



UNIVERSIDADE FEDERAL DO CEARÁ
CAMPUS QUIXADÁ
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

LUCAS DA SILVA COSTA

UMA NOVA HEURÍSTICA DE COLORAÇÃO GULOSA SEQUENCIAL

QUIXADÁ

2018

LUCAS DA SILVA COSTA

UMA NOVA HEURÍSTICA DE COLORAÇÃO GULOSA SEQUENCIAL

Monografia apresentada no curso de Ciência da Computação da Universidade Federal do Ceará, como requisito parcial à obtenção do título de bacharel em Ciência da Computação. Área de concentração: Computação.

Orientador: Prof. Dr. Wladimir Araujo Tavares

QUIXADÁ

2018

Dados Internacionais de Catalogação na Publicação
Universidade Federal do Ceará
Biblioteca Universitária
Gerada automaticamente pelo módulo Catalog, mediante os dados fornecidos pelo(a) autor(a)

C873n Costa, Lucas da Silva.

Uma nova heurística de coloração gulosa sequencial / Lucas da Silva Costa. – 2018.
23 f.

Trabalho de Conclusão de Curso (graduação) – Universidade Federal do Ceará, Campus de Quixadá,
Curso de Redes de Computadores, Quixadá, 2018.
Orientação: Prof. Dr. Wladimir Araujo Tavares.

1. Coloração de Grafos. 2. Heurística. I. Título.

CDD 004.6

LUCAS DA SILVA COSTA

UMA NOVA HEURÍSTICA DE COLORAÇÃO GULOSA SEQUENCIAL

Monografia apresentada no curso de Ciência da Computação da Universidade Federal do Ceará, como requisito parcial à obtenção do título de bacharel em Ciência da Computação. Área de concentração: Computação.

Aprovada em: ___/___/_____.

BANCA EXAMINADORA

Prof. Dr. Wladimir Araujo Tavares (Orientador)
Universidade Federal do Ceará – UFC

Prof. Dr. Arthur Rodrigues Araruna
Universidade Federal do Ceará - UFC

Prof. Me. Francisco Erivelton Fernandes de Aragão
Universidade Federal do Ceará - UFC

RESUMO

Nesta monografia, apresentamos uma nova heurística de coloração gulosa sequencial com uma nova estratégia, proposta pelos autores. A heurística obtém uma nova ordem de coloração a partir de uma coloração já existente. Experimentos computacionais mostram que a heurística proposta apresenta uma diminuição na quantidade de cores necessária para se colorir um grafo, em comparação com outras heurísticas de coloração gulosa sequencial existentes na literatura.

Palavras-chave: Coloração de Grafos. Heurística.

ABSTRACT

In this monograph, we present a new heuristic of sequential greedy coloration with a new strategy, proposed by the authors. The heuristic obtains a new order of coloration from an existing coloration. Computational experiments show that the proposed heuristic presents a decrease in the amount of color necessary to color a graph, in comparison to other sequential greedy color heuristics in the literature.

Keywords: Coloring of Graphs. Heuristic.

SUMÁRIO

1	INTRODUÇÃO	6
2	TRABALHOS RELACIONADOS	8
3	FUNDAMENTAÇÃO TEÓRICA	10
3.1	COLORAÇÃO DE GRAFOS	10
3.2	HEURÍSTICAS DE COLORAÇÃO	10
3.3	CRITÉRIOS DE ORDENAÇÃO DOS VÉRTICES	11
3.3.1	<i>LF</i>	12
3.3.2	<i>SL</i>	12
3.3.3	<i>LFI</i>	13
3.3.4	<i>SLI</i>	13
3.4	DSATUR	13
3.5	CG	14
4	PROCEDIMENTOS METODOLÓGICOS	15
4.1	ALGORITMOS <i>LF</i> , <i>SL</i> , <i>LFI</i> , <i>SLI</i> , <i>DSATUR</i> E <i>CG</i>	15
4.2	ESTRATÉGIA DA NOVA HEURÍSTICA DE COLORAÇÃO	15
4.3	COMPARAÇÃO ENTRE OS MÉTODOS	15
4.4	ORDEM OBTIDA PELA NOVA HEURÍSTICA	16
5	RESULTADOS	17
6	DISCUSSÃO	21
7	CONSIDERAÇÕES FINAIS	22
	REFERÊNCIAS	23

1 INTRODUÇÃO

O problema de coloração de grafos consiste em colorir os vértices de um grafo de forma que os vértices adjacentes tenham diferentes cores e que a coloração utilize a menor quantidade de cores possível. A coloração de grafos possui diversas aplicações. No geral, problemas que separem objetos em grupos não conflitantes, podem ser modelados como um problema de coloração de grafos.

Um dos problemas que utilizam a coloração de grafos é a alocação de registradores, estudada em compiladores, que consiste em designar variáveis para os registradores. Os registradores são circuitos digitais capazes de armazenar e deslocar informações binárias, e são tipicamente usados como um dispositivo de armazenamento temporário. O processo de mapeamento dos registradores é importante, pois permite a otimização na geração de código, afetando diretamente o desempenho do programa gerado (SILVA, 2011). Neste problema, as variáveis representam os vértices do grafo, as dependências entre as variáveis representam as arestas e os registradores representam as cores disponíveis para o problema de coloração de grafos.

A coloração de grafos é um dos problemas de otimização combinatória mais estudados. O problema representa um grande desafio teórico uma vez que o problema pertence a classe dos problemas *NP-Completo*, onde, nenhum algoritmo polinomial é conhecido para os problemas desta classe (CORMEN et al., 2009). Geralmente, heurísticas são empregadas para encontrar soluções sub-ótimas para o problema.

Este trabalho tem como objetivo criar uma nova heurística de coloração sequencial. A coloração sequencial consiste em colorir os vértices seguindo uma determinada sequência. Algumas heurísticas de coloração sequenciais como a LF (*largest-first*) (WELSH; POWELL, 1967), SL (*smallest-last*) (MATULA; BECK, 1983), LFI (*largest-first interchange*) e SLI (*smallest-last interchange*) (RADIN, 2000) serão apresentadas. O método heurístico DSATUR, apresentado inicialmente por (BRÉLAZ, 1979) é um algoritmo, em que a ordem dos vértices a serem coloridos é obtida de forma dinâmica. A nova heurística é um trabalho independente ao trabalho realizado por Culberson e Luo (1996). A criação dessa nova heurística propõe ser mais eficiente na utilização de cores necessárias para colorir o grafo, se comparado as heurísticas de coloração gulosas sequenciais citadas anteriormente. Com isso, este trabalho visa ajudar a comunidade acadêmica-científica que estuda ou trabalha com algoritmos na área de coloração de grafos.

Na Seção 2, são mostrados os trabalhos relacionados. Na Seção 3, apresentamos a fundamentação teórica. Na Seção 4, apresentamos os procedimentos metodológicos. Na seção 5, apresentamos os resultados computacionais. Na seção 6, apresentaremos as conclusões e trabalhos futuros.

2 TRABALHOS RELACIONADOS

Nessa Seção serão apresentados alguns trabalhos que serviram de base para a construção da nova heurística de coloração gulosa sequencial.

Welsh e Powell (1967) observou que os vértices de maior grau eram mais difíceis de se colorir, se comparado aos vértices de menor grau. Os vértices de menor grau apresentam uma maior flexibilidade em sua coloração, devido a sua simplicidade. Com base nisso foi desenvolvido um método heurístico de coloração gulosa sequencial chamado LF (*largest-first*) que colore os vértices na ordem de maior grau.

Matula e Beck (1983) propôs um outro método, baseado na observação feita anteriormente por Welsh e Powell (1967), chamado de SL (*smallest-last*) que realiza um refinamento na sequência obtida pelo método LF e inverte esta ordem.

Um procedimento interessante criado por Matula e Beck (1983), consiste na coloração por troca (*interchange*). Esse procedimento tenta trocar as cores nos subgrafos bipartido do grafo, afim de atribuir uma cor, que antes não era permitida. Essas trocas tem como objetivo evitar a criação de uma nova cor, diminuindo a quantidade de cores necessária pra colorir o grafo. Os métodos anteriores foram modificados utilizando esse procedimento, dando origem aos métodos SLI (*smallest-last interchange*) e LFI (*largest-first interchange*).

Brélaz (1979) criou um método heurístico de coloração chamado DSATUR que escolhe os vértices a serem coloridos de forma dinâmica. Inicialmente o vértice com o maior grau de saturação é escolhido e então colorido. O grau de saturação de um vértice é determinado pelo grau do mesmo, logo, quanto maior o grau de um vértice, maior será o grau de saturação. Desta forma o vértice de maior grau é escolhido, como o vértice com o maior grau de saturação. Depois dessa coloração inicial o próximo vértice com maior grau de saturação é escolhido. Se houver vértices com mesmo grau de saturação, o método opta por qualquer um de grau máximo pertencente ao sub-grafo ainda não colorido e então atribuí ao vértice selecionado a cor de menor índice disponível. Ele mostra ótimos resultados se comparado aos métodos heurísticos de coloração gulosa sequenciais existentes.

Os trabalhos citados anteriormente tem com objetivo procurar a melhor ordem possível, sempre pensando no uso da menor quantidade possível de cores para colorir um grafo. No geral os métodos heurísticos gulosos sequenciais LF, SL, LFI, SLI e DSATUR, mostram resultados bons para instâncias de grafos com poucos vértices e menos densos (RADIN, 2000).

Este trabalho propõe uma nova heurística de coloração gulosa sequencial denominada

CG (*color group*) que utiliza as ordens obtidas pelos métodos LF, SL, LFI, SLI e DSATUR.

A Tabela 1 apresenta de forma resumida a relação deste trabalho com os trabalhos apresentados nesta seção. A métrica comparada, se refere a complexidade computacional ao descobrir uma ordem inicial, na escolha do vértice da ordem obtida e da atualização da ordem.

Tabela 1 – Trabalhos Relacionados

TRABALHOS	Ordem Inicial	Escolha do Vértice	Atualização da Ordem
Welsh e Powell (1967)	$O(n \log n)$	$O(1)$	$O(1)$
Matula e Beck (1983)	$O(m)$	$O(1)$	$O(1)$
Brélaz (1979)	$O(1)$	$O(n)$	$O(n)$
CG	$O(n^2)$	$O(1)$	$O(1)$

Fonte – Produzido pelo próprio autor

3 FUNDAMENTAÇÃO TEÓRICA

Nessa Seção o problema de coloração de grafos será apresentado, bem como as heurísticas de coloração gulosas sequenciais LF, SL, LFI e SLI. O método heurístico DSATUR é mostrado. Além disso, a nova heurística de coloração, denominada CG é apresentada.

3.1 COLORAÇÃO DE GRAFOS

Um grafo simples G é um par ordenado $G = (V, E)$, composto por um conjunto finito $V = \{1, 2, \dots, n\}$, cujos elementos são chamados de vértices, e por um conjunto de pares não ordenados $E \subseteq \{\{u, v\} : u, v \in V, u \neq v\}$, cujos elementos são chamados de arestas. Denotaremos por $n = |V|$ e $m = |E|$. Os vértices $u, v \in V$ são chamados adjacentes (ou vizinhos) se $\{u, v\} \in E$ e não adjacentes se $\{u, v\} \notin E$. As arestas $e, f \in E$ são ditas serem adjacentes se $e \cap f \neq \emptyset$ e não adjacente se $e \cap f = \emptyset$. O grau de um vértice v em um grafo G , denotado por $deg(v)$, corresponde ao número de vizinhos de v no grafo G , isto é $|\{u : \{v, u\} \in E\}|$. O grau máximo de um vértice no grafo G é denotado por $\Delta(G)$ e o grau mínimo é denotado por $\delta(G)$. A densidade do grafo G , denotada por $g(G)$, é dada por $g(G) = 2m/(n(n-1))$.

Uma coloração de vértice válida de G é uma função $c : V \rightarrow \mathbb{N}$ em que qualquer dois vértices vizinhos u e v são atribuídas cores diferentes, isto é $\{u, v\} \in E \Rightarrow c(u) \neq c(v)$. A função c é chamada de função de coloração. Um grafo G para o qual existe uma coloração de vértice que requer k cores é chamado k -colorível, enquanto essa coloração é chamada de uma k -coloração. O menor número k para o qual existe uma k -coloração do grafo G é chamada de número cromático do grafo G e denotada por $\chi(G)$. O problema de coloração de um grafo G é encontrar o número mínimo de cores, $\chi(G)$, tal que G admite uma $\chi(G)$ -coloração.

3.2 HEURÍSTICAS DE COLORAÇÃO

O problema de coloração de grafos é um problema que pertence a classe de problemas NP -difícil. Logo, existe a necessidade do desenvolvimento de heurísticas de coloração que permitam encontrar soluções sub-ótimas em tempo polinomial. Entre as heurísticas de coloração conhecidas, vamos destacar as heurísticas de coloração gulosas sequenciais.

Dado um grafo G e uma sequência de vértices $K = (v_1, v_2, \dots, v_n)$ usaremos o termo coloração gulosa sequencial para descrever o seguinte procedimento de atribuição de cores:

Algoritmo 1: Coloração-Gulosa(G, K)

```

início
  | para  $v \leftarrow v_1$  até  $v_n$  faça
  |   | atribua a menor cor possível ao vértice  $v$ 
  |   fim
fim

```

Fonte – Produzido pelo próprio autor

Observe que um algoritmo de coloração gulosa sequencial de G é dividido nos dois seguintes estágios:

- (1) Determinar uma sequência de coloração K dos vértices de G .
- (2) Aplicar o algoritmo Coloração-Gulosa(G, K).

Coloração com troca é um método relativamente simples que aumenta a eficiência dos algoritmos de coloração sequencial. Ele é baseado na seguinte observação: se a próxima escolha gulosa necessitar de uma nova cor para colorir o vértice v_k , então pode ser possível trocar as cores em um dos subgrafos bipartidos de G e assim permitir a atribuição de uma cor até então não permitida ao vértice v_k .

Algoritmo 2: Coloração-com-Troca(G, K)

```

início
  | para  $v \leftarrow v_1$  até  $v_n$  faça
  |   | se  $v$  necessitar de uma nova cor então
  |   |   | tente trocar as cores em um dos subgrafos bipartidos de  $G$  para liberar uma cor para  $v$ 
  |   |   fim
  |   | atribua a menor cor possível ao vértice  $v$ 
  |   fim
fim

```

Fonte – Produzido pelo próprio autor

3.3 CRITÉRIOS DE ORDENAÇÃO DOS VÉRTICES

Nesta subseção, apresentaremos quatro critérios de ordenação de vértices, sendo eles: LF (*largest-first*), SL (*smallest-last*), LFI (*largest-first interchange*) e SLI (*smallest-last interchange*):

3.3.1 LF

O método LF foi apresentado pela primeira vez por (WELSH; POWELL, 1967). É um dos métodos sequenciais mais antigos e simples. O método é baseado na observação de que vértices com menor grau são mais flexíveis na escolha de cores se comparado a vértices com maior grau, logo é natural colorir primeiro os outros vértices de maior grau. Apesar de sua simplicidade o método LF mostra ser eficiente, na quantidade de cores necessárias para colorir um grafo.

Algoritmo 3: Coloração-LF(G)

```

início
  |  $K :=$  vértices de  $G$  organizados em ordem decrescente de grau
  | Coloração-Gulosa( $G, K$ )
fim

```

Fonte – Versão baseada em Welsh e Powell (1967)

3.3.2 SL

Outro método de coloração sequencial bastante conhecido e similar ao método LF é o método SL (MATULA; BECK, 1983). O método SL, assim como, o método LF é baseado na observação de que os vértices com poucos vizinhos devem ser os últimos vértices a serem coloridos, já que, são mais flexíveis na escolha de cores. A criação da sequência de vértices do método SL passa por um processo de refinamento que adiciona a K o vértice com o menor grau no subgrafo gerado por V / K . Por fim o método inverte a sequência obtida por K . Dessa forma para instâncias de grafos que sejam cíclicos, uni-cíclicos direcionados e bipartidos o método mostra-se ser ótimo. O símbolo \bar{K} utilizado no algoritmo de coloração SL abaixo representa a inversão da ordem da sequência K .

Algoritmo 4: Coloração-SL(G)

```

início
  |  $K := \emptyset$ 
  | while  $V / K \neq \emptyset$  do
  |   | adicione a  $K$  o vértice com o menor grau no subgrafo gerado por  $V / K$ 
  | end
  | Coloração-Gulosa( $G, \bar{K}$ )
fim

```

Fonte – Versão baseada em Matula e Beck (1983)

3.3.3 LFI

O método LFI é uma versão modificada do método LF que inclui o procedimento de coloração com troca. O algoritmo LFI pode ser implementado em tempo $O(mn)$.

Algoritmo 5: Coloração-LFI(G)

```

início
  |  $K :=$  vértices de  $G$  organizados em ordem não crescente de grau
  | Coloração-com-Troca( $G, K$ )
fim

```

Fonte – Versão baseada em Matula e Beck (1983)

3.3.4 SLI

Por último, o método SLI é uma modificação do método SL análogo ao LFI. Como discutido anteriormente, o método tira vantagem da mecânica de coloração com troca e é possível implementar em tempo $O(mn)$.

Algoritmo 6: Coloração-SLI(G)

```

início
  |  $K := \emptyset$ 
  | while  $V / K \neq \emptyset$  do
  |   | adicione a  $K$  o vértice com o menor grau no subgrafo gerado por  $V / K$ 
  | end
  | Coloração-com-Troca( $G, \bar{K}$ )
  |  $\bar{K}$  denota a inversão da ordem da sequência  $K$ 
fim

```

Fonte – Versão baseada em Matula e Beck (1983)

3.4 DSATUR

O método DSATUR escolhe os vértices durante o processo de coloração, escolhendo o primeiro vértice da entrada de acordo com sua pontuação. A pontuação varia de acordo com os critérios específicos do problema.

Algoritmo 7: DSATUR(G)

início

Atribua ao vértice de maior grau a menor 1

Selecione o vértice com maior grau de saturação.

Se houver vértices com mesmo grau de saturação, opte por qualquer um de grau máximo pertencente ao sub-grafo ainda não colorido

Atribua ao vértice selecionado a cor de menor índice possível

se todos os vértices estiverem coloridos então

| pare

fim

senão

| retorne para a seleção do vértice com maior grau de saturação

fim

fim

Fonte – Versão baseada em Brélaz (1979)

3.5 CG

O método CG recebe uma coloração de vértices válida como entrada. Então ele separa os vértices em grupo de cores. Os grupos são organizados de acordo com a cor dos vértices. Cada grupo possui somente os vértices coloridos de uma mesma cor. Ao separar os vértices em grupos é possível adquirir uma nova ordem de coloração a partir desses grupos.

Dado uma coloração C como entrada. Podemos conseguir uma nova ordem.

Algoritmo 8: CG(G, C)

início

$K := \emptyset$

Crie os grupos de cores de acordo com a coloração de entrada C

Selecione os vértices de cada grupo em ordem crescente de tamanho de grupo

Adicione os vértices selecionados a ordem K

Coloração-Gulosa(G, K)

fim

Fonte – Produzido pelo próprio autor

4 PROCEDIMENTOS METODOLÓGICOS

Nessa Seção os procedimentos metodológicos serão apresentados.

4.1 ALGORITMOS LF, SL, LFI, SLI, DSATUR E CG

Implementaremos as heurísticas de coloração SL, LF, LFI, SLI, DSATUR e a nova heurística CG. Todos os algoritmos serão desenvolvidos utilizando a linguagem de programação C++.

A estrutura base a ser utilizada para a implementação desses algoritmos foi apresentada na Seção 4.2. Com isso, poderemos utilizar essas implementações para a construção da nossa nova heurística.

4.2 ESTRATÉGIA DA NOVA HEURÍSTICA DE COLORAÇÃO

A estratégia da nova heurística de coloração, separa os vértices em grupo de cores. Os grupos são organizados de acordo com a cor dos vértices. Cada grupo possui somente os vértices coloridos com uma mesma cor. Ao separar os vértices em grupos é possível adquirir uma nova ordem a partir desses grupos. A ideia é minimizar os grupos de cores maximais e com isso tentar rearranjar os vértices dos grupos menores, afim de diminuir a quantidade de grupos e assim diminuindo a quantidade de cores necessárias para se colorir o grafo.

A nova ordem de vértices a serem recoloridos será dada de acordo com os grupos de cores criados. Os primeiros vértices a serem escolhido, serão aqueles pertencentes aos menores grupos de cores.

4.3 COMPARAÇÃO ENTRE OS MÉTODOS

A comparação dos métodos será realizada analisando alguns critérios. O principal critério de avaliação, consiste na quantidade de cores utilizadas em cada método heurístico de coloração gulosa sequencial em comparação a nova heurística. Avaliar a complexidade dá uma ideia, se o método utilizado é bom ou ruim ao solucionar o problema proposto. A partir da complexidade é possível também procurar melhorias na estrutura do método, afim de diminuir sua complexidade e torná-lo mais eficiente.

4.4 ORDEM OBTIDA PELA NOVA HEURÍSTICA

Analisar se a ordem obtida pela nova heurística é melhor ou pior que as ordens obtidas pelo métodos heurísticos gulosos sequenciais citados anteriormente é um passo extremamente importante, pois determina a quantidade de cores a serem utilizadas para colorir um grafo. A ideia para verificar se a ordem da nova heurística é melhor, consiste em realizar testes para grandes instâncias de grafos com diferentes níveis de densidade e quantidade de vértices, abordando a maior variedade de grafos possíveis. Esse processo será feito utilizando instâncias retiradas de sites *Benchmark* (*Benchmark* é o ato de executar um programa de computador ou um conjunto de programas), que possuem as instâncias de entradas necessárias para a realização dos testes computacionais.

A quantidade de cores utilizada para cada método em comparação ao novo método será visualizada em forma de tabela e gráfico, facilitando a visualização das diferenças, na eficiência ao colorir os grafos. Com isso é possível ter uma visão de como a nova heurística se comporta em relação aos outros métodos.

5 RESULTADOS

Nesta seção são apresentados os resultados dos algoritmos LF, LFI, SL, SLI, DSATUR, CGLF (*color group largest-first*), CGLFI (*color group largest-first interchange*), CGSL (*color group smallest-last*), CGSLI (*color group smallest-last interchange*) e CGDSATUR (*color group dsatur*). Os resultados são apresentados na tabela 2.

Os algoritmos foram executados em um computador com o processador Intel(R) Core(TM) i5-7200U CPU @ 2.50GHz 2.70GHz, 8GB de memória RAM e Linux 64 bit. O desempenho dos algoritmos são avaliados pela quantidade de cores necessária para se colorir o grafo. Utilizadas instâncias de grafos do benchmark *graph coloring*.

As modificações feitas nos algoritmos LF, LFI, SL, SLI e DSATUR, deram origem aos algoritmos CGLF, CGLFI, CGSL, CGSLI e CGDSATUR. Esses algoritmos modificados, utilizam a ideia da nova heurística, apresentada nesse trabalho.

Na Tabela 2, o número de cores necessária pra se colorir o grafo é comparado para cada algoritmo e o menor resultado é destacado.

Tabela 2 – Resultados

Grafo			CGLF	CGLFI	LF	LFI	CGSL	CGSLI	SL	SLI	CGDSATUR	DSATUR
Nome	V	Densidade										
DSJR500.1	500	0.028497	13	12	13	13	12	11	12	12	13	13
DSJR500.5	500	0.471840	134	131	134	133	127	126	127	127	133	133
DSJC500.1	500	0.099864	18	16	19	17	18	16	19	18	17	17
4-Insertions_4	475	0.015945	5	4	5	4	5	4	5	4	5	5
le450_25d	450	0.172482	31	29	31	30	31	30	31	30	30	30
le450_5c	450	0.097035	12	10	13	11	10	8	11	8	10	11
le450_25c	450	0.171670	30	29	30	29	30	29	31	29	29	29
le450_5d	450	0.096580	13	11	13	13	14	13	15	14	11	12
mug100_25	100	0.033535	4	3	4	3	4	3	4	3	4	4
2-Insertions_4	149	0.049066	5	4	5	4	5	4	5	5	5	5
myciel7	191	0.130063	8	7	8	8	8	7	8	7	8	8
queen10_10	100	0.593939	15	14	17	14	15	13	18	14	15	17
3-FullIns_4	405	0.043075	7	6	7	7	7	6	7	7	7	7
4-FullIns_4	690	0.027976	8	7	8	8	8	7	8	8	8	8
5-FullIns_4	1085	0.019377	9	8	9	9	9	8	9	9	9	9
queen16_16	256	0.387255	23	21	27	21	23	21	24	22	24	26
mug88_1	88	0.038140	4	3	4	3	4	3	4	3	4	4
miles250	128	0.095226	8	7	8	8	8	7	8	8	8	8
2-FullIns_4	212	0.072476	6	5	6	6	7	6	8	6	6	6
queen12_12	144	0.504274	17	16	19	16	17	16	19	17	18	20
le450_25a	450	0.081762	26	24	26	25	25	24	25	25	25	25
myciel3	11	0.363636	4	3	4	3	4	3	4	3	4	4
queen8_8	64	0.722222	13	11	13	13	12	11	15	11	12	13
DSJC250	250	0.503390	39	38	40	38	39	37	39	38	39	39
qg.order40	1600	0.048780	40	40	64	64	40	40	64	64	40	64
mulsoi.i.4	185	0.231845	31	30	31	30	31	30	32	31	31	31
le450_15a	450	0.080851	18	17	18	18	18	17	18	17	18	18
queen5_5	25	1.066667	5	5	7	6	7	6	8	7	5	7
inithx.i.1	864	0.050178	54	53	54	53	54	53	54	54	54	54
DSJC500.5	500	0.501996	69	66	70	69	69	67	70	69	67	68
queen8_12	96	0.600000	14	13	15	14	14	13	15	14	15	16
le450_5a	450	0.056560	11	10	11	11	11	10	11	11	11	11
games120	120	0.178711	9	8	9	8	9	8	9	9	9	9
school1_nsh	352	0.236532	32	31	33	31	20	18	21	20	25	26
4-FullIns_5	4146	0.008997	9	8	9	9	9	8	9	9	9	9
3-Insertions_3	56	0.071429	4	3	4	3	4	3	4	4	4	4
1-Insertions_5	202	0.060440	6	5	6	5	6	5	6	6	6	6
myciel4	23	0.280632	5	4	5	4	5	4	5	4	5	5
zeroin.i.3	206	0.167653	30	29	30	29	30	29	30	30	30	30
wap08a	1870	0.059614	50	47	50	47	46	45	46	46	47	48
1-FullIns_5	282	0.081951	6	5	6	6	6	5	7	6	6	6
fpsol2.i.1	496	0.094933	65	64	65	64	65	64	65	65	65	65
3-FullIns_5	2030	0.016388	8	7	8	8	8	7	9	9	8	8
abb313GPIA	1557	0.053981	14	13	15	14	13	12	18	18	12	13
david	87	0.217054	11	10	11	11	11	10	11	11	11	11
1-Insertions_6	607	0.034455	7	6	7	6	7	6	7	6	7	7
mulsoi.i.3	184	0.232597	31	30	31	30	31	30	32	31		
2-Insertions_5	597	0.022124	6	5	6	5	6	5	6	5	6	6
miles1500	128	1.279035	73	72	73	73	73	72	73	73	73	73
queen13_13	169	0.468864	19	17	23	18	19	17	19	18	19	23
DSJC1000.9	1000	0.899798	305	302	312	309	310	303	319	311	301	306
wap02a	2464	0.036825	48	47	50	48	46	44	47	48	47	47
DSJC1000	1000	1.000304	121	118	122	121	123	119	125	121	121	121
DSJC125	125	0.094968	7	6	7	6	6	5	7	7	7	7
wap04a	5231	0.021559	50	49	51	50	48	47	49	48	50	51
2-Insertions_3	37	0.108108	4	3	4	3	4	3	4	3	4	4
le450_15c	450	0.165108	25	24	26	25	26	25	26	25	24	25

Grafo			CGLF	CGLFI	LF	LFI	CGSL	CGSLI	SL	SLI	CGDSATUR	DSATUR
Nome	V	Densidade										
5-FullIns_3	154	0.067227	8	7	8	8	8	7	8	8	8	8
queen6_6	36	0.920635	8	7	9	8	9	8	10	8	8	9
fpsol2.i.2	451	0.085647	30	29	30	29	30	29	30	30	30	30
mulsol.i.2	188	0.221015	31	30	31	30	31	30	32	31	31	31
miles750	128	0.519931	32	31	32	32	31	30	31	31	31	31
inithx.i.3	621	0.072562	31	30	31	30	31	30	32	31	31	31
myciel5	47	0.218316	6	5	6	5	6	5	6	5	6	6
wap05a	905	0.105317	50	49	50	50	51	50	51	50	50	51
inithx.i.2	645	0.067307	31	30	31	30	31	30	32	31	31	31
myciel6	95	0.169093	7	6	7	6	7	6	7	6	7	7
le450_15b	450	0.080861	18	17	18	17	18	17	18	17	17	17
queen9_9	81	0.651852	14	12	15	14	13	12	15	13	14	15
miles1000	128	0.791339	42	41	43	43	42	41	42	42	42	42
DSJC1000.1	1000	0.099357	29	28	29	28	29	28	30	28	28	28
ash958GPIA	1916	0.006817	7	6	10	7	6	5	10	7	7	10
1-Insertions_4	67	0.104930	5	4	5	4	5	4	5	5	5	5
4-Insertions_3	79	0.050633	4	3	4	3	4	3	4	3	4	4
queen11_11	121	0.545455	16	15	17	15	16	15	16	15	16	17
3-Insertions_5	1406	0.009816	6	5	6	5	6	5	6	6	6	6
2-FullIns_3	52	0.151584	5	4	5	5	5	4	5	5	5	5
huck	74	0.222880	11	10	11	10	11	10	11	11	11	11
zeroin.i.2	211	0.159828	30	29	30	29	30	29	30	30	30	30
anna	138	0.104306	11	10	11	10	11	10	11	11	11	11
DSJC250.1	250	0.103390	11	10	11	11	11	10	12	10	11	11
DSJC125.9	125	0.898194	52	51	54	52	50	50	52	51	54	55
zeroin.i.1	211	0.185060	49	48	49	48	49	48	49	49	49	49
mulsol.i.5	186	0.230921	31	30	31	30	31	30	31	31	31	31
3-Insertions_4	281	0.026589	5	4	5	4	5	4	5	4	5	5
DSJC125.5	125	0.502065	23	21	24	22	24	23	25	24	23	23
will199GPIA	701	0.028796	9	8	10	9	8	7	8	8	8	9
jean	80	0.160759	10	9	10	9	10	9	10	10	10	10
wap07a	1809	0.063209	50	49	50	49	47	46	49	47	49	48
mug100_1	100	0.033535	4	3	4	3	4	3	4	3	4	4
le450_15d	450	0.165801	26	25	26	26	26	25	26	25	25	25
le450_5b	450	0.056758	11	10	12	11	11	10	12	10	11	12
mug88_25	88	0.038140	4	3	4	3	4	3	4	3	4	4
mulsol.i.1	197	0.203305	49	48	49	48	49	48	49	49	49	49
wap06a	947	0.097272	48	45	48	46	45	43	45	47	49	49
wap03a	4730	0.025637	54	53	56	54	51	49	55	54	52	53
queen15_15	225	0.411111	23	20	25	21	22	20	25	21	23	25
latin_square_10	900	0.759733	150	141	219	196	150	141	219	196	147	213
DSJC500.5	500	1.003992	69	66	70	69	69	67	70	69	67	68
wap01a	2368	0.039561	51	49	52	52	47	46	48	47	49	52
homer	561	0.020741	13	12	13	12	13	12	13	13	13	13
school1	385	0.258320	34	31	34	31	15	14	15	14	33	34
DSJR500.1c	500	0.972144	95	94	99	98	99	98	104	105	91	91
ash608GPIA	1216	0.010618	7	5	10	7	6	5	8	7	7	10
le450_25b	450	0.081792	25	24	25	25	25	24	25	25	25	26
DSJC1000.5	1000	0.500152	121	118	122	121	123	119	125	121	121	122
DSJC250.9	250	0.896289	91	88	91	91	91	89	96	92	92	92
qg.order100	10000	0.019802	104	104	128	125	104	104	128	126	104	128
qg.order30	900	0.064516	30	30	32	33	30	30	32	33	30	32
queen14_14	196	0.438095	21	19	25	19	20	18	22	19	21	23
3-FullIns_3	80	0.109494	6	5	6	6	6	5	6	6	6	6
fpsol2.i.3	425	0.096426	30	29	30	29	30	29	30	30	30	30
1-FullIns_4	93	0.138616	5	4	5	5	5	4	5	5	5	5
1-FullIns_3	30	0.229885	4	3	4	4	4	3	4	4	4	4

Grafo			CGLF	CGLFI	LF	LFI	CGSL	CGSLI	SL	SLI	CGDSATUR	DSATUR
Nome	V	Densidade										
ash331GPIA	662	0.019128	6	5	10	7	6	5	10	7	7	10
2-FullIns_5	852	0.033656	7	6	7	7	7	6	9	8	7	7
4-FullIns_3	114	0.083993	7	6	7	7	7	6	7	7	7	7
miles500	128	0.287894	20	19	20	20	20	19	20	20	20	20
DSJC500.9	500	1.802597	168	162	175	166	167	165	173	167	166	169
queen7_7	49	0.809524	11	10	12	10	10	9	10	10	11	12
qg.order60	3600	0.032787	60	60	64	65	60	60	65	65	60	64

Fonte – Produzido pelo próprio autor

6 DISCUSSÃO

Nesta monografia foram apresentadas algumas heurísticas de coloração de grafos para o problema de coloração de grafos. O trabalho teve como objetivo apresentar uma nova heurística de coloração de grafos, que diminuísse a quantidade de cores necessária para se colorir um grafo. Com base nessa nova heurística, foram criados cinco novos algoritmos, com base nas heurísticas apresentadas nesse trabalho.

Os experimentos realizados mostram que a nova heurística de coloração de grafos apresenta uma redução no número de cores necessária pra se colorir um grafo em comparação com as outras heurísticas. Os resultados da nova heurística mostram que ela diminuí a quantidade de cores necessária para se colorir o grafo para a maioria dos casos. É importante notar que os algoritmos que utilizam a nova heurística foram capazes de resolver as instâncias nos grafos do benchmark do *graph coloring* em um tempo muito pequeno e por este motivo, a análise no tempo de execução se mostrou desnecessária.

7 CONSIDERAÇÕES FINAIS

A nova heurística apresenta a estratégia de criar grupos de cores e com base nisso modificar a ordem e recolorir o grafo, afim de diminuir o número de cores. O trabalho é interessante no ponto de vista teórico, pois usa a nova heurística de coloração, que possui uma nova estratégia de coloração, não apresentada por outros trabalhos.

Para trabalhos futuros é interessante realizar mais testes entre os algoritmos com grafos de diferentes tamanhos e densidades, para melhores conclusões sobre o comportamento da nova heurística de coloração. Além disso é possível randomizar a ordem que os grupos de cores são escolhidos, de forma a gerar novas ordens de coloração.

REFERÊNCIAS

- BRÉLAZ, D. New methods to color the vertices of a graph. **Communications of the ACM**, ACM, v. 22, n. 4, p. 251–256, 1979.
- CORMEN, T. H.; LEISERSON, C. E.; RIVEST, R. L.; STEIN, C. **Introduction to algorithms**. [S.l.]: MIT press, 2009.
- CULBERSON, J. C.; LUO, F. Exploring the k-colorable landscape with iterated greedy. **Cliques, coloring, and satisfiability: second DIMACS implementation challenge**, American Mathematical Society: Providence, RI, v. 26, p. 245–284, 1996.
- MATULA, D. W.; BECK, L. L. Smallest-last ordering and clustering and graph coloring algorithms. **Journal of the ACM (JACM)**, ACM, v. 30, n. 3, p. 417–427, 1983.
- RADIN, A. **Graph coloring heuristics from investigation of smallest hard to color graphs**. [S.l: s.n], 2000.
- SILVA, L. da C. **Algoritmos para Escalonamento de Instruções e Alocação de Registradores na Infraestrutura LLVM**. Tese (Doutorado) — Master's thesis, Universidade Federal de Mato Grosso do Sul, 2013. Faculdade de Computação, 2011.
- WELSH, D. J.; POWELL, M. B. An upper bound for the chromatic number of a graph and its application to timetabling problems. **The Computer Journal**, Oxford University Press, v. 10, n. 1, p. 85–86, 1967.