Enumerating Costas Arrays

By Emils Matiss, BSc Computer Science
supervised by Dr. Illias Kotsireas

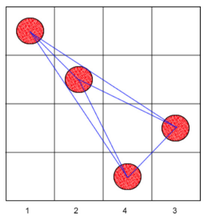
Wilfrid Laurier University

23rd March 2018

Abstract

Costas arrays are constructed similarly to the n-queens problem, an n-by-n grid has one point in every column. Any displacement vector from any two points on the Costas array must be unique. This property allows any two points to be identifiable, resulting in an auto-ambiguity function useful in RADAR and SONAR. Costas arrays are named after John P. Costas who published the concept in 1965. Later in the same year Edgar Gilbert independently wrote about the same concept. Enumeration is a computationally expensive problem, and improvements are translatable to other similar problems. [1]

Order 4 Costas

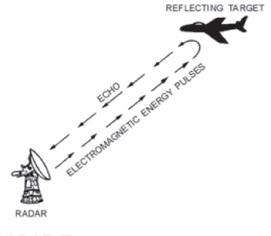


Known Costas Arrays

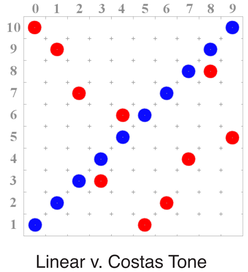
1	1	10	2160/28	19	10240/12	28	712/0
2	2	11	4368/36	20	6464/8	29	≥ 164/10
3	4	12	7852/34	21	3536/16	30	≥ 664/8
4	12/2	13	12828/50	22	2052/10	31	≥ 8/0
5	40/4	14	12752/46	23	872/20	32	?
6	116/10	15	19612/62	24	200/0	33	?
7	200/20	16	21104/40	25	88/4		
8	444/18	17	18276/38	26	56/4		
9	760/20	18	15096/20	27	204/14		

1 Applications

Since Costas arrays are auto-ambiguity functions, they are useful in SONAR and RADAR. In a perfect environment the distance can be calculated with RADAR by timing how long the echo takes to return to the antenna, and the speed is calculated by the change in frequency due to the Doppler effect. However, in a real environment many objects can reflect the same pulse back, resulting in noise that would prevent the target from being correctly identified. The US military used an ascending tone to reduce noise, which was an imperfect solution since the Doppler effect would shift the frequency, resulting in the distance being shifted. Costas arrays increased the accuracy of RADAR by making them immune to Doppler shifts since any two points can be uniquely identified. [2]



RADAR Example



Linear v. Costas Tone

2 Construction

The Costas enumeration problem is computational expensive, it falls in the factorial complexity category. Some orders however can be constructed mathematically. The Welch method named after Lloyd R. Welch, originally found by Gilbert, allows for costas arrays to be constructed by raising a primitive root to the power of its column, and the finding the modulo of the corresponding prime. [2]

3 Problem

To confirm if a sequence is a Costas array, we compute all possible displacement vectors, and then check if duplicates exist. This is an expensive operation, requiring n! operations - resulting in 3.049 ×10^29 comparisons for an array of order 28.

3.1 Computational Representation

Since all displacement vectors must be unique, their position on the grid is not relevant in determining if a sequence is a Costas array. Therefore, a boolean 2D array of size (n-1)*2(n-1) can be used to represent all possible, with the x dimension (of size n-1) representing the x component, and the y dimension (of size 2(2-1), since we can have a positive/upwards or negative/downwards displacement) representing the y component.

To test if an array is a Costas array all combinations of points are iterated over, and their corresponding displacement vector is marked on the above mentioned 2D array. If at any time two points represent a location on the 2D grid that is already marked, we stop the process and conclude that the array is not a Costas array. If after all combinations of points are evaluated, and all displacement vectors are uniquely represented on the 2D grid, the array is deemed to be a Costas array. In C a char datatype is used in place of booleans.

3.2 C implementation

```
int isCostas(int *array, int order) {
   int i, j;
   int cell;
   char vectors[order-1][2*(order-1)]; //2D grid

   for(j=0; j<2*(order-1); j++) { //Initialize grid with zeros
      for(i=0; i<order - 1; i++) {
         vectors[i][j] = 0;
      }
   }
   for(i=0; i<order; i++) { //Compute all displacement vectors
      for(j=i+1; j<order; j++) {
         if(array[i] > 0 && array[j] > 0) { //Treat 0 as wildcard
            cell = vectors[j - i - 1][order - array[j] + array[i] - 2]++;
            if(cell>0) {
               return 0;
            }
         }
      }
   }
   return 1;
}
```

3.3 Tree Traversal

A recursive breadth-first traversal is not computationally feasible since it would require a large amount of RAM, which would slow the enumeration immensely due to slow access times.

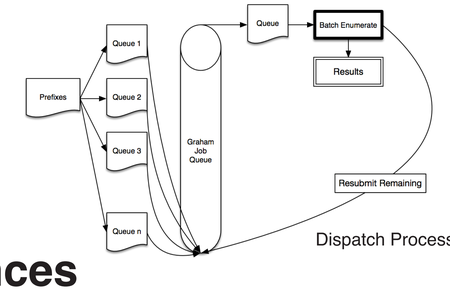
However, depth-first searching is possible as it only requires 2(n-1)^2 n bytes for an array of order n. The benefits of depth-first traversal is that calculations can be passed forward, reducing redundant calculations; On the nth recursive step, the nth column's point is compared to the remaining right columns, adding their corresponding displacement vectors to the 2D grid, if a duplicate displacement vector is found the algorithm returns to the previous level and tries the next available number.

3.4 Current Implementation

A variation of the tree traversal is currently used to enumerate Costas arrays. For efficiency, instead of relying on RAM the problem is optimized to fit inside the smallest cache in the processor. Due to the amount of computation required, the problem is split into sections; For example, all valid four position prefixes are precomputed, and then run separately. For Costas arrays of order 28 there are approximately 1.8 × 10^5 valid prefixes, and each one takes approximately 4 hours on Graham - Canada's newest academic supercomputer on SHARCNET. [3]

4 Enumeration

Current enumeration efforts are being run on Graham, the newest SHARCNET cluster, hosted around the corner at the University of Waterloo. Graham has 33,448 cores (32 per node).
Serial farming is used to process the 1.8 X 10^5 prefixes, which are split into smaller queues that are dispatched to nodes. Each node can process 32 prefixes simultaneously, processing on average 384 prefixes in a 48 hour block. Once a prefix is calculated, it is removed from the queue and the result is stored in a file. Occasionally all queues are collapsed into one list, and then redistributed to balance the queue lengths. [3]



Dispatch Process

References

[1] Drakakis, K. (2010, November 3). An Introduction to Costas Arrays. Retrieved from http://www1.spms.ntu.edu.sg/~ccrg/documents/basicTalkSingapore.pdf

[2] Wolff, C. (n.d.). Radar Basics. Retrieved March 22, 2018, from http://www.radartutorial.eu/08.transmitters/Costas%20Code.en.html

[3] SHARCNET. (n.d.). Cluster graham.sharcnet.ca. Retrieved March 22, 2018, from https://www.sharcnet.ca/my/systems/show/114