

# 1 Introdução

## 1.1 Motivação

A Computação Evolutiva ou Evolucionária (CE) consiste no projeto e/ou na análise de algoritmos estocásticos inspirados em princípios da seleção natural de Darwin e suas variações [1]. Segundo este princípio, uma população de indivíduos evolui, ao longo de gerações ou ciclos, pela sobrevivência dos mais aptos. Por sua vez, os algoritmos que compõem a CE são denominados Algoritmos Evolutivos (AEs). Há diferentes áreas de pesquisa em AEs, tais como Algoritmos Genéticos (AGs) [2], Estratégias Evolutivas [3], Programação Evolutiva [4] etc. Uma abordagem comparativamente recente neste contexto é a Programação Genética (PG).

Em sua definição mais simples, a Programação Genética é uma técnica que permite que computadores resolvam problemas sem que precisem ser explicitamente programados para tal [5]. Ou seja, a PG parte de uma declaração de alto nível sobre “o que se necessita ser feito” e cria automaticamente um programa de computador para resolver o problema. Este mecanismo também é conhecido como “programação automática”.

Diferentes tipos de PG existentes evoluem diferentes tipos de programas e os representam por diferentes tipos de estruturas. Este trabalho se baseia na Programação Genética Linear, que possui esta denominação porque seus programas são representados por estruturas lineares (listas). A PG Linear é utilizada para a evolução de programas em linguagens imperativas, como por exemplo, C [6] e linguagem de montagem [7], onde as listas são constituídas por uma sequência de instruções, ou de programas em código de máquina, onde então tais listas podem conter o próprio código de máquina.

Portanto, atualmente entende-se por PG toda forma evolutiva de indução de programas [8]. Ou seja, como linha de pesquisa, a PG compreende qualquer técnica de CE que forneça soluções sob a forma específica de programas de computador, independentemente da linguagem ou da forma de representação destes programas.

Desde sua concepção [9, 10] e formalização [5], entre as décadas de 1980 e 1990, a PG vem sendo utilizada para resolver uma ampla variedade de problemas

práticos, produzindo resultados competitivos com os de especialistas humanos, incluindo novas descobertas científicas e invenções patenteáveis [11]. Como diversas outras áreas da Ciência da Computação, a PG vem evoluindo rapidamente, com novas ideias, técnicas e aplicações sendo constantemente propostas [12].

Entretanto, os AEs apresentam, em algumas ocasiões, características que podem prejudicar o seu desempenho. Como estes algoritmos necessitam avaliar diversas vezes a qualidade das soluções encontradas, os problemas onde esta avaliação seja computacionalmente custosa podem tornar proibitivo o uso de tais algoritmos. Além disso, e ainda no contexto do custo computacional, a PG apresenta um problema particular denominado “inchaço” (*bloat*), que se caracteriza pelo rápido aumento médio do tamanho dos programas ao longo da evolução, sendo que, tipicamente, este aumento de tamanho não é acompanhado por uma melhoria correspondente na qualidade das soluções [12]. Portanto, uma vez que cada avaliação de cada programa (solução) implica na sua execução completa por diversas vezes, o crescimento médio dos indivíduos pelo efeito de inchaço implica em um rápido aumento do custo computacional total do algoritmo ao longo da sua execução.

Neste sentido, os algoritmos evolutivos com inspiração quântica (AEIQs) representam um dos mais recentes avanços na área de CE. Estes algoritmos se baseiam em ideias inspiradas na mecânica quântica, em particular no conceito de superposição de estados, apresentando melhor desempenho em diversos tipos de aplicações. Mais especificamente, o AEIQ usando representação binária (*Quantum-Inspired Evolutionary Algorithm* – QEA) [13] foi utilizado com sucesso em problemas de otimização combinatória, apresentando resultados superiores em relação aos AGs convencionais em termos de tempo de convergência, ou seja, um menor número de avaliações necessárias para se atingir os mesmos resultados. O AG com inspiração quântica para otimização do problema do caixeiro viajante [14] também apresenta resultados promissores quando comparado com os AEs convencionais. Em [15] é apresentado um AEIQ para otimização numérica, inspirado no princípio de múltiplos universos da física quântica e com representação de números reais, que apresenta um tempo de convergência menor para problemas *benchmark* quando comparado com algoritmos convencionais em [16].

Portanto, a partir de diversos exemplos de trabalhos desenvolvidos nesta área, inclusive os supracitados, pode-se notar que a aplicação do paradigma da inspiração quântica melhorou o desempenho de diferentes modelos de AEs. Entretanto, a despeito de tal sucesso, ainda não foi proposto nenhum AEIQ cujo objetivo seja o mesmo da PG: a criação automática de programas de computador.

## 1.2 Objetivos

Sendo assim, com base na discussão anterior, o principal objetivo deste trabalho é propor um novo modelo de AE, inspirado no conceito de superposição de estados da física quântica, para a síntese automática (evolução) de programas de computador, denominado “Programação Genética Linear com Inspiração Quântica” (PGLIQ). A “Programação Genética Linear” é assim denominada porque cada um dos seus indivíduos é representado por uma lista de instruções (estruturas lineares), as quais são executadas sequencialmente. Desta forma, também se pretende investigar o quão significativamente a aplicação do paradigma da inspiração quântica é capaz de melhorar o desempenho do processo de busca da PG por soluções, mais especificamente, em problemas de regressão simbólica e de classificação binária (vide seção 4).

Uma vez que diferentes modelos de PG existentes evoluem diferentes tipos de programas, optou-se pela evolução de programas em código de máquina para o modelo aqui proposto. O modelo de PG Linear convencional mais bem sucedido na evolução de programas em código de máquina denomina-se AIMGP (*Automatic Induction of Machine Code by Genetic Programming*) [17] e, portanto, é utilizado como referência para fins de comparação neste trabalho.

Uma das principais motivações para se decidir pela evolução de programas em código de máquina reside no fato de que, em termos de velocidade total de execução, a evolução de programas diretamente em código de máquina é a abordagem de PG mais rápida existente [8]. Tal velocidade advém do processo de avaliação de cada um dos muitos indivíduos, o qual é efetuado através da execução direta do indivíduo pelo microprocessador, sem a necessidade de etapas intermediárias e computacionalmente custosas, como compilação ou interpretação. Uma vez que um número elevado de avaliações é efetuado ao longo do processo evolutivo, pode-se perceber o impacto significativo deste tipo de avaliação na redução do tempo total de execução da PG.

Uma vez que todo processo de desenvolvimento de um novo modelo de AE demanda uma quantidade elevada de experimentos, seja para ajustes de parâmetros ou para comparações, a velocidade de execução desta abordagem de PG é uma característica útil para este trabalho. Ou seja, o fato dos modelos PGLIQ e AIMGP evoluírem programas em código de máquina representa um aspecto positivo de grande relevância para o desenvolvimento e avaliação do modelo proposto, uma vez que esta abordagem possui execução mais rápida que as demais abordagens de PG, o que possibilita um número maior de experimentos em um tempo menor.

### 1.3

#### Contribuições

Uma das contribuições deste trabalho é o desenvolvimento de um novo modelo de AEIQ para a síntese automática de programas de computador, ou seja, um novo modelo de PG que se diferencia dos demais existentes pelo fato da sua idealização ser inspirada em fenômenos da física quântica. Algumas das principais características pretendidas para este modelo são:

- Rápida convergência – Como a avaliação das soluções potenciais é, em geral, a responsável pela maior parte do tempo computacional gasto com o processo de busca, deseja-se que o novo modelo seja capaz de encontrar as soluções com um número menor de avaliações da função objetivo do problema em questão, quando comparado ao modelo AIMGP.
- Escalabilidade – Objetiva-se que o modelo seja robusto com relação à dimensionalidade do problema, ou seja, que o aumento do número de variáveis de entrada do problema não acarrete em um aumento exponencial do esforço computacional para a obtenção de soluções satisfatórias.
- Parametrização simplificada – Objetiva-se que o modelo possua poucos parâmetros a serem ajustados para se obter resultados satisfatórios, em comparação com o modelo de clássico de referência. Também é importante que a configuração dos valores destes parâmetros seja simples e que os mesmos variem o mínimo possível entre diferentes problemas.

Outra contribuição deste trabalho, que é uma consequência da anterior, diz respeito a uma avaliação inédita sobre o ganho de desempenho obtido ao se aplicar o paradigma da inspiração quântica na evolução de programas, além de apontar aspectos positivos e negativos do uso de tal paradigma em relação à abordagem clássica. Desta forma, abre-se a possibilidade para o desenvolvimento de outros AEIQs que visem à síntese automática de outros tipos de programas a serem aplicados nos mais diversos tipos de problemas, tais como: ajuste de curvas, modelagem de dados, regressão simbólica, síntese de circuitos analógicos e digitais, processamento de imagens e de sinais, controle de processos industriais etc. Em [12], pode-se encontrar uma listagem ampla e detalhada destes e de outros tipos de problemas abordados por PG, assim como suas respectivas referências bibliográficas.

## 1.4

### Descrição do Trabalho

O desenvolvimento deste trabalho seu deu nas seguintes etapas:

- Estudo de AEIQs existentes. Esta etapa consistiu na pesquisa e estudo de alguns AEIQs existentes, de forma a compreender, sob uma análise mais prática e funcional, como e por que o paradigma da inspiração quântica tem melhorado os diferentes tipos de AEs onde tem sido aplicado. O QEA [13] foi estudado de forma detalhada, uma vez que este modelo foi elucidativo com relação à possibilidade de se aplicar o fenômeno da superposição de estados no modelo aqui proposto.
- Estudo de tipos de PG existentes. Nesta etapa, foram analisados os aspectos positivos e negativos de cada tipo de PG quanto ao desempenho e, principalmente, quanto aos tipos de estruturas utilizados na representação dos programas. Esta etapa foi decisiva para a definição do tipo de PG mais adequado ao novo modelo proposto.
- Definição do modelo inicial da PGLIQ. Como consequência da pesquisa realizada nas etapas anteriores, foi idealizado um modelo inicial para PGLIQ que, por sua vez, consistiu em definir: os tipos de estruturas a serem utilizados na representação dos programas, tanto no domínio quântico quanto no clássico; um operador evolutivo com inspiração quântica; o funcionamento do algoritmo evolutivo em si, como um todo.
- Implementação do modelo da PGLIQ. Nesta etapa, foi desenvolvido um sistema para implementar o modelo inicial definido na etapa anterior. Ao longo do desenvolvimento do sistema, o modelo foi sendo aperfeiçoado, conforme novas heurísticas eram criadas e modificadas, em um ciclo iterativo envolvendo testes, avaliações e novas alterações. Tal processo de desenvolvimento culminou na versão atual do modelo empregado neste trabalho. Este sistema foi desenvolvido em duas linguagens simultaneamente: C e C# (*C Sharp*), utilizando-se o ambiente integrado de desenvolvimento Microsoft Visual Studio em um computador de plataforma Intel (PC). O núcleo evolutivo do modelo foi desenvolvido em C, enquanto que os processos relativos a entrada e saída de dados (leitura e escrita em arquivos) foram desenvolvidos em C#. Esta abordagem permite conjugar a velocidade de execução da programação em C com a velocidade de desenvolvimento da linguagem C#. Visando a agilizar e flexibilizar a parametrização do modelo da PGLIQ e a definição dos experimentos, o sistema é configurado a partir da leitura de um arquivo no formato XML.

- Avaliação de desempenho do modelo. Esta etapa teve como objetivo conduzir a avaliação de desempenho do modelo implementado na etapa anterior. Para tal, foram executados experimentos comparativos entre os modelos PGLIQ e AIMGP. Estes experimentos envolveram quatro problemas de regressão simbólica e dois de classificação binária, uma vez que estes tipos de problema, por sua relevância em aplicações reais, são os mais comumente utilizados para se avaliar o desempenho de sistemas de PG em geral.

## 1.5

### Organização do Trabalho

O restante do trabalho está dividido em quatro capítulos adicionais. No capítulo 2 é apresentado um resumo dos fundamentos teóricos necessários para a compreensão deste trabalho, onde são abordados assuntos que vão de mecânica quântica e computação quântica, passando por algoritmos evolutivos com inspiração quântica, até programação genética e, mais especificamente, programação genética linear. Já no capítulo 3, o modelo proposto de programação genética linear com inspiração quântica é apresentado em detalhes, começando pela descrição da sua plataforma de execução, da representação dos seus indivíduos e programas evoluídos, além da forma de avaliação destes programas. Também apresenta o operador quântico proposto e finaliza descrevendo a estrutura geral e o funcionamento do modelo, assim como seu algoritmo evolutivo. O capítulo 4 mostra e analisa os resultados obtidos a partir dos estudos de caso, compostos por problemas de regressão simbólica e de classificação binária, além de comparar estes resultados aos obtidos pelo modelo clássico de referência. Finalmente, o capítulo 5 apresenta discussões e conclusões sobre o modelo e os resultados obtidos, além de apontar direções para trabalhos futuros.