Evolving the Maximum Segment Length of a Golomb Ruler

Tiago Leitão

Centre for Informatics and Systems of the University of Coimbra,
Polo II - Pinhal de Marrocos, 3030 Coimbra, Portugal
tleitao@student.dei.uc.pt

Abstract. An evolutionary algorithm based on Random Keys to represent Golomb Rulers segments has been found to be a reliable option for finding Optimal Golomb Rulers in a short amount of time, when comparing with standard methods. This paper presents a modified version of this evolutionary algorithm where the maximum segment length for a Golomb Ruler is also part of the evolutionary process. Attained experimental results shows us that this alteration doesn't seems to provide significant benefits to the static version of the algorithm.

1 Introduction

A Golomb ruler is defined as a ruler that has marks unevenly spaced at integer locations in such a way that the distance between any two marks is unique. They were named after the relevant work of the mathematician Solomon Golomb [1], and, unlike usual rulers, they have the ability to measure more discrete measures than the number of marks they carry. Also Golomb rulers are not redundant, since they do not measure the same distance twice.

Although the definition of a Golomb ruler does not place any restriction on the length of the ruler, researchers are usually interested in rulers with minimum length. An Optimal Golomb Ruler (OGR) is defined as the shortest length ruler for a given number of marks. There may exist multiple different OGRs for a specific number of marks. OGRs are used in a wide range of real world situations. For example, in the field of communications when setting up an interferometer for radio astronomy, placing the antennas on the marks of a Golomb ruler maximizes the recovery of information about the phases of the signal received [2], [3].

Evolutionary Computation (EC) approaches are a promising alternative to brute force methods that usually need too much time to obtain an answer and so cannot be considered as a realistic option in real world situations. To the best of our knowledge, there are just three applications of EC to this problem [4], [5] and [6]. The first two, when searching for solutions evolve the length of a fixed number of segments. This way, during search EC algorithms try to discover good rulers for a specific number of marks. In [6] a different evolutionary approach is proposed. Prior to the application of the algorithm, a maximum ruler length is specified and then the search procedure tries to determine how many marks can be placed in such a ruler as well as where each one of the marks should be located. In this paper we continue the study presented in [5], by analyzing the influence of evolving the maximum length of a segment.

2 Golomb Rulers

A n-mark Golomb ruler is an ordered set of n distinct nonnegative integers $\{a_1, a_2, \ldots, a_n\}$ such that all possible differences $|a_i - a_j|$, $i, j = 1, \ldots, n$ with $i \neq j$, are distinct. Values a_i correspond to positions where marks are placed. By convention, the first mark a_1 is placed on position 0, whereas the length of the ruler is given by the position of the rightmost mark a_n. A Golomb ruler with 4 marks can be defined as $\{0, 1, 4, 6\}$. The length of a segment of a ruler is defined as the distance between two consecutive marks. This way, it is also possible to represent a Golomb ruler with n marks through the specification of the length of the $n-1$ segments that compose it. According to this notation the example presented before can be defined as $\{1, 3, 2\}$. This raises the following issues: Should a maximum value for λ, the maximum segment length, be pre-established or should it be adjusted during the construction of a ruler? How to select the $n-1$ elements from a set of λ values? How to build a valid permutation with the $n-1$ elements selected? This paper focus on the first question: should the maximum value for a segment be fixed or evolved.

3 An Evolutionary Approach with Random Keys

The representation chosen for individuals plays a crucial role on the performance of EC algorithms. In [5], an evolutionary approach based on the evolution of ruler segments is proposed. A candidate solution for an OGR-n instance is composed by a permutation of λ distinct values, where λ is the maximum segment length. Encoding of the permutation is done with Random Keys (RK) [7], [8], which ensures that the application of standard genetic operators (e.g., one point crossover) to chromosomes obtains feasible individuals. It is important to notice that the representation proposed in this approach is redundant, since a chromosome contains more segments than necessary to build the ruler. To obtain a possible solution (i.e., a ruler), a two-step process is performed: the RK encoding is translated into a permutation of integers and then an interpretation algorithm chooses the segments used in the ruler. Figure 1 illustrates this process.

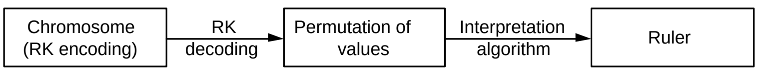

Fig. 1. Decoding and interpretation of the information contained in a chromosome

Evaluation of an individual follows two criteria: ruler length and legality of the solution (i.e., whether it contains repeated measurements). The effect of the addition of a simple heuristic to the interpretation process is also analyzed [5]. Results presented show a small improvement in the performance of the EC algorithm.

In this paper, the value for λ is also evolved. This is accomplished in a simple manner. The value for λ is given by the chromosome length, this means that by having chromosomes with variable length each individual will have a different λ value. The only modification that is necessary to the previous approach, with fixed length, is made on the crossover operator, that for each individual, a different cut point is randomly selected. This will ensure the swap of genetic material of different lengths, thus providing individuals with variable length.

4 Experimental Results

To evaluate our approach we performed a set of experiments with several OGR instances. More precisely, we used the evolutionary algorithm to seek for good rulers with 10 to 17 marks. The settings of the EC algorithm are the following: Number of generations: 5000; Population size: 100; Tournament selection with tourney size: 5; Elitist strategy; One point crossover with rate: 0.75; An evolutionary strategy like mutation operator is used. When undergoing mutation, the new value v_{new} for a given gene (i.e. a key in the chromosome) is obtained from the original value v_{old} in the following way:

$$v_{new} = v_{old} + \sigma \times N(0,1) \quad (1)$$

Where $N(0,1)$ represents a random value sampled from a standard normal distribution and σ is a parameter from the algorithm. In our experiments we used $\sigma = 0.1$. Mutation rate was set to 0.25 per gene. For every OGR instance we performed 30 runs with the same initial conditions and with different random seeds. All initial populations were randomly generated with values for keys selected from the real interval $[0,1]$. Significance of the results was tested with a t-test with level of significance 0.05.

Table 1. Best rulers found and averages of the best rulers for the 30 runs, for all the tested approaches. Statistical analysis of te results with t-test.

Instances	Std GA		RK Fix λ		RK Evolve λ		t-test
	Best	Avg	Best	Avg	Best	Avg	$p \leq 0.05$
OGR-10	79	89.93	**55**	60.14	**55**	**57.83**	**0.000159**
OGR-11	98	117.10	**72**	76.00	**72**	**74.93**	**0.031262**
OGR-12	136	149.27	**91**	95.64	**91**	**94.40**	**0.008301**
OGR-13	152	187.87	**111**	**117.21**	114	118.50	**0.017769**
OGR-14	253	283.27	**131**	**143.57**	**131**	144.37	0.672176
OGR-15	295	344.07	**167**	**172.82**	**167**	174.63	0.054341
OGR-16	337	435.60	**200**	**207.82**	202	210.53	**0.008406**
OGR-17	434	500.10	236	**245.46**	**230**	246.53	0.157805

Examining table 1 show us that both RK approaches have better results than the standard segment evolution. Comparing the best rulers found for each instance, we can say that the static λ (RK Fix) approach seems to discover better rulers, although the evolved λ approach (RK Evolve) obtained the best ruler for 17 marks. For lower instances, the averages of the best solutions found in each one of the 30 runs are better in RK Evolve than RK Fix. For larger instances, more than 12 marks, the best averages are attained with the RK Fix approach. In spite of these results, by looking at the column with the statistic analysis, we verify that significant differences between both RK approaches can be found for instances from 10 to 13 and 16 marks.

5 Conclusion

In this paper we presented some experiments regarding the evolution of the maximum segment length of a Golomb Ruler. This effect was attained by slightly altering the genetic algorithm presented in [5]. Results presented in this paper show us that there aren't significant gains in evolving the maximum segment length. Even so, it seems that there might be some advantages for lower instances.

6 Acknowledgements

The author wishes to thank the faculty mentors Jorge Tavares and Ernesto Costa, as well as the members of the ECOS group, for all the helpfull discussions.

References

1. Golomb, S.: How to Number a Graph. In: Graph Theory and Computing. Academic Press (1972) 23–37
2. Blum, E., Biraud, F., Ribes, J.: On optimal synthetic linear arrays with applications to radioastronomy. IEEE Transactions on Antennas and Propagation **AP-22** (1974) 108–109
3. Hayes, B.: Collective wisdom. American Scientist **86** (1998) 118–122
4. Soliday, S., Homaifar, A., G., L.: Genetic algorithm approach to the search for golomb rulers. In: Proceedings of the Sixth International Conference on Genetic Algorithms (ICGA-95), Morgan Kaufmann (1995) 528–535
5. Pereira, F.B., Tavares, J., Costa, E.: Golomb rulers: The advantage of evolution. In: Proceedings of the 11th Portuguese Conference on Artificial Intelligence, Workshop on Artificial Life and Evolutionary Algorithms (ALEA), EPIA'03. (2003) 27–33
6. Tavares, J., Pereira, F.B., Costa, E.: Understanding the role of insertion and correction in the evolution of golomb rulers. In: Proceedings of the Congress on Evolutionary Computation (CEC 2004). (2004)
7. Bean, J.: Genetic algorithms and random keys for sequencing and optimization. ORSA Journal on Computing **6** (1994) 154–160
8. Rothlauf, F., Goldberg, D., Heinzl, A.D.: Network random keys - a tree representation scheme for genetic and evolutionary algorithms. Evolutionary Computation **10** (2002) 75–97