Mixed Covering Arrays on Graphs

Presenter:

Latifa Zekaoui

Joint work with Karen Meagher and Lucia Moura
to appear in the Journal of Combinatorial Designs

School of Information and Technology
University of Ottawa

Outline

- Covering Arrays
- Applications
- Covering Arrays on Graphs
- Mixed Covering Arrays on Graphs
- Graph Homomorphisms
- Mixed Qualitative Independence Graph
- ***n***-chromatic Graphs for n = 2,3,4,5
- Graph Operations: Tree & Cycle Construction
- Bipartite Graph Construction

Covering Array

A ***Covering Array,*** denoted **CA(*N*, k, g),** is a **k x *N*** array with:

- entries from Z_g (g is the alphabet)
- and between any two rows any pair from Z_g occurs in some column.

(Pairs of rows satisfying this property are said to be ***Qualitatively Independent***.)

CAN(k, g) is the smallest ***N*** such that a **CA(*N*, k, g)** exists.

Test light wiring in your home: 0 => OFF, 1 => ON

room \ test:	**1**	**2**	**3**	**4**	**5**
bedroom	0	1	1	1	0
hall	0	1	1	0	1
bathroom	0	1	0	1	1
kitchen	0	0	1	1	1

Example of an optimal CA: CAN(4, 2) = 5.

Applications

Covering Arrays are used in:

- **Circuit Testing** (Boroday and Grunskii, 1992)
- **Network Testing** (Williams and Probert, 1996)
- **Software Testing** (Cohen, Dalal, Fredman and Patton, 1997; Cheng, Dumitrescu and Schroeder, 2003)

Software Testing Application

Software Testing:

- Test parameters individually.
- But faults generally result from the interaction between certain parameters.
- Test every possible combination of parameter values. g^k GROWS EXPONENTIALLY!
- Enormous test suite sizes even for a small number of parameters.
- Covering arrays are used to test interaction between every pair of parameters.

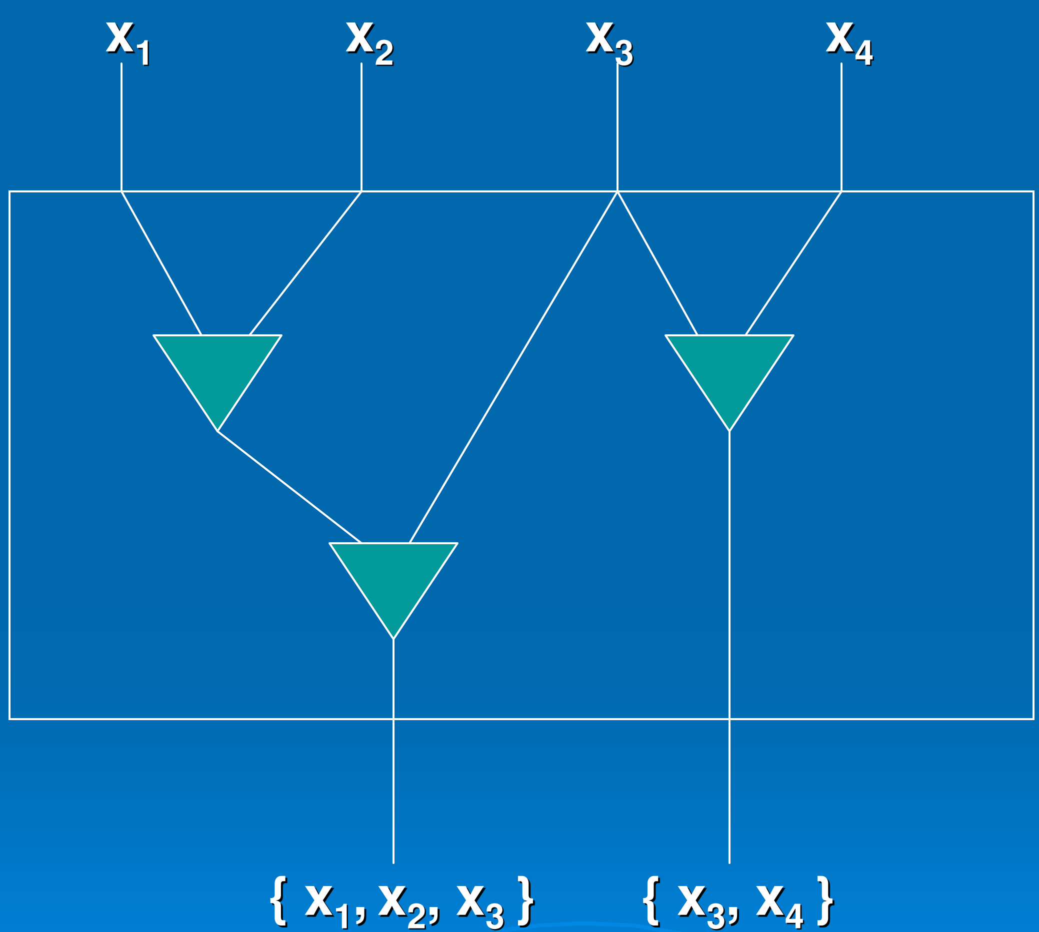
Example: 10 parameters with 4 input values

k = 10, g = 4, All possible interactions => 4^{10} = 1,048,576 test suites.
k = 10, g = 4, All pair-wise interactions => 29 test suites using a covering array.

Asymptotic result (Gargano, Korner and Vaccaro, 1990)

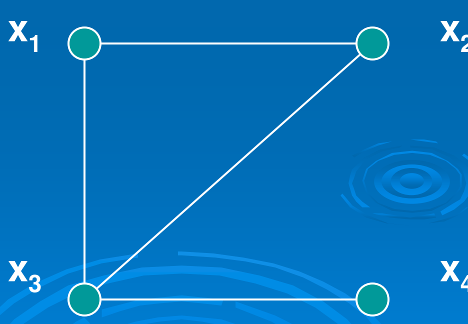
$$\lim_{k \to \infty} \mathrm{CAN}(g, k) = \frac{g}{2} \log k$$

Circuit Testing Application



We do not need to test the interaction between {x_1, x_4} or {x_2, x_4} .

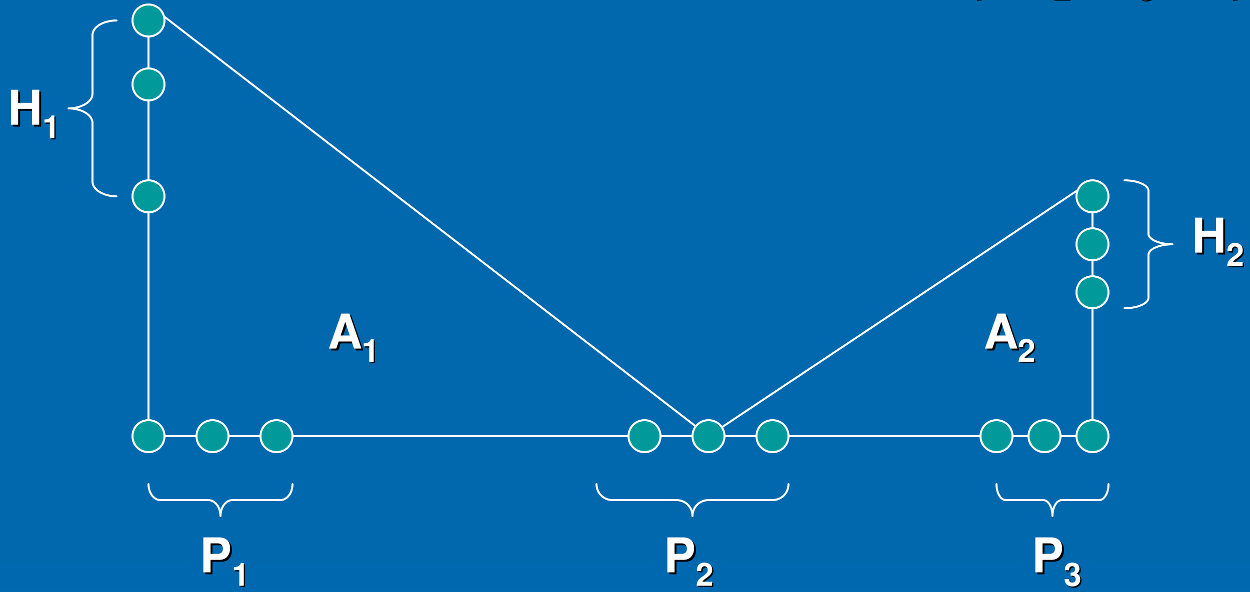
Build this graph:



Build a CA on the above graph.

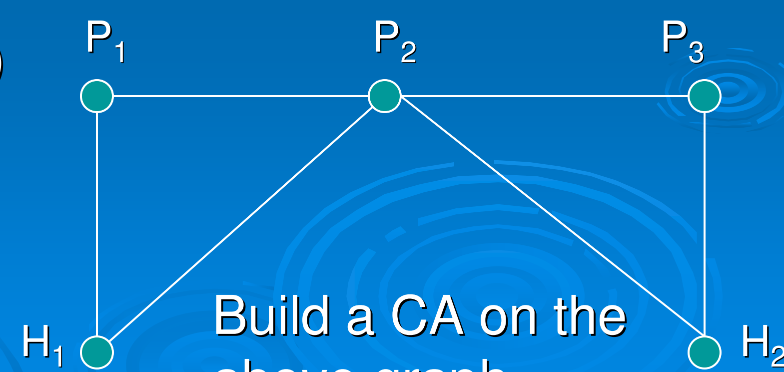
Software Testing: Relevant Interactions

Find the area of two triangles given P_1, P_2, P_3, H_1, H_2 each with 3 values.



We do not need to test the interaction between $\{P_1, P_3\}$, $\{P_1, H_2\}$, $\{P_3, H_1\}$, or $\{H_1, H_2\}$.

Build this graph:




```
calculateTriangleArea(P1, P2, P3, H1, H2)
{
        A1 = 0.5 * (P2 – P1) * H1;
        A2 = 0.5 * (P3 – P2) * H2;
return (A1, A2)
}
```

Build a CA on the above graph.

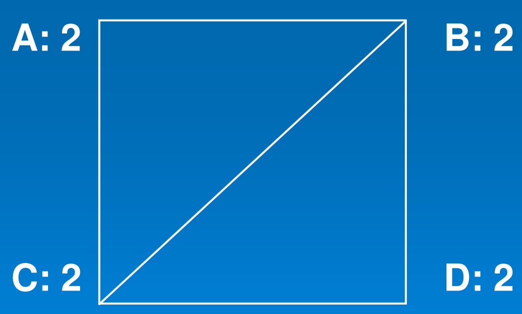
Covering Arrays on Graphs

A ***Covering Array on a Weighted Graph G,*** denoted **CA(*N*, *G*, g),** is a

- k x ***N*** array where k = |V(***G***)|
- with entries from Z_g (g is the alphabet and weight on vertices)
- rows for adjacent vertices are qualitatively independent

CAN(*G* , g) is the smallest ***N*** such that a CA(***N***, ***G***, g) exists.

Test wiring in your home:



Graph *G*

room \ test:	1	2	3	4
A: bedroom	0	0	1	1
B: hall	0	1	0	1
C: bathroom	0	1	1	0
D: kitchen	0	0	1	1

Example of an optimal CA: CAN(*G*, 2) = 4.

Covering Array

A ***Covering Array***, denoted **CA(*N*, k, g)**, is a **k x *N*** array with:

- entries from Z_g (g is the alphabet)
- and between any two rows any pair from Z_g occurs in some column.

(Pairs of rows satisfying this property are said to be ***Qualitatively Independent***.)

CAN(k, g) is the smallest ***N*** such that a **CA(*N*, k, g)** exists.

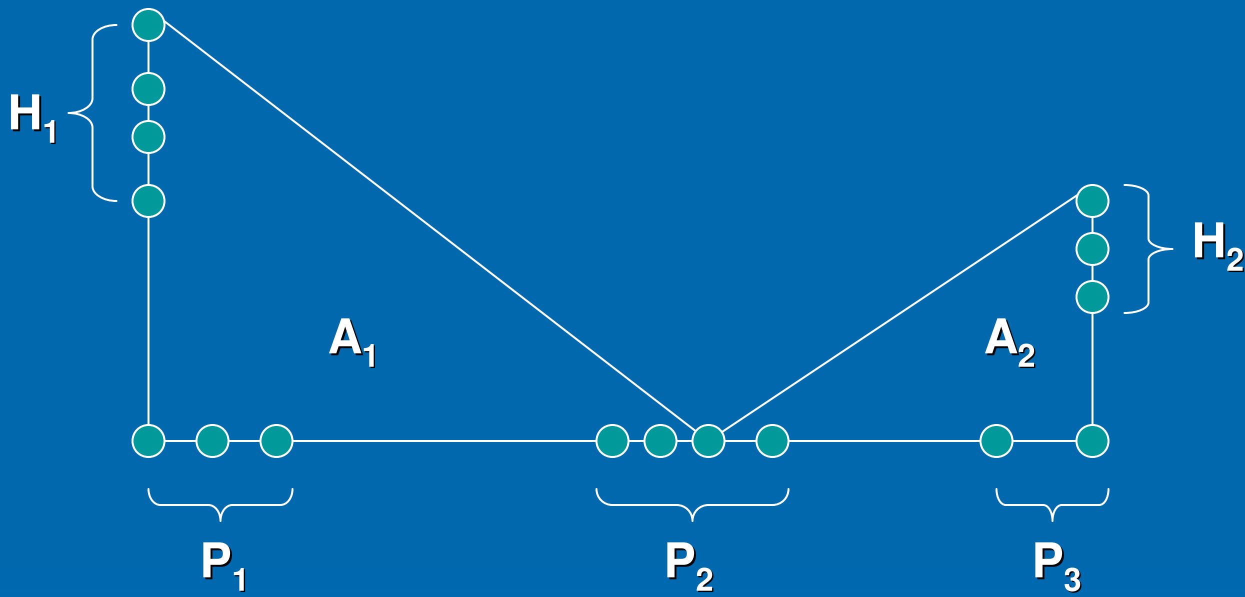
Test light wiring in your home: 0 => OFF, 1 => ON

room \ test:	1	2	3	4	5
bedroom	0	1	1	1	0
hall	0	1	1	0	1
bathroom	0	1	0	1	1
kitchen	0	0	1	1	1

Example of an optimal CA: CAN(4, 2) = 5.

Software Testing: Mixed Case

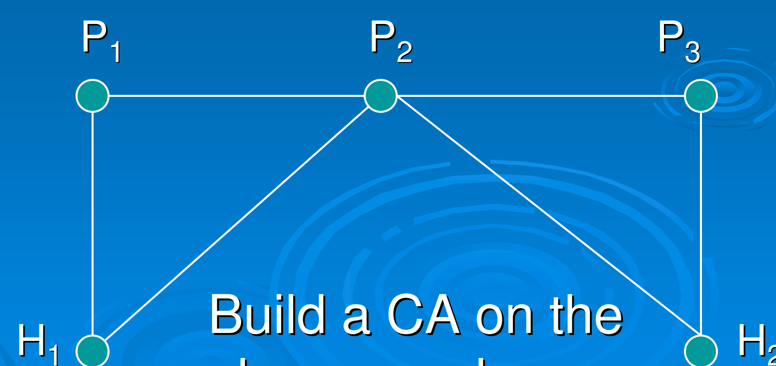
Find the area of two triangles given P_1, P_2, P_3, H_1, H_2 each with a different number of values.



We do not need to test the interaction between $\{P_1, P_3\}$, $\{P_1, H_2\}$, $\{P_3, H_1\}$, or $\{H_1, H_2\}$.

Build this graph:

```
calculateTriangleArea(P1, P2, P3, H1, H2)
{
        A1 = 0.5 * (P2 – P1) * H1;
        A2 = 0.5 * (P3 – P2) * H2;
return (A1, A2)
}
```



Build a CA on the above graph.

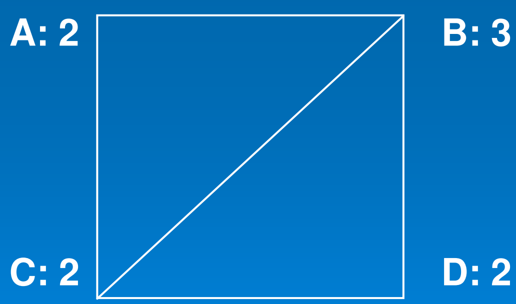
Mixed Covering Arrays on Graphs

A ***Mixed Covering Array on a weighted Graph*** *G*, denoted by **CA(N, *G*, $g_1g_2…g_k$)**, has mixed alphabet sizes for different rows in the graph.

The ***Product Weight*** of a graph ***G***, denoted PW(***G***), is

$$PW(G) = \max \{w_G(u) * w_G(v) : \{u,v\} \in E(G)\}.$$

$CAN(G, g_1,g_2,…, g_k) \geq PW(G)$



Graph *G*

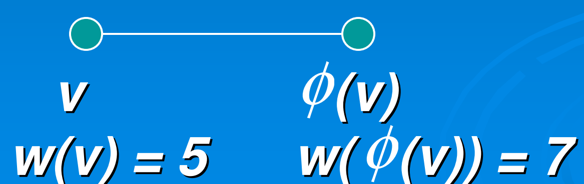
room \ test:	1	2	3	4	5	6
A: bedroom	0	1	0	1	0	1
B: hall	0	0	1	1	2	2
C: bathroom	0	1	1	0	1	0
D: kitchen	0	1	1	0	0	1

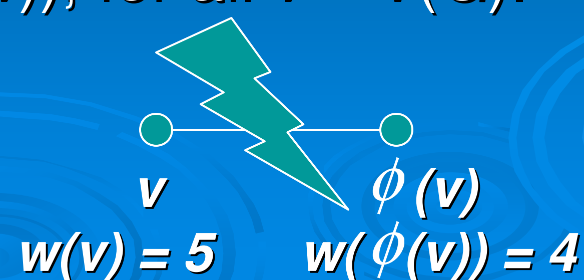
Example of an optimal CA: CAN(*G*, $2^3 3$) = 6.

Graph Homomorphisms

A mapping ϕ from V(***G***) to V(***H***) is a ***graph homomorphism*** from ***G*** to ***H*** if for all $v, w \in$ V(***G***), the vertices $\phi(v)$ and $\phi(w)$ are adjacent in ***H*** whenever *v* and *w* are adjacent in ***G***.

Let ***G*** and ***H*** be weighted graphs. A mapping ϕ from V(***G***) to V(***H***) is a ***weight-restricted graph homomorphism,*** denoted $G \xrightarrow{w} H$, if ϕ is a graph homomorphism from ***G*** to ***H*** such that $w_G(v) \le w_H(\phi(v))$, for all $v \in$ V(***G***).





Graph Homomorphisms

The following theorems are generalizations of work done by Meagher and Stevens (2002) for the uniform alphabet case.

Theorem 1: (Meagher, Moura, Zekaoui)
Let ***G*** and ***H*** be weighted graphs with weights g_1, g_2,…, g_k and h_1, h_2, …, h_l respectively. If there exists a weight-restricted graph homomorphism
ϕ: ***G*** $\xrightarrow{w}$ ***H*** then **CAN(*G*,** $\prod_{i=1}^{k} g_i$ **) ≤ CAN(*H*,** $\prod_{j=1}^{l} h_j$ **).**

Theorem 2: (Meagher, Moura, Zekaoui)
Let ***G*** be a weighted graph with ***k*** vertices and $g_1 \leq g_2 \leq \ldots \leq g_k$ be positive weights. Then,
CAN(*K*$_{\omega(G)}$, $\prod_{i=1}^{\omega(G)} g_i$ **) ≤ CAN(*G*,** $\prod_{j=1}^{k} g_j$ **) ≤ CAN(*K*$_{\chi(G)}$,** $\prod_{l=k-\chi(G)+1}^{k} g_l$ **).**

n-chromatic Graphs for n = 2,3,4,5

From Theorem 2 and results from the paper by Moura, Stardom, Stevens, and Williams (2003), we get the next theorem.

Theorem 3: (Meagher, Moura, Zekaoui)
Let G be a weighted graph with k vertices with

weights $g_1 \leq g_2 \leq \cdots \leq g_k$. If one of the following holds:

1) $\chi(G) = 2, 3,$

2) $\chi(G) = 4$ and $\prod_{i=k-3}^{k} g_i \notin \{2^4, 6^4\}$, or

3) $\chi(G) = 5$ and $\prod_{i=k-4}^{k} g_i \notin \{2^5, 3^5, 23^4\}$ and $g_{k-1} \notin \{4, 6, 10\}$, then

CAN(G, $\prod_{i=1}^{k} g_i$) $\leq g_{k-1} g_k$.

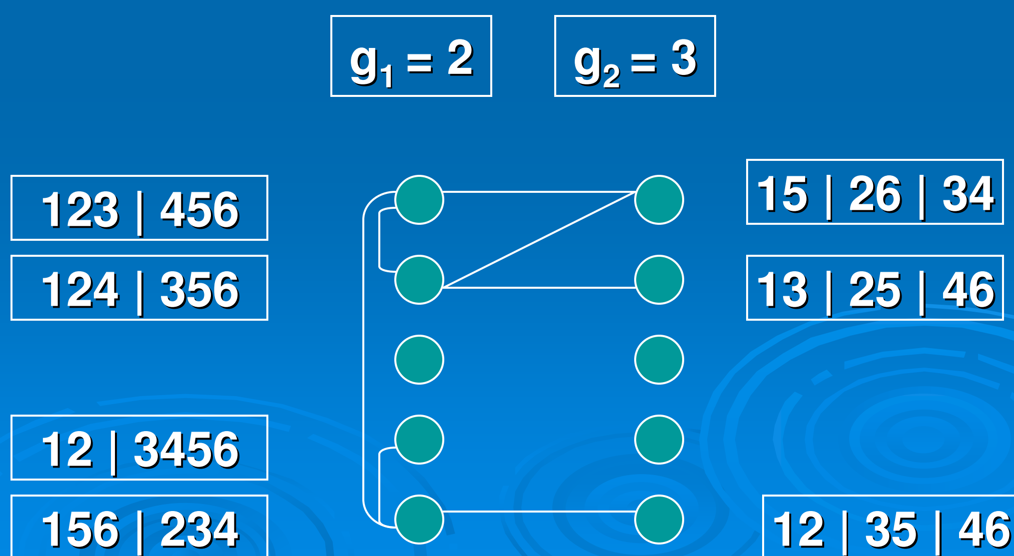
The covering array number we are providing is an upper bound.

Mixed Qualitative Independence Graph

Mixed Qualitative Independence Graph, denoted **QI**(N, $\prod_{i=1}^{l} g_i$), is a graph:

- whose vertex set is the set of all g_i-partitions of an N-set
- vertices are adjacent if and only if their corresponding partitions are qualitatively independent.

Example: QI(6, 2x3)



Mixed Qualitative Independence Graph

Theorem 4: (Meagher, Moura, Zekaoui)

For a weighted graph ***G*** and positive integers ***N*** and $g_1, g_2, \ldots, g_k$ there exists a CA(***N***, ***G***, $\prod_{i=1}^{k} g_i$) if and only if there exists a weight-restricted graph homomorphism $\boldsymbol{G} \xrightarrow{w}$ **QI**(***N***, $\prod_{i=1}^{k} g_i$).

Corollary 5: (Meagher, Moura, Zekaoui)

Let ***N*** be a positive integer and let ***G*** be a weighted graph with distinct weights $g_1, g_2, \ldots, g_r$, repeated $s_1, s_2, \ldots, s_r$ times, respectively. If $\omega(\boldsymbol{G}) > \omega(\mathbf{QI}(\boldsymbol{N}, \prod_{i=1}^{r} g_i))$ or $X(\boldsymbol{G}) > X(\mathbf{QI}(\boldsymbol{N}, \prod_{i=1}^{r} g_i))$, then **CAN**(***G***, $\prod_{i=1}^{r} g_i^{s_i}$) > ***N***.

Mixed Covering Arrays on Graphs

The problem of finding an optimal covering array on a general graph has been shown to be NP-hard, even when restricted to the binary alphabet case. (Seroussi and Bshouty, 1988)

We will build optimal covering arrays for special classes of graphs:

- trees,
- cycles, and
- bipartite graphs.

From Theorem 3, for ***G*** in one of these classes we have

$$\mathbf{CAN}(G, g_1, g_2, \ldots, g_k) \leq g_{k-1} g_k.$$

Theorem 6: (Meagher, Moura, Zekaoui)
Let G be a weighted tree, cycle or bipartite graph then,

$$\mathrm{CAN}(G, g_1, g_2, \ldots, g_k) = \mathrm{PW}(G).$$

Graph Operations

- **One-vertex Edge Hooking**

  Insert a new edge where one end is in V(***G***) and the other is a new vertex.

- **Edge Duplication**

  Create an edge that is parallel to an existing edge in ***G***.

- **Weight-Restricted Edge Subdivision**

  Edge subdivision such that if x is the new vertex in ***G*** adjacent to vertices y and z then $w_G(x)w_G(y) \leq PW(G)$ and $w_G(x)w_G(z) \leq PW(G)$.

The above operations will have no effect on the covering array number of the modified graph.

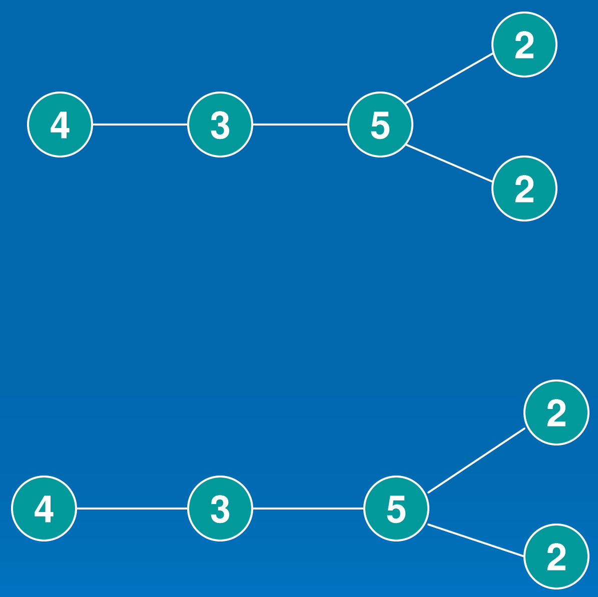
Optimal Tree Construction

- Build a tree T by starting with an edge {u, v} such that PW(T) = *w*(u) * *w*(v).

- Next, apply successive one-vertex edge-hooking in the proper order so as to obtain T.

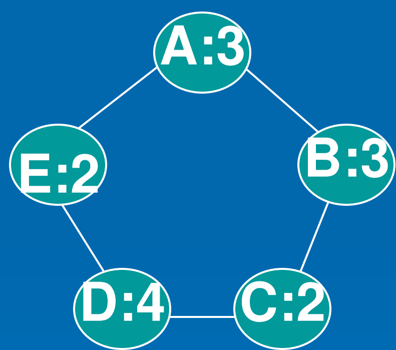
- CAN(T, g_1 g_2…g_k)=PW(T)

PW(T) = 15



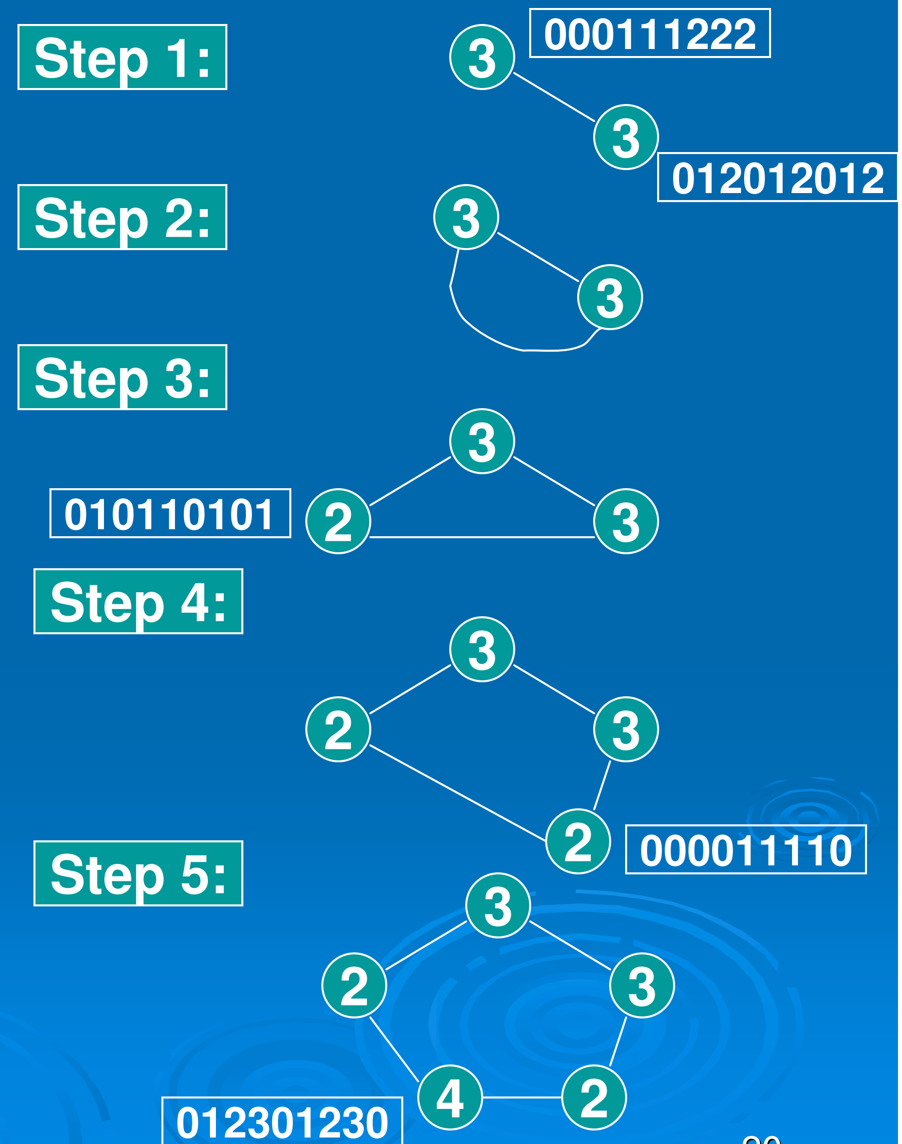
Optimal Cycle Construction

To build a CA on the cycle *C* below with PW(*C*) = 9



A: 000111222
B: 012012012
C: 000011110
D: 012301230
E: 010110101

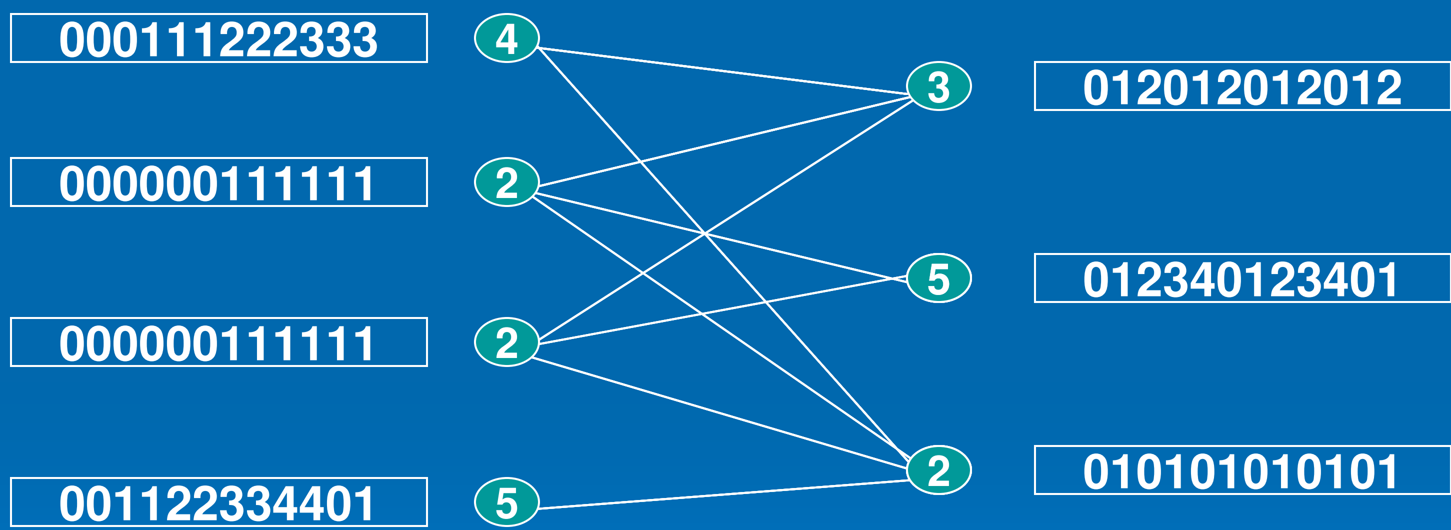
CA(9, *C*, $2^2 3^2 4$)



Optimal Bipartite Construction

Repeat each symbol 0,1,…, g-1 (PW(***G***) / g_i) times

Repeat the symbols 0,1,…, g-1 (PW(***G***) / g_i) times



Graph ***G:*** PW(***G***) = 12

Future Work

Finding Optimal Covering Arrays for other classes of graphs

- Solved for the uniform alphabet size cubic graphs and wheels

Implementing Tabu Search Methods for Covering Arrays

- Stardom's Algorithm (2001).
- Nurmela's Algorithm (2004).
- Moura and Zekaoui's Algorithm (in progress) which combines greedy techniques with a tabu search method that adds or deletes a test case at each iteration.

References

- S.Y. Boroday and I.S Grunskii. Recursive generation of locally complete tests. Cybernetics and Systems Analysis 28 (1992), 20- 25.
- C. Cheng, A. Dumitrescu and P. Schroeder. Generating small combinatorial test suites to cover input-output relationships. Proceedings of the Third International Conference on Quality software. Dallas (2003), p. 76-82.
- D.M.Cohen, S.R.Dalal, M.L.Fredman, and G.C.Patton. The AETG system: an approach to testing based on combinatorial design. IEEE Transactions on Software Engineering. 23(1997), p.437-44.
- C.J.Colbourn. Combinatorial aspects of covering arrays. Le Matematiche(Catania) 58(2004), p. 121-167.
- L.Garagano, J.Korner, and U.Vaccaro. Capacities: from information to extremal set theory. Journal of Combinatorial Theory. 68(1994), p. 296-316.
- K. Meagher, L. Moura, L. Zekaoui. Mixed Covering Arrays on Graphs. Journal of Combinatorial design. to appear.
- K. Meagher, B. Stevens. Covering arrays on graphs. Journal Combinatorial Theory. Ser. B 95(2005), p. 134-151.
- L. Moura, J. Stardom, B. Stevens, A. Williams. Covering Arrays with Mixed Alphabet Sizes. Journal of Combinatorial Design. 11(2003), p. 416-432.
- G. Seroussi and N.H.Bshouty. Vector sets for exhaustive testing of logic circuits. IEEE Trans. on Infor. Theory. 34(1988), p. 513-522.
- A.Williams and R.L.Probert. A measure for component interaction test coverage. IEEE(2001), p. 301-311.

Workshops, Ottawa, May 12-16

http://www.fields.utoronto.ca/programs/scientific/05-06/discrete_math/
http://www.fields.utoronto.ca/programs/scientific/05-06/covering_arrays/

- Ottawa-Carleton DISCRETE MATH DAY
- May 12-13 (Friday-Saturday)
- Plenary Speakers:
  Bill Cook, Anthony Evans, Jonathan Jedwab, Pierre Leroux, Kieka Mynhardt

- Workshop on COVERING ARRAYS
- May 14-16 (Sunday-Tuesday)
- Plenary Speakers:
  Rick Brewster, Charlie Colbourn, Peter Gibbons, Alan Hartman, Brett Stevens, Doug Stinson.

DEADLINE April 26

- Student financial support to travel to Ottawa
- Submission of abstracts for contributed talks

THANK YOU