



UMA REVISÃO SISTEMÁTICA SOBRE ALGORITMOS PARALELOS PARA OS PROBLEMAS DE REDUÇÕES DE LARGURA DE BANDA E DE *PROFILE* DE MATRIZES

Jean Antonio Ribeiro

Universidade Federal de Lavras - UFLA

Lavras - MG

jeankusanagi@posgrad.ufla.br

Sanderson L. Gonzaga de Oliveira

Universidade Federal de Lavras - UFLA

Lavras - MG

sanderson@dcc.ufla.br

RESUMO

Neste trabalho, é apresentada uma revisão sistemática de heurísticas paralelas para reduções de largura de banda e de *profile* de matrizes simétricas e esparsas. Foram identificadas 13 heurísticas paralelas. Nesses trabalhos, a maioria dos pesquisadores compararam os resultados experimentais entre as heurísticas paralelas e sequenciais. Somente em duas publicações os autores compararam suas heurísticas paralelas com outras heurísticas paralelas. Com isso, não é possível reconhecer o estado da arte nos problemas. Nesta revisão, essas publicações são analisadas, as abordagens adotadas para a paralelização das heurísticas são identificadas, bem como quais bibliotecas paralelas foram utilizadas para a paralelização dos algoritmos.

PALAVRAS CHAVE. redução de largura de banda, redução de *profile*, computação paralela.

Otimização combinatória.

ABSTRACT

This work shows a systematic review of parallel heuristics for bandwidth and profile reductions of symmetric sparse matrices. We identified 13 parallel heuristics for these problems. In these publications, most of the researchers compared the results between their parallel and sequential heuristics. Only in two publications the authors compared the results of their parallel heuristics with other parallel algorithms. Thus, it is not possible to recognize the state-of-the-art heuristics for addressing the problems. In this review, these publications are analyzed, the adopted approaches for the parallelization of the heuristics are identified, and the parallel libraries that were used for the parallelization of the algorithms are listed.

KEYWORDS. bandwidth reduction, profile reduction, parallel computing.

Combinatorial optimization.



1. Introdução

Uma variedade de problemas científicos, de engenharia e industriais, são descritos por equações diferenciais parciais (EDPs). Em geral, na resolução numérica de EDPs elípticas e parabólicas, são gerados sistemas de equação lineares. As matrizes contidas nesses sistemas de equações lineares são de grande porte e esparsas. Tais sistemas são provenientes da utilização de métodos como os dos elementos finitos (MEF), das diferenças finitas e dos volumes finitos [Gonzaga de Oliveira e Chagas, 2014, p. 3-12].

Para a resolução de um sistema de equações lineares, podem ser aplicados tanto métodos diretos como iterativos. A eliminação de Gauss e a decomposição LU são exemplos de métodos diretos. Para métodos iterativos, o método mais popular é o método dos gradientes conjugados (MGC) [Hestenes e Stiefel, 1952; Lanczos, 1952], indicado para resolução de sistemas de equações lineares de grande porte, nas ocorrências em que a matriz dos coeficientes é simétrica e positiva definida. O custo computacional exigido nessa resolução, seja por métodos diretos ou iterativos, é alto quando aplicado a sistemas de equações lineares de grande porte, tornando-se o verdadeiro gargalo em tais simulações computacionais.

Com o objetivo de reduzir o tempo de execução de métodos iterativos ou diretos na resolução de sistemas de equações lineares, pode ser realizado o reordenamento das linhas e colunas da matriz esparsa, antes de serem realizados os cálculos matriciais. Esse reordenamento é realizado como um passo de pré-processamento na matriz de coeficientes. Há métodos que realizam esse reordenamento para reduzir a largura de banda ou o *profile* de matrizes. A redução de largura de banda de matrizes consiste em realizar permutações de linhas e colunas, deixando-a com uma estrutura compacta, de modo que os coeficientes não nulos estejam próximos à diagonal principal. Desse modo, a banda de uma matriz é definida como a banda diagonal que compreende as diagonais da matriz, contendo coeficientes não nulos mais distantes da diagonal principal em ambas as direções. Por sua vez, largura de banda de uma matriz simétrica é a quantidade de subdiagonais da matriz triangular inferior ou da superior [Gonzaga de Oliveira e Chagas, 2014, p. 9]. O *profile* é definido como o somatório das larguras de banda de cada linha da matriz [Gonzaga de Oliveira e Chagas, 2014, p. 10].

Os problemas de minimização de largura de banda e de *profile* são pertencentes à classe NP-Difícil [Papadimitriou, 1976; Lin e Yuan, 1994]. Por causa da importância desses problemas, existe uma grande variedade de heurísticas para reduções de largura de banda e de *profile* de matrizes [Bernardes e Gonzaga de Oliveira, 2015; Chagas e Gonzaga de Oliveira, 2015; Gonzaga de Oliveira e Chagas, 2015; Gonzaga de Oliveira et al., 2016a,b, 2017]. Atualmente, com a utilização de processamento paralelo, pode-se obter custo de processamento muito menor nas resoluções, especialmente quando os tamanhos dos problemas tratados são muito grandes e impedem a obtenção de resultados precisos em tempos de computação aceitáveis por algoritmos sequenciais. Assim, neste trabalho, é apresentada uma revisão sistemática de algoritmos paralelos para os problemas de reduções de largura de banda e de *profile* de matrizes.

A seguir, na Seção 2, mostra-se como foi realizada a revisão sistemática. Na Seção 3, são descritas, de forma breve, as abordagens paralelas das heurísticas para reduzir largura de banda ou *profile* de matrizes, encontradas durante a realização da revisão sistemática da literatura. Por fim, a conclusão deste texto é apresentada na Seção 4.

2. Revisão sistemática

O presente estudo sobre implementações paralelas de heurísticas para reduções de largura de banda e de *profile* de matrizes teve início em abril de 2016. Trata-se de uma revisão sistemática realizada na base de dados *Scopus*[®]. A busca na base foi realizada utilizando a seguinte chave de busca: ((*TITLE-ABS-KEY(bandwidth)* OR *TITLE-ABS-KEY(profile)*) AND (*TITLE-ABS-KEY(reduction)* OR *TITLE-ABS-KEY(reordering)*) AND *TITLE-ABS-KEY(matrix)* AND *TITLE-ABS-KEY(parallel)*) OR (*TITLE-ABS-KEY(ordering)* AND *TITLE-ABS-KEY(preconditioned conjugate gradient)* AND *TITLE-ABS-KEY(parallel)*).



Essa busca resultou em 164 artigos. Em cada um dos artigos recuperados na busca, foram verificados título, resumo e palavras-chave. Somente abordagens paralelas de métodos para reduzir largura de banda ou *profile* de matrizes foram selecionadas entre os artigos recuperados. Além disso, foi realizada uma busca na ferramenta de pesquisa *Google Scholar* e foram identificadas as publicações de [Rodrigues et al., 2016a] e [Rodrigues et al., 2016b]. Dessa maneira, os dados foram extraídos de forma a adquirir uma comparação clara entre cada uma das publicações. Com isso, foram selecionados 13 artigos para serem analisados detalhadamente.

Na Tabela 1, são listadas as publicações com algoritmos paralelos para reduções de largura de banda ou de *profile* encontrados na literatura. Na primeira coluna, estão listados 13 algoritmos paralelos para reduzir largura de banda ou *profile* de matrizes esparsas. Na segunda coluna, são listadas as bibliotecas utilizadas para a paralelização dos algoritmos. Na terceira e última colunas, são listadas as heurísticas sequenciais (ou técnica utilizada) que foram paralelizadas para reduzir largura de banda e/ou *profile* de matrizes esparsas, respectivamente. Na Tabela 2, são listadas informações sobre as instâncias de testes utilizadas pelos autores em seus respectivos artigos.

Algoritmo paralelo (publicação)	Biblioteca utilizada	Redução de largura de banda (publicação)	Redução de <i>profile</i> (publicação)
[Prasad et al., 1992]	-	[Puttonen, 1983]	-
[Esposito e Tarricone, 1996]	CRAY Shared Memory Access e Parallel Virtual Machine	Busca tabu [Reeves, 1993]	-
[Tarricone, 2000]	Parallel Virtual Machine	GA [Tarricone, 2000]	-
[Manguoglu et al., 2010]	METIS e ParMETIS	Reordenação espectral ponderada [Barnard et al., 1995] / RCM [George, 1971]	-
[Gibou e Min, 2012]	OpenMP	[Cuthill e McKee, 1969]	-
[Deb e Srirama, 2014]	pacote GA em R	[Cuthill e McKee, 1969]	-
[Karantasis et al., 2014]	Galois	RCM [George, 1971]	[Sloan, 1989]
[Tsuburaya et al., 2015]	OpenMP	RCM [George, 1971]	-
[Rodrigues et al., 2016a]	OpenMP	L-RCM [Chan e George, 1980]	-
[Rodrigues et al., 2016b]	OpenMP	L-RCM [Chan e George, 1980]	-
[Maftéiu-Scai e Cornigeanu, 2016a]	MPI	[Maftéiu-Scai e Cornigeanu, 2016a]	-
[Maftéiu-Scai e Cornigeanu, 2016b]	MPI	[Maftéiu-Scai e Cornigeanu, 2016b]	-
[Azad et al., 2017]	MPI e OpenMP	RCM [George, 1971]	-

Tabela 1: Publicações com algoritmos paralelos para reduções de largura de banda e de *profile* de matrizes.

Algoritmo paralelo (publicação)	Dimensão mínima	Dimensão máxima	Quantidade mínima de coeficientes	Quantidade máxima de coeficientes
[Prasad et al., 1992]	-	-	-	-
[Esposito e Tarricone, 1996]	500	1.000	-	-
[Tarricone, 2000]	284	1.231	-	-
[Manguoglu et al., 2010]	7.548	2.063.494	41.746	14.612.663
[Gibou e Min, 2012]	-	-	-	-
[Deb e Srirama, 2014]	-	-	-	-
[Karantasis et al., 2014]	503.712	23.947.347	7.660.826	100.663.202
[Tsuburaya et al., 2015]	93.879	381.197	34.396.220	82.050.304
[Rodrigues et al., 2016a]	146.689	1.585.478	3.636.643	89.306.020
[Rodrigues et al., 2016b]	8.192	1.585.478	41.764	89.306.020
[Maftéiu-Scai e Cornigeanu, 2016a]	936	12.452	-	-
[Maftéiu-Scai e Cornigeanu, 2016b]	24	23.133	-	-
[Azad et al., 2017]	72.000	27.993.600	28.715.634	760.648.352

Tabela 2: Informações sobre as instâncias de testes utilizadas pelos autores de algoritmos paralelos para reduções de largura de banda e de *profile* de matrizes.

Esse é um tópico de pesquisa recente. Na Figura 1, enfatiza-se que, entre as 13 heurísticas



paralelas identificadas, nove foram publicadas nos últimos cinco anos.

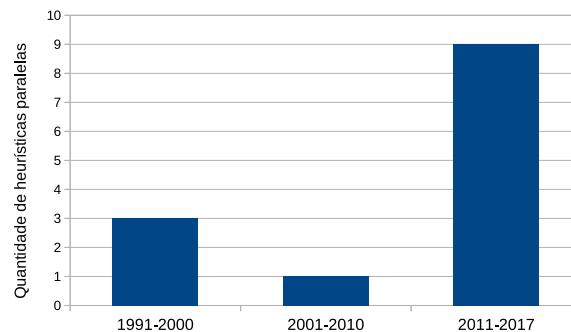


Figura 1: Quantidade de heurísticas paralelas para reduções de largura de banda e/ou de *profile* de matrizes, de 1991 a 2017.

Como descrito, na maioria dessas publicações, os autores comparam o algoritmo paralelo com a versão sequencial, e não compararam o algoritmo paralelo proposto com outros algoritmos paralelos. Em [Rodrigues et al., 2016b], os autores mostraram que seu algoritmo paralelo apresentou desempenho computacional melhor que o algoritmo paralelo proposto em [Karantasis et al., 2014]. Em [Maftieu-Scai e Cornigeanu, 2016b], os autores apresentaram duas heurísticas paralelas, comparando os resultados experimentais entre elas. Como não há na literatura avaliações entre os 13 algoritmos identificados, não é possível reconhecer o estado da arte nos problemas.

3. Heurísticas paralelas para reduções de largura de banda e de *profile* de matrizes

Nesta seção, são descritas, de forma breve, 13 heurísticas paralelas para reduzir largura de banda ou *profile* de matrizes identificadas na literatura. Além disso, são abordadas as comparações, realizadas pelos pesquisadores, dos resultados apresentados pelas heurísticas paralelas identificadas nesta revisão sistemática.

3.1. Proposta de Prasad e colaboradores

Em [Prasad et al., 1992], os autores introduziram um método paralelo para a redução da largura de banda de matrizes simétricas. Trata-se de uma permutação realizada com linhas e colunas de matrizes projetada para o mapeamento de matrizes lineares em arquiteturas *Single Instruction, Multiple Data*. Esse método é uma versão modificada do método sequencial publicado em [Puttonen, 1983], também utilizado para redução de largura de banda de matrizes simétricas. Na resolução de sistemas de equações lineares, foi utilizada a decomposição LU. A paralelização do método foi apenas introduzida e mostrada possível de ser utilizada por meio de análise de complexidade, ou seja, o método em paralelo não foi implementado.

3.2. Proposta de Esposito e Tarricone

Em [Esposito e Tarricone, 1996], os autores descreveram um método paralelo para redução de largura de banda de matrizes esparsas, baseado na meta-heurística busca tabu [Reeves, 1993]. Por meio de uma função de custo, uma pontuação é calculada para cada vizinho e a melhor delas é selecionada como a solução na iteração seguinte. A função de custo, nesse caso, é representada pela soma ponderada da largura de banda ou do *profile* da matriz de coeficientes.

A implementação do método em paralelo foi realizada e testada nas plataformas IBM SP2 e Cineca CRAY T3D. Na plataforma SP2, foi testado um paradigma de programação mestre-escravo e uma abordagem *Single Program, Multiple Data* (SPMD). Para a implementação dos códigos, foi utilizada a interface de programação *Parallel Virtual Machine*. Na plataforma T3D, o paradigma de programação utilizado foi SPMD com memória compartilhada. A comunicação entre os processos foi implementada com a biblioteca *CRAY Shared Memory Access*, utilizando um recurso de mapeamento lógico com memória compartilhada. Para ambas as plataformas em que essas duas abordagens foram projetadas, há várias abordagens possíveis de paralelização da meta-heurística.



Uma forma possível consiste na execução de várias buscas em paralelo e na comparação periódica dos resultados. A busca continua em todos os processos, a partir da melhor solução encontrada até o momento. Cada pesquisa se diferencia uma da outra por alguns parâmetros de ajustes. Outra técnica utilizada nessa proposta consiste em distribuir partições do problema entre os processos e intercalar os resultados em uma única solução.

Foram aplicados dois métodos numéricos na resolução de sistemas de equações lineares. Um deles, chamado de *Mode-Matching* [Tarricone et al., 1995], alcançou altos *speedups* ao utilizar uma estratégia de renumeração apropriada. O outro, baseado no método dos elementos finitos, oferece pacotes apropriados com redutores de largura de banda. Ao apresentar soluções ineficientes, em que a largura de banda não foi reduzida, dois níveis de otimizações heurísticas foram propostos para a estratégia utilizando o métodos de elementos finitos: *otimizações intermediária e fina*. Essas abordagens diferem basicamente na quantidade de iterações executadas e no critério de parada.

Nos resultados experimentais, foi verificado que o método paralelo apresentou resultados (computacionalmente) eficientes em aplicações reais. A estratégia utilizando o método numérico *Mode-Matching* foi mais eficaz e escalável do que a estratégia utilizando o método dos elementos finitos, garantindo um equilíbrio de carga na execução da aplicação.

3.3. Proposta de Tarricone

Em [Tarricone, 2000], o autor apresentou uma solução por algoritmos genéticos para reduzir largura de banda de matrizes esparsas. Nesse método, a geração da população inicial foi realizada em paralelo e os resultados da pesquisa evolutiva, de cada população, foram coletados periodicamente, de modo que o cruzamento e as mutações foram realizados sobre os cromossomos de populações diferentes com um aumento do nível da hibridação. Em seguida, a função de custo é computada em paralelo, ao realizar uma decomposição em blocos adequada das matrizes e dos vetores de permutações. Contudo, a tarefa de computar a função de custo é difícil de ser realizada, especialmente quando o tamanho do problema é grande.

Aplicado em problemas de modelagem de circuitos eletromagnéticos, os resultados apresentados pelo método paralelo baseado em algoritmos genéticos foram comparados com resultados apresentados pelo pacote MEF do *software* EMAP1, ambas as abordagens em conjunto com um método direto de resolução de sistemas de equações lineares para matrizes em banda. O método paralelo baseado em algoritmos genéticos foi implementado utilizando a interface de programação *Parallel Virtual Machine*, em uma máquina IBM SP2 com oito núcleos. Segundo os autores, os resultados, em relação ao desempenho computacional e eficácia, apresentados pelo método paralelo baseado em algoritmos genéticos foi superior em relação aos resultados apresentados por algoritmos contidos no pacote MEF do *software* EMAP1.

3.4. Abordagem de Manguoglu e colaboradores

Em [Manguoglu et al., 2010], os autores apresentaram uma estratégia de reordenamento das linhas e colunas de uma matriz de tal maneira que é possível obter um preconditionador para resolutores iterativos de sistemas de equações lineares. Para isso, é utilizada uma reordenação não simétrica para mover os coeficientes para perto da diagonal principal. Uma reordenação espectral ponderada é utilizada para mover os maiores coeficientes para perto da diagonal principal. Desse modo, uma matriz com largura de banda pequena é obtida para ser utilizada como um preconditionador para métodos baseados no subespaço de *Krylov*, deixando de fora da banda diagonal os coeficientes de menor magnitude.

Na versão paralela do método, a matriz foi particionada em blocos para que fossem distribuídos entre os processadores, combinando os resultados de cada processador no final da computação. Nesse caso, o tamanho reduzido do sistema de equações lineares é determinado pela largura de banda da matriz e pela quantidade de partições da matriz.

De acordo com os autores, essa abordagem superou os resultados apresentados por resolutores de sistemas de equações lineares dos pacotes MUMPS e SuperLU. Nesse caso, o reordenamento das linhas e colunas da matriz foi realizado por funções contidas nas bibliotecas METIS e



ParMETIS, respectivamente. Por ser um esquema de condicionamento de matrizes em banda, a estratégia em paralelo foi considerada mais robusta que uma classe de condicionadores baseados em fatoração incompleta, já que proporcionou um desempenho computacional melhor, tornando-se escalável em arquiteturas paralelas.

3.5. Abordagem de Gibou e Min

Em [Gibou e Min, 2012], os autores aplicaram a reordenação de linhas e colunas da matriz pelo método Cuthill-McKee [Cuthill e McKee, 1969], em conjunto com o método dos gradientes conjugados condicionado paralelo, para resolver a equação de Poisson em domínios irregulares. O condicionador paralelo utilizado foi o ILU.

Dada a estrutura de nível gerada a partir de um vértice inicial, a ordenação dos vértices do grafo associado à matriz A é aplicada de modo incremental, ou seja, a ordenação é iniciada a partir dos níveis com menor quantidade de vértices para os níveis com maior quantidade de vértices. Além disso, vértices pertencentes a um mesmo nível são ordenados lexicograficamente. Dessa forma, foi possível obter um equilíbrio entre paralelismo e condicionamento.

O método Cuthill-McKee permite uma implementação paralela natural dos blocos diagonais da matriz correspondente ao grafo e o condicionamento é tão eficiente quanto a ordenação lexicográfica. O método Cuthill-McKee foi implementado totalmente em paralelo. A ordenação lexicográfica também foi implementada em paralelo; contudo, o condicionador ILU foi implementado sequencialmente. Nessa abordagem, os autores mostraram que a aplicação é escalável em arquiteturas *multicore*.

3.6. Proposta de Deb e Srirama

Em [Deb e Srirama, 2014], os autores projetaram um algoritmo genético paralelo baseado em uma técnica de agrupamento *two-mode* que, por sua vez, é uma adaptação do método Cuthill-McKee [Cuthill e McKee, 1969]. O problema da escalabilidade da organização de um conjunto de dados dispostos em matrizes de dados *two-mode* foi abordado utilizando *clusters* com processadores *multicores*. Como exemplo de matrizes de dados *two-mode*, tem-se os dados gerados a partir de experiências com DNA, RNA e micromatrizes de proteínas. Nessas matrizes, as linhas podem estar relacionadas aos genes e as colunas podem ser representações das particularidades de cada gene.

Na paralelização do método, foi utilizado um pacote de algoritmos genéticos em R [Scrucca, 2013], baseado no padrão de paralelismo *Simple Network of Workstations*. Nessa abordagem, a codificação dos cromossomos e os dados são enviados para cada um dos núcleos. Em seguida, o algoritmo genético paralelo é executado em cada um dos núcleos, com diferentes valores atribuídos aos parâmetros de entrada.

O método foi projetado por um modelo de programação mestre-escravo, em que o núcleo mestre envia tarefas para os núcleos trabalhadores, que executam as tarefas e retornam os resultados para o mestre. Por sua vez, o núcleo mestre agrega os resultados e seleciona a solução com menor largura de banda.

Nessa abordagem, os autores consideraram que o algoritmo paralelo é escalável em processadores *multicore* quando comparado com sua implementação sequencial. A consistência dos resultados foi verificada com base no valor médio do tempo de execução do método paralelo baseado em algoritmos genéticos executado cinco vezes. Além disso, foi verificado que a aceleração proporcionada pelo algoritmo genético paralelo foi aumentada de 1 para 2,42, 3,33, 4,0, 5,74 e 6,49 com processamento de dois a seis núcleos, respectivamente.

3.7. Proposta de Karantasis e colaboradores

Em [Karantasis et al., 2014], os autores apresentaram implementações em paralelo aplicadas em ambiente *multicore*, de dois algoritmos de reordenação: *Reverse Cuthill-McKee* (RCM) [George, 1971] (apesar de os autores referenciam o método Cuthill-McKee, os autores descrevem o método RCM) e [Sloan, 1989]. Para encontrar vértices pseudo-periféricos, os autores utilizaram o algoritmo descrito em [Kumfert, 2000], que é o algoritmo de Sloan com algumas melhorias.



As paralelizações de ambos os métodos foram baseadas na paralelização da busca em largura não ordenada. Além disso, a biblioteca Galois foi utilizada na otimização dos algoritmos. Uma otimização adotada foi dividir um conjunto de iteradores ordenados em um conjunto de iteradores não ordenados. Em um conjunto de iteradores não ordenados, cada iteração pode ser paralelizada processando múltiplas iterações de uma só vez, verificando se iterações simultâneas acessam os mesmos dados. Por outro lado, um conjunto de iterações ordenadas é como um conjunto de iterações não ordenadas com a restrição de que a serialização das iterações deverá ser consistente com uma relação de ordenação definida pelo usuário.

Foram utilizadas duas abordagens na paralelização do método RCM. Na primeira, chamada de algoritmo nivelado, foi utilizada uma abordagem incremental, ou seja, em cada passo do algoritmo, é realizada a permutação dos vértices, que não se altera em passos subsequentes. Na segunda, chamada de método RCM não ordenado, é gerada uma permutação direta em relação ao método RCM, mas que só é válida após a geração da estrutura de nível.

Na paralelização da heurística de Sloan, os autores propuseram uma abordagem em que a ordem de prioridade dos vértices visitados é relaxada. Para isso, utilizaram uma fila de prioridade aproximada, fornecida pela biblioteca Galois. Nessa biblioteca, os vértices são retornados em ordem de prioridade, mas podem admitir um certo número de inversões de prioridade quando um vértice é recuperado fora da ordem estrita. Para evitar conflitos na atualização dos vértices do grafo, cada atualização foi realizada de forma atômica, por meio de instruções de comparação e troca.

Em [Karantasis et al., 2014], os autores compararam a qualidade das soluções apresentadas pela paralelização das heurísticas *Reverse Cuthill-McKee* [George, 1971] e [Sloan, 1989] com os resultados decorrentes de uma implementação baseada na biblioteca de funções matemáticas HSL [HSL, 2017]. Foi verificado que a execução dos algoritmos paralelos alcançou desempenho computacional melhor em relação ao desempenho computacional apresentado pelas implementações da biblioteca HSL. Além disso, a qualidade das soluções permaneceu a mesma em todos os casos.

3.8. Proposta de Tsuburaya e colaboradores

Em [Tsuburaya et al., 2015], os autores apresentaram um método, chamado de *Reverse Block-Multicolor* (RBMC), para reordenação de linhas e colunas da matriz, baseado na coloração multicolor dos vértices e no método RCM. O objetivo foi melhorar o desempenho computacional da reordenação multicolor em blocos na paralelização do preconditionador incompleto de Cholesky em conjunto com o método dos gradientes conjugados. Como a quantidade de vértices em cada nível se torna não uniforme, o balanceamento de carga no preconditionamento se torna também não uniforme. Desse modo, o método RBMC foi combinado com a técnica de decomposição da matriz de coeficientes em blocos, baseada na árvore rubro-negro [Iwashita e Shimasaki, 2003; Iwashita et al., 2005].

O objetivo dessa abordagem foi reduzir a quantidade de sincronizações progressiva e retroativa despendidas na resolução de sistemas de equações lineares. Adicionalmente, um balanceamento de carga uniforme no preconditionamento foi conseguido por meio do método RBMC paralelo.

O método RBMC paralelo foi avaliado ao ser comparado com os métodos de reordenação multicolor [Adams e Jordan, 1986] e reordenação multicolor em blocos [Iwashita et al., 2012; Semba et al., 2013]. Como resultado, o método RBMC foi considerado computacionalmente mais eficaz que os demais métodos de reordenação testados, devido à redução na quantidade de sincronização nas substituições progressiva e retroativa e na melhoria do balanceamento de carga realizado pelo preconditionamento.

3.9. Propostas de Rodrigues e colaboradores

Em [Rodrigues et al., 2016a], os autores descreveram uma implementação paralela do método RCM não ordenado, chamada pelos autores de RCM *Unordered*, que é comparado com sua versão sequencial. Para o conjunto de matrizes testadas, a redução de tempo de computação obtido



variou entre 78% e 94%. Os autores também relataram que o método RCM *Unordered* paralelo obteve *speedups* de 2 a 17 vezes em relação a versão sequencial do algoritmo. Sobre a qualidade das soluções, a redução de largura de banda alcançada foi a mesma para as implementações sequenciais e paralelas.

No método RCM *Unordered*, a geração da estrutura de nível é realizada utilizando um algoritmo de busca em largura não ordenado. Com a criação da estrutura de nível, o algoritmo mantém uma lista de trabalho *ws* de vértices, a partir da qual qualquer vértice pode ser selecionado. Assim, vários vértices podem ser computados em paralelo. Para cada iteração do algoritmo na lista *ws*, uma *thread* remove um vértice arbitrário *v* de *ws* e calcula o nível apropriado para os vértices adjacentes de *v*.

Em [Rodrigues et al., 2016b], os autores apresentaram uma implementação paralela e otimizada do método *Leveled Reverse Cuthill-McKee* (L-RCM) [Chan e George, 1980], chamada pelos autores de OL-RCM. Para a determinação dos vértices pseudo-periféricos, os autores utilizaram o algoritmo descrito em [Kumfert, 2000]. A representação das matrizes foi realizada por meio de *buckets*. Para evitar conflitos no algoritmo otimizado, a operação de determinar o nível de cada vértice na estrutura de nível é realizada de forma atômica. Além disso, a maioria das operações realizadas em cada vértice ocorre no *bucket*. Como exemplo, cada *thread* utiliza a quantidade de vértices no *bucket* para determinar os rótulos dos vértices. Com isso, o custo envolvido no acesso a um *bucket* está relacionado com operações executadas por uma função *hash*, adicionado ao custo de acessar uma posição específica no *bucket*.

No método OL-RCM, os autores compararam os resultados de sua implementação paralela com os resultados apresentados pela sua respectiva versão sequencial e pelo método RCM paralelo descrito em [Karantasis et al., 2014]. Nesse caso, os resultados experimentais apresentados pela abordagem OL-RCM foram considerados melhores que os outros métodos avaliados.

3.10. Propostas de Mafteiu-Scai e Cornigeanu

Em [Mafteiu-Scai e Cornigeanu, 2016a], os autores apresentaram uma heurística paralela e híbrida para reduzir largura de banda de matrizes esparsas e simétricas. Nessa heurística, é utilizada uma seleção gulosa de linhas/colunas para serem permutadas, dependendo da posição dos coeficientes não nulos localizados nas extremidades de cada linha/coluna e de alguns parâmetros relacionados com a matriz de coeficientes.

Em [Mafteiu-Scai e Cornigeanu, 2016b], os autores apresentaram duas heurísticas paralelas e híbridas, utilizadas para redução da largura de banda média de matrizes esparsas. Segundo os autores, a largura de banda da matriz não fornece informações suficientes sobre o padrão de agrupamento de coeficientes não nulos ao redor da diagonal principal. Com isso, a largura de banda média da matriz foi definida como a razão entre a largura de banda e a quantidade de coeficientes não nulos da matriz.

Nessa abordagem, o objetivo é obter uma distribuição uniforme dos coeficientes não nulos ao redor da diagonal principal. Com base em um processamento direto na matriz e inspirada em leis da física, a primeira heurística realiza uma seleção gulosa de linhas e colunas da matriz para realizar a permutação. Na segunda heurística, o objetivo foi melhorar a primeira heurística por meio da utilização de uma fórmula exata, determinando as permutações mais adequadas na matriz.

Na paralelização dos algoritmos, foi utilizado o modelo mestre-escravo. Para isso, as matrizes foram divididas em blocos compactos de tamanhos iguais, dependendo da quantidade de processos utilizados. Além disso, foi utilizado um modelo síncrono de paralelização, produzindo um balanceamento de carga entre os processos escravos.

Todos os algoritmos foram implementados em um supercomputador *IBM Blue Gene/P*, utilizando o compilador *IBM C++*. A biblioteca *MPI* foi empregada para a comunicação entre processos. Para o armazenamento das matrizes, na abordagem proposta em [Mafteiu-Scai e Cornigeanu, 2016b], os autores utilizaram o formato de coordenadas e, na abordagem proposta em [Mafteiu-Scai e Cornigeanu, 2016a], foi utilizada uma variante desse formato, o *Matrix Market*.



Segundo os autores, os resultados experimentais apresentados pelas heurísticas paralelas foram considerados melhores quando comparados com os resultados experimentais apresentados pelas heurísticas sequenciais. Além disso, o *speedup* da segunda heurística foi inferior ao *speedup* da primeira heurística por causa das tarefas paralelas adicionais no processo mestre e da comunicação realizada entre os processos.

3.11. Proposta de Azad e colaboradores

Em [Azad et al., 2017], os autores apresentaram uma implementação paralela do método RCM com memória distribuída para reduzir o *profile* de matrizes esparsas. Para encontrar um vértice pseudo-periférico, foi utilizado o algoritmo de Gibbs, Poole e Stockmeyer [Gibbs et al., 1976]. Na paralelização do método, foi utilizada uma decomposição da matriz de coeficientes em submatrizes, de modo que o desempenho computacional melhorasse em relação ao algoritmo sequencial. Além disso, operações em grafos não estruturados foram substituídas por operações algébricas de matrizes por vetores. Com isso, foi apresentado um algoritmo baseado em primitivas paralelas síncronas e otimizadas para serem executadas em sistemas com memória compartilhada e distribuídas.

Para a distribuição das matrizes e dos vetores esparsos entre os processadores, foi utilizado o *framework Combinatorial BLAS* (CombBLAS) [Buluç e Gilbert, 2011]. Além disso, foram utilizadas as bibliotecas OpenMP e MPI. A implementação com essas ferramentas atingiu um *speedup* satisfatório em matrizes originadas de várias aplicações executadas em um supercomputador Cray XC30 com 1024 núcleos. Somente uma *thread* em cada processo realizou chamadas MPI para comunicação entre os nós da rede.

Os autores mostraram que a implementação é escalável em computadores com até 1024 núcleos para matrizes de pequeno porte e até 4096 núcleos para matrizes de grande porte. A memória distribuída, utilizada pela primitiva SPMSPV é o passo mais caro na implementação desse algoritmo paralelo. O tempo de comunicação despendido pela primitiva SPMSPV começa a dominar o tempo de computação quando a concorrência é maior. Além disso, grafos com diâmetros maiores possuem uma sobrecarga de comunicação maior do que grafos com diâmetros menores.

4. Conclusão

Neste trabalho, mostra-se uma revisão sistemática de algoritmos paralelos para os problemas de reduções de largura de banda e de *profile* de matrizes esparsas. Foram identificadas 13 heurísticas paralelas para esses problemas. A maioria dessas heurísticas foi baseada no método *Reverse Cuthill-McKee* [George, 1971]. Apenas a abordagem apresentada em [Karantasis et al., 2014] está relacionada com a heurística de [Sloan, 1989], utilizada para a redução de *profile* de matrizes esparsas.

É possível que existam outras heurísticas paralelas para reduções de largura de banda e de *profile* de matrizes; contudo, é altamente provável que os resultados das principais heurísticas foram considerados nesta revisão sistemática. O próximo passo deste trabalho é implementar essas heurísticas paralelas para verificar o(s) algoritmo(s) no estado da arte nos problemas.

Com os resultados experimentais dessas heurísticas paralelas, em relação à eficiência computacional e qualidade das soluções, espera-se compará-los com os resultados de outras possíveis heurísticas a serem paralelizadas. As heurísticas mais promissoras para serem implementadas em paralelo foram identificadas em [Bernardes e Gonzaga de Oliveira, 2015; Chagas e Gonzaga de Oliveira, 2015; Gonzaga de Oliveira e Chagas, 2015; Gonzaga de Oliveira et al., 2016a]. Algumas delas são: RCM-GL [George e Liu, 1981], Sloan [Sloan, 1989], MPG [Medeiros et al., 1993], NSloan [Kumfert e Pothén, 1997] e hGPHH-GL [Koohestani e Poli, 2011].

5. Agradecimentos

Este trabalho foi realizado com o apoio da FAPEMIG - Fundação de Amparo à Pesquisa do Estado de Minas Gerais. Agradecemos a(o)s revisore(a)s as valiosas sugestões e comentários pertinentes de forma a contribuir para melhorar a qualidade deste artigo.



Referências

- Adams, L. M. e Jordan, H. F. (1986). Is SOR color-blind? *SIAM Journal on Scientific and Statistical Computing*, 7(2):490–506.
- Azad, A., Jacquelin, M., Buluç, A., e Ng, E. G. (2017). The reverse Cuthill-McKee algorithm in distributed-memory. In *Proceedings of the IPDPS*, Florida, USA.
- Barnard, S. T., Pothén, A., e Simon, H. (1995). A spectral algorithm for envelope reduction of sparse matrices. *Numerical Linear Algebra with Applications*, 2(4):317–334.
- Bernardes, J. A. B. e Gonzaga de Oliveira, S. L. (2015). A systematic review of heuristics for profile reduction of symmetric matrices. *Procedia Computer Science*, 51:221 – 230.
- Buluç, A. e Gilbert, J. R. (2011). The combinatorial blas: design, implementation, and applications. *The International Journal of High Performance Computing Applications*, 25(4):496–509.
- Chagas, G. O. e Gonzaga de Oliveira, S. L. (2015). Metaheuristic-based heuristics for symmetric-matrix bandwidth reduction: A systematic review. *Procedia Computer Science*, 51:211 – 220.
- Chan, W. M. e George, A. (1980). A linear time implementation of the Reverse Cuthill-McKee algorithm. *BIT Numerical Mathematics*, 20(1):8–14.
- Cuthill, E. e McKee, J. (1969). Reducing the bandwidth of sparse symmetric matrices. In *Proceedings of the 1969 24th National Conference, ACM '69*, p. 157–172, New York, NY, USA. ACM.
- Deb, B. e Srirama, S. N. (2014). Scalability of parallel genetic algorithm for two-mode clustering. *International Journal of Computer and Applications*, 94(14):23–26.
- Esposito, A. e Tarricone, L. (1996). Parallel heuristics for bandwidth reduction of sparse matrices with IBM SP2 and Cray T3D. In Waśniewski, J., Dongarra, J., Madsen, K., e Olesen, D., editors, *Applied Parallel Computing Industrial Computation and Optimization*, p. 239–246. Springer Berlin Heidelberg.
- George, A. e Liu, J. W. H. (1981). *Computer Solution of Large Sparse Positive Definite Systems*. Prentice-Hall, New York, USA.
- George, J. A. (1971). *Computer implementation of the finite element method*. PhD thesis, Computer Science Department, Stanford University, USA.
- Gibbs, N. E., Poole, W. G., Jr, e Stockmeyer, P. K. (1976). An algorithm for reducing the bandwidth and profile of a sparse matrix. *SIAM Journal on Numerical Analysis*, 13(2):236–250.
- Gibou, F. e Min, C. (2012). On the performance of a simple parallel implementation of the ILU-PCG for the Poisson equation on irregular domains. *Journal of Computational Physics*, 231(14): 4531 – 4536.
- Gonzaga de Oliveira, S. L., Abreu, A. A. A. M., Robaina, D. T., e Kischnevsky, M. (2017). An evaluation of four reordering algorithms to reduce the computational cost of the Jacobi-preconditioned conjugate gradient method using high-precision arithmetic. *International Journal of Business Intelligence and Data Mining*, 12(2):190–209.
- Gonzaga de Oliveira, S. L., Bernardes, J. A. B., e Chagas, G. O. (2016a). An evaluation of low-cost heuristics for matrix bandwidth and profile reductions. *Computational & Applied Mathematics*. URL <http://dx.doi.org/10.1007/s40314-016-0394-9>.



- Gonzaga de Oliveira, S. L. e Chagas, G. O. (2015). A systematic review of heuristics for symmetric-matrix bandwidth reduction: methods not based on metaheuristics. In *Proceedings of the Brazilian Symposium on Operations Research (SBPO 2015)*, Pernambuco, Brazil. Sobrapo.
- Gonzaga de Oliveira, S. L., de Abreu, A. A. A. M., Robaina, D., e Kischinhevsky, M. (2016b). A new heuristic for bandwidth and profile reductions of matrices using a self-organizing map. In *Proceedings of the 16th ICCSA 2016: Beijing, China, LCNS, Part I*, volume 9786, p. 54–70. Springer International Publishing.
- Gonzaga de Oliveira, S. L. e Chagas, G. O. (2014). *Introdução a heurísticas para redução de largura de banda de matrizes*. SBMAC, São Carlos, SP, Brasil.
- Hestenes, M. R. e Stiefel, E. (1952). Methods of conjugate gradients for solving linear systems. *Journal of Research of the National Bureau of Standards*, 49(36):409–436.
- HSL (2017). The HSL mathematical software library. <http://www.hsl.rl.ac.uk>. Acessado: 2017-06-16.
- Iwashita, T., Nakashima, H., e Takahashi, Y. (2012). Algebraic block multi-color ordering method for parallel multi-threaded sparse triangular solver in ICCG method. In *2012 IEEE 26th International Parallel and Distributed Processing Symposium*, p. 474–483.
- Iwashita, T. e Shimasaki, M. (2003). Algebraic block red-black ordering method for parallelized ICCG solver with fast convergence and low communication costs. *IEEE Transactions on Magnetics*, 39(3):1713–1716.
- Iwashita, T., Nakanishi, Y., e Shimasaki, M. (2005). Comparison criteria for parallel orderings in ILU preconditioning. *SIAM Journal on Scientific Computing*, 26(4):1234–1260.
- Karantasis, K. I., Lenharth, A., Nguyen, D., Garzarán, M. J., e Pingali, K. (2014). Parallelization of reordering algorithms for bandwidth and wavefront reduction. In *SCI4: International Conference for High Performance Computing, Networking, Storage and Analysis*, p. 921–932, New Orleans, USA. IEEE.
- Koohestani, B. e Poli, R. (2011). A hyper-heuristic approach to evolving algorithms for bandwidth reduction based on genetic programming. In *Proceedings of AI-2011, the Thirty-first SGAI International Conference on Innovative Techniques and Applications of Artificial Intelligence*, p. 93–106. Springer London, London.
- Kumfert, G. e Pothen, A. (1997). Two improved algorithms for envelope and wavefront reduction. *BIT Numerical Mathematics*, 37(3):559–590.
- Kumfert, G. K. (2000). *An Object-Oriented Algorithmic Laboratory For Ordering Sparse Matrices*. PhD thesis, Old Dominion University.
- Lanczos, C. (1952). Solution of systems of linear equations by minimized iterations. *Journal Of Research Of The National Bureau Of Standards Section B Mathematics And Mathematical*, 49 (1):33–53.
- Lin, Y. e Yuan, J. (1994). Profile minimization problem for matrices and graphs. *Acta Mathematicae Applicatae Sinica*, 10(1):107–112.
- Maftiu-Scai, L. O. e Cornigeanu, C. A. (2016a). A parallel heuristic for bandwidth reduction based on matrix geometry. In *2016 18th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC)*, p. 424–427.



- Maftiu-Scai, L. O. e Cornigeanu, C. A. (2016b). Parallel heuristics for systems of equations preconditioning. In *2016 18th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC)*, p. 319–322, Timisoara, Romania. IEEE.
- Manguoglu, M., Koyutürk, M., Sameh, A. H., e Grama, A. (2010). Weighted matrix ordering and parallel banded preconditioners for iterative linear system solvers. *SIAM Journal on Scientific Computing*, 32(3):1201–1216.
- Medeiros, S., Pimenta, P., e Goldenberg, P. (1993). An algorithm for profile and wavefront reduction of sparse matrices with a symmetric structure. *Engineering Computations*, 10(3):257–266.
- Papadimitriou, C. H. (1976). The NP-Completeness of the bandwidth minimization problem. *Computing*, 16(3):263–270.
- Prasad, I. D., Patnaik, L., e Murthy, I. (1992). An optimal parallel algorithm for sparse matrix bandwidth reduction on a linear array. *International Journal of Computer Mathematics*, 42(1-2): 7–20.
- Puttonen, J. (1983). Simple and effective bandwidth reduction algorithm. *International Journal for Numerical Methods in Engineering*, 19(8):1139–1152.
- Reeves, C. R., editor (1993). *Modern Heuristic Techniques for Combinatorial Problems*. John Wiley & Sons, Inc., New York, NY, USA.
- Rodrigues, T. N., Boeres, M. C. S., e Catabriga, L. (2016a). An implementation of the unordered parallel RCM for bandwidth reduction of large sparse matrices. In *Proceeding Series of the Brazilian Society of Computational and Applied Mathematics*, volume 5, p. 010097–1–7, Gramado, RS. SBMAC.
- Rodrigues, T. N., Boeres, M. C. S., e Catabriga, L. (2016b). An optimized leveled parallel RCM for bandwidth reduction of sparse symmetric matrices. In *Proceedings of the Brazilian Symposium on Operations Research (SBPO 2016)*. SOBRAPO.
- Scrucca, L. (2013). GA: a package for genetic algorithms in R. *Journal of Statistical Software*, 53 (4):1–37.
- Semba, K., Tani, K., Yamada, T., Iwashita, T., Takahashi, Y., e Nakashima, H. (2013). Parallel performance of multithreaded ICCG solver based on algebraic block multicolor ordering in finite element electromagnetic field analyses. *IEEE Transactions on Magnetics*, 49(5):1581–1584.
- Sloan, S. W. (1989). A FORTRAN program for profile and wavefront reduction. *International Journal for Numerical Methods in Engineering*, 28(11):2651–2679.
- Tarricone, L. (2000). A genetic approach for the efficient numerical analysis of microwave circuits. *Applied Computational Electromagnetics Society Journal*, 15(2):87 – 93.
- Tarricone, L., Dionigi, M., e Sorrentino, R. (1995). A strategy for the efficient mode matching analysis of complex waveguide networks. In *1995 25th European Microwave Conference*, volume 1, p. 425–429, Perugia, Italy. IEEE.
- Tsuburaya, T., Okamoto, Y., e Sato, S. (2015). Parallelized ICCG method using block-multicolor orderings in real symmetric linear system derived from voltage-driven FEM in time domain. *COMPEL - The international journal for computation and mathematics in electrical and electronic engineering*, 34(5):1433–1446.