Um Sistema de Busca de Motifs em Redes Biológicas

Alternative Title: A System For Motif Search In Biological Networks

Alexandre da Silva Freire
Escola de Artes, Ciências e Humanidades
Universidade de São Paulo
São Paulo, SP
asfreire@usp.br

Karla R. P. S. Lima
Escola de Artes, Ciências e Humanidades
Universidade de São Paulo
São Paulo, SP
ksampaiolima@usp.br

Diego Ignacio Zurita Rojas
Escola de Artes, Ciências e Humanidades
Universidade de São Paulo
São Paulo, SP
diego.zurita@usp.br

ABSTRACT

In this work, we investigate an important problem from the biology field, which consists in searching for specific patterns, named *motifs*, in networks which represent certain biological interactions, such as metabolic networks, regulatory networks or Protein-Protein Interaction (PPI) networks. Two linear integer models are proposed, one of them using the concept of representatives. We present computational experiments made with instances generated from PPI networks with approximately 8.000 proteins and 29.000 interactions among them. As experimentally verified, the two proposed models were able to solve all instances in a very satisfactory amount of computational time.

CCS CONCEPTS

• **Mathematics of computing** → **Combinatorial optimization**; **Discrete optimization**; *Combinatoric problems*; *Linear programming*; Solvers;

KEYWORDS

Integer Linear Programming, Motifs.

ACM Reference Format:
Alexandre da Silva Freire, Karla R. P. S. Lima, and Diego Ignacio Zurita Rojas. 2018. Um Sistema de Busca de Motifs em Redes Biológicas: Alternative Title: A System For Motif Search In Biological Networks. In *SBSI'18: XIV Brazilian Symposium on Information Systems, June 4–8, 2018, Caxias do Sul, Brazil.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3229345.3229378

1 INTRODUÇÃO

O uso de sistemas de informação na análise de dados biológicos, com a finalidade de reconhecer determinados padrões estruturais, dentre outras, tem auxiliado na obtenção de grandes avanços no campo da biologia. O vertiginoso avanço tecnológico tem viabilizado cada vez mais a automatização do tratamento de dados biológicos, que por sua natureza possuem estruturas complexas, cuja análise se torna impraticável sem a utilização de sistemas especializados de apoio. Grande parte dos problemas computacionais envolvidos na análise de tais dados são de difícil resolução, impondo grandes desafios tanto do ponto de vista prático como teórico. Tais dificuldades têm sido vencidas em grande parte graças à utilização de técnicas computacionais avançadas, bem como o emprego de tecnologia de ponta. Não raro, se faz necessário o apoio de todo um arsenal teórico (modelagem por programação linear inteira, teoria dos grafos, técnicas avançadas de pesquisa operacional, etc...) e tecnológico (softwares comerciais de otimização, computadores com múltiplos processadores, etc...) para a obtenção do êxito almejado. Um outro aspecto fundamental na realização de pesquisas científicas interdisciplinares, como no caso da bioinformática, é a sinergia com a qual pesquisadores de áreas distintas devem interagir, o que torna esse tipo de trabalho ainda mais enriquecedor e ao mesmo tempo desafiador.

Neste trabalho, investigamos o problema de busca por padrões estruturais específicos, denominados *motifs*, em redes que representam determinadas interações biológicas, tais como redes metabólicas, reguladoras ou de interação entre proteínas. Diferentes definições de motif têm sido apresentadas na literatura, levando em conta as peculiaridades de cada aplicação, sendo que a utilizada neste trabalho é a proposta por Lacroix, Fernandes e Sagot [15], que investigaram o problema no contexto de análise de redes metabólicas (ver também [9, 14]), no qual o conjunto de reações envolvidas na síntese e degradação de certas moléculas é representado através de uma rede e os motifs correspondem aos "módulos" em que essa rede pode ser decomposta, de forma a facilitar a interpretação das funções que cada um desses módulos desempenha no metabolismo celular.

1.1 Trabalhos Relacionados

A literatura sobre o problema de busca de motifs é vasta. Já foram consideradas diversas variantes do problema e gran-de parte dos trabalhos publicados propõem métodos não exatos (heurísticas, como por exemplo em [3], ou algoritmos de aproximação) ou não eficientes na prática (algoritmos de enumeração, algoritmos FPT, etc). No primeiro caso, trata-se de trabalhos mais práticos. Uma parcela significativa de tais trabalhos encontra-se catalogada em [16, 18].

Informalmente, no PROBLEMA DE BUSCA DE MOTIFS CONEXOS (PBMC), dado um grafo G em que cada um dos vértices possui uma determinada cor, deseja-se encontrar um subgrafo conexo de

SBSI'18, June 4–8, 2018, Caxias do Sul, Brazil

ACM ISBN 978-1-4503-6559-8/18/06.
https://doi.org/10.1145/3229345.3229378

G cujo conjunto de vértices contém exatamente as quantidades desejadas de cada uma das cores. Um tal subgrafo é chamado de *motif*. Na variante considerada neste trabalho, que denominaremos PROBLEMA DE BUSCA DE MOTIFS COM NÚMERO MÍNIMO DE COMPONENTES CONEXAS (PBM-MinCC), deseja-se minimizar o número de componentes conexas do motif (ou seja, não necessariamente o motif será um subgrafo conexo). Na Seção 3.2 definimos mais precisamente esses problemas.

Os principais resultados teóricos para o PBMC foram obtidos na linha de complexidade parametrizada [1], inaproximabilidade [11, 17] e algoritmos exatos [12, 13]. Primeiramente, foi provado que o PBMC em árvores (caso especial em que a rede de entrada pode ser representada como um grafo conexo acíclico) e com a restrição adicional de que cada cor deve ocorrer exatamente uma única vez no motif (sem repetição de cores no motif) é NP-completo [15]. Posteriormente, foi provado em [13] que o problema permanece NP-completo quando G é uma árvore de grau máximo três e sem repetição de cores no motif. Além disso, em termos de complexidade parametrizada, nesse mesmo trabalho foi provado que o PBMC em árvores é W[1]-difícil.

Para o PBM-MinCC, em [10] foram apresentados os seguintes resultados relacionados a aproximabilidade: o problema é APX-difícil, mesmo no caso especial em que G é um caminho; o problema é FPT (*fixed parameter tractable*) quando parametrizado pelo número de vértices do motif; se G for uma árvore, então o problema não é aproximável dentro da razão $c \log n$, para alguma constante $c > 0$, onde $n = |V|$; o problema é W[2]-difícil quando parametrizado pelo número de componentes conexas do motif.

Do ponto de vista de programação linear inteira, a modelagem da restrição de conexidade das componentes através de inequações lineares constitui o ponto central a ser cuidadosamente considerado. Esse tipo de restrição aparece em outros problemas de otimização, diferentes do investigado aqui, e há resultados na literatura a esse respeito. No caso em que o grafo de entrada pode conter ciclos, em [6] é apresentado um estudo sobre alternativas de como modelar esse tipo de restrição. No caso em que o grafo de entrada é uma árvore, em [5] é proposto um modelo para o problema de recoloração convexa de árvores, no qual há restrições de conexidade. Em [7], tratando do mesmo problema, é apresentado um modelo mais compacto e que "domina" a formulação proposta em [5], sendo assim o trabalho [7] o que melhor modela a restrição de conexidade em árvores. Conforme será discutido na Seção 4, somente as ideias encontradas em [7] não são suficientes para se chegar a um bom modelo para o PBM-MinCC, do ponto de vista teórico. Nos baseaeremos também no conceito de *representantes*, introduzido em [4], em um dos modelos que será proposto.

1.2 Contribuições

Como contribuição, neste trabalho propomos um novo modelo de programação linear inteira – inspirado no modelo proposto em [7] para o problema de recoloração convexa de árvores – para uma das variantes do problema de busca de motifs, na qual a rede contendo os dados biológicos possui uma estrutura de árvore, podendo ser representada através de um grafo conexo acíclico. Apresentamos experimentos computacionais com instâncias obtidas através de dados reais de redes de interação entre proteínas. Apesar de os resultados práticos obtidos terem sido satisfatórios, conforme verificado experimentalmente, identificamos um ponto fraco no modelo proposto, do ponto de vista teórico. Para sanar tal problema, apresentamos um segundo modelo, no qual utilizamos o conceito de *representantes*, introduzido em [4]. Para o conjunto de instâncias considerado neste trabalho, conforme verificado experimentalmente, ambos os modelos são computacionalmente eficientes.

1.3 Organização do trabalho

O restante do texto está organizado da seguinte maneira. Na Seção 2, mencionamos quais são os principais objetivos deste trabalho e expomos a motivação para investigar o problema em questão. Na Seção 3 enunciamos o problema de busca de motifs de forma mais precisa, estabelecemos a metodologia empregada neste trabalho, bem como formalizamos os principais conceitos e notações utilizadas no restante do texto. Na Seção 4 introduzimos duas novas formulações de programação linear inteira para o problema de busca de mofis em árvores; os resultados experimentais obtidos com tais formulações são apresentados na Seção 5. Por fim, na Seção 6 apresentamos algumas conclusões sobre o que foi exposto e sugerimos alguns trabalhos futuros.

2 MOTIVAÇÃO E OBJETIVOS

Pesquisadores de diversas áreas frequentemente se deparam com problemas de natureza discreta, conhecidos como NP-difíceis. Sabe-se que poucas são as técnicas que apresentam bons resultados práticos, em termos de solução exata, para problemas dessa magnitude. De fato, a complexidade envolvida restringe bastante as possibilidades computacionais. Por outro lado, as técnicas de modelagem por programação linear inteira têm se mostrado uma poderosa ferramenta para se abordar problemas desta natureza. Em se tratando do problema de busca de motifs, pelo fato de o problema ser relativamente recente, os resultados mais relevantes foram obtidos nos últimos dez anos. Assim, a motivação para estudar tal problema surgiu da constatação de que, até o início desta pesquisa, não havia na literatura abordagens baseadas em programação linear inteira para a variante do problema considerada aqui.

Os principais objetivos deste trabalho são: propor novos modelos de programação linear inteira para o problema de busca de motifs com número mínimo de componentes conexas em árvores (na Seção 3.2 este problema será enunciado formalmente) e apresentar experimentos computacionais com instâncias reais de uma aplicação em biologia.

3 METODOLOGIA

A metodologia empregada neste trabalho consiste em utilizar conceitos de teoria dos grafos e de programação linear inteira (PLI), a fim de propor novos modelos de PLI para o PBM-MinCC. Do ponto de vista teórico, temos que argumentar acerca da corretude de tais modelos e tentar prever o comportamento que eles terão na prática, quando utilizados para resolver instâncias reais do problema, com o apoio de softwares de otimização. Para tanto, necessitaremos de alguns conceitos e notações apresentados nas Seções 3.1, 3.2 e 3.3 e que serão utilizadas no restante do texto. Do ponto de vista prático,

apresentaremos resultados experimentais obtidos através dos modelos propostos, juntamente com o apoio de um software comercial de otimização. Neste aspecto, explicamos na Seção 3.4 como as instâncias da aplicação foram obtidas e fornecemos todas as informações relevantes relacionadas à realização dos experimentos computacionais.

3.1 Teoria dos grafos e multiconjuntos

Os conceitos de teoria dos grafos a serem apresentados aqui são básicos e podem ser encontrados no livro de Bondy e Murty [2].

Um *grafo simples* G é um par ordenado (V, E), onde V é um conjunto finito de elementos chamados *vértices* e E é um conjunto de elementos chamados *arestas*, sendo que cada aresta é um par não-ordenado de vértices distintos.

Se $\{u, v\}$ (ou simplesmente uv) é uma aresta, dizemos que uv *incide* em u e em v, ou que u e v são seus *extremos* ou ainda que u e v são *adjacentes*. Um *caminho* num grafo G é uma sequência de vértices distintos $P = (v_1, v_2, \ldots, v_k)$, tal que $v_i v_{i+1} \in E$ para $i = 1, \ldots, k-1$. Um grafo G é *conexo* se para qualquer par de vértices distintos u e v existe um caminho de u a v em G. Uma *árvore* é um grafo conexo e *acíclico*, no sentido de que para todo par de vértices u e v, existe apenas um único caminho de u para v em G. Se dois grafos $G = (V, E)$ e $H = (W, B)$ são tais que $V \subseteq W$ e $E \subseteq B$, então G é dito um *subgrafo* de H. Todo grafo G, conexo ou não, pode ser expresso unicamente como a união disjunta de subgrafos conexos *maximais* de G, no sentido de que não existe subgrafo conexo de G que contém propriamente nenhum desses tais subgrafos, que são chamados de *componentes conexas* (ou simplesmente *componentes*) de G.

Uma *coloração* de um grafo $G = (V, E)$ é uma função $C : V \to \mathcal{C}$, onde $\mathcal{C}$ é um conjunto de cores. A coloração aqui definida corresponde a uma simples atribuição de cores aos vértices do grafo, sem nenhuma restrição. Um *grafo colorido* é um par (G, C) que consiste em um grafo G e uma coloração C dos vértices de G. Se C é uma coloração, então $\mathcal{C}$ denota o conjunto de cores usadas por C.

Um *multiconjunto* é definido como um par (A, m), onde A é um conjunto qualquer e $m : A \to \mathbb{N}_+$ é uma função que associa a cada elemento a de A um número natural positivo $m(a)$, que corresponde à multiplicidade de a.

3.2 Definição do Problema

No problema de busca de motifs (PBM), a rede contendo os dados biológicos é representada através de um grafo $G = (V, A)$. A cada vértice v de V é atribuída uma "cor" $C(v)$. Um *motif* em G é um subgrafo de G cujo conjunto de vértices contém exatamente as quantidades desejadas de cada uma das cores. Mais especificamente, dado um certo conjunto de cores $\mathcal{C}$ e um multiconjunto $\mathcal{M} = (C, m)$, o valor de $m(c)$ representa a quantidade de vértices da cor c que devem estar contidos no motif a ser encontrado em G, para cada cor c de $\mathcal{C}$. Por simplicidade, às vezes nos referimos a $\mathcal{M}$ como sendo um motif, embora nossa definição indique que um motif é um subgrafo de G e não um multiconjunto de cores.

Na versão clássica do problema, que denominaremos problema de busca de motifs conexos (PBMC), os vértices do motif devem formar um subgrafo conexo de G. Consideraremos neste trabalho uma outra variante do problema, denominada problema de busca de motifs com número mínimo de componentes conexas (PBM-MinCC), na qual deseja-se minimizar o número de componentes conexas do motif (ou seja, não necessariamente o motif será um subgrafo conexo).

Note que a resolução do PBM-MinCC implica na resolução do PBMC, no sentido de que se o número de componentes do motif encontrado for 1, temos uma solução para o segundo problema; caso contrário, a instância em questão não possui solução viável do PBMC (ou seja, G não possui um subgrafo conexo cujo conjunto de vértices contenha exatamente as quantidades desejadas de cada uma das cores). Mesmo neste último caso, uma solução do PBM-MinCC pode ser bastante útil, pois frequentemente os dados biológicos estão corrompidos – seja por falhas nas metodologias utilizadas na captação e/ou interpretação dos dados ou seja pela indisponibilidade de parte dos dados por questão de limitações técnicas – e, nessas condições, um motif com número mínimo de componentes conexas serve como uma boa aproximação do padrão que se desejava encontrar ou também como um ponto de partida para a correção de tais dados.

A seguir, enunciamos de forma mais precisa o problema a ser investigado neste trabalho.

Problema: Busca de motifs com número mínimo de componentes conexas (PBM-MinCC) em árvores:
Entrada: Uma tupla $(G, \mathcal{C}, C, \mathcal{M})$, onde $G = (V, A)$ é uma árvore, $C\colon V \to \mathcal{C}$ é uma coloração dos vértices de G e $\mathcal{M} = (C, m)$ é um multiconjunto de cores, onde $m\colon C \to \mathbb{N}_+$.
Saída: Um subgrafo H de G cujo multiconjunto de cores de seus vértices seja $\mathcal{M}$ e o número de componentes conexas de H seja mínimo.

3.3 Conceitos de programação linear inteira

Os conceitos de programação linear inteira a serem apresentados aqui são básicos e podem ser encontrados no livro de Wolsey [19].

Por simplicidade, consideraremos em nossas definições apenas problemas de minimização. Um problema de programação linear inteira (PLI) pode ser enunciado da seguinte forma: dados vetores $c \in \mathbb{R}^m$, $b \in \mathbb{R}^n$ e uma matriz $A \in \mathbb{R}^{n \times m}$, encontrar um vetor $x \in \mathbb{N}^m$ que minimize $c^\top x$, tal que $Ax \geq b$. Abaixo, apresentamos um programa linear inteiro na notação tipicamente utilizada na literatura.

$$(\text{PLI}_{\text{EXMPL}}) \qquad \begin{array}{lrcl} & \min & cx & \\ \text{sujeito a} & Ax & \geq & b \\ & x & \in & \mathbb{N}^m \end{array}$$

Dizemos que $(\text{PLI}_{\text{EXMPL}})$ é uma *formulação* (ou um *modelo*) para um problema P se qualquer solução ótima de $(\text{PLI}_{\text{EXMPL}})$ corresponde a uma solução ótima de P. Muitos problemas de otimização combinatória podem ser formulados como um programa linear inteiro. Para obter a *relaxação linear* de $(\text{PLI}_{\text{EXMPL}})$, denotada por $(\text{RL}_{\text{EXMPL}})$, substituímos a restrição de integralidade por $x \geq \mathbf{0}$, obtendo assim um programa linear cujo valor de uma solução ótima pode ser utilizado como um *limitante inferior* no valor de uma solução ótima de $(\text{PLI}_{\text{EXMPL}})$. É sabido que qualquer programa linear pode ser resolvido em tempo polinomial. Porém, de forma geral, o problema de encontrar uma solução ótima para um programa linear inteiro é NP-difícil. Usualmente, os resolvedores de modelos

de PLI utilizam a relaxação linear do modelo em questão, juntamente com heurísticas, *branch-and-bound* e outras técnicas, para obter uma solução ótima inteira. Para que este processo seja eficiente, é importante que a relaxação linear seja boa o suficiente para produzir um bom limitante inferior. Mais precisamente, quanto melhor for o modelo em questão, mais os valores cx^* e cx^{**} estarão próximos um do outro, onde x^* e x^{**} correspondem a soluções ótimas de $(\text{PLI}_{\text{EXMPL}})$ e $(\text{RL}_{\text{EXMPL}})$, respectivamente.

3.4 Metodologia dos experimentos

A seguir, descrevemos como os dados de redes biológicas foram obtidos e como os modelos de PLI foram implementados, bem como especificamos as configurações do ambiente computacional utilizado.

3.4.1 Geração de instâncias a partir de redes PPI. Para a realização dos experimentos computacionais, foram obtidas através do sítio http://igm.univ-mlv.fr/AlgoB/gramofone/ três redes de interação entre proteínas, conhecidas como *Protein-Protein Interaction Networks* (PPI-*Networks*), a saber: *Saccharomyces Cerevisiae* (SC) (com aproximadamente 5.500 proteínas e 40.000 interações entre elas), *Drosophila Melanogaster* (DM) (com aproximadamente 6.500 proteínas e 21.000 interações entre elas) e *Homo Sapiens* (HS) (com aproximadamente 8.000 proteínas e 29.000 interações entre elas). No mesmo sítio foram obtidos três arquivos, denominados *Yeast*, *Homo Sapiens* e *Fly*, cada um deles contendo diversos motifs de interesse a serem buscados em uma das redes mencionadas acima. Dado o grande número de motifs contidos em cada um dos arquivos, selecionamos alguns deles, filtrando por um certo tamanho mínimo estipulado.

Cada rede mencionada acima é representada através de um grafo H com "pesos" associados às arestas. Os vértices representam proteínas e a existência de uma aresta entre dois vértices indica que possivelmente há interação entre as respectivas proteínas, sendo que o peso da aresta, que está no intervalo (0,1], indica o "grau de confiabilidade" de que a interação realmente ocorre. A partir de cada grafo H, obtemos uma árvore geradora G de H utilizando uma adaptação do algoritmo de Kruskal [8] em que a ordenação das arestas é invertida, resultando em uma *árvore geradora máxima*. A justificativa para tal abordagem é a preferência em manter as arestas de maior peso, que são justamente as que indicam maior grau de confiabilidade de que as respectivas proteínas realmente interagem entre si.

Cada motif contido nos arquivos mencionados acima é representado por um conjunto de proteínas, de forma que deseja-se encontrar um subgrafo de G que contenha proteínas similares às proteínas do motif. A similaridade entre duas proteínas é expressa através de uma função que atribui "cores" às proteínas, sendo que duas proteínas consideradas similares recebem uma mesma cor. Nas aplicações reais, tal comparação entre proteínas é realizada comparando-se as respectivas sequências de DNA, através de softwares especializados, a fim de determinar se há similaridade funcional e/ou estrutural entre elas. No entanto, como não nos foi possível obter tais informações, como alternativa optamos por comparar os tamanhos das sequências, embora a rigor tal comparação não seja adequada para determinar o grau de similaridade entre as proteínas. Mais precisamente, foram definidos intervalos numéricos de tamanho fixo e a coloração dos vértices de G foi gerada comparando os tamanhos das sequências de DNA das proteínas, sendo que dois vértices recebem a mesma cor se e só se os respectivos tamanhos de suas sequências estão em um mesmo intervalo numérico.

3.4.2 Implementação e ambiente computacional. Os modelos de PLI foram codificados em linguagem de programação Julia (https://julialang.org/). Para resolver os modelos gerados a partir das instâncias descritas acima, foi utilizado o software comercial de otimização CPLEX 12.5 (https://www.ibm.com/analytics/data-science/prescriptive-analytics/cplex-optimizer), sendo que o tempo medido em nossos experimentos foi apenas o tempo de resolução de cada modelo, tendo sido desprezado o tempo necessário para carregar os modelos no resolvedor. As configurações do ambiente computacional são as seguintes: 4 processadores Intel(R) Xeon(TM) CPU 3.00GHz; 8GB de memória RAM; sistemas operacional Debian GNU/Linux 9.3 (stretch).

4 FORMULAÇÕES DE PLI

Na Seção 4.1, propomos uma nova formulação de programação linear inteira para o PBM-MinCC, baseada no modelo apresentado em [7]. Conforme será discutido nas próximas seções, para tal modelo, identificamos um ponto fraco do ponto de vista teórico, ainda que os resultados práticos tenham sido satisfatórios. Para sanar tal problema, na Seção 4.2 propomos um segundo modelo, no qual utilizamos o conceito de *representantes*, introduzido em [4].

4.1 Uma primeira formulação

Denotamos por $(G, C, \mathcal{M})$ uma instância do PBM-MinCC, onde G é uma árvore, $C : V \to C$ é uma coloração de G e $\mathcal{M} = (C, m)$ é um multiconjunto de cores de C, que indica a multiplicidade com que cada cor deve ocorrer no motif. Por simplicidade, assumiremos que as cores usadas na coloração dos vértices de G são as mesmas que aparecem no motif; caso contrário, poderíamos realizar um pré-processamento removendo de G todos os vértices que são coloridos com cores que não pertencem ao motif. Tal pré-processamento poderia fazer com que o grafo de entrada se tornasse uma *floresta* (coleção de árvores disjuntas) e não mais uma árvore. No entanto, isso não afetaria em basicamente nada os modelos que serão propostos.

Definimos uma variável binária $x_u \in \{0, 1\}$, para todo vértice u de V, com a interpretação de que $x_u = 1$ indica que o vértice u foi selecionado para compor a solução e $x_u = 0$, caso contrário. Para modelar a restrição de conexidade das componentes do motif, introduzimos, como em [7], uma variável binária $y_{uv} \in \{0, 1\}$, para toda aresta uv de E, com a interpretação de que $y_{uv} = 1$ indica que a aresta uv foi selecionada para compor a solução e $y_{uv} = 0$, caso contrário. Conforme mostraremos adiante, esses dois conjuntos de variáveis serão vinculados para que haja coerência entre os valores de x e y, através de um conjunto de restrições.

A seguir, apresentamos uma primeira formulação para o PBM-MinCC.

$$\min \sum_{v \in V} x_v - \sum_{uv \in E} y_{uv} \tag{1}$$

$$s.a \sum_{v \in V_c} x_v = m(c), \qquad \forall c \in C \tag{2}$$

$$y_{uv} \leq x_u, \qquad \forall uv \in E \tag{3}$$

$$y_{uv} \leq x_v, \qquad \forall uv \in E \tag{4}$$

$$x_v \in \{0,1\}, \qquad \forall v \in V \tag{5}$$

$$y_{uv} \in \{0,1\}, \qquad \forall uv \in E \tag{6}$$

Denotamos por (PLI-1) o modelo acima. A função objetivo (1) minimiza a diferença entre o número de vértices e arestas selecionados para compor a solução, o que corresponde ao número de componentes da solução. A restrição (2) garante que, para cada cor c, a quantidade de vértices escolhidos com a cor c, no conjunto V_c (vértices com a cor c em V), seja exatamente $m(c)$, ou seja, a multiplicidade com que cada cor ocorre na solução seja a mesma definida no motif buscado. A vinculação entre as variáveis x e y se dá através da função objetivo e das restrições (3) e (4). Mais precisamente, a função objetivo faz com que cada variável y_{uv} assuma o valor 1 sempre que possível; já as restrições (3) e (4) fazem com que isso só possa acontecer se $x_u = x_v = 1$. Por fim, a integralidade das variáveis é garantida através das restrições (5) e (6).

4.1.1 Qualidade do limitante inferior. Para obter a relaxação linear de (PLI-1), que denotaremos por (RL-1), substituímos as restrições de integralidade (5) e (6) por $x_u \geq 0$, para todo $u \in V$, e $y_{uv} \geq 0$, para todo $uv \in E$, respectivamente, obtendo assim um programa linear cujo valor de uma solução ótima pode ser utilizado como um limitante inferior no valor de uma solução ótima de (PLI-1). Mais precisamente, temos que $optRL \leq optPLI$, onde $optRL$ e $optPLI$ correspondem aos valores na função objetivo das soluções ótimas de (RL-1) e (PLI-1), respectivamente. Por se tratar de uma relaxação linear, o modelo (RL-1) pode ser resolvido em tempo polinomial, o que já não ocorre com (PLI-1), que corresponde a encontrar uma solução ótima inteira para o PBM-MinCC, que é NP-difícil. Usualmente, os resolvedores de modelos de PLI utilizam a relaxação linear do modelo em questão como parte do processo de obtenção de uma solução ótima inteira. Para que este processo seja eficiente, é importante que a relaxação linear seja boa o suficiente para produzir um bom limitante inferior; ou seja, seria importante que o valor de $gap = optPLI - optRL$ fosse o menor possível. A seguir, mostraremos como construir uma classe de instâncias para as quais o valor de gap cresce proporcionalmente ao tamanho da instância.

Sejam k e q duas constantes inteiras e positivas, tais que $q < k$. Para qualquer inteiro $z > 0$, construiremos uma instância $I^z_{k,q} = (G, C, \mathcal{M})$, onde G é um caminho de $n = k^2z + kz - k$ vértices; $\mathcal{M} = (C, m)$ é um multiconjunto de cores, tal que $m(c) = q$, para todo $c \in C$ (ou seja, a frequência com que cada cor deve ocorre no motif é fixa e igual a q); o conjunto de cores C terá cardinalidade $|C| = zk + z - 1$. Considere os vértices do caminho G rotulados de 1 a n, assim o caminho G será escrito da forma $(1, 2, \ldots, n)$. A coloração $C : V \to C$ dos vértices de G é construída como segue. O caminho G é composto por $zk - 1$ subcaminhos com k vértices, separados por um vértice. Há também um vértice antes do primeiro subcaminho e outro depois do último subcaminho. Chamaremos esses subcaminhos de "blocos internos". Todos os vértices que não estão contidos em nenhum desses blocos internos serão chamados de vértices "separadores". Assim, teremos kz vértices separadores e $zk - 1$ blocos internos, cada um deles de tamanho k. Todos os k vértices de um bloco interno são coloridos com a mesma cor e cada um desses blocos possui uma cor distinta. Os vértices separadores são agrupados em z "blocos espaçados" de tamanho k, sendo que cada um desses blocos contém os próximos k vértices separadores, considerando a ordem em que eles ocorrem no caminho G. Analogamente à coloração dos blocos internos, todos os vértices de um bloco espaçado são coloridos com a mesma cor e cada um desses blocos possui uma cor distinta que não coincide com a cor de nenhum dos blocos internos.

Por exemplo, para uma instância $I^z_{k,q} = (G, c, \mathcal{M})$ tal que $k = 2$, $q = 1$ e $z = 3$, supondo que $C = \{1, 2, \ldots, |C|\}$, onde $|C| = zk + z - 1 = 8$, temos que G será o caminho $(1, 2, \ldots, n)$ com $n = k^2z + kz - k = 2^2 \cdot 3 + 2 \cdot 3 - 2 = 16$ vértices, sendo $zk - 1 = 3 \cdot 2 - 1 = 5$ blocos internos de tamanho $k = 2$, com as cores $\{1, 2, \ldots, 5\}$, e $z = 3$ blocos espaçados de tamanho $k = 2$, com as cores $\{6, 7, 8\}$. Portanto, as cores dos vértices são $(\mathbf{6}, 1, 1, \mathbf{6}, 2, 2, \mathbf{7}, 3, 3, \mathbf{7}, 4, 4, \mathbf{8}, 5, 5, \mathbf{8})$, ou qualquer permutação dessas cores. As cores em negrito indicam as posições dos vértices separadores. A frequência com que cada cor deve ocorrer no motif é $q = 1$. Repare que se atribuirmos o valor $\frac{1}{2}$ para todas as variáveis do modelo (RL-1), teremos uma solução que satisfaz todas as restrições do modelo e cujo valor na função objetivo será $|V| \cdot \frac{1}{2} - |E| \cdot \frac{1}{2} = \frac{1}{2}$. Já o valor na função objetivo de uma solução ótima de (PLI-1) para a mesma instância será 3.

De forma geral, para uma instância $I^z_{k,q}$, podemos atribuir o valor $\frac{q}{k}$ para todas as variáveis e, assim, obteremos uma solução que satisfaz todas as restrições do modelo (RL-1) e cujo valor na função objetivo será $|V| \cdot \frac{q}{k} - |E| \cdot \frac{q}{k} = \frac{q}{k} < 1$. Como (RL-1) é um problema de minimização, temos que $optRL \leq \frac{q}{k}$. De fato, $optRL = \frac{q}{k} = O(1)$ mas não há necessidade de demonstrar esse fato. O valor de $optRL$ não é, absolutamente, um bom limitante inferior, pois é fácil ver que qualquer solução viável do PBM-MinCC, para qualquer instância, sempre terá pelo menos uma componente e, portanto, o limitante obtido através da relaxação linear não está sendo melhor do que o limitante trivial.

Vejamos agora qual é o valor na função objetivo de uma solução ótima de (PLI-1) para a mesma instância $I^z_{k,q}$. Como $q < k$, o tamanho de um bloco interno será sempre maior do que a frequência com que a respectiva cor deve ocorrer no motif, impossibilitando assim que dois vértices separadores fiquem em uma mesma componente. Além disso, a cor correspondente a cada um dos z blocos espaçados deve ocorrer q vezes no motif, assim teremos pelo menos zq componentes na solução e, por último, é fácil ver que, zq componentes são suficiente para compor uma solução viável, já que, para cada bloco interno os q vértices necessários da cor desse bloco estarão na mesma componente de algum vértice separador que é "vizinho". Logo, o valor de uma solução ótima de (PLI-1) para a instância $I^z_{k,q}$ será $optPLI = qz = \left(\frac{n+k}{k^2+k}\right) \cdot q = \Theta(n)$, onde n

é a quantidade de vértices de G e, consequentemente, temos que o valor de $gap = optPLI - optRL = \Theta(n)$ cresce proporcionalmente ao tamanho da instância.

4.2 Uma formulação mais forte

Usualmente, o que se faz para melhorar a qualidade do limitante obtido através de um modelo de PLI é "fortalecer" o modelo com novas desigualdades que são satisfeitas por todas as soluções inteiras mas que, por outro lado, são violadas por muitas soluções fracionárias que antes eram viáveis na relaxação linear.

Observe que, para quaisquer dois vértices u e v, se a frequência com que alguma cor c ocorre no caminho de u a v for maior que $m(c)$, temos que u e v não podem estar em uma mesma componente; neste caso, diremos que u e v são *incompatíveis*. Ademais, dado um conjunto de vértices W dois a dois incompatíveis, temos que cada componente da solução pode conter no máximo um vértice de W. Esta observação será explorada por uma nova classe de desigualdades que será apresentada adiante.

No modelo (PLI-1), não é simples determinar se dois vértices u e v estão ou não em uma mesma componente, pois as variáveis x indicam apenas quais vértices estão ou não na solução. Para obter tal informação, poderíamos verificar se todas as variáveis y correspondentes às arestas internas ao caminho de u a v possuem valor igual a 1, mas isso complicaria a inclusão de desigualdades que utilizam essa informação no modelo.

Uma outra forma, mais simples, de obter tal informação é redefinir as variáveis x, de modo que elas indiquem um rótulo para a componente na qual cada vértice selecionado para compor a solução está contido. Mais precisamente, seria definida uma variável binária $x_{ui} \in \{0,1\}$, para todo $u \in V$ e $i \in I$, onde I é um conjunto de rótulos para as componentes, com a interpretação de que $x_{ui} = 1$ se e só se o vértice u está na componente de rótulo i. Porém, ao redefinir as variáveis x dessa forma, o conjunto de soluções viáveis aumenta de forma drástica, devido à *simetria* que passa a existir entre as soluções obtidas a partir de permutação de rótulos, podendo comprometer a eficiência do método de resolução do modelo em questão.

Uma maneira de eliminar simetria entre soluções diferentes, mas que essencialmente são iguais, é através do uso de *representantes*, como apresentado em [4] para o problema de coloração de vértices. A seguir, explicamos como utilizar essa técnica para remodelar (PLI-1). Ao invés de rotular as componentes, elegemos um representante para cada componente S da solução, sendo que esse representante deve ser um vértice de S a ser determinado de maneira unívoca, conforme descreveremos a seguir.

Redefiniremos as variáveis x de forma que $x_{uv} = 1$ se e só se o vértice u está na componente representada pelo vértice v; neste caso, por simplicidade, diremos que u é representado por v. Diremos que um vértice $u \in V$ é um *representante* se $x_{uu} = 1$. Tomamos um vértice r qualquer de V e *enraizamos* a árvore G em r. Denotaremos por G^r a árvore G enraizada em r. Dada uma subárvore S de G, o vértice s que corresponde à raiz de S em G^r será o representante de S se a componente S for selecionada para compor a solução, ou seja, $x_{us} = 1$, para todo $u \in S$. Desta forma, elimina-se as simetrias indesejadas.

Definiremos as variáveis $x_{uv} \in \{0,1\}$, para todo $u \in V$ e $v \in R_u$. O conjunto R_u deve conter apenas os vértices que possivelmente seriam representantes de alguma componente em que u está contido. Observe que, dado um vértice u, os únicos vértices "candidatos" a pertencer a R_u são os vértices contidos no caminho de u até a raiz r de G^r (ou seja, os "ancestrais" de u em G^r, ou o próprio u), exceto os vértices que são incompatíveis com u, já que, conforme mencionado anteriormente, dois vértices incompatíveis nunca estarão em uma mesma componente na solução. Mais precisamente, para todo $u \in V$, definimos $R_u = \{v \in V \mid v = u$ ou v é ancestral de u em G^r e não é incompatível com $u\}$. Sejam $H_w = \{u \in V \mid w \in R_u\}$, para todo $w \in V$, e $Q_w = \{uv \in E \mid w \in R_u \cap R_v\}$, para todo $uv \in E$ e $w \in V$. Ou seja, o conjunto H_w contém todos os vértices que podem ser representados por w e o conjunto Q_w contém todas as arestas cujos extremos podem ser representados por w.

As variáveis y também serão redefinidas. Para toda aresta uv de E e todo vértice $w \in R_{uv} = R_u \cap R_v$, seja $y^w_{uv} \in \{0,1\}$, com a interpretação de que $y^w_{u,v} = 1$ se e só se, a aresta uv está na componente representada por w. A seguir, apresentamos um segundo modelo de PLI, denotado por (PLI-2), para o PBM-MinCC:

$$\min \sum_{w \in V} \left(\sum_{u \in H_w} x_{uw} - \sum_{uv \in Q_w} y^w_{uv} \right) \tag{7}$$

$$\text{s.a} \sum_{v \in R_u} x_{uv} \leq 1, \qquad \forall u \in V \tag{8}$$

$$\sum_{u \in V_c} \sum_{v \in R_u} x_{uv} = m(c), \qquad \forall c \in C \tag{9}$$

$$y^w_{uv} \leq x_{uw}, \qquad \forall uv \in E, w \in R_{uv} \tag{10}$$

$$y^w_{uv} \leq x_{vw}, \qquad \forall uv \in E, w \in R_{uv} \tag{11}$$

$$x_{uv} \leq x_{vv}, \qquad \forall u, v \in V \tag{12}$$

$$x_{vw} \in \{0,1\}, \qquad \forall v \in V, \forall w \in R_u, \tag{13}$$

$$y^w_{uv} \in \{0,1\}, \qquad \forall uv \in E, \forall w \in R_{uv} \tag{14}$$

A função objetivo (7) minimiza a soma das diferenças entre o número de vértices e arestas de cada componente selecionada para compor a solução, o que equivale a minimizar o número de componentes da solução. Note que as funções objetivos (1) e (7) são similares, com a diferença que em (7) há um somatório externo em que cada possível representante w é fixado, para então no somatório interno ser calculada a diferença entre o número de vértices e arestas da respectiva componente (1 se há uma componente representada por w e 0 caso contrário).

A restrição (8) garante que cada vértice u de V pode ser representado por no máximo um vértice de R_u. A restrição (9) impõe que a multiplicidade com que cada cor ocorre na solução seja a mesma definida no motif buscado. As variáveis x e y estão interligadas através da função objetivo e das restrições (10) e (11), de forma análoga ao que ocorre no modelo (PLI-1), porém cada componente representada por um vértice w é considerada separadamente. A restrição (12) impede que um vértice u seja representado por um vértice v que não foi selecionado para representar uma componente da solução. As restrições (13) e (14) garantem a integralidade das variáveis.

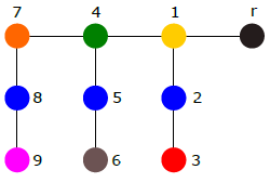

Figura 1 - Um conjunto $I = \{3, 6, 9\}$ em que os vértices são dois a dois incompatíveis, considerando que $m(azul) = 1$ no motif de interesse.

Claramente, o modelo (PLI-1) apresentado na seção 4.1 é mais compacto do que o modelo (PLI-2), comparando o número de variáveis e restrições incluídas em tais modelos. Em contrapartida, para a classe de instâncias construídas conforme exposto na Seção 4.1.1, utilizando a relaxação linear de (PLI-2), denotada por (RL-2), obtem-se um limitante inferior justo (ou seja, igual ao valor da solução ótima inteira). Isso ocorre porque na definição de R_u, para cada vértice u de V, são considerados apenas os vértices compatíveis com u, implicando na eliminação em (RL-2) de diversas soluções fracionárias que seriam viáveis no modelo (RL-1). Além disso, utilizando a relaxação (RL-2) é possível explorar as observações feitas no início desta seção para incluir novas desigualdades, de forma a produzir um limitante inferior ainda melhor, conforme apresentaremos abaixo.

Seja $\mathcal{I} = \{I \subset V \mid$ para todo par de vértices distintos u e v de I, temos que u e v são incompatíveis$\}$ e seja $R_I = \bigcap_{u \in I} R_u$. Considere as seguintes desigualdades:

$$\sum_{u \in I} x_{uw} \leq 1, \quad \forall\, I \in \mathcal{I}, \forall\, w \in R_I \tag{15}$$

Para um exemplo da aplicação da restrição (15), considere o exemplo da Figura 1, em que $m(azul) = 1$ no motif buscado. Note que os vértices do conjunto $I = \{3, 6, 9\}$ são dois a dois incompatíveis e compartilham dois candidatos a representante em uma componente na solução, a saber $R_I = \{1, r\}$. Para este exemplo, a restrição (15) correspondente é $x_{3v} + x_{6v} + x_{9v} \leq 1$, para todo $v \in \{1, r\}$.

Como a quantidade de restrições (15) pode ser exponencial no tamanho de G, tais restrições devem ser adicionadas no modelo utilizando o método de *planos-de-corte* [19]. Para tanto, é necessário desenvolver um algoritmo de *separação* para essas desigualdades, o que pretendemos fazer em trabalhos futuros.

5 RESULTADOS EXPERIMENTAIS

Apresentamos nesta seção os resultados dos experimentos computacionais realizados com os modelos de PLI propostos, bem como com suas respectivas relaxações lineares, utilizando as instâncias obtidas conforme descrito na Seção 3.4.

Na Tabela 1, identificamos cada instância através da abreviatura do nome da respectiva rede PPI, juntamente com o nome do motif. Para cada instância, são apresentadas as seguintes informações: o número de vértices do grafo G; o número de cores da coloração C; o número de vértices de V que foram coloridos com alguma cor cuja frequência é maior que zero no motif, denotado por $|V_{\mathcal{M}}|$; o tamanho do motif, que corresponde à soma das frequências com que cada cor aparece no motif, ou seja $|\mathcal{M}| = \sum_{c \in C} m(c)$; e, por fim, o número de cores distintas que ocorrem com frequência maior que zero no motif, denotado por $|C_{\mathcal{M}}|$.

Na Tabela 2, para cada instância, são apresentadas as seguintes informações: o valor da solução ótima inteira, obtida através a resolução dos modelos (PLI-1) e (PLI-2); os valores dos limitantes inferiores obtidos através da resolução das relaxações lineares (RL-1) e (RL-2); e, por fim, o número de coeficientes não-nulos da matriz de restrição de cada um dos modelos, o que nos fornece uma ideia bastante precisa do tamanho de tais modelos.

Tabela 1 - Identificação das Instâncias

		Grafo			Motif	
Id.	Nome Rede-Motif	$\|V\|$	$\|C\|$	$\|V_{\mathcal{M}}\|$	$\|\mathcal{M}\|$	$\|C_{\mathcal{M}}\|$
1	DM-mitochondrial	6594	269	1998	36	24
2	DM-kinetochore	6594	269	1930	61	48
3	DM-mediator	6594	269	1569	30	25
4	DM-VCB	6594	269	1502	31	20
5	DM-microtubule	6594	269	1733	41	37
6	HS-C	7664	318	2984	81	39
7	HS-Spliceosome	7664	318	4040	141	57
8	HS-Nop56p-associated	7664	318	3384	106	40
9	HS-60S	7664	318	1665	50	15
10	HS-Ribosome	7664	318	2092	82	20
11	SC-ribonucleoprotein	5385	216	4797	356	108
12	SC-transcription	5385	216	3409	131	108
13	SC-spliceosome5681	5385	216	3174	90	53
14	SC-ribosome5840	5385	216	3461	135	115
15	SC-DNA-directed	5385	216	2620	73	76

Tabela 2 - Limitantes inferiores e tamanhos dos modelos

		Lim. Inf.		Coef. Não-Nulos	
Id	Otm	(RL-1)	(RL-2)	(RL-1)	(RL-2)
1	6	5,22	5,23	4217	11764
2	19	18,00	18,50	4125	11628
3	9	8,88	8,88	3236	8496
4	7	6,17	6,67	2909	7142
5	**11**	**11,00**	**11,00**	3443	9122
6	8	7,13	7,13	8016	25738
7	16	14,98	15,03	12943	53248
8	**18**	**18,00**	**18,00**	9596	32718
9	12	11,40	11,42	3329	8420
10	21	20,25	20,25	4604	13044
11	**4**	**4,00**	**4,00**	21527	214620
12	9	8,56	8,75	13198	57722
13	6	5,21	5,28	9769	37416
14	10	8,29	8,38	13670	71094
15	5	4,33	4,33	7564	28940

Para quase a todalidade das instâncias, os dois modelos de PLI propostos, bem como suas respectivas relaxações lineares, foram resolvidos em menos de 15 segundos, sendo que na maioria dos casos o tempo gasto foi inferior a 3 segundos (conforme mencionado na Seção 3.4, apenas o tempo gasto na resolução dos modelos foi medido, tendo sido desprezado o tempo para carregar os modelos no CPLEX). No entanto, para as instâncias 11 e 14, o modelo (PLI-2) foi resolvido em 344 e 31 segundos, respectivamente, e o modelo (RL-2) foi resolvido em 52 e 7 segundos, respectivamente. Para essas mesmas instâncias, os modelos (PLI-1) e (RL-1) mantiveram-se

com as mesma eficiência obtida nas demais instâncias (neste caso, abaixo de 10 segundos).

Acreditamos que essa queda no desempenho do modelo (PLI-2), principalmente para a instância 11, se deve ao fato de se tratar dos maiores valores de $|\mathcal{M}|$ e $|V_{\mathcal{M}}|$, resultando em um número muito grande de coeficientes não-nulos na matriz de restrições desse modelo. Por ser mais compacto, o modelo (PLI-1) não teve seu desempenho tão afetado quanto o (PLI-2). Ainda assim, consideramos que, dado o tamanho da instância 11, o tempo gasto pelo modelo (PLI-2) foi satisfatório, de modo que podemos concluir que para o conjunto de testes considerados os resultados obtidos, em termos de tempo computacional gasto, foram excelentes.

Conforme mostrado na Tabela 2, os limitantes inferiores produzidos pelas relaxações lineares estão muito próximos dos respectivos valores das soluções inteiras ótimas, sendo que para as instâncias 5, 8 e 11, os limitantes obtidos pelas duas relaxações correspondem ao valor da solução ótima inteira. Isso explica o baixo tempo computacional gasto para a resolução dos modelos de PLI. Apesar de os limitantes obtidos por (RL-2) terem sido melhores do que os obtidos por (RL-1), não foi observada diferença significativa. Em experimentos preliminares, verificamos que o limitante obtido por (RL-2) se altera à medida em que alteramos a escolha do vértice r que será fixado como a raiz da árvore G. Porém, ainda não sabemos como fazer uma "boa escolha" de r. Além disso, a inclusão das restrições (15), utilizando o método de planos-de-corte, deve melhorar ainda mais a qualidade dos limitantes a serem obtidos por (RL-2).

Para melhor validar os modelos propostos, há a necessidade de se realizar experimentos computacionais mais amplos, obtendo-se mais dados de outras redes biológicas, tanto de redes PPI como de redes provenientes de outras aplicações.

6 CONCLUSÃO

Apresentamos neste trabalho dois modelos de programação linear inteira para um problema importante da área de biologia, que consiste na busca de motifs em redes bológicas. Apresentamos uma classe de instâncias para as quais o limitante inferior obtido pela relaxação linear do primeiro modelo é comprovadamente muito distante do valor da solução ótima inteira, o que inviabilizaria seu uso. O segundo modelo proposto se baseia na utilização de representantes e sua estrutura diferenciada possibilitou a inclusão de uma classe de desigualdades para fortalecer este modelo. Apresentamos experimentos computacionais realizados com instâncias geradas a partir de redes biológicas de interação entre proteínas, conhecidas como redes PPI, com até aproximadamente 8.000 proteínas e 29.000 interações entre elas. Conforme verificado experimentalmente, os dois modelos propostos foram capazes de solucionar todas as instâncias em um tempo computacional satisfatório.

Como trabalhos futuros, vislumbramos a realização de experimentos computacionais mais amplos, obtendo-se mais dados de outras redes biológicas, tanto de redes PPI como de redes provenientes de outras aplicações. Além disso, outro aspecto a ser explorado é a inclusão das desigualdades propostas através do método de planos-de-corte.

AGRADECIMENTOS

Durante o desenvolvimento deste trabalho, os autores A.S. Freire e K.R.P.S. Lima receberam apoio do CNPq (Projeto Universal – Proc. 456792/2014-7).

REFERÊNCIAS

[1] Nadja Betzler, Michael R. Fellows, Christian Komusiewicz, and Rolf Niedermeier. 2008. Parameterized Algorithms and Hardness Results for Some Graph Motif Problems. In *Combinatorial Pattern Matching*, Paolo Ferragina and Gad M. Landau (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 31–43.

[2] J.A. Bondy and U.S.R. Murty. 2008. *Graph Theory*. Springer.

[3] Felipe Brigatto and Karla R. Lima. 2015. Desenvolvimento de uma Formulação Linear Inteira para o Problema de Motifs em Grafos. In *II Workshop de Iniciação Científica em Sistemas de Informação – XI Simpósio Brasileiro de Sistemas de Informação*. 29–32.

[4] Manoel Campêlo, Victor A. Campos, and Ricardo C. Corrêa. 2008. On the asymmetric representatives formulation for the vertex coloring problem. *Discrete Applied Mathematics* 156, 7 (2008), 1097 – 1111. https://doi.org/10.1016/j.dam.2007.05.058 GRACO 2005.

[5] Manoel Campêlo, Alexandre S. Freire, Karla R. Lima, Phablo F. S. Moura, and Yoshiko Wakabayashi. 2016. The convex recoloring problem: polyhedra, facets and computational experiments. *Mathematical Programming* 156, 1 (2016), 303–330.

[6] Rodolfo Carvajal, Miguel Constantino, Marcos Goycoolea, Juan Pablo Vielma, and Andrés Weintraub. 2013. Imposing Connectivity Constraints in Forest Planning Models. *Operations Research* 61, 4 (2013), 824–836. https://doi.org/10.1287/opre.2013.1183 arXiv:https://doi.org/10.1287/opre.2013.1183

[7] Sunil Chopra, Bartosz Filipecki, Kangbok Lee, Minseok Ryu, Sangho Shim, and Mathieu Van Vyve. 2017. An extended formulation of the convex recoloring problem on a tree. *Mathematical Programming* 165, 2 (01 Oct 2017), 529–548. https://doi.org/10.1007/s10107-016-1094-3

[8] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. 2009. *Introduction to Algorithms, Third Edition* (3rd ed.). The MIT Press.

[9] Yves Deville, David R. Gilbert, Jacques van Helden, and Shoshana J. Wodak. 2003. An overview of data models for the analysis of biochemical pathways. *Briefings in Bioinformatics* 4, 3 (2003), 246–259. https://doi.org/10.1093/bib/4.3.246

[10] Riccardo Dondi, Guillaume Fertin, and Stéphane Vialette. 2007. Weak pattern matching in colored graphs: Minimizing the number of connected components. In *10th Italian Conference on Theoretical Computer Science (ICTCS 2007) (World-Scientific Conference Proceedings)*. World-Scientific Conference Proceedings, Rome, Italy, 27–38. https://hal.archives-ouvertes.fr/hal-00417910

[11] Riccardo Dondi, Guillaume Fertin, and Stéphane Vialette. 2009. Maximum Motif Problem in Vertex-Colored Graphs. In *Combinatorial Pattern Matching*, Gregory Kucherov and Esko Ukkonen (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 221–235.

[12] Riccardo Dondi, Guillaume Fertin, and StÃ©phane Vialette. 2013. Finding approximate and constrained motifs in graphs. *Theoretical Computer Science* 483 (2013), 10 – 21. https://doi.org/$http://dx.doi.org/10.1016/j.tcs.2012.08.023$ Special Issue Combinatorial Pattern Matching 2011.

[13] Michael R. Fellows, Guillaume Fertin, Danny Hermelin, and StÃ©phane Vialette. 2007. Sharp Tractability Borderlines for Finding Connected Motifs in Vertex-Colored Graphs.. In *ICALP* (2007-09-03) *(Lecture Notes in Computer Science)*, Lars Arge, Christian Cachin, Tomasz Jurdzinski, and Andrzej Tarlecki (Eds.), Vol. 4596. Springer, 340–351. http://dblp.uni-trier.de/db/conf/icalp/icalp2007.html#FellowsFHV07

[14] Brian P. Kelley, Roded Sharan, Richard M. Karp, Taylor Sittler, David E. Root, Brent R. Stockwell, and Trey Ideker. 2003. Conserved pathways within bacteria and yeast as revealed by global protein network alignment. *Proceedings of the National Academy of Sciences of the United States of America* 100, 20 (30 Sept. 2003), 11394–11399. https://doi.org/10.1073/pnas.1534710100

[15] Vincent Lacroix, Cristina G. Fernandes, and Marie-France Sagot. 2006. Motif Search in Graphs: Application to Metabolic Networks. *IEEE/ACM Transactions on Computational Biology and Bioinformatics* 3, 4 (2006), 360–368. https://doi.org/$http://doi.ieeecomputersociety.org/10.1109/TCBB.2006.55$

[16] Angela Makolo. 2016. A Comparative Analysis of Motif Discovery Algorithms. *Computational Biology and Bioinformatics* (2016). https://doi.org/10.11648/j.cbb.20160401.11

[17] Romeo Rizzi and Florian Sikora. 2015. Some results on more flexible versions of Graph Motif. *Theory of Computing Systems* 56, 4 (2015), 612–629. https://doi.org/$10.1007/s00224-014-9564-6$

[18] Geir Kjetil Sandve and Finn Drablos. 2006. A survey of motif discovery methods in an integrated framework. *Biology Direct* (2006). https://doi.org/10.1186/1745-6150-1-11

[19] Laurence A. Wolsey. 1998. *Integer Programming*. Wiley - Interscience Series in Discrete Mathematics and Optimization.