Mathematics Interdisciplinary Research **x** (x) (202x) $xx - xx$

Original Scientific Paper

Simulated Annealing and Integer Linear Programming Approaches to Mutual-Visibility Coloring

Alireza Masoudi, Mostafa Tavakoli * *and Zahra Hamed-Labbafian*

Abstract

Vertex coloring is a fundamental concept in graph theory, where a color is assigned to each vertex, and vertices sharing the same color form a color class. This paper introduces and investigates a specific type of coloring known as mutual-visibility coloring. In this coloring, each color class must satisfy the property that for every pair of vertices within that class, there exists at least one shortest path between them that contains no other vertex from the same class. The primary objective is to find the minimum number of colors required to achieve such a coloring, which is known as the mutual-visibility chromatic number. To solve this problem, this research employs two complementary approaches: a heuristic algorithm based on *simulated annealing*, which is efficient for exploring the solution space in large graphs, and an exact approach using *integer linear programming*, which guarantees finding the optimal solution for graphs of smaller dimensions.

Keywords: Mutual-visibility coloring, Simulated annealing algorithm, Integer linear programming.

2020 Mathematics Subject Classification: 05C15, 05C85, 90C10.

Corrected Proof

How to cite this article
A. Masoudi, M. Tavakoli and Z. Hamed-Labbafian, Simulated annealing and integer linear programming approaches to mutual-visibility coloring, *Math. Interdisc. Res.* **x** (x) (202x) xx–yy.

*Corresponding author (E-mail: m_tavakoli@um.ac.ir)
Academic Editor: Gholam Hossein Fath-Tabar
Received 24 February 2026, Accepted 20 August 2026
DOI: 10.22052/mir.2026.258483.1575

© 2026 University of Kashan
This work is licensed under the Creative Commons Attribution 4.0 International License.

1. Introduction

Let G be a connected graph. *The distance* between two vertices u and v in G, denoted by $d(u, v)$, is defined as the number of edges in a shortest path joining u and v. The diameter of a graph G, denoted $diam(G)$, is the maximum shortest-path distance between any two vertices in G.

Let G be a graph and let $X \subseteq V(G)$. The subset X is said to be a *mutual visibility set* of G if, for every two vertices $u, v \in X$, there exists a shortest (u, v)-path $P_{u,v}$ containing no other vertices of X. This problem attracts significant attention owing to its applicability to many practical problems, including the mobile robot problem; see [1, 2]. This notion was introduced by Di Stefano in [3] and has since been the subject of extensive research. In particular, the computational complexity of the problem was investigated in [3], where its NP-hardness was confirmed. Furthermore, in [4–6], mutual-visibility sets were studied with respect to several graph products. Recently, Hamed-Labbafian et al. [7] extended this line of research by investigating mutual visibility in hierarchical graph products, proposing a genetic algorithm, and presenting several applications of the concept.

Vertex coloring is a fundamental problem in graph theory, aiming to assign colors to the vertices of a graph. Due to numerous practical applications, various extensions of vertex coloring have been proposed. A recent variant, introduced by Klavžar et al. in [8], is known as *mutual-visibility coloring*, where each color class is required to be a mutual-visibility set. Hence, mutual-visibility coloring extends classical vertex coloring by incorporating visibility constraints. The minimum number of colors needed for a mutual-visibility coloring of a graph G, denoted by $\chi_\mu(G)$, is called *the mutual-visibility chromatic number* of G. It has been shown in [9] that computing the chromatic number of the visibility graph is NP-complete. Moreover, an approximation algorithm for this problem was presented in [8].
These complexity results motivate the development of efficient exact and heuristic solution methods for the problem. Since the Mutual-Visibility Coloring problem is NP-hard, we first propose a Simulated Annealing (SA) algorithm, which is the first metaheuristic approach developed for this problem. The algorithm is applicable to arbitrary graph classes and is capable of finding near-optimal solutions within reasonable computational time. Furthermore, to obtain exact solutions and provide a benchmark for evaluating the heuristic, we present the first Integer Linear Programming (ILP) formulation for the mutual-visibility coloring problem. The ILP model provides an exact optimization framework for computing optimal solutions; however, due to the large number of variables and constraints in the formulation, it is primarily suitable for small- and medium-sized graph instances. Finally, extensive computational experiments demonstrate the effectiveness of both approaches and show that the exact solutions obtained by the ILP model provide reliable benchmarks for assessing the performance of the SA algorithm.
In this work, Section 3 presents a metaheuristic algorithm designed to address the problem. Section 4 is devoted to the formulation of a linear programming model. Section 5 reports the computational results and presents a comparative

Corrected Proof

performance analysis of the proposed approaches.

2. Mutual-visibility coloring

Let G be a connected graph. A vertex coloring $\varphi : V(G) \to C$ is called a mutual-visibility coloring of G if, for any two vertices $x, y \in V(G)$, the following condition holds: Vertices x and y receive the same color, i.e., $\varphi(x) = \varphi(y)$, if there exists a shortest x, y-path whose internal vertices are colored with colors different from $\varphi(x)$ and $\varphi(y)$. In particular, if x and y are adjacent, they may receive the same color. The minimum number of colors required in a mutual-visibility coloring of G is called the mutual-visibility chromatic number of G, denoted by $\chi_\mu(G)$.
To illustrate the concept of mutual-visibility coloring, Figure 1 presents two colorings of the $(5, 6)$-fullerene graph (Generalized Petersen graph). Figure 1(a) shows a valid mutual-visibility coloring using four colors. In contrast, Figure 1(b) illustrates an invalid coloring. Although the highlighted vertices receive the same color, every shortest path connecting them contains an internal vertex with the same color. Hence, the mutual-visibility condition is violated, and the coloring is not a mutual-visibility coloring.

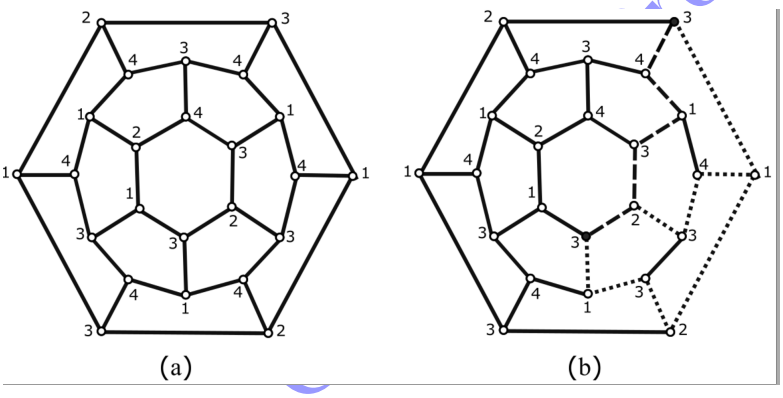


Figure 1: (a) A valid mutual-visibility 4-coloring of the $(5, 6)$-fullerene (Generalized Petersen) graph. (b) An invalid 4-coloring, where the two vertices in color class 3 violate mutual visibility due to an internal same-colored vertex on every shortest path.

3. Simulated annealing algorithm for mutual-visibility coloring of graphs

In this section, we present a SA algorithm for solving the mutual-visibility coloring problem. The proposed approach combines a binary search framework with

a fitness-guided SA procedure to compute the mutual-visibility chromatic number. The main components of the proposed method are described in the following subsections.

3.1 Simulated annealing and fitness function explanation

The algorithm takes as input the graph G and the number of available colors k. The initial value of k is obtained from the upper bound on the mutual-visibility chromatic number presented in [8]. A binary search strategy is then employed to determine the minimum feasible number of colors. At each iteration, the SA algorithm is executed to determine whether a feasible mutual-visibility coloring exists using k colors. If a feasible coloring is found, the upper bound is updated and a smaller value of k is tested; otherwise, the lower bound is increased. This process continues until the minimum feasible number of colors is identified.

For a given value of k, the algorithm generates an initial random coloring C, assigning each vertex of the graph a color from the set $\{1, \ldots, k\}$. This initial solution is then evaluated using the FITNESS function, yielding $(f, conf) = F(C)$. Simultaneously, a variable C_{best} is defined to store the best solutions found during the algorithm based on the highest fitness value.

If the initial coloring achieves perfect fitness ($f = 1$), the algorithm terminates, and this solution is returned as the optimal mutual-visibility coloring. Otherwise, for each current solution C, a set of neighboring solutions is generated. The number of neighbors is equal to the number of vertices $|V|$, where each neighbor differs from the current solution in the color of exactly one vertex while other colors remain unchanged. This ensures that the search space is explored in a controlled and incremental manner, allowing local improvements.

Next, using the fitness function, the value f' of each neighbor is computed, and the best neighbor with the highest fitness is selected and stored in C'. The value f' is compared with f_{best}, and if it is higher, C_{best} is updated. The current solution C is then replaced by C', and the main loop continues.

During the algorithm, if the fitness of a neighbor f' is lower than the current f, it may still be accepted according to the Boltzmann criterion. First, the fitness difference is computed as $\Delta = f' - f$, and the acceptance probability is:

$$p = e^{\Delta/T},$$

where T is the current temperature. A random number $r \in [0, 1]$ is generated. If $r < p$, the neighbor C' is accepted as the current solution C, even if it is worse than the current solution. This mechanism allows the algorithm to escape local minima and explore a broader search space.

Unlike the classical simulated annealing framework, where a single neighbor solution is randomly selected at each iteration, the proposed implementation evaluates a set of candidate neighbors and chooses the best one according to the objective function. Random-neighbor selection provides stronger diversification

Corrected Proof

by exploring the search space more broadly and decreasing the probability of premature convergence to local optima. However, this strategy may slow down the convergence process. The best-neighbor strategy, although computationally more expensive, provides a more guided search and generally leads to higher-quality solutions in fewer iterations. Since the instances considered in this work have moderate sizes, the additional computational cost of evaluating multiple neighbors is acceptable, and the resulting improvement in convergence and solution quality justifies this choice.

After each iteration, the system temperature T is gradually decreased using the cooling factor α provided as an algorithm input:

$$T \leftarrow T \times \alpha.$$

Gradual cooling ensures that in the early stages, weaker solutions are more likely to be accepted (increasing search diversity), while at lower temperatures, the algorithm performs a more precise and local search toward optimal solutions.

The algorithm continues iterating until one of the predefined stopping criteria is satisfied. In the proposed approach, the stopping criterion is problem-dependent. Specifically, the algorithm terminates if a feasible mutual-visibility coloring is found, i.e., when the fitness value reaches 1, indicating that no mutual-visibility conflicts remain. Otherwise, the search continues until the maximum number of iterations is reached, in which case the best solution found during the search is returned.

Corrected Proof

3.1.1 Fitness function evaluation

In the first evaluation step, based on the current coloring C, vertices are grouped according to their color. That is, if a vertex v has color i, it belongs to the color class C_i.

According to the mutual-visibility coloring problem definition, between any two vertices u and v of the same color, there must exist at least one shortest path that does not contain any other vertex of the same color.

The algorithm checks all possible pairs (u, v) of vertices with the same color and computes all shortest paths between them. If none of these paths satisfy the condition (i.e., all paths contain at least one other vertex of the same color), the total conflict count f is incremented by one. Otherwise, the conflict count remains unchanged.

Finally, the fitness of the solution is calculated as:

$$\text{Fitness} = \frac{1}{1+f}.$$

The lower the number of conflicts f, the higher the fitness value, indicating a better solution. This fitness value is then used in the simulated annealing algorithm for neighbor selection and updating the best solution. The steps of the algorithm are presented in Algorithm 1.

Algorithm 1 Simulated annealing for mutual-visibility coloring

Require: Graph $G(V, E)$, number of colors k, maximum iterations N, initial temperature T_0, cooling rate α $(0 < \alpha < 1)$

Ensure: Best coloring C_{best}, best fitness f_{best}

Generate an initial random coloring: $C \leftarrow \text{RandomColoring}(G, k)$
Evaluate C using the Fitness function: $(f, conf) \leftarrow \text{Fitness}(G, C)$
$C_{\text{best}} \leftarrow C$, $f_{\text{best}} \leftarrow f$, $conf_{\text{best}} \leftarrow conf$
$T \leftarrow T_0$
for $iter = 1$ to N **do**
  Generate neighbors of C by changing the color of only one vertex
  Compute the fitness of each neighbor using the Fitness function
  Set C' to the neighbor that has the maximum fitness value
  $(f', conf') \leftarrow \text{Fitness}(G, C')$
  $\Delta \leftarrow f' - f$
  if $\Delta > 0$ **then**
    $accept \leftarrow True$
  else
    $p \leftarrow e^{(\Delta/T)}$
    Accept the neighbor probabilistically using the Boltzmann criterion
    $accept \leftarrow (\text{Random}(0, 1) < p)$
  end if
  if $accept$ **then**
    $C \leftarrow C'$, $f \leftarrow f'$, $conf \leftarrow conf'$
    if $f > f_{\text{best}}$ **or** $(f = f_{\text{best}}$ **and** $conf < conf_{\text{best}})$ **then**
      $C_{\text{best}} \leftarrow C'$, $f_{\text{best}} \leftarrow f'$, $conf_{\text{best}} \leftarrow conf'$
    end if
  end if
  $T \leftarrow T \times \alpha$
  if $f_{\text{best}} = 1.0$ **then**
    break
  end if
end for

return $(C_{\text{best}}, f_{\text{best}}, conf_{\text{best}})$

Corrected Proof

3.2 Computational complexity analysis

The proposed SA algorithm consists of two main phases: a preprocessing phase and an iterative optimization phase.

Before the optimization process begins, all shortest paths between every unordered pair of vertices are computed and stored in a cache. Let $P_{u,v}$ denote the set of all shortest paths between vertices u and v. This preprocessing stage is performed only once and eliminates the need to recompute shortest paths during the search process. Let $P = \sum_{u<v} |P_{u,v}|$, denote the total number of shortest paths stored in the cache, where $P_{u,v}$ is the set of all shortest paths between vertices u and v. The preprocessing stage consists of two steps. First, shortest-path distances are computed from every vertex to all other vertices. For an unweighted graph, this requires $O(n+m)$ time per source vertex, yielding a total complexity of $O(n(n+m))$. Second, all shortest paths are enumerated and stored in the cache. Since a total of P shortest paths are stored and each path contains, on average, L vertices, this step requires $O(PL)$ time. Therefore, the overall complexity of the preprocessing stage is

$$O(n(n+m) + PL)\,,$$

where n and m denote the numbers of vertices and edges, respectively, and L denotes the average length of a shortest path.

Assume that the vertices are partitioned into color classes $C_1, C_2, \ldots, C_k$, where $|C_i| = s_i$. For each color class, the fitness function examines every pair of vertices having the same color. Since all shortest paths are already available in the cache, the algorithm only inspects the internal vertices of the stored paths to verify the mutual-visibility condition. Consequently, the complexity of one fitness evaluation is $O\left(\sum_{i=1}^{k} \sum_{\substack{u,v\in C_i \\ u<v}} |P_{u,v}|\, L\right)$. In sparse graphs, where the number of shortest paths between two vertices is typically small, this complexity reduces approximately to $O\left(\sum_{i=1}^{k} s_i^2\right)$, which is bounded by $O(n^2)$ in the worst case.

During each iteration, one neighboring solution is generated for every vertex by changing its assigned color. Therefore, the neighborhood contains $O(n)$ candidate solutions. Since each candidate solution requires one fitness evaluation, the computational complexity of a single iteration becomes $O\left(n \sum_{i=1}^{k} \sum_{\substack{u,v\in C_i \\ u<v}} |P_{u,v}|\, L\right)$. For N iterations, the total complexity of the optimization phase is

$$O\left(Nn \sum_{i=1}^{k} \sum_{\substack{u,v\in C_i \\ u<v}} |P_{u,v}|\, L\right).$$

Finally, the cache stores all shortest paths between every pair of vertices. Hence, the memory requirement is $O(PL)$, where $P = \sum_{u<v} |P_{u,v}|$ is the total number of stored shortest paths.

Corrected Proof

4. Integer linear programming model for the mutual-visibility coloring

In this section, we present the ILP model for the mutual-visibility coloring problem. We begin by defining the decision variables, followed by four essential constraints that describe and enforce the logical structure of the problem.

Definitions and variables

Let $G = (V, E)$ be a graph with $n = |V|$ vertices. According to the upper bound theorem in [8], the maximum number of colors is assumed to be $k_{max} = \lceil \frac{n(G) - \Delta(G) + 2}{2} \rceil$, where $\Delta(G)$ is the maximum degree of the graph G. Decision variables are defined as follows:

- $x_i \in \{0, 1\}$ for $i = 1, \ldots, k_{max}$: $x_i = 1$ if color i is used, and $x_i = 0$ otherwise.
- $y_{iv} \in \{0, 1\}$ for each $i \in \{1, ..., k_{max}\}$ and $v \in V$: $y_{iv} = 1$ if vertex v is colored with color i, and $y_{iv} = 0$ otherwise. .
- $z_{u,v,p,i} \in \{0, 1\}$ for each vertex pair (u, v), each shortest path $p \in P_{u,v}$, and each color i. The variable $z_{u,v,p,i}$ takes the value 1 if color i does not appear on any internal vertex of path p; otherwise, it takes the value 0, meaning that at least one internal vertex of the path is assigned color i.

Corrected Proof

Model explanation

The goal is to minimize the number of active colors. Basic constraints ensure that each vertex receives exactly one color and that colors without any assigned vertex remain inactive. The mutual-visibility constraints ensure that if two vertices u and v share the same color i, there exists at least one path $p \in P_{u,v}$ with no vertex of color i.

Complete ILP formulation

The full ILP model is as follows:

$$\min \quad \sum_{i=1}^{k_{\max}} x_i$$

$$\text{s.t.} \quad \sum_{i=1}^{k_{\max}} y_{iv} = 1, \qquad \forall v \in V,$$

(Each vertex receives exactly one color),

$$y_{iv} \le x_i, \qquad \forall i = 1, \ldots, k_{max},\ \forall v \in V,$$

(Color active only if used),

$$y_{iu} + y_{iv} - \sum_{p \in P_{u,v}} z_{u,v,p,i} \le 1, \qquad \forall u, v \in V,\ u \neq v,\ \forall i = 1, \ldots, k_{max},$$

(Mutual-visibility condition),

$$z_{u,v,p,i} + y_{iw} \le 1, \qquad \forall u, v,\ \forall p \in P_{u,v},\ \forall w \in \mathrm{Int}(p),\ \forall i,$$

(Path vertices cannot use color i),

$$x_i \ge x_{i+1}, \qquad \forall i = 1, \ldots, k_{\max} - 1,$$

$$\sum_{v \in V} y_{iv} \ge \sum_{v \in V} y_{(i+1)v}, \qquad \forall i = 1, \ldots, k_{\max} - 1,$$

$$x_i, y_{iv}, z_{u,v,p,i} \in \{0, 1\} \qquad \forall i, u, v, p.$$

where $\mathrm{Int}(p)$ denotes the set of internal vertices of the shortest path p.

Corrected Proof

Constraint descriptions

Since the objective of the problem is to determine the minimum number of colors required to obtain a mutual-visibility coloring of a graph, the objective function is defined as $\min \sum_{i=1}^{k_{max}} x_i$.

Here, the binary variable x_i indicates whether color i is used in the coloring. Since each vertex must be assigned exactly one color, the following constraint is imposed:

$$\sum_{i=1}^{k_{max}} y_{iv} = 1, \qquad \forall v \in V. \tag{1}$$

This constraint guarantees that every vertex receives exactly one color.
Furthermore, if a vertex is assigned color i, then color i must be activated in the solution. This relationship is enforced by the following constraint:

$$y_{iv} \le x_i, \qquad \forall i = 1, \ldots, k_{max},\ \forall v \in V. \tag{2}$$

The most important part of the formulation is the enforcement of the mutual-visibility condition. According to the definition of the problem, if two vertices u and v receive the same color i, then there must exist at least one shortest path between them whose internal vertices are not assigned color i. This requirement is modeled by the following constraints:

$$y_{iu} + y_{iv} - \sum_{p \in P_{u,v}} z_{u,v,p,i} \leq 1, \qquad \forall u, v \in V,\ u \neq v,\ \forall i = 1, \ldots, k_{max}, \tag{3}$$

and

$$z_{u,v,p,i} + y_{iw} \leq 1, \qquad \forall u, v \in V,\ \forall p \in P_{u,v},\ \forall w \in \mathrm{Int}(p),\ \forall i = 1, \ldots, k_{max}. \tag{4}$$

To clarify the role of these constraints, suppose that two vertices u and v are both assigned color i, that is $y_{iu} = y_{iv} = 1$. Substituting these values into the first constraint yields $\sum_{p \in P_{u,v}} z_{u,v,p,i} \geq 1$. Therefore, there exists at least one shortest path p between u and v such that $z_{u,v,p,i} = 1$.

Now, by the second constraint, for every internal vertex w of this path we have $z_{u,v,p,i} + y_{iw} \leq 1$. Since $z_{u,v,p,i} = 1$, it immediately follows that

$$y_{iw} = 0, \qquad \forall w \in \mathrm{Int}(p).$$

Hence, none of the internal vertices on the selected shortest path is assigned color i. Consequently, the vertices u and v satisfy the mutual-visibility condition.

Symmetry-breaking constraints

Since the color labels are interchangeable, the ILP formulation admits many symmetric optimal solutions corresponding to permutations of the same coloring. To reduce this symmetry and improve the efficiency of the branch-and-bound search, we incorporate the following symmetry-breaking constraints.

First, the color activation variables are enforced to be consecutive:

$$x_i \geq x_{i+1}, \qquad i = 1, \ldots, k_{max} - 1. \tag{5}$$

This guarantees that color $i + 1$ cannot be used unless color i is also used. Furthermore, the sizes of the color classes are required to be non-increasing:

$$\sum_{v \in V} y_{iv} \geq \sum_{v \in V} y_{(i+1)v}, \qquad i = 1, \ldots, k_{max} - 1. \tag{6}$$

These constraints eliminate equivalent symmetric solutions without excluding any feasible coloring or changing the optimal objective value.

The above discussion shows that every feasible solution of the proposed ILP model induces a valid mutual-visibility coloring of the graph. We now prove the converse, namely that every valid mutual-visibility coloring of the graph can be transformed into a feasible solution of the ILP formulation by satisfying all of its constraints (after an appropriate relabeling of the color classes).

Corrected Proof

Lemma 4.1. *For every valid mutual-visibility coloring of a graph G, there exists an equivalent feasible solution to the proposed ILP formulation.*

Proof. Let $\phi : V \to \{1, \ldots, h\}$ be an arbitrary valid mutual-visibility coloring of G, where h denotes the number of colors used by ϕ. Since $k_{\max}$ is a valid upper bound on the mutual-visibility chromatic number of G, we have $h \leq k_{\max}$. Since color labels are arbitrary, we may relabel the colors without affecting the coloring so that the used colors are indexed consecutively by $1, 2, \ldots, h$, and the color classes are ordered in nonincreasing cardinality.

We define the decision variables as:

$$y_{iv} = \begin{cases} 1, & \text{if } \phi(v) = i, \\ 0, & \text{otherwise,} \end{cases} \qquad x_i = \begin{cases} 1, & \text{if color } i \text{ is used,} \\ 0, & \text{otherwise,} \end{cases}$$

where $i = 1, \ldots, h$. Furthermore, for every pair of vertices u, v with $\phi(u) = \phi(v) = i$, the definition of a valid mutual-visibility coloring guarantees the existence of at least one shortest path $p \in P_{u,v}$ whose internal vertices are not colored i. For every such witnessing path, we set $z_{u,v,p,i} = 1$.

We now verify that all constraints of the ILP formulation are satisfied. Since every vertex receives exactly one color, exactly one variable y_{iv} equals 1 for each vertex v. Hence,

$$\sum_{i=1}^{k_{\max}} y_{iv} = 1,$$

and Constraint (1) is satisfied. If $y_{iv} = 1$, then vertex v is assigned color i. Therefore, color i is used, implying $x_i = 1$. Consequently, $y_{iv} \leq x_i$, so Constraint (2) holds.

Now suppose that vertices u and v are assigned the same color i. Then, $y_{iu} = y_{iv} = 1$. Since ϕ is a valid mutual-visibility coloring, there exists at least one witnessing shortest path $p \in P_{u,v}$. Therefore,

$$\sum_{p \in P_{u,v}} z_{u,v,p,i} \geq 1,$$

which yields

$$y_{iu} + y_{iv} - \sum_{p \in P_{u,v}} z_{u,v,p,i} \leq 2 - 1 = 1.$$

If u and v are not both assigned color i, then at least one of y_{iu} and y_{iv} equals 0, and the inequality is satisfied trivially. Hence, Constraint (3) holds.

Next, let $z_{u,v,p,i} = 1$. By construction, p is a witnessing shortest path whose internal vertices are not colored i. Thus, for every $w \in \mathrm{Int}(p)$, we have $y_{iw} = 0$, implying

$$z_{u,v,p,i} + y_{iw} = 1 + 0 = 1 \leq 1.$$

Corrected Proof

If $z_{u,v,p,i} = 0$, the inequality also holds immediately. Therefore, Constraint (4) is satisfied. Since the used colors have been relabeled consecutively, we have $x_i = 1$ for $1 \leq i \leq h$ and $x_i = 0$ for $h < i \leq k_{\max}$. Hence,

$$x_i \geq x_{i+1}, \qquad i = 1, \ldots, k_{\max} - 1,$$

and Constraint (5) holds. Finally, because the color classes have been ordered by non-increasing cardinality,

$$\sum_{v \in V} y_{iv} \geq \sum_{v \in V} y_{(i+1)v}, \qquad i = 1, \ldots, k_{\max} - 1,$$

which verifies Constraint (6).

Therefore, all constraints of the ILP formulation are satisfied. Hence, every valid mutual-visibility coloring, after an appropriate relabeling of its color classes, corresponds to a feasible solution of the proposed ILP model. □

4.1 Computational complexity analysis

Let $n = |V|$ denote the number of vertices of the graph, and let $P = \sum_{u<v} |P_{u,v}|$ be the total number of shortest paths between all unordered pairs of vertices, where $P_{u,v}$ denotes the set of all shortest paths between vertices u and v. Furthermore, let L denote the average number of internal vertices of a shortest path.

The proposed ILP formulation introduces three classes of binary variables. The variables x_i indicate whether color i is used, resulting in n binary variables. The assignment variables $y_{i,v}$ specify whether vertex v is assigned color i, yielding n^2 binary variables. Finally, for every color, every shortest path, and every pair of vertices, the formulation introduces one binary variable $z_{u,v,p,i}$. Therefore, the total number of path variables is nP. Consequently, the total number of binary variables is

$$n + n^2 + nP,$$

which is dominated by the path variables whenever the number of shortest paths becomes large.

The formulation consists of six groups of constraints. The assignment constraints require each vertex to receive exactly one color, producing n constraints. The linking constraints between x_i and $y_{i,v}$ contribute n^2 constraints. The visibility constraints that connect vertices having the same color generate one constraint for every unordered vertex pair and every color, resulting in $O(n^3)$ constraints. Finally, the path constraints relate each path variable to every internal vertex of the corresponding shortest path. Since each stored path contains, on average, L internal vertices, these constraints contribute $O(nPL)$ constraints.

In addition, two groups of symmetry-breaking constraints are incorporated. The first group enforces consecutive activation of colors (Constraint (5)), while the second requires the color-class sizes to be ordered in a non-increasing manner (Constraint (6)). Together, these constraints contribute an additional $2(k_{max} - 1) =$

Corrected Proof

$O(n)$ constraints.
Hence, the total number of constraints remains $O(n^3 + nPL)$. Regarding memory consumption, the formulation is dominated by the path variables and their associated constraints. Therefore, the memory requirement is proportional to $O(nPL)$. The computational complexity of solving the proposed ILP formulation is determined primarily by the branch-and-bound procedure employed by modern MILP solvers. Since binary integer programming is NP-hard, the worst-case running time is exponential in the number of binary variables. Consequently, the proposed formulation is intended primarily for small- and medium-sized graph instances, where optimal solutions can still be obtained within reasonable computational time.
Although the introduced symmetry-breaking constraints do not change the asymptotic complexity of the formulation, they reduce the number of equivalent symmetric solutions explored by the branch-and-bound solver, thereby improving its practical performance.
It is worth noting that the size of the formulation depends directly on the total number of shortest paths in the graph. For many sparse graph classes considered in this paper, this number remains moderate. However, for graph families possessing a large number of shortest paths, such as grid graphs, hypercubes, and other highly symmetric graphs, the number of shortest paths may grow exponentially. Consequently, the number of path variables and constraints also increases significantly, which limits the practical applicability of the exact ILP formulation to relatively small instances. Nevertheless, these exact solutions provide reliable benchmarks for evaluating the quality of heuristic algorithms. The size and computational complexity of the proposed ILP formulation, including the numbers of binary variables, constraints, and memory requirements, are summarized in Table 1.

Corrected Proof

Table 1: Size of the proposed ILP formulation.

Component	Size
Binary variables x_i	n
Binary variables $y_{i,v}$	n^2
Binary variables $z_{u,v,p,i}$	nP
Total binary variables	$O(n^2 + nP)$
Assignment constraints	n
Linking constraints	n^2
Visibility constraints	$O(n^3)$
Path constraints	$O(nPL)$
Symmetry-breaking constraints	$2(k_{max} - 1) = O(n)$
Total constraints	$O(n^3 + nPL)$
Memory requirement	$O(nPL)$

5. Comparison of theoretical and computational results

Several solution approaches have been proposed in the literature for the mutual visibility coloring problem, including the ILP model introduced in section 4, the greedy algorithm proposed in [8], and the SA metaheuristic described in section 3. Although the ILP model is an exact optimization method capable of finding optimal solutions, its practicality is limited by the rapid growth in the number of variables and constraints as the graph complexity increases. Consequently, this approach is only feasible for small-sized graphs. For larger graphs, the computational time increases dramatically, making the method practically intractable. Similarly, as stated in [8], the greedy algorithm requires repeated computation of mutual-visibility sets, whose determination is NP-hard for general graphs. Consequently, the algorithm runs in polynomial time only for graph classes in which the mutual-visibility property can be tested in polynomial time, whereas for arbitrary graphs, its repeated solution of an NP-hard subproblem prevents it from being a general-purpose polynomial-time algorithm. For these reasons, the SA metaheuristic is employed in this work. Unlike the greedy algorithm, SA is applicable to arbitrary graph classes without requiring polynomial-time computation of mutual-visibility sets. Moreover, SA demonstrates the high computational efficiency, and its accuracy and effectiveness are clearly reflected in the reported results.

In what follows, we present a table summarizing the results obtained by applying SA and ILP to several graphs. All experiments were carried out in Python 3.11.5, using the PuLP 3.3.0 optimization package, on a laptop equipped with an Intel(R) Core(TM) i5-4300M processor running at 2.60 GHz and 64 GB of RAM. The ILP model was solved using the COIN-OR Branch-and-Cut (CBC) MILP Solver (Version 2.10.3) with its default parameter settings. To assess the correctness of the algorithms, we first determine the mutual-visibility chromatic number of this graph class using Theorem 5.1 and Theorem 5.2. This value serves as an exact benchmark for evaluating the accuracy and performance of the proposed methods. To investigate the sensitivity of the SA performance with respect to the main control parameters, a parameter tuning analysis was performed using the combinations $T_0 \in 5, 10, 20$ and $\alpha \in 0.8, 0.9, 0.95$. Two representative fullerene instances with different structural characteristics were considered for this analysis. Each parameter configuration was independently executed several times, and the average solution value, standard deviation, and runtime were calculated. The results are summarized in Table 2. The obtained results show that the tested parameter combinations generally lead to similar solution qualities, demonstrating the robustness of the proposed SA approach. Although some configurations achieve comparable objective values, the setting $T_0 = 10$ and $\alpha = 0.95$ provides a suitable trade-off between exploration capability, convergence speed, and computational efficiency. Consequently, this parameter setting was adopted in all subsequent SA experiments. For the graph classes considered in our computational experiments, the corresponding theoretical values are summarized in the following results.

Corrected Proof

Theorem 5.1. *For the cycle graph C_n with n vertices, the mutual-visibility chromatic number is*

$$\chi_\mu(C_n) = \left\lceil \frac{n}{3} \right\rceil .$$

Proof. As said in [8], every connected graph G satisfies

$$\chi_\mu(G) \geq \left\lceil \frac{n(G)}{\mu(G)} \right\rceil . \tag{7}$$

Since $\mu(C_n) = 3$ [3], it follows that

$$\chi_\mu(C_n) \geq \left\lceil \frac{n}{3} \right\rceil . \tag{8}$$

On the other hand, Let the vertices of C_n be labeled consecutively as $V(C_n) = \{v_1, v_2, \ldots, v_n\}$, and let $t = \lceil \frac{n}{3} \rceil$. Define a coloring by assigning color i to the vertices v_i, v_{i+t}, v_{i+2t}, whenever the corresponding indices do not exceed n, for $i = 1, 2, \ldots, t$. Thus, each color class contains at most three vertices. We claim that this coloring is a mutual-visibility coloring. Now, we prove our claim.
Let u and v be two vertices receiving the same color j. Then, u and v are two vertices from the set $\{v_j, v_{j+t}, v_{j+2t}\}$, whenever the third vertex exists. Since $t = \lceil \frac{n}{3} \rceil$, we have $3t \geq n$, and therefore $n - 2t \leq t$. Hence, the three arcs of the cycle determined by v_j, v_{j+t}, and v_{j+2t} have lengths $t, t, n - 2t$, all of which are at most t. Consequently, the shortest path between any two vertices of the same color is exactly one of these three arcs.

Moreover, the third vertex of the color class (if it exists) lies on one of the other two arcs and therefore does not belong to this shortest path. Since the only vertices colored j are v_j, v_{j+t}, and possibly v_{j+2t}, none of the internal vertices of the shortest path has color j.

Therefore, for every pair of vertices with the same color, there exists a shortest path whose internal vertices have colors different from the endpoints. Hence, every pair of vertices in the same color class is mutually visible.
Thus, the proposed coloring is a valid mutual-visibility coloring using $\lceil \frac{n}{3} \rceil$ colors. Therefore,

$$\chi_\mu(C_n) \leq \left\lceil \frac{n}{3} \right\rceil . \tag{9}$$

Combining (8) and (9), we conclude that

$$\chi_\mu(C_n) = \left\lceil \frac{n}{3} \right\rceil .$$

□

Theorem 5.2. *([8]). If T is a tree, then*

$$\chi_\mu(T) = \begin{cases} rad(T) + 1, & \text{if } diam(T) \text{ is even,} \\ rad(T), & \text{otherwise.} \end{cases}$$

Corrected Proof

Figure 2 presents the binary search procedure and the convergence behavior of the proposed SA algorithm on the $(5, 6)$-fullerene (Generalized Peterson graph). Figure 2 illustrates different aspects of the performance of the proposed algorithm

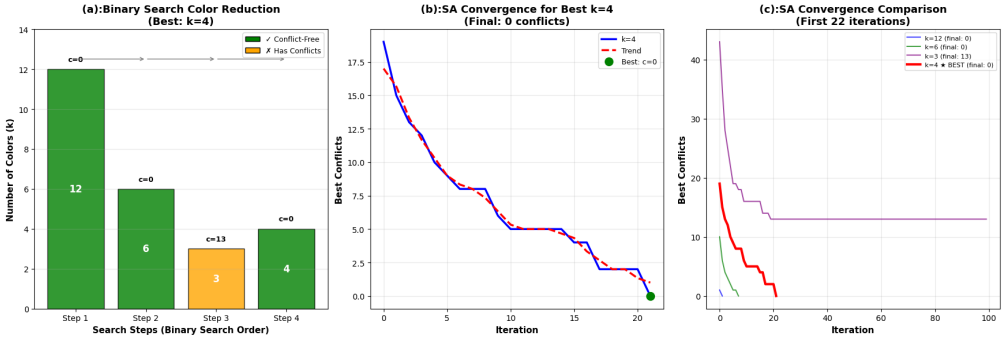


Figure 2: Illustration of the proposed solution procedure on the $(5, 6)$-fullerene (Generalized Petersen) graph. (a) Binary search procedure for determining the minimum feasible number of colors k. (b) Convergence behavior of the SA algorithm toward the optimal solution at $k = 4$. (c) Comparative convergence profiles of the algorithm for different values of k.

on the $(5, 6)$-fullerene graph (Generalized Peterson graph).

Figure 2(a) shows the binary search procedure used to determine the minimum number of colors required for a mutual-visibility coloring. The search starts from the upper bound of 12 colors, for which a conflict-free coloring is obtained ($c = 0$). Consequently, the number of colors is reduced to 6, which also yields a valid coloring without conflicts. The search then proceeds to $k = 3$, where the algorithm is unable to obtain a feasible coloring, resulting in 13 conflicts. Finally, the value $k = 4$ is examined and a conflict-free coloring is obtained. Since $k = 3$ is infeasible whereas $k = 4$ is feasible, the algorithm concludes that the mutual-visibility chromatic number of the graph is $\chi_\mu(G) = 4$. This figure also demonstrates that the binary search strategy significantly reduces the search space by evaluating only four candidate values of k.

Figure 2(b) presents the convergence behavior of the proposed SA algorithm for the optimal value $k = 4$. The horizontal axis represents the iteration number, while the vertical axis indicates the best conflict value obtained up to each iteration. The initial solution contains 19 conflicts, indicating that the randomly generated coloring is far from satisfying the mutual-visibility constraints. As the search progresses, the number of conflicts decreases steadily through successive neighborhood moves and the probabilistic acceptance mechanism of the SA algorithm. The improvement is more pronounced during the early iterations, whereas the convergence becomes slower near the optimum. A conflict-free solution is ob-

Corrected Proof

tained after approximately 21 iterations, indicating that the algorithm successfully finds a valid mutual-visibility coloring using four colors. The dashed red line illustrates the overall convergence trend.

Figure 2(c) compares the convergence behavior of the proposed SA algorithm for four different values of k, namely 12, 6, 4, and 3. For $k = 12$, the algorithm reaches a conflict-free coloring almost immediately, since the large number of available colors makes the problem relatively easy. A similar behavior is observed for $k = 6$, where all conflicts are eliminated within a few iterations. For $k = 4$, which corresponds to the optimal solution, the algorithm gradually reduces the number of conflicts from 19 to 0, requiring approximately 21 iterations. This represents the most challenging feasible case. In contrast, when $k = 3$, the algorithm initially decreases the number of conflicts but eventually stagnates at 13 conflicts without reaching a feasible solution. This behavior provides experimental evidence that three colors are insufficient for the considered graph, whereas four colors are necessary to obtain a valid mutual-visibility coloring.

Table 2: Sensitivity analysis of SA parameters (T_0, α) for two fullerene graphs: (a) $(5, 6)$-fullerene (Generalized Petersen graph), and (b) $(4, 6)$-fullerene ($(BN)_{16}$ graph).

(a)

T_0	α	Average Solution(k)	Standard Deviation	Average Time (s)
5	0.80	4.30	0.46	2.27
5	0.90	4.40	0.49	2.36
5	0.95	4.40	0.49	2.33
10	0.80	4.40	0.49	2.29
10	0.90	4.30	0.46	2.55
10	0.95	4.10	0.30	2.62
20	0.80	4.50	0.50	2.38
20	0.90	4.30	0.46	2.51
20	0.95	4.30	0.46	2.45

(b)

T_0	α	Average Solution(k)	Standard Deviation	Average Time (s)
5	0.80	5.80	0.40	5.13
5	0.90	5.90	0.30	5.50
5	0.95	5.90	0.30	5.36
10	0.80	5.80	0.40	5.76
10	0.90	5.90	0.30	5.31
10	0.95	5.80	0.40	5.73
20	0.80	5.80	0.40	5.54
20	0.90	6.00	0.00	5.12
20	0.95	5.80	0.40	5.53

Corrected Proof

Table 3 compares the theoretical results with the solutions obtained by the proposed ILP model and SA algorithm. For each graph instance, the table reports the number of vertices, the theoretical value (when available), the exact result and computational time of the ILP model, and the best, mean, standard deviation, and execution time obtained by the SA algorithm over 10 independent runs. The results show that the ILP model successfully computes optimal solutions for small- and medium-sized instances; however, as the graph size increases, the rapid growth in the number of variables and constraints renders the exact approach computationally impractical. In contrast, the proposed SA algorithm remains applicable to all tested graph classes and consistently produces optimal or near-optimal solutions within reasonable computational time.

Table 3: Comparison of the theoretical value, ILP results, and SA performance.

Graph	$n(G)$	Theory	ILP		SA				Runs
			Result	Time (s)	Best	Mean	Std. Dev.	Time (s)	
C_{15}	15	5	5	18.85	5	5.00	0	0.0491	10
C_{25}	25	9	9	535.72	9	9.00	0	0.7985	10
C_{35}	35	12	–	–	12	12.00	0	1.2603	10
P_{10}	10	5	5	12.29	5	5.00	0	0.1272	10
P_{20}	20	10	10	120.89	10	10.00	0	0.5171	10
P_{30}	30	15	–	–	15	15.00	0	1.1841	10
Complete bipartite graph $K_{1,10}$	11	2	2	1.35	2	2.00	0	0.0035	10
Kite tree $(4, 5)$	9	4	4	3.54	4	4.00	0	0.0704	10
$(4, 6)$-fullerene ($(BN)_{16}$ graph)	32	≤ 16	5	2126.38	5	5.70	0.4830	8.0577	10
$(5, 6)$-fullerene (Generalized Peterson graph)	24	≤ 12	4	4.80	4	4.30	0.4830	2.1417	10
$(5, 7)$-fullerene (Dodecahedral graph)	20	≤ 10	3	320.72	4	4.00	0	2.6539	10
C_{60}-fullerene (Truncated icosahedral graph)	60	≤ 30	–	–	7	7.60	0.5164	30.6932	10

Corrected Proof

6. Conclusion

In this paper, we investigated the mutual-visibility coloring problem by proposing both heuristic and exact optimization approaches. Since the problem is NP-hard, we first developed a SA algorithm capable of producing high-quality near-optimal solutions for arbitrary graph classes within reasonable computational time. We also presented the first ILP formulation for the problem, which provides exact optimal solutions and serves as a benchmark for evaluating the performance of the proposed heuristic. Computational experiments conducted on a diverse collection of benchmark graph instances demonstrate the effectiveness of both approaches. While the proposed ILP model successfully computes optimal solutions for small- and medium-sized instances, the SA algorithm is more suitable for larger graphs due to its computational efficiency.

As future work, other metaheuristic approaches, particularly population-based algorithms such as Genetic Algorithms and Ant Colony Optimization, may be developed for the mutual-visibility coloring problem and compared with the proposed SA algorithm. Furthermore, improving the mathematical formulation or designing hybrid exact–heuristic approaches may further enhance the efficiency of solving larger problem instances.

Conflicts of Interest. The authors declare that they have no conflicts of interest regarding the publication of this article.

References

[1] A. Aljohani and G. Sharma, Complete visibility for mobile robots with lights tolerating faults, *Int. J. Netw. Comput.* **8** (2018) 32 – 52.

[2] G. A. Di Luna, P. Flocchini, S. G. Chaudhuri, F. Poloni, N. Santoro and G. Viglietta, Mutual visibility by luminous robots without collisions, *Inform. and Comput.* **254** (2017) 392 – 418, https://doi.org/10.1016/j.ic.2016.09.005.

[3] G. Di Stefano, Mutual visibility in graphs, *Appl. Math. Comput.* **419** (2022) #126850, https://doi.org/10.1016/j.amc.2021.126850.

[4] S. Cicerone, G. Di Stefano, S. Klavžar and I. G. Yero, Mutual-visibility in strong products of graphs via total mutual-visibility, *Discrete Appl. Math.* **358** (2024) 136 – 146, https://doi.org/10.1016/j.dam.2024.06.038.

[5] D. Korže and A. Vesel, Mutual-visibility sets in Cartesian products of paths and cycles, *Results Math.* **79** (2024) #116.

[6] D. Kuziak and J. A. Rodríguez-Velázquez, Total mutual-visibility in graphs with emphasis on lexicographic and Cartesian products, *Bull. Malays. Math. Sci. Soc.* **46** (2023) #197, https://doi.org/10.1007/s40840-023-01590-3.

[7] Z. Hamed-Labbafian, M. Tavakoli, N. Sabeghi and A. Vesel, Mutual visibility in graphs: Hierarchical products, genetic algorithm and some applications, *J. Supercomput.* **82** (2026) #455, https://doi.org/10.1007/s11227-026-08581-8.

[8] S. Klavžar, D. Kuziak, J. C. V. Tripodoro and I. G. Yero, Coloring the vertices of a graph with mutual-visibility property, *Open Math.* **23** (2025) #20250193.

[9] S. Babu, G. Di Stefano and A. Lakshmanan S, Mutual-visibility coloring of graphs, (2026), https://arXiv:2512.12251v2.

Corrected Proof

Alireza Masoudi
Department of Applied Mathematics,
Faculty of Mathematical Sciences,
Ferdowsi University of Mashhad,
P. O. Box 1159-91775, Mashhad, I. R. Iran
e-mail: alimasoodix@gmail.com

Mostafa Tavakoli
Department of Applied Mathematics,
Faculty of Mathematical Sciences,
Ferdowsi University of Mashhad,
P. O. Box 1159-91775, Mashhad, I. R. Iran

e-mail: m_tavakoli@um.ac.ir

Zahra Hamed-Labbafian
Department of Applied Mathematics,
Faculty of Mathematical Sciences,
Ferdowsi University of Mashhad,
P. O. Box 1159-91775, Mashhad, I. R. Iran
e-mail: hamedlabbafianzahra@gmail.com

Corrected Proof