Bang 3: A Computational Multi-Agent System

Roman Neruda, Pavel Krušina, Petra Kudová[∗], Pavel Rydvan
Institute of Computer Science, ASCR, P.O. Box 5
Prague 8, Czech Republic, bang@cs.cas.cz

Gerd Beuster
Universität Koblenz-Landau
Koblenz, Germany,

Abstract

A multi-agent system targeted toward the area of computational intelligence modeling is presented. The purpose of the system is to allow both experiments and high-performance distributed computations employing hybrid computational models. The focus of the system is the interchangeability of computational components, their autonomous behavior, and emergence of new models.

1. Computational Intelligence

Bang 3 is a platform for the development of Multi-Agent Systems (MAS) [3]. Its main areas of application are computational intelligence methods (genetic algorithms, neural networks, fuzzy controllers) on single machines and clusters of workstations.

Hybrid models, including combinations of artificial intelligence methods such as neural networks, genetic algorithms and fuzzy logic controllers, seem to be a promising and extensively studied research area [1]. We have designed a distributed multi-agent system [4] called Bang 3 that provides a support for an easy creation of hybrid AI models by means of autonomous software agents [2].

Besides serving as an experimental tool and a distributed computational environment, this system should also allow to create new agent classes consisting of several cooperating agents. The *MAS scheme* is a concept for describing the relations within such a set of agents. The basic motivation for schemes is to describe various computational methods. It should be easy to ‘connect’ a particular computational method (implemented as an agent) into hybrid methods, using schemes description. The scheme description should be strong enough to describe all the necessary relations within a set of agents that need to communicate one with another in a general manner.

Here we show, how two computational intelligence methods — artificial neural network of the RBF type, and a genetic algorithm — are represented by a MAS. This further serves for more complicated MAS schemes containing representing hybrid computational intelligence models.

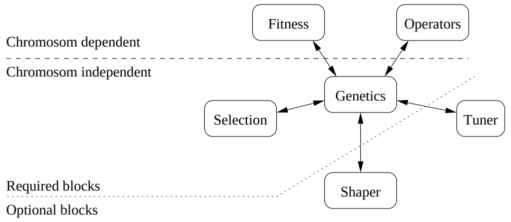

Figure 1. Genetic Algorithm as a multi-agent system.

The genetic algorithm itself, from this point of view, consists of several parts: the *Genetics* agent, which performs the basic genetic algorithm logic and glues all parts together, the *Fitness* agent, that evaluates the fitness function for each individual, the *Operators* agent, that provides genetic operators, metrics operators, and creation operators, the *Selection* agent, that provides the selection of individuals. There are also two optional agent types that can further optimize overall performance: the *Shaper* agent, that provides global processing of population individuals fitness function values — such as sigma scaling — and the *Tuner* agent, that by exploiting information about the genetic algorithm performance (like best individual fitness, average fitness, first and second derivatives of these etc) tunes genetic operators probabilities (cf. Fig 1). It is supposed that there will exist more rival agents implementing a particular function (such as fitness evaluating) and it will be possible to choose among them.

Similar situation is with the Radial Basis Function neural network, which is realized as a MAS containing the central *RBF* agent which cooperates with several other agents realizing training subtasks for it (cf. Fig 2). Namely, it is the *Vector quantization*, *Gradient learning*, and *Least Square Solver*, or the above mentioned *GA*. Again, it is possible

[∗] This research has been supported by the National Research Program Information Society project no. 1ET100300419.

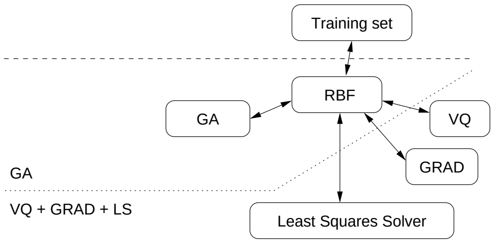

Figure 2. Radial Basis Function Network as a multi-agent system.

that several competing agents realize a particular function for the RBF agent (we have several VQ agents based on Kolmogorov network and k-means clustering, e.g.).

2. Evolution of MAS schemes

The scheme is a set of agents with a given topology of communication channels. The following mechanism for scheme specification has been designed. The agents that comprise the scheme are called *building blocks*. Building blocks communicate through *incoming* and *outgoing gates*. Each agent can have any number of both incoming and outgoing gates. One purpose of the schemes is that the hybrid computational methods designed in a form of scheme can be easily stored and used. Second, perhaps more interesting, challenge of the schemes concept is the *automatic scheme generation*. The scheme definition is a data structure consisting of the list of the building blocks and the interconnection among them. Actually, the scheme definition is a directed acyclic graph. This offers the possibility of automatic searching the space of schemes in order to find a suitable solution.

The proposed evolutionary algorithm operates on schemes definitions in order to find a suitable scheme solving a specified problem. The genetic algorithm has three inputs: First, the number and the types of inputs and outputs of the scheme. Second, the *training set*, which is a set of prototypical inputs and the corresponding desired outputs, it is used to compute the fitness of a particular solution. And third, the list of types of building blocks available for being used in the scheme. We supply three operators that would operate on graphs representing schemes: *random scheme creation*, *mutation* and *crossover*. Fig 3 shows an example of the GA run for the problem of finding suitable representation for data samples taken from a polynomial function. The algorithm finds the MAS realizing the function within three thousands of iterations. Nevertheless, this still means minutes to hours of run on current hardware.

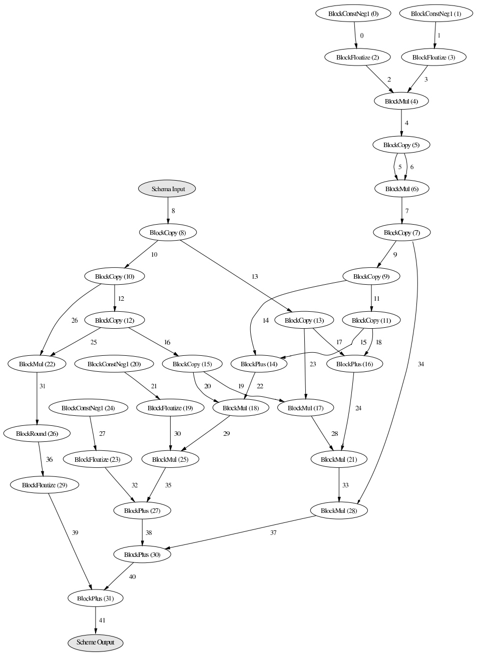

Figure 3. MAS scheme realizing function $x^3 - 2x^2 - 3$, taken from 3000th GA generation.

3. Conclusions and Future Work

We have demonstrated that *Bang 3* is able to help both scientists and end-users with data analysis tasks. The niche for this software has been prototype building and testing various hybrid models so far. However, it is possible to employ it for large scale distributed computations running on a cluster of workstations. The nature of evolution of MAS schemes has brought several issues that are currently being solved. We are building an ontological descriptions of computational agents and data tasks, and we are enhancing the evolution by reasoning component. The resulting hybrid search for MAS solution to a particular problem represented by data should be not only automatic, but also feasible in terms of computational time and resources consumption.

References

[1] P. Bonissone. Soft computing: the convergence of emerging reasoning technologies. *Soft Computing*, 1:6–18, 1997.

[2] S. Franklin and A. Graesser. “Is it an agent, or just a program?”: A taxonomy for autonomous agents. In *Intelligent Agents III*, pages 21–35. Springer-Verlag, 1997.

[3] P. Krušina, R. Neruda, and Z. Petrová. More autonomous hybrid models in Bang. In *International Conference on Computational Science*, pages 935–942, 2001.

[4] G. Weiss, editor. *Multiagent Systems*. The MIT Press, 1999.