

Universidade de São Paulo
Instituto De Ciências Matemáticas e de Computação

Desenvolvimento de um Simulador de Inteligência de Enxames
Acelerado em Hardware

Prof. Dr. Eduardo do Valle Simões (Orientador)

Programa Unificado de Bolsas de Estudo Para Apoio e Formação de
Estudantes de Graduação — PUB/USP

Resumo

Esse projeto propõe o desenvolvimento de um sistema capaz de simular a interação entre diversos conceitos da área de Inteligência de Enxame, possibilitando a realização de experimentos com enxames de abelhas e colônias de formigas artificiais aplicados na solução de diversos problemas como exploração, forragiamento, busca e resgate, reposicionamento das colônias e otimização de caminhos. O simulador deve fornecer ao pesquisador uma interface amigável, na qual será possível planejar e configurar experimentos de simulação de centenas de milhares de indivíduos (insetos), possibilitando a visualização de suas interações com um ambiente gráfico acelerado em hardware que pode conter obstáculos, objetos de interesse, fontes de alimentos e sede das colônias. Essa interface também deverá viabilizar a configuração e a aplicação de um Algoritmo Evolutivo (AE) para otimizar os parâmetros de controle da população de agentes (formigas, abelhas e outros insetos) para resolver problemas e melhorar seu desempenho no desenvolvimento das tarefas alocadas. A implementação do sistema contará com a utilização de duas APIs (*Application Programming Interface*): OpenGL, uma API de computação gráfica usada para a parte visual da simulação, e CUDA, uma API para trabalho com computação paralela em placas de vídeo desenvolvidas pela Nvidia. Com a disponibilização de todo o software desenvolvido e seus manuais de instalação e operação em repositórios públicos como o Github, espera-se que o simulador proposto possa disponibilizar para cientistas de todo o Brasil uma ferramenta de fácil configuração para desenvolver experimentos acelerados em hardware contendo enxames de centenas de milhares de indivíduos.

1. Introdução e Justificativa

Com o desenvolvimento tecnológico de *drones* e robôs autônomos de pequeno porte, as técnicas de controle de grandes times de robôs aplicados em problemas reais que envolvem exploração, forragiamento, busca e resgate e otimização de caminhos, vem sendo repensadas para que permitir o emprego de centenas de indivíduos (Ko & Lau, n.d.). Neste contexto, a tecnologia dos Sistemas de Inteligência de Enxame vem abrindo espaço para novas abordagens no campo da robótica. A Inteligência de Enxame (IE), ou *Swarm Intelligence*, foi abordada por Craig W. Reynolds, em 1987, em seu artigo “*Flocks, Herds, and Schools: A Distributed Behavioral Model*” (Reynolds, 1987) no qual modelou um programa de vida artificial que ele chamara de “*Boids*”, termo proveniente do comportamento em bando dos pássaros. Esse sistema iniciara o estudo que atualmente se tornou um campo da inteligência artificial que

estuda sistemas descentralizados e auto-organizados pelo comportamento coletivo de seus elementos. Tal tecnologia é utilizada em problemas nos quais há uma grande quantidade de elementos a serem trabalhados, que possuem algum tipo de interconexão e soluções que exigem dinamicidade. Para controlar esse grande número de agentes autônomos são utilizadas tecnologias bioinspiradas, que são metodologias que adaptam processos biológicos para soluções computacionais, como em aplicações de Ant Colony Optimization (Dorigo et al., 2006), aplicações inspiradas no comportamento de abelhas Artificial Bee Colony (Yang et al., 2007), ou otimização via enxame de partículas (Kennedy & Eberhart, 1995), por exemplo.

Na última década, tecnologias como ACO, ABC, Particle Swarm Optimization, dentre outras derivadas dos enxames bioinspirados, vem crescendo em popularidade (J. Tang, 2021), sendo utilizadas em diferentes trabalhos de otimização. Isso posto, de fato é possível encontrar vários trabalhos recentes que se utilizam dessas técnicas para estudar possíveis soluções de otimização de problemas complexos, como na engenharia na área de gestão de recursos hídricos (M. Janga, 2020), na robótica com o planejamento de rotas para robôs (Diep, 2021), ou na área de segurança, com melhorias em sistemas de detecção de intrusos (Ankit Thakkar, 2020). No entanto, desenvolver experimentos com dezenas, centenas ou até milhares de pequenos robôs físicos, impõe várias dificuldades operacionais, como por exemplo, o manejo de uma grande quantidade de componentes eletrônicos, carregamento da bateria de cada um dos agentes do enxame, posicionamento e inicialização manual de cada robô no ambiente de testes ou atuação, entre outros problemas que podem surgir quando se trata de um grande volume de robôs autônomos.

Assim, para tais estudos, trabalhar em uma simulação é uma boa opção, uma vez que se abstrai a complexidade de lidar com robôs físicos, possibilitando a criação de simulações das mais variadas aplicações de comportamento de robôs autônomos em grupo. Nessas simulações, a virtualização do ambiente no qual os robôs seriam testados, bem como do hardware desses,

permite que testes sejam realizados com mais praticidade e rapidez, realizando alterações de parâmetros da simulação via aplicação ou de comportamento dos agentes via scripts como em (s. Lim, 2021), no trabalho chamado BeeGround, que gerou uma ferramenta de simulação gráfica de enxame de abelhas, a qual contava com uma população de 1000 robôs, viabilizando experimentos sobre o mecanismo de agregação dos agentes (abelhas).

Na bibliografia atual, nota-se uma incipiente procura por ferramentas, frameworks ou interfaces para se trabalhar com enxames, como ao simular enxames de abelhas, ou em (E. Soria, 2020) com o trabalho chamado SwarmLab, que buscou criar processos em MatLab para viabilizar a avaliação de enxames de drones. Ainda assim, nota-se a falta de um conjunto de recursos que possibilitem, num só lugar, a parametrização, simulação, otimização e avaliação de um enxame, que possivelmente se dá pela dificuldade em alinhar um grande número de agentes autônomos em simulação com o grande processamento e tempo demandado para simular e avaliar o comportamento desses.

A implementação de um grande número de robôs simulados acaba afetando a performance da máquina que hospeda a simulação, uma vez que, para cada agente, todas as ações devem ser executadas, as interações com o ambiente tratadas e sua performance no enxame armazenada e avaliada. Isso posto, percebe-se que o gargalo do número de agentes acaba limitando a atuação dos enxames nas simulações, como em (s. Lim, 2021) que contou com apenas 1000 abelhas, ou seja, caso o processamento seja otimizado a ponto de suportar um maior número de agentes, abre-se uma gama de novas possibilidades de simulação, novos tipos de enxames com mais agentes bem como a possibilidade da união entre simulação e avaliação e otimização de parâmetros dos agentes numa mesma ferramenta.

É nesse contexto que é proposta essa pesquisa, que visa o desenvolvimento de um simulador de Inteligência de Enxame acelerado em hardware para estudar a eficiência e a performance do sistema em relação às variações possíveis dos parâmetros, como a influência do número de

agentes, da quantidade de objetos de interesse a serem coletados, da interação entre os agentes do enxame, etc. Com a aceleração em hardware, espera-se que uma grande quantidade de agentes possa estar presente simultaneamente na simulação, sem que haja perda de performance do sistema, o que pode potencializar a ação do Enxame e possibilitar testes diversos, variando a quantidade de agentes e estudando suas interações.

Portanto, para viabilizar a utilização de tais sistemas e, principalmente estudar a influência de um grande número de agentes no enxame, a aceleração de hardware em placa gráfica pode ser uma boa alternativa para otimizar a performance do sistema ao transferir algumas operações custosas do software da CPU para a GPU. Então, espera-se que os múltiplos processadores da placa gráfica possam auxiliar com seu processamento paralelo, o que pode ser fundamental para processar toda a movimentação, colisão e reconhecimento do ambiente de cada agente simulado ao mesmo tempo.

Para isso, esse trabalho promoverá um estudo com três vertentes da computação: a computação gráfica com OpenGL (*The Industry's Foundation for High Performance Graphics*, n.d.), a aceleração de hardware em CUDA (*CUDA®*, n.d.) e a inteligência artificial (Algoritmos Evolutivos e Inteligência de Enxame). A integração dessas tecnologias resultará no desenvolvimento de um sistema de simulação gráfica, acelerado em hardware, que tem o objetivo de processar simultaneamente centenas, ou até milhares de agentes num enxame, sem perda de performance.

3. Objetivos

Esse projeto visa desenvolver uma ferramenta para possibilitar a realização de experimentos com inteligência de enxames e colônia de formigas artificiais aplicados a diversos problemas como exploração, forragiamento, busca e resgate, reposicionamento das colônias e otimização de caminhos. O sistema proposto deve ser capaz de simular a interação de diversos indivíduos

que se comportam de acordo com os conceitos de Inteligência de Enxame, possibilitando a realização de experimentos com enxames de abelhas e colônias de formigas artificiais aplicados a diversos problemas clássicos destas técnicas.

O simulador deve fornecer ao pesquisador uma interface amigável, na qual será possível planejar e configurar experimentos de simulação de centenas de milhares de indivíduos (insetos), possibilitando a visualização de suas interações com um ambiente gráfico acelerado em hardware que pode conter obstáculos, objetos de interesse, fontes de alimentos e sede das colônias. Essa interface também deverá viabilizar a configuração e a aplicação de um Algoritmo Evolutivo (AE) para otimizar os parâmetros de controle da população de agentes (formigas, abelhas e outros insetos) para resolver problemas e melhorar seu desempenho no desenvolvimento das tarefas alocadas. Em sistemas de enxames, o grande número de agentes necessários provoca grande demanda de execução e processamento, assim, com a aceleração na GPU (*Graphics Processing Unit*) espera-se que uma grande quantidade de agentes, cerca de 500 mil indivíduos, possam ser simulados ao mesmo tempo, o que pode solucionar o gargalo de processamento que muitos agentes provocam e também potencializar a ação do enxame. Um exemplo de possível aplicação da ferramenta de simulação proposta seria criar um experimento no qual uma população de formigas deverá coletar alimento para o formigueiro e, assim como na natureza, a simulação do AE deverá adaptar a inteligência coletiva de busca e resgate das formigas através da adaptação computacional da função de estigmergia (depósito de feromônios) das formigas.

A maior contribuição proposta neste trabalho deverá ser a disponibilização de todo o software desenvolvido e seus manuais de instalação e operação em repositórios públicos como o github e o gitlab. Espera-se que o simulador proposto possa disponibilizar para cientistas de todo o Brasil uma ferramenta intuitiva e de fácil configuração para desenvolver experimentos acelerados em hardware contendo enxames de centenas de milhares de indivíduos.

4. Metodologia

Propõe-se que um sistema com aceleração em hardware possa agregar novas abordagens para os sistemas de enxame, visto que, devido ao processamento paralelo das GPUs, uma quantidade maior de agentes poderão ser simulados simultaneamente, quando comparado a um sistema convencional limitado pelo processamento da CPU, o que pode acelerar e viabilizar possíveis testes em simulação, que antes demandariam muito tempo, energia e infraestrutura no ambiente real, com robôs físicos. Assim, serão possibilitados tanto estudos a respeito da importância da quantidade de agentes em simulação, quanto a modelagem de robôs com comportamentos complexos e em ambientes dinâmicos.

Com isso, esse projeto tem como objetivo o desenvolvimento de um sistema que simulará graficamente diferentes tipos de enxame, bem como a sua parametrização pré simulação e pré disposição de todos os componentes do próprio swam e do ambiente no qual ele será simulado. No início, o enxame a ser implementado será o inspirado em colônia de formigas, com posterior adição do enxame de abelhas e, possivelmente, a generalização para qualquer swam.

Os agentes do sistema baseado em ACO serão adaptações artificiais de formigas, que se moverão num ambiente terrestre e terão a sua estigmergia, ou seja, seu sistema de comunicação a base de feromônios, estudada e adaptada logicamente e graficamente para a simulação (da Silva et al., 2005). Essa comunicação será implementada como uma matriz de deposição de feromônio, representada por uma matriz de *pixels* que serão percebidos artificialmente pelas antenas das formigas que se sobrepuserem a eles.

O processamento da sobreposição das formigas com a matriz de feromônio talvez seja o processo mais custoso de todo o sistema, visto que a matriz deve ser atualizada e acessada por cada formiga a cada iteração do *looping* principal da simulação. Portanto, para que não haja uma queda de desempenho com o aumento do número de formigas em simulação, todo o

processamento dessa função deverá ser passada para a GPU, a fim de otimizar e amplificar a capacidade de processamento da simulação. A parte visual, com a representação gráfica na tela, será implementada em OpenGL, e a conexão com a placa de vídeo para o processamento lógico será feita através da API desenvolvida pela Nvidia CUDA.

Como a ferramenta proposta oferecerá suporte a tecnologias conjuntas pouco antes exploradas, integrando simulação gráfica do enxame, algoritmo evolutivo para sua otimização junto a uma única interface que gerenciará os parâmetros de ambas, contando com uma aparência intuitiva e prática, é necessário um estudo prévio de técnicas de desenvolvimento e design de interfaces gráficas. Em adição, como a ferramenta será disponibilizada para comunidade científica, problemas de usabilidade causados por um design mal pensado, assim como por um mal planejamento da experiência do usuário, como os citados em (Khairat, 2022), devem ser evitados. Ou seja, além da implementação e desenvolvimento prático da ferramenta, um estudo prévio de arquitetura e design de ferramentas de grande porte deve ser feito.

Tendo como possíveis materiais de apoio os livros "Game Engine Architecture", de Jason Gregory, e "C++ Templates: The Complete Guide", de David Vandevor, Nicolai M. Josuttis e Douglas Gregor, o aluno poderá apreender boas técnicas de desenvolvimento bem como arquitetar uma boa estrutura para a ferramenta a ser desenvolvida. Utilizando-se de bons templates para modularização de classes em c++, bem como de boa organização de código para integração com todas as APIs a serem utilizadas e abertura para possíveis contribuições open-source.

A princípio, a ferramenta deverá oferecer suporte a input de mouse e teclado, de forma que, antes de iniciar a simulação, o usuário possa selecionar a quantidade e posições iniciais de cada colônia, número de agentes, disposição do terreno e demais parâmetros que forem necessários para a simulação em questão, como obstáculos, fontes de interesse dos agentes, etc. Assim, a interface contará com botões e campos para input de dados, bem como funções de "segurar e

arrastar" para posicionamento dos componentes da simulação. Além disso, a ferramenta deverá oferecer suporte à inicialização via arquivos de input de parâmetros, como Jason por exemplo. Ainda antes da simulação ser iniciada, um quadro destinado aos algoritmos evolutivos deve conter as opções de parametrização do AE, bem como diferentes técnicas de crossover ou mutação, além da forma do cromossomo a ser avaliado, junto com diferentes técnicas e objetos parâmetros de avaliação.

Após iniciada a simulação, as estatísticas e atuais estados do enxame devem estar presentes na tela, junto a gráficos de desempenho e afins. A interface deve conter opções para pausar, interromper, resetar e reiniciar a simulação a qualquer momento, bem como fazer alterações em tempo real e aceitar novos inputs de objetos e valores de parâmetros. Em adição, como a ferramenta tem fins de instrumentalização científica, é interessante que a ferramenta suporte o armazenamento de todas as informações relevantes de um experimento, como estado inicial da simulação, estados dos agentes no decorrer da simulação, assim como resultados de avaliações de desempenho de cada enxame fornecidos pelo AE, como objetivo de reproduzir um experimento e gerar facilmente dados e estatísticas para estudos, artigos, etc.

Após a implementação do sistema, serão realizados testes de desempenho e integração dos módulos, visando a aceleração do processamento e fazendo os ajustes necessários em cada parte da simulação, ambiente, colônia de agentes, agentes, obstáculos, objetivos, etc. Por fim, deverá ser realizada a integração de todos os componentes do sistema, integrando OpenGL, C++ e CUDA e a interface gráfica (possivelmente feita com base na biblioteca ImGui – Disponível em: <https://github.com/ocornut/imgui>), buscando sempre a otimização do tempo de processamento das iterações do sistema e algoritmos.

Com o sistema finalizado, a fase de documentação deve ser realizada, visando disponibilizar todo o software desenvolvido e seus manuais de instalação e operação em repositórios públicos

como o github e o gitlab. Serão produzidos, juntamente com o relatório final deste projeto, artigos científicos para a divulgação das potencialidades do simulador proposto.

5. Detalhamento das atividades a serem desenvolvidas pelo bolsista

O aluno irá desenvolver diversas atividades, partindo do aperfeiçoamento das tecnologias do sistema e APIs, seguido da busca por uma bibliografia atualizada a respeito do tema, desenvolver o sistema com todos os componentes propostos, realizar testes e, por fim analisá-los a fim de buscar melhorias. Assim, o aluno terá as seguintes tarefas:

1 Revisão bibliográfica sobre Inteligência de Enxame, Algoritmos Evolutivos e criação de interface gráfica para simulações.

2 Estudo da API de computação Gráfica OpenGL para a parte gráfica da simulação, bem como a otimização desta em GPU, buscando funções otimizadas e eficientes para grandes simulações.

3 Estudo da API CUDA de aceleração de hardware, para a realização dos cálculos necessários para a simulação da população de agentes de simulação.

4 Projeto da arquitetura e desenvolvimento do sistema que integrará a simulação de enxames de formigas ou abelhas, ao suporte a algoritmos evolutivos, unidos pela interface gráfica de gerenciamento.

5 Implementação de um Algoritmo evolutivo para otimização dos enxames, bem como a possível personalização do mesmo.

6 Integração de todas as tecnologias estudadas e desenvolvidas, para desenvolvimento do sistema final.

7 Realização de testes de desempenho do enxame em diferentes cenários e situações, análises de melhorias.

8 Modularização e generalização da simulação para qualquer tipo de agente e enxame, bem como a generalização da aplicação dos algoritmos evolutivos possibilitando a fácil personalização do mesmo.

9 Escrita dos manuais de instalação e operação do software, do relatório final e de artigos para a divulgação dos resultados.

6. Cronograma de execução

Na Tabela 1 apresenta-se o cronograma de 12 meses com o planejamento das atividades da pesquisa.

Atividade	1 Bim	2 Bim	3 Bim	4 Bim	5 Bim	6 Bim
1	•	•	•			
2	•	•	•			
3	•	•				
4	•	•				
5		•	•	•	•	
6			•	•	•	
7				•	•	•
8					•	•
9						•

Tabela 1: Cronograma de Atividades

7. Resultados previstos e seus respectivos indicadores de avaliação

Resultado	Indicadores de avaliação
Interface com estrutura principal da ferramenta, ambiente, controle de fluxo, controle de arquivos, etc.	Teste da ferramenta com casos de borda, testes de processamento e fluxo de funcionalidades.
Enxame de formigas funcional em simulação gráfica.	Testes e análises de movimentação da formiga em interação com o ambiente.
Interface para gerenciamento do enxame das formigas em integração com o ambiente.	Testes com experimentos e casos de borda na inicialização dos parâmetros.
Upload dos dados a respeito da simulação em armazenamento local.	Testes com reexecução de experimentos com importação dos dados armazenamentos.
Integração da simulação do enxame com a aplicação de um algoritmo evolutivo, junto com sua interface de gerenciamento.	Testes de avaliação e evolução da população de enxames em simulação.
Modularização e generalização dos códigos envolvendo enxame a fim de generalizar os agentes de formigas, para qualquer tipo.	Testes com diferentes tipos de enxame.

Tabela 2: Resultados previstos e seus indicadores de avaliação

Referências

- 1 Ko, A. W. Y., & Lau, H. Y. K. (n.d.). *Intelligent Robot-assisted Humanitarian Search and Rescue System*.
- 2 Reynolds, C. W. (1987). *Flocks, herds and schools: A distributed behavioral model*. *Proceedings of the 14th Annual Conference on Computer Graphics and Interactive Techniques*, 25–34.
- 3 Dorigo, M., Birattari, M., & Stutzle, T. (2006). *Ant colony optimization*. *IEEE Computational Intelligence Magazine*, 1(4), 28–39. <https://doi.org/10.1109/MCI.2006.329691>
- 4 Yang, C., Chen, J., & Tu, X. (2007). *Algorithm of Fast Marriage in Honey Bees Optimization and Convergence Analysis*. *2007 IEEE International Conference on Automation and Logistics*, 1794–1799. <https://doi.org/10.1109/ICAL.2007.4338865>
- 5 Kennedy, J., & Eberhart, R. (1995). *Particle swarm optimization*. *Proceedings of ICNN'95 - International Conference on Neural Networks*, 4, 1942–1948 vol.4. <https://doi.org/10.1109/ICNN.1995.488968>
- 6 J. Tang, G. Liu and Q. Pan, "A Review on Representative Swarm Intelligence Algorithms for Solving Optimization Problems: Applications and Trends," in *IEEE/CAA Journal of Automatica Sinica*, vol. 8, no. 10, pp. 1627-1643, October 2021, doi: 10.1109.
- 7 M. Janga Reddy, D. Nagesh Kumar; *Evolutionary algorithms, swarm intelligence methods, and their applications in water resources engineering: a state-of-the-art review*. *H2Open Journal* 1 January 2020; 3 (1): 135–188. doi: <https://doi.org/10.2166/h2oj.2020.128>
- 8 Diep, Q.B., Truong, T.C., Zelinka, I. (2021). *Swarm Intelligence and Swarm Robotics in the Path Planning Problem*. In: Piunovskiy, A., Zhang, Y. (eds) *Modern Trends in Controlled Stochastic Processes: Emergence, Complexity and Computation*, vol 41. Springer, Cham. https://doi.org/10.1007/978-3-030-76928-4_16
- 9 Ankit Thakkar, Ritika Lohiya, *Role of swarm and evolutionary algorithms for intrusion detection system: A survey*, *Swarm and Evolutionary Computation*, Volume 53, 2020, 100631, ISSN 2210-6502, <https://doi.org/10.1016/j.swevo.2019.100631>.
- 10 S. Lim, S. Wang, B. Lennox and F. Arvin, "BeeGround - An Open-Source Simulation Platform for Large-Scale Swarm Robotics Applications," *2021 7th International Conference on Automation, Robotics and Applications (ICARA)*, 2021, pp. 75-79, doi: 10.1109/ICARA51699.2021.9376494.
- 11 E. Soria, F. Schiano and D. Floreano, "SwarmLab: a Matlab Drone Swarm Simulator," *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020, pp. 8005-8011, doi: 10.1109/IROS45743.2020.9340854
- 12 CUDA®. (n.d.). <https://developer.nvidia.com/cuda-zone>
- 13 *The Industry's Foundation for High Performance Graphics*. (n.d.). Retrieved June 3, 2021, from <https://www.opengl.org/>
- 14 M. I. S. B. Khairat, Y. Priyadi and M. Adrian, "Usability Measurement in User Interface Design Using Heuristic Evaluation & Severity Rating (Case Study: Mobile TA Application based on MVVM)," *2022 IEEE 12th Annual Computing and Communication Workshop and Conference (CCWC)*, 2022, pp. 0974-0979, doi: 10.1109/CCWC54503.2022.9720876.