MINIMUM WEIGHTED VERTEX COVER by Dynamic Programming on Tree Decompositions

Dominik Leko

February 2024

1 Problem description

Given a graph $G = (V, E)$, a tree decomposition $\mathcal{T} = (T, \chi)$ of G of width k and a function $w : V \to \mathbb{R}$, find $X \subseteq V$ such that X is a vertex cover of G and $\sum_{x \in X} w(x)$ is minimal.

2 Dynamic Programming description

The problem is solved via a dynamic programming approach on T. The algorithm has 3 steps:

1. For each $t \in T$ and each $U \subseteq \chi(t)$, initialize $M_t(U)$. $M_t(U)$ is the best (i.e. minimum weight) vertex cover VC found for subtree T_t where $\chi(t) \cap VC = U$.
2. Update $M_t(U)$ with dynamic programming.
3. Output $max\{n_r(U) | U \subseteq \chi(r)\}$.

3 Update nodes of nice Tree Decomposition via Dynamic Programming

Let t be the current node for which to update each $M_t(U)$ and t' and t'' its potential children.

1. Leaf Nodes. Let v be the single vertex in $\chi(t)$:

$$M_t(\{\}) = 0$$
$$M_t(\{v\}) = w(v)$$

2. Introduce Nodes. Let v be the newly introduced vertex, i.e. $\chi(t') \setminus \chi(t) = \{v\}$.

$$M_t(U) = \begin{cases} M_{t'}(U), & \text{if } v \notin U \\ M_{t'}(U) + w(v), & \text{if } v \in U \end{cases}$$

3. Forget Nodes. Let v be the newly forgotten vertex, i.e. $\chi(t) \setminus \chi(t) = \{v\}$.

$$M_t(U) = min(M_{t'}(U), M_{t'}(U \cup \{v\}))$$

4. Join Nodes.

$$M_t(U) = M_{t'} + M_{t''} - \sum_{v \in U} w(v)$$