

**Resumo do Evoluindo código de máquina  
de GPU de Cleomar Pereira da Silva,  
Douglas Mota Dias, Cristiana Bentes,  
Marco Aurélio Cavalcanti Pacheco e  
Leandro Fontoura Cupertino**

*Fábio Moreira Duarte*

## Sumário

1	Introdução . . . . .	3
2	Computação quântica e Algoritmos inspirados quantificamente .	4
2.1	Sistemas quântico multinível . . . . .	4
3	Programação genética Linear inspirada quântica . . . . .	4
3.1	Representação . . . . .	4
3.2	Observações . . . . .	4
3.3	Avaliação de indivíduos clássicos . . . . .	4
3.4	Operador quantum . . . . .	5

## Resumo

Implementações da GPU (Parallel Graphics Processing Unit) de GP apareceram na literatura utilizando três metodologias: (I) compilação, gerando os indivíduos no código da GPU e requisitos da compilação; (II) pseudo-assembly, gera os indivíduos em código assembly intermediário e requisitos de compilação; (III) interpretação, onde interpreta o código. O artigo propõe uma tecnologia que utiliza o conceito de computação quântica e lidagem direta das instruções do código da GPU da máquina. A metodologia utiliza uma representação probabilística de um indivíduo para aprimorar a capacidade de busca global. A evolução do código da máquina, limita a sobrecarga da compilação do código e o custo de análise do programa durante a avaliação.

## 1 Introdução

Programação genética (GP) é método heurístico para programas de computador gerado automaticamente ou subcomponentes-chaves. Funcionalidade é baseada nos princípios darwinianos de seleção natural, onde uma população de programas de computador ou indivíduos é mantida e modificada baseada em variações genéticas. Os indivíduos são avaliados de acordo com o função fitness para conseguir melhores soluções. GP foi aplicada com sucesso em vários problemas. O processo de avaliação consome tempo. O poder computacional requerido para GP é enorme, e técnicas de alta performance são utilizadas para reduzir o tempo computacional. Paralelização de GP pode ser explorada em dois níveis: múltiplos indivíduos podem ser avaliados simultaneamente, múltiplos casos de fitness para um indivíduo podem ser avaliados em paralelo.

O surgimento de computação geral em unidades de processamento gráfico (GPUs) proporcionou a oportunidade para acelerar a execução de algoritmos custosos.

Divide o esforço em três principais metodologias:

1. Compilação: cada programa evoluído, ou GP individual, é compilada para código de máquina GP e avaliada em paralelo na GPU.
2. Pseudo-assembly: os individuais são gerados no pseudo código da GPU, e compilação just in time é realizada para cada indivíduo.
3. Interpretação: executa os programas.

Propõe-se uma metodologia para uso de GPUs no processo de evolução de GP. Utiliza um algoritmo evolutivo inspirado quanticamente (QEA) lidando com as instruções de máquina da GPU.

A metodologia é chamada de programação genética em código de máquina GPU, GMGP, baseada em programação genética linear (LGP).

Compara-se a metodologia inspirada em quantificação com as tentativas para acelerar GP utilizando GPUs.

## 2 Computação quântica e Algoritmos inspirados quanticamente

Em computadores clássicos um bit pode assumir valores de 0 ou 1. Em computadores quânticos o bit chama qubit pode assumir os valores de 0, 1 ou sobreposição dos 2.

Sobreposição de estados providos por computadores quânticos com grau de paralelismo. O paralelismo quando explorado permite computadores realizarem tarefas inviáveis a computadores clássicos.

### 2.1 Sistemas quântico multinível

Muitas abordagens de computação quântica utiliza codificação qubits em sistemas quantitativos de dois níveis. Candidatos a sistema tem estrutura física complexa, com acessíveis níveis de liberdade.

## 3 Programação genética Linear inspirada quântica

A proposta é baseada em programação genética linear inspira quântica (QILGP).

As instruções são representadas por dois tokens: A função token (FT), e token terminal (TT), cada função tem um terminal. Quando a função não possui terminal, o valor do token é ignorado. Cada token é um valor inteiro que apresenta o índice do conjunto de funções no conjunto de terminais.

### 3.1 Representação

QILGP é baseada nas seguintes entidades: O indivíduo quântico, representa a superposição dos possíveis programas para o espaço de busca definido, e indivíduos clássicos, representam o código de máquina programado nos valores do token.

### 3.2 Observações

O cromossomo do indivíduo quântico é representado pela lista de estruturas denominadas genes quânticos. O indivíduo quântico compreende a observações de todos genes do cromossomo. O processo de observação consiste de geração aleatória de valores  $r$   $0 \leq r \leq 1$  e busca por intervalo  $r$  pertence a todos estados possíveis que o indivíduo representa.

### 3.3 Avaliação de indivíduos clássicos

O processo começa com geração de programas de código máquina do indivíduo clássico sobre avaliação, onde os cromossomos são sequencialmente atravessado, gene por gene e token por token, para gerar em série o corpo do programa do código de máquina relatado para o indivíduo clássico.

### 3.4 Operador quantum

O operador quantum de QILGP manipula a probabilidade  $p_i$  de qudit, satisfazendo a normalização somatório de  $i=0$  a  $d-1 \bmod \lambda$  elevado a 2 igual a 1. Primeiro passo é incrementar a probabilidade de qudit. Segundo é ajustar o valor de todas as probabilidades de qudit para satisfazer a condição de normalização.

QILGP tem uma população híbrida, por população quântica e população clássica, ambas com  $m$  indivíduos.