Bull. Math. Soc. Sci. Math. Roumanie
Tome 65 (113), No. 4, 2022, 405–419

A variation of Boolean distance

by
Guillaume Ducoffe

Dedicated to Professor Tomescu on the occasion of his 80^{th} birthday

Abstract

Let $\mathcal{P}$ be a family of paths in a connected graph G so that there exists a uv-path in $\mathcal{P}$ for every choice of vertices u and v. The $\mathcal{P}$-interval $J_{\mathcal{P}}(u,v)$ is the union of vertex sets of all uv-paths in $\mathcal{P}$. We address the problem of maximizing $|J_{\mathcal{P}}(u,v)|$ over all vertices u and v, for various families $\mathcal{P}$ of paths.

- First, the Boolean distance between two vertices u and v, in a connected graph G, equals the total number of vertices on uv-paths. The *Boolean diameter* of a connected graph is its maximum Boolean distance. We observe that the Boolean diameter of any graph G can be computed in linear time, due to an elegant characterization of Boolean distances by [Harary, Melter, Peled & Tomescu; Discr. Math. 1982].
- Then, we restrict our study to induced paths. Unless P=NP, we cannot compute in polynomial time the maximum number of vertices on all induced paths between two vertices. However, on the positive side, we can solve this problem in polynomial time on graphs of bounded clique-width, and even in linear time on bounded-treewidth graphs and distance-hereditary graphs.
- Finally, we introduce the *interval semidistance* for graphs, that only accounts for vertices on shortest paths. Equivalently, the interval semidistance between two vertices u and v, in a connected graph G, equals the number of vertices that are metrically between u and v. The *interval diameter* (maximum interval semidistance) can be computed in $\mathcal{O}(n^3)$ time for n-vertex connected graphs. We prove that in contrast to Boolean diameter there is an $\Omega(n^{2-o(1)})$ time lower bound for this problem under the Strong Exponential-Time Hypothesis, even for graphs with only $n^{1+o(1)}$ edges. However, on the positive side, we present linear-time algorithms for computing the interval diameter within various graph classes with tree-likeness properties, such as: trees, cacti, block graphs and distance-hereditary graphs.

Key Words: Boolean distance, intervals of graphs, transit functions, NP-hardness, SETH-hardness.
2010 Mathematics Subject Classification: Primary: 05C85, 05C12.

1 Introduction

We refer to [3] and [19] for standard textbooks in Graph Theory and Complexity Theory, respectively. Unless stated otherwise, all graphs considered are unweighted, undirected,

simple (*i.e.*, without loops or multiple edges) and connected. Given a graph $G = (V, E)$, let $n = |V|$ be its order (number of vertices) and $m = |E|$ be its size (number of edges). A path is a sequence of pairwise different vertices such that every two consecutive vertices are adjacent. It is called a uv-path if it starts and ends with vertices u and v, respectively. Real-life communication networks can be conveniently modelled as a graph [6]. Therefore, studying the properties of various families of paths in a graph is of utmost importance. Let $\mathcal{P}$ be a family of paths so that there exists a uv-path in $\mathcal{P}$ for every possible choice of vertices u and v. For concreteness, we may see $\mathcal{P}$ as the set of all available roads for routing messages in a network. For every vertices u and v, let $J^G_{\mathcal{P}}(u, v)$ be the union of vertex sets of all uv-paths in $\mathcal{P}$. The mapping $(u, v) \to J^G_{\mathcal{P}}(u, v)$ is sometimes called a transit function [18]. We denote by $\sigma^G_{\mathcal{P}}(u, v) = |J^G_{\mathcal{P}}(u, v)|$ the number of vertices on all uv-paths in $\mathcal{P}$. We sometimes omit the superscript if G is clear from the context. In this paper, we are interested in computing $\Sigma_{\mathcal{P}}(G) = \max_{u,v \in V} \sigma^G_{\mathcal{P}}(u, v)$.

Let the length of a path be equal to its number of vertices minus one. Note that, if the length of any uv-path in $\mathcal{P}$ is at most ℓ, then there are at least $\lceil \sigma_{\mathcal{P}}(u, v)/(\ell + 1) \rceil$ uv-paths in $\mathcal{P}$. Therefore, if all paths in $\mathcal{P}$ have bounded length, the larger $\sigma_{\mathcal{P}}(u, v)$ the more roads there exist to route messages between u and v. Another motivation for computing $\Sigma_{\mathcal{P}}(G)$ could be in the context of broadcasting a message in a network. Indeed, if we send a message from u to v on all the uv-paths in $\mathcal{P}$, then the larger $\sigma_{\mathcal{P}}(u, v)$ the more vertices receive the message at once.

We consider in what follows three different families $\mathcal{P}$ of paths.

- As a starter, let $\mathcal{P}_{\text{all}}$ be the family of all paths. We denote by $\sigma_{\text{all}}(u, v)$ the number of vertices on all uv-paths. In [12], $\sigma_{\text{all}}(u, v)$ was introduced and studied under the name of *Boolean distance* between u and v. To our best knowledge, computational aspects of Boolean distances have not been studied before our work. Let the *Boolean diameter* $\Sigma_{\text{all}}(G)$ of a graph G equal its maximum Boolean distance. Our first result is that the Boolean diameter can be computed in linear time (Theorem 2). The proof of this result readily follows from the nice relationship between Boolean distances in a graph and its *block-cut tree* [12].

- Then, let $\mathcal{P}_{\text{ind}}$ be the family of all *induced* paths, where we recall that a path is induced if no two non-consecutive vertices are adjacent. We denote by $\sigma_{\text{ind}}(u, v)$ the number of vertices on all induced uv-paths. The existence of a uv-path is well-known to be equivalent to that of an induced uv-path. However, the same is not true for the vertices on uv-paths. For instance, let the bull graph be made of a triangle with vertices x, y, z and of two additional pendant vertices u, v with respective unique neighbours x, y. Then, we have $z \in J_{\text{all}}(u, v) \setminus J_{\text{ind}}(u, v)$. We refer to [17] for previous works on the properties of the intervals $J_{\text{ind}}(u, v)$, within the realm of induced path convexity [8]. Unsurprisingly, the restriction of our maximization problem to induced paths makes it intractable (Theorem 3). Standard applications of Courcelle's optimization theorem [4] allows us to derive a polynomial-time algorithm for the important special case of graphs of bounded clique-width.

- Finally, let $\mathcal{P}_{\text{sp}}$ be the family of all *shortest* paths. The union $J_{\text{sp}}(u, v)$ of vertex sets of all shortest uv-paths is sometimes called a metric interval or a geodesic interval. Metric intervals play an important role in the characterization of graph classes studied in Metric Graph Theory [1], some fellow-traveller properties on graphs [16], and

geodesic graph convexity [13]. The interval semidistance between u and v, denoted in what follows by $\sigma_{\rm sp}(u,v)$, equals $|J_{\rm sp}(u,v)|$. Note that in general, $\sigma_{\rm sp}()$ fails in satisfying the triangle inequality, and therefore it is not a distance function. For instance, let $K_{2,p}$ denote the complete bipartite graph with its partite sets of respective cardinalities 2 and p. Let u and v be the two vertices in the smallest partite set, and let x be an arbitrary vertex from the largest partite set. Then, $\sigma_{\rm sp}(u,v) = p+2$ whereas we have $\sigma_{\rm sp}(u,x) = \sigma_{\rm sp}(v,x) = 2$. Let the *interval diameter* $\Sigma_{\rm sp}(G)$ of G be equal to its maximum interval semidistance. While we prove that the interval diameter can be computed in cubic time (Theorem 4), we present a (conditional) quadratic lower bound for this problem (Theorem 6). However, on the positive side, we obtain linear-time algorithms for this problem on various graph classes.

All graph classes considered are first presented in Sec. 2. Our results for $\mathcal{P}_{\rm all}$, $\mathcal{P}_{\rm ind}$ and $\mathcal{P}_{\rm sp}$ are presented in Sec. 3, 4 and 5, respectively. We conclude this paper in Sec. 6.

2 Preliminaries

We start recalling a few standard notations. Let $G = (V, E)$ be a connected graph. For an arbitrary vertex v, let its (open) neighbourhood $N_G(v)$ contain every adjacent vertex to v. Let its closed neighbourhood be defined as $N_G[v] = N_G(v) \cup \{v\}$. The *distance* $d_G(u,v)$ between two vertices u and v equals the length of a shortest uv-path. We sometimes omit the subscript if G is clear from the context. A vertex v is *pendant* if $|N(v)| = 1$. Two vertices u, v are *twins* if $N(u) \setminus v = N(v) \setminus u$. In particular, we say that u and v are *true twins* if they are twins and adjacent, and they are *false twins* if they are twins and nonadjacent.

A *cut-vertex* is a vertex whose removal leaves the graph disconnected. A graph is 2-connected if it has no cut-vertex. Let a *block* in a graph be a maximal 2-connected subgraph. We may define a bipartite graph whose partite sets are the cut-vertices and the blocks of G respectively, so that there is an edge between every cut-vertex and every block that contains it. This graph is a tree, sometimes called the *block-cut tree* of G [11]. A *cactus* is a graph whose blocks are either edges or cycles. A *block graph* is a graph whose blocks are cliques. A *Gallai tree* is a graph whose blocks are either odd cycles or cliques.

A chordal graph is a graph with no induced cycle of length greater than three. Special cases of chordal graphs are the block graphs and the *split graphs*, where a split graph is a graph that can be vertex-partitioned in a clique and a stable set. A k-tree is a chordal graph with clique number at most $k+1$. We say that a graph has *treewidth* at most k if it is a (not necessarily induced) subgraph of a k-tree.

A graph is *distance-hereditary* if every induced path is also a shortest path. *Clique-width* is a parameter generalizing distance-hereditary graphs, whose formal definition (unused in this paper) can be found in [5]. In particular, every graph of bounded treewidth also has bounded clique-width [5] and every distance-hereditary graph has clique-width at most three [9].

3 Boolean distances

We revisit the beautiful characterization of Boolean distances in [12], deriving from the former a simple linear-time algorithm for Boolean diameter computation.

We need to introduce a few additional notations. First, being given two nodes x and y in any tree T, the unique xy-path of T is denoted by $P_T(x, y)$. Now, let $G = (V, E)$ be an arbitrary graph. For every vertex v, let us define $b(v)$ such that: $b(v) = v$ if v is a cut-vertex; otherwise, $b(v)$ is the unique block of G containing vertex v. We may see b as a mapping of V to the nodes of the block-cut tree of G. In general, b is neither injective nor surjective. For instance, if G is 2-connected, then $b(v) = G$ for every vertex v, and so, b is not injective. If G is a path with at least four nodes, then there are blocks composed of two cut-vertices, to which no node can be mapped by b. In this situation, b is not surjective.

Theorem 1 ([12])**.** *Let T be the block-cut tree of a graph $G = (V, E)$ and let u and v be arbitrary distinct vertices of G. Then, $J_{all}(u, v)$ is the union of vertex sets of all blocks on $P_T(b(u), b(v))$.*

By Theorem 1, computing the Boolean diameter of an arbitrary graph can be reduced to computing the diameter of a weighted tree. We prove it next.

Theorem 2. *The Boolean diameter of every $G = (V, E)$ can be computed in linear time.*

Proof. We assume without loss of generality that $n > 1$. We first describe our algorithm, before analyzing its running time and proving its correctness.

- We compute the block-cut tree T of G.
- For each node x of T, let $w(x) = -1$ if $x = v$ is a cut-vertex of G; otherwise, $x = B$ is a block of G, and we set $w(x) = |B|$.
- We compute $D(T, w) = \max_{x,y \in V(T)} \sum\{w(z) \mid z \in P_T(x, y)\}$.

The block-cut tree of G can be computed in linear time [14], and so can be the weight function w simply by scanning T once. Then, we are left computing the diameter of a weighted tree, that can also be done in linear time by using a folklore dynamic programming approach.

It remains to prove that $\Sigma_{\text{all}}(G) = D(T, w)$. For convenience, in what follows we write $\delta_{T,w}(x, y) = \sum\{w(z) \mid z \in P_T(x, y)\}$ for nodes x and y of T. Let us consider arbitrary distinct vertices u and v of G. Let $x = b(u), y = b(v)$ be as defined earlier in this section. We call z_u, z_v the nodes representing the first and last blocks on $P_T(x, y)$ (ordering nodes on the path from x to y). We claim that $\sigma_{\text{all}}(u, v) = \delta_{T,w}(z_u, z_v)$. In order to prove the claim, let $P_T(z_u, z_v) = (z_u = B_0, c_1, B_1, \ldots, c_\ell, B_\ell = z_v)$ where each c_i is a cut-vertex and each B_j is a block. We have $\delta_{T,w}(z_u, z_v) = \sum_{i=0}^{\ell} |B_i| - \ell = |B_0| + \sum_{i=1}^{\ell} (|B_i| - 1) = |B_0| + \sum_{i=1}^{\ell} |B_i \setminus \{c_i\}| = \left|\bigcup_{i=0}^{\ell} B_i\right|$. The claim now follows from Theorem 1. In particular, this implies $\Sigma_{\text{all}}(G) \leq D(T, w)$.

Let us prove that conversely, $\Sigma_{\text{all}}(G) \geq D(T, w)$. We first assume that G is 2-connected. In this situation, T is reduced to a single node with label G. Therefore, $D(T, w) = n$. Since $n > 1$, there exist at least two distinct vertices u and v. By Theorem 1, $J_{\text{all}}(u, v) = V$. Therefore, $\Sigma_{\text{all}}(G) = n = D(T, w)$. From now on, we assume that G has at least two blocks.

Let x and y be arbitrary nodes of T such that $\delta_{T,w}(x,y) = D(T,w)$. We first observe that x, y are blocks of G. Indeed, if x (resp., y) were a cut-vertex, then path $P_T(x,y)$ could be extended with a neighbour of x (resp., y) and this neighbour has a positive cost, thus contradicting the maximality of $D(T,w)$. Furthermore, we claim that there always exists a pair of *distinct* blocks x and y such that $\delta_{T,w}(x,y) = D(T,w)$. Indeed, for any given block x, there exists another distinct block y because we assume that G is not 2-connected. Since the path in T between x and y alternates between cut-vertices, of weight equal to -1, and blocks of weight at least 1, $\delta_{T,w}(x,y) \geq w(x) = \delta_{T,w}(x,x)$. Thus, we may assume in what follows that x and y are different. Let the path $P_T(x,y)$ start with edge xc_x and end with edge $c_y y$, where c_x and c_y are cut-vertices in blocks x and y, respectively. We pick a vertex u in block x such that $u \neq c_x$. Note that either u is a cut-vertex and so $b(u) = u$, or $b(u) = x$. Similarly, we pick a vertex v in block y such that $v \neq c_y$. According to our choices for u and v, the first and last blocks on $P_T(b(u), b(v))$ must be x and y. As a result, $D(T,w) = \delta_{T,w}(x,y) = \sigma_{\text{all}}(u,v) \leq \Sigma_{\text{all}}(G)$. □

4 The case of induced paths

Our main result in this section is that unless P=NP, we cannot compute in polynomial time the maximum number of vertices on all induced paths between any two vertices (Theorem 3). We start with the following simple reduction.

Lemma 1. *There is a polynomial-time reduction from the problem of computing $\sigma^G_{ind}(u,v)$ for a prescribed pair of vertices (u,v) in some graph G to the problem of computing $\Sigma_{ind}(G')$ for some graph G'.*

Proof. Let P_u, P_v be paths with $2n$ nodes each and respective end-vertices u', x and v', y. We construct the graph G' from G, P_u, P_v by adding the two edges ux, vy. In doing so, we get $\Sigma_{\text{ind}}(G') = \sigma^{G'}_{\text{ind}}(u', v') = 4n + \sigma^G_{\text{ind}}(u,v)$. □

Theorem 3. *The problem of computing $\Sigma_{ind}(G)$ for a given graph G cannot be solved in polynomial time, unless P=NP.*

Proof. The three-in-a-path problem, that consists in deciding whether there exists an induced path with three prescribed vertices, is NP-complete (*e.g.*, see [10]). It implies that unless P=NP, being given three vertices u, v and w we cannot decide in polynomial time whether vertex w lies on some induced uv-path. Let $\sigma^G_{\text{ind}}(u,v), \sigma^{G\setminus w}_{\text{ind}}(u,v)$ denote the number of vertices on all induced uv-paths in G and $G \setminus w$, respectively. Then, vertex w lies on some induced uv-path if and only if we have $\sigma^G_{\text{ind}}(u,v) > \sigma^{G\setminus w}_{\text{ind}}(u,v)$. As a result, unless P=NP we cannot compute in polynomial time the number of vertices on all induced paths between two prescribed vertices. We are done applying Lemma 1. □

Recall that the MS_1 logic is a fragment of second-order logic where the second-order quantification is limited to quantification over sets. The formulas that are written in MS_1

logic for graphs use lower-case variables $u, v, \ldots$ to represent vertices and upper-case variables $X, Y, \ldots$ to represent vertex-subsets. The atomic formulas available are set containment denoted $x \in X$ and the adjacency relation - denoted $adj(u, v)$. By Courcelle's Theorem [4], if the vertex-subsets in a given family can be expressed in MS_1 logic, then a maximum-cardinality subset in this family can be computed in polynomial time (resp., in linear-time) on graphs of bounded clique-width (resp., on bounded-treewidth graphs). It implies the following result.

Proposition 1. *If $G = (V, E)$ has bounded clique-width (resp., bounded treewidth), then we can compute $\Sigma_{ind}(G)$ in polynomial time (resp., in $\mathcal{O}(n)$ time).*

Proof. Let us first assume to be given a MS_1 formula $\varphi(u, v, x)$ expressing the property for a vertex x to belong to $J_{\rm ind}(u, v)$. Then, the MS_1 formula $\psi(X) = \exists u \exists v \, (x \in X \iff \varphi(u, v, x))$ expresses the property for a vertex-subset X to be an interval $J_{\rm ind}(u, v)$ for some vertices u and v. In particular, using Courcelle's theorem [4], we can compute such an interval of maximum cardinality. As a result, it now suffices to prove that $J_{\rm ind}(u, v)$ can be expressed in MS_1 logic. We do so by exhibiting the following MS_1 formula (for $u \neq v$):

$$\begin{aligned}\varphi(u,v,x) = \exists P(\; & u, v, x \in P \\ & \wedge \forall Q \subset P \; (\; (\exists r \in Q \wedge \exists r' \in P \setminus Q) \implies (\exists s \in Q \; \exists t \in P \setminus Q \; adj(s,t))) \\ & \wedge \forall r, s, t, w \in P \; (adj(w,r) \wedge adj(w,s) \wedge adj(w,t) \implies r = s \vee s = t \vee t = r) \\ & \wedge \forall s, t \in P \; (adj(u,s) \wedge adj(u,t) \implies s = t) \\ & \wedge \forall s, t \in P \; (adj(v,s) \wedge adj(v,t) \implies s = t) \;).\end{aligned}$$

The second line ensures that P is a connected subset. The third line ensures that the maximum degree of a vertex in P is at most two. Hence, P is either a cycle or a path. The last two lines ensure that both u and v have at most one neighbour in P. Overall, P must be an induced uv-path containing x. □

In Sec. 5, we present a linear-time algorithm for computing $\Sigma_{\rm sp}(G)$ for every distance-hereditary graph G (see Theorem 8). Since every induced path in a distance-hereditary graph G is also a shortest-path, we obtain the following result.

Proposition 2. *If $G = (V, E)$ is distance-hereditary, then we can compute $\Sigma_{ind}(G)$ in linear time.*

5 Interval semidistance

The main section of this paper is devoted to the computational aspects of the interval semidistance. In Sec. 5.1, we present a simple cubic-time algorithm for computing the interval diameter of any graph. We complete this positive result with a conditional quadratic time lower bound in Sec. 5.2. We end up the section presenting linear-time algorithms for special graph classes, in Sec. 5.3.

5.1 A polynomial-time algorithm

Theorem 4. *For every $G = (V, E)$, we can compute $\Sigma_{sp}(G)$ in $\mathcal{O}(n^3)$ time.*

Proof. We compute all distances $d_G(u, v)$, for every vertices u and v, in total $\mathcal{O}(nm)$ time. Then for every $u, v, w \in V$, we test in $\mathcal{O}(1)$ time whether $d_G(u, v) = d_G(u, w) + d_G(w, v)$. Doing so, for every vertices u and v we can compute $J_{\text{sp}}(u, v)$ (and so, $\sigma_{\text{sp}}(u, v)$) in $\mathcal{O}(n)$ time. We are done as there are only $\mathcal{O}(n^2)$ pairs (u, v) of vertices to be considered. □

We left open whether there is an $\mathcal{O}(nm)$-time algorithm for computing the interval diameter. This would be conditionally optimal due to the hardness result presented in the next Sec. 5.2.

5.2 SETH-hardness

Recall that k-SAT is the problem of deciding whether a given CNF formula, with at most k literals per clause, is satisfiable. The *Strong Exponential-Time Hypothesis* (SETH) posits that for every $\epsilon > 0$, there exists some k such that we cannot solve k-SAT in $(2 - \epsilon)^n n^{\mathcal{O}(1)}$ time, where n is the number of variables [15]. The Orthogonal Vectors problem (OV) takes as inputs two families A and B of n sets over a common universe C, and it asks whether there exist sets $a \in A,\ b \in B$ such that $a \cap b = \emptyset$.

Theorem 5 ([21])**.** *Under SETH, for every $\epsilon > 0$, there exists a constant $c > 0$ such that we cannot solve* OV *in $\mathcal{O}(n^{2-\epsilon})$ time, even if $|C| \leq c \log n$.*

Theorem 6. *Under SETH, for every $\epsilon > 0$, we cannot compute $\Sigma_{sp}(G)$ in $\mathcal{O}(n^{2-\epsilon})$ time, even if G is a split graph with at most $cn \log n$ edges for some constant c depending on ϵ.*

Proof. Let (A, B, C) be an instance of the OV problem. For convenience, let $d = |C|$. We may assume that there is no empty subset in A (resp., in B) nor any subset equal to C. Then, let $A_1, A_2, \ldots, A_{d-1}$ partition A so that every set of A_i has cardinality equal to i. In the same way, let $B_1, B_2, \ldots, B_{d-1}$ partition B so that every set of B_j has cardinality equal to j. For every $1 \leq i, j \leq d-1$ such that both A_i and B_j are nonempty, we construct a split graph $G_{i,j}$ as follows:

- The vertex-set is $A_i \cup B_j \cup C \cup \{x_A, y_B\}$;
- The subset $C \cup \{x_A, y_B\}$ is a clique, and the subset $A_i \cup B_j$ is an independent set;
- For every $a \in A_i$ and for every $x \in C$, we add an edge ax if and only if $x \in a$. Similarly, for every $b \in B_j$ and for every $y \in C$, we add an edge by if and only if $y \in b$.
- Finally, there is an edge between x_A and every $a \in A_i$. In the same way, there is an edge between y_B and every $b \in B_j$.

We observe that this above graph $G_{i,j}$ can be computed in $\mathcal{O}(nd + d^2)$ time.

We claim that $\Sigma_{\text{sp}}(G_{i,j}) = i + j + 4$ if and only if there exist $a \in A_i,\ b \in B_j$ such that $a \cap b = \emptyset$. Indeed, let u and v be arbitrary vertices of $G_{i,j}$. If u and v are adjacent, then $\sigma_{\text{sp}}(u, v) = 2$. Otherwise, if $d(u, v) = 2$, then $J_{\text{sp}}(u, v) = \{u, v\} \cup (N(u) \cap N(v))$. Since

$C \cup \{x_A, y_B\}$ is a clique, at least one of u or v, say it is u, belongs to $A_i \cup B_j$. Therefore, $\sigma_{\rm sp}(u,v) \leq 3 + \max\{i,j\}$ because the degree of u is at most $\max\{i,j\}+1$. From now on, we assume that $d(u,v) \geq 3$. We may assume by symmetry that $u \in A_i,\ v \in B_j$. In this situation, $u \cap v = \emptyset$, $d(u,v) = 3$, $J_{\rm sp}(u,v) = N[u] \cup N[v]$, and therefore $\sigma_{\rm sp}(u,v) = 4+i+j$, thus proving our claim.

Overall, in order to test whether there are $a \in A,\ b \in B$ such that $a \cap b = \emptyset$, it suffices to compute $\Sigma_{\rm sp}(G_{i,j})$ for every $1 \leq i,j \leq d-1$. Suppose by contradiction that the interval diameter can be computed in $\mathcal{O}(n^{2-\epsilon})$ time, for some $\epsilon > 0$. Then, we can solve OV in $\mathcal{O}(nd^3 + d^4 + d^2 n^{2-\epsilon})$ time which, in the special case $d \leq c \log n$ for some $c > 0$, is in $\mathcal{O}(n^{2-\epsilon'})$ time for some $0 < \epsilon' < \epsilon$. Assuming SETH, the latter contradicts Theorem 5. □

5.3 Linear-time algorithms on some special graph classes

Trees. A graph is geodetic if there exists a unique shortest path between every two vertices. We start making the following useful observation:

Lemma 2. *For every two vertices u and v in a geodetic graph G, $\sigma_{sp}(u,v) = d_G(u,v) + 1$.*

Proof. It suffices to observe that $J_{\rm sp}(u,v)$ is reduced to the vertex set of a unique shortest path of length $d_G(u,v)$, and so, with $d_G(u,v)+1$ vertices. □

By Lemma 2, the interval diameter can be computed in linear time on any class of geodetic graphs for which there exists a linear-time diameter computation algorithm.

Corollary 1. *The interval diameter can be computed in linear time on trees, block graphs and Gallai trees.*

Cacti. In general, a cactus is not a geodetic graph. However, we can design a simple dynamic programming approach on its block-cut tree for computing the interval diameter. We start with the following reduction rule:

Lemma 3. *If c is a cut-vertex in a graph $G = (V,E)$, and vertices u and v are in separate connected components of $G \setminus c$, then $\sigma_{sp}(u,v) = \sigma_{sp}(u,c) + \sigma_{sp}(c,v) - 1$.*

Proof. The result follows from the property that every uv-path must go by c. In particular, every shortest uv-path is the union of a shortest uc-path with a shortest cv-path. □

Let us consider the following weighted version of our problem. A graph $G = (V,E)$ is given together with a vertex-weight function β. For every $v \in V$, we further assume $\beta(v) \geq 0$. Let us define, for every vertices u and v:

$$\sigma_{\rm sp}(u,v,\beta) = \begin{cases} \beta(u) + \sigma_{\rm sp}(u,v) + \beta(v) \text{ if } u \neq v \\ \beta(u) + 1 \text{ if } u = v. \end{cases}$$

Similarly, let us define:

$$d(u,v,\beta) = \begin{cases} \beta(u) + d(u,v) + \beta(v) \text{ if } u \neq v \\ \beta(u) \text{ if } u = v. \end{cases}$$

The weighted interval diameter of G equals $\Sigma_{\rm sp}(G,\beta) = \max_{u,v\in V} \sigma_{\rm sp}(u,v,\beta)$. Its weighted diameter equals $diam(G,\beta) = \max_{u,v\in V} d(u,v,\beta)$. We will use the following result:

Lemma 4 ([20])**.** *The weighted diameter of a cycle can be computed in linear time for every nonnegative vertex-weight function β.*

The proof of Lemma 4 can be found in Section 2 of Reference [20]. The authors' definition of weighted diameter is slightly different than ours because they only consider $\max_{u\neq v} d(u,v,\beta)$ whereas we also allow u,v to be equal. However, the difference is not that important because we can separately compute $\max_{u\in V} d(u,u,\beta) = \max_{u\in V} \beta(u)$ in linear time. At first glance, Lemma 4 seems more general than the results from [20]. The reason is that Section 2 of Reference [20] is devoted to the computation of the (unweighted) diameter of *unicycle graphs*, *i.e.*, graphs G with a unique cycle C. More precisely, it is well-known that if we remove from a unicycle graph G all edges from its cycle C, then we obtain a forest whose each tree is rooted at some vertex of C. For every vertex v of cycle C, the authors of Reference [20] defined its weight $\beta(v)$ as the maximum distance between v and a node of the tree rooted at v. Then, they compute in $\mathcal{O}(|C|)$ time the weighted diameter of C with respect to β. Therefore, the results from [20] are only presented for specific vertex-weight functions β. However, these results hold for any function β with nonnegative *integer* weights: indeed, we can simulate any such function β by attaching to each vertex v of a cycle C a path of length $\beta(v)$. Furthermore, a closer look at the correctness proof of [20, Algorithm 1] (see also Observation 1 and Lemmas 3 to 6 in Reference [20]) shows that all arguments remain valid for general nonnegative vertex-weight functions β.

Corollary 2. *The weighted interval diameter of a cycle can be computed in linear time for every nonnegative vertex-weight function β.*

Proof. Let C_n be a cycle with n vertices, and let β be a vertex-weight function. We first describe our algorithm before proving its correctness.

- We compute $D_0 = diam(C_n,\beta)$, the weighted diameter. By Lemma 4, it can be done in linear time.
- If n is odd, then we output $D_0 + 1$.
- Else, let the vertices of C_n be enumerated in clockwise order, that results in the ordering $(v_0, v_1, \ldots, v_{n-1})$. Let $D_1 = \max\{\beta(v_i) + \beta(v_{i+n/2}) \mid 0 \leq i < n/2\}$. We output $\max\{D_0 + 1, D_1 + n\}$.

For every vertices u and v we always have that $\sigma_{\rm sp}(u,v,\beta) \geq d(u,v,\beta) + 1$. Therefore, $D_0 + 1$ is a lower bound on the weighted interval diameter. In order to prove correctness of the algorithm, it suffices to characterize the cases when it is larger than $D_0 + 1$. For that, let u and v satisfy $\sigma_{\rm sp}(u,v,\beta) > d(u,v,\beta) + 1$. Since there are at least two shortest uv-paths, n must be even and u,v are diametrically opposed. In particular, there is an i such that

$\{u, v\} = \{v_i, v_{i+n/2}\}$. Furthermore, $\sigma_{\rm sp}(u, v) = n$. It implies that if the weighted interval diameter is larger than $D_0 + 1$, then it is at most $D_1 + n$ and n must be even. Conversely, if n is even then $D_1 + n$ is a lower bound on the weighted interval diameter. □

Theorem 7. *The interval diameter can be computed in linear time for cacti.*

Proof. Let $G = (V, E)$ be a cactus, and let T denote its block-cut tree. It can be computed in linear time [14]. We may assume that G is not a cycle by Corollary 2, and therefore there is at least one cut-vertex. Let $c_0, c_1, \ldots, c_p$ be the cut-vertices of G. We root T in c_0. Then, for every $0 \leq i \leq p$, let V_i denote the union of vertex sets of all blocks in the subtree rooted at c_i.

We often use in what follows that after pre-processing a cycle C in $\mathcal{O}(|C|)$ time, we can compute $\sigma_{\rm sp}(u, v)$ in $\mathcal{O}(1)$ time for every vertices u and v. Indeed, $\sigma_{\rm sp}(u, v) = |C|$ if $|C|$ is even and u, v are diametrically opposed; otherwise, $\sigma_{\rm sp}(u, v) = d_C(u, v) + 1$.

Our algorithm proceeds in two main steps.

Step 1. For every cut-vertex c_i, let $C_i^1, C_i^2, \ldots, C_i^{k_i}$ denote its children nodes in T. For every $1 \leq j \leq k_i$, let $V_{i,j}$ be the union of vertex sets of all the blocks in the subtree rooted at C_i^j. We aim at computing $D_i = \max_{v \in V_i} \sigma_{\rm sp}(c_i, v)$. For that, it is sufficient to compute $D_{i,j} = \max_{v \in V_{i,j}} \sigma_{\rm sp}(c_i, v)$ for every $1 \leq j \leq k_i$. We do so by considering all the blocks in postorder. For a given block B, let the cut-vertex c_i be its parent node in T. We have $B = C_i^j$ for some $1 \leq i \leq k_i$. Then, we assign weights $\beta_j(v)$ to every vertex v of B so that:

$$\beta_j(v) = \begin{cases} D_{i'} - 1 \text{ if } v = c_{i'} \text{ is a cut-vertex different than } c_i \\ 0 \text{ otherwise.} \end{cases}$$

By Lemma 3, $D_{i,j} = \max\{\sigma_{\rm sp}(c_i, v) + \beta_j(v) \mid v \in B\}$.

Clearly, according to the above formula we can compute $D_{i,j}$ in $\mathcal{O}(|C_i^j|)$ time. As a result, the first step runs in total linear time.

Step 2. For every cut-vertex c_i, we consider each child block $C_i^1, C_i^2, \ldots, C_i^{k_i}$ sequentially. For every $1 \leq j \leq k_i$, we assign weights $\beta'_j(v)$ to the vertices of C_i^j so that:

$$\beta'_j(v) = \begin{cases} \max_{v' \in V_i \setminus V_{i,j}} \sigma_{\rm sp}(c_i, v') - 1 \text{ if } v = c_i \\ D_{i'} - 1 \text{ if } v = c_{i'} \text{ is a cut-vertex different than } c_i \\ 0 \text{ otherwise.} \end{cases}$$

Let $\Sigma_{i,j} = \Sigma_{\rm sp}(C_i^j, \beta'_j)$. Finally, let $\Sigma_i = \max\{D_i\} \cup \{\Sigma_{i,j} \mid 1 \leq j \leq k_i\}$.

Note that all the values $\beta'_1(c_i), \beta'_2(c_i), \ldots, \beta'_{k_i}(c_i)$ can be computed in $\mathcal{O}(k_i)$ time, simply by keeping track of the two largest values amongst $D_{i,1}, D_{i,2}, \ldots, D_{i,k_i}$. By Corollary 2, the weighted interval diameter $\Sigma_{i,j}$ can be computed in $\mathcal{O}(|C_i^j|)$ time. Hence, the second step runs in total linear time.

We output $\Sigma_{\rm sp}(G) = \max_{0 \leq i \leq p} \Sigma_i$.

Correctness. We first observe that $\Sigma_{i,j} \leq \Sigma_{\rm sp}(G)$, for every $0 \leq i \leq p$ and $1 \leq j \leq k_i$, that follows from repeated applications of Lemma 3 to all cut-vertices which are contained

in the block C_i^j. Similarly, we have $D_i \leq \Sigma_{\rm sp}(G)$ for every $0 \leq i \leq p$. Suppose by contradiction $\Sigma_{\rm sp}(G) > \max_{0 \leq i \leq p} \Sigma_i$. Let u and v be satisfying $\sigma_{\rm sp}(u,v) = \Sigma_{\rm sp}(G)$. We consider the deepest cut-vertex c_i such that $u, v \in V_i$. Note that $c_i \notin \{u, v\}$ because otherwise, $D_i = \Sigma_i = \Sigma_{\rm sp}(G)$. Let j be such that $u \in V_{i,j}$, and let u' be the unique vertex of $C_i^j \setminus \{c_i\}$ such that:

$$\begin{cases} u' = u \text{ if } u \text{ is a vertex of } C_i^j \\ u' = c_{i'} \text{ is a cut-vertex such that } u \in V_{i'} \text{ else.} \end{cases}$$

Let v' be the unique vertex of C_i^j such that:

$$\begin{cases} v' = c_i \text{ if } v \notin V_{i,j} \\ v' = v \text{ if } v \text{ is a vertex of } C_i^j \\ v' = c_{i''} \text{ is a cut-vertex of } C_i^j \setminus \{c_i\} \text{ such that } v \in V_{i''} \text{ else.} \end{cases}$$

Note that we cannot have $u' = v'$ because otherwise, $u' = v'$ would be a cut-vertex deeper than c_i and such that u, v are in its rooted subtree. By Lemma 3,

$$\begin{aligned} \sigma_{\rm sp}(u,v) &= \sigma_{\rm sp}(u,u') - 1 + \sigma_{\rm sp}(u',v) \\ &= \sigma_{\rm sp}(u,u') - 1 + \sigma_{\rm sp}(u',v') + \sigma_{\rm sp}(v',v) - 1 \\ &\leq \beta'_j(u') + \sigma_{\rm sp}(u',v') + \beta'_j(v') \\ &\leq \Sigma_{i,j}. \end{aligned}$$

A contradiction. □

Distance-hereditary graphs. We use the following characterization of distance-hereditary graphs in our algorithm.

Lemma 5 ([2])**.** *A graph is distance-hereditary if and only if it can be reduced to one vertex graph by a pruning sequence of one-vertex deletions: removing a pendant vertex or a single vertex from a pair of twin vertices. Moreover, if G is distance-hereditary, then such a pruning sequence can be computed in linear time.*

Based on the existence of a pruning sequence, we introduce the following reduction rules for distance-hereditary graphs. We need to introduce a more complicated weighted version of our problem. Namely, let β, γ be vertex-weight functions that only take nonnegative and positive values, respectively. Let us define, for every distinct vertices u and v:

$$\sigma_{\rm sp}(u,v,\beta,\gamma) = \beta(u) + \sum \{\gamma(w) \mid w \in J_{\rm sp}(u,v) \setminus \{u,v\}\} + \beta(v)$$

For a graph G with $n > 1$, let $\Sigma_{\rm sp}(G,\beta,\gamma) = \max_{u \neq v} \sigma_{\rm sp}(u,v,\beta,\gamma)$.

Lemma 6. *Let v be a pendant vertex in a graph $G = (V,E)$, let vertex u be its unique neighbour, and let β, γ be vertex-weight functions on G. We define a vertex-weight function β' on $G \setminus v$, such that:*

$$\beta'(x) = \begin{cases} \beta(x) \text{ if } x \neq u \\ \max\{\beta(u), \gamma(u) + \beta(v)\} \text{ if } x = u. \end{cases}$$

Let also γ' be the restriction of γ to $G \setminus v$. If $n > 2$, then we have

$$\Sigma_{sp}(G, \beta, \gamma) = \max\{\Sigma_{sp}(G \setminus v, \beta', \gamma'), \sigma^G_{sp}(u, v, \beta, \gamma)\}.$$

Proof. Let $x, y \in V$ be arbitrary distinct vertices. If $\{x, y\} \cap \{u, v\} = \emptyset$, then we have $\sigma^{G\setminus v}_{\rm sp}(x, y, \beta, \gamma) = \sigma^G_{\rm sp}(x, y, \beta', \gamma')$. From now on, we assume that $\{x, y\} \cap \{u, v\} \neq \emptyset$.

Let us consider the case when $x \in \{u, v\},\ y \notin \{u, v\}$. We have

$$\begin{aligned}
\sigma^{G\setminus v}_{\rm sp}(u, y, \beta', \gamma') &= \beta'(u) + \sum\{\gamma'(w) \mid w \in J_{\rm sp}(u, y) \setminus \{u, y\}\} + \beta'(y)\\
&= \max\{\beta(u), \gamma(u) + \beta(v)\} + \sum\{\gamma(w) \mid w \in J_{\rm sp}(u, y) \setminus \{u, y\}\} + \beta(y)\\
&= \max\{\beta(u) + \sum\{\gamma(w) \mid w \in J_{\rm sp}(u, y) \setminus \{u, y\}\} + \beta(y),\\
&\qquad \beta(v) + \gamma(u) + \sum\{\gamma(w) \mid w \in J_{\rm sp}(u, y) \setminus \{u, y\}\} + \beta(y)\}\\
&= \max\{\sigma^G_{\rm sp}(u, y, \beta, \gamma), \sigma^G_{\rm sp}(v, y, \beta, \gamma)\}.
\end{aligned}$$

As a result, we always have $\Sigma_{\rm sp}(G, \beta, \gamma) \geq \Sigma_{\rm sp}(G \setminus v, \beta', \gamma')$. In particular, we must have $\Sigma_{\rm sp}(G, \beta, \gamma) = \Sigma_{\rm sp}(G \setminus v, \beta', \gamma')$, unless maybe if $\Sigma_{\rm sp}(G, \beta, \gamma) = \sigma^G_{\rm sp}(u, v, \beta, \gamma)$. □

Lemma 7. *Let u and v be twins in a graph $G = (V, E)$, and let β, γ be vertex-weight functions on G. We define vertex-weight functions β', γ' on $G \setminus v$, such that:*

$$\beta'(x) = \begin{cases} \beta(x) \text{ if } x \neq u \\ \max\{\beta(u), \beta(v)\} \text{ if } x = u. \end{cases}$$

and

$$\gamma'(x) = \begin{cases} \gamma(x) \text{ if } x \neq u \\ \gamma(u) + \gamma(v) \text{ if } x = u \end{cases}$$

If $n > 2$, then we have $\Sigma_{sp}(G, \beta, \gamma) = \max\{\Sigma_{sp}(G \setminus v, \beta', \gamma'), \sigma^G_{sp}(u, v, \beta, \gamma)\}$.

Proof. The proof is quite similar to that of Lemma 6. Let $x, y \in V$ be arbitrary. There are several cases to be considered.

- Case $u, v \notin J_{\rm sp}(x, y)$. In this situation, $\sigma^G_{\rm sp}(x, y, \beta, \gamma) = \sigma^{G\setminus v}_{\rm sp}(x, y, \beta', \gamma')$.
- Case $u, v \in J_{\rm sp}(x, y) \setminus \{x, y\}$. In this situation, $\sigma^{G\setminus v}_{\rm sp}(x, y, \beta', \gamma') = (\sigma^G_{\rm sp}(x, y, \beta, \gamma) - \gamma(u) - \gamma(v)) + \gamma'(u) = \sigma^G_{\rm sp}(x, y, \beta, \gamma)$.
- Case $x \in \{u, v\},\ y \notin \{u, v\}$. In particular, $J_{\rm sp}(u, y) \setminus \{u\} = J_{\rm sp}(v, y) \setminus \{v\}$. As a result,

$$\begin{aligned}
\sigma^{G\setminus v}_{\rm sp}(u, y, \beta', \gamma') &= \beta'(u) + \sum\{\gamma'(w) \mid w \in J_{\rm sp}(u, y) \setminus \{u, y\}\} + \beta'(y)\\
&= \max\{\beta(u), \beta(v)\} + \sum\{\gamma(w) \mid w \in J_{\rm sp}(u, y) \setminus \{u, y\}\} + \beta(y)\\
&= \max\{\beta(u) + \sum\{\gamma(w) \mid w \in J_{\rm sp}(u, y) \setminus \{u, y\}\} + \beta(y),\\
&\qquad \beta(v) + \sum\{\gamma(w) \mid w \in J_{\rm sp}(v, y) \setminus \{v, y\}\} + \beta(y)\}\\
&= \max\{\sigma^G_{\rm sp}(u, y, \beta, \gamma), \sigma^G_{\rm sp}(v, y, \beta, \gamma)\}.
\end{aligned}$$

Therefore, we always have $\Sigma_{\rm sp}(G,\beta,\gamma) \geq \Sigma_{\rm sp}(G \setminus v, \beta', \gamma')$. In particular, we must have $\Sigma_{\rm sp}(G,\beta,\gamma) = \Sigma_{\rm sp}(G \setminus v, \beta', \gamma')$, unless maybe if $\Sigma_{\rm sp}(G,\beta,\gamma) = \sigma^G_{\rm sp}(u,v,\beta,\gamma)$. □

Theorem 8. *The interval diameter can be computed in linear time for distance-hereditary graphs.*

Proof. Let $G = (V,E)$ be a distance-hereditary graph. We may assume that $n > 1$. Let $V = (v_1, v_2, \ldots, v_n)$ such that, for every $1 \leq i < n$, vertex v_i is either pendant or it has a twin in the subgraph $G_i := G[\{v_i, v_{i+1}, \ldots, v_n\}]$. Such a pruning sequence can be computed in linear time by Lemma 5. For every $1 \leq i \leq n$, we set $\beta(v_i) = \gamma(v_i) = 1$. Doing so, we have $\Sigma_{\rm sp}(G) = \Sigma_{\rm sp}(G,\beta,\gamma)$. We set $\Sigma := 0$. Then, we consider all vertices v_i, for $i = 1, \ldots, n-2$ sequentially, and we proceed as follows:

- If v_i is a pendant vertex in G_i, then let v_j, for some $j > i$, be its unique neighbour. We set $\Sigma := \max\{\Sigma, \beta(v_i) + \beta(v_j)\}$. Note that $\beta(v_i) + \beta(v_j) = \sigma^{G_i}_{\rm sp}(v_i, v_j, \beta, \gamma)$. Then, we set $\beta(v_j) := \max\{\beta(v_j), \gamma(v_j) + \beta(v_i)\}$.

- Otherwise, v_i has a twin v_j, for some $j > i$, in G_i. We may assume v_j to be known because it can be computed at the same time as the pruning sequence, with no computational overhead [7]. If v_i, v_j are true twins, then we set $\Sigma := \max\{\Sigma, \beta(v_i) + \beta(v_j)\}$. Otherwise, we set $\Sigma := \max\{\Sigma, \beta(v_i) + \beta(v_j) + \sum\{\gamma(w) \mid w \in N(v_i) \cap V(G_i)\}\}$. Note that in both cases, $\Sigma := \max\{\Sigma, \sigma^{G_i}_{\rm sp}(v_i, v_j, \beta, \gamma)\}$. Finally, we set $\beta(v_j) = \max\{\beta(v_i), \beta(v_j)\}$, $\gamma(v_j) := \gamma(v_j) + \gamma(v_i)$.

Each step runs in $\mathcal{O}(|N_G(v_i)|)$ time, and therefore the total running time is linear. At the end of the for loop, we output $\Sigma_{\rm sp}(G) = \max\{\Sigma, \Sigma_{\rm sp}(G_{n-1}, \beta, \gamma)\}$. Correctness follows from repeated applications of either Lemma 6 or Lemma 7 at each step of the algorithm. □

6 Conclusion

In this paper, we initiated the study of the problem of computing the maximum cardinality of a $\mathcal{P}$-interval, with respect to various path families $\mathcal{P}$. We established the computational complexity of this problem for general paths, induced paths and shortest-paths. Other path families, in relation with prior studies on transit functions, might be worth investigating. Finally, it would be interesting to extend our approach and our techniques in this paper to some more graph classes. For instance, can the interval diameter be computed in (almost) linear time on bounded-treewidth graphs?

Acknowledgement *This work was supported by a grant of the Ministry of Research, Innovation and Digitalization, CCCDI - UEFISCDI, project number PN-III-P2-2.1-PED-2021-2142, within PNCDI III.*

References

[1] H. J. Bandelt, V. Chepoi, Metric graph theory and geometry: a survey, *Contemporary Mathematics*, **453**, 49-86 (2008).

[2] H. J. Bandelt, H. M. Mulder, Distance-hereditary graphs, *Journal of Combinatorial Theory, Series B*, **41 (2)**, 182-208 (1986).

[3] A. Bondy, U. S. R. Murty, *Graph Theory*, Springer, London (2008).

[4] B. Courcelle, J. A. Makowsky, U. Rotics, Linear time solvable optimization problems on graphs of bounded clique-width, *Theory of Computing Systems*, **33 (2)**, 125-150 (2000).

[5] B. Courcelle, S. Olariu, Upper bounds to the clique width of graphs, *Discrete Applied Mathematics*, **101 (1–3)**, 77-114 (2000).

[6] J. de Rumeur, *Communications dans les réseaux de processeurs*, Elsevier, Masson (1997).

[7] F. F. Dragan, H. M. Guarnera, Eccentricity function in distance-hereditary graphs, *Theoretical Computer Science*, **833**, 26-40 (2020).

[8] P. Duchet, Convex sets in graphs. II. Minimal path convexity, *Journal of Combinatorial Theory, Series B*, **44**, 307-316 (1988).

[9] M. C. Golumbic, U. Rotics, On the clique-width of some perfect graph classes, *International Journal of Foundations of Computer Science*, **11 (3)**, 423-443 (2000).

[10] R. Haas, M. Hoffmann, Chordless paths through three vertices., *Theoretical Computer Science*, **351 (3)**, 360-371 (2006).

[11] F. Harary, *Graph Theory*, Addison-Wesley (1969).

[12] F. Harary, R. A. Melter, U. N. Peled, I. Tomescu, Boolean distance for graphs, *Discrete Mathematics*, **39 (2)**, 123-127 (1982).

[13] F. Harary, J. Nieminen, Convexity in graphs, *Journal of Differential Geometry*, **16 (2)**, 185-190 (1981).

[14] J. Hopcroft, R. E. Tarjan, Algorithm 447: efficient algorithms for graph manipulation, *Communications of the ACM*, **16 (6)**, 372-378 (1973).

[15] R. Impagliazzo, R. Paturi, On the complexity of k-SAT, *Journal of Computer and System Sciences*, **62 (2)**, 367-375 (2001).

[16] A. O. Mohammed, F. F. Dragan, H. M. Guarnera, Fellow Travelers Phenomenon Present in Real-World Networks, *Complex Networks & Their Applications X. COMPLEX NETWORKS 2021. Studies in Computational Intelligence*, **1015**, 194-206 (2021).

[17] M. A. Morgana, H. M. Mulder, The induced path convexity, betweenness, and svelte graphs, *Discrete Mathematics*, **254 (1–3)**, 349-370 (2002).

[18] H. M. Mulder, Transit functions on graphs (and posets), *Convexity in Discrete Structures, RMS Lecture Notes Series*, **5**, 117-130 (2008).

[19] C. H. Papadimitriou, *Computational Complexity*, Addison-Wesley (1994).

[20] H. Wang, Y. Zhao, Algorithms for diameters of unicycle graphs and diameter-optimally augmenting trees, *Theoretical Computer Science*, **890**, 192-209 (2021).

[21] R. Williams, A new algorithm for optimal 2-constraint satisfaction and its implications, *Theoretical Computer Science*, **348 (2–3)**, 357-365 (2005).

Received: 28.09.2022
Revised: 29.10.2022
Accepted: 01.11.2022

Department of Computer Science, University of Bucharest, Romania
and National Institute for Research and Development in Informatics, Romania
E-mail: guillaume.ducoffe@fmi.unibuc.ro