Proceeding Series of the Brazilian Society of Computational and Applied Mathematics, v. 12, n. 1, 2026.

Trabalho apresentado no XLIV CNMAC, Fundação Getulio Vargas - Rio de Janeiro - RJ, 2025.

Determinação dos k Caminhos Mínimos Disjuntos Mais Confiáveis em Uma Rede de Fluxo Multiestado

Majid Forghani-elahabad[1], Nicolas T. Matos[2]
CMCC/UFABC, Santo André, SP

Resumo. A determinação de k caminhos mínimos disjuntos (CMDs) em uma rede de fluxo multiestado (RFM) desempenha um papel fundamental na garantia da confiabilidade e disponibilidade de sistemas reais, como telecomunicações, transporte e redes de distribuição. Embora estudos anteriores tenham se concentrado na identificação do número máximo de CMDs entre dois nós em um grafo, a confiabilidade dos caminhos selecionados em uma RFM tem sido pouco explorada. Neste trabalho, introduzimos o problema de determinar os k CMDs mais confiáveis capazes de transmitir d unidades de fluxo de uma fonte a um destino dentro de T unidades de tempo. Além disso, propomos um algoritmo para sua resolução e demonstramos sua correção.

Palavras-chave. Caminhos Mínimos Disjuntos, Redes de Fluxo Multiestado, Confiabilidade, Algoritmos.

1 Introdução

Uma rede de fluxo multiestado (RFM) é um grafo direcionado ou não direcionado no qual as capacidades dos arcos assumem valores aleatórios discretos de um conjunto predefinido, seguindo uma distribuição de probabilidade. Os nós podem ser determinísticos (totalmente confiáveis) ou exibir estados variáveis como os arcos, levando a sistemas multiestado [2, 3]. No entanto, um nó não confiável pode ser substituído por dois nós totalmente confiáveis (determinísticos) conectados por um único arco não confiável [7]. Consequentemente, a maioria dos estudos se concentra em RFMs com nós totalmente confiáveis. Muitos sistemas do mundo real, incluindo redes de telecomunicações, transporte e logística, podem ser modelados como RFMs [4, 10, 14, 15, 20, 21].

Estendendo o problema do caminho mais rápido para a confiabilidade do sistema, o problema da confiabilidade do caminho mais rápido foi introduzido pela primeira vez em [13] e tem sido amplamente estudado desde então [3]. Este problema foi posteriormente estendido para considerar dois ou mais caminhos mínimos disjuntos (CMDs) [5, 6]. Neste contexto, um caminho é uma sequência de arcos consecutivos conectando um nó de origem a um nó de destino, enquanto um caminho mínimo (CM) é um caminho livre de ciclos. Dois caminhos são disjuntos (arcos-disjuntos) se não compartilharem arcos comuns, embora possam ter nós comuns.

Apesar da extensa pesquisa sobre a confiabilidade do caminho mais rápido, pouca atenção tem sido dada à identificação dos CMs mais confiáveis [3]. Enquanto isso, os problemas de determinar o número máximo de CMDs ou identificar k CMDs em um grafo dado têm sido amplamente estudados por décadas devido à sua importância na robustez e eficiência da rede [12, 16, 18, 19]. No entanto, poucos estudos abordaram a confiabilidade de k CMDs [5], e, até onde sabemos, nenhum trabalho anterior se concentrou em determinar os k CMDs mais confiáveis em uma RFM. Para preencher

[1] m.forghani@ufabc.edu.br
[2] m.nicolas@ufabc.edu.br

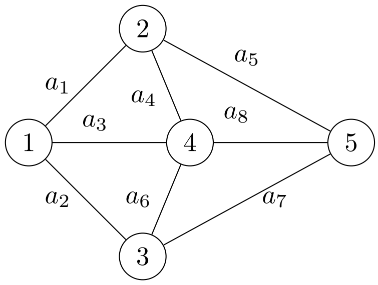

Figura 1: Uma rede de referência com nove arestas e seis nós. Fonte: Forghani-elahabad et al.[4].

essa lacuna, iniciamos o estudo deste problema para dois CMDs em [8] e o estendemos neste trabalho para o caso geral de k CMDs.

O restante deste artigo está estruturado da seguinte forma. A Seção 2 introduz a notação, as suposições e os preliminares necessários. A Seção 3 apresenta uma investigação detalhada do problema juntamente com o algoritmo proposto. Finalmente, a Seção 4 conclui o estudo.

2 Preliminares

Denotamos uma rede de fluxo multiestado (RFM) como $G(N, A, M, L)$. Nessa rede, $N = \{1, 2, \ldots, n\}$ representa o conjunto de nós, e $A = \{a_1, a_2, \ldots, a_m\}$ representa o conjunto de arcos (arestas). Assim, n é o número total de nós e m o número total de arcos. O vetor $M = (M_1, \ldots, M_m)$ especifica as capacidades máximas dos arcos, onde M_i é a capacidade máxima do arco a_i, para $i = 1, \ldots, m$. Similarmente, o vetor $L = (l_1, \ldots, l_m)$ indica os tempos de espera nos arcos. Além disso, os nós 1 e n representam, respectivamente, o nó de origem e o nó de destino.

Definimos o estado operacional da rede através de um vetor $X = (x_1, x_2, \cdots, x_m)$, onde cada componente x_i representa a capacidade atual do arco a_i. Essa capacidade varia aleatoriamente entre zero e M_i, assumindo valores inteiros, para cada i de 1 a m. É importante notar que o vetor M em si representa um possível estado do sistema, especificamente o estado de capacidade máxima. O tempo máximo permitido para a transmissão de dados é denotado por T. A demanda, representada por d, é um valor inteiro não negativo que especifica a quantidade mínima de fluxo a ser transferida do 1 para o nó n.

Seja h o número total de caminhos mínimos (CMs) existentes na rede, representados por $P_1, P_2, \ldots, P_h$, e f o número máximo de caminhos mínimos disjuntos (CMDs) entre os nós 1 e n na rede. Observe que o número f pode ser calculado de forma eficiente [11], e apresentamos uma abordagem para isso na Seção 3.4. Considerando a RFM da Figura 1, observe que existem $h = 9$ CMs e, no máximo, $f = 3$ CMDs entre os nós 1 e 5. Um conjunto possível de três CMDs é formado pelos caminhos $P_1 = \{a_1, a_5\}, \quad P_4 = \{a_3, a_8\}, \quad P_7 = \{a_2, a_7\}$. Observe, também, que, se considerarmos $P_3 = \{a_1, a_4, a_6, a_7\}$, não é possível encontrar mais dois CMs que, junto com esse caminho, formem um conjunto de três CMDs.

Seja $k \le f$ o número de CMDs requeridos para transmitir d unidades de dados dentro de T unidades de tempo. Denotamos por $\Psi^k = \{\psi_1^k, \psi_2^k, \ldots, \psi_{\sigma_k}^k\}$ o conjunto de todos os possíveis subconjuntos de k CMDs, onde ψ_i^k representa um conjunto de k CMDs, para $i = 1, 2, \ldots, \sigma_k$. Por exemplo, na Figura 1, temos os seguintes CMs $P_1 = \{a_1, a_5\},\ P_2 = \{a_1, a_4, a_8\},\ P_3 = \{a_1, a_4, a_6, a_7\},\ P_4 = \{a_3, a_8\},\ P_5 = \{a_3, a_4, a_5\},\ P_6 = \{a_3, a_6, a_7\},\ P_7 = \{a_2, a_7\},\ P_8 = \{a_2, a_6, a_8\},\ P_9 = \{a_2, a_6, a_4, a_5\}$ e assim, $\psi_1^2 = \{P_1, P_4\},\ \psi_2^2 = \{P_1, P_6\},\ \psi_3^2 = \{P_1, P_7\},\ \psi_4^2 = \{P_1, P_8\},\ \psi_5^2 = \{P_2, P_6\},\ \psi_6^2 =$

$\{P_2, P_7\}$, $\psi_7^2 = \{P_3, P_4\}$, $\psi_8^2 = \{P_4, P_7\}$, $\psi_9^2 = \{P_4, P_9\}$, $\psi_{10}^2 = \{P_5, P_7\}$, $\psi_{11}^2 = \{P_5, P_8\}$, e, portanto, há $\sigma_2 = 11$ conjuntos diferentes de dois CMDs.

Definimos um vetor $D = (d_1, d_2, \ldots, d_k)$, no qual d_j representa a quantidade de fluxo destinada ao CM, P_j no conjunto ψ_i^k, para $j = 1, \ldots, k$ e $i = 1, \ldots, \sigma_k$. Este vetor D, chamado de *vetor de política*, especifica a distribuição do fluxo entre os k CMDs, com o objetivo de transferir d unidades de fluxo do nó 1 ao nó n. Assim, devemos ter $d_1 + \cdots + d_k = d$. A capacidade de um CM é definida como a quantidade máxima de fluxo que ele pode suportar simultaneamente. Matematicamente, a capacidade do CM, P_j sob o vetor de estado do sistema (VES) X é dada por $KP_j(X) = \min\{x_i | a_i \in P_j\}$, para todo $j = 1, 2, \cdots, h$.

Conforme estabelecido em trabalhos anteriores [1, 4, 9], o tempo de espera associado a um CM é calculado pela soma dos tempos de espera de seus arcos constituintes. Assim, designando LP_j como o tempo de espera de P_j, temos $LP_j = \sum_{i:\ a_i \in P_j} l_i$. Ao enviar d_j unidades de fluxo do nó 1 até o nó n por meio do caminho P_j, são necessárias inicialmente LP_j unidades de tempo para que o fluxo percorra o caminho. A partir desse momento, $KP_j(X)$ unidades de fluxo podem ser recebidas no nó de destino por unidade de tempo, onde $KP_j(X)$ representa a capacidade do caminho P_j sob X. Considerando que as capacidades e os fluxos enviados são inteiros, o tempo total necessário para transmitir d_j unidades de fluxo por meio de P_j, sob X, é dado por $\xi(d_j, X, P_j) = LP_j + \left\lceil \frac{d_j}{KP_j(X)} \right\rceil$. Seja $C_{dj}(P_j)$ a confiabilidade do caminho P_j para a transmissão simultânea de d_j unidades de fluxo do nó 1 ao nó n. A confiabilidade, em termos gerais, representa a probabilidade de sucesso. Assim, $C_{dj}(P_j)$ corresponde à probabilidade de que a capacidade do caminho P_j seja ao menos d_j. Considerando a definição acima de capacidade de um caminho, tem-se:

$$C_{dj}(P_j) = \prod_{a_i \in P_j} \Pr(x_i \geq d_j). \tag{1}$$

Neste estudo, adotamos as seguintes suposições, consistentes com a literatura [3, 6]:

1. Todos os nós são considerados perfeitamente confiáveis.
2. As capacidades dos arcos seguem distribuições de probabilidade conhecidas.
3. As capacidades dos arcos são estatisticamente independentes entre si.
4. O fluxo vai da origem ao destino por até k CMDs, respeitando a conservação do fluxo.

3 Bloco Principal

O problema principal é encontrar os k CMDs mais confiáveis para transmitir d unidade de fluxo dentro de T unidade de tempo. Então, temos os seguintes passos:

1. Encontrar todos os CM da rede dada.
2. Determinar todos os conjuntos de k CMDs, se houver algum.
3. Verificar quais conjuntos do passo 2 são soluções viáveis.
4. Determinar o conjunto mais confiável.

Existem vários algoritmos para o primeiro passo [3], porém os passos 2 e 3 são desafiadores. Vamos discutir algumas possíveis soluções nesta seção, com base nas quais propomos um algoritmo para resolver o problema principal.

DOI: 10.5540/03.2026.012.01.0275

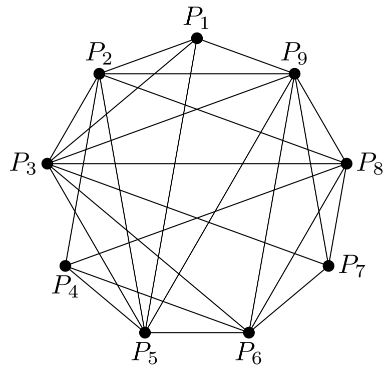

Figura 2: O grafo conflito correspondente à rede da Figura 1. Fonte: Os Autores.

3.1 Determinação de k CMDs

Lembrando que existem, no total, h CMs e, no máximo, f CMDs na rede, primeiramente verificamos se $k \leq f$. Se não, a resposta é simplesmente que não há k CMDs. Se sim, uma solução direta seria listar todas as $\binom{h}{k}$ possibilidades de escolher k CMs entre os h CMs e depois verificar quais conjuntos formam um conjunto de CMDs. Porém, mesmo utilizando técnicas combinatórias, o custo computacional dessa abordagem é muito alto, especialmente quando h é suficientemente grande. Aqui, discutimos duas soluções que são mais eficientes do que essa abordagem direta.

1. Buscar os conjuntos independentes em um grafo de conflito: Nesta abordagem, construímos uma nova rede com h nós, onde cada nó representa um CM da rede original. Adicionamos uma aresta entre dois nós nessa nova rede se a interseção dos CMs associados for não vazia. Em seguida, buscamos os conjuntos independentes de nós na nova rede, ou seja, procuramos todos os conjuntos de k nós em que não há nenhuma aresta entre os nós de cada conjunto. Esses conjuntos representam os conjuntos de CMDs na rede original. Observamos que o problema de encontrar conjuntos independentes em um grafo é bem estudado [17]. Uma abordagem direta para encontrar todos os conjuntos independentes de k nós consiste em começar com um nó qualquer, remover todos os nós conectados a ele por um arco (aresta), escolher um novo nó entre os restantes e continuar esse processo até formar um conjunto de k nós. Para ilustrar essa ideia, a Figura 2 representa o grafo de conflito correspondente aos CMs da rede dada na Figura 1, lembrando que essa rede contém nove CMs.

2. Modelagem como uma Programação Linear Inteira: Podemos definir as variáveis de decisão $z_j = 1$ se o CM P_j for escolhido e $z_j = 0$ caso contrário, para $j = 1, \ldots, h$. Além disso, definimos a matriz $B_{m \times h}$ com $b_{ij} = 1$ se o arco a_i pertence a P_j e $b_{ij} = 0$ caso contrário, para $i = 1, \ldots, m$ e $j = 1, \ldots, h$. Dessa forma, é necessário determinar todas as soluções binárias $z = (z_1, z_2, \ldots, z_h)$ do seguinte problema, para o qual diversos métodos foram propostos na literatura [11].

$$\begin{cases} (i) \ \sum_{j=1}^{h} z_j = k \\ (ii) \ \sum_{j=1}^{h} b_{ij} z_j \leq 1, \qquad \forall i = 1, \ldots, m. \end{cases} \tag{2}$$

Observamos que a Eq. (i) no sistema 2 garante que tenhamos exatamente k componentes não nulos em cada solução $z = (z_1, z_2, \ldots, z_h)$, ou seja, que exatamente k CMs sejam escolhidos. Já a Eq. (ii) garante que os CMs escolhidos sejam disjuntos, pois não compartilham nenhum arco em comum.

3.2 Verificação dos CMDs escolhidos

Seja $\Psi^k = \{\psi_1^k, \psi_2^k, \ldots, \psi_{\sigma_k}^k\}$ o conjunto de todos os conjuntos de k CMDs escolhidos. Agora, precisamos verificar quais desses conjuntos de k CMDs permitem o envio de d unidades de fluxo dentro de T unidades de tempo. O resultado a seguir fornece um limite superior para a quantidade de fluxo que pode ser transmitida através de P_j dentro de T unidades de tempo.

Lema 3.1. *[8] A quantidade máxima de dados que pode ser enviada por P_j dentro de T unidades de tempo é $\gamma_j = KP_j(M)(T - LP_j)$, onde $KP_j(M)$ é a capacidade máxima de P_j e LP_j é o tempo de espera de P_j.*

Portanto, considerando $\psi_r^k = \{P_{r1}, P_{r2}, \ldots, P_{rk}\}$ um conjunto de k CMDs escolhidos, se tivermos $\sum_{j=1}^k \gamma_{rj} < d$, então a solução através desse conjunto é inviável, e ele deve ser removido. Observamos que, se $\sum_{j=1}^k \gamma_{rj} \geq d$, então qualquer vetor $D = (d_1, \ldots, d_k)$ com $0 \leq d_j \leq \gamma_j$ e $d_1 + \cdots + d_k = d$ é um vetor de política viável. Na próxima seção, investigamos como escolher o conjunto mais confiável dentre os CMDs selecionados.

3.3 O conjunto dos CMDs mais confiáveis

Eq. (1) mostra que a confiabilidade de um CM depende do fluxo transmitido e das capacidades atuais dos arcos do caminho. Como essas capacidades são não negativas, a confiabilidade de um caminho ao transmitir $d = 0$ unidades é igual a 1. Assim, para calcular a confiabilidade de um conjunto de CMs, é necessário identificar quais caminhos serão usados. Para isso, definimos $\alpha_j = 1$ se $d_j > 0$ e $\alpha_j = 0$ caso contrário, para $j = 1, \ldots, k$.

Logo, a confiabilidade do conjunto $\psi_r^k = \{P_{r1}, P_{r2}, \ldots, P_{rk}\}$ ao transmitir d unidades de fluxo sob a política $D = (d_1, \ldots, d_k)$ é dada por

$$R_D(\psi_r^k) = \prod_{t=1}^{k} \alpha_t C_{d_t}(P_{rt}). \tag{3}$$

Portanto, considerando $\Theta_r^k(d)$ como o conjunto de todos os vetores de política viáveis correspondentes ao conjunto ψ_r^k dos CMDs escolhidos, podemos calcular a confiabilidade de ψ_r^k para todos os vetores em $\Theta_r^k(d)$ usando a Eq. (3) e considerar o maior valor obtido como a confiabilidade do conjunto ψ_r^k para transmitir d unidades de fluxo dentro de T unidades de tempo, salvando o vetor de política correspondente.

Assim, após calcular a confiabilidade de todos os conjuntos não removidos na seção anterior, escolhemos aquele com a maior confiabilidade como o conjunto de k CMDs mais confiáveis da rede dada. Agora, podemos escrever o algoritmo completo.

3.4 O algoritmo proposto

Observamos que é recomendável verificar a existência de k CMDs na rede antes de iniciar o processo de sua determinação. Para isso, é necessário calcular o número máximo de CMDs na rede, denotado por f. Diversas técnicas podem ser utilizadas para essa estimativa, dentre as quais adotamos a abordagem baseada no cálculo do fluxo máximo em uma rede onde todos os arcos ativos possuem capacidade igual a 1 [11, 16]. De fato, ao atribuir capacidade 1 a cada arco com capacidade positiva, garantimos que ele possa ser utilizado em apenas um CMD. Dessa forma, o valor do fluxo máximo obtido nesse modelo corresponde ao número máximo de CMDs na rede, ou seja, f. Assim, se $k > f$, concluímos que não há uma solução viável para o problema. Com essa verificação prévia estabelecida, podemos agora descrever o algoritmo proposto.

DOI: 10.5540/03.2026.012.01.0275

Algoritmo 1

Entrada: $G(N, A, M, L)$, d, T e o conjunto de todos os CMs $P_1, \ldots, P_h$.

Saída: Os k CMDs mais confiáveis da rede.

Passo 0. Calcule f utilizando o algoritmo de fluxo máximo na rede modificada, definindo a capacidade de cada arco ativo para um. Se $k > f$, pare e informe que não existem k CMDs na rede dada.

Passo 1. Determine $\Psi^k = \{\psi_1^k, \psi_2^k, \ldots, \psi_{\sigma_k}^k\}$, o conjunto de todos os conjuntos possíveis de k CMDs, utilizando as abordagens descritas na Seção 3.1.

Passo 2. Remova os conjuntos de k CMDs pelos quais é impossível transmitir d unidades de fluxo dentro de T unidades de tempo, utilizando a abordagem descrita na Seção 3.2. Se todos os conjuntos forem removidos, pare e retorne que não há solução viável.

Passo 3. Calcule a confiabilidade dos conjuntos não removidos conforme descrito na Seção 3.3 e escolha o conjunto mais confiável. Pare.

De acordo com todas as discussões nas seções anteriores, o seguinte resultado é imediato.

Teorema 3.1. *O algoritmo proposto determina corretamente o conjunto dos k CMDs mais confiáveis na rede dada.*

4 Considerações finais

Neste trabalho, apresentamos o problema de encontrar os k CMDs mais confiáveis em uma rede de multiestado para transmitir d unidades de fluxo do nó fonte ao nó destino dentro de T unidades de tempo. Investigamos detalhadamente cada etapa do problema e descrevemos algumas abordagens para cada passo. Com base nisso, propusemos um algoritmo para resolvê-lo.

No entanto, há espaço para melhorias no algoritmo. Por exemplo, as abordagens utilizadas nos Passos 0 e 1 podem ser substituídas por alternativas mais eficientes, caso existam. Ademais, se for possível descartar alguns CMs que não fazem parte da solução logo no início do processo, o esforço para encontrar todos os conjuntos possíveis de k CMDs será significativamente reduzido. Além disso, ao reduzir o número de vetores de políticas viáveis para cada conjunto de CMDs identificado — desde que o melhor vetor de política correspondente seja mantido — o desempenho do algoritmo pode ser aprimorado. Essas possibilidades serão exploradas em trabalhos futuros.

Agradecimentos

O primeiro autor agradece à FAPESP pelo suporte financeiro (2023/13667-5), e o segundo autor agradece à CAPES (Código de Financiamento 001) e à FAPESP (2024/06845-7) pelo apoio.

Referências

[1] Y. L. Chen e Y. H. Chin. “The quickest path problem”. Em: **Computers & Operations Research** 17.2 (1990), pp. 153–161.

[2] M. Forghani-elahabad. “1 Exact reliability evaluation of multistate flow networks”. Em: **Systems Reliability Engineering**. De Gruyter, 2021, pp. 1–24.

[3] M. Forghani-elahabad. “3 The Disjoint Minimal Paths Reliability Problem”. Em: **Operations Research**. CRC Press, 2022, pp. 35–66.

[4] M. Forghani-elahabad e O. M. Alsalami. “Using a Node–Child Matrix to Address the Quickest Path Problem in Multistate Flow Networks under Transmission Cost Constraints”. Em: **Mathematics** 11.24 (2023), p. 4889.

DOI: 10.5540/03.2026.012.01.0275

[5] M. Forghani-elahabad e N. Mahdavi-Amiri. "A New Algorithm for Generating All Minimal Vectors for the q SMPs Reliability Problem With Time and Budget Constraints". Em: **IEEE Transactions on Reliability** 65.2 (2015), pp. 828–842.

[6] M. Forghani-elahabad e N. Mahdavi-Amiri. "An efficient algorithm for the multi-state two separate minimal paths reliability problem with budget constraint". Em: **Reliability Engineering & System Safety** 142 (2015), pp. 472–481.

[7] M. Forghani-elahabad e N. Mahdavi-Amiri. "An improved algorithm for finding all upper boundary points in a stochastic-flow network". Em: **Applied Mathematical Modelling** 40.4 (2016), pp. 3221–3229.

[8] M. Forghani-elahabad e N. T. Matos. "Buscando os dois caminhos mínimos disjuntos mais confiáveis em uma rede de fluxo multiestado com restrição de tempo". Em: **Proceeding Series of the Brazilian Society of Computational and Applied Mathematics** 11.1 (2025), pp. 1–7.

[9] M. Forghani-elahabad e W. C. Yeh. "An improved algorithm for reliability evaluation of flow networks". Em: **Reliability Engineering & System Safety** 221 (2022), p. 108371.

[10] Z. Hao, W. C. Yeh, Z. Liu e M. Forghani-elahabad. "General multi-state rework network and reliability algorithm". Em: **Reliability Engineering & System Safety** 203 (2020), p. 107048.

[11] J. Kleinberg e E. Tardos. **Algorithm design**. Pearson Education India, 2006.

[12] C. L. Li, S.T. McCormick e D. Simchi-Levi. "Finding disjoint paths with different path-costs: Complexity and algorithms". Em: **Networks** 22.7 (1992), pp. 653–667.

[13] Y. K. Lin. "Extend the quickest path problem to the system reliability evaluation for a stochastic-flow network". Em: **Computers & Operations Research** 30.4 (2003), pp. 567–575.

[14] Y. F. Niu, Xiao-Yu Wan, X. Z. Xu e Dong Ding. "Finding all multi-state minimal paths of a multi-state flow network via feasible circulations". Em: **Reliability Engineering & System Safety** 204 (2020), p. 107188.

[15] Y. F. Niu, J. H. Wei e X. Z. Xu. "Computing the Reliability of a Multistate Flow Network with Flow Loss Effect". Em: **IEEE Transactions on Reliability** (2023).

[16] D. Ronen e Y. Perl. "Heuristics for finding a maximum number of disjoint bounded paths". Em: **Networks** 14.4 (1984), pp. 531–544.

[17] W. Samotij. "Counting independent sets in graphs". Em: **European journal of combinatorics** 48 (2015), pp. 5–18.

[18] J. Vygen. "NP-completeness of some edge-disjoint paths problems". Em: **Discrete Applied Mathematics** 61.1 (1995), pp. 83–90.

[19] J. Weiner, T.E. Andreas, X. Li, Y. Sun e K. Deb. "Solving the maximum edge disjoint path problem using a modified Lagrangian particle swarm optimisation hybrid". Em: **European Journal of Operational Research** 293.3 (2021), pp. 847–862.

[20] W. C. Yeh e M. Forghani-elahabad. "An efficient algorithm for sorting and duplicate elimination by using logarithmic prime numbers". Em: **Big Data and Cognitive Computing** 8.9 (2024), p. 96.

[21] W. C. Yeh e M. Forghani-elahabad. "An efficient parallel approach for binary-state network reliability problems". Em: **Annals of Operations Research** (2024), pp. 1–22.

DOI: 10.5540/03.2026.012.01.0275