MATH 8500 Algorithmic Graph Theory, Spring 2017, OSU
Lecture 18: Baker's Technique
Instructor: Anastasios Sidiropoulos
Scribe: Ryan Slechta

1 Baker's Technique

Baker's technique is a method for finding approximate solutions to problems in planar graphs. It can be used to create a *polynomial time approximation scheme*: given an optimization problem and some $\epsilon > 0$, can generate a solution that is within a $(1 + \epsilon)$ factor of being optimal. This is the best we can realistically hope for in the case of approximation algorithms for NP-hard problems.

While we focus on planar graphs, there are generalizations of Baker's technique. We demonstrate Baker's technique on the "Independent Set" problem.

2 Independent Set

Input: $G = (V, E)$
Goal: Find a maximal $S \subseteq V$ such that $\forall u \neq v \in S$, $\{u, v\} \notin E$.

Essentially, we look for the largest set of vertices that are pairwise nonadjacent.

Theorem 1. *Independent Set is NP-hard, even in the case of planar graphs.*

In general, there does not exist a constant factor approximation. In the case of planar graphs, we will use Baker's technique to get an approximation arbitrarily close to an optimal solution. The algorithm developed by Baker is as follows.

Algorithm 1 Baker's Algorithm for Independent Set

1: **procedure** IndependentSet($G = (V, E), \epsilon$)
2: Pick some arbitrary vertex; denote it r. ▷ r is the "root"
3: $k \leftarrow \lceil 1/\epsilon \rceil$
4: $\forall i \in \{0, \ldots, k-1\}$, let $V_i = \{v \in V \mid d(r, v) \pmod k = 1\}$
5: $G^i \leftarrow G \setminus V_i$
6: Let $G^i_1, G^i_2, \ldots$ denote the connected components of G^i
7: For all j, i, compute a maximal Independent Set S^i_j of G^i_j
8: $S_i \leftarrow \cup_j S^i_j$
9: $i^* \leftarrow \text{argmax}\{|S_i|\}$
10: **return** S_{i^*}
11: **end procedure**

An example of step 4 for an arbitrary graph and $i = 2$ is given in Figure **??**. Essentially, the algorithm works by partitioning a graph into "thin sausages." These are marked by the black ovals in Figure **??**. We then solve the Independent Set problem on each of these ovals, and use them to construct an Independent Set for the entire graph.

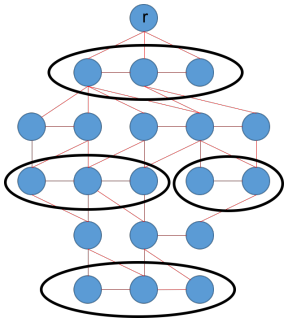

Figure 1: Baker's Algorithm for independent set, step 4, $i = 2$

Note in particular that in step 7, this algorithm requires a means to compute an independent set for some subsets of G. Our ability to do this comes from Lemma **??**, which we discuss after the following two results.

Theorem 2. *Algorithm* **??** *outputs an Independent Set.*

Proof. Note that S_{i^*} is defined to be the union of independent sets, each of which is defined on some connected component of $G \setminus V_{i^*}$. If S_{i^*} were not an independent set, then there must be some pair of vertices $v_i, v_j \in S_{i^*}$ in distinct connected components of $G \setminus V_{i^*}$, where $\{v_i, v_j\} \in G \setminus V_{i^*}$. But if there exists such an edge, then v_i, v_j must exist in the same connected component. Hence, there is no such edge, and S_{i^*} is an independent set. □

Lemma 1. $|S_{i^*}| \geq (1-\epsilon) \cdot OPT$, *where OPT is the cardinality of some maximal independent set in G.*

Proof. Fix some optimal solution X. Let $t = \text{argmin}_{i \in \{0,1,\ldots k-1\}} |X \cap V_i|$. Then $|S_{i^*}| \geq |S_t| = \Sigma_j |S_j^t|$. S_t is, by construction, a maximal Independent Set of $G \setminus V_t$. Ergo, $\Sigma_j |S_j^t| \geq |X \setminus V_t| \geq |X|(1 - 1/k)$, because V_i and V_j are pairwise disjoint for all $i \neq j$ and S_t intersects X less than all other S_i. By the definition of k, it follows that $|S_{i^*}| \geq |X|(1 - \epsilon)$. □

All that remains to be shown is a means to compute an independent set on the various connected components. We will utilize the following result:

Lemma 2. *There exists an algorithm for Independent Set on graphs of treewidth k with running time* $2^{O(k)} n^{O(1)}$.

Proof. This result uses dynamic programming, and a similar idea has been discussed in class for 3-coloring. □

All that remains to be done is to to show that the various connected components all have bounded tree width. To do this, we develop some machinery.

Definition 1. *An embedding of a graph G is "1-outerplanar" or "outerplanar" if it is planar, and all vertices lie on the outer face.*

Definition 2. *For* $k \geq 2$, *an embedding is "k-outerplanar" if it is planar and when all vertices on the outer face are deleted, then a* $(k-1)$*-outerplanar embedding is obtained.*

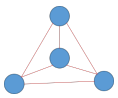

Figure 2: An embedding of a graph that is not outerplanar.

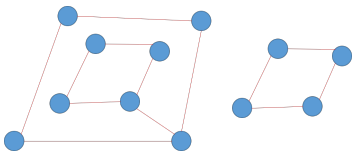

Figure 3: The left embedding is 2-outerplanar, because upon deleting all vertices on the outer face, we are left with the right embedding, which is 1-outerplanar.

Definition 3. *A graph G is k-outerplanar if it admits a k-outerplanar embedding.*

Lemma 3. *$\forall i, j$, G^i_j is $O(k)$-outerplanar.*

Once Lemma 3 is shown, it remains to be shown that $O(k)$-outerplanar graphs are of bounded treewidth, which will be the topic of the remaining lectures.