Proceeding Series of the Brazilian Society of Computational and Applied Mathematics, v. 10, n. 1, 2023.

Trabalho apresentado no XLII CNMAC, Universidade Federal de Mato Grosso do Sul - Bonito - MS, 2023

Proceeding Series of the Brazilian Society of Computational and Applied Mathematics

Base de ciclos mínima no fluxo de potência ótimo

Fernanda L. Teixeira[1]
IMECC/Unicamp, Campinas, SP
Aurelio R. L. Oliveira[2]
IMECC/Unicamp, Campinas, SP

Resumo.
Uma base de ciclos mínima fornece uma estrutura esparsa para a matriz ciclo de um grafo, que é utilizada como uma etapa de pré-processamento em várias aplicações. Neste trabalho, analisamos métodos para encontrar uma base de ciclos mínima, implementamos e realizamos testes computacionais utilizando os sistemas de teste do Institute of Electrical and Electronics Engineers (IEEE) e sistemas brasileiros de grande porte. Os resultados apontam que a matriz ciclo gerada por uma base aproximada de ciclos mínima é uma alternativa interessante para aplicações em redes elétricas, pois apresenta um equilíbrio viável entre o tempo de processamento e a esparsidade da matriz.

Palavras-chave. Base de Ciclos Mínima, Matriz Ciclo, Esparsidade, Fluxo de Potência Ótimo.

1 Introdução

Ciclos em grafos desempenham um papel importante em muitas aplicações, como por exemplo em análise de redes elétricas [9], análises de vias químicas e biológicas [3], entre outros.

Este trabalho tem como motivação a aplicação em redes elétricas descrita no trabalho de Oliveira, Soares e Nepomuceno [9]. Os autores apresentaram um modelo de fluxo de potência ótimo DC, onde as leis de Kirchhoff são representadas por um problema de fluxo em redes com restrições adicionais. A estrutura matricial resultante é explorada de forma a reduzir o sistema linear a ser resolvido para um sistema com dimensão igual ao número de barras ou, opcionalmente, ao número de laços independentes (ou ciclos). Essa matriz é invariante ao longo das iterações, o que permite que o método seja muito eficiente, tornando as iterações mais rápidas.

Essa matriz de ciclos, denotada como matriz de reatância, é formada pelos ciclos que formam uma base para o espaço ciclo da rede. A forma mais esparsa da matriz de reatância é aquela em que cada laço independente (ciclo) é o menor possível. Quanto mais esparsa for essa matriz, mais rápidas serão as iterações do método para resolver o problema de fluxo de potência ótima. Assim, encontrar uma matriz de reatância esparsa é um pré-processamento importante para o problema em questão.

Para entender a estrutura da matriz de reatância consideramos a rede elétrica um grafo não orientado conexo $G(N, E)$, onde as n barras formam o conjunto de nós N e as m ligações entre barras formam o conjunto arestas E. O conjunto de todos os ciclos do grafo G é um subespaço vetorial do espaço vetorial formado por todos subconjuntos de arestas de G. Em grafos não orientados, a cada ciclo C em um grafo G é associado um vetor de incidência x, onde $x_e = 1$ se a aresta e pertence a C e $x_e = 0$ caso contrário. O espaço ciclo gerado por esses vetores de incidência formam um espaço vetorial sobre o corpo $\mathbf{Z}_2 = \{0, 1\}$, no qual a dimensão é dada pelo número ciclomático $\nu = m - n + 1$.

[1]fernanda.teixeira@ifsp.edu.br
[2]aurelio@ime.unicamp.br

Uma coleção de ciclos cujos vetores de incidência formam uma base para o espaço de ciclos de um grafo é chamada de base de ciclos. A matriz formada pelos vetores de uma base de ciclos é chamada de matriz ciclo. No caso do problema de redes elétricas, a matriz ciclo é matriz de reatância.

Neste trabalho, sempre que mencionado um grafo G, estaremos tratando de um grafo não orientado, conexo, com n vértices e m arestas, cujas arestas possuem peso unitário. Uma maneira simples de determinar uma base de ciclos é utilizando uma árvore geradora. Dada qualquer árvore geradora T de um grafo $G = (N, E)$, podemos construir um conjunto de ciclos da seguinte forma: para cada aresta $e = (u; v) \in E \backslash T$, um ciclo é formado por e e o caminho único de u até v em T. Note que o número de arestas de uma árvore geradora é $n-1$. Logo, o número de arestas de $E \backslash T$ é $m-(n-1) = m-n+1$. Além disso, cada arestas $e = (u; v) \in E \backslash T$ pertence a um único ciclo no conjunto formado. Dessa forma, o conjunto de ciclos formado é linearmente independente e forma uma base para o espaço de ciclos, denominada de base de ciclos fundamental. Assim, construir uma base cíclica é um processo de complexidade linear em relação ao tempo.

Em Oliveira, Soares e Nepomuceno [9], a matriz de reatância é determinada por uma base de ciclos fundamental. No entanto, para aumentar a esparsidade, os autores utilizam o método de busca em largura para determinar a árvore geradora que forma os ciclos da base.

Bases de ciclos consistindo em ciclos de comprimento mínimo, ou em grafos de custo, ciclos de custo mínimo, são interessantes, pois geram uma matriz mais esparsa possível. Por esta razão, o foco desse trabalho é compreender os métodos para determinar uma base de ciclos mínima. Vale destacar que um grafo pode ter mais de uma base de ciclos mínima, mas todas têm o mesmo comprimento (ou custo).

Nas últimas décadas, muitos resultados novos em bases de ciclos foram publicados, incluindo classificações de diferentes tipos de bases de ciclos, resultados estruturais, limitantes de comprimento e peso mínimo de bases de ciclo, algoritmos de tempo polinomial para construção exata ou aproximada de bases de ciclos mínimas para alguns tipos e resultados fortes para outros tipos de base de ciclos, como apresentado em Kavitha et al. [8].

Uma opção de base esparsa consiste em encontrar uma base de ciclos formada por ciclos canônicos, onde um ciclo canônico é um ciclo que não possui ciclos internos, ou seja, ciclos canônicos não possuem cordas ou caminhos internos. O interesse principal nessa base está na sua estrutura, que só permite que uma aresta pertença a dois ciclos da base. No entanto, podemos ter mais de uma representação gráfica para um mesmo grafo, no caso de representações isomorfas, o que torna a tarefa de determinar se um caminho é (ou não) interno ao ciclo um problema topológico, inviável para uma etapa de pré-processamento de outras aplicações.

Além disso, o teste $IEEE30$ ilustrado na Figura 1 mostrou que a base de ciclos canônicos não é necessariamente uma base de ciclos mínima, logo não é a mais esparsa do grafo. Para ilustrar, considere as Figuras 2 e 3 que representam o mesmo subgrafo de $IEEE30$. Note que na Figura 2, temos o ciclo canônico $\{15, 18, 19, 20, 10, 21, 22, 24, 23, 15\}$ em destaque com comprimento igual a 9 e o ciclo $\{10, 21, 22, 10\}$, ambos contidos na base de ciclos canônica do grafo $IEEE30$. Já na Figura 3, temos o ciclo mínimo $\{15, 18, 19, 20, 10, 22, 24, 23, 15\}$ em destaque com comprimento igual a 8 e o ciclo $\{10, 21, 22, 10\}$, ambos contidos na base de ciclos mínima do grafo $IEEE30$. Dessa forma, é possível observar que a base de ciclos canônica possui uma entrada (aresta) a mais que a base de ciclos mínima.

Dessa forma, a base de ciclos mínima gera a melhor estrutura para a matriz ciclo do ponto de vista matemático e tem sido amplamente estudada. Sua importância está em entender a estrutura cíclica dos grafos e seu uso como uma etapa de pré-processamento em vários algoritmos. Isso se deve ao fato de que, em algumas aplicações, uma base de ciclos é usada como entrada para os algoritmos, que, em geral, resolvem algum sistema linear baseado nessa base. O uso de uma base de ciclos mínima não é obrigatório, mas fornece uma estrutura esparsa ao sistema linear e, portanto, resulta em menor tempo de processamento.

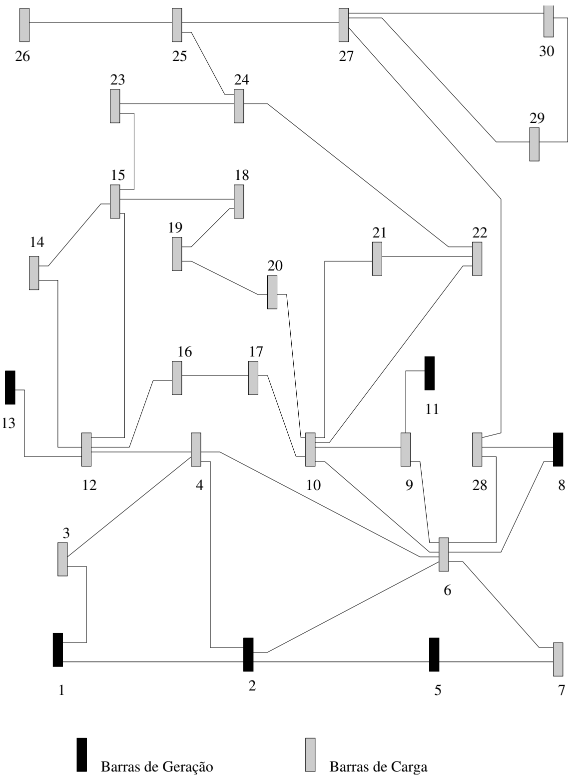

Figura 1: Grafo representando o teste IEEE30 do sistema IEEE. Fonte: Elaborado por Leonardo Nepomuceno.

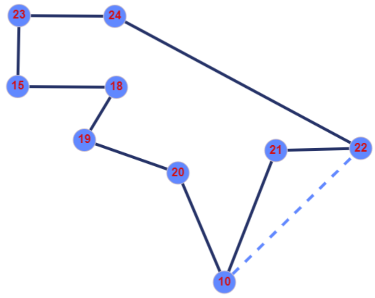

Figura 2: Ciclo canônico

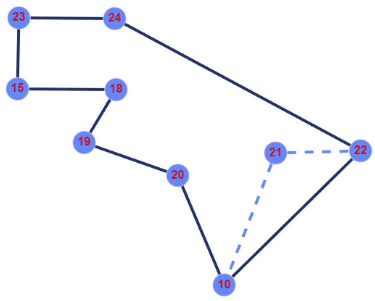

Figura 3: Ciclo mínimo

O primeiro algoritmo de tempo polinomial para o problema de encontrar uma base de ciclos mínima (MCB, do inglês *Minimum Cycle Basis*) foi proposto por Horton [6] e tem complexidade computacional $O(m^3n)$. A abordagem de Horton consiste em criar um conjunto M de ciclos, com mn ciclos, que ele provou ser um superconjunto de uma MCB. Em seguida, a MCB é extraída como os ν menores ciclos linearmente independentes de M usando eliminação de Gauss.

Pina [10] apresentou um algoritmo com ordem de complexidade $O(m^3 + mn^2log(n))$ para determinar uma MCB, utilizando uma abordagem diferente da de Horton, baseado em um algoritmo combinatorial.

Em Golynski e Horton [4], os autores resolvem o problema MCB para matróides regulares em tempo polinomial $O(m^\omega n)$, onde $\omega < 2,376$ é o expoente para matriz multiplicação.

Hartvigsen e Mardon [5] mostraram que, em grafos planares, o problema de corte mínimo de todos os pares é equivalente ao problema de base de ciclos mínima no grafo dual. Eles também apresentaram um novo algoritmo para resolver esses dois problemas em grafos planares, com uma complexidade de ordem $O(n^2log(n))$.

Em Kavitha et al. [7], os autores mostram que relaxar a manutenção do conjunto de testemunhas de Pina mantém a precisão do algoritmo, resultando em um tempo de complexidade de ordem $O(m^2n + mn^2log(n))$. Isso ocorre porque apenas um vetor é necessário nesse subespaço. Essa é a menor complexidade para o problema MCB.

No entanto, mesmo com a evolução dos métodos exatos para calcular uma base de ciclos mínima, o tempo de processamento necessário para um método exato ainda não é realmente prático. Por essa razão, surgiram vários trabalhos que calculam bases aproximadas de MCB. Tais bases são ainda consideradas esparsas e, além disso, podem ser calculadas muito mais rapidamente.

2 Algoritmos para Base de Ciclo Mínimo e outras bases

Nesta seção, apresentaremos uma breve descrição dos métodos clássicos para encontrar uma base de ciclos mínima, que são o algoritmo de Horton [6] e o algoritmo de Pina [10]. Além disso, apresentaremos uma abordagem heurística para uma base aproximada de ciclos mínima e a abordagem de Oliveira, Soares e Nepomuceno [9] para uma base de ciclos fundamental. Os resultados obtidos por esses métodos serão apresentados na seção seguinte.

2.1 Algoritmo de Horton

Horton [6] mostrou que uma base de ciclos mínima (orientada ou não orientada) pode ser construída por um algoritmo guloso simples. Para isso mostrou os seguintes resultados:

Lema 2.1. *Se B é uma base de ciclos para um grafo, $C \in B$, e $C = C_1 + C_2$, então $B \backslash C \cup C_i$ é uma base de ciclos para um dos $i = 1$ ou $i = 2$.*

Teorema 2.1. *Sejam x e y dois vértices em um ciclo C de uma base de ciclos mínima B. Então $P(x, y)$ está contido em C, onde $P(x, y)$ é o menor caminho de x a y em G.*

Teorema 2.2. *Seja x um vértice qualquer de um ciclo C em uma base de ciclos mínimo B. Então existe uma aresta (y, z) em C tal que $C = P(x, y) + P(x, z) + (y, z)$.*

Com isso, Horton propôs o algoritmo que é iniciado com uma base vazia e, a cada iteração, um ciclo é adicionado a base parcial se for linearmente independente dos ciclos anteriormente adicionados. O algoritmo processa os ciclos de G em ordem não decrescente de peso. Esse processo é repetido até que se obtenha ν ciclos linearmente independentes. A verificação da independência linear pode ser facilmente realizada por meio da eliminação gaussiana.

Os ciclos são selecionados do conjunto de Horton e são formados da seguinte maneira: para cada vértice v e aresta (x, y) no grafo, é criado um ciclo $C(v, x, y) = P(v, x) + P(v, y) + (x, y)$. Dessa forma, o conjunto de Horton possui mn ciclos. Após serem determinados, os ciclos são ordenados em ordem não decrescente.

A determinação dos caminhos mínimos pode ser feita pelo algoritmo Dijkstra [1] ou Floyd [2].

A estrutura do algoritmo é dada por:

1. Encontrar um caminho mínimo entre todos os pares de vértices;
2. Determinar o conjunto de Horton;
3. Ordenar os ciclos por peso;
4. Usar um algoritmo guloso para encontrar a base de ciclos mínima do conjunto de Horton.

2.2 Algoritmo de Pina

Na abordagem proposta por Pina [10], os ciclos da base de ciclos mínima são determinados sequencialmente. A independência linear é garantida mantendo, a cada passo, uma base do subespaço ortogonal ao subespaço formado pelos ciclos selecionados até o momento. Os vetores desta base são referidos como testemunhas.

Seja T qualquer árvore geradora em $G(V, E)$ e sejam $E'_T = \{e_1, \cdots, e_N\}$ as arestas de $E \backslash T$ em alguma ordem arbitrária, mas fixa. Uma testemunha S de um ciclo C é um subconjunto de E'_T que vai provar que C pertence à base de ciclos mínima. A base inicial de testemunhas $\{S_1, \cdots, S_N\}$ é a base canônica de E'_T sobre $\mathbf{Z}_2$, onde cada S_j , para $1 \leq j \leq N$, tem apenas a n-ésima componente igual a 1. Note que tanto os ciclos quanto suas testemunhas são vetores no espaço $\{0; 1\}^N$.

Denotando $\langle C; S\rangle$ o produto interno usual dos vetores C e S, dizemos que um vetor S é ortogonal a C se $\langle C; S\rangle = 0$. Como estamos no corpo $\mathbf{Z}_2$, observe que $\langle C; S\rangle = 1$ se, e somente se, C contém um número ímpar de arestas de S.

Em cada etapa $i, com \quad 1 \leq i \leq N$, um novo ciclo C_i é calculado. Para que esse novo ciclo C_i seja linearmente independente do conjunto de ciclos $\{C_1, \cdots, C_{i-1}\}$ calculado anteriormente, o algoritmo primeiro determina um vetor não nulo S_i que seja ortogonal aos ciclos. Dado tal S_i, o algoritmo então calcula o ciclo C_i mais curto em G tal que $\langle C_i, S_i\rangle = 1$.

Dado S_i, podemos calcular um ciclo mínimo C_i tal que $\langle C_i, S_i\rangle = 1$, reduzindo-o a k cálculos de caminhos mínimos em um grafo apropriado G_i. O grafo G_i pode ser visualizado como dois níveis de G, o nível $+$ e o nível $-$, dentro de cada nível temos as arestas de $E \backslash S_i$, e entre os níveis temos as arestas de S_i. Assim, G_i é definido a partir de $G = (V, E)$ e $S_i \subset E$ da seguinte maneira: G_i possui duas cópias de cada vértice v de G, chamadas de v^+ e v^-, e suas respectivas arestas. Para toda aresta $e = (v, u) \in E$, se $e \in S_i$, exclui as arestas (v^+, u^+) e (v^-, u^-) e adiciona as arestas (v^+, u^-) e (v^-, u^+) em G_i.

Dado qualquer caminho de v^+ para v^- em G_i, podemos corresponder a ele um ciclo em G identificando os vértices e arestas em G_i com seus vértices e arestas correspondentes em G. Assim, encontrando um caminho mínimo de v^+ para v^- em G_i, determinamos um ciclo mínimo em G.

A estrutura do algoritmo de Pina é dada por:

1. Inicializar as testemunhas $S_{i,1} = \{e_i\}$ para $i = \{1, \cdots, N\}$

2. Em cada iteração:

    (a) Encontrar um ciclo C_k, tal que $\langle C_k, S_{k,k}\rangle = 1$;

    (b) Atualizar as testemunhas restantes $S_{k+1,i}$ de forma a manter a ortogonalidade.

2.3 Heurística inspirada em Pina

Na abordagem de Pina [10], em cada iteração para encontrar o ciclo mínimo C_i, precisamos calcular k caminhos mínimos em G_i e tomar o menor deles, onde k é o números de vértices distintos das arestas em S_i.

Nossa abordagem heurística consiste em tomar o primeiro caminho mínimo encontrado pelo algoritmo, independente de ser o ciclo mínimo procurado. Dessa forma, mantemos os ciclos da base linearmente independentes, e a base resultante ainda é considerada esparsa. Essa abordagem é uma pequena variação do método exato e, até o momento, não foi encontrada na literatura.

2.4 Método para Base de Ciclos Fundamental

Na abordagem de Oliveira, Soares e Nepomuceno [9], os autores utilizam o método de busca em largura para determinar a árvore geradora que forma os ciclos da base fundamental.

Dada a árvore geradora de busca em largura T do grafo $G = (N, E)$, construímos um conjunto de ciclos da seguinte forma: para cada aresta $e = (u; v) \in E \backslash T$, um ciclo é formado por e e o caminho único de u até v em T. Todos esses ciclos formam uma base de ciclos fundamental.

DOI: 10.5540/03.2023.010.01.0027

3 Testes Computacionais

Os métodos foram implementados na linguagem Octave 5.2.0 em um processador AMD Ryzen 7 com 12GB de memória, executando no sistema operacional Windows 7. Para os testes preliminares, foram utilizados sistemas de testes do IEEE e sistemas brasileiros de grande porte, com o número de barras descrito na coluna "Barras"da Tabela 1.

O Octave não foi capaz de encontrar uma base de ciclos mínima pelo método de Horton nas instâncias $SSECO1654$, $SSECO1732$ e $BRASIL$ por extrapolar o tempo limite. Por outro lado, o algoritmo Pina encontrou uma base em todas as instâncias, sendo o pior tempo de execução de $76842,9$ segundos, que ocorreu na instância $SSECO1732$.

Para análise de esparsidade, foram implementados três métodos: o método proposto por Pina [10] para base de ciclos mínima (MCB), o método utilizado por Oliveira, Soares e Nepomuceno [9] para uma base de ciclos fundamental (FCB, do inglês *Fundamental Cycle Basis*) utilizando uma árvore de busca em largura, e o método heurístico inspirado no algoritmo de Pina (HCB, do inglês *Heuristc Cycle Basis*) para uma base aproximada de ciclos. A Tabela 1 apresenta o número de entradas não nulas da base gerada por cada método nas colunas "nnz"(do inglês *number of nonzero*) e o tempo de processamento em segundos de cada método nas colunas "tempo".

Tabela 1: Número de entradas não nulas na base e tempo de processamento.

Sistema	Barras	MCB		HCB		FCB	
		nnz	tempo	nnz	tempo	nnz	tempo
IEEE30	30	55	0,34	55	0,23	56	0,10
IEEE118	118	270	25,04	287	7,13	442	2,69
BRASIL810	810	1416	7515,00	1459	546,43	3258	147,74
SSECO1654	1654	2303	53103,60	2446	3029,45	5707	456,70
SSECO1732	1732	2410	76842,90	2547	3228,39	5893	492,80
BRASIL	1993	2631	14014,90	2710	3349,71	5822	581,31

Podemos observar nos resultados apresentados na Tabela 1 que a base de ciclos mínima (MCB) possui um número significativamente menor de entradas não nulas em comparação com a base de ciclos fundamental (FCB), o que valida a ideia inicial de que a matriz ciclo de uma base de ciclos mínima é muito mais esparsa do que a gerada por uma base de ciclos fundamental. No entanto, o tempo de processamento necessário para gerar uma base de ciclos mínima pelo método de Pina (MCB) é consideravelmente maior do que o tempo gasto para encontrar uma base de ciclos fundamental (FCB), o que torna o método exato de Pina inviável para uma etapa de pré-processamento. Essa limitação motivou a realização de testes com o método heurístico inspirado no método de Pina (HCB). Podemos notar que o número de entradas não nulas da base heurística (HCB) é maior do que o número de entradas não nulas da base de ciclos mínima (MCB), mas é significativamente menor do que o número de entradas não nulas da base de ciclos fundamental (FCB). Além disso, o tempo de processamento da heurística (HCB) é consideravelmente menor que o tempo de processamento do método de Pina (MCB), o que torna a matriz ciclo gerada pela heurística uma opção interessante para aplicações em redes elétricas.

4 Considerações Finais

Este trabalho apresentou métodos para encontrar uma base de ciclos para a matriz de reatância utilizada como entrada no problema de Fluxo de Potência Ótimo de redes elétricas, tanto nos sistemas IEEE quanto nos sistemas brasileiros de grande porte. Além disso, foi realizado uma

DOI: 10.5540/03.2023.010.01.0027

análise da esparsidade das matrizes obtidas e viabilidade de aplicação considerando o tempo de processamento. Até o momento, não encontramos na literatura a aplicação desses métodos em sistemas de redes elétricas.

Como continuidade deste trabalho, planejamos implementar esses métodos e outros métodos heurísticos, utilizando a linguagem Python. O objetivo é encontrar uma alternativa viável na prática para o Problema de Fluxo de Potência Ótimo, considerando a eficiência computacional e a esparsidade da matriz ciclo.

Agradecimentos

Agradecemos o apoio do Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq), processo $140783/2018-0$, e do Instituto Federal de São Paulo (IFSP).

Referências

[1] E. W. Dijkstra. "A note on two problems in connexion with graphs". Em: **Edsger Wybe Dijkstra: His Life, Work, and Legacy**. Ed. por Hoare T. Krzysztof R. A. Vol. 45. Association for Computing Machinery, 2022. Cap. 14, pp. 287–290.

[2] R. W. Floyd. "Algorithm 97: Shortest Path". Em: **Commun. ACM** 6 (1962), pp. 345–350. DOI: 10.1145/367766.368168.

[3] P. M. Gleiss. "Short Cycles Minimum Cycle Bases of Graphs from Chemistry and Biochemistry". Tese de doutorado. Faculty of Science e Mathematics from the University of Wien, 2001.

[4] A. Golynski e J. D. Horton. "A polynomial time algorithm to find the minimum cycle basis of a regular matroid". Em: **Scandinavian Workshop on Algorithm Theory**. 2002, pp. 200–209. DOI: 10.1007/3-540-45471-3_21.

[5] D. Hartvigsen e R. Mardon. "The all-pairs min cut problem and the minimum cycle basis problem on planar graphs". Em: **SIAM Journal on Discrete Mathematics** 3 (1994), pp. 403–418. DOI: 10.1137/S0895480190177042.

[6] J. D. Horton. "A polynomial-time algorithm to find the shortest cycle basis of a graph". Em: **SIAM Journal on Computing** 2 (1987), pp. 358–366. DOI: 10.1137/0216026.

[7] T. Kavitha et al. "A faster algorithm for minimum cycle basis of graphs". Em: **Automata, Languages and Programming: 31st International Colloquium, ICALP 2004, Turku, Finland, July 12-16, 2004. Proceedings 31**. 2004, pp. 846–857. DOI: 10.1007/978-3-540-27836-8_71.

[8] T. Kavitha et al. "Cycle bases in graphs characterization, algorithms, complexity, and applications". Em: **Computer Science Review** 4 (2009), pp. 199–243. DOI: 10.1016/j.cosrev.2009.08.001.

[9] A. R. L. Oliveira, S. Soares e L. Nepomuceno. "Optimal active power dispatch combining network flow and interior point approaches". Em: **IEEE Transactions on Power Systems** 4 (2003), pp. 1235–1240. DOI: 10.1109/TPWRS.2003.814851.

[10] J. C. Pina. "Applications of shortest path methods". Tese de doutorado. University of Amsterdam, 1995.

