Approximation Algorithms for Independent Sets in Map Graphs

Zhi-Zhong Chen
Department of Mathematical Sciences, Tokyo Denki University
Hatoyama, Saitama 350-0394, Japan. Email: chen@r.dendai.ac.jp

Abstract

This paper presents polynomial-time approximation algorithms for the problem of computing a maximum independent set in a given map graph G with or without weights on its vertices. If G is given together with a map, then a ratio of $1+\delta$ can be achieved for any given constant $\delta > 0$, no matter whether each vertex of G is given a weight or not. In case G is given without a map, a ratio of 4 can be achieved if no vertex is given a weight, while a ratio of $O(\log n)$ can be achieved otherwise. Behind the design of our algorithms are several fundamental results for map graphs; these results can be used to design good approximation algorithms for coloring and vertex cover in map graphs, and may find applications to other problems on map graphs as well.

1 Introduction

An *independent set* in a graph $G = (V, E)$ is a subset I of V such that no two vertices of I are connected by an edge in E. If each vertex of G is associated with a weight, the *weight* of an independent set I in G is the total weight of vertices in I. Given a vertex-weighted graph $G = (V, E)$, the *maximum independent set* (MIS) problem requires the computation of an independent set of maximum weight in G. The special case where each vertex of G has weight 1, is called the *unweighted maximum independent set* (UMIS) problem. The MIS problem is a fundamental problem in many areas; unfortunately, even the UMIS problem is widely known to be NP-hard.

Since the MIS problem is NP-hard, we are interested in designing *approximation algorithms*, i.e. polynomial-time algorithms that find an independent set of large but not necessarily maximum weight in a given graph. An approximation algorithm $\mathcal{A}$ *achieves a ratio* of ρ if for every vertex-weighted graph G, the independent set I found by $\mathcal{A}$ on input G has weight at least $\frac{OPT(G)}{\rho}$, where $OPT(G)$ is the maximum weight of an independent set in G.

Håstad [8] gave strong evidence that no approximation algorithm for the UMIS problem achieves a ratio of $n^{1-\epsilon}$ for any $\epsilon > 0$, where n is the number of vertices in the input graph. Due to this, a lot of work has been devoted to designing approximation algorithms for the MIS problem restricted to certain classes of graphs (such as planar graphs, bounded-degree graphs, etc). This paper considers the MIS problem restricted to *map graphs*, which were introduced by Chen et al. [7] recently.

Intuitively speaking, a *map* is a plane graph whose faces are classified into *nations* and *lakes*. Traditional planarity says that two nations on a map $\mathcal{M}$ are adjacent if and only if they share at least a borderline. Motivated by topological inference, Chen et al. [7] considered a modification of traditional planarity, which says that two nations on $\mathcal{M}$ are adjacent if and only if they share at least a border-point. Consider the simple graph G whose vertices are the nations on $\mathcal{M}$ and whose edges are all $\{f_1, f_2\}$ such that f_1 and f_2 are nations sharing at least a border-point. G is

called a *map graph* [7, 16]; if for some integer k, no point in $\mathcal{M}$ is shared by more than k nations, G is called a *k-map graph* [7]. The UMIS problem restricted to map graphs is NP-hard, because planar graphs are exactly 3-map graphs [7] and the UMIS problem restricted to planar graphs is NP-hard.

It is known [1, 9] that the MIS problem restricted to planar graphs has a *polynomial-time approximation scheme* (PTAS). Recall that a PTAS for a maximization (respectively, minimization) problem Π is a polynomial-time algorithm $\mathcal{A}$ such that given an instance I of Π and an error parameter $\epsilon > 0$, $\mathcal{A}$ computes a solution S of I in time polynomial in the size of I such that the value of S is at least $(1-\epsilon) \cdot OPT(I)$ (respectively, at most $(1+\epsilon) \cdot OPT(I)$), where $OPT(I)$ is the optimal value of a solution of I. An interesting question is to ask whether the MIS problem restricted to map graphs also has a PTAS. This paper partially answers this question in the affirmative, by giving a PTAS for the MIS problem restricted to k-map graphs for any fixed integer k. Also, if we require that the input graph be given together with a map, then the MIS problem restricted to map graphs has a PTAS. It is worth noting that although Thorup [16] presented a polynomial-time algorithm for constructing a map for a given map graph, his algorithm is extremely difficult to understand and the exponent of its polynomial time bound is about 120.

In the case where the input graph is not given with a map, we are unable to design a PTAS for the MIS problem restricted to map graphs. However, we are able to design two nontrivial approximation algorithms for it. One of them achieves a ratio of $O(\log n)$. The other is for the UMIS problem and achieves a ratio of 4.

Behind the design of our algorithms are several fundamental results for map graphs; these results can be used to obtain the following results. First, there is a PTAS for the minimum weighted vertex cover (MWVC) problem restricted to map graphs. Second, a map graph G with n vertices and m edges can be colored in $O(n^2 m)$ time using at most $2 \cdot \chi(G)$ colors, where $\chi(G)$ is the chromatic number of G. Third, for all optimization problems Π such that Π restricted to planar graphs has a PTAS as shown in [1], Π restricted to k-map graphs has a PTAS as well for any fixed integer k. These results may find applications to other problems on map graphs as well.

The rest of this paper is organized as follows. Section 2 gives precise definitions and states known results that are needed later on. Section 3 presents a PTAS for the MIS problem where the input graph is given together with a map. Section 4 describes two PTASs for the MIS problem restricted to k-map graphs. Section 5 presents nontrivial approximation algorithms for the MIS and UMIS problems restricted to map graphs. Section 6 asks several open questions and sketches the PTAS for the MWVC problem restricted to map graphs.

Attention: All omitted proofs may be found in the appendix.

2 Preliminaries

Throughout this paper, a graph may have multiple edges but no loops, while a *simple* graph has neither multiple edges nor loops. Let $G = (V, E)$ be a graph. The *size* of G is the total number of vertices and edges in G, and is denoted by $|G|$. An *induced subgraph* of G is a subgraph H whose edges are those of G between vertices of H. An induced subgraph of G with vertex-set U is denoted by $G[U]$, and called the *subgraph of G induced by U*. An *independent set* of G is a subset U of V such that $G[U]$ has no edge. The *independence number* of G, denoted by $\alpha(G)$, is the maximum number of vertices in an independent set of G. Let $v \in V$. The *degree* of v in G is the number of edges incident to v in G; v is an *isolated* vertex if its degree is 0. The

neighborhood of v in G, denoted by $N_G(v)$, is the set of vertices adjacent to v in G. Note that $|N_G(v)|$ bounds the degree of v in G from below. For a subset U of V, let $N_G(U) = \cup_{v \in U} N_G(v)$.

A *simple cycle* in G is a sequence of distinct vertices and edges $v_1, e_1, v_2, e_2, \ldots, v_k, e_k$ with $k \geq 2$ such that for all $i \in \{1, \ldots, k\}$, e_i is an edge between v_i and v_{i+1} where $v_{k+1} = v_1$. G is *biconnected* if (1) G has at least two edges and (2) for every pair of distinct edges e_1 and e_2 in G, G has a simple cycle on which both e_1 and e_2 appear. A *biconnected component* of a graph is a maximal biconnected subgraph.

A graph is *planar* if it can be embedded in the plane so that any pair of edges can only intersect at their endpoints; a *plane* graph is a planar one together with such an embedding. Let $\mathcal{E}$ be a plane graph. Consider the set of all points in the plane that lie on no edge of $\mathcal{E}$. This set consists of a finite number of topologically connected regions; the closure of each such region is called a *face* of $\mathcal{E}$. Let f be a face of $\mathcal{E}$. The *boundary* of f is the graph (V_f, E_f), where V_f and E_f are the set of vertices and edges surrounding f, respectively. The *size* of f, denoted by $|f|$, is $|V_f|$. We call f a *cycle face* if its boundary is a simple cycle.

A *map* $\mathcal{M}$ is a pair $(\mathcal{E}, \varphi)$, where (1) $\mathcal{E}$ is a plane graph $(\mathcal{P}, E_{\mathcal{E}})$ and each connected component of $\mathcal{E}$ is biconnected, and (2) φ is a function that maps each inner face f of $\mathcal{E}$ to 0 or 1 in such a way that whenever $\varphi(f) = 1$, f is a cycle face. A face f of $\mathcal{E}$ is called a *nation* on $\mathcal{M}$ if f is an inner face with $\varphi(f) = 1$, while called a *lake* on $\mathcal{M}$ otherwise. For an integer k, a *k-point* in $\mathcal{M}$ is a $p \in \mathcal{P}$ that appears on the boundaries of exactly k nations on $\mathcal{M}$; a *k^+-point* in $\mathcal{M}$ is a $p \in \mathcal{P}$ that appears on the boundaries of more than k nations on $\mathcal{M}$. $\mathcal{M}$ is a *k-map* if there is no k^+-point in $\mathcal{M}$. Two nations are *adjacent* if their boundaries share a $p \in \mathcal{P}$. The *graph* of $\mathcal{M}$ is the simple graph $G = (V, E_G)$ where V consists of the nations on $\mathcal{M}$ and E_G consists of all $\{f_1, f_2\}$ such that f_1 and f_2 are adjacent. We call G a *map graph*, while call G a *k-map graph* if $\mathcal{M}$ is a k-map. For clarity, we call the elements of $\mathcal{P}$ *points*, while call the elements of V *vertices*. A nation f on $\mathcal{M}$ is *isolated* if f is an isolated vertex of G.

The *half square* of a bipartite graph $H = (X, Y; E_H)$ is the simple graph $G = (X, E)$ where $E = \{\{x_1, x_2\} \mid \text{there is a } y \in Y \text{ with } \{x_1, y\} \in E_H \text{ and } \{x_2, y\} \in E_H\}$.

Lemma 1 [7] *Every map graph $G = (V, E)$ is the half square of a bipartite planar graph $H = (V, \mathcal{P}; E_H)$ with $|\mathcal{P}| \leq 3|V| - 6$.*

For a map $\mathcal{M} = (\mathcal{E}, \varphi)$ and a face f in $\mathcal{E}$, we say that f *touches* a point p if p appears on the boundary of f.

Lemma 2 [7] *Let G be the graph of a map $\mathcal{M} = (\mathcal{E}, \varphi)$. Let C be a clique of G containing at least five vertices. Then, at least one of the following holds:*

- *C is a* pizza, *i.e., there is a point p in $\mathcal{E}$ such that all $f \in C$ touch p.*
- *C is a* pizza-with-crust, *i.e., there are a point p in $\mathcal{E}$ and an $f_1 \in C$ such that (1) all $f \in C - \{f_1\}$ touch p and (2) for each $f \in C - \{f_1\}$, some point $q \neq p$ in $\mathcal{E}$ is touched by both f_1 and f.*
- *C is a* hamantasch, *i.e., there are three points in $\mathcal{E}$ such that each $f \in C$ touches at least two of the points and each point is touched by at least two nations.*

3 A PTAS for the case with a map

Let $G = (V, E)$ be the input map graph given with a map $\mathcal{M} = ((\mathcal{P}, E_{\mathcal{E}}), \varphi)$ and a nonnegative weight to each nation on $\mathcal{M}$. Let ϵ be the given error parameter.

Let H be the bipartite planar graph $(V, \mathcal{P}; E_H)$ where $E_H = \{\{v, p\} \mid v \in V, p \in \mathcal{P}$, and p is a point on the boundary of nation $v\}$. Obviously, H can be constructed in linear time from $\mathcal{M}$. Moreover, G is the half square of H. Indeed, Lemma 1 is proved by further removing all *redundant points* from H where a point $p \in \mathcal{P}$ is redundant if for every pair $\{u, v\}$ of neighbors of p in H, there is a point $q \in \mathcal{P} - \{p\}$ with $\{u, v\} \subseteq N_H(q)$.

So, one naive method for computing an independent set of large weight in G is to emulate Lipton and Tarjan [10]. That is, we first remove redundant points from H to make H small, then find a small separator S of H, and further compute an independent set I_C of G in each connected component C of $G - S$. The output is $\cup_C I_C$. However, $\cup_C I_C$ may not be an independent set of G, implying that this naive method does not work.

Another approach to the design of PTASs for a certain class of NP-hard problems restricted to planar graphs was suggested by Baker [1]. The essence of this approach is the notion of *k-outerplanar graphs* defined inductively as follows. *1-outerplanar graphs* are outerplanar graphs, i.e., those planar graphs having a plane embedding such that all vertices lie on the outer face. For positive integers $k \geq 2$, a planar graph is *k-outerplanar* if it has a plane embedding such that the removal of all vertices on the outer face and all edges incident to them leaves a $(k-1)$-outerplanar graph.

Our PTAS is a nontrivial application of the Baker approach. Let $h = \lceil \frac{1}{\epsilon} \rceil$. We may assume that G and hence H are connected. Starting at an arbitrary vertex $r \in V$, we perform a breadth-first search (BFS) on H to obtain a BFS tree T_H. For each vertex v in H, we define the level number $\ell(v)$ of v to be the number of edges on the path from r to v in T_H. Note that only r has level number 0. For each $i \in \{0, \ldots, h-1\}$, let H_i be the subgraph of H obtained from H by deleting all vertices v with $\ell(v) \equiv 2i \pmod{2h}$. Obviously, each $v \in V$ appears in exactly $h-1$ of H_0 through H_{h-1}. Moreover, H_i is clearly a $(2h-1)$-outerplanar graph.

For each $i \in \{0, \ldots, h-1\}$, let G_i be the half square of H_i. The crux is that each G_i is an induced subgraph of G, because all $p \in \mathcal{P}$ appear in H_i. Our strategy is to compute an independent set I_i of maximum weight in G_i for each $i \in \{0, \ldots, h-1\}$. Since each vertex of G appears in all but one of G_0 through G_{h-1}, the independent set of maximum weight among $I_0, \ldots, I_{h-1}$ has weight at least $(1 - \frac{1}{h}) \cdot OPT(G) \geq (1 - \epsilon) \cdot OPT(G)$. Recall that $OPT(G)$ is the maximum weight of an independent set in G.

So, it remains to show how to compute a maximum weight independent set I_i in G_i for each $i \in \{0, \ldots, h-1\}$. To do this, we first recall the notion of *treewidth*, introduced by Robertson and Seymour [14]. A *tree-decomposition* of a graph $K = (V_K, E_K)$ is a pair $(\{X_j \mid j \in J\}, \mathcal{T})$, where $\{X_j \mid j \in J\}$ is a family of subsets of V_K and $\mathcal{T}$ is a tree with vertex-set J such that the following hold:

- $\cup_{j \in J} X_j = V_K$.
- For every edge $\{v, w\} \in E_K$, there is a $j \in J$ with $\{v, w\} \subseteq X_j$.
- For all $j_1, j_2, j_3 \in J$, if j_2 lies on the path from j_1 to j_3 in $\mathcal{T}$, then $X_{j_1} \cap X_{j_3} \subseteq X_{j_2}$.

Each X_j is called the *bag* of vertex j. The *treewidth* of a tree-decomposition $(\{X_j \mid j \in J\}, \mathcal{T})$ is $\max\{|X_j| - 1 \mid j \in J\}$. The *treewidth* of K is the minimum treewidth of a tree-decomposition of K, taken over all possible tree-decompositions of K.

It is widely known that the MIS problem restricted to graphs with bounded treewidth can be solved in linear time. However, the treewidths of our graphs $G_0, \ldots, G_{h-1}$ may not be bounded. Our strategy is to resort to $H_0, \ldots, H_{h-1}$ instead, and do a dynamic programming on them as shown below.

Lemma 3 *For every $i \in \{0, \ldots, h-1\}$, we can compute a maximum weight independent set I_i of G_i in $O(|H_i|^2 + 2^{12h-8}h|H_i|)$ time.*

By Lemma 3, we have:

Theorem 4 *There is a PTAS for the special case of the MIS problem restricted to map graphs where the input graph is given together with a map $\mathcal{M} = (\mathcal{E}, \varphi)$. It runs in $O(h|\mathcal{E}|^2 + 2^{12h-8}h^2|\mathcal{E}|)$ time, where $h = \lceil \frac{1}{\epsilon} \rceil$ and ϵ is the given error parameter.*

4 PTASs for k-map graphs

We use an extension of the Baker approach, suggested in [6]. The only difference between the extension and the Baker approach is that instead of decomposing the given graph G into k'-outerplanar subgraphs, we decompose G into subgraphs of bounded treewidth such that each vertex of G appears in all but one of these subgraphs. The next lemma shows that such a decomposition is possible for a k-map graph.

Lemma 5 *Given a k-map graph G and an integer $h \geq 2$, it takes $O(h|G|)$ time to compute a list $\mathcal{L}$ of h induced subgraphs of G such that (1) each subgraph in $\mathcal{L}$ is of treewidth $\leq k(6h-4)$ and (2) each vertex of G appears in exactly $h-1$ of the subgraphs in $\mathcal{L}$.*

By the proof of Lemma 5, for all optimization problems Π such that Π restricted to planar graphs has a PTAS as shown in [1], we can emulate Baker [1] to show that Π restricted to k-map graphs has a PTAS as well for any fixed integer k. In particular, we have:

Theorem 6 *For any fixed integer k, there is a PTAS for the MIS problem restricted to k-map graphs. It runs in $O(2^{32(6kh-4k)^3}hn)$ time, where n is the size of the given k-map graph, $h = \lceil \frac{1}{\epsilon} \rceil$, and ϵ is the given error parameter.*

The PTAS in Theorem 6 is not practical. We next show a different PTAS for the special case of the UMIS problem where the input graph is given together with a k-map. This PTAS is based on the Lipton and Tarjan approach. Although this PTAS is not better than the one in Section 3, the tools used in its design seem to be fundamental. Indeed, the tools will be used in the next section, and we believe that they can be used in other applications.

A map $\mathcal{M} = (\mathcal{E}, \varphi)$ is *well-formed* if (1) the degree of each point in $\mathcal{E}$ is at least 3, (2) no edge of $\mathcal{E}$ is shared by two lakes, (3) no lake touches a point whose degree in $\mathcal{E}$ is 4 or more, and (4) the boundary of each lake contains at least 4 edges.

Lemma 7 *Given a map $\mathcal{M} = (\mathcal{E}, \varphi)$ with at least three nations and without isolated nations, it takes $O(|\mathcal{E}|)$ time to obtain a well-formed map $\mathcal{M}'$ such that (1) the graph of $\mathcal{M}'$ is the same as that of $\mathcal{M}$ and (2) for every integer $k \geq 2$, $\mathcal{M}$ is a k-map if and only if so is $\mathcal{M}'$.*

Let $\mathcal{M} = (\mathcal{E}, \varphi)$ be a well-formed map. Then, each point on the boundary of a lake on $\mathcal{M}$ is a 2-point in $\mathcal{M}$ and has degree 3 in $\mathcal{E}$; moreover, for all integers $i \geq 3$, each i-point in $\mathcal{M}$ has degree i in $\mathcal{E}$.

Lemma 8 *Let $k \geq 3$ and G be a k-map graph with $n \geq 3$ vertices and m edges. Then, $m \leq kn - 2k$.*

It is known that k-map graphs are $2k$-colorable [13], but the following corollary has the advantage of being algorithmic.

Corollary 9 *Given an integer k and a k-map graph G, it takes $O(|G|)$ time to color G with at most $2k$ colors. Consequently, given a k-map $\mathcal{M} = (\mathcal{E}, \varphi)$, it takes $O(k|\mathcal{E}|)$ time to color the graph of $\mathcal{M}$ with at most $2k$ colors.*

It is known that k-map graphs with $k \geq 8$ can be colored with at most $2k - 3$ colors [5], but the proof does not yield an efficient algorithm.

An important consequence of Corollary 9 is the following:

Corollary 10 *Given a map graph $G = (V, E)$, we can color G in $O(|V|^2 \cdot |E|)$ time using at most $2 \cdot \chi(G)$ colors, where $\chi(G)$ is the chromatic number of G.*

Proof: According to [7], G has only $O(|V|)$ maximal cliques. Thus, the maximal cliques of G can be enumerated in $O(|V|^2 \cdot |E|)$ time [12]. Consequently, the size k of the maximum clique(s) of G can be computed within the same time bound. Note that $\chi(G) \geq k$. On the other hand, G must be a k-map graph. Now, the corollary follows from Corollary 9. □

Let G be a connected graph with nonnegative weights on its vertices. The *weight* of a subgraph of G is the total weight of vertices in the subgraph. A *separator* of G is a subgraph H of G such that deleting the vertices of H from G leaves a graph whose connected components each have weight $\leq \frac{2W}{3}$, where W is the weight of G. The *size* of a separator H is the number of vertices in H.

Lemma 11 *Given a k-map $\mathcal{M} = (\mathcal{E}, \varphi)$ on which each nation has an associated nonnegative weight and at least two nations exist, it takes $O(|\mathcal{E}|)$ time to find a separator H of the graph G of $\mathcal{M}$ such that the size of H is at most $2\sqrt{2k(n-1)}$.*

Using Lemma 11, we can emulate Lipton and Tarjan [10] to prove the following theorem:

Theorem 12 *For any fixed integer k, there is a PTAS for the special case of the UMIS problem restricted to k-map graphs where the input graph is given together with a k-map $\mathcal{M} = (\mathcal{E}, \varphi)$. It runs in $O(|\mathcal{E}| \log |\mathcal{E}| + 2^{288h^2k^3} n)$ time, where n is the number of nations on $\mathcal{M}$, $h = \lceil \frac{1}{\epsilon} \rceil$, and ϵ is the given error parameter.*

5 Approximation algorithms for map graphs

This section presents four approximation algorithms. The first three are for the UMIS problem restricted to map graphs. The fourth is for the MIS problem restricted to map graphs.

For a well-formed map $\mathcal{M} = (\mathcal{E}, \varphi)$, let $\mathcal{G}(\mathcal{M})$ be the plane graph obtained from the dual plane graph $\mathcal{E}^*$ of $\mathcal{E}$ by deleting all vertices that are lakes on $\mathcal{M}$. Note that corresponding to each 2^+-point p in $\mathcal{E}$, there is a unique face $\mathcal{F}_p$ in $\mathcal{G}(\mathcal{M})$ such that the vertices on the boundary of $\mathcal{F}_p$ are exactly the nations on $\mathcal{M}$ that touch p; the boundary of $\mathcal{F}_p$ is a simple cycle. We call $\mathcal{F}_p$ a *point-face* of $\mathcal{G}(\mathcal{M})$. Those faces of $\mathcal{G}(\mathcal{M})$ that are not point-faces are called *lake-faces*.

Consider the following algorithm, called $MapIS_1$:

Input : A map $\mathcal{M} = (\mathcal{E}, \varphi)$.

1. If at most two nations are on $\mathcal{M}$, then output a maximum weight independent set of the graph of $\mathcal{M}$ and halt.

2. If there is an isolated nation on $\mathcal{M}$, then find such a nation f; otherwise, perform the following substeps:

2.1. Modify $\mathcal{M}$ to a well-formed map, without altering the graph of $\mathcal{M}$.

2.2. Construct $\mathcal{G}(\mathcal{M})$.

2.3. Find a minimum-degree vertex f in $\mathcal{G}(\mathcal{M})$.

3. Construct a new map $\mathcal{M}' = (\mathcal{E}, \varphi')$, where for all faces f_1 of $\mathcal{E}$, if $f_1 = f$ or f_1 is a neighbor of f in the graph of $\mathcal{M}$, then $\varphi'(f_1) = 0$, while $\varphi'(f_1) = \varphi(f_1)$ otherwise.

4. Recursively call $MapIS_1$ on input $\mathcal{M}'$, to compute an independent set J in the graph of $\mathcal{M}'$.

5. Output $\{f\} \cup J$ and halt.

Theorem 13 *$MapIS_1$ achieves a ratio of 5 and runs in $O(|\mathcal{E}|^2)$ time.*

Proof: The proof is by induction on n, the number of nations on the input map $\mathcal{M}$. The case where $n \leq 2$ is trivial. Consider the case where $n \geq 3$. Let G be the graph of $\mathcal{M}$. Let f be the nation found in Step 2. If f is an isolated nation, then we are done. So, assume that f is found in Substep 2.3. Since $\mathcal{M}$ is modified to a well-formed map in Substep 2.1, $\mathcal{G}(\mathcal{M})$ is a plane graph in which the boundary of each face contains at least three edges. Thus, the number of edges in $\mathcal{G}(\mathcal{M})$ is at most $3n - 6$. So, the degree of f in $\mathcal{G}(\mathcal{M})$ is at most 5.

We claim that the neighborhood of f in G can be divided into at most five cliques. To see this, let $\mathcal{F}_1, \ldots, \mathcal{F}_\ell$ be the faces in $\mathcal{G}(\mathcal{M})$ whose boundaries each contain vertex f. Then, $\ell \leq 5$. For each $i \in \{1, \ldots, \ell\}$, if $\mathcal{F}_i$ is a point-face, then the vertices on $\mathcal{F}_i$ form a clique C_i of G. For each neighbor f_j of f in G that is in no C_i, $\{f, f_j\}$ must be an edge in $\mathcal{G}(\mathcal{M})$, and f_j forms a singleton clique of G. The total number of such cliques as C_i and f_j is bounded from above by ℓ. This establishes the claim. By the claim, each independent set of G contains at most five vertices among f and its neighbors in G. This fact and the inductive hypothesis imply the theorem. □

Consider the following algorithm, called $MapIS_2$:

Input : A map graph $G = (V, E)$.

1. If G has exactly at most two vertices, then output a maximum weight independent set of G and halt.

2. Find a vertex f in G such that $\alpha(G_f) \leq 5$, where $G_f = G[N_G(f)]$.

3. Remove f and its neighbors from G.

4. Recursively call $MapIS_2$ on input G, to compute an independent set J.

5. Output $\{f\} \cup J$ and halt.

Theorem 14 *$MapIS_2$ achieves a ratio of 5, and runs in $O(|V| \cdot \sum_{f \in V} |N_G(f)|^6)$ time.*

Proof: By the proof of Theorem 13, Step 2 of $MapIS_2$ can always be done. Moreover, it can be done in $O(\sum_{f \in V} |N_G(f)|^6)$ time. □

Consider the following algorithm, called $MapIS_3$:

Input : A map graph $G = (V, E)$, an integer $k \geq 4$, and a positive real number ϵ less than 1.

1. If G has exactly at most two vertices, then output a maximum weight independent set of G and halt.

2. Find a vertex f in G such that $\alpha(G_f)$ is minimized among all vertices f of G, where $G_f = G[N_G(f)]$.

3. If $\alpha(G_f) \leq 4$, then perform the following substeps:

3.1. Remove f and its neighbors from G.

3.2. Recursively call $MapIS_3$ on input G to compute an independent set J.

3.3. Output $\{f\} \cup J$ and halt.

4. Repeat the following substeps until G has no clique of size $\geq k+1$ (and hence is a k-map graph):

4.1. Find a maximal clique C in G such that C contains at least $k+1$ vertices.

4.2. Remove all vertices of C from G.

5. Color G with at most $2k$ colors (cf. Corollary 9); let J_1 be the maximum set of vertices with the same color.

6. Compute an independent set J_2 in G with $|J_2| \geq (1-\epsilon)\alpha(G)$.

7. Output the bigger one among J_1 and J_2, and halt.

Theorem 15 *If we fix $k = 28$ and $\epsilon = 0.25$, then $MapIS_3$ achieves a ratio of 4 and runs in $O(|V| \cdot \sum_{f \in V} |N_G(f)|^6 + |V|^3 \cdot |E|)$ time.*

Proof: Let G_0 be the input graph. We may assume that G_0 is connected. By the proof of Theorem 14, it suffices to prove that if $\alpha(G_f) \geq 5$, then J_1 found in Step 5 or J_2 found in Step 6 has size $\geq \frac{\alpha(G_0)}{4}$. So, suppose that $\alpha(G_f) \geq 5$.

Let n be the number of vertices in G_0. Fix a well-formed map $\mathcal{M}$ of G_0. For each integer $i > k$, let h_i be the number of cliques C in G_0 such that (1) C contains exactly i vertices and (2) C is found in Substep 4.1 by the algorithm. We claim that the number x of edges in $\mathcal{G}(\mathcal{M})$ is bounded from above by $3n - 6 - \sum_{i=k+1}^{\infty} h_i(i-4)$. To see this claim, let $\mathcal{F}_1, \ldots, \mathcal{F}_t$ be the point-faces in $\mathcal{G}(\mathcal{M})$ whose boundaries each contain at least three vertices. For each $j \in \{1, \ldots, t\}$, fix a vertex f_j on the boundary of $\mathcal{F}_j$. We can triangulate each face $\mathcal{F}_j$ by adding exactly $|\mathcal{F}_j| - 3$ new edges incident to f_j; call these new edges *imaginary edges* incident to f_j. There are exactly $|\mathcal{F}_j| - 3$ imaginary edges incident to f_j. Thus, $x \leq 3n - 6 - \sum_{j=1}^{t}(|\mathcal{F}_j| - 3)$. Let C be a clique of G_0 found in Substep 4.1. We say that an imaginary edge $\{f_j, f'\}$ with $j \in \{1, \ldots, t\}$ is *contributed* by C, if f' is contained in C. Let i be the number of vertices in C. To prove the claim, it remains to prove that the number of imaginary edges contributed by C is at least $i - 4$. By Lemma 2, only the following three cases can occur:

- *Case 1:* C is a pizza. Then, there is a point p in $\mathcal{M}$ such that all nations f' in C touch p on $\mathcal{M}$. Let $\mathcal{F}_j$ be the point-face in $\mathcal{G}(\mathcal{M})$ that corresponds to p. No matter whether f_j is in C or not, at least $i - 3$ imaginary edges incident to f_j are contributed by C.

- *Case 2:* C is a pizza-with-crust. Then, there is a point p in $\mathcal{M}$ such that $|C| - 1$ nations in C touch p on $\mathcal{M}$. Similarly to Case 1, at least $(i-1) - 3$ imaginary edges are contributed by C.

- *Case 3:* C is a hamantasch. Then, there are three points p_1 through p_3 in $\mathcal{M}$ such that each nation $f' \in C$ touches at least two of the three points on $\mathcal{M}$ and each point is touched by at least two nations in C. Since $i \geq k+1 \geq 5$, at most one of p_1 through p_3 is a 2-point in $\mathcal{M}$. Moreover, if one of p_1 through p_3 is a 2-point, then C is actually a pizza-with-crust and Case 2 applies. So, we may assume that p_1 through p_3 are 2^+-points in $\mathcal{M}$. Let $\mathcal{F}_{j_1}$ be the point-face in $\mathcal{G}(\mathcal{M})$ that corresponds to p_1, and a_1 be the number of nations in C that touch p_1 on $\mathcal{M}$. Define $\mathcal{F}_{j_2}$, a_2, $\mathcal{F}_{j_3}$, and a_3 similarly. Then, the number of imaginary edges incident to f_{j_1} (respectively, f_{j_2}, or f_{j_3}) contributed by C is at least $a_1 - 3$ (respectively, $a_2 - 3$, or $a_3 - 3$). Thus, the number of imaginary edges contributed by C is at least $(a_1 + a_2 + a_3) - 9 \geq 2i - 9 \geq i - 4$.

By the claim, $x \leq 3n - 6 - \sum_{i=k+1}^{\infty} h_i(i-4)$. The degree of f in $\mathcal{G}(\mathcal{M})$ is at most $\lfloor \frac{2x}{n} \rfloor$. On the other hand, since $\alpha(G_f) \geq 5$, the degree of f in $\mathcal{G}(\mathcal{M})$ is at least 5. Thus, $2\sum_{i=k+1}^{\infty} h_i(i-4) \leq n$. In turn, $\sum_{i=k+1}^{\infty} h_i i \leq \frac{n(k+1)}{2(k-3)}$ and $\sum_{i=k+1}^{\infty} h_i \leq \frac{n}{2(k-3)}$.

Let J be the bigger one among J_1 and J_2. By Step 6, $|J| \geq (1-\epsilon)(\alpha(G_0) - \sum_{i=k+1}^{\infty} h_i)$. By Step 5 and Corollary 9, $|J| \geq \frac{1}{2k} \cdot (n - \sum_{i=k+1}^{\infty} i h_i)$. Thus, $\frac{|J|}{\alpha(G_0)} \geq (1-\epsilon)(1 - \frac{1}{2(k-3)} \cdot \frac{n}{\alpha(G_0)})$ and $\frac{|J|}{\alpha(G_0)} \geq \frac{k-7}{4k(k-3)} \cdot \frac{n}{\alpha(G_0)}$. As the value of $\frac{n}{\alpha(G_0)}$ decreases, the first lower bound on $\frac{|J|}{\alpha(G_0)}$ increases while the second lower bound on $\frac{|J|}{\alpha(G_0)}$ decreases. The worst case happens when $\frac{n}{\alpha(G_0)} = \frac{4k(k-3)(1-\epsilon)}{3k-2k\epsilon-7}$. Therefore, $\frac{|J|}{\alpha(G_0)} \geq (1-\epsilon) \cdot \frac{k-7}{3k-2k\epsilon-7}$. By fixing $\epsilon = \frac{1}{4}$ and $k = 28$, we have $\frac{|J|}{\alpha(G_0)} \geq \frac{1}{4}$.

By Theorem 14, the total time needed by Steps 1 through 3 is $O(|V| \cdot \sum_{f \in V} |N_G(f)|^6)$. G has only $O(|V|)$ maximal cliques [7], and hence the maximal cliques can be enumerated in $O(|V|^2 \cdot |E|)$ time [12]. By Corollary 9 and Theorem 6, Steps 6 and 7 take $O(|G|)$ time. So, the total time needed is as claimed. □

Finally, we show that the proof of Theorem 13 leads to an approximation algorithm for the MIS problem restricted to map graphs that achieves a logarithmic ratio.

Theorem 16 *Given a vertex-weighted map graph $G = (V, E)$, we can find an independent set of G whose weight is at least $\frac{\log \frac{7}{6}}{6 \log |V|} \cdot OPT(G)$, in $O((\sum_{f \in V} |N_G(f)|^7 + |G|) \cdot \log |V|)$ time.*

6 Concluding remarks

The impracticalness of the PTAS in Theorem 6 and hence that of $MaxIS_3$ mainly result from the impracticalness of the Bodlaender algorithm [4] for tree-decomposition. A more practical algorithm for tree-decomposition can make our algorithms more practical.

Is there a practical PTAS for the MIS or UMIS problem restricted to map graphs? It would also be nice if one can design approximation algorithms for the problems that achieve better ratios than ours. Can $MapIS_1$ and $MapIS_2$ be implemented to run faster than as claimed?

Finally, we point out that Lemma 5 can be used to design a PTAS for the *minimum weighted vertex cover* (MWVC) problem restricted to map graphs. Recall that the MWVC problem requires the computation of a minimum weighted subset S of vertices of a given graph G with nonnegative weights on its vertices such that every edge of G is incident to a vertex of S.

Theorem 17 *There is a PTAS for the MWVC problem restricted to map graphs.*

Proof: (Sketch) Using Lemma 5, we can emulate Baker [1] to obtain a PTAS $\mathcal{A}$ for the MWVC problem restricted to k-map graphs for any fixed integer k.

The PTAS $\mathcal{B}$ for the MWVC problem restricted to map graphs uses $\mathcal{A}$ as a subroutine and is based on the Local Ratio Theorem in [2]. More specifically, $\mathcal{B}$ works as follows. Let G and ϵ be the given map graph and the error parameter, respectively. Let k be the smallest integer such that $\frac{2}{k+1} \leq \epsilon$. First, $\mathcal{B}$ repeats the following: if G has a clique C such that $|C| > k$ and the current weight on each vertex of C is positive, then find C and decrease the weight of each vertex of C by the minimum weight of a vertex of C. Then, $\mathcal{B}$ computes the set U_0 of vertices of weight 0 in G (note that $G - B$ is a k-map graph). Finally, $\mathcal{B}$ calls $\mathcal{A}$ on input $G - B$ and $\epsilon/2$ to obtain a vertex cover U_1 of $G - B$. The output of $\mathcal{B}$ is $U_0 \cup U_1$. The weight of $U_0 \cup U_1$ is at most $(1 + \epsilon)$ times the optimal, because of the Local Ratio Theorem [2] and the fact that the minimum vertex cover of a clique C contains all but one vertex of C. □

Acknowledgments: I thank Professors Seinosuke Toda and Xin He for helpful discussions.

References

[1] B.S. Baker. Approximation algorithms for NP-complete problems on planar graphs. *J. ACM* **41** (1994), 153-180.

[2] R. Bar-Yehuda and S. Even. A local-ratio theorem for approximating the weighted vertex cover problem. *Annals of Discrete Mathematics* **25** (1985), 27-45.

[3] H.L. Bodlaender. Planar graphs with bounded treewidth. Technical Report RUU-CS-88-14, Dept. of Comput. Sci., Univ. of Utrecht, March 1988.

[4] H.L. Bodlaender. A linear time algorithm for finding tree-decompositions of small treewidth. *SIAM J. Comput.* **25** (1996), 1305-1317.

[5] O.V. Borodin. Cyclic coloring of plane graphs. *Discrete Math.* **100** (1992), 281-289.

[6] Z.-Z. Chen. Efficient approximation schemes for maximization problems on $K_{3,3}$-free or K_5-free graphs. *J. Algorithms* **26** (1998), 166-187.

[7] Z.-Z. Chen, M. Grigni, and C.H. Papadimitriou. Planar map graphs. *STOC'98*, 514-523. See http://www.r.dendai.ac.jp/~chen/papers/mg2.ps.gz for a full version.

[8] J. Håstad. Clique is hard to approximate within $n^{1-\epsilon}$. *STOC'96*, 627-636.

[9] R.J. Lipton and R.E. Tarjan. A separator theorem for planar graphs. *SIAM J. Appl. Math.* **36** (1979), 177-189.

[10] R.J. Lipton and R.E. Tarjan. Applications of a planar separator theorem. *SIAM J. Computing* **9** (1980), 615-627.

[11] G.L. Miller. Finding small simple cycle separators for 2-connected planar graphs. *JCSS* **32** (1986), 265-279.

[12] T. Nishizeki and N. Chiba. *Planar Graphs: Theory and Algorithms.* North-Holland, 1988.

[13] O. Ore and M.D. Plummer. Cyclic coloration of plane graphs. *Recent Progress in Combinatorics*, pp. 287-293, Academic Press, 1969.

[14] N. Robertson and P. Seymour. Graph minors. II. Algorithmic aspects of tree-width. *J. Algorithms* **7** (1986), 309-322.

[15] J.A. Telle and A. Proskurowski. Practical algorithms on partial k-trees with an application to domination-like problems. *WADS'93* (LNCS 709), 610-621.

[16] M. Thorup. Map graphs in polynomial time. *FOCS'98*, 396-405.

Appendix

Proof of Lemma 3: We show how to compute I_0. Since H_0 is $(2h-1)$-outerplanar, its treewidth is at most $6h-4$ [3]. So, we can compute a tree-decomposition $(\{X_j \mid j \in J\}, \mathcal{T})$ of H_0 with treewidth at most $6h-4$ in $O(|H_i|^2)$ time [3]. We may assume that the degree of each vertex in $\mathcal{T}$ is at most 3; in case some $j \in J$ has d neighbors $j_1, \ldots, j_d$ in $\mathcal{T}$ with $d \geq 4$, we delete the edges $\{j, j_3\}, \ldots, \{j, j_d\}$ from T, introduce $d-3$ new vertices $\ell_1, \ldots, \ell_{d-3}$ all having the same bag X_j, and further add the edges $\{j, \ell_1\}$, $\{\ell_1, \ell_2\}, \ldots, \{\ell_{d-4}, \ell_{d-3}\}$, $\{\ell_1, j_3\}, \ldots, \{\ell_{d-3}, j_{d-1}\}$, $\{\ell_{d-3}, j_d\}$; $\mathcal{T}$ remains to be a tree-decomposition of H_0.

Root $\mathcal{T}$ at an arbitrary vertex. Then, $\mathcal{T}$ is a binary tree. For each vertex j of $\mathcal{T}$, let $H(j)$ be the subgraph of H_0 induced by the union of the bags of j and its descendants; let $H(j) = (V(j), \mathcal{P}(j); E(j))$ where $V(j) \subseteq V$ and $\mathcal{P}(j) \subseteq \mathcal{P}$. Let $G(j)$ be the half square of $H(j)$. Note that an independent set of $G(j)$ may not be an independent set of G.

To compute I_1, we process the vertices of $\mathcal{T}$ in postorder. Let j be a vertex of $\mathcal{T}$. A *target pair* for j is (S, W) where $S \subseteq X_j \cap V$, S is an independent set of G_0, $W \subseteq X_j \cap \mathcal{P}$, and $N_{H_0}(S) \cap (X_j - W) = \emptyset$. During processing j, for every target pair (S, W) for j, we want to compute a maximum weight subset $I(j, S, W)$ of $V(j)$ satisfying the following conditions:

(a) $I(j, S, W)$ is an independent set of G_0.

(b) $I(j, S, W) \cap X_j = S$.

(c) $N_{H_0}(I(j, S, W)) \cap X_j = W$.

Of course, it is possible that no $I(j, S, W) \subseteq V(j)$ satisfies Conditions (a) through (c). In this case, we record $I(j, S, W) = \emptyset$ during processing j.

The processing of each leaf j of $\mathcal{T}$ is straightforward, because $H(j)$ has at most $3h-4$ vertices. Next, consider the processing of a non-leaf vertex j of $\mathcal{T}$. We assume that j has exactly two children in $\mathcal{T}$, say j_1, j_2; the case where j has exactly one child in $\mathcal{T}$ is simpler. Let (S, W) be a target pair for j. Let $B = W \cap X_{j_1} \cap X_{j_2}$. For each $i \in \{1, 2\}$, let $S_i = S \cap X_{j_i}$, $W_i = ((W \cap X_{j_i}) - N_{H_0}(S)) \cup (N_{H_0}(S_i) \cap X_{j_i})$, and $Z_i = ((W \cap X_{j_i}) - (B \cup N_{H_0}(S))) \cup (N_{H_0}(S_i) \cap X_{j_i})$. For a class $\{U_1, \ldots, U_\ell\}$ of subsets of V, let $\max\{U_1, \ldots, U_\ell\}$ denote the maximum weight subset in the class. We claim that the following hold:

1. If $W - (X_{j_1} \cup X_{j_2}) \not\subseteq N_{H_0}(S)$, then $I(j, S, W) = \emptyset$.

2. Otherwise, $I(j, S, W) = S \cup \max\{\Gamma, \Delta\}$, where $\Gamma = \max\{I(j_1, S_1 \cup S_1', W_1 \cup W_1') \cup I(j_2, S_2 \cup S_2', Z_2 \cup W_2') \mid S_i' \subseteq (X_{j_i} - X_j) \cap V$ and $W_i' \subseteq (X_{j_i} - X_j) \cap \mathcal{P}$ for each $i \in \{1, 2\}$, and $(S_1 \cup S_1', W_1 \cup W_1')$ and $(S_2 \cup S_2', Z_2 \cup W_2')$ are a target pair for j_1 and j_2, respectively$\}$, and $\Delta = \max\{I(j_1, S_1 \cup S_1', Z_1 \cup W_1') \cup I(j_2, S_2 \cup S_2', W_2 \cup W_2') \mid S_i' \subseteq (X_{j_i} - X_j) \cap V$ and $W_i' \subseteq (X_{j_i} - X_j) \cap \mathcal{P}$ for each $i \in \{1, 2\}$, and $(S_1 \cup S_1', Z_1 \cup W_1')$ and $(S_2 \cup S_2', W_2 \cup W_2')$ are a target pair for j_1 and j_2, respectively.$\}$.

To see Claim 1, consider a point $p \in W - (X_{j_1} \cup X_{j_2} \cup N_{H_0}(S))$. Since $(\{X_j \mid j \in J\}, \mathcal{T})$ is a tree-decomposition of H_0, p cannot belong to the bag of a descendant of j, and every $v \in N_{H_0}(p) \cap V(j)$ must belong to X_j. However, by Conditions (b) and (c) required for $I(j, S, W)$, no $v \in N_{H_0}(p) \cap X_j$ can be in $I(j, S, W)$. This establishes Claim 1.

To prove Claim 2, suppose $W - (X_{j_1} \cup X_{j_2}) \subseteq N_{H_0}(S)$. Since $(\{X_j \mid j \in J\}, \mathcal{T})$ is a tree-decomposition of H_0, no $a \in (X_j \cap V) - (X_{j_1} \cup X_{j_2})$ can belong to $\max\{\Gamma, \Delta\}$. Moreover, by the definition of Γ and Δ, and by Condition (b) (for $I(j_1, S_1 \cup S_1', W_1 \cup W_1')$, etc.), no $a \in ((X_{j_1} \cap X_j \cap V) \cup (X_{j_2} \cap X_j \cap V)) - S$ can belong to $\max\{\Gamma, \Delta\}$. Hence, $(S \cup \max\{\Gamma, \Delta\}) \cap X_j = S$, implying that Condtion (b) required for $I(j, S, W)$ is satisfied.

To see that Condtion (c) required for $I(j,S,W)$ is satisfied, first recall that $W-(X_{j_1}\cup X_{j_2})\subseteq N_{H_0}(S)$. By the definition of Γ and Δ, and by Condition (c) (for $I(j_1,S_1\cup S_1',W_1\cup W_1')$, etc.), $(W\cap(X_{j_1}\cup X_{j_2}))-N_{H_0}(S))\subseteq N_{H_0}(\max\{\Gamma,\Delta\})$. So, $W\subseteq N_{H_0}(S\cup\max\{\Gamma,\Delta\})$. Thus, to establish Condition (c) for $I(j,S,W)$, it remains to show that there is no $p\in N_{H_0}(S\cup\max\{\Gamma,\Delta\})\cap(X_j-W)$. To see this, assume on the contrary that such a p exists. Then, there is $a\in S\cup\max\{\Gamma,\Delta\}$ with $p\in N_{H_0}(a)$. Since (S,W) is a target pair for j, $a\notin S$. In turn, $a\in\max\{\Gamma,\Delta\}$. So, by Condition (b) already established for (j,S,W), $a\in V(j_1)-X_j$ or $a\in V(j_2)-X_j$. We assume $a\in V(j_1)-X_j$; the case where $a\in V(j_2)-X_j$ is similar. Then, since $\{a,p\}$ is an edge in H_0 and $(\{X_j\mid j\in J\},\mathcal{T})$ is a tree-decomposition of H_0, p must belong to $X_j\cap X_{j_1}$. So, by the definition of Γ and Δ, and by Condition (c) (for $I(j_1,S_1\cup S_1',W_1\cup W_1')$, etc.), $p\in W_1$ or $p\in Z_1$. However, by the definition of W_1 and Z_1, $p\in W\cap X_{j_1}$ or $p\in N_{H_0}(S_1)\cap X_{j_1}$, contradicting that $p\notin W$ and (S,W) is a target pair for j. This establishes Condition (c) required for $I(j,S,W)$.

We next establish Condition (a) required for $I(j,S,W)$. To this end, we first prove that there is no $\{a,b\}\in E$ such that $a\in S$ and $b\in\max\{\Gamma,\Delta\}-S$. Towards a contradiction, assume that such an edge $\{a,b\}\in E$ exists. Then, by Condition (b) already established for (j,S,W), either $b\in V(j_1)-X_j$ or $b\in V(j_2)-X_j$. We assume $b\in V(j_1)-X_j$; the other case is similar. Then, $a\in S-X_{j_1}$, by the definition of Γ and Δ, and by Condition (a) (for $I(j_1,S_1\cup S_1',W_1\cup W_1')$, etc.). Since $(\{X_j\mid j\in J\},\mathcal{T})$ is a tree-decomposition of H_0, the facts that $a\in S-X_{j_1}$, $b\in V(j_1)-X_j$ and $\{a,b\}\in E$ imply that there exists $p\in X_j\cap X_{j_1}$ with $\{a,b\}\subseteq N_{H_0}(p)$. Since $b\in\max\{\Gamma,\Delta\}-S$, Conditions (a) and (c) (for $I(j_1,S_1\cup S_1',W_1\cup W_1')$, etc.) imply that $p\notin N_{H_0}(S_1)$ and that $p\in W_1$ or $p\in Z_1$. However, this contradicts the definition of W_1 and Z_1, because $p\in N_{H_0}(S)$.

So, to establish Condition (a) for $I(j,S,W)$, it remains to prove that $\max\{\Gamma,\Delta\}-S$ is an independent set of G_0. Towards a contradiction, assume that $\max\{\Gamma,\Delta\}-S$ is not an independent set of G_0. Then, some $a\in\max\{\Gamma,\Delta\}\cap(V(j_1)-X_j)$ and some $b\in\max\{\Gamma,\Delta\}\cap(V(j_2)-X_j)$ are adjacent in G_0. Let $p\in N_{H_0}(a)\cap N_{H_0}(b)$. Since $(\{X_j\mid j\in J\},\mathcal{T})$ is a tree-decomposition of H_0, p must belong to all of X_{j_1}, X_{j_2} and X_j. In turn, Conditions (a) and (c) (for $I(j_1,S_1\cup S_1',W_1\cup W_1')$, etc.) imply that $p\notin N_{H_0}(S_1)$, $p\notin N_{H_2}(S_2)$, $p\in W_1\cup Z_1$, and $p\in W_2\cup Z_2$. However, this contradicts the definition of W_1 and Z_1, because $p\in N_{H_0}(S)$. This completes the proof of Claim 2.

Before processing the vertices of $\mathcal{T}$, we compute and store the adjacency matrix of H_0 and that of its half square. Then, by Claims 1 and 2, processing each vertex j of $\mathcal{T}$ takes $O(2^{2(6h-4)}h)$ time. Since $|\mathcal{T}|=O(|H_0|)$, the whole processing takes $O(2^{2(6h-4)}h|H_0|)$ time. □

Proof of Lemma 5: We may assume that G is connected. Starting at an arbitrary vertex r, we perform a breadth-first search (BFS) on G to obtain a BFS tree T of G. For each vertex v in G, we define the level number $\ell(v)$ of v to be the number of edges on the path from r to v in T. For each $i\in\{0,\ldots,h-1\}$, let G_i be the subgraph of G obtained from G by deleting all vertices v with $\ell(v)\equiv i\pmod h$. Obviously, each vertex of G appears in exactly $h-1$ of these subgraphs. It remains to show that each G_i has treewidth $\leq k(6h-4)$.

Let $H=(V,\mathcal{P};E_H)$ be a bipartite planar graph whose half square is G. Such H exists, by Lemma 1. Starting at r, we perform a BFS on H to obtain a BFS tree T'. For each vertex v in H, we define the level number $\ell'(v)$ of v to be the number of edges on the path from r to v in T'. It can be verified that $\ell'(v)=2\ell(v)$ for all vertice v of G. Thus, for each $i\in\{0,\ldots,h-1\}$, G_i is the half square of the subgraph H_i of H obtained from H by deleting all vertices v with $\ell'(v)\equiv 2i\pmod{2h}$. Clearly, H_i is a $(2h-1)$-outerplanar graph, and hence has treewidth $\leq 6h-4$ [3].

Let $(\{X_j\mid j\in J_i\},\mathcal{T}_i)$ be a tree-decomposition of H_i with treewidth $\leq 6h-4$. For each $j\in J_i$,

let X'_j be the set obtained from X_j by replacing each $p \in X_j \cap \mathcal{P}$ with the vertices $v \in V - X_j$ such that v is adjacent to p in H_i. Obviously, the treewidth of $(\{X'_j \mid j \in J_i\}, \mathcal{T}_i)$ is at most $k(6h-4)$. It remains to show that $(\{X'_j \mid j \in J_i\}, \mathcal{T}_i)$ is a tree-decomposition of G_i.

Since G_i is the half square of H_i, $\cup_{j \in J_i} X'_i$ is the set of all vertices in G_i. Let $\{u, v\}$ be an edge in G_i. Then, there is a $p \in \mathcal{P}$ such that both u and v are adjacent to p in H. Since p is contained in H_i, there is a $j \in J_i$ with $p \in X'_j$. So, both u and v are in X'_j. Next, let $j_1, j_2, j_3 \in J_i$ and suppose that j_2 lies on the path P between j_1 and j_3 in $\mathcal{T}$. We want to show that $X'_{j_1} \cap X'_{j_3} \subseteq X'_{j_2}$. To this end, let $u \in X'_{j_1} \cap X'_{j_3}$. Consider the following cases:

Case 1: $u \in X_{j_1}$ and $u \in X_{j_3}$. Then, $u \in X_{j_2}$ and certainly $u \in X'_{j_2}$.

Case 2: $u \in X_{j_1}$ and $u \notin X_{j_3}$. Then, there is a $p \in X_{j_3} \cap \mathcal{P}$ such that u is adjacent to p in H_i. Since $\{p, u\}$ is an edge in H_i, there is a $j' \in J_i$ with $\{p, u\} \subseteq X_{j'}$. Because $\mathcal{T}$ is a tree, j_2 appears either on the path between j' and j_1 in $\mathcal{T}$ or on the path between j' and j_3 in $\mathcal{T}$. In the former case, $u \in X_{j_2}$ and certainly $u \in X'_{j_2}$. In the latter case, $p \in X_{j_2}$ and in turn $u \in X'_{j_2}$.

Case 3: $u \notin X_{j_1}$ and $u \in X_{j_3}$. Similar to Case 2.

Case 4: $u \notin X_{j_1}$ and $u \notin X_{j_3}$. Then, there are $p_1 \in X_{j_1} \cap \mathcal{P}$ and $p_3 \in X_{j_3} \cap \mathcal{P}$ such that u is adjacent to both p_1 and p_3 in H_i. If $p_1 = p_3$, then $p_1 \in X_{j_2}$ and in turn $u \in X'_{j_2}$. So, suppose $p_1 \neq p_3$. Since $\{p_1, u\}$ is an edge in H_i, there is a $j' \in J_i$ with $\{p_1, u\} \subseteq X_{j'}$. Because $\mathcal{T}$ is a tree, j_2 appears either on the path between j' and j_1 in $\mathcal{T}$ or on the path between j' and j_3 in $\mathcal{T}$. In the former case, $p_1 \in X_{j_2}$ and in turn $u \in X'_{j_2}$. So, suppose it is the latter case. Since $\{p_3, u\}$ is an edge in H_i, there is a $j'' \in J_i$ with $\{p_3, u\} \subseteq X_{j''}$. Because $\mathcal{T}$ is a tree, j_2 appears either on the path between j'' and j' in $\mathcal{T}$ or on the path between j'' and j_3 in $\mathcal{T}$. In the former case, $u \in X_{j_2}$ and certainly $u \in X'_{j_2}$. In the latter case, $p_3 \in X_{j_2}$ and in turn $u \in X'_{j_2}$. □

Proof of Theorem 6: Let G be the given vertex-weighted k-map graph and ϵ be the given error parameter. Let $h = \lceil \frac{1}{\epsilon} \rceil$. Using Lemma 5, we compute h induced subgraphs of G in $O(h|G|)$ time such that (1) each of the subgraphs is of treewidth $\leq k(6h-4)$ and (2) each vertex of G appears in exactly $h-1$ of the subgraphs. For each such subgraph G_i, we first compute a tree-decomposition of G_i in $O(2^{32(6kh-4k)^3}|G_i|)$ time [4], and then use the tree-decomposition to find a maximum weight independent set I_i of G_i in $O(2^{6kh-4k+2\log(6kh-4k)}|G_i|)$ time [15]. The I_i whose weight is maximized over all such subgraphs G_i has weight at least $(1-\epsilon) \cdot OPT(G)$. □

Proof of Lemma 7: Let $\mathcal{M} = (\mathcal{E}, \varphi)$ be as in the lemma. We modify $\mathcal{M}$ by performing the following steps in turn:

1. Delete all edges e of $\mathcal{E}$ such that there is no nation on $\mathcal{M}$ whose boundary contains e.
2. Delete all isolated points in $\mathcal{E}$. (Comment: Steps 1 and 2 have no effect on the nations and their adjacencies on $\mathcal{M}$.)
3. If a point p of degree 4 or more in $\mathcal{E}$ is touched by a lake, then for each pair (e_1, e_2) of consecutive edges incident to p such that the boundary of a lake L contains both e_1 and e_2, perform the following substeps:
   (a) Split e_2 into two edges by putting a new point x at the middle of e_2.
   (b) Split L into two smaller faces L and L' by adding a new edge e_x between x and the endpoint of e_1 other than p, where the boundary of L' is a triangle formed by the edges e_1, e_x and $\{x, p\}$.
   (c) Find the nation $f_1 \neq L'$ whose boundary contains e_1, and delete e_1 to merge L' and f_1 into a bigger nation f_1.

Repeat Step 3 until no such point as p exists in $\mathcal{E}$. (Comment: After this, each connected component of $\mathcal{E}$ becomes biconnected and the adjacencies of the nations remain the same as before.)

4. If a point p is incident to exactly two edges $e_1 = \{p, q_1\}$ and $e_2 = \{p, q_2\}$ in $\mathcal{E}$, then replace edges e_1 and e_2 by a single edge $\{q_1, q_2\}$. (Comment: $q_1 \neq q_2$ because each connected component of $\mathcal{E}$ is biconnected and no nation is isolated.)
   Repeat Step 4 until no such point as p exists in $\mathcal{E}$.

5. For each lake L on $\mathcal{M}$ whose boundary contains exactly two edges e_1 and e_2, find the nation f_1 whose boundary contains e_1, and delete e_1 to merge f_1 and L into a bigger nation f_1. (Comment: After Step 5, if a point p is incident to exactly two edges e_1 and e_2, then neither e_1 nor e_2 appears on the boundary of a lake.)

6. Same as Step 4. (Comment: After Step 6, no lake on $\mathcal{M}$ can contain exactly two edges.)

7. For each lake L whose boundary contains exactly three edges $e_1 = \{p_1, p_2\}$, $e_2 = \{p_2, p_3\}$, and $e_3 = \{p_3, p_1\}$, perform the following substeps:

   (a) Put a new point q inside L.

   (b) Split L into three smaller faces L_1, L_2 and L_3 by adding three new edges $\{q, p_1\}$, $\{q, p_2\}$ and $\{q, p_3\}$, where the boundary of L_i contains e_i for each $i \in \{1, 2, 3\}$.

   (c) For each $i \in \{1, 2, 3\}$, find the nation $f_i \neq L_i$ whose boundary contains e_i, and delete e_i to merge L_i and f_i into a bigger nation f_i.

After the above steps are performed, $\mathcal{M}$ becomes well-formed and its graph remains unchanged. □

Proof of Lemma 8: We may assume that G is connected. Let $\mathcal{M} = (\mathcal{E}, \varphi)$ be a well-formed k-map whose graph is G. For each integer $j \geq 4$, let h_j be the number of j-points in $\mathcal{M}$. Let H be the simple and bipartite plane graph constructed from $\mathcal{E}$ by (1) putting a new point p_f inside each nation f, (2) connecting p_f to each 3^+-point on the boundary of f, and (3) deleting all original edges. H has at most $2(n + \sum_{j=4}^{k} h_j) - 4$ edges and exactly $\sum_{j=4}^{k} j h_j$ edges. Thus, $\sum_{j=4}^{k} (j-2) h_j \leq 2n - 4$.

Suppose that a nation L on $\mathcal{M}$ touches a 3^+-point p. Then, the two edges e_1 and e_2 on the boundary of L and incident to p must appear around p consecutively. *Detaching L from p* is the operation of modifying $\mathcal{M}$ by performing Substeps 3a through 3c in the proof of Lemma 7.

After detaching a nation from a j-point p with $j \geq 4$ in $\mathcal{M}$, we lose at most $j-3$ edges from G; moreover, p becomes a $(j-1)$-point and no new 3^+-point is created. Thus, after repeatedly detaching a nation from a 3^+-point until no 3^+-point exists in $\mathcal{M}$, we lose at most $\sum_{j=4}^{k} \frac{(j-3)(j-2)}{2} h_j$ edges from G; moreover, G becomes planar. Hence, $m \leq 3n - 6 + \sum_{j=4}^{k} \frac{(j-3)(j-2)}{2} h_j$.

Combining the two inequalities above, we obtain $m \leq kn - 2k$. □

Proof of Corollary 9: The corollary trivially holds when $k \leq 2$. So, assume $k \geq 3$. By Lemma 8, the vertex v with minimum degree in G has degree $2k-1$ or less. We delete v from G, recursively color the resulting graph H which must be a k-map graph, and extend the returned $2k$-coloring of H to a $2k$-coloring of the original graph G. □

Proof of Lemma 11: Obviously, a separator of the heaviest connected component G' of G is also a separator of G, and the lemma trivially holds when G' has at most two vertices. So,

we may assume that G is connected and $n \geq 3$. Moreover, if $k \leq 3$, then G is planar and the lemma holds [9]. So, assume $k \geq 4$. Then, by Lemma 7, we may assume that $\mathcal{M}$ is a well-formed k-map. In turn, $\mathcal{E}$ is biconnected and the dual plane graph $\mathcal{E}^*$ of $\mathcal{E}$ is biconnected. Since $\mathcal{M}$ is a well-formed k-map, the boundary of each face in $\mathcal{E}^*$ contains at most k edges. Each vertex of $\mathcal{E}^*$ that is a nation on $\mathcal{M}$ has its associated weight. To each vertex of $\mathcal{E}^*$ that is a lake on $\mathcal{M}$, we assign a weight 0. Then, we find a separator C of $\mathcal{E}^*$ that is a simple cycle and is of size at most $2\sqrt{k(n+n_L)}$ [11], where n_L is the number of lakes on $\mathcal{M}$. Note that the vertices of C that are nations on $\mathcal{M}$ form a separator of G. So, it suffices to show that $n_L \leq n-2$.

Let n_p be the number of points in $\mathcal{E}$. Then, $n_p \geq 4n_L$ because $\mathcal{M}$ is well-formed. Let H be the simple and bipartite plane graph constructed from $\mathcal{E}$ by (1) putting a new point q_f inside each face f of $\mathcal{E}$, (2) connecting q_f to each point on the boundary of f, and (3) deleting all original edges. H has at most $2(n+n_L+n_p)-4$ edges and at least $3n_p$ edges. Thus, $n_L \leq n-2$. □

Proof of Theorem 12: By Lemma 11 and the discussions in [10], we can prove the following: Given a k-map $\mathcal{M} = (\mathcal{E}, \varphi)$ and a positive integer x, we can compute a set S of nations on $\mathcal{M}$ in $O(|\mathcal{E}|\log|\mathcal{E}|)$ time such that $|S| \leq 6n\sqrt{\frac{2k}{x}}$ and the removal of S from the map graph G of $\mathcal{M}$ leaves a graph G' whose connected components each contain at most x vertices. For each connected component C of G', we compute an independent set I_C of maximum size in $O(|C|2^x)$ time. The output is $I = \cup_C I_C$. By Lemma 8, $\frac{|I|}{\alpha(G)} \geq 1 - 12k\sqrt{\frac{2k}{x}}$. Thus, setting $x = 288h^2k^3$ yields $\frac{|I|}{\alpha(G)} \geq 1 - \frac{1}{h}$. □

Proof of Theorem 16: Let I_0 be the set of all vertices $v \in V$ such that the connected component of G containing v has at most two vertices. Consider a well-formed map $\mathcal{M}$ of $G[V-I_0]$. As observed in the proof of Theorem 13, $\mathcal{G}(\mathcal{M})$ is a plane graph with at most $3|V-I_0|-6$ edges. Thus, $\mathcal{G}(\mathcal{M})$ has at least $\frac{|V-I_0|+12}{7}$ vertices of degree at most 6. From the proof of Theorem 13, we can see that for every vertex f of degree at most 6 in $\mathcal{G}(\mathcal{M})$, $\alpha(G_f) \leq 6$ where $G_f = G[N_G(f)]$.

So, we compute the set $V_1 = I_0 \cup \{f \in V - I_0 \mid \alpha(G_f) \leq 6\}$. This takes $O(\sum_{f \in V} |N_G(f)|^7)$ time. Note that $|V_1| \geq \frac{|V|}{7}$ and $V_1 = \{f \in V \mid \alpha(G_f) \leq 6\}$.

Similarly, we compute $G_2 = G[V - V_1]$ and $V_2 = \{f \in V - V_1 \mid \alpha(G_2[N_{G_2}(f)]) \leq 6\}$. Continuing this process, we can obtain a sequence $G_3, V_3, G_4, V_4, \ldots$ such that for each $i \geq 3$, $G_i = G[V - (V_1 \cup \cdots \cup V_{i-1})]$ and $V_i = \{f \in V - (V_1 \cup \cdots \cup V_{i-1}) \mid \alpha(G_i[N_{G_i}(f)]) \leq 6\}$. Let $\ell = \log_{\frac{7}{6}} |V|$. Obviously, for all $i > \ell$, G_i is an empty graph.

Finally, for each $i \in \{1, \ldots, \ell\}$, we compute an independent set I_i of G_i greedily whose weight is at least $\frac{OPT(G_i)}{6}$. This takes $O(|G|\log|V|)$ time in total. The maximum weight subset among $I_1, \ldots, I_\ell$ has weight at least $\frac{\log \frac{7}{6} \cdot OPT(G)}{6 \log |V|}$. □