



O Algoritmo GPU-ColorAnt3-RT

Bruno Cesar Puli Dala Rosa (PIBIC/CNPq/FA/Uem), Anderson Faustino Da Silva (Orientador), e-mail: 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, GPU.

Resumo:

O Problema de Coloração de Grafos não possui algoritmos exatos em tempo polinomial, desta forma meta-heurísticas são empregadas para resolver este problema. O algoritmo ColorAnt3-RT é um promissor algoritmo heurístico por apresentar bons resultados, porém, ao custo de um tempo de execução longo. Este trabalho apresenta e analisa uma estratégia de minimização do tempo de execução de tal algoritmo, uma versão paralela para arquiteturas GPU.

Introdução

O Problema de Coloração de Grafos (PCG) consiste, basicamente, em colorir os vértices de um grafo G com uma quantidade k de cores, de modo que vértices adjacentes não tenham a mesma cor. O objetivo é encontrar um número k mínimo que respeite as restrições do problema, denominado número cromático de G (Bondy e Murty, 2008). Devido ao caráter NP-Difícil do problema, meta-heurísticas e algoritmos heurísticos vem sendo explorados e implementados. A Otimização por Colônia de Formigas, ou 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 et al, 2006). O ColorAnt3-RT é um promissor algoritmo ACO (Lintzmayer, 2011) para coloração de grafos, capaz de obter boas soluções para os melhores resultados conhecidos, entretanto, devido ao seu longo tempo de execução ele é inviável em





sistemas cujo tempo de execução é um fator crítico. Desta forma, o presente trabalho apresenta uma estratégia de minimização de seu tempo de execução: a paralelização de seu método construtivo ACO para arquiteturas GPU.

Materiais e métodos

ColorAnt3-RT utiliza o algoritmo Ant_Fixed_k, proposto por Costa e Hertz (1997), como método construtivo para cada formiga da meta-heurística ACO e, opcionalmente, também utiliza um algoritmo de busca local para melhorar as soluções. O algoritmo GPU-ColorAnt3-RT diferencia-se do algoritmo ColorAnt3-RT pelo modo como encontra uma solução. GPU-ColorAntRT-3 utiliza uma versão paralela do algoritmo Ant_fixed_k, denominado GPU_Ant_fixed_k, desenvolvido e implementado para ser executado na GPU. Desta forma as soluções são geradas paralelamente pela GPU, um processo diferente ocorre no algoritmo original, onde as soluções são geradas sequencialmente na CPU. Os demais procedimentos como a busca tabu e a manipulação da trilha do feromônio são similares aos do algoritmo ColorAnt3-RT. A Figura 1 apresenta a arquitetura do algoritmo GPU-ColorAnt3-RT.

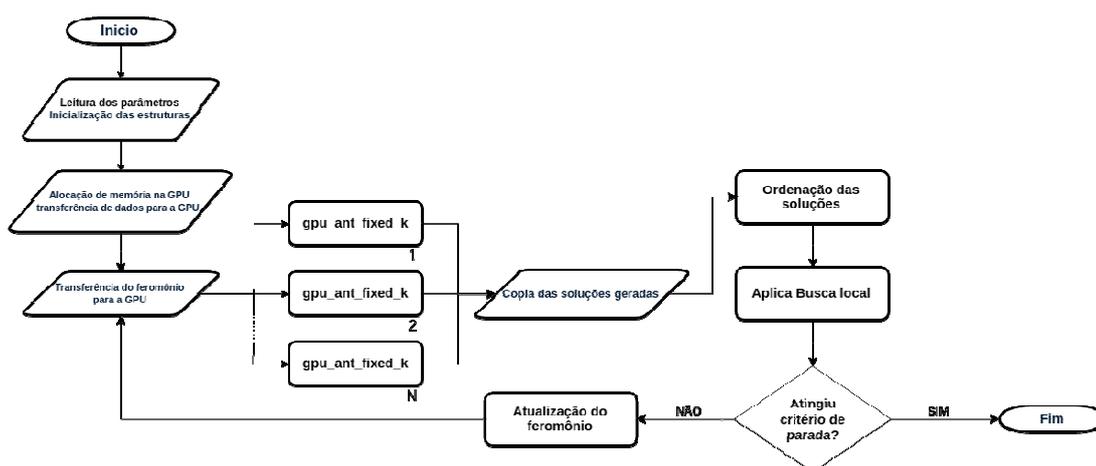


Figura 1 – Algoritmo Gpu-ColorAnt3-RT

Ambos algoritmos são executados continuamente até que algum critério de parada seja atingido, podendo ser: solução ótima encontrada; o





número máximo de ciclos atingido; número máximo de ciclos sem melhora atingido (convergência); ou tempo máximo alcançado. Estes critérios são parâmetros do algoritmo e podem ser combinados nas execuções.

Resultados e Discussão

Os algoritmos implementados foram executados em um computador Intel Core i7-3820 com a seguinte configuração: 3.60GHz, 32GB RAM, GPU GeForce GTX 680 e sistema operacional Ubuntu 15.10.

Os experimentos foram realizados utilizando 6 grafos do Desafio DIMACS. Os critérios de paradas são: número máximo de 2500 ciclos, número máximo de 500 ciclos sem melhora e número de cores k igual ao melhor número cromático encontrado para a instância. Além disto, a quantidade de formigas variou entre: 64, 128, 256 e 512.

A Tabela 1 apresenta os resultados alcançados por ColorAnt3-RT (CA3-RT) e GPU-ColorAnt3-RT (GPU-CA3-RT).

Tabela 1 – Resultados obtidos

Grafo	Quantidade de Formigas				
	Algoritmo	64	128	256	512
DSJC500.5	CA3-RT	0.47s / 30	0.93s / 15	1.80s / 19	3.60s / 15
	GPU-CA3-RT	0.47s / 34	0.70s / 35	1.20s / 37	2.13s / 28
DSJR500.5	CA3-RT	0.74s / 3	1.49s / 5	3.0s / 6	6.45s / 2
	GPU-CA3-RT	0.76s / 20	1.06s / 20	1.73s / 20	3.06s / 18
flat300_28_0	CA3-RT	0.16s / 55	0.32s / 54	0.67s / 43	1.26s / 56
	GPU-CA3-RT	0.19s / 63	0.27s / 58	0.48s / 47	1.04s / 59
latin_square	CA3-RT	1.42s / 18	2.95s / 23	7.30s / 3	11.25s / 13
	GPU-CA3-RT	1.38s / 18	2.06s / 20	4.07s / 20	6.31s / 12
le450_15d	CA3-RT	0.21s / 0	0.42s / 0	0.84s / 0	1.66s / 0
	GPU-CA3-RT	0.29s / 0	0.40s / 0	0.75s / 0	1.41s / 0
r1000.1.c	CA3-RT	1.79s / 3	3.76s / 3	7.19s / 2	14.37 / 2
	GPU-CA3-RT	1.84s / 0	2.84s / 0	5.23s / 3	9.46s / 0

Os resultados apresentam o tempo médio de execução do método construtivo em cada ciclo do algoritmo e o número de vértices conflitantes no fim da execução. Pode-se observar que, em ambos os algoritmos, o tempo de execução cresce conforme a quantidade de formigas aumenta, porém como era esperado, este crescimento é menor no algoritmo paralelo e desta forma a diferença entre os resultados também cresce com o aumento das formigas. A diferença varia de intensidade e chegou a um máximo de 4.94





segundos, que representa 44% de melhora, este comportamento foi observado para todos os grafos.

Conclusões

A análise dos resultados obtidos revela uma melhora no tempo de processamento do método construtivo ACO, entretanto, esta melhora não é muito expressiva quando comparada ao ganho de outras paralelizações para GPU conhecidas. A causa para tal comportamento é a estrutura do algoritmo ColorAnt3-RT e da natureza de suas operações, este apresenta diversos trechos de instruções que são necessariamente sequenciais em sua lógica.

Os trabalhos futuros compreendem (1) A paralelização do algoritmo de busca local, (2) O estudo de novas estratégias para o feromônio e (3) Melhores adequações do algoritmo a arquitetura GPU. Desta forma é esperado que a próxima versão do algoritmo GPU-ColorAnt3-RT tente ganhos compatíveis com aqueles apresentados na literatura, por paralelizações que utilizam GPU.

Agradecimentos

Agradeço ao CNPq, a Fundação Araucária 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. **Algoritmos Heurísticos para Coloração de Grafos**. 2011. Monografia, Graduação em Ciência da Computação, do Centro de tecnologia da Universidade Estadual de Maringá, Maringá, 2011.

