The Snake in the Box Problem

Mathematical Conjecture and a Genetic Algorithm Approach

Pedro A. Diaz-Gomez
Robotics, Evolution, Adaptation and Learning Laboratory (REAL Lab)
School of Computer Science
University of Oklahoma, OK, USA
pdiazg@ou.edu

Dean F. Hougen
Robotics, Evolution, Adaptation and Learning Laboratory (REAL Lab)
School of Computer Science
University of Oklahoma, OK, USA
hougen@ou.edu

ABSTRACT

With applications in coding theory and hypercube-based computing and networking, the "snake in the box" problem is of great practical importance. Moreover, it is both mathematically elegant and highly difficult. The problem is simply to find the longest "snake" in a hypercube. However, as the hypercube grows in dimensionality, the size of the search space increases exponentially. Moreover, as the maximum snake length is only known for the smallest dimensions (where the snakes themselves have already been identified), there is no known stopping criterion for the search in higher dimensions. In this paper we make a mathematical conjecture about the possible maximum length of a snake in a hypercube of dimension d. We use a genetic algorithm for finding snakes in a 8-hypercube to show some results.

Categories and Subject Descriptors

I.2.6 [**Artificial Intelligence**]: Learning

General Terms

Algorithms

Keywords

Genetic algorithms, hypercubes

1. INTRODUCTION

A *snake* in a hypercube of dimension d is a *connected open path* in the hypercube d, where each node in the path has exactly 2 neighbors that are also in the snake, except for the *head* (start) and the *tail* (end) that each have only one neighbor in the snake. The constraint, then, is that a node in the hypercube may be visited if it *is not a neighbor* of any previously visited node in the snake.

There have been some non-heuristic methods to solve the problem in hypercubes of dimension less than 8 [1] and it has been shown experimentally that Genetic Algorithms (GAs) are a powerful heuristic tool to solve this type of problem [2].

GECCO'06, July 8–12, 2006, Seattle, Washington, USA.
ACM 1-59593-186-4/06/0007.

2. SNAKES AND GENETIC ALGORITHMS

In order to *encode a hypercube* we use a two dimensional *matrix of neighbors MN*, where each row indicates the node and each column indicates the neighbors of that node. The matrix multiplication between the MN matrix of neighbors and the chromosome I gives us the number of neighbors of each node in the path—we call this product MP.

In order to ensure that the path is connected we use as our fitness function Equation 1:

$$F(I) = \left(\frac{\sum_{j=0}^{2^d-1} (MP)_j - Penalty}{\sum_{j=0}^{2^d-1} (MP)_j} \right) \left(\frac{Length\_S + 1}{\#P} \right) \quad (1)$$

where *Penalty* corresponds to those points that violate the constraint, $Length\_S$ counts the points in the path that are connected, from the head to the tail—if that is the case—or to the point where the constraint is violated or the connection is broken, and $\#P$ is the number of points that belong to the snake. The *constraint* takes into account the following parts: (1) genes with value 1 in the chromosome and with number of neighbors greater than 2, (2) *isolated points*, that is, genes with value 1 in the chromosome and with no neighbors, (3) *lazy points*, that is, genes with value 0 in the chromosome and with no neighbors, and (4) chromosomes with more than one head and one tail.

2.1 Experimental Setting

To search for snakes in an 8-dimensional hypercube, we encode the problem as a binary array of length $2^8 = 256$. The initial population of $1,000$ individuals (versus $10,000$ used in Casella's work [1]) is seeded with a maximum length snake from dimension 7 found by Rajan [3]. The first 128 genes of all the individuals correspond to the length 49 snake found in the previous hypercube of dimension 7. The remaining 128 genes, i.e., genes from 128 to 255 are randomly generated, taking into account to turn off (give value 0 to) the neighbors of the snake that is seeded.

Crossover and mutation are performed on the chromosome only above gene 127, in order to preserve the initial seed. We used *tournament selection*: two chromosomes are selected randomly and with a probability of 75% the one with higher fitness value is selected and with a probability of 25% the one with the lower fitness value is selected.

2.2 Results

In order to evaluate the length of the snake we used different approaches. One was to find a head and a tail and count the connected points from the head to the tail that did not violate the constraint, then subtract one. However, there could be other heads (or tails) that we can take into account to find a bigger snake. So, another approach was to look at all points in the chromosome as candidates to be head or tail, then we evaluate the length from each one of those, and score as the length the biggest of those. There could also be better candidates that have only a head but no tail. So the GA used can run looking at the length of the snake taking into consideration the path length from each head-tail pair that exists in the chromosome, or can run taking into consideration individual heads (tails) that exist in the chromosome and giving as the length the maximum connected path found without violations in the chromosome. For this type of experiment it turns out to be better to look at only one starting point (head) to find the length of the connected path that does not violate the constraint. We ran the algorithm for $1,000$ iterations and found snakes of length 81.

We can obtain longer snakes using some variants of the fitness function proposed in Equation 1, but that is not the topic of this paper.

3. MATHEMATICAL CONJECTURES

We define the *energy* of a snake as the *dot product* between the snake (the chromosome I) and the product vector MP which is the result of the matrix multiplication between the matrix of neighbors MN and the snake I. Formally,

$$E \;=\; I \cdot (MN * I)^T \;=\; I \cdot (MP)^T \qquad (2)$$

For instance, for a snake {0, 1, 3, 7, 6} the energy is the scalar value $E = (1\,1\,0\,1\,0\,0\,1\,1) \cdot (1\,2\,3\,2\,2\,2\,1\,2)^T = 8$

Conjecture 1. The number of points of a longest snake in a hypercube of dimension d where $d > 3$, is greater than or equal to the energy of the longest snake in the hypercube of dimension $d-1$.

Table 1 shows the values of the energy for longest snakes and the corresponding number of points for each snake. We use the energy definition *inductively*, i.e., we begin with a known longest snake in the 3-dimensional hypercube and from there we begin to conjecture as to the number of points for the longest snake in the 4-dimensional hypercube, and so forth. Table 1 compares our conjectures with theoretical results from Abbot [5] and practical bounds [1].

Hyp.	*Energy*	*#P Conjecture 1*	*Abbott*	*Casella*
3	8	5	2.4	5
4	14	8	4.8	8
5	26	14	9.6	14
6	50	26	19.2	27
7	98	50	38.5	51
8	194	98	77.0	98
9	386	194	154.0	187
10	770	386	308.0	359
11	1,538	770	616.0	681
12	3,074	1,538	1232.0	1,261

Table 1: Conjecture 1 of Number of Points in Longest Snakes vs. Theoretical and Casella's Findings

Again, we can observe values quite near those of our conjectures with Casella's experimental findings.

Conjecture 2. A *lower bound* for the number of points of a snake in a hypercube of dimension d, with $d > 3$ is

$$3 * 2^{d-3} + 2$$

This conjecture holds directly from the energy of a snake[1].

4. CONCLUSIONS AND FUTURE WORK

We have presented a Genetic Algorithm approach to find snakes in hypercubes, using a small number of individuals in the population, few iterations, and an effective fitness function. We use *only the head* as a starting point to find the length of a snake, taking into account the connected path until the tail is found and/or the constraint is violated. Besides that, we take into account all the "heads" in the chromosome and give as the length of the snake the highest value found in the unviolated connected path. We give the definition of the energy of a snake and use it to conjecture a lower bound on the number of points of a longest snake in a hypercube of dimension d. We compare our conjecture with theoretical and experimental findings by other researches and we proposed a new greater lower bound for the number of points of longest snakes in hypercubes.

For future work we can analyze the fitness function proposed in this paper and others than can be used to find snakes in the box. We are going to propose a new conjecture related with the upper bound of the number of points in longest snakes.

5. ACKNOWLEDGEMENTS

A special thanks to Benjamin P. Carlson and Lawrence Kincheloe for giving excellent ideas and sharing time with us analyzing the snake in the box problem.

6. REFERENCES

[1] D. A. Casella and W. D. Potter. New lower bounds for the snake–in–the–box problem: Using evolutionary techniques to hunt for snakes. In *Proceedings of Florida Artificial Intelligence Research Society Conference*, pages 264–269.

[2] W. D. Potter, R. W. Robinson, J. A. Miller, K. Kochut, and D. Z. Redys. Using the genetic algorithm to find snake-in-the-box codes. In *Proceedings of the Industrial and Engineering Applications of Artificial Intelligence and Expert Systems*, pages 421–426, 1994.

[3] D. Rajan and A. Shende. Maximal and reversible snakes in hypercubes. In *Proceedings of the Annual Australian Conference on Combinatorial Mathematics and Combinatorial Computing*, 1999.

[4] H. S. Snevily. The snake-in-the-box problem: A new upper bound. In *Discrete Mathematics 133*, 1994.

[5] E. W. Weisstein. Snake. MathWorld–A Wolfram Web Resource. http://mathworld.wolfram.com/Snake.html, accessed January 31, 2006.

[1]This result was conjectured by Snevily [4] as an *upper bound* for dimension $d \geq 5$.