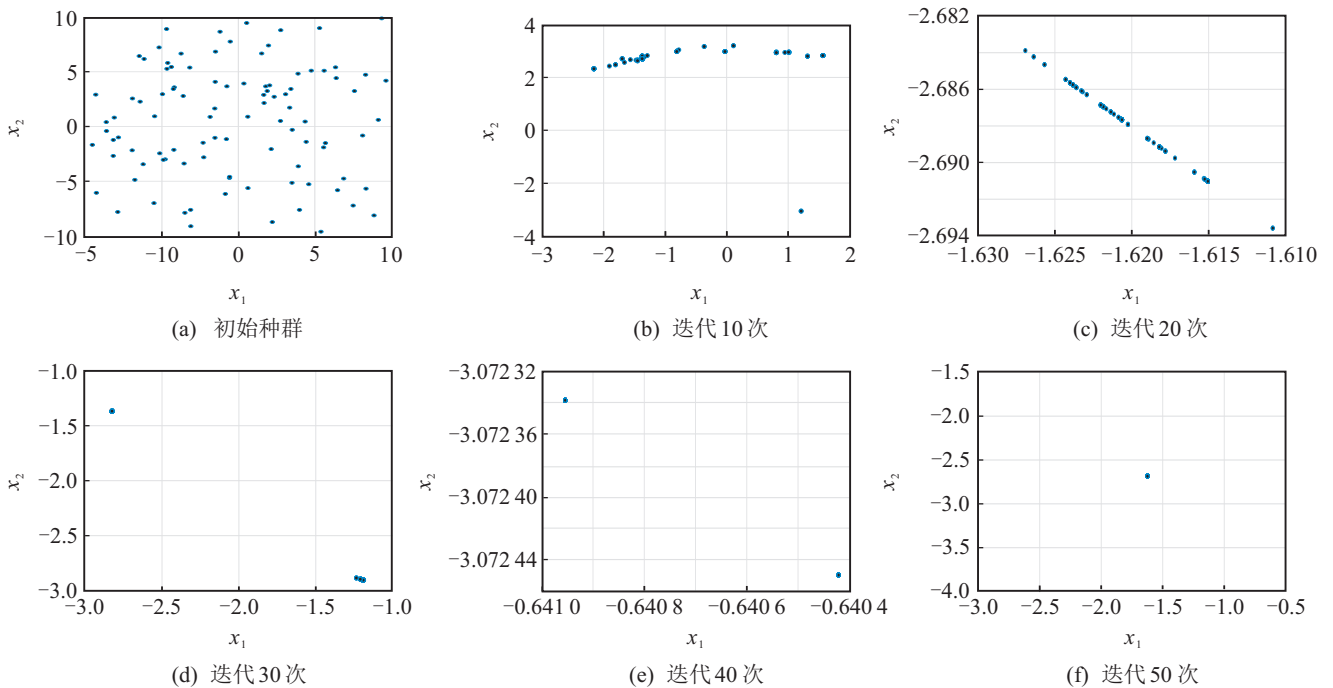


图 3 不同迭代次数下, IRCGA-1 种群中个体的分布情况

从图 3 可以看出, 随着迭代次数的增加, IRCGA-1 种群中个体的多样性变差, 当迭代次数为 50 次时, 种群中所有的个体都相同, 此时目标函数的最大值为 0.990 284, 陷入了局部极值点, 无法求得全局最优解. 从图 4 可以看出, 随着迭代次数的增加, IRCGA-2 种群的多样性没有明显的变化. 由此可知, IRCGA-2 的种群多样性明显优于 IRCGA-1 的种群多样性.

为了统计 IRCGA-1 和 IRCGA-2 陷入局部极值点

的概率, 以 4.2 节中的第 1 个测试函数为例, 统计 1 000 次. IRCGA-1 有 98 次能够求出全局极值点, 陷入局部极值点的概率为 90.2%; IRCGA-2 有 1 000 次求出全局极值点, 陷入局部极值点的概率为 0%. 由此可知, IRCGA-1 求解有较多极值点的待优化问题, 陷入局部极值点的可能性较大, 多数情况下无法求得全局最优解.

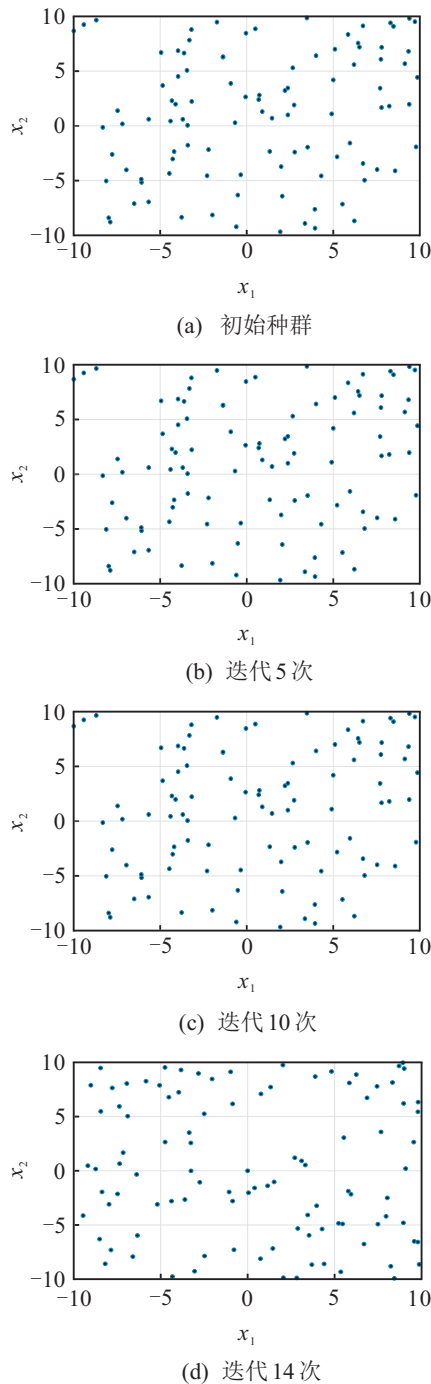


图4 不同迭代次数下, IRCGA-2种群中个体的分布情况

4.5 测试结果

为了使文献[22]中给出的改进实数遗传算法能够求解有较多局部极值点的优化问题,将本文中的替代操作用于文献[22]中给出的改进实数遗传算法中,从而使得文献[22]中给出的改进实数遗传算法能够求解有较多极值点的优化问题。

为了获得一个公平的性能比较基础,使用4.2节中的10个测试函数,每个测试函数在同一台计算机上运行1000次,收集两种改进实数遗传算法对应各个测试函数的计算数据,两种改进实数遗传算法的计算数据如表1所示。

表1 两种改进实数遗传算法的计算结果

函数	实数遗传算法	平均计算时间/s	平均迭代次数/次
f_1	IRCGA-1	1.0539	326.7900
	IRCGA-3	0.9985	105.6200
	RCQGA	0.0953	35.3200
	IRCGA-2	0.0147	16.1240
f_2	IRCGA-1	0.0471	15.6400
	IRCGA-3	0.0486	19.2500
	RCQGA	0.0533	18.1200
	IRCGA-2	0.0127	12.9370
f_3	IRCGA-1	0.0696	23.9900
	IRCGA-3	0.0530	15.6320
	RCQGA	0.0273	9.6300
	IRCGA-2	0.0091	8.9790
f_4	IRCGA-1	0.1372	47.7500
	IRCGA-3	0.1468	41.6020
	RCQGA	0.1736	64.2400
	IRCGA-2	0.0353	42.4600
f_5	IRCGA-1	0.0441	26.4610
	IRCGA-3	0.0460	30.5120
	RCQGA	0.0909	31.7000
	IRCGA-2	0.0217	20.6140
f_6	IRCGA-1	0.0459	27.4500
	IRCGA-3	0.0580	30.5680
	RCQGA	0.0548	29.6500
	IRCGA-2	0.0207	20.2160
f_7	IRCGA-1	0.0842	74.6910
	IRCGA-3	0.0967	68.6950
	RCQGA	0.5905	70.9604
	IRCGA-2	0.0520	49.5400
f_8	IRCGA-1	0.5078	86.4830
	IRCGA-3	0.5968	90.2560
	RCQGA	0.5672	50.1100
	IRCGA-2	0.0567	35.5840
f_9	IRCGA-1	0.3968	85.7000
	IRCGA-3	0.4230	92.6920
	RCQGA	0.3868	82.6800
	IRCGA-2	0.0826	73.4500
f_{10}	IRCGA-1	0.1359	79.8500
	IRCGA-3	0.1462	83.5800
	RCQGA	0.1225	78.8020
	IRCGA-2	0.0743	63.5720

从表1可以看出,无论是平均迭代时间还是平均迭代次数,IRCGA-2都明显优于IRCGA-1、IRCGA-3和RCQGA.因此,与IRCGA-1、IRCGA-3和RCQGA相比,IRCGA-2具有较快的运算速度,优于IRCGA-1、IRCGA-3和RCQGA.

5 结论

本文针对轮盘赌选择、锦标赛选择和局部选择存在的缺点,提出了一种排序分组选择方法,该方法具有只需要计算目标函数值、计算简单和易于实现的优点.同时,对于DBX算子产生较优交叉方向的可能性较小的问题,提出了一种基于方向的启发式交叉算子,该交叉算子有很大的可能性产生较优的交叉方向,从而得到更好的子代个体,提高GA的收敛速度。

现有文献中的遗传算法经过多次迭代后,种群中

有可能出现相同的个体,使得种群的多样性变差,因此,本文在交叉之后增加了一个替代操作,使得种群中没有相同的染色体存在,种群的多样性较好。由于单一变异算子无法既兼顾局部搜索能力,又兼顾全局搜索能力,本文提出了一种组合变异方法,使得变异操作既能够兼顾算法的局部搜索能力,又能够兼顾全局搜索能力。另外,本文提出的改进实数遗传算法克服了文献[22]在求解具有较多局部极值的待优化问题时,多数情况下无法收敛于全局最优解的缺点。10个实例的计算结果表明,本文提出的改进实数遗传算法无论是平均计算时间还是平均迭代次数,都明显少于以往文献中的实数遗传算法,从而验证了本文提出的改进实数遗传算法的有效性、可行性和优越性。

参考文献(References)

- [1] Holland J H. Adaptation in natural and artificial systems[M]. Ann Arbor: University of Michigan, 1975: 102-138.
- [2] 郑生荣, 赖家美, 刘国亮, 等. 一种改进的实数编码混合遗传算法[J]. 计算机应用, 2006, 26(8): 1959-1962. (Zheng S R, Lai J M, Liu G L, et al. An improved real-coded hybrid genetic algorithm[J]. Computer Applications, 2006, 26(8): 1959-1962.)
- [3] 刘荷花, 崔超, 陈晶. 一种改进的遗传算法求解旅行商问题[J]. 北京理工大学学报, 2013, 33(4): 390-393. (Liu H H, Cui C, Chen J. An improved genetic algorithm for solving traveling salesman problem[J]. J of Beijing Institute of Technology, 2013, 33(4): 390-393.)
- [4] 安建成, 靳海娟. 自适应策略在实数编码遗传算法中的应用研究[J]. 微电子学与计算机, 2011, 28(4): 140-142. (An J C, Jin H Y. Application of adaptive strategy in real-coded genetic algorithm [J]. Microelectronics and Computers, 2011, 28(4): 140-142.)
- [5] De Jong K A. An Analysis of the behavior of a class of genetic adaptive systems[D]. Michigan: University of Michigan, Literature Science and Arts Computer and Communication Sciences Department, 1975: 101-128.
- [6] 祁荣宾, 钱锋, 杜文莉, 等. 基于精英选择和个体迁移的多目标遗传算法[J]. 控制与决策, 2007, 22(2): 164-168. (Qi R B, Qian F, Du W L, et al. Multi-objective genetic algorithm based on elitist selection and individual migration[J]. Control and Decision, 2007, 22(2): 164-168.)
- [7] Liu Q, Wang X Y, Fu Q M, et al. Collaborative coevolutionary genetic algorithm[J]. J of Software, 2012, 23(4): 765-775.
- [8] 喻寿益, 邝溯琼. 保留精英遗传算法收敛性和收敛速度的鞅方法分析[J]. 控制理论与应用, 2010, 27(7): 843-848. (Yu S Y, Kuang S Q. Martingale analysis of convergence and convergence rate of elite genetic algorithm[J]. Control Theory & Applications, 2010, 27(7): 843-848.)
- [9] Chen C T, Wu C K, Hwang C. Optimal design and control of CPU heat sink processes[J]. IEEE Trans on Components & Packaging Technologies, 2008, 31(1): 184-195.
- [10] Chen C T, Chuang Y C. An intelligent run-to-run control strategy for chemical-mechanical polishing processes[J]. IEEE Trans on Semiconductor Manufacturing, 2010, 23(1): 109-120.
- [11] Dyer J D, Hartfield R J, Dozier G V, et al. Aerospace design optimization using a steady state real-coded genetic algorithm[J]. Applied Mathematics & Computation, 2012, 218(9): 4710-4730.
- [12] Tsai C W, Lin C L, Huang C H. Microbrushless DC motor control design based on real-coded structural genetic algorithm[J]. IEEE/ASME Trans on Mechatronics, 2011, 16(1): 151-159.
- [13] Valarmathi K, Devaraj D, Radhakrishnan T K. Real-coded genetic algorithm for system identification and controller tuning[J]. Applied Mathematical Modelling, 2009, 33(8): 3392-3401.
- [14] Michalewicz Z, Janikow C Z, Krawczyk J B. A modified genetic algorithm for optimal control problems[J]. Computers & Mathematics with Applications, 1992, 23(12): 83-94.
- [15] Goldberg D E, Deb K, Clark J H. Genetic algorithms, noise, and the sizing of populations[J]. Complex Systems, 1991, 6(4): 333-362.
- [16] Eiben A E, Hinterding R, Michalewicz Z. Parameter control in evolutionary algorithms[J]. IEEE Trans on Evolutionary Computation, 1999, 3(2): 124-141.
- [17] Koumousis V K, Katsaras C P. A saw-tooth genetic algorithm combining the effects of variable population size and reinitialization to enhance performance[J]. IEEE Trans on Evolutionary Computation, 2006, 10(1): 19-28.
- [18] Affenzeller M, Wagner S, Winkler S. Self-adaptive population size adjustment for genetic algorithms[C]. 2007 Int Conf on Computer Aided Systems Theory. Berlin: Springer, 2007: 820-828.
- [19] Meysenburg M M, Foster J A. The quality of pseudo-random number generators and simple genetic algorithm performance[C]. Proc of th Int Conf on Genetic Algorithms. New York: Morgan Kaufmann Publishers Inc, 1997: 276-281.
- [20] Paz E Cant. On random numbers and the performance of genetic algorithms[C]. Proc of Genetic and Evolutionary Computation Conf. New York: Morgan Kaufmann Publishers Inc, 2002: 311-318.
- [21] Meysenburg M M, Hoelting D, McElvain D, et al. How random generator quality impacts genetic algorithm performance[C]. Proc of Genetic and Evolutionary Computation Conf. New York: Morgan Kaufmann Publishers Inc, 2002: 480-487.
- [22] Chuang Y C, Chen C T, Hwang C. A simple and efficient real-coded genetic algorithm for constrained optimization[J]. Applied Soft Computing, 2016, 38(C): 87-105.
- [23] Deep K, Das K N. Performance improvement of real coded genetic algorithm with quadratic approximation

- based hybridization[J]. *Int J of Intelligent Defence Support Systems*, 2009, 2(2): 319-334.
- [24] Nakanishi H, Kinjo H, Oshiro N, et al. Searching performance of a real-coded genetic algorithm using biased probability distribution functions and mutation[J]. *Artificial Life & Robotics*, 2007, 11(1): 37-41.
- [25] Ortiz-Boyer D, Hervás-Martínez C, García-Pedrajas N. Improving crossover operator for real-coded genetic algorithms using virtual parents[J]. *J of Heuristics*, 2007, 13(3): 265-314.
- [26] Tang P H, Tseng M H. Adaptive directed mutation for real-coded genetic algorithms[J]. *Applied Soft Computing*, 2013, 13(1): 600-614.
- [27] Zhang G, Zhang G, Ma J, et al. Three real-coded genetic algorithms with new mutation operators[C]. *Proc of the 2007 Int Conf on Intelligent Systems and Knowledge Engineering*. Atlantis: Atlantis Press, 2007: 531-534.
- [28] Singh A, Deep K. Real coded genetic algorithm operators embedded in gravitational search algorithm for continuous optimization[J]. *Int J of Intelligent Systems Technologies & Applications*, 2015, 7(12): 1-12.
- [29] Tang P H, Tseng M H. Adaptive directed mutation for real-coded genetic algorithms[J]. *Applied Soft Computing*, 2013, 13(1): 600-614.
- [30] Yushchenko A G, Pashko D A, Gatilova O V. Evolutionary strategies of a genetic algorithm for a traveling salesman and new york taxi driver problems[R]. Kharkiv: Anniversary Edition "Information Systems" Department NTU "KhPI", 2014.
- [31] Kita H. A comparison study of self-adaptation in evolution strategies and real-coded genetic algorithms[J]. *Evolutionary Computation*, 2001, 9(2): 223-241.
- [32] Magalhães-Mendes J. A comparative study of crossover operators for genetic algorithms to solve the job shop scheduling problem[J]. *Wseas Trans on Computers*, 2013, 12(4): 164-173.
- [33] Subbaraj P, Rengaraj R, Salivahanan S. Enhancement of combined heat and power economic dispatch using self adaptive real-coded genetic algorithm[J]. *Applied Energy*, 2009, 86(6): 915-921.
- [34] Amjady N, Nasiri-Rad H. Solution of nonconvex and nonsmooth economic dispatch by a new adaptive real coded genetic algorithm[J]. *Expert Systems with Applications*, 2010, 37(7): 5239-5245.
- [35] Subbaraj P, Rengaraj R, Salivahanan S. Enhancement of self-adaptive real-coded genetic algorithm using Taguchi method for economic dispatch problem[J]. *Applied Soft Computing*, 2011, 11(1): 83-92.
- [36] Chuang Y C, Chen C T, Hwang C. A real-coded genetic algorithm with a direction-based crossover operator[J]. *Information Sciences*, 2015, 305: 320-348.
- [37] 汪定伟, 方述诚. 准时化生产计划的半无限规划模型及其遗传算法[J]. *控制与决策*, 1996, 11(4): 446-451. (Wang D W, Fang S C. Semi-infinite programming model and its genetic algorithm for quasi-time production planning[J]. *Control and Decision*, 1996, 11(4): 446-451.)
- [38] Peltokangas R, Sorsa A. Real-coded genetic algorithms and nonlinear parameter identification[R]. University of OULU Control Engineering Laboratory Report A, 2008: 1-28.
- [39] Michalewicz Z. Genetic algorithms + data structures = evolution programs[M]. 3rd ed. Berlin: Springer-Verlag, 1996: 631-653.
- [40] Mäkinen R A E, Jacques P, Jari T. Multidisciplinary shape optimization in aerodynamics and electromagnetics using genetic algorithms[J]. *Int J for Numerical Methods in Fluids*, 1999, 30(2): 149-159.
- [41] Deep K, Thakur M. A new mutation operator for real coded genetic algorithms[J]. *Applied Mathematics & Computation*, 2007, 193(1): 211-230.
- [42] Michalewicz Z, Logan T D, Swaminathan S. Evolutionary operators for continuous convex parameter spaces[J]. *Revista Española De Anestesiología Y Reanimación*, 1994, 52: 377-378.
- [43] Ripon K S N, Kwong S, Man K F. A real-coding jumping gene genetic algorithm(RJGGA) for multiobjective optimization[J]. *Information Sciences*, 2007, 177(2): 632-654.
- [44] 胡运权. 运筹学教程[M]. 北京: 清华大学出版社, 2012: 176-178. (Hu Y Q. Operations research course[M]. Beijing: Tsinghua University Press, 2012: 176-178.)
- [45] 谢晓锋, 张文俊, 杨之廉. 一种防止浮点遗传算法早熟收敛的父代选择策略[J]. *控制与决策*, 2002, 17(5): 625-628. (Xie X F, Zhang W J, Yang Z L. A parents selection strategy fighting premature convergence in floating genetic algorithms[J]. *Control & Decision*, 2002, 17(5): 625-628.)
- [46] Wang J Q, Wang F L, Dong Z G, et al. Research on evolutionary strategy of real-number genetic algorithm[J]. *Mathematics and Practice*, 2017, 47(11): 118-125.
- [47] 王晓慧, 刘雪英, 白梅花. 引入高斯变异和最速下降算子的人口迁移算法[J]. *计算机工程与应用*, 2009, 45(20): 57-60. (Wang X H, Liu X Y, Bai M H. Migration algorithm with gaussian variation and steepest descent operator[J]. *J of Computer Engineering and Applications*, 2009, 45(20): 57-60.)
- [48] 李晶晶, 戴月明. 自适应混合变异的蛙跳算法[J]. *计算机工程与应用*, 2013, 49(10): 58-61. (Li J J, Dai Y M. Frog leaping algorithm based on adaptive mixed variation[J]. *J of Computer Engineering and Applications*, 2013, 49(10): 58-61.)
- [49] 陈辉, 张家树, 张超. 实数编码混沌量子遗传算法[J]. *控制与决策*, 2005, 20(11): 1300-1303. (Chen H, Zhang J S, Zhang C. Quantum chaos quantum genetic algorithm[J]. *Control and Decision*, 2005, 20(11): 1300-1303.)
- [50] Ren Y T, Zhang D, Xu D Y, et al. Prediction and analysis of agricultural machinery total power based on real-valued genetic algorithm and neural network[J]. *Agricultural Mechanization Research*, 2018, 40(7): 13-18.