# Connected Vertex Cover on AT-Free Graphs

**Joydeep Mukherjee** ✉
Ramakrishna Mission Vivekananda Educational and Research Institute, Belur, India

**Tamojit Saha** ✉
Ramakrishna Mission Vivekananda Educational and Research Institute, Belur, India
Institute of Advancing Intelligence, TCG CREST, Kolkata, India

## Abstract

Asteroidal Triple (AT) in a graph is an independent set of three vertices such that every pair of them has a path between them avoiding the neighbourhood of the third. A graph is called AT-free if it does not contain any asteroidal triple. A connected vertex cover of a graph is a subset of its vertices which contains at least one endpoint of each edge and induces a connected subgraph. Settling the complexity of computing a minimum connected vertex cover in an AT-free graph was mentioned as an open problem in Escoffier et al. [6]. In this paper we answer the question by presenting an exact polynomial time algorithm for computing a minimum connected vertex cover problem on AT-free graphs.

**2012 ACM Subject Classification** Theory of computation → Graph algorithms analysis

**Keywords and phrases** Graph Algorithm, AT-free graphs, Connected Vertex Cover, Optimization

**Digital Object Identifier** 10.4230/LIPIcs.ISAAC.2023.54

## 1 Introduction

An *Asteroidal Triple (AT)* of a graph $G = (V, E)$ is a set of three vertices of $V(G)$ such that these three vertices are mutually nonadjacent and for any two vertices of this set there exists a path between these two vertices which avoids the neighborhood of the third vertex. A graph is called *asteroidal triple free (AT-free)* if it does not contain any asteroidal triple.

We assume, in the rest of the paper, the graph $G$ is undirected, unweighted and simple graph. A subset of $V(G)$ is a vertex cover of $G$ if every edge of $G$ has an endpoint in that subset. The minimum vertex cover problem is to find a vertex cover of minimum cardinality. A vertex cover which also induces a connected subgraph of $G$ is called a connected vertex cover. The minimum connected vertex cover problem is to find a vertex cover of minimum cardinality such that the vertices of the vertex cover induces a connected subgraph. In the rest of the paper we denote the minimum vertex cover problem by MVC and the minimum connected vertex cover problem by MCVC.

In this paper we present a polynomial time algorithm for MCVC on connected AT-free graphs. More precisely we provide an $O(n^4)$ algorithm for minimum connected vertex cover in AT-free graphs. In [3] Broersma et al. presented a polynomial time algorithm for maximum independent set problem. Our work is inspired by the technique developed in that paper. In the following we define the problem more formally.

> ***Connected Vertex cover On AT-free graphs***
> ***Instance:*** A connected AT-free graph $G = (V, E)$, $|V| = n$, $|E| = m$.
> ***Output:*** A set $S^* \subseteq V(G)$ of minimum cardinality such that $G[S^*]$ is connected and $S^*$ contains at least one end point of every edge in $G$, i.e. $S^*$ is a vertex cover.

The MCVC problem is studied in several graph classes, and there exist various algorithms for this problem in the fields of approximation algorithm, fixed parameter algorithm, and polynomial time exact algorithm. In the following we discuss some of the known results for

© Joydeep Mukherjee and Tamojit Saha;
licensed under Creative Commons License CC-BY 4.0
34th International Symposium on Algorithms and Computation (ISAAC 2023).
Editors: Satoru Iwata and Naonori Kakimura; Article No. 54; pp. 54:1–54:12
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

this problem. This problem was first introduced by Garey and Johnson [8]. This problem is known to be NP-hard in planar bipartite graphs of maximum degree 4 [7], in planar bi-connected graphs of maximum degree 4 [17], in $H-$free graphs if $H$ contains a cycle or a claw [16], and in 3-connected graphs [20]. It is APX-complete in bipartite graphs of maximum degree 4, even if each vertex of one partite set has a degree at most 3 [6].

MCVC is polynomial time solvable in many special graph classes like graphs of maximum degree 3 [19], $(sP_1 + P_5)-$free graphs [11]. Escoffier et al. [6] proved several results regarding the connected vertex cover problem in special graph classes. They showed this problem is polynomial time solvable in chordal graphs; in bipartite graphs if each vertex of one partite set has maximum degree 2 and the vertices of the other partite set have no restriction on the degree. They proved a PTAS for MCVC in planar graphs. In the same paper, they provided a $\frac{5}{3}-$approximation algorithm for MCVC on all those graphs for which MVC is solvable in polynomial time. On the complexity side they proved that MCVC is APX-hard in bipartite graphs. Note that results of this paper along with the polynomial time algorithm for independent set problem presented by Broersma et al. in [3], impliy a $\frac{5}{3}-$approximation algorithm for AT-free graphs. Escoffier et al. in the paper [6], posed the complexity of MCVC on AT-free graphs as an open problem.

We state known approximation algorithm and FPT algorithm results for MCVC. A 2-approximation algorithm for MCVC is known in general graphs [1, 18] but it is not possible to approximate MCVC within ratio $(10\sqrt{5} - 21)$ in general graphs unless $P = NP$ [7]. Several results for computing connected vertex cover are known in the field of fixed parameter algorithms. First result was an algorithm with running time $O(6^k)$ [10] which was later improved to $O(2.7060^k)$ [14], where $k$ is the length of a minimum vertex cover in the given graph and also an algorithm with running time $O(2^t \cdot t^{(3t+2)} n)$ where $t$ is the treewidth and $n$ is the number of vertices in the given graph [15].

Asteroidal triple free graph class contains graph classes like permutation graphs, interval graphs, trapezoid graphs, and cocomparability graphs [5]. AT-free graphs have many desirable properties which make them amenable for designing polynomial time algorithms for many problems which are NP-complete in general graphs. Such problems include minimum feedback vertex set problem [13], maximum independent set [3], dominating set, total dominating set [12] and connected dominating set [2], induced disjoint path problem [9]. However, to the best of our knowledge, the complexity of computing connected vertex cover problem is unknown in AT-free graphs.

## 2 Preliminaries

Let $G = (V, E)$ be a simple unweighted graph. We denote the set of vertices by $V(G)$ and the set of edges by $E(G)$. A graph $H = (V', E')$ is a subgraph of $G = (V, E)$ if $V' \subseteq V$ and $E' \subseteq E$. We denote $|V|$ by $n$ and $|E|$ by $m$. A subgraph $H = (V', E')$ of $G$ is an induced subgraph if $V' \subseteq V$ and for $u, v \in V'$, $(u, v) \in E'$ if and only if $(u, v) \in E$. The induced subgraph on any subset $S \subseteq V$ is denoted by $G[S]$.

The neighbourhood of a vertex $v$, denoted by $N(v)$, is the set of all vertices that are adjacent to $v$. Closed neighbourhood of $v$ is denoted by $N[v] = \{v\} \cup N(v)$. The neighbourhood of a set of vertices $\{v_1, v_2, \ldots, v_k\}$ is denoted by $N(v_1, v_2, \ldots, v_k) = \bigcup_{i=1}^{k} N(v_i)$ and the closed neighbourhood is denoted by $N[v_1, v_2, \ldots, v_k] = \bigcup_{i=1}^{k} N[v_i]$. Assume $C$ is a connected component of $G$. The set $N_C(v)$ where $v \in V(C)$, denotes the set of neighbour of $v$ that are in the component $C$.

A path is a graph, $Y = (V, E)$, such that $V = \{y_1, y_2, \ldots, y_k\}$ and $E = \{y_1y_2, y_2y_3, \ldots, y_{k-1}y_k\}$. We denote a path by the sequence of its vertices, that is $Y = y_1y_2 \ldots y_k$. Here $y_1$ and $y_k$ are called endpoints of path $Y$. The number of vertices present in $Y$ is denoted by $|Y|$. We denote $y_iYy_j = y_iy_{i+1}\dots y_j$ where $1 \le i \le j \le k$. A path on $k$ vertices is denoted by $Y_k$ and the length of the path is denoted by the number of edges present on the path that is $k-1$. The distance between two vertices in a graph is the length of the shortest path between them. A cycle is a graph, $C = (V, E)$, such that $V(C) = \{c_1, c_2, \dots, c_l\}$ and $E(C) = \{c_1c_2, \dots, c_{l-1}c_l, c_lc_1\}$. The shortest distance between $u$ and $v$ is denoted by $dist_C(u, v)$ where $u, v \in V(C)$. The number of vertices present in the cycle $C$ is denoted by $|C|$.

A *dominating set* $D$ of $G$ is a subset of vertices of $G$ such that for every $v$ outside $D$, $N(v) \cap D \ne \phi$. A *dominating pair* is a pair of vertices such that any path between them is a dominating set. There is a linear time algorithm to find a dominating pair [4] in AT-free graphs. We denote a shortest path between a dominating pair by $DSP$.

This paper is inspired by the technique developed by Broersma et al. in [3]. We use their method of graph decomposition and supplement it with our new observations for connected vertex cover in AT-free graphs to derive the results stated in this paper. Let $G(V, E)$ denote a connected AT-free graph. The components in the graph $G \setminus N[x]$, where $x \in V$, are denoted by $C_1^x, \dots, C_r^x$.

Let $x$ and $y$ be two nonadjacent vertices of the graph. We define an *interval* to be $I(x, y) \subseteq V(G)$ in the following :

$I(x, y) = \{s \in V(G)$ : there is a $s, x$-path which does not contain any neighbours of $y$ and there is a $s, y$-path which does not contain any neighbour of $x\}$.

Assume a connected component $C$ containing vertices $x$ and $y$. We denote the interval $I(x, y)$ by $I_C(x, y)$ when we consider the induced subgraph on $C$ instead of the whole graph $G$.

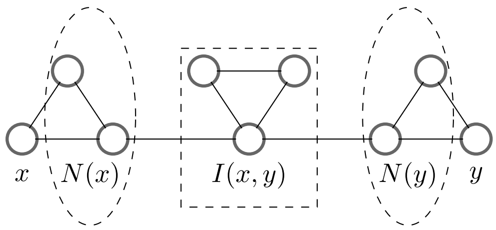


**Figure 1** An example of interval in an AT-free graph.

Let $y \in V$ and the component of $G \setminus N[x]$ containing $y$ is $C^x(y)$. The component in $G \setminus N[y]$ containing $x$ is $C^y(x)$. The vertices in $C^x(y) \cap C^y(x)$ form a separator of $x$ and $y$ which is precisely the set $I(x, y)$. Note that $C^x(y) \cap C^y(x)$ may be empty.

In the following we state the necessary lemma from [3] for our purpose which provide us with some characterization for $I(x, y)$.

In the following lemma we consider a connected AT-free graph $G(V, E)$. We consider an interval $I(x, y)$ of $G$ and assume that $s \in I(x, y)$. Lemma 1 is obtained using the fact that $G$ is AT-free.

▶ **Lemma 1** (Broersma et al. [3]). *The vertices $x$ and $y$ are in different components of $G \setminus N[s]$ for each $s \in I(x, y)$.*

Thus every path between $x$ and $y$ either contains $s$ or some neighbour of $s$. The next three lemma states a decomposition of an interval into disjoint intervals and disjoint components. Lemma 2 states that the intervals $I(x, s)$ and $I(s, y)$ have empty intersection. This implies that $P_{s,x} \cap P_{s,y} \subseteq N[s]$, where $P_{s,x}$ is an arbitrary $s, x-$path and $P_{s,y}$ is an arbitrary $s, y-$path in $G$.

▶ **Lemma 2** (Broersma et al. [3]). *The intervals $I(x, s)$ and $I(s, y)$ have no vertices in common, that is $I(x, s) \cap I(s, y) = \phi$.*

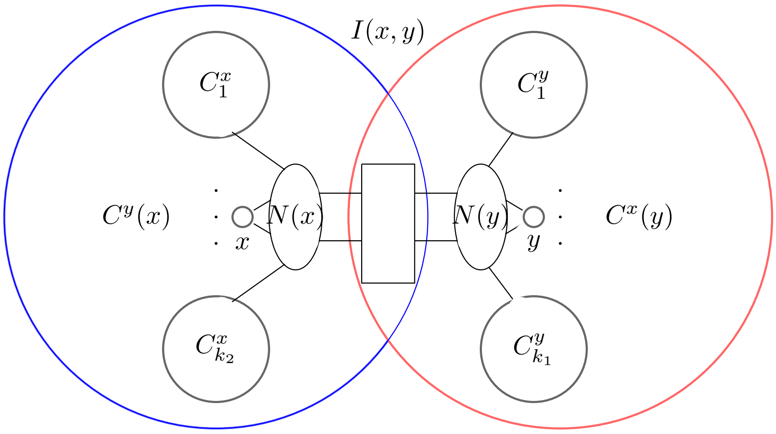


**Figure 2** The component $C^y(x)$ which contains $x$ in $G \setminus N[y]$.

Lemma 3 states a containment relation among the intervals. More precisely, if $s \in I(x,y)$ then $I(x,s) \subseteq I(x,y)$ and so does $I(s,y)$.

▶ **Lemma 3** (Broersma et al. [3]). *The intervals $I(x,s)$ and $I(s,y)$ are both contained in $I(x,y)$, that is $I(x,s) \subseteq I(x,y)$ and $I(s,y) \subseteq I(x,y)$, where $s \in I(x,y)$.*

Combining Lemma 2 and Lemma 3 we arrive at Lemma 4.

▶ **Lemma 4** (Broersma et al. [3]). *In the graph $G \setminus N[s]$ there are components $C_1^s, C_2^s, \ldots, C_t^s$ such that $I(x,y) \setminus N[s] = I(x,s) \cup I(s,y) \cup (\bigcup_{i=1}^t C_i^s)$.*

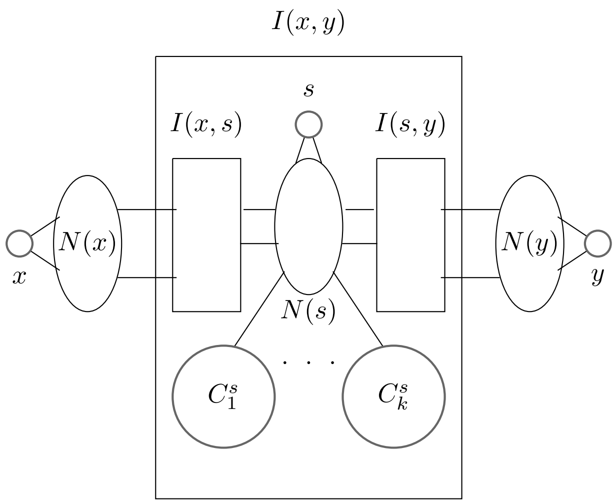


**Figure 3** The interval decomposition.

Similarly the components of $G \setminus N[x]$ can also be decomposed. Consider such a component containing $y$, recall that $y \in C^x(y)$. The following lemma describes the structure of the graph induced on $C^x(y) \setminus N[y]$. In the following we denote the component of $G \setminus N[y]$ containing $x$ by $C^y(x)$.

Consider the graph induced on $C^x(y) \setminus N[y]$ and let $D$ be a connected component of that graph. Lemma 5 essentially states that any vertex of $D$ reaches $N[x]$ using at least one vertex from $I(x,y)$.

▶ **Lemma 5** (Broersma et al. [3]). *Let $D$ be a component of the graph $C^x(y) \setminus N[y]$. Then $N[D] \cap (N[x] \setminus N[y]) = \phi$ if and only if $D$ is a component of $G \setminus N[y]$.*

## 3 Connected Vertex Cover

In the following sections we make some important observations related to the connectivity constraint of the vertex cover and then we formulate the dynamic programming recurrence relations.

### 3.1 Some structural observations

In this section and subsequent sections we assume $G(V, E)$ is a connected AT-free graph. Let $\alpha_c$ be an independent set with maximum cardinality, while ensuring that the subgraph $G[V \setminus \alpha_c]$ remains connected. The complement of $\alpha_c$ forms a connected vertex cover with the smallest possible size. Observe that $\alpha_c$ cannot include a cut vertex. This is because if a vertex $v$ belongs to $\alpha_c$, none of its neighbors can be in $\alpha_c$. If $v$ is a cut vertex, its neighbors would be divided into separate components, leading $G[V \setminus \alpha_c]$ to be disconnected. Hence we have the following observation.

▶ **Observation 6.** *Let $\alpha_c$ be an independent set with maximum cardinality, while ensuring that the subgraph $G[V \setminus \alpha_c]$ remains connected. The set $\alpha_c$ does not include any cut vertex of $G$.*

Let $V'$ denote the set of all cut vertices in $G$. We define some notations that are necessary in the following set of lemma. Let $x$ be a vertex of $G$ which is not a cut vertex. Let $C_1^x, \ldots, C_r^x$ be the components of $G \setminus N[x]$. Let $Z_i$ be those vertices of $N(x)$ which are reachable from $C_i^x$ in the graph $G \setminus \{x\}$, that is without using the vertex $x$ or vertices from any other components. In other words let $C$ be the connected component in $G[N(x) \cup V(C_i^x)]$, then $Z_i = V(C) \setminus V(C_i^x)$. Note that $G[Z_i]$ may not be connected. Suppose $S$ is a connected vertex cover of $G$ and let $x \in V \setminus V'$ such that $x \notin S$. That is $N(x) \subseteq S$. Let $S_i = S \cap (Z_i \cup V(C_i^x))$. Note that $S_i$ contains $Z_i$, since $Z_i \subseteq N(x)$.

▶ **Lemma 7.** *The graph induced on $S \cap (Z_i \cup V(C_i^x))$ is connected.*

**Proof.** Assume for the sake of contradiction, $G[S \cap (Z_i \cup V(C_i^x))]$ is not connected and $H_1, \ldots, H_k$ are the components of $G[S \cap (Z_i \cup V(C_i^x))]$. Each component $H_j$ has $V(H_j) \cap Z_i \neq \phi$, because otherwise $H_j$ is a component of $G \setminus N[x]$. Consider two components $H_l$, $H_r$. There is some vertex $v \in Z_i$ which is adjacent to some vertex of $v' \in V(H_l)$ and there is a vertex $u \in Z_i$ which is adjacent to some vertex $u' \in V(H_r)$. Note that, $v$ is not adjacent to $u$ and $v'$ is not adjacent to $v'$ since $H_l$ and $H_r$ are different components. Hence $v', u', x$ forms an AT. The paths leading $v', u', x$ to form an AT is as follows. The vertices $v'$ and $u'$ are in same component of $G \setminus N[x]$ but not adjacent, hence there is a $u', v'$ path avoiding the neighbours of $x$. The path $x, u, u'$ avoids the neighbours of $v'$ and similarly the path $x, v, v'$ avoids the neighbours of $v'$. ◀

Also note that the graph $S \cap (Z_i \cup V(C_i^x))$ is a vertex cover of the graph $G[Z_i \cup V(C_i^x)]$, since $S$ is a vertex cover of $G$.

▶ **Lemma 8.** *Let $S_i^*$ be a minimum connected vertex cover in $G[Z_i \cup V(C_i^x)]$ and let $S^*$ be a minimum connected vertex cover in $G$. Then $|S_i^*| \leq |S^* \cap (Z_i \cup V(C_i^x))|$.*

**Proof.** Please find the proof in full version of the paper. ◀

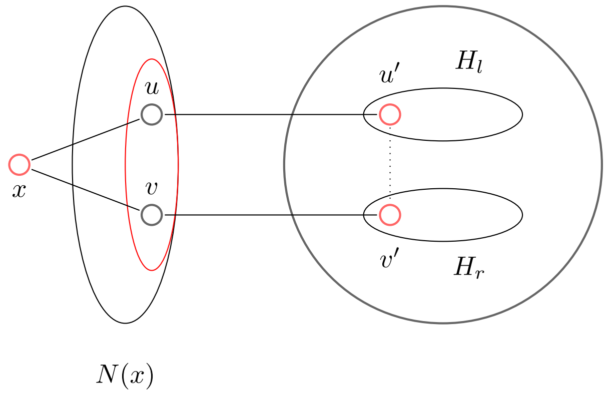


**Figure 4** Illustrating proof of Lemma 7.

▶ **Lemma 9.** *Let $S'_i$ denote a connected vertex cover in $G[Z_i \cup V(C_i^x)]$ containing $Z_i$. The graph induced on the set $N(x) \cup \left(\bigcup_{i=1}^{r} S'_i\right)$ is connected.*

**Proof.** Please find the proof in full version of the paper. ◀

## 3.2 The Dynamic Programming Formulation

The definition of intervals implies that, the set $I(x, y)$ is unique for each pair of non adjacent vertices $x$ and $y$. The above property implies that the number of intervals is bounded by a polynomial in $|V(G)|$. We shall use these intervals to decompose the AT-free graph into smaller disjoint graphs. In the following sections, using this broad idea, we frame the recurrences to find an independent set of maximum size such that its complement is connected. We begin by a graph modification to incorporate the recurrence relation in terms of the intervals.

### 3.2.1 Graph modification

We begin by constructing a modified graph. A result by Corneil et al. [5], ensures that there exists a dominating pair in every AT-free graph which is *pokable*, that is we can append pendant vertices to both of the vertices of the pair maintaining the AT-free property. The following theorem by Corneil et al. [5] states that the process of composing two AT-free graphs.

▶ **Theorem 10** (The Composition Theorem; Corneil et al. [5]). *Given two AT-free graphs $G_1$ and $G_2$, and pokable dominating pairs $(x_1, y_1)$ and $(x_2, y_2)$ in $G_1$ and $G_2$, respectively, let $G'$ be the graph constructed from $G_1$ and $G_2$ by identifying vertices $x_1$ and $x_2$. Then, $G'$ is an AT-free graph.*

Let $p_1, \ldots, p_k$ be a dominating path where $p_1$ and $p_k$ is a pokable dominating pair in $G$. An edge is AT-free, hence we append an edge $(u, v)$ to $p_1$, that is $v$ is adjacent to $p_1$ and an edge $(u', v')$ to $p_k$, that is $v'$ is adjacent to $p_k$. We add edges between $v$ and $p_2$, $v'$ and $p_{k-1}$. We denote this graph by $G'$. More formally, $G'(V, E)$ where,

$$V(G') = V(G) \cup \{u, v, u', v'\}$$

$$E(G') = E(G) \cup \{(u, v), (u', v'), (v, p_1), (v, p_2), (v', p_k), (v', p_{k-1})\}$$

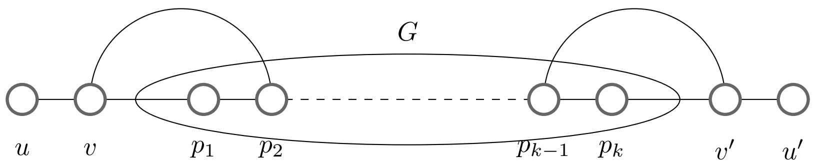


**Figure 5** Illustrating graph modification.

We denote by $\alpha_c$ an independent set such that the remaining set of vertices induces a connected graph.

▶ **Lemma 11.** *The set $\alpha_c$ is a maximum independent set of $G$ such that $G[V \setminus \alpha_c]$ is connected if and only if $\alpha_c$ is a maximum independent set of $G'[I(u,u')]$ such that $G'[V \setminus \alpha_c]$ is connected.*

**Proof.** The set $I(u,u')$ contains all the vertices that has a path to $u$ avoiding the neighbourhood of $u'$ and a path to $u'$ avoiding the neighbourhood of $u$ in the graph $G'$. From the construction of $G'$, every vertex of $V(G)$ satisfies this property. Hence $G$ and $I(u,u')$ are the same. The claim follows since $G$ and $I(u,u')$ are the same. ◀

Now we define the recurrence relations for dynamic programming on the modified graph.

### 3.2.2 The dynamic programming

We decompose the graph in such a way that for any two non adjacent vertices $x$ and $y$ belonging to some connected component $C$ has the property, $I_C(x,y) = I_G(x,y)$. More precisely we remove the closed neighbourhood of a vertex to achieve the smaller subgraphs. From Lemma 1, we can see that the invariant $I_C(x,y) = I_G(x,y) = I(x,y)$ is maintained while solving the subproblems. Broadly our approach is to compute minimum connected vertex cover in smaller connected components and take their disjoint union to obtain a minimum connected vertex cover of a larger component of which the smaller components are part. It is sufficient to calculate the minimum connected vertex cover for smaller components and combine them, which is ensured by Lemma 8 and Lemma 9. We begin by stating a recurrence for a component. In this recurrence we compute a maximum independent set of the component, such that the complement of this independent set (w.r.t the component) is connected. Note that we want the complement to be connected because of our observation in Lemma 7. The recurrence consists of decomposing a given component into interval and some connected components whose disjoint union is the given component as claimed in Lemma 2, Lemma 3 and Lemma 4.

Now we define required notations to state the recurrence formally. Suppose $C$ is a connected component. Let $x$ be a vertex in $V(C)$ which belongs to the independent set. Let the components in $C \setminus N[x]$ be denoted by $C_1^x, \ldots, C_k^x$ if $C \setminus N[x]$ has $k$ connected components.

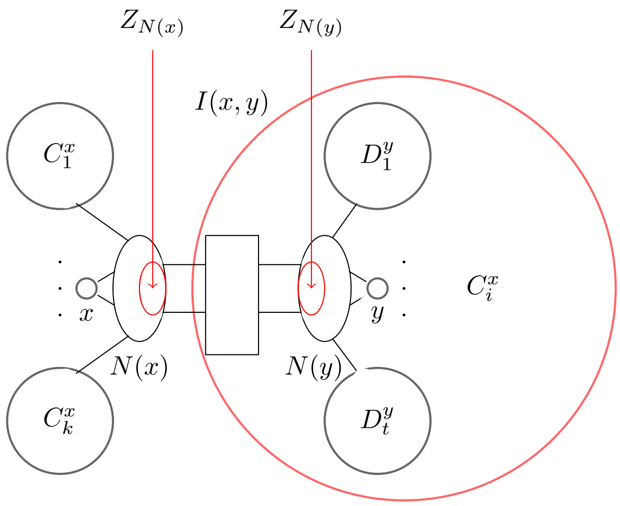


**Figure 6** Illustrating Lemma 12.

In the following we define some notations which are necessary to state the recurrence.

- Consider a component $C_i^x$ in $C \setminus N(x)$.
- Let $V'_{C_i^x}$ be the set of cut vertices of $C[V(C_i^x) \cup Z_{N(x)}]$. We choose $y \in V(C_i^x) \setminus V'_{C_i^x}$ as a candidate for the independent set since from Lemma 7 we know that $C[(V(C_i^x) \cup Z_{N(x)}) \setminus \{y\}]$ is connected.
- Let $I(x,y)$ be the interval for vertices $x$ and $y$.
- Let $Z_{N(x)}$ be those vertices of $N(x)$ that are reachable from $C_i^x$ in $C[N(x)]$ without using $x$ or vertices from any other components.
- Let $Z_{N(y)}$ be those vertices of $N(y)$ that are reachable from $I(x,y)$ in $C[N(y)]$ without using $y$ or vertices from any other components.
- Let $D_1^y, \ldots, D_t^y$ be the components of $C[C_i^x \setminus N[y]]$, and let $H_j$ be those vertices of $N(y)$ that are reachable from $N(y) \cap N(V(D_j^y))$ in $C[N(y)]$ without using $y$ or vertices from any other components.
- We define $\beta(C_i^x, Z_{N(x)})$ to be a maximum independent set in $C_i^x$ such that $C[Z_{N(x)} \cup (V(C_i^x) \setminus \beta(C_i^x, Z_{N(x)}))]$ is connected.
- We define $\gamma(I(x,y), Z_{N(x)} \cup Z_{N(y)})$ to be a maximum independent set in $I(x,y)$ such that $G[Z_{N(x)} \cup Z_{N(y)} \cup (I(x,y) \setminus \gamma(I(x,y), Z_{N(x)} \cup Z_{N(y)}))]$ is connected.

▶ **Lemma 12.** *The recurrence for $\beta$ is as follows.*

$$|\beta(C_i^x, Z_{N(x)})| = 1 + \max_{y \in C_i^x \setminus V'_{C_i^x}} \left( |\gamma(I(x,y), Z_{N(x)} \cup Z_{N(y)})| + \sum_{j=1}^{t} |\beta(D_j^y, H_j)| \right)$$

**Proof.** Please find the proof in the appendix. ◀

Note that if $C$ is the whole graph then $x$ is not a cut vertex of $G$.

Now we state recurrence for an interval. In this recurrence we compute a maximum independent set of the interval, such that the complement of this independent set (w.r.t the interval) is connected. Note that we want the complement to be connected because of our observation in Lemma 7. The recurrence consists of decomposing a given interval into disjoint sub intervals and some connected components whose disjoint union is the given interval as claimed in Lemma 2, Lemma 3 and Lemma 4.

▶ **Observation 13.** *The graph $G[I(x,y) \cup Z_{N(x)} \cup Z_{N(y)}]$ is connected.*

**Proof.** Please find the proof in full version of the paper. ◀

- Note that the definitions of $x$, $y$, $Z_{N(x)}$, $Z_{N(y)}$ and $\gamma(I(x,y), Z_{N(x)} \cup Z_{N(y)})$ remains same as earlier.
- Let $V'_{I(x,y)}$ be the set of cut vertices in $C[I(x,y) \cup Z_{N(x)} \cup Z_{N(y)}]$. We choose $s \in I(x,y) \setminus V'_{I(x,y)}$ as a candidate for the independent set in $I(x,y)$, since from Lemma 7, we know that $C[(I(x,y) \cup Z_{N(x)} \cup Z_{N(y)}) \setminus \{s\}]$ is connected.
- Let $A_{N(x)}$ be those vertices of $N(x)$ that are reachable from $I(x,s)$ in $C[N(x)]$ without using $x$ and any vertex from other components.
- Let $A_{N(s)}$ be those vertices of $N(s)$ that are reachable from $I(x,s)$ in $C[N(s)]$ without using $s$ and any vertex from other components.
- Let $B_{N(y)}$ be those vertices of $N(y)$ that are reachable from $I(y,s)$ in $C[N(y)]$ without using $y$ and any vertex from other components.
- Let $B_{N(s)}$ be those vertices of $N(s)$ that are reachable from $I(s,y)$ in $C[N(s)]$ without using $s$ and any vertex from other components.
- Let $Y_1^s, \ldots, Y_l^s$ are the components of $G[I(x,y) \setminus N[s]]$, and $H_j$ are those vertices of $N(s)$ that are reachable from $Y_j^s$ is $N(s)$ in $G[N(s)]$ without using the vertex $s$ and vertices from other components.

We need the following lemma to prove the correctness of the recurrence for the intervals. Lemma 14 is similar to 7 and Lemma 15 is similar to Lemma 9.

▶ **Lemma 14.** *Let $S$ be a connected vertex cover of $G$ such that $x, y \notin S$. Then $S \cap (I(x,y) \cup Z_{N(x)} \cup Z_{N(y)})$ induces a connected subgraph.*

**Proof.** Please find the proof in full version of the paper. ◀

Let $S_{I(x,s)}$ denote vertices of a connected vertex cover in $G[Z_{N(x)} \cup I(x,s) \cup A_{N(s)}]$ such that $(Z_{N(x)} \cup A_{N(s)}) \subseteq S_{I(x,s)}$ and let $S_{I(s,y)}$ denote a connected vertex cover in $G[Z_{N(y)} \cup I(s,y) \cup B_{N(s)}]$ such that $(Z_{N(y)} \cup B_{N(s)}) \subseteq S_{I(s,y)}$. Let $S_j$ denote a connected vertex cover in $G[H_j \cup V(Y_j^s)]$ containing $H_j$.

▶ **Lemma 15.** *The graph induced on the set $N(s) \cup \left(\bigcup_{i=1}^{r} S_i\right) \cup S_{I(x,s)} \cup S_{I(s,y)}$ is connected.*

**Proof.** Please find the proof in full version of the paper. ◀

Please see the Figure 7 for clarification of the following lemma. Note that $Z_{N(x)}$ and $A_{N(x)}$ are same and $Z_{N(y)}$ and $B_{N(y)}$ are same.

▶ **Lemma 16.** *The recurrence for $\gamma$ is as follows. Let $Z = Z_{N(x)} \cup Z_{N(y)}$.*

$$|\gamma(I(x,y), Z)| =$$

$$1 + \max_{s \in I(x,y) \setminus V'_{I(x,y)}} \left( |\gamma(I(x,s), A_{N(x)} \cup A_{N(s)})| + |\gamma(I(s,y), B_{N(y)} \cup B_{N(s)})| + \sum_{j=1}^{s} |\beta(Y_j^s, H_j)| \right)$$

**Proof.** Please find the proof in the appendix. ◀

Consider the modified graph $G'$. Since $(p_1, p_2)$ and $(p_{k-1}, p_k)$ are edges of $G$ (also of $G'$), the connected vertex cover must contain at least one endpoint from each of those edges. The $\gamma(I(u,u'), \{v,v'\})$ is a maximum independent set such that, $G'[\{v,v'\} \cup (I(u,u') \setminus \gamma(I(u,u'), \{v,v'\})]$ is connected. Since $I(u,u')$ is the graph $G$, $G'[V \setminus \gamma(I(u,u'), \{v,v'\})]$ is our desired solution from Lemma 11.

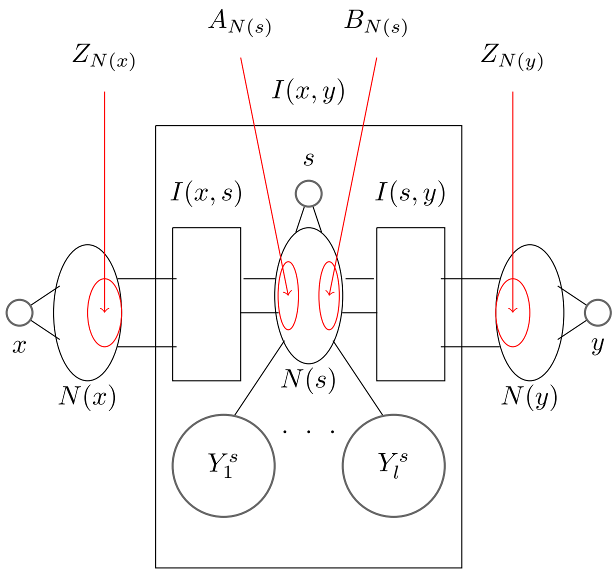


**Figure 7** Illustrating Lemma 16.

## 3.3 Running time analysis

We employ dynamic programming technique to solve the above recurrences to obtain the minimum connected vertex cover. Let $\mathcal{C}$ be a set of components of $G$ as defined below.

$$\mathcal{C} = \bigcup_{v \in V(G)} \{C : C \text{ is a component of } G \setminus N[v]\}$$

We prove in the following observation that the components for which we compute $\beta$ are precisely the members of $\mathcal{C}$. The proof of the following observation is a repeated application of Lemma 5.

▶ **Observation 17.** *The non-interval components in the above recurrence relations are members of $\mathcal{C}$.*

**Proof.** Please find the proof in full version of the paper. ◀

Observation 17 ensures that the number of non-interval components is $|\mathcal{C}| = O(n^2)$.

Let $\mathcal{I}$ denote the set of all possible intervals. The collection $\mathcal{I}$ has cardinality at most $n^2$, that is $|\mathcal{I}| \leq n^2$, since $I(x,y)$ is unique for each pair of non adjacent vertex $x$ and $y$. We arrange the list of all intervals and components in the non decreasing order of the number of vertices. We compute the recurrences for this two lists in the order they are arranged.

First we discuss the complexity to solve the recurrence for intervals. Consider a particular interval $I(x,y)$. We have to go through all the vertices in that $I(x,y)$ and there is at most $O(n)$ such vertices and for each vertex there can be at most $O(n)$ components. Note that these components are smaller than the component that $I(x,y)$ is part of. Hence the solution for each of the components are already stored in the dynamic programming table. Since there is at most $O(n^2)$ intervals and each can take $O(n^2)$ time it takes $O(n^4)$ to find the solutions.

The time complexity to solve the components is calculated similarly and it is also upper bounded by $O(n^4)$. Since all the other computation can be done in time $O(n^4)$ the complexity of our algorithm is $O(n^4)$.

## 4 Conclusion

In this paper, we present a polynomial time algorithm to compute a minimum connected vertex cover on AT-free graphs. Note that even though we have considered an unweighted graph, this algorithm can be modified in such a way that it also works for weighted AT-free graphs.

It will be interesting to explore the complexity of MCVC for those graph classes where MVC is solvable in polynomial time.

### References


1 Esther M Arkin, Magnús M Halldórsson, and Rafael Hassin. Approximating the tree and tour covers of a graph. *Information Processing Letters*, 47(6):275–282, 1993.

2 Hari Balakrishnan, Anand Rajaraman, and C Pandu Rangan. Connected domination and steiner set on asteroidal triple-free graphs. In *Algorithms and Data Structures: Third Workshop, WADS'93 Montréal, Canada, August 11–13, 1993 Proceedings 3*, pages 131–141. Springer, 1993.

3 Hajo Broersma, Ton Kloks, Dieter Kratsch, and Haiko Müller. Independent sets in asteroidal triple-free graphs. *SIAM Journal on Discrete Mathematics*, 12(2):276–287, 1999.

4 Derek G Corneil, Stephan Olariu, and Lorna Stewart. Computing a dominating pair in an asteroidal triple-free graph in linear time. In *Workshop on Algorithms and Data Structures*, pages 358–368. Springer, 1995.

5 Derek G Corneil, Stephan Olariu, and Lorna Stewart. Asteroidal triple-free graphs. *SIAM Journal on Discrete Mathematics*, 10(3):399–430, 1997.

6 Bruno Escoffier, Laurent Gourvès, and Jérôme Monnot. Complexity and approximation results for the connected vertex cover problem in graphs and hypergraphs. *Journal of Discrete Algorithms*, 8(1):36–49, 2010.

7 Henning Fernau and David F Manlove. Vertex and edge covers with clustering properties: Complexity and algorithms. *Journal of Discrete Algorithms*, 7(2):149–167, 2009.

8 Michael R Garey and David S. Johnson. The rectilinear steiner tree problem is np-complete. *SIAM Journal on Applied Mathematics*, 32(4):826–834, 1977.

9 Petr A Golovach, Daniël Paulusma, and Erik Jan van Leeuwen. Induced disjoint paths in at-free graphs. In *Algorithm Theory–SWAT 2012: 13th Scandinavian Symposium and Workshops, Helsinki, Finland, July 4-6, 2012. Proceedings 13*, pages 153–164. Springer, 2012.

10 Jiong Guo, Rolf Niedermeier, and Sebastian Wernicke. Parameterized complexity of generalized vertex cover problems. In *WADS*, pages 36–48. Springer, 2005.

11 Matthew Johnson, Giacomo Paesani, and Daniël Paulusma. Connected vertex cover for $(sp_1 + p_5)$-free graphs. In *Algorithmica82*, pages 20–40, 2020.

12 Dieter Kratsch. Domination and total domination on asteroidal triple-free graphs. *Discrete Applied Mathematics*, 99(1-3):111–123, 2000.

13 Dieter Kratsch, Haiko Müller, and Ioan Todinca. Feedback vertex set on at-free graphs. *Discrete Applied Mathematics*, 156(10):1936–1947, 2008.

14 Daniel Mölle, Stefan Richter, and Peter Rossmanith. Enumerate and expand: New runtime bounds for vertex cover variants. In *Computing and Combinatorics: 12th Annual International Conference, COCOON 2006, Taipei, Taiwan, August 15-18, 2006. Proceedings 12*, pages 265–273. Springer, 2006.

15 Hannes Moser. Exact algorithms for generalizations of vertex cover. *Institut für Informatik, Friedrich-Schiller-Universität Jena*, 12, 2005.

16 Andrea Munaro. Boundary classes for graph problems involving non-local properties. *Theoretical Computer Science*, 692:46–71, 2017.

17 PK Priyadarsini and T Hemalatha. Connected vertex cover in 2-connected planar graph with maximum degree 4 is np-complete. *International Journal of Mathematical, Physical and Engineering Sciences*, 2(1):51–54, 2008.

18 Carla Savage. Depth-first search and the vertex cover problem. *Information processing letters*, 14(5):233–235, 1982.

19 Shuichi Ueno, Yoji Kajitani, and Shin'ya Gotoh. On the nonseparating independent set problem and feedback set problem for graphs with no vertex degree exceeding three. *Discrete Mathematics*, 72(1-3):355–360, 1988.

20 Toshimasa Watanabe, Satoshi Kajita, and Kenji Onaga. Vertex covers and connected vertex covers in 3-connected graphs. In *1991 IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 1017–1020. IEEE, 1991.