

# 面向离散优化问题的量子协同演化算法\*

崔晓晖<sup>†</sup>, 齐建东, 蔡 祥  
(北京林业大学 信息学院, 北京 100083)

**摘 要:** 针对现有优化算法在求解具有时效要求的离散问题时容易出现过早或难以收敛问题,提出了面向离散优化问题的量子协同演化算法,旨在有限的求解时间内获得精度较高的求解方案。在算法的初始化阶段,通过种群初始化策略构建分布均匀的初始种群。在算法的执行阶段,将粒子群和单点优化算法改进为具有不同搜索能力的协同演化策略,利用量子旋转门根据种群个体的进化情况自适应地选择合适的演化策略。在每次迭代后利用精英保持策略避免种群退化。通过标准离散问题和背包问题对算法进行测试。实验结果表明已提出的算法在较短的迭代时间内能够稳定地收敛到精度较高的求解方案,即已提出的算法可用于求解具有时效要求的离散优化问题。

**关键词:** 离散优化问题; 协同演化算法; 量子旋转门

**中图分类号:** TP301.6

**文献标志码:** A

**文章编号:** 1001-3695(2018)08-2315-05

doi:10.3969/j.issn.1001-3695.2018.08.019

## Quantum-coevolutional algorithm for discrete optimization problem

Cui Xiaohui<sup>†</sup>, Qi Jiandong, Cai Xiang

(School of Information Science & Technology, Beijing Forestry University, Beijing 100083, China)

**Abstract:** Existing optimization algorithms are prone to premature convergence or difficult to converge when solving discrete problem that requiring solving efficiency. This paper proposed a quantum-coevolutional algorithm (QCA) for discrete optimization problem, aiming to obtain the solution with a higher precision within the limited time. In the initial phase, QCA used initialization strategy to generate the initial population with uniform distribution. In the operation phase, QCA improved the existing particle swarm optimization algorithm and the single point algorithm to the coevolutional strategies with various searching ability. Then, QCA involved quantum rotation gate to adaptively select the appropriate evolution strategy according to the individual evolution. Also, QCA adopted elitist strategy to avoid the degradation of population after each iteration. In the test environment of standard discrete problem and knapsack problem, the experimental results show that QCA can steadily converge to the solution of higher precision within the limited iterations, and will be used to solve the discrete problem requiring solving efficiency.

**Key words:** discrete optimization problem; coevolutional algorithm; quantum rotation gate

许多离散问题,如疾病控制过程中的病毒 DNA 序列识别<sup>[1]</sup>和入侵检测中的恶意特征识别<sup>[2]</sup>等,都需要求解算法在有限的时间内取得精度较高的求解方案。现有离散演化算法<sup>[3,4]</sup>,如离散粒子群优化算法 (particle swarm optimization, PSO) 和单点变异算法 (single point, SP) 等,在有限时间内通常会过早地收敛到局部最优解或无法收敛。发生该问题的主要原因是现有离散演化算法通常只倾向于一种解空间的搜索策略,如全局搜索策略或局部搜索策略<sup>[5]</sup>。在解空间分布未知的普遍情况下,单纯使用一种搜索策略往往难以在有限时间内获得较优的求解方案。

协同演化技术利用现有演化策略所具备的不同搜索能力,构建多个演化策略相融合的多策略演化算法,以期能够求解多类优化问题<sup>[6]</sup>。本文基于协同演化技术,面向具有时效要求的离散优化问题,提出量子协同演化算法 (quantum-coevolutional algorithm, QCA)。其主要工作为:首先,改进现有离散 PSO 和 SP 算法,使它们成为全局搜索能力更强的改进离散 PSO 策略 (improved discrete particle swarm optimization, IDPSO) 和局部搜索能力更强的改进离散 SP 策略 (improved discrete single point, IDSP);其次,依据迭代过程中种群的进化情况,使用量子旋转门自适应地调整 IDPSO 和 IDSP 的选择概率,使得能够促进种群进化的演化策略成为优势策略,并进一步通过优势策略的选择促使种群快速收敛。同时,为加快算法收敛速度,在算法开始阶段使用种群初始化策略生成分布均匀的初始

种群,并在每次演化后使用精英保持策略避免种群退化。

依据上述工作内容,本文首先分析协同演化技术的国内外研究现状,并提出 QCA 的算法框架;其次阐述框架中各环节的实现方法,再进行对比实验;最后总结全文并指明日后的研究方向。

### 1 国内外研究现状

对于诸多实际问题所抽象的离散优化问题,传统确定性求解方法如整数规划等,虽然在理论上可获得最优解,但需要消耗大量的计算资源和存储资源,算法的实际应用价值较低。为此,学者们提出了随机性求解方法,如遗传算法 (genetic algorithm, GA)、模拟退火算法 (simulated annealing, SA)、粒子群算法<sup>[3,4]</sup>等,这些算法通过构建具有随机性的演化策略在解空间中高效地寻找最优解。

随着人类对客观世界认知程度的提升,许多实际问题,如治病基因的监测<sup>[1]</sup>、恶意页面的识别以及入侵行为的检测<sup>[2]</sup>等,不仅要考虑更多的求解因素,还需要算法能够在有限的时间内获得精度较高的求解方案。针对这类问题,现有的随机性算法如 GA 和 SP 等,容易出现早熟或无法收敛的情况。主要原因是 GA 和 SP 等算法属于单策略的演化算法,这种单策略的演化算法在较短的迭代过程中仅会偏向一种搜索能力,或者为全局搜索,或者为局部搜索<sup>[5]</sup>。如果想发挥单一策略演化算法的搜索优势,必须了解解空间的分布情况,而这在问题求解的初期是很难达到的。因此,单策略演化算法所提供的单一

收稿日期: 2017-04-10; 修回日期: 2017-06-05 基金项目: 中央高校基本科研业务费专项资金资助项目 (BLX2014-27); 国家自然科学基金青年基金资助项目 (31400621)

作者简介: 崔晓晖 (1984-), 男 (通信作者), 山东招远人, 讲师, 博士, 主要研究方向为智能优化算法、协同演化技术 (cuixiaohui@bjfu.edu.cn); 齐建东 (1976-), 男, 教授, 博士, 主要研究方向为智能信息处理; 蔡祥 (1985-), 男, 副教授, 博士, 主要研究方向为农业智能信息处理。

搜索能力制约了随机性算法的问题适应性和求解效率。

为提高单策略演化算法的问题适应性和求解效率,学者们尝试将具有不同搜索能力的演化策略结合在一起,并形成多策略的协同演化算法。依据不同的协同方式,现有协同演化算法可分为顺序执行的协同演化算法和概率执行的协同演化算法。

1) 顺序执行的协同演化算法 对种群中的每个个体顺序地执行具备不同搜索能力的演化策略,从而使每个个体获得多次演化机会。代表成果包括文献[7]提出的一种朴素的协同方案(PSO-SA),它在每次迭代中对种群的每个个体顺序地执行具有全局搜索能力的 PSO 策略和局部搜索能力的 SA 策略;在文献[7]的基础上,文献[8,9]进一步提出了改进的顺序协同方式,即先对种群中每个个体执行全局搜索策略,然后仅对适应度最高的个体执行局部搜索策略。实验表明,顺序执行的协同演化算法比单策略的演化算法更容易获得较高的收敛精度。但其主要缺点是,顺序执行的协同演化算法在每次迭代时需对每个个体至少执行两次演化计算。若固定迭代次数,顺序执行的协同演化算法的复杂度相当于单策略演化算法的两倍或多倍。同时,理想的多策略协同过程应根据算法当前的演化情况,选择具有特定搜索能力的演化策略,而非同顺序协同演化算法一样,机械地执行多个演化策略。因此,顺序执行的协同演化算法只能有限程度地提高单策略演化算法的问题适应性和求解效率。

2) 概率执行的协同演化算法 依据演化策略的选择概率,随机地从具有不同搜索能力的演化策略中选择一个策略对种群进行演化。与顺序执行的协同方式相比,概率执行的协同方式对演化策略的使用更加灵活和科学。它会在每次迭代后,根据种群的进化情况,提高优势策略的选择概率,削弱劣势策略的选择概率,从而使得适合种群进化的优势策略成为下一次迭代的首选策略。因此,概率执行的协同演化算法在算法复杂度提升并不显著的前提下,提高种群的收敛效率和收敛精度。代表成果包括结合高斯变异和列维变异的协同演化算法<sup>[6]</sup>、结合 PSO 和 SA 策略的协同演化算法<sup>[10]</sup>等。在经典和实际的连续型优化问题上,概率执行的协同方式被证明比顺序执行的协同方式具有更快的收敛速度和更高的收敛精度。但面对具有时效要求的复杂离散问题,如何借助概率执行的策略协同优势,选择合适的演化策略并设计策略选择概率的调整方式,是目前该领域的研究热点。为此,面对离散优化问题,本文将依据协同演化算法对演化策略的选用要求,设计在搜索能力上具有显著差异的演化策略。同时,根据概率执行的策略协同方式,建立新的策略选择概率调整方法,从而实现在有限时间内求解复杂离散优化问题的高效算法。

## 2 离散优化问题描述

通常使用二元编码方式或整数编码方式对离散问题进行编码。对于整数编码方式,如果将求解方案中每个维度限制在  $\{0, 1\}$  内,则整数编码方式等价于二元编码方式,即二元编码方式是整数编码方式的特殊形式。因此,对于  $n$  维离散优化问题,本文使用更加通用的整数编码方式,如图 1 所示。

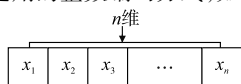


图1 整数编码方式

在整数编码方式中,每一维  $x_i$  的取值约束如式(1)所示。其中: $l_i$  和  $h_i$  分别表示  $x_i$  的取值下限和取值上限。

$$x_i \in [l_i, h_i] \wedge x_i \in \mathbb{Z} \quad (1)$$

为衡量求解方案的适应度,需根据离散问题的领域信息以及问题维度,建立相应的目标函数。假设目标为最大化求解方案的适应度,则目标函数如式(2)所示。

$$\max f(x_1, x_2, \dots, x_n) \quad (2)$$

## 3 量子协同演化算法

### 3.1 量子协同演化算法框架

为在有限时间内高效求解第 2 章描述的离散优化问题,本

文提出面向离散优化问题的量子协同演化算法(quantum-coevolutionary algorithm for discrete optimization problem, QCA)。QCA 的算法框架如图 2 所示。

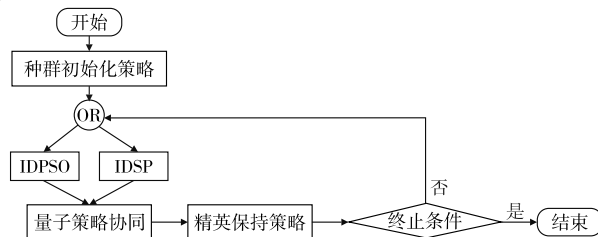


图2 QCA的算法框架

根据图 2 所示的算法框架,QCA 首先依据种群初始化策略获得分布均匀的初始种群,以期快速发现最优解;然后在 OR 过程中,确定本次迭代所使用的演化策略;其次根据 OR 过程的选择结果,使用改进的离散粒子群策略(improved discrete particle swarm optimization, IDPSO)或改进的离散单点变异策略(improved discrete single point, IDSP)对种群进行演化;然后在量子策略协同阶段,基于量子旋转门调整各演化策略的选择概率,对于适合当前种群进化的演化策略,提高其选择概率,反之,则降低其选择概率;接下来依据精英保持策略,保留种群中的精英个体并避免种群退化;最后,判断终止条件,若不满足则算法进入下一次迭代,否则算法终止并输出适应度最高的个体。QCA 算法框架中各环节的实现方法如下。

### 3.2 种群初始化策略

初始种群的分布情况对演化算法的收敛速度具有重要影响。通常分布均匀的初始种群比分布集中的初始种群更易收敛到全局最优解。

为获得分布更加均匀的初始种群,QCA 采用解空间中心对称的思想设计了种群初始化策略。种群初始化策略的具体步骤如下:

a) 针对每一维度  $x_i$ ,在其取值约束  $[l_i, h_i]$  内均匀生成  $N$  个随机数。将这  $N$  个随机数按生成顺序排列成一个序列  $X_i = [x_{i1}, x_{i2}, \dots, x_{iN}]$ 。对于  $n$  维离散优化问题,本步骤将产生  $n$  个随机数序列。

b) 对步骤 a) 产生的  $n$  个序列进行转置处理,并按照  $i$  的大小关系组合成  $N$  行  $n$  列的矩阵  $X = [X_1^T; X_2^T; \dots; X_n^T]$ 。矩阵的每一行是该离散优化问题的求解方案。

c) 将矩阵  $X$  中每一行  $[x_1, x_2, \dots, x_n]$  考虑成解空间中的一个点,求取点  $[x_1, x_2, \dots, x_n]$  关于解空间中心对称的点  $[x_1^s, x_2^s, \dots, x_n^s]$ ,其中  $x_i$  和  $x_i^s$  的关系满足式(3)。本步骤将产生  $N$  个对称的求解方案。

$$(l_i + h_i)/2 = (x_i^s + x_i)/2 \Leftrightarrow x_i^s = l_i + h_i - x_i \quad (3)$$

d) 使用目标函数衡量步骤 b) c) 形成的  $2N$  个求解方案,保留适应度较优的前  $N$  个求解方案作为初始种群。

在步骤 a) 中,若存在两个维度有同样的取值约束  $[l_i, h_i]$ ,则可以选择其中一个维度,将其产生的随机数以  $\text{mod}(n+1)$  的方式排列成序列  $X_i$ ,即  $X_i = [x_{iN}, x_{i1}, x_{i2}, \dots, x_{iN-1}]$ ,以避免两个维度产生相同的随机序列。当多个维度具有相同约束时,处理方式类似。

除生成的初始种群外,还需初始化 IDPSO 的局部最优个体、全局最优个体等。

### 3.3 OR 过程

OR 过程根据演化策略的选择概率,确定每次迭代使用的演化策略。具体步骤如下:

a) 使用  $P_{\text{IDPSO}}$  和  $P_{\text{IDSP}}$  分别表示 IDPSO 和 IDSP 的选择概率。在 QCA 演化过程中, $P_{\text{IDPSO}}$  和  $P_{\text{IDSP}}$  始终满足式(4)。

$$P_{\text{IDPSO}} + P_{\text{IDSP}} \equiv 1 \quad (4)$$

b) 比较随机数  $\text{rand}()$  与  $P_{\text{IDPSO}}$  的数量关系,若  $\text{rand}() < P_{\text{IDPSO}}$ ,则在本次迭代中使用 IDPSO 对种群进行演化,否则使用 IDSP 对种群进行演化。

### 3.4 改进的离散粒子群演化策略

根据 QCA 的设计思路,粒子群演化策略充当全局搜索的

角色。为此,可改进经典粒子群中的参数,构建全局搜索能力更强的改进离散粒子群演化策略(IDPSO)。

IDPSO 涉及的参数包括  $k$  标志不同粒子、 $d$  表示粒子的维度、 $t$  表示迭代次数、 $v_{k,d}(t)$  表示粒子的速度、 $x_{k,d}(t)$  表示粒子的位置。

为使得粒子能够更加有效地进行全局搜索,针对粒子的每一个维度  $d$ ,IDPSO 采用具有差异化的最大速度  $v_{d,\max}$  构成速度约束  $[-v_{d,\max}, v_{d,\max}]$ ,  $v_{d,\max}$  的计算方法如式(5)所示。其中:  $\lambda$  用于控制粒子的速度约束,  $\lambda = 0.8$ 。

$$v_{d,\max} = \lambda(h_i - d_i) \quad (5)$$

在上述参数定义的基础上,IDPSO 的具体步骤如下:

a) 使用式(6)调整粒子的速度。

$$v_{k,d}(t+1) = \omega v_{k,d}(t) + c_1 r p_{k,d}(t)(p_{k,d}(t) - x_{k,d}(t)) + c_2 r g_{k,d}(t)(g_{k,d}(t) - x_{k,d}(t)) \quad (6)$$

式(6)由三部分构成,分别为惯性部分、趋向局部最优部分和趋向全局最优部分。惯性部分表明邻代粒子速度的继承程度,其中  $\omega \in [0,1]$  称为速度惯性系数,其数值越大,粒子在解空间中的全局搜索范围越广。因为 QCA 需要 IDPSO 具备全局搜索能力,所以  $\omega = 1$ 。趋向于局部最优部分表明粒子在局部最优个体吸引下的移动情况,  $r p_{k,d}(t)$  为该环节的随机数;趋向全部最优部分表明粒子在全局最优位置引导下的移动情况,  $r g_{k,d}(t)$  为该环节的随机数。文献[11]指出,  $r p_{k,d}(t)$  和  $r g_{k,d}(t)$  随机度越高,粒子在解空间的移动越灵活,且更容易逃离局部最优位置。为此,IDPSO 基于随机性较强的混沌随机数,产生  $r p_{k,d}(t)$ ,如式(7)所示。在式(7)中,当  $\mu = 4$  时,会产生混沌效应。  $r g_{k,d}(t)$  的产生方式与  $r p_{k,d}(t)$  类似。

$$r p_{k,d}(t) = \mu r p_{k,d}(t-1)(1 - r p_{k,d}(t-1)) \quad (7)$$

b) 使用式(8)判断式(7)调整后的速度是否在其约束范围内。

$$v_{k,d}(t+1) = \max(\min(v_{k,d}(t+1), v_{d,\max}), -v_{d,\min}) \quad (8)$$

c) 使用式(9)调整粒子的位置。当粒子的位置超过约束的上限或下限时,则以约束的上限和下限代替该维度的位置。

$$x_{k,d}(t+1) = \max(\min(\lceil x_{k,d}(t) + v_{k,d}(t+1) \rceil, h_d), l_d) \quad (9)$$

### 3.5 改进的离散单点变异策略

单点变异演化策略负责解空间的局部搜索。经典的单点变异算法会随机地选择种群中一些个体,并只针对该个体的某一维度进行随机扰动。这种变异个体的选择以及变异维度的扰动方法均存在一定的盲目性。在 QCA 的演化过程中,适应度较高的个体通常位于最优解的邻域内,因此,可以利用这一特点来改进现有单点变异的演化算法,构建改进的离散单点变异策略(IDSP)。IDSP 的具体步骤如下:

a) 将局部最优个体、全局最优个体和现有种群中适应度最高的个体构成单点变异的候选集。

b) 针对候选集中的每个个体,随机选取个体的某一维度并执行 +1 和 -1 操作。

c) 使用式(10)调整变异后不在其约束范围内的维度。

$$x_{k,d}(t+1) = \max(x_{k,d}(t), h_d), l_d) \quad (10)$$

### 3.6 量子策略协同

量子策略协同基于量子旋转门调整演化策略的选择概率,实现各演化策略的概率协同。量子策略协同的设计思路是:在量子信息学中,一个量子比特可由两个量子态线性叠加得到。在  $R^2$  空间中,所有量子比特构成了单位圆上的点,即所有量子态的线性叠加均为 1,因此,可以使用量子态表示演化策略的选择概率。而量子旋转门描述了量子比特从一个状态到另一个状态的迁移过程,迁移后的量子比特仍然在单位圆上,因此,可以通过量子旋转门调整演化策略的选择概率。量子策略协同的设计思路如图 3 所示。

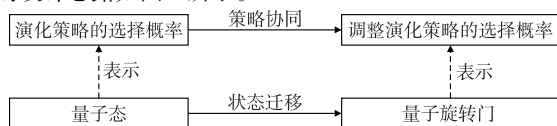


图3 量子策略协同的设计思路

QCA 中量子策略协同过程如下:

a) 将策略的选择概率  $P_{IDPSO}$  和  $P_{IDSP}$  表示成量子态。

b) 分析种群中的每个个体演化前后适应度的变化情况。如果适应度提高,则说明当前使用的演化策略为优势策略,应上调策略选择概率对应的量子态;反之,则下调对应的量子态。而量子态的上调和下调均可以看作是量子比特在量子旋转门作用下的迁移,如式(11)所示。

$$\begin{bmatrix} \sqrt{P_{IDPSO}(t+1)} \\ \sqrt{P_{IDSP}(t+1)} \end{bmatrix} = \begin{bmatrix} \cos(c\theta/2N) & -\sin(c\theta/2N) \\ \sin(c\theta/2N) & \cos(c\theta/2N) \end{bmatrix} \begin{bmatrix} \sqrt{P_{IDPSO}(t)} \\ \sqrt{P_{IDSP}(t)} \end{bmatrix} \quad (11)$$

其中:  $\theta$  表示量子态的夹角;符号变量  $c = \{-1, 0, 1\}$  表示了量子旋转门的旋转方向,用于控制量子态的上调和下调。当  $c = 1$  时,量子旋转门以顺时针方向旋转,旋转后  $P_{IDPSO}$  会增加而  $P_{IDSP}$  会下降;当  $c = -1$  时,量子旋转门以逆时针方向旋转,旋转后  $P_{IDSP}$  会增加而  $P_{IDPSO}$  会下降;当  $c = 0$  时,量子旋转门不旋转,  $P_{IDPSO}$  和  $P_{IDSP}$  保持不变。结合 QCA 的迭代过程,量子旋转门的顺时针旋转适合在 IDPSO 促进种群个体进化或 IDSP 导致种群个体退化情况下调整  $P_{IDPSO}$  和  $P_{IDSP}$ ,量子旋转门的逆时针旋转适合在 IDSP 促进种群个体进化或 IDPSO 导致种群个体退化情况下调整  $P_{IDPSO}$  和  $P_{IDSP}$ ,量子旋转门不旋转适合在  $P_{IDPSO}$  和  $P_{IDSP}$  在边界值时仍需调整的情况。根据上述量子旋转门的旋转方式,可建立表 1,方便 QCA 在演化过程中快速确定式(10)中  $c$  的取值。

表 1 不同条件下  $c$  的取值

当前策略	是否进化	$c$ 的取值		
		一般情况	$P_{IDPSO} = 0$ $P_{IDSP} = 1$	$P_{IDSP} = 0$ $P_{IDPSO} = 1$
IDPSO	N	+1	0	+1
IDPSO	Y	-1	-1	0
IDSP	N	-1	-1	0
IDSP	Y	+1	0	+1

### 3.7 精英保持策略

离散优化算法在对问题进行求解时通常会出现种群退化问题。为解决该问题,将演化前后的个体按照个体适应度进行排序,选择适应度排在前  $N$  位的精英个体作为下一次迭代的种群。同时,为避免精英保持对个体产生趋同的影响,当种群平均适应度在五次迭代后仍保持不变时,使用 IDSP 对种群中的次优个体和最差个体进行随机扰动,并将扰动后的个体重新加入到种群中,从而提高种群的多样性。

## 4 仿真分析

### 4.1 实验准备

1) 对比算法及选取依据 本文分别选取三类对比算法:a) 单一策略的演化算法,用于分析演化策略的数量差异对优化结果的影响,代表算法包括离散粒子群算法(DPSO)和单点优化算法(SP);b) 顺序执行的演化算法,用于分析演化策略的不同协同方式对优化结果的影响,代表算法为文献[7]建立的 DPSOWSP;c) 与 QCA 策略协同方式类似但在策略调整方式和算法框架上存在差异的算法,代表算法为文献[10]建立的 DPSO-SP。

2) 参数设定方法 为确保实验结果的可比性,将 QCA 中 IDPSO 和其他粒子群算法的  $c_1$  和  $c_2$  设置为 2。除 IDPSO 外,其他粒子群算法的  $v_{\max}$  设置为 6。鉴于 QCA 面对的是具有时效的离散优化问题,应考虑在种群规模适中的情况下( $N = 30$ ),各算法在有限迭代次数内的收敛情况。将最大迭代次数  $t_{\max}$  作为各算法的终止条件,  $t_{\max} = 100$ 。

### 4.2 算法的标准测试环境

使用检验函数<sup>[12]</sup>构建算法的标准测试环境。检验函数是学者根据众多实际问题抽象的函数集合,具有与实际问题结构类似且代表性强等特点。选取不同类型的检验函数( $f_1 \sim f_5$ )构成实验环境,综合比较不同算法在不同类型解空间下的实际

应用价值。算法测试所使用的检验函数如表 2 所示。

表 2 检验函数

No	函数表达式	取值范围	最优值	类型
$f_1$	$f(x) = \sum_{i=1}^n x_i^2$	$[-10, 10]$	$x^* = (0, \dots, 0)$ $f(x) = 0$	单峰
$f_2$	$f(x) = \sum_{i=1}^n x_i^2 + (\sum_{i=1}^n 0.5ix_i)^2 + (\sum_{i=1}^n 0.5ix_i)^4$	$[-10, 10]$	$x^* = (0, \dots, 0)$ $f(x) = 0$	单峰
$f_3$	$f(x) = 10n + \sum_{i=1}^n (x_i^2 - 10 \cos(2\pi x_i))$	$[-10, 10]$	$x^* = (0, \dots, 0)$ $f(x) = 0$	多峰
$f_4$	$f(x) = 10 + e - 20e^{-0.2\sqrt{\frac{1}{n}\sum_{i=1}^n x_i^2}} - \frac{1}{e} \sum_{i=1}^n \cos(2\pi x_i)$	$[-10, 10]$	$x^* = (0, \dots, 0)$ $f(x) = 0$	多峰
$f_5$	$f(x) = \sum_{i=1}^n \frac{x_i^2}{4000} - \prod_{i=1}^n \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	$[-10, 10]$	$x^* = (0, \dots, 0)$ $f(x) = 0$	多峰

表 2 中的检验函数均为连续型函数,在对不同离散优化算法进行测试时,需对每个测试函数进行离散化处理。

#### 4.3 标准测试环境的收敛精度分析

在  $f_1 \sim f_5$  维度为  $\{10, 20, 30, 40, 50\}$  等情况下,执行 QCA 和各对比算法 100 轮。记录各函数上各算法每轮迭代后的最优收敛精度,统计最优收敛精度的平均值。各函数上 QCA 和对比算法的收敛精度如表 3~7 所示。

表 3  $f_1$  函数上 QCA 及各对比算法的收敛精度

维度	DPSO	SP	DPSOWSP	DPSO-SP	QCA
10	2.22	6.75	0	0	0
20	67.1	45.52	10.54	0	0
30	191.88	235.48	15.38	7.12	0
40	360.72	432.54	22.45	10.64	5.24
50	530.72	782.12	30.94	17.67	12.54

表 4  $f_2$  函数上 QCA 及各对比算法的收敛精度

维度	DPSO	SP	DPSOWSP	DPSO-SP	QCA
10	927.026	964.14	332.5	136.41	137.11
20	2.84E7	3.79E8	1.56E3	362.28	258.32
30	6.713E9	9.652E10	2.665E4	864.57	535.15
40	1.716E11	2.648E13	5.486E5	1.344E3	687.98
50	2.224E12	3.448E15	8.547E7	1.542E6	1.35E3

表 5  $f_3$  函数上 QCA 及各对比算法的收敛精度

维度	DPSO	SP	DPSOWSP	DPSO-SP	QCA
10	2.42	5.64	0	0	0
20	65.08	78.21	15.61	0	0
30	196.92	200.36	50.36	23.11	21.12
40	371.0	432.15	134.4	53.48	52.15
50	528.28	681.58	357.36	166.78	126.87

表 6  $f_4$  函数上 QCA 及各对比算法的收敛精度

维度	DPSO	SP	DPSOWSP	DPSO-SP	QCA
10	1.627	5.216	0	0	0
20	6.22	10.78	0	0	0
30	8.139	12.36	1.745	0	0
40	8.983	22.45	2.974	1.68	1.24
50	9.552	26.74	5.169	2.31	1.96

表 7  $f_5$  函数上 QCA 及各对比算法的收敛精度

维度	DPSO	SP	DPSOWSP	DPSO-SP	QCA
10	1.001	2.02	0	0	0
20	1.02	3.09	0	0	0
30	1.050	4.24	1.005	1	1
40	1.089	4.98	1.05	1	1
50	1.134	6.35	1.09	1.009	1

根据表 3~7,总体上 QCA 及各对比算法的平均收敛精度均随问题维度的升高呈现下降趋势。具体情况如下:

第一类对比算法(单一策略的演化算法 DPSO 和 SP)在维度较低(小于 10)和优化问题较简单( $f_1$  和  $f_3$ )的情况下,能够收敛到最优精度 0 的领域范围内;当问题维度以及问题复杂性

上升时,DPSO 和 SP 的收敛精度显著下降,且在迭代结束时难以收敛。

第二类对比算法(DPSOWSP 算法)在高维度问题的测试环境中,收敛精度显著优于第一类演化算法。该实验结果表明多策略演化算法比单一策略更适用于复杂、高维的离散优化问题。

概率协同的演化算法(DPSO-SP 和 QCA 算法)在维度较低的简单问题上,能够收敛到全局最优解 0;当面对复杂的高维离散优化问题时,DPSO-SP 和 QCA 的平均收敛精度要优于顺序执行的协同演化算法。该实验结果表明概率执行的协同演化算法在复杂离散优化问题上更容易获得精度较高的求解方案。

单纯对比 DPSO-SP 和 QCA,两者虽然均属概率执行的协同优化算法,但是在选择概率的调整方式上存在差别。根据实验结果,QCA 在各项对比环境下的平均收敛精度均优于 DPSO-SP。该实验结果表明 QCA 和 DPSO-SP 的差异环节,如初始策略、量子策略协同和精英保持策略等,对提高算法在有限时间内的求解精度具有积极作用。

综合上述分析,各算法在有限迭代次数内,问题求解能力的排序为:QCA > DPSO-SP > DPSOWSP > DPSO > SP。

#### 4.4 标准测试环境的收敛情况分析

为更加精细地对比各算法的收敛情况,观察 DPSO、DPSOWSP 和 QCA 在高维(50 维)单峰问题  $f_1$  和多峰问题  $f_3$  上的收敛情况,如图 4 和 5 所示。

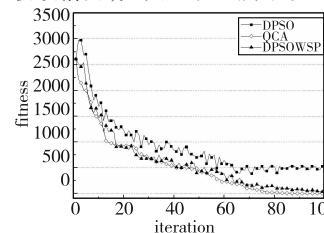


图 4 各算法在高维  $f_1$  上的收敛情况

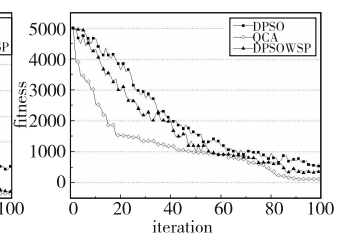


图 5 各算法在高维  $f_3$  上的收敛情况

根据图 4 和 5,在高维问题  $f_1$  和  $f_3$  上,各算法总体上的精度随迭代次数的增加呈现收敛的趋势。具体情况如下:

从各算法的迭代过程来看,DPSO 和 DPSOWSP 的适应度均出现了一定程度的波动情况,而 QCA 则呈现稳定的下降趋势,其原因是 QCA 采用的精英保持策略能够有效保持适应度较高的优势个体,避免种群在离散优化过程中普遍存在的退化现象。

从迭代结束后的求解精度上来看,单一策略的演化算法 DPSO 在高维  $f_1$  和  $f_3$  问题上更容易陷入局部最优解,顺序执行的协同演化算法更加适合单峰复杂的优化问题,但对于多峰高维问题  $f_3$ ,QCA 的最终收敛精度要优于各对比算法。

上述分析结果表明,在有限迭代次数内,概率执行的协同演化算法对高维离散优化问题的寻优能力要优于顺序执行的协同演化算法和单一策略的演化算法。

#### 4.5 背包问题的测试环境

使用多维多选择背包问题<sup>[13]</sup>分析各算法对实际离散问题的优化效果。多维多选择背包问题是许多实际离散优化问题<sup>[14]</sup>的抽象形式。它的目标是在背包容量  $C$  和物品容量  $w$  的约束下,选择一定数量的物品,使得所选物品的价值最大。实验时选取 Brunel University 提供的七个多维多选择背包的测试用例<sup>[15]</sup>,每个测试用例使用  $M$  表示物品数量, $N$  表示物品总数, $R$  表示每类物品的数量。

#### 4.6 背包问题的平均收敛精度

在七个测试用例上,分别运行贪心算法(Greedy)、DPSO 算法、DPSOWSP 算法和 QCA 20 轮。20 轮优化后各算法的平均收敛精度如表 8 所示。根据表 8 中的实验结果,在第 1 行和第 2 行等维度较低的简单离散优化问题上,DPSO 的平均收敛精度要低于 Greedy 的平均收敛精度;在第 3~7 行等维度相对较高的离散优化问题上,DPSO 的平均收敛精度均高于 Greedy 的平均收敛精度。该实验结果表明,在简单离散优化问题上,传统的贪心算法反而比单一策略的演化算法更具优势。

表8 各算法在多维多选择背包问题上的平均收敛精度

背包参数设置			Greedy	DPSO	DPSOWSP	QCA
<i>N</i>	<i>M</i>	<i>R</i>				
25	5	5	169.67	163.23	170.32	171.62
50	10	5	351.61	344.71	352.63	355.14
150	15	10	1 430.60	1 604.62	1 620.35	1 624.15
200	20	10	3 374.45	3 525.31	3 482.87	3 501.21
250	25	10	3 577.66	3 759.83	3 876.76	3 890.64
300	30	10	4 200.58	4 680.67	4 726.35	4 750.16
1 000	100	10	2.08E4	2.17E4	2.34E4	2.48E4

横向对比 DPSOWSP、DPSO 和 Greedy 的平均收敛精度, DPSOWSP 在每个测试用例上均优于 DPSO 和 Greedy。该实验结果进一步说明具有多个策略的协同演化算法在各类离散优化问题上比单一策略的演化算法和传统算法更具优势。对比 DPSOWSP 和 QCA 的平均收敛精度, QCA 在维度较低的离散优化问题上收敛优势并不显著;随着问题维度的上升, QCA 的收敛优势越发显著。该实验结果表明采用概率方式协同多个策略比顺序执行的协同方式更适合实际的离散优化问题。

综合上述多维多选择背包问题的实验分析结果,与现有单一策略的演化算法 DPSO、顺序执行的多策略演化算法(DPSOWSP)相比, QCA 更适用于实际离散优化问题的求解,能够在有限时间内获得较高的平均收敛精度。

## 5 结束语

针对具有时效要求的离散优化问题,本文综合了种群初始化策略、改进的离散粒子群策略,改进的离散单点变异策略和量子策略协同等过程,建立了面向离散优化问题的量子协同演化算法。实验结果显示,已提出的算法能够在有限时间内收敛到精度较高的求解方案,可用于求解具有时效要求的离散优化问题。未来将 QCA 用于求解实际高维离散问题,并通过解空间的分布特征形成启发性信息,促进种群收敛,减缓维度变化对求解效率的消极影响,提高算法的实际应用价值。

## 参考文献:

- [1] Wong Kachun, Peng Chengbin, Wong M H, *et al.* Generalizing and learning protein-DNA binding sequence representations by an evolutionary algorithm[J]. *Soft Computing*, 2011, 15(8): 1631-1642.
- [2] Wargh S, Kolhe S. Effective semi-supervised approach towards intrusion detection system using machine learning techniques[J]. *International Journal of Electronic Security and Digital Forensics*, 2015, 7(3): 1290-1304.
- [3] Li D, Cheng B, Deng N, *et al.* QoS-aware service composition algorithm and architecture by discrete PSO[J]. *Journal of Computational Information Systems*, 2010, 6(2): 501-511.
- [4] Qin Jin, Lin Xin, Yin Yixin. An algorithmic framework of discrete particle swarm optimization[J]. *Applied Soft Computing*, 2012, 12(3): 1125-1130.
- [5] 崔晓晖, 印桂生, 董红斌. 面向服务匹配问题的协同演化算法[J]. *软件学报*, 2015, 26(7): 1601-1614.
- [6] Dong Hongbin, He Jun, Huang Houkuan, *et al.* Evolutionary programming using a mixed mutation strategy[J]. *Information Science*, 2007, 177(1): 312-327.
- [7] Behnamian J, Fatemi S M T. Development of a PSO-SA hybrid meta-heuristic for a new comprehensive regression model to time-series forecasting[J]. *Expert Systems with Application*, 2010, 37(2): 974-984.
- [8] Idoumghar L, Melkemi M, Schott R, *et al.* Hybrid PSO-SA type algorithms for multimodal function optimization and reducing energy consumption in embedded systems[J]. *Applied Computational Intelligence and Soft Computing*, 2011, 2011(10): 1-12.
- [9] Kathpal S, Vohra R, Singh J, *et al.* Hybrid PSO-SA algorithm for achieving partitioning optimization in various network application [C]// Proc of Conference on Modeling Optimization and Computing. Holland: Elsevier, 2012: 1728-1734.
- [10] Zhang Xin, Dong Hongbin. A new co-evolutionary programming using operators of single-point and particle swarm [C]//Proc of Conference on Modeling Optimization and Computing. Washington DC: IEEE Computer Society, 2010: 5528-5534.
- [11] Bharti K K, Singh P K. Opposition chaotic fitness mutation based adaptive inertia weight BPSO for feature selection in text clustering [J]. *Applied Soft Computing*, 2016, 43(6): 20-34.
- [12] Chen Ning, Gravin N, Lu Pinyan. Competitive analysis via benchmark decomposition [C]// Proc of ACM Conference on Economics and Computation. New York: ACM Press, 2015: 363-367.
- [13] Han Bing, Leblet J, Simon G. Hard multidimensional multiple choice knapsack problems, an empirical study[J]. *Computers & Operations Research*, 2010, 37(1): 172-181.
- [14] 苏爱东, 高维珉, 张永翼. 基于贪心程度和区域界定的预期效率模型求解 0-1 背包问题[J]. *计算机应用研究*, 2015, 32(11): 3304-3308.
- [15] Khan S, Li K F, Manning E G, *et al.* Solving the knapsack problem for adaptive multimedia system[J]. *Studia Informatica, Special Issue on Cutting, Packing and Knapsacking Problems*, 2001, 9: 157-178.
- [16] 程与电子技术, 2009, 31(10): 2476-2479.
- [14] 江岳春, 张丙江, 邢方方, 等. 基于混沌时间序列 GA-VNN 模型的超短期风功率多步预测[J]. *电网技术*, 2015, 39(8): 2160-2166.
- [15] Zhang Xuguang, Ouyang Meiling, Zhang Xufeng. Small scale crowd behavior classification by Euclidean distance variation-weighted network[J]. *Multimedia Tools & Applications*, 2016, 75(19): 11945-11960.
- [16] 杜杰, 曹一家, 刘志坚, 等. 混沌时间序列的局域高阶 Volterra 滤波器多步预测模型[J]. *物理学报*, 2009, 58(9): 5997-6005.
- [17] 王桓, 何怡刚, 侯周国, 等. 基于夹角余弦的混沌局域加权线性预测算法[J]. *高电压技术*, 2009, 35(6): 1483-1487.
- [18] Guo Yina, Liu Qijia, Wang Anhong, *et al.* Optimized phase-space reconstruction for accurate musical-instrument signal classification [J]. *Multimedia Tools & Applications*, 2017, 76(20): 20719-20737.
- [19] Pang Yanjie, Zhang Jiashu. A hierarchical alternative updated adaptive Volterra filter with pipelined architecture[J]. *Digital Signal Processing*, 2016, 56(9): 67-78.
- [20] Pascoal C, Oliveira M R, Pacheco A, *et al.* Theoretical evaluation of feature selection methods based on mutual information[J]. *Neurocomputing*, 2016, 226(2): 168-181.
- [21] Oh B S, Sun Lei, Ahn C S, *et al.* Extreme learning machine based mutual information estimation with application to time-series change-points detection[J]. *Neurocomputing*, 2017, 261(10): 204-206.

(上接第 2273 页)

- [7] Tariq I, Chen Tao, Kirkby N F, *et al.* Modelling and Bayesian adaptive prediction of individual patients' tumour volume change during radiotherapy[J]. *Physics in Medicine & Biology*, 2016, 61(5): 2145-2161.
- [8] Vinoth K C, Natarajan V. Hybrid local prediction error-based difference expansion reversible watermarking for medical images[J]. *Computers & Electrical Engineering*, 2016, 53(7): 333-345.
- [9] Wen Long, Li Xinyu, Gao Liang, *et al.* Surface roughness prediction in end milling by using predicted point oriented local linear estimation method[J]. *International Journal of Advanced Manufacturing Technology*, 2016, 84(9): 2523-2535.
- [10] Patra A, Das S, Mishra S N, *et al.* An adaptive local linear optimized radial basis functional neural network model for financial time series prediction[J]. *Neural Computing & Applications*, 2015, 28(1): 1-10.
- [11] Li Yanling, Zhang Yunpeng, Wang Jing, *et al.* The Volterra adaptive prediction method based on matrix decomposition[J]. *Journal of Interdisciplinary Mathematics*, 2016, 19(2): 363-377.
- [12] Batista E L O, Rui S. A reduced-rank approach for implementing higher-order Volterra filters[J]. *Eurasip Journal on Advances in Signal Processing*, 2016, 2016(12): 118.
- [13] 方芬. 多变量混沌时间序列 Volterra 自适应实时预测[J]. *系统工*