

Tight extended formulations for independent set*

Austin Buchanan Sergiy Butenko

September 30, 2015

Abstract

This paper describes tight extended formulations for independent set. The first formulation is for arbitrary independence systems and has size $O(n + \mu)$, where μ denotes the number of inclusion-wise maximal independent sets. Consequently, the extension complexity of the independent set polytope of graphs is $O(1.4423^n)$. The size $O(2^{\mathbf{tw}}n)$ of the second extended formulation depends on the treewidth \mathbf{tw} of the graph, which is a common measure of how tree-like it is. This improves upon the size $O(n^{\mathbf{tw}+1})$ extended formulations implied by the Sherali-Adams reformulation procedure (as shown by Bienstock and Ozbay). This implies size $O(n)$ extended formulations for outerplanar, series-parallel, and Halin graphs; size $2^{O(\sqrt{n})}$ extended formulations for planar graphs; and size $O(1.2247^n)$ extended formulations for graphs of maximum degree three.

Keywords: independent set; extended formulation; stable set polytope; treewidth; independence system; series-parallel; fpt extended formulations;

1 Introduction

Many combinatorial optimization problems are \mathcal{NP} -hard and are unlikely to be solvable in polynomial time. Since linear programs can be solved in

*This paper was first made publicly available on September 15, 2014 on the preprint website Optimization Online. Subsequently, an anonymous referee notified us that our main result of a size $O(2^{\mathbf{tw}}n)$ extended formulation for independent set had been previously shown by Monique Laurent using different techniques (see page 134 of the updated version of [25]). In the months that followed, the size $O(2^{\mathbf{tw}}n)$ bound has been generalized for other problems [2, 23, 24]. For these reasons, we have decided not to pursue the publication of this paper in a journal. However, our proofs are different than those given in the previously mentioned papers and may be valuable to some. In particular, we take advantage of the framework for developing extended formulations due to Martin et al. [30].

polynomial time, this suggests that any collection of linear programs that encodes instances of an \mathcal{NP} -hard problem must have superpolynomial size (or at least must take superpolynomial time to construct). Indeed, recent research has shown that \mathcal{NP} -hard problems such as the traveling salesman problem and the 0-1 knapsack problem admit no polynomial-size extended formulation [15, 37]—irrespective of whether $\mathcal{P} = \mathcal{NP}$. Even worse, the independent set problem does not even admit *approximate* extended formulations of polynomial size, cf. [10, 11].

Consequently, there is no use in searching for extended formulations for independent set that are of polynomial-size for all graphs. However, there are classes of graphs that admit small (extended) formulations. For example, it is known that the edge-formulation for the independent set problem is tight if the graph is a tree (or, more generally, bipartite [18]). Moreover, the maximum weight independent set problem can be solved in time $O(2^w n)$ when given a tree decomposition of width w [6]¹. This gives hope that graphs that are sufficiently “tree-like” admit small extended formulations.

Indeed, as shown in Section 3, the independent set polytope admits an extended formulation with $O(2^{\mathbf{tw}} n)$ variables and constraints, where \mathbf{tw} denotes the graph’s treewidth. This implies small extended formulations for independent set for particular classes of graphs. For example, it is known that planar graphs have $\mathbf{tw} = O(\sqrt{n})$; graphs of maximum degree three have $\mathbf{tw} \leq n/6 + o(n)$; and outerplanar graphs, series-parallel graphs, and Halin graphs have treewidth bounded by a constant. (See the treewidth survey of [5] for results of this type.) This extended formulation generalizes results of Barahona and Mahjoub [1], who showed that series-parallel graphs admit linear-size extended formulations. It also improves upon the size $O(n^{\mathbf{tw}+1})$ extended formulations implied by the Sherali-Adams reformulation procedure [39, 3].

The other extended formulation that we present is more general, applying to arbitrary independence systems. There is a trivial extended formulation for an independence system’s polytope—just write the solution vector as a convex combination of the at most 2^n independent sets. In fact, Braun et al. [9] declare 2^n to be the best known upper bound for the extension complexity of the independent set polytope of graphs. However, as we show in Section 2, this bound can be substantially improved to $O(3^{n/3}) = O(1.4423^n)$ by restricting ourselves to *maximal* independent sets and then relaxing some equalities to inequalities. Another reason for

¹Interestingly, there is reason to believe that this algorithm cannot be improved to $O(1.9999^{\mathbf{tw}} \text{poly}(n))$ [27].

studying this extended formulation is for comparison purposes—to show that the size $O(1.2247^n)$ bound for graphs of maximum degree three is an improvement over that for arbitrary graphs.

1.1 Preliminaries and related work

This paper is concerned primarily with extended formulations for independent set, though the techniques surely apply to other problems. A key lemma to prove the integrality of the treewidth-based formulation comes from Martin et al. [30], who show how to write a variety of dynamic programs as linear programs. Recall that the *size* of an extended formulation is the number of inequalities in its description [20]. For more information about extended formulations in combinatorial optimization, consult the surveys of [14, 20].

We consider a simple graph $G = (V, E)$ with vertex set V and edge set E . Usually, we let $n = |V|$ and $m = |E|$. A subset $S \subseteq V$ of vertices is said to be an independent set if no two vertices in S are adjacent. In literature, independent sets are also referred to as *stable sets* and *vertex packings*. The maximum (weight) independent set problem is the task of finding a largest (weight) independent set in the graph. See the survey of [7] for more information about this problem. The independent set polytope $P(G)$ of G is the convex hull of (characteristic vectors of) independent sets of G . The characteristic vector x^S of S has $x_i^S = 1$ if $i \in S$, and $x_i^S = 0$ otherwise.

$$\begin{aligned} P(G) &= \text{conv}\{x^S \in \{0, 1\}^n \mid S \text{ is an independent set of } G\} \\ &= \text{conv}\{x \in \{0, 1\}^n \mid x_i + x_j \leq 1 \text{ for every } \{i, j\} \in E\} \end{aligned}$$

The structure of $P(G)$ is well-studied in literature [36, 34, 35, 41, 1, 31] and full characterizations of the independent set polytope are known for several graph classes. For example, the edge inequalities and nonnegativity bounds are sufficient precisely for bipartite graphs without isolated vertices. A generalization of the edge inequality is the clique inequality $\sum_{i \in C} x_i \leq 1$, where $C \subseteq V$ is a clique, i.e., a subset of pairwise-adjacent vertices [36]. The clique inequalities and nonnegativity bounds are sufficient to describe $P(G)$ precisely when G is perfect [18]. While $P(G)$ has a small description for bipartite graphs, perfect graphs do not—at least in the original space of variables. This is because there can be exponentially many maximal cliques in perfect graphs, and each corresponding clique inequality induces a facet of $P(G)$. While the maximum independent set problem is polynomial-time solvable in perfect graphs (via the ellipsoid method and the Lovász theta

function [18]), there are no known polynomial-size extended formulations for $P(G)$ for arbitrary perfect graphs. The best known bound is $n^{O(\log n)}$ by Yannakakis [43]. However, there are polynomial size formulations for comparability graphs and chordal graphs, which are subclasses of perfect graphs [43].

Chvátal [13] conjectured that, for series-parallel graphs, the edge inequalities, odd-hole inequalities, and 0-1 bounds are sufficient to describe $P(G)$. (An odd hole is an odd-cardinality subset $H \subseteq V$ of vertices that induces a cycle graph, and the inequality $\sum_{i \in H} x_i \leq \frac{|H|-1}{2}$ is valid for $P(G)$ [36].) That these inequalities are sufficient when G is series-parallel was proven by Boulala and Uhry [8], but in French. For a short proof in English, consult [28]. It has been noted [1] that $P(G)$ for series-parallel graphs may have exponentially many facets, yet they admit linear-size extended formulations. Since series-parallel graphs have treewidth at most two, the extended formulation provided in this paper generalizes the results of [1]. We note that for any t -perfect graph (i.e., a graph whose independent set polytope is well-described by the 0-1 bounds, edge inequalities, and odd-hole inequalities) there are size $O(n^3)$ extended formulations [43], which are obtained by writing the odd-hole separation problem within the linear program à la Martin [29].

Pulleyblank and Shepherd [38] provide a dynamic programming algorithm for the maximum independent set problem, whose runtime is polynomial for *distance claw-free graphs*. These are the graphs such that for each vertex, neither its neighborhood nor the set of nodes at distance two contain an independent set of size three. They formulate the dynamic program as a linear program of polynomial-size for distance claw-free graphs.

Perhaps the most interesting relationship between the independent set polytope and treewidth is that, starting with the edge formulation for independent set, at most $\text{tw}(G)$ levels of the Sherali-Adams [39] reformulation procedure are needed to obtain $P(G)$ [3]. This shows that for graphs of bounded treewidth, there is a compact extended formulation for $P(G)$. However, the number of variables in the k -th level of the Sherali-Adams reformulation procedure is $\binom{n}{k+1} = O(n^{k+1})$. In contrast, the treewidth-based extended formulation proposed in this paper has size $O(2^{\text{tw}}n)$, which is of *linear-size* for graphs of bounded treewidth.

Very recently there has been a flurry of activity showing *negative* results about the extension complexity of the independent set polytope. It has been noted that any linear program achieving an $O(n^{1/2-\epsilon})$ approximation for independent set has exponential size [10]. A stronger inapproximability

of $O(n^{1-\epsilon})$ [11] holds under the uniform model (described in the following). Note that these results are unconditional and do not depend on the assumption that $\mathcal{P} \neq \mathcal{NP}$. Braun et al. [9] study the average case extension complexity of independent set with two types of formulations in mind which they call the uniform model and the non-uniform model. Essentially, the uniform model requires that the formulation’s constraints be the same for each n -vertex graph, while in the non-uniform model the constraints can depend on the graph’s topology. In this sense, the extended formulations proposed in this paper are non-uniform.

2 Formulation based on maximal independent sets

The first extended formulation is fairly simple and is based on introducing a variable for each maximal independent set of the graph. The number μ of maximal independent sets of a graph is at most $3^{n/3}$, and this bound is tight on what we will call the MM graphs, which are the disjoint union of $n/3$ triangles. Historically, MM referred to Moon and Moser [33], but the same results were given several years earlier by Miller and Muller [32]. Further, all maximal independent sets of an arbitrary graph can be listed in time $O(3^{n/3})$ [12, 40]. In fact, there are output-sensitive algorithms that, for example, list all maximal independent sets in time $O(nm\mu)$, where μ denotes the number of maximal independent sets [42]. As a consequence, if a graph has polynomially many maximal independent sets, not only does its independent set polytope admit a compact extended formulation, but it can be constructed in polynomial time.

While the focus of this paper is on the independent set polytope of graphs, we will state the extended formulation for the more general case of an arbitrary independence system. An independence system is a pair (I, \mathcal{I}) , where I is a finite “ground” set and \mathcal{I} is a collection of subsets of I satisfying:

1. (non-emptiness) $\emptyset \in \mathcal{I}$, and
2. (down-monotonicity) $S \subseteq S' \in \mathcal{I}$ implies $S \in \mathcal{I}$.

In the extended formulation below, x is the decision vector representing the chosen independent set, and for every maximal independent set S , there is a variable y_S . Denote by \mathcal{I}_M the set of all inclusion-wise maximal independent sets.

Extended Formulation 1:

$$\sum_{S \in \mathcal{I}_M} y_S = 1 \tag{1}$$

$$x_i \leq \sum_{S \in \mathcal{I}_M: i \in S} y_S, \text{ for every } i \in I \tag{2}$$

$$y_S \geq 0, \text{ for every maximal independent set } S \in \mathcal{I}_M \tag{3}$$

$$x_i \geq 0, \text{ for every } i \in I \tag{4}$$

Lemma 1. *For an independence system (I, \mathcal{I}) , let $F_1(I, \mathcal{I})$ be the set of all (x, y) satisfying constraints 1, 2, 3, 4. Then the projection of $F_1(I, \mathcal{I})$ onto the x variables is precisely (I, \mathcal{I}) 's independence system polytope $P(I, \mathcal{I})$.*

Proof. First see that $P(I, \mathcal{I}) \subseteq \text{proj}_x F_1(I, \mathcal{I})$. Consider $x' \in P(I, \mathcal{I})$, which we can assume, without loss of generality, is integer. Then x' is the characteristic vector of some independent set I' which is a subset of a maximal independent set I' . Then the binary vector (x', y') belongs to $F_1(I, \mathcal{I})$, where $y'_S = 1$ iff $I' = S$.

To show $P(I, \mathcal{I}) \supseteq \text{proj}_x F_1(I, \mathcal{I})$, let $(u, v) \in F_1(I, \mathcal{I})$. Then also $(x, v) \in F_1(I, \mathcal{I})$, where, for each $i \in I$, $x_i := \sum_{S \in \mathcal{I}_M: i \in S} v_S$. Note that $x \in P(I, \mathcal{I})$, since it belongs to the maximal independent set face of $P(I, \mathcal{I})$. Then, since $0 \leq u \leq x$, and by down-monotonicity of $P(I, \mathcal{I})$ (see, e.g., [19]), we have $u \in P(I, \mathcal{I})$. \square

Theorem 1. *An independence system with n ground elements and μ maximal independent sets admits an extended formulation of size $O(n + \mu)$.*

Corollary 1. *The extension complexity of a graph's independent set polytope $P(G)$ is $O(3^{n/3})$.*

It is not too hard to see that similar results hold if, instead of down-monotonicity, we enforce up-monotonicity, i.e., that $S \supseteq S' \in \mathcal{I}$ implies $S \in \mathcal{I}$. This allows us to write extended formulations for, say, the dominating set polytope of a graph with $O(1.7159^n)$ variables and constraints [16]. The extended formulation also implies that the dominating set polytope admits a compact extended formulation whenever the graph has polynomially many minimal dominating sets. However, it is not yet clear if such an extended formulation could be constructed in polynomial time, as, to date, there is no known *output-polynomial time* algorithm for enumerating minimal dominating sets [21]. This is to be expected for some independence systems, as it has been shown that no algorithm lists all maximal independent sets of an independence system in output-polynomial time, unless $\mathcal{P} = \mathcal{NP}$ [26].

3 Formulation based on treewidth

The second extended formulation that we describe borrows ideas from a treewidth-based dynamic programming algorithm for independent set. We will first represent the problem as a network flow problem of sorts. The directed network that we construct has hyperarcs, complicating the proof of the linear programming formulation’s integrality. For clarity, we will refer to the input graph of the independent set problem as a graph with vertices and edges; the directed graph that represents the network flow problem will be called a network with nodes and (hyper)arcs.

If we based the extended formulation on a pathwidth-based dynamic programming algorithm, then there would be no hyperarcs. In this case, it is pretty straightforward to achieve an extended formulation with $O(2^{\mathbf{pw}}n)$ entities, where \mathbf{pw} denotes pathwidth. It turns out that $\mathbf{pw}(G) = O(\mathbf{tw}(G) \log n)$ so this would yield polynomial-size extended formulations for graphs of bounded treewidth. However, if we construct the formulation from a treewidth-based dynamic programming algorithm, then we can make a stronger claim—that graphs of bounded treewidth admit *linear-size* extended formulations for their independent set polytopes.

Since there is the possibility for hyperarcs, the usual total unimodularity argument is not enough to show that the proposed formulation is integral. Fortunately for us, Martin et al. [30] have shown how to craft extended formulations for these types of dynamic programs. We will only need to construct the necessary directed acyclic hypergraph and show that it fits into their paradigm. First, however, we will need some background information about treewidth and the treewidth-based dynamic programming algorithm.

Definition 1. A tree decomposition of a graph $G = (V, E)$ is a pair (\mathcal{B}, T) , where $T = (J, F)$ is a tree and $\mathcal{B} = \{B_j \mid j \in J\}$ is a collection of subsets of V (each B_j is called a bag) such that

- $\bigcup_{j \in J} B_j = V$;
- for every edge $\{u, v\} \in E$ there is a bag that contains u and v ; and
- for all $i, j, k \in J$: if j is on the path from i to k in T then $B_i \cap B_k \subseteq B_j$.

The width of the decomposition is $\max_i \{|B_i|\} - 1$. The treewidth of G , denoted $\mathbf{tw}(G)$, is the minimum width among the tree decompositions of G .

The “−1” in the definition of width is merely a cosmetic detail done so that the treewidth of a tree is one. A path decomposition is a tree

decomposition, where T is further required to be a path graph. Pathwidth is defined similarly.

While many problems are quickly solvable on graphs of small treewidth, actually determining a graph's treewidth is \mathcal{NP} -hard. However, Bodlaender's theorem states that, for any fixed w , there is a linear-time algorithm that finds a tree decomposition of width w (if one exists). Even though Bodlaender's algorithm runs in linear time for fixed w , its dependence on w is very large and the algorithm is notoriously impractical. Still, there are practical, linear-time algorithms for small values of treewidth, e.g., for $\mathbf{tw} = 1, 2, 3, 4$. Consult the surveys of Bodlaender for these and other facts about treewidth [4, 5].

It will be convenient to work with a *nice* tree decomposition, and from now on we will assume, without loss of generality, that our tree decompositions will be nice and will have $O(n)$ bags. This follows by a standard linear-time algorithm that, when given a tree decomposition, outputs a nice tree decomposition of the same width and with at most $4n$ bags (see Lemma 13.1.2 of [22]).

Definition 2. *A tree decomposition is nice if it is a rooted binary tree such that each node $j \in J$ is one of the following four types:*

- **Leaf nodes** j are leaves of T and have $|B_j| = 1$.
- **Introduce nodes** j have one child c with $B_j = B_c + v$ for some vertex $v \in V$.
- **Forget nodes** j have one child c with $B_j = B_c - v$ for some vertex $v \in V$.
- **Join nodes** j have two children c_1 and c_2 with $B_j = B_{c_1} = B_{c_2}$.

We will now describe the treewidth-based dynamic programming algorithm for weighted independent set [6]. For each bag $B_j \subseteq V$ and for every subset $S \subseteq B_j$ of the bag, let $f(j, S)$ be the weight of a maximum weight independent set I of the subgraph induced by V_j such that $S = I \cap B_j$. Here, V_j is the union of B_j along with all of its descendant bags (not necessarily direct descendants). Whenever S is itself not independent, the subproblem is infeasible with the convention that its objective is $-\infty$. The formula for computing $f(j, S)$ depends on the type of bag B_j . The weight of a vertex v is denoted w_v , and the weight of $S \subseteq V$ is denoted by $w(S) := \sum_{v \in S} w_v$.

- **Leaf node**, where $B_j = \{v\}$. Set $f(j, \emptyset) = 0$ and $f(j, \{v\}) = w_v$.

- **Introduce node**, where $B_j = B_c + v$. For every $S \subseteq B_c$, set

$$f(j, S) = f(c, S), \text{ and}$$

$$f(j, S + v) = \begin{cases} w_v + f(c, S) & \text{if } S + v \text{ is independent} \\ -\infty & \text{otherwise.} \end{cases}$$

- **Forget node**, where $B_j = B_c - v$. For every $S \subseteq B_j$, set

$$f(j, S) = \max\{f(c, S), f(c, S + v)\}.$$

- **Join node**, where $B_j = B_{c_1} = B_{c_2}$. For every $S \subseteq B_j$, set

$$f(j, S) = f(c_1, S) + f(c_2, S) - w(S).$$

The objective of the maximum independent set problem for the original graph can be found by looking at the root bag B_r and computing the maximum of $f(r, S)$ such that $S \subseteq B_r$.

Notice that the algorithm does not depend on the graph's structure, in the sense that dependent subsets are penalized in the objective with a weight of $-\infty$, instead of being explicitly excluded during algorithm's execution. For example, the complete graph on n nodes and the empty graph on n nodes both admit the trivial tree decomposition where a single bag contains all vertices. The algorithm's execution on these two graphs with the trivial decomposition is essentially the same, and hence, a polyhedral representation of this dynamic programming algorithm will not describe the graph's independent set polytope. Hard constraints are necessary.

We are now ready to construct our directed acyclic hypergraph $D = (N, A)$ that will model the treewidth-based dynamic programming algorithm for the independent set problem for a graph $G = (V, E)$. The main idea is to disallow nodes that represent infeasible solutions, i.e., dependent subsets of vertices. We can assume, without loss of generality, that the given tree decomposition is *nicer* and has width w .

Definition 3. A **nicer tree decomposition** is nice tree decomposition with $O(n)$ bags that is rooted at an empty bag.

The node set N is created as follows. For every bag B_j in the tree decomposition, and for every subset $S \subseteq B_j$ that is independent in G (including the empty set), create a node S^j . This implies, by the nicer tree decomposition, a single node $t = \emptyset^r \in N$ from the empty root bag B_r that we will call the sink node. Finally, for every leaf bag B_j , create a source node s_j .

The number of nodes is $|N| = O(2^w n)$, since there are $O(n)$ bags, and for each bag B_j there are at most $2^{|B_j|} \leq 2^{w+1}$ independent sets.

The arc set A will allow a partial solution to “grow” at introduce bags and “shrink” at forget bags. Create A as follows depending on the type of bag B_j .

- **Leaf node**, where $B_j = \{v\}$. Add the arcs (\emptyset, s_j) , (s_j, \emptyset^j) , and $(s_j, \{v\}^j)$. Note that (\emptyset, s_j) is strange in that it has no tail and is called a boundary arc in Theorem 2.
- **Introduce node**, where $B_j = B_c + v$. For every independent $S \subseteq B_c$, add the arc (S^c, S^j) and if $S + v$ is also independent, then add the arc $(S^c, (S + v)^j)$.
- **Forget node**, where $B_j = B_c - v$. For every independent $S \subseteq B_j$, add the arc (S^c, S^j) , and if $S + v$ is also independent, then add the arc $((S + v)^c, S^j)$.
- **Join node**, where $B_j = B_{c_1} = B_{c_2}$. For every independent subset $S \subseteq B_j$, add the *hyperarc* $(\{S^{c_1}, S^{c_2}\}, S^j)$.

The c in S^c and $(S + v)^c$ refers to bag B_c and not to the set’s complement.

Examples of the constructed hypergraphs can be found in Figures 1, 2, and 5. Figure 1 illustrates the most basic case, where each node in the directed network represents at most one vertex in the input graph and there are no ‘join’ bags in the tree decomposition. Figure 2 shows an example where some bags contain independent sets of size two. Figure 3 shows the smallest graph with $\mathbf{tw} = 1$ and $\mathbf{pw} = 2$. A nicer tree decomposition and constructed hypergraph follow in Figures 4 and 5. Since the given tree decomposition has ‘join’ bags, there are hyperarcs in the directed network.

We are now ready to provide the extended formulation. For each (hyper)arc $a \in A$ of D , there is a variable y_a representing the amount of flow across it. As usual, x is the decision vector representing the chosen independent set of G . For a node $v \in N$, $\delta^{out}(v)$ is the set of (hyper)arcs that have v as (one of) its tail(s). The set $\delta^{in}(v)$ is defined similarly. The set $\text{FORGET}(v)$ is the set of all arcs that “forget” $v \in N$, i.e., arcs of the form $((S + v)^c, S^j)$. The polytope $F_2(G)$ is the set of all (x, y) satisfying the following constraints.

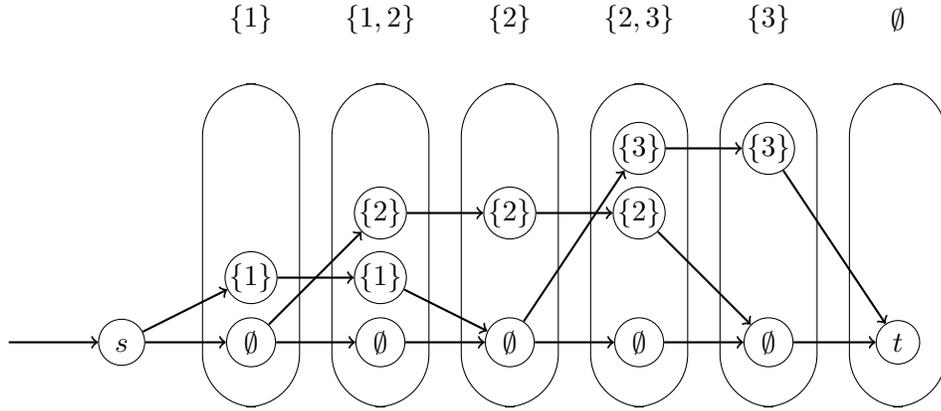


Figure 1: A *nicer* tree decomposition of P_3 (the path on 3 vertices) and the proposed construction D . (This is also a nice path decomposition.) There are no “join” nodes in the tree decomposition, so there is no need for hyperarcs.

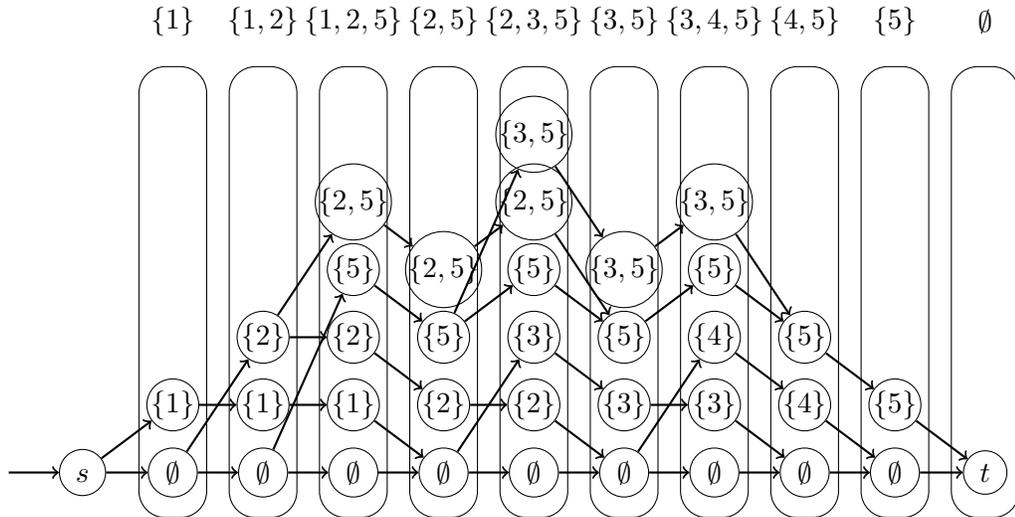


Figure 2: A width-2 *nicer* tree decomposition of the cycle graph on five vertices and the proposed construction D . (This is also a path decomposition.)

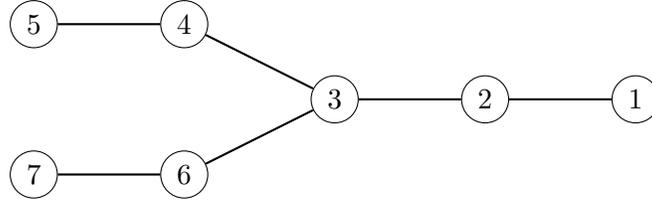


Figure 3: A tree (of pathwidth 2).

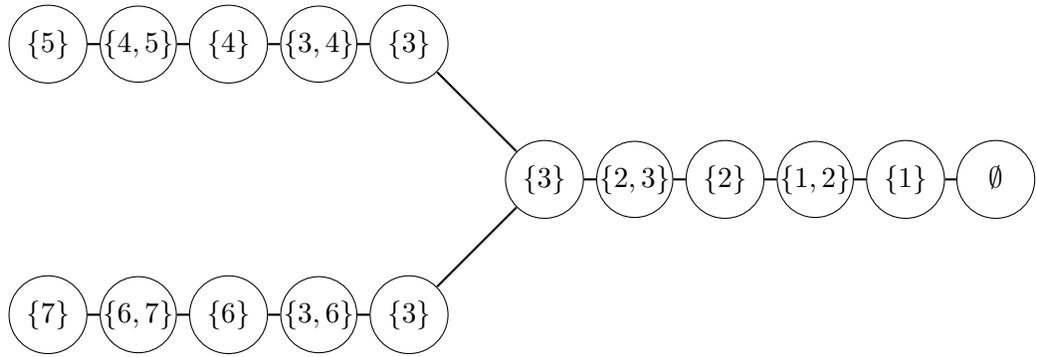


Figure 4: A *nicer* tree decomposition of width 1 that is rooted at the right.

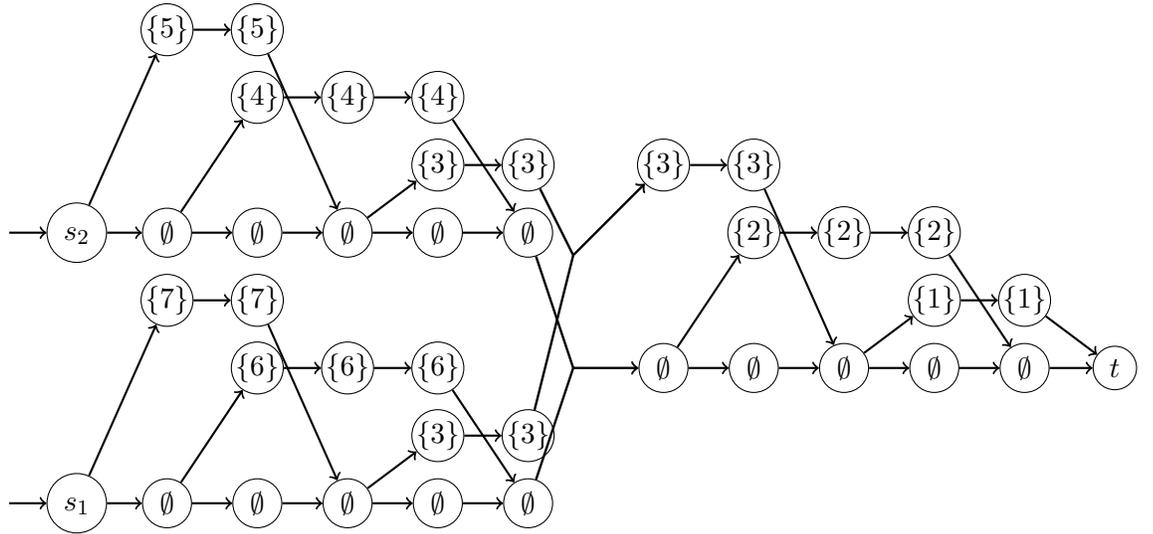


Figure 5: The proposed directed acyclic hypergraph D . Since there is a “join” node in the tree decomposition, D has hyperarcs.

Extended Formulation 2:

$$\sum_{a \in \delta^{in}(t)} y_a = 1, \text{ for sink node } t \quad (5)$$

$$\sum_{a \in \delta^{out}(v)} y_a - \sum_{a \in \delta^{in}(v)} y_a = 0, \text{ for every node } v \in N \setminus \{t\} \quad (6)$$

$$x_i - \sum_{a \in \text{FORGET}(i)} y_a = 0, \text{ for every vertex } i \in V \quad (7)$$

$$y_a \geq 0, \text{ for every (hyper)arc } a \quad (8)$$

Theorem 2 (Martin et al. [30]). *Let $\mathcal{H} = (\mathcal{V}, \mathcal{A})$ be a directed hypergraph such that*

1. *each hyperarc has a single head, i.e., hyperarcs are of the form (J, i) where $J \subseteq \mathcal{V}$ and $i \in \mathcal{V}$;*
2. *\mathcal{H} is acyclic; more specifically, there is a mapping $\sigma : \mathcal{V} \rightarrow \mathbb{R}$ such that for every hyperarc $(J, i) \in \mathcal{A}$ and every $j \in J$, we have $\sigma(j) < \sigma(i)$;*
3. *there is finite set Q and a mapping $f : \mathcal{V} \rightarrow 2^Q$ such that*
 - (a) *f is “consistent” with the acyclicity, namely, for every hyperarc $(J, i) \in \mathcal{A}$ and for every $j \in J$, we have $f(j) \subseteq f(i)$;*
 - (b) *for every hyperarc $(J, i) \in \mathcal{A}$ and for distinct “tails” $j, j' \in J$ of the hyperarc, we have $f(j) \cap f(j') = \emptyset$;*
 - (c) *there is a single “sink” node t with $f(t) = Q$.*
4. *every $i \in \mathcal{V}$ has at least one incoming arc. Since the graph is acyclic this implies that some arcs (called boundary arcs) will have no tail nodes, i.e., arcs of the type (J, i) with $J = \emptyset$.*

Then, the set of all z satisfying the following constraints is a 0-1 polytope.

$$\sum_{a=(J,t) \in \mathcal{A}} z_a = 1 \quad (9)$$

$$\sum_{a=(J,i) \in \mathcal{A}} z_a - \sum_{a=(J,j) \in \mathcal{A}: i \in J} z_a = 0, \text{ for every node } i \in V \setminus \{t\} \quad (10)$$

$$z_a \geq 0, \forall a \in \mathcal{A}. \quad (11)$$

Lemma 2. $\text{proj}_y(F_2(G))$ *is a 0-1 polytope.*

Proof. Apply Theorem 2. The directed hypergraph that we constructed clearly satisfies points 1, 2, and 4. For point 3, let Q be the set of source nodes, and for $v \in N$, let $f(v)$ be the set of source nodes from which there is a directed path to v in D . \square

Lemma 3. *In a nicer tree decomposition, each vertex is ‘forgotten’ once, i.e., for each $v \in V$, there is one pair (B_j, B_c) of bags, where B_j is the parent of B_c , such that $B_j = B_c - v$.*

Proof. Each vertex is forgotten at least once, since each vertex belongs to at least one bag and all vertices have been forgotten by the empty root bag. Now suppose that a vertex is forgotten at least twice, so that there are distinct bags $B_{j_1} = B_{c_1} - v$ and $B_{j_2} = B_{c_2} - v$ that forget v . We consider two cases. In the first case, assume that one of the bags that forgets v is a descendant of the other bag that forgets v . Without loss of generality suppose that B_{j_2} is a descendant of B_{j_1} . Then, bags B_{c_2} and B_{c_1} both contain v , but bag B_{j_2} does not, yet it lies between B_{c_1} and B_{c_2} , contradicting the tree decomposition. In the second case, B_{j_1} is neither a descendant nor an ancestor of B_{j_2} . In this case, they lie in different branches of the tree and both of B_{j_1} and B_{j_2} lie on the unique path between B_{c_1} and B_{c_2} , and the same contradiction occurs. \square

Note that for a feasible solution (x, y) to $F_2(G)$ there will be one unit of flow ‘from’ bag B_c ‘to’ its parent B_j . For example, when $B_j = B_c - v$, we have

$$\sum_{\substack{a=(S^c, S^j) \in A \\ \text{s.t. } S \subseteq B_c \text{ is independent}}} y_a + \sum_{\substack{a=((S+v)^c, S^j) \in A \\ \text{s.t. } S+v \subseteq B_j \text{ is independent}}} y_a = 1. \quad (12)$$

If this flow were greater (less) than one, then the flow into the sink node t would be greater (less) than one, violating constraint (5).

Lemma 4. *$F_2(G)$ is a 0-1 polytope.*

Proof. First see that $F_2(G)$ is an integral polytope, since $\text{proj}_y(F_2(G))$ is a 0-1 polytope (by Lemma 2), and since there is a nonnegative integer matrix M such that $x = My$. Now we must show that the x variables are bounded by zero and one. By Lemma 3, for any vertex $v \in V$, there will be one bag

$B_j = B_c - v$ that forgets v . Then, for any $(x, y) \in F_2(G)$, we have that

$$\begin{aligned}
0 \leq x_v &= \sum_{a \in \text{FORGET}(v)} y_a \\
&\leq \sum_{\substack{a=(S^c, S^j) \in A \\ \text{s.t. } S \subseteq B_c \text{ is independent}}} y_a + \sum_{a \in \text{FORGET}(v)} y_a \\
&= \sum_{\substack{a=(S^c, S^j) \in A \\ \text{s.t. } S \subseteq B_c \text{ is independent}}} y_a + \sum_{\substack{a=((S+v)^c, S^j) \in A \\ \text{s.t. } S+v \subseteq B_j \text{ is independent}}} y_a = 1.
\end{aligned}$$

□

Lemma 5. $P(G) \subseteq \text{proj}_x(F_2(G))$.

Proof. Consider $x \in P(G)$. Without loss of generality, suppose that x is an extreme point of $P(G)$, and is thus the characteristic vector of an independent set I . We construct an integral feasible point of $F_2(G)$ as follows. For every non-boundary arc $a = (S_1^c, S_2^j) \in A$ that is not a hyperarc, set

$$y_a = \begin{cases} 1, & \text{if } S_1 = B_c \cap I \text{ and } S_2 = B_j \cap I \\ 0, & \text{otherwise} \end{cases}$$

For each boundary arc, set the corresponding variable to one. Similarly, for every hyperarc, say $a = (\{S^{c_1}, S^{c_2}\}, S^j) \in A$, set $y_a = 1$ iff $S = B_j \cap I$. Then, for every arc, say $a = (s_j, S^i)$, emanating from a source node s_j , set $y_a = 1$ iff $S = B_i \cap I$. It can be verified that $(x, y) \in F_2(G)$. □

Lemma 6. $\text{proj}_x(F_2(G)) \subseteq P(G)$.

Proof. Consider $(x', y') \in F_2(G)$. Without loss of generality, suppose that (x', y') is an extreme point of $F_2(G)$. By Lemma 4, this means that (x', y') is 0-1. We are to show that $x' \in P(G)$. By the flow constraints of $F_2(G)$, the integrality of (x', y') , and equality (12), the set of all arcs with positive flow induce a directed tree of D —a sort of reverse arborescence rooted at the sink \emptyset^t with the boundary arcs at the leaves. We claim that $S' := \{i \in V \mid x'_i > 0\}$ is an independent set in G . Suppose not, then there exist adjacent $u, v \in S'$. By the tree decomposition, there is a bag B_{j_1} that contains u and v . Further, there is a unique path $(S_1^{j_1}, S_2^{j_2}, S_3^{j_3}, \dots, \emptyset^t)$ leading to the sink node \emptyset^t crossing only arcs of nonzero flow. Notice that, by Lemma 3, there is a single opportunity to “forget” u and a single opportunity to “forget” v along this path, and both arcs must be taken to have $x_u > 0$ and $x_v > 0$. Moreover,

u and v cannot be re-introduced along this path, since this would contradict the tree decomposition. This implies that $S_1^{j_1}$ must contain both u and v , but this contradicts the construction of N , since for every node $S^{j_1} \in N$, S is independent in G . Thus, S' is independent, so $x' = x^{S'} \in P(G)$. \square

Theorem 3. *The extension complexity of a graph's independent set polytope is $O(2^{\mathbf{tw}n})$, where \mathbf{tw} denotes its treewidth.*

Proof. Lemmata 5 and 6 show that $\text{proj}_x(F_2(G)) = P(G)$. Since $F_2(G)$ has size $O(2^{\mathbf{tw}n})$, the result follows. \square

4 Conclusion

There is a trivial extended formulation for the independent set polytope—just take a convex combination of the $O(2^n)$ independent sets of the graph. In this paper, we show how to significantly improve this bound, based on two different approaches. In the first we take ideas from the convex combination approach, but restrict ourselves to maximal independent sets and then relax some equalities to inequalities. As we show in Section 2, this approach achieves an extended formulation with $O(n + \mu)$ entities for *arbitrary* independence systems with μ maximal independent sets. In the second extended formulation, described in Section 3, we use ideas from dynamic programming and treewidth \mathbf{tw} to achieve an extended formulation with $O(2^{\mathbf{tw}n})$ entities.

It should be noted that neither of the two extended formulations proposed in this paper always gives a smaller size bound. For example, the number $\mu(P_n)$ of maximal independent sets of the n -vertex path graph P_n satisfies the recurrence $\mu(P_n) = \mu(P_{n-2}) + \mu(P_{n-3})$ with initial values $\mu(P_{-1}) = \mu(P_0) = \mu(P_1) = 1$, and this sequence, the Padovan sequence, grows as ρ^n , where $\rho = 1.3247\dots$ is the plastic number [17]. This implies that the first extended formulation would use exponentially many variables, but the treewidth-based formulation would have size $O(n)$. In the other extreme, the complete graph K_n on n vertices has $\mathbf{tw}(K_n) = n - 1$, but K_n has n maximal independent sets.

We conclude with an open question. Is the size $O(2^{\mathbf{tw}n})$ bound optimal, or is there an $\epsilon > 0$ for which it can be improved to $O((2 - \epsilon)^{\mathbf{tw}n} \text{poly}(n))$?

Acknowledgements

This material is based upon work supported by the AFRL Mathematical Modeling and Optimization Institute. Partial support by AFOSR under

grants FA9550-12-1-0103 and FA8651-12-2-0011 is also gratefully acknowledged. The authors would like to thank Yiming Wang for providing comments on an early version of this paper.

References

- [1] F. Barahona and A.R. Mahjoub. Compositions of graphs and polyhedra II: stable sets. *SIAM Journal on Discrete Mathematics*, 7(3):359–371, 1994.
- [2] D. Bienstock and G. Munoz. On optimization problems with bounded treewidth. *arXiv preprint arXiv:1501.00288*, 2015.
- [3] D. Bienstock and N. Ozbay. Tree-width and the Sherali–Adams operator. *Discrete Optimization*, 1(1):13–21, 2004.
- [4] H.L. Bodlaender. Treewidth: Algorithmic techniques and results. In *Mathematical Foundations of Computer Science 1997: 22nd International Symposium, MFCS’97, Bratislava, Slovakia, August 25-29, 1997, Proceedings*, volume 22, page 19. Springer, 1997.
- [5] H.L. Bodlaender. A partial- k -arboretum of graphs with bounded treewidth. *Theoretical Computer Science*, 209(1):1–45, 1998.
- [6] H.L. Bodlaender and A.M.C.A. Koster. Combinatorial optimization on graphs of bounded treewidth. *The Computer Journal*, 51(3):255–269, 2008.
- [7] I.M. Bomze, M. Budinich, P.M. Pardalos, and M. Pelillo. The maximum clique problem. In *Handbook of combinatorial optimization*, pages 1–74. Springer, 1999.
- [8] M. Boulala and J.-P. Uhry. Polytope des indépendants d’un graphe série-parallèle. *Discrete Mathematics*, 27(3):225–243, 1979.
- [9] G. Braun, S. Fiorini, and S. Pokutta. Average case polyhedral complexity of the maximum stable set problem. In *RANDOM 2014*, volume 28 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 515–530, 2014.
- [10] G. Braun, S. Fiorini, S. Pokutta, and D. Steurer. Approximation limits of linear programs (beyond hierarchies). In *Foundations of Computer Science (FOCS), 2012 IEEE 53rd Annual Symposium on*, pages 480–489. IEEE, 2012.
- [11] M. Braverman and A. Moitra. An information complexity approach to extended formulations. In *Proceedings of the 45th Annual ACM Symposium on Theory of Computing*, pages 161–170. ACM, 2013.
- [12] C. Bron and J. Kerbosch. Algorithm 457: finding all cliques of an undirected graph. *Communications of the ACM*, 16(9):575–577, 1973.
- [13] V. Chvátal. On certain polytopes associated with graphs. *Journal of Combinatorial Theory, Series B*, 18(2):138–154, 1975.

- [14] M. Conforti, G. Cornuéjols, and G. Zambelli. Extended formulations in combinatorial optimization. *Annals of Operations Research*, 204(1):97–143, 2013.
- [15] S. Fiorini, S. Massar, S. Pokutta, H.R. Tiwary, and R. de Wolf. Linear vs. semidefinite extended formulations: exponential separation and strong lower bounds. In *Proceedings of the 44th Annual ACM Symposium on Theory of Computing*, pages 95–106. ACM, 2012.
- [16] F.V. Fomin, F. Grandoni, A.V. Pyatkin, and A.A. Stepanov. Combinatorial bounds via measure and conquer: Bounding minimal dominating sets and applications. *ACM Transactions on Algorithms*, 5(1):9, 2008.
- [17] Z. Füredi. The number of maximal independent sets in connected graphs. *Journal of Graph Theory*, 11(4):463–470, 1987.
- [18] M. Grötschel, L. Lovász, and A. Schrijver. *Geometric Algorithms and Combinatorial Optimization*. Springer, 1988.
- [19] P.L. Hammer, E.L. Johnson, and U.N. Peled. Facet of regular 0–1 polytopes. *Mathematical Programming*, 8(1):179–206, 1975.
- [20] V. Kaibel. Extended formulations in combinatorial optimization. *Optima*, 85:2–7.
- [21] M.M. Kanté, V. Limouzy, A. Mary, and L. Nourine. Enumeration of minimal dominating sets and variants. In *Fundamentals of Computation Theory*, pages 298–309. Springer, 2011.
- [22] T. Kloks. Treewidth: Computations and approximations. *Lecture Notes in Computer Science*, 842, 1994.
- [23] P. Kolman and M. Koutecký. Extended formulation for CSP that is compact for instances of bounded treewidth. *arXiv preprint arXiv:1502.05361*, 2015.
- [24] P. Kolman, M. Koutecký, and H.R. Tiwary. Courcelle’s Theorem: An extension complexity analogue. *arXiv preprint arXiv:1507.04907*, 2015.
- [25] M. Laurent. Sums of squares, moment matrices and optimization over polynomials. In *Emerging applications of algebraic geometry*, pages 157–270. Springer, 2009. Updated version available at <http://homepages.cwi.nl/~monique/files/moment-ima-update-new.pdf>.
- [26] E.L. Lawler, J.K. Lenstra, and A.H.G. Rinnooy Kan. Generating all maximal independent sets: NP-hardness and polynomial-time algorithms. *SIAM Journal on Computing*, 9(3):558–565, 1980.
- [27] D. Lokshtanov, D. Marx, and S. Saurabh. Known algorithms on graphs of bounded treewidth are probably optimal. In *Proceedings of the 22nd Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 777–789. SIAM, 2011.
- [28] A.R. Mahjoub. On the stable set polytope of a series-parallel graph. *Mathematical Programming*, 40(1):53–57, 1988.

- [29] R.K. Martin. Using separation algorithms to generate mixed integer model reformulations. *Operations Research Letters*, 10(3):119–128, 1991.
- [30] R.K. Martin, R.L. Rardin, and B.A. Campbell. Polyhedral characterization of discrete dynamic programming. *Operations Research*, 38(1):127–138, 1990.
- [31] B. McClosky and I.V. Hicks. Composition of stable set polyhedra. *Operations Research Letters*, 36(5):615–617, 2008.
- [32] R.E. Miller and D.E. Muller. A problem of maximum consistent subsets. *IBM Research Rep. RC-240, IBM Research Center, Yorktown Heights, New York, USA*, 1960.
- [33] J.W. Moon and L. Moser. On cliques in graphs. *Israel Journal of Mathematics*, 3(1):23–28, 1965.
- [34] G.L. Nemhauser and L.E. Trotter Jr. Properties of vertex packing and independence system polyhedra. *Mathematical Programming*, 6(1):48–61, 1974.
- [35] G.L. Nemhauser and L.E. Trotter Jr. Vertex packings: structural properties and algorithms. *Mathematical Programming*, 8(1):232–248, 1975.
- [36] M.W. Padberg. On the facial structure of set packing polyhedra. *Mathematical Programming*, 5(1):199–215, 1973.
- [37] S. Pokutta and M. Van Vyve. A note on the extension complexity of the knapsack polytope. *Operations Research Letters*, 41(4):347–350, 2013.
- [38] W.R. Pulleyblank and B. Shepherd. Formulations for the stable set polytope. In *Proceedings Third IPCO Conference*, pages 267–279, 1993.
- [39] H.D. Sherali and W.P. Adams. A hierarchy of relaxations between the continuous and convex hull representations for zero-one programming problems. *SIAM Journal on Discrete Mathematics*, 3(3):411–430, 1990.
- [40] E. Tomita, A. Tanaka, and H. Takahashi. The worst-case time complexity for generating all maximal cliques and computational experiments. *Theoretical Computer Science*, 363(1):28–42, 2006.
- [41] L.E. Trotter Jr. A class of facet producing graphs for vertex packing polyhedra. *Discrete Mathematics*, 12(4):373–388, 1975.
- [42] S. Tsukiyama, M. Ide, H. Ariyoshi, and I. Shirakawa. A new algorithm for generating all the maximal independent sets. *SIAM Journal on Computing*, 6(3):505–517, 1977.
- [43] M. Yannakakis. Expressing combinatorial optimization problems by linear programs. *Journal of Computer and System Sciences*, 43:441–466, 1991.