APPLICATION OF GENETIC ALGORITHMS FOR INCREASE OF AN OVERALL PERFORMANCE OF ARTIFICIAL NEURAL NETWORKS

Margarita Karpunina[1]

[1]Nizhniy Novgorod State Technical University, Faculty of Information Systems & Technologies, Dept. of Applied Mathematics, Russia, Nizhniy Novgorod, Minina 24, 603024. levati@yandex.ru

ABSTRACT

In the given paper utilization of genetic algorithms to increase performance of neural networks is concerned. This issue is described by example of the synthesis of optimal logical structure of distributed databases. Author suggests an approach where genetic algorithm plays a role of "tuning" framework for specialized Hopfield neural network. In this case a neural network algorithm gives in generally local-optimum decisions of an extreme task of data base's logical structure synthesis. Within the framework, the function of genetic algorithm is the selection among the obtained local-optimum decisions global-optimum, satisfying to the given constraints of a task. The large number of experiments shows suitability of the proposed approach and allows synthesizing optimal logical structures of distributed databases in presence of various constraints.

1. INTRODUCTION

The problems of the optimum decisions acceptance arise in various areas of science and engineering, rendering decisive influence on scientific researches [1]. The common point for these problems is that their mathematical model can be formulated as an optimization task connected to search of such values of controlled variables, which provide extreme value for one of the most important technical and economic characteristics of the object, if other characteristics satisfy to the given set of the technical requirements. Thus the main difficulties of the numerical decision of the extreme task are connected to its dimension and a form of optimized function, which generally can be nonlinear, explosive, non differential and multi-extreme.

One of the approaches allowing overcoming specified difficulties is the evolution-genetic approach, which allows building algorithms of search of the optimum decisions named by ***genetic algorithms***, on the basis of modeling biological mechanisms of population genetics.

The task solved in given paper, is taken from area of synthesis optimum structures of *the Distributed Databases* (DDB), belongs to a class of nonlinear tasks of integer programming and is a NP-complete task.

The practice of DDB logic structures development recommends using heuristic methods of designing the rational and formalized methods of synthesis the optimum DDB logic structures [2]. Different kinds of heuristic methods are therefore often used to find reasonably good solutions. *The Artificial Neural Networks* (ANN) approach falls within this category.

The ANN approach allows obtaining a set of locally optimum decisions of optimum DDB logic structure synthesis task. Thus each of such decisions corresponds to the certain values of factors-parameters of energy function of neural network. As there are no regular ways of definition of these factors values, it was decided to use genetic algorithms for a finding of parameters appropriate to the optimum task decision.

2. STATEMENT OF A TASK

The task, considered in the given work, consists in optimal synthesis of DDB logical structure by criteria of a minimum general time of consecutive processing of the set of users queries.

For the proved construction of the task's goal function it is necessary to discribe the characteristics used for synthesis of DDB logic structures. The formalized descriptions and characteristics of initial DDB structure, set of queries, DDB users, hosts and topology of the computer networks (CN), average initial time characteristics other usfull variables are shown in [2].

The soluble task can be formulated as follows.

The goal function of a task:

$$\min_{\{x_{it},y_{tr}\}} \sum_{k=1}^{K_0}\sum_{p=1}^{P_0}\xi_{kp}^{3}\cdot\varphi_{kp}^{3}\cdot\{\sum_{r1=1}^{R_0}\nu_{kr1}\cdot[\sum_{r2=1}^{R_0}z_{pr2}\cdot(t^{раз}+t_{r1r2}^{сер}+t_{r1r2}^{пер}\cdot(1+ \\ +\sum_{t=1}^{t_0}z_{pr2}^{t}))+t^{сб}]+\sum_{r2=1}^{R_0}\sum_{t=1}^{t_0}z_{pr2}^{t}\cdot(t_{r2}^{n}+t_{r2})\} \quad \textbf{(1)}$$

Constraints of a task: 1) on number of groups in logic record $\sum_{i=1}^{I}x_{it}\le F_t,\ \forall t=\overline{1,t_0}$, where F_t − maximum quantity of groups in t record; 2) on single groups inclusion in logic record $\sum_{t=1}^{t_0}x_{it}=1,\ \forall i=\overline{1,I}$; 3) on length of formed logic record $\sum_{i=1}^{I}x_{it}y_{tr}\rho_i\psi_0\le\theta_{tr}$, where θ_{tr} − as much as possible allowable length of t record determined by characteristics of the server r; 4) on general number of synthesized logic records placed on the r server $\sum_{t=1}^{t_0}y_{tr}\le h_r$, where h_r − maximum quantity of logic records supported by local database management system of the r server-host; 5) on volume of accessible external memory of servers for a storage of local databases $\sum_{t=1}^{t_0} \sum_{i=1}^{I} \psi_0 \rho_i \pi_i x_{it} y_{tr} \le \eta_r^{\text{ВЗУ}}$; 6) on a required level of information safety of the system $x_{it} x_{i't} = 0$ for given d_i and $d_{i'}$; 7) on a total processing time of operative queries on servers $\sum_{r2=1}^{R_0} \sum_{t=1}^{t_0} z_{pr2}^{t} \cdot (t_{r2}^{n} + t_{r2}) \le T_p$ for given $З_p \in З$, where T_p − allowable processing time of p operative search.

3. BASIC IDEA OF THE OFFERED APPROACH

The decision of the task (1) is divided into two stages. For each stage the neural network model is constructed. The approach used in translation process of mathematical statement of our task in the terms of neural network is described in detail in paper [3]. The description of neural network model for each stage is a separate question, and it will not be considered here. Let's note only, that for each stage of the task's decision Hopfield neural networks are used.

As a result of translation of constraints in the terms of constructed neural networks the following energy functions of networks for the 1st and 2nd stages of the task's decision are received:

1 stage (splitting of the set of data groups into types of logic records)

$$E = -\frac{1}{2} \cdot \sum_i \sum_j \sum_x \sum_y [-A_1 \cdot \delta_{xy} \cdot (1-\delta_{ij}) + B_1 \cdot \delta_{ij} \cdot (1-\delta_{xy}) \cdot (2 \cdot a_{xy}^{г} - 1) - C_1 - E_1 \cdot \delta_{ij} \cdot (incomp\_gr_{xy} + incomp\_gr_{yx})] \cdot OUT_{xi} \cdot OUT_{yj} + \sum_i \sum_x [\frac{B_1}{2} \cdot \sum_{y \ne x} (a_{xy}^{г})^2 + \frac{D_1}{2 \cdot F_i}] \cdot OUT_{xi} - \sum_i \sum_x [C_1 \cdot n] \cdot OUT_{xi} + \frac{C_1}{2} \cdot n^2 \qquad \textbf{(2)}$$

2 stage (excessless accommodation of the synthesized types of records in computer network)

$$E = \frac{A_2}{2} \cdot \sum_{t=1}^{T} \sum_{r_1=1}^{R_0} \sum_{r_2=1}^{R_0} OUT_{tr_1} \cdot OUT_{tr_2} + \frac{B_2}{2} \cdot \left[\left(\sum_{t=1}^{T} \sum_{r=1}^{R_0} OUT_{tr} \right) - T \right]^2 + \frac{C_2}{2} \cdot \sum_{t=1}^{T} \sum_{r=1}^{R_0} \left(\frac{OUT_{tr} \cdot \psi_0}{\theta_{tr}} \cdot \sum_{i=1}^{n} (x_{it} \cdot \rho_i) \right) + \frac{D_2}{2} \cdot \sum_{r=1}^{R_0} \left(\frac{1}{h_r} \cdot \sum_{t=1}^{T} OUT_{tr} \right) + \frac{E_2}{2} \cdot \sum_{r=1}^{R_0} \left(\frac{\psi_0}{\eta_r^{\text{ВЗУ}}} \cdot \sum_{t=1}^{T} \sum_{i=1}^{n} (\rho_i \cdot \pi_i \cdot x_{it} \cdot OUT_{tr}) \right) + \frac{F_2}{2} \cdot \sum_{p=1}^{P_0} \left(\frac{1}{T_p} \cdot \sum_{r=1}^{R_0} \sum_{t=1}^{T} OUT_{tr} \cdot SN_{pt} \cdot (t_r^n + t_r) \right) \qquad \textbf{(3)}$$

Constructed by us neural network algorithm as the initial data accepts not only characteristics of initial DDB structure, characteristics of computer network hosts, characteristics of users queries, characteristics of users and constraints of a task, but also two vectors of weight coefficients $\vec{c_1} = (A_1, B_1, C_1, D_1, E_1)$ and $\vec{c_2} = (A_2, B_2, C_2, D_2, E_2, F_2)$ for construction of energy functions of neural networks on the first and second stages accordingly. The regular way of vector's elements definition also does not exist. It is known only, that the constraint $A_1, B_1, C_1, D_1, E_1, A_2, B_2, C_2, D_2, E_2, F_2 > 0$ should be carried out. The neural network algorithm gives ***optimum*** task decision for used goal function and constraints ***at concrete vectors*** $\vec{c_1}$ and $\vec{c_2}$.

Thus, we have the following situation: neural algorithm is capable to produce a set of pairs result-matrixes (matrix of groups distribution on logic records and matrix of distribution of logic records types on hosts in computer network), each such pair corresponds to pair of elements from set of "realizations" of vectors $\vec{c_1}$ and $\vec{c_2}$ (i.e. set of concrete values of elements of these vectors). It is necessary to find such "realizations" of vectors $\vec{c_1}$ and $\vec{c_2}$ (to pick up values of their elements), at which the neural algorithm gives optimum task decision according to goal function and constraints of the task. Such "realizations" of vectors $\vec{c_1}$ and $\vec{c_2}$ we shall name $\vec{c}_{1\,optimum}$ and $\vec{c}_{2\,optimum}$. For a finding $\vec{c}_{1\,optimum}$ and $\vec{c}_{2\,optimum}$ it is offered to use genetic algorithms.

Necessity of genetic algorithms usage. Let's consider the general scheme of genetic optimization algorithms (GOA). Let's given some complex goal function dependent from several variables. It is required to solve an optimization task, i.e. to find such variable's values, at which the value of function is maximum or is minimal [1,4].

The genetic algorithms represent faster approach, than uniform algorithms. They require substantial filling for the decision of each concrete task.

The general scheme of GOA can be submitted by a sequence of steps given in [1].

The search of the optimum decision in genetic algorithms is carried out by a direct manipulation with set from the several allowable decisions forming a population, each of which is coded in a binary code. Since our neural algorithm makes the various decisions of a task of DDB logic structure synthesis at various pairs "realizations" of vectors $\vec{c_1}$ and $\vec{c_2}$, the genetic algorithm can be used as the intellectual tool of movement to the optimum task's decision.

Genetic algorithm as the framework for neural network algorithm. Let's develop model of application of genetic algorithms to the given task [5]. The individual in our case is a point $\vec{c}_{i,\, i=1,2}$ in k-dimensional space (if $i = 1$, then $k = 5$, if $i = 2$, then $k = 6$, i.e. the dimension depends on quantity of constraints for the first and second stages of the task's decision). The quantity of genes is equal to $k = 6$ for each individual. Chromosome − consecutive association of all genes.

For construction of a genotype the following symbolical model is developed: each gene is represented by a string of 32 bits, such representation turns out with the help of two operations: 1) translation of value of a gene in a decimal notation in binary (at addition of the senior categories of zero, if it is necessary), 2) translation of a gene in a binary notation in a Gray code. Thus, the genotype of each individual is represented as a $32 \cdot k$-bit string, where first, second and third etc. 32 elements are Gray codes decimal values of first, second, third etc. elements of a vector accordingly.

At realization of genetic algorithm the neural algorithm is used as a "black box", carrying out display of individual $A_i - B_i - C_i - \ldots$ in the matrix $\underset{(n \times T)}{X}$ or $\underset{(T \times R_0)}{Y}$, appropriate to it (depending on number of a stage of the task's decision), on which $A_i - B_i - C_i - \ldots$ individual's suitability is defined: $A_i - B_i - C_i - \ldots \to \{ \underset{(n \times T)}{X}, \underset{(T \times R_0)}{Y} \}$.

Suitability function of genetic algorithm for the 1st stage task's decision. For definition individual's suitability the following suitability function is chosen

$$F_{suit.} = 1/(1 + \sum_{x} \sum_{i} \sum_{j \neq i} OUT_{xi} \cdot OUT_{xj} + \sum_{i} \sum_{x} \sum_{y \neq x} OUT_{xi} \cdot \left(OUT_{yi} - a^{\text{г}}_{xy}\right)^2 + \left(\left(\sum_{x} \sum_{i} OUT_{xi}\right) - n\right)^2 + \sum_{i} \left[\frac{1}{F_i} \cdot \sum_{x} OUT_{xi}\right] + \sum_{i} \sum_{x} \sum_{y} OUT_{xi} \cdot OUT_{yi} \cdot \left(incomp\_gr_{xy} + incomp\_gr_{yx}\right)) \quad \textbf{(4)}$$

Here square brackets mean a capture of the whole part from division. In a denominator terms number 2, 4 – 6 respond for performance of task's constraints and are considered first of all. If the sum of these four terms is equal to zero, it means, that all constraints are satisfied. The third term shows, as far as the well given decision takes into account semantic contiguity of data groups. Than its value less, it's taken better into account semantic contiguity. As against constraints of a task the complete account semantic contiguity is not necessity, it is desirable. Generally, conditions of semantic contiguity and constraints of a task can be incompatible. The priority is always given back to satisfy of task's constraints. Thus, at the first stage the genetic algorithm allows to select from set of the decisions satisfying to constraints, such decisions which take into account a semantic contiguity of data groups in the best way.

Suitability function of genetic algorithm for a 2nd stage task's decision. For definition individual's suitability the following suitability function is chosen

$$F_{suit.} = 1/(1 + \sum_{t=1}^{T} \sum_{r_1=1}^{R_0} \sum_{r_2=1}^{R_0} OUT_{tr_1} \cdot OUT_{tr_2} + \left(\left(\sum_{t=1}^{T} \sum_{r=1}^{R_0} OUT_{tr}\right) - T\right)^2 + \sum_{t=1}^{T} \sum_{r=1}^{R_0} \left[\frac{OUT_{tr} \cdot \psi_0}{\theta_{tr}} \cdot \sum_{i=1}^{n} \left(x_{it} \cdot \rho_i\right)\right] + \sum_{r=1}^{R_0} \left[\frac{1}{h_r} \cdot \sum_{t=1}^{T} OUT_{tr}\right] + \sum_{r=1}^{R_0} \left[\frac{\psi_0}{\eta_r^{\text{ВЗУ}}} \cdot \sum_{t=1}^{T} \sum_{i=1}^{n} \left(\rho_i \cdot \pi_i \cdot x_{it} \cdot OUT_{tr}\right)\right] + \sum_{p=1}^{P_0} [\frac{1}{T_p} \cdot \sum_{r=1}^{R_0} \sum_{t=1}^{T} OUT_{tr} \cdot SN_{pt} \cdot \left(t_r^n + t_r\right)] + \sum_{k=1}^{K_0} \sum_{p=1}^{P_0} \xi_{kp}^{3} \cdot \varphi_{kp}^{3} \cdot \{ \sum_{r1=1}^{R_0} v_{kr1} \cdot (\sum_{r2=1}^{R_0} z_{pr2} \cdot (t^{\text{раз}} + t_{r1r2}^{\text{сер}} + t_{r1r2}^{\text{пер}} \cdot (1 + \sum_{t=1}^{t_0} z_{pr2}^{t})) + t^{\text{сб}}) + \sum_{r2=1}^{R_0} \sum_{t=1}^{t_0} z_{pr2}^{t} \cdot (t_{r2}^{n} + t_{r2}) \}) \quad \textbf{(5)}$$

Here square brackets mean a capture of the whole part from division. In a denominator first 6 terms after unit respond for satisfy to constraints of a task and are considered first of all. If the sum of these six terms is equal to zero, it means, that all constraints are satisfied. In this case the given individual-decision is located in a new formed population, the value of goal function of our task (1), which is last term in the denominator, is calculated. If the sum of the first, following after unit, six terms is not equal to zero, the given individual is excluded from consideration and, hence, does not get in a new population. In such case among all decisions satisfying to constraints (a part of suitability function, checking satisfying of constraints, is equal to zero), such gets out, which delivers a minimum of task's goal function. From (5) it is obvious, that than smaller value the goal function of a task accepts, the closer to unit value of suitability function of genetic algorithm. And on logic of a task $F_{\text{npucn.}} \in (0;1]$, and best for us value of individual's suitability is $\max\{F_{\text{npucn.}}\} = 1$. The suitability value, calculated by this way, "defines the further destiny" of $A_i - B_i - C_i - \ldots$ individual.

The decision of a task with the help of genetic algorithms for each of stages is the point $\vec{c}_{i\,optimum}$, which application as entrance data for neural algorithm will ensure a finding of optimum result-matrix of splitting of data group's set on logic records – for the first stage, or result-matrix of accommodation of logic records types on computer network's hosts – for the second stage.

In the given work the idea of genetic algorithm's application as a framework for central neural algorithm, both for first, and for the second stages of the task's decision is offered. The function of this framework is the selection among the decisions received with the help neural algorithm (generally locally-optimum) and appropriate to various values of coefficients, global-optimum decisions at the given constraints.

In created NN-GA algorithm it is possible to allocate the following roles. The NN-part responds for satisfying to constraints of a task and maximal account of a semantic contiguity of data groups. As generally conditions of a semantic contiguity and the constraints of a task can be incompatible, the priority is given to constraints satisfying. The GA-part responds for minimization of task's goal function, i.e. provides an optimality of the decision.

4. PROGRAM REALIZATION OF THE ALGORITHM

During realization of NN-GA algorithm an OO-library of C++ classes was constructed, allowing to simulate work of Hopfield neural networks and to use these networks for the decision of the task. An OO-model of genetic algorithms used as a framework for neural network's classes also was written.

The result of the task's decision: 1) vector $\vec{c}_1$ and matrix of groups splitting on logic records (stage 1); 2) vector $\vec{c}_2$ and matrix of accommodation of logic records types on hosts in computer network (stage 2).

Features of genetic algorithm's implementation. In the current implementation of genetic algorithm the following mechanisms are used. 1) Gray code for representation of symbolical model of a genotype. 2) Formation of an initial population on fenotype. 3) A way of a pair choice – punmixing. Punmixing – casual crossing. The only one constraint at a choice of parental pair – this parental pair should have not repeated. 4) Genetic operations – simple crossover and dot mutation.

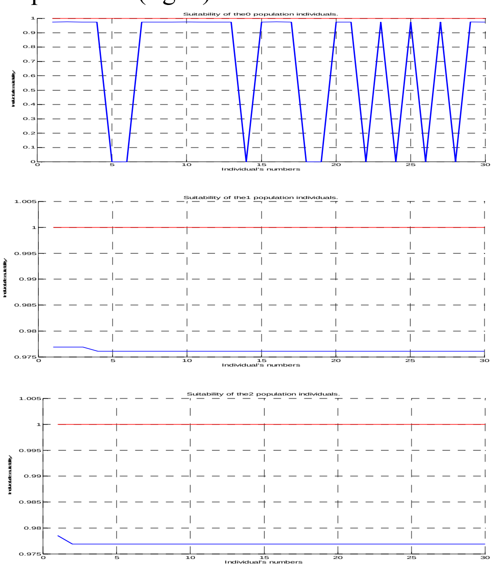
5) A general way of parental group formation, i.e. in reproduction group includes both children, and parents. 6) Strict natural selection.

The algorithm stops, if during several populations the champion on suitability level does not vary, the quantity of such populations is set to algorithm as parameter. The second criterion of algorithm's stopping: if suitability of although one individual of the current population is equal to1. In this case answer is the vector $\vec{c_i}$ appropriate to individual which suitability is equal to 1.

5. RESULTS OF EXPERIMENTS AND ANALYSIS OF THE RECEIVED RESULTS

During testing the optimization task for various configurations of constraints was solved, the optimality of the decision was estimated on value of goal function of a task on the received decision.

To show abilities of genetic algorithms in selection of the optimum decisions, we shall show the diagrams of suitability values change on initial and first two populations during the decision of the second stage of synthesis optimum DDB logic structure with quantity of groups equal to 20 (fig. 1).



(a)

(b)

(c)

Figure 1. Suitabilities of the initial (a), first (b) and second (c) population individuals.

The quality of the decisions received with the help of neural algorithm, was compared with quality of the decisions received with the help of a "branches and borders" method, constructed for a soluble task in [2], for similar configurations of constraints; for each of the decisions percent of a deviation was calculated by the following formula

$$P_{deviation} = \left(\frac{\textit{the value of goal function on decision with help of NN} - GA - \text{algorithm}}{\textit{the value of goal function on decision with help of branches and borders method}} - 1 \right) \cdot 100\% \qquad \textbf{(6)}$$

The average percent of a deviations received as a result of testing on tasks of various dimensions has made $P_{deviation_{average}} = -6.44\%$.

In [6,7] the approaches with usage of Hopfield networks for the decision of NP-complete optimization tasks, in particular to the decision of a traveling salesman problem, are described. The main problem of the approach described in these papers − uncertainty in a choice of coefficients of energy function of a network. Besides that, it is mentioned, that regular methods of definition of these coefficient's values does not exist. In work [8] the useful recommendations for allocation of areas of acceptable coefficient's values are given. In such situation use of genetic algorithms is preferable alternative. Such approach, though it is heuristic, allows to move in a correct direction during search of the most suitable values of coefficients.

The algorithm, offered in the given paper, remains efficient and at removal of the genetic framework from the neural networks. However at the "included" genetic framework the quality of the decision is considerably improved, as neural network though is robust system, but at the large changes of the entrance characteristics nevertheless gives results far from optimum. It is possible to improve results by correctly picked up coefficients of energy function. The rules of such selection are offers by genetic algorithms. These rules do not carry casual character, and work in such way, that it is possible better to approach value of energy function of a task to optimum.

6. REFERENCES

[1] Batishev D.I "Genetic algorithms of the extreme tasks decision". The Nizhniy Novgorod State University. Nizhniy Novgorod, 1995 (electronic variant).

[2] V.V. Kulba, S.S. Kovalevskiy, S.A. Kosyachenko, V.O. Sirotyuck " Theoretical bases of designing optimum structures of the distributed databases ". A series "Information of Russia on a threshold of XXI-th century". - M.: SINTEG, 1999, 660 p. (in Russian).

[3] Edward Babkin, Margarita Karpunina "Towards application of neural networks for optimal synthesis of distributed database systems"

[4] Kruglov V.V., Dli M.I., Golunov R.Y. "Fuzzy logic and artificial neural networks": publishing house of the physical and mathematical literature, 2001. - 224 p.

[5] Karpunina M.E. "Use of genetic algorithms for increase of an overall performance artificial neural networks". News AES A.M.Prokchorov. Busines - computer science. 2005 - 11 p.

[6] F. Wossermen "Neurocomputing engineering: the theory and practice" Translation on Russian, Y. I. Zuev, V. I. Tochenov, 1992. (in Russian).

[7] Horfield J.J., Tank D.W. "Neural computation of decisions in optimization problems". Biological Cybernetics52: 141-52, 1985.

[8] "Using Hopfield Networks to Solve Traveling Salesman Problems Based on Stable State Analysis Technique", Gang Feng and Christos Douligeris, Department of Electrical and Computer Engineering, University of Miami Coral Gables, FL 33124-0640, USA. Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks, 2000.