Hypercube problems

Lecture 15

January 4, 2017

Lecturer: Petr Gregor | Scribe by: Miloš Chromý | Updated: January 26, 2017

This lecture is based on an article "A Survey of Binary Covering Arrays" [LKL+11].

1 Vertex Turán number and other related problems

1.1 Turán number

For graphs G, H the **Turán number of H in G**, $\boldsymbol{ex(G,H)}$, is the maximal number of edges in a subgraph of G not containing H. Formally we can write

$$ex(G,H) = \max_{H \not\subseteq S \subseteq G} |E(S)|. \tag{1}$$

For $G = Q_n$ and $H = Q_2$ (a 4-cycle) there is the following conjecture.

Conjecture 1 (Erdős) $ex(Q_n, Q_2) = (\frac{1}{2} + o(1))|E(Q_n)|$.

We can see that in a hypercube a 4-cycle has to cover three subsequent levels. So if we take all edges between levels $2i$ and $2i+1$ for every $0 \le 2i < n$ we get $ex(Q_n, Q_2) \ge \frac{1}{2}|E(Q_n)|$. Furthermore in the remaining layers we can take some additional edges. Good candidates are edges that form a maximal matching between levels $2i-1$ and $2i$ for every $2 \le 2i \le n$. Number of such edges is in $o(E(Q_n))$ and we get $ex(Q_n, Q_2) \ge (\frac{1}{2} + o(1))|E(Q_n)|$.

1.2 Vertex Turán number, s-face transversal, (binary) covering array

Our interest will be in vertex Turán numbers for hypercubes and two other problems, which can be easily transformed one to the other.

Definition 1 *Let $s \le k$ be integers. The **vertex Turán number**, $\boldsymbol{ex_v(Q_k, Q_s)}$, is the maximal number of vertices in an induced subgraph of Q_k not containing Q_s as a subgraph, i.e.*

$$ex_v(Q_k, Q_s) = \max_{\substack{A \subseteq V(Q_k) \\ Q_s \not\subseteq Q_k[A]}} |A|. \tag{2}$$

Definition 2 ([CKMZ83]) *A set $S \subseteq V(Q_k)$ is a **s-face transversal** (shortly s-transversal, a.k.a. a "piercing set") if S contains some vertex from each s-dimensional subcube (s-face), i.e. $Q_k - S$ has no s-face. Let $\boldsymbol{tr(k,s)}$ be the minimal size of **s-transversal** in Q_k for $0 \le s \le k$, i.e.*

$$tr(k,s) = \min_{\substack{S \subseteq V(Q_k) \\ S \text{ is } s\text{-face transversal}}} |S| \tag{3}$$

Definition 3 *A* ***(binary)*** $\boldsymbol{n \times k}$ ***covering array of strength*** $\boldsymbol{t}$ *(an* ***(n,k,t)-array****) is a (binary) $n \times k$ matrix such that the projection into any t columns gives $n \times t$ submatrix containing all 2^t possible rows. Let* $\textsc{can}\boldsymbol{(k,t)}$ *be the minimal number of rows in an $n \times k$ covering array of strength t for $0 \leq t \leq k$.*

Observation 2 *For any $0 \leq s \leq k$*

$$2^k - ex_v(Q_k, Q_s) = tr(k,s) = \textsc{can}(k, k-s). \tag{4}$$

So as we can see, the Definitions 1,2, and 3 are only different formulations of the same problem.

2 Application

2.1 Fault-tolerance of hypercube computers [BS88, GHLS93]

If we need to run some task on a subcube Q_s for our solution and we have a hypercube computer Q_k, $tr(k,s) = \textsc{can}(k, k-s)$ tells us how many faulty vertices in the worst case we can tolerate until we will not be able to run the computation.

"Good news:" $\textsc{can}(k, k-s)$ grows as fast as 2^k for s fixed. More precisely,

$$\frac{\textsc{can}(k, k-s)}{2^k} \leq \frac{1}{s+1}$$

and is this fraction is non-decreasing, which we will show in Section 3.4.

2.2 Software testing

We have an algorithm $A(x_1, \ldots, x_k)$ with k binary inputs and we assume that bugs occur in interaction of small number of input parameters. Therefore we can test algorithm A for all interactions between any t parameters $\Rightarrow$ straightforward approach will need $\binom{k}{t}2^t$ runs. But we need only $\textsc{can}(k,t)$ runs. Each row of our (n,k,t)-array is one input for tests of our algorithm and by Definition 3 for every choice of t parameters (columns) we have all 2^t possible values which we need to test.

"Good news:" $\textsc{can}(k,t)$ grows as slow as $\log_2 k$ for t fixed. More precisely,

$$\textsc{can}(k,t) \leq \frac{t}{\log_2(\frac{2^t}{2^t-1})} \log_2 k,$$

which we will show in Section 3.4, Corollary 11.

2.3 Other

There are also other applications, e.g. synchronization of blind dyslectic robots on a line [Gal99, Har05, LA96], ...

3 Results

Next, we explore some properties of $\textsc{can}(k,t)$. First, we show some basic properties and how $\textsc{can}(k,t)$ behaves for small t or small $k-t$. Then we describe recursive constructions of covering arrays and finally we show some asymptotic estimations of this function. Most of this part is based on [LKL+11], Chapter 3.

$s\backslash t$	1	2	3	4	5	6
0	2	4	8	16	32	64
1	2	4	8	16	32	64
2	2	5	10	21	42	85
3	2	6	12	24	48–52	96–108
4	2	6	12	24	48–54	96–116
5	2	6	12	24	48–56	96–118
6	2	6	12	24	48–64	96–128
7	2	6	12	24	48–64	96–128
8	2	6	12	24	48–64	96–128
9	2	7	15	30–32	60–64	120–128
10	2	7	15–16	30–35	60–79	120–179

Table 1: Values of $\mathrm{CAN}(s+t,t)$.

3.1 Basic properties

First, we explore some basic properties. For every $k \geq t \geq 1$,

1. $\textsc{can}(k+1,t) \geq \textsc{can}(k,t)$
   Any projection of an $(n,k+1,t)$-array to t columns has to contain all 2^t possible combinations in rows, so it holds that every projection of the first k columns of an $(n,k+1,t)$-array to t columns contains all 2^t combinations in rows. Hence from the first k columns we get an (n,k,t)-array.

2. $\textsc{can}(k+1,t+1) \geq 2\textsc{can}(k,t)$
   Any $(k-t)$-transversal in Q_{k+1} consists of two disjoint $(k-t)$-transversals in Q_k^0, Q_k^1.

3. $\textsc{can}(k,t) \geq 2^t$
   All 2^t vectors must appear in projection into any t columns.

3.2 Exact results for small numbers

Now we will show values of $\textsc{can}(k,t)$ for small t and $k-t$.

- $\textsc{can}(k,1) = 2$ for every $k \geq 1$
  Let us have a matrix

$$A = \overbrace{\begin{pmatrix} 0 & 0 & \dots & 0 \\ 1 & 1 & \dots & 1 \end{pmatrix}}^{k}.$$

We can see that any projection of A to a single column contains 0 and 1. Therefore A is a $2 \times k$ array of strength 1.

- $\textsc{can}(k,k) = 2^k$ for every $k \geq 1$.
  We need all possible 2^k combinations in k columns.

- $\textsc{can}(k,k-1) = 2^{k-1}$ for every $k \geq 2$.
  From Observation 2 we have $\textsc{can}(k,k-1) = tr(k,1)$. 1-face is an edge. We can take the bipartition of Q_k, i.e. the set S_e containing all vertices from even levels and the set S_o containing remaining vertices. Both these sets are 1-transversals and $|S_e| = |S_o| = 2^{k-1}$.

- $\textsc{can}(k,k-2) = \left\lfloor \frac{2^k}{3} \right\rfloor$ for every $k \geq 3$.
  As in previous example, we know that $\textsc{can}(k,k-2) = tr(k,2)$. We consider three sets S_0, S_1, S_2: each set S_i consists of all vertices of levels equal to $i \mod 3$. Each of the sets S_0, S_1, S_2 is a 2-transversal since any Q_2 (4-cycle) meets 3 consecutive levels. At least one set has size
  $$S_i \leq \left\lfloor \frac{2^k}{3} \right\rfloor.$$

  The lower bound was shown by Johnson and Entringer [JE89].

  Observation 3 ([TC84]) $\textsc{can}(k,s) \leq \frac{2^k}{s+1}$ *for every* $0 \leq s \leq k$.

  This proof is similar to the preceding argumentation.

  Proof We use the s-transversal definition (Observation 2). Any subcube Q_s has to go through $s+1$ consecutive levels of Q_k so if we partition the cube Q_k into sets $S_j = \bigcup_{i \equiv j \mod (s+1)} X^{(i)}$ for $0 \leq j \leq s$, where $X^{(i)}$ is a set of vertices at level i, we obtain s-transversal sets S_j for every $0 \leq j \leq s$. As $S_0, \ldots, S_s$ partition Q_k, there is an s-transversal of size at most $\lfloor \frac{2^k}{s+1} \rfloor$. ■

- $\textsc{can}(k,2) = n$ where n is the least positive integer such that $\binom{n-1}{\lceil \frac{n}{2} \rceil} \geq k$

  Proof Let us have a matrix $A \in \{0,1\}^{n \times k}$ with $k \leq \binom{n-1}{\lceil \frac{n}{2} \rceil}$

$$A = \begin{pmatrix} 0 & 0 & \cdots & 0 \\ \hline 1 & 0 & & \\ \vdots & \vdots & & \\ 1 & 0 & & \\ 1 & 1 & \cdots & \\ 0 & 1 & & \\ \vdots & \vdots & & \\ 0 & 1 & & \end{pmatrix} \begin{matrix} \\ \left.\begin{matrix} \\ \\ \\ \\ \\ \\ \\ \end{matrix}\right\} n-1 \end{matrix}$$
$$\underbrace{\qquad\qquad\qquad\qquad}_{k}$$

where each column in a red $(n-1) \times k$ submatrix has exactly $\lceil \frac{n}{2} \rceil$ ones and any two columns are distinct. Such matrix A exists since $k \leq \binom{n-1}{\lceil \frac{n}{2} \rceil}$. If we take any two columns, we have 00 from the first row and we have 11 in some row because we have more ones than zeros. Any two columns are distinct, so we have at least one point, where there is 01 or 10. We have a fixed number of 1's and 0's in each column, hence we get the other combination 10 or 01. So we have an $(n, k, 2)$-array, thus the upper bound $\textsc{can}(k, 2) \leq n$ holds.

To show the lower bound, first we consider $\boldsymbol{n}$ **even**. We will need Sperner's lemma [Spe28].

Lemma 4 (Sperner [Spe28]) *The maximal size of an independent family of sets over n-element set is* $\binom{n}{\lfloor \frac{n}{2} \rfloor}$.

Let $A_1, \ldots, A_k \subseteq [n]$ be sets with characteristic vectors in columns of an optimal $(n \times k)$ array of strength 2. Then $\mathcal{A} = \{A_i, \overline{A_i} | i \in [k]\}$ is an **independent family** of size $2k$. By Lemma 4,

$$2k \leq \binom{n}{\frac{n}{2}}.$$

Thus

$$k \leq \frac{1}{2}\binom{n}{\frac{n}{2}} = \binom{n-1}{\lceil \frac{n}{2} \rceil}.$$

For $\boldsymbol{n}$ **odd** from $\mathcal{A}$ we can take a subfamily $\mathcal{B} \subseteq \mathcal{A}$ of size k such that each $B \in \mathcal{B}$ has $|B| < \frac{n}{2}$. Furthermore, $\mathcal{B}$ is **intersecting** (for every $B, B' \in \mathcal{B} : B \cap B' \neq \emptyset$) and it can be shown that we may assume $|B| = \frac{n-1}{2}$ for every $B \in \mathcal{B}$. Thus,

$$k \leq \binom{n-1}{\frac{n-1}{2} - 1} = \binom{n-1}{\lceil \frac{n}{2} \rceil}$$

by Erdös-Ko-Rado theorem [EKR61] (recall Lecture 12). ■

3.3 Recursive constructions

Now we show how we can construct (n, k, t)-arrays recursively. First we show a construction using an (n, k, t)-array and an $(n, k, t-1)$-array. Then we generalize this approach to constructions with $(n, k, t-i)$-array, for $i = 1, \ldots, l$, $l \leq t$. Then we will describe some alternative approaches.

Labeling approach.

Proposition 5 *For every* $k \geq t \geq 1$,

$$\textsc{can}(k+1, t) \leq \textsc{can}(k, t) + \textsc{can}(k, t-1).$$

Proof Let A be an (n_1, k, t)-array, B be an $(n_2, k, t-1)$ array, and let

$$C = \begin{pmatrix} A & 0 \\ B & 1 \end{pmatrix}.$$

By taking t of the first $k-1$ columns, we have all 2^t possible rows from the array A. By taking $t-1$ of the first columns and the last one, we have all possible 2^{t-1} rows concatenated with 0 from the first n_1 rows and all possible 2^{t-1} rows concatenated with 1 from the last n_2 rows (B is an $(n_2, k, t-1)$-array). Hence C is an $(n_1+n_2, k+1, t)$-array. ∎

Proposition 6 *For every $k \geq t \geq 2$,*

$$\text{CAN}(k+2, t) \leq \text{CAN}(k, t) + 2\text{CAN}(k, t-1) + \text{CAN}(k, t-2).$$

Proof Let A be an (n_1, k, t)-array, B be an $(n_2, k, t-1)$-array, C be an $(n_3, k, t-2)$ array, and let

$$D = \begin{pmatrix} A & 0 & 0 \\ B & 0 & 1 \\ B & 1 & 0 \\ C & 1 & 1 \end{pmatrix}.$$

Using similar approach as one in the proof of Proposition 5, we obtain, that D is an $(n_1 + 2n_2 + n_3, k+2, t)$-array. ∎

Now we generalize this approach. First we have to define a **good labeling** of hypercube vertices.

Definition 4 *A labeling $l : V(Q_d) \to \{0, \ldots, d\}$ is* ***good*** *if every s-subcube Q_s of Q_d contains a vertex v with label $l(v) \leq d - s$ for every $s \in \{0, \ldots, d\}$.*

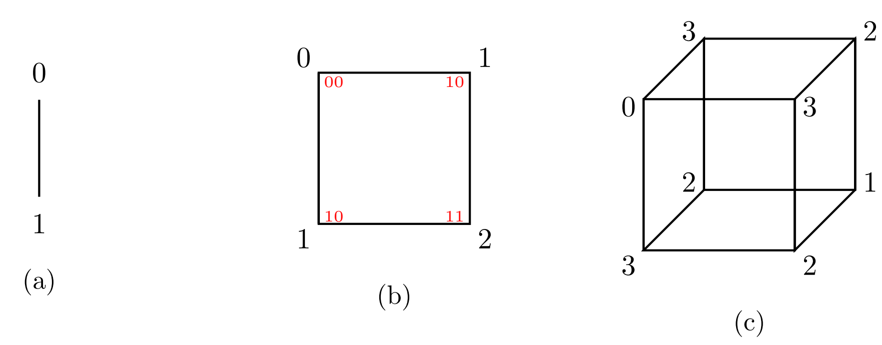

Figure 1: Examples of good labeling of Q_1,Q_2 (with red suffixes) and Q_3.

In Figure 1 we can see some examples of good labelings. For Q_2 on Figure 1(b) we have for each label the corresponding suffix next to labeled vertex. This suffix we use to construct an $(n, k+2, t)$-array in the proof of the Proposition 6.

Theorem 7 *Let $l : V(Q_d) \to \{0, \ldots, d\}$ be a good labeling. Then*

$$\textsc{can}(k+d, t) \leq \sum_{u \in V(Q_d)} \textsc{can}(k, t - l(u)) \tag{5}$$

For details of the proof we refer to [GHLS93]. Now we will show two alternative recursive constructions.

Proposition 8 *For every $k \geq t \geq 2$,*

$$\textsc{can}(k+1, t) \leq \textsc{can}(k, t) + 2\textsc{can}(k-1, t-2).$$

Proof Let $A = A' a_{k-1}$ be an (n_1, k, t)-array, let B be an $(n_2, k-1, t-2)$-array. Then

$$C = \begin{pmatrix} A' & a_{k-1} & a_{k-1} \\ B & 0 & 1 \\ B & 1 & 0 \end{pmatrix}$$

is an $(n_1 + 2n_2, k+1, t)$-array, using similar argument as before. ■

Proposition 9 *For every $k \geq 1$,*

$$\textsc{can}(2k, 3) \leq \textsc{can}(k, 3) + \textsc{can}(k, 2).$$

Proof Let A be an $(n_1, k, 3)$-array, B be an $(n_2, k, 2)$-array, and let

$$C = \begin{pmatrix} A & A \\ B & \overline{B} \end{pmatrix}.$$

We have following cases of choosing three columns of C.

1. All three columns are in the same half of C. Then we are done since A is an array of strength 3.

2. Two columns are corresponding to columns from one half and the third one is from the other. Let us have three columns a, b, c and without loss of generosity assume $0 \leq a < b < k$ (the first half of C) and $k \leq c < 2k$ (the second half of C). Here we have two possibilities.

   - $c - k \neq a$ and $c - k \neq b$. Here we are done, because these columns corresponds to three different columns in matrix A, that is an array of strength 3.
   - $c - k = a$ or $c - k = b$. Without loss of generality we can assume that $c - k = b$. From the first n_1 rows and columns a and b we will get all four combinations of rows and because b and c are equal on n_1 rows we have row *00 and *11 from first n_1 rows. For the last n_2 rows, we get all 4 combinations of rows from columns a and b (B is an array of strength 2). Values of last n_2 rows in columns b and c are not equal, and from that we will get all remaining combinations of rows *10 and *10. Hence we have an array of strength 3.

Therefore C is an $(n_1 + n_2, 2k, 3)$-array. ■

3.4 Asymptotic behaviour

Now we show asymptotic behaviour of $\textsc{can}(k,t)$ for fixed t and for fixed $s = k - t$. First, lets describe the case in which $\boldsymbol{s = k - t}$ is **fixed**.

By $\textsc{can}(k+1, t+1) \geq 2\textsc{can}(k,t)$ (Section 3.1, 2. inequality), the function $\frac{\textsc{can}(k,k-s)}{2^k}$ is non-decreasing, and by $\textsc{can}(k, k-s) \leq \frac{2^k}{s+1}$ (Observation 3) it is bounded, so $\lim\limits_{k\to\infty} \frac{\textsc{can}(k,k-s)}{2^k}$ exists.

Problem 1 *What is the value of* $\lim\limits_{k\to\infty} \frac{\textsc{can}(k,k-s)}{2^k}$ *?*

It could be true that it is $\frac{2^k}{s+1}$. Next we explore the case when $\boldsymbol{t}$ is **fixed**.

Observation 10 *Let A be a random binary $(n \times k)$ matrix with entries chosen independently with equal probability of 0 and 1. Then*

$$Pr[A \textit{ is not an } (n,k,t)\textit{-array}] \leq 2^t \binom{k}{t} \left(1 - \frac{1}{2^t}\right)^n . \tag{6}$$

Proof Probability that a fixed t-tuple does not occur in given t columns is $\left(1 - \frac{1}{2^t}\right)^n$. There are 2^t such t-tuples and $\binom{k}{t}$ choices for t columns. ■

By choosing n large enough so that probability (6) is less then 1 we obtain

Corollary 11 $\textsc{can}(k,t) \leq \frac{t}{\log_2\left(\frac{2^t}{2^t - 1}\right)}$

On the other hand, it can be shown [KS73] that

$$\left(\frac{t-2}{H(\frac{1}{2^{t-1}}) - \frac{1}{2^{t-2}}} - o(1)\right) \log_2 k \leq \textsc{can}(k,t),$$

where $H(\alpha) = -(\alpha \log_2 \alpha + (1-\alpha)\log_2(1-\alpha))$ is the (binary) entropy function.

Problem 2 *Does* $\lim\limits_{k\to\infty} \frac{\textsc{can}(k,t)}{\log_2 k}$ *exists?*

References

[BS88] B. Becker and H.-U. Simon. How robust is the n-cube? *Information and Computation*, 77(2):162 – 178, 1988.

[CKMZ83] A. K. Chandra, L. T. Kou, G. Markowsky, and S. Zaks. On sets of boolean n-vectors with all k-projections surjective. *Acta Informatica*, 20(1):103–111, 1983.

[EKR61] P. Erdős, C. Ko, and R. Rado. Intersection theorems for systems of finite sets. *The Quarterly Journal of Mathematics*, 12:313–320, 1961.

[Gal99] S. Gal. Rendezvous search on a line. *Oper. Res. 47*, pages 974–976, 1999.

[GHLS93] N. Graham, F. Harary, M. Livingston, and Q.F. Stout. Subcube fault-tolerance in hypercubes. *Information and Computation*, 102(2):280 – 314, 1993.

[Har05] A. Hartman. *Software and Hardware Testing Using Combinatorial Covering Suites*, pages 237–266. Springer US, Boston, MA, 2005.

[JE89] K. A. Johnson and R. Entringer. Largest induced subgraphs of the n-cube that contain no 4-cycles. *Journal of Combinatorial Theory, Series B*, 46(3):346 – 355, 1989.

[KS73] D. J. Kleitman and J. Spencer. Families of k-independent sets. *Discrete Mathematics*, 6(3):255 – 262, 1973.

[LA96] W. S. Lim and S. Alpern. Minimax rendezvous on the line. *SIAM Journal on Control and Optimization*, 34(5):1650–1665, 1996.

[LKL+11] J. Lawrence, R. N. Kacker, Y. Lei, D. R. Kuhn, and M. Forbes. A survey of binary covering arrays. *Electronic Journal of Combinatorics*, 18, 2011.

[Spe28] E. Sperner. Ein satz über untermengen einer endlichen menge. *Mathematische Zeitschrift*, 27:544–548, 1928.

[TC84] D. T. Tang and C. L. Chen. Iterative exhaustive pattern generation for logic testing. *IBM J. Res. Dev.*, 28(2):212–219, March 1984.