Complexity of Maximum Cut on Interval Graphs

Ranendu Adhikary1 · Kaustav Bose2 · Satwik Mukherjee1 · Bodhayan Roy3

Received: 30 July 2021 / Revised: 4 June 2022 / Accepted: 18 June 2022 /
Published online: 14 January 2023
© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2023

Abstract
We resolve the longstanding open problem concerning the computational complexity of MAXCUT on interval graphs by showing that it is NP-complete.

Keywords Maximum cut · Interval graph · NP-complete

Mathematics Subject Classification 68Q17 · 90C27

1 Introduction

For a graph $G = (V, E)$, a cut is a partition of $V$ into two subsets. Any cut determines a cut set which is the set of all edges that have one endpoint in one part of the partition and the other endpoint in the other part of the partition. The size of a cut is the cardinality of its cut set. The maximum cut problem or MAXCUT asks for a cut of maximum size. In the decision version of the problem, we are given a graph $G$ and an integer $k$, and the
question is whether \( G \) has a cut of size at least \( k \). This is a fundamental and well-known NP-complete problem [22, pp. 37–79]. The weighted version of the problem is one of Karp’s original 21 NP-complete problems [33]. In the weighted version, each edge is associated with a real number, its weight. The size of a cut is defined in this case as the sum of the weights of the edges in its cut set. Besides its theoretical importance, the MaxCut problem has applications in VLSI circuit design [13], statistical physics [5] etc.

In view of the hardness, the problem has been studied from various points of view, such as approximation algorithms, fixed-parameter tractability and special graph classes. Regarding approximation algorithms, the problem is known to be APX-hard [43], meaning that there is no polynomial-time approximation scheme (PTAS) for the problem unless \( P = NP \). In fact, it is NP-hard to approximate MaxCut within a factor of \( 16/17 \approx 0.941 \) [27, 45]. The naive random partition of the vertices gives a simple 0.5-approximation algorithm. This algorithm can be derandomized in polynomial time using method of conditional expectations, therefore giving a deterministic polynomial-time 0.5-approximation algorithm. For many years this remained the best known approximation ratio achievable in polynomial-time for the MaxCut problem.

In 1994 Michel Goemans and David Williamson in their celebrated paper [23] gave an algorithm based on semidefinite programming (SDP), which achieves an approximation ratio of \( \approx 0.878 \). If the Unique Games conjecture [35] is true, then this is the best possible approximation ratio for MaxCut. As for fixed parameter tractability, several parameterized algorithms have been proposed over the years with respect to different parameters including solution size [44], solution size above guaranteed values [17, 39, 40], tree-width [9], clique-width [20] and crossing number [36]. Regarding the computational complexity of MaxCut on restricted graph classes, the problem remains NP-hard for cubic graphs [6], split graphs [9], co-bipartite graphs [9], unit disk graphs [18], and total graphs [25]. On the positive side, polynomial time algorithms are known for planar graphs [26], line graphs [25], graphs not contractible to \( K_5 \) [4], and graphs with bounded tree-width [9].

This paper concerns the computational complexity of MaxCut on interval graphs. An interval graph is the intersection graph of a collection of intervals on the real line. The class of interval graphs is widely regarded as an important graph class with many real-world applications. Interval graphs arise naturally in modelling problems that involve temporal reasoning, e.g. scheduling problems. Interval graphs are also extensively used in bioinformatics (e.g. DNA mapping [47], protein sequencing [31]) and mathematical biology (e.g. food webs in population biology [15]). It is well known that many classical NP-complete problems like colourability [24], Hamiltonian cycle [34], minimum dominating set [14], minimum feedback vertex set [38], minimum vertex cover [41] and maximum clique [29] are polynomial time solvable for interval graphs. This is because interval graphs are well structured graphs with many nice properties and decomposition models that are often exploited to design efficient dynamic programming or greedy algorithms. Few problems that are known to be NP-hard in interval graphs include optimal linear arrangement [16], achromatic number [7], harmonious colouring [3], geodetic set [12], minimum sum colouring [42], metric dimension [21], identifying code [21], and locating-dominating set [21].
The weighted version of MaxCut is NP-complete for interval graphs. This is because the problem is NP-complete even for complete graphs [32]. Surprisingly, the computational complexity of unweighted MaxCut for interval graphs is a longstanding open problem. The first time that the problem was mentioned as open was probably in 1985 [30]. No polynomial time algorithm is known even for the subclass of unit interval graphs. There are however two previous papers reporting polynomial time algorithms solving MaxCut for unit interval graphs: one by Bodlaender et al. [10] and another one by Boyacı et al. [11]. The result of the first paper was disproved later by the authors themselves in [8]. The algorithm of the second paper has been recently shown to be flawed as well by Kratochvíl et al. in [37]. In this paper, we resolve the longstanding open problem concerning the status of (unweighted) MaxCut on interval graphs by showing that it is NP-complete.

2 Preliminaries

For any simple undirected graph $G = (V, E)$, a cut is a partition of $V$ into two subsets $A$ and $B$, i.e., $V = A \cup B$ and $A \cap B = \emptyset$. The corresponding cut set is the set of all edges that have one endpoint in $A$ and the other endpoint in $B$, i.e., the set $\{(u, v) \in E \mid (u \in A, v \in B) \lor (u \in B, v \in A)\}$. The size of the cut is the cardinality of its cut set. A typical instance of the decision version of MaxCut consists of a simple undirected graph $G = (V, E)$ and an integer $k$ such that $1 \leq k \leq |E|$. $(G, k)$ is a yes-instance of MaxCut if and only if $G$ has a cut of size at least $k$.

Interval graphs are the intersection graphs of intervals on the real line. Formally, $G = (V, E)$ is said to be an interval graph if there is a set $S$ of intervals on the real line and a bijection $\varphi : V \rightarrow S$ such that $u, v \in V$ are adjacent if and only if $\varphi(u) \cap \varphi(v) \neq \emptyset$. Cubic graphs are graphs in which all vertices have degree three.

3 NP-Completeness

In this section, we show that MaxCut is NP-complete on interval graphs. MaxCut is known to be NP-complete on cubic graphs [6]. We reduce MaxCut on cubic graphs to MaxCut on interval graphs.

3.1 Outline of the Reduction

We now give a brief and intuitive idea of the reduction. We first describe the main gadget that we repeatedly use in the reduction (see Fig. 1). Consider a number of short intervals placed in a consecutive manner, such that no two of them intersect. Now consider a bunch of long intervals each of which contains all the short intervals. Let the number of short intervals be significantly more than the number of long intervals. In such a scenario, a maximum cut is obtained if all the long intervals belong to one part of the partition, and all the short ones to the other. Now make a copy of all these intervals and place them to the left of the original so that only the long intervals of each
copy intersect. In such a gadget, the left short intervals and the right long intervals must belong to one part of the cut, and the left long intervals and right short intervals to the other, to ensure the maximum number of cut edges.

Observe that in the gadget, there are equal number of intervals in each part of the optimum partition. Now consider some very long intervals, much longer than all the intervals in the gadget. We call these intervals link intervals. Suppose that such a link interval intersects all the intervals in the gadget. No matter which part it belongs to, it will generate an equal number of cut edges by intersecting the gadget. Thus the partition of the gadget remains optimum. Now consider a number of link intervals intersecting all intervals of the gadget. If the number of link intervals is significantly less than the number of intervals in the gadget, then the partition of the gadget remains as described before in any optimal solution. The partition of the gadget remains optimum even when a small number of link intervals intersect only a part of the gadget.

Now we consider a given cubic graph $G$. We construct an interval graph $G'$ from it in the following way. We replace the vertices and edges with gadgets called V-gadgets and E-gadgets, respectively. Both V-gadgets and E-gadgets look like the gadget described above, but the number of intervals in V-gadgets is much more than in E-gadgets. If an edge is incident to a vertex in $G$, then in the interval graph we join the corresponding V-gadget and E-gadget with a pair of link intervals. The link intervals should intersect only the right long intervals of the V-gadget. Each E-gadget is connected to two V-gadgets corresponding to its two end vertices in $G$, by two pairs of link intervals. One of the pairs of link intervals intersect all left intervals of the E-gadget, while the other pair intersects only the left long intervals of the E-gadget (see Fig. 5).

Suppose that two vertices $v_i$ and $v_j$ in $G$ belong to the same part of a maximum cut. Now consider a cut $A' \cup B'$ of the interval graph $G'$ obtained by the reduction. In order to maximize the number of cut edges, we put the left long intervals and right short intervals of both V-gadgets in the same part, say $A'$, and then put the left short and right long intervals of both V-gadgets in $B'$. Then to maximize the cut edges, the link intervals from both V-gadgets must be kept in part $A'$ irrespective of the placement of the corresponding E-gadget intervals in the cut. This is because the V-gadgets contain significantly more intervals than the E-gadget and thus completely determine placement of the link intervals in a maximum cut. Now, there are two general ways to partition the E-gadget, namely to put its left long intervals and right short intervals in $A'$ and right long intervals and left short intervals in $B'$, or the opposite (see Figs. 6(a) and 6(b), the blue intervals are in $A'$ and the red intervals are in $B'$). In the first case,
cut edges are created only between a pair of link intervals and the left short intervals of the E-gadget, while in the second case, cut edges are created between both pair of link intervals and the left long intervals of the E-gadget.

Now suppose that the two vertices \( v_i \) and \( v_j \) in \( G \) belong to two different parts. Then in the interval graph we put the left long intervals and right short intervals of the gadget of \( v_i \), and the right long intervals and left short intervals of the gadget of \( v_j \) in part \( A' \). We put the right long intervals and left short intervals of the gadget of \( v_i \), and the left long intervals and right short intervals of the gadget of \( v_j \) in part \( B' \) (see Figs. 6(c) and 6(d)). To maximize the cut edges, the link intervals from V-gadgets of \( v_i \) and \( v_j \) must be kept in parts \( A' \) and \( B' \) respectively. As before, there are two general ways to partition the E-gadget, namely to put its left long intervals and right short intervals in \( A' \) and right long intervals and left short intervals in \( B' \), or the opposite. In the first case, cut edges are created only between a pair of link intervals and the left long intervals of the E-gadget, while in the second case, cut edges are created between a pair of link intervals and left long intervals of the E-gadget and another pair of link intervals and left short intervals of the E-gadget. Clearly, the number of cut edges in the second case is more than the first, and also more than both the cases where \( v_i \) and \( v_j \) belonged to the same part in the maximum cut of \( G \). This establishes a correspondence between a maximum cut of the original cubic graph and a maximum cut of the constructed interval graph.

3.2 Formal Description of the Reduction

Let \( (G, x) \) be an instance of \textsc{MaxCut} where \( G = (V, E) \) is a cubic graph. Let \( |V| = n \) and hence \( |E| = 3n/2 \). We shall reduce it to an instance \( (G', f(x)) \) of \textsc{MaxCut} where \( G' = (V', E') \) is an interval graph. The construction of \( G' \) is outlined in the following. \( G' = (V', E') \) is described as the intersection graph of a set of intervals on the real line and the vertices of \( G' \) are referred to as intervals.

1. Fix an arbitrary ordering of the vertices and edges of \( G = (V, E) \) as \( v_1, v_2, \ldots, v_n, e_1, e_2, \ldots, e_m \). We shall write any edge \( e \in E \) as an ordered pair of vertices that respects the following convention. If \( e \) is an edge between \( v_i \) and \( v_j \), where \( i < j \), then we shall write \( e = (v_i, v_j) \) (as opposed to \( e = (v_j, v_i) \)).

2. For each vertex \( v \in V \), we construct a V-gadget \( I(v) \) and for each edge \( e \in E \), we construct an E-gadget \( I(e) \). They are shown in Fig. 2. The structure of a V-gadget is identical to that of an E-gadget, the only difference is their size. Each V-gadget (resp. E-gadget) consists of \( q \) (resp. \( q' \)) left long intervals, \( p \) (resp. \( p' \)) left short intervals, \( q \) (resp. \( q' \)) right long intervals and \( p \) (resp. \( p' \)) right short intervals. The left long intervals and the right long intervals of a V-gadget (resp. E-gadget) all intersect each other to form a clique of size \( 2q \) (resp. \( 2q' \)). All left short intervals of a V-gadget (resp. E-gadget) are mutually disjoint and each of them intersect only the \( q \) (resp. \( q' \)) left long intervals. Similarly all right short intervals of a V-gadget (resp. E-gadget) are mutually disjoint and each of them intersect only the \( q \) (resp. \( q' \)) right long intervals. Therefore, the number of edges in each V-gadget (resp. E-gadget) is \( q(2q - 1) + 2pq \) (resp. \( q'(2q' - 1) + 2p'q' \)).
3. We set $q = 200n^3$, $p = 2q + 7n$, $q' = 10n^2$, $p' = 2q' + 7n$, where $n$ is the number of vertices in $G$. Note that the following inequalities hold:

- $p > 2q > 2p' > 4q' > 9n^2$,
- $q^2 > 6n(p - q)$,
- $q^2 > 6n(p' - q')$,
- $q > 3n(p' + q') + 9n^2$.

4. There are a total of $n$ V-gadgets, and $3n/2$ E-gadgets. All $5n/2$ gadgets are arranged in the following order as shown in Fig. 3: $I(v_1), I(v_2), \ldots, I(v_n), I(e_1), I(e_2), \ldots, I(e_{3n/2})$. No two intervals belonging to different gadgets intersect.

5. To establish relationships between the V-gadgets and E-gadgets we introduce 6 link intervals (see Fig. 3). Link intervals connect V-gadgets to E-gadgets. A link interval can intersect a gadget in four different ways as described in the following.

- A link interval is said to cover a gadget if it intersects all intervals of the gadget (see Fig. 4(a)).
- A link interval is said to intersect a V-gadget in the first manner if it intersects only the $q$ right long intervals of the gadget (see Fig. 4(b)).
- A link interval is said to intersect an E-gadget in the second manner if it intersects only the $p'$ left long intervals of the gadget (see Fig. 4(c)).
- A link interval is said to intersect an E-gadget in the third manner if it intersects only the $q'$ left long intervals and the $p'$ left short intervals of the gadget (see Fig. 4(d)).

6. For each edge $e = (v_i, v_j) \in E$, we introduce four link intervals: 1) a pair intersecting $I(v_i)$ in the first manner and $I(e)$ in the second manner, and 2) another pair intersecting $I(v_j)$ in the first manner and $I(e)$ in the third manner (See Fig. 5). Note that since $G$ is cubic, the total number of link intervals covering a V-gadget is $6k$ for some integer $k$ between 0 to $n - 1$. Similarly, the total number of link intervals covering an E-gadget is $4k$ for some integer $k$ between 0 to $3n/2 - 1$. Also, the total number of link intervals intersecting a V-gadget in the first manner is 6.
Fig. 3 Arrangement of the gadgets and the link intervals

(a) A gadget is covered by a link interval.  
(b) A link interval intersects a V-gadget in the first manner.

(c) A link interval intersects an E-gadget in the second manner.  
(d) A link interval intersects an E-gadget in the third manner.

Fig. 4 Illustrations showing the four different ways a link interval can intersect a gadget

Fig. 5 Link intervals connecting an E-gadget $I((v_i, v_j))$ with V-gadgets $I(v_i)$ and $I(v_j)$
3.3 Properties of the Reduction

In this section, we study the properties of the interval graph $G'$ constructed from $G$ in the previous section. Let us first state a general fact about false twins and maximum cut partitions of a graph. Two non-adjacent vertices of a graph are called false twins if they have the same neighbourhood.

**Lemma 1** Every graph has a maximum cut in which any pair of false twins belong to the same part of the partition.

**Proof** Let $A \cup B$ be a maximum cut of a graph. If every pair of false twins in the graph belong to the same subset, then we are done. So assume that there are false twins $u$ and $v$ that are in different subsets, say $u \in A$ and $v \in B$. Since the partition yields a maximum cut, we must have $|N(u) \cap B| \geq |N(u) \cap A|$ and $|N(v) \cap A| \geq |N(v) \cap B|$. But since $N(u) = N(v)$, we can substitute $N(u)$ with $N(v)$ in the first inequality and $N(v)$ with $N(u)$ in the second inequality to obtain respectively $|N(v) \cap B| \geq |N(v) \cap A|$ and $|N(u) \cap A| \geq |N(u) \cap B|$. Form these four inequalities we can conclude that $|N(u) \cap B| = |N(v) \cap A|$ and $|N(v) \cap A| = |N(v) \cap B|$. Therefore, moving $u$ from $A$ to $B$ (or moving $v$ from $B$ to $A$) does not change the cut size. Thus we can obtain a maximum cut in which every pair of false twins belong to the same part of the partition. \qed

Let us now introduce a definition. For any V or E-gadget of $G'$, we shall say that the gadget is well partitioned by a cut if all left long and right short intervals of the gadget are in one subset, while all the right long and left short intervals are in the other.

**Lemma 2** There is a maximum cut of $G'$ in which all V-gadgets and E-gadgets are well partitioned.

**Proof** By Lemma 1, there is a maximum cut of $G'$ in which any pair of false twins belong to the same part of the partition. Consider such a maximum cut $A \cup B$. Let us first prove that all V-gadgets are well partitioned by this cut. So take any V-gadget $I(v_i)$. All its left short intervals are in the same subset as they are all false twins. Without loss of generality, assume that they are in $A$. Suppose that a left long interval of $I(v_i)$ is also in $A$. Then moving it to $B$ results in losing at most $2q - 1$ cut edges due to its intersections with other long intervals of $I(v_i)$, and at most $6n$ cut edges due to its intersections with the link intervals of $I(v_i)$. However, we gain $p$ cut edges. Since $p = 2q + 7n > 2q - 1 + 6n$, the size of the cut increases. This contradicts the fact that the partition yields a maximum cut. Hence, all left long intervals of $I(v_i)$ must be in $B$.

Since all right short intervals of $I(v_i)$ also false twins, they all belong to any one subset. It follows from the above arguments that all right long intervals of $I(v_i)$ must be in the other subset. Therefore, there are two possible cases: either (a) all left short and right long intervals of $I(v_i)$ are in $A$, all right short and left long intervals of $I(v_i)$ are in $B$, or (b) all the short intervals of $I(v_i)$ are in $A$ and all the long intervals of $I(v_i)$ are in $B$.

If case (a) holds then we are done. So we show that case (b) is impossible. For the sake of contradiction, assume that (b) holds. Now let us move all the right short intervals...
of $I(v_i)$ to $B$ and all right long intervals of $I(v_i)$ to $A$. Due to their intersections with link intervals, this removes at most $6n(p - q)$ edges from the cut. But due to the intersections among the left and right long intervals, it also adds at least $q^2$ edges to the cut. By our choice of $q$ and $p$, we have $q^2 - 6n(p - q) > 0$. Hence the total number of edges in the cut increases. This contradicts the fact that the partition yields a maximum cut and hence this case is impossible.

We have shown that all V-gadgets are well partitioned by the maximum cut $A \cup B$. Since V-gadgets and E-gadgets are structurally similar, it is easy to see that the arguments also hold for E-gadgets. Hence, all V and E-gadgets are well partitioned by the maximum cut $A \cup B$. □

**Lemma 3** If all V-gadgets and E-gadgets of $G'$ are well partitioned by a cut, then

- the number of cut edges obtained from the intersections between the intervals of all V-gadgets is $n(2pq + q^2)$,
- the number of cut edges obtained from the intersections between the intervals of all E-gadgets is $3n(2p'q' + q'^2)/2$,
- the number of cut edges obtained from the intersections between the V-gadgets and the link intervals that cover them is $3n(n - 1)(p + q)$,
- the number of cut edges obtained from the intersections between the E-gadgets and the link intervals covering them is

$$3n\left(\frac{3n}{2} - 1\right)(p' + q').$$

(1)

**Proof** It is easy to see that each well-partitioned V-gadget gives $2pq + q^2$ cut edges. Hence the total number of cut edges obtained from all V-gadgets is $(2pq + q^2)n$. Also each well-partitioned E-gadget gives $2p'q' + q'^2$ cut edges. Since $G$ is cubic, the number of E-gadgets in $G'$ is $3n/2$. Hence the total number of cut edges obtained from all E-gadgets is $3n(2p'q' + q'^2)/2$. Now we calculate the number of cut edges obtained from the intersections between the V-gadgets and the link intervals that cover them. Notice that the intersection between a well-partitioned V-gadget and a link interval covering it gives $p + q$ cut edges. Now observe that for each $i$, $1 \leq i \leq n$, the V-gadget $I(v_i)$ is covered by $6(i - 1)$ link intervals. Hence the total number of cut edges obtained in this manner is $3n(n - 1)(p + q)$. Similarly, we get $p' + q'$ cut edges each time a link interval covers an E-gadget. Observe that for each $j$, $1 \leq j \leq 3n/2$, the E-gadget $I(v_j)$ is covered by $4(3n/2 - j)$ link intervals. So the total number of cut edges obtained from these intersections is (1). □

**Lemma 4** $G$ has a cut of size at least $x$ if and only if $G'$ has a cut of size at least

$$2pq + q^2)n + \frac{3n}{2}(2p'q' + q'^2) + 3n(n - 1)(p + q) + 3n\left(\frac{3n}{2} - 1\right)(p' + q') + 6nq + 3np' + 2xq'.$$

(2)
**Proof** First suppose that $G$ has a cut of size at least $x$. Denote the subsets in the partition of the vertices of $G$ by $A$ and $B$. We partition the vertices of $G'$ as follows. If a vertex $v_i$ of $G$ is in $A$, then in the corresponding V-gadget $\mathcal{I}(v_i)$ of $G'$, all left short intervals and right long intervals are placed in $A'$, all right short intervals and left long intervals are placed in $B'$. Finally, all link intervals intersecting $\mathcal{I}(v_i)$ in the first manner are placed in $B'$. If $v_i$ is in $B$ instead, then all the above placements of intervals are swapped. Recall that for each E-gadget exactly two link intervals intersect it in the second manner and exactly two link intervals intersect it in the third manner. If the link intervals that intersect an E-gadget in the third manner is in $A'$, then we place the left short intervals and right long intervals of the E-gadget in $B'$, and the left long intervals and right short intervals in $A'$. If the link intervals are in $B'$, then the placements of the intervals are swapped.

All the V-gadgets and E-gadgets of $G'$ are well partitioned by the cut. Hence by Lemma 3, the internal cut edges of V-gadgets and E-gadgets, and the cut edges formed between gadgets and the link intervals that cover them are in total

$$(2pq + q^2)n + \frac{3n}{2}(2p'q' + q'^2) + 3n(n - 1)(p + q) + 3n\left(\frac{3n}{2} - 1\right)(p' + q'). \tag{3}$$

For each V-gadget, the link intervals intersecting it in the first manner give $6q$ cut edges, resulting in a total of $6nq$ cut edges. Each link interval that intersects an E-gadget in the third manner gives $p'q'$ cut edges, thus we have $3np'$ in total. However, a link interval that intersects an E-gadget in the second manner can produce cut edges from the E-gadget only when the other link interval mentioned above is in a different subset, i.e., the vertices of $G$ corresponding to the V-gadgets of these link intervals are in $A$ and $B$, and produce a cut edge. This means that such link intervals produce at least $2xq'$ cut edges in total, proving the forward direction of the claim.

Now we prove the backward direction of the claim. Assume that $G'$ has a cut of size (2) at least. So the size of a maximum cut of $G'$ is at least this much. By Lemma 2, there exists a maximum cut of $G'$ in which all $V$ and $E$-gadgets are well partitioned. Consider such a maximum cut $A' \cup B'$. Corresponding to this cut of $G'$, we define a cut $A \cup B$ of $G$ in the following way. If the left long and right short intervals of $\mathcal{I}(v_i)$ are in $A'$ (resp. $B'$), then we put $v_i$ in $A$ (resp. $B$). If $y$ is the size of the cut $A \cup B$, then we have to show that $y \geq x$.

By Lemma 3, the internal cut edges of V-gadgets and E-gadgets, and the cut edges formed between gadgets and the link intervals that cover them amount to (3) cut edges in total. Hence, the remaining $6nq + 3np' + 2xq'$ cut edges are obtained from the partial intersections of the link intervals with the V-gadgets and E-gadgets, and the intersections among link intervals. The number of cut edges due to the intersections among the link intervals is at most $(3n)^2 = 9n^2$. Also the partial intersections between link intervals and E-gadgets cannot give more than $3n(p' + q')$ cut edges. The partial intersections between link intervals and V-gadgets can contribute at most $6nq$ cut edges. This happens when for each V-gadget, the link intervals intersecting it in the first manner are all in the subset which contains the left long and right short intervals of the gadget. If this is not the case for at least one link interval, then we lose at least $q$ cut edges. Since $q > 3n(p' + q') + 9n^2$, this loss cannot be compensated by cut edges due
to the partial intersections between link intervals and E-gadgets and the intersections among the link intervals, as they cannot give more than $5n(p' + q') + 9n^2$ cut edges. Hence it implies that exactly $6nq$ of the remaining cut edges are obtained from link intervals intersecting V-gadgets in the first manner. Thus for each V-gadget, the link intervals intersecting it in the first manner are all in the subset which contains the left long and right short intervals of the gadget. Hence, the placement of the intervals of the V-gadget in the subsets $A'$ and $B'$ (and hence the placement of the corresponding vertex of $G$ in $A$ or $B$) determines the placements of the link intervals.

The remaining $3np' + 2xq'$ cut edges should come from the partial intersections of the link intervals with the E-gadgets, and the intersections among link intervals. We show that this is not possible if $y < x$ where $y$ is the size of the cut $A \cup B$. For this, consider an E-gadget $\mathcal{I}(v_i, v_j)$. Let $\ell_i, \ell_i', \ell_j, \ell_j'$ be the two link intervals from $\mathcal{I}(v_i)$ that intersect $\mathcal{I}(v_i, v_j)$ in the second manner and $\ell_i, \ell_j'$ be the two link intervals from $\mathcal{I}(v_j)$ that intersect $\mathcal{I}(v_i, v_j)$ in the third manner. Consider the following cases: $\ell_i, \ell_i', \ell_j, \ell_j'$ are in the same subset, say $\ell_i, \ell_i', \ell_j, \ell_j' \in A'$ (case 1) and $\ell_i, \ell_i'$ are in one subset and $\ell_j, \ell_j'$ are in the other, say, $\ell_i, \ell_i' \in A', \ell_j, \ell_j' \in B'$ (case 2). In case 2, the edge $(v_i, v_j)$ appears in the cut set of $A \cup B$, while in case 1, it does not. For each case, we have two subcases as described in the following (see also Fig. 6).

Case 1a: $A'$ contains $\ell_i, \ell_i', \ell_j, \ell_j'$ and the left long and right short intervals of $\mathcal{I}(v_i, v_j)$. $B'$ contains the right long and left short intervals of $\mathcal{I}(v_i, v_j)$. Hence, the intersections between $\mathcal{I}(v_i, v_j)$ and $\ell_i, \ell_i', \ell_j, \ell_j'$ give $2p'$ cut edges.

Case 1b: $A'$ contains $\ell_i, \ell_i', \ell_j, \ell_j'$ and the left long and short intervals of $\mathcal{I}(v_i, v_j)$. $B'$ contains the left long and right short intervals of $\mathcal{I}(v_i, v_j)$. Hence, the intersections between $\mathcal{I}(v_i, v_j)$ and $\ell_i, \ell_i', \ell_j, \ell_j'$ give $4q'$ cut edges.

Case 2a: $A'$ contains $\ell_i, \ell_i'$ and the left long and right short intervals of $\mathcal{I}(v_i, v_j)$. $B'$ contains $\ell_j, \ell_j'$ and the right long and short intervals of $\mathcal{I}(v_i, v_j)$. Hence, the intersections between $\mathcal{I}(v_i, v_j)$ and $\ell_i, \ell_i', \ell_j, \ell_j'$ give $2q'$ cut edges.

Case 2b: $A'$ contains $\ell_i, \ell_i'$ and the left long and right short intervals of $\mathcal{I}(v_i, v_j)$. $B'$ contains $\ell_j, \ell_j'$ and the left long and right short intervals of $\mathcal{I}(v_i, v_j)$. Hence, the intersections between $\mathcal{I}(v_i, v_j)$ and $\ell_i, \ell_i', \ell_j, \ell_j'$ give $2p' + 2q'$ cut edges.

Therefore, we see that an E-gadget gives at most $2p'$ cut edges from its partial intersections with link intervals if the link intervals belong to the same subset (since $2p' > 4q'$), and at most $2(p' + q')$ cut edges if the link intervals belong to different subsets (since $2p' + 2q' > 2q'$). Notice that the later case occurs for exactly $y$ E-gadgets. The number of cut edges obtained from the partial intersections of E-gadgets with link intervals is at most $2p'(3n/2 - y) + 2y(p' + q') = 3np' + 2yq'$. Hence if $y < x$, then at least $2q'(x - y) > 2q'$ cut edges must come from the intersections among the link intervals. But this is not possible as $2q' > 9n^2$. Hence, $y \geq x$ as required.

\begin{flushright}
\textbf{Theorem 1} \ (\textbf{MAXCut is NP-complete on interval graphs}.)
\end{flushright}
Proof It can be checked in polynomial time if a given partition of an interval graph produces a cut of a given size. Thus the problem is in NP. The construction of $G'$ from $G$ clearly takes polynomial time. The NP-hardness follows from Lemma 4.

4 Concluding Remarks

In this paper, we have settled the longstanding open question concerning the computational complexity of MAXCUT on interval graphs. However, the status of MAXCUT on unit interval graphs still remains open. Interval graphs can be characterized in terms
of interval count which is the smallest number of interval lengths used by an interval representation of the graph. Therefore, unit interval graphs are the graphs with interval count 1. Notice that the interval count of our reduction is unbounded. If MAXCUT on unit interval graphs is also NP-complete, then an easier problem would be to first show hardness for interval graphs with bounded interval count, which would sharpen our result. This has been very recently shown by de Figueiredo et al. [19]. Their reduction is also from MAXCUT on cubic graphs. Recall that in our reduction, the V-gadgets and E-gadgets are linearly arranged with each one disjoint from the others. Link intervals of different lengths are then used to establish the relations between the V-gadgets and E-gadgets and thus giving rise to unbounded interval count. To circumvent this, the V-gadgets in the reduction of [19] are all bunched together with each one a bit perturbed to the left of the previous one. Each V-gadget of [19] consists of several disjoint copies of V-gadgets of our reduction that are connected by link intervals of equal length. The E-gadgets in [19] are the same as ours. The E-gadgets are placed inside the bunch of V-gadgets and thus intersects all the V-gadgets. Another set of intervals, each of the same length, is introduced to establish the relation between each E-gadget with its two corresponding V-gadgets. This allows them to bound the interval count to 4.

On the algorithmic side, two flawed polynomial time algorithms for unit interval graphs were previously proposed, one by Bodlaender et al. in [10] and another one by Boyacı et al. in [11]. The result of the first paper was later disproved by the authors themselves in [8]. In the second paper, the authors used the so-called bubble model of unit interval graphs to design the algorithm. The bubble model, introduced by Heggernes et al. in [28], is a matrix-like representation of unit interval graphs. The bubble model representation is particularly useful in designing efficient algorithms for unit interval graphs. The algorithm of [11] was however disproved by Kratochvíl et al. in [37]. However, the authors in [37] were able use the bubble model representation to give a subexponential algorithm for MAXCUT on unit interval graphs. In fact, the result holds for even of the more general class of mixed unit interval graphs where unit intervals of more than one type (open, closed, semi-closed) are allowed. In particular, they introduced the \(U\)-bubble model which is an extension of the bubble model to the class of mixed unit interval graphs and used it to design the subexponential time algorithm. Furthermore, the problem can be solved in polynomial time if the \(U\)-bubble model representation has a constant number of columns.

We proved the NP-completeness of MAXCUT on interval graphs via a reduction from MAXCUT on cubic graphs. A reduction from MAXCUT on general graphs was given in an earlier version [1] of this paper. Both reductions are similar and use the same gadgets, but the correctness proof of the reduction of the earlier version was cumbersome. It is known that it is NP-hard to approximate MAXCUT with an approximation ratio better than \(131/132 \approx 0.997\) for cubic graphs [6] and \(16/17 \approx 0.941\) in general [27, 45]. Our reductions however do not seem to be approximation preserving and hence do not imply any approximation hardness for MAXCUT on interval graphs. In general, the polynomial-time approximation algorithm for MAXCUT with the best known approximation ratio is by Goemans and Williamson [23] which achieves approximation ratio \(\approx 0.878\). Assuming the Unique Games conjecture [35], this is the best possible approximation ratio. An interesting question is whether this can
be improved for interval graphs or unit interval graphs. Recently in [46] a near-linear time 0.66 approximation algorithm for unit interval graphs has been proposed.

**Acknowledgements** The first and third authors are supported by Council of Scientific and Industrial Research (CSIR), Govt. of India. The second author is supported by National Board for Higher Mathematics (NBHM), Department of Atomic Energy (DAE), Govt. of India. The fourth author is supported by grants from Sponsored Research and Industrial Consultancy (SIRC), IIT Kharagpur under the Institute Scheme for Innovative Research and Development (ISIRD), and Science and Engineering Research Board (SERB) under the Mathematical Research Impact-Centric Support (MATRICS) scheme. We would like to thank the anonymous reviewers for their suggestions which helped us to improve the clarity and presentation of the paper. We especially thank one reviewer for suggesting Lemma 1 which helped us to simplify the proof of Lemma 2.

**Data Availability** Data sharing not applicable to this article as no datasets were generated or analysed during the current study.

**References**

1. Adhikary, R., Bose, K., Mukherjee, S., Roy, B.: Complexity of maximum cut on interval graphs (2020). arXiv:2006.00061
2. Adhikary, R., Bose, K., Mukherjee, S., Roy, B.: Complexity of maximum cut on interval graphs. In: 37th International Symposium on Computational Geometry. Leibniz International Proceedings in Informatics, vol. 189, # 7. Leibniz-Zentrum für Informatik, Wadern (2021)
3. Asdre, K., Ioannidou, K., Nikolopoulos, S.D.: The harmonious coloring problem is NP-complete for interval and permutation graphs. Discrete Appl. Math. 155(17), 2377–2382 (2007)
4. Barahona, F.: The max-cut problem on graphs not contractible to $K_5$. Oper. Res. Lett. 2(3), 107–111 (1983)
5. Barahona, F., Grötschel, M., Jünger, M., Reinelt, G.: An application of combinatorial optimization to statistical physics and circuit layout design. Oper. Res. 36(3), 493–513 (1988)
6. Berman, P., Karpinski, M.: On some tighter inapproximability results (extended abstract). In: 26th International Colloquium on Automata, Languages and Programming (Prague 1999). Lecture Notes in Computer Science, vol. 1644, pp. 200–209. Springer, Berlin (1999)
7. Bodlaender, H.L.: Achromatic number is NP-complete for cographs and interval graphs. Inform. Process. Lett. 31(3), 135–138 (1989)
8. Bodlaender, H.L., de Figueiredo, C.M.H., Gutierrez, M., Kloks, T., Niedermeier, R.: Simple max-cut for split-indifference graphs and graphs with few $P_4$’s. In: 3rd International Workshop on Experimental and Efficient Algorithms (Angra dos Reis 2004). Lecture Notes in Computer Science, vol. 3059, pp. 87–99. Springer, Berlin (2004)
9. Bodlaender, H.L., Jansen, K.: On the complexity of the maximum cut problem. Nordic J. Comput. 7(1), 14–31 (2000)
10. Bodlaender, H.L., Kloks, T., Niedermeier, R.: Simple MAX-CUT for unit interval graphs and graphs with few $P_4$’s. In: 6th Twente Workshop on Graphs and Combinatorial Optimization (Enschede 1999). Electronic Notes in Discrete Mathematics, vol. 3, pp. 19–26. Elsevier, Amsterdam (1999)
11. Boyaci, A., Ekim, T., Shalom, M.: A polynomial-time algorithm for the maximum cardinality cut problem in proper interval graphs. Inform. Process. Lett. 121, 29–33 (2017)
12. Chakraborty, D., Das, S., Foucaud, F., Gahlawat, H., Lajou, D., Roy, B.: Algorithms and complexity for geodetic sets on planar and chordal graphs. In: 31st International Symposium on Algorithms and Computation. Leibniz International Proceedings in Informatics, vol. 181, # 7. Leibniz-Zentrum für Informatik, Wadern (2020)
13. Chang, K.C., Du, D.H.-Ch.: Efficient algorithms for layer assignment problem. IEEE Trans. Comput. Aided Des. Integr. Circuits Syst. 6(1), 67–78 (1987)
14. Chang, M.-Sh.: Efficient algorithms for the domination problems on interval and circular-arc graphs. SIAM J. Comput. 27(6), 1671–1694 (1998)
15. Cohen, J.E., Stephens, D.W.: Food Webs and Niche Space. Monographs in Population Biology, vol. 11. Princeton University Press, Princeton (1978)
16. Cohen, J., Fomin, F., Heggernes, P., Kratsch, D., Kucherov, G.: Optimal linear arrangement of interval graphs. In: Mathematical Foundations of Computer Science (Staré Lázně 2006). Lecture Notes in Computer Science, vol. 4162, pp. 267–279. Springer, Berlin (2006)
17. Crowston, R., Jones, M., Mnich, M.: Max-cut parameterized above the Edwards–Erdős bound. Algorithmica 72(3), 734–757 (2015)
18. Diaz, J., Kamiński, M.: MAX-CUT and MAX-BISECTION are NP-hard on unit disk graphs. Theoret. Comput. Sci. 377(1–3), 271–276 (2007)
19. de Figueiredo, C.M., de Melo, A.A., Oliveira, F.S., Silva, A.: Maximum cut on interval graphs of interval count four is NP-complete. In: 46th International Symposium on Mathematical Foundations of Computer Science (Tallinn 2021). Leibniz International Proceedings in Informatics, vol. 202, # 38. Leibniz-Zentrum für Informatik, Wadern (2021)
20. Fomin, F.V., Golovach, P.A., Lokshtanov, D., Saurabh, S.: Almost optimal lower bounds for problems parameterized by clique-width. SIAM J. Comput. 43(5), 1541–1563 (2014)
21. Foucaud, F., Mertzios, G., Naserasr, R., Parreau, A., Valicov, P.: Identification, location-domination and metric dimension on interval and permutation graphs II. Algorithms and complexity. Algorithmica 78(3), 914–944 (2017)
22. Garey, M.R., Johnson, D.S.: Computers and Intractability: A Guide to the Theory of NP-completeness. W.H. Freeman, New York (1990)
23. Goemans, M., Williamson, D.: Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. J. Assoc. Comput. Mach. 42(6), 1115–1145 (1995)
24. Golumbic, M.Ch.: Algorithmic Graph Theory and Perfect Graphs. Annals of Discrete Mathematics, vol. 57. Elsevier, Amsterdam (2004)
25. Gururwami, V.: Maximum cut on line and total graphs. Discrete Appl. Math. 92(2–3), 217–221 (1999)
26. Hadlock, F.: Finding a maximum cut of a planar graph in polynomial time. SIAM J. Comput. 4(3), 221–225 (1975)
27. Håstad, J.: Some optimal inapproximability results. J. Assoc. Comput. Mach. 48(4), 798–859 (2001)
28. Heggernes, P., Meister, D., Papadopoulos, Ch.: A new representation of proper interval graphs with an application to clique-width. In: DIMAP Workshop on Algorithmic Graph Theory (Coventry 2009). Electronic Notes in Discrete Mathematics, vol. 32, pp. 27–34. Elsevier, Amsterdam (2009)
29. Imai, H., Asano, T.: Finding the connected components and a maximum clique of an intersection graph of rectangles in the plane. J. Algorithms 4(4), 310–323 (1983)
30. Johnson, D.S.: The NP-completeness column: an ongoing guide. J. Algorithms 6(3), 434–451 (1985)
31. Jungck, J.R., Dick, G., Gleason Dick, A.: Computer-assisted sequencing, interval graphs, and molecular evolution. Biosystems 15(3), 259–273 (1982)
32. Kamiński, M.: MAX-CUT and containment relations in graphs. Theoret. Comput. Sci. 438, 89–95 (2012)
33. Karp, R.M.: Reducibility among combinatorial problems. In: Complexity of Computer Computations (Yorktown Heights 1972), pp. 85–103. Plenum, New York (1972)
34. Keil, J.M.: Finding Hamiltonian circuits in interval graphs. Inform. Process. Lett. 20(4), 201–206 (1985)
35. Khot, S.: On the power of unique 2-prover 1-round games. In: 34th Annual ACM Symposium on Theory of Computing (Montréal 2002), pp. 767–775. ACM, New York (2002)
36. Kobayashi, Y., Kobayashi, Y., Miyazaki, Sh., Tamaki, S.: An improved fixed-parameter algorithm for max-cut parameterized by crossing number. In: 30th International Workshop on Combinatorial Algorithms (Pisa 2019). Lecture Notes in Computer Science, vol. 11638, pp. 327–338. Springer, Cham (2019)
37. Kratochvíl, J., Masařík, T., Novotná, J.: U-bubble model for mixed unit interval graphs and its applications: the MaxCut problem revisited. In: 45th International Symposium on Mathematical Foundations of Computer Science (Prague 2020). Leibniz International Proceedings in Informatics, vol. 170, # 57. Leibniz-Zentrum für Informatik, Wadern (2020)
38. Lu, Ch.L., Tang, Ch.Y.: A linear-time algorithm for the weighted feedback vertex problem on interval graphs. Inform. Process. Lett. 61(2), 107–111 (1997)
39. Madathil, J., Saurabh, S., Zehavi, M.: Fixed-parameter tractable algorithm and polynomial kernel for max-cut above spanning tree. Theory Comput. Syst. 64(1), 62–100 (2020)
40. Mahajan, M., Raman, V.: Parameterizing above guaranteed values: MaxSat and MaxCut. J. Algorithms 31(2), 335–354 (1999)
41. Marathe, M.V., Ravi, R., Pandu Rangan, C.: Generalized vertex covering in interval graphs. Discrete Appl. Math. 39(1), 87–93 (1992)
42. Marx, D.: A short proof of the NP-completeness of minimum sum interval coloring. Oper. Res. Lett. 33(4), 382–384 (2005)
43. Papadimitriou, Ch.H., Yannakakis, M.: Optimization, approximation, and complexity classes. J. Comput. System Sci. 43(3), 425–440 (1991)
44. Raman, V., Saurabh, S.: Improved fixed parameter tractable algorithms for two “edge” problems: MAXCUT and MAXDAG. Inform. Process. Lett. 104(2), 65–72 (2007)
45. Trevisan, L., Sorkin, G.B., Sudan, M., Williamson, D.P.: Gadgets, approximation, and linear programming. SIAM J. Comput. 29(6), 2074–2097 (2000)
46. Utkarsh, J., Rahul, S., Thoppil, J.J.: Algorithms for max cut on unit interval and laminar interval graphs. In: Computational Geometry: Young Researchers Forum 2021, pp. 37–39. https://cse.buffalo.edu/socg21/files/YRF-Booklet.pdf
47. Zhang, P., Schon, E.A., Fischer, S.G., Cayanis, E., Weiss, J., Kistler, S., Bourne, Ph.E.: An algorithm based on graph theory for the assembly of contigs in physical mapping of DNA. Bioinformatics 10(3), 309–317 (1994)

Publisher’s Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.