

On feasible and infeasible search for equitable graph coloring

Wen Sun

Université d'Angers
2 Bd Lavoisier, 49045 Angers, France
sun@info.univ-angers.fr

Jin-Kao Hao*

Université d'Angers
2 Bd Lavoisier, 49045 Angers, France
hao@info.univ-angers.fr

Xiangjing Lai

Nanjing University of Posts and Telecommunications
66 Xinmofan Road, Nanjing 210023, China
laixiangjing@gmail.com

Qinghua Wu

Huazhong University of Science and Technology
1037 Luoyu Road, Wuhan 430074, China
qinghuawu1005@gmail.com

ABSTRACT

An equitable legal k -coloring of an undirected graph $G = (V, E)$ is a partition of the vertex set V into k disjoint independent sets, such that the cardinalities of any two independent sets differ by at most one (this is called the equity constraint). As a variant of the popular graph coloring problem (GCP), the equitable coloring problem (ECP) involves finding a minimum k for which an equitable legal k -coloring exists. In this paper, we present a study of searching both feasible and infeasible solutions with respect to the equity constraint. The resulting algorithm relies on a mixed search strategy exploring both equitable and inequitable colorings unlike existing algorithms where the search is limited to equitable colorings only. We present experimental results on 73 DIMACS and COLOR benchmark graphs and demonstrate the competitiveness of this search strategy by showing 9 improved best-known results (new upper bounds).

KEYWORDS

Equitable coloring; tabu search; heuristics; penalty-based fitness function.

ACM Reference format:

Wen Sun, Jin-Kao Hao, Xiangjing Lai, and Qinghua Wu. 2017. On feasible and infeasible search for equitable graph coloring. In *Proceedings of GECCO '17, Berlin, Germany, July 15-19, 2017*, 8 pages. DOI: <http://dx.doi.org/10.1145/3071178.3071267>

1 INTRODUCTION

Given an undirected graph $G = (V, E)$ with the vertex set V and the edge set $E \subset V \times V$, an independent set of G is a subset of V such that any pair of its vertices are not linked by an edge of E . An equitable legal k -coloring of G is a partition of the vertex set V into k disjoint independent sets $\{V_1, V_2, \dots, V_k\}$ such that $||V_i| - |V_j|| \leq 1, i \neq j, 1 \leq i, j \leq k$. This last constraint is called

*Corresponding author, also affiliated with the Institut Universitaire de France.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

GECCO '17, Berlin, Germany

© 2017 ACM. 978-1-4503-4920-8/17/07...\$15.00

DOI: <http://dx.doi.org/10.1145/3071178.3071267>

the equity constraint of a coloring. The equitable coloring problem (ECP) in graphs involves finding an equitable legal k -coloring with k minimum for general graphs. This minimum k is called the equitable chromatic number of G and denoted by $\chi_e(G)$.

As a variant of the conventional graph coloring problem (GCP), the decision version of the ECP is NP-complete. This can be proved by a straightforward reduction from graph coloring to equitable coloring by adding sufficiently many isolated vertices to a graph and testing whether the graph has an equitable coloring with a given number of colors [8]. The ECP model has a number of practical applications related to garbage collection [25], load balancing [3], timetabling [16], scheduling [7, 15, 24] and so on.

Much effort has been devoted to theoretical studies of the ECP. For example, Meyer conjectured that $\chi_e(G) \leq \Delta(G)$ for any connected graph except the complete graphs and the odd circuits, where $\Delta(G)$ is the maximum vertex degree of G [24]. This conjecture has been proved to be true for trees and graphs with $\Delta(G) = 3$ [6], connected bipartite graphs [20], graphs with the average degree at most $\Delta/5$ [18] and outerplanar graphs [17]. Bodlaender and Fomin [4] identified that the ECP can be solved in polynomial time for graphs with bounded treewidth. Furmańczyk and Kubale investigated the computational complexity of the ECP for some special graphs [9]. Yan and Wang discussed the ECP for kronecker products of the complete multipartite graphs and complete graphs [26].

From a perspective of solution methods for the ECP in the general case, several exact algorithms have been proposed. Specifically, Bahiense et al. presented a branch-and-cut algorithm based on a formulation by representatives [1] and showed computational results only on a set of small random instances (with 60 vertices). Méndez-Díaz et al. investigated a polyhedral approach [22] and a Dsatur-based algorithm [23] and presented computational results for a subset of benchmark instances from the DIMACS and COLOR competitions.

Given the computational challenge of the ECP, exact algorithms suffer inevitably from an exponential time complexity and thus are only applicable to graphs of limited sizes (typically with less than 150 vertices). To handle larger graphs, heuristic algorithms are often used to find sub-optimal solutions in a reasonable time frame. The first heuristics are based on greedy constructive principles [8]. More recently, two powerful heuristics were proposed, which are based on the tabu search method: TabuEqCol [21] and BITS [19]. TabuEqCol is a straightforward adaptation of the well-known TabuCol algorithm designed for the classical graph coloring problem

[10, 14]. BITS improves TabuEqCol by embedding a backtracking scheme under the iterated local search framework.

We observe that unlike the popular graph coloring problem for which many heuristic algorithms have been proposed, research on heuristics for the ECP is quite limited and is still in its infancy. In particular, one important feature of the problem identified by its equity constraint is not explicitly explored by the existing algorithms, which visit equity-feasible solutions only. On the other hand, it is well known that for constrained optimization problems (like the ECP), allowing a controlled exploration of infeasible solutions may facilitate transitions between structurally different solutions and help discover high-quality solutions that are difficult to locate if the search is confined to the feasible region [12].

In this work, we present a feasible and infeasible search algorithm for the ECP which enlarges the search to include equity-infeasible solutions. To prevent the search from going too far away from the feasible boundary, we devise an extended penalty-based fitness function which is used to guide the search for an effective examination of candidate solutions. We show computational results on a set of 73 benchmark graphs from the DIMACS and COLOR competitions to assess the interest of the proposed approach. These results include especially 9 improved best solutions (new upper bounds) which can be used to assess other algorithms for the ECP in the future.

The remainder of the paper is organized as follows. Section 2 introduces some preliminary definitions. Section 3 is dedicated to the description of the proposed algorithm. Section 4 presents computational results and comparisons with state-of-the-art algorithms. Section 5 analyzes the impact of some key components of the proposed algorithm. Conclusions and future work are discussed in the last section.

2 BASIC DEFINITIONS

We introduce the following basic definitions which are useful for the description of the proposed approach, where $G = (V, E)$ is a given graph.

Definition 2.1. A candidate coloring of G is any partition of the vertex set V into k subsets V_1, V_2, \dots, V_k , where each V_i is called a color class.

Definition 2.2. A legal coloring is a conflict-free coloring composed of independent sets, i.e., any pair of vertices of any color class are not linked by an edge in E . Otherwise, it is an illegal or conflicting coloring.

Definition 2.3. An equitable coloring or equity-feasible solution is any candidate coloring satisfying the equity constraint, i.e., the cardinalities of any two color classes differ by at most one. Otherwise, it is an equity-infeasible solution.

3 GENERAL APPROACH

The equitable coloring problem (ECP) involves finding the smallest number of colors k such that an equitable legal k -coloring exists for a given graph G . Like for the conventional GCP [11], the ECP can be approximated by finding a series of equitable legal k -colorings for decreasing k values. To seek an equitable legal k -coloring for a given k , one typically explores the space of equity-feasible colorings

while minimizing a fitness function f which counts the number of conflicting edges [19, 21]. The ECP problem with a given k is called the k -ECP problem.

This study follows this general approach of solving a series of k -ECP problems. However for each fixed k , our algorithm explores candidate solutions which include both equity-feasible and equity-infeasible colorings. For this, our feasible and infeasible search algorithm (FISA) introduces an extended fitness function F which is employed to measure the quality of any candidate solution.

The proposed FISA algorithm is composed of two search phases (see Sections 3.1 and 3.2). The first phase examines only the space of equity-feasible colorings to seek a legal (i.e., conflict-free) k -coloring. If a legal k -coloring is found, the k -ECP problem is solved with the current k value and we continue with the new k -ECP problem by setting $k = k - 1$. To be effective, the first phase is based on the basic tabu search procedure of the BITS algorithm [19]. If the first phase fails to find a legal k -coloring with the equity-feasible space, the second phase is invoked to enlarge the search to include equity-infeasible colorings. To explore the enlarged search space, this second phase relies on the extended fitness function F to guide the search process. The infeasible search phase terminates if an equitable coloring is found or if the best solution found so far cannot be improved during 10000 consecutive iterations. The pseudo-code of the FISA algorithm is given in Algorithm 1. The algorithm starts with an initial equity-feasible solution which is generated with a simple greedy heuristic presented in [19]. In the next sections, we explain the search strategies of both phases of the FISA algorithm.

Algorithm 1 Main Scheme of the FISA algorithm for the ECP

```

1: Input: Graph  $G$ ;  $k$  colors
2: Output: An equitable legal  $k$ -coloring if found
3:  $s_0 \leftarrow \text{greedy\_initial}(G, k)$  /* Generate an initial equity-feasible
    $k$ -coloring */
4:  $s_1 \leftarrow \text{feasible\_search}(s_0)$  /* Section 3.1 */
5: if  $f(s_1) = 0$  then
6:   return( $s_1$ ) and stop
7: end if
8: while stopping condition is not met do
9:    $s_2 \leftarrow \text{infeasible\_search}(s_1)$  /* Section 3.2 */
10:  if  $F(s_2) = 0$  then
11:    return( $s_2$ ) and stop
12:  end if
13:   $s_1 \leftarrow \text{perturbation}(s_2)$ 
14: end while

```

3.1 Searching equity-feasible solutions [19]

The first phase of the proposed FISA algorithm searches the space of candidate solutions which verifies the equity constraint and tries to find a legal k -coloring. This is achieved by minimization of the number of conflicting edges of candidate equitable k -colorings, an edge is conflicting if its endpoints belong to the same color class.

3.1.1 Equity-feasible space and fitness function. We define the equity-feasible space Ω_k to be the set of all candidate colorings verifying the equity constraint. Formally, Ω_k is given by

$$\Omega_k = \{\{V_1, V_2, \dots, V_k\} : ||V_i| - |V_j|| \leq 1, \cup_{i=1}^k V_i = V, V_i \cap V_j = \emptyset\} \quad (1)$$

where $i \neq j, 1 \leq i, j \leq k$.

To assess the quality of a candidate solution s in Ω_k , the evaluation or fitness function counts the number of conflicting edges in the color classes of s . Specifically, let $s = \{V_1, V_2, \dots, V_k\} \in \Omega_k$ be a candidate solution, let $C(V_i)$ denote the set of conflicting edges with both endpoints in V_i . The fitness function f (which is to be minimized) is given by

$$f(s) = \sum_{i=1}^k |C(V_i)| \quad (2)$$

Therefore, a solution s with $f(s) = 0$ is an equitable legal k -coloring satisfying both the equity and coloring constraints. When such a solution is found, the associated k -ECP problem is solved.

The feasible search phase of the FISA algorithm uses this fitness function to guide its search process to visit solutions of Ω_k in order to obtain a solution s with $f(s) = 0$.

3.1.2 Move operators to explore space Ω_k . To explore the space Ω_k , the feasible search phase applies two move operators to generate neighboring solutions from the current solution. Let $s = \{V_1, V_2, \dots, V_k\}$ be the current solution. Let $C(s)$ denote the set of conflicting vertices involved in the conflicting edges of s .

- (1) **One-move operator:** It transfers a conflicting vertex v from its current color class V_i to a different color class V_j ensuring that the equity constraint is always respected, i.e., $|V_i| > \lfloor \frac{n}{k} \rfloor, |V_j| < \lceil \frac{n}{k} \rceil$. Let $\langle v, V_i, V_j \rangle$ denote such a move. We use $s \oplus \langle v, V_i, V_j \rangle$ to denote the neighboring solution generated by applying the move to s . Then the neighborhood N_1 induced by this move operator contains all possible solutions obtained by applying “one-move” to s , i.e.,

$$N_1(s) = \{s \oplus \langle v, V_i, V_j \rangle : v \in V_i \cap C(s)\} \quad (3)$$

where $1 \leq i, j, \leq k, i \neq j, |V_i| > \lfloor \frac{n}{k} \rfloor, |V_j| < \lceil \frac{n}{k} \rceil$.

Note that the one-move operator is not applicable if $\lfloor \frac{n}{k} \rfloor = \lceil \frac{n}{k} \rceil$. In this case, the neighborhood N_1 is empty.

- (2) **Swap operator:** It exchanges a conflicting vertex v of color class V_i with another vertex u of color class V_j ($i \neq j$). Let $swap(v, u)$ denote such a move. The neighborhood N_2 induced by the swap operator is composed of all possible solutions obtained by applying “swap” to s (recall that $C(s)$ is the set of conflicting vertices of s).

$$N_2(s) = \{s \oplus swap(v, u) : v \in V_i, u \in V_j, i \neq j, \{v, u\} \cap C(s) \neq \emptyset\} \quad (4)$$

Since this operator does not change the cardinality of any color class, a neighboring solution generated by this operator is always equity-feasible (given that the current solution is an equitable k -coloring).

3.1.3 Exploration of the space Ω_k . Starting from an equitable (conflicting) k -coloring of Ω_k , the first phase of FISA iteratively improves the solution according to the tabu search method [13]. Specifically, the basic tabu search procedure (TS⁰) described in [19] is applied to find a conflict-free k -coloring. At each iteration, a best admissible candidate solution is taken among the neighboring solutions of N_1 and N_2 to replace the current solution. The underlying move ($\langle v, V_i, V_j \rangle$ for one-move or $swap(v, u)$) is recorded in the so-called tabu list in order to forbid the reverse move for a fixed number of next iterations. This tabu search process continues until either a solution s with $f(s) = 0$ is found in which case, the current k -ECP problem is solved, or the current solution is not improved during a fixed number of consecutive iterations in which case the FISA algorithm moves to the second search phase.

3.2 Searching equity-infeasible solutions

When the first phase fails to identify an equitable legal k -coloring within the equity-feasible space Ω_k , the FISA algorithm invokes the second phase to explore an enlarged space Ω_k^+ including both equity-feasible and equity-infeasible solutions.

3.2.1 Equity-infeasible space and extended fitness function. The enlarged search space Ω_k^+ explored by the second phase contains all possible partitions of the vertex set V into k disjoint subsets as follows.

$$\Omega_k^+ = \{\{V_1, V_2, \dots, V_k\}, \cup_{i=1}^k V_i = V, V_i \cap V_j = \emptyset\} \quad (5)$$

where $i \neq j, 1 \leq i, j \leq k$.

We note that this enlarged search subsumes the equity-feasible space Ω_k and additionally includes the equity-infeasible solutions.

To evaluate the quality of the solutions of Ω_k^+ , we devise an extended penalty-based fitness function F . For this purpose, we first introduce some notations.

Let $W^+ = \lceil n/k \rceil$ and $W^- = \lfloor n/k \rfloor$, which represent respectively the theoretical cardinality of the largest and smallest color classes in an equitable k -coloring. Then for an equitable k -coloring $s = \{V_1, V_2, \dots, V_k\}$, $W^- \leq |V_i| \leq W^+$ ($i = 1, \dots, k$) holds. Let $s = \{V_1, V_2, \dots, V_k\}$ be a candidate solution in Ω_k^+ , we define the penalty ρ_i ($i = 1, \dots, k$) for each color class V_i of the solution s to be the gap between $|V_i|$ and the theoretical cardinalities as follows.

$$\rho_i = \begin{cases} |V_i| - W^+, & \text{if } |V_i| \geq W^+ \\ W^- - |V_i|, & \text{if } |V_i| \leq W^- \end{cases} \quad (6)$$

Then we define our extended fitness function F (to be minimized) as a linear combination of the basic fitness function f (Equation (2)) and a penalty function as follows.

$$F(s) = \sum_{i=1}^k |C(V_i)| + \varphi \sum_{i=1}^k \rho_i \quad (7)$$

where $C(V_i)$ is the set of conflicting edges in color class V_i and φ (a parameter with $\varphi \geq 1$) is the penalty coefficient which is used to control the importance given to the penalty function (see Section 5.1 for an analysis of φ). According to this definition, a candidate solution violating strongly (weakly) the equity constraint will be

penalized more harshly (slightly). Since in general the number of conflicting edges (the first term) of F decreases quickly when the search progresses, the penalty term has the desirable property of helping the search process to avoid infeasible solutions which are too far from the feasibility boundaries. Note that the penalty term of an equitable coloring equals 0.

Therefore, a partition $s \in \Omega_k^+$ with $F(s) = 0$ corresponds to an equitable and legal k -coloring, i.e., satisfying both the equity and coloring constraints and is thus a solution to the k -ECP problem.

3.2.2 Move operators to explore the space Ω_k^+ . To explore the search space Ω_k^+ , the infeasible search phase also applies two move operators to generate neighboring solutions. Let $s = \{V_1, V_2, \dots, V_k\}$ be the current solution. Let $C(s)$ denote the set of conflicting vertices of s , i.e., the vertices involved in a conflicting edge.

- (1) One-move operator: Like for the first search phase, this operator displaces a conflicting vertex v from its current color class V_i to another color class V_j . However, the equity constraint is no more considered. This leads to the following enlarged neighborhood.

$$N_1^+(s) = \{s \oplus \langle v, V_i, V_j \rangle : v \in V_i \cap C(s), 1 \leq i, j \leq k, i \neq j\} \quad (8)$$

Clearly N_1^+ is bounded by $O(|C(s)| \times k)$ in size. To effectively calculate the move gain which identifies the change in the fitness function F (Equation (7)), we adapt the fast incremental evaluation technique of [19]. The main idea is to maintain a matrix A of size $n \times k$ with elements $A[v][q]$ recording the number of vertices adjacent to v in color class V_q ($1 \leq q \leq k$). Another $n \times k$ matrix B is maintained with elements $B[v][q]$ representing the penalty value of vertex v assigned to color class q in the current solution. Then, the move gain of each one-move in terms of extended fitness variation can be conveniently computed by

$$\Delta F = A[v][j] - A[v][i] + \varphi(B[v][j] - B[v][i]) \quad (9)$$

where φ is the penalty coefficient used in the extended fitness function F .

Each time a one-move operation involving vertex v is performed, we just need to update a subset of values affected by this move as follows. For each vertex u adjacent to vertex v , $A[u][i] \leftarrow A[u][i] - 1$, and $A[u][j] \leftarrow A[u][j] + 1$. For any vertex w , $B[w][j] \leftarrow \sum_{i=1}^k \rho_i$, $1 \leq j \leq k$. $B[w][j] = B[u][j]$, if w and u belong to the same color class.

- (2) Swap operator: The same swap operator as for the first phase is applied to exchange a pair of vertices (u, v) from different color classes where at least one of them is a conflicting vertex. However, there is an important difference. Since the second search phase operates in the enlarged space Ω_k^+ instead of Ω_k , the equity-feasibility of a neighboring solution fully depends on the current solution. That is, if the current solution is equity-infeasible (equity-feasible), application of swap leads to an equity-infeasible (equity-feasible) solution. The resulting swap-based neighborhood is thus given as follows.

$$N_2^+(s) = \{s \oplus \text{swap}(v, u) : v \in V_i, u \in V_j, i \neq j, \{v, u\} \cap C(s) \neq \emptyset\} \quad (10)$$

where $C(s)$ is the set of conflicting vertices of s . Notice that the swap operation has no impact on the penalty value of the neighboring solution and can only change the number of conflicting edges. Then the fitness gain of a swap operation can be computed by

$$\Delta F = A[u][i] - A[u][j] + A[v][j] - A[v][i] - 2e_{v,u} \quad (11)$$

where $e_{v,u} = 1$ if v and u are adjacent vertices, otherwise $e_{v,u} = 0$.

3.2.3 Exploration of the enlarged space Ω_k^+ . To explore the enlarged space Ω_k^+ , we apply again the tabu search method. Specifically, each iteration of tabu search selects the best admissible solutions among the neighboring solution of N_1^+ and N_2^+ . The procedure makes transitions between various k -coloring while minimizing the extended fitness function F with the purpose of attaining a solution s with $F(s) = 0$.

3.2.4 Perturbation of infeasible search. The tabu list used by the equity-infeasible exploration phase helps the search process to go beyond some local optima. Yet, this mechanism may not be sufficient to escape deep traps. To overcome this problem, we apply a perturbation procedure inspired by the procedure of [19]. This operator follows the perturbation scheme of breakout local search [2] and combines directed and random applications of the one-move and swap operators. To avoid a too strong deterioration of the perturbed solution, a directed perturbation move takes into consideration the fitness variation and performs the most favorable move (i.e., deteriorating the solution the least). In contrary, a random perturbation performs a one-move or swap operation without considering the fitness deterioration. To combine these two types of perturbations, the number of performed moves dynamically varies in an adaptive way while the application of each type of perturbation is determined probabilistically. The resulting solution from the perturbation procedure is then used as the new starting solution of the next round of the infeasible search phase.

4 EXPERIMENTAL RESULTS AND COMPARISONS

In this section, we assess the performance of the proposed FISA algorithm on the set of 73 benchmark instances which are commonly used in the literature and were initially proposed for the DIMCAS and COLOR competitions for graph coloring problems^{1,2}.

4.1 Experiment settings

The proposed algorithm was coded in C++ and compiled by GNU g++ 4.1.2 with -O3 flag (option). The experiments were conducted on a computer with an Intel Xeon E5-2670 processor (2.5 GHz and 2 GB RAM) running Ubuntu 12.04. When solving the DIMACS

¹<http://www.dimacs.rutgers.edu/>

²<http://www.cs.hbg.psu.edu/txn131/graphcoloring.html/>

Table 1: Comparative results of FISA with state-of-the-art algorithms on the 73 benchmark instances.

Instance	V	LB [22, 23]	UB [22, 23]	TabuEqCol [21]	BITS [19]		FISA				$\Delta(k_1)$	$\Delta(k_{best})$	
				k_1	$k_1(pre)$	$k_{best}(pre)$	k_1	k_{best}	k_{avg}	SR			t_{avg}
DSJC125.1.col	125	5	5	5*	5*	5*	5*	5*	5	20/20	0.617	0	0
DSJC125.5.col	125	9	18	18	17	17	17	17	17	20/20	428.281	0	0
DSJC125.9.col	125	43	45	45	44	44	44	44	44	20/20	0.094	0	0
DSJC250.1.col	250	5	8	8	8	8	8	8	8	20/20	3.619	0	0
DSJC250.5.col	250	12	32	32	32	29	29	29	29.35	13/20	5235.95	-3	0
DSJC250.9.col	250	63	83	83	72	72	72	72	72	20/20	892.236	0	0
DSJC500.1.col	500	5	13	13	13	13	13	13	13	20/20	3.569	0	0
DSJC500.5.col	500	13	62	63	57	56	53	52	53.25	1/20	8197.02	-4	-4
DSJC500.9.col	500	101	148	182	130	129	131	130	131	3/20	6269.63	1	1
DSJR500.1.col	500	12	12	12*	12*	12*	12*	12*	12	20/20	0.38	0	0
DSJR500.5.col	500	120	131	133	126	126	126	126	126.5	10/20	4459.5	0	0
DSJC1000.1.col	1000	5	22	22	22	21	21	21	21	20/20	1866.63	-1	0
DSJC1000.5.col	1000	15	112	112	112	101	98	95	96.1	2/20	15698.6	-14	-6
DSJC1000.9.col	1000	126	268	329	254	252	253	252	252.2	16/20	2240.02	-1	0
R125.1.col	125	-	-	-	5	5	5	5	5	20/20	0.0025	0	0
R125.5.col	125	-	-	-	36	36	36	36	36	20/20	0.6745	0	0
R250.1.col	250	-	-	-	8	8	8	8	8	20/20	0.0045	0	0
R250.5.col	250	-	-	-	67	66	67	66	66.9	2/20	3041.22	0	0
R1000.1.col	1000	-	-	-	20	20	20	20	20	20/20	2.244	0	0
R1000.5.col	1000	-	-	-	269	250	251	250	250.45	11/20	11564.2	-18	0
le450_5a.col	450	5	5	5	5*	5*	5*	5*	5	20/20	30.1971	0	0
le450_5b.col	450	5	5	7	5*	5*	5*	5*	5	20/20	44.2914	0	0
le450_5c.col	450	-	-	-	5	5	5	5	5	20/20	16.391	0	0
le450_5d.col	450	5	8	8	5*	5*	5*	5*	5	20/20	14.0655	0	0
le450_15a.col	450	15	15	-	15*	15*	15*	15*	15	20/20	2.993	0	0
le450_15b.col	450	15	15	15*	15*	15*	15*	15*	15	20/20	2.4125	0	0
le450_15c.col	450	-	-	-	15	15	15	15	15.2	16/20	553.786	0	0
le450_15d.col	450	15	16	16	15*	15*	15*	15*	15.85	3/20	638.127	0	0
le450_25a.col	450	25	25	-	25*	25*	25*	25*	25	20/20	0.41	0	0
le450_25b.col	450	25	25	25*	25*	25*	25*	25*	25	20/20	0.46	0	0
le450_25c.col	450	-	-	-	26	26	26	26	26	20/20	86.9035	0	0
le450_25d.col	450	25	27	27	26	26	26	26	26	20/20	95.845	0	0
wap01a.col	2368	41	46	46	43	42	42	42	42.95	1/20	4544.77	-1	0
wap02a.col	2464	40	44	44	42	41	42	41	41	2/20	2538.33	0	0
wap03a.col	4730	40	50	50	46	45	46	45	45.7	6/20	20201.8	0	0
wap04a.col	5231	-	-	-	46	44	46	44	44.45	11/20	15614.2	0	0
wap05a.col	905	-	-	-	50	50	50	50	50	20/20	99.2625	0	0
wap06a.col	947	-	-	-	42	41	42	41	41.9	2/20	9340.42	0	0
wap07a.col	1809	-	-	-	43	43	43	43	43.05	19/20	4077.71	0	0
wap08a.col	1870	-	-	-	43	43	43	43	43.1	10/20	4872.74	0	0
flat300_28_0.col	300	11	36	36	35	34	33	32	33.6	1/20	4910.48	-2	-2
flat1000_50_0.col	1000	-	-	-	112	101	96	94	94.7	6/20	17321.4	-16	-7
flat1000_60_0.col	1000	-	-	-	112	101	97	94	94.8	5/20	10488.8	-15	-7
flat1000_76_0.col	1000	14	112	112	112	102	98	94	95.15	2/20	15246.4	-14	-8
latin_square_10.col	900	90	130	130	129	113	105	104	104.55	10/20	12666.2	-24	-9
C2000.5.col	2000	-	-	-	202	201	198	183	183.4	13/20	19702.3	-4	-18
C2000.9.col	2000	-	-	-	504	502	503	493	495.21	2/20	21163.9	-1	-9
mulsol.i.1.col	197	49	49	50	49*	49*	49*	49*	49	20/20	44.3365	0	0
mulsol.i.2.col	188	34	39	48	36	36	36	36	36.95	2/20	1914.22	0	0
fpsol2.i.1.col	496	65	65	78	65*	65*	65*	65*	65	20/20	1723.52	0	0
fpsol2.i.2.col	451	47	47	60	47*	47*	47*	47*	47.2	17/20	2357.15	0	0
fpsol2.i.3.col	425	55	55	79	55*	55*	55*	55*	55	20/20	1310.01	0	0
inithx.i.1.col	864	54	54	66	54*	54*	54*	54*	56.9	7/20	3356.31	0	0
inithx.i.2.col	645	30	93	93	36	36	36	36	38.8	5/20	3275.5	0	0
inithx.i.3.col	621	-	-	-	38	37	38	37	39.85	4/20	2891.78	0	0
zeroin.i.1.col	211	49	49	51	49*	49*	49*	49*	49.6	8/20	1088.94	0	0
zeroin.i.2.col	211	36	36	51	36*	36*	36*	36*	36	20/20	123.876	0	0
zeroin.i.3.col	206	36	36	49	36*	36*	36*	36*	36	20/20	129.445	0	0
myciel6.col	95	7	7	7*	7*	7*	7*	7*	7	20/20	0.0035	0	0
myciel7.col	191	8	8	8*	8*	8*	8*	8*	8	20/20	0.0185	0	0
4_FullIns_3.col	114	7	7	7	7*	7*	7*	7*	7	20/20	0.0005	0	0
4_FullIns_4.col	690	6	8	8	8	8	8	8	8	20/20	0.12	0	0
4_FullIns_5.col	4146	6	9	9	9	9	9	9	9	20/20	0.12	0	0
1_Insertions_6.col	607	3	7	7	7	7	7	7	7	20/20	0.1655	0	0
2_Insertions_5.col	597	3	6	6	6	6	6	6	6	20/20	0.056	0	0
3_Insertions_5.col	1406	3	6	6	6	6	6	6	6	20/20	0.3525	0	0
school1.col	385	15	15	15*	15*	15*	15*	15*	15	20/20	0.932	0	0
school1_nsh.col	352	14	14	14*	14*	14*	14*	14*	14	20/20	1.774	0	0
qg_order40.col	1600	40	40	40*	40*	40*	40*	40*	40	20/20	3.437	0	0
qg_order60.col	3600	60	60	60*	60*	60*	60*	60*	60	20/20	14.534	0	0
ash331GPIA.col	662	4	4	4*	4*	4*	4*	4*	4	20/20	0.7755	0	0
ash608GPIA.col	1216	3	4	4	4	4	4	4	4	20/20	0.249	0	0
ash958GPIA.col	1916	3	4	4	4	4	4	4	4	20/20	10.887	0	0

machine benchmark procedure ‘dfmax.c’³ without compiler optimization flag, the run time on our computer is 0.46, 2.68 and 10.70 seconds for graphs r300.5, r400.5 and r500.5, respectively.

For our comparative study, we use the most recent heuristic algorithms [19, 21] as our references. The TabuEqCol algorithm (2014) [21] was run on an Intel i5 CPU with 750 2.67 GHz and tested under a time limit of 1 hour. The BITS algorithm (2015) [19] was run on an Intel Xeon E5440 CPU with 2.83 GHz and 2 GB RAM and tested under a time limit of 1 hour and a relaxed limit (10^4 seconds for the instances with up to 500 vertices and 2×10^4 seconds for larger instances with more than 500 vertices). As shown in [19], the computational results of the more recent BITS algorithm dominate those of TabuEqCol. We also include the lower and upper bounds reported in [22, 23] which were obtained by exact methods under various test conditions. These bounds provide useful information when they are contrasted with the results (upper bounds) obtained by the compared heuristic algorithms (TabuEqCol, BITS and FISA).

The FISA algorithm requires the tuning of some parameters related to tabu search and the extended fitness function F . Since our tabu search procedures are adaptations of the basic tabu search procedure of [19], we adopted the parameter settings used in the original paper. As to the penalty coefficient φ of the extended fitness function F , we provide an analysis in Section 5.

Following [19, 21], we present a first experiment where we ran our FISA algorithm only once per instance with a cutoff time of 3,600 seconds (1 hour). Like [19], we carried out a second experiment where we ran FISA 20 times to solve each instance under the extended stopping condition – 10^4 seconds for the instances with up to 500 vertices and 2×10^4 seconds for larger instances with more than 500 vertices. We note that our Intel Xeon E5-2670 2.5 GHz processor is slightly slower than those used by the reference algorithms. As a result, our adopted stopping conditions can be considered as reasonable with respect to those used by the reference algorithms. Finally, as shown in [19], the main reference BITS algorithm fully dominates the TabuEqCol algorithm. So the results of BITS have the most significant reference value.

4.2 Comparison with state-of-the-art algorithms

From Table 1, we can see that FISA shows a remarkable performance compared to the TabuEqCol and BITS algorithms under both stopping conditions. Under the 1 hour condition, FISA dominates TabuEqCol (Columns 5 and 8) on the 50 instances tested by both algorithms, by obtaining better results for 30 instances and the same results for the remaining 20 instances. Compared to the most recent BITS algorithm, FISA obtains better solutions for 14 instances (see negative entries in Column $\Delta(k_1)$), the same best results for other 58 instances and one worse result.

When comparing FISA with BITS under the long time condition (the results of TabuEqCol under this condition are unavailable), one observes that FISA also performs very well (Columns 7 and 9). Specifically, FISA improves the best results of BITS for 9 instances (see negative entries in Column $\Delta(k_{best})$) while matching the best results of BITS for other 63 instances. Only in one case, FISA obtains a slightly worse result.

³dfmax: <ftp://dimacs.rutgers.edu/pub/dsj/cliquote/>

Finally, when comparing to the upper bounds obtained by exact algorithms (Column 4), it is clear that the bounds of FISA (Column 9) are much better. If we contrast the current best lower bounds (Column 3) obtained by exact approaches of [22, 23] and the best upper bounds from FISA (Column 9), we observe important gaps for a number of instances. This indicates that both the current best lower and upper bounds can be further improved.

5 ANALYSIS

This section performs additional experiments to analyze the proposed FISA algorithm: the penalty coefficient φ and the perturbation strategy. These experiments were performed on a selection of 26 instances which are relatively difficult according to the results reported in Table 1, i.e., the best-known results of these instances cannot be attained by all algorithms.

5.1 Analysis of the penalty coefficient φ

Table 2: Comparative results of the FISA algorithm with 3 different values of φ on the 26 instances. The best results are in bold.

Instance	k_1^*	$k_{1,\varphi=1}(Avg)$	$k_{1,\varphi=2}(Avg)$	$k_{1,\varphi=3}(Avg)$
DSJC125.5.col	17	17(17)	17(17)	17(17)
DSJC250.1.col	8	8(8)	8(8)	8(8)
DSJC250.5.col	29	29(29.8)	29(29.65)	29(29.75)
DSJC250.9.col	72	72(72.1)	72(72.05)	72(72.05)
DSJC500.5.col	53	53(53.35)	53(53.95)	53(54)
DSJC500.9.col	130	131(131.6)	131(131.4)	130(130.75)
DSJR500.5.col	126	126(126.85)	126(126.65)	126(126.9)
DSJC1000.1.col	21	21(21)	21(21.15)	21(21.15)
DSJC1000.5.col	98	98(99.45)	100(101.6)	101(104.75)
DSJC1000.9.col	253	253(254.35)	254(254.6)	254(254)
R250.5.col	66	67(67)	66(66.9)	67(67)
R1000.5.col	250	251(253.9)	251(255.05)	250(252.85)
flat300_28_0.col	32	33(33.85)	32(33.95)	32(33.85)
flat1000_50_0.col	96	96(97.4)	98(99.9)	99(102.25)
flat1000_60_0.col	97	97(97.9)	98(100.35)	99(102.2)
flat1000_76_0.col	98	98(98.65)	99(100.7)	101(102.95)
latin_square_10.col	105	105(106.1)	114(115.15)	115(116.6)
C2000.5.col	198	198(199.55)	200(201.25)	199(200.85)
multsol1.2.col	36	36(36.95)	36(36.95)	36(36.95)
fpsol2.i.1.col	65	65(65)	65(65.05)	65(65)
fpsol2.i.2.col	47	47(47.2)	47(47.25)	47(47.3)
inithx.i.2.col	36	36(38.8)	37(38.65)	36(39.95)
inithx.i.3.col	37	38(39.85)	37(40.1)	37(39.55)
zeroin.i.1.col	49	49(49.75)	49(49.4)	49(49.6)
zeroin.i.2.col	36	36(36)	36(36)	36(36)
zeroin.i.3.col	36	36(36)	36(36)	36(36)
#Equal	21		16	18
#Worse	5		10	8

This section investigates the influence of the penalty coefficient φ on the performance of the proposed algorithm (Section 3.2, Equation 7). For this purpose, we tested FISA with 3 different values of $\varphi = 1, 2, 3$. We ran 20 times the algorithm with each φ value to solve each selected instance with a cutoff time of 1 hour.

The experimental results are presented in Table 2. The first column shows the names of instances, and the second column indicates the best results (k_1^*) obtained in this experiment. The results of FISA with different φ values are respectively listed in columns 3 to 5 including the best values with the averaged values between parentheses. The rows #Equal and #Worse respectively indicate the number of instances for which each φ values attains an equal and worse result compared to k_1^* . We note that the best results were obtained with $\varphi = 1$. This justifies the setting of this parameter used in our previous experiments.

Table 3: Analysis of the influence of the perturbation on the performance of the FISA algorithm.

Instance	k_{best}^*	FISA		REST	
		k_{best}	k_{avg}	k_{best}	k_{avg}
DSJC125.5.col	17	17	17	17	17
DSJC250.1.col	8	8	8	8	8
DSJC250.5.col	29	29	29.8	29	29.95
DSJC250.9.col	72	72	72.1	72	72.1
DSJC500.5.col	53	53	53.35	54	55.15
DSJC500.9.col	131	131	131.6	131	131.7
DSJR500.5.col	126	126	126.85	126	127.05
DSJC1000.1.col	21	21	21	21	21
DSJC1000.5.col	98	98	99.45	102	102.15
DSJC1000.9.col	253	253	254.35	253	254.3
R250.5.col	67	67	67	67	67.05
R1000.5.col	251	251	253.9	252	255.35
flat1000_50_0.col	96	96	97.4	101	101.5
flat1000_60_0.col	97	97	97.9	101	101.75
flat1000_76_0.col	98	98	98.65	101	101.95
flat300_28_0.col	33	33	33.85	33	34.3
latin_square_10.col	105	105	106.1	105	106.5
C2000.5.col	198	198	199.55	201	201
mulsol.i.2.col	36	36	36.95	37	37.75
fpsol2.i.1.col	65	65	65	77	79.4
fpsol2.i.2.col	47	47	47.25	56	73.05
inithx.i.2.col	36	36	38.8	60	63.85
inithx.i.3.col	38	38	39.05	57	64.85
zeroin.i.1.col	49	49	49.75	55	55.9
zeroin.i.2.col	36	36	36	39	42
zeroin.i.3.col	36	36	36	38	43.05
#Equal		26		14	
#Worse		0		12	

5.2 Impact of the perturbation operation

As shown in Section 3.2.4, the proposed algorithm uses a perturbation strategy to ensure a global diversification within the enlarged search space Ω_k^+ . In order to assess this strategy, we compare it with a traditional restart strategy (denoted as REST), where each restart begins its search with a new equitable k -coloring generated by the greedy procedure mentioned in Section 3. The two algorithms were run 20 times on the 26 selected instances with a time limit of 1 hour per run.

The results of this experiment are shown in Table 3. Column 1 lists the names of instances. Column 2 indicates the best results (k_{best}^*) obtained in this experiment. The best results (k_{best}) and the average results (k_{avg}) of FISA and REST are respectively listed in columns 3 to 6. The rows #Equal and #Worse respectively indicate the number of instances for which FISA and REST attain an equal and worse result compared to k_{best}^* . It is clear that FISA dominates the REST variant by obtaining 12 better results out of the 26 tested instances and no worse result. This experiment confirms thus the interest of the adopted perturbation strategy.

6 CONCLUSIONS

The equitable coloring problem (ECP) is an NP-hard problem with a number of practical applications. In addition to the conventional coloring constraint (i.e., adjacent vertices must receive different colors), a solution of the ECP must satisfy the equity constraint (the cardinalities of the color classes must differ by at most one). In this work, we investigated the benefit of examining both feasible and infeasible solutions with respect to the equity constraint. The resulting algorithm (called FISA) combines an equity-feasible search phase where only equitable colorings are considered and an equity-infeasible search phase where the search is enlarged to include non-equitable solutions. To guide the search procedure (which is based on tabu search), we devised an extended fitness function which uses a penalty to discourage candidate solutions which violates the equity constraint. A perturbation procedure was also used as a means of diversification to help the algorithm to explore new search regions.

We assessed the performance of the FISA algorithm on the set of 73 benchmark instances from DIMACS and COLOR competitions and presented comparative results with respect to state-of-the-art algorithms. The comparisons showed that FISA performs very well by discovering 9 improved best results (new upper bounds) and matching the best-known results for the remaining instances except one case. The new bounds can be used for assessment of other ECP algorithms. This study demonstrates the benefit of the mixed search strategy examining both equity-feasible and equity-infeasible solutions for solving the ECP.

For future work, several directions could be followed. First, the penalty term of the extended fitness function could be improved by introducing adaptive techniques like [5, 12] to enable a strategic oscillation for dynamically transitioning between feasible and infeasible space. Second, other search operators (rather than those used in this work) can be sought to further improve the performance of the search algorithm. Third, the proposed algorithm could be advantageously integrated into a hybrid population-based method (e.g., memetic search, path-linking) as a key intensification component. Finally, the instances tested in this work are based on the conventional DIMACS coloring benchmark graphs. These graphs can be considered as being limited in size with respect to massive graphs obtained from a number of modern applications like complex networks and biological networks. Contrary to DIMACS graphs, these massive graphs are typically very sparse with very low edge density. It would be interesting to investigate the ideas of this work in the context of coloring massive graphs.

ACKNOWLEDGMENTS

The authors would like to thank the anonymous referees for their valuable comments and helpful suggestions. Support for the first author from the China Scholarship Council (2015-2019) is also acknowledged.

REFERENCES

- [1] Laura Bahiense, Yuri Frota, Thiago F. Noronha, and Celso C. Ribeiro. A branch-and-cut algorithm for the equitable coloring problem using a formulation by representatives. *Discrete Applied Mathematics* 164, (2014), 34–46.
- [2] Una Benlic and Jin-Kao Hao. Breakout local search for maximum clique problems. *Computers and Operations Research* 40, 1 (2013), 192–206.

- [3] Jacek Blazewicz, Klaus H. Ecker, Erwin Pesch, Günter Schmidt, and Jan Weglarz. Scheduling computer and manufacturing processes. *Journal of the Operational Research Society* 48, 6 (1997), 659–659.
- [4] Hans L. Bodlaender and Fedor V. Fomin. Equitable colorings of bounded treewidth graphs. In *International Symposium on Mathematical Foundations of Computer Science*. Springer, (2004), 180–190.
- [5] Yuning Chen, Jin-Kao Hao, and Fred Glover. A hybrid metaheuristic approach for the capacitated arc routing problem. *European Journal of Operational Research* 253, 1 (2016), 25–39.
- [6] Bor-Liang Chen, Ko-Wei Lih, and Pou-Lin Wu. Equitable coloring and the maximum degree. *European Journal of Combinatorics* 15, 5 (1994), 443–447.
- [7] Jian-Ya Ding, Shi-ji Song, Jatinder N.D. Gupta, Rui Zhang, Raymond Chiong, and Cheng Wu. An improved iterated greedy algorithm with a Tabu-based reconstruction strategy for the no-wait flowshop scheduling problem. *Applied Soft Computing* 30, (2015), 604–613.
- [8] Hanna Furmańczyk and Marek Kubale. Equitable coloring of graphs. *Contemporary Mathematics* 352, (2004), 35–54.
- [9] Hanna Furmańczyk and Marek Kubale. The complexity of equitable vertex coloring of graphs. *Journal of Applied Computer Science* 13, 2 (2005), 95–106.
- [10] Philippe Galinier and Jin-Kao Hao. Hybrid evolutionary algorithms for graph coloring. *Journal of Combinatorial Optimization* 3, 4 (1999), 379–397.
- [11] Philippe Galinier, Jean-Philippe Hamiez, Jin-Kao Hao, and Daniel Porumbel. Recent advances in graph vertex coloring. In Zelinka I., Abraham A., Snasel V. (Eds.) *Handbook of Optimization*, Springer (2012).
- [12] Fred Glover and Jin-Kao Hao. The case for strategic oscillation. *Annals of Operations Research* 183, 1 (2011), 163–173.
- [13] Fred Glover and Manuel Laguna. *Tabu Search*. Springer. (1997).
- [14] Alain Hertz and Dominique de Werra. Using tabu search techniques for graph coloring. *Computing* 39, 4 (1987), 345–351.
- [15] Sandy Irani and Vitus Leung. Scheduling with conflicts, and applications to traffic signal control. In *Symposium on Discrete Algorithms*, 96, (1996), 85–94.
- [16] Fumio Kitagawa and Hideto Ikeda. An existential problem of a weightcontrolled subset and its application to school timetable construction. *Annals of Discrete Mathematics* 38, (1988), 195–211.
- [17] Alexandr V Kostochka. Equitable colorings of outerplanar graphs. *Discrete Mathematics* 258, 1 (2002), 373–377.
- [18] Alexandr V Kostochka and Kittikorn Nakprasit. On equitable Δ -coloring of graphs with low average degree. *Theoretical Computer Science* 349, 1 (2005), 82–91.
- [19] Xiangjing Lai, Jin-Kao Hao, and Fred Glover. Backtracking based iterated tabu search for equitable coloring. *Engineering Applications of Artificial Intelligence* 46, (2015), 269–278.
- [20] Ko-Wei Lih and Pou-Lin Wu. On equitable coloring of bipartite graphs. *Discrete Mathematics* 151, 1 (1996), 155–160.
- [21] Isabel Méndez Diaz, Graciela Nasini, and Daniel Severin. A tabu search heuristic for the equitable coloring problem. In *International Symposium on Combinatorial Optimization*. Springer, (2014), 347–358.
- [22] Isabel Méndez-Díaz, Graciela Nasini, and Daniel Severin. A polyhedral approach for the equitable coloring problem. *Discrete Applied Mathematics* 164, (2014), 413–426.
- [23] Isabel Méndez-Díaz, Graciela Nasini, and Daniel Severin. A DSATUR-based algorithm for the Equitable Coloring Problem. *Computers & Operations Research* 57, (2015), 41–50.
- [24] Walter Meyer. Equitable coloring. *The American Mathematical Monthly* 80, 8 (1973), 920–922.
- [25] Alan Tucker. 1973. Perfect graphs and an application to optimizing municipal services. *Siam Review* 15, 3 (1973), 585–590.
- [26] Zhidan Yan and Wei Wang. Equitable coloring of Kronecker products of complete multipartite graphs and complete graphs. *Discrete Applied Mathematics* 162, (2014), 328–333.