Constructing Optimal Golomb Rulers: Greedy Search and Minimal Seed Sizes

Chad Brewbaker; Iowa State University

May 10, 2002

Abstract

Contents

1 What is a Golomb Ruler?

A Golomb ruler is a ruler that instead of having the marks evenly spaced, all marks are unevenly spaced in the sense that the distance between any two marks is unique. An optimal Golomb ruler of size n has n marks at non-negative integer coordinates starting at zero, with the distance between any to marks unique, and the largest mark used is the smallest possible

Golomb rulers [6] are named after the University of Southern California mathematician Solomon W. Golomb [5]. They are used in applications such as building antenna arrays for astronomers.

For an index of the best known optimal Golomb rulers visit the website maintained by James B. Shearer [2].

Currently, the largest project for confirming optimal golomb rulers is done by the members of Distributed.net [7]. Also, a previous attempt at generating optimal Golomb rulers with an evolutionary technique was[3].

2 Greedy Search Algorithm

INSERT BACKGROUND OF CONWAY'S WORK

A greedy approach to constructing a Golomb ruler is to start with the first mark at zero, and add the next mark at the lowest possible spot that doesn't violate the "all distances between two marks are distinct" rule.

Since marks on an optimal Golomb ruler are dependent on other marks present, we can search for a set of "seeds", or marks on an optimal ruler that exclude other unwanted marks.

Example: An optimal golomb ruler of size 3 is :$\{0, 1, 4, 6\}$. However, a greedy constructed ruler with no seeds would be $\{0, 1, 3, 7\}$. To force out the 3 and 7 we need either the seed $\{4\}$ or $\{6\}$ hard coded in the ruler so that the greedy construction outputs the ruler $\{0, 1, 4, 6\}$.

The reason why we might want to search the seed space is because seeds require less marks than an actual ruler. Searching for seeds tightens the search space, and hopefully speeds up the search for optimal golomb rulers.

3 Minimal Seeds

To get an idea of how large our seed sets needed to be we took some of the proven optimal golomb rulers size 1-24 and checked subsets of them to find out what was the smallest seed that we would need to generate a seed of a given size. Since there can be more than one optimal seed of minimal length we also include the number of distinct seeds in the union of all minimal seed sets for an optimal Golomb ruler of given length, and the number of minimal seed sets that exist for an optimal Golomb ruler of a given length.

Ruler Size	Min. Seed Size	Distinct Seeds	Distinct Points
1	0	0	0
2	0	0	0
3	1	2	2
4	2	3	3
5	2	4	4
6	3	5	5
7	2	2	4
8	3	1	3
9	3	3	5
10	5	15	9
11	5	5	8
12	6	11	10
13	6	8	11
14	6	1	6
15	6	14	14
16	6	1	6
17	7	4	10
18	8	7	16
19	9	94	18
20	10	28	19
21	8	1	8
22	9	24	18
23	9	2	10
24	10	17	19

This table includes minimal seed data for only one set of optimal Golomb rulers. Other optimal Golomb rulers exist, and their data should also be included in this table to make it accurate.

A quick glance at the table shows that a seed needs to be at about $\frac{1}{2}$ as long as the ruler we are searching for, although some seeds are smaller. Hopefully this $\frac{1}{2}$ rule applies to larger rulers. Minimal seed length is an open problem for $n > 24$.

4 Evolutionary Seed Search

To search for a seed that yields an optimal Golomb ruler we implemented an evolutionary search technique. The genetic algorithm kept a population of candidate seeds, where the length of each seed was estimated off of the minimal seed data above, and each seed contained the implicit mark 0, and any other n marks between 1 and the best known ruler of that size.

Roulette selection was used, and mutation was done at 5% by perturbing a random mark from a random seed. Two point crossover was done on the seeds, which were logicaly stored as rings.

Fitness was calculated as the largest mark used after a greedy construction with the seed, plus a penalty for each distance between marks that was repeated

in the seed.

Also, to speed things up, a look-up table was implemented as a circular buffer containing the last 500 unique seeds and their evaluated fitness. This gave a noticeable speed-up after the first generation.

5 Results

Due to a bug in the code found a few days before this paper was finished, the CPU time to do re-testing with was rather limited. The following table is a list of the results.

Ruler Size	Ruler Found	Best Ruler Known	Generation Found	Time Run
3	3	3	0	seconds
4	6	6	0	seconds
5	11	11	0	seconds
6	17	17	3	seconds
7	25	25	0	seconds
8	34	34	47	seconds
9	44	44	260	seconds
10	55	55	178	seconds
11	72	72	4391	20 minutes
12	85	85	121259	12 hours
13	117	106	10459	6 hours
84	22050*	6159	60	12 hours

*Note that the 22050 was a ruler of length 6050 with a penalty score of 16000 for repeated marks.

The above table sugests that this technique blows up around rulers of size 12. This technique might be ok for generating near optimal Golomb rulers if you had a few weeks of CPU time, but in it's current state it dosen't look to promising.

Source code and data files can be found at http://www.public.iastate.edu/ crb002/OGR.html.

6 Future Work

As with any evolutionary algorithm your fitness landscape is very important. For this technique to work better a new fitness function that behaves better for this problem's topology would be nice. Another adaptation that might help is a variable size seed length that is evolved along with the seeds.

7 Acknowledgments

Thanks to Dan Ashlock for proposing these problems.

8 Appendix

The appendix is rather large, so to save paper minimal seed sets, runs, and source code are posted at http://www.public.iastate.edu crb002/OGR.html.

References

[1] *Project Website*, Chad Brewbaker, http://www.public.iastate.edu crb002/OGR.html

[2] *Table of Shortest Known Golomb Rulers*, James B. Shearer, http://www.research.ibm.com/people/s/shearer/grtab.html

[3] *Genetic Algorithm Approach to the Search for Golomb Rulers*, Stephen W. Soliday, Abdollah Homaifar, and Gary L. Lebby, Genetic Algorithms: Proceedings of the Sixth International Conference on Genetic Algorithms, University of Pittsburgh PA, vol:1,pp 528-535, Morgan Kaufman, July 1995, ISBN:1-55860-370-0

[4] *Genetic Algorithms + Data Structures = Evolution Programs 3rd ed.*, Zbigniew Michalewicz, Springer-Verlag Berlin Heidelberg, 1996, ISBN:3-540-60676-9

[5] *Homepage of Solomon W. Golomb*, Solomon W. Golomb, http://commsci.usc.edu/faculty/golomb.html

[6] *Applications of numbered undirected graphs*, Gary S. Bloom and Solomon W. Golomb, Proceedings of the IEE, 65(4):562-570,1977

[7] *Distributed.net Project OGR*, Jerome me Froment-Curtil and John Vender, http://www.distributed.net/ogr/