

基于量子进化的给定围长图构造算法

冯晓华, 孙永奇

(北京交通大学 计算机与信息技术学院 交通数据分析与挖掘北京市重点实验室, 北京 100044)

摘 要: 构造给定围长的极图是图论难题之一, 特别是在顶点数规模不断增大的情况下会出现组合爆炸的现象。针对该问题, 提出一种构造给定围长图的算法, 通过在生成个体、调整个体中充分利用极图的特性, 使算法达到较高的收敛速度与收敛精度。实验结果表明, 通过构造围长为 10 的图, 与粒子群优化算法、遗传算法相比, 该算法达到次优解和最优解的准确率最高, 构造围长为 11 的图, 可得到相应的极图边数的下界。

关键词: 进化算法; 量子进化算法; 极图; 围长; 圈

中文引用格式: 冯晓华, 孙永奇. 基于量子进化的给定围长图构造算法[J]. 计算机工程, 2017, 43(10): 141-146.

英文引用格式: FENG Xiaohua, SUN Yongqi. Construction Algorithm of Given Girth Graphs Based on Quantum Evolution[J]. Computer Engineering, 2017, 43(10): 141-146.

Construction Algorithm of Given Girth Graphs Based on Quantum Evolution

FENG Xiaohua, SUN Yongqi

(Beijing Key Lab of Traffic Data Analysis and Mining, School of Computer and Information Technology,
Beijing Jiaotong University, Beijing 100044, China)

[Abstract] To construct an extremal graph with a given girth is still a challenging problem of graph theory. Especially when the vertex number increases combination explosion will appear. Thus, this paper proposes an algorithm for constructing graphs with given girth based on quantum evolution. By making full use of the characteristics of the extremal graphs in producing and adjusting the individuals, it makes the algorithm achieve higher convergence speed and precision. Experimental results show that, by constructing the graphs with the girth of 10, the proposed algorithm is compared with the other algorithms, namely Particle Swarm Optimization (PSO) algorithm and Genetic Algorithm (GA), the algorithm can achieve the optimal solutions and near optimal solutions with the highest accuracy. It uses the proposed algorithm to construct the graphs with the girth of 11, and gives lower bounds of the sizes of extremal graphs with the girth 11.

[Key words] evolutionary algorithm; Quantum Evolutionary Algorithm (QEA); extremal graph; girth; cycle

DOI: 10.3969/j.issn.1000-3428.2017.10.024

0 概述

智能优化算法主要包括量子进化算法 (Quantum-inspired Evolutionary Algorithm, QEA)、粒子群优化算法 (Particle Swarm Optimization, PSO)、遗传算法 (Genetic Algorithm, GA) 等。人们对量子计算的研究可以追溯到上世纪中叶, 但直到 20 世纪 90 年代中期才开始得到研究者的广泛关注。文献[1]基于量子计算给出了在多项式时间内寻找离散对数和整数分解的方法, 并将该量子因子分解算法应用于 NP 问题的求解, 但该算法的鲁棒性有待提高。文献[2]根据量子计算多状态叠加的特点提出了量子搜索算法, 并将其应用于由 N 个名字随机排列的电话号码簿搜索问题上, 可以使搜索次数由 $(N/2)$ 减少到 \sqrt{N} 。文献[3]

将量子计算与进化算法相结合, 提出了量子衍生遗传算法 (QGA), 并将其用于求解旅行商问题, 实验结果表明, QGA 的效果明显优于 GA。但是由于 QGA 采用的是传统的编码和种群更新方式, 因此存在种群中染色体状态单一和进化速率慢的问题。文献[4]提出了量子进化算法 (QEA), 采用基于量子比特进行编码, 并应用量子旋转门对种群进行更新。因为 QEA 采用了一种新颖的编码和种群更新方式, 所以具有种群规模小、迭代次数少、收敛速度快、搜索效率高和全局搜索能力强等优点, 但是该算法局限于应用在单目标优化问题上。

鉴于量子进化算法的优点, 国内外许多研究者还对其算法性能、框架和应用等方面进行了深入研究, 特别是在组合优化、基准函数优化、信号处理等

基金项目: 国家自然科学基金 (61572005, 61272004)。

作者简介: 冯晓华 (1991—), 女, 硕士研究生, 主研方向为智能计算、高性能计算; 孙永奇 (通信作者), 教授、博士生导师。

收稿日期: 2016-08-29 **修回日期:** 2016-10-24 **E-mail:** yqsun@bjtu.edu.cn

方面得到了广泛应用,已经成为计算智能领域的研究热点之一^[5-6]。但是,量子进化算法本身仍然有不成熟的地方,在理论研究、工程应用、多目标约束函数优化和算法融合等方面尚有待推广、深入和拓展^[7]。基于此,本文提出一种基于量子进化的构造给定围长图的算法。

1 极图问题与量子进化算法

1.1 极图问题

一个简单图 G 可被定义为一个二元组 $(V(G), E(G))$, 其中, $V(G)$ 和 $E(G)$ 分别表示其顶点集和边集。对于顶点 $v \in V(G)$, 与其关联的边数称为度, 用 $d(v)$ 表示。在图中如果 2 个顶点 v, w 之间存在一条无向边则用 (v, w) 或 vw 表示。用 C_k 表示长度为 k ($k \geq 3$) 的圈, 在图 G 中长度最小的圈称为图的围长, 用 g 表示。若一个图 H 满足 $V(H) \subseteq V(G)$ 和 $E(H) \subseteq E(G)$, 则称 H 为 G 的子图。

对于给定图集合 $\psi = \{H_1, H_2, \dots, H_m\}$, n 个顶点不包含任意 $H_i \in \psi$ ($1 \leq i \leq m$) 的边数最多的图称为不含 ψ 的极图。用 $EX(n; \psi)$ 表示由 n 个顶点不含 ψ 的所有极图组成的集合, 用 $ex(n; \psi)$ 表示这些极图的边数。在 ψ 为圈集即 $\psi = \{C_3, C_4, \dots, C_k\}$ 的极图研究方面, 已经取得了很多的成果。文献[8]给出了 $n \leq 31$ 时的极图集合 $EX(n; \{C_4\})$ 及相应 $ex(n; \{C_4\})$ 的值。文献[9]将该研究推进了一步, 证明了 $ex(32; \{C_4\}) = 92$ 。文献[10]给出了顶点数 $n \leq 24$ 的极图集合 $EX(n; \{C_3, C_4\})$ 以及 $ex(n; \{C_3, C_4\})$ 的值。文献[11]给出 $n \leq 21$ 时 $ex(n; \{C_4, C_5\})$ 的值。由围长的定义可知, 不含圈集 $\{C_3, C_4, \dots, C_k\}$ 的极图的围长至少为 $k+1$ 。为简便起见, 用 $EX(n; g)$ 和 $ex(n; g)$ 分别表示 $EX(n; \{C_3, C_4, \dots, C_k\})$ 和 $ex(n; \{C_3, C_4, \dots, C_k\})$, 其中 $g = k+1$ 。文献[12]将混合模拟退火算法和遗传算法应用到极图问题的研究中, 利用这些算法得到了围长为 6, 7 和 8 的一些极图。文献[13]则研究了当 n 和 g 满足一定关系时, $ex(n; g)$ 的变化规律。

本文主要研究围长至少为 10 或 11 的极图问题, 即确定 $ex(n; 10)$ 和 $ex(n; 11)$ 的值以及构造相应的图集合。当 n 较小时可以通过穷举的方式逐步构造极图, 但是随着顶点数的不断增加, 求解问题的计算量将以指数级增长, 使用穷举的方法去构造极图是非常困难的。因此, 本文将研究量子进化算法在构造给定围长的极图问题中的应用, 首先建立极图问题的优化模型, 再设计相应的量子进化算法, 并通过构造围长为 10 的图优化量子进化算法的参数, 最后利用该算法构造围长为 11 的图。

1.2 量子进化算法

量子进化算法是以量子的矢量态为基础, 将量子的叠加状态应用于染色体的编码, 使得一条染色

体可以叠加多个状态, 并且利用量子旋转门进行量子的更新, 从而达到种群进化的目的。与传统的优化算法的编码方式不同, 量子进化算法采用一种基于量子比特(Q-bit)的编码方式, 一个量子比特可以是 0 状态也可以是 1 状态, 其表示形式为:

$$|\varphi\rangle = \alpha|0\rangle + \beta|1\rangle$$

其中, $|\alpha|^2$ 和 $|\beta|^2$ 是对应状态概率幅度的一对复数, $|\alpha|^2$ 表示测量量子比特时得到 0 的概率, $|\beta|^2$ 表示测量量子比特时得到 1 的概率, α 和 β 满足 $|\alpha|^2 + |\beta|^2 = 1$ 。

在量子进化算法中, 一个规模为 n 的种群进化到 t 代时定义为:

$$Q(t) = q'_1, q'_2, \dots, q'_n$$

其中, $q'_i = \begin{bmatrix} \alpha'_i & \alpha'_2 & \dots & \alpha'_m \\ \beta'_1 & \beta'_2 & \dots & \beta'_m \end{bmatrix}$, $i = 1, 2, \dots, n$ 。量子进化算法主要包括种群初始化、生成个体、评估和更新 4 个阶段, 其流程如图 1 所示。

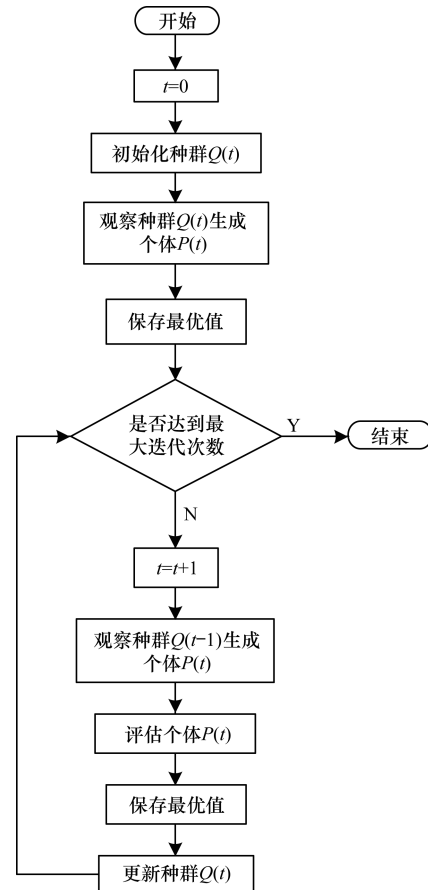


图 1 量子进化算法流程

在初始化阶段, 首先要初始化种群 $Q(t)$, 种群 $Q(t)$ 中的每个个体的 (α, β) 位初始化为 $1/\sqrt{2}$, 即 $\alpha'_i = \beta'_i = 1/\sqrt{2}$ ($i = 1, 2, \dots, m$)。然后通过观察 $Q(t)$ 的状态, 采用二进制编码方式生成个体 $P(t)$, 其中在第 0 代时 $P(0) = \{x_1^0, x_2^0, \dots, x_m^0\}$, 是一个长度为 m 的二进制字符串, 简记为 X_i^0 ($i = 1, 2, \dots, m$)。对于

X_i^0 中的每一位 x_i 取值由 $|\alpha_i^0|^2$ 决定。

在以 t 为参数的循环体内,包括生成个体、评估和种群更新 3 个子过程:

1) 生成个体 $P(t)$ 。由于染色体 $P(t)$ 中的每一位 x_i^t 是由种群 $Q(t-1)$ 中的 q_i^{t-1} 得到的,因此可以通过观察 $Q(t-1)$ 的状态,采用二进制编码方式生成个体 $P(t)$ 。

2) 评估个体 $P(t)$ 。通过适应度函数评估每个个体 $P(t)$,根据评估结果找到解最优的个体。如果这个最优解优于已保存的最优解,则保存该最优解和相应的个体。

3) 更新种群 $Q(t)$ 。利用量子门(Q-gates)更新种群 $Q(t)$ 中每个个体的量子比特。

当迭代进化次数 t 未达到设定的最大迭代进化次数时, $t = t + 1$,继续循环体内的操作;否则,算法结束,得到最优解。

2 基于量子进化的极图构造算法

2.1 极图问题规划模型

根据 1.2 节中对量子进化算法的描述可知,在进化过程中, $[\alpha, \beta]$ 角控制着算法的进化方向。由于量子进化算法通过适应度函数控制 $[\alpha, \beta]$ 角的更新方向,因此需要根据实际问题设计适应度函数。所以,首先需要对极图问题进行建模,得到相应的数学模型和适应度函数。用 G_n 表示 n 个顶点且围长满足要求的图,根据对极图的描述可知,极图问题可以转化为 0-1 规划问题,其规划模型可描述为:

$$\begin{aligned} \max f(X) &= \sum_{i=1}^m x_i \\ \text{s. t. } &\begin{cases} e_i \in E(G_n), x_i = 1 \\ C_j \notin G_n, j \in \{3, 4, \dots, g-1\} \\ x_i \in \{0, 1\} \end{cases} \end{aligned}$$

其中, $m = n(n-1)/2$; x_i 与完全图 K_n 的边 e_i ($i = 1, 2, \dots, m$) 一一对应。

2.2 极图构造算法

由 2.1 节的规划模型可知,求解极图问题的实质是在限定条件下最大化适应度函数 $f(X)$,属于函数优化问题。利用量子进化算法求解该类问题的关键是找到恰当的编码方式,把图的邻接矩阵的上三角按行优先转换成一个一维向量,该一维向量对应一条染色体,即每一条边作为染色体基因的一个因子。如果一条边被选中,则将对应该位置为 1,否则为 0。在此基础上就可以进行量子进化算法的操作,而在每次进化时仅保留围长符合要求的图。

本节给出一种基于量子进化的构造给定围长的图的算法 ConstructEG。该算法的输入为要构造的图的顶点数 n 、围长 g 、最大进化代数 M 和种群规模 K ,输出为满足围长要求的图,它们的边数等于或接近最优解 $ex(n; g)$ 。可以看出,调整个体

$P(t)$ 的时间复杂度最高,为 $O(Kn^2)$,主要是选择不符合围长要求的个体,并进行删边操作。因此,算法 ConstructEG 的时间复杂度为 $O(MKn^2)$ 。算法的各个子过程的详细描述为:

```
ConstructEG(n, g, M, K)
{
    t ← 0 // t 为进化代数
    初始化种群 Q(t)
    生成个体 P(t)
    调整个体 P(t)
    评估个体 P(t)
    保存最优解和相应的个体
    while(t < M) do
        Begin
            通过 Q(t-1) 生成个体 P(t)
            对个体 P(t) 进行交叉变异操作
            调整个体 P(t)
            评估个体 P(t)
            扰动种群 Q(t)
            更新种群 Q(t)
            保存最优解和相应个体
            t ← t + 1
        End
    }
```

1) 生成个体 $P(t)$ 。量子进化算法是基于比特进行编码,而进化算法的操作对象为 0,1 字符串,因此要将比特编码映射为二进制编码。根据极图的结构特点,可以令度数小的顶点优先邻接。生成过程为:如果 x_i 对应的边的 2 个顶点的度均小于 2,则 x_i 置为 1;否则采用随机函数产生一个 $[0, 1)$ 之间的随机数 r ,如果 $r < |\beta_i|^2$,则 x_i 置为 1;否则置为 0。

2) 交叉和变异操作。在调整个体 $P(t)$ 之前,对其进行交叉和变异。交叉的目的是为了更好地模拟生物的进化过程,保证种群中的个体之间有进化信息的交流。具体方法为:从种群中随机选取 2 个个体 i 和 j ,然后对于个体 i 使用随机函数生成范围在 $[0, 1)$ 的随机数 r ,根据式(1)计算交叉概率 pc ,其中, f_{\max} 为种群中最大适应度值; f_{avg} 为种群平均适应度值; f 为 2 个交叉个体适应度较高者; pc_1 的取值为 0.9; pc_2 的取值为 0.6。

$$pc = \begin{cases} pc_1 \frac{(f_{\max} - f)}{f_{\max} - f_{\text{avg}}}, & f \geq f_{\text{avg}} \\ pc_2, & f < f_{\text{avg}} \end{cases} \quad (1)$$

如果有 $r < pc$,则使用随机函数产生交叉点,在交叉点之后对个体 i 和 j 的量子状态进行交换,实现交叉操作。

变异的目的是使算法具有局部搜索能力,通过非门设计量子变异操作。具体方法为:对种群中的个体 i 先通过随机函数产生 $[0, 1)$ 之内的随机数 r ,然后根据式(2)计算其变异概率 pm ,其中, f_{\max} 为种群中最大适应度值; f_{avg} 为种群平均适应度值; f 为变异个体 i 的适应度值; pm_1 的取值为 0.1; pm_2 的取值

为 0.001。

$$pm = \begin{cases} pm_1 \frac{(f_{\max} - f)}{f_{\max} - f_{\text{avg}}}, & f \geq f_{\text{avg}} \\ pm_2, & f < f_{\text{avg}} \end{cases} \quad (2)$$

如果有 $r < pm$, 则再用随机函数生成变异点, 对该点的量子状态进行非门操作, 实现变异操作。

3) 调整个体 $P(t)$ 。在生成个体 $P(t)$ 时, 可能产生不满足围长要求的个体, 这时在个体调整时要通过删边使其满足要求。文献[14]证明了极图的一些性质, 表明极图的结构近似于正则图, 即在极图中顶点度的分布是比较均匀的。根据这一性质, 在删除禁止子图 C_k ($3 \leq k < g$) 的边时, 优先选择度数较大的顶点关联的边进行删除, 从而保证图中各顶点度分布均匀。由于要不断循环查找禁止子图, 并删除相应的边, 直到图的围长符合要求, 因此删边操作的时间复杂度为 $O(n^2)$ 。

4) 评估个体 $P(t)$ 。在算法 ConstructEG 中, 个体 (即满足围长要求的图) 的边数越多, 其适应度就越高。

5) 扰动种群 $Q(t)$ 。扰动的作用是防止出现早熟收敛现象。在进化停滞的时候对算法中种群个体的量子状态和旋转角进行一个扰动, 打破当前的进化方向, 使算法跳出局部最优解。具体做法为: (1) 设定一个适当的进化代数阈值 $tvalue$; (2) 在迭代进化的过程中, 如果算法在 $tvalue$ 代内进化停滞, 则改变 $\Delta\theta$ 的大小, 并且初始化种群个体的量子状态; (3) 算法根据新的量子状态继续进行迭代进化。

6) 更新种群 $Q(t)$ 。在 QEA 的进化过程中, $P(t+1)$ 的生成不是由 $P(t)$ 决定, 而是由 $P(t)$ 中最优的个体和 $Q(t)$ 的状态决定。通过量子门作用于种群 $Q(t)$ 的量子叠加态, 使各个量子的状态相互干扰, 从而更新 $Q(t)$ 的状态。因此, 量子门的设计直接影响算法的性能, 本文采用的量子旋转门为:

$$U(\theta_i) = \begin{bmatrix} \cos \theta_i & -\sin \theta_i \\ \sin \theta_i & \cos \theta_i \end{bmatrix}$$

其中, θ_i ($i=1, 2, \dots, m$) 是量子比特向 0 或者 1 状态改变的旋转角度。

7) 保存最优解和相应个体。由已经得到的极图结果可知, 极图的结构比较对称。因此, 在每一步保存完最优解后, 挑选对称性好的图作为输入图, 进行下一代进化。保留对称图实际上是对种群的一个“优胜劣汰”的操作, 将进化较好的个体保留到下一代的进化过程中, 淘汰掉进化不好的个体。

3 实验结果与分析

首先通过对围长为 10 的图的构造, 对算法 ConstructEG 进行参数优化; 然后比较遗传算法、粒子群进化算法和算法 ConstructEG 在构造围长为 10 的图的效果; 最后使用算法 ConstructEG 构造围长为 11 的图, 从而给出 $ex(n; 11)$ 的下界。在本节的实验

中, 硬件平台为 Intel i3-2120 3.3 GHz CPU, 4 GB 内存。软件环境为 Windows 操作系统, C 编程语言。

3.1 参数优化实验

3.1.1 初始图的选择

进化算法在进化的过程中, 算法的收敛速度和精度不仅受到进化策略的影响, 同时也会受到初始解的影响。本节以对构造围长为 10、顶点数为 n ($44 \leq n \leq 48$) 的图为例, 分别选用圈、树形图和随机图这 3 种结构作为算法 ConstructEG 的初始解, 对其运行结果进行评价。

当圈作为初始图时, 对于每个 n ($44 \leq n \leq 48$), 分别采用图 $(n-k)K_1 \cup C_k$ ($11 \leq k \leq n$) 为初始解构造围长为 10 的图, 其中, $(n-k)K_1$ 为 $(n-k)$ 个孤立点。当树形图作为初始图时, 选择采用图 $(n-28)K_1 \cup T_{28}$ ($44 \leq n \leq 48$) 为初始解, 其中, T_{28} 为如图 2 所示的树形图。当随机图作为初始图时, 每次使用随机图生成器产生 n ($44 \leq n \leq 48$) 个顶点的图作为初始解。对于每种初始解, 分别运行算法 40 次, 取最好的结果进行统计, 如表 1 所示。由实验结果可知, 当选用树形图作为算法的初始解时, 构造围长为 10 的图的效果最好。因此, 在算法 ConstructEG 中将采用树形图作为初始解。

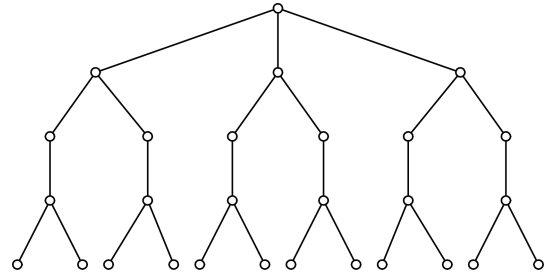


图 2 树形图 T_{28}

表 1 不同初始解的构造 G_n

n	随机图	圈	树形图
44	55	55	57
45	56	56	59
46	57	57	61
47	57	58	62
48	59	60	64

3.1.2 旋转角步长的调整

在进化过程中, 旋转角步长影响着算法的进化方向和收敛速度, 选择合适的旋转角步长策略可以提高算法的效率。本节将针对固定旋转角步长、自适应旋转角步长和定量调整旋转角步长 3 种不同的旋转角步长策略进行对比实验。

1) 固定旋转角步长。在更新种群 $Q(t)$ 时, 采用量子门 $U(\Delta\theta_i)$ 对量子比特进行更新, $\Delta\theta_i$ 选用统一的旋转角步长, 即当 (α_i, β_i) 位于第一、第三象限时, $\Delta\theta_i$ 取值为 0.001π ; 当 (α_i, β_i) 位于第二、第四象限时, $\Delta\theta_i$ 取值为 -0.001π 。

2) 自适应旋转角步长。在自适应旋转角步长策略中, $\Delta\theta_i$ 的取值根据种群的进化在 $[0.001\pi, 0.05\pi]$ 之间线性调整, 当 (α_i, β_i) 位于第一、第三象限时, $\Delta\theta_i$ 取值为 θ^j ; 当 (α_i, β_i) 位于第二、第四象限时, $\Delta\theta_i$ 取值为 $-\theta^j$ 。 θ^j 的计算公式为:

$$\theta^j = \begin{cases} \frac{(k_2 - k_1)(f_i - f_{\max})}{f_{\max} - f_{\min}}, & f_{\max} \neq f_{\min} \\ k_1, & f_{\max} = f_{\min} \end{cases}$$

其中, k_1 取值为 0.001π ; k_2 取值为 0.05π ; f_{\max} 为种群中最优的适应度值; f_{\min} 为种群中最差的适应度值; f_i 为当前个体的适应度值。

3) 定量调整旋转角步长。结合了固定角步长策略和自适应角步长策略的思想, 在进化的过程中采用自适应角步长调整公式得到 $\Delta\theta_i$ 的值; 当扰动种群 $Q(t)$ 时, 则采用固定角步长策略得到 $\Delta\theta_i$ 的值, 并且经多次实验表明, 这时选取固定角步长为 0.031π , 算法的效果最好。

针对每一种旋转角步长策略, 分别进行 40 次构造顶点数为 n ($44 \leq n \leq 48$), 围长为 10 的图的实验, 取最好的结果进行统计, 如表 2 所示。由实验结果可知, 在更新种群 $Q(t)$ 时, 选用定量调整旋转角步长策略时构造图的效果最好。因此, 在进行更新操作时, 将选取定量调整旋转角步长的策略。

表 2 不同旋转角步长的构造 G_n

n	固定旋转 脚步长	自适应旋转 脚步长	定量调整 旋转脚步长
44	55	55	57
45	58	57	59
46	59	58	61
47	60	60	62
48	61	62	64

3.2 3 种优化算法的对比

本节以构造围长为 10、顶点数为 48 的图为例, 对比遗传算法^[15]、粒子群优化算法^[16]和 ConstructEG 这 3 种优化算法的实验结果。由 3.1 节参数优化实验可知, 在利用算法 ConstructEG 构造围长为 10 的图时, 初始解选取树形图。种群个数均设置为 1 000 个, 进化代数设置为 1 500 代。

由于优化算法是一种基于概率的搜索算法, 因此每种算法分别运行 40 次, 对 3 种算法求得最优解和次优解的准确率进行统计, 如图 3 所示。从图 3 可以看出, GA 算法的求解效果最差, 在 40 次构造图的实验中达到最优解的准确率仅为 2.5%, 而 PSO 和 ConstructEG 达到最优解的准确率分别为 20.0% 和 42.5%。由实验结果可以看出, 虽然 3 种优化算法经过迭代进化都可以达到次优解和最优解, 但是在求解问题的准确率方面, 算法 ConstructEG 要明显优于 PSO 和 GA 算法。

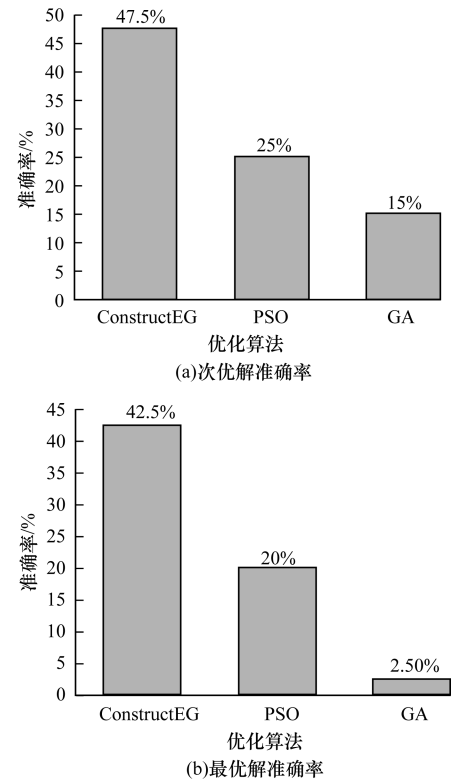


图 3 3 种算法构造图 G_{48} 的准确率

3.3 $ex(n; 11)$ 的下界

首先分析算法 ConstructEG 的收敛速度。先利用与文献[17]中类似的穷举算法(以下简称为算法 A)构造围长为 11、顶点数为 n ($11 \leq n \leq 30$) 的图, 从而得到 $ex(n; 11)$ 的准确值; 再利用算法 ConstructEG 构造出边数为 $ex(n; 11)$ 的图, 即得到最优解。当 $21 \leq n \leq 30$ 时, 这 2 种算法的执行时间对比见表 3, 其中 T_A 和 T_Q 分别为算法 A 和算法 ConstructEG 的执行时间。从表中可以看出, 当 n 较大时, 算法的执行时间很短。而且, 随着 n 的增加, 算法的执行时间增加幅度很小。结果表明, 算法 ConstructEG 的收敛速度很快。

表 3 2 种算法的执行时间对比

n	$ex(n; 11)$	T_A/s	T_Q/s
21	23	1 167	170
22	24	2 245	201
23	26	2 252	223
24	27	6 575	262
25	28	27 388	290
26	30	27 644	418
27	31	54 356	440
28	33	54 793	486
29	34	203 905	534
30	35	800 940	590

然后利用算法 ConstructEG,以树形图为初始解构造出围长为 11、顶点数为 n ($31 \leq n \leq 89$) 的图,从而得到相应的 $ex(n;11)$ 的下界,如表 4 所示。

表 4 $ex(n;11)$ 的下界值

n	0	1	2	3	4	5	6	7	8	9
30	-	36	37	39	40	41	42	44	45	46
40	48	49	50	52	53	55	56	57	58	60
50	61	63	64	65	67	68	70	71	73	73
60	75	77	78	80	81	82	84	85	87	88
70	89	91	92	94	95	96	97	99	99	102
80	103	104	106	107	108	110	111	99	114	116

4 结束语

本文提出一种基于量子进化的构造给定围长的图的算法。首先通过参数优化实验,选择树形图作为初始图。然后根据极图的度序列分布均匀的特性,在生成个体时选择度数小的顶点优先邻接,而在调整个体删除禁止子图的边时,则选择度数较大的顶点关联的边进行删除。最后利用极图的结构比较对称的特性,在进化过程中通过挑选对称性较好的图进入下一代进化,从而加快了算法的收敛速度。此外,在更新种群时提出一种新的旋转角步长策略,将固定调整旋转角步长与自适应调整旋转角步长相结合,同时在进化过程中添加扰动操作,使得算法可以快速跳出局部最优解。通过构造顶点数为 n ($44 \leq n \leq 48$)、围长为 10 的图,将本文算法 ConstructEG 与 PSO,GA 算法进行了对比实验。实验结果表明,该算法在求解问题的次优解和最优解的准确率上要明显优于 PSO 和 GA 算法。另外通过构造顶点数为 n ($21 \leq n \leq 30$)、围长为 11 的图,分析了 ConstructEG 算法的收敛速度,结果表明,该算法可以在较短的时间内达到最优解。最后,利用该算法构造出顶点数为 n ($31 \leq n \leq 89$) 围长为 11 的图,从而给出了相应的 $ex(n;11)$ 的下界。但是,对于更大的顶点数 n ,算法 ConstructEG 的收敛速度和收敛精度还有待于提高,这是下一步要研究的重点。

参考文献

- [1] SHOW P W. Algorithms for Quantum Computation: Discrete Logarithms and Factoring[C]//Proceedings of the 35th Annual Symposium on Foundations of Computer Science. Washington D. C., USA: IEEE Press,1994:124-134.
- [2] GROVER L K. A Fast Quantum Mechanical Algorithm for Database Search[C]//Proceedings of the 28th ACM Symposium on Theory of Computing. New York, USA: ACM Press,1996:212-219.
- [3] NARAYANAN A, MOORE M. Quantum-inspired Genetic Algorithms [C]//Proceedings of IEEE CEC '96. Washington D. C., USA: IEEE Press,1996:61-66.
- [4] HAN K H. Quantum-inspired Evolutionary Algorithm[J]. IEEE Transactions on Evolutionary Computation, 2003, 6(6):580-593.
- [5] 王 凌. 量子进化算法研究进展[J]. 控制与决策, 2008,23(12):1321-1326.
- [6] 钱 洁,郑建国,张超群,等. 量子进化算法研究现状综述[J]. 控制与决策,2011,26(3):321-326.
- [7] ZHANG G X. Quantum-inspired Evolutionary Algorithms: A Survey and Empirical Study[J]. Journal of Heuristics, 2011,17(3):303-351.
- [8] YANG Y S, ROWLINSON P. On Extremal Graphs Without Four-cycles[J]. Utilitas Mathematica,1992,41(1):204-210.
- [9] SHAO Z, XU J, XU X. A New Turin Number for Quadrilateral[J]. Utilitas Mathematica,2009,79(4):51-58.
- [10] GARNICK D K, HARRIS K Y H, FELIX L. Extremal Graphs Without Three-cycles or Four-cycles[J]. Journal of Graph Theory,1993,17(5):633-645.
- [11] SUN Y Q, LIN X H, YANG Y S. Extremal Graphs Without Four-cycles or Five-cycles[J]. Utilitas Mathematica,2009,80(5):115-130.
- [12] TANG J, LIN Y, MILLER M. Calculating the Extremal Number $ex(n; \{C_3, C_4, \dots, C_n\})$ [J]. Electronic Notes in Discrete Mathematics,2006,27:101-102.
- [13] ABAJO E, DIANEZ A. Graphs with Maximum Size and Lower Bounded Girth[J]. Applied Mathematics Letters, 2012,25(3):575-579.
- [14] WANG P, DUECK G W, MACMILLAN S. Using Simulated Annealing to Construct Extremal Graphs[J]. Discrete Mathematics,2001,235(3):125-135.
- [15] MICHALEWICZ Z, ARABAS J. Genetic Algorithms for the 0/1 Knapsack Problem[M]. Berlin, Germany: Springer,2015.
- [16] VENTER G. Particle Swarm Optimization [C]//Proceedings of International Conference on Biomedical Engineering & Informatics. Washington D. C., USA: IEEE Press,2015:129-132.
- [17] 孙永奇,杨元生. 三色拉姆塞数 $R_3(C_8)$ 研究[J]. 北京交通大学学报,2011,35(2):14-17.

编辑 索书志