Self-Adaptive Mutation Rates in Genetic Algorithm for Inverse Design of Cellular Automata

Ron Breukelaar
Leiden Institute of Advanced Computer Science,
Universiteit Leiden
P.O. Box 9512, 2300 RA Leiden,
The Netherlands
and
Blueridge Analytics Inc., 101 W. Worthington
Ave. Suite 206, Charlotte, NC, 28203, USA
ron@siteops.com

Thomas Bäck
Leiden Institute of Advanced Computer Science,
Universiteit Leiden
P.O. Box 9512, 2300 RA Leiden,
The Netherlands
and
Nutech Solutions GmbH,
Martin-Schmeisser-Weg, 44227 Dortmund,
Germany
baeck@liacs.nl

ABSTRACT

Self-adaptation is used a lot in Evolutionary Strategies and with great success, yet for some reason it is not the mutation adaptation of choice for Genetic Algorithms. This poster describes how a self-adaptive mutation rate was used in a Genetic Algorithms to inverse design behavioral rules for a Cellular Automata. The unique characteristics of this search space gave rise to some interesting convergence behavior that might have implications for using self-adaptive mutation rates in other Genetic Algorithm applications and might clarify why self-adaptation in Genetic Algorithms is less successful than in Evolutionary Strategies.

Categories and Subject Descriptors

I.2.8 [**Computing Methodologies**]: ARTIFICIAL INTELLIGENCE—*Problem Solving, Control Methods, and Search*

General Terms

Algorithms, Experimentation, Theory

Keywords

Self Adaptation, Genetic Algorithms, Cellular Automata

1. THE PROBLEM

The Majority Problem can be defined as follows:

Given a set $A = \{a_1, ..., a_n\}$ *with* n *odd and* $a_m \in \{0, 1\}$ *for all* $1 \leq m \leq n$*, answer the question: 'Are there more ones than zeros in A?'.*

The challenge as discussed in this poster is to solve this problem by inverse designing rules for Cellular Automata using a Genetic Algorithm.

The genetic algorithm for all experiments is a $(10, 100)$ strategy where 100 stands for the number of individuals in the pool and 10 for the number of parents that is selected from that pool. Parents are selected by selecting the ten fittest individuals from the population. Every parent is then copied ten times every generation and all individuals are mutated using a mutation operator.

A self-adaptive method is used first proposed by Bäck et al. [2] which suggests to add a floating point value to every individual which represents their mutation rate. The individual will then be mutated using a probabilistic bit flip operator using that mutation rate. The mutation rate in turn is mutated using the following formula:

$$p' = \left(1 + \frac{1-p}{p} \cdot (-\gamma \cdot N(0,1))\right)^{-1}$$

Where γ is a constant to impact the convergence speed of the mutation rate and is usually set to 0.22. $N(0,1)$ represents a random value from a normal distribution with mean 0 and standard deviation 1.

2. EXPERIMENT

Figure 1 shows the average mutation rate of the self-adaptive method compared to the three fixed mutation rates. And the important thing to notice here is that not only does the mutation rate go down quickly and constantly, but it hit the minimum of $p = 1/n$ as early as generation 50. This is surprising given the fact that in the first 100 generation a mutation rate of $p = 4/n$ and even $p = 8/n$ clearly out performs $p = 2/n$. Mutation using those settings should have been better during this part of the algorithm and self-adaptive mutation should have evolved into something close to this mutation rate. There clearly is another force pulling the mutation rate down.

3. HIDDEN PLUS STRATEGY

Even though every individual is mutated in a comma strategy, there is always the possibility that non of the bits is flipped, even though $p \geq 1/n$.

Let p_{clone} be the chance that an individual is not changed by mutation. Then: $p_{clone} = (1-p)^n$, where p is the mutation rate and n is the bit string length. This means that for $p = 1/n$ (minimum allowed in experiments) and $n = 128$ (true for all experiments):

$$p_{clone} = (1 - \frac{1}{128})^{128} \approx 0.3664$$

GECCO'08, July 12–16, 2008, Atlanta, Georgia, USA.
ACM 978-1-60558-130-9/08/07.

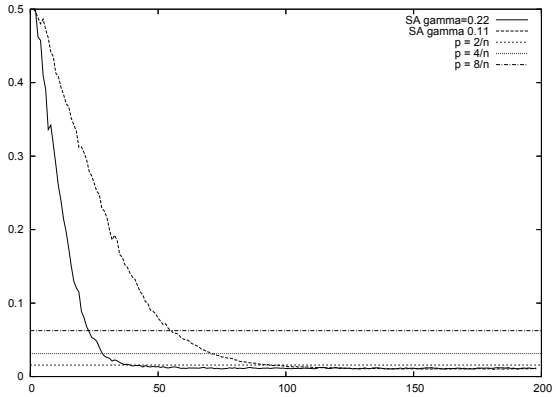

Figure 1: Shows the average mutation rate of the self-adaptive methods compared to the three fixed mutation rates.

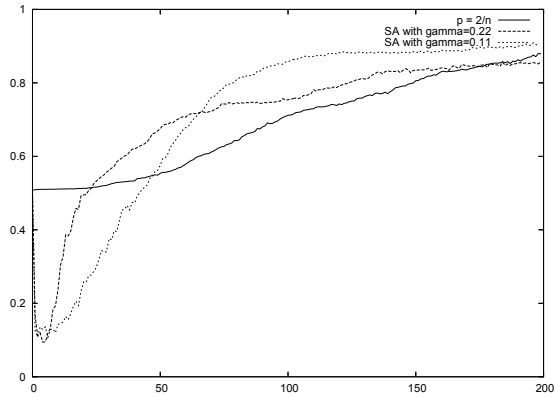

Figure 2: Shows the average fitness of the two self-adaptive mutation experiments with different γ settings vs. fixed mutation rates tested with $p = 2/n$.

Every parent in the experiments above was copied 10 times. This means that if the mutation rate $p = 1/n$ in for an individual and that individual is chosen to be a parent, then on average 3 to 4 copies of this parent will reside in the next generation, unaltered. Making the comma strategy effectively some kind of "probabilistic plus strategy".

4. COUCH POTATOES

An important aspect of the self-adaptive mutation rule is that the rule should have the same probability to change up, as it has to change down. This ensures that the search for an optimal mutation rate is not biased in any way. The experiments in this poster suggest that the self-adaptive mutation used does seem to have such a bias. Surprisingly the "probabilistic plus strategy" as explained above could create such a bias.

Every parent in the experiments was copied ten times. And right before the bit flip mutation is employed, the mutation rates are themselves mutated. If the mutation rate of a child is decreased its p_{clone} increases and it is more likely to stay identical than an child of which the mutation rate is increased. This simple fact makes that the average mutation rate of the children that are unchanged is lower than the mutation rate of their parent and the average mutation rate of the changed children is higher than that of their parent. At the beginning of the algorithm there are still a lot of improvements to be made and they are relatively easy to find. In a "rugged" landscape though, the success rate of a mutation might go down drastically. In a comma strategy the fitness of the next generation is allowed to be lower than that of the previous one, but if p_{clone} is high enough, enough clones will be generated such that the fitness will in practise only go up. This means that the harder it becomes to find an improvement, the more appealing it will be to stay where you are. Which will decrease the mutation rate and only make it harder to leave the comfort of the "couch".

5. CONCLUSIONS

Self-adaptive mutation rates have been successfully applied to the inverse design of rules for Cellular Automata. The mutation rates adapted from an initial very high rate to a level that was usable to run the Genetic Algorithm.

Self-adaptive mutation rates as it was applied here, did have some problems with the complexities of the search space of the Majority Problem. In particular there was a premature convergence measured that did not seem to be the effect of any speed setting of the algorithm or noise level on the fitness function.

The "Couch Potato"-effect as described in section 4 fits the measured effect in the experiments and therefore seems to be a viable explanation of the convergence behavior. Further and more general research seems needed to understand this potentially important convergence force.

6. REFERENCES

[1] Bäck, T.: Evolutionary Algorithms in Theory and Practice. Oxford University Press, NY (1996)

[2] Bäck, T. and Schutz, M.: Intelligent mutation rate control in canonical genetic algorithms. Proceedings of the 9th International Symposium, ISMIS 96, pages 158–167, Springer-Verlag, Berlin (1996)

[3] Breukelaar, R. and Bäck, Th.: Using a Genetic Algorithm to Evolve Behavior in Multi Dimensional Cellular Automata proceedings of Genetic and Evolutionary Computation Conference, GECCO 2005, ACM 1-59593-010-8/05/0006, pg. 107–114 (2005).

[4] Eiben, A.E., Hinterding, R., Michalewicz, Z.: Parameter control in evolutionary algorithms Evolutionary Computation, IEEE Transactions on Evolutionary Computation‘ Volume 3, Issue 2, Jul (1999), pages:124–141

[5] Mitchell, M., Crutchfield, J.P.: The Evolution of Emergent Computation. Proceedings of the National Academy of Sciences (1994), SFI Technical Report 94-03-012

[6] Li, W., Packard, N. H., Langton, C. G.: Transition phenomena in cellular automata rule space. Physica D, (1990), 45:77-94

[7] Wolfram, S.: Statistical mechanics of Cellular Automata. Reviews of Modern Physics volume 55 (1983)

[8] Wolfram, S.: Theory and Applications of Cellular Automata. World Scientific, Singapore (1986)