O problema do caminho mínimo k-rotulado

Emanuel Elias Silva Castelo
Universidade Federal do Ceará
Campus do Pici, Bloco 910, 60440-900. Fortaleza, CE, Brasil
eliaslvcastelo@gmail.com

Rafael Castro de Andrade
Universidade Federal do Ceará
Campus do Pici, Bloco 910, 60440-900. Fortaleza, CE, Brasil
rafael.andrade@ufc.br

Rommel Dias Saraiva
Universidade de Fortaleza – Núcleo de Ciência de Dados e Inteligência Artificial
Av. Washington Soares, 1321, 60811-905. Fortaleza, CE, Brasil
rommelds@unifor.br

RESUMO

Dado um digrafo ponderado $D = (V, A, c, d)$ com coloração de arcos e dois nós $s, t \in V$, desejamos encontrar um (s, t)-caminho p^* de custo mínimo em D, de sorte que a quantidade de cores em p^* seja limitada superiormente por um dado $k \in \mathbb{N}$. Introduzimos desigualdades válidas para o problema, uma formulação de programação linear inteira, e duas classes de instâncias difíceis. Os experimentos computacionais mostram que as novas desigualdades foram eficientes para resolver a maioria das instâncias em tempos computacionais bem inferiores quando comparados tanto com o modelo original quanto ao modelo alternativo. Mostram também que as classes de instâncias introduzidas são mais difíceis do que as utilizadas na literatura.

PALAVRAS CHAVE. Caminho mínimo. Caminho com restrição em número de cores dos arcos. Desigualdades válidas.

Otimização Combinatória. Programação Matemática.

ABSTRACT

Given a weighted edge-colored digraph $D = (V, A, c, d)$ and a pair of nodes $s, t \in V$, we want to find a (s, t)-path p^* of minimum cost in D, so that the amount of arc colors in p^* is limited above by a given $k \in \mathbb{N}$. We introduce valid inequalities for the problem, an integer linear programming formulation, and two new classes of difficult instances. The computational experiments show that the new inequalities are efficient to solve the majority of instances in much lower execution times when compared with both the original model and the alternative one. They also show that the introduced instances are more difficult than those from the literature.

KEYWORDS. Shortest path. Constrained path in the number of arc colors. Valid inequalities.

Combinatorial Optimization. Mathematical Programming.

https://proceedings.science/p/157375?lang=pt-br

1. Introdução

Problemas de otimização em grafos com coloração de arestas codificam informações qualitativas através de rótulos (ou cores), podendo estes representarem, por exemplo, diferentes meios de transporte em uma rede de fluxo. Dentre os existentes com essa característica na literatura, tem-se o problema da árvore geradora com quantidade mínima de cores [Chang e Shing-Jiuan, 1997; Saraiva et al., 2018], o problema da cobertura de ciclos coloridos [Silvestri et al., 2016] e o problema da floresta colorida [Carrabs et al., 2018], para citar apenas alguns.

Em se tratando, especificamente, de problemas de otimização para obtenção de um caminho com rotulação em arestas, o problema do caminho com quantidade mínima de cores (MCP, de *minimum colored path*) já foi estudado em diferentes áreas sob nomenclaturas distintas [Eiben e Kanj, 2020]. Kumar [2019], por exemplo, concebeu métodos heurísticos para o problema com base na seleção de cores mais frequentes no grafo de entrada. Um algoritmo aproximativo e um modelo exato também foram propostos. Grafos em camadas, de disco unitário e *road networks* foram observadas como instâncias difíceis para o problema. Já Granata et al. [2013] apresentaram duas formulações para o problema de fluxo máximo com quantidade mínima de cores. A primeira é derivada da descrição formal do problema, enquanto a segunda a incrementa por meio dos conceitos de rótulo e vértice necessários. Os autores também propuseram uma meta-heurística *Skewed Variable Neighborhood Search* para a geração de soluções viáveis, utilizando-se de rotinas de fluxo máximo como ponto de partida. Uma vasta revisão da literatura de grafos com coloração de arestas também é realizada. Em Carrabs et al. [2019] os autores investigaram o problema do caminho mais longo com cores ordenadas. Modelos de programação matemática baseados em fluxo e em *scheduling* foram detalhados, além da relação do problema com aplicações relevantes, como na área de ressonância magnética nuclear.

Neste trabalho, abordamos o problema do caminho mínimo k-rotulado (k-CSP, de *k-color shortest path problem*), definido da seguinte forma: seja um digrafo ponderado com coloração de arcos $D = (V, A, c, d)$, com conjunto de vértices V, conjunto de arcos A, função de coloração de arcos $c : A \to \mathbb{N}$, e função custo $d : A \to \mathbb{R}_+$. Além disso, considere $s, t \in V$ e um inteiro $k > 0$. O k-CSP consiste em encontrar um (s, t)-caminho com custo mínimo que utilize no máximo k cores distintas. A Figura 1 apresenta um exemplo de instância para o problema.

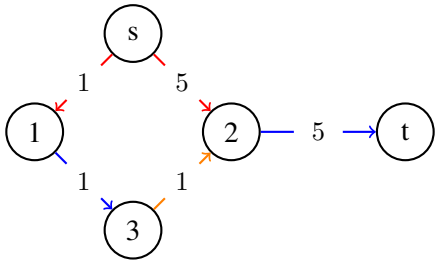


Figura 1: Ilustração do k-CSP. Supondo ser $k = 3$, é evidente que os caminhos $p_1 = \{(s, 1), (1, 3), (3, 2), (2, t)\}$ e $p_2 = \{(s, 2), (2, t)\}$ são viáveis para a instância, sendo p_1 a solução ótima com $d(p_1) = 8$. Entretanto, ao considerarmos $k = 2$ observamos que p_1 se torna inviável e a solução ótima é dada por p_2 com $d(p_2) = 10$.

O k-CSP foi recente e independentemente introduzido por Ferone et al. [2019] no contexto de redes confiáveis, assim como outros problemas de minimização no uso de cores [Shengli Yuan et al., 2005; Coudert et al., 2007], e por Dehouche [2020] no cenário de linhas de metrô, aparecendo como um subproblema do problema de encontrar o diâmetro restrito k-rotulado (*k-color constrained*

diameter) de um grafo, sendo essa a maior distância que um passageiro poderá percorrer usando um caminho mínimo com não mais que k linhas. Um modelo de programação linear inteira (PLI) também foi proposto por Ferone et al. [2019]. Em ambos os artigos que o introduziram, o k-CSP foi provado ser NP-Difícil, tendo Ferone et al. [2019] utilizado o fato de o problema de encontrar um (s,t)-caminho com no máximo k cores distintas ser um caso especial onde os pesos são nulos. Ainda, observamos que toda solução deste é uma solução viável para aquele, de tal forma que encontrar uma solução viável para o k-CSP é um problema NP-Completo.

Com base no modelo da literatura, propomos uma formulação de PLI equivalente fundamentada em uma inversão das desigualdades e variáveis de cores do modelo proposto por Ferone et al. [2019]. Apresentamos novas desigualdades válidas para o k-CSP baseadas no limite de arcos de mesma cor partindo ou incidindo em um vértice, e na seleção de cores na vizinhança da origem e do destino do caminho. Além disso, introduzimos duas novas classes de instâncias difíceis para o problema: a classe de digrafos em camadas, adotada por Kumar [2019] como instâncias difíceis para o MCP, e a classe de digrafos em camadas com arcos de salto.

A sequência deste trabalho está estruturada da seguinte maneira. A Seção 2 traz o modelo existente para o k-CSP e uma nova formulação para o problema, cujas algumas desigualdades válidas se encontram na Seção 3. A Seção 4 apresenta uma análise comparativa dos modelos através de experimentos computacionais. Conclusões e menções a trabalhos futuros são realizadas na Seção 5.

2. Modelos matemáticos

Esta seção apresenta formulações de PLI para o k-CSP. Preliminarmente, faz-se necessário introduzir algumas notações. Seja $C(D)$ o conjunto de cores do digrafo D, $A(p)$ (resp. $V(p)$) o conjunto de arcos (resp. vértices) em um caminho p, A_h o conjunto de arcos que possuem rótulo $h \in C(D)$ e $N^-(i)$ (resp. $N^+(i)$) as extremidades iniciais (resp. finais) dos arcos que incidem no (resp. partem do) vértice $i \in V$.

A começar pela formulação proposta por Ferone et al. [2019], seja $x_{ij} \in \{0,1\}$ variável binária que determina se o arco $(i,j) \in A$ está (valor 1) ou não (valor 0) na solução p, para todo $(i,j) \in A$. Além disso, seja $y_h \in \{0,1\}$ variável binária que expressa se uma cor h está (valor 1) ou não (valor 0) em p, para todo $h \in C(D)$. O modelo para o k-CSP é dado por:

$$\text{(FFP)} \quad \min \sum_{(i,j)\in A} d_{ij} x_{ij} \tag{1}$$

$$\text{s.a.:} \sum_{j\in N^-(i)} x_{ji} - \sum_{j\in N^+(i)} x_{ij} = \begin{cases} -1 & \text{se } i = s \\ +1 & \text{se } i = t \\ 0 & \text{caso contrário} \end{cases}, \quad \forall\, i \in V \tag{2}$$

$$x_{ij} \leq y_{c(ij)}, \quad \forall\, (i,j) \in A \tag{3}$$

$$\sum_{h\in C(D)} y_h \leq k \tag{4}$$

$$x_{ij} \in \{0,1\}, \quad \forall\, (i,j) \in A \tag{5}$$

$$y_h \in \{0,1\}, \quad \forall\, h \in C(D) \tag{6}$$

A função objetivo (1) minimiza o custo $d(p)$ de um (s,t)-caminho p. As restrições (2) são responsáveis pela conservação de fluxo unitário ao longo do caminho p. As restrições (3) impõem que uma cor estará presente na solução se ao menos um arco que a tem também estiver.

A restrição (4) limita superiormente a quantidade de cores de p ao valor k. As restrições (5) e (6) informam a natureza (domínio) das variáveis de decisão.

Apresentamos uma formulação alternativa (equivalente) de PLI ao considerarmos uma interpretação inversa das restrições (3) e (4). Enquanto a formulação (FFP) realiza a escolha dos arcos que se deseja utilizar na solução e, consequentemente, as respectivas cores, de forma análoga podemos definir tais restrições de modo a descartar rótulos que não devem aparecer na solução ótima. Dito isso, definimos variáveis binárias l_h de forma que $l_h = 1$ implica que a cor h não será utilizada na solução, e $l_h = 0$ caso contrário, para todo $h \in C(D)$. A formulação proposta é dada por:

$$\text{(INV)} \quad \min \sum_{(i,j)\in A} d_{ij}x_{ij}$$

$$\text{s.a.: (2), (5)}$$

$$x_{ij} \leq 1 - l_{c(ij)}, \quad \forall (i,j) \in A \tag{7}$$

$$\sum_{h\in C(D)} l_h \geq c(D) - k \tag{8}$$

$$l_h \in \{0,1\} \quad \forall h \in C(D) \tag{9}$$

As restrições (7) impõem que nenhum arco de cor $h \in C(D)$ será utilizado caso $l_h = 1$, enquanto a restrição (8) limita inferiormente a quantidade de cores que devem ser descartadas. A restrição (9) informa as variáveis de decisão.

3. Desigualdades válidas

Nesta seção, propomos desigualdades válidas para o k-CSP. Para isso, definimos $N_h^-(i)$ (resp. $N_h^+(i)$) como o conjunto de extremidades iniciais (resp. finais) dos arcos de cor h que incidem no (resp. partem do) vértice i, e $N_\emptyset(i)$ como o conjunto de vértices não alcançados pelo vértice i, para todo $i \in V$.

A primeira desigualdade válida para o problema baseia-se na observação do seguinte fato: visto que a desigualdade (3) implica que nenhum arco (i,j) poderá ser escolhido sem que sua cor $c(ij)$ também o seja, podemos afirmar que para todo vértice $i \neq t$ (resp. $i \neq s$) a quantidade de arcos de mesma cor que partem de (resp. incidem em) i é limitado superiormente pela variável do rótulo em questão. Em razão disso, obtemos as desigualdades (10) e (11).

$$\sum_{j\in N_h^+(i)} x_{ij} \leq y_h, \quad \forall i \in V, \quad \forall h \in C(D) \tag{10}$$

$$\sum_{j\in N_h^-(i)} x_{ji} \leq y_h, \quad \forall i \in V, \quad \forall h \in C(D) \tag{11}$$

Agora, considere dois vértices $u, v \in V$ tal que não exista um (u,v)-caminho ou (v,u)-caminho em D. Considere o grafo suporte $\bar{G} = (\bar{V}, \bar{A})$ de uma solução relaxada de um modelo do problema com respeito às variáveis x em que $u, v \in \bar{V}$. Veja que u, v estão em dois caminhos distintos de s a t em $\bar{G}$. Logo, deve haver algum vértice $w \in \bar{V}$ tal que existe um (w,v)-caminho e um (w,u)-caminho em $\bar{G}$. Consequentemente, podemos generalizar as desigualdades (10) e (11) de modo a incluir os casos de vértices não alcançáveis, resultando nas desigualdades (12)–(15).

$$\sum_{j \in N_h^+(v)} x_{vj} + \sum_{i \in N_h^+(u)} x_{ui} \leq y_h, \quad \forall v \in V, \quad \forall u \in N_\emptyset(v), \quad \forall h \in C(D) \tag{12}$$

$$\sum_{j \in N_h^+(v)} x_{vj} + \sum_{i \in N_h^-(u)} x_{iu} \leq y_h, \quad \forall v \in V, \quad \forall u \in N_\emptyset(v), \quad \forall h \in C(D) \tag{13}$$

$$\sum_{j \in N_h^-(v)} x_{jv} + \sum_{i \in N_h^+(u)} x_{ui} \leq y_h, \quad \forall v \in V, \quad \forall u \in N_\emptyset(v), \quad \forall h \in C(D) \tag{14}$$

$$\sum_{j \in N_h^-(v)} x_{jv} + \sum_{i \in N_h^-(u)} x_{iu} \leq y_h, \quad \forall v \in V, \quad \forall u \in N_\emptyset(v), \quad \forall h \in C(D) \tag{15}$$

No que tange a restrição (12) (resp. (15)), essa desigualdade nos diz que considerando dois nós $v \in V$ e $u \in V$ distintos não alcançáveis, a quantidade de arcos de mesma cor que partem de (resp. incidem em) ambos os vértices é limitado superiormente pela variável de decisão da cor. De maneira semelhante, a desigualdade (13) (resp. (14)) limita superiormente a quantidade de arcos que saem de (resp. incidem em) v e incidem em (resp. partem de) u que possuem a mesma cor.

Por fim, ao observarmos que toda solução viável para o k-CSP deve possuir uma cor na vizinhança de arcos que partem de s (resp. incidem em t), derivamos a desigualdade (16) (resp. (17)), que também é válida para o problema.

$$\sum_{j \in N^+(s)} y_{c(sj)} \geq 1 \tag{16}$$

$$\sum_{j \in N^-(t)} y_{c(jt)} \geq 1 \tag{17}$$

4. Resultados computacionais

Experimentos computacionais foram realizados para avaliar a qualidade das contribuições propostas. Para isso, implementamos os modelos matemáticos em Julia 1.7.2 e recorremos às bibliotecas CPLEX.jl e JuMP.jl para uso do *solver* CPLEX 22.1. Os experimentos foram executados em um PC Intel Core i7-3770, 8 $\times$ 3.40 GHz, 16 GB RAM DDR3 com Ubuntu 20.04 LTS 64 bits. Para cada teste, utilizamos um limite de tempo de 700 segundos.

Nos experimentos, avaliamos a qualidade da relaxação linear dos modelos e o respectivo tempo de execução, assim como o tempo de processamento até a convergência para a solução ótima inteira do algoritmo de *branch-and-bound* e a contagem de nós associada à árvore. Denotamos por (FFP+) o modelo (FFP) acrescido das restrições (12)–(17).

Como instâncias, utilizamos digrafos em camadas por serem de difícil resolução em problemas de caminho com rotulação de arcos [Kumar, 2019]. Um digrafo em camadas é organizado em w colunas de r nós. Seja v_{ij} o i-ésimo nó da j-ésima coluna; em toda coluna, temos $(v_{ij}, v_{bj+1}) \in A$, para $j = 1, \ldots, w$ e $i, b = 1, \ldots, r$. Ainda, criamos arcos (s, v_{i1}) para todo vértice i na primeira coluna e (v_{iw}, t) para todo nó i na w-ésima coluna. Nos experimentos realizados, consideramos digrafos em camadas com $w = 15$, $r = 10$, e quantidade de cores equivalente a 70% da quantidade de arcos. Além disso, a escolha do valor de k se dá pela solução do problema do caminho com quantidade mínima de cores. A geração das cores e dos pesos dos arcos foi uniforme como explicado adiante.

https://proceedings.science/p/157375?lang=pt-br

Denotamos instâncias de digrafos em camadas na Tabela 1 com o prefixo LG. Também consideramos uma modificação dos digrafos em camadas que utiliza arcos de salto. Um arco $(i, j) \in A$ é dito ser de salto se $i \in V$ e $j \in V$ não se encontram em camadas consecutivas. O peso de um arco de salto é gerado uniformemente a partir do intervalo inteiro $[d_{max}, d_{max} + d_{offset}]$, onde d_{max} é o maior peso observado no digrafo D e d_{offset} é um valor suficientemente grande (consideramos o valor 400). Multiplicamos então o valor gerado pela quantidade de camadas entre i e j. Esse grupo de instâncias foi criado visando facilitar a obtenção de uma solução viável, mas por um custo maior que percorrer as camadas em D. As instâncias de digrafos em camadas que implementam arcos de salto serão denotadas na Tabela 1 pelo prefixo JLG. Quanto à Tabela 2, são utilizadas as instâncias apresentadas por Ferone et al. [2019] referentes aos grupos G_1, G_3, R_1 e R_2, dentre os quais os dois primeiros são grafos *grid* com dimensões 100×100 e 100×200, com 5.940 e 11.910 cores respectivamente, enquanto os dois últimos são classes de grafos aleatórios com 75.000 nós e 750.000 arcos, com 112.500 e 150.000 cores respectivamente.

As Tabelas 1 e 2 possuem um total de 30 e 39 instâncias, respectivamente. Nelas, *opt* indica o valor ótimo da instância $inst$, z indica o valor da relaxação linear, cpu_z se trata do tempo para resolução da relaxação, *cpu* é o tempo para resolução do modelo inteiro, e *bb* é a quantidade de subproblemas resolvidos na árvore de *branch-and-bound*.

No que concerne as novas instâncias propostas e os resultados da Tabela 1, o modelo (FFP+) obteve a solução ótima inteira no menor tempo em 96,66% dos casos (29 instâncias), enquanto o (INV) obteve o menor tempo em apenas uma. Para esse grupo de instâncias, o modelo (FFP+) obteve uma relaxação linear mais forte que os demais modelos. O modelo (FFP+) foi 78,53% e 79,33% mais rápido que (FFP) e (INV), respectivamente. O número de subproblemas *bb* de (FFP+) foi cerca de 4 vezes menor que o de (FFP), em média. O que relatamos acima mostra o potencial das desigualdades válidas concebidas. Salientamos que (INV) obteve o resultado no menor tempo em 40% dos casos (12 instâncias) quando comparado a (FFP).

Tabela 1: Experimentos computacionais em grafos em camadas e grafos em camadas com saltos.

Instances		**(FFP)**				**(FFP+)**				**(INV)**			
inst	*opt*	z	cpu_z	*cpu*	*bb*	z	cpu_z	*cpu*	*bb*	z	cpu_z	*cpu*	*bb*
LG15x10-01	709	361.33	0.07	55.85	5818	394.57	0.18	**23.27**	2007	361.33	0.08	97.24	7872
LG15x10-02	528	309.26	0.05	66.79	8559	347.39	0.16	**18.80**	1275	309.26	0.06	54.44	5817
LG15x10-03	457	311.92	0.05	93.14	9857	332.92	0.13	**9.31**	264	311.92	0.05	53.81	5482
LG15x10-04	671	321.47	0.07	88.20	4695	361.47	0.19	**26.36**	1328	321.47	0.07	90.08	5290
LG15x10-05	533	378.88	0.07	50.74	2888	406.76	0.17	**9.26**	216	378.88	0.07	18.57	989
LG15x10-06	583	325.93	0.05	177.56	26160	354.47	0.14	**36.94**	6234	325.93	0.06	178.80	23537
LG15x10-07	636	331.09	0.06	242.39	26512	362.32	0.13	**26.90**	3905	331.09	0.05	319.54	34138
LG15x10-08	670	361.56	0.09	70.50	5284	408.29	0.21	**24.27**	1075	361.56	0.06	106.74	7073
LG15x10-09	683	319.84	0.07	142.75	8572	354.50	0.23	**30.50**	1522	319.84	0.06	108.32	7200
LG15x10-10	645	329.93	0.05	247.38	29696	359.18	0.05	**21.83**	9425	329.93	0.05	255.86	28683
LG15x10-11	693	392.09	0.08	127.44	7606	422.69	0.07	**13.08**	2061	392.09	0.06	106.56	6388
LG15x10-12	589	314.93	0.06	273.11	31393	333.97	0.05	**23.55**	7280	314.93	0.05	196.99	25196
LG15x10-13	420	323.58	0.05	15.94	1221	345.37	0.07	**1.94**	0	323.58	0.05	15.28	1043
LG15x10-14	474	288.13	0.06	109.13	11644	322.23	0.08	**6.38**	897	288.13	0.06	84.51	8869
LG15x10-15	507	289.88	0.04	101.72	15574	316.63	0.07	**15.51**	5195	289.88	0.04	106.21	12721
LG15x10-16	423	273.78	0.05	16.93	1023	291.98	0.06	**6.30**	417	273.78	0.05	41.83	5894
LG15x10-17	610	323.24	0.05	225.97	26936	353.71	0.06	**24.97**	6741	323.24	0.05	154.25	18236
LG15x10-18	605	296.13	0.05	285.09	38059	320.72	0.07	**23.09**	8329	296.13	0.06	190.18	23410
LG15x10-19	596	298.76	0.07	215.58	29090	323.38	0.10	**29.33**	5564	298.76	0.07	157.02	19880
LG15x10-20	516	273.46	0.05	87.24	10881	301.58	0.13	**21.10**	2558	273.46	0.06	105.23	15695
JLG15x10-01	555	301.28	0.06	61.37	6155	334.55	0.20	**38.52**	3471	301.28	0.06	71.56	6935
JLG15x10-02	547	297.27	0.05	109.35	14139	328.03	0.20	**44.45**	3950	297.27	0.06	152.80	14921
JLG15x10-03	851	350.45	0.07	260.38	20647	400.42	0.32	**42.35**	2359	350.45	0.06	289.97	25771
JLG15x10-04	464	309.87	0.06	20.98	1464	331.93	0.28	13.30	287	309.87	0.06	**11.05**	580
JLG15x10-05	582	314.49	0.08	176.68	20232	354.19	0.23	**46.68**	4200	314.49	0.07	184.11	21797
JLG15x10-06	661	305.39	0.07	400.96	47125	343.24	0.26	**154.72**	16475	305.39	0.06	559.87	60129
JLG15x10-07	792	351.37	0.08	177.64	13495	388.47	0.29	**54.84**	3296	351.37	0.08	180.80	14675
JLG15x10-08	493	288.99	0.07	99.09	11195	316.11	0.24	**24.18**	1298	288.99	0.06	128.00	16819
JLG15x10-09	536	300.75	0.05	87.02	9194	347.06	0.24	**38.64**	3326	300.75	0.05	108.57	11487
JLG15x10-10	579	309.28	0.08	120.88	14103	326.61	0.24	**53.05**	6591	309.28	0.07	240.86	25047
Média		-	0.06	140.26	15307.23	-	0.16	30.11	3718.20	-	0.06	145.64	15385.80
Mediana		-	0.06	109.24	11419.50	-	0.16	24.22	2927.00	-	0.06	108.45	13698.00
Máximo		-	0.09	400.96	47125.00	-	0.32	154.72	16475.00	-	0.08	559.87	60129.00
Mínimo		-	0.04	15.94	1023.00	-	0.05	1.94	0.00	-	0.04	11.05	580.00

https://proceedings.science/p/157375?lang=pt-br

Tabela 2: Experimentos computacionais nas instâncias da literatura.

Instances		**(FFP)**			**(FFP+)**			**(INV)**		
inst	*opt*	z	cpu_z	*cpu*	z	cpu_z	*cpu*	z	cpu_z	*cpu*
G1-01	6131	6120	5.76	**37.78**	6120	22.50	60.19	6120	26.60	58.08
G1-02	6233	6233	4.13	9.28	6233	16.70	**1.45**	6233	20.41	8.80
G1-03	6336	6327	10.00	149.97	6327	43.62	**36.28**	6327	42.48	35.82
G1-04	6200	6195.50	5.05	23.28	6195.50	30.83	**6.52**	6195.50	19.06	25.44
G1-05	6079	6077.16	20.49	**41.48**	6077.16	152.40	52.46	6077.16	83.00	42.95
G1-06	6109	6106.33	7.64	**19.92**	6106.33	42.03	50.02	6106.33	36.71	24.14
G1-07	6197	6197	5.68	10.52	6197	21.91	**2.79**	6197	20.70	15.68
G1-08	6181	6181	6.36	15.10	6181	47.58	**5.99**	6181	22.73	17.23
G1-09	6375	6375	15.95	41.77	6375	106.65	**10.32**	6375	170.63	50.53
G1-10	6193	6193	5.86	11.19	6193	48.19	**7.44**	6193	38.49	15.22
G3-01	9808	9805.67	46.60	94.52	9805.67	342.54	**38.47**	9805.67	256.50	104.83
G3-02	9652	9651.67	87.98	**190.01**	0	701.65	704.27	0	703.04	216.99
G3-03	9448	9446.50	33.12	58.31	9446.50	217.74	**31.41**	9446.50	192.79	89.92
G3-04	10149	10147.38	45.03	148.62	10147.38	357.13	**75.96**	10147.38	245.03	167.37
G3-05	9654	9653.33	35.14	80.68	9653.33	264.01	**51.07**	9653.33	215.43	98.85
G3-06	9535	9535	36.19	66.17	9535	251.50	**17.38**	9535	193.50	70.14
G3-07	9455	9453.50	60.54	159.11	9453.50	425.30	**36.92**	9453.50	421.83	155.42
G3-08	9424	9424	69.60	155.48	9424	489.62	**24.00**	9424	441.83	162.24
G3-09	9786	9783	140.65	**444.00**	0	700.94	702.91	0	701.62	382.32
G3-10	9793	9793	120.06	**186.94**	0	701.16	702.34	0	700.50	335.85
R1-01	201	201	11.74	62.60	201	42.28	**34.08**	201	42.50	56.50
R1-02	253	252	9.97	60.75	252	20.21	**36.14**	252	21.65	56.37
R1-03	333	333	28.47	81.17	333	64.57	**46.85**	333	133.48	68.10
R1-04	236	236	14.21	80.86	236.00	56.28	**46.30**	236	48.73	71.59
R1-05	253	238.50	15.91	83.69	239	73.66	**52.27**	238.50	43.07	121.13
R1-06	255	255	11.20	57.44	255	43.82	**34.87**	255	21.92	68.11
R1-07	401	401	18.07	68.52	401	68.28	**43.98**	401	80.36	64.17
R1-08	426	426	92.77	150.59	426	293.68	**27.93**	426	113.36	198.26
R1-09	242	240.50	9.79	86.42	240.50	20.40	**68.92**	240.50	50.57	79.81
R1-10	152	152	9.55	59.01	152	17.37	**31.62**	152	20.59	54.70
R2-01	289	289	13.59	65.23	289	59.90	**42.73**	289	50.10	58.93
R2-02	246	246	10.81	59.28	246	27.89	**35.06**	245	25.10	66.59
R2-03	198	192.67	10.92	62.37	192.67	22.61	**30.90**	192.67	22.67	58.36
R2-04	196	190	11.72	64.93	190	43.91	**35.40**	190	44.79	60.96
R2-05	246	246	16.19	65.63	246	46.93	**30.60**	246	52.77	59.82
R2-06	238	238	17.07	64.10	238	46.44	**34.21**	238	53.05	61.75
R2-07	219	205	12.40	65.26	205	45.53	**36.31**	205	41.60	71.54
R2-08	231	231	12.82	60.90	231	46.55	**33.07**	231	39.38	60.87
R2-09	245	244	10.43	61.41	244	20.79	**43.33**	244	22.44	58.30
Média		-	33.95	98.51	-	185.36	104.56	-	167.96	106.50
Mediana		-	16.05	65.90	-	58.09	36.23	-	51.67	69.13
Máximo		-	140.65	444.00	-	701.65	704.27	-	703.04	382.32
Mínimo		-	5.86	11.19	-	17.37	7.44	-	20.59	15.22

Para o grupo de instâncias de Ferone et al. [2019] na Tabela 2, o modelo (FFP+) obteve o menor tempo de resolução do problema inteiro em 84,62% dos casos (33 instâncias), enquanto (FFP) obteve o menor tempo em 6 instâncias. Note-se que o modelo (FFP+) e o modelo (INV) não obtiveram soluções ótimas em três instâncias. Observe que a coluna *bb* referente à quantidade de subproblemas no *branch-and-bound* é omissa, visto que a maioria das instâncias foi resolvida na raiz. Ao desconsiderarmos as instâncias G3-02, G3-09, e G3-10, vemos que o modelo (FFP+) foi 51,87% e 54,55% mais rápido que (FFP) e (INV), respectivamente.

Por fim, destacamos, quanto ao tempo para obtenção da solução ótima inteira, que a média das instâncias na Tabela 1 é superior àquela da Tabela 2 no modelo (FFP), e nos modelos (FFP+) e (INV) ao desconsiderarmos as três instâncias supracitadas. Isso é um forte indício que as instâncias de digrafos em camadas são mais difíceis do que as que foram propostas na literatura por Ferone et al. [2019]. Além disso, revelamos que 40% dos grafos *grid* (8 instâncias) e 63,16% dos grafos aleatórios (12 instâncias) possuem valor da relaxação linear igual ao custo da solução ótima inteira.

5. Conclusões

Neste trabalho, propusemos desigualdades válidas para o problema do caminho mínimo k-rotulado (k-CSP), além de uma nova formulação de PLI. Os experimentos computacionais mos-

tram que o uso das desigualdades válidas tornam o modelo proposto por Ferone et al. [2019] mais eficiente.

Para comprovar a qualidade das desigualdades, abordamos dois grupos de instâncias: aquelas fornecidas por Ferone et al. [2019] e digrafos em camadas, sugeridos por Kumar [2019] como instâncias desafiadoras para problemas de caminho com rótulo em arcos. Isso, de fato, foi evidenciado nos experimentos através do tempo de resolução dos modelos matemáticos. A partir disso, concebemos uma nova classe de instâncias desafiadoras: digrafos em camadas com arcos de salto. Isso também ficou evidente ao observamos a quantidade de subproblemas resolvidos pela árvore de *branch-and-bound*, cerca de 4 vezes menor (média) com a introdução das desigualdades.

Já o novo modelo proposto, quando comparado com o já existente na literatura, obteve melhores resultados em um total de 12 digrafos em camadas.

Pesquisas futuras serão direcionadas para a elaboração de uma relaxação lagrangiana para o k-CSP, além de métodos heurísticos para obtenção de soluções viáveis em tempo razoável, e um estudo da variação do limite superior k e da densidade de cores introduzidas para o aumento na dificuldade das instâncias. Acreditamos que a característica do problema também abre espaço para métodos de decomposição.

Agradecimentos

Este estudo foi financiado em partes pela Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES), pelo Conselho Nacional de Desenvolvimento Científico e Tecnológico (Proc. CNPq 408084/2018-9 e 303988/2021-5), e pela FUNCAP (Proc. BMD-0008-01460.01.15/21).

Referências

Carrabs, F., Cerrone, C., Cerulli, R., e Silvestri, S. (2018). The rainbow spanning forest problem. *Soft Computing*, 22(8):2765–2776.

Carrabs, F., Cerulli, R., Felici, G., e Singh, G. (2019). Exact approaches for the orderly colored longest path problem: Performance comparison. *Computers & Operations Research*, 101:275–284.

Chang, R.-S. e Shing-Jiuan, L. (1997). The minimum labeling spanning trees. *Information Processing Letters*, 63(5):277–282.

Coudert, D., Datta, P., Perennes, S., Rivano, H., e Voge, M.-E. (2007). Shared risk resource group complexity and approximability issues. *Parallel Processing Letters*, 17(02):169–184.

Dehouche, N. (2020). The k-interchange-constrained diameter of a transit network: a connectedness indicator that accounts for travel convenience. *Transportation Letters*, 12(3):197–201.

Eiben, E. e Kanj, I. (2020). A Colored Path Problem and Its Applications. *ACM Transactions on Algorithms*, 16(4):1–48.

Ferone, D., Festa, P., e Pastore, T. (2019). The k-Color Shortest Path Problem. In Paolucci, M., Sciomachen, A., e Uberti, P., editors, *Advances in Optimization and Decision Science for Society, Services and Enterprises*, volume 3, p. 367–376. Springer International Publishing, Cham.

Granata, D., Cerulli, R., Scutellà, M. G., e Raiconi, A. (2013). Maximum Flow Problems and an NP-Complete Variant on Edge-Labeled Graphs. In Pardalos, P. M., Du, D.-Z., e Graham, R. L., editors, *Handbook of Combinatorial Optimization*, p. 1913–1948. Springer New York, New York, NY.

Kumar, N. (2019). Computing a Minimum Color Path in Edge-Colored Graphs. In Kotsireas, I., Pardalos, P., Parsopoulos, K. E., Souravlias, D., e Tsokas, A., editors, *Analysis of Experimental Algorithms*, volume 11544, p. 35–50. Springer International Publishing, Cham.

Saraiva, R., Andrade, R., e Adasme, P. (2018). Mathematical programming techniques applied to the label-constrained minimum spanning tree problem. *50º Simpósio Brasileiro de Pesquisa Operacional*.

Shengli Yuan, Varma, S., e Jue, J. (2005). Minimum-color path problems for reliability in mesh networks. In *Proceedings IEEE 24th Annual Joint Conference of the IEEE Computer and Communications Societies.*, volume 4, p. 2658–2669, Miami, FL, USA. IEEE.

Silvestri, S., Laporte, G., e Cerulli, R. (2016). The rainbow cycle cover problem. *Networks*, 68(4): 260–270.

https://proceedings.science/p/157375?lang=pt-br