

Computing the geodesic number of a graph

Júlio Araújo

ParGO, Departamento de Matemática, Universidade Federal do Ceará
Av. Humberto Monte, s/n, Campus do Pici, Bloco 914. CEP 60440-900. Fortaleza, Ceará, Brasil
julio@mat.ufc.br

Manoel Campêlo

ParGO, Departamento de Estatística e Matemática Aplicada, Universidade Federal do Ceará
Av. Humberto Monte, s/n, Campus do Pici, Bloco 910. CEP 60440-900. Fortaleza, Ceará, Brasil
mcampelo@ufc.br

Gabriel H. de Sousa

ParGO, Departamento de Computação, Universidade Federal do Ceará
Av. Humberto Monte, s/n, Campus do Pici, Bloco 910. CEP 60440-900. Fortaleza, Ceará, Brasil
gabrielsousa.9957@gmail.com

RESUMO

Seja $G = (V, E)$ um grafo simples e não-orientado. $S \subseteq V(G)$ é um *conjunto de intervalo* na convexidade geodésica se, para todo $w \in V \setminus S$, existem $u, v \in S$ tais que w é vértice interno de algum u, v -caminho mínimo (u, v -geodésica). O *número de intervalo* de G na convexidade geodésica, denotado por $\text{in}_g(G)$, é a menor cardinalidade de um conjunto de intervalo de G . Esse parâmetro também é conhecido na literatura como *número geodésico*. Determinar o número geodésico de um grafo é um problema NP-Difícil, mesmo quando o grafo dado como entrada é bipartido [Dourado et al., 2010]. Neste trabalho, foram propostas duas formulações matemáticas para determinar tal parâmetro, uma exponencial e outra compacta. Realizamos experimentos computacionais, com o uso do CPLEX, para avaliar e comparar suas performances sobre instâncias geradas aleatoriamente.

PALAVRAS CHAVE. Convexidade em Grafos, Número de intervalo, Formulação matemática, Número geodésico.

Tópicos: Programação Matemática, Otimização Combinatória, Teoria e Algoritmos em Grafos.

ABSTRACT

Let $G = (V, E)$ be a simple, undirected graph. $S \subseteq V(G)$ is an *interval set* in geodesic convexity if, for every $w \in V \setminus S$, there exist $u, v \in S$ such that w is an internal vertex of some u, v -shortest path (u, v -geodesic). The *interval number* of G in geodesic convexity, denoted by $\text{in}_g(G)$, is the smallest cardinality of an interval set of G . This parameter is also known in the literature as the *geodesic number*. Determining the geodesic number of a graph is an NP-Hard problem, even when the input graph is bipartite [Dourado et al., 2010]. In this work, two mathematical formulations are proposed to determine this parameter, one exponential and the other compact. We conduct computational experiments, using CPLEX, to evaluate and compare their performances on randomly generated instances.

KEYWORDS. Graph Convexity, Interval number, Mathematical formulation, Geodesic number.

Topics: Mathematical Programming, Combinatorial Optimization, Theory and Algorithms in Graphs.

1. Introduction

A set of combinatorial optimization problems that has been receiving considerable attention in the Graph Theory area corresponds to what is generically termed problems of convexity, infection, or contamination in graphs. Several optimization parameters involving convexity in graphs are explored in the literature. In general, they are related to an iterative process, in which, starting from a subset of initially contaminated vertices, step by step, others will also be contaminated [Pelayo, 2013; Araújo et al., 2023].

The convexity concept originates from Euclidean geometry, where the shortest path between two points is the line segment between them. In this context, a set is said to be *convex* when the segment (shortest path) between any two points in the set is contained within it. The *convex hull* of a given set of points is well-known to be the inclusion-wise minimum convex set containing such points. For instance, by translating points to vertices of a graph, and line segment linking two points u, v to all shortest u, v -paths in the graph, we obtain the analogous definitions of a convex set, and of the convex hull of a set of vertices, in a graph in well-known Geodesic Convexity [Pelayo, 2013].

Note that we can obtain the convex hull of a finite set S of points in the plane by iteratively adding points belonging to the line segments linking any two points in S . Analogously, in a graph, to compute the convex hull of a subset of vertices S , one can iteratively add vertices to a set H that starts with S , by adding vertices belonging to any shortest u, v -path for every u, v in H . Since we only consider finite graphs, this process halts with the convex hull of S . This can often be seen as an infection, contamination, or diffusion process in the graph through shortest paths. Other infection models are also considered in the literature, giving rise to various Graph Convexities.

Let us formalize these notions. A *convexity space*¹ is an ordered pair $(\mathcal{V}, \mathcal{C})$, where $\mathcal{V}$ is a non-empty finite set and $\mathcal{C}$ is a family of subsets of $\mathcal{V}$, called *convex sets*, satisfying:

- (i) $\emptyset, \mathcal{V} \in \mathcal{C}$;
- (ii) $C \cap C' \in \mathcal{C}$, for all $C, C' \in \mathcal{C}$.

The complements of convex sets with respect to $\mathcal{V}$ are called here *co-convex sets*. Thus, if $S \in \mathcal{C}$ is convex, we denote $C = \mathcal{V} \setminus S$ as its corresponding co-convex set. Given a subset $C \subseteq \mathcal{V}$, the *convex hull* of C with respect to $(\mathcal{V}, \mathcal{C})$ is the unique inclusion-wise minimal convex set $C' \in \mathcal{C}$ that contains C , and it is denoted by $H_{(\mathcal{V}, \mathcal{C})}(C)$.

If $H_{(\mathcal{V}, \mathcal{C})}(C) = \mathcal{V}$, then C is called a *hull set* of $(\mathcal{V}, \mathcal{C})$. The *hull number* of $\mathcal{V}$ with respect to $\mathcal{C}$ is the minimum cardinality of a hull set, denoted by $hn_{(\mathcal{V}, \mathcal{C})}$. These notions are related to the works of Farber and Jamison [1986]; Duchet [1988]; Everett and Seidman [1985].

We highlight two basic properties of a convexity space:

1. If $S \subseteq S' \subseteq \mathcal{V}$, then $H_{(\mathcal{V}, \mathcal{C})}(S) \subseteq H_{(\mathcal{V}, \mathcal{C})}(S')$;
2. If $C \neq \emptyset$ is co-convex and S is a hull set, then $S \cap C \neq \emptyset$.

When it comes to convexity in graphs, the vast majority of convexities defined in the literature take $\mathcal{V}$ as the set of vertices of a given graph, and the family $\mathcal{C}$ as subsets of vertices that are fixed points of an interval function. Given a graph $G = (V, E)$, an *interval function* is a function $I : 2^{|V|} \rightarrow 2^{|V|}$ such that: (i) $I(S) \supseteq S$, for every $S \subseteq V(G)$, and (ii) $I(S') \subseteq I(S)$, for every $S' \subseteq S$. Let $S \subseteq V(G)$. The vertices in $I(S)$ are called *infected* or *generated* by S . We say that S

¹Since we only deal with finite graphs, we restrict this definition to this case but, for infinite sets $\mathcal{V}$, there is a third condition ensuring that *nested unions* of elements of $\mathcal{C}$ must also be in the family $\mathcal{C}$.

is an *interval set* if $I(S) = V(G)$. The *interval number* of G is the cardinality of a minimal interval set, denoted by $in(G)$.

It is well-known (see Araújo et al. [2023], Lemma 2.1.) that the fixed points of an interval function define a convexity in G , called an *interval convexity*. Let us consider a graph convexity defined by an interval function $I : 2^V \rightarrow 2^V$. We say that such a function defines a path convexity if, for every $S \subseteq V$, and for every $w \in I(S) \setminus S$, there exist $u, v \in S$ such that w lies on a predetermined type of path between the vertices u and v , meaning that the function I includes vertices that are on this type of path. For example, $I(S)$ could be the vertices lying on any path on three vertices (of length 2) between pairs of vertices in S [Centeno et al., 2009]. This gives rise to the P_3 -convexity.

A geodesic between two vertices u and v in a graph G is the shortest path between them in G . In the geodesic convexity, the interval function, denoted by $I_g(S)$, infects all vertices that belong to any shortest path between any two vertices of S . Thus, a set of vertices S is convex in the geodesic convexity if, for any pair of vertices within this set, all vertices along any shortest path between these two vertices also belong to the set. Taking the graph in Figure 1 as an example and $S = \{1, 6\}$ highlighted in black, we have that the shortest paths between 1 and 6 are $\langle 1, 2, 6 \rangle$ and $\langle 1, 5, 6 \rangle$, $I_g(S)$ includes vertices 2 and 5 in the geodesic convexity.

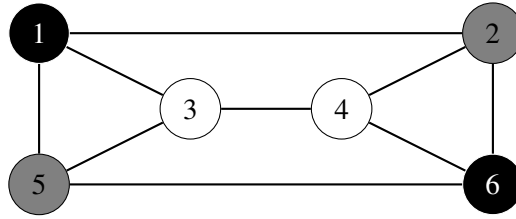


Figure 1: geodesic convexity

In addition to geodesic convexity, many other path convexities are explored in the literature, each of which differs exactly by the specific characteristic of the paths defining the interval function. In Table 1, we present the main path convexities in graphs and their characterizations for each path type.

Table 1: Main convexities of paths in the literature

Convexity	Path characteristic
Geodesic [Harary and Nieminem, 1981]	Shortest paths
Monophonic [Jamison, 1982]	Induced paths
g^3 [Nielsen and Oellermann, 2011]	Shortest paths with length at least three
m^3 [Dragan et al., 1999]	Induced paths with length at least three
g_k [Farber and Jamison, 1987]	Shortest paths of length at most k
P_3 [Centeno et al., 2009]	Paths of length two
P_3^* [Araújo et al., 2013]	Induced paths of length two
P_4 [Enriquez and Junior, 2014]	Paths of length three
triangle-path [Changat and Mathews, 1999]	Paths allowing chords of length two
total [Chepoi, 1997]	Paths allowing chords of length three
detour [Chartrand et al., 2003]	Maximum paths in the graph
all-path [Changat et al., 2001]	All paths

Given a graph $G = (V, E)$, the interval number in geodesic convexity of G corresponds

to the smallest set of vertices $S \subseteq V$ such that $I_g(S) = V$. Consequently, for every vertex $v \in V$, either $v \in S$, or v is on some shortest path between two vertices of S . The minimum cardinality of such an interval set of G is denoted by $\text{in}_g(G)$. This parameter was proposed by Harary et al. [1993], and it is also known in the literature as the *geodesic number*. In addition to the interval number, many other graph convexity parameters have been defined in the literature. For example, we can mention the hull number, which corresponds to the size of the smallest set S of vertices of the graph such that $H(S) = V$, proposed by Everett and Seidman [1985] for geodesic convexity. Besides the hull number, we can mention the convexity number [Chartrand et al., 2002], Carathéodory number [Carathéodory, 1911], Helly number [Helly, 1923], Radon number [Radon, 1921], rank [Jamison, 1981], general position [Manuel and Klavžar, 2018] and percolation time [Benevides and Przykucki, 2013] as the main graph convexity parameters studied in the literature.

With respect to the computational complexity, determining the geodesic number of a given graph G is NP-hard [Harary et al., 1993], even if G is bipartite [Dourado et al., 2010]. For further computational complexity results when the input graph G is restricted to some specific classes of graphs, see Dourado et al. [2010], Dourado et al. [2016], and Chakraborty et al. [2020]. Given the NP-hardness of determining the geodesic number, it is thus not expected that there are efficient algorithms to determine such parameter. In this case, as usual, an alternative is to propose computational methods to try to solve this problem of determining $\text{in}_g(G)$, for a given graph G .

To our knowledge, there are no mathematical programming formulations in the literature for determining the geodesic number of a graph. Despite the absence of formulations for the geodesic number, it is worth mentioning that there is a formulation to compute the interval number in the P_3 -convexity, which is known as the P_3 -interval number, proposed by Blaum and Marenco [2021] and Blaum and Marenco [2023]. However, no computational experiments have been presented to evaluate its performance.

Thus, aiming to address the literature gap regarding the geodesic number, we propose two integer programming formulations to compute this parameter: one exponential and one compact, alongside computational experiments to test their performance over randomly generated instances. In the following section, we present the proposed formulations. Section 3 is devoted to present some computational experiments we made and in Section 4 we present some concluding remarks and avenues for further research.

2. Mathematical Formulations

In this section, we present two integer programming models to determine the geodesic number of a given graph G on $n = |V(G)|$ vertices. The first one has a linear number of variables, but an exponential number of constraints with respect to n ; the second one is compact, having a quadratic number of variables and constraints (w.r.t. n).

2.1. Exponential Formulation

Given a graph G , we recall that a set $S \subseteq V(G)$ is an interval set if $I_g(S) = V(G)$. Define $CI(G)$ as the family of all non-interval sets, that is, $CI(G) = \{S \subseteq V(G) : I_g(S) \subset V\}$. To present the mathematical formulation with an exponential number of constraints, we use the following variables, for every $u \in V$:

$$x_u = \begin{cases} 1, & \text{if } u \text{ belongs to the interval set;} \\ 0, & \text{otherwise.} \end{cases}$$

Since our objective is to determine the smallest number of vertices that infect the whole graph, we want to minimize $\sum_{u \in V} x_u$. Let $S \in CI(G)$, and so $I_g(S) \subset V$. Note that at least one vertex from $V \setminus S$ must be selected for the solution, since for every $S' \subseteq S$, we have that

$I_g(S') \subseteq I_g(S) \subset V(G)$. In fact, these conditions, besides being necessary, are sufficient to define an interval set. Thus, we obtain the following formulation:

$$\min \sum_{u \in V} x_u \quad (1)$$

$$\begin{aligned} \text{s.t. } \sum_{u \in V \setminus S} x_u &\geq 1 & \forall S \in CI(G) \\ x_u &\in \{0, 1\} & \forall u \in V(G) \end{aligned} \quad (2)$$

Note that this formulation has a linear number of variables with respect to the number of vertices. However, it has an exponential number of constraints, which need to be separated when implementing the model.

2.2. Compact Formulation

Let H_w be the set containing all pairs of vertices $\{u, v\}$ such that w is on some geodesic between u and v . Below, we propose a compact formulation to compute the geodesic number. We adopt the x variables above and define the following binary variables, for every pair $u, v \in V$ of distinct vertices:

$$y_{uv} = \begin{cases} 1, & \text{if } u, v \text{ belong to the interval set;} \\ 0, & \text{otherwise.} \end{cases}$$

The relationship between variables x and y is such that $y_{uv} = x_u \cdot x_v$, for any two vertices u and v in V . This relationship is nonlinear; however, Fortet [1959a] and Fortet [1959b] show that it can be expressed by the following linear inequalities:

$$\begin{aligned} y_{uv} &\leq x_u, & \forall u, v \in V, u \neq v \\ y_{uv} &\leq x_v, & \forall u, v \in V, u \neq v \\ y_{uv} &\geq x_u + x_v - 1, & \forall u, v \in V, u \neq v \end{aligned} \quad (3)$$

Due to the objective function, we can remove the group (3) of constraints and still guarantee the relationship $y_{uv} = x_u \cdot x_v$ in an optimal solution. Finally, to ensure that the vertex selection represents an interval set, for every vertex $w \in V$, we need to guarantee that either w lies on some geodesic between two vertices of the solution, or w is selected for the interval set. In this way, we obtain the following constraints:

$$x_w + \sum_{\{u,v\} \in H_w} y_{uv} \geq 1, \quad \forall w \in V.$$

Thus, we obtain the following compact formulation:

$$\begin{aligned} \min \quad & \sum_{u \in V} x_u \\ \text{s.t. } \quad & y_{uv} \leq x_u, y_{uv} \leq x_v, & \forall u, v \in V, u \neq v, \\ & x_w + \sum_{\{u,v\} \in H_w} y_{uv} \geq 1, & \forall w \in V, \\ & x_u \in \{0, 1\}, & \forall u \in V, \\ & y_{uv} \in \{0, 1\}, & \forall u, v \in V, u \neq v. \end{aligned}$$

Unlike the exponential model, this formulation is compact and, with respect to the number of vertices of the input graph, it has a quadratic number of variables and constraints.

3. Computational Results

3.1. Instances

We generated two groups of instances for the computational tests: random arbitrary graphs, and random bipartite graphs. Note that the study in the class of bipartite graphs is relevant, since even in bipartite graphs, determining the geodesic number is NP-hard. Let us first describe how we generated such instances.

To generate each graph, two input parameters were used: the number of vertices $n \in \mathbb{N}^*$; and a density $d \in (0, 1)$ of a graph. Below we list the precise values of each parameter that we tested:

- Number of vertices n ($|V(G)|$) varying in $\{15, 30, 45, 60\}$;
- Density d of a graph in $\{0.2, 0.4, 0.6, 0.8\}$.

Let n and d be the two fixed values. For arbitrary random graphs, we initialize the graph of an instance as a spanning tree with n vertices to ensure graph connectivity (note that for a non-connected graph G , one can compute $\text{in}_g(G)$ by summing up the interval numbers of its connected components). The selection of edges composing such a tree is random, with each edge having an equal probability of being chosen. After creating the tree, we randomly select a pair $\{u, v\} \subset V(G)$, if $\{u, v\}$ is not an edge in the initial tree, we add the edge $\{u, v\}$ to the graph. We repeat this process until the graph has $\lfloor d \times \frac{n(n-1)}{2} \rfloor$ edges.

For random bipartite graphs, we partition the vertex set as $V = V_1 \cup V_2$, where $V_1 = \{1, 2, \dots, \lfloor \frac{n}{2} \rfloor\}$ and $V_2 = \{\lfloor \frac{n}{2} \rfloor + 1, \lfloor \frac{n}{2} \rfloor + 2, \dots, n\}$. To generate the spanning tree, the process is similar, but now ensuring that each selected edge has one extremity in V_1 and the other in V_2 . In this case, we repeat the process until the graph has $d \times (\lfloor \frac{n}{2} \rfloor \lceil \frac{n}{2} \rceil)$ edges. Note that it is not possible to generate a bipartite graph with $n = 15$ and $d = 0.2$. In this case, the graph should have 11 edges, preventing it from being connected. For each (n, d) combination, four instances were generated to ensure a more accurate and reliable evaluation during the tests.

3.2. Implementing the models

The models were solved using the CPLEX solver (IBM, Version 22.10). Their implementations were in the C++ programming language. Each experiment was conducted on a machine with a 12th Gen Intel® Core™ i3-12100 CPU running at 4.30 GHz, 16 GB RAM, and Ubuntu 22.04 operating system. The time limit for solving each instance for each formulation was 600 seconds (10 minutes).

To construct the sets H_w and compute $I_g(S)$, for $w \in V(G)$ and $S \subseteq V(G)$, we adapt the classic dynamic programming algorithm of Floyd [1962] and Warshall [1962]. It is applied to calculate all shortest paths between all pairs of vertices in the graph. Besides the cost of the shortest path, we also save the vertices in the shortest path between these pairs.

The exponential formulation has a potentially exponential number of constraints, therefore, implementing it directly into the solver becomes practically intractable. To circumvent this trouble, we adopt an iterative strategy to solve it with a subset of constraints (2). Specifically, we start the formulation only with the constraint $\sum_{u \in V} x_u \geq 2$, as at least 2 vertices must be selected for any interval set. Then, we use the solver to solve this relaxed (integer) model. Here, the term “relaxed” refers to the addition of only a subset of constraints (2). Thus, we get an integer solution defining a set of vertices, say S . We then check whether S is an interval. If so, we have that $|S|$ is the optimal solution. Otherwise, constraint (2) is violated for the set S . Then, we add the corresponding constraint to the current relaxed model, and the process is repeated. Note that the relaxation is with respect to the set of constraints and not to the integrality of the variables.

On the other hand, we can directly implement the compact formulation by adding all its components (variables, objective function, and constraints) directly to the solver. In the following section, we present the computational results obtained by each formulation on each set of randomly generated instances.

3.3. Experiments

In Tables 2 and 3, we present the results obtained by the compact and exponential formulations on random arbitrary graphs and random bipartite graphs, respectively. The columns ‘Group’, ‘ n ’, and ‘ d ’ respectively represent the group index (each representing one of the 4 generated instances), the number of vertices of the graph, and its density. In the columns ‘Time’, ‘Sol’, and ‘GAP’, we present the averages, for the 4 instances, of time (in seconds), the best feasible solution, and GAP obtained by each formulation. Note that, for cases where the GAP is ‘0.00’, the best solution obtained by the model is also optimal.

Table 2: Computational Experiments for arbitrary random graphs.

Group	n	d	Exponential			Compact		
			Time	Sol	GAP	Time	Sol	GAP
1	15	0,2	0,75	5,00	0,00	0,00	5,00	0,00
2	15	0,4	0,25	4,00	0,00	0,00	4,00	0,00
3	15	0,6	0,00	3,50	0,00	0,00	3,50	0,00
4	15	0,8	0,00	3,25	0,00	0,00	3,25	0,00
5	30	0,2	600,00	7,25	0,31	0,00	6,25	0,00
6	30	0,4	600,00	6,75	0,26	0,50	6,00	0,00
7	30	0,6	218,75	4,75	0,00	0,00	4,75	0,00
8	30	0,8	1,75	3,25	0,00	0,00	3,25	0,00
9	45	0,2	600,00	9,50	0,58	2,25	7,25	0,00
10	45	0,4	600,00	8,75	0,54	25,50	6,25	0,00
11	45	0,6	572,50	4,75	0,15	1,25	4,75	0,00
12	45	0,8	42,75	3,25	0,00	0,00	3,25	0,00
13	60	0,2	600,00	12,25	0,75	323,50	8,25	0,15
14	60	0,4	600,00	9,25	0,67	600,75	7,00	0,44
15	60	0,6	600,00	6,50	0,54	6,50	5,00	0,00
16	60	0,8	463,75	3,75	0,19	0,75	3,50	0,00
Average			343,78	5,98	0,25	60,06	5,08	0,04

The data analysis in Tables 2 and 3 reveals that the compact formulation outperformed the exponential formulation. The compact formulation not only solved a greater number of instances but also had the shortest average execution time. In particular, all instances with bipartite graphs were solved within the time limit by this formulation. However, for arbitrary graphs, instances with 60 vertices and low density (20% and 40%) were the only ones where the compact formulation failed to solve all instances.

Furthermore, the variation in density shows an interesting behavior for the compact formulation. For graphs with 45 and 60 vertices, the average solution time increases when the density grows from 20% to 40%. However, it starts to decrease from 40% onwards. This indicates that the most challenging graphs for this formulation are those with density around 40%. On the other hand, the exponential formulation did not perform satisfactorily compared to the compact one. For graphs with 30 vertices, this formulation reached the time limit for some instances. We also observed a

Table 3: Computational Experiments for bipartite graphs

Group	n	d	Exponential			Compact		
			Time	Sol	GAP	Time	Sol	GAP
1	15	0,4	0,25	2,75	0,00	0,00	2,75	0,00
2	15	0,6	0,00	3,50	0,00	0,00	3,50	0,00
3	15	0,8	0,00	3,00	0,00	0,00	3,00	0,00
4	30	0,2	458,75	6,75	0,24	0,00	6,50	0,00
5	30	0,4	132,75	4,25	0,00	0,25	4,25	0,00
6	30	0,6	6,50	4,00	0,00	0,25	4,00	0,00
7	30	0,8	1,00	3,00	0,00	0,00	3,00	0,00
8	45	0,2	600,00	7,00	0,41	0,50	6,25	0,00
9	45	0,4	366,00	4,75	0,13	0,75	4,00	0,00
10	45	0,6	381,00	4,50	0,10	1,25	4,00	0,00
11	45	0,8	13,75	3,00	0,00	0,50	3,00	0,00
12	60	0,2	600,00	7,50	0,60	10,25	5,75	0,00
13	60	0,4	600,00	7,25	0,58	13,25	4,75	0,00
14	60	0,6	600,00	5,75	0,48	6,25	4,25	0,00
15	60	0,8	453,25	3,75	0,19	2,25	3,75	0,00
Average			280,88	4,72	0,18	2,37	4,18	0,00

direct relationship between the resolution time and the value of the obtained solution: as the density increases, the value of the geodesic number decreases. When the optimal solution is smaller, eventually, the number of constraints (2) added is also smaller, which explains the reduction in the average execution time of the exponential formulation as the optimal value decreases.

In the Table 4 we present the solution value of the linear relaxation of the compact formulation and compare it with the integer solution on all tested instances. The columns 'Group', ' n ', and ' d ' have the same representation as in Tables 2 and 3. The columns 'Relax', 'Sol', and 'GAP' respectively represent the averages, for the 4 instances, of the linear relaxation solution, the integer solution, and the relative GAP between the fractional solution and the integer solution, in each set of instances. We observed that, on average, the relative GAP is, respectively, 66% for arbitrary random graphs and 72% for bipartite graphs. Despite this large initial gap, almost all instances were resolved optimally, as shown in Tables 2 and 3. In the case of random graphs, the computing time (additional to that for solving the relaxation) was relatively large, indicating that it could be reduced by properly strengthening the formulation.

4. Conclusion

In this work, we propose two mathematical formulations to determine the geodesic number of a graph, one compact and the other exponential. We conducted computational experiments on random arbitrary graphs and random bipartite graphs. We observed that the compact formulation performed better than the exponential formulation in all tests.

We observed a relationship between density and the geodesic number: as the graph density increases, the value of the parameter decreases. Thus, we determined relationships between time, density, and solution value for each formulation. We found that graphs with 40% density required the most computational time for the compact formulation. At this density, random graphs with 60 vertices were the only instances where the formulation failed to prove optimality of the found solutions, indicating that graphs with this density are more difficult for this model. For the exponential formulation, we related the required time with the solution value. As the solution value decreases,

Table 4: Computational experiments for the linear relaxation of the compact formulation

Group	n	d	Random Graphs			Bipartite Graphs		
			Relax	Sol	GAP	Relax	Sol	GAP
1	15	0,2	4,61	5,00	0,08	-	-	-
2	15	0,4	2,74	4,00	0,32	2,08	2,75	0,24
3	15	0,6	2,12	3,50	0,39	1,73	3,50	0,51
4	15	0,8	1,96	3,25	0,40	1,10	3,00	0,63
5	30	0,2	3,78	6,25	0,40	4,90	6,50	0,25
6	30	0,4	1,53	6,00	0,75	1,26	4,25	0,70
7	30	0,6	0,91	4,75	0,81	0,59	4,00	0,85
8	30	0,8	0,93	3,25	0,71	0,41	3,00	0,86
9	45	0,2	2,57	7,25	0,65	3,05	6,25	0,51
10	45	0,4	0,93	6,25	0,85	0,65	4,00	0,84
11	45	0,6	0,55	4,75	0,88	0,39	4,00	0,90
12	45	0,8	0,51	3,25	0,84	0,29	3,00	0,90
13	60	0,2	1,90	8,25	0,77	1,31	5,75	0,77
14	60	0,4	0,66	7,00	0,91	0,44	4,75	0,91
15	60	0,6	0,37	5,00	0,93	0,28	4,25	0,93
16	60	0,8	0,37	3,50	0,89	0,20	3,75	0,95
Average			1,65	5,08	0,66	1,25	4,18	0,72

the number of (2) inequalities added tends to be smaller, which causes the average resolution time to also reduce.

As future work, there are at least two lines for research. The first line of improvement is to enhance the formulations to solve larger instances. For the compact formulation, we will propose valid inequalities to strengthen its linear relaxation and, if efficient, use advanced integer programming methods: Lagrangian relaxation and decompositions. For the exponential formulation, we will seek alternative ways to add (2) constraints, as adding a single constraint per iteration proved inefficient. Indeed, using this strategy only instances with 15 vertices were solved to optimality within the time limit. One possibility is to start the resolution process and iteratively add a subset of the (2) constraints. The other line of improvement involves adapting the formulations to determine the interval number in other convexities, for example, P_3 and P_3^* . In this way, we could evaluate and compare their performances with the formulations of Blaum and Marenco [2021] and Blaum and Marenco [2023] for the P_3 convexity.

Acknowledgments

This study was partly financed by the CAPES-Brazil (Finance Code 001), CNPq-Brazil (404479/2023-5, 442977/2023-9, 312417/2022-5 and 313153/2021-3) and FUNCAP-Brazil (MLC-0191-00056.01.00/22, PS1-0186-00155.01.00/21, PGP-0192-00040.01.00/22, ITR-0214-00096.01.00/23).

To the reviewers for their constructive feedback and evaluations that contributed to the refinement of this work.

References

Araújo, J., Dourado, M., Protti, F., and Sampaio, R. (2023). *Uma introdução à convexidade em grafos*. 34º Colóquio Brasileiro de Matemática. IMPA.

- Araújo, R. T., Sampaio, R. M., and Szwarcfiter, J. L. (2013). The convexity of induced paths of order three. *Electronic Notes in Discrete Mathematics*, 44:109–114.
- Benevides, F. and Przykucki, M. (2013). On slowly percolating sets of minimal size in bootstrap percolation. *Electronic Journal of Combinatorics*, 20(2).
- Blaum, M. and Marengo, J. (2021). Valid inequalities and complete characterizations of the 2-domination and the p_3 -hull number polytopes. *XI Latin and American Algorithms, Graphs and Optimization Symposium*, p. 532–542.
- Blaum, M. and Marengo, J. (2023). Complete characterizations of the 2-domination and p_3 -hull number polytopes. *Discrete Applied Mathematics*, 338:30–45.
- Carathéodory, C. (1911). Über den variabilitätsbereich der fourier'schen konstanten von positiven harmonischen funktionen. *Rendiconti del Circolo Matematico di Palermo*, 32(1):193–217.
- Centeno, C. C., Dourado, M. C., and Szwarcfiter, J. L. (2009). On the convexity of paths of length two in undirected graphs. *Electronic Notes in Discrete Mathematics*, 32:11–18.
- Chakraborty, D., Foucaud, F., Gahlawat, H., Ghosh, S. K., and Roy, B. (2020). Hardness and approximation for the geodetic set problem in some graph classes. *Algorithms and discrete applied mathematics*, 12016:102–115.
- Changat, M., Klavzar, S., and Mulder, H. M. (2001). The all-paths transit function of a graph. *Czechoslovak Mathematical Journal*, 51:439–448.
- Changat, M. and Mathews, J. (1999). On triangle path convexity in graphs. *Discrete Mathematics*, 206:91–95.
- Chartrand, G., Garry, L., and Zhang, P. (2003). The detour number of a graph. *Utilitas Mathematica*, 64:97–113.
- Chartrand, G., Wall, C. E., and Zhang, P. (2002). The convexity number of a graph. *Graphs and Combinatorics*, 18:209–217.
- Chepoi, V. (1997). Peakless functions on graphs. *Discrete Applied Mathematics*, 73:175–189.
- Dourado, M. C., Penso, L. D., and Rautenbach, D. (2016). On the geodetic hull number of pk -free graphs. *Theoretical Computer Science*, 640:52–60.
- Dourado, M. C., Protti, F., Rautenbach, D., and Szwarcfiter, J. L. (2010). Some remarks on the geodetic number of a graph. *Discrete Mathematics*, 310:832–837.
- Dragan, F. F., Nicolai, F., and Brandstädt, A. (1999). Convexity and HHD-free graphs. *SIAM Journal on Discrete Mathematics*, 12:119–135.
- Duchet, P. (1988). Convex sets in graphs, ii. minimal path convexity. *Journal of Combinatorial Theory, Series B*, 44(3):307 – 316.
- Enriquez, E. L. and Junior, S. R. C. (2014). A short note on convexity and convex domination in $g[k_m]$. *Applied Mathematical Sciences, HIKARI, Ltd*, 8:5737 – 5741.
- Everett, M. G. and Seidman, S. B. (1985). The hull number of a graph. *Discrete Mathematics*, 57(3):217 – 223.

- Farber, M. and Jamison, R. E. (1986). Convexity in graphs and hypergraphs. *SIAM J. Algebraic Discrete Methods*, 7:433–444.
- Farber, M. and Jamison, R. E. (1987). On local convexity in graphs. *Discrete Mathematics*, 66: 231–247.
- Floyd, R. W. (1962). Algorithm 97: Shortest path. *Commun. ACM*, 5(6):345.
- Fortet, R. (1959a). Applications de l’algèbre de boole en recherché opérationnelle. *Revue française d’automatique d’informatique et de recherché opérationnelle*, 4:5–36.
- Fortet, R. (1959b). L’algèbre de boole et ses applications en recherche operationnelle. *Cahiers du centre d’études de recherche opérationnelle*, 4:5–36.
- Harary, F., Loukakis, E., and Tsouros, C. (1993). The geodetic number of a graph. *Mathematical and Computer Modelling*, 17:89–95.
- Harary, F. and Nieminem, J. (1981). Convexity in graphs. *Journal of Differential Geometry*, 16: 185–190.
- Helly, E. (1923). Über mengen konvexer körper mit gemeinschaftlichen punkten. *Jahresbericht der Deutschen Mathematiker-Vereinigung*, 32:175–176.
- Jamison, R. E. (1981). Partition numbers for trees and ordered sets. *Pacific Journal of Mathematics*, 96(1):115–140.
- Jamison, R. E. (1982). A perspective on abstract convexity: classifying alignments by varieties. *Convexity and Related Combinatorial Geometry*, New York.
- Manuel, P. and Klavžar, S. (2018). A general position problem in graph theory. *Bulletin of the Australian Mathematical Society*, 98(2):177–187.
- Nielsen, M. H. and Oellermann, O. R. (2011). Steiner trees and convex geometries. *SIAM J. Discrete Math*, 23:680–693.
- Pelayo, I. (2013). *Geodesic Convexity in Graphs*. SpringerBriefs in Mathematics. Springer New York.
- Radon, J. (1921). Mengen konvexer körper, die einen gemeinsamen punkt enthalten. *Mathematische Annalen*, 83(1):113–115.
- Warshall, S. (1962). A theorem on boolean matrices. *J. ACM*, 9(1):11–12.