Uma Heurística baseada em Iterated Local Search para o problema de Coloração Equilibrada de Vértices

Amadeu Almeida Coco [1], **Thiago F. Noronha**[1]

[1]Departamento de Ciência da Computação
Universidade Federal de Minas Gerais – UFMG
Avenida Presidente Antônio Carlos 6627
Belo Horizonte – Minas Gerais – Brazil

{amadeuac,tfn}@dcc.ufmg.br

Resumo. *O problema de coloração de grafos é um dos mais estudados na otimização combinatória e possui inúmeras variações. Uma delas é o problema de Coloração Equilibrada de Vértices. Uma coloração equilibrada de um grafo é definida pela partição de seus vértices em k subconjuntos onde a diferença de cardinalidade de dois subconjuntos quaisquer é no máximo um. O problema de Coloração Equilibrada de Vértices (ECP) consiste em minimizar o valor de k de modo que um grafo possa ser colorido com k cores de forma equilibrada. Este trabalho propõe uma heurística ILS para este problema. Os resultados computacionais demonstram que a heurística proposta, na média, obteve soluções melhores que os trabalhos comparados.*

PALAVRAS CHAVE. *Coloração Equilibrada de Vértices, Grafos, Heurísticas.*

Abstract. *The Graph Cloring Problem is one of the most studied problems in optimization and has many variations. For instance, the Equitable Coloring Problem. A equitable coloring of a graph is defined as a partition of its vertices in k subsets, where the difference on the cardinality of any two subsets is at most one. The Equitable Coloring Problem (ECP) tries to minimize the value of k so that a graph can be equitable colored with k colors. Computational results show that the proposed heuristic, on average, had better solutions compared with the related works.*

KEYWORDS. *Equitable Coloring Problem, Graphs, Heuristics.*

1. Introdução

O problema da Coloração Equilibrada de Vértices (ECP, do inglês Equitable Coloring Problem) consiste em colorir os vértices de um grafo de forma que (i) vértices adjacentes possuam cores diferentes e que (ii) o número de vértices coloridos com cada cor se diferencie em no máximo uma unidade. O problema foi proposto por Erdös [10] ao conjecturar que um grafo com grau máximo Δ pode ser colorido equitativamente com $\Delta+1$ cores. Apesar da semelhança entre o ECP e o problema clássico de coloração de vértices (GCP, do inglês Graph Coloring Problem), existe uma diferença entre ambos. A diferença é o fato de existir uma coloração para ECP com K cores não implica que exista uma coloração com $K+1$ cores [28]. Por exemplo, o grafo bipartido $K_{3,3}$ admite uma coloração equilibrada com duas cores, porém não admite uma coloração equilibrada com três cores. Este problema é NP-Completo [13, 22].

2. Trabalhos Relacionados

A versão de decisão do *Graph Coloring Problem (GCP)* é um problema NP-completo [16]. Várias classes de problemas reais podem ser modeladas como extensões do GCP como por exemplo, os problemas de construção de tabelas [5] e de alocação de frequências [9]. Devido à dificuldade de se resolver este problema de forma exata, a maioria dos trabalhos apresentados na literatura sugerem o uso de heurísticas [8, 11, 14, 15, 19, 20, 23, 24, 30].

Em 1973, Meyer [28], motivado por uma aplicação prática de coleta de lixo municipal, conjecturou uma forma do teorema de Brooks [4] para colorir grafos de forma equilibrada. Cada grafo com grau máximo Δ tem uma equidade com Δ ou menos cores, exceto para grafos completos.

Lozano et al.[25] utilizaram o ECP aplicado ao problema de redes de interconexão. Uma rede de interconexão é uma estrutura composta por um conjunto P de $n > 1$ processadores e um conjunto T de ligações. Os autores do trabalho apresentaram uma representação natural para o processamento paralelo neste tipo de rede através de uma coloração total equilibrada com no máximo $D+2$ cores, onde D é o grau máximo do grafo que representa a rede.

Chen et al. [6, 7] provaram que o ECP é polinomialmente resolvível quando o grafo é uma árvore [6] ou quando o grafo é do tipo split [7]. Hajnal e Szemerédi [17] provaram que todo grafo $G = (V,E)$ possuí coloração $(\Delta+1)$-equilibrada, onde $\Delta = max_{v\in V}\{d(v)\}$ e $d(v)$ denota o grau do vértice $v \in V$. Kierstead e Kostochka [21] apresentaram uma prova mais simples para tal resultado além de uma heurística aproximativa que alcança este limite superior $(\Delta+1)$. Além disso, os autores de [21] demonstraram que todo grafo que satisfaz $d(u)+d(v) \leq 2k+1$ para toda aresta $(u,v) \in E$ possuí uma $(k+1)$-coloração equilibrada.

Furmanczyk e Kubale [13] provaram que o ECP é NP-difícil e propuseram uma heurística construtiva para o problema. Esta heurística será discutida em detalhes na seção 3.1, já que ela faz parte da estratégia de solução proposta neste trabalho.

Méndez-Díaz et al. [27] realizaram um estudo poliédrico do modelo de programação linear inteira (PLI) do ECP, visto que no GCP o estudo poliédrico foi bastante utilizado no aperfeiçoamento dos algoritmos para o problema. O trabalho teve como objetivo estudar os planos de corte para a formulação do ECP. Neste trabalho ([27]) foram resolvidas instâncias com até 35 vértices de forma exata.

2.1. O Modelo de PLI de Bahiense et. al. [2]

Bahiense et. al. [2] propuseram uma formulação baseada na formulação de [12] utilizando uma formulação através de representantes. Para quebrar as simetrias na formulação abaixo, a formulação assimétrica formulada por representantes é generalizada. É estabelecido que um vértice $u \in V$ pode apenas representar um outro vértice $v \in V$ se $u < v$. Portanto, o vértice representativo de um conjunto de cores é o vértice com o menor índice dentre aqueles coloridos com aquela determinada cor. Definido-se $A_>(u) = \{v \in A(u) : u < v\}$ como *out-anti-neighbourhood* de um vértice $u \in V$, ou seja, os vértices que não podem representar o vértice u e $A_<(u) = \{v \in A(u) : v < u\}$ como *in-anti-neighbourhood* de um vértice $u \in V$, ou seja, os vértices que podem representar o vértice u, exceto ele mesmo. Além disso, define-se $A'_>(u) = A_>(u) \cup \{u\}$ e $A'_<(u) = A_<(u) \cup \{u\}$, para todo $u \in V$.

Também é definido $V^s = \{u \in V : A_<(u) = \emptyset\}$ como conjunto de vértices cujo *in-anti-neighbourhood* em G é vazia, ou seja, o conjunto de vértices que sempre são representantes. Como os vértices em V^s são sempre representantes, $x_{vv} = 1$ em qualquer solução viável para qualquer $v \in V^s$. Portanto estas variáveis podem ser removidas da formulação e reescritas na formulação assimétrica (1) - (8) abaixo.

Nas restrições abaixo as variáveis binárias x_{uv} $\forall v \in A(v)$ são iguais a 1 se e somente se o vértice u representa a cor do vértice v, caso contrário $x_{uv} = 0$. A variável de equilíbrio $w \in \mathbb{R}, w \geq 0$, indica a cardinalidade do conjunto máximo estável relativo à coloração equilibrada, ou seja, se a cardinalidade de cada conjunto estável é w ou $w-1$. A variável β_u é referente a assimetria da formulação e $\beta_u = 1$ se $u \in V^s$, caso contrário $\beta_u = x_{uu}$. L_w e U_w são definidos, respectivamente, como os limites mínimo e máximo inteiro para o valor de w.

A função objetivo (1) conta o número de vértices representantes, ou seja, o número de cores (ou o número de conjuntos de cores). As restrições do conjunto (2) garantem que cada vértice $u \in V$ tem que ser representado por si mesmo ou por outro vértice v na sua anti-vizinhança. As inequações (3) cumprem a restrição de que vértices adjacentes precisam possuir cores diferentes. As restrições (3) juntamente com as restrições (4) garantem que cada vértice pode apenas ser representado por um vértice representante. As restrições (5) e (6) garantem que a diferença entre as cardinalidades entre quaisquer dois conjuntos de cores é no máximo um. As restrições (7) e (8) definem o domínio das variáveis do problema.

$$\min \sum_{(v)\in V\setminus V^s} x_{vv} + |V^s| \quad \text{subject to:} \tag{1}$$

$$\sum_{v\in A'_<(u)} x_{vu} = 1 \forall u \in V\setminus V^s \tag{2}$$

$$x_{uv} + x_{ub} \leq \beta_u,\ \forall u \in V, \forall (v,b) \in E : v,b \in A_>(u) \tag{3}$$

$$x_{uv} \leq x_{uu},\ \forall u \in V\setminus V^s,\ \forall v \in A_>(u) : v \text{ é isolado em } A_>(u) \tag{4}$$

$$\beta_u + \sum_{v\in A_>(u)} x_{uv} \leq w \cdot \beta_u,\ \forall u \in V \tag{5}$$

$$\beta_u + \sum_{v\in A_>(u)} x_{uv} \geq (w-1) \cdot \beta_u,\ \forall u \in V \tag{6}$$

$$x_{uv} \in \{0,1\}\ \forall u \in V, \forall v \in A_>(u) \tag{7}$$

$$w \in \mathbb{R},\ \text{onde } \beta_u = 1 \text{ se } u \in V^s;\ \textit{caso contrário}\ \beta_u = x_{uu}. \tag{8}$$

As restrições (5) e (6) podem ser linearizadas e substituídas pelas restrições (9-12). São introduzidas novas variáveis y_i, para todo $i \in [L_w, U_w]$. $y_i = 1$ se a cardinalidade do conjunto máximo na partição de uma coloração equilibrada de G é i, caso contrário $y_i = 0$. Portanto w pode ser substituído por $\sum_{i=L_w}^{i=U_w} i \cdot y_i$ adicionando a restrição (9):

$$\sum_{i=L_w}^{U_w} y_i = 1 \tag{9}$$

Para todo $u \in V$ e todo inteiro $i \in [L_w, U_w]$, é introduzido novas váriaveis z_{ui} tais que $z_{ui} = x_{uu} \cdot y_i$ para toda solução inteira do ECP, juntamente com as inequações lineares:

$$z_{ui} \leq y_i,\ z_{ui} \leq x_{uu},\ z_{ui} \geq y_i + x_{uu} - 1,\ \forall u \in V, \forall i \in [L_w, U_w]. \tag{10}$$

Por substituição temos que as restrições do tipo (5) podem ser reescritas como:

$$\beta_u + \sum_{v\in A_>(u)} x_{uv} \leq \sum_{i=L_w}^{U_w} i \cdot z_{ui},\ \forall u \in V \tag{11}$$

De forma similar, a mesma transformação pode ser aplicada em (6), resultando em :

$$2 \cdot \beta_u + \sum_{v\in A_>(u)} x_{uv} \geq \sum_{i=L_w}^{U_w} i \cdot z_{ui},\ \forall u \in V \tag{12}$$

Portanto, o ECP pode ser alternativamente formulado através da formulação resultante definida pela função objetivo (1) e as restrições (2)-(4),(7-12). O número de variáveis nesta formulação é $|V\backslash V^s| + \overline{m} + \rho(1+|V|)$, onde $\rho = U_w - L_w$. Contudo, como $z_{ui} = y_i$ para todo $u \in V^s$, z_{ui} pode ser substituído por y_i para todo $u \in V^s$, o que reduz o número de variáveis da formulação para $|V\backslash V^s| + \overline{m} + \rho(1+|V\backslash S|)$.

Em [2] também é proposta uma heurística baseada em busca tabu [18] para realizar o cálculo dos limites superiores relativos a cada subproblema dos algoritmos *branch-and-cut* gerados. O algoritmo é baseado na heurística para o problema de assinalamento de frequências proposto em [31].

A solução inicial define o vértice representante $v \in V$ como o $argmax_{u \in A'_<(v)} x^*_{uv}$, onde x^*_{uv} é o valor da variável x_{uv} na solução ótima da relaxação linear LF_2. Essa solução inicial pode violar as restrições (3), assim como as restrições de coloração equilibrada (5) e (6).

A função objetivo da heurística de busca tabu consiste em minimizar a soma de quatro custos c_1, c_2, c_3, c_4, $\forall u \in V$ e para cada $v \in A'_>(u)$, onde $\overline{x}_{uv} = 1$ se e somente se o vértice u representa a cor do vértice v na solução atual da tabu e caso contrário $\overline{x}_{uv} = 0$. Além disso, define-se $\overline{w} = \lceil \frac{|V|}{K} \rceil$ como a cardinalidade de um "conjunto máximo"estável em uma solução viável com K cores (a cardinalidade estável de cada conjunto é w ou $w-1$). Os custos de c_1, c_2, c_3, c_4 são calculados como:

$$c_1 = \sum_{V \in V \backslash V^s} \overline{x}_v v + |V^s|; \tag{13}$$

$$c_2 = \sum_{u \in V} \sum_{(v,b) \in E, v,b \in A_>(u)} max\{\overline{x}_u v + \overline{x}_u b - 1, 0\}; \tag{14}$$

$$c_3 = \sum_{u \in V} max\{\overline{w} - 1 - \beta_u - \sum_{v \in A_>(u)} \overline{x}_{uv}, 0\}; \tag{15}$$

$$c_4 = \sum_{u \in V} max\{-\overline{w} + \beta_u - \sum_{v \in A_>(u)} \overline{x}_{uv}, 0\}; \tag{16}$$

A componente de custo c_1 conta o número de vértices representantes, ou seja, número de cores na solução definida por $\overline{x}$ e $\overline{w}$, enquanto as outras três componentes da função objetivo medem o grau de inviabilidade da solução. A componente de custo c_2 mede o grau de inviabilidade de acordo com a restrição (3) através da contagem do número de vértices adjacentes com o mesmo vértice representativo. As componentes c_3 e c_4 medem o grau de inviabilidade das restrições de equilíbrio de cores (5) e (6), respectivamente. Para uma solução viável temos $c_2 = c_3 = c_4 = 0$.

A vizinhança N de uma solução consiste em todas as soluções que podem ser obtidas trocando-se o representante $u \in A'_<$ de um vértice $v \in V$ por outro vértice $b \in A'_<$. Por questões de eficiência não são avaliadas todas as $O(n^2)$ soluções da vizinhança. A busca tabu é implementada utilizando a estratégia de busca local *best improvement*. A cada iteração a solução corrente é trocada pela melhor solução dentro da vizinhança restrita N'. Portanto o representativo $u \in A'_<(v)$ de um vértice $v \in V$ é trocado por outro vértice $b \in A'_<(v)$.

3. Iterated Local Search para o Problema de Coloração Equilibrada

A heurística proposta por [8], baseada na meta heurística *Iterated Local Search*, é a heurística que produz os melhores resultados na literatura de GCP. Enquanto a heurística construtiva de [13] produz bons resultados para o ECP. Uma grande desvantagem do algoritmo de [13] é a utilização da heurística construtiva *smallest-last* [26] que é bastante ineficiente na resolução do GCP. O *smallest last* [26] consiste na ordenação dos vértices do grafo através de seus respectivos graus. Em seguida, os vértices são coloridos de forma que os vértices com menor grau são coloridos por último. A heurística de [26] é rápida, porém os resultados são ruins.

Este trabalho combina a heurística proposta por [8], para o GCP, com a heurística proposta para o ECP em [13] de forma a obter melhores resultados para o ECP.

3.1. ILS de Chiarandini e Stützle [8] para GCP

Em [8] os autores dividem a heurística ILS em três fases. A primeira, chamada de fase de inicialização é realizada pela heurística de [3], que retorna uma solução válida para o problema. A segunda fase, chamada de decremento do número cores utilizadas, ocorre quando a heurística de [3] ou a busca local MinMaxConfict [29] fornecem uma solução viável. Nesta fase uma cor é removida e os nós que foram coloridos com aquela cor são recoloridos com outra cor de forma aleatória, gerando assim uma solução inviável. A terceira fase, chamada de busca local, ocorre quando a coloração retornada na segunda fase não é viável. Nesta fase, algoritmos de busca local são utilizados para que uma coloração viável seja utilizada. O algoritmo de [8] funciona da seguinte forma:

Algoritmo 1: Iterated Local Search [8]

procedure Iterated Local Search
s_0 = GeraSolucaoInicial();
s = MinMaxConflict(s_0);
while *Condição Final não é Atingida* **do**
    s' = Pertubação(s, histórico);
    s” = MinMaxConflict(s');
    s = CriterioDeAceitação(s, s”, histórico);
    end
end

Neste trabalho o algoritmo *ILS* proposto segue três fases sugeridas em [8]. Porém na primeira fase ao invés da utilização de [3] o algoritmo utilizado é o *smallest last* [26]. Na segunda fase é utilizado o mesmo algoritmo formulado por [8] baseado em decremento do número de cores. Na terceira fase a busca local utilizada é baseada na heurística *MinMaxConflict* que é proposta para a resolução de problemas de satisfazibilidade. Em cada passo da busca local, um vértice conflitante é escolhido de forma aleatória e a cor que minimiza o número de conflitos com outros vértices é assinalada para este vértices. Esta heurística de busca local foi proposta em [29]. O pseudo-código desta heurística é apresentado em (1).

3.2. Heurística de Furmanzyck e Kubale [13] para ECP

Furmanczyk e Kubale [13] propuseram uma heurística que inicialmente gera uma solução não equilibrada através do algoritmo *Smallest-Last* [26]. Em seguida, enquanto o grafo ainda não está equilibrado o algoritmo faz as seguintes operações. (i) encontra a cor presente em menos vértices, (ii) encontra a cor existente em mais vértices e (iii) recolore um vértice quem contém a cor com mais vértices com a cor com menos vértices, mantendo a viabilidade do grafo. Se nenhuma opção para essa operação for encontrada, a heurística aumenta o número de cores em uma unidade. A complexidade deste algoritmo é $O(n^4)$. O pseudocódigo desta heurística é apresentado no algoritmo 2.

Algoritmo 2: Algoritmo de Furmanczyk e Kubale [13]

procedure FurmanczykKubale(G)[13]
while *O grafo não está equilibrado* **do**
    colmin = número da cor menos utilizada;
    colmax = número da cor mais utilizada;
    Encontre vértices coloridos com *colmin* e *colmax*;
    if *É possível alterar a cor de um vértice colorido com colmin para colmax* **then**
        Troque a cor do vértice de *colmin* para *colmax*;
        endif
    else
        Assinale uma nova cor para alguns vértices coloridos com *colmax*
end

3.3. O Equitable Iterated Local Search (EILS)

A heurística proposta neste trabalho, chamada de *Equitable Iterated Local Search* (EILS) utiliza os dois algoritmos descritos nas seções 3.1 e 3.2 buscando uma melhor solução para o ECP. O EILS funciona da seguinte forma: (i) Gera-se uma solução viável para o GCP através do algoritmo *smallest-last* [26], (ii) executa-se a busca local *MinMaxConflict* [29] para encontrar uma solução melhorada para o GCP, (iii) executa-se a heurística *FurmanczykKubale* [13] para obter uma solução viável para o ECP e (iv) enquanto uma condição de parada não é atingida perturba-se a solução encontrada em (iii) e repete-se os passos (ii) e (iii). O pseudocódigo desta heurística é apresentado no algoritmo 3.

4. Experimentos Computacionais

As heurísticas Furmanzyck e Kubale [13] e EILS foram implementadas em C++ versão 4.3.2. Os experimentos computacionais foram realizados em um Intel Core2Quad 2.50GHz, com 4 GB de RAM e Sistema Operacional UNIX Ubuntu 11.10. As instâncias utilizadas foram as mesmas utilizadas na avaliação de Tabu-LF2 [2]. As heurísticas [2] e [13] foram executadas apenas uma vez para cada uma das instâncias, pois estas heurísticas são determinísticas. A heurística EILS foi executada durante cinco segundos por dez vezes para cada instância da tabela, variando-se a semente aleatória do algoritmo.

A Tabela 1 apresenta uma comparação entre as heurísticas Tabu-LF2 [2], Furmanczyk e Kubale [13] e EILS para as instâncias usadas em [2], que por sua vez foram

Algoritmo 3: Equitable Iterated Local Search

```
procedure EILS
s0 = GeraSolucaoInicialAtravesDoSmallestLast();
s = MinMaxConflict(s0);
s = FurmanczykKubale(s);
while Condição Final não é Atingida do
    s’ = Pertubação(s, histórico);
    s” = MinMaxConflict(s’);
    s = FurmanczykKubale(s”);
    s = CriterioDeAceitação(s, s”, histórico);
end
end
```

coletadas em [1]. Na primeira coluna estão os nomes das instâncias executadas. Na segunda coluna está o valor ótimo de cada instância. Nas seis colunas seguintes são apresentados o *Upper Bound (UB)*, o GAP ($\frac{UB-LB}{LB}$) e o tempo gasto por cada instância para as heurísticas [2] e [13] respectivamente. Nas quatro últimas colunas, são apresentados o melhor resultado, a média dos resultados para as dez execuções, o GAP (calculado pela fórmula $\frac{Media-LB}{LB}$) e o tempo (em segundos) da heurística EILS.

Instância	Ótimo	Furmanzyck e Kubale [13]			Tabu-LF2 [2]			ILSE			
		UB	GAP	Time(s)	UB	GAP	Time(s)	Melhor	Média	GAP	Time(s)
miles750	31	32	3,23%	0	35	12,90%	3	31	**31,2**	0,65%	5
miles1000	42	43	2,38%	0	48	14,29%	3	42	**42,3**	0,71%	5
miles1500	73	**73**	0,00%	0	**73**	0,00%	3	73	**73**	0,00%	5
zeroin.i.1	49	65	32,65%	0	**49**	0,00%	4	55	63	28,57%	5
zeroin.i.2	36	**58**	61,11%	0	84	133,33%	5	58	**58**	61,11%	5
zeroin.i.3	36	**58**	61,11%	0	81	125,00%	5	58	**58**	61,11%	5
Queen6_6	7	10	42,86%	0	**7**	0,00%	0	7	**7**	0,00%	5
Queen7_7	7	12	71,43%	0	8	14,29%	0	7	**7,9**	12,86%	5
queen8_8	9	14	55,56%	0	**10**	11,11%	2	10	10,3	14,44%	5
myciel3	4	**4**	0,00%	0	**4**	0,00%	0	4	**4**	0,00%	5
myciel4	5	**5**	0,00%	0	**5**	0,00%	0	5	**5**	0,00%	5
jean	10	**10**	0,00%	0	**10**	0,00%	0	10	**10**	0,00%	5
anna	11	**11**	0,00%	0	12	9,09%	1	11	**11**	0,00%	5
david	30	**30**	0,00%	0	**30**	0,00%	2	30	**30**	0,00%	5
games120	9	**9**	0,00%	0	**9**	0,00%	0	9	**9**	0,00%	5
K5,2	3	**3**	0,00%	0	**3**	0,00%	0	3	**3**	0,00%	5
K7,2	6	**6**	0,00%	0	**6**	0,00%	0	6	**6**	0,00%	5
K7,3	3	**4**	33,33%	0	5	66,67%	1	4	**4**	33,33%	5
K9,4	3	5	66,67%	0	**3**	0,00%	0	5	5	66,67%	5
1-FullIns-3	4	**4**	0,00%	0	6	50,00%	0	4	**4**	0,00%	5
2-FullIns-3	5	**5**	0,00%	0	8	60,00%	0	5	**5**	0,00%	5
3-FullIns-3	6	7	16,67%	0	9	50,00%	1	6	**6,6**	10,00%	5
4-FullIns-3	7	**7**	0,00%	0	11	57,14%	0	7	7,3	4,29%	5
5-FullIns-3	8	**8**	0,00%	0	13	62,50%	2	8	8,1	1,25%	5
Média			18,62%			27,76%				**12,29%**	

Tabela 1. *Benchmark* de instâncias utilizadas e seus respectivos valores ótimos para o ECP. Nas colunas seguintes são apresentados o valor encontrado, o GAP de otimalidade (em porcentagem) e o tempo gasto (em s) pelas heurísticas [2] e [13] e o melhor valor encontrado, a média, o GAP de otimalidade (em porcentagem) e o tempo gasto pela heurística EILS.

Dos algoritmos analisados, o EILS apresentou o melhor GAP médio entre os três algoritmos. Desconsiderando as instâncias nas quais os três algoritmos atingiram o valor ótimo, o EILS é melhor em onze instâncias, o algoritmo de [13] obteve melhores resultados em seis instâncias e o algoritmo de [2] obteve o melhor resultado em três instâncias. Desta forma, na média, o algoritmo EILS apresentou um resultado médio melhor que as outras heurísticas implementadas na literatura.

5. Conclusão

Neste artigo foi proposta uma nova heurística para o problema de Coloração equilibrada de vértices (ECP). Tal heurística foi comparada com as melhores heurísticas apresentadas pela literatura. Os resultados mostram que o EILS obteve resultados melhores ou iguais na grande maioria das instâncias apresentadas.

Para trabalhos futuros são sugeridos a utilização da meta heurística BRKGA para o problema de coloração de grafos equilibrada.

Referências

[1] Graph coloring instances. http://mat.gsia.cmu.edu/COLOR/instances.html,2002, última visita em 20/04/2012.

[2] L. Bahiense, Y. Frota, T. F. Noronha, and C. C. Ribeiro. A branch-and-cut algorithm for the equitable coloring problem using a formulation by representatives. *Aceito para Discrete Applied Mathematics*, 2012.

[3] D. Brélaz. New methods to color the vertices of a graph. *Communications of the ACM*, 22:251–256, 1979.

[4] R. L. Brooks. On colouring the nodes of a network. In *Proceedings of Cambridge Philosophical Society, Math. Phys. Sci*, volume 37, pages 194–197, 1941.

[5] M. W. Carter. A survey of practical applications of examination timetabling algorithms. *Operations Reserach*, 34:193–202, 1986.

[6] B.-L. Chen and K.-W. Lih. Equitable coloring of trees. *Journal of Combinatorial Theory*, 61:83–87, 1994.

[7] B.-L. Chen and K.-W. Lih. Equitable and m-bounded coloring of split graphs. *Lecture Notes in Computer Science*, 1120:1–5, 1996.

[8] M. Chiarandini and T. Stützle. An application of iterated local search to graph coloring problem. In *Proceedings of the Computational Symposium on Graph Coloring and its Generalizations*, pages 112–125, 2002.

[9] R. Dorn, J. K. Kao, and P. Galinier. Tabu search for frequency assignment in mobile radio networks. *Journal of Heuristics*, 4:47–62, 1998.

[10] P. Erdös. Problem 9. In *Theory of Graphs and its Applications*, page 159. in Fieldler, M, 1964.

[11] C. Fleurent and J. A. Ferland. Genetic and hybrid algorithms for graph coloring. *Annals of Operations Research*, 63:437–461, 1996.

[12] Y. Frota, N. Maculan, T. F. Noronha, and C. C. Ribeiro. A branch-and-cut algorithm for partition coloring. *Networks*, 55:194–204, 2010.

[13] H. Furmanczyk and M. Kubale. The complexity of equitable vertex coloring of graphs. *Journal of Applied Computer Science*, 13:95–107, 2005.

[14] P. Galinier and J. K. Hao. Hybrid evolutionary algorithms for graph coloring. *Journal of Combinatorial Optimization*, 3:379–397, 1999.

[15] P. Galinier and A. Hertz. A survey of local search methods for graph coloring. *Computers and Operations Research*, 33:2547–2562, 2006.

[16] M. Garey and D. S. Johnson. *Computers and intractability: A guide to the theory of NP-completeness*. W. H. Freeman and Company, 1979.

[17] A. Hajnal and E. Szemerédi. *Proof of a conjecture of P. Erdös*, pages 601–623. Combinatorial Theory and its Application. North-Holland, London, 1970.

[18] J. P. Hart and A. W. Shogan. Semi-greedy heuristics: An empirical study. *Operations Reserach Letters*, 6:107–114, 1987.

[19] A. Hertz and D. D. Werra. Using tabu search techniques for graph coloring. *Computing*, 39:345–351, 1987.

[20] D. S. Johnson, C. R. Aragon, L. A. McGeoch, and C. Schevon. Optimization by simulated annealing: An experimental evaluation. part i, graph partitioning. *Operations Research*, 37:865–892, 1989.

[21] H. A. Kierstead and A. V. Kostochka. An ore-type theorem on equitable coloring. *Journal of Combinatorial Theory*, 44:226–234, 2008.

[22] M. Kubale. *Graph Colorings*. American Mathematical Society, 2004.

[23] M. Laguna and R. Martí. A grasp for coloring sparse graphs. *Combinatorial Optimization and Application*, 19:165–178, 2001.

[24] F. T. Leighton. A graph coloring algorithm for large scheduling problems. *Journal of Research of the National Bureau of Standards*, 84:79–100, 1979.

[25] A. Lozano, C. Friedmann, and S. Jurkiewicz. Coloração total equilibrada de grafos - um modelo para redes de interconexão. *Pesquisa Operacional*, 28:161–171, 2008.

[26] D. Matula and L. L. Beck. Smallest last ordering and clustering and graph coloring algorithms. *Journal of ACM*, 30:417–427, 1983.

[27] I. Méndez-Díaz, G. Nasini, and D. Severin. A polyhedral approach for the graph equitable coloring problem. In *Proceedings of the VI ALIO/EURO Workshop on Applied Combinatorial Optimization*, 2008. Buenos Aires, Argentina.

[28] W. Meyer. Equitable coloring. *American Mathematical Monthly*, 80:920–922, 1973.

[29] S. Minton, M. D. Johnston, A. B. Philips, and P. Laird. Minimizing conflicts: A heuristic repair method for constraint satisfaction and scheduling problems. *Artificial Intelligence*, 52:161–205, 1992.

[30] L. Paquete and T. Stützle. An experimental investigation of iterated local search for coloring graphs. In S. Cagnoni, J. Gottlieb, E. Hart, M. Middendorf, and G. Raidl, editors, *Proceedings of the Applications of evolutionary computing on Evoworkshops*, volume 2279, pages 122–131. Springer-Verlag, 2002.

[31] S. Touhami. *Optimization problems in cellular Networks*. PhD thesis, John Molson School of Business, Concordia University, Montreal, 2004.