

1. Introdução

1.1. Motivações

Dentre os tipos de técnicas de Inteligência Artificial existentes, as técnicas de Programação Genética (PG) continuam mudando rapidamente conforme os pesquisadores e profissionais desenvolvem novos métodos e aplicações [1]. Estas técnicas se baseiam geralmente numa representação de estrutura de árvore, cujos elementos guardam entre si uma relação hierárquica. Outras representações da PG foram propostas e implementadas com sucesso, como a Programação Genética Linear (PGL), que é mais adequada à síntese de programas em linguagens imperativas tradicionais, como no modelo AIMGP (Automatic Induction of Machine Code by Genetic Programming), considerado atualmente o modelo de PG mais eficiente na evolução (síntese) de código de máquina [2].

A Programação Genética Linear com Inspiração Quântica (PGLIQ) é outra das técnicas de PG inovadoras. Ela combina vantagens da PGL com as de algoritmos evolutivos com inspiração quântica, particularmente no princípio de superposição de estados, apresentando um desempenho geral superior em problemas de regressão simbólica e classificação binária. Isso se torna possível ao se encontrar melhores soluções (menores erros) a partir de um número menor de avaliações, com a vantagem adicional de utilizar um número menor de parâmetros e operadores que o modelo AIMGP [3].

A Inteligência Artificial está sempre trabalhando em conjunto com a indústria da computação, seja para PG (nosso caso de estudo) ou outras técnicas que precisem de um alto poder computacional para sua execução. Esta indústria da computação está se movendo cada vez na direção do maior desempenho, tanto para CPU, quanto para GPU. As GPUs de hoje podem processar enormes quantidades de dados e são usadas não apenas para renderização de cenas 3D, mas também para o processamento de imagens e realização de computação maciçamente paralela, tais como em estatísticas financeiras ou análise de terreno para encontrar novos campos

de petróleo. Um novo domínio chamado “computação de propósito geral em processadores gráficos” (GPGPU) surgiu quando os *pipelines* de gráficos de funções fixas começaram a evoluir para *pipelines* programáveis em tempo real.

Embora o critério para a seleção das ferramentas de alto desempenho normalmente dependa de cada problema, as GPUs se destacam por sua inerente arquitetura *Single Instruction Multiple Threads* (SIMT) e acesso rápido à memória do dispositivo *on-board* para alcançar melhor desempenho em termos de aceleração, em comparação com os sistemas multi-processadores e *multi-core* convencionais.

Além disso, os *clusters* de GPU têm o potencial de reduzir significativamente o espaço, energia e refrigeração, demandas relativas aos *clusters* de CPU tradicionais com capacidade computacional elevada. O segundo supercomputador mais rápido do mundo, Titan, com um desempenho máximo teórico de 27 *petaflops* por segundo, utiliza NVIDIA K20X GPUs como aceleradores de *hardware* [4]. Entretanto, uma das claras desvantagens que as GPUs apresentam é a transferência de dados GPU-CPU e vice-versa, especialmente se grandes quantidades de dados são transferidas em cada instante, além de ser amplamente aceito que a programação da GPU é mais difícil do que a programação da CPU. Para obter melhor desempenho e gerenciar os diferentes processos que envolvem a paralelização heterogênea da PGLIQ, cria-se uma estrutura de alto desempenho onde múltiplas GPUs e múltiplas CPUs interagem, formando vários níveis de paralelismo. Desta forma, elas executam diferentes tarefas na GPU e CPU e ao mesmo tempo têm uma aceleração escalável.

Conjuntamente ao paralelismo, a implementação de técnicas de visualização gráfica na representação dos padrões do algoritmo permite compreender, de uma forma potencialmente rica, a dinâmica que sustenta as técnicas de PG. As animações 3D destacam-se da mesma forma representando o desempenho do algoritmo em processos evolutivos, como em aprendizagem de robô, estudo de caso nesta dissertação. Em um estudo de comparação realizado por [5], entre as bibliotecas *Bullet*, ODE and *PhysX*, conclui-se que a biblioteca *Bullet* é o melhor motor de física devido a seus bons resultados entre as várias avaliações realizadas, especialmente pelo seu preciso modelo de atrito. *Bullet* foi utilizado em conjunto com o motor gráfico OGRE encarregado da renderização, pelo suporte e compatibilidade entre eles [6].

1.2. Objetivos

O objetivo principal desta pesquisa é desenvolver uma topologia de alto desempenho (utilizando o paralelismo heterogêneo) para a Programação Genética Linear com Inspiração Quântica na evolução de processos com alto custo computacional. Assim, nesta dissertação, é analisado um estudo de caso central para a aprendizagem do andar de um robô hexápode. A topologia proposta melhora o desempenho da PGLIQ a partir de dois aspectos fundamentais de um algoritmo: a eficiência, reduzindo o tempo de processamento e acelerando a convergência na busca de soluções; e a eficácia, aumentando a qualidade dos resultados através de soluções de busca mais eficazes. Para reduzir o tempo de processamento, utilizam-se diversos recursos de alto desempenho (paralelismo heterogêneo), paralelizando as tarefas em arquiteturas híbridas *multi-core* e *many-core*, além de utilizar subpopulações isoladas de indivíduos, chamados *demes* (seção 2.2.2), nos quais o processo se torna ainda mais paralelizável. Outro objetivo é a construção de uma extensão PGLIQ com uma melhor qualidade dos resultados usando as ferramentas de visualização gráfica e representando os parâmetros da PGLIQ, que permitem entender o desenvolvimento ao longo das gerações e explicar as melhorias propostas na busca de soluções.

Os objetivos secundários desta pesquisa são:

- Paralelizar completamente a PGLIQ utilizando técnicas otimizadas para minimizar todo tipo de transferência entre GPU-CPU, acesso à memória global, entre outras.
- Analisar o comportamento dos *demes* em função da escalabilidade e apoio ao desempenho da PGLIQ.
- Definir tipos de visualização na qual se observa o comportamento do processo evolutivo da PGLIQ.
- Comparar estudos de casos com e sem a melhora da técnica PGLIQ.
- Implementar o sistema de visualização gráfica em tempo real de forma remota analisando o custo computacional do processamento da PGLIQ que é executado no *cluster*.

- Avaliar o desempenho da PGLIQ e utilizar o suporte de visualização gráfica remota em aplicações de robótica simulada.

Analisando a compatibilidade entre a programação CUDA e OpenCL [7], foi escolhida a linguagem de programação de médio nível C-CUDA, pela facilidade para atualizar para novas arquiteturas, que são principalmente lideradas pela NVIDIA, e, posteriormente, padronizar com OpenCL para diferentes tipos de placas gráficas.

1.3.

Descrição do Trabalho

Este trabalho foi elaborado através das seguintes etapas:

- Estudo sobre o modelo de inteligência computacional utilizado: Programação Genética Linear com Inspiração Quântica.
- Estudo de *demes* e as vantagens do paralelismo.
- Estudo sobre a ferramenta de paralelismo com Multi-GPU NVIDIA.
- Estudo sobre as ferramentas de alto desempenho: OpenMP (Open Multi-Processing), biblioteca de troca de mensagens openMPI (open Message Passing Interface) e sistemas híbridos OpenMP/MPI dentro do paralelismo heterogêneo, além da utilização do gerenciador de recursos Torque.
- Implementação de uma comunicação TCP/IP para transferência de informação entre um computador local e o *cluster*.
- Estudo da visualização científica e visualização de parâmetros da PGLIQ.
- Estudo de bibliotecas de simulação dinâmica e motores de renderizado.
- Análise dos estudos de casos.

Na primeira etapa do trabalho, foi feito um estudo sobre Programação Genética Linear com Inspiração Quântica, que é a técnica de inteligência computacional utilizada como análise central nesta pesquisa. Trata-se de um estudo que compreende os conceitos gerais de programação genética e algoritmos com inspiração quântica, bem como um estudo sobre o modelo com *demes* na programação genética. Também são estudadas as bibliotecas de alto desempenho OpenMP, OpenMPI, CUDA e técnicas otimizadas, criando um sistema híbrido que paraleliza os operadores da PGLIQ para formar estruturas matriciais dentro da GPU

e o modelo com *demes* em Multi-GPU e Multi-CPU criando um paralelismo heterogêneo. Finalmente será estudada a ferramenta para gerenciar recursos Torque.

Na implementação do sistema de Visualização em Tempo Real, foi criada uma biblioteca para transferências de dados a partir do *cluster* por protocolo TCP/IP e visualizado em um computador local.

Analisa-se o desempenho da PGLIQ baseado nos gráficos obtidos pela visualização e propõe-se uma extensão que seja mais eficaz na busca de soluções. É realizado também um estudo de bibliotecas para simulação dinâmica e renderização em simulações em robótica evolutiva, como a criação de um ambiente onde um robô possa evoluir ou realizar uma tarefa.

Na etapa de estudos de casos, foi realizada uma análise detalhada dos *benchmarks* de regressão simbólica desenvolvidos como casos referenciais pela PGLIQ, com e sem *demes*, bem como uma análise da PGLIQ em aplicações de robótica evolutiva, utilizando o sistema de transferência de dados em tempo real para visualizar a evolução do robô ao longo das gerações.

1.4. Organização do Trabalho

Esta dissertação contém cinco capítulos, cujos conteúdos estão descritos a seguir.

No Capítulo 2, faz-se uma introdução sobre todas as técnicas utilizadas para a conclusão do objetivo principal. Tais técnicas incluem Programação Genética, Programação Genética Linear, Programação Genética Linear com Inspiração Quântica, conceitos gerais da Visualização Gráfica e fundamentos da Computação de Alto Desempenho.

O Capítulo 3 tem como finalidade a análise da PGLIQ, começando pela configuração de diferentes estruturas, desde a criação dos operadores evolutivos dentro da GPU, até os operadores de migração utilizados em *demes*, os quais serão paralelizados em sistemas híbridos com interfaces de troca de mensagens e memória compartilhada junto aos operadores em GPU. Também são representados graficamente os parâmetros mais importantes da PGLIQ que visualizam o

comportamento geral da evolução. Nela é esquematizada uma extensão da PGLIQ proposta que melhora essencialmente a eficácia na busca de soluções.

O Capítulo 4 descreve os estudos de casos. Inicialmente, são analisados dois *benchmarks* de regressão simbólica, analisados previamente pela PGLIQ e comparados com a extensão proposta. Seu desempenho com *demes*, tempos de processamento e eficácia na busca de soluções do algoritmo são também analisados. Além disso, trata de um estudo de caso no qual se analisa seu desempenho em resolver casos em robótica evolucionária. Os resultados obtidos em ambos os casos são apresentados e discutidos.

Finalmente, no Capítulo 5, as conclusões são apresentadas e os possíveis direcionamentos para trabalhos futuros.