Cumhuriyet Science Journal
e-ISSN: 2587-246X
ISSN: 2587-2680
Cumhuriyet Sci. J., 41(2) (2020) 493-505
http://dx.doi.org/10.17776/csj.628518

Two lagrangian relaxation based heuristics for vertex coloring problem

Ali İrfan MAHMUTOĞULLARI

Department of Industrial Engineering, TED University, 06420, Ankara/TURKEY

Abstract

Vertex coloring problem is a well-known NP-Hard problem where the objective is to minimize the number of colors used to color vertices of a graph ensuring that adjacent vertices cannot have same color. In this paper, we first discuss existing mathematical formulations of the problem and then consider two different heuristics, namely HEUR-RA and HEUR-RC, based on Lagrangian relaxation of adjacency and coloring constraints. HEUR-RA does not require solving any optimization problem through execution whereas at each iteration of HEUR-RC another NP-Hard problem, maximal weight stable set problem, is solved. We conduct experiments to observe computational performances of these heuristics. The experiments reveal that although it requires longer running times, HEUR-RC outperforms HEUR-RA since it provides lower optimal gaps as well as upper bound information.

Article info

History:
Received: 02.10.2019
Accepted:15.06.2020
Keywords:
Vertex coloring problem, Heuristics, Optimization.

1. Introduction

Let $G = (V, E)$ be a graph where V is set of vertices and E is set of edges. Two vertices $i, j \in V$ are called as *adjacent* if $e = (i, j) \in E$. A *vertex coloring* of a graph G is a mapping $c: V \to C$ such that $c(i) \neq c(j)$ whenever i and j are adjacent. The elements of the set C are called *colors*. A graph G is said to be *k-colorable* if there exist a vertex coloring $c: V \to \{1, \dots, k\}$ and such c is called as *k-coloring* of G. The *vertex coloring problem* (VCP) is to determine *chromatic number*, denoted as $\chi(G)$, of a given graph G where chromatic number of a graph is the smallest number of colors to color it. VCP is an NP-Hard problem according to [1].

Aside from its theoretical importance, VCP has many practical applications in scheduling, timetabling, map coloring etc. For each instance of these problems, we can construct a graph and solve VCP to find the desired solutions. Figure 1 depicts an example graph whose chromatic number is 4.

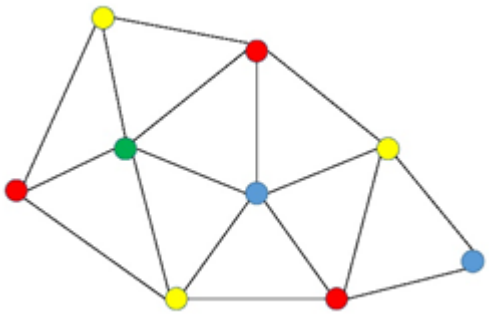

Figure 1. An example of graph G with χ(G)=4.

There are many articles and books (such as [2] from which our notation is adapted) on graph theory that include VCP. However, in this paper, we will provide only studies with operations research perspective. Although first coloring problems were proposed in 19th century, the number of exact approaches of VCP is relatively recent compared to heuristic approaches. First exact approach was proposed by [3]. The idea is based on coloring one vertex at each step and obtain upper and lower bounds for the optimal value. Their approach for VCP can be seen

*Corresponding author. *Email address: ali.mahmutogullari@tedu.edu.tr*

as an analogous to that of Branch-and-Bound method for solving mixed-integer problems. Later, [4,5] suggested improvements for the method such as tie breaking rules and computing initial upper bounds for the optimal value.

[6] proposed a column generation approach for the problem. Their formulation has exponential number of variables; thus, they solve its continuous relaxation with a subset of variables and then solve a pricing problem to detect that if a negative reduced cost exists. The pricing problem is a weighted independent set problem which is initially solved by a heuristic. If the heuristic does not generate a negative reduced cost values, an exact solution of the problem is performed. When LP relaxation of their model has a fractional solution, they perform a branching operation to recover integrality. This column generation approach solves VCP problems up to random instances of 70 vertices.

[7,8] proposed some extra constraints to eliminate symmetry (see Section 2) and proposed several valid inequalities for the models with these new constraints. With these valid inequalities, they could have solved random instances with more than 80 vertices.

Most of the heuristic approaches for VCP are greedy algorithms, that is ranking vertices (or independent sets) based on some criteria, selecting the one with the highest (or smallest rank) and color it either via color which is already used or a new color. [9-11] are examples of studies that proposed greedy algorithms for VCP. Some metaheuristic approaches were also employed for solving VCP. Local search ([12,13]), tabu search ([14]) and simulated annealing algorithms ([15]) enable us to solve VCP problems with 1000 vertices near-optimally.

VCP lies in the intersection of several disciplines such as graph theory, computer science and optimization. Although it attracts attention of researchers from various disciplines, our study is the first to propose heuristic methods based on different Lagrangian relaxations of a VCP. Therefore, the contribution of this paper is proposal of these methods and a discussion of the results obtained from the computational experiments on randomly generated instances.

The rest of the paper is organized as follows: In Section 2, we provide several existing mathematical models of VCP in the literature and give a detailed discussion on these models. In Section 3, we propose two different Lagrangian relaxations of a VCP model by relaxing adjacency and coloring constraints. We also discuss how to solve these relaxed problems. In Section 4, we propose two heuristics based on the two relaxations discussed in Section 3. In Section 5, we conduct a computational experiment on randomly generated VCP instances to observe the quality of bounds obtained from the proposed heuristics and discuss the results of the experiment. Finally, in Section 6, we present some concluding remarks and possible future extensions of the existing study.

2. Mathematical Models

Let $n = |V|$ be the number of vertices. Since any graph with n vertices is n-colorable (because each vertex can be colored with a different color), number of colors to color a graph is at most n. Moreover, we can label vertices of the graph from 1 to n without loss of generality.

Let $y_h = 1$ if color h is used and, 0 otherwise for all $h \in \{1, \dots, n\}$. Also let $x_{ih} = 1$ if vertex i is colored with color h and, 0 otherwise for all $i \in \{1, \dots, n\}$ and for all $h \in \{1, \dots, n\}$. Then, the following mathematical model VCP-1 solves VCP.

$$(VCP-1) \quad \text{minimize} \quad \sum_{h=1}^{n} y_h \tag{1}$$

$$\text{subject to} \quad \sum_{h=1}^{n} x_{ih} = 1 \quad \forall i \in \{1, \dots, n\} \tag{2}$$

$$x_{ih} + x_{jh} \leq y_h \quad \forall (i,j) \in E, \forall h \in \{1, \dots, n\} \tag{3}$$

$$x_{ih} \in \{0,1\} \quad \forall i \in \{1, \dots, n\}, \forall h \in \{1, \dots, n\} \tag{4}$$

$$y_h \in \{0,1\} \quad \forall h \in \{1, \dots, n\} \tag{5}$$

Objective function (1) minimizes number of used colors. Constraints (2) and (3) ensure that each vertex is colored exactly with one color and adjacent vertices are colored with different colors, respectively. Constraints (4) and (5) are domain constraints.

VCP-1 has two major drawbacks as stated in [16]. The first one is the symmetry due to fact that colors are indistinguishable. There are $\binom{n}{k}$ possible ways to choose k colors and once k colors are selected, they can be

permutated in $k!$ ways. Thus, given a solution with k colors, there exists $\binom{n}{k} k!$ equivalent solutions. The second drawback is the weakness of LP relaxation of the model. Indeed, LP relaxation of VCP-1 has an optimal value of 2 by letting $x_{i1} = x_{i2} = 1/2$ for all $i \in \{1, \dots, n\}$, $y_1 = y_2 = 1$, $x_{ih} = 0$ for all $i \in \{1, \dots, n\}, h > 2$ and $y_h = 0$ for $h > 2$. [7,8] tried to overcome these drawbacks. To eliminate symmetry, they proposed adding some extra constraints to VCP-1. The first one is:

$$y_h \geq y_{h+1} \quad h = 1, \dots, n-1. \tag{6}$$

Constraints (6) ensure that color $h+1$ can be used only if color h is used. An alternative constraint to reduce symmetry is using following constraint:

$$\sum_{i=1}^{n} x_{ih} \geq \sum_{i=1}^{n} x_{i,h+1} \quad h = 1, \dots, n-1. \tag{7}$$

Constraints (7) ensure that the number of vertices with color $h+1$ is not greater than the number of vertices with color h.

Finally, following constraints completely eliminate symmetry of VCP-1:

$$x_{ih} = 0 \quad h \geq i+1, \tag{8}$$

$$x_{ih} \leq \sum_{k=h-1}^{i-1} x_{k,h-1} \quad \forall i \in \{2, \dots, n\}, 2 \leq h \leq i-1. \tag{9}$$

Constraints (8) and (9) ensure that independent sets are labeled with the minimum label of its vertices and the vertices in this independent set h are colored with color h.

[6] proposed another formulation of VCP by using independent sets. An *independent (or stable) set* S of a graph $G = (V, E)$ is a subset of V such that whenever $i, j \in S$ then $(i, j) \notin E$. Hence, all vertices in an independent set can be colored with same color. Let $\mathcal{S}$ be collection of all independent sets of G and $z_S = 1$ if all vertices in S are colored with same color, and 0 otherwise for all $S \in \mathcal{S}$. Then, following mathematical model VCP-2 solves VCP.

$$(VCP-2) \quad \text{minimize} \quad \sum_{S \in \mathcal{S}} z_S \tag{10}$$

$$\text{subject to} \quad \sum_{S \in \mathcal{S}: i \in S} z_S \geq 1 \quad \forall i \in \{1, \dots, n\} \tag{11}$$

$$z_S \in \{0,1\} \quad \forall S \in \mathcal{S} \tag{12}$$

Objective function (10) minimizes number of selected independent sets. Constraints (11) ensure that each vertex is included in some independent set and (12) are domain constraints. Although number of variables, that is $|\mathcal{S}|$, is huge, a column generation algorithm can be employed to solve VCP-2. Also, LP relaxation of VCP-2 is as good as LP relaxation of VCP-1 due to [17].

Another mathematical model was proposed in [18] by manipulating constraint (11) in VCP-2. The following *set packing* model VCP-3 solves VCP.

$$(VCP-3) \quad \text{maximize} \quad \sum_{S \in \Omega} (|S|-1) z_S \tag{13}$$

$$\text{subject to} \quad \sum_{S \in \Omega: i \in S} z_S \leq 1 \quad \forall i \in \{1, \dots, n\} \tag{14}$$

$$z_S \in \{0,1\} \quad \forall S \in \Omega \tag{15}$$

where $\Omega = \{S \in \mathcal{S}: |S| \geq 2\}$. The number of used colors is $n - z^*$ where z^* is optimal value of VCP-3. [18] also showed that LP relaxations of VCP-2 and VCP-3 are equivalent in terms of solution value and proposed valid inequalities for VCP-3. VCP-2 and VCP-3 have exponential number of constraints and hence these models cannot be solved directly even for moderate size instances. These models can only be considered for the solution methods such as column generation and branch-and-cut.

3. Lagrangian Relaxations for VCP

In this section, we propose two different Lagrangian relaxations of VCP.

3.1. Relaxing adjacency constraints

Consider VCP-1 with additional constraints (8):

$$(VCP-1a) \quad \text{minimize } \sum_{h=1}^{n} y_h \tag{1}$$

$$\text{subject to } \sum_{h=1}^{n} x_{ih} = 1 \quad \forall i \in \{1,\dots,n\} \tag{2}$$

$$x_{ih} + x_{jh} \le y_h \quad \forall (i,j) \in E, \forall h \in \{1,\dots,n\} \tag{3}$$

$$x_{ih} = 0 \quad h \ge i+1, \tag{8}$$

$$x_{ih} \in \{0,1\} \quad \forall i \in \{1,\dots,n\}, \forall h \in \{1,\dots,n\} \tag{4}$$

$$y_h \in \{0,1\} \quad \forall h \in \{1,\dots,n\} \tag{5}$$

Relaxing adjacency constraints (3) in a Lagrangian manner, we get LR1:

$$(LR1) \quad \text{minimize} \quad \sum_{h=1}^{n} y_h + \sum_{(i,j)\in E} \sum_{h=1}^{n} u_{ijh}\left(x_{ih} + x_{jh} - y_h\right) \tag{16}$$

$$\text{subject to (2), (4), (5) and (8).}$$

LR1 is a relaxation of VCP-1a for any $u \ge 0$. By rearranging terms of the objective function (16), LR1 becomes:

$$(LR1) \quad \text{minimize} \quad \sum_{h=1}^{n} \left(1 - \sum_{(i,j)\in E} u_{ijh}\right) y_h + \sum_{i=1}^{n} \sum_{h=1}^{n} \left(\sum_{e\in\delta(i)} u_{eh}\right) x_{ih} \tag{17}$$

$$\text{subject to (2), (4), (5) and (8).}$$

where $\delta(i) := \{e \in E : e = (i,j) \text{ or } e = (j,i) \text{ for some } j \in \{1,\dots,n\}\}$ is the set of edges adjacent to i.

LR-1 decomposes for each color and vertex. Also, an optimal solution (x^*, y^*) of LR-1 can be found by inspection. Since y variables appear only in objective function, y_h^* takes value 1 only if its objective value coefficient is non-positive for any $h \in \{1,\dots,n\}$. Moreover, constraints (2) and (8) ensures that $\sum_{h=1}^{i} x_{ih} = 1$. Therefore, the color from set $\{1,\dots,i\}$ with smallest objective value coefficient is used to color vertex $i \in \{1,\dots,n\}$. Hence,

$$y_h^* = \begin{cases} 1, & if \quad 1 - \sum_{(i,j)\in E} u_{ijh} \le 0 \\ 0, & otherwise \end{cases} \quad \forall h \in \{1,\dots,n\} \tag{18}$$

$$x_{ih}^* = \begin{cases} 1, & if \quad h = \underset{k\in\{1,\dots,i\}}{\operatorname{argmin}} \sum_{e\in\delta(i)} u_{ek} \\ 0, & otherwise \end{cases} \quad \forall i \in \{1,\dots,n\}, \forall h \in \{1,\dots,n\} \tag{19}$$

3.2. Relaxing coloring constraints

Consider VCP-1: (1)-(5) without anti-symmetry constraints. Relaxing coloring constraints (2) in a Lagrangian manner, we get LR2:

$$(LR2) \quad \text{minimize} \quad \sum_{h=1}^{n} y_h + \sum_{i=1}^{n} \left(1 - \sum_{h=1}^{n} x_{ih}\right) v_i \tag{20}$$

$$\text{subject to } (3) - (5).$$

LR2 is a relaxation of VCP for any choice of Lagrangian multipliers v. By rearranging terms of the objective function (20), LR2 becomes:

$$(LR2) \quad \text{minimize} \sum_{h=1}^{n} \left(y_h - \sum_{i=1}^{n} v_i x_{ih}\right) + \sum_{i=1}^{n} v_i \tag{21}$$

$$\text{subject to } (3) - (5).$$

LR2 decomposes for each color such that:

$$LR2(v) = \sum_{h=1}^{n} LR2_h(v) + \sum_{i=1}^{n} v_i \tag{22}$$

where

$$LR2_h(v) = \text{minimize} \quad y_h - \sum_{i=1}^{n} v_i x_{ih} \tag{23}$$

$$\text{subject to} \quad x_{ih} + x_{jh} \leq y_h \quad \forall (i,j) \in E \tag{24}$$

$$x_{ih} \in \{0,1\}, y_h \in \{0,1\} \quad \forall i \in \{1,\dots,n\} \tag{25}$$

The value of $LR2_h(v)$ depends on whether y_h takes value 0 or 1. If $y_h = 0$, then $x_{ih} = 0, \forall i \in V$ due to constraint (24); otherwise $LR2_h(v)$ takes value of following problem:

$$\text{minimize } 1 - \sum_{i=1}^{n} v_i x_i \tag{26}$$

$$\text{subject to} \quad x_i + x_j \leq 1 \quad \forall (i,j) \in E \tag{27}$$

$$x_i \in \{0,1\}, \quad \forall i \in \{1,\dots,n\}. \tag{28}$$

which can equivalently be restated as:

$$1 - \text{maximize} \sum_{i=1}^{n} v_i x_i \tag{29}$$

$$\text{subject to } (27) \text{ and } (28)$$

Hence,

$$LR2_h(v) = \min\{0, 1 - f(v)\} \tag{30}$$

where $f(v)$ is the objective of the maximum weight stable set problem

$$(MWSSP) \quad \text{maximize} \quad \sum_{i=1}^{n} v_i x_i \tag{31}$$

$$\text{subject to } (27) \text{ and } (28)$$

Let $\tilde{x}_i, \forall i \in \{1,\dots,n\}$ be an optimal solution of MWSSP. Based on the value that minimum in (30) is attained, an optimal solution of LR2 can be found as:

$$y_h^* = \begin{cases} 0, & if \quad f(v) \leq 1 \\ 1, & otherwise \end{cases} \qquad \forall h \in \{1,\dots,n\} \tag{32}$$

$$x_{ih}^* = \begin{cases} 0, & if \quad f(v) \leq 1 \\ \tilde{x}_i, & otherwise \end{cases} \qquad \forall i \in \{1,\dots,n\}, \forall h \in \{1,\dots,n\} \tag{33}$$

Note that in any optimal solution of MWSSP, we have $\tilde{x}_i = 0$ for any $i \in \{1,\dots,n\}$ with $v_i < 0$ since otherwise we can always have feasible solution with a better objective value by changing the value of the variable with a strictly negative coefficient to 0.

4. Heuristics

In this section, we propose two heuristics for VCP based on the Lagrangian relaxations proposed in the previous section. The first **heur**istic is based on the **r**elaxation of **a**djacency constraint and called as **HEUR-RA**. The outline of HEUR-RA is given below.

HEUR-RA
Initialize: Iteration counter $t \leftarrow 1$, Lagrangian multipliers $u_{ijh}^t = 0$ for all $(i,j) \in E$ and $h \in \{1,\dots,n\}$, step-sizes $\{\mu_t\}_{t=1}^{\infty}$, lower bound $LB \leftarrow -\infty$.
while some termination criteria is not met **do**
Compute an optimal solution (x^t, y^t) of the Lagrangian relaxation by (18) and (19).
Update lower bound $$LB_t \leftarrow \sum_{h=1}^{n} y_h^t + \sum_{(i,j)\in E} \sum_{h=1}^{n} u_{ijh}^t \left(x_{ih}^t + x_{jh}^t - y_h^t\right)$$

$LB \leftarrow \max\{LB, LB_t\}$
Update Lagrangian multipliers $u_{ijh}^{t+1} \leftarrow \max\{0, u_{ijh}^t + \mu^t(x_{ih}^t + x_{jh}^t - y_h^t)\}$ for all $(i,j) \in E$ and $h \in \{1, \dots, n\}$
$t \leftarrow t + 1$
end
Return: A lower bound LB and an upper bound UB.

At each iteration of HEUR-RA, an optimal solution of LR1 is found by inspection using equations (18) and (19) and a lower bound for VCP is obtained from this solution. The Lagrangian multipliers are also updated in each iteration. Since adjacency constraints are relaxed in LR1, a solution of the relaxed problem is only an assignment of colors to the vertices. Therefore, it does not provide any information about a feasible coloring and hence an upper bound.

The second **heur**istic is based on **r**elaxation of **c**oloring constraints and hence called as **HEUR-RC**. The outline of HEUR-RC is given below.

HEUR-RC
Initialize: Iteration counter $t \leftarrow 1$, Lagrangian multipliers $v_i^t = 0$ for all $i \in \{1, \dots, n\}$, lower bound $LB \leftarrow -\infty$ and upper bound $UB \leftarrow \infty$.
while some termination criteria is not met **do**
Solve MWSSP problem given with given $v_i^t, i \in \{1, \dots, n\}$ values and compute an optimal solution (x^t, y^t) of the Lagrangian relaxation by (32) and (33). Update lower bound $LB_{t+1} \leftarrow \max\{LB_t, \sum_{h=1}^n y_h^t + \sum_{i=1}^n (1 - \sum_{h=1}^n x_{ih}^t)v_i\}$ $LB \leftarrow \max\{LB, LB_{t+1}\}$
Let $\overline{x} \leftarrow x^t$ and $\overline{y} \leftarrow y^t$ be a coloring which does not necessarily satisfy coloring constraints (2)
For vertices $i \in \{1, \dots, n\}$ such that $\sum_{h=1}^n \overline{x}_{ih} \geq 2$ only keep the color with smallest index, that is, $\overline{x}_{ih} \leftarrow \begin{cases} 1, & if \quad h = \min\{k \in \{1, \dots, n\} : \overline{x}_{ik} = 1\} \\ 0, & otherwise \end{cases}$
For vertices $i \in \{1, \dots, n\}$ such that $\sum_{h=1}^n \overline{x}_{ih} = 0$ assign the color with the smallest index that does not create a conflict, that is, $\overline{x}_{ih} \leftarrow \begin{cases} 1, & if \quad h = \min\{\{1, \dots, n\} \backslash \mathcal{C}\} \\ 0, & otherwise \end{cases}$ where $\mathcal{C} :=$ $\{h : \overline{x}_{i'h} = \overline{x}_{j'h} = 1 \text{ for some} (i', j') \in \delta(i)\}$ is the set of colors that creates a conflict.
Update used colors $y_h^t \leftarrow \begin{cases} 1, & if \quad \exists i \in \{1, \dots, n\} \text{s.t.} x_{ih}^t = 1 \\ 0, & otherwise \end{cases} \quad \forall h \in \{1, \dots, n\}$
Update upper bound $UB_{t+1} \leftarrow \min\{UB_t, \sum_{h=1}^n y_h^t\}$
Update Lagrangian multipliers $u_{ijh}^{t+1} \leftarrow \max\{0, u_{ijh}^t + \mu^t(x_{ih}^t + x_{jh}^t - y_h^t)\}$ for all $(i,j) \in E$ and $h \in \{1, \dots, n\}$
$t \leftarrow t + 1$
end

Return: A feasible coloring, lower and upper bounds.

At each iteration of HEUR-RC, a solution of LR2 is obtained by solving MWSSP as discussed in the previous section. Unlike LR1, the adjacency requirement is maintained in LR2, it is possible to generate a feasible solution to VCP using x_{ih}^* in equation (33). However, some vertices are colored more than once and some of them are not colored since coloring constraint (2) does not appear in LR2. If a vertex has been colored more than once, only the color with the smallest index is maintained and other colors are discarded. If a vertex has not been colored, then any color which does not violate adjacency requirement can be assigned to that vertex. Such color with the smallest index is used to color any uncolored vertex. Therefore, a feasible coloring and hence an upper bound for VCP is obtained. Finally, Lagrangian multipliers are updated.

5. Computational Study

In order to observe the efficiency of the proposed heuristics, some computational experiments are conducted on randomly generated instances. *Random* class on Java 1.6.023 running on NetBeans IDE 6.9.1 is used to generate random values. IBM ILOG CPLEX Optimization Studio 12.6 is used to solve optimization problems on a PC with specifications Intel(R) Core (TM)2 Duo CPU P7450 2 x 2.13GHz. with 4.00 GB RAM. Five instances are generated for each $|V| = n \in \{10,20,30,40\}$ and edge density coefficient $f \in \{0.2,0.5,0.8\}$ pair. Edge density coefficient indicates the probability of existence of an edge between each edge pairs.

The classical subgradient algorithm discussed in [19] is used to solve Lagrangian dual problems with stepsize rule $\mu_t = 1/t$ for HEUR-RA (due to absence of upper bound information) and $\mu_t = \frac{UB - z^t}{||infea||^2}$ for HEUR-RC where $||infea||$ indicates the Euclidean norm of the infeasibility vector of the relaxed constraint, UB indicates current upper bound and z^t is the value of Lagrangian relaxation at iteration t. The algorithm terminates if no improvement is made in last $10n$ iterations in LB so as to prevent memory errors.

Table 1 summarizes optimal values and required CPU seconds for solving VCP-1 with anti-symmetry constraints (8) and (9) of all five instances for each parameter settings (n, f). Although all instances with $n \leq 30$ are able to be solved less than 10 seconds, instances with $n = 40$ require higher CPU times. For the instances with $n > 40$, memory errors occur due to the fact that the Branch-and-Bound tree is quite large.

Required CPU time and optimality gap percentage of HEUR-RA on each instance are summarized in Table 2. Since LR1 can only be solved by inspection, CPU times are under 36 seconds for all instances. On the other hand, lower bounds obtained by LR1 is not tight with average optimality gap of 40.48%. Also, no upper bound information is obtained during execution of HEUR-RA. Therefore, the optimality gap for these instances are cacluated as $Gap\ \% = 100\frac{Obj.value - Lower\ bound}{bower\ Bound}$.

Table 3 summarizes lower and upper bounds, optimality gaps, number of iterations and CPU times of execution o HEUR-RC. For 53 out of 60 instances the heurıstic is able to find optimal solution. These instances are indicated with bolded upper bound values in Table 3. For the instances with $n = 40$ and $f = 0.5$ or 0.8 CPU times of Lagrangian dual problem is smaller than CPU times required to solve VCP-1 with constraints (8) and (9). Also, the average optimality gap of the instances in Table 3 is 2%.

HEUR-RC is also tested with larger instances presented in Table 4. For six out of 20 instances, the heuristic yields optimal solutions (these instances are indicated with bolded upper bound values in Table 4). Also, average gap for the instances in Table 4 is 11.18 %. Although, VCP-1 with constraints (8) and (9) cannot find optimal values for the instances in Table 4, the algorithm figures out reasonable bounds even though CPU times are high. In Tables 3 and 4, the optimality gaps are calculated as $Gap\ \% = 100\frac{Upper\ bound - Lower\ bound}{Lower\ bound}$ since both lower and upper bounds are available in HEUR-RC.

6. Conclusion

In this study, two different Lagrangian relaxations of VCP are investigated and then we propose two heuristics based on these relaxations. First, adjacency constraints (3) in VCP-1a are relaxed in Lagrangian manner. Although corresponding relaxed problem can be solved by easily, corresponding lower bounds are not tight. A possible improvement in lower bound can be obtained by adding inequalities that are valid for VCP-1a to LR1. For example,

$$x_{ih} \leq y_h \quad i \in \{1, \dots, n\}, h \in \{1, \dots, n\} \tag{34}$$

or

$$|K| \leq \sum_{h=1}^{n} y_h \tag{35}$$

where $K \subseteq V$ is a *maximum cardinality clique* of G. Adding (34) and/or (35) to LR1 definitely improves quality of the lower bound. However, solving the relaxed problem with (34) or finding a maximum cardinality clique prohibit color-wise decomposition of LR1. This trade off should be investigated by conducting computational experiments.

Second, coloring constraints (2) are relaxed in Lagrangian manner. The lower bounds obtained by Lagrangian dual problem induced by LR2 are much tighter than the previous relaxation with a computational cost of solving another NP-Hard problem, namely weighted stable set problem, at each iteration. Also, an upper bound can also be obtained by a heuristic whose input is optimal solution of the relaxed problem. Indeed, optimal solution is attained for 58 of 80 instances with average gap is 4%. An improvement for the bound of this relaxation can be achieved solving weighted stable set problem at each iteration more efficiently.

Table 1. Computational performance of VCP-1 with constraints (8) and (9)

	n	10			20			30			40		
#	f	0.2	0.5	0.8	0.2	0.5	0.8	0.2	0.5	0.8	0.2	0.5	0.8
1	Opt. value	3	3	5	4	5	9	4	7	12	5	8	15
	CPU time	0.016	0.016	0.031	0.031	0.109	0.141	0.764	2.293	3.073	14.274	292.516	656.012
2	Opt. value	3	4	7	3	5	10	4	7	12	5	8	15
	CPU time	0.016	0.046	0.015	0.141	0.156	0.156	0.671	3.417	5.819	11.279	1303.49	647.323
3	Opt. value	3	4	5	4	6	10	4	7	12	5	9	15
	CPU time	0.01	0.312	0.374	0.124	0.203	0.063	0.39	3.026	1.7	0.92	830.14	19.032
4	Opt. value	3	5	6	4	6	10	4	8	13	5	8	16
	CPU time	0.01	0.015	0	0.047	0.109	0.046	0.234	0.562	0.749	4.431	505.581	235.139
5	Opt. value	3	3	5	4	5	10	4	7	13	5	9	14
	CPU time	0.01	0.031	0.031	0.094	0.562	0.125	0.92	7.316	0.858	5.054	632.956	3.37
Average Opt. value		0.012	0.084	0.09	0.087	0.228	0.106	0.596	3.323	2.44	7.192	712.937	312.175
Max CPU time		0.016	0.312	0.374	0.141	0.562	0.156	0.92	7.316	5.819	14.274	1303.49	656.012

Table 2. Computational performance of HEUR-RA

	n	10			20			30			40		
#	f	0.2	0.5	0.8	0.2	0.5	0.8	0.2	0.5	0.8	0.2	0.5	0.8
1	Opt. Value	3	3	5	4	5	9	4	7	12	5	8	15
	Lower Bound	2	3	4	3	3	5	3	3	5	3	3	4
	Gap %	33.3	0	20	25	40	44.4	25	57.1	58.3	40	62.5	73.3
	CPU time	0.016	0.078	0.047	1.123	0.405	1.201	8.829	1.607	1.467	25.334	5.179	3.759
2	Opt. Value	3	4	7	3	5	10	4	7	12	5	8	15
	Lower Bound	3	3	4	3	3	6	3	3	5	3	3	5
	Gap %	0	25	42.9	0	40	40	25	57.1	58.3	40	62.5	66.7
	CPU time	0.047	0.016	0.016	1.747	0.156	1.185	9.594	1.779	3.916	11.372	2.231	17.971
3	Opt. Value	3	4	5	4	6	10	4	7	12	5	9	15
	Lower Bound	3	3	4	3	3	4	3	3	4	3	3	4
	Gap %	0	25	20	25	50	60	25	57.1	66.7	40	66.7	73.3
	CPU time	0.109	0.063	0.53	0.842	0.562	0.358	1.186	0.936	0.936	23.291	8.643	4.758
4	Opt. Value	3	5	6	4	6	10	4	8	13	5	8	16
	Lower Bound	3	3	4	3	3	6	3	3	4	3	3	5
	Gap %	0	40	33.3	25	50	40	25	62.5	69.2	40	62.5	68.8
	CPU time	0.249	0.016	0.046	0.39	0.328	0.889	1.778	1.076	0.717	10.608	1.794	9.454
5	Opt. Value	3	3	5	4	5	10	4	7	13	5	9	14
	Lower Bound	3	3	4	3	3	4	3	3	4	3	3	5
	Gap %	0	0	20	25	40	60	25	57.1	69.2	40	66.7	64.3
	CPU time	0.25	0.062	0.016	1.06	1.357	0.219	20.031	1.965	2.745	4.836	10.342	35.256
Avg CPU time		0.134	0.047	0.131	1.032	0.562	0.77	8.284	1.473	1.956	15.088	5.638	14.24
Max CPU time		0.25	0.078	0.53	1.747	1.357	1.201	20.031	1.965	3.916	25.334	10.342	35.256
Average Gap %		6.7	18	27.2	20	44	48.9	25	58.2	64.4	40	64.2	69.3
Max Gap %		33.3	40	42.9	25	50	60	25	62.5	69.2	40	66.7	73.3

Table 3. Computational performance of HEUR-RC

	n	10			20			30			40		
#	f	0.2	0.5	0.8	0.2	0.5	0.8	0.2	0.5	0.8	0.2	0.5	0.8
1	Lower Bound	3	3	5	4	5	9	4	7	12	5	8	14
	Upper Bound	**3**	**3**	**5**	**4**	**5**	**9**	**4**	**7**	**12**	**5**	**8**	**15**
	Gap %	0	0	0	0	0	0	0	0	0	0	0	6.7
	# iter	12	9	16	27	22	39	49	214	299	487	569	49
	CPU time	0.23	0.187	0.234	0.671	0.734	0.84	1.55	20.015	44.444	85.207	210.725	17.394
2	Lower Bound	3	4	7	3	5	10	4	7	12	4	8	15
	Upper Bound	**3**	**4**	**7**	**3**	**5**	**10**	**4**	**7**	**12**	**5**	9	**15**
	Gap %	0	0	0	0	0	0	0	0	0	20	11.1	0
	# iter	9	14	14	15	23	35	53	50	48	1346	1032	854
	CPU time	0.03	0.078	0.39	0.093	0.265	0.47	0.66	4.29	4.103	230.29	333.513	467.298
3	Lower Bound	3	4	5	4	6	10	4	7	12	5	9	15
	Upper Bound	**3**	5	6	**4**	**6**	**10**	**4**	8	**12**	**5**	**9**	**15**
	Gap %	0	20	16.7	0	0	0	0	12.5	0	0	0	0
	# iter	12	207	139	95	69	92	35	757	71	40	26	224
	CPU time	0.19	1.342	0.765	1.545	1.248	1.58	0.34	111.74	7.114	1.731	1.779	72.634
4	Lower Bound	3	5	6	4	6	10	4	8	13	5	8	16
	Upper Bound	**3**	**5**	**6**	**4**	7	**10**	**4**	**8**	**13**	**5**	9	**16**
	Gap %	0	0	0	0	14.3	0	0	0	0	0	11.1	0
	# iter	12	14	22	33	482	38	33	126	418	459	1312	195
	CPU time	0.05	0.078	0.125	0.234	4.103	0.39	0.28	9.142	12.183	55.817	404.166	29.141
5	Lower Bound	3	3	5	4	5	10	4	7	13	5	8	14
	Upper Bound	**3**	**3**	**5**	**4**	**5**	**10**	**4**	**7**	**13**	**5**	10	**14**
	Gap %	0	0	0	0	0	0	0	0	0	0	20	0
	# iter	11	10	11	30	22	44	30	444	107	29	625	585
	CPU time	0.02	0.031	0.046	0.14	0.171	0.62	0.42	65.941	7.114	0.843	177.949	214.594
Avg CPU time		0.1	0.343	0.312	0.537	1.304	0.78	0.65	42.226	14.992	74.777	225.626	160.212
Max CPU time		0.23	1.342	0.765	1.545	4.103	1.58	1.55	111.74	44.444	230.29	404.166	467.298
Average Gap %		0	4	3.3	0	2.9	0	0	2.5	0	4	8.4	1.3
Max Gap %		0	20	16.7	0	14.3	0	0	12.5	0	20	20	6.7

Table 4. Computational performance of HEUR-RC on larger instances					
	n	50			60
#	f	0.2	0.5	0.8	0.2
1	Lower Bound	4	9	16	5
	Upper Bound	**4**	11	19	7
	Gap %	0	18.2	15.8	28.6
	# iter	64	1024	1211	1944
	CPU time	9.02	511.38	474.12	1204.85
2	Lower Bound	5	9	16	7
	Upper Bound	**5**	11	17	8
	Gap %	0	18.2	5.9	12.5
	# iter	553	812	1763	1603
	CPU time	140.65	380.61	680.82	1574.36
3	Lower Bound	4	10	16	8
	Upper Bound	**4**	**10**	19	**8**
	Gap %	0	0	15.8	0
	# iter	66	526	1018	1056
	CPU time	10.17	244.33	411.79	1008.83
4	Lower Bound	5	6	17	6
	Upper Bound	**5**	**6**	19	7
	Gap %	0	0	10.5	14.3
	# iter	236	38	1097	2615
	CPU time	65.83	7.64	462.06	1876.73
5	Lower Bound	5	8	17	6
	Upper Bound	6	11	20	8
	Gap %	16.7	27.3	15	25
	# iter	1153	1268	1040	1862
	CPU time	376.23	584.85	402.54	1755.13
Avg CPU time		120.38	345.76	486.27	1483.98
Max CPU time		376.23	584.85	680.82	1876.73
Average Gap %		3.3	12.7	12.6	16.1
Max Gap %		16.7	27.3	15.8	28.6

Conflicts of interest

There is no conflict of interest.

References

[1] Karp R. M., Reducibility among combinatorial problems. Springer (1972).

[2] Diestel R., Graph theory (graduate texts in mathematics). Springer (2005).

[3] Brown J. R., Chromatic scheduling and the chromatic number problem. *Management Science*, 19 (4) (1972) 456–463.

[4] Brélaz D., New methods to color the vertices of a graph. *Communications of the ACM*, 220 (4) (1979) 251–256.

[5] Sewell E., An improved algorithm for exact graph coloring. *DIMACS series in discrete mathematics and theoretical computer science*, 26 (1996) 359-373.

[6] Mehrotra A. and Trick M. A., A column generation approach for graph coloring. *Informs Journal on Computing*, 80 (4) (1996) 344–354.

[7] Méndez-Diaz I. and Zabala P., A branch-and-cut algorithm for graph coloring. *Discrete Applied Mathematics*, 1540 (5) (2006) 826-847.
[8] Méndez-Diaz I. and Zabala P., A cutting plane algorithm for graph coloring. *Discrete Applied Mathematics*, 1560 (2) (2008) 159-179.
[9] Leighton F. T., A graph coloring algorithm for large scheduling problems. *Journal of research of the national bureau of standards*, 840 (6) (1979) 489–506.
[10] Bollobás B. and Thomason A., Random graphs of small order. *North-Holland Mathematics Studies*, 118 (1985) 47-97.
[11] Culberson J. C. and Luo F., Exploring the k-colorable landscape with iterated greedy. *Cliques, coloring, and satisfiability: second DIMACS implementation challenge*, 26 (1996) 245–284.
[12] Morgenstern C., Distributed coloration neighborhood search. *Discrete Mathematics and Theoretical Computer Science*, 26 (1996) 335–358.
[13] Chiarandini M., Dumitrescu I., and Stützle T., Stochastic local search algorithms for the graph colouring problem. Handbook of approximation algorithms and metaheuristics. Chapman & Hall, CRC (2007).
[14] Hertz A. and de Werra D., Using tabu search techniques for graph coloring. *Computing*, 390 (4) (1987) 345–351.
[15] Johnson D. S., Aragon C. R., McGeoch L. A. and Schevon C., Optimization by simulated annealing: An experimental evaluation; part i, graph partitioning. *Operations research*, 370 (6) (1989) 865–892.
[16] Malaguti E. and Toth P., A survey on vertex coloring problems. *International Transactions in Operational Research*, 170 (1) (2010) 1–34.
[17] Mehrotra A., Constrained graph partitioning: decomposition, polyhedral structure and algorithms, (1992).
[18] Hansen P., Labbé M. and Schindl D., Set covering and packing formulations of graph coloring: algorithms and first polyhedral results. *Discrete Optimization*, 60 (2) (2009) 135–147.
[19] Wolsey L.A., Integer programming. Wiley (1998).