Generating Efficient Automata for Negotiations – An Exploration with Evolutionary Algorithms

D.D.B. van Bragt
CWI
P.O. Box 94079
1090 GB Amsterdam, The Netherlands
E-mail: bragt@cwi.nl

J.A. La Poutré
CWI
P.O. Box 94079
1090 GB Amsterdam, The Netherlands
E-mail: hlp@cwi.nl

Abstract

The rapid growth of a global electronic market place, together with the establishment of standard negotiation protocols, currently leads to the development of multi-agent architectures in which artificial agents can negotiate on behalf of their users (Maes et al., 1999). Most of today's (prototype) systems for automated negotiations, like Kasbah or Tête-à-Tête, use simple and static negotiation rules. Ideally, however, negotiating agents should be able to deal successfully with a variety of opponents (with different tactics and different preferences). Furthermore, they should be able to *adapt* their strategies to deal with changing opponents.

Such flexible and powerful bargaining agents can be obtained by representing the agents' bargaining strategies as finite automata. A finite automaton representation allows an agent to behave differently against different opponents. We demonstrate that highly effective bargaining automata are generated by an evolutionary algorithm (EA).

Our application domain is rather complex compared to the simple games considered in previous works [e.g., the iterated prisoner's dilemma (IPD) (Miller, 1996) or Nash's demand game (Ashlock, 1997)]. We focus on so-called *multi-issue* negotiations. In multi-issue negotiations not only the price of a product is important, but other aspects are also taken into account (for instance the quality of the product, the delivery time, etc.). Obviously, the complexity of such multi-issue negotiations increases rapidly when the number of issues becomes large.

We show how successful strategies (represented as finite automata) can be generated for this class of complex bargaining problems. We follow Miller's (1996) approach by encoding the automata as linear strings. We show that very efficient strategies can be generated by evolving these strings using a GA (when the strings are binary coded). We also propose a hybrid EA model, based upon evolutionary programming (EP) and evolution strategies (ES), which performs well in case of real codings.

We validate our approach in a series of experiments by testing the performance of the evolving automata in a competition with a variety of fixed opponents. A similar approach has been used in the past to generate robust strategies for the IPD (Axelrod, 1987). Current work focusses on the development of automata that perform well against *co-evolving* opponents.

References

D. Ashlock (1997). GP-Automata for Dividing the Dollar. In J.R. Koza, K. Deb, M. Dorigo, D.B. Fogel, M. Garzon, H. Iba, and R.L. Riolo, *Genetic Programming 1997: Proceedings of the Second Annual Conference*, 18-26. San Francisco, CA: Morgan Kaufmann.

R. Axelrod (1987). The evolution of strategies in the iterated prisoner's dilemma. In L. Davis (ed.), *Genetic Algorithms and Simulated Annealing*, 32-41. London: Pitman.

P. Maes, R.H. Guttman, and A.G. Moukas (1999). Agents that buy and sell. *Communications of the ACM* **42**(3):81-91.

J.H. Miller (1996). The coevolution of automata in the repeated prisoner's dilemma. *Journal of Economic Behavior and Organization* **29**(1):87-112.

Operator Adaptation in Structure Optimization of Neural Networks

Christian Igel
Institut für Neuroinformatik
Ruhr-Universität Bochum
44780 Bochum, Germany
Christian.Igel@neuroinformatik.ruhr-uni-bochum.de

Martin Kreutz
ZN Vision Technologies AG
44781 Bochum, Germany
Martin.Kreutz@zn-ag.de

The ability of an evolutionary algorithm (EA) to adapt its search strategy during the optimization process is a central concept in evolutionary computation, because (i) the best setting of an EA is not known *a priori* for a given task, (ii) the optimal search strategy is normally not constant during the evolutionary process. Thinking in terms of search (or generating) distributions, population and search strategy as well as selection and strategy adaptation are of equal importance. We propose a derandomized algorithm that adapts operator probabilities on population level. A detailed description, theoretical and more empirical results, and in particular references to related work can be found in [1].

Let Φ be a fitness function to maximize, Ω the set of (asexual) variation operators employed, and $p_o^{(t)}$ the probability at generation t that $o \in \Omega$ is chosen. Further, let $O_o^{(t)}$ contain all offsprings produced in generation t by application of operator o, e.g., if individual $\boldsymbol{g}$ has been generated by consecutive application of the operators o_i and o_j then $\boldsymbol{g}$ is added to $O_{o_i}^{(t)}$ and $O_{o_j}^{(t)}$. The adaptation of operator probabilities is based on a function q that measures the value of a single modification by an operator. One possible choice for this measure is the *local delta* or *credit* given by $q(\boldsymbol{g}) := \max\{\Phi(\boldsymbol{g}) - \Phi(\boldsymbol{g}_{\text{best}}), 0\}$, where $\boldsymbol{g}_{\text{best}}$ is the best individual in the current population. Replacing $\Phi(\boldsymbol{g}_{\text{best}})$ with the fitness of the parent of $\boldsymbol{g}$ yields an alternative measure called *benefit*. The generation dependent quality of $o \in \Omega$ is defined as $q_o^{(t)} := 1/\left|O_o^{(t)}\right| \sum_{\boldsymbol{g}\in O_o^{(t)}} q(\boldsymbol{g})$. The operator probabilities are adjusted every τ generations. Let $q_{\text{all}} := \sum_{i=0}^{\tau-1} \sum_{o'\in\Omega} q_{o'}^{(t-i)}$. Then we set

$$\tilde{p}_o^{(t+1)} := \begin{cases} \zeta/q_{\text{all}} \cdot \displaystyle\sum_{i=0}^{\tau-1} q_o^{(t)} + (1-\zeta)\cdot \tilde{p}_o^{(t)} & \text{if } q_{\text{all}} > 0 \\ \zeta/|\Omega| \qquad\qquad\quad + (1-\zeta)\cdot \tilde{p}_o^{(t)} & \text{else} \end{cases}$$

for all $o \in \Omega$. Then

$$p_o^{(t+1)} := p_{\min} + (1 - |\Omega|\cdot p_{\min}) \frac{\tilde{p}_o^{(t+1)}}{\sum_{o'\in\Omega} \tilde{p}_{o'}^{(t+1)}} \ .$$

Herein $\zeta \in]0,1]$ controls a momentum effect and $p_{\min} < 1/|\Omega|$ serves as a lower bound on the operator probabilities. We initialize $\tilde{p}_o^{(0)} = p_o^{(0)}$ (e.g., $p_o^{(0)} = 1/|\Omega|$) for all $o \in \Omega$. This adaptation algorithm itself has free parameters, but (i) their number is reduced compared to the number of parameters that are adapted, (ii) to our experience, the new parameters are very robust, and (iii) sensible guidelines for their choice exist.

The proposed adaptation scheme has proven to be beneficial in structure optimization of neural networks (NNs). In our experiments, the goal is to find an NN that solves a real-world classification task and has as few as possible degrees of freedom (DOF). It turns out that (i) structure optimization with operator adaptation performs statistically significantly better than optimization without operator adaptation, (ii) the *benefit* tends to give better results than the *local delta*, and (iii) the operator probabilities change in an intuitive way during evolution. Early in the search process, operators that add DOF to the NN perform better. But after NNs that solve the classification task evolved, operators that reduce the DOF are preferred: First, complete nodes are pruned, then the NNs are fine-tuned by removing single connections.

References

[1] C. Igel and M. Kreutz. Operator adaptation in evolutionary computation and its application to structure optimization of neural networks. Technical Report IRINI 2001-03, Institut für Neuroinformatik, Ruhr-Universität Bochum, 44780 Bochum, Germany, 2001.

How to Evolve a Cooperative Population by Minimizing Mutual Information

Yong Liu
The University of Aizu
Tsuruga, Ikki-machi, Aizu-Wakamatsu
Fukushima 965-8580, Japan

Combining individual neural network (NNs) in a population into an NN ensemble has a close relationship with the design of NN ensembles. The population of NNs can be regarded as an ensemble. The evolutionary process can be regarded as a natural and automatic way to design NN ensembles. Based on negative correlation learning and evolutionary learning, an evolutionary learning system EENCL was proposed for learning and designing of NN ensembles [1].

The negative correlation learning and fitness sharing were adopted in EENCL to encourage the formation of species in the population. Fitness sharing refers to a class of speciation techniques in evolutionary computation. The fitness sharing used in EENCL was based on the idea of "covering" the same training patterns by shared individuals. It would be useful to explore possible connection between fitness sharing and information theory. The similarity measurement between two NNs in a population can be defined by the explicit mutual information of output variables extracted by two NNs. The mutual information between two variables, output F_i of network i and output F_j of network j, is given by

$$I(F_i; F_j) = H(F_i) + H(F_j) - H(F_i, F_j) \qquad (1)$$

where $H(F_i)$ is the entropy of F_i, $H(F_j)$ is the entropy of F_j, and $H(F_i, F_j)$ is the joint differential entropy of F_i and F_j. The equation shows that joint differential entropy can only have high entropy if the mutual information between two variables is low, while each variable has high individual entropy. That is, the lower mutual information two variables have, the more different they are. By minimizing the mutual information between variables extracted by two NNs, two NNs are forced to convey different information about some features of their input.

From Eq.(1), we may make the following statements:

1. If F_i and F_j are uncorrelated, the mutual information $I(F_i; F_j)$ becomes very small.
2. If F_i and F_j are highly positively correlated, the mutual information $I(F_i; F_j)$ becomes very large.

Both theoretical and experimental results have indicated that when individual networks in an ensemble are unbiased, average procedures are most effective in combining them when errors in the individual networks are negatively correlated and moderately effective when the errors are uncorrelated. There is little to be gained from average procedures when the errors are positively correlated. In order to create a population of NNs that are as uncorrelated as possible, the mutual information between each individual NN and the rest of population should be minimized. The fitness f_i of individual network i in the population can therefore be evaluated by the mutual information:

$$f_i = \frac{1}{\sum_{j \neq i} I(F_i, F_j)} \qquad (2)$$

Minimization of mutual information has the similar motivations as fitness sharing. Both of them try to generate individuals that are different from others, though overlaps are allowed.

This paper introduces mutual information into EENCL. Through minimization of mutual information, a diverse and cooperative population of neural networks can be evolved by EENCL. The effectiveness of such evolutionary learning approach was tested on two real-world problems.

References

[1] Y. Liu, X. Yao, and T. Higuchi. Evolutionary ensembles with negative correlation learning. *IEEE Transactions on Evolutionary Computation*, 4(4):380–387, 2000.