Theoretical Computer Science 898 (2022) 20–29

Contents lists available at ScienceDirect

Theoretical Computer Science

www.elsevier.com/locate/tcs

On the maximum cardinality cut problem in proper interval graphs and related graph classes

Arman Boyacı [a], Tınaz Ekim [a], Mordechai Shalom [b,*]

[a] *Department of Industrial Engineering, Boğaziçi University, Istanbul, Turkey*
[b] *Department of Computer Engineering, Işık University, Istanbul, Turkey*

A R T I C L E I N F O

Article history:
Received 7 July 2020
Received in revised form 17 September 2021
Accepted 18 October 2021
Available online 30 October 2021
Communicated by F.V. Fomin

Keywords:
Maximum cut
Proper interval graph
Clique decomposition
Clique-width
Bubble model

A B S T R A C T

Although it has been claimed in two different papers that the maximum cardinality cut problem is polynomial-time solvable for proper interval graphs, both of them turned out to be erroneous. In this work we consider the parameterized complexity of this problem. We show that the maximum cardinality cut problem in proper/unit interval graphs is FPT when parameterized by the maximum number of non-empty bubbles in a column of its bubble model. We then generalize this result to a more general graph class by defining new parameters related to the well-known clique-width parameter.
Specifically, we define an (α, β, δ)-clique-width decomposition of a graph as a clique-width decomposition in which at each step the following invariant is preserved: after discarding at most δ labels, a) every label consists of at most β sets of twin vertices, and b) all the labels together induce a graph with independence number at most α. We show that for every two constants $\alpha, \delta > 0$ the problem is FPT when parameterized by β plus the smallest width of an (α, β, δ)-clique-width decomposition.

© 2021 Elsevier B.V. All rights reserved.

1. Introduction

Maximum Cardinality Cut: A *cut* of a graph $G = (V(G), E(G))$ is a partition of $V(G)$ into two subsets $S, \bar{S}$ where $\bar{S} = V(G) \setminus S$. The *cut-set* of $(S, \bar{S})$ is the set of edges of G having exactly one endpoint in S. *The maximum cardinality cut problem (*MaxCut*)* is to find a cut with a maximum size cut-set, of a given graph.

MaxCut remains NP-hard when restricted to the following graph classes: chordal graphs, undirected path graphs, split graphs, tripartite graphs, co-bipartite graphs [2], unit disk graphs [7] and total graphs [15]. On the positive side, it was shown that MaxCut can be solved in polynomial-time in planar graphs [16], in line graphs [15], in graphs with bounded clique-width [12], and the class of graphs factorable to bounded treewidth graphs [2]. None of these results applies to proper interval graphs. As for the parameterized complexity of the problem in general graphs, MaxCut when parameterized by the clique-width of the input graph is not in FPT unless the Exponential Time Hypothesis (ETH) collapses [12].

Maximum Cardinality Cut in Proper Interval Graphs: Polynomial-time algorithms for some subclasses of proper interval graphs (also known as indifference graphs) are proposed in [1] and in [11], for split indifference graphs and co-bipartite chain graphs (a.k.a. co-chain graphs), respectively. A polynomial-time algorithm for proper interval graphs is proposed in [3]. However, as pointed out in [1] this algorithm contains a flaw and may return sub-optimal solutions. A polynomial-time

* Corresponding author at: Işık University, Istanbul, Turkey.
E-mail address: cmshalom@gmail.com (M. Shalom).

https://doi.org/10.1016/j.tcs.2021.10.014
0304-3975/© 2021 Elsevier B.V. All rights reserved.

algorithm for proper interval graphs was also proposed by the authors of this work [5]. However, this algorithm too, is flawed as explained in detail in [18]. Consequently, the question of whether MaxCut can be solved in polynomial time for proper interval graphs is open.

Bubble Model and Clique-width: In [17] the authors introduce the bubble model of proper interval graphs which is a partition of the vertices into a rectangular array of *bubbles*. Recently the bubble model is generalized to mixed unit interval graphs and a sub-exponential exact algorithm is given for this graph class [18].

Proper interval graphs have unbounded clique-width. Specifically, the authors of [14] show that the clique-width of a specific interval graph $I_{n,n}$ whose bubble model has n rows and n columns is at least n. On the other hand, the clique-width of a mixed unit interval graph (and therefore of a unit interval graph) is upper bounded by the number of rows and also by the number of columns of its bubble model (up to a small constant term) [17,18]. An upper bound of $\Omega(n^3)$ is proven in [19] for $I_{n,n}$-free proper interval graphs. The work [20] considers proper interval graphs with specific bubble models and presents an efficient algorithm to compute the clique-width of proper interval graphs of this type.

Our Contribution: In this work we consider the parameterized complexity of MaxCut when the parameter is the width of a clique-width decomposition of the input graph, having certain properties. Since, as mentioned before, the problem is unlikely to be in FPT in general [12], in order to find FPT algorithms, we confine ourselves to clique-width decompositions having certain properties. The bubble model of proper interval graphs and the extended bubble model of mixed unit interval graphs lead to such clique-width decompositions for these families of graphs.

We introduce new parameters of clique-width decompositions. Specifically, we define a (α, β, δ)-clique-width decomposition of a graph as a clique-width decomposition in which at each step the following invariant is preserved: after discarding at most δ labels, a) every label consists of at most β sets of twin vertices, and b) all the labels together induce a graph with independence number at most α.

We present FPT algorithms for MaxCut when parameterized by the width of the decomposition in which these parameters are bounded by a constant. Since mixed unit interval graphs have decompositions in which these parameters are bounded by constants and the width of these decompositions are equal (up to a small constant term) to the number of non-empty rows in a column, this result implies an FPT algorithm for MaxCut in mixed unit interval graphs when the parameter is the number of non-empty rows in a column of the bubble model. This parameter is in turn equal to (again, up to a small constant term) to the clique size of the corresponding twin-free graph [17,18].

In Section 3 we introduce the notion of bubble partitions. We characterize bubble partitions by two parameters, namely their independence number α and their width. This notion generalizes the two dimensional bubble model of proper interval and mixed unit interval graphs. The parts of the bubble partition of a mixed unit interval graph correspond to the columns of its two dimensional bubble model. Since every column in the bubble model is a clique, a mixed unit interval graph has a bubble partition with independence number 1. Moreover, such a partition can be found in polynomial time [17,18]. We show in Theorem 4 that for every fixed α there is an algorithm that computes a maximum cardinality cut of a graph G given with an α-bubble partition, (i.e., a bubble partition with independence number α). This algorithm runs in FPT time when the parameter is the width of the bubble decomposition. This implies an FPT algorithm for mixed unit interval graphs where the parameter is the number of non-empty bubbles in a column of its two dimensional bubble model.

In Section 4, we extend the scope of this FPT algorithm to a wider domain. In Theorem 6, we show that the XP algorithm for MaxCut when parameterized by clique-width presented in [12] runs in FPT time when parameterized by the smallest width of an (α, β, δ)-clique-width decomposition, denoted by $\mathsf{cw}_{\alpha,\beta,\delta}(G)$ plus β. We also show (in Lemma 1) that a bubble partition with independence number α can be used to find an $(2\alpha, 1, 1)$-clique-width decomposition of similar width. Therefore, in some sense, the main result of Section 4 generalizes the main result of Section 3. This result implies that a mixed unit interval graph has a clique-width decomposition in which at every step, discarding the vertices labeled zero, the graph has independence number two and every label consists of a set of twins. In our terminology, this is a $(2, 1, 1)$-clique-width decomposition implying that the result applies to mixed unit interval graphs.

Denoting by BW_α the class of graphs having a bubble partition with independence number α, we present structural results relating BW_1 to the classes of interval, mixed unit interval, chordal, co-bipartite and split graphs.

We conclude in Section 5 with several open questions about FPT algorithms for MaxCut and related parameters introduced in this paper, both in general and in specific graph classes.

2. Preliminaries

Graph notations and terms: Given a simple graph (i.e., with no loops or parallel edges) $G = (V(G), E(G))$ and a vertex v of G, $N_G(v)$ denotes the set of neighbors of v in G. Two adjacent (resp. non-adjacent) vertices u, v of G are *twins* (resp. *false twins*) if $N_G(u) \setminus \{v\} = N_G(v) \setminus \{u\}$. For a graph G and $U \subseteq V(G)$, we denote by $G[U]$ the subgraph of G induced by U, and $G \setminus U \stackrel{def}{=} G[V(G) \setminus U]$. For a singleton $\{x\}$ and a set Y, $Y + x \stackrel{def}{=} Y \cup \{x\}$ and $Y - x \stackrel{def}{=} Y \setminus \{x\}$. We use $[n] = [1..n]$ to denote the set of positive integers less than or equal to n. A vertex set $U \subseteq V(G)$ is a *clique* (resp. *independent set*) (of G) if every pair of vertices in U is adjacent (resp. non-adjacent). We denote by $\alpha(G)$ the maximum size of an independent set of a graph G. We refer the reader to [8] for general notation and terminology regarding graphs.

Some graph classes: A graph is *chordal* if it does not contain holes, i.e., induced cycles of four or more vertices. It is known that a graph G is chordal if and only if it is the vertex-intersection graph of subtrees of a tree, i.e., there exists a tree T and subtrees $T_1, \ldots, T_n$ of it such that v_i and v_j are adjacent in G if and only if T_i and T_j have a common vertex [13]. A graph

G is *interval* if it is the intersection graph of intervals on a straight line. The subtree intersection characterization of chordal graphs implies that interval graphs are chordal. An interval graph is *proper* (resp. *unit*) if it has an interval representation such that no interval properly contains another (resp. every interval has unit length). It is known that the class of proper interval graphs is equivalent to the class of unit interval graphs [4]. However, if one is allowed to use a mixture of open and closed unit intervals in the representation, richer families of graphs are obtained. This can be easily demonstrated by the set $\{[0,1],(1,2),[2,3],[1,2]\}$ of unit intervals that represent a claw which is not a proper interval graph. The most general such family is the family of *mixed unit interval graphs* of the intersection graphs of unit intervals where each interval can be open, closed, or open at one end and closed at the other.

Cuts: We denote a cut of a graph G by one of the subsets of the partition. $E(S,\bar{S})$ denotes the *cut-set* of S, i.e. the set of the edges of G with exactly one endpoint in S, and $\text{cs}(S) \stackrel{def}{=} \left|E(S,\bar{S})\right|$ is termed the *cut size* of S. A *maximum cut* of G is one having the biggest cut size among all cuts of G. We refer to this size as the *maximum cut size* of G. Clearly, S and $\bar{S}$ are dual; we thus can replace S by $\bar{S}$ and $\bar{S}$ by S everywhere. In particular, $E(S,\bar{S}) = E(\bar{S},S)$, and $\text{cs}(S) = \text{cs}(\bar{S})$.

Parameterized Complexity: A *parameterized problem* is a decision problem each instance of which is a pair (I,k) where k is a number that is termed the *parameter* of the instance. An algorithm that decides a parameterized problem Π is an FPT (resp. XP) algorithm if its running time is bounded by $f(k)\cdot|I|^c$ (resp. $f(k)\cdot|I|^{g(k)}$) for some computable functions f,g and some constant c. The class FPT (resp. XP) is the class of all parameterized problems for which an FPT (resp. XP) algorithm exists. Clearly, FPT ⊆XP. A parameterized problem that is in FPT is termed *fixed-parameter tractable*. The notation $\mathcal{O}^*$ is used to omit polynomial factors. For instance, for an FPT algorithm of time complexity $\mathcal{O}(f(k)\cdot|I|^c)$ for some constant c, we omit the polynomial factor of $|I|^c$ and say that the time complexity of the algorithm is $\mathcal{O}^*(f(k))$. We refer the reader to [10,6] for basic background on parameterized complexity.

Bubble models: A *2-dimensional bubbles model* $\mathcal{B}$ for a finite non-empty set A is a 2-dimensional arrangement of bubbles $\left\{B_{i,j} \mid j\in[k], i\in[r_j]\right\}$ for some positive integers $k, r_1,\ldots,r_k$, such that $\mathcal{B}$ is a *near-partition* of A. That is, $A=\cup\mathcal{B}$ and the sets $B_{i,j}$ are pairwise disjoint, allowing for the possibility of $B_{i,j}=\emptyset$ for arbitrarily many pairs i,j. For an element $a\in A$ we denote by $i(a)$ and $j(a)$ the unique indices such that $a\in B_{i(a),j(a)}$. Given a bubble model $\mathcal{B}$, the graph $G(\mathcal{B})$ has $\cup\mathcal{B}$ as its vertex set. Two vertices u,v are adjacent (in $G(\mathcal{B})$) if and only if $j(u)=j(v)$ or, $j(u)=j(v)+1$ and $i(u)<i(v)$. We say that $\mathcal{B}$ is a *bubble model* for $G(\mathcal{B})$. Observe that every *bubble* $B\in\mathcal{B}$ is a set of twins. A *compact representation* for a bubble model is an array of *columns* each of which contains a list of non-empty bubbles given by their row numbers and their vertices.

Theorem 1. *[17] A graph is proper interval if and only if it has a bubble model. Moreover, a compact representation of a bubble model for a proper interval graph can be computed in linear time.*

Recently the bubble model is extended to a model for mixed unit interval graphs [18]. An extended bubble model corresponds to a bubble model of a proper interval graph. In this model the bubbles are arranged in a rectangular grid. At every point of the grid, instead of one bubble there are four bubbles $B^1_{i,j}, B^2_{i,j}, B^3_{i,j}, B^4_{i,j}$ termed *quadrants* where i and j are the row and column numbers, respectively. Every quadrant contains interval of one type (open at both ends, closed at both ends, and so on). Let G be a mixed-unit interval graph given with an extended bubble model. Consider the bubble model obtained by combining every set of quadrants into a bubble. Let G' be the unit interval graph defined by this bubble model. Two vertices u,v are adjacent in G if and only if they are either adjacent in G' or they are in adjacent columns and same row ($j(u)=j(v)+1$ and $i(u)=i(v)$) and they belong to the appropriate quadrants, i.e., u is in one of the left-closed quadrants and v is in one of the right-closed quadrants. In other words, G' is the intersection graph of the same set of unit intervals except that all endpoints are open.

Theorem 2. *[18] A graph is mixed unit interval if and only if it has an extended bubble model. Moreover, a compact representation of an extended bubble model for a mixed unit interval graph can be computed in linear time.*

We denote by $p(G)$ the maximum number of non-empty bubbles in a column of the bubble model (resp. extended bubble model) $\mathcal{B}$ of a proper (resp. mixed unit) interval graph G.

In this work, we use the term *bubble* as a maximal set of twins and extend the scope of this definition to general graphs, not restricted to proper interval or mixed unit interval graphs. Whenever an ambiguity arises, we use the adjective 2-dimensional for the bubble model of a proper interval graph.

Clique-width: The *clique-width* of a graph G is the minimum number of labels needed to construct G by using the following four operations defined on vertex-labeled graphs:

1. The operation $\ell(v)$ returns a graph with one vertex v labeled ℓ.
2. The disjoint union $G\cup G'$ of two vertex-disjoint labeled graphs G and G' is the graph $(V(G)\cup V(G'), E(G)\cup E(G'))$ and every vertex in $V(G)\cup V(G')$ preserves its original label.
3. The graph $\eta_{i,j}(G)$ is obtained from the graph G by connecting all the vertices labeled i with all the vertices labeled j.
4. The graph $\rho_{i\to j}(G)$ is obtained from the graph G by replacing all labels i with j.

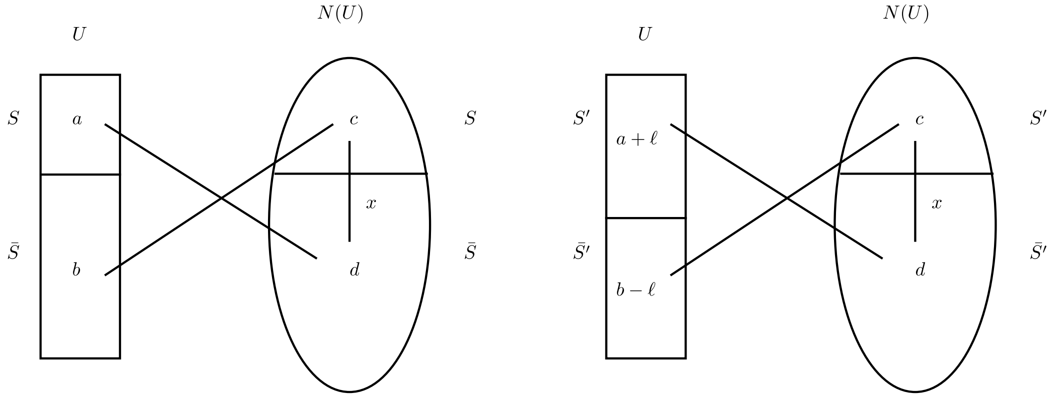

Fig. 1. The cut S contains a vertices of U and c vertices of $N(U)$. The number of edges of $G[N(U)]$ separated by S is x. Then the number of edges of $G[N[U]]$ separated by S is $a \cdot d + b \cdot c + x$. The cut S' on the right side is obtained from S by adding ℓ vertices of U. The number of edges of $G[N[U]]$ separated by S' is $(a+\ell) \cdot d + (b-\ell) \cdot c + x$. Then $\text{cs}(S') - \text{cs}(S) = \ell(d-c)$.

A *clique-width decomposition* of a graph is a rooted binary tree that represents an expression involving the above four operations. The graph G_t corresponding to a node t of T is the value of the expression represented by the subtree of T rooted at t. Let $\mathcal{L}(T)$ be the set of labels used by T. The *width* $w(T)$ of a decomposition T is the number $|\mathcal{L}(T)|$ of labels it uses. The *clique-width* $\text{cw}(G)$ of an (unlabeled) graph G is the smallest width of an expression whose value is G (with some labeling function). For a label $\ell \in \mathcal{L}$, we denote by V_ℓ the set of vertices labeled ℓ and by $V_{t,\ell}$ the set of vertices of G_t labeled ℓ. We denote by $\mathcal{V}_t$ the partition $\{V_{t,\ell} \mid \ell \in \mathcal{L}(T)\}$ of $V(G_t)$.

Decomposition by clique separators: The concept of decomposition by clique separators is introduced by Tarjan [21]. If G is a connected graph and K a clique of G such that $G \setminus K$ is disconnected with connected components $V_1, V_2, \ldots, V_k$ then we decompose G into k subgraphs $G[K \cup G_1], G[K \cup G_2], \ldots, G[K \cup G_k]$. By continuing recursively for every subgraph until a subgraph does not contain a clique separator, we obtain a decomposition of G. This decomposition can be modeled by a tree T where an internal node of T represents a clique of G and the leaves of T represent subgraphs of G termed *atoms* that do not contain clique separators. Given any graph, such a decomposition can be found in polynomial-time.

3. Bubble partitions

Following the definition of 2-dimensional bubble representations of proper interval graphs, we term *bubble* a maximal set of twins. Given a graph G, we denote by G^- the graph obtained by contracting every bubble of G to a single vertex. Two bubbles of G are adjacent (resp. non-adjacent) if the corresponding vertices in G^- are adjacent (resp. non-adjacent).

A *bubble partition* of a graph G is a partition $\mathcal{V} = \{V_1, \ldots V_k\}$ of $V(G)$ such that every $V_i \in \mathcal{V}$ is a union of bubbles and the graph obtained from G by contracting every set V_i to a single vertex is a tree $T(\mathcal{V})$. Note that a bubble partition $\mathcal{V} = \{V_1, \ldots, V_k\}$ of G corresponds to a partition $\mathcal{V}^- = \{V_1^-, \ldots, V_k^-\}$ of $V(G^-)$.

A bubble partition always exists, since $\{V\}$ is a partition whose contraction results in a (trivial) tree. The *independence number* $\alpha(\mathcal{V})$ of a bubble partition $\mathcal{V}$ is $\max\{\alpha(V_i) \mid V_i \in \mathcal{V}\}$, and the *width* $w(\mathcal{V})$ of $\mathcal{V}$ is $\max\{|V_i^-| \mid V_i \in \mathcal{V}\}$, the largest number of bubbles in a set of $\mathcal{V}$. A bubble partition $\mathcal{V}$ with $\alpha(\mathcal{V}) \le \alpha$ is termed an α-bubble partition. The *α-bubble width* $\text{bw}_\alpha(G)$ of G is the smallest width of an α-bubble partition (and ∞ if no such partition exists).

Given a cut S, a set U of (false or true) twin vertices and an integer $\ell \in [-|S \cap U|, |S \setminus U|]$ we denote by $S(U, \ell)$ the cut obtained by adding ℓ vertices of U to S if $\ell \ge 0$ and by removing $|\ell|$ vertices of U from S otherwise.

Observation 1. *Let U be an independent set of pairwise (false) twin vertices of a graph G, and S a cut of G. Then*

$$\text{cs}(S(U,\ell)) - \text{cs}(S) = \ell \cdot \delta(U,S)$$

where $\delta(U,S) \stackrel{def}{=} |N(U) \setminus S| - |N(U) \cap S|$ is the marginal contribution of U to S (see Fig. 1).

Given two adjacent bubbles B, B' and a cut S of a graph G, we denote by $S(B, B', \ell)$ the cut obtained from S by adding to it ℓ vertices of $B \setminus S$ and removing from it ℓ vertices of $B' \cap S$, provided that $\ell \le \min\{|B \setminus S|, |B' \cap S|\}$. Note that $B \cup B'$ is a clique. Since the number of edges of $E[S, \bar{S}]$ in this clique is not affected by this operation, applying Observation 1 twice, we get the following.

Observation 2.

$$\text{cs}(S(B,B',\ell)) - \text{cs}(S) = \ell \cdot (\delta(B,S) - \delta(B',S)) = -(\text{cs}(S(B',B,\ell)) - \text{cs}(S)).$$

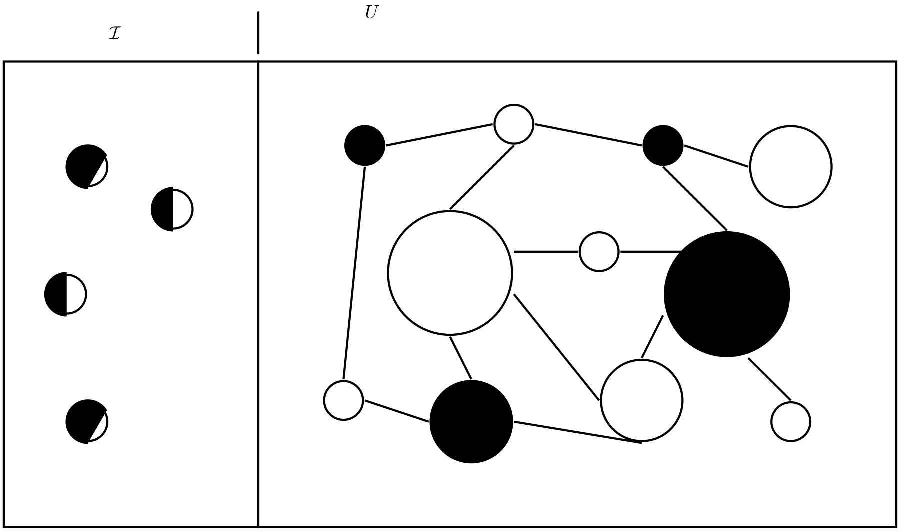

Fig. 2. A tight cut S and a set U of bubbles. The white parts of the bubbles depict the set of vertices in S and the black parts depict the set of vertices not in S. The set $\mathcal{I}$ is the set of bubbles of U crossed by S. The corresponding configuration is $\gamma(S,U) = (\mathcal{I}, \mathbf{s}, \bar{U}^-)$ where **S** contains five positive integers each of which corresponds to a bubble in $\mathcal{I}$ and denotes the number of vertices of S in the corresponding bubble (i.e., the size of the white part of the bubble). The last part of the configuration, i.e., $\bar{U}^-$ is the set of white bubbles on the right side, i.e., the set of bubbles of that are completely contained in S.

Two sets A and B are *crossing* (or A crosses B and vice versa) if their intersection is non-empty and none of them is a subset of the other. A cut S of a graph G is *tight* if the set of bubbles of G crossed by S corresponds to an independent set of G^-. Note that (since a singleton cannot be crossed) if G is twin-free then every cut is of G is tight.

Theorem 3. *Every graph has a maximum cut that is also tight.*

Proof. Suppose that the statement does not hold. Let S be a maximum cut of G that is not tight, such that the number of bubbles that S crosses is smallest possible. Then S crosses at least two adjacent bubbles B, B' of G. We recall that each bubble consists of twin vertices. Therefore, $B \cup B'$ is a clique of G. By Observation 2, at least one of $\text{cs}(S(B, B', \ell)) - \text{cs}(S)$ and $\text{cs}(S(B', B, \ell)) - \text{cs}(S)$ is non-negative for every feasible $\ell \geq 0$. In the sequel we assume that $\text{cs}(S(B, B', \ell)) \geq \text{cs}(S)$ with the other case being symmetric. Let $\ell = \min\left\{|B \setminus S|, \left|B' \cap S\right|\right\}$, and note that $\ell > 0$. Then $S(B, B', \ell)$ is a maximum cut that does not cross at least one of B and B'. Since the intersection of S with other bubbles is not affected, the number of bubbles that S crosses is reduced by one, contradicting the way S is chosen. □

Consult Fig. 2 for this definition. Let U be a union of bubbles. A *configuration* of U for some tight cut S is an encoding $\gamma(S,U)$ of $U \cap S$, defined as follows. $\gamma(S,U)$ is a triple $(\mathcal{I}, \mathbf{s}, \bar{U}^-)$ where $\mathcal{I}$ is a (possibly empty) independent set of $G[U]^-$ that indicates the set of bubbles of U that S crosses, $\mathbf{s}$ is a vector of (non-negative) integers indexed by the elements of $\mathcal{I}$ and $\bar{U}^-$ is a subset of $U^- \setminus \mathcal{I}$. For a bubble $B \in U$ corresponding to a vertex of $\mathcal{I}$, the number $\mathbf{s}_B \in [|B|-1]$ indicates the number of vertices of B in S. The set $\bar{U}^- \subseteq U^- \setminus \mathcal{I}$ indicates the set of bubbles that are completely in S. Therefore, in the sequel we denote $S \cap U$ also as $\gamma(S,U)$, interchangeably. We also denote $\Gamma(U) = \left\{\gamma(S,U) \mid S \text{ is tight}\right\}$.

The first entry of $\gamma(S,U)$ can be chosen in at most $\sum_{i \leq \alpha(G[U]^-)} \binom{|U^-|}{i} < \left|U^-\right|^{\alpha(G[U]^-)+1}$ different ways. The second entry can be chosen in at most $|U|^{\alpha(G[U]^-)}$ different ways, and the last entry can be chosen in at most $2^{\left|U^-\right|}$ ways. Therefore,

$$|\Gamma(U)| \leq |U|^{2\alpha(G[U]^-)+1}\, 2^{\left|U^-\right|} = |U|^{2\alpha(G[U])+1}\, 2^{\left|U^-\right|}.$$

Theorem 4. *Given a bubble partition $\mathcal{V}$ of a graph G, a maximum cut of G can be computed in time $\mathcal{O}^*\left(|V(G)|^{4\alpha(\mathcal{V})}\, 4^{w(\mathcal{V})}\right)$.*

Proof. Consider the tree $T = T(\mathcal{V})$ of the bubble partition $\mathcal{V}$ with an arbitrarily chosen root r. We denote by $\mathcal{C}(t)$ the set of children of a node t in T, and $V_t \in \mathcal{V}$ is the set of $\mathcal{V}$ from the contraction of which t is obtained. Let T_t be the subtree of T induced by t and all of its descendants. Accordingly, G_t denotes the subgraph of G induced by all the vertices of G represented by the nodes of T_t. We process the nodes of T from the bottom to the top and compute a set of best cuts of G_t, namely one cut for each possible configuration of V_t. We terminate after the root r is processed, and choose a configuration in $\Gamma(V_r)$ leading to a maximum cut of G.

For a node t of T and a configuration $\bar{\gamma} \in \Gamma(V_t)$, we denote by $OPT_t(\bar{\gamma})$ the maximum size of a (tight) cut S_t of G_t such that $\bar{\gamma}$ encodes $S_t \cap V_t$, i.e.,

$$OPT_t(\bar{\gamma}) = \max \left\{ \mathrm{cs}(S_t) \mid \gamma(S_t, V_t) = \bar{\gamma} \right\}.$$

By Theorem 3, the maximum cut size of G is $\max_{\gamma \in \Gamma(V_r)} OPT_r(\gamma)$. In the sequel we show how to compute the values $OPT_t(\gamma)$ from the values $OPT_{t'}(\gamma')$ of the children t' of t.

Let S_t be a tight cut of G_t, and for $t' \in \mathcal{C}(t)$, let $S_{t'}$ denote the cut induced by S_t on $G_{t'}$. Denote by $E_{t,t'}(S_t, \bar{S}_t)$ the set of edges between V_t and V'_t that are separated by S_t, i.e. $E_{t,t'}(S_t, \bar{S}_t) = E(G) \cap \left((S_t \cap V_t) \times (\bar{S}_t \cap V_{t'}) \cup (\bar{S}_t \cap V_t) \times (S_t \cap V_{t'}) \right)$. Since the vertices of V_t are adjacent (in G_t) only to vertices of $\cup_{t' \in \mathcal{C}(t)} V_{t'}$, we have

$$\mathrm{cs}(S_t) = \mathrm{cs}(S_t \cap V_t) + \sum_{t' \in \mathcal{C}(t)} \left(\left| E_{t,t'}(S_t, \bar{S}_t) \right| + \mathrm{cs}(S_{t'}) \right).$$

We fix a node t of T and a configuration $\bar{\gamma} \in \Gamma(V_t)$. By definition, we have

$$\begin{aligned} OPT_t(\bar{\gamma}) &= \max \left\{ \mathrm{cs}(S_t \cap V_t) + \sum_{t' \in \mathcal{C}(t)} \left(\left| E_{t,t'}(S_t, \bar{S}_t) \right| + \mathrm{cs}(S_{t'}) \right) \mid \gamma(S_t, V_t) = \bar{\gamma} \right\} \\ &= \mathrm{cs}(\bar{\gamma}) + \max \left\{ \sum_{t' \in \mathcal{C}(t)} \left(\left| E_{t,t'}(S_t, \bar{S}_t) \right| + \mathrm{cs}(S_{t'}) \right) \mid \gamma(S_t, V_t) = \bar{\gamma} \right\}. \end{aligned}$$

We now observe that the terms of the summation corresponding to two children t'_1 and t'_2 of t depend only on the cuts $S_{t'_1}$ and $S_{t'_2}$ induced by S_t and on $S_t \cap V_t$ which is fixed. Therefore, the individual terms can be maximized independently, i.e.,

$$OPT_t(\bar{\gamma}) = \mathrm{cs}(\bar{\gamma}) + \sum_{t' \in \mathcal{C}(t)} \max \left\{ \left| E_{t,t'}(S_t, \bar{S}_t) \right| + \mathrm{cs}(S_{t'}) \mid \gamma(S_t, V_t) = \bar{\gamma} \right\}. \tag{1}$$

For every $t' \in \mathcal{C}(t)$ we partition the set of cuts according to the configurations of t' to get

$$\begin{aligned} \max_{\gamma(S_t, V_t) = \bar{\gamma}} \left(\left| E_{t,t'}(S_t, \bar{S}_t) \right| + \mathrm{cs}(S_{t'}) \right) &= \max_{\bar{\gamma}' \in \Gamma(V_{t'})} \max_{\substack{\gamma(S_t, V_t) = \bar{\gamma} \\ \gamma(S_t, V_{t'}) = \bar{\gamma}'}} \left(\left| E_{t,t'}(S_t, \bar{S}_t) \right| + \mathrm{cs}(S_{t'}) \right) \\ = \max_{\bar{\gamma}' \in \Gamma(V_{t'})} \max_{\substack{\gamma(S_t, V_t) = \bar{\gamma} \\ \gamma(S_t, V_{t'}) = \bar{\gamma}'}} \left(\left| E_{t,t'}(\bar{\gamma}, \bar{\gamma}') \right| + \mathrm{cs}(S_{t'}) \right) \\ = \max_{\bar{\gamma}' \in \Gamma(V_{t'})} \left(\left| E_{t,t'}(\bar{\gamma}, \bar{\gamma}') \right| + \max_{\substack{\gamma(S_t, V_t) = \bar{\gamma} \\ \gamma(S_t, V_{t'}) = \bar{\gamma}'}} \mathrm{cs}(S_{t'}) \right) &= \max_{\bar{\gamma}' \in \Gamma(V_{t'})} \left(\left| E_{t,t'}(\bar{\gamma}, \bar{\gamma}') \right| + OPT_{t'}(\bar{\gamma}') \right) \end{aligned}$$

and substitute in (1)

$$OPT_t(\bar{\gamma}) = \mathrm{cs}(\bar{\gamma}) + \sum_{t' \in \mathcal{C}(t)} \max_{\bar{\gamma}' \in \Gamma(V_{t'})} \left(\left| E_{t,t'}(\bar{\gamma}, \bar{\gamma}') \right| + OPT_{t'}(\bar{\gamma}') \right).$$

Clearly, $E_{t,t'}(\gamma, \gamma')$ can be computed in time $\mathcal{O}(|E(G)|)$ and $OPT_t(\gamma)$ can be computed in time

$$\mathcal{O}^* \left(\sum_{t' \in \mathcal{C}(t)} |\Gamma(V_{t'})| \right) \le \mathcal{O}^* \left(\sum_{t' \in \mathcal{C}(t)} |V_{t'}|^{2\alpha(V_{t'})+1} 2^{\left| V^-_{t'} \right|} \right) \le \mathcal{O}^* \left(|V(G)|^{2\alpha(\mathcal{V})} 2^{w(\mathcal{V})} \right).$$

For every node t, we compute $|\Gamma(V_t)| = \mathcal{O}^*(|V(G)|^{2\alpha(\mathcal{V})} 2^{w(\mathcal{V})})$ values OPT_t. Therefore, the running time of the algorithm is $\mathcal{O}^* \left(|V(G)|^{4\alpha(\mathcal{V})} 4^{w(\mathcal{V})} \right)$. $\square$

Denoting by BW_α the class of graphs G such that $\mathsf{bw}_\alpha(G) < \infty$, we formulate the following corollary of Theorem 4.

Corollary 1. *For every $\alpha > 0$, there is an* FPT *algorithm for* MaxCut *for BW_α when parameterized by $\mathsf{bw}_\alpha(G)$ provided that a bubble partition $\mathcal{V}$ of width $\mathsf{bw}_\alpha(G)$ can be found in time $\mathcal{O}^*(f(\mathsf{bw}_\alpha(G)))$ for some computable function f.*

Recall that $p(G)$ denotes the maximum number of non-empty bubbles in a column of the 2-dimensional bubble model $\mathcal{B}$ of a proper (or mixed unit) interval graph G.

Corollary 2. *There is an* FPT *algorithm for* MaxCut *in mixed unit interval graphs when parameterized by* $p(G)$*. Moreover,* $\mathsf{bw}_1(G) \le p(G)$ *whenever* G *is a mixed unit interval graph.*

Proof. Let G be a mixed unit interval graph. The 2-dimensional bubble-representation $\mathcal{B}$ of G can be computed in polynomial time [18]. Let V_j be a column of $\mathcal{B}$, i.e., $V_j = \bigcup_{i=1}^{r_j} \bigcup_{q=1}^{4} B_{i,j}^q$, and consider the partition $\mathcal{V} = \{V_j \mid j \in [k]\}$. Every set $V_j \in \mathcal{V}$ is a clique and also a union of bubbles. Moreover, the graph obtained from the contraction of every V_j to a single vertex is a path. Therefore, $\mathcal{V}$ is a bubble partition with $\alpha(\mathcal{V}) = 1$ and $w(\mathcal{V}) = p(G)$. By Theorem 4, there is an algorithm for MaxCut that runs in time $\mathcal{O}^*\left(|V(G)|^4 4^{w(\mathcal{V})}\right) = \mathcal{O}^*\left(4^{p(G)}\right)$. □

We conclude this section by relating BW_1 to some known graph classes. By the proof of Corollary 2, BW_1 contains the class of mixed unit interval graphs. It is easy to see that BW_1 contains also the classes of split graphs and co-bipartite graphs. Therefore, we have the following.

Observation 3. *Split* ∪ *Co-Bipartite* ∪ *MixedUnitInterval* ⊆ BW_1.

Clearly, $G \in \mathsf{BW}_1$ if and only if G has a bubble partition where each set is a clique. At first glance, such a bubble partition seems to be a special case of decomposition by clique separators. A result of Dirac [9] implies that a graph is chordal if and only if it has a decomposition by clique separators the atoms of which are cliques. Given these facts it is natural to investigate the relationship between the class BW_1 and the class of chordal graphs.

Theorem 5. BW_1 *crosses both classes of chordal and interval graphs.*

Proof. Since every clique is both chordal, interval and BW_1, the intersection of these classes is non-empty. Moreover, a C_4 (being co-bipartite) is in BW_1 but not chordal. It is now sufficient to show that there is an interval graph which is not BW_1.

Consider the graph G on 8 vertices obtained by adding a universal vertex v_0 to a path P on 7 vertices $v_1, \ldots, v_7$ where the vertices are numbered according to their order on P. It is trivial to construct an interval representation for G. We claim that $G \notin \mathsf{BW}_1$. Assume for a contradiction that $G \in \mathsf{BW}_1$, and let $\mathcal{V} = \{V_0, V_1, \ldots, V_k\}$ be a bubble partition of G such that $\alpha(V_i) = 1$ for $i = 0, \ldots, k$, i.e., every V_i is a clique. Note that each bubble consists of a single vertex as G is twin-free. Assume without loss of generality that $v_0 \in V_0$. Then, the node 0 of $T(\mathcal{V})$ corresponding to V_0 is adjacent to every other node. In other words, $T(\mathcal{V})$ is a star with center 0 and leaves $1, \ldots, k$. Since every V_i is a clique, V_0 contains at most two vertices of P (in addition to v_0). Then $P \setminus V_0$ has at least 5 vertices and at most two connected components, implying that $P \setminus V_0$ has a connected component with at least three vertices. This yields two adjacent nodes in $T(\mathcal{V})$, contradicting that $T(\mathcal{V})$ is a star with center 0. □

4. Clique-width decompositions

It is shown in [12] that a) MaxCut problem can be solved in polynomial time for graphs with bounded clique-width, and b) an FPT algorithm for MaxCut when parameterized by the clique-width of the input graph is impossible under the ETH. In this section, we consider clique-width decompositions with special properties, and the behavior of the MaxCut algorithm under such decompositions. We show that these decompositions in some sense extend bubble partitions. Specifically, we construct a clique-width decomposition the width of which is a constant factor away of the width of a given bubble partition.

We start with properties of clique-width decompositions that we will assume without loss of generality. Let r be the root of a clique-width decomposition T of G (i.e., $G_r = G$), t be a node of T with parent t'.

- If t' is a union node then G_t is an induced subgraph of G. Indeed, if this is not the case, there are pairs of sets $V_{t,\ell}, V_{t,\ell'}$ such that the vertices of $V_{t,\ell}$ and $V_{t,\ell'}$ are adjacent in G but not adjacent in G_t. Then, we can insert an $\eta_{\ell,\ell'}$ node between t and t' for every such pair ℓ, ℓ'. This modification does not affect the width of the decomposition.
- If two vertices u, v are twins in G and $u \in V(G_t)$, then $v \in V(G_t)$. Moreover, u and v have the same label ℓ in G_t. If this is not the case, we can remove from T the node $\ell'(v)$ (and every parent node with one child), and replace the expression $\ell(u)$ by the expression $\eta_{\ell,\ell}(\ell(u) \cup \ell(v))$. Therefore,
- $V(G_t), V_{t,\ell}$ are non-crossing sets, and for every t and ℓ, $\mathcal{V}_t$ is a partition of $V(G_t)$ each set of which does not cross bubbles.
- $u, v \in V(G_t)$ are twins in G if and only if they are twins in G_t and they have the same label in G_t.
- If a cut S is tight then the cut S_t that S induces on G_t is tight.

A clique-width decomposition T is *an* (α, β, δ)*-clique-width decomposition* if for every node t of T, there exists a set $L_t \subseteq \mathcal{L}(T)$ of at most δ labels such that

- the independence number of $G[\bigcup_{\ell \notin L_t} V_{t,\ell}]$ is at most α, and
- the number of bubbles $\left|V_{t,\ell}^-\right|$ of $V_{t,\ell}$ is at most β whenever $\ell \notin L_t$.

We now analyze the running time of the algorithm in [12] that solves MaxCut when provided with an (α, β, δ)-clique-width decomposition T of the input graph G. The algorithm presented in [12] is based on the following observation. For every graph G_t and every label ℓ the vertices of $V_{t,\ell}$ are identical with respect to vertices not in G_t, i.e., every vertex of $G \setminus G_t$ is either adjacent to every vertex of $V_{t,\ell}$ or not adjacent to any of them. Therefore, two cuts S and S' of G that differ only on the vertices of G_t and $\left|S \cap V_{t,\ell}\right| = \left|S' \cap V_{t,\ell}\right|$ for every label ℓ have the same number of edges in $V_t \times \bar{V}_t$. For every node t of T and every vector $\mathbf{s} \in \mathbb{N}^{w(T)}$ such that $0 \le s_\ell \le \left|V_{t,\ell}\right|$ for every $\ell \in \mathcal{L}(T)$ the algorithm computes the maximum cut size among all cuts S such that $\left|S \cap V_{t,\ell}\right| = s_\ell$. The running time of the algorithm is dominated by the computation at union nodes in which, in order to compute the result for a vector $\mathbf{s}'$ of a parent node t', the algorithm considers all the vectors of $\mathbf{s}$ of one of the children and for each such vector, the vector $\mathbf{s}' - \mathbf{s}$ for the other child. Since the number of vectors $\mathbf{s}$ is bounded by $n^{w(T)}$ it follows that the running time of the algorithm is $n^{\mathcal{O}(w(T))}$.

We now improve this upper bound using the above observations. Let S be a tight maximum cut of G. Then the cut S_t that S induces on G_t is also tight, for every node t of T. Therefore, it suffices to consider only cuts that are tight in G_t. More precisely, it is sufficient to compute the results only for vectors that result from a tight cut.

To guess a tight cut S_t we first guess an independent set $\mathcal{I}$ of $G[\bigcup_{i \notin L_t} V_{t,i}]$ in one of the at most $(\beta \cdot (w(T) - \delta))^{\alpha+1}$ ways. Then, for every bubble B that intersects $\mathcal{I}$, we guess the number of vertices in $S_t \cap B$. This can be done in at most $|V(G_t)|^\alpha$ different ways. For every bubble that is a) not labeled with a label from L_t, and b) does not intersect $\mathcal{I}$ we guess whether or not it is contained in S_t. This can be done in at most $2^{\beta \cdot (w(T) - \delta)}$ ways. Finally, we guess the number of vertices of $S_t \cap V_{t,\ell}$ for every label $\ell \in L_t$. This can be done in at most $|V(G_t)|^\delta$ ways. We conclude that the number of vectors $\mathbf{s}$ to consider is at most

$$(\beta \cdot (w(T) - \delta))^{\alpha+1} \, |V(G_t)|^\alpha \, 2^{\beta \cdot (w(T)-\delta)} \, |V(G_t)|^\delta = \mathcal{O}(|V(G_t)|^{\alpha+\delta} \, (\beta \cdot w(T))^{\mathcal{O}(\alpha)} 2^{\beta \cdot w(T)}).$$

Let $\mathsf{cw}_{\alpha,\beta,\delta}(G)$ be the smallest width of an (α, β, δ)-clique-width decomposition of G (and ∞ if no such decomposition exists). We conclude that for every two constants $\alpha, \delta > 0$ the running time of the algorithm is $\mathcal{O}^*((\beta \cdot \mathsf{cw}_{\alpha,\beta,\delta}(G))^{\mathcal{O}(1)} 2^{\beta \cdot \mathsf{cw}_{\alpha,\beta,\delta}(G)})$.

Theorem 6. *The* MaxCut *problem when parameterized by $\mathsf{cw}_{\alpha,\beta,\delta}(G) + \beta$ is in* FPT *for every two constants $\alpha, \delta > 0$.*

We now note that bubbles can be computed efficiently using, for instance, modular decomposition. As already observed, clique-width decompositions can be modified so that twins appear together all the way in the decomposition. Moreover, such a modification does not affect the width of the decomposition. Therefore, for any given clique-width decomposition, the modified algorithm performs at least as good as the original algorithm presented in [12]. When the graph is twin-free or it contains only a small number of twins, both algorithms coincide and their performance depends only on the given clique-width decomposition.

By definition, we have $\mathsf{cw}(G) \le \mathsf{cw}_{\alpha,\beta,\delta}(G) \le \mathsf{cw}_{\alpha,\beta,\delta}(G) + \beta$. The inequality being in the "wrong" direction, it does not allow to deduce an FPT algorithm for MaxCut when parameterized by the clique-width of the graph, what would contradict the impossibility result of [12]. Furthermore, this impossibility result forbids the existence of a computable function f such that $\mathsf{cw}_{\alpha,\beta,\delta}(G) + \beta \le f(\mathsf{cw}(G))$ for every graph G. On the positive size, our result implies an FPT algorithm when the parameter is the clique-width exists for every graph class that admits such a function. Indeed, assume that some graph class $\mathcal{C}$ admits such a function $f_{\alpha,\delta}$ for two constants α, δ. Then, by Theorem 6, there is an algorithm that solves the problem in time $\mathcal{O}^*(g(\mathsf{cw}_{\alpha,\beta,\delta}(G) + \beta) \subseteq \mathcal{O}^*(g(f(\mathsf{cw}(G))))$ for every $G \in \mathcal{C}$.

We now present the following lemma which implies that Theorem 6 in fact extends Theorem 4.

Lemma 1. *Given a bubble partition $\mathcal{V}$ of a graph G, one can find an $(2\alpha(\mathcal{V}), 1, 1))$-clique-width decomposition of G of width $2w(\mathcal{V}) + 1$ in polynomial time. In particular,*

$$\mathsf{cw}_{2\alpha,1,1}(G) \le 2\mathsf{bw}_\alpha(G) + 1.$$

Proof. We present an algorithm to construct a $(2\alpha(\mathcal{V}), 1, 1))$-clique-width decomposition from a given a bubble decomposition $\mathcal{V}$. In this proof we refer to the constructed clique-width decomposition and its parts as *expressions*. Therefore, whenever trees and nodes are mentioned they refer to the bubble decomposition tree and its nodes. We process the tree of the bubble decomposition $\mathcal{V}$ in a bottom-up manner and for every node of the bubble decomposition we construct a clique-width decomposition of the subgraph of G induced by the vertices of the subtree rooted at that node. During this processing, we use a new set of labels for the bubbles of the current node (one label per bubble) and process one child node at a time. For every child we construct the edges connecting it to its parent node and labeled all its vertices with a special label 0 that is never used to construct edges. This leads to a clique-decomposition that uses at most $2w(\mathcal{V}) + 1$ labels and the independence number of which is at most twice the independence number of $\mathcal{V}$.

We now proceed with notation used in the formal description of the process. Let $T = T(\mathcal{V})$ rooted at an arbitrary node r, $\alpha = \alpha(\mathcal{V})$, $w = w(\mathcal{V})$. For every node t of T, let G_t be the subgraph of G induced by the set of vertices that are in the subtree of T rooted at t. Let $\mathcal{L} = \{\ell_1, \ldots, \ell_w\}$, $\mathcal{L}' = \left\{\ell'_1, \ldots, \ell'_w\right\}$ two disjoint label sets. For every node t of T, we construct an expression that satisfies the following conditions:

- it uses only labels from $\mathcal{L} \cup \mathcal{L}' \cup \{0\}$,
- its value is G_t labeled with labels from $\mathcal{L} \cup \{0\}$,
- $\left|V^-_{t,\ell}\right| = 1$ for every label $\ell \in \mathcal{L} \cup \mathcal{L}'$ (in other words, for every label except the special label 0), and
- $\alpha(G_t \setminus V_{t,0}) \le 2\alpha$.

These conditions imply that the expression for G_t (and in particular for $G = G_r$) is an $(2\alpha, 1, 1)$ clique-width decomposition of G of width at most $\left|\mathcal{L} \cup \mathcal{L}' \cup \{0\}\right| = 2w + 1$.

Let $B(k, \ell)$ be the expression $\eta_{\ell,\ell}\left(\bigcup_{i=1}^{k} \ell(i)\right)$ whose value is a k-clique with every vertex labeled by ℓ. For a set X of pairs of labels, we denote by η_X a path consisting of nodes $\eta_{\ell,\ell'}$, one for every pair $(\ell, \ell') \in X$. If $X \subseteq A \times B$ for two disjoint sets A, B we denote by ρ_X a path of $\rho_{i\to j}$ nodes, one for every pair $(i, j) \in X$. Note that $\eta_{i,j}$ operations are commutative and the operations $\rho_{i\to j}$, $\rho_{i'\to j'}$ are commutative if $\left\{i, i'\right\} \cap \left\{j, j'\right\} = \emptyset$, i.e. the above definitions are non-ambiguous. In particular, we denote by $\rho_{\ell\to\ell'}$ the operation of relabeling all nodes labeled ℓ_i with ℓ'_i for every $i \in [w]$ and by $\rho_{\ell'\to 0}$ the operation of relabeling all nodes labeled ℓ'_i with 0 for every $i \in [w]$.

We now describe the construction of the expression for the node t. If t is a leaf of T the graph G_t is the subgraph of G induced by the vertices of V_t. We have $\alpha(V_t) \le \alpha$ and $\left|V^-_t\right| \le w$, i.e. V_t is a union of at most w bubbles. Let k_i be the number of vertices in bubble i. Let E_t be the edge set of $G[V^-_t]$. Then the value of $\eta_{E_t}(\bigcup_{i\in[w]} B(k_i, \ell_i))$ is G_t and it satisfies all the conditions above.

If t is an internal node of T with children $t_1, \ldots, t_k$, let $e_1, \ldots, e_k$ be the expressions for $G_{t_1}, \ldots, G_{t_k}$ each of which satisfies the conditions. We first construct an expression for $G[V_t]$ in the same way as it is done for a leaf. Then we iterate over all the children of t and for every child $t_{k'}$ of t we

- relabel all the labels $\mathcal{L}$ of $G_{t_{k'}}$ by $\mathcal{L}'$,
- take the disjoint union with $G_{t_{k'}}$,
- add the edges between V_t and $V_{t_{k'}}$, and
- relabel all the labels $\mathcal{L}'$ by 0.

Again, it is an easy task to check that all the conditions above are satisfied.

We finish our proof by giving a formal description of the expression corresponding to an internal node t. For $k' \in [0, k]$ we construct an expression e'_i whose value is the subgraph of G_t induced by the vertices V_t and all the vertices in the subtrees of $t_1, \ldots, t_{k'}$ where vertices of V_t are labeled with labels from $\mathcal{L}$ and the rest are labeled 0. Then, the expression e'_k is an expression for G_t. The expression e'_0 is $\eta_{E_t}(\bigcup_{i\in[w]} B(k_i, \ell_i))$ where k_i is the number of vertices in bubble i of V_t. For $k' \in [k]$, let $E_{k'} \subseteq \mathcal{L} \times \mathcal{L}'$ be such that $(i.j') \in E_{k'}$ if and only if there is an edge between vertices labeled i in V_t and vertices labeled j in $V_{t_{k'}}$. Then $e'_{k'} = \rho_{\mathcal{L}'\to 0}(\eta_{E_{k'}}(e'_{i-1} \cup \rho_{\mathcal{L}\to\mathcal{L}'}(e_i)))$. □

Combining Lemma 1 with Corollary 2 we get the following corollary.

Corollary 3. *A $(2, 1, 1)$-clique-width decomposition of width $2p(G) + 1$ can be computed in polynomial-time whenever G is a mixed unit interval graph.*

We note that result in Corollary 3 is not optimal. Using the nested neighborhood structure of the adjacent columns, one can construct a $(2, 1, 1)$-clique-width decomposition of width $p(G) + 1$ of an interval graph or mixed-unit interval graph [17,18].

5. Conclusion and future work

Discussion of the results: In this work, we introduced bubble partitions of graphs and new parameters for clique-width decompositions that we denote by α, β and δ. Our results imply that the existing XP algorithm for MaxCut parameterized by clique-width presented in [12] can be modified to run in FPT time for the parameter $\mathsf{cw}(\alpha, \beta, \delta) + \beta$ for every two constants α, δ. We have shown that bubble partitions with bounded width and independence number can be used to find a clique-width decomposition with bounded values of α, β and δ. It is known that for mixed unit interval graphs such bubble partitions can be found in polynomial time [17,18].

Our results make use of the concept of tight cuts. We show that tight cuts have the following property: any set of distinct representatives of the sets of twins that are crossed by such a cut constitutes an independent set. Moreover, every graph has a maximum cut that is also tight. Since a singleton cannot be crossed, the above mentioned independent set is empty whenever the graph is twin-free. For the same reason, i.e., since a cut cannot cross a singleton, the part of our algorithm that guesses an independent set can be modified to discard vertices without twins. In other words, the modified algorithm will exhaustively search independent sets of non-singleton bubbles. In the extreme case where the graph is twin-free, there is no need to guess an independent set because the only relevant independent set is the empty set.

Possible Extensions and Open Problems: Our work can be extended in the following directions. The dynamic programming algorithm for bubble partitions can be extended to cases where the partition induces a graph with small tree-width instead of a tree. One can extend the "look-ahead" for bubbles, to structures that can be decomposed into a small number of modules. One can study the time complexity of constructing a bubble partition $\mathcal{V}$ having particular $\alpha(\mathcal{V}), \mathsf{bw}(\mathcal{V})$ parameters. Such a study may also be confined to specific graph classes. It would be also interesting to use such partitions when dealing with problems other than MaxCut.

A bubble partition with independence number 1 for unit interval graphs and mixed unit interval graphs easily follows from [17] and [18]. The complexity of computing $\mathsf{bw}_\alpha(G)$ in general, or for specific graph classes is a research problem that was out of the scope of this work.

It is known that an FPT algorithm for MaxCut when parameterized by clique-width is unlikely for general graphs [12]. The existence of such an algorithm for proper interval graphs is an open question. Though, our algorithm is such one for a subclass of proper interval graphs. Namely, this is the family of graphs $\left\{G | G^- = I_{n,m}, n \leq m\right\}$ whose two dimensional bubble model has more columns than rows and does not contain empty bubbles. Indeed, if $G^- = I_{n,m}$ and $n \leq m$, we have $p(G) = n \leq \mathsf{cw}(G)$ where the last inequality is by [14].

There seems to be a close relation between bubble partitions with independence number 1 and decomposition by clique separators whose atoms are cliques. The latter is known to coincide with the class of chordal graphs. On the other hand, we have shown that the former class neither includes nor is included in the class of chordal graphs. The characterization of chordal graphs (and interval graphs) that admit a bubble partition with independence number 1 is an interesting research question too.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] H. Bodlaender, C. de Figueiredo, M. Gutierrez, T. Kloks, R. Niedermeier, Simple max-cut for split-indifference graphs and graphs with few P_4's, in: C. Ribeiro, S. Martins (Eds.), Experimental and Efficient Algorithms, in: Lecture Notes in Computer Science, vol. 3059, Springer, Berlin Heidelberg, 2004, pp. 87–99.
[2] H.L. Bodlaender, K. Jansen, On the complexity of the maximum cut problem, Nord. J. Comput. 7 (2000) 14–31.
[3] H.L. Bodlaender, T. Kloks, R. Niedermeier, SIMPLE MAX-CUT for unit interval graphs and graphs with few P_4s, Electron. Notes Discrete Math. 3 (1999) 19–26.
[4] K.P. Bogart, D.B. West, A short proof that 'proper = unit', Discrete Math. 201 (1–3) (1999) 21–23.
[5] A. Boyacı, T. Ekim, M. Shalom, A polynomial-time algorithm for the maximum cardinality cut problem in proper interval graphs, Inf. Process. Lett. 121 (May 2017) 29–33.
[6] M. Cygan, F.V. Fomin, L. Kowalik, D. Lokshtanov, D. Marx, M. Pilipczuk, M. Pilipczuk, S. Saurabh, Parameterized Algorithms, Springer, 2015.
[7] J. Diaz, M. Kaminski, Max-cut and max-bisection are NP-hard on unit disk graphs, Theor. Comput. Sci. 377 (1–3) (2007) 271–276.
[8] R. Diestel, Graph Theory, 4th edition, Graduate Texts in Mathematics, vol. 173, Springer, 2012.
[9] G.A. Dirac, On rigid circuit graphs, Abh. Math. Semin. Univ. Hamb. 25 (1961) 71–76.
[10] R.G. Downey, M.R. Fellows, Fundamentals of Parameterized Complexity, Texts in Computer Science, Springer, 2013.
[11] T. Ekim, A. Boyacı, M. Shalom, The maximum cardinality cut problem in co-bipartite chain graphs, J. Comb. Optim. 35 (Jan 2018) 250–265.
[12] F.V. Fomin, P.A. Golovach, D. Lokshtanov, S. Saurabh, Algorithmic lower bounds for problems parameterized by clique-width, in: Proceedings of the Twenty-First Annual ACM-SIAM Symposium on Discrete Algorithms, Society for Industrial and Applied Mathematics, 2010, pp. 493–502.
[13] F. Gavril, The intersection graphs of subtrees in trees are exactly the chordal graphs, J. Comb. Theory 16 (1974) 47–56.
[14] M.C. Golumbic, U. Rotics, On the clique-width of some perfect graph classes, Int. J. Found. Comput. Sci. 11 (03) (2000) 423–443.
[15] V. Guruswami, Maximum cut on line and total graphs, Discrete Appl. Math. 92 (2–3) (1999) 217–221.
[16] F. Hadlock, Finding a maximum cut of a planar graph in polynomial time, SIAM J. Comput. 4 (3) (1975) 221–225.
[17] P. Heggernes, D. Meister, C. Papadopoulos, A new representation of proper interval graphs with an application to clique-width, Electron. Notes Discrete Math. 32 (2009) 27–34.
[18] J. Kratochvíl, T. Masařík, J. Novotná, U-bubble model for mixed unit interval graphs and its applications: the maxcut problem revisited, in: J. Esparza, D. Král' (Eds.), 45th International Symposium on Mathematical Foundations of Computer Science, MFCS 2020, Prague, Czech Republic, August 24–28, 2020, in: LIPIcs, vol. 170, Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020, pp. 57:1–57:14.
[19] V. Lozin, Minimal classes of graphs of unbounded clique-width, Ann. Comb. 15 (2011) 707–722.
[20] D. Meister, U. Rotics, Clique-width of full bubble model graphs, Discrete Appl. Math. 185 (2015) 138–167.
[21] R.E. Tarjan, Decomposition by clique separators, Discrete Math. 55 (2) (1985) 221–232.