Um Algoritmo Exato em *clusters* de GPUs para o *Hitting Set* Aplicado à Inferência de Redes de Regulação Gênica

Danilo Carastan dos Santos - Autor[1*], Luiz Carlos da Silva Rozante - Orientador[1]

[1]Centro de Matemática, Computação e Cognição
Universidade Federal do ABC (UFABC)
Santo André – SP – Brasil

{danilo.santos,luiz.rozante}@ufabc.edu.br

Abstract. *We propose a Hitting Set problem (HSP) algorithm suited for GRN inference applications by introducing innovations in the data structures and a sorting scheme to allow efficient discarding of Hitting Set non-solution candidates. We provided an implementation for multi-core CPUs and for (homogeneous and heterogeneous) GPU clusters. Experimental results show that the optimizations offered significant individual performance gains and a high scalability with increasing number of GPUs. The conjunction of these gains resulted in speedups above 60 for the parallel part of the algorithm. We publish two papers: one in conference (Qualis CC A1) and the other in journal (Qualis CC A2).*

Resumo. *Propomos um algoritmo para resolver o problema do Hitting Set (HSP), adequado para aplicações de inferência de GRNs, introduzindo inovações nas estruturas de dados e um mecanismo de ordenação que permite um descarte eficiente de candidatos que não são solução. Foram providas implementações em CPU multi-core e em clusters (homogêneos e heterogêneos) de GPU. Resultados experimentais mostraram que as otimizações ofereceram ganhos de desempenho individuais significativos e alta escalabilidade com o aumento do número de GPUs. A conjunção desses ganhos resultou em speedups acima de 60 para a parte paralela do algoritmo. Publicamos dois artigos: um em congresso (Qualis CC A1) e outro em periódico (Qualis CC A2).*

1. Introdução e Motivação

Diversas áreas de estudo possuem problemas teóricos ou práticos os quais podem ser modelados, em parte ou como um todo, como uma instância do problema da Transversal Mínima (do Inglês, *Hitting Set Problem* ou HSP). A área de Biologia de Sistemas não é uma exceção, apresentando aplicações tais como distância reversa genômica [Kolman and Walen 2007], experimentos com reação em cadeia de polimerase [Pearson et al. 1996] e inferência de redes de regulação gênica (do Inglês *Gene Regulatory Networks* ou GRNs) [Ideker et al. 2000, Ruchkys and Song 2003]. Ideker *et al.* [Ideker et al. 2000], por exemplo, utiliza uma abordagem iterativa de perturbação de sinais, na qual se aplicam perturbações sucessivas nos níveis de expressão dos genes e as

*Financiado pelo CNPq

alterações consequentes são analisadas com o objetivo de inferir interações regulatórias entre os genes. Uma característica dessa abordagem é que um de seus passos pode ser modelado via HSP. Essas interações entre os genes são usualmente essenciais para diversos tratamentos de doenças [Madhamshettiwar et al. 2012] e estudos de um sistema biológico [Barrera et al. 2007]. Em geral, é bastante habitual que os subsequentes tamanhos de entrada para o HSP dessas aplicações sejam consideravelmente grandes, fazendo com que a obtenção das soluções exatas seja inviável ou até mesmo impossível para os algoritmos atuais.

Uma vez que o HSP é um problema NP-Difícil [Garey and Johnson 1999], existem alguns algoritmos que tentam contornar esse problema, tais como algoritmos exatos não polinomiais [Shi and Cai 2010], algoritmos de aproximação e heurísticas [Ruchkys and Song 2003, Cendic 2014]. Ruchkys e Song [Ruchkys and Song 2003] propuseram algoritmos de aproximação sequencial e paralelo para resolver o *Hitting Set* de modo a inferir quais genes afetam o nível expressão de um determinado gene-alvo. Mais recentemente, Steinbach e Postohoff [Steinbach and Posthoff 2012] propuseram um algoritmo exato que utiliza GPUs (do Inglês, *Graphics Processing Units*) para determinar a cobertura mínima de expressões Booleanas, o qual é um problema dual do *Hitting Set*. A exceção do algoritmo de aproximação de Ruchkys e Song [Ruchkys and Song 2003], nenhum dos algoritmos supracitados são capazes de lidar com grandes conjuntos de entrada (ordem de milhares de variáveis), o que é precisamente o caso no contexto de inferência de GRNs. Além disso, nenhum destes algoritmos permitem utilizar múltiplas GPUs.

Nesse trabalho foi proposto um algoritmo que faz uso de múltiplas GPUs para se obter as soluções exatas do HSP, incluindo um mecanismo de ordenação capaz de desclassificar eficientemente as não soluções do problema. O algoritmo foi utilizado para resolver o problema de inferência de GRNs e os resultados experimentais mostraram que o mecanismo de ordenação resultou em *speedups* de até 3,5, em comparação com o algoritmo sem esse mecanismo. Adicionalmente, a implementação em GPU foi capaz de fornecer um *speedup* adicional de 4,7, em comparação com uma implementação em CPU *multi-core*. Por fim, utilizando 8 GPUs Tesla K20c, dividida em duas *workstations*, foi obtido um *speedup* adicional de 6,6, em comparação com o algoritmo executando em apenas uma GPU. Todas essas otimizações combinadas fornecerem *speedups* acima de 60, levando em consideração as partes paralelizadas do algoritmo.

O restante desse texto está estruturado como segue. Na Seção 2 são apresentadas algumas das definições preliminares desse trabalho, incluindo a definição do HSP e o problema de inferência de GRNs. Na Seção 3 são apresentados os objetivos e as contribuições obtidas nesse trabalho. Os resultados alcançados, incluindo os artigos publicados, são apresentados na Seção 4 e as conclusões são apresentadas na Seção 5.

2. Algumas Definições

2.1. Inferência de Redes de Regulação Gênica

Dado um conjunto de dados de entrada e um determinado gene alvo, o problema de identificação dos preditores consiste em determinar os genes cujos níveis de expressão realizam uma melhor predição do nível de expressão desse gene alvo. Um gene *preditor* é um gene cujo perfil de expressão possui informação significante (sozinho ou em conjunto com outros preditores) sobre o perfil de expressão do gene alvo [Barrera et al. 2007].

Sendo assim, o problema de inferência de GRNs consiste em determinar o melhor conjunto de preditores para todos os genes alvo em estudo, levando em conta os dados de entrada. Em geral, os dados de entrada são medidas de expressão dos genes de um determinado sistema biológico. Embora estas expressões sejam de natureza contínua, adotamos uma representação Booleana, que é um bom ponto de partida para a modelagem de GRNs [Kauffman 1969]. A literatura referente aos métodos de inferência de GRNs expande rapidamente. Mais informações relativas a esses diversos métodos de inferência podem ser encontradas na seguinte revisão: [Ristevski 2013].

Muitos dos métodos existentes de inferência de GRNs não fazem utilização de técnicas de paralelismo. Entretanto, publicações recentes indicam que há um crescimento em pesquisa de métodos paralelos de inferência de GRNs, em especial, métodos que utilizam o processamento paralelo em GPUs, como por exemplo os trabalhos de Shi *et al.* [Shi et al. 2011] e Borelli *et al.* [Borelli et al. 2012]. Nesse trabalho consideramos a abordagem adotada por [Ruchkys and Song 2003], na qual é utilizada uma estratégia baseada no HSP (ver Seção a seguir) para efetuar a inferência de GRNs.

2.2. Problema do *Hitting Set*

Em linhas gerais, o HSP é um problema de otimização que é capaz de modelar diversos problemas combinatórios e pertence à classe dos problemas NP-difíceis [Garey and Johnson 1999]. Formalmente, podemos definir o HSP da maneira a seguir. Dado um conjunto finito X, uma coleção $\mathcal{S}$ de subconjuntos de X (alternativamente denominada como coleção de cláusulas) e um inteiro positivo k, o objetivo é encontrar um subconjunto $H \subseteq X$, de menor cardinalidade, tal que:

$$|H| \leq k \text{ e } H \cap S \neq \emptyset, \forall S \in \mathcal{S}. \quad (1)$$

Mais de um subconjunto de X pode satisfazer as condições acima. É importante notar que o inteiro positivo k, no contexto de inferência de GRNs, denota a quantidade de preditores por gene alvo e, portanto, o grau máximo dos vértices da GRN inferida. Neste trabalho foi definido o grau máximo $k \leq 5$, uma vez que existem várias evidências que sugerem que o grau médio dos genes de uma GRN está entre 2 e 3 [Kauffman 1993].

3. Objetivos e Contribuições

O objetivo desse trabalho consistiu no desenvolvimento de um algoritmo capaz de obter soluções exatas do HSP aplicado ao problema de inferência de GRNs. Essa proposta conseguiu lidar, de maneira eficiente, com conjuntos de entrada contendo milhares de variáveis, pela introdução de diversas inovações nas estruturas de dados utilizadas no algoritmo. Além disso, propusemos um mecanismo de ordenação o qual permite um descarte eficiente de candidatos que não são soluções do *Hitting Set*. Tais aprimoramentos permitiram que o algoritmo proposto seja utilizado no método de inferência de GRNs de Ruchkys e Song [Ruchkys and Song 2003]. Em seguida, desenvolvemos uma implementação do algoritmo proposto capaz de executar em aglomerados (*clusters*) de GPU de maneira eficiente e escalável. O uso de *clusters* de GPUs permitiu a execução de conjuntos de entrada maiores (ver mais detalhes na Seção 4) .

As principais contribuições deste trabalho são:

1. Um mecanismo de ordenação que possibilita um descarte no conjunto de soluções candidatas do problema *Hitting Set* que proporciona, dependendo da configuração da entrada, um considerável ganho de desempenho (*speedups* entre 2,2 e 3,5).
2. Um algoritmo paralelo e eficiente baseado na arquitetura GPU/CUDA (*speedup* entre 1,6 e 4,7), para resolver o problema *Hitting Set* – aplicado à inferência de GRNs – com quantidade arbitrária de elementos (variáveis).
3. Uma representação computacional dos elementos do problema *Hitting Set* que permite o uso relativamente eficiente dos recursos de memória das GPUs.
4. Extensão do algoritmo mencionado no item 2, o qual é capaz de fazer uso de aglomerados (*clusters*) de GPUs homogêneas, com curva de *speedup* próxima do linear em função da quantidade de GPUs (*speedup* de 6,6 para até 8 GPUs).
5. Aprimoramento do algoritmo proposto no item 2, incluindo um novo mecanismo de geração e atribuição de candidatos a solução à *threads* da GPU, capaz de reduzir severamente as porções sequenciais do algoritmo e a transferência de dados para as GPUs, aumentando ainda mais a escalabilidade do algoritmo.
6. Aprimoramento do algoritmo do item 4, adicionando suporte para *clusters* heterogêneos, com um mecanismo automático de balanceamento de carga que determina os tamanhos de entrada mais apropriados para as GPUs disponíveis.

4. Resultados Obtidos

As implementações do algoritmo proposto foram avaliadas utilizando dois computadores, cada um com um processador Intel® Xeon E5-2690 - com 10 núcleos, 25MB de *cache*, 3.0 GHz e 256GB de memória RAM - e quatro placas de vídeo NVIDIA® Tesla K20c, com 2496 núcleos e 5GB de memória global. As implementações estão disponíveis através da URL `https://bitbucket.org/dancarastan/grnhspgpu`. Em todos os experimentos, os tamanhos de entrada adotados foram os tamanhos tipicamente encontrados no problema de inferência de GRNs, com instâncias do HSP com conjunto X de tamanho até 4096 (número de genes) ou instâncias cujas soluções possuam cardinalidade k até 5 (número de preditores por gene-alvo). Os tempos de execução apresentados são uma média dos tempos de execução medidos de dez execuções do algoritmo com instâncias distintas. O parâmetro de otimização `-O3` foi utilizado durante a compilação de todas as implementações os algoritmos. Foram elaborados três experimentos para mensurar a eficiência do algoritmo em relação ao tempo de execução e a escalabilidade em função do número de GPUs. Nesses experimentos foram utilizados dados artificiais gerados aleatóriamente utilizando a função `rand()` da biblioteca padrão da linguagem C, o que permitiu a criação de instâncias de tamanhos e cardinalidades definidos. Abaixo é apresentado um breve resumo dos resultados experimentais obtidos. Informações detalhadas sobre os experimentos podem ser obtidas nos artigos da Seção 4.1.

1. Foi mensurada a eficiência do mecanismo de ordenação para resolver o *Hitting Set*. Para tal, o algoritmo sequencial proposto foi executado com e sem o mecanismo de ordenação. Foi adotada a biblioteca OpenMP para paralelizar em CPU o algoritmo e foram utilizados 10 núcleos de CPU para esse experimento. Nos casos avaliados foram obtidos *speedups* entre 2,2 e 3,5, evidenciando a eficiência desse mecanismo de pré-processamento.
2. Executamos a implementação em GPU utilizando apenas uma GPU e seu tempo de execução foi comparado com o tempo de execução da implementação em

CPU utilizando dez núcleos. Em ambos os casos, foi efetuado o procedimento de ordenação antes de avaliar as soluções. Nesse experimento, levando em consideração as partes paralelizadas do algoritmo, foram obtidos *speedups* entre 1,6 e 4,7 e crescentes com o aumento do número total de genes e com o aumento do número de preditores por gene alvo. Esse resultado mostra que o algoritmo em GPU proposto é capaz de utilizar os recursos da GPU de maneira eficiente.

3. Avaliamos a escalabilidade da implementação utilizando um aglomerado de GPUs. Para isso utilizamos duas máquinas com 4 GPUs cada e executamos o algoritmo para 1, 2, 4 e 8 GPUs. Nas partes paralelizadas do algoritmo, foi obtida uma curva de *speedup* em função da quantidade de GPU crescente, quase linear e com valor de 6,6 para 8 GPUs, mostrando que o algoritmo proposto consegue lidar de maneira eficiente com a adição de mais GPUs para o processamento.

Ao combinarmos as técnicas avaliadas nos experimentos 1, 2 e 3, obtemos *speedups* acima de 60 nas partes paralelizadas do algoritmo, em comparação ao algoritmo sequencial paralelizado em dez núcleos e sem o mecanismo de ordenação.

4.1. Artigos Publicados

- **Artigo publicado nos anais de congresso internacional:** Danilo Carastan-Santos, Raphael Y. de Camargo, David C. Martins-Jr, Siang W. Song, Fabrizio F. Borelli e Luiz C. S. Rozante. A multi-GPU Hitting Set Algorithm for GRNs Inference. In *15th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGrid)*, Shenzhen, Guangdong, China, 2015. DOI: 10.1109/ccgrid.2015.29. Conferência classificada no comitê de CC da CAPES como Qualis A1. Publicamos as contribuições referentes aos itens 1 a 4 da Seção 3.
- **Artigo publicado em periódico:** Danilo Carastan-Santos, Raphael Y. de Camargo, David C. Martins-Jr, Siang W. Song, Luiz C. S. Rozante. Finding exact hitting set solutions for systems biology applications using heterogeneous GPU clusters. *Future Generation Computer Systems.* Elsevier. DOI: 10.1016/j.future.2016.02.009. Periódico com fator de impacto da Journal Citation Reports (JCR) de 2,786 e é classificado no comitê de CC da CAPES como Qualis A2. Publicamos as contribuições referentes aos itens 5 e 6 da Seção 3.

5. Conclusão

Nesse trabalho desenvolvemos um algoritmo exato para resolver o problema do *Hitting Set* aplicado à inferência de GRNs. Foram introduzidas inovações nas estruturas de dados e foi proposto um mecanismo de ordenação que permitiu efetuar um descarte eficiente dos candidatos que não são soluções do problema. Foram solucionadas instâncias do HSP na ordem de milhares de elementos em tempo razoável, alcançando bom desempenho e escalabilidade. Assim, mostramos que é possível resolver o HSP aplicado à inferência de GRNs fazendo o uso de GPUs, em tempo razoável e para entradas típicas no contexto de inferência de GRNs (milhares de genes). É importante observar também que o algoritmo proposto nesse trabalho pode ser adaptado para outras aplicações que possam ser modeladas via HSP e que possuam tamanho de entrada da ordem de milhares de elementos.

Referências

Barrera, J., Cesar-Jr, R. M., Martins-Jr, D. C., Vencio, R. Z. N., Merino, E. F., Yamamoto, M. M., Leonardi, F. G., Pereira, C. A. B., and del Portillo, H. A. (2007). Constructing

probabilistic genetic networks of *Plasmodium falciparum* from dynamical expression signals of the intraerythrocytic development cycle. In *Methods of Microarray Data Analysis V*, chapter 2, pages 11–26. Springer.

Borelli, F. F., Camargo, R. Y., Martins, D. C., Stransky, B., and Rozante, L. C. (2012). Accelerating gene regulatory networks inference through gpu/cuda programming. In *Computational Advances in Bio and Medical Sciences (ICCABS), 2012 IEEE 2nd International Conference on*, pages 1–6. IEEE.

Cendic, B. L. (2014). A genetic algorithm for the minimum hitting set. *Scientific Publications of the State University of Novi Pazar*, 6(2):107.

Garey, M. R. and Johnson, D. S. (1999). *Computers and Intractability - A guide to the Theory of NP-completeness*. W. H. Freeman and Company.

Ideker, T. E., Thorsson, V., and Karp, R. M. (2000). Discovery of regulatory interactions through perturbation: inference and experimental design. *Pacific symposium on biocomputing*, 5:302–313.

Kauffman, S. A. (1969). Homeostasis and differentiation in random genetic control networks. *Nature*, 224(215):177–178.

Kauffman, S. A. (1993). *The Origins of Order*. Oxford University Press, New York.

Kolman, P. and Walen, T. (2007). Reversal distance for strings with duplicates: Linear time approximation using hitting set. *The Electronic Journal of Combinatorics*, 14(1):11.

Madhamshettiwar, P. B., Maetschke, S. R., Davis, M. J., Reverter, A., and Ragan, M. A. (2012). Gene regulatory network inference: evaluation and application to ovarian cancer allows the prioritization of drug targets. *Genome medicine*, 4(5):1–16.

Pearson, W. R., Robins, G., Wrege, D. E., and Zhang, T. (1996). On the primer selection problem in polymerase chain reaction experiments. *Discrete Applied Mathematics*, 71(1):231–246.

Ristevski, B. (2013). A survey of models for inference of gene regulatory networks. *Nonlinear Analysis: Modelling and Control*, 18(4):444–465.

Ruchkys, D. P. and Song, S. W. (2003). A parallel solution to infer genetic network architectures in gene expression analysis. *International Journal of High Performance Computing Applications*, 17(2):163–172.

Shi, H., Schmidt, B., Liu, W., and Muller-Wittig, W. (2011). Parallel mutual information estimation for inferring gene regulatory networks on gpus. *BMC Research Notes*, 4:189.

Shi, L. and Cai, X. (2010). An exact fast algorithm for minimum hitting set. *Int. Joint Conference on Computational Science and Optimization*, 1:64–67.

Steinbach, B. and Posthoff, C. (2012). Sources and obstacles for parallelization-a comprehensive exploration of the unate covering problem using both CPU and GPU. *GPU Computing with Applications in Digital Logic*, page 63.