

## O ALGORITMO IGPU-COLORANT3-RT

Bruno Cesar Puli Dala Rosa (PIBIC/CNPq/FA/Uem), Anderson Faustino Da Silva (Orientador), e-mail: [bcesar.g6@gmail.com](mailto:bcesar.g6@gmail.com)

Universidade Estadual de Maringá / Departamento de informática /Maringá, PR.

**Área e subárea: Ciências Exatas e da Terra / Ciência da computação**

**Palavras-chave:** Problema de Coloração de Grafos, Otimização por Colônia de Formigas Artificiais, Algoritmos paralelos.

### Resumo:

Não existem algoritmos polinomiais que solucionam o Problema de Coloração de Grafos, em tempo satisfatório, assim o uso de abordagens heurísticas vem sendo explorado. O algoritmo *ColorAnt* emprega as heurísticas de Otimização por Colônia de Formigas e Busca Local na resolução do problema e apresenta bons resultados, porém ao custo de um tempo de execução longo. Neste trabalho foram desenvolvidos e implementados algoritmos que aplicam técnicas de paralelização, com o objetivo de minimizar o tempo de execução do algoritmo *ColorAnt*. Os resultados obtidos comprovam a eficácia desta técnica ao exibirem ganhos de velocidade consistentes entre os experimentos realizados.

### Introdução

No Problema de Coloração de Grafos (PCG) busca-se atribuir valores, representados como cores, aos vértices de um grafo  $G$  respeitando a restrição de que vértices adjacentes não podem partilhar da mesma cor (BONDY e MURTY, 2008). Por se tratar de um problema NP-Difícil o desenvolvimento de algoritmos heurístico é considerado uma abordagem viável a sua resolução. A Otimização por Colônia de Formigas, do inglês *Ant Colony Optimization* (ACO), é uma meta-heurística que se baseia no comportamento apresentado por formigas durante a busca por alimento em um ambiente (DORIGO e KRZYSZTOF, 2006) e vem mostrando resultados promissores para o PCG como visto no algoritmo *ColorAnt* proposto por Lintzmayer e outros (2015). *ColorAnt* combina a metaheurística ACO com o método construtivo *Ant\_fixed\_k* (COSTA e HERTZ, 1997), além disto este utiliza uma meta-heurística de Busca Local para refinamento dos resultados.

Em um primeiro trabalho foi explorada a paralelização para arquiteturas GPU. Apesar do sucesso no desenvolvimento do algoritmo seu desempenho foi abaixo do esperado. Este novo trabalho tem como foco a

otimização do algoritmo desenvolvido anteriormente e também a aplicação da técnica de paralelização com foco em processadores tradicionais.

## Materiais e métodos

A versão mais recente do algoritmo *ColorAnt* remove as restrições de execução do algoritmo original e implementa a mudança dinâmica do esquema de atualização do feromônio durante a execução, estas alterações trouxeram maior flexibilidade ao algoritmo. Utilizando esta nova versão como base foram desenvolvidas e implementadas duas novas versões do algoritmo: *Pthreads-ColorAnt* e *iGPU-ColorAnt*, ambas versões modificam o método construtivo do algoritmo *GPU-ColorAnt* possibilitando que ele seja executado em mais de um fluxo de execução simultaneamente.

No algoritmo *Pthreads-ColorAnt* o laço de repetição principal do método construtivo é dividido em  $N$  fluxos de execução, onde  $N$  é um parâmetro do algoritmo, de forma a distribuir o processamento pelos núcleos do computador. Já o algoritmo *iGPU-ColorAnt* é uma evolução direta do algoritmo *GPU-ColorAnt* (ROSA, 2016), desenvolvido para a execução em unidades de processamento gráfico. Em *iGPU-ColorAnt* todas as funções do método construtivo foram revisadas, consequentemente otimizando as estratégias de paralelização e uso da memória. Além disso, a etapa de atualização do feromônio passou a ser executada paralelamente na GPU, de forma a reduzir a transferência de dados entre a memória principal e memória da GPU. As demais etapas do algoritmo *ColorAnt*, como a inicialização do algoritmo, a Busca Local e os esquemas de atualização da trilha de feromônio permaneceram iguais nas novas versões.

## Resultados e Discussão

Os algoritmos implementados foram executados em um computador Intel Xeon E5, 32GB RAM, GPU GeForce GTX 680 e sistema operacional CentOS 6.5.

Para a realização dos experimentos foram selecionados 5 grafos, já conhecidos na literatura, como instâncias de entrada do algoritmo. As execuções foram configuradas para realizar a coloração com o melhor número cromático conhecido para a instância. Os parâmetros de execução foram os seguintes: 1024 formigas no sistema ACO, X ciclos de busca local e tempo de execução de uma hora.

Para avaliar o desempenho dos algoritmos e a qualidade das soluções geradas os experimentos foram executados cinco vezes para cada instancia de entrada. No caso do algoritmo *Pthreads-ColorAnt* o número de formigas é dividido pelo número de fluxos de execução simultaneos, um parâmetro do algoritmo que neste experimento variou entre 6, 12 e 24. Já no algoritmo *iGPU-ColorAnt* cada formiga representa um fluxo de execução na

GPU, desta forma, a cada ciclo do algoritmo todas as instâncias do método construtivo são executadas paralelamente.

A Figura 1 apresenta os resultados obtidos. Nesta figura, cada barra representa a média do tempo de execução do método construtivo em um ciclo do algoritmo e acima de cada barra é exibido o valor médio de conflitos (quantidade de vértices adjacentes com a mesma coloração) ao fim das execuções.

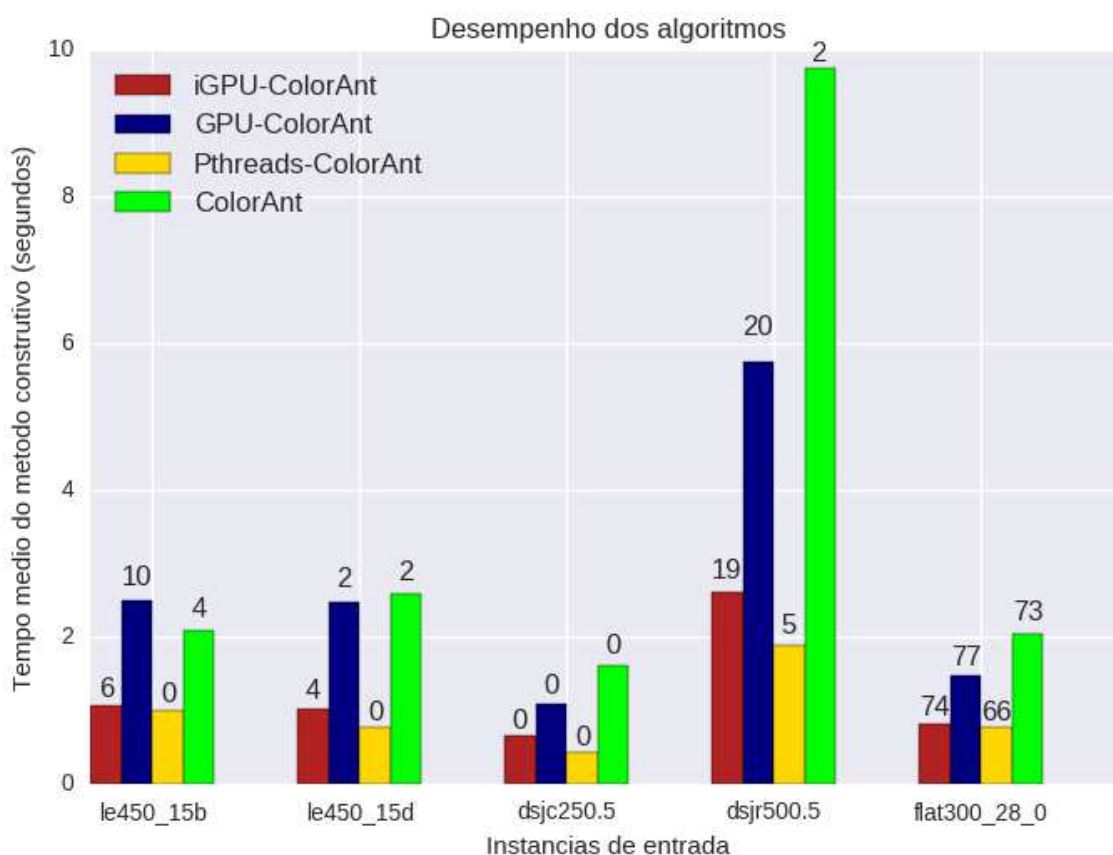


Figura 1 – Resultados dos experimentos

Os resultados demonstram a eficiência das estratégias de minimização do tempo de execução aplicadas aos algoritmos desenvolvidos e também evidenciam as melhorias na nova versão do algoritmo GPU, especialmente quando comparado a sua versão anterior. De maneira geral os resultados foram consistentes independentemente da instância de entrada, salva exceção em *le450\_15b*, de forma que os algoritmos podem ser ordenados quanto a seu desempenho na seguinte ordem crescente: *Pthreads-ColorAnt*, *iGPU-ColorAnt*, *GPU-ColorAnt* e *ColorAnt*.

Por meio destes experimentos pode-se identificar como melhores algoritmos os algoritmos *Pthreads-ColorAnt* e *iGPU-ColorAnt*, com destaque

especial a *Pthreads-ColorAnt* que apresentou os melhores resultados tanto para velocidade de execução quanto na qualidade das soluções encontradas. Estas duas características estão relacionadas, pois um método construtivo mais rápido garante que mais ciclos do algoritmo serão executados dentro do tempo limite de execução, refinando assim a solução final.

## Conclusões

O estudo do desempenho dos algoritmos revela diferenças expressivas entre eles, especialmente entre os algoritmos *Pthreads-ColorAnt* e *iGPU-ColorAnt*, desenvolvidos neste trabalho. Tal comportamento era esperado pois os novos algoritmos implementam estratégias mais eficientes de processamento, as quais são capazes de distribuir a computação entre diversos fluxos de execução. De fato, tal característica é especialmente proveitosa nas arquiteturas modernas, pois estas disponibilizam diversas unidades de processamento.

Com o intuito de otimizar os algoritmos desenvolvidos trabalhos futuros compreendem: o desenvolvimento de novas funções de avaliação das soluções na atualização do feromônio e a paralelização do algoritmo de busca local.

## Agradecimentos

Agradeço ao CNPq e a UEM por possibilitarem este trabalho.

## Referências

BONDY, J. A.; MURTY, U. S. R. **Graph Theory**. New York, NY, USA: Springer, 2008. 654 p. (Graduate Texts in Mathematics, v. 244).

COSTA, D.; HERTZ, A. **Ants Can Colour Graphs**. The Journal of the Operational Research Society, Palgrave Macmillan, Houndmills, UK, v. 48, n. 3, p. 295–305, 1997.

DORIGO, M.; KRZYSZTOF, S. **An Introduction to Ant Colony Optimization**. IRIDIA Technical Report Series, 2006.

LINTZMAYER, C. N.; MULATI, M.; SILVA, A. F. Da; **The Hybrid ColorAnt-RT Algorithms and an Application to Register Allocation**. Inteligencia Artificial, volume 18, número 55, p. 81-111, 2015.

ROSA, B. C. P. D. **O algoritmo GPU-ColorAnt3-RT**. In: ENCONTRO ANUAL DE INICIAÇÃO CIENTÍFICA, 25., 2016. Maringá. **Anais...** Maringá: UEM, 2016. p. 1-4.