Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 169 (2020) 168–172

Procedia Computer Science

www.elsevier.com/locate/procedia

Postproceedings of the 10th Annual International Conference on Biologically Inspired Cognitive Architectures, BICA 2019 (Tenth Annual Meeting of the BICA Society)

Educational Intelligent System Using Genetic Algorithm

Julia Protopopova[1] and Sergey Kulik[1]

National Research Nuclear University MEPHI, Kashirskoe shosse 31, Moscow 115409, Russia

Abstract

This paper presents an Intelligent Information System for education. The system was created for teaching students to use genetic algorithm in application to optimization tasks. The system allows to quickly encode a solution of the problem and pick up most suitable configuration of genetic algorithm. The paper also demonstrates a specific example of usage of educational system to solve an optimization task. The paper contains a description of the educational system and its features, a description of its capabilities of working with genetic algorithm and its graphical interface.

Keywords: Genetic Algorithm, Optimization Tasks, Intelligent System.

1. Introduction

The use of information technologies (IT) in various spheres of human activity, the exponential growth of information volumes and the need to respond quickly in any situations necessitate the search for adequate ways to solve new and new emerging problems [1]. One of the most effective ways is the study, application, development and further improvement of algorithms and methods of machine computing.

Finding the optimal solution is one of the problems that researchers face, both in the theoretical field and in solving engineering and production problems. Optimization problems imply the need to choose from the set of possible solutions the best from the point of view of certain criteria, satisfying given conditions and limitations.

10.1016/j.procs.2020.02.130

There are classes of optimization problems (the so-called NP-complete problems), the solution of which cannot be found without a complete enumeration of all possible solutions. In particular, the latter include many varieties of multicriteria optimization problems. Due to the large time costs and the large dimension of these tasks, the implementation of enumeration of options is almost impossible. In this situation, an alternative approach to solving the mentioned problems is the use of methods based on the methodology of evolutionary computing.

This work is devoted to genetic algorithms (GA) as a universal method for optimizing multiparameter functions [2]. GAs essentially are search algorithms based on selection and genetics mechanisms. The use of GA in solving engineering and industrial problems allows us to reduce the amount and time of calculations, simplify the modeling of functions, and reduce the number of modeling errors.

Genetic algorithms can be used in different fields for example: cognitive technology [6], Big and Fast Data [7], mobile robot [8], IT for Regular Agent [9], information retrieval [10] and other fields. Note that Bayesian Network [11] can be used in IT.

2. The Principle of GA

The genetic algorithm is a stochastic algorithm for finding the optimal solution. It belongs to the group of heuristic ones, and is applicable to a fairly wide range of problems [3]. It is stochastic, since its result is based on the use of random/pseudo-random variables and generators. The most important feature of these is being heuristic: it does not have a rigorous theoretical justification, but is justified practically, i.e. it should give a practical suitable result.

GA involves the principles and terminological apparatus of genetics. Each individual in the GA is a potential solution to the task. The search for the optimal or suboptimal solution is carried out during the evolution of the population of individuals, i.e. sequential conversion using genetic operators of reproduction, crossover and mutation from one finite set of solutions to another.

The fundamental concept in genetic algorithms is the goal (or objective) function, which is designed to assess the characteristics of an individual relative to the ultimate goal. This function allows to determine the degree of fitness of specific individuals in the population and calculate among them the most adapted, i.e. having maximum fitness function values [4].

There are some existing implementations of genetic algorithms that have their pros and cons. They include the gmulti package for R programming language, Genetic Algorithm Tool for Matlab that includes gatool utility, and DEAP package for Python. What is common for them is that they are not aimed at learning. Also only Matlab package includes graphical interface for configuring and launching GA. Due to these limitations, we have decided to create an implementation that allows to focus on teaching students how to work with GA and on quick prototyping of a solution.

3. The GALA software system

The GALA (Genetic Algorithm Learning Application) have been developed in order to teach students how to use GAs when solving applied problems. The main goal of system development is to provide a simple and intuitive tool for studying the use of genetic algorithms in solving applied problems.

The system has modular architecture consisting of 3 main sections that group its functions by areas of operation:

- The “Objects” subsystem that works with user-defined objects like chromosomes, genotypes and functions and allows to create, edit, remove and browse them;
- The “Problems” subsystem that is designed to manage user’s problems and their solutions. It allows to create, edit and remove any created problem, browse them, and browse and manage all solutions that have been made for particular task;
- The “Import/Export” subsystem is assigned to synchronize data between client app and database server.

The GALA system is implemented as a three-tier client-server architecture and consists of the following components: a database server running MongoDB database management system, a web-server implemented on NodeJS platform using Express framework, and client web-app that processes mathematical computations (with use of MathJS library) and renders a graphical user interface (based on VueJS framework).

The genotype of an individual is a way of encoding a solution to a problem being solved. It is built of chromosomes, which, in turn, are built from a sequence of genes. In the GALA system, a gene is a predefined combination of encoded and decoded representations of a decision gene. Genes can be of the following types:

- *Int* is an integer both in the genotype (encoded solution) of the chromosome and in its phenotype (decoded solution). Mutations are made by changing its value by adding a random integer picked from a given range;
- *Float* is represented by a fractional number in both the encoded and decoded representation of a solution. Mutation occurs similarly to the Int gene type;
- *BinaryFloat* in encoded form is represented by a binary number, and in decoded form by a floating-point number with a given accuracy. A mutation is a random inversion of a gene locus;
- *BinaryInt* in encoded form is represented by a sequence of bits that take the values “0” or “1” (Boolean vector). In decoded form, is an integer. Mutations occur by changing the value of a randomly selected bit of an encoded representation;
- *GrayInt* in encoded form is represented by Gray's binary code, and in decoded form by a whole number. The advantage of the Gray code is that the encoded representation of the phenotype of neighboring numbers differs by only one bit. Thus, during mutations, sharp losses of good decisions are eliminated. Mutation occurs by inverting the locus of a gene.

To describe functions, including the goal function, the system uses a special language that allows them to be described as mathematical expressions. Language implementation is provided by math.js library.

The configuration of solution search with use of the genetic algorithm is done with the following parameters:

- Genotypes of the individuals processed by algorithm;
- Goal function to maximize/minimize;
- Specific values of initial individuals;
- A number of independent populations being processed;
- The strategy of selection of individuals into intermediate population and a fraction of selected ones;
- The pair form strategy;
- A number of points and a probability of a crossover;
- A number of points and a probability of a mutation;
- The strategy of merging intermediate and main population;
- Termination criteria: maximum of iterations, achievement of desired goal function value, no significant change of average goal function in populations (stagnation).

3.1. Example of a Problem Solution Using GALA

As mentioned above, the algorithm can be applied to solve engineering and production optimization problems. Usually, the number of combinations in the case of direct enumeration of options is too large, while GA allows one to obtain a solution quite close to optimal in a relatively small number of steps. Consider an example of such a task: let there be a need to provide a settlement with disinfected water. This can be done using five different models of disinfecting stations, each of which has a specific performance and consumption of three resources (fuel, electricity, disinfecting reagents). There is also a limit on the maximum number of stations of each type used. It is necessary to ensure maximum productivity of the complex of stations in the presence of restrictions on resource reserves. The numerical parameters of the problem are given in table 1.

Each solution to this problem is represented of five positive integers showing the number of stations used in the corresponding model. The total number of existing combinations of these numbers, taking into account the available number of stations (and therefore the number of options that need to be checked, in the case of direct enumeration) is $481 \times 81 \times 181 \times 121 \times 101 = 86\ 181\ 770\ 961$, i.e. over 86 billion possible configurations. In this paper, the task is given to demonstrate the operation of the GA and evaluate its effectiveness. The optimal solution to this problem is known and equals to 6650.

Table 1. Parameters of the problem.

Resource	Disinfection station model					Resource availability
	A	B	C	D	E	
Fuel (L/h)	4	8	12	9	10	3690
Reactive (L/h)	0.4	0.6	0.3	0.2	0.3	432
Energy (kW)	0	0	6	4	5	2400
Performance (m3/h)	8	10	16	13	17	
Available quantity	480	80	180	120	100	

To solve the problem, a genotype with a single chromosome was created in the system. The chromosome represents a configuration of a purifying complex and contains five integer genes (of type Int). Let the name of the chromosome be c, and the names of the genes - *a, b, c, d, e* (each gene describes the number of corresponding stations used). This encoding of the solution does not take into account the restrictions imposed on the stock of resources. To do this, we use the penalty function (1), which will significantly reduce the value of the objective function for incorrect decisions. To calculate it, we additionally introduce 3 functions for calculating the overconsumption of resources: *fl* (2), *rc* (3) and *en* (4) (for fuel, reactive and energy, respectively):

$$penalty(a, \dots, e) = \begin{cases} 10000, fl(a, \dots, e) > 0 \vee rc(a, \dots, e) > 0 \vee en(a, \dots, e) > 0 \\ 0, \; otherwise \end{cases} \tag{1}$$

$$fl(a, b, c, d, e) = 3690 - 4a - 8b - 12c - 9d - 10e \tag{2}$$

$$rc(a, b, c, d, e) = 432 - 0{,}4a - 0{,}6b - 0{,}3c - 0{,}2d - 0{,}3e \tag{3}$$

$$en(a, b, c, d, e) = 2400 - 6c - 4d - 5e \tag{4}$$

Thus, the goal function for the problem will have form

$$goal(o) = 8o.c.a + 10o.c.b + 16o.c.c + 13o.c.d + 17o.c.e - fine(o.c.a, \dots, o.c.e) \tag{5}$$

The following configuration was supplied to GA:

- The only independent population is being processed;
- Intermediate population is selected with ranked selection strategy;
- Positive associative crossover strategy;
- Single point crossover with probability of 1;
- Single point mutation with probability of 0.1;
- Elitary merge strategy with 10% of individuals saved from previous generation;
- Stop criterion is achieving goal function value of 6550.

With these parameter values, the GA finds a solution close to optimal (above 6550, i.e., within 98% of the optimal) on average over 778 generations (in a series of 30 experiments). The average number of actually tested solutions is 778 × 30 = 23,340, which is significantly lower than the full number. Thus, GA is rational to use to search for practical applicable results when solving engineering problems in the conditions of limited computing resources and time.

4. Conclusions

Optimization tasks are ubiquitous in almost every area of industry. There are a lot of them that require full enumeration of solutions to find the optimal one. One of popular and efficient methods that can deal with such problems is the genetic algorithm. It has a lot of various enhancements that suit specific cases, as well as numerous implementations for different programming languages. However, none of them concentrates on teaching and educational purposes. This paper introduces the educational system GALA that has been created to teach students how to apply GA to different optimization problems. The system allows to quickly encode a solution of the problem and pick up most suitable configuration of GA. The paper also demonstrates a specific example of usage of GALA system to solve an optimization task. We will continue to make efforts in developing our results for various fields for example: Fast Data [7] and convolutional neural network [8], special agent technology [9] and factographic information retrieval [10, 17], electronic document [13], decision-making system [14] and situational express analysis [15], fuzzy business processes [16] and special cognitive technology [18]. This will help researchers to speed up the development of special algorithm solutions to various important tasks.

References

[1] Aseev G. G. (2011) "Methods of data mining in electronic storages: genetic algorithms." *Radio electronics, computer science, management* **2** (**25**): 82-85.

[2] Alzhanov A. K. et al. (2017) "Problem Solving and a General Approach to Prediction Using Neural Networks." *International Scientific and Practical Conference World science* **2**(**3**): 26-31.

[3] Yandybaeva N. V. (2009) "Genetic algorithm in the problem of optimizing the university curriculum." *Modern high technology* **11**: 97-98.

[4] Batishchev D. I., Neymark E. A. and Starostin N. V. (2006) "Application of genetic algorithms for solving discrete optimization problems." N. Novgorod: Publishing House of Nizhny Novgorod State University: 16-27.

[5] Mastitsky S.E., Shitikov V.K. (2014) "Statistical analysis and visualization of data using R." - E-book, access address: http://r-analytics.blogspot.com

[6] Samsonovich, A.V.: On semantic map as a key component in socially-emotional BICA. Biol. Inspired Cogn. Arch. 23, 1–6 (2018).

[7] Miloslavskaya, N., Tolstoy, A.: Big Data, Fast Data and Data Lake Concepts. Procedia Computer Science, vol. 88, pp. 300-305 (2016).

[8] Verbitsky, N.S., Chepin, E.V., Gridnev, A.A.: Experimental studies of a convolutional neural network for application in the navigation system of a mobile robot. Procedia Computer Science, vol. 145, pp. 611-616 (2018).

[9] Artamonov, A., Onykiy, B., Ananieva, A., Ionkina, K., Kshnyakov, D., Danilova, V., Korotkov, M.: Regular Agent Technologies for the Formation of Dynamic Profile. Pro-cedia Computer Science, vol. 88, pp. 482-486 (2016).

[10] Kulik, S.D.: Factographic information retrieval for semiconductor physics, micro - and nanosystems. AMNST 2017, IOP Conference Series: Materials Science and Engineering, 498, 012026 (2019).

[11] Zaharov, A.S., Borisov, V.V., Senkov, A.V. Accident Risks Assessment by Temporal Fuzzy Bayesian Network. International Journal of Applied Engineering Research, vol. 11, Number 22, pp 10731-10736 (2016).

[12] Borisov, V.V., Garanin, O. I.: A Method of Dynamic Visual Scene Analysis Based on Convolutional Neural Network. Proc. of 16th Russian Conference on Artificial Intelligence, RCAI-2018 (Moscow, Russia) Springer: Communications in Computer and Information Science vol. 934, pp 60–69 (2018).

[13] Artamonov, A., Ionkina, K., Tretyakov, E., Timofeev, A.: Electronic document pro-cessing operating map development for the implementation of the data management system in a scientific organization. Procedia Computer Science, 145, pp. 248-253. (2018).

[14] Gridnev A.A., Voznenko T.I., Chepin E.V.: The Decision-Making System for a Multi-Channel Robotic Device Control. Procedia Computer Science, vol. 123, pp. 149-154 (2018).

[15] Koshlan, D.I., Tretyakov, E.S., Korenkov, V.V., Onykij, B.N., Artamonov, A.A.: Agent technology situational express analysis in assessment of technological development level of the BRICS countries. CEUR Workshop Proceedings, 2267, pp. 436-440 (2018).

[16] Senkov, A., Borisov, V. Risk assessment in fuzzy business processes based on high level fuzzy petri net. International Journal of Applied Engineering Research vol. 11, Number 16, pp 9052-9057 (2016).

[17] Kulik, S.: Factographic Information Retrieval for Competences Forming. The Third In-ternational Conference on Digital Information Processing, Data Mining, and Wireless Communications (DIPDMWC2016), Proceedings, July 06-08, Moscow, Russian Federation, pp. 245-250 (2016).

[18] Ushakov, V.L., Alexei V. Samsonovich, A.V.: Toward a BICA-Model-Based Study of Cognition Using Brain Imaging Techniques. Procedia Computer Science, vol. 71, pp. 254–264 (2015).