

# Algoritmo Paralelo de Dinâmica de Partículas Aplicado ao Problema de Agrupamento em Grafos

Hugo Resende<sup>1</sup>, Álvaro Luiz Fazenda<sup>2</sup>, Marcos Gonçalves Quiles<sup>2</sup>

<sup>1</sup>Instituto de Educação, Ciência e Tecnologia do Sul de Minas Gerais (IFSULDEMINAS)  
Campus Passos – 37903-358 – Passos–MG

<sup>2</sup>Universidade Federal de São Paulo (UNIFESP)  
Instituto de Ciência e Tecnologia – 12247-014 – São José dos Campos–SP

**Resumo.** *Diversas aplicações podem ser naturalmente modeladas por meio de estruturas matemáticas combinatoriais como as redes complexas. Nessas estruturas, vértices bem relacionados podem indicar alguma informação potencialmente útil e, além disso, proporcionar uma melhor avaliação dos dados. Por essa razão, o problema de agrupamento em redes complexas, ou agrupamento em grafos, tem sido bastante estudado por pesquisadores de várias áreas. No entanto, a maioria dos algoritmos propostos têm encontrado dificuldades em agrupar vértices em grafos com milhares de vértices e arestas, sendo um dos grandes desafios, projetar algoritmos eficientes para tal fim. Um dos algoritmos propostos recentemente para o problema de agrupamento em grafos, que se utiliza da técnica de dinâmica de partículas como mecanismo de construção de soluções (agrupamentos) apresentou uma eficiência considerável em relação a demais algoritmos conhecidos, entretanto, possui complexidade quadrática, limitando a sua aplicação em grafos com muitas arestas e vértices. Objetivando superar limitações no desempenho computacional, bem como citado, pesquisadores recorrem, normalmente, a estratégias relacionadas ao processamento de alto desempenho, tal como a paralelização dos algoritmos em CPUs e em GPUs. Tendo isso em mente, o presente trabalho objetiva demonstrar que a paralelização do algoritmo de agrupamento em grafos baseado na técnica de dinâmica de partículas, implementada para CPUs, por meio do uso de diretivas OpenMP, pode ser uma alternativa válida na aplicação desse algoritmo em instâncias de grafos maiores.*

**Palavras-chave:** *Agrupamento em grafos, dinâmica de partículas, Programação Paralela, OpenMP.*

## 1. Introdução

Um tópico de pesquisa investigado nas últimas décadas é a detecção de vértices (nós) bem relacionados na estrutura matemática combinatoria grafos. Tais estruturas permitem uma modelagem eficiente de aplicações como a Internet, as redes sociais, as redes biológicas, entre outras. Por meio do agrupamento de vértices nos grafos que modelam essas aplicações, é possível de se obter informações relevantes, como a identificação de padrões entre os objetos da aplicação modelada [Newman 2006, Porter et al. 2009].

A tarefa de detectar grupos de vértices em grafos é conhecida, segundo a literatura, como problema de agrupamento em grafos. Para lidar com esse problema, vários

algoritmos, baseados em estratégias distintas, como a programação linear, a programação inteira, a hibridização entre métodos inteiros e heurísticos, entre outras, foram propostos nas últimas décadas [Clauset et al. 2004]. Um algoritmo proposto recentemente, baseado em uma técnica da física conhecida como *dinâmica de partículas* tem apresentado bons resultados ao agrupar vértices [Quiles et al. 2016]. Entretanto, similarmente à maioria dos algoritmos encontrados na literatura para essa finalidade, este possui complexidade computacional elevada ( $O(n^2)$ , onde  $n$  é o número de vértices do grafo), tornando inviável a sua aplicação em grafos com milhares de vértices e arestas.

Com o advento das arquiteturas computacionais *multicore*, a computação de alto desempenho foi massificada tornando-a ainda mais atrativa na superação das limitações de algoritmos que possuem altos custos computacionais. Em diferentes cenários de aplicação, a paralelização de tais algoritmos por meio de APIs para programação paralela em CPUs como a *Open Multi-Processing* (OpenMP) e com o auxílio da plataforma CUDA para programação paralela em GPUs, segundo a literatura, podem levar à uma melhoria de desempenho na busca por soluções [Resende 2014, Pacheco 2011, Farber 2011].

Ao considerar a complexidade do problema de agrupamento em grafos e a possibilidade da otimização de desempenho de algoritmos que o resolvem por meio de técnicas de processamento paralelo em arquiteturas *multicore*, este trabalho investiga o problema de agrupamento em grafos e propõe uma abordagem paralela para o algoritmo de detecção de comunidades em grafos, baseado na técnica de dinâmica de partículas. A técnica para melhoria de desempenho consiste na utilização de diretivas da API OpenMP como mecanismos de paralelização do algoritmo em CPUs. O algoritmo paralelo desenvolvido poderá ainda futuramente ser estendido para utilizar recursos de programação da plataforma CUDA para GPU. Resultados preliminares do algoritmo paralelo implementado para CPUs indicam que essa abordagem é válida.

## 2. Materiais e Métodos

O algoritmo de detecção de comunidades, proposto por [Quiles et al. 2016], pode ser descrito, de forma resumida, da seguinte maneira. Cada vértice de um grafo é associado, inicialmente, à uma partícula de sistema de interação, o qual é distribuído de forma aleatória em um espaço tridimensional. A cada iteração do modelo, são calculados os valores das interações para cada par de partículas. Essas interações, que podem ser atração (caso exista uma aresta conectando o par de vértices) ou repulsão (caso não haja aresta entre o par de vértices), fornecem os parâmetros para a atualização das posições das partículas no espaço. Após as iterações, o sistema alcança um estado conhecido por *estado de agrupamento* ou *estado de equilíbrio* e as partículas mais próximas no espaço, naturalmente, tendem a formar pequenos grupos (que posteriormente são identificados por um algoritmo particionador de grafos), os quais caracterizam as diferentes comunidades no grafo.

Diante do exposto, identificou-se que a tarefa mais complexa do algoritmo objeto de estudo é a realização do cálculo da interação (atração ou repulsão) para cada par de partículas, que gera um custo algorítmico de complexidade  $O(n^2)$ . Por meio de diretivas OpenMP, dividiu-se a quantidade de cálculos das interações entre os pares, pela quantidade de *threads*, calculando assim, essa parte do algoritmo de forma paralela. Para assegurar a corretude dessa primeira abordagem paralela, optou-se por fixar as posições iniciais das partículas no espaço tridimensional para os grafos experimentados. Os resul-

tados dessa primeira abordagem, apresentados na seção seguinte, comprovam que houve um aumento no desempenho do algoritmo sem perda significativa de escalabilidade.

Sobre a abordagem paralela para GPU, a qual está em fase de desenvolvimento, identificou-se, como característica substancial, que a parte mais complexa do algoritmo original não possui operações de entrada/saída de dados, restringindo-se, especificamente, ao cálculo das interações. Por conseguinte, o desenvolvimento está sendo realizado com base na mesma ideia do algoritmo paralelizado para CPUs.

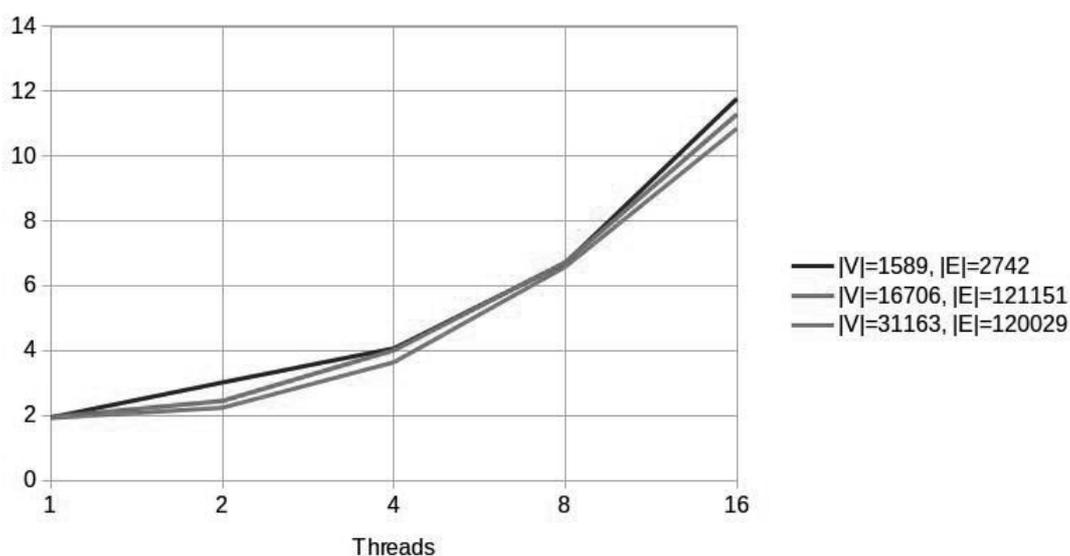
### 3. Resultados e Discussão

Os experimentos com os algoritmos original e paralelo para CPUs foram executados em uma máquina com processador AMD Opteron 6272, totalizando 16 núcleos, com velocidade de *clock* igual a 2.1GHz, e memória principal de 64Gb. A Tabela 1 apresenta os tempos de processamento demandados pelo algoritmo serial e pelo algoritmo paralelo para agrupar vértices em três instâncias de grafos. Para experimentar o algoritmo paralelo foram utilizadas diferentes quantidades de *threads*, conforme apresentado a seguir.

**Tabela 1. Tempos de execução (em segundos) do algoritmo paralelo para CPU**

V ,  E	Serial	1 thread	2 threads	4 threads	8 threads	16 threads
1589, 2742	16,962	19,977	10,330	4,914	2,981	1,697
16706, 12151	2265,673	2496,22	1279,8	622,80	371,314	221,165
31163, 120029	9971,209	11382,9	5938,4	3127,2	1729,13	1049,08

Para melhor demonstrar o desempenho da abordagem paralela para CPUs, a Figura 1 ilustra os valores de *speedup* [Pacheco 2011] obtidos por meio dos tempos de execução, demandados para agrupar o grafos reportados na Tabela 1.



**Figura 1. Speedup obtido para as instâncias de grafos da Tabela 1**

Por meio de uma análise dos resultados apresentados, percebe-se que há um ganho de desempenho do algoritmo paralelo para CPUs com relação ao algoritmo serial. Diante

desse cenário, a paralelização do algoritmo de detecção de comunidades se apresentou como uma boa alternativa no que diz respeito ao agrupamento de grafos maiores, entretanto, há ainda espaço para melhoria da eficiência paralela do algoritmo, o que está sendo desenvolvido atualmente.

#### **4. Considerações Finais**

Neste documento foi apresentada uma breve motivação sobre o estudo do problema de agrupamento em grafos, principalmente no tocante às altas complexidades de algoritmos que são propostos para tratá-lo de forma eficiente. Dentre esses algoritmos, foi apresentada uma abordagem baseada na técnica de dinâmica de partículas que possui uma boa eficiência para lidar com o problema mencionado. De modo a aumentar o seu poder na busca por soluções, foi apresentada uma versão paralela desse modelo. Por meio dos resultados preliminares reportados, percebe-se que o algoritmo paralelizado com o auxílio de diretivas OpenMP é uma alternativa válida para tratar grafos maiores.

A futura versão a ser implementada em CUDA para GPU pode surgir como uma alternativa ainda melhor, uma vez que o ponto crítico do algoritmo original é o cálculo das distâncias entre todos os pares de vértices (partículas), a cada iteração. Tendo em vista que as GPUs possuem como característica intrínseca o alto poder de realização de operações de ponto flutuante e que o algoritmo, em sua essência, implementa muitas operações do gênero concorrentemente, espera-se que essa segunda versão seja bastante promissora.

#### **Agradecimentos**

Agradecemos à Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP) e à Fundação de Amparo à Pesquisa do Estado de Minas Gerais (FAPEMIG) pelo apoio financeiro para o desenvolvimento dessa pesquisa.

#### **Referências**

- Clauset, A., Newman, M. E., and Moore, C. (2004). Finding community structure in very large networks. *Physical review E*, 70(6):066111.
- Farber, R. (2011). *CUDA application design and development*. Elsevier.
- Newman, M. E. (2006). Modularity and community structure in networks. *Proceedings of the national academy of sciences*, 103(23):8577–8582.
- Pacheco, P. (2011). *An introduction to parallel programming*. Elsevier.
- Porter, M. A., Onnela, J.-P., and Mucha, P. J. (2009). Communities in networks. *Notices of the AMS*, 56(9):1082–1097.
- Quiles, M. G., Macau, E. E., and Rubido, N. (2016). Dynamical detection of network communities. *Scientific reports*, 6.
- Resende, H. (2014). Algoritmos paralelos para determinar agrupamento em grafos com única ou múltiplas camadas. Master's thesis, Universidade Federal de São Paulo.