

单目标优劣交叉的微分进化解决答辩分组问题^①

曹 莉, 许玉龙, 李亚威

(河南中医药大学, 郑州 450046)

通讯作者: 许玉龙, E-mail: flyxyl@qq.com

摘 要: 论文答辩分组是高校管理中的常见问题, 为保证分组的公平性和科学性, 须考虑老师和学生间的若干限制条件, 存在两个相互矛盾的条件——互回避原则和均匀原则. 寻求最优的分组方案使两个条件都能尽量满足, 是本文要解决的核心问题. 通过建立数学模型, 使答辩分组问题可用矩阵编码表示, 并将两个冲突的条件整合为一个目标函数. 然后采用单目标优劣交叉的微分进化求解该问题, 通过构造合适染色体、适应度函数, 进行初始化种群、优劣交叉、变异修正等操作, 逐代进化得到最优解. 为验证该方法的优越性, 与一般算法求解效果对比, 结果表明优劣交叉的微分算法能够得到更科学的分组方案、更多的可行方案.

关键词: 单目标; 微分进化; 答辩分组; 数学模型; 矩阵编码; 适应度

引用格式: 曹莉, 许玉龙, 李亚威. 单目标优劣交叉的微分进化解决答辩分组问题. 计算机系统应用, 2017, 26(9): 32-39. <http://www.c-s-a.org.cn/1003-3254/5966.html>

Based on Single-Objective Differential Evolution with Superior-Inferior Crossover Scheme to Solve the Problem of Defense Grouping

CAO Li, XU Yu-Long, LI Ya-Wei

(Henan University of Chinese Medicine, Zhengzhou 450046, China)

Abstract: The thesis defense grouping is a common problem in the college management. To ensure the fairness and scientificity, it is necessary to consider some constraints between supervisors and students when grouping. There are two inherent contradictions: the principles of mutual avoidance and uniformity. In this paper, the main issue is to find out an optimal solution that satisfies the two conditions as far as it is possible. Through the establishment of mathematical model, the respondent grouping problem is summarized as the matrix encoding. Then two conflict conditions are consolidated into one objective function. The single objective differential evolution with superior-inferior crossover scheme is adopted to solve this problem. A suitable chromosome representation and fitness function are designed. A series of operations such as mutation, superior-inferior crossover and modification are performed. The optimal solution is obtained when the evolution is terminated. To test the advantages of this method, a general algorithm is designed for comparison with it. The results show that the grouping solution obtained by differential evolution using a superior-inferior crossover scheme is more scientific and feasible than the general algorithm.

Key words: single-objective; differential evolution; defense grouping; matrix coding; mathematical model; fitness degree

毕业生答辩分组问题是高校教学管理中的一个常见问题, 为保证分组的公平性和科学性, 在分组时须考虑老师和学生间、老师和老师间, 以及学生之间的关

系, 同时须考虑每个组的若干限制条件^[1]. 此类答辩分组问题已经被证明是一个受限性 NP(Non-Deterministic Polynomial)难问题^[2].

① 基金项目: 河南省教育厅高校重点科学技术研究项目(15A520083, 16A520060, 17B520017); 河南省科技厅2014年基础与前沿技术研究项目(142300410391); 2015年河南中医学院博士基金(2015BSJJ-19); 2017年河南省科技攻关研究项目(172102210361, 172102310536)

收稿时间: 2016-12-26; 采用时间: 2017-01-23

该问题一般的描述是: 已知有 m 个老师, n 个学生, 其中每个老师指导若干名学生, 但每名学生只被一名老师指导. 要求分成 g 组答辩小组, 每组中由 m/g 个老师, 来面试答辩 n/g 学生, 假设 g 能够被 m 和 n 整除. 限制条件有:

(1) 每个老师不能答辩面试自己指导的学生, 自回避原则, 必须满足.

(2) 存在两个老师相互面试对方学生的情况, 要尽可能的少, 互回避原则. 该条件尽量满足.

(3) 每个老师指导的学生, 尽量均匀分散到不同的面试组中, 均匀原则. 该条件尽量满足.

条件 2 和 3 相互冲突, 所以要尽量满足, 可以看出该问题是一个较为复杂的组合优化问题. 针对此类问题, 用常规算法求解计算量较大且难以获取最优解^[2], 较适合用启发式进化算法来解决.

微分进化算法(Differential Evolution algorithm, DE)是一种模拟生物自然化过程搜索全局最优解的随机优化算法, 是近年来最著名的进化算法之一^[3,4]. 它具有简单, 快速, 鲁棒性好, 易于实现, 控制参数少且搜索能力强等特点, 因此被广泛应用于各个领域, 尤其是在处理复杂优化问题方面得到了广泛关注^[5,6].

DE 算法在求解连续变量的函数优化问题时能快速、稳定地收敛到全局最优解^[7,8]. 本文主要研究基于单目标的 DE 算法在答辩分组中的应用. 通过设计合适的染色体编码方式、适应度函数, 然后基于 DE 的进化策略进行初始化种群、变异、交叉、修正等操作, 对问题的解进行搜索. 最后给出结果并与常规的求解方法进行对比.

1 问题建模分析

1.1 模型假设

为简化模型, 对问题中的一些元素作以下约定:

(1) 所有参加答辩的学生, 按十进制序号升序排序, 学生个体在建模中不作区分, 认为完全一样.

(2) 所有老师按十进制序号升序排序, 不同老师个体之间需要区分.

(3) 分组结果中每组的先后顺序没有特别要求, 可以调换, 不影响整体的效果.

(4) 为了简便处理, 每位老师指导学生人数相同, 每一组分配老师学生人数分别相同.

同时约定以下符号:

m : 老师总人数.

n : 学生总人数.

p : 每位老师指导学生人数.

g : 分组总数.

s : 分组结果中每组参加答辩学生人数.

t : 分组结果中每组参加答辩老师人数.

A : 已知老师指导学生矩阵, $m*n$.

B : 分组矩阵, $m*n$.

C : 检验矩阵, $m*m$.

num : 检验矩阵 C 中不满足老师互回避情况的对数与零元素个数的总和.

为了简化处理, 其中人数均为整数, 而且有 $n=m*p$, $m=g*t$, $n=g*s$, 如 $m=6$, $n=12$, $p=2$, $g=3$, $s=4$, $t=2$ 情况, 则表示 6 位老师, 12 名学生, 每位老师指导 2 名学生, 共分为 3 组, 每组学生 4 人, 老师 2 人.

1.2 模型定义

定义 1. 老师指导学生矩阵 A

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix}$$

其中,

$$a_{ij} = \begin{cases} 1, & \text{第 } i \text{ 位老师指导第 } j \text{ 名} \\ 0, & \text{第 } i \text{ 位老师不指导第 } j \text{ 名} \end{cases}$$

$$i = 1, 2, \dots, m, j = 1, 2, \dots, n$$

定义 2. 分组矩阵 B

$$B = \begin{bmatrix} b_{11} & b_{12} & \cdots & b_{1n} \\ b_{21} & b_{22} & \cdots & b_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ b_{m1} & b_{m2} & \cdots & b_{mn} \end{bmatrix}$$

其中,

$$b_{ij} = \begin{cases} 1, & \text{第 } i \text{ 位老师答辩第 } j \text{ 名学生} \\ 0, & \text{第 } i \text{ 位老师不答辩第 } j \text{ 名学生} \end{cases}$$

$$i = 1, 2, \dots, m, j = 1, 2, \dots, n$$

定义 3. 检验矩阵 C

$$C = \begin{bmatrix} c_{11} & c_{12} & \cdots & c_{1m} \\ c_{21} & c_{22} & \cdots & c_{2m} \\ \cdots & \cdots & \cdots & \cdots \\ c_{m1} & c_{m2} & \cdots & c_{mm} \end{bmatrix}$$

其中, $c_{ij} = 0, i, j = 1, 2, \dots, m$,

且 $C = B^T * A$, 即 $c_{ij} = b_{i1}a_{j1} + b_{i2}a_{j2} + \cdots + b_{im}a_{jn}$.

1.3 模型定理

定理 1. 分组结果中每组的先后顺序不影响整体的效果.

证明: 假设分组中每组的先后顺序影响分组效果, 随意变换分组中的先后顺序, 每组参与老师学生不变, 通过计算变换顺序后的互回避指标, 发现任意变化分组的先后顺序, 互回避指标不变, 评价整体效果的互回避指标与分组结果中每组的先后顺序无关, 故与假设矛盾, 因此分组结果中每组的先后顺序不影响整体的效果, 证毕.

定理 2. 检验矩阵 C 可以用来评估分组结果的优劣.

证明: 由于老师指导学生矩阵 A 乘以分组矩阵 B 的转置矩阵是矩阵 C , 对于矩阵 $C(i, j)$, 就是老师 i 指导的学生接受老师 j 面试的人数. C 主对角线元素是 0, 说明分组结果必须满足老师自回避原则. 关于主对角线对称位置的两元素同时不为零说明是满足互回避原则的, 矩阵 C 中每列零元素越少说明学生分布的越均匀. num 是检验矩阵 C 中关于主对角线对称的两元素同时不为零的对数加上每列零元素的个数的总和, 可以反馈分组方案满足老师互回避原则的情况以及满足均匀原则的情况信息, 因此, 检验矩阵 C 可以评估分组结果的优劣, 证毕.

1.4 答辩分组问题数学模型

根据以上的研究分析, 建立答辩分组问题数学模型如下:

$$s.t. = \begin{cases} \sum_{j=1}^n a_{ij} = p, i = 1, 2, \dots, m \\ m * p = n \\ m = g * t \\ n = g * s \\ \sum_{j=1}^n b_{ij} = s, i = 1, 2, \dots, m \\ \sum_{i=1}^m b_{ij} = t, j = 1, 2, \dots, n \\ a_{ij} + b_{ij} \leq 1, i = 1, 2, \dots, m, j = 1, 2, \dots, n \\ c_{ij} = 0, i = j = 1, 2, \dots, m \\ B^T * A = C \\ num = \sum(c_{ij} * c_{ji} > 0) / 2 + \sum(c_{ij} = 0), \\ i = 1, 2, \dots, m, j = 1, 2, \dots, m \end{cases} \quad (1)$$

2 DE 算法解决答辩分组问题

2.1 种群初始化

针对答辩分组问题, 采用随机的 0/1 矩阵构造一个老师矩阵 A (如图 1)^[11,12]. 矩阵的行属性定义为老师, 列

属性定义为老师, 则矩阵的行数为 $m * n$, 列数为 m . 矩阵元素 0 代表老师不带某个学生, 1 表示老师带的学生, 并满足每个老师带 2 个学生, 每个学生只属于一个老师. 图 1 中的矩阵 A 表示老师 1 指导学生 2 和 6, 老师 2 指导学生 1 和 12, 老师 3 指导学生 3 和 7, 老师 4 指导学生 9 和 10, 老师 5 指导学生 4 和 8, 老师 6 指导学生 5 和 11.

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}$$

图 1 老师矩阵 A

然后设置种群规模 $N=10$, 来构造 10 个染色体矩阵, 即种群矩阵. 该矩阵用于表现答辩分组后的结果, 问题的最终目标是得到该矩阵的最优解. 种群矩阵的行列属性与矩阵 A 相同, 假设老师和学生数都可以被均匀的分到 g 个组中, 每个答辩老师答辩 n/g 个学生, 每个学生被 m/g 个老师答辩. 图 2 中的矩阵 B 表示老师 1 和 3 答辩学生 1, 4, 5, 8; 老师 2 和 5 答辩学生 3, 9, 10, 11; 老师 4 和 6 答辩学生 2, 6, 7, 12 (如图 2).

$$B = \begin{bmatrix} 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 \end{bmatrix}$$

图 2 分组矩阵 B

2.2 变异操作

变异操作是微分进化算法的核心. 当种群中个体的适应度相差不大时, 说明种群中的各个体基本上趋

于一致,因而可能导致进化停滞,过早地收敛于局部的极值解.为此必须通过变异操作来改变不利因素,使算法具有全局收敛性^[3].变异机制如下所示:

$$V_{i,G+1} = X_{r_1,G} + F(X_{r_2,G} - X_{r_3,G}) \quad (2)$$

对 G 代中的第 i 个个体进行变异,产生的中间个体为 $V_{i,G+1}$,其中 r_1, r_2 和 r_3 取自 $1 \sim NP$ 种群个数的随机数, $F[0, 1]$ 是一个常数,称为变异因子,它控制着偏差量 $(X_{r_2,G} - X_{r_3,G})$ 的放大作用^[9].对于答辩分组问题的随机变异方法是在矩阵中随机选择一行,将这行按反序插入原位置中,得到其变异产生的新的子代个体.

2.3 优劣个体的交叉策略

为增加干扰参数向量的多样性,经典 DE 中的交叉机制是:

$$u_{ji,G+1} = \begin{cases} v_{ji,G+1} & \text{if } (\text{randb}(i) < C_R) \\ x_{ji,G} & \text{else} \end{cases} \quad (3)$$

对于基因 j 随机生成一个 $0 \sim 1$ 的随机数,如果随机数小于 C_R 那么就将变异后的个体 v 基因 j 给交叉后的个体 u_i , 否则将变异前的 i 种群的 j 基因给变异后的种群个体 u_i , 交叉后的个体为 u .对于答辩分组问题采取行列交叉操作,即随机选取双亲中相对应的一行,将其互换得到两个新的子代个体^[4,5].

但是,在标准进化算法的进化后期,当大多数个体聚集在局部最优解时,种群多样性减少,种群不能通过变异和交叉操作产生新的更优个体.为增加种群多样性,加大探索空间,依据前期研究成果^[6,7],这里采用一种基于优劣个体交叉策略.该方案中优劣交叉具体是指,在目标向量(父代向量)和变异向量中调整出部分优秀和劣质个体之间完成,其中包括优劣交叉和优优交叉,若种群多样性指标非常小,即大部分个体过度聚集,采用优劣个体间交叉可提高种群多样性;否则,采用优优个体间交叉提高种群的探索能力.

用种群多样性指标 η 来判断采用哪种交叉策略.该指标受种群中个体之间的距离和适应度值的影响.为计算个体间的距离,首先介绍基因的多样性程度 $dx_{i,j}$.

$$dx_{i,j} = \begin{cases} 0, & \text{if } |(x_{ij} - x_{bj})| / R < \varepsilon_1 \\ 1, & \text{otherwise} \end{cases} \quad (4)$$

$i = 1, 2, \dots, NP, i \neq b, j = 1, 2, \dots, D$

每一代中, $dx_{i,j}$ 表示基因的多样性程度, $\varepsilon_1 \in [0, 1]$ 是每个基因的多样容忍度.在公式(4)中, $x_{i,j}$ 表示在第 i 代个体中的第 j 代基因, $x_{b,j}$ 表明使用参照第 b 个

个体来计算所有基因的多样性程度.因此,当前种群多样性程度 ξ 可以从多基因多样性总量来获得:

$$\xi = \frac{\sum_{i=1}^{NP} \sum_{j=1}^D dx_{ij}}{D \cdot (NP - 1)} \quad (5)$$

从个体之间距离的角度计算多样性,用每五个个体(种群大小为 100)中随机选择一个作为参照来计算种群多样性程度,然后使用 20 个值得到均值为最终的多样性程度值 ξ .从上面公式(5)中可以知道, ξ 的值在 $(0, 1)$ 之间.

另一方面,从适应度值 φ 来测量种群多样性:

$$\varphi = \frac{\sigma_f}{|f_{\text{worst}} - f_{\text{best}}|} \quad (6)$$

在公式(6)中用 φ 来表示多样性. σ_f 表示种群个体适应度值的标准偏差, f_{worst} 和 f_{best} 分别表示种群中最优的和最差的个体适应度值.显然, φ 的值也在 0 和 1 之间.

结合 ξ 和 φ 来计算种群多样性,并在公式(7)中使用 η 为最终参数表示.

$$\eta = 0.5 \times \xi + 0.5 \times \varphi \quad (7)$$

η 的值在 0 到 1 之间变化,根据 ξ 和 φ ,可判断应该使用哪种交叉策略.如果 $\eta < \varepsilon$,即当前种群多样性小于容忍度 ε ,执行优劣交叉以增加种群多样性.否则,当 $\eta \geq \varepsilon$,执行优优交叉来提高种群的探索性.基于我们前期的工作,仿真实验将 $\varepsilon = \varepsilon_1 = 0.05$ 作为经验值,更多关于优劣交叉相关参见文献^[6,8],在此不再赘述.

2.4 选择操作

选择的目的是要对个体进行适应度函数评价,以此来决定是否在下一代中用候选个体替换当前目标个体.其对应法则如下:

$$X_{i,G+1} = \begin{cases} U_{i,G+1} & \text{if } (f(U_{i,G+1}) < f(X_{i,G})) \\ X_{i,G} & \text{else} \end{cases} \quad (8)$$

适应度函数是选择下一代种群的一个重要依据,用来评估种群优劣的一个指标,有的适应度函数评价价值越大表示个体越优秀,有的则是越小越优秀,要根据具体情况而定,DE 算法主要以适应度函数来指导搜索策略.假设适应度最小的为最优解,那么当变异交叉后的 U 种群的个体适应度比初始种群的适应度小的时候,就将 U 种群选取为第 $G+1$ 代;反之,则将原来的第 G 代个体选取为第 $G+1$ 代^[9,10].

2.5 适应度计算

对于答辩分组问题,要求的最优解应当满足每个

老师的学生尽可能的分布在多个答辩老师中同时尽量避免两个老师相互面试对方学生的情况,使用检验矩阵 C (见公式 1)来反馈分组方案中满足老师互回避原则的情况信息(定理 2). 适应度计算依据定理 2 中的 num 的值.

用图 1 中的矩阵 A 和图 2 中的矩阵 B 代入定义 3 得到检验矩阵 C (如图 3). 从图 3 中可以看出, 矩阵 C 主对角线的元素全部为零, 说明满足自回避原则, 关于主对角线对称的两元素同时不为零的对数为 6, 矩阵中的非零元素个数为 18, 因此该分组方案的适应度 num 为 24.

$$C = \begin{bmatrix} 0 & 1 & 0 & 0 & 2 & 1 \\ 0 & 0 & 1 & 2 & 0 & 1 \\ 0 & 1 & 0 & 0 & 2 & 1 \\ 2 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 2 & 0 & 1 \\ 2 & 1 & 1 & 0 & 0 & 0 \end{bmatrix}$$

图 3 检验矩阵 C

2.6 修正操作

在对种群始化, 或者变异和交叉时, 可能会产生不合法的个体, 比如产生了与矩阵 A 相同行列位置同为 1 的情况(不满足自回避原则), 或者矩阵中某个数变成小数, 此时就要把不合法个体修正为合法的染色体.

修正过程既是对个体进行限制条件的验证过程, 使得个体中的值为 0 或 1, 而且行满足限制条件 $\sum_{j=1}^n b_{ij} = s, i = 1, 2, \dots, m$, 列满足限制条件 $\sum_{i=1}^m b_{ij} = t, j = 1, 2, \dots, n$.

2.7 总体流程

使用 DE 求解答辩分组问题步骤如下:

- (1) 生成初始种群, 种群中包含 10 个体, 进化代数 $G=0$.
- (2) 修正操作, 对随机生成的个体依据限制条件进行修正.
- (3) 开始进化, 对种群中每个个体进行交叉, 变异和选择操作, 得到新一代的个体.
- (4) 判断是否满足终止进化条件, 即是否进化迭代次数达到 50 代, 若不满足则进化到下一代, 继续执行(3)中的交叉变异等操作. 如果满足终止条件, 则停止搜索, 算法结束并输出结果.

图 4 是 DE 算法流程图, 图中 G 表示代数, NP 表示种群大小.

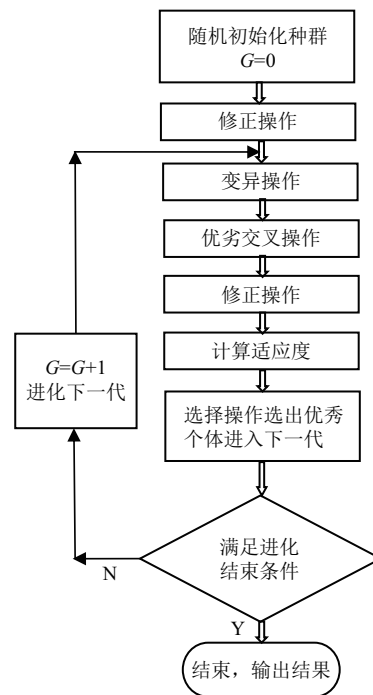


图 4 算法总体流程图

3 一般算法解答辩分组问题

为了与 DE 算法求解答辩分组问题做对比, 给出一般方法的求解过程. 一般算法即用常规的程序设计方法, 在设计过程中应考虑参数的限制, 死循环的处理等问题, 用数组和循环来求解该问题. 确定已知参数 m 个老师, n 个学生, 共分成 g 组, 每组老师个数为 m/g , 每组学生个数为 n/g , 根据已知老师进行答辩组分组.

(1) 选择老师

随机生成 $1 \sim m$ 个数的随机排列, 每次顺序取两个作为每答辩组的答辩老师, 这种老师与老师的组合是随机的, 共可以选择 g 次.

(2) 选择学生

逐一遍历每个老师, 每次从老师指导的学生中选一名学生进入答辩组, 直到选够 n/g 个为止, 这样才能近可能满足均匀原则. 迭代到某个老师时, 首先将该老师与本组中已分配的答辩老师进行对比, 如果没有相同结果, 再从该老师的学生中进行选择(遵循自回避原则).

(3) 死锁处理

迭代过程中, 可能会出现排除自答辩老师组学生, 其它学生已全被选取的情况, 此时会产生死锁, 出现这种情况需要重新进行分组. 设置运行次数 $Runtime$, 如

果死锁就重新开始分组,最多运行 10 次.算法流程如图 5 所示.

(4) 适应度

为了检验分配方案的均匀性和互回避性,需要计算分配结果的适应度,这里定义适应度由两个指标组成,一个是分配结果中每个答辩组里属于同一个老师的学生个数;另一个是老师互相答辩对方学生的对数,最终适应度等于这两个值之和.

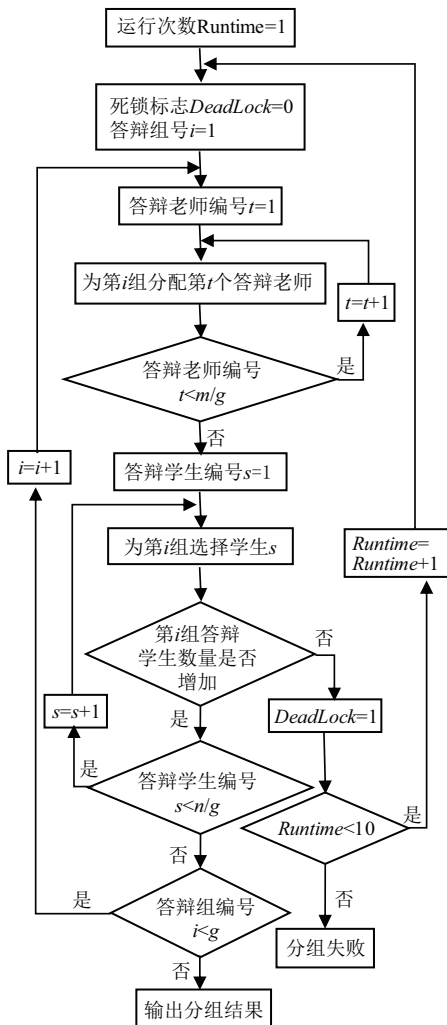


图 5 一般算法流程图

4 算法实验测试结果

对于以上两种算法,分别用两组数据测试算法的优劣.程序中用 T1, T2, ……Tm 来表示老师的编号,用 T1_1, T1_2, ……T1_n 来表示同属于 T1 老师指导的学生编号,依此类推, Tm_n 则表示由 Tm 老师指导的第 n 个学生.第一组数据: m=6, n=12, g=3, 对应的老

师指导学生信息如表 1.

第二组数据: m=8, n=32, g=4, 对应的老师指导学生信息如表 2.

表 1 6 名老师指导 12 名学生的信息

老师	指导学生
T1	T1_1, T1_2
T2	T2_1, T2_2
T3	T3_1, T3_2
T4	T4_1, T4_2
T5	T5_1, T5_2
T6	T6_1, T6_2

表 2 8 名老师指导 32 名学生的信息

老师	指导学生
T1	T1_1, T1_2, T1_3, T1_4
T2	T2_1, T2_2, T2_3, T2_4
T3	T3_1, T3_2, T3_3, T3_4
T4	T4_1, T4_2, T4_3, T4_4
T5	T5_1, T5_2, T5_3, T5_4
T6	T6_1, T6_2, T6_3, T6_4
T7	T7_1, T7_2, T7_3, T7_4
T8	T8_1, T8_2, T8_3, T8_4

4.1 DE 算法测试结果

用 DE 算法运行第一组数据,初始种群包含 10 个个体,进化代数为 50 代,运行结果显示了答辩分组矩阵 B 对应的 10 个个体,均为 12×6 的矩阵,由于篇幅所限,不再一一展开.运行结束后,种群中个体最小的适应度值是 20,图 6 给出了 DE 算法进化的曲线.在末代种群中,个体 1, 5, 8 的适应度均为 20,即是理想的答辩分组结果,表 3,表 4,表 5 分别给出了他们的分配方案.

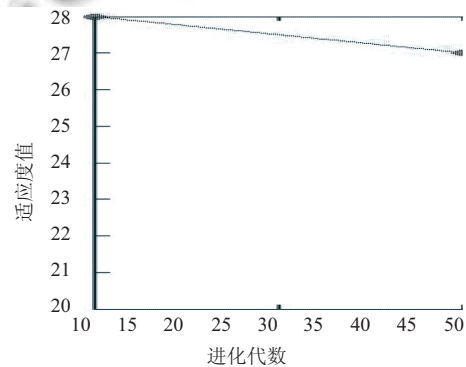


图 6 DE 运行第 1 组数据进化曲线

表 3 个体 1 对应的分组方案

答辩组	答辩老师	答辩学生
1	T1, T4	T2_1, T3_1, T6_1, T6_2
2	T2, T3	T1_1, T5_1, T1_2, T5_2
3	T5, T6	T3_2, T4_1, T4_2, T2_2

表4 个体5对应的分组方案

答辩组	答辩老师	答辩学生
1	T1, T4	T2_1, T3_2, T6_2, T2_2
2	T2, T5	T1_1, T3_1, T6_1, T1_2
3	T3, T6	T5_1, T5_2, T4_1, T4_2

表5 个体8对应的分组方案

答辩组	答辩老师	答辩学生
1	T1, T5	T2_1, T2_2, T4_1, T4_2
2	T2, T6	T1_2, T3_1, T3_2, T5_1
3	T3, T4	T1_1, T5_2, T6_1, T6_2

从上述三个分组方案中可以看出, DE 算法得出的解决方案呈现了多样化的特征.

运行第二组数据, 仍然设定初始种群包含 10 个个体, 进化 50 代, 输出 10 个 32×8 的矩阵, 以此表示答辩分组结果, 同时输出这 10 个分组方案对应的适应度. 进化曲线如图 7 所示, 图中显示种群中适应度最小为 28, 也就是最理想的答辩分组结果. 运行结束后, 种群中个体 1、2 的适应度都是 28, 即他们两种分配方案有同样的适应度, 都可以供使用者选择.

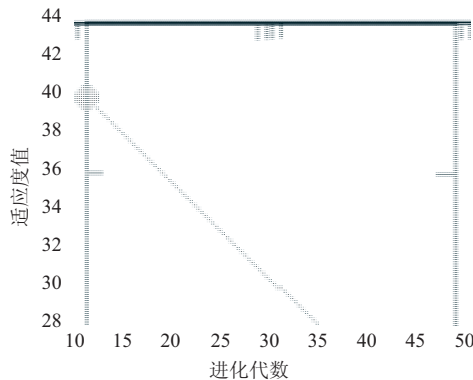


图7 DE 运行第 2 组数据进化曲线

个体 1 和 2 对应的分组方案如表 6 和表 7 所示.

表6 个体 1 对应的分组方案

答辩组	答辩老师	答辩学生
1	T1, T8	T2_3, T3_2, T3_4, T4_2, T4_4, T7_2, T7_3, T7_4,
2	T2, T5	T1_3, T1_4, T4_1, T6_2, T6_4, T7_1, T8_1, T8_3
3	T3, T4	T1_1, T2_4, T5_4, T5_2, T6_1, T6_3, T8_2, T8_4
4	T6, T7	T1_2, T2_1, T2_2, T3_1, T3_3, T4_3, T5_1, T5_3

表7 个体 2 对应的分组方案

答辩组	答辩老师	答辩学生
1	T1, T4	T5_2, T5_4, T6_2, T6_3, T6_4, T7_1, T8_2, T8_4
2	T2, T8	T1_1, T3_2, T3_4, T4_4, T5_3, T6_1, T7_3, T7_4
3	T3, T6	T1_3, T2_2, T4_1, T4_3, T5_1, T7_2, T8_1, T8_3
4	T5, T7	T1_2, T1_4, T2_1, T2_3, T2_4, T3_1, T3_3, T4_2

4.2 一般算法测试结果

一般算法运行一次只能得到一个解, 运行第 1 组数据结果如表 8 所示, 这个结果的适应度为 12.

表8 一般算法运行第 1 组数据的结果

答辩组	答辩老师	答辩学生
1	T1, T2	T3_1, T4_1, T5_1, T6_1
2	T3, T5	T1_1, T2_1, T4_2, T6_2
3	T6, T4	T1_2, T2_2, T3_2, T5_2

用一般算法运行第 2 组数据, 结果如表 9 所示, 适应度为 24.

表9 一般算法运行第 2 组数据的结果

答辩组	答辩老师	答辩学生
1	T6, T5	T1_1, T2_1, T3_1, T4_1, T7_1, T8_1, T1_2, T2_2
2	T8, T1	T2_3, T3_2, T4_2, T5_1, T6_1, T7_2, T2_4, T3_3
3	T2, T7	T1_3, T3_4, T4_3, T5_2, T6_2, T8_2, T1_4, T4_4
4	T3, T4	T5_3, T6_3, T7_3, T8_3, T5_4, T6_4, T7_4, T8_4

一般算法在运行数据时, 有时会发生死锁的情况, 如表 10 所示, 在完成 3 组分配后, 进行第 4 组分配时遇到了排除自答辩老师没有学生可分配的情况. 此时, 需要重新运行程序进行分组.

表10 一般算法运行第 2 组数据时遇到死锁的情况

答辩组	答辩老师	答辩学生
1	T6, T4	T1_1, T2_1, T3_1, T5_1, T7_1, T8_1, T1_2, T2_2
2	T7, T1	T2_3, T3_2, T4_1, T5_2, T6_1, T8_2, T2_4, T3_3
3	T2, T3	T1_3, T4_2, T5_3, T6_2, T7_2, T8_3, T1_4, T4_3
4	T5, T8	排除自答辩老师没有学生可分配

根据实验结果, 死锁发生的概率与测试数据的规模成正比, 测试第 1 组数据($m=6, n=12, g=3$)时, 程序运行 30 次, 发生死锁的次数为 0. 测试第 2 组数据($m=8, n=32, g=4$)时, 程序运行 10 次, 发生了 9 次死锁, 死锁概率为 90%; 运行 20 次时, 发生了 15 次死锁, 死锁概率为 75%; 运行 30 次, 发生了 20 次死锁, 死锁概率为 67%. 由此可以推断, 随着一般算法运行次数增多, 死锁的概率会逐渐下降, 但是想得到多个分组方案的话, 程序至少要运行十次以上. 由此可见, 一般算法在问题规模较小的情况下输出可行解是快速、高效的, 但输出的解不具有进化过程. 当问题规模稍大时, 输出无效解的概率过大, 算法效率明显降低.

5 结果分析

从测试结果来看, 两种算法都能解决答辩分组问

题,并且都能满足自回避原则,对于互回避原则和均匀原则都能尽量满足.但一般算法运行一次只能得到一个解,只有运行多次才能得到多个解,且这些解不具有进化的过程,不能从所有可能的解当中寻得最优值,在实际应用中难以为决策者提供更全面可行的方案.而DE算法运行一次能得到多个最优解,且这些解是经过 N 代进化选择的结果.因此,单目标微分进化算法所具备的进化过程使最优解能从众多的备选方案中快速显现,表现出较好的优越性.

由于篇幅所限,只用了两组数据测试,当每增加一位老师时会增加更多的学生,如果学生足够多就能完全满足自回避和互回避条件,但 m, n 具体增加到多少才能满足条件,还留待以后进行深入探究.

参考文献

- 1 李剑,朱延峰,吴畏.学生面试问题的分配策略.数学的实践与认识,2007,37(14):153-160. [doi: 10.3969/j.issn.1000-0984.2007.14.020]
- 2 司守奎,孙玺菁.数学建模算法与应用.北京:国防工业出版社,2012.
- 3 Das S, Suganthan PN. Differential evolution: A survey of the state-of-the-art. IEEE Trans. Evolutionary Computation, 2011, 15(1): 4-31. [doi: 10.1109/TEVC.2010.2059031]
- 4 许玉龙,方建安,王晓鹏,等.基于非支配解排序的快速多目标微分进化算法.计算机应用,2014,34(9):2547-2551,2561. [doi: 10.11772/j.issn.1001-9081.2014.09.2547]
- 5 黄仁全,靳聪,贺筱军,等.自适应局部增强微分进化改进算法.空军工程大学学报(自然科学版),2011,12(3):84-89.
- 6 Xu YL, Fang JA, Zhu W, et al. Differential evolution using a superior-inferior crossover scheme. Computational Optimization and Applications, 2015, 61(1): 243-274. [doi: 10.1007/s10589-014-9701-9]
- 7 许玉龙,方建安,赵灵东,等.微分进化求解无线传感器网络中的覆盖问题.计算机工程与设计,2014,35(9):3007-3013.
- 8 Cheng SL, Hwang C. Optimal approximation of linear systems by a differential evolution algorithm. IEEE Trans. on Systems, Man, and Cybernetics-Part A: Systems and Humans, 2010, 31(6): 698-707.
- 9 Plagianakos VP, Vrahatis MN. Parallel evolutionary training algorithms for "hardware-friendly" neural networks. Natural Computing, 2011, 1(2): 307-322.
- 10 Gamperle R, Dmuller S, Koumoutsakos P. A parameter study for differential evolution. International Conference on Advances in Intelligent Systems, Fuzzy Systems, Evolutionary Computation. Interlaken, Switzerland. 2002. 293-298.
- 11 刘靖洁,陈桂明,刘小方.基于矩阵编码的遗传算法研究.计算机工程,2011,37(13):160-162. [doi: 10.3969/j.issn.1000-3428.2011.13.051]
- 12 战红,杨建军.基于工序矩阵编码遗传算法的车间作业调度优化.制造业自动化,2013,35(7):86-88.