

文章编号: 2095-2163(2024)02-0041-07

中图分类号: TP391

文献标志码: A

基于改进蝗虫优化算法的智能组卷策略

严恺轩, 余隆勇

(浙江理工大学 计算机科学与技术学院(人工智能学院), 杭州 310018)

摘要: 针对传统组卷策略组卷时间长、试卷质量低等问题,本文提出了一种基于改进蝗虫优化算法的智能组卷策略。首先,综合知识点覆盖率、难度系数等因素,建立多目标评价模型;其次,将试题组合方案编码为蝗虫个体,通过蝗虫优化算法寻找最优组合方案,引入最优导向与随机竞争策略更新个体位置,以增强算法全局搜索能力;结合模拟退火思想,以使算法有能力跳出局部最优;引入余弦自适应函数调整搜索步长,以提升算法收敛速度。根据最优试题组合方案,生成一套高质量的试卷。实验结果表明,基于改进蝗虫优化算法的智能组卷策略与 PSO-GA 相比,组卷时间降低了 52%,适应度提升了 19%;与传统蝗虫优化算法相比,组卷时间降低了 30%,适应度提升了 14%。

关键词: 智能组卷; 蝗虫优化算法; 自适应函数; 随机竞争; 模拟退火

Intelligent test paper generation strategy based on enhanced Grasshopper Optimization Algorithm

YAN Kaixuan, YU Longyong

(College of Computer Science and Technology (School of Artificial Intelligence), Zhejiang Sci-Tech University, Hangzhou 310018, China)

Abstract: Aiming at the problems of long time and low quality of traditional paper forming strategy, an intelligent paper forming strategy based on enhanced Grasshopper Optimization Algorithm is proposed. Firstly, a multi-objective evaluation model is established by integrating factors such as the coverage rate of knowledge points and the difficulty coefficient. Secondly, the test question combination scheme is coded as Grasshopper individual, and the Grasshopper Optimization Algorithm is used to find the best combination scheme. At the same time, the optimal guidance and random competition strategy are introduced to update the individual position to enhance the global search ability of the algorithm. The idea of simulated annealing is combined so that the algorithm has the ability to jump out of the local optimum. The cosine adaptive function is introduced to adjust the search step to improve the convergence speed of the algorithm. According to the optimal combination scheme, a set of high-quality test papers is generated. The simulation results show that compared with PSO-GA, the intelligent test paper generation strategy based on the improved Grasshopper Optimization Algorithm reduces the response time of test paper generation by 53% and improves the fitness value by 19%. Compared with traditional Grasshopper Optimization Algorithm, the test paper generation time has been reduced by 30% and the fitness value has been improved by 14%.

Key words: intelligent test paper generation; Grasshopper Optimization Algorithm; self-adaptive function; random competition; simulated annealing

0 引言

随着计算机技术的高速发展,领域知识快速迭代,在线教育已经成为构建终身学习型社会的关键途径^[1]。从 20 世纪 70 年代计算机教育的兴起,到 2010 年左右 MOOC 在全球范围内的推广,计算机技

术与教育的深度融合已经成为提升教育信息化发展水平和质量的重要手段^[2]。在线考试是计算机技术在教育领域中的主要应用,能降低考试成本、避免纸张浪费、节省人力物力^[3]。然而,由于知识点繁杂,难度参差不齐,在组卷的过程中容易出现知识点覆盖率不高,没有区分度的情况,导致试卷质量低

基金项目: 浙江省重点研发计划项目(2020C03094); 浙江省教育厅一般科研项目(Y202147659, Y202250706, Y202250677); 国家自然科学基金(6207050141); 浙江省基础公益研究计划项目(QY19E050003)。

作者简介: 严恺轩(1998-),男,硕士研究生,主要研究方向:智能算法与数据挖掘; 余隆勇(1997-),男,硕士研究生,主要研究方向:智能算法与数据挖掘。

收稿日期: 2023-03-02

下,无法准确掌握学生的水平^[4]。因此,如何尽可能地覆盖知识点、把握试卷的难度、高效地生成一套试卷,是当前在线考试的研究重点。

目前常见的组卷策略包括随机抽取法和回溯试探法^[5]。该方法在题库中选择试题并匹配组成试卷的基本条件,通过反复抽题的迭代过程实现组卷^[6];回溯试探法改进了随机抽题法,搜索解的过程采用深度优先的方式,随机抽取试题并记录试题组合,搜索失败则重新试探^[7]。这两种组卷策略实现简单、组卷速度快,但考虑因素少。越来越多的学者对智能组卷策略提出了改进,贺建英等^[8]提出了一种基于遗传算法的智能组卷策略,取得了良好的效果,但是对于知识点覆盖率欠缺考虑;陈春燕等^[9]提出了一种基于遗传算法和粒子群算法的智能组卷策略,能够生成一套高质量的试卷,但是算法存在收敛速度慢的问题;潘婷婷等^[10]对知识点权重与错误率的关联展开研究,组卷质量得到提升。随着智能技术的发展,出现了越来越多的群智能算法。蝗虫优化算法(Grasshopper Optimization Algorithm, GOA)^[11]是2017年诞生的新型群智能算法,该算法的搜索效率优于大部分传统的群智能算法,具有较强的寻优能力^[12]。针对传统组卷策略组卷时间长,试卷质量低的问题,本文提出了一种基于改进蝗虫优化算法(Enhanced Grasshopper Optimization Algorithm, EGOA)的智能组卷策略。首先,综合考虑知识点覆盖率、难度系数等因素,建立评价模型;其次,通过改进的蝗虫优化算法寻找最优试题组合方案,算法结合随机竞争策略、模拟退火思想和余弦自适应函数提升最优试题组合方案的搜索效率;最后,根据试题组合方案,生成一套高质量的试题。

1 智能组卷问题分析与建模

组卷问题会受到许多条件的约束,是一个多目标优化问题^[13]。要生成一套高质量的试卷,应紧跟考试大纲,试题应具有一定难度,且覆盖足够的知识点,才能考察应试者的知识储备。因此,一套试卷需要包含难度、区分度、知识点覆盖率等指标^[14-15]。

假设一套试卷共有 m 道题,每道题有 n 个属性,则一套试卷可以建模为一个 $m \times n$ 的矩阵,如式(1)所示:

$$Q = \begin{bmatrix} q_{00} & q_{01} & \cdots & q_{0n} \\ q_{10} & q_{11} & \cdots & q_{1n} \\ \cdots & \cdots & \cdots & \cdots \\ q_{m0} & q_{m1} & \cdots & q_{mn} \end{bmatrix} \quad (1)$$

本文取试题的5个属性,目标矩阵满足的约束条件如下:

(1)题型: q_{i0} 表示试题的题型,试卷的题型有单选题、多选题、填空题、判断题、简答题。为了对题型加以区分,1表示单选题,2表示多选题,3表示填空题,4表示判断题,5表示简答题。

(2)分数: q_{i1} 表示试题的分值,所有的题目分值加起来需要满足预设的试卷总分。

(3)知识点: q_{i2} 表示试题的知识点,每道题目都涉及了不同的知识点。设题库中知识点的个数为 k_1 ,试题中涉及到的知识点个数为 k_2 ,因此,整套试题的知识点覆盖率 K 的计算方法如式(2)所示:

$$K = \frac{k_2}{k_1} \quad (2)$$

(4)难度系数: q_{i3} 表示试题的难度系数。设 S 为试卷的总分, D 为试卷难度系数,计算方法如式(3)所示:

$$D = \frac{\sum_{i=0}^m q_{i1} \times q_{i3}}{S} \quad (3)$$

(5)区分度: q_{i4} 表示试题的区分度,能够区分学生的实际水平。试卷总体区分度 T 计算方法如式(4)所示:

$$T = \frac{\sum_{i=0}^m q_{i1} \times q_{i4}}{S} \quad (4)$$

由于组卷系统的结果和出题人的意愿往往存在偏差,不可能满足所有条件。偏差主要体现在难度系数、知识点覆盖率和区分度上。因此,应该根据实际情况,选择偏差值最小的试题组合。

知识点覆盖率偏差值为 Ek , 计算方法如式(5)所示:

$$Ek = \frac{|Uk - K|}{Uk} \times 100\% \quad (5)$$

其中, Uk 为用户设定的知识点覆盖率。

难度系数偏差值为 Ed , 计算方法如式(6)所示:

$$Ed = \frac{|Ud - D|}{Ud} \times 100\% \quad (6)$$

其中, Ud 为难度系数。

区分度偏差值为 Et , 计算方法如式(7)所示:

$$Et = \frac{|Ut - T|}{Ut} \times 100\% \quad (7)$$

其中, U_t 为区分度。

组卷策略应该满足知识点覆盖率、难度系数、区分度的偏差最小,因此,目标函数的设置如式(8)~式(9)所示:

$$F = 1 - \alpha_1 \times Ek - \alpha_2 \times Ed - \alpha_3 \times Et \quad (8)$$

$$\alpha_1 + \alpha_2 + \alpha_3 = 1 \quad (9)$$

2 基于改进蝗虫优化算法的智能组卷策略

2.1 蝗虫优化算法

蝗虫优化算法是 Semari 在 2017 年首次提出的,用来解决优化问题。通过模拟蝗虫的觅食行为,将幼虫的小范围移动映射为算法的局部开发,成虫的大范围跳跃映射为算法的全局勘探。蝗虫的位置受到蝗虫间的作用力、重力、风平流作用力的影响,实际问题中一般仅考虑蝗虫之间的作用力。根据蝗虫之间的距离,作用力可以分为吸引力和排斥力。基本 GOA 算法的位置更新如式(10)所示:

$$X_i^d = c \left\{ \sum_{j=1, j \neq i}^K c \frac{ub_d - lb_d}{2} s(|x_j^d - x_i^d|) \frac{x_j - x_i}{d_{ij}} \right\} + T_d \quad (10)$$

其中, ub_d 和 lb_d 分别为第 d 维空间的上下界; X_i 为蝗虫 i 的位置; d_{ij} 为蝗虫 i 和蝗虫 j 的距离; T_d 为当前迭代轮次的最优解。

c 为线性递减系数,用于平衡算法开发和勘探的过程,如式(11)所示:

$$c = c_{\max} - t \cdot \frac{c_{\max} - c_{\min}}{T_{\max}} \quad (11)$$

其中, c_{\max} 和 c_{\min} 分别为 c 的最大值和最小值; T_{\max} 为最大迭代次数; t 为当前迭代轮次。

GOA 的搜索过程是在迭代过程中依靠蝗虫间的作用力、最优位置、当前位置等信息,更新每只蝗虫位置,靠近食物的过程。在 GOA 中,每只蝗虫的位置被映射为搜索空间的每个解,通过适应度函数衡量每个解的优劣,食物的位置被映射为全局最优解,搜索食物源泉的过程即为算法的寻优过程^[16]。

2.2 改进策略

在蝗虫优化算法中,每只蝗虫都会参与搜索过程,与其它群智能算法相比,GOA 具有更高的搜索效率。为了进一步提升 GOA 的搜索性能,本文对 GOA 做出了以下几点改进,并将改进后的算法应用于智能组卷策略。

2.2.1 自适应权重

GOA 中的 c 是关键参数,控制着算法从勘探到开发过程的转变。在传统蝗虫优化算法中, c 是线性递减的,无法动态调整算法的搜索步长,从而影响算法的收敛速度。因此,本文提出了余弦自适应权重调整算法的搜索步长,式(12):

$$C(l) = \cos\left(\frac{\pi}{2} \times \frac{l}{L}\right) \times (c_{\max} - \frac{l}{L} \times (c_{\max} - c_{\min})) \quad (12)$$

其中, L 为最大迭代数, l 为当前迭代轮次。

改进后的公式在算法初期步长更大并且下降幅度更慢,有助于算法的全局搜索;算法中期步长下降幅度更大,有助于提升算法收敛速度;算法末期步长更小,下降幅度更慢,有助于算法的局部搜索。

2.2.2 随机竞争策略

由蝗虫优化算法的位置更新式(10)~式(11)可知,每只蝗虫的位置由当前位置、蝗虫间的交流信息和当前迭代的最优位置决定,因此 GOA 具有较强的局部搜索能力,但全局搜索能力偏弱^[17]。因此,本文在迭代过程的前半段引入了随机竞争策略。随机竞争策略是指每轮迭代中,随机选择两只蝗虫,比较适应度值,适应度更高的为胜利个体,适应度更低的为失利个体,参与算法的迭代过程,以扩大算法的搜索空间。优化后的位置更新如式(13)~式(14)所示:

$$X_{\text{new}}(t+1) = \alpha_1 C + \alpha_2 (X_{\text{win}}(t) - X_i(t)) + \alpha_3 (X_{\text{loss}}(t) - X_i(t)) + \alpha_4 (X_{\text{best}}(t) - X_i(t)) \quad (13)$$

$$C = \sum_{j=1, j \neq i}^N c \frac{ub_d - lb_d}{2} s(|x_j^d - x_i^d|) \frac{x_j - x_i}{d_{ij}} \quad (14)$$

其中, C 是蝗虫的搜索步长,由蝗虫间的距离和交流信息共同决定; X_{best} 是当前迭代轮次中保存的最优解; X_i 是第 i 只蝗虫的位置; X_{win} 是竞争胜利的蝗虫; X_{loss} 是竞争失败的蝗虫; α_1 是蝗虫之间的交流因子,设为 1; α_2 是胜利权重,为了让竞争胜利的蝗虫以更大的权重参与搜索过程, α_2 取值为 $[0.7, 1]$ 的随机值; α_3 是失利权重,为了让竞争失败的蝗虫以更小的权重参与算法搜索过程, α_3 取值为 $[0, 0.3]$ 的随机值; α_4 是最优解记忆权重,参考参数 c 的自适应权重,有助于控制算法勘探和开发过程的转变,式(15):

$$\alpha_4 = \cos\left(\frac{\pi}{2} \times \frac{t}{T_{\max}}\right) \quad (15)$$

新的位置更新公式结合了所有蝗虫的交流信息、每只蝗虫的自身位置、迭代过程的最优位置与随机竞争的位置,其中蝗虫间的交流信息能够增强算法全局搜索的能力,随机竞争位置能够扩大算法搜索空间,迭代过程中的最优位置能让每只蝗虫朝最优的方向去搜索。

2.2.3 模拟退火思想

随着迭代次数的增加,算法容易陷入局部最优,而此刻无论是优胜组或者是失利组的蝗虫都会靠近这个局部的最优解,导致算法过早收敛。因此在算法后半段,为了使算法能够跳出局部最优,本文引入了模拟退火的思想。

模拟退火算法是1953年由Metropolis^[18]提出的,来源于固体的退火原理,固体在高温情况下冷却至常温状态,内部的粒子会随着温度的降低,自身内能的减小,粒子的状态会区域稳定,最后在常温情况下达到基态;某个特定温度下,由于粒子的运动,固体内部的内能会发生改变,如果系统朝着内能减小的方向进行,就接受这种变化,反之,则以一定的概率接受这种变化^[19]。

模拟退火算法遵循Metropolis准则,以一定的概率接受较差的解。概率计算规则如式(16)所示:

$$P_{ij}^T = \begin{cases} 1, & E(j) \leq E(i) \\ e^{-\frac{E(j)-E(i)}{KT}}, & \text{others} \end{cases} \quad (16)$$

其中, $E(i)$ 为固体在*i*状态下的内能; $E(j)$ 为固体在*j*状态下的内能; K 为玻尔兹曼常数; T 为温度。

在温度*T*时,固体从状态*i*到状态*j*,如果状态*j*的内能小于状态*i*,则接受*j*为当前状态,否则按照公式(16)以一定的概率接受该状态。

在模拟退火思想中,蝗虫的随机位置如式(17)所示^[13]:

$$X_{\text{new}i} = \frac{1}{2} \left(x_i + \frac{(ub - lb) \times 2 \times rand}{l^2} - \frac{ub - lb}{l^2} \right) \quad (17)$$

其中, $rand$ 为(0,1)中的随机数。

将模拟退火思想引入蝗虫优化算法,有助于扩大算法搜索空间,跳出局部最优,增强全局搜索能力。

2.3 改进的蝗虫优化算法步骤

本文提出的改进的蝗虫优化算法(EGOA)流程如图1所示。

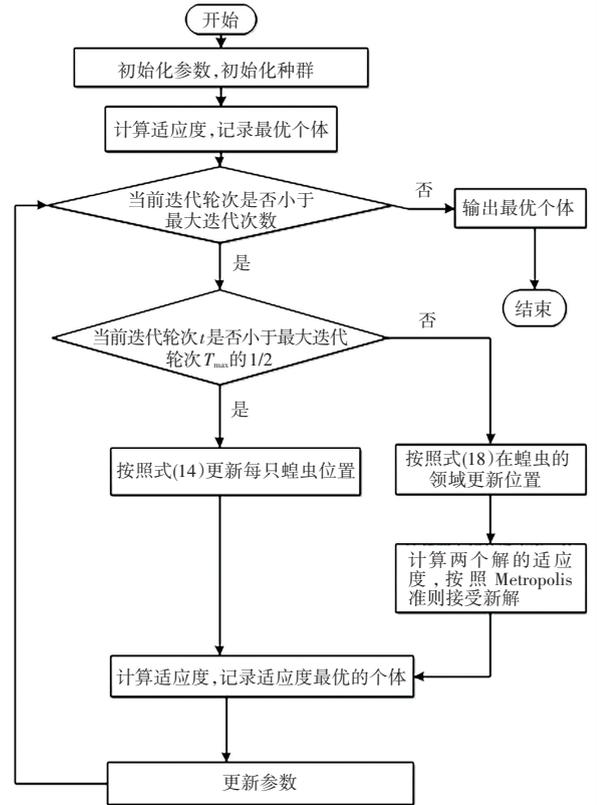


图1 EGOA流程图

Fig. 1 Flow chart of EGOA

2.4 基于EGOA的智能组卷策略

提升算法的搜索效率,编码方式非常重要。目前常见的编码方式为二进制编码,这样的编码方式比较简单,但是在本文的智能组卷策略中,题库中题目的数量非常大,采用二进制编码会导致编码过长^[20]。由于题库在创建的时候,题目已经按照题型进行分类,因此本文采用分段实数编码,不同题型间的编码互不干扰。比如一套试题中,有选择题5题,填空题2题,判断题3题,简答题2题。其中一种编码方案见表1。

表1 试卷编码方案

Table 1 Test paper coding scheme

题型	选择题	填空题	判断题	简答题
编码	15, 24, 33, 56, 78	100, 22	34, 56, 22	134, 27

本文将试卷的编码方案映射为EGOA中的蝗虫,通过式(8)~式(9)计算适应度函数值,衡量每种组合方案的优劣。

基于EGOA智能组卷策略步骤:

步骤1 初始化试卷编码方案,每一种组合方案编码为算法中的每只蝗虫;

步骤2 初始化迭代次数、试题总分、题型、难度系数、区分度等参数;

步骤 3 求出每种编码方案的适应度值, 保存适应度值最优的组合方案;

步骤 4 如果迭代轮次小于最大迭代次数的 1/2, 跳转至步骤 5, 否则跳转至步骤 6;

步骤 5 按照式 (13) 更新蝗虫位置;

步骤 6 按照式 (17) 在蝗虫邻域范围内更新位置, 计算两个解的适应度, 并且由 Metropolis 准则接受解;

步骤 7 计算适应度, 保存适应度最优的个体;

步骤 8 更新参数;

步骤 9 如果迭代次数未达到最大值, 则跳到步骤 4; 如果达到了最大值, 则输出最优试题组合方案, 算法结束。

3 实验

3.1 EGOA 性能测试

本文通过 4 个基准测试函数验证 EGOA 的性能, 基准测试函数详细信息见表 2。

表 2 基准测试函数

Table 2 Benchmark test function

函数	表达式
Sphere	$F_1 = \sum_{i=0}^n x_i^2$
Schwefel 2.22	$F_2 = \sum_{i=1}^n x_i + \prod_{i=1}^n x_i $
Schwefel 2.21	$F_3 = \max\{ x_i , 1 \leq i \leq n\}$
Ackley	$F_4 = -20 \exp(-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2}) - \exp(\frac{1}{n} \sum_{i=1}^n \cos(2\pi x_i)) + 20 + e$

为了验证算法的寻优精度和稳定性, 本文取算法独立运行 50 次的最小值、平均值、方差作为评价指标, 将 PSO-GA、GOA 和本文提出的 EGOA 的运行结果进行对比。设置种群规模为 50, 迭代次数为 200, c_{\max} 为 1, c_{\min} 为 0.000 01, 初始温度设置为 100, 结束温度设置为 1, 退火系数设置为 0.95, 玻尔兹曼常数设置为 10。实验结果见表 3。

根据表 3 可知, 针对本文选取的 4 个测试函数, EGOA 的寻优精度和稳定性都优于其它算法。对于函数 F_4 , EGOA 和 PSO-GA 都能取得全局最优解, 但是 EGOA 算法的方差和平均值都小于其他算法。对于函数 F_1 、 F_2 、 F_3 , EGOA 取得最优解的平均值, 方差和最小值都小于其它算法。

表 3 算法性能对比

Table 3 Comparison of algorithm performance

函数	指标	PSO-GA	GOA	EGOA
F_1	平均值	1.74×10^{-6}	4.38×10^{-14}	2.54×10^{-33}
	方差	1.97×10^{-6}	8.33×10^{-14}	4.35×10^{-33}
	最小值	1.12×10^{-7}	4.83×10^{-25}	9.55×10^{-33}
F_2	平均值	1.45	5.11×10^{-7}	2.46×10^{-21}
	方差	1.32	4.32×10^{-7}	3.23×10^{-21}
	最小值	1.03	1.66×10^{-7}	5.42×10^{-21}
F_3	平均值	1.66×10^{-2}	6.63×10^{-8}	2.21×10^{-20}
	方差	2.61×10^{-2}	5.19×10^{-8}	4.74×10^{-21}
	最小值	2.41×10^{-1}	7.12×10^{-9}	6.43×10^{-23}
F_4	平均值	1.02	7.14×10^{-16}	5.56×10^{-25}
	方差	1.78	5.34×10^{-16}	4.43×10^{-26}
	最小值	0.14	0	0

3.2 基于 EGOA 的智能组卷策略性能测试

为了验证本文提出的 EGOA 算法在智能组卷策略中的性能, 本文选择 PSO-GA、GOA 与 EGOA 算法进行对比。本实验采用 WebLearn 的某个课程题库作为数据来源, 采用 MongoDB 数据库, 实验环境为 Windows10 操作系统, 处理器为 AMD Ryzen 5 5600X, 内存为 16 GB, 实验平台为 IntelliJ IDEA2020。实验参数见表 4。

表 4 实验参数设置

Table 4 Experimental parameter setting

参数名称	取值
种群大小	100
α	$\alpha_1 = 0.14, \alpha_2 = 0.42, \alpha_3 = 0.36$
c_{\max}	1
c_{\min}	0.000 01
初始温度	100
结束温度	1
玻尔兹曼常数	10
退火系数	0.95
总分	100
题型(题量)	单选题(10), 多选题(5), 填空题(10), 判断题(10), 主观题(2)
题型(分数)	单选题(2), 多选题(4), 填空题(2), 判断题(2), 主观题(10)
难度系数	0.7
区分度	0.7
知识点覆盖率	1

本文通过两类实验来验证 EGOA 智能组卷策略的效率,分别是改变算法迭代次数和改变题库的题目数量,并将 EGOA 应用于智能组卷策略的实验结果与 PSO-GA、GOA 进行比较。

(1) 迭代次数能度量算法找到最优解所耗费的时间,衡量算法的收敛速度,因此本实验通过改变算法的迭代次数,比较在不同迭代次数的条件下,算法应用于组卷策略中的组卷时间和适应度函数值,来验证本文提出的 EGOA 的优越性。实验设定题库数量为 600,迭代次数由 10 到 100 递增,将 PSO-GA、GOA、EGOA 应用于组卷策略中,组卷时间和适应度值的变化如图 2 和图 3 所示。

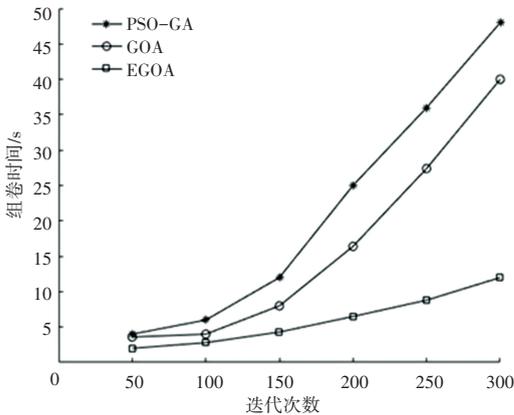


图 2 组卷时间随迭代次数变化图

Fig. 2 Change of test paper generation time with iteration number

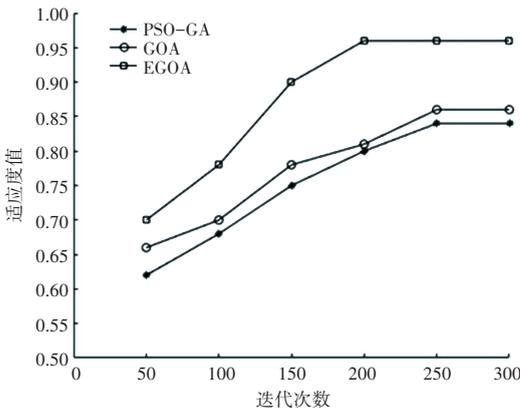


图 3 适应度值随迭代次数变化图

Fig. 3 Change of fitness value with iteration number

由图 2 可知,随着迭代次数的增加,组卷时间逐渐增大。迭代次数小于 100 时,组卷时间差别不大,当迭代次数大于 100,GOA 与 PSO-GA 的组卷时间显著增长,而 EGOA 的组卷时间平稳上升。由图 3 可知,随着迭代次数增加,适应度值逐渐增大,当迭代次数达到 200 时,本文提出的 EGOA 优先达到平稳状态。无论是组卷时间还是适应度值,PSO-GA 算法的劣势都非常明显。而 EGOA 在组卷时间的

表现上更突出,适应度值也更大,说明 EGOA 的收敛速度更快,能够生成质量更好的试题。

(2) 题库中题目数量不同的情况下,不同算法应用于智能组卷策略的效果也是不同的。本实验向题库中依次加入 100 道题,加 10 次,迭代次数设置为 200,并且对比 PSO-GA、GOA 与 EGOA 的效果。不同试题数量下,PSO-GA、GOA 和 EGOA 应用于智能组卷策略中的组卷时间和适应度值的变化如图 4 和图 5 所示。

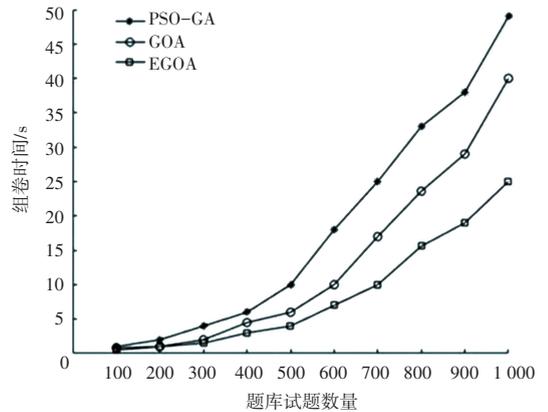


图 4 组卷时间随试题数量变化图

Fig. 4 Change of test paper generation time with test question number

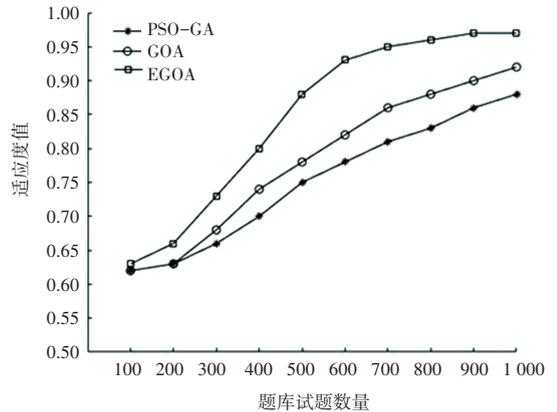


图 5 组卷时间随试题数量变化图

Fig. 5 Change of fitness value with test question number

由图 4 可知,随着题库中试题数量的增加,组卷时间逐渐增大。试题数量较少的时候,3 种算法的组卷时间相差无几,当试题数目大于 500,PSO-GA 的组卷时间上升明显,EGOA 算法的组卷时间缓慢上升。由图 5 可知,当试题数量较少时,3 种算法的适应度值没有太大差别,随着试题数量的增加,适应度值逐渐增大。当试题数目大于 500 时,EGOA 的适应度值明显大于其他算法。当迭代次数达到 900 时,本文提出的 EGOA 优先达到平稳状态,证明在智能组卷策略中,EGOA 算法优于其他算法,在题库规模较大的情况下优势更明显。

4 结束语

为了解决传统组卷策略组卷时间长,试题质量低的问题,本文提出了一种改进的蝗虫优化算法,并将其应用于智能组卷策略中。综合考虑试卷知识点覆盖度、难度系数等因素,建立多目标评价模型;通过改进的蝗虫优化算法找出最优试题组合方案,同时通过引入余弦自适应函数、随机竞争策略和模拟退火思想,来解决 GOA 收敛速度慢、容易陷入局部最优的问题。基准函数测试结果显示,与 GOA、PSO-GA 相比,本文提出的 EGOA 的寻优精度更高,收敛速度更快,稳定性更强。通过仿真实验,将 EGOA 和 PSO-GA、GOA 对比,验证不同算法在智能组卷策略中的效果。实验结果表明,本文提出的 EGOA 的智能组卷策略的组卷时间更低,适应度更高,生成的试卷质量更好。和 PSO-GA、GOA 相比,基于 EGOA 的智能组卷策略具有更好的竞争力,但当题库中试题数量较少的时候,EGOA 效果一般。如何让算法在题库规模比较小的情况下也具有足够的竞争力,这是后续研究的方向。

参考文献

- [1] 歹节,李青山,储华,等.突破智慧教育:云基于图学习的课程推荐系统[J].软件学报,2022,33(10):3656-3672.
- [2] 郑庆华,董博,钱步月,等.智慧教育研究现状与发展趋势[J].计算机研究与发展,2019,56(1):209-224.
- [3] 赵崇阳.基于遗传算法的学生在线竞赛系统研究[J].自动化与仪器仪表,2022(4):146-150.
- [4] 赵博,宁慧,张汝波.基于遗传算法的智能组卷系统的研究与

- 实现[J].应用科技,2021,48(2):50-53.
- [5] 郭秀娟,张坤鹏.基于蚁群混合遗传算法的组卷问题研究[J].吉林建筑大学学报,2017,34(4):79-83,102.
- [6] 彭云建,欧善国,梁进.在线气象科普知识竞赛试题的自动组卷方法[J].计算机技术与发展,2021,31(5):209-214.
- [7] 唐静,舒小松.基于回溯法的智能组卷算法的研究[J].信息与电脑(理论版),2018(19):69-71.
- [8] 贺建英,王光琼,唐青松.一种基于遗传算法的智能组卷策略优化研究[J].计算机与数字工程,2019,47(1):130-135.
- [9] 陈春燕,刘梦赤.基于粒子群遗传算法的智能组卷策略[J].计算机与现代化,2021(8):16-23.
- [10] 潘婷婷,詹国华,李志华.基于知识点与错误率关联的个性化智能组卷模型[J].计算机系统应用,2018,27(5):139-144.
- [11] SAREMI S, MIRJALILI S, LEWIS A. Grasshopper optimisation algorithm: theory and application [J]. Advances in Engineering Software, 2017, 105:30-47.
- [12] 邵鸿南,梁倩,王李森,等.基于反向学习和种群引导的多目标蝗虫优化算法[J].计算机工程与科学,2021,43(5):944-950.
- [13] 刘雅莉.一种改进遗传算法的自动组卷系统优化研究[J].微型电脑应用,2020,36(8):28-30,40.
- [14] 肖桂霞,刘武初,朱伟,等.基于遗传算法智能组卷的去重题方法[J].计算机工程,2021,38(11):150-152.
- [15] 胡新源,赵当丽,李辉,等.基于定向变异遗传算法的智能组卷算法研究[J].电子设计工程,2021,29(17):65-69.
- [16] 杨文珍,何庆,杜逆索.具有扰动机制和强化莱维飞行的蝗虫优化算法[J].小型微型计算机系统,2022,43(2):247-253.
- [17] 何庆,林节,徐航.混合柯西变异和均匀分布的蝗虫优化算法[J].控制与决策,2021,36(7):1558-1568.
- [18] METROPOLIS N, ROSENBLUTH A W, ROSENBLUTH M N, et al. Equation of state calculations by fast computing machines [J]. The Journal of Chemical Physics, 1953,21(6):1087-1092.
- [19] 李阳州,顾磊.基于曲线自适应和模拟退火的蝗虫优化算法[J].计算机应用研究,2019,36(12):3637-3643.
- [20] 罗少华.基于自适应技术的教育类课程考试智能组卷方法[J].微型电脑应用,2022,38(4):40-43.