ON DEFECT SETS IN BIPARTITE GRAPHS

P. E. Haxell, M. Loebl

March 10, 1998

Abstract

A *defect set* in a bipartite graph with vertex classes V and W is a subset $X \subset V$ such that the neighbourhood $N(X)$ satisfies $|N(X)| < |X|$. We study a lemma on defect sets in bipartite graphs with certain expanding properties from the algorithmic complexity point of view. This lemma is the core of a result of Friedman and Pippenger which states that expanding graphs contain all small trees. We also discuss related problems of finding shortest circuits of matroids represented over a field. In particular, we propose a new straightforward method to derive a weaker form (PR-completeness) of the recent NP-completeness results of Khachiyan [11] and Vardy [17] concerning this problem for the field of rationals and $GF(p^m)$, respectively.

1 Introduction.

Let $G = (V, E)$ be a graph with vertex set V and edge set E. For each $X \subset V$, we let $N(X) = N_G(X)$ be the set of vertices of G joined by an edge to at least one vertex of X. In 1987 Friedman and Pippenger [7] proved the following nice theorem.

Theorem 1.1 *If H is a non-empty graph such that, for every $X \subset V(H)$ with $|X| \leq 2n-2$, we have*

$$|N_H(X)| \geq (d+1)|X|,$$

then H contains every tree with n vertices and maximum degree at most d.

The statement of this theorem and its proof are used to obtain several interesting consequences in combinatorics and computer science (see Feldman et.al [6], Aggarwal et.al. [1], Haxell, Kohayakawa [8]). The proof is non-constructive: it only shows the existence of the tree, but does not give an efficient (i.e. polynomial-time) method to find it. Actually no such method has been found so far, unless the assumptions of Theorem 1.1 are made much stronger (see Aggarwal et.al. [1]).

In this note we study the algorithmic complexity of Theorem 1.1. We restrict ourselves to bipartite graphs, for which the theorem has the following form.

Theorem 1.2 *If $H = (V, W, E)$ is a non-empty bipartite graph such that, for every $X \subset V$ and every subset $X \subset W$ with $|X| \leq 2n-2$,*

$$|N_H(X)| \geq (d+1)|X|,$$

then H contains every tree with n vertices and maximum degree at most d.

In Section 2 of this note, we give Friedman and Pippenger's proof of Theorem 1.2 in a different form, in particular we isolate the core of the proof as a lemma (Lemma 2.1). A proof of the lemma based on an efficient algorithm would lead to an efficient method to find the trees whose existence is guaranteed by Theorem 1.2. In order to study the algorithmic complexity of 2.1 we introduce the following notion.

Let $\mathcal{P}$ be an NP-complete problem. A polynomial-time algorithm A will be called an *NP-decider* if, given three instances of $\mathcal{P}$ such that the answer of $\mathcal{P}$ is "YES" to one instance and 'NO' to the other two, A chooses one of the "NO"-instances.

It seems unlikely that an NP-decider exists. More concretely, its concept is similar to the definition of the complexity class DIF^P (see Papadimitriou, Yannakakis [13], Welsh [19]). A typical complete problem for DIF^P is $SAT - UNSAT$: Given two Boolean formulas E_1 and E_2 in conjunctive normal form, is E_1 satisfiable and E_2 not satisfiable?

The $UNIQUE - SAT$ problem, which asks whether a formula in conjunctive normal form has exactly one satisfying assignment, belongs to DIF^P. There is no efficient algorithm to solve $UNIQUE - SAT$ unless $NP = RP$ (see Vazirani, Vazirani [18], Welsh [19]), where RP denotes the class of decision problems which may be solved by a random polynomial algorithm. We postpone the definition of RP to Section 4, but remark that $RP = NP$ is considered as unlikely as $P = NP$.

In Section 3 we introduce a decision problem called "Special Defect Set (SDS)" and show that its algorithmic complexity is equivalent to the algorithmic complexity of Lemma 2.1 in the following sense. If SDS may be solved in polynomial time then an efficient method exists to solve 2.1. On the other hand, no such method exists if SDS is NP-complete, unless an NP-decider exists. We also include some results on the complexity of SDS, in particular a formal variant of SDS is in the complexity class UP (see Section 3 for the definition of UP).

Finally in Section 4 we make a connection with "shortest circuit" problems in matroids represented over a field. In particular, we use the notion of *random polynomial completeness for NP* (see Vazirani, Vazirani [18], Johnson [9]) to derive weaker forms of two recent results of Khachiyan [11] and Vardy [17]: we show that the following problem is PR-complete: given a matrix over a field $\mathbf{F}$ and a number k, find out whether the matrix has k linearly dependent columns.

2 The Lemma.

A *weighted bipartite graph* $H = (V, W, E, \Delta)$ is a bipartite graph H with vertex classes V and W and edge set E, together with a function Δ which assigns a non-negative integer to each vertex of V. For $X \subset V$ let $\Delta(X) = \sum_{x \in X} \Delta(x)$.

Let $H = (V, W, E, \Delta)$ be a weighted bipartite graph and let $v \in V$ and $w \in W$ be such that $vw \in E$ and $\Delta(v) > 0$. We denote by $H(v, w)$ the weighted bipartite graph $H' = (V', W', E', \Delta')$ where $V' = V$, $W' = W - \{w\}$, $\Delta'(x) = \Delta(x)$ for $x \neq v$ in V and $\Delta'(v) = \Delta(v) - 1$.

The following lemma is the core of Friedman and Pippenger's proof of Theorems 1.1 and 1.2. It is more natural to formulate it in the language of weighted bipartite graphs here. We remark though that as a result on a density of hypergraphs it may be of independent interest.

Lemma 2.1 *Let l be a positive integer and $H = (V, W, E, \Delta)$ be a weighted bipartite graph with the following properties.*

(1) For each $X \subset V$ with $|X| \leq 2l$, $|N_H(X)| \geq \Delta(X)$,

(2) If for $X \subset V$ with $|X| \le 2l$, $|N_H(X)| = \Delta(X)$, then $|X| \le l$.

Then for each $v \in V$ with $\Delta(v) > 0$ there is $w \in W$ such that $H(v,w)$ satisfies (1).

Proof. Let $\Delta(v,w)$ denote the weight function of $H(v,w)$. Fix a vertex v with $\Delta(v) > 0$.

Suppose the result is not true. Then for each $w \in N(v)$, there exists a 'bad' set Y_w with $|Y_w| \le 2l$ and $|N_{H(v,w)}(Y_w)| < \Delta(v,w)(Y_w)$.

Then by (1), each Y_w must satisfy $|N_H(Y_w)| = \Delta(Y_w)$, and moreover $v \notin Y_w$ (see the definition of $\Delta(v,w)$). Hence by (2) of 2.1 we see that $|Y_w| \le l$.

The trick is to observe that the function F defined on the subsets of V by $F(X) = |N_H(X)| - \Delta(X)$ is submodular for any weighted bipartite graph H: it is well-known for the function $N_H(X)$, and $\Delta(X)$ is modular by its definition.

Let w_1, w_2 be two vertices in W adjacent to v. We have $|Y_{w_1} \cup Y_{w_2}| \le 2l$ and, using submodularity of F, $|N_H(Y_{w_1} \cup Y_{w_2})| = \Delta(Y_{w_1} \cup Y_{w_2})$. Hence $|Y_{w_1} \cup Y_{w_2}| \le l$ by (2).

In this way we find that the union Y of all Y_w, $w \in N_H(v)$, satisfies $|N_H(Y)| = \Delta(Y)$ and $|Y| \le l$. Then however $Y \cup \{v\}$ does not satisfy (1) for H, which is a contradiction. □

Note that Lemma 2.1 is no longer true if Property (2) is deleted. Next we show how to prove Theorem 1.2 using 2.1. It is easy to see that the proof would give a polynomial time algorithm if the Lemma 2.1 subroutine were polynomial. If G is a graph and v a vertex of G, we denote by $deg_G(v)$ the degree of v in G.

Proof of Theorem 1.2.

Let $G = (V, W, E)$ be a non-empty bipartite graph such that, for every $X \subset V$ or $X \subset W$ with $|X| \le 2n-2$,

$$|N_G(X)| \ge (d+1)|X|.$$

We want to show that G contains every tree with n vertices and maximum degree at most d.

Let S be a subgraph of G. We associate a function C_S defined on the subsets of V and the subsets of W as follows:

$C_S(X) = A_S(X) - B_S(X)$, where $A_S(X) = |N_G(X) \setminus S|$ and $B_S(X) = \sum_{x \in X} B_S(x)$, where $B_S(x) = d - deg_S(x)$.

A subgraph S of G is called *correct* if $C_S(X) \ge 0$ for each $X \subset V$ and $X \subset W$ with $|X| \le 2n-2$.

Let T be a fixed tree with n vertices and maximum degree at most d. We will show that for each subtree S' of T there is a correct subtree S of G which is isomorphic to S'. Let R' be a minimal counterexample. R' must have at least two vertices. Let v be a leaf of R' and let $S' = R' - v$. Let S be a correct subtree of G which is isomorphic to

S' and let w be a vertex of S' such that attaching a vertex of degree one to it leads to a tree isomorphic to R'. Without loss of generality let $w \in V$. Let H be the weighted bipartite subgraph of G induced by $V \cup (W \setminus S)$, where the weight of each $u \in V$ is defined to be $B_S(u)$. Then note that $B_S(w) \geq 1$.

According to our assumptions, H satisfies (1) of 2.1 with $l = n - 1$, and it also satisfies (2) since for $|X| \leq 2n - 2$, $C_S(X) \geq (d+1)|X| - n + 1 - d|X|$ and thus $C_S(X) = 0$ only if $|X| \leq n - 1$.

Let y be the vertex of H whose existence is guaranteed by 2.1 and let R be obtained from S by attaching vertex y to w. Then R is isomorphic to R' and it is a correct subgraph of G: if $X \subset W$ then $C_R(X) \geq C_S(X)$, and if $X \subset V$, $|X| \leq 2n - 2$ then $C_R \geq 0$ follows from 2.1.

This contradicts the choice of R', and so the proof is complete.

$\square$

3 Complexity Questions.

Let $G = (V, W, E)$ be a bipartite graph and D positive integer. A subset $X \subset V$ is called a D-*defect* set if $|N(X)| < D|X|$. If Δ is a function on the subsets of V then $X \subset V$ is called Δ-*defect* if $|N(X)| < \Delta(X)$.

To find the smallest cardinality of a D-defect subset of V is an NP-complete problem even for D fixed. This was first observed by Blum et. al. in [5], who proved that to decide whether a bipartite graph $B = (V_1, V_2, E)$ with $|V_1| = |V_2|$ has a 1-defect set of at most $|V_1/2|$ vertices is NP-complete. Proposition 3.7 below shows that another special case of the "smallest 1-defect set problem" is NP-complete.

There is a basic difficulty concerning the investigation of the algorithmic complexity of Lemma 2.1 (and Theorem 1.2), namely, that it is already an NP-complete task to check whether the assumptions of the lemma are satisfied for an arbitrary weighted bipartite graph. In a typical application of Theorem 1.2 however, we know, usually by the way the graph was constructed, that the assumptions of the theorem are satisfied, and we want to know how hard it is to find the tree whose existence is guaranteed by the theorem. Since there is no algorithmic proof of Theorem 1.2, to actually construct the tree one must start with a bipartite graph satisfying much stronger assumptions (see Aggarwal [1]).

Therefore the algorithmic complexity of Lemma 2.1 that interests us lies in finding the vertex w, given that the assumptions hold, rather than in checking the assumptions themselves. Strictly speaking then, the problem of finding w does not satisfy the definition of "problem" according to Johnson [10], for example, since it is not known that the input can be recognised in polynomial time. This is not an unusual situation, see e.g. Barany, Onn [3], where the complexity of Colourful Caratheodory's Theorem is studied.

To overcome this difficulty we introduce the following formal construction. We assume that each input graph for our given problem is accompanied by a 1-bit "certificate" which states whether or not the input satisfies the conditions required for the problem. Then the complexity question becomes that of solving the problem, given the information that the input is acceptable.

Using this device, we now introduce the *Special Defect Set Problem (SDS)*.

Definition 3.1 *Let SDS be the following decision problem: We are given a positive integer l and a weighted bipartite graph $G = (V, W, E, \Delta)$ and a certificate equal to "YES" if G satisfies property EXP, and "NO" otherwise.*

EXP: if $X \subset V$, $|X| \leq 2l$ is a Δ-defect subset then $|N(X)| = |\Delta(X)| - 1$ and $|X| \leq l$.

SDS outputs "YES" if the certificate is equal to "YES" and G has a Δ-defect set of at most l vertices.

Theorem 3.2 *If SDS is polynomially solvable then there is an efficient method to find the vertex which existence is assured by Lemma 2.1.*

Proof. Following the proof of 2.1, it is enough to recognize which choice of w is bad. Let $\Delta(w)$ be the weight function of $H(v, w)$. We need to know whether $H(v, w)$ has a $\Delta(w)$-defect set of at most l vertices. Moreover from the assumptions of 2.1 we know that if $X \subset V$, $|X| \leq 2l$ is Δ-defect then $|N(X)| = \Delta(X) - 1$ and $|X| \leq l$. SDS answers "YES" to $H(v, w)$ if and only if w is a bad choice for 2.1.

□

Theorem 3.3 *If SDS is NP-complete then there is no efficient method to find a vertex whose existence is assured by Lemma 2.1 unless an NP-decider exists.*

Proof. We design an NP-decider under the assumption that SDS is NP-complete and there is an efficient method to find the vertex w in Lemma 2.1.

Let l be a positive integer. Let $G_i = (V_i, W_i, E_i, \Delta_i)$, $i = 1, 2, 3$ be weighted bipartite graphs and let (l, G_i, C_i) be three inputs of SDS, where C_i is a certificate for G_i. Moreover let the answer to SDS be "YES" for one of the three inputs and "NO" for the other two inputs.

If one of the C_i is equal to "NO" then SDS answers "NO" to the input with that certificate. Hence let all three certificates equal "YES".

Let $G = (V, W, E, \Delta)$ be the weighted bipartite graph constructed as follows:

1. $V = V_1 \cup V_2 \cup V_3 \cup \{z\}$, where z is a new vertex,

2. $W = W_1 \cup W_2 \cup W_3 \cup \{z_1, z_2, z_3\}$, where z_1, z_2, z_3 are new vertices,

3. E is obtained from $E_1 \cup E_2 \cup E_3$ by adding all edges between z_i and $V_i \cup \{z\}$, $i = 1, 2, 3$,

4. $\Delta(z) = 1$ and for each $x \in V_i$, $\Delta(x) = \Delta_i(x)$, $i = 1, 2, 3$.

Here G and l satisfy the assumptions of Lemma 2.1 since all three certificates are equal to "YES". However we also know that $G(z, z_1)$ does not satisfy (1) of 2.1. Hence if an efficient method exists to choose i such that $G(z, z_i)$ satisfies (1) of 2.1 then an NP-decider exists.

$\square$

This leaves us with the interesting question of determining the complexity of SDS. By making just a formal adjustment to the definition, we can show that SDS with $\Delta(v) = 1$ for all $v \in V$ belongs to the complexity class UP. This class is defined to be the class of decision problems that may be solved by a non-deterministic polynomial-time algorithm with at most one accepting computation (see Welsh [19]).

Definition 3.4 *Let minSDS be the following decision problem: We are given a positive integer l and a bipartite graph $G = (V, W, E)$ and a certificate equal to "YES" if G satisfies property EXP(1), and "NO" otherwise.*

EXP(1): if $X \subset V$, $|X| \leq 2l$ is a 1-defect set then $|N(X)| = |X| - 1$ and $|X| \leq l$.

minSDS outputs "YES" if the certificate equals "YES" and G has a 1-defect set of at most l vertices which is minimal with respect to inclusion.

Note that minSDS is just a formal variant of SDS with $\Delta = 1$: for every input to SDS, the output of SDS is "YES" if and only if the output of minSDS is "YES" for the same input.

Lemma 3.5 *Let l be a positive integer and $G = (V, W, E)$ be a bipartite graph with Property EXP(1). Suppose X, $Y \subset V$ are both 1-defect sets of size at most l, and that X is minimal with respect to inclusion. Then $X \subseteq Y$.*

Proof. It suffices to recall that the function on the subsets of V which maps $U \subset V$ to $|N(U)|$ is submodular. Therefore

$$|N(X \cup Y)| - |X \cup Y| + |N(X \cap Y)| - |X \cap Y| \leq |N(X)| - |X| + |N(Y)| - |Y| = -2.$$

Since EXP holds for both $X \cup Y$ and $X \cap Y$ we have that $|N(X \cup Y)| = |X \cup Y| - 1$ and $|N(X \cap Y)| = |X \cap Y| - 1$. Hence in particular $X \cap Y$ is a 1-defect set, which implies by minimality of X that $X = X \cap Y$. Thus $X \subseteq Y$.

$\square$

This lemma shows that if a bipartite graph $G = (V, W, E)$ satisfies the conditions for minSDS then the minimal 1-defect set of size at most l is unique, if it exists. Note also that for $X \subset V$ it can be verified in polynomial time that X is a minimal defect set, since it suffices to check that X is a defect set, and that for each $x \in X$ there is a complete matching from $X \setminus \{x\}$ to W.

Therefore for a given input to minSDS, there is at most one accepting computation, and so minSDS belongs to UP.

We conclude this section by showing that another SDS-type problem is in fact NP-complete.

Definition 3.6 *Let SDS(ϵ) be the following decision problem: We are given $\epsilon > 0$, a positive integer l and a bipartite graph $G = (V, W, E)$ and a certificate equal to "YES" if G satisfies property EXP(ϵ), and "NO" otherwise.*

EXP(ϵ): if $X \subset V$ is a 1-defect set with $|X| \leq (1 + l^{-\epsilon})l$, then $|N(X)| = |X| - 1$ and $|X| \leq l$.

SDS(ϵ) outputs "YES" if the certificate equals "YES" and G has a 1-defect set with at most l vertices.

Proposition 3.7 *The decision problem SDS(ϵ) is NP-complete for every $\epsilon > 0$.*

Proof. Let $\epsilon > 0$ be given. Let $t = \lceil 2/\epsilon \rceil$. Let k be an integer satisfying $k \geq t^2$ such that k divides $\binom{k}{t}$. We give a transformation from the decision problem (k, x)-CLIQUE, which decides, given an instance consisting of an integer k, a graph G and a vertex x of G, whether G contains a clique of size k that contains x. Let such an instance G with vertex set V and edge set E be given. We construct a bipartite graph $H = (U, W, E')$ by letting U be the set of all subgraphs of G isomorphic to the complete graph K_t, and $W = \{(v, i) : v \in V \setminus \{x\}, 1 \leq i \leq \binom{k}{t}/k\} \cup \{(x, i) : 1 \leq i \leq \binom{k}{t}/k - 1\}$, and we join $u \in U$ and $(v, i) \in W$ by an edge if v is a vertex of u in G.

We claim that H satisfies the conditions for SDS(ϵ) with $l = \binom{k}{t}$, and that H contains a 1-defect set if and only if G contains a k-clique that contains x. Suppose that $X \subset U$ satisfies $|X| > |N(X)|$. Let $V[X]$ denote the set of vertices of G spanned by the K_t-subgraphs in X. Note that if $x \notin V[X]$ then the neighbourhood $N(X)$ of X in H consists of $\binom{k}{t}/k$ copies of $V[X]$, and if $x \in V[X]$ then $N(X)$ consists of $\binom{k}{t}/k$ copies of $V[X] \setminus \{x\}$ together with $\binom{k}{t}/k - 1$ copies of x. Hence $|N(X)| \geq |V[X]|\binom{k}{t}/k - 1$, with equality holding only if $x \in V[X]$. But since $|X| \leq \binom{|V[X]|}{t}$ we have $|V[X]| \geq k$. Now if X is the set of K_t-subgraphs of a k-clique that contains x then clearly $|N(X)| = \binom{k}{t} - 1 = |X| - 1$.

Now suppose that X satisfies $|X| > |N(X)|$ but it is not the set of K_t-subgraphs of a k-clique in G that contains x. Then we must have $|V[X]| \geq k+1$, and so $|N(X)| \geq (k+1)\binom{k}{t}/k - 1 \geq (1+1/k)l - 1$.

By the choice of t and k we have that $l = \binom{k}{t} \geq (k/t)^t \geq k^{t/2}$. Hence $l^\epsilon \geq k$, and so $|X| \geq (1+l^{-\epsilon})l$. Therefore H satisfies the required conditions.

□

The importance of the class UP lies in the fact that one-way functions exist if and only if $P \neq UP$ (see Welsh [19]). For the definition of one-way functions see *e.g.* Welsh [19]; loosely speaking, a function f is one-way if it is computable in polynomial time but it is difficult to solve equations of type $f(x) = y$ for given values of y. Hence a one-way function exists if minSDS is hard. This makes the fact that SDS(ϵ) is NP-complete more interesting.

4 Short Circuits of Matroids.

Let $G = (V, W, E)$ be a bipartite graph. The collection $\mathcal{T}$ of subsets S of V such that S does not have a 1-defect subset, is called a *transversal matroid*. The elements of $\mathcal{T}$ are called *independent sets* of the matroid. All the remaining subsets of V are called *dependent*. The minimal (with respect to inclusion) dependent sets are called *circuits*.

Let $\mathbf{F}$ be a field. We say that $\mathcal{T}$ is *representable* over $\mathbf{F}$ if there is a matrix of elements of $\mathbf{F}$ whose columns are indexed by V, and each $S \subset V$ is independent if and only if the columns indexed by S are linearly independent over $\mathbf{F}$.

Transversal matroids are representable over all sufficiently large fields, *i.e.* over fields with an arbitrary characteristic. The proof given in Oxley [12] is not constructive, however it is not hard to propose a natural construction.

Definition 4.1 *Let $G = (V, W, E)$ be a bipartite graph. Let x_e, $e \in E$, be distinct variables. We let $P(G)$ be the matrix whose columns are indexed by V and rows are indexed by W, and each entry (w, v) equals x_e if $e = wv$ is an edge of G, and 0 otherwise.*

If $S \subset V$ then let $P(G)_S$ denote the subset of the columns of $P(G)$ indexed by the elements of S.

A non-zero transversal of $P(G)_S$ is a set of $|S|$ non-zero entries of $P(G)$, no two in the same row, such that each column of $P(G)_S$ contains exactly one of them.

Proposition 4.2 *Let G be a bipartite graph and $S \subset V$. Then the following properties are equivalent.*

(1) S does not have a 1-defect subset,

(2) $P(G)_S$ has a non-zero transversal,

(3) $P(G)_S$ has a square submatrix A of full row-length such that, when A is considered as a matrix of different indeterminants over an arbitrary field $\mathbf{F}$, $det(A)$ is not identically equal to zero.

Proof. The conditions (2) and (3) are clearly equivalent. We also have immediately that (2) implies (1). Finally, if (2) does not hold then by Hall's theorem this means that a 1-defect subset of S exists.

□

We will use the following notation. If $\mathbf{F}$ is a field, we denote by $\mathbf{F}[\mathbf{x}]$ the field of polynomials of degree at most 1 with coefficients in $\mathbf{F}$, where the field operations are performed modulo an irreducible polynomial over $\mathbf{F}$ of degree 2.

Definition 4.3 *Let $\mathbf{F}$ be a field and let $G = (V, W, E)$ be a bipartite graph. We define the field $\mathbf{F}(\mathbf{G})$ as follows. Let $e_1, ..., e_m$ be an arbitrary total order of the edges of G. Then we let $\mathbf{F}(\mathbf{G}) = \mathbf{F}[\mathbf{x_{e_1}}][\mathbf{x_{e_2}}]...[\mathbf{x_{e_m}}]$.*

Proposition 4.4 *Let $\mathbf{F}$ be a field and let $G = (V, W, E)$ be a bipartite graph. Then $P(G)$ represents the transversal matroid defined by G on V, over the field $\mathbf{F}(\mathbf{G})$.*

Proof. Let A be a square submatrix of $P(G)$. We show A has a non-zero transversal if and only if $det(A) \neq 0$ in $\mathbf{F}(\mathbf{G})$. This proves the proposition by 4.2.

It is clear that if A has no non-zero transversal then $det(A) = 0$ in $\mathbf{F}(\mathbf{G})$. Hence assume A has a non-zero transversal. We will show by induction on the size of A that $det(A) \neq 0$ in $\mathbf{F}(\mathbf{G})$.

Let i be the highest index of an edge of G such that x_{e_i} appears in a non-zero transversal of A. Regard $det(A)$ as a polynomial of degree 1 with variable x_{e_i}. The coefficient of this polynomial at x_{e_i} is, up to a sign, the determinant of a submatrix A' of A and it is non-zero by the induction assumption, since A' has a non-zero transversal by the choice of x_{e_i}.

We conclude that $det(A)$ is non-zero as well.

□

Proposition 4.4 gives a construction of a representation of a transversal matroid over a sufficiently large extension of an arbitrary field, *i.e.* in particular over a field with an arbitrary characteristic.

The extension needed is very large unfortunately. Let $\mathbf{F} = GF(p)$, where p is a prime. The construction shows how to represent a transversal matroid given by a bipartite graph G over $GF(p^{2^m})$, where m is the number of edges of G. We do not know whether it is possible to construct efficiently a representation over $GF(p^{Q(m)})$, where Q is a polynomial.

Applying randomness one can nevertheless find such a representation.

We use the following well-known theorem (see Schwartz [15], Zippel [20]).

Theorem 4.5 *Let* $\mathbf{F}$ *be a field and let* $H(x_1, ..., x_n)$ *be a multivariate polynomial over* $\mathbf{F}$ *of total degree* d*, where the total degree of a polynomial is the maximum over its terms of the sum of the exponents of the variables. Fix any finite set* $S \subset \mathbf{F}$*, and let* $r_1, ..., r_n$ *be chosen independently and uniformly at random from* S*. Then*

$$\mathbf{Pr}[H(r_1, ..., r_n) = 0 | H(x_1, ..., x_n) \not\equiv 0] \leq d/|S|.$$

Theorem 4.6 *Let* $G = (V, W, E)$ *be a bipartite graph,* $|V| = n$ *and let* $\mathcal{T}$ *be the transversal matroid defined by* G*. Let* $\mathbf{F}$ *be a field with at least* 2^{2n} *elements and let* $R \subset \mathbf{F}$*,* $|R| = 2^{2n}$*. Let* $P(R)$ *be obtained from* $P(G)$ *be replacing each* x_{e_i} *by an element chosen independently and uniformly at random from* R*. Then the following two properties hold.*

(1) If a subset S *of columns of* $P(R)$ *is linearly independent over* $\mathbf{F}$ *then* $S \in \mathcal{T}$*,*

(2) The probability that "there exists a subset S *of* V *such that* $S \in \mathcal{T}$ *but the columns of* $P(R)$ *indexed by* S *are linearly dependent over* $\mathbf{F}$*" is at most* $n/2^n$*.*

Proof. If a subset of columns of $P(R)$ is linearly independent then it must have a square submatrix with non-zero determinant of full row-length, and hence also a non-zero transversal. Using Theorem 4.2 we get the first property.

Let S be an arbitrary subset of V and assume $S \in \mathcal{T}$. Then by 4.2 $P(G)_S$ has a submatrix A of full row-length such that, when A is considered as a matrix of different indeterminants over the field $\mathbf{F}$, $det(A)$ is not identically equal to zero.

The columns of $P(R)$ indexed by S are linearly dependent over $\mathbf{F}$ if and only if they do not contain a regular submatrix of full row-length.

Hence the probability that "the columns of $P(R)$ indexed by S are linearly dependent over $\mathbf{F}$" is at most the probability that submatrix A becomes singular after the substitution of random elements of R for the indeterminants, which is at most $|S|/2^{2n}$ by Theorem 4.5. Summing this over all subsets of V we get that property (2) holds.

$\square$

Theorem 4.6 gives a "very likely" representation of a transversal matroid on n elements over an arbitrary field with at least 2^{2n} elements. It has also some interesting algorithmic consequences.

The result of Blum et.al. [5] mentioned in the beginning of Section 3, that it is NP-complete to find out whether a bipartite graph has a 1-defect set of at most l vertices, has the following form for transversal matroids.

Theorem 4.7 *The problem to find out whether a transversal matroid has a circuit of at most l elements is NP-complete.*

□

Theorems 4.4 and 4.7 have the following immediate consequence.

Corollary 4.8 *Let K be a fixed characteristic. Then the following problem is NP-hard: Given a matrix A over field $\mathbf{F}$ of characteristic K, together with an oracle for the operations of addition and multiplication in $\mathbf{F}$, and an integer l, find out whether A has l linearly dependent columns over $\mathbf{F}$.*

Further consequences use the notions of random polynomial algorithms and PR-completeness. These definitions follow Vazirani and Vazirani [18] and Johnson [9].

Definition 4.9 *A random polynomial algorithm is a deterministic polynomial algorithm whose steps may depend on flips of an unbiased coin. A computation that results from a particular sequence of K coin flips has probability $1/2^K$.*

A random polynomial algorithm P solves a decision problem $\mathcal{D}$ if P is always correct when answering "NO" and wrong with a fixed probability independent of the input and less than 1 when answering "YES".

The class of decision problems that admit a random polynomial algorithm is denoted by RP.

Note that by running a random polynomial algorithm repeatedly the probability of an error decreases exponentially.

Definition 4.10 *A decision problem $\mathcal{A}$ is called random polynomial complete for NP (PR-complete) if there is an NP-complete problem $\mathcal{B}$ and a random polynomial algorithm $P(\mathcal{B})$ which solves $\mathcal{B}$, provided there is a random polynomial algorithm to solve $\mathcal{A}$.*

The random polynomial completeness of a problem for NP is convincing evidence that an efficient algorithm to solve it is unlikely. See Vazirani, Vazirani [18] for another problem of this type, and [9] for a discussion on random polynomial completeness.

Definition 4.11 *We let $Que(\mathbf{F}, A, k)$ be the following decision problem: Given a field $\mathbf{F}$, integers m, n, k, $k < n$, such that $|\mathbf{F}| > 2^{2n}$, and an $(m \times n)$ matrix A over $\mathbf{F}$, find out whether A has k columns linearly dependent over $\mathbf{F}$.*

We will show that $Que(\mathbf{F}, A, k)$ is random polynomial complete for NP, even when restricted to inputs where the field $\mathbf{F}$ is chosen from a fixed sequence of fields, whose sizes grow to infinity: an example is when $\mathbf{F}$ is the field of rationals always, or when $\mathbf{F} = GF(p^n)$ for some prime p and integer n.

Having these two examples in mind, we will not discuss here in detail the issue of how the fields $\mathbf{F}$ are given: for simplicity we assume from now on that addition and multiplication in $\mathbf{F}$ are given by an oracle, and the elements of $\mathbf{F}$ are numbers in binary notation.

Theorem 4.12 *$Que(\mathbf{F}, A, k)$ is random polynomial complete for NP, even when restricted to inputs where the field $\mathbf{F}$ is chosen from a fixed sequence of fields, whose sizes grow to infinity.*

Proof. Let ALQ be a random polynomial algorithm which solves Que. Without loss of generality assume that the probability of error when ALQ answers "YES" is at most 1/4. Using ALQ, we design a random polynomial algorithm ALT for the NP-complete problem to decide whether a transversal matroid has a circuit of at most k elements. (see 4.7).

Let $G = (V, W, E)$ be a bipartite graph defining the transversal matroid $\mathcal{T}$, let $|V| = n$ and $k < n$. Let $\mathbf{F}$ be a field of at least 2^{2n} elements, from the fixed set of fields.

Our algorithm ALT to find out whether $\mathcal{T}$ has a circuit of at most k elements is very simple: It constructs the matrix $P(R)$ as in the proof of Theorem 4.6 and then applies ALQ to $(\mathbf{F}, P(R), k)$.

It follows from 4.6 that ALT is a random polynomial algorithm.

$\square$

Khachiyan [11] proved that the problem discussed in Theorem 4.12 is NP-complete when $\mathbf{F}$ is the field of rationals. In fact, he can show that the problem to decide whether a set of rational vectors is in general position is NP-complete. Using essentially the same method as Khachiyan, Vardy [17] has proven recently that the problem discussed in Theorem 4.12 is NP-complete when $\mathbf{F} = GF(p^m)$ for an arbitrary fixed p. Vardy [17] has been able to use his result to answer affirmatively a longstanding open conjecture (see Berlekamp et.al. [4]) on whether finding the smallest cardinality of a circuit of a matroid represented over a finite field, particularly $GF(2)$, is an NP-complete task. This problem was known to be NP-complete if we restricted ourselves to circuits containing a given element (see Berlekamp et.al. [4]).

Theorem 4.12 and its proof have some interesting consequences.

Definition 4.13 *Let p be a fixed prime. Let MOD_p be the following decision problem: Given a rational matrix A and integers m and k, find out whether A has k columns which are linearly dependent modulo p^m.*

Definition 4.14 *Let c be a fixed positive integer. Let MOD^c be the following decision problem: Given a rational matrix A, prime q and integer k, find out whether A has k columns which are linearly dependent modulo q^c.*

Corollary 4.15 *Both MOD_p and MOD^c are PR-complete, for each c and $p > 1$.*

Proof. We proceed in the same way as in the proof of Theorem 4.12 for the field of rationals, but instead of applying algorithm ALQ we apply a random polynomial algorithm which solves MOD_p or MOD^c for the matrix $P(R)$, k and $m = n^2 + 2n + 1$ or $q > 2^{n^2+2n}$ respectively.

By the choice of $P(R)$, the absolute value of each of its subdeterminants is at most 2^{n^2+2n}. Hence the counting modulo p^m or q^c respectively is the same as counting in the rationals as far as the subdeterminants of $P(R)$ are concerned.

□

References

[1] A. Aggarwal, A. Bar-Noy, D. Coppersmith, R. Ramaswami, B. Schieber, M. Sudan, *Efficient Routing and Scheduling Algorithms for Optical Networks*, in Proceedings of the Fifth Annual ACM-SIAM Symposium on Discrete Algorithms, Philadelphia, Pennsylvania, 23-25 January 1994, pp. 412-423.

[2] M. Ajtai, *Generating Hard Instances of Lattice Problems*, preprint 1996.

[3] I. Barany, S. Onn, *Colourful Linear Programming And its Relatives*, Mathematics of Operations Research, to appear.

[4] E.R. Berlekamp et. al., *On the Inherent Intractability of Certain Coding Problems*, IEEE Transactions on Information Theory 24 (1978), pp.384-386.

[5] M. Blum, R.M. Karp, O. Vornberger, C.H. Papadimitriou, M. Yannakakis, *The Complexity of Testing Whether a Graph Is a Superconcentrator*, Information Processing Letters 13 (1981).

[6] P. Feldman, J. Friedman, N. Pippenger, *Wide-Sense Nonblocking Networks*, SIAM J. Disc. Math. 1 (1988).

[7] J. Friedman, N. Pippenger, *Expanding Graphs Contain All Small Trees*, Combinatorica 7 (1) (1987), pp.71-76.

[8] P.E. Haxell, Y. Kohayakawa, *The Size-Ramsey Number of Trees*, Israel J. of Mathematics 89 (1995), pp. 261-274.

[9] D.S. Johnson, *The NP-Completeness Column: An Ongoing Guide*, J. of Algorithms 5 (1984), pp. 433-447.

[10] D.S. Johnson, *A Catalog of Complexity Classes*, in Handbook of Theoretical Computer Science, Vol. A (J. Van Leeuwen, ed.), Elsevier 1990, pp. 67-162.

[11] L. Khachiyan, *On the Complexity of Approximating Extremal Determinants in Matrices*, J. of Complexity 11 (1995), pp.138-153.

[12] J. Oxley, *Matroid Theory*, Oxford University Press, 1992.

[13] C.H. Papadimitriou, M. Yannakakis, *The Complexity of Facets (And Some Facets of Complexity)*, J. Comput. System Sci. 28 (1984), pp.244-259.

[14] A. Schrijver, *Theory of Integer and Linear Programming*, Wiley, Chichester, 1986.

[15] J.T. Schwartz, *Fast Probabilistic Algorithms For Verification of Polynomial Identities*, Journal of the ACM 27(1980), pp.701-717.

[16] L.G. Valiant, V. Vazirani, *NP Is As Easy As Detecting Unique Solutions*, Theoretical Computer Science 47 (1986), pp.85-93.

[17] A. Vardy, *The Intractability of Computing the Minimum Distance of a Code*, manuscript November 1996.

[18] U.V. Vazirani, V.V. Vazirani, *A Natural Encoding Scheme Proved Probabilistic Polynomial Complete*, Theoretical Computer Science 24 (1983), pp. 291-300.

[19] D.J.A. Welsh, *Complexity: Knots, Colourings and Counting*, Cambridge University Press, 1993.

[20] R.E. Zippel, *Probabilistic Algorithms for Sparse Polynomials*, In Proceedings of EUROSAM 79, volume 72 of Lecture Notes in Computer Science, (1979), pp. 216-226.