

## 4

### Algoritmos com Inspiração Quântica e PG

Algoritmos com inspiração quântica exploram os conceitos e princípios da mecânica quântica para informar e inspirar métodos mais eficientes de computação evolucionária. A ideia da computação com inspiração quântica não é de desenvolver novos algoritmos para execução em computadores quânticos ou de fazer seu uso possível. A proposta da inspiração quântica é de criar algoritmos clássicos (i.e., executam em computadores clássicos) que utilizam os paradigmas da mecânica quântica para melhorar a qualidade e o desempenho da solução do problema. Em particular, algoritmos evolutivos de inspiração quântica (*quantum-inspired evolutionary algorithms*–QEAs) têm recebido especial atenção na computação evolutiva.

A superposição linear de estados permite a representação de diferentes indivíduos probabilisticamente. QEAs pertencem à classe de algoritmos de estimação de distribuição (*estimation of distribution algorithms*–EDAs) [60]. O mecanismo probabilístico provê aos QEAs um mecanismo de evolução diferente que é muitas vezes mais efetivo do que muitos algoritmos evolucionários clássicos [26]. Existem duas razões principais para isto. A primeira é que a representação probabilística das possíveis soluções reduz consideravelmente o número de cromossomos necessários para garantir uma boa diversidade de busca [72]. A segunda razão é que o uso da interferência quântica oferece uma ferramenta poderosa para reforçar a estabilidade da busca. Desta forma, ela fornece um guia para a população de indivíduos possibilitando uma boa exploração das vizinhanças das soluções correntes na busca das melhores soluções [73].

QEAs têm sido utilizados com sucesso para resolver uma série de problemas, tais como: o problema da mochila [73], problemas de otimização combinatorial [74], problemas de otimização na engenharia [75], segmentação de imagens [76], e registro de imagem [77]. Em [26], encontramos mais exemplos de QEAs e suas aplicações.

As metodologias de compilação em linguagem intermediária e criação de indivíduos em código de máquina propostas neste trabalho utilizam o modelo de Programação Genética Linear com Inspiração Quântica (*Quantum Inspired Linear Genetic Programming* – QILGP) desenvolvido por Dias e Pacheco [78, 63].

## 4.1

**Programação Genética Linear com Inspiração Quântica – QILGP**

QILGP evolui indivíduos de PG ou programas de computador utilizando código de máquina de CPU. O modelo apresenta viabilidade para ser utilizado em qualquer plataforma de CPUs, desde que sejam conhecidas as suas instruções em código de máquina, contudo, atualmente são conhecidas e utilizadas as instruções em código de máquina de CPUs para a plataforma Intel x86. Para evoluir cada indivíduo são utilizadas instruções de ponto flutuante, com acesso a dados da memória principal ( $m$ ) e/ou oito registradores da FPU ( $ST(i) \mid i \in [0..7]$ ). O conjunto de funções é composto por instruções de adição, subtração, multiplicação, divisão, transferência de dados, operações trigonométricas e aritméticas. Um indivíduo ou programa é composto por uma combinação de diversas destas funções ou instruções, sendo que cada instrução é representada por um linha de código no programa e é chamada de gene em PG. QILGP gera programas com tamanhos variados através da adição da instrução NOP ao conjunto de instruções. Durante a geração do código, qualquer gene que contenha a instrução NOP é ignorado. A Tabela 4.1 apresenta um exemplo do conjunto de instruções.

Tabela 4.1: Descrição funcional das instruções. A primeira coluna apresenta as instruções da plataforma Intel x86. A segunda coluna apresenta as operações realizadas. A terceira coluna apresenta o argumento das instruções.

Instrução	Operação	Arg.
NOP	Nenhuma Operação	-
FADD $m$	$ST(0) \leftarrow ST(0) + m$	$m$
FADD $ST(0), ST(i)$	$ST(0) \leftarrow ST(0) + ST(i)$	$i$
FADD $ST(i), ST(0)$	$ST(i) \leftarrow ST(i) + ST(0)$	$i$
FSUB $m$	$ST(0) \leftarrow ST(0) - m$	$m$
FSUB $ST(0), ST(i)$	$ST(0) \leftarrow ST(0) - ST(i)$	$i$
FSUB $ST(i), ST(0)$	$ST(i) \leftarrow ST(i) - ST(0)$	$i$
FMUL $m$	$ST(0) \leftarrow ST(0) \times m$	$m$
FMUL $ST(0), ST(i)$	$ST(0) \leftarrow ST(0) \times ST(i)$	$i$
FMUL $ST(i), ST(0)$	$ST(i) \leftarrow ST(i) \times ST(0)$	$i$
FXCH $ST(i)$	$ST(0) \leftrightarrow ST(i)$ (swap)	$i$
FDIV $m$	$ST(0) \leftarrow ST(0) \div m$	$m$
FDIV $ST(0), ST(i)$	$ST(0) \leftarrow ST(0) \div ST(i)$	$i$
FDIV $ST(i), ST(0)$	$ST(i) \leftarrow ST(i) \div ST(0)$	$i$
FABS	$ST(0) \leftarrow  ST(0) $	-
FSQRT	$ST(0) \leftarrow \sqrt{ST(0)}$	-
FSIN	$ST(0) \leftarrow \sin ST(0)$	-
FCOS	$ST(0) \leftarrow \cos ST(0)$	-

Cada indivíduo é representado por uma sequência linear de instruções

em código de máquina. Cada instrução pode usar um ou nenhum argumento. A avaliação de um programa requer que os dados de entrada sejam lidos da memória principal, os quais são constituídos pelas variáveis de entrada do problema e por algumas constantes opcionais fornecidas pelo usuário. Os dados de entrada são representados por um vetor, tal como

$$I = (V[0], V[1], 1, 2, 3), \quad (4-1)$$

onde  $V[0]$  e  $V[1]$  representam os dois valores de entrada do problema (isto é, uma amostra de dados) e 1, 2, e 3 são os três valores das constantes.

As instruções são representadas em QILGP através de dois *tokens*:

- **Token de Função** (TF): que representa a função; e
- **Token de Terminal** (TT): que representa o argumento da função.

Cada função possui um único terminal. Quando a função não tem terminal, o seu valor de token correspondente é ignorado. Cada token é um valor inteiro que representa um índice para o conjunto de funções ou para o conjunto de terminais.

## 4.2 Representação

QILGP é baseada nas seguintes entidades:

- **indivíduo quântico**, que representa a superposição de todos os possíveis programas para o espaço de busca definido; e
- **indivíduo clássico**, que representa o programa em código de máquina codificado através dos valores dos tokens. Na fase de observação da QILGP, cada indivíduo quântico é observado para gerar um indivíduo clássico.

## 4.3 Observação

O cromossomo de um indivíduo quântico é representado por uma lista de estruturas chamadas de **genes quânticos**. Cada um destes genes quânticos representa a superposição de todas as possíveis instruções do conjunto de funções apresentado na Tabela 4.1 e durante o processo de observação ocorre o colapso de cada gene quântico para uma instrução específica, originando o gene clássico. A observação de um indivíduo quântico compreende a observação de todos os genes de seu cromossomo. A representação de cada uma das instruções tem um probabilidade de ocorrer,  $p_i$ , a qual é chamada de estado. Um estado é

representado por um intervalo fechado de probabilidades  $I_i$  cujos limites finais representam as probabilidades acumuladas. O processo de observação consiste em gerar aleatoriamente um valor  $r$   $\{r \in \mathbb{R} \mid 0 \leq r \leq 1\}$  e de uma busca pelo intervalo ao qual  $r$  pertence dentre todos os estados possíveis que podem representar o indivíduo. Por exemplo, o processo de observação de um gene quântico representado por 10 estados diferentes segue a função:

$$T(r) = \begin{cases} 0 & \text{se } 0 \leq r < p'_0 \\ 1 & \text{se } p'_0 \leq r < p'_1 \\ 2 & \text{se } p'_1 \leq r < p'_2 \\ \vdots & \vdots \\ 9 & \text{se } p'_8 \leq r \leq p'_9, \end{cases} \quad (4-2)$$

onde  $\{r \in \mathbb{R} \mid 0 \leq r \leq 1\}$  é o valor gerado aleatoriamente com uma distribuição uniforme;  $p'_0, p'_1, p'_2, \dots, p'_8$  e  $p'_9$ , são os valores dos intervalos das probabilidades associados com cada token; e  $T(r)$  retorna o valor observado para o token.

O processo de observação tem um papel importante em um algoritmo evolutivo com inspiração quântica. A representação com inspiração quântica de um gene implica que a criação de cada instrução segue uma distribuição probabilística, onde é possível representar as instruções que são mais suscetíveis de serem observadas. Além disso, o algoritmo evolutivo pode ser alimentado com os resultados das avaliações dos indivíduos, e a superposição dos estados possibilita que os valores das probabilidades sejam melhorados iterativamente. Os melhores indivíduos clássicos contribuem para melhorar os valores das probabilidades dos indivíduos quânticos. Este mecanismo permite que o algoritmo encontre soluções melhores com um número menor de avaliações.

QILGP é inspirada pelos sistemas quânticos multiníveis [79] e usa o *qudit* como a unidade básica de informação. Esta informação pode ser descrita por um vetor de estados de  $d$  níveis, onde  $d$  é o número de estados nos quais o *qudit* pode ser medido. Assim,  $d$  representa a cardinalidade do token. O estado de um *qudit* é uma superposição linear dos  $d$  estados e pode ser representado como segue:

$$|\psi\rangle = \sum_{i=0}^{d-1} \alpha_i |i\rangle, \quad (4-3)$$

onde  $|\alpha_i|^2$  é a probabilidade de que o *qudit* irá colapsar para o estado  $i$  quando for observado. As probabilidades de todos os estados num *qudit* são as mesmas no início da evolução, exceto para a instrução NOP (a qual depende de um parâmetro de execução do programa).

Por exemplo, suponha que cada instrução na Tabela 4.1 tenha um único valor de token em  $T = \{0,1,2,3,\dots\}$ . A Equação (4-4) fornece o estado de um

*qudit* de função (QF), o qual tem seu estado dado por:

$$|\psi\rangle = \frac{1}{\sqrt{5}} |0\rangle + \frac{1}{\sqrt{4}} |1\rangle + \frac{1}{\sqrt{10}} |2\rangle + \frac{1}{\sqrt{8}} |3\rangle + \dots \quad (4-4)$$

A probabilidade de medir o estado  $|0\rangle$  (instrução NOP) é  $(1/\sqrt{5})^2 = 0.200$ , para o estado  $|1\rangle$  (instrução FADD  $m$ ) é  $(1/\sqrt{4})^2 = 0.250$ , para o estado  $|2\rangle$  (instrução FADD ST(0), ST(i)) é  $(1/\sqrt{10})^2 = 0.100$ , e assim por diante para os demais casos. O *qudit* de estado deste exemplo é implementado numa estrutura de dados como mostrado na Figura 4.1.

$p_0 = 0,20$	$p_1 = 0,25$	$p_2 = 0,10$	$p_3 = 0,125$	$\dots$
Instrução <sub>0</sub>	Instrução <sub>1</sub>	Instrução <sub>2</sub>	Instrução <sub>3</sub>	$\dots$
$I_0$	$I_1$	$I_2$	$I_3$	
[0,00;0,20]	[0,20;0,45]	[0,45;0,55]	[0,55;0,675]	

Figura 4.1: Ilustração de uma implementação de *qudit* que representa a Equação (4-4). O *qudit* representa a superposição de todas as possíveis instruções. Cada instrução representa um estado e tem uma probabilidade de ocorrer,  $p_i$ . A soma de todas as probabilidades é dada por  $p_0 + p_1 + p_2 + p_3 + \dots + p_n = 1.0$ . O estado de uma instrução é representado por um intervalo fechado de probabilidades  $I_i$ , o qual descreve uma probabilidade acumulada. O processo de observação gera um número aleatório  $r$  e seleciona um token baseado no intervalo de probabilidade no qual  $r$  se encaixa.

A Figura 4.2 ilustra a criação de um gene clássico através da observação de um gene quântico de um exemplo baseado na Tabela 4.1 e no vetor de entradas  $I = (V[0], V[1], 1, 2, 3)$  (Equação (4-1)). Este processo pode ser explicado através de três passos básicos, indicados pelos números circulos na Figura 4.2:

1. O QF é observado, e o valor resultante (por exemplo: 7) é atribuído ao TF deste gene.
2. O valor do TF determina o *qudit* de terminal (QT) que será observado, uma vez que cada instrução requer um tipo diferente de terminal: registrador ou memória.
3. O QT definido pelo valor do TF é observado e o valor resultante (por exemplo, 1) é atribuído ao TT deste gene.

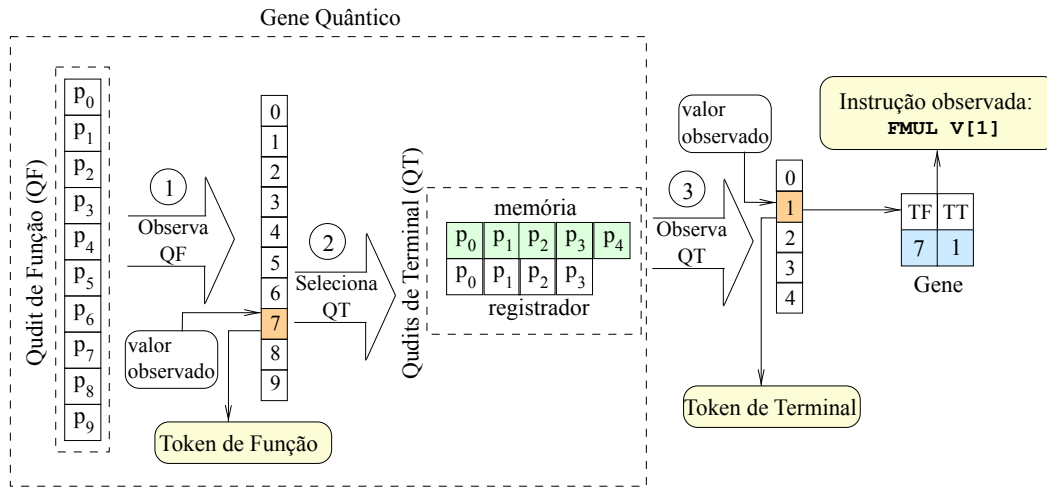


Figura 4.2: Criação de um gene clássico a partir da observação de um gene quântico. O QF é observado e o valor de token selecionado é 7. O *qudit* de memória é selecionado no QT. O QT é observado e o valor do TT selecionado é 1. A instrução observada neste exemplo é FMUL V[1], uma vez que ‘7’ é o valor do TF para esta instrução (Tabela 4.1), e ‘1’ é o valor do TT que representa V[1] na posição I do vetor de entradas definido pela Equação (4-1).

#### 4.4 Avaliação de um Indivíduo Clássico

O processo de avaliação de um indivíduo clássico se inicia com a geração de um programa em código de máquina a partir do indivíduo clássico a ser avaliado. O seu cromossomo é varrido sequencialmente, gene por gene e *token* por *token* (ambos os TFs e TTs), para serialmente gerar o corpo do programa em código de máquina relacionado com o indivíduo clássico. Então, o programa é executado para todas as amostras de dados do problema (isto é, as amostras do conjunto de treinamento).

Para cada amostra de dados, o valor atribuído como resultado da avaliação desta amostra é zero ( $V[0] \leftarrow 0$ ) quando a instrução FDIV realiza divisão por zero ou a instrução FSQRT realiza o cálculo de uma raiz quadrada de um número negativo.

#### 4.5 Operador Quântico

QILGP não utiliza os operadores de mutação e cruzamento dos algoritmos genéticos convencionais para criar novas populações de indivíduos. Ao invés disso é utilizado o operador quântico  $P$ . O operador quântico da QILGP manipula a probabilidade  $p_i$  de um *qudit*, satisfazendo a condição de normalização  $\sum_{i=0}^{d-1} |\alpha_i|^2 = 1$ , onde  $d$  é a cardinalidade do *qudit* e  $|\alpha_i|^2 = p_i$ . O

operador  $P$  trabalha em dois passos principais. Primeiro, ele incrementa uma dada probabilidade de um *qudit* da seguinte forma:

$$p_i \leftarrow p_i + s \times (1 - p_i), \quad (4-5)$$

onde  $s$  é um parâmetro chamado de **tamanho do passo**, o qual pode assumir qualquer valor real entre 0 e 1. O segundo passo é ajustar os valores de todas as probabilidades daquele *qudit* para satisfazer a condição de normalização. Então, o operador modifica o estado de um *qudit* incrementando  $p_i$  por um valor que é diretamente proporcional a  $s$ . O comportamento assintótico de  $p_i$  na equação do operador  $P$  indica que a probabilidade nunca atinge o valor unitário. Evitar probabilidades unitárias é uma importante característica deste operador, uma vez que isto evita o colapso da probabilidade do *qudit*, o que poderia causar uma convergência prematura do processo de busca evolutiva.

QILGP tem uma população híbrida, composta por uma população quântica e por uma população clássica, ambas as quais possuem  $M$  indivíduos. QILGP também tem  $M$  indivíduos clássicos auxiliares  $C_i^{obs}$ , os quais resultam das observações dos indivíduos quânticos  $Q_i$ , onde  $1 \leq i \leq M$ .

## 4.6

### Algoritmo Evolutivo

O algoritmo evolucionário da QILGP segue cinco passos, como mostrado na Figura 4.3. Inicialmente, a população de indivíduos quânticos é criada. O Tamanho da População  $M$  e o Comprimento Máximo do Programa são parâmetros de entrada do algoritmo. A cardinalidade do *qudit*  $d$  é o número de instruções do conjunto de funções. As probabilidades de todos os estados num *qudit* são as mesmas no início da evolução, exceto para a instrução NOP (a qual depende de um parâmetro). O algoritmo procede realizando os seguintes passos iterativamente até que um critério de parada seja atingido (por exemplo, um número predefinido de gerações, tempo computacional máximo, etc.):

1. Cada um dos  $M$  indivíduos quânticos é observado uma vez, resultando em  $M$  indivíduos clássicos  $C_i^{obs}$ .
2. Os  $M$  indivíduos clássicos  $C_i^{obs}$  são executados e seus resultados são avaliados de acordo com a função de avaliação da aptidão definida.
3. Os  $M$  indivíduos da população clássica são agrupados com os  $M$  indivíduos clássicos observados (auxiliares)  $C_i^{obs}$  e em seguida ordenados pelas suas avaliações, do melhor para o pior, de  $C_0$  para  $C_{M-1}$ .
4. O operador  $P$  é aplicado a cada indivíduo quântico  $Q_i$ , tomando para isso o seu indivíduo correspondente  $C_i$  na população clássica como referência.

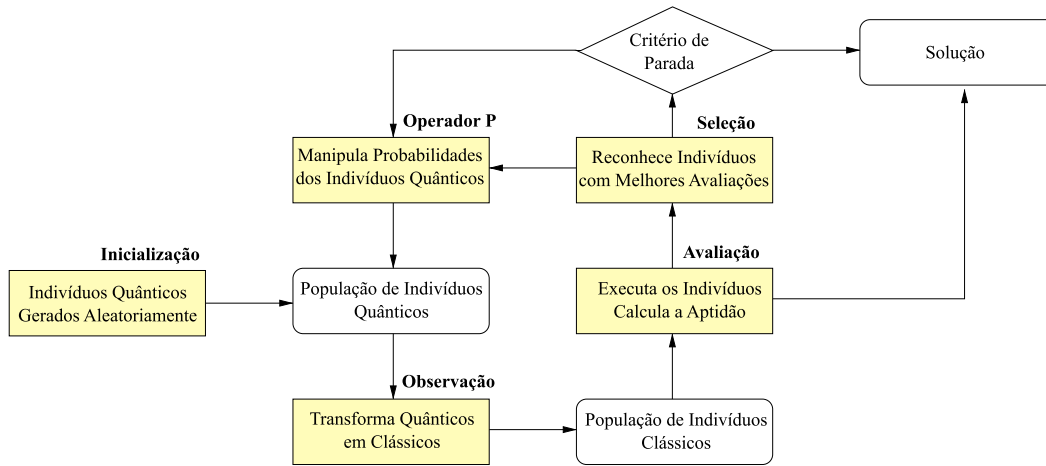


Figura 4.3: Algoritmo evolucionário da QILGP.

Então, a cada nova geração, a aplicação deste operador aumenta a probabilidade de que as observações dos indivíduos quânticos gerem indivíduos clássicos mais similares aos melhores indivíduos encontrados até então.

5. O melhor indivíduo clássico avaliado na geração corrente é armazenado como sendo a solução do problema, se ele for melhor do que o melhor indivíduo clássico encontrado nas gerações anteriores, utilizando como heurística para a comparação dos indivíduos, os seus resultados obtidos para a Função de Avaliação da Aptidão. A qual poderia ser, por exemplo, para alguns estudos de caso, o Erro Absoluto Médio (MAE) no problema de regressão simbólica *Salutowicz* [68].