Evolutionary Algorithms in Control Systems Engineering

Eduard Ushakov

Abstract – **This paper introduces the session describing the state-of-the-art in the application of evolutionary computation in control system engineering. Evolutionary methods such as genetic algorithms (GAs) and genetic programming (GP) are particularly suitable in problems for which conventional optimizers are inefficient or inappropriate, rather than simply as an alternative to conventional optimization. This session will be of interest to the control engineering community as a whole, and will provide an educational background to those not familiar with these methods as well as presenting new results and applications.**

Keywords - g**enetic algorithms; genetic programming; model predictive control.**

Completeness and harmony in nature are largely the result of evolutionary forces that have adapted species to their surroundings and to each other. Examining natural phenomena, one can appreciate the potential of nature's ready-made solutions, which are often more efficient than man-made ones. For example, consider spider silk, which is more elastic than nylon and stronger than steel, and witness the abilities of the spider, essentially blind with a limited nervous system, to use six variants of silk to build a robust, complex structure in an unpredictable environment.

With nature as a motivator, recent decades have seen increasing attempts to mimic natural evolution using computers (Fogel et al., 1965; Holland, 1975; Koza, 1992). These efforts are stimulated by Darwin's notion of "the survival of the fittest," are generally referred to as *evolutionary computation,* and have been applied successfully to solve particularly complex problems. In evolution, the problem each species faces is one of searching for beneficial adaptations to a changing environment, with the "lessons" learned captured in the chromosomes of its population. Furthermore, evolutionary methods keep track of *populations* of potential solutions, and are thus less sensitive to arbitrary initial guesses of the solution than classical optimization methods, which often rely on local gradient search.

This paper presents a review of the state-of-the-art in the two most prevalent evolutionary computation methods that are applied to control systems engineering: genetic algorithms and genetic programming.

The most commonly-used evolutionary computation algorithm is the *genetic algorithm* (GA). In most implementations, these algorithms manipulate binary encodings of the decision variables to be optimized, which are concatenated into so-called *chromosomes.* Mimicking nature, the algorithm starts its search from an initial population of solutions, usually generated randomly, or occasionally, based on problem-specific knowledge. The performance of each individual is evaluated using a fitness function that gauges its performance, with the most successful (efficient) chromosomes having a higher probability to reproduce. In synthetic evolution, reproduction is similar to that in biological reproduction, which by mimicking natural operators like *crossover* and *mutation* creates a generation of offspring solutions. *Crossover* generates new features in the solution space by combining genetic information, while *mutation* does this by adding random perturbations. The subsequent generations are subjected to these evolutionary operators, thus producing generation after generation of offspring solutions. Since the more appropriate solutions are given higher probabilities to reproduce, one would expect a growing improvement of the solutions over generations.

It has been shown that genetic algorithms are efficient and appropriate optimization methods for control system design (see the excellent survey by Fleming and Purshouse, 2002), for both feedback controllers (e.g. Fonseca and Fleming, 1993; Fonseca and Fleming, 1998) and feedforward controllers (Brown, 1996). All these studies have involved the optimization of the parameters of a control system of fixed structure in order to achieve robust stability and specified performance. The multi-objective approach (MOGA), first suggested by Fonseca and Fleming (1993), permits the modification of the objective as performance is gauged in the course of optimization. This methodology allows multiple, often non-commensurate, performance objectives to be handled separately, and allows goals and priorities for these to be updated by the decision-maker during the course of the optimisation as the effect of trade-offs between them are discovered. MOGAs have been successfully applied in the design of several complex control problems, such as a gas turbine aero-engine, relying on a classical control framework (Chipperfield and Fleming, 1996), as well as in several contributions in this session (Ferreira et al. (2005); Molina-Cristobal et al, 2005; Stirrup and Chipperfield, 2005). It is noted that the MOGA approach is a generalization of the trade-off line approach advocated by Brown (1994), in the automated design of MIMO systems. Exploiting the population-based nature of GAs has shown that statistical hypothesis testing can be applied to a population of successful solutions in order to eliminate controller parameters, which have little impact on the optimal solution.

Dr. Eduard Ushakov – the Chair of Cybernetic Department,
SMC “Agromet”, NAS, Kyiv, UKRAINE
E-mail: eushakov2002@ukr.net
Tel. +38 050 3138670.