Georgia Institute of Technology

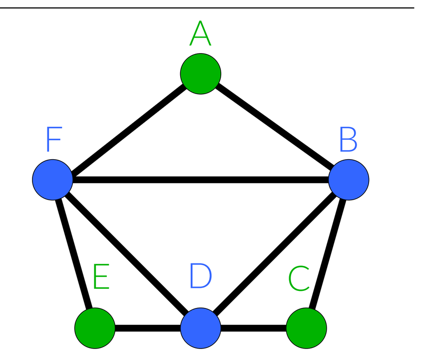
Weighted Independent Set And Bounded Odd Cycle Tree Decompositions

Anshul Tripathy
Joint work with Rose McCarty, Caleb McFarland, and Zach Walsh

U.S. National Science Foundation

Introduction

An **independent set** of a graph $G = (V, E)$ is a set of vertices $S \subseteq V$ such that **no two vertices** in S **share an edge** in G.



Example: In the graph on the right hand side, the set $\{A, C, E\}$ forms an independent set.

Maximum Weighted Independent Set (MWIS) Problem

Given a graph $G = (V, E)$ and a weight on each vertex, v, denoted $w(v)$.

Find an **independent set** $S \subseteq V$ that **maximizes sum of weights** of vertices in S.

In general, **finding the MWIS** in a graph in **NP-Complete**. Thus, searching for a polynomial algorithm for MWIS in general graphs is impractical. However, perhaps special classes of graphs admit polynomial solutions.

Goal: Find **new class of graphs** that admit **polynomial algorithms** for MWIS?

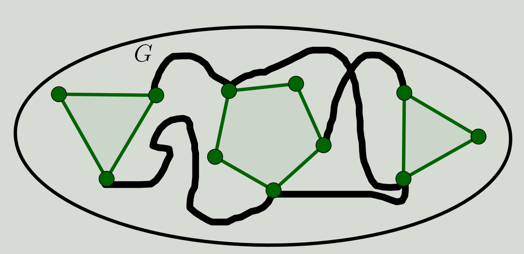
Motivation

In 2022, Fiorini, Joret, Weltge, and Yuditsky give a polynomial algorithm to **solve integer programming** with bounded subdeterminants and only two nonzeros entries per row (or column) via a **reduction to MWIS with bounded vertex disjoint odd cycles**.

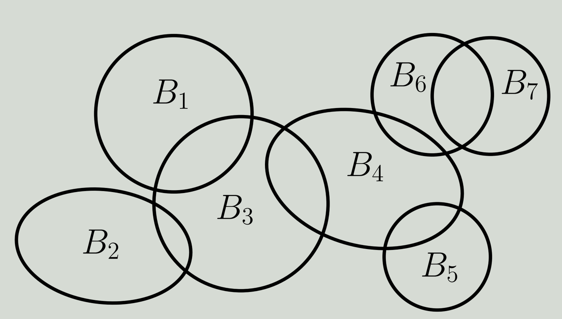
Known Classes of Graphs Admitting Polynomial Algorithm for MWIS

For any integer constant $k > 0$,

1.) graphs with at most k **vertex disjoint odd cycles** admit polynomial algorithms, and



Graph with **three** vertex disjoint odd cycles



All vertex sets $|B_i| \leq k + 1$ for all $i \in \{1, \dots, 7\}$

2.) graphs with **treewidth** at most k admit polynomial algorithms.

Tree Decomposition

A graph $G = (V, E)$ has a tree decomposition (T, B) with **tree** T and a **collection of sets of vertices** B such that

there is a **mapping** between **any vertex** $v \in T$ to a **set of vertices in** G (called a **bag**) $B_v \in \mathsf{B}$.

A tree decomposition must have that

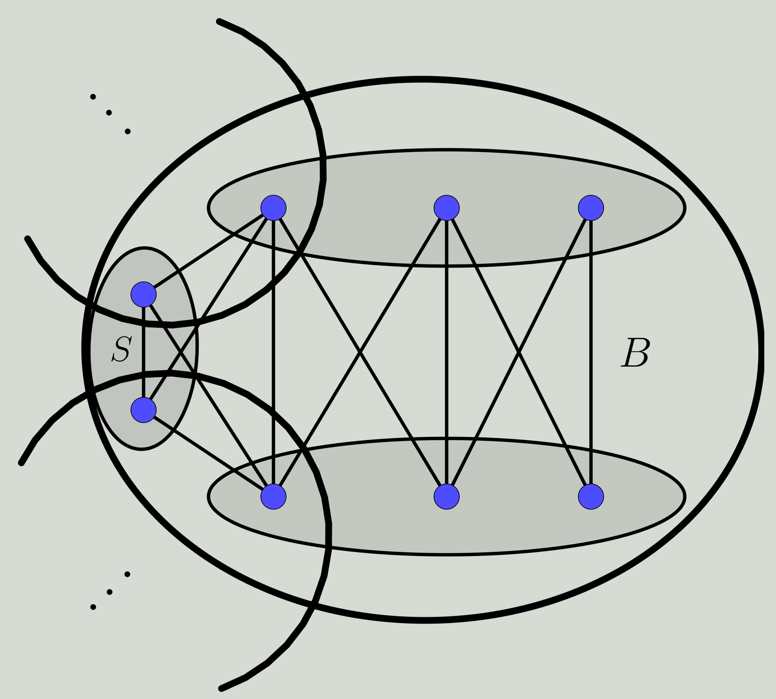
- every vertex $v \in V$ is in some bag $B_i \in \mathsf{B}$,
- every edge $uv \in E$ is contained in a bag B_i (i.e both u,v are in B_i for some bag $B_i \in \mathsf{B}$), and
- for any path ijk in T, we have that $B_i \cap B_k \subseteq B_k$.

Our Bounded Odd Cycle Tree Decomposition

Let us say that a graph G has a **bounded odd cycle** tree decomposition (T, B) if for each bag $B \in \mathsf{B}$ you can remove a **constant** k number of vertices, S, from B to get that

- G no odd cycle among vertices $B \setminus S$, and
- for any other bag $B' \neq B$, the intersection between bags $|B' \cap (B \setminus S)| \leq 1$.

The first rule ensures there are **few vertex disjoint odd cycles** in any bag $B \in \mathsf{B}$. The second rule ensures that **different bags** have **few vertices in common**

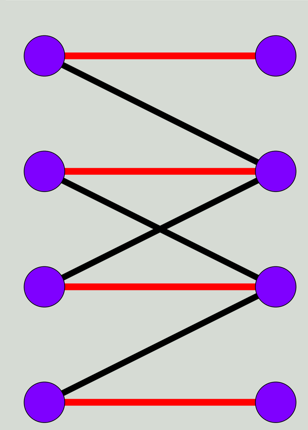


Question: Does the class of graphs with a bounded odd cycle tree decomposition for a constant k admit polynomial algorithms for MWIS.

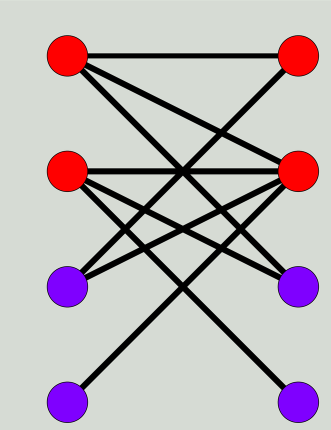
Subroutine: Find the maximum weighted independent set in a given bag

Since k is a constant, we can simply guess which vertices are in S of size $|S| = k$. We can test all 2^k possible subsets of S that can be in our maximum weighted independent set. Both computations would be polynomial in terms of n.

Try all $\binom{n}{k}$ ways to choose k vertices to be S	$O(n^k)$
Check if the graph on vertices $B \setminus S$ has no odd cycles and that $\forall B' \neq B, \lvert B' \cap (B \backslash S)\rvert \leq 1$	$O(n^2)$
Try all 2^k subsets of S that could be in the maximum weighted independent set	$O(2^k)$
Find a **maximum weight matching** in G among vertices $B \setminus S$ using Ford Fulkerson's Algorithms	$O(n^3)$



Matching is red



Independent set is blue
Vertex cover is red

Theorem: For any graph G, I is an **independent set** if an only if $V \setminus I$ is a **vertex covering**, a set of vertices that are incident to every edge in G.

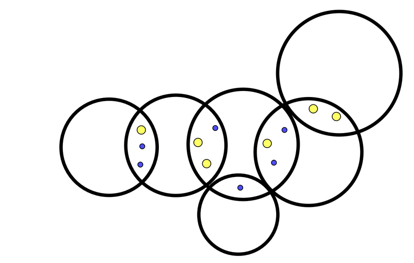
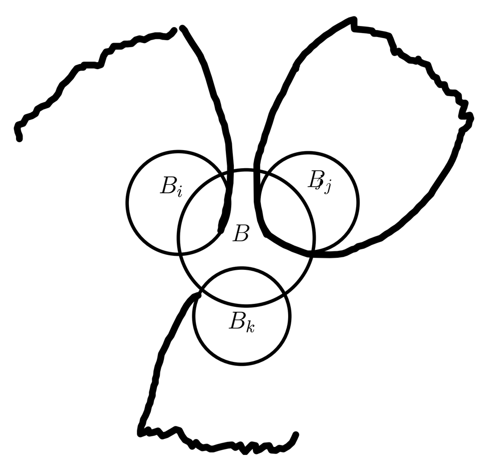
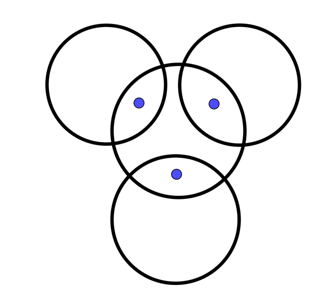
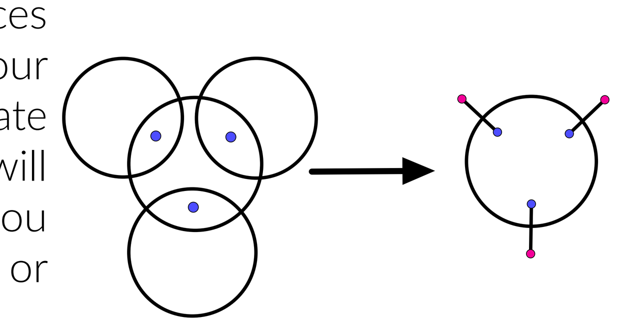
Egervary's Theorem: If G has no odd cycles, the weight of the **minimum vertex cover** is **equal** to the weight of the **maximum matching**.

Combining both theorems, we get that

$$(|B| - k) - \text{ maximum weight matching } = \text{ maximum weight independent set}$$

Our subroutine works in $O(2^k n^{k+5})$ time. We will reference it as $IndSet(B)$ later.

=

Observations	Diagram	Calls to $IndSet[B]$, for given B, in Improved Algorithm
If we knew which subsets of each intersection between bags were in a MWIS, we could find the MWIS for for the entire graph		$2^{\sum_{i,j \in V(T)} \lvert B_i \cap B_j \rvert}$
For any given bag, B, if we knew which subsets of its intersections were in a MWIS and we knew the MWIS of the resulting graph when removing B, we could find the MWIS of the entire graph G.		$2^{\sum_{i \in V(T)} \lvert B_i \cap B \rvert}$
For any given bag, B, if we knew which set S to use, we could reduce the intersections with any other bag B' to just 1 vertex.		$2^{\sum_{i \in V(T)} 1 \text{ if } \lvert B_i \cap B \rvert \neq \emptyset}$
Instead of guessing which of the single intersection vertices should be included in our independent set, we can create an auxilarly graph which will store the size of MWIS if you choose to keep that vertex or not.		1 (but bag is now larger)

Dynamic Programming Algorithm for MWIS

Given a graph $G = (V, E)$ and tree decomposition (T, B), we can start by rooting the tree.
Let us define $pa(i)$ = parent of $i \in V(T)$ and similarly $ch(i)$ = child of i. We will also let $P(i)$ be the set $B_i \cap B_P(i)$.
Table is going to be filled with entries dp$[i][x]$ for all $i \in V(T)$ and $x \subseteq P(i)$ where dp$[i][x]$ represents the maximum weight independent set where $x \subseteq P(i)$ is in the independent set and

An initial recurrence relation is guessing all possible intersection with B_i. It would look something like dp[i][x] = $\max_{x_c \subseteq P(c)\ \forall c \in ch(i)} \left(\sum_{c \in chi(i)} \text{dp}[c][x_c] + IndSet(B') \right)$
But unfortunately, going over all possible x_c is time consuming. Choosing S beforehand can reduce the cases because intersection between bags will be a single vertex. It might still not be polynomial because a bag can have up to n neighbors and doing test all 2^n possible ways to include them in the independent set is infeasible.
But, it also makes sent there would be quicker solution, since the graph is bipatite after removign S. Thus, we can use the augmented graph.

Acknowledgement

I am thankful for the support of everyone in the project group and especially for my mentor Dr. McCarty for her guidance throughout the summer. I gratefully acknowledge NSF funding in this REU.