Using Neural Networks to Tune Heuristic Parameters in Evolutionary Optimization

MARTIN HOLEŇA
Institute of Computer Science, Academy of Sciences of the Czech Republic
Pod vodárenskou věží 2, 18207 Praha 8
CZECH REPUBLIC

Abstract: Evolutionary optimization algorithms contain, due to their heuristic inspiration, many heuristic parameters, which need to be empirically tuned for the algorithm to work most properly. This paper deals with tuning those parameters in situations when the values of the objective function have to be obtained in a costly experimental way. It suggests to use a neural-network based approximation of the objective function for parameter tuning in such situations. In this way, the convergence speed of the algorithm and the density of the poplation of points can be investigated for many various combinations of heuristic parameters. To construct the approximating neural network, some initial amount of data is needed, usually obtained from running the algorithm for several generations with default values. The feasibility of the approach is demonstrated on a case study in materials science.

Key-Words: Evolutionary optimization, Genetic algorithms, Heuristic parameters, Parameter tuning, Artificial neural networks, Convergeence speed, Population diversity

1. Introduction

Evolutionary and especially genetic algorithms (GAs) have been used since the last decade for optimization tasks in many application domains [3,4,7], mainly for the following reasons:

they do not need gradient nor higher order derivatives of the objective function;

they follow a collection of optimization paths instead of a single one;

they require a comparatively low number of function calls parallelized with respect to the followed paths;

they tend to find global optima rather than local ones.

Due to these reasons, genetic algorithms are particularly suitable for the optimization of functions for which the analytic form is not known and the costs of obtaining function values can not be neglected, typically when those values have to be obtained through experimental measurements. Their suitability further increases if there is a straightforward correspondence between the parallelism of the collection of optimization paths, and the way how the function values are experimentally obtained. Such situation occurs, for example, in materials science or in high-throughput chemical experiments.

On the other hand, the basic principle of evolutionary algorithms, i.e., the imitation of survival in biological evolution, implies that they contain many heuristic parameters. To tune those parameters empirically would substantially increase the cost of solving the optimization task and completely invalidate the advantage of evolutionary and genetic algorithms being comparatively undemanding in terms of parallelized function calls.

This paper shows that instead of tuning the heurisitic parameters by means of the experimentally obtained objective function, an approximation of that function with artificial neural networks can be used. This needs some initial amount of data, usually obtained from running the algorithm for several generations with some default parameter values, but then any number of parameters can be tuned for no additional costs. The principles of that approach are explained, and the approach is illustrated on a case study in materials science [8].

2. Genetic Algorithms and Their Heuristic Parameters

Genetic algorithms [3,4,7] are a stochastic optimization method. This means that when searching for maxima or minima of the objective function, the available information about its values is combined with random influences. The term "genetic algorithm" refers to the fact that their particular way of incorporating random influences into the optimization process has been inspired by the biological evolution of a genotype. Basically, that optimization method consists of:

- random exchange of coordinates of two particular points in the domain of the objective function, called *crossover* or *recombination*;

- random modification of coordinates of a particular point in the input space, called *mutation*; sometimes, a differnece is made between *qualitative mutation* if the coordinate corresponds to a discrete-valued attribute, the individual values of which represent qualitatively different states of objects, and *quantitative mutation* otherwise, especially if the coordinate corresponds to an interval-valued attribute;

- *selection* of the points for crossover and mutation according to a probability distribution, either uniform or skewed towards points at which the objective function takes high values (the latter being a probabilistic expression of the survival-of-the-fittest principle).

The operations selection, crossover and mutation require only function values of the objective function, and they can be performed in parallel for a whole population of points. At the same time, the incorporated random variables enable the optimization paths to leave the attraction area of the nearest local optimum, and to continue searching for a global one. The probability that at least one optimization path will reach the global optimum increases with the diversity of the population of points. On the other hand, those random variables heavily depend on the underlying probability distributions of possible values. Due to the biological inspiration of the random variables, a particular distribution can not be justified mathematically, but its choice is a heuristic task. The most important heuristic choices entailed by a genetic algorithm are:

- *overall probability of any modification* (crossover, qualitative or quantitative mutation) of an individual;

- *ratio between the conditional probabilities* of crossover and qualitative or quantitative mutation, conditioned on any modification;

- *distribution of the intensity of quantitaive mutation*, e.g., distribution the coefficient with which the respective coordinate has to be multiplied / divided.

In addition, also the population size is sometimes a matter of heuristic choice, though in other cases it is determined by hardware limitations of how the values of the objective function are obtained (e.g., the number of measurement sensors, the number of channels in a chemical reactor).

3. Tuning Heuristic Parameters by Means of Artificial Neural Networks

If the values of the objective function have to be obtained in a costly experimental way, then it is not affordable to employ that costly evaluation also for tuning the involved heuristic parameters. The aim of this paper is to suggest that, in such situations, a *neural-network based approximation of the objective function* be used for parameter tuning, instead of that function itself. The cost of computing such an approximation in a population of points even for a large number of combinations of values of heuristic parameters is negligible compared with the cost of experimentally evaluating the original objective function in those points for any single combination of their values. In addition, experimental evaluation of one generation of individuals typically needs hours to days of time, whereas if the neural-network based approximation is computed instead, the genetic algorithm advances to the next generation in a fragment of second. The algorithm then can be run for many generations in that case, allowing to investigate, through this neural-network simulation, its *convergence speed* for any particular combination of values of heuristic parameters. Moreover, also the evolution of the *diversity of the population of points* can be investigated in that way, which is important due to the role that the diversity plays in enabling optimization paths to reach the global optimum.

Needless to say, training the neural network requires some *initial amount of data* from experimentally evaluating the objective function to be gathered first. To this end, data from several early generations of the genetic algorithm are usually sufficient, especially if the population size is large. Actually, data from the early generations are more uniformly distributed, making it more likely that the neural network will correctly approximate all optima, including the global optimum. Sometimes, also data from other experiments concerning the same problem are available. Once the parameters have been tuned, they can be used for all the remaining generations of the algorithm, or the tuning can be repeated every several generations, using a new neural network. In the latter case, the data from experimentally evaluating the objective function that were gathered since the last tuning are addet to the training data of the network.

An obvious drawback of the proposed apporach is that until the amount of data needed for network training gets available, the values of heuristic parameters remain untuned, and have to be set to some kind of default values. However, first experience with the approach confirms the expectation that this drawback is outweighed with the possibility to investigate, for any combination of values of heuristic parameters, the convergence speed of the algorithm and the evolution of the diversity of the population of points.

The approach has been already used in several appliacations in chemistry and materials science. One of them will be described in some detail in the following section.

4. A Case Study Using the Proposed Approach

Most of industrially important chemical reactions are catalysed, and the choice of their catalysts can significantly influence the yield and cost of the desired reaction product(s), as well as the production of nuisant by-products, which in turn entail separation and cleaning costs of the desired product(s). Therefore, much attention has been paid to the optimization of catalysts for important reactions, including their optimization by means of genetic algorithms. In this case study, heuristic parameters of a genetic algorithm implemented specificaly for the purpose of catalyst optimization [9] have been tuned on data about 328 materials tested as catalysts for the oxidative dehydrogenation of propane to propene [2]. The materials consisted of a mesoporous alpha-aluminium support covered externally and internally with metal oxide mixtures selected from elements B, Fe, Ga, La, Mg, Mn, Mo, and V. All the materials were tested at the same standard conditions: T = 273 K, $p_{C_3H_8}$ = 30,3 kPa, p_{O_2} = 10,1 kPa and p_{N_2} = 60,6 kPa. The data came from running the genetic algorithm for the 1.–5. generation with default parameters and population size 60, and were complemented with 28 catalysts from supplementary experiments.

With those data, a series of multilayer perceptrons have been trained. To obtain perceptrons with possibly high generalization ability, two overtraining-reduction methods have been used – early stopping and Bayesian regularization. The atomic proportions of all included oxides of these elements are connected through the concentration constraint (requiring that they have to sum up to 100 %), hence one of those proportions is always derivable from the remaining seven and cannot be used as an input to the network. In the reported experiments, the proportion of V has been chosen to this end. Each multilayer perceptron that approximates the dependency of the propene yield on the proportions of the oxides of considered metals must then have 7 input neurons and one output neuron. As far as hidden neurons are concerned, a restriction to perceptrons with one hidden layer was made. This restriction was based on the one hand on the fact that even multilayer perceptrons with one hidden layer have the universal approximation property [1,6], on the other hand on the experience from research reported in [5], in which other data of the same reaction were used for training one- and two-hidden-layer perceptrons. In that research, architectures with only one hidden layer showed a clearly better generalization ability. To find the most appropriate number of neurons in the single hidden layer, perceptrons with 1–20 neurons were trained in combination with each of the considered overtraining reduction methods and their generalization error, estimated with mean squared error on test data was compared. For the experiments with the genetic algorithm, the combination of architecture and overtraining reduction method leading to the least generalization error was selected. This was the architecture with 5 hidden neurons, combined with early stopping. In accordance with the proposed approach, the function computed by that network was used as an approximation of the unknown objective function, which was the yield of C_3H_6 in the considered reaction. To evaluate the success of the genetic algorithm in searching for the global maximum of the approximation, that maximum was first estimated with a deterministic optimization method. To this end, the Levenberg-Marquardt method was used, more precisely its modification for constrained optimization. Since that method searches only for local maxima, it was started from 7 different starting points. The highest of those local maxima will be in the sequel considered as global maxima of the approximation.

The approach was applied to 24 particular combinations of heuristic parameters, combined with 4 different population sizes.

For each of those 96 final combinations, the algorithm was run repeatedly 10 times. This entails a certain variability of the obtained results, documented in Fig. 1 and Fig. 2. The significance of the information conveyed by those figures consists in the fact that it indicates which part of the difference between the presumable global maximum of the approximation and the maximum found by the genetic algorithm and which part of the population diversity were due to the randomness-caused variability. In particular, repeated runs of the GA accounted in average for an absolute error of 0.07 % of the maximal propene yield found by the algorithm (corresponding to a relative error 0.8 %), and to an absolute error of 0.1 % of the population diversity of propene yield (corresponding to a relative error nearly 100 %).

For each final combination, the following results have been recorded:

- The convergence of the algorithm during the first 50 generations, which is measured as the evolution of the maximal value of the objective function among the population of points

- The decreasing diversity among the population of points during the first 50 generations, where diversity is measured as the difference between the maximal and mean value of the objective function among the population of points.

In those results, the following indicators of convergence have been employed:

(i.) Average of the value of the global maximum of the objective function, and the maximal value of the objective function among the population of points proposed in the first generation, i.e., the value "½(global maximum + 1st generation)".

(ii.) Global maximum of the objective function minus 0.8 % relative error, which was the average relative error due to repeated runs of the algorithm with an identical combination of values of heuristic parameters.

(iii.) The global maximum of the objective function within one decimal digit precision.

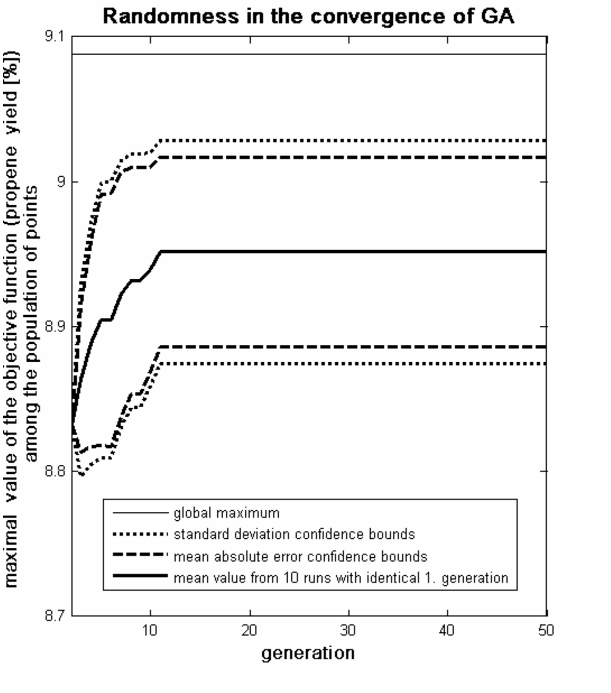

Fig. 1. Example of randomness-caused variability in convergence of the GA: the algorithm was run repeatedly 10 times with an identical population size and an identical combination of values of heuristic parameters, starting from the an identical first generation

As indicators of decreasing diversity among the population of points, the following have been employed:

(i.) Half the diversity in the first generation.

(ii.) Diversity 1 %.

(iii.) Diversity 0.1 %.

The most important knowledge pertaining to the choice of the population size and of the values of heuristic parameters, which has been obtained from running the genetic algorithm with the 96 considered combinations, can be summarized as follows:

1. The indicators of convergence that were fulfilled for a given combination of values of heuristic parameters were typically fulfilled already after early generations. Moreover, distributions of the number of generations needed to fulfil the three

considered indicators of convergence are clearly skewed towards low values. Those distributions are depicted in Fig. 3, which thus significantly documents that early generations are much more important in optimization by means of genetic algorithms than later generations.

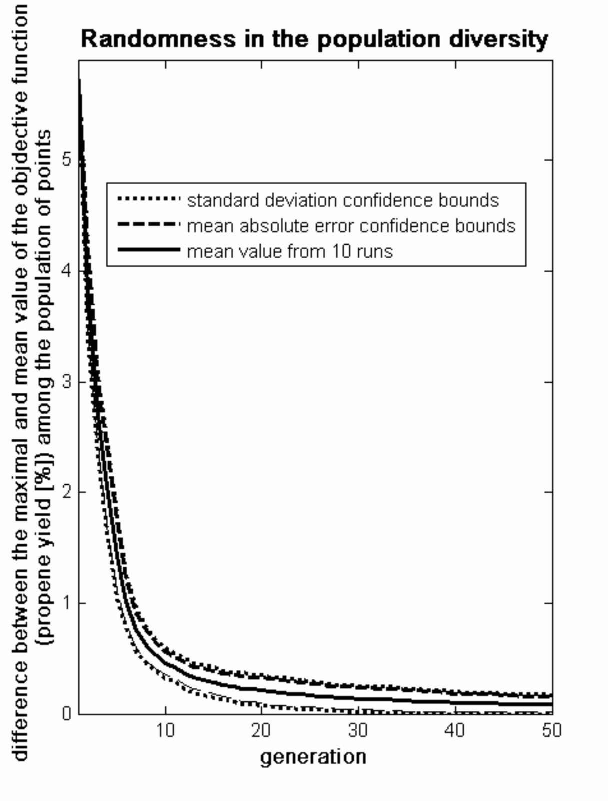

Fig. 2. Example of randomness-caused variability in the diversity of the population of points: the algorithm was run repeatedly 10 times with an identical population size and an identical combination of values of heuristic parameters, starting from an identical first generation

2. The convergence speed of the GA tends to increase with increasing population size (see Fig. 4 for an example).

3. For a given generation above approximately the 10th generation, the maximal value of the objective fuction among the population of points tends to increase with increasing population size (cf. Fig. 4).

4. The diversity among the proposed points decreases more quickly if crossover and mutation occur with equal probability than if one of them substantially prevails (Fig. 5).

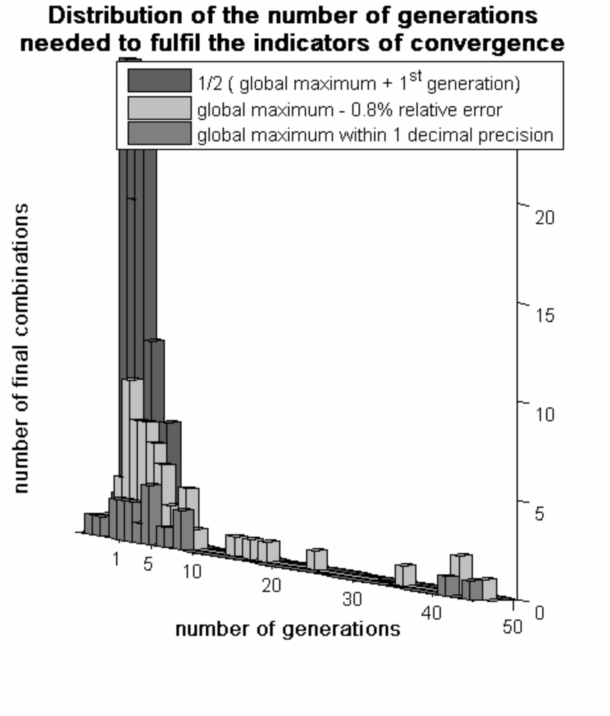

Fig. 3. Distribution of the number of generations needed for the value of the objective function among the population of points to fulfil three increasingly strong indicators of convergence of the GA (the algorithm has been run till the 50th generation, irrespectively of which of those indicators had been fulfilled)

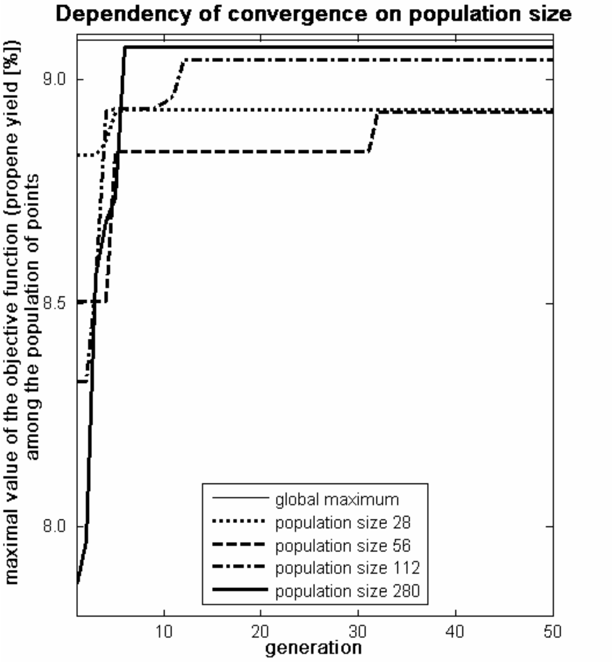

Fig. 4. Example dependency of the maximal value of the objective function among the population of points on the generation, for 4 considered population sizes

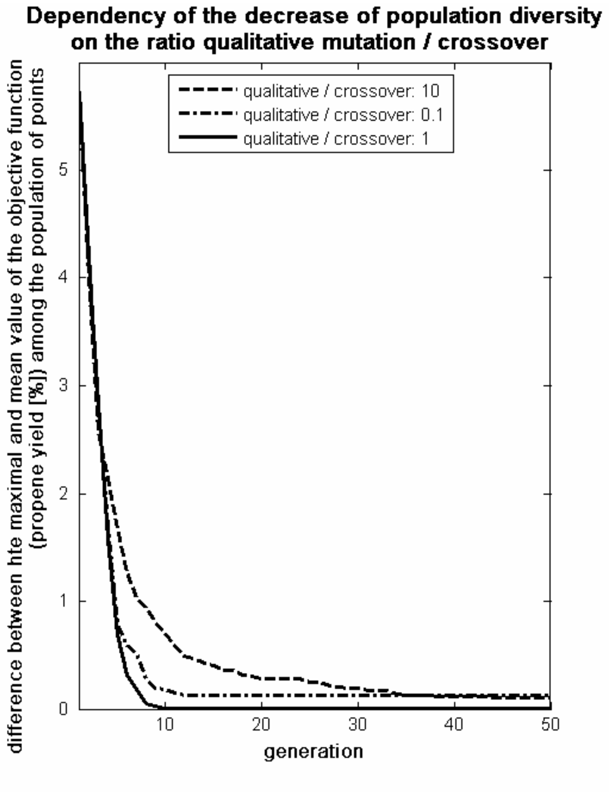

Fig. 5. Example illustrating that the diversity of the population of points in one generation of the GA decreases more quickly if crossover and qualitative mutation occur with equal probability than if one of them substantially prevails

5. Conclusion

This paper dealt with the problem of tuning heuristic parameters of genetic algorithms in situations when the values of the objective function have to be obrtained in a costly experimental way. It suggested to use a neural-network based approximation of the objective function istead of the function itself in such situations. In this way, it is possible to investigate the convergence speed of the algorithm and the diversity of the population of points for many various combinations of heuristic parameters. The feasibility of the approach has been demonstrated on a case study in materials science. The study has shown that by means of the aproximation, knowldge useful for the choice of the population size and the values of heuristic parameters can be obtained. So far, only multilayer perceptrons have been used in the applications of the approach, extensions to other kinds of artificial neural networks are a subject of ongoing research.

Acknowledgement

The research reported in this paper has been supported by the grant No. 201/05/0325 of the Grant Agency of the Czech Republic and partially supported by the Institutional Research Plan AV0Z10300504.

References:

1. G. Brightwell, C. Kenyon, H. Paugam-Moisy, Multilayer neural networks: one or two hidden layers? In *Mozer, Jordan, Petsche, Eds., Advances in Neural Information Processing Systems,* MIT Press: Cambridge, 1997*, 148-154.*

2. O.V. Buyevskaya, A. Brückner, E.V. Kondratenko, D. Wolf, M. Baerns, Fundamental and combinatorial approaches in the search for and optimisation of catalytic materials for the oxidative dehydrogenation of propane to propene. *Catalysis Today*, 67 (2001), 369–378.

3. D.B. Fogel, *Evolutionary Computation: Toward a New Philosophy of Machine Intelligence*, 2nd Edition, IEEE Press: New York, 1999.

4. A.A. Freitas, *Data Mining and Knowledge Discovery with Evolutionary Algorithms*, Springer: Berlin, 2002.

5. M. Holeňa, M. Baerns, Artificial Neural Networks in Catalysts Development. In *Cawse, Ed., Experimental Design for Combinatorial and High Throughput Materials Development*, Wiley: New York, 2003, 163–202.

6. K. Hornik, Approximation capabilities of multilayer neural networks. *Neural Networks*, 4 (1991), 251–257.

7. J.R. Koza, F.H. Bennett, D. Andre M.A. Keane, *Genetic Programming III: Darwinian Invention and Problem Solving*, Morgan Kaufmann: San Francisco, 1999.

8. U. Rodemerck, M. Baerns, M. Holeňa, D. Wolf, Application of a genetic algorithm and a neural network for the discovery and optimization of new solid catalytic materials. *Applied Surface Science*, 223 (2004), 168–174.

9. D. Wolf, O.V. Buyevskaya, M. Baerns, An evolutionary approach in the combinatorial selection and optimization of catalytic materials. *Applied Catalyst A: General*, 200 (2000), 63–77.