

控制与决策

Control and Decision

基于跳跃基因算子的改进实数遗传算法

宋莹莹, 王福林, 兰佳伟

引用本文:

宋莹莹, 王福林, 兰佳伟. 基于跳跃基因算子的改进实数遗传算法[J]. *控制与决策*, 2020, 35(9): 2277–2284.

在线阅读 View online: <https://doi.org/10.13195/j.kzyjc.2019.0024>

您可能感兴趣的其他文章

Articles you may be interested in

改进非支配排序精英遗传算法的篦冷机参数优化

Optimization of grate cooler parameters based on improved no-dominated sorting genetic algorithm II

控制与决策. 2020, 35(5): 1217–1225 <https://doi.org/10.13195/j.kzyjc.2018.0749>

求解有约束优化问题的实数遗传算法改进研究

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

控制与决策. 2019, 34(5): 937–946 <https://doi.org/10.13195/j.kzyjc.2017.1425>

自适应的非支配排序遗传算法

Adaptive non-dominated sorting genetic algorithm

控制与决策. 2018, 33(12): 2191–2196 <https://doi.org/10.13195/j.kzyjc.2017.1032>

求解离散调度问题的双机制头脑风暴优化算法

A brain storm optimization algorithm integrating diversity and discussion mechanism for solving discrete production scheduling problem

控制与决策. 2017, 32(9): 1583–1590 <https://doi.org/10.13195/j.kzyjc.2016.0620>

基于交叉模型的改进遗传算法

An improved genetic algorithm based on crossover model

控制与决策. 2016, 31(10): 1837–1844 <https://doi.org/10.13195/j.kzyjc.2015.1082>

基于跳跃基因算子的改进实数遗传算法

宋莹莹, 王福林[†], 兰佳伟

(东北农业大学 工程学院, 哈尔滨 150030)

摘要: 为了避免遗传算法在求解数值优化问题时出现搜索能力差、多样性缺失等弊端, 提出一种基于实数编码的改进遗传算法(IRCGA). 算法集成两个特别设计的算子: 模拟二进制跳跃基因算子(SBJG)和多方向交叉算子(MX). SBJG算子以染色体为操作对象, 本质上模拟了二进制跳跃基因操作中的插入运动, 即利用一种随机的方式将选定的染色体块插入到染色体位点, 实现种群内部染色体间的转位, 为种群提供额外的遗传多样性; MX算子通过增加交叉方向的方式扩大算子的搜索区域, 从而提升后代个体质量与算法的搜索能力. 在 11 个实例的基础上进行对比实验, 结果表明, 采用改进算子能够明显提升算法在求解数值优化问题时的性能, 同时, 相比于其他先进有效的算法, IRCGA 具有较强的搜索能力且能够维持一定的种群多样性, 从而验证了改进算法的有效性和可行性.

关键词: 实数遗传算法; 数值优化; 模拟二进制跳跃基因算子; 种群多样性; 多方向交叉算子; 搜索能力

中图分类号: TP391

文献标志码: A

DOI: 10.13195/j.kzyjc.2019.0024

引用格式: 宋莹莹, 王福林, 兰佳伟. 基于跳跃基因算子的改进实数遗传算法[J]. 控制与决策, 2020, 35(9): 2277-2284.

Improved real-coded genetic algorithm based on jumping gene operator

SONG Ying-ying, WANG Fu-lin[†], LAN Jia-wei

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

Abstract: In order to avoid the disadvantages of genetic algorithms in solving numerical optimization problems, such as poor search ability and lack of diversity, an improved real-coded genetic algorithm (IRCGA) is proposed. The algorithm integrates two specially designed operators: the simulated binary jumping gene (SBJG) operator and the multi-directional crossover (MX) operator. The SBJG operates on chromosomes. It essentially simulates the insertion motion in binary jumping gene operation, that is, inserts the selected chromosomal block into the chromosomal locus in a random way, and realizes the translocation between the chromosomes within the population, which provides additional genetic diversity for the population. The MX operator expands the search area of the operator by increasing the crossover direction to improve the quality of the offspring and the search ability of the algorithm. The comparative experiment is carried out on the basis 11 examples. The results show that the improved operator can significantly improve the performance of the algorithm in solving numerical optimization problems. At the same time, compared with other advanced and effective algorithms, the IRCGA has strong search ability and can maintain a certain population diversity, thus verifying the effectiveness and feasibility of the improved algorithm.

Keywords: real-coded genetic algorithm; numerical optimization; simulated binary jumping gene operator; population diversity; multi-directional crossover; search ability

0 引言

遗传算法的有效性通过在适者生存的原则下进化搜索能力与种群多样性实现, 交叉作为影响搜索能力的核心算子, 其与精英保留操作的结合会在一定程度上降低种群多样性, 影响算法性能^[1].

种群失去多样性, 会导致搜索过程停滞. 为了避免这种情况, 常用的方法是对变异算子进行改进, 但程度往往不够, 因此, 需要引入其他机制进行抵

消^[2-3]. 为此, Mitchell 等^[4]对遗传物质重排进行了研究, 发现跳跃基因能够为种群提供额外的多样性. 与跳跃基因算子一样, 设计高效的交叉算子能够促使算法有效地探索解空间, 因此, 对跳跃基因算子与交叉算子进行改进研究对于提升算法性能有着重要的意义.

在对跳跃基因算子的研究中, 文献[3, 5]最先提出了跳跃基因的概念, 即基因能够在染色体与染色

收稿日期: 2019-01-04; 修回日期: 2019-03-27.

基金项目: 公益性行业专项课题项目(201503116-04); 国家重点研发计划课题项目(2018YFD0300105); 黑龙江省哲学社会科学研究规划项目(18GLC205).

责任编辑: 王燕舞.

[†]通讯作者. E-mail: fulinwang1462@163.com.

体之间随意跳动,从而影响基因组的遗传行为.随后,Kasat等^[6]提出了一种二进制跳跃基因算子(JG),并将其应用于遗传算法中.但二进制编码的随机性会降低算法的局部搜索能力,为此,Sankararao等^[7-8]通过向算法内部引入特定适应机制,模拟JG的概念提出了SBJG-1算子,并成功将其应用于整数变量.Ripon等^[3]提出的SBJG-2算子根据随机选定的染色体数目调整算法所执行的操作.Ramteke等^[8]提出的SBJG-3算子利用指数分布对基因进行动态调控,在变量及其组合方面保留了大量原始信息.

在对交叉算子的研究中,最常见的是SBX算子,通过模拟单点交叉算子,使子代与父代的跨度成正比,但却限制了算子的搜索区域^[8].王吉权^[9]提出的DBHX算子利用目标函数值确定交叉方向,仅适用于单峰函数,并不适用于全局优化的普遍情形.文献[1, 8, 10]根据算法区间模式的概念提出了BLX-算子,但当个体间差异性较大时,算子的搜索方式类似于随机搜索,收敛性不强.

综上所述,传统的跳跃基因操作过度专注于模拟JG的过程,忽视了算子的效果,而交叉算子的方向过于单一,搜索区域受限,无法有效引导个体向最优区域进化,启发性不强.因此,为了更好地维持种群多样性,提升算法搜索能力,本文提出一种新的模拟二进制跳跃基因算子SBJG与多方向的交叉算子MX,并在11个实例的基础上进行对比实验.结果表明,采用改进算子能够明显提升算法在求解数值优化问题时的性能,同时,相比于其他先进有效的算法,IRCGA具有较强的搜索能力且能够维持一定的种群多样性,从而验证了改进算法的有效性和可行性.

1 改进算法描述

1.1 选择算子

改进算法选用排序分组选择^[9],能够增大配对个体间的差异性,有效避免近亲繁殖.

1.2 模拟二进制跳跃基因算子

传统的跳跃基因算子以基因为操作对象,利用基因的剪切、插入与互换运动实现染色体内部基因间的转位^[6-8].但对于实数编码的算法而言,由于变量的取值范围不同,剪切与互换会产生越界问题.此外,剪切与互换是在原基因库上进行的操作,只有插入会向基因库内引入新的遗传信息^[3].因此,本文提出的SBJG算子以染色体为操作对象,通过模拟基因的插入运动为种群提供多样性,具体操作步骤如下.

SBJG算子在规模为 n 的种群(X_1, X_2, \dots, X_n)内利用取值范围 $[1, n]$ 的整数随机数 R_x 选定两个染

色体位点 jgx_1, jgx_2 ,满足 $1 \leq jgx_1 \leq jgx_2 \leq n$,染色体位点 jgx_1 与 jgx_2 间的染色体块长度为 $l_{jgx} = (jgx_2 - jgx_1 + 1)$.利用 R_x 选定染色体的插入位点 I_x ,满足插入位点的个数 $N_{I_x} \leq l_{jgx}$.SBJG算子利用新产生的染色体块替换原染色体块并以随机插入选定位点的方式实现种群内部染色体间的转位,新染色体块中个体 X_i^{new} 的产生公式为

$$X_i^{new} = a + \text{rand}(1, D)(b - a). \quad (1)$$

其中: $a = [a_1, a_2, \dots, a_D]$ 、 $b = [b_1, b_2, \dots, b_D]$ 为变量取值下限和上限, D 为染色体维数, $\text{rand}(1, D)$ 为1行 D 列取值范围 $[0, 1]$ 的随机向量, $\text{rand}(1, D)(b - a)$ 为 $(b - a)$ 与 $\text{rand}(1, D)$ 中相同位置元素的乘积.以种群规模 $n = 6$,染色体块长度 $l_{jgx} = 3$ 为例,SBJG算子操作示意图如图1所示.

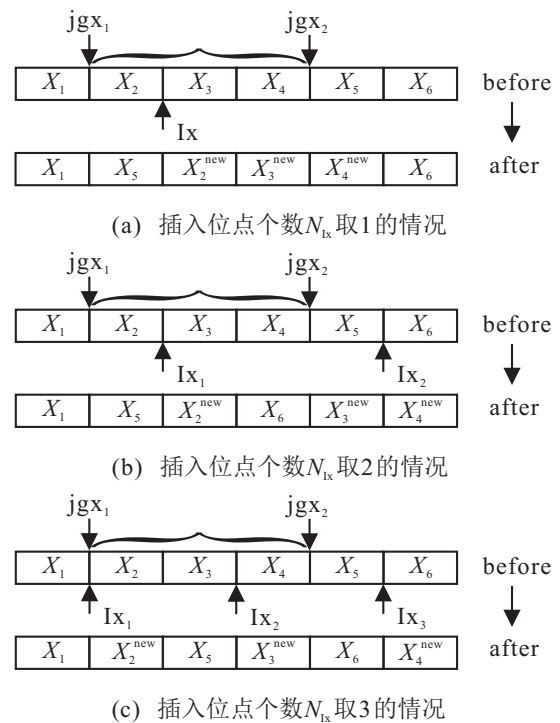


图1 SBJG算子的操作示意

图1中SBJG算子利用随机产生的新染色体块 $X_2^{new}, X_3^{new}, X_4^{new}$ 替换原染色体块.由关系式 $N_{I_x} \leq l_{jgx}$ 可知, N_{I_x} 的取值有3种情况,依次如图1(a)~图1(c)所示,其中图1(b)还可以表示在 I_{x_1} 处插入两条染色体、 I_{x_2} 处插入一条染色体的情况.

为了验证SBJG算子是否模拟了JG算子的本质,对JG和SBJG算子进行定性分析.设二进制编码字符串长度 $l_{str} = 20$,变量维数 $D = 3$,变量取值上下限为 $a_i = 0, b_i = 1 (i = 1, 2, 3)$.用随机生成的5000个实例 X_{old} 作为初始变量分别进行JG与SBJG操作,得到新变量 X_{new} ,其中JG操作得到的 X_{new} 需解码回实际值^[8].利用 X_{old} 与 X_{new} 的实际值计算操作对变量扰

动的分布函数 α ,有

$$\alpha = (X_{old} - X_{new}) / (b - a), \quad (2)$$

其中 α 在区间 $[-1, 1]$ 内波动,当 α 趋近于0时表示变量受到较低的扰动;当 α 接近-1和1时表示变量受到较大的扰动. JG算子(图2(a)和图2(b))与SBJG算子(图2(c)和图2(d))对变量扰动的分布如图2所示.

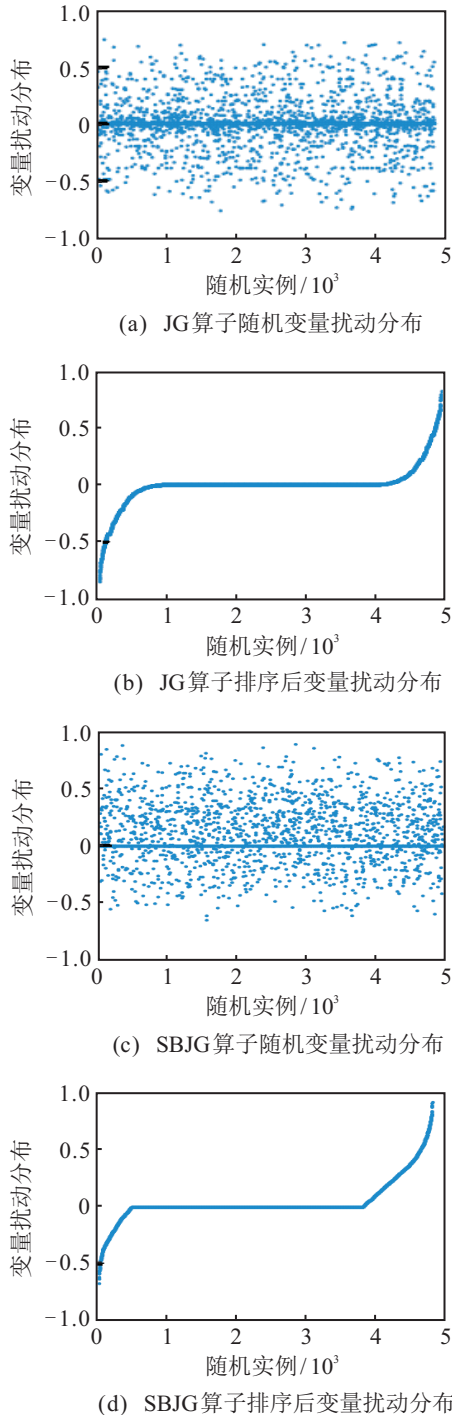


图2 JG算子与SBJG算子的变量扰动分布

图2(a)与图2(c)中分别有约80%和70%的点位于0附近,意味着SBJG与JG算子一样对大量实例产生了局部扰动. 将 α 按升序排序,如图2(b)和图2(d)所示,两幅图中均有明显的递增和递减曲线,表明

SBJG与JG算子均对部分变量产生了较大扰动,因此证实了SBJG算子模拟了JG算子对变量扰动的性质.

SBJG算子虽与变异作用相似,但是两者机制不同,SBJG在交叉操作前进行,不需要概率的设定,可以为种群提供额外的多样性. 此外,SBJG以染色体为操作对象,通过模拟基因的插入运动实现种群内部染色体间的转位,因此不需要计算复杂的种群结构,程序简单.

1.3 多方向的交叉算子

为了增大算子的搜索区域,MX算子在传统交叉方向的基础上新增了两个交叉方向. 传统方向以较大的概率指向目标函数值较优的一侧,为了使算子能够产生优质个体,新增方向与传统方向指向同侧,因此新方向通过将传统方向分别向两边旋转 90° 和 45° 获得的,这样既能保证新增方向与传统方向指向同侧,又能保证新增方向以不同的程度扩大搜索区域. 以配对个体 X_1 与 X_2 为例, X_1 的目标函数值优于 X_2 ,传统交叉方向为 X_2X_1 , X_2X_3 和 X_2X_4 分别为 X_2X_1 向两边旋转 90° 和 45° 形成的新方向. 算法具体步骤如下.

step 1: 确定点 X_3 ,根据向量垂直、模长相等的关系,对随机生成的点 $X' = [x'_1, x'_2, \dots, x'_D]$ 中 x'_D 进行归一化处理,使其位于 X_2X_3 所在的直线上,进而确定点 X_3 ,有

$$\begin{cases} x'_D = x_{2D} - \frac{\sum_{i=1}^{D-1} (x'_i - x_{2i})(x_{1i} - x_{2i})}{x_{1D} - x_{2D}}, \\ X_3 = X_2 + \frac{\sqrt{\sum_{i=1}^D (x_{1i} - x_{2i})^2}}{\sqrt{\sum_{i=1}^D (x'_i - x_{2i})^2}} (X' - X_2). \end{cases} \quad (3)$$

step 2: 作 $Rt\Delta X_1X_2X_3$ 斜边 X_1X_3 中点关于直角边的对称点,即点 X_4 ,有

$$X_4 = \frac{X_1 + X_2}{2} + \left(\frac{X_1 + X_2}{2} - \frac{X_1 + X_3}{2} \right). \quad (4)$$

由式(3)和(4)可以确定交叉方向 X_2X_1 、 X_2X_3 和 X_2X_4 ,算子在交叉方向 X_iX_j 上按下式产生个体:

$$\begin{cases} X_i^{ofs} = X_j - r_1(X_j - X_i), \\ X_j^{ofs} = X_j + r_2(X_j - X_i), \\ r_1 = [r_{11}, \dots, r_{1D}], (r_{11}, \dots, r_{1D}) \in [0, 0.5], \\ r_2 = [r_{21}, \dots, r_{2D}], (r_{21}, \dots, r_{2D}) \in [0, 0.5]. \end{cases} \quad (5)$$

step 3: 从产生的个体中选择两个目标函数值

最优的个体作为配对个体产生的后代. 以变量维数 $D = 2$ 为例, MX算子产生的交叉方向的空间分布如图3所示, 其中 X_{best} 为全局最优解.

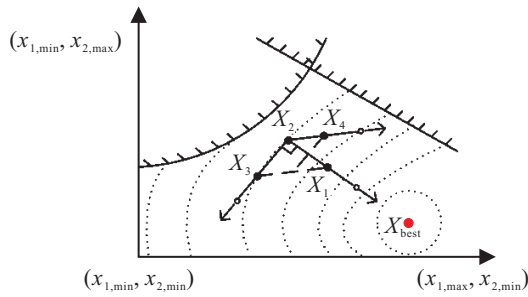


图3 交叉算子的空间分布

由图3可见, 不同于单方向的交叉算子^[9], MX算子中新增的交叉方向在不同程度上扩大了算子的搜索区域, 使算法能够更加全面地探索解空间, 从而提升算法的搜索能力.

1.4 变异算子

本文改进算法使用组合变异算子(CM). 当迭代次数为奇数代时, 执行均匀变异^[11], 此时算子具有较强的全局搜索能力; 否则, 执行高斯变异^[9], 此时算子具有较强的局部搜索能力.

1.5 进化策略

改进算法选用罚函数法处理约束条件^[4], 同时, 以算法是否达到给定最大迭代次数为终止条件. 改进算法的伪代码如下所示, 算法中的参数设置为: $n = 100$, $P_c = 1$, $P_m = 0.5$, $M = 10^{10}$, $\text{maxgen} = 1000$, $s = 40$.

输入: 种群大小 n , 交叉概率 P_c , 变异概率 P_m , 罚函数 M , 最大迭代次数 maxgen , 精英个体数 s ;

输出: 实际最优值 X^* , 种群多样性 div .

1. 初始种群 $X = \{X_i, i = 1, 2, \dots, n\}$

2. $\{F(X_i), i = 1, 2, \dots, n\} \leftarrow$ 计算目标函数值

(X_i)

3. $s \leftarrow$ 精英保留(X_i)

4. $\text{gen} = 1 //$ 迭代次数

5. while(满足最大迭代次数) do

6. $\{X^1, X^2\} \leftarrow$ RS算子(X)

7. $\{X^1, X^2\} \leftarrow$ SBJG算子 $\{X^1, X^2\}$

8. $X^c \leftarrow$ MX算子 $\{X^1, X^2\}$

9. $s \leftarrow$ 精英保留(X^c, s)

10. $x^M \leftarrow$ CM算子(X^c)

11. $s \leftarrow$ 精英保留(X^M, s)

12. $X \leftarrow$ 替换操作(X^M, s)

13. end while

14. 输出 X^* , div

操作: $s \leftarrow$ 精英保留(X_i)

1. $s \leftarrow$ 从(X_i)中选择 s 个最优个体

2. 输出 s

操作: $X \leftarrow$ 替换操作(X^M, s)

1. $X \leftarrow$ 从(X^M, s)中选择最优的 n 个个体

2. 输出 X

2 实验结果与分析

2.1 算子的基本功能

为了系统地研究各算子在算法中的作用, 实验依次禁用算法中的某一算子, 其余算子正常使用. 禁用MX算子时记为Test-1, 禁用SBJG算子时记为Test-2, 禁用CM算子时记为Test-3, 算法同时使用所有算子时记为Test-4^[12]. 选择11个基准测试函数^[1,9-10,13-15], 其中除 f_6 和 f_{10} 除外, $f_1 \sim f_{11}$ 为高维或非线性优化问题, f_6 和 f_{10} 为工程设计中减速器与热换热器优化问题, 测试函数如下:

$$f_1(x) = 1 + \frac{1}{4000} \sum_{i=1}^{10} x_i^2 - \prod_{i=1}^{10} \cos\left(\frac{x_i}{\sqrt{i}}\right);$$

$$\text{s.t. } -600 \leq x_i \leq 600, i = 1, 2, \dots, 10.$$

$$f_2(x) = 20 + e - 20 \exp\left(-\frac{1}{5} \sqrt{\frac{1}{10} \sum_{i=1}^{10} x_i^2}\right) - \exp\left(\frac{1}{10} \sum_{i=1}^{10} \cos(2\pi x_i)\right);$$

$$\text{s.t. } -32 \leq x_i \leq 32, i = 1, 2, \dots, 10.$$

$$f_3(x) = \sum_{i=1}^{10} x_i^2;$$

$$\text{s.t. } -600 \leq x_i \leq 600, i = 1, 2, \dots, 10.$$

$$f_4(x) = \left(\sum_{i=1}^{10} |x_i|\right) e^{-\sum_{i=1}^{10} \sin(x_i^2)};$$

$$\text{s.t. } -2\pi \leq x_i \leq 2\pi, i = 1, 2, \dots, 10.$$

$$f_5(x) = \sum_{i=1}^{10} x_i^2 + \left(\frac{1}{2} \sum_{i=1}^{10} i x_i\right)^2 + \left(\frac{1}{2} \sum_{i=1}^{10} i x_i\right)^4;$$

$$\text{s.t. } -5 \leq x_i \leq 10, i = 1, 2, \dots, 10.$$

$$f_6(x) = 0.7854x_1x_2^2(3.3333x_3^2 + 14.9334x_3 - 43.0934) - 1.508x_1(x_6^2 + x_7^2) + 7.477(x_6^3 + x_7^3) + 0.7854(x_4x_6^2 + x_5x_7^2);$$

$$\text{s.t. } \frac{27}{x_1x_2^2x_3} - 1 \leq 0, \frac{397.5}{x_1x_2^2x_3^2} - 1 \leq 0,$$

$$\frac{1.93}{x_2x_3x_4^3x_6^4} - 1 \leq 0, \frac{1.93}{x_2x_3x_4^3x_7^4} - 1 \leq 0,$$

$$\sqrt{\left(\frac{745x_4}{x_2x_3}\right)^2 + 1.69 \times 10^6} - 1 \leq 0,$$

$$\frac{\sqrt{\left(\frac{745x_4}{x_2x_3}\right)^2 + 1.57 \times 10^6}}{85x_7^2} - 1 \leq 0,$$

$$\frac{x_2x_3}{40} - 1 \leq 0, \frac{5x_2}{x_1 - 1} - 1 \leq 0, \frac{x_1}{12x_2} - 1 \leq 0,$$

$$2.6 \leq x_1 \leq 3.6, 0.7 \leq x_2 \leq 0.8, 17 \leq x_3 \leq 28,$$

$$7.3 \leq x_4, x_5 \leq 8.3, 2.9 \leq x_6 \leq 3.9, 5 \leq x_7 \leq 5.5.$$

$$f_7(x) = x_1^2 + x_2^2 + x_1x_2 - 14x_1 - 16x_2 + (x_3 - 10)^2 + 4(x_4 - 5)^2 + (x_5 - 3)^2 + 2(x_6 - 1)^2 - 5x_7^2 - 7(x_8 - 11)^2 + 2(x_9 - 10)^2 - (x_{10} - 7)^2 - 45;$$

s.t. $5x_1^2 + 8x_2 + (x_3 - 6)^2 - 2x_4 \leq 40,$

$$x_1^2 + 2(x_2 - 2)^2 - 2x_1x_2 + 14x_5 - 6x_6 \leq 0,$$

$$0.5(x_1 - 8)^2 + 2(x_2 - 4)^2 + 3x_5^2 - x_6 \leq 30,$$

$$3x_1 - 6x_2 - 12(x_9 - 8)^2 + 7x_{10} \geq 0,$$

$$4x_1 + 5x_2 - 3x_7 + 9x_8 \leq 105,$$

$$10x_1 - 8x_2 - 17x_7 + 2x_8 \leq 0,$$

$$8x_1 - 2x_2 - 5x_9 + 2x_{10} + 12 \geq 0,$$

$$3(x_1 - 2)^2 + 4(x_2 - 3)^2 + 2x_3^2 - 7x_4 \leq 120,$$

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

$$f_8(x) = 5.3578547x_3^2 + 0.8356891x_1x_5 + 37.293239x_1 - 40792.141;$$

s.t. $85.334407 + 0.0056858x_2x_5 + 0.0006262x_1x_4 - 0.0022053x_3x_5 \geq 0,$

$$-85.334407 - 0.0056858x_2x_5 - 0.0006262x_1x_4 + 0.0022053x_3x_5 + 92 \geq 0,$$

$$80.51249 + 0.0071317x_2x_5 + 0.0029955x_1x_2 - 0.0021813x_3^2 + 90 \geq 0,$$

$$-80.51249 - 0.0071317x_2x_5 - 0.0029955x_1x_2 - 0.0021813x_3^2 + 110 \geq 0,$$

$$9.300961 + 0.0047026x_3x_5 + 0.0012547x_1x_3 + 0.0019085x_3x_4 - 20 \geq 0,$$

$$-9.300961 - 0.0047026x_3x_5 - 0.0012547x_1x_3 - 0.0019085x_3x_4 + 25 \geq 0,$$

$$78 \leq x_1 \leq 102, 33 \leq x_i \leq 45, i = 2, 3, 4, 5.$$

$$f_9(x) = x_1;$$

s.t. $x_1 - 42x_2 - 44x_3 - 45x_4 - 47(x_5 + x_6) +$

$$50(x_2^2 + x_3^2 + x_4^2 + x_5^2 + x_6^2) \geq 0,$$

$$20x_2 + 12x_3 + 11x_4 + 7x_5 + 4x_6 - 40 \leq 0,$$

$$-20 \leq x_1 \leq 20, 0 \leq x_i \leq 1, i = 2, 3, \dots, 6.$$

$$f_{10}(x) = x_1 + x_2 + x_3;$$

s.t. $1 - 0.0025(x_4 + x_6) \geq 0,$

$$1 - 0.0025(x_5 + x_7 - x_4) \geq 0,$$

$$1 - 0.01(x_8 - x_5) \geq 0,$$

$$x_2x_7 - 1250x_5 - x_2x_4 + 1250x_4 \geq 0,$$

$$1250000 - x_3x_8 + x_3x_5 - 2500x_5 \leq 0,$$

$$100 \leq x_1 \leq 10000, 1000 \leq x_2, x_3 \leq 10000,$$

$$10 \leq x_i \leq 1000, i = 4, 5, \dots, 8.$$

$$f_{11}(x) = (x_1 - 10)^2 - 5(x_2 - 12)^2 - x_3^4 - 3(x_4 - 11)^2 - 10x_5^6 - 7x_6^2 - x_7^4 + 4x_6x_7 + 10x_6 + 8x_7;$$

s.t. $127 - 2x_1^2 - 3x_2^4 - x_3 - 4x_4^2 - 5x_5 \geq 0,$

$$282 - 7x_1 - 3x_2 - 10x_3^2 - x_4 + x_5 \geq 0,$$

$$-4x_1^2 - x_2^2 + 3x_1x_2 - 2x_3^2 - 5x_6 + 11x_7 \geq 0,$$

$$196 - 23x_1 - x_2^2 - 6x_6^2 + 8x_7 \geq 0,$$

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

为了获得公平的性能比较基础,算法在同一台计算机上运行,每个测试函数均运行1000次,实验评价指标取最优解的最小值(min),均值(mean)以及平均种群多样性(div)^[12],结果如表1~表3所示.

表1 评价指标min的实验结果

min	Test-1	Test-2	Test-3	Test-4
f_1	0.18	3.0e-3	2.0e-3	2.0e-3
f_2	0.09	0.00	0.00	0.00
f_3	8.51	0.00	0.00	0.00
f_4	6.0e-4	0.00	0.00	0.00
f_5	0.03	0.00	0.00	0.00
f_6	2995.13	2994.47	3000.60	2994.47
f_7	26.02	25.15	25.27	24.80
f_8	-30662.94	-30659.24	-30657.06	-30664.72
f_9	-16.80	-16.39	-14.83	-16.88
f_{10}	7282.38	7180.29	7951.50	7160.89
f_{11}	680.83	681.01	680.80	680.68

由实验结果可知,在11个测试函数上,对于指标div而言,实验Test-1最优,表明同时使用SBJG和CM能够提升种群多样性,且SBJG提升多样性的程度远大于CM(对比Test-2与Test-3可知).对于指标min和mean而言,实验Test-4的结果最优,表明MX具有较强的搜索能力.同时,也证实了算法同时使用所有算子时性能最优(对比实验Test-1与Test-2、Test-3、Test-4可知).

表2 评价指标 mean 的实验结果

min	Test-1	Test-2	Test-3	Test-4
f_1	0.42	0.03	8.8e-3	8.7e-3
f_2	0.66	0.00	0.00	0.00
f_3	40.16	0.00	0.00	0.00
f_4	6.0e-4	2.0e-4	1.0e-4	0.00
f_5	0.03	0.00	0.00	0.00
f_6	2996.12	2997.41	3009.48	2994.59
f_7	28.83	32.26	31.14	28.83
f_8	-30636.63	-30570.65	-30573.34	-30650.10
f_9	-14.59	-12.58	-8.69	-14.62
f_{10}	8016.80	8720.47	9740.48	8003.52
f_{11}	682.40	691.16	683.23	681.54

表3 评价指标 div 的实验结果

div	Test-1	Test-2	Test-3	Test-4
f_1	3.2e+4	3.1e-4	1.7e+3	1.2e+3
f_2	1.7e+3	0.0e+0	1.0e+2	5.4e+1
f_3	3.0e+4	0.0e+0	1.5e+3	1.4e+3
f_4	3.3e+2	0.0e+0	3.6e+1	3.3e+1
f_5	2.6e+2	0.0e+0	2.0e+1	1.6e+1
f_6	3.2e+1	1.6e-6	6.9e+0	4.6e+0
f_7	5.7e+3	5.1e-3	7.0e+1	4.4e+1
f_8	2.3e+2	2.3e-1	3.9e+1	9.6e+1
f_9	4.8e+1	2.3e-1	1.7e+1	1.5e+1
f_{10}	1.0e+5	2.6e+1	1.4e+4	2.1e+2
f_{11}	3.9e+2	3.7e-2	4.1e+1	2.3e+1

2.2 算法的有效性验证

为了验证改进算法的有效性,首先,将其对比于在使用跳跃基因算子方面较为有效的算法RCGA-S-1^[7]、RCGA-S-2^[3]、RCGA-S-3^[8];然后,替换改进算法中的SBJG算子为以往跳跃基因算子SBJG-1、SBJG-2、SBJG-3,替换后算法记为IRCGA-S-1、IRCGA-S-2、IRCGA-S-3;最后,将算法对比于在求解测试函数上较为有效的算法RCGA-1^[13]、RCGA-2^[9]、RCGA-3^[16]、RCGA-4^[1]、RCGA-5^[10].对于实验中涉及的算法而言,RCGA-3与RCGA-4的时间复杂度最大,为 $O(n^2\log(\maxgen))$,其余算法均为 $O(n\log(\maxgen))$,而算法的空间复杂度均为 $O(n \times D)$,因此改进算法与对比算法在运行过程中无明显差异.实验结果如表4~表7所示,其中指标time为算法的平均运行时间.

由实验结果可知,IRCGA求得的最小值min与均值mean最优,表明IRCGA较对比算法而言具有较强的搜索能力.通过计算各算法在测试函数上的整体平均种群多样性^[15]可以得出,IRCGA的平均种群多样性优于RCGA-S-2、IRCGA-S-3、RCGA-3、RCGA-4,表明IRCGA能够将种群多样性维持在一个相对较高的水平.同理,计算各算法的整体平均运行时间可知,除RCGA-3外,IRCGA的平均运行时间最短,而RCGA-3的搜索能力较差,因此相较于对比算法,IRCGA能够在相同的迭代次数内以最短的时间搜索到最优解,具有较快的收敛速度.综上,实验强有力地表明了改进算法的有效性.

表4 不同算法作用下测试函数 f_1 、 f_2 、 f_3 的实验结果

Algorithm	f_1				f_2				f_3			
	min	mean	div	time	min	mean	div	time	min	mean	div	time
RCGA-S-1	1.38	1.43	1.3e+4	5.44	2.17	2.38	8.0e+2	8.69	1359.42	1580.35	1.3e+4	5.44
RCGA-S-2	0.74	0.87	6.7e-14	26.63	2.37	2.38	5.0e-15	30.32	291.00	750.56	5.9e-14	28.23
RCGA-S-3	1.04	1.05	2.2e+4	4.92	2.73	2.90	1.4e+3	7.41	3259.50	4467.08	2.6e+4	5.06
IRCGA-S-1	8.6e-3	0.06	1.9e+2	5.51	0.00	0.00	8.8e+0	11.98	0.00	0.00	1.6e+2	5.15
IRCGA-S-2	9.9e-3	0.15	8.5e+3	19.68	0.00	0.03	4.5e+2	22.09	0.01	1.35	8.4e+3	17.22
IRCGA-S-3	5.5e-3	0.05	2.0e-12	11.70	0.00	0.00	3.7e-12	19.55	0.00	0.00	0.0e+0	12.05
RCGA-1	0.03	0.03	8.9e+4	5.22	0.01	0.02	1.2e+3	8.22	0.17	0.39	2.3e+4	5.09
RCGA-2	1.11	1.30	4.6e+4	5.03	2.03	2.40	2.4e+3	7.58	0.01	387.00	4.4e+4	4.83
RCGA-3	0.01	0.04	2.3e-7	3.06	0.00	0.00	1.9e-8	4.16	0.00	0.00	7.0e-12	2.42
RCGA-4	0.02	0.04	2.7e-8	7.58	0.00	0.00	3.8e-9	13.49	0.00	0.00	3.1e-13	7.31
RCGA-5	0.14	1.12	5.9e+4	4.66	1.51	2.26	3.6e+3	8.57	10.16	1799.08	2.9e+4	4.73
IRCGA	2.5e-3	0.01	1.2e+3	3.60	0.00	0.00	5.4e+1	9.85	0.00	0.00	1.4e+3	3.14

表5 不同算法作用下测试函数 f_4 、 f_5 、 f_6 的实验结果

Algorithm	f_4				f_5				f_6			
	min	mean	div	time	min	mean	div	time	min	mean	div	time
RCGA-S-1	0.00	1.9e-3	3.1e+2	5.30	0.17	0.21	1.6e+2	8.75	3 212.92	3 394.33	9.1e+1	13.21
RCGA-S-2	0.17	0.26	1.3e-15	28.79	0.01	0.04	3.9e-16	30.12	3 009.80	3 014.67	4.4e-14	28.01
RCGA-S-3	0.00	2.5e-3	3.6e+2	5.62	0.46	0.59	2.3e+2	7.56	3 014.11	3 017.43	3.9e+0	10.56
IRCGA-S-1	0.00	3.0e-4	7.0e+0	5.41	0.00	0.00	2.8e+0	12.67	2 994.53	2 996.84	4.0e-1	20.84
IRCGA-S-2	0.00	6.0e-4	1.1e+2	16.85	0.00	0.00	1.0e+2	21.33	2 994.52	2 994.72	1.4e+1	27.98
IRCGA-S-3	0.00	1.0e-4	0.0e+0	12.74	0.00	0.00	0.0e+0	19.51	3 047.66	1.9e+6	1.7e-13	25.25
RCGA-1	0.00	7.0e-4	3.7e+2	5.29	0.01	0.03	3.7e+2	8.74	2 995.16	2 995.49	3.9e+1	12.97
RCGA-2	0.00	1.7e-3	6.5e+2	5.32	0.63	0.86	3.9e+2	7.63	3 033.06	3 045.98	3.6e+1	11.29
RCGA-3	0.00	2.8e-3	0.0e+0	2.46	0.26	0.40	5.5e-18	3.73	3 433.71	3.4e+5	3.5e-16	0.42
RCGA-4	0.00	7.0e-4	0.0e+0	8.02	0.00	0.01	8.9e-6	13.66	2 994.48	2 994.66	9.9e-15	21.36
RCGA-5	0.00	2.3e-3	8.0e+2	4.69	0.10	0.26	6.3e+2	8.79	2 994.55	2 994.85	5.3e-7	13.54
IRCGA	0.00	0.00	3.3e+1	3.61	0.00	0.00	1.6e+1	9.76	2 994.47	2 994.59	4.6e+0	17.92

表6 不同算法作用下测试函数 f_7 、 f_8 、 f_9 的实验结果

Algorithm	f_8				f_7				f_9			
	min	mean	div	time	min	mean	div	time	min	mean	div	time
RCGA-S-1	39.36	42.22	2.9e+2	5.49	-30 644.68	-30 630.73	2.0e+2	4.97	-14.83	-14.06	3.4e+1	4.85
RCGA-S-2	28.84	32.09	4.7e-14	26.15	-30 635.46	-30 634.69	2.6e-13	21.98	-6.98	-6.97	3.8e-15	21.72
RCGA-S-3	33.24	39.93	5.9e+1	5.90	-30 638.39	-30 628.35	3.2e+1	5.20	-11.88	-10.77	9.1e+0	2.05
IRCGA-S-1	24.89	30.00	4.3e+0	6.43	-30 664.16	-30 617.25	4.4e+0	5.49	-16.70	-13.95	3.9e+0	5.06
IRCGA-S-2	25.11	28.86	2.2e+2	18.83	-30 662.89	-30 631.70	3.8e+1	16.69	-16.74	-14.62	4.3e+1	16.72
IRCGA-S-3	30.16	6.5e+8	1.9e-2	10.81	-30 588.92	-30 165.69	2.0e+0	9.69	-15.23	-7.08	1.7e-1	9.57
RCGA-1	25.79	32.44	6.5e+2	5.52	-30 639.81	-30 612.64	3.9e+2	5.07	-13.64	-13.48	9.2e+1	5.13
RCGA-2	33.27	34.74	4.9e+2	5.69	-30 636.75	-30 624.16	1.0e+3	4.93	-12.65	-10.40	2.3e+2	4.85
RCGA-3	1.3e+11	1.6e+12	0.0e+0	0.02	-30 160.59	-30 057.20	7.1e-16	0.50	2.65	9.53	7.5e-17	0.07
RCGA-4	25.12	32.08	3.8e-9	7.74	-30 555.64	-30 414.16	1.3e+0	7.24	-14.99	-13.38	8.9e-4	7.28
RCGA-5	39.10	62.00	1.1e+3	4.61	-30 614.55	-3 051.64	5.0e+2	4.01	-13.56	-8.74	8.8e+1	4.17
IRCGA	24.80	28.83	4.4e+1	3.72	-30 664.72	-30 650.11	9.6e+1	2.83	-16.88	-14.63	1.5e+1	2.65

表7 不同算法作用下测试函数 f_{10} 、 f_{11} 的实验结果

Algorithm	f_{10}				f_{11}			
	min	mean	div	time	min	mean	div	time
RCGA-S-1	8 034.98	8 515.80	5.5e+4	5.07	683.30	685.18	2.1e+2	5.41
RCGA-S-2	8 354.66	8 429.59	3.9e-12	22.90	684.55	686.67	1.2e-14	21.60
RCGA-S-3	8 007.91	8 219.06	1.1e+4	4.99	682.90	686.54	4.8e+1	4.88
IRCGA-S-1	7 283.82	8 358.29	3.3e+3	5.49	680.71	682.93	5.5e+0	6.47
IRCGA-S-2	7 171.80	8 005.15	7.0e+4	16.43	680.69	681.64	1.3e+2	17.82
IRCGA-S-3	7 764.41	3.3e+8	1.5e+2	10.64	682.87	697.09	6.1e-3	13.27
RCGA-1	7 300.66	8 435.39	1.2e+5	5.23	682.58	686.52	4.7e+2	5.30
RCGA-2	8 223.49	8 674.09	1.1e+6	5.07	681.93	682.46	3.1e+2	5.42
RCGA-3	2.8e+6	1.2e+9	1.1e-14	2.57	750.61	853.01	0.0e+0	2.45
RCGA-4	7 238.57	8 030.54	1.8e+2	7.28	681.26	692.22	8.8e-17	7.87
RCGA-5	8 478.76	12 353.42	1.4e+5	4.42	682.68	689.97	8.1e+2	4.94
IRCGA	7 160.89	8 003.53	2.1e+2	3.01	680.68	681.54	2.3e+1	3.65

3 结论

为了在提升算法搜索能力的同时维持一定的种群多样性,本文提出了一种模拟二进制跳跃基因算子SBJG与多方向的交叉算子MX.不同于传统跳跃基因算子,SBJG算子以染色体为操作对象,通过模拟基因的插入运动实现种群内部染色体间的转位,具有对大部分变量产生局部扰动的性质,为种群提供了额外的多样性.MX算子在传统交叉方向的基础上新增了两个与其指向同侧的交叉方向,在扩大算子搜索区域的同时保证算子以较大的概率产生优质的个体,提升了算法的搜索能力.在11个测试函数上与一些先进的算法进行了对比实验,由实验结果可以看出,改进算法具有较强的搜索能力和较快的收敛速度,且能够维持一定的种群多样性,从而验证了算法的有效性和可行性.

参考文献(References)

- [1] Ismkhan H. Black box optimization using evolutionary algorithm with novel selection and replacement strategies based on similarity between solutions[J]. Applied Soft Computing, 2018, 64(3): 260-271.
- [2] 何庆, 吴意乐, 徐同伟. 改进遗传模拟退火算法在TSP优化中的应用[J]. 控制与决策, 2018, 33(2): 219-225. (He Q, Wu Y L, Xu T W. Application of improved genetic simulated annealing algorithm in TSP optimization[J]. Control and Decision, 2018, 33(2): 219-225.)
- [3] Ripon K S N, Kwong S, Man K F. A real-coding jumping gene genetic algorithm (RJGGA) for multi objective optimization[J]. Information Sciences, 2007, 177(2): 632-654.
- [4] Mitchell M, Forrest S. Genetic algorithms and artificial life[J]. Artificial Life, 1994, 1(3): 267-289.
- [5] Freeling M. The dynamic genome: Barbara McClintock's ideas in the century of genetics[J]. Trends in Genetics, 1992, 8(12): 463.
- [6] Kasat R B, Gupta S K. Multi-objective optimization of an industrial fluidized-bed catalytic cracking unit (FCCU) using genetic algorithm (GA) with the jumping genes operator[J]. Computers & Chemical Engineering, 2003, 27(12): 1785-1800.
- [7] Sankararao B, Gupta S K. Multi objective optimization of the dynamic operation of an industrial steam reformer using the jumping gene adaptations of simulated annealing[J]. Asiapacific Journal of Chemical Engineering, 2010, 1(1/2): 21-31.
- [8] Ramteke M, Ghune N, Trivedi V. Simulated binary jumping gene: A step towards enhancing the performance of real-coded genetic algorithm[J]. Information Sciences, 2015, 325(20): 429-454.
- [9] 王吉权, 程志文, 代伟婷, 等. 求解有约束优化问题的实数遗传算法改进研究[DB/OL]. (2017-10-26)[2018-04-16]. <http://kns.cnki.net/kcms/detail/21.1124.TP.20180416.0932.024.html>. (Wang J Q, Cheng Z W, Dai W T, et al. Research on improvement of real-coded genetic algorithm for solving constrained optimization problems[DB/OL]. (2017-10-26)[2018-04-16]. <http://kns.cnki.net/kcms/detail/21.1124.TP.20180416.0932.024.html>.)
- [10] Pathan M V, Patsias S, Tagarielli V L. A real-coded genetic algorithm for optimizing the damping response of composite laminates[J]. Computers & Structures, 2018, 198(3): 51-60.
- [11] Contaldi C, Vafae F, Nelson P C. Bayesian network hybrid learning using an elite-guided genetic algorithm[J]. Artificial Intelligence Review, 2018, 52(293): 245-272.
- [12] Song Y, Wang F, Chen X. An improved genetic algorithm for numerical function optimization[J]. Applied Intelligence, 2019, 49(5): 1880-1902.
- [13] 杨新武, 杨丽军. 基于交叉模型的改进遗传算法[J]. 控制与决策, 2016, 31(10): 1837-1844. (Yang X W, Yang L J. An improved genetic algorithm based on crossover model[J]. Control and Decision, 2016, 31(10): 1837-1844.)
- [14] Yao X, Liu Y, Lin G. Evolutionary programming made faster[J]. IEEE Transactions on Evolutionary Computation, 1999, 3(2): 82-102.
- [15] 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(1): 87-105.
- [16] 任永泰, 张达, 许东阳, 等. 基于实数遗传算法与神经网络的农机总动力预测及分析[J]. 农机化研究, 2018, 40(7): 13-18. (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.)

作者简介

宋莹莹(1993—),女,博士生,从事遗传算法、数据挖掘、图像处理的研究, E-mail: songyingying@neau.edu.cn;

王福林(1982—),男,教授,博士生导师,从事遗传算法理论及应用、神经网络理论及应用等研究, E-mail: fulinwang1462@163.com;

兰佳伟(1993—),男,硕士生,从事遗传算法、组合预测的研究, E-mail: 269005186@qq.com.

(责任编辑: 郑晓蕾)