An Improved Complex System Optimization Method Hybridized Structure-based Neural Networks with Orthogonal Genetic Algorithm

Si-Jun tao

School of Mathematics and Computer Sciences, Xinyu University, Xinyu 338003, P.R. China

Abstract—The research on optimization methods to complex systems is an important issue in both theoretical and practical significance. For this reason, an improved complex system optimization method is proposed which hybridized the structure-based neural networks with the orthogonal genetic algorithm. Experimental results suggest that this approach outperforms other existing approaches.

Keywords-Optimal design, genetic algorithm, neural network, knowledge.

I Introduction

Various systems become more and more complex and the corresponding influence factors also enjoy a great increase. Therefore, it becomes a very difficult job to make an overall system optimal design. For example, the design of weapon system is the feedback and iteration process of technical information between the whole system and equipment. Usually many conflicts exist among different performances of the weapon system and in the performance between the whole system and equipment. They should be coordinated constantly to improve the performance of the whole system (Camara et al., 2009; Yuan et al., 2008; Mukhopadhyay et al., 2009; Zhou, 2009). Therefore, most of the countries in the world have shifted their focus to the integrity of combat ability, survival ability, rapidity, maneuverability and compatibility when designing weapon systems (Wang et al., 2009; Marano et al., 2009; Zhou et al., 2009) during recent years.

Since the system optimal design is a difficult job, this paper first employs structure-based neural networks to fit the input-output relationship of the weapon system, then adopts genetic algorithms to train parameters of the network and optimize the neural network model, and finally achieves very satisfactory experimental results. The whole design approach can be concluded as below: extracting some knowledge from the optimization process of genetic algorithms and then employing it to guide the subsequent optimization process.

II Discussed problems

This problem can be concluded as: employing simulation optimization approaches to study system optimal design problems and attempting to make the system to be studied output satisfactory results through fewer times of system simulation. The corresponding target function is described as:

$$\begin{cases} \max\left\{P_1, P_2, \cdots, P_s\right\} \\ \min\left\{T, Q_1, Q_2, \cdots, Q_r\right\} \end{cases} \tag{1}$$

Here, $P_1, P_2, \cdots, P_s$ denote some indicators that need to be maximized in the system to be studied, $Q_1, Q_2, \cdots, Q_r$ denote ones that need to be minimized, and T denotes the total time cost by the system simulation during the optimization process. Constraint conditions of the problem are described as:

$$\begin{cases} P_i = Sim(S_1 S_2, \cdots, S_n), i = 1, 2, \cdots, s \\ Q_i = Sim(S_1 S_2, \cdots, S_n), i = 1, 2, \cdots, r \\ T = \sum_{i=1}^{n+1} t_i c_i \\ S_i = Sim(x_{i1} x_{i2}, \cdots, x_{i(k_i)}), i = 1, 2, \cdots, n \\ x_{ij} \in \left[x_{ij}^{low}, x_{ij}^{up}\right], i = 1, 2, \cdots, n, j = 1, 2 \cdots, k_i \end{cases} \tag{2}$$

In the first constraint condition and the second one, take simulation outputs of all the subsystems as the input of the whole system and then make simulations to obtain the output value of all the indicators of the system to be studied. $S_i (1 \le i \le n)$ denotes the simulation output of the i^{th} subsystem, either a single value or a vector. The third constraint condition denotes the total time cost by simulations during the optimization process. $t_i \left(1 \le i \le n\right)$ denotes the average time spent by the i^{th} subsystem on a single simulation. t_{n+1} denotes the average time spent by the system on the overall simulation. $c_i \left(1 \le i \le n\right)$ denotes the number of simulations implemented by the i^{th} subsystem during the optimization process, and c_{n+1} denotes the total number of overall simulations. The fourth constraint condition denotes that simulation outputs of subsystems can be obtained solely through subsystem simulations. $k_i \left(1 \le i \le n\right)$ denotes the number of input variables of the i^{th} subsystem. The fifth constraint condition defines the feasible region of input variables of the system to be studied.

III The Proposed Method

The proposed approach in this paper employs the structure-based neural networks with highly-nonlinear mapping

© 2015. The authors - Published by Atlantis Press

capability to fit the input-output relationship of the weapon system, trains parameters of the structure-based network through genetic algorithms with global searching ability. Genetic algorithms can also optimize the neural network model at the same time. This approach extracts some knowledge from the optimization process of genetic algorithms and then uses it to guide the subsequent optimization process.

A. Structure-based Neural Networks

The structure-based neural network is established based on the causality theory, and connections between its nodes are all based on the causal relationships in real systems, so it is fairly powerful in interpreting models. The structure-based neural network modelling has settled many defects confronting traditional neural network modelling, such as the unstructured models, uncertainty of the number of neurons, slow convergence and local minimization, etc. It has acted as a new powerful processing tool for non-linear systems, especially for the system modelling and structure parameter optimization of those large-scale non-linear systems (Whittaker et al., 2009; Chung et al., 2009; Martin, 2009; Kirnbrough et al., 2008; Vilcot, 2008).

Fig. 1 shows the input-output relationship model for system optimal design problems based on the structure-based neural networks. $x_{11}, x_{12}, \cdots, x_{1n}$ denote the input of the first subsystem, $x_{21}, x_{22}, \cdots, x_{2n}$ denote the input of the second one and $x_{m1}, x_{m2}, \cdots, x_{mn}$ denote that of the m^{th} one. $P_1, P_2, \cdots, P_s$ denote some indicators of the system to be studied that need to be maximized and $Q_1, Q_2, \cdots, Q_r$ denote those that need to be minimized.

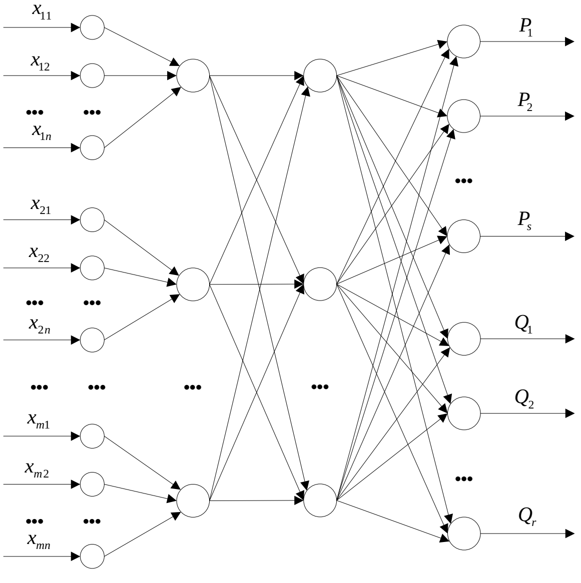

Figure 1. Structure-based neural network model

B. Orthogonal Genetic Algorithm with Quantification

Initialization of the population: (1) Divide the feasible region $[L, U]$ of the problem to be optimized into B subspaces according to the following equation

$$u_k - l_k = \max_{1 \le i \le n} \{u_i - l_i\}$$

$$\begin{cases} L_i = L + (i-1)\left[(u_k - l_k)/B\right]1_k \\ U_i = U + (B-i)\left[(u_k - l_k)/B\right]1_k \end{cases}, \quad i = 1, 2, \cdots, B$$

Here, $L = \left[l_1, l_2, \cdots, l_n\right]^T$ and $U = \left[u_1, u_2, \cdots, u_n\right]^T$ respectively denote the lower boundaries and upper boundaries of n independent variables of the problem. B denotes the design parameter. 1_k is the n-dimensional vector of which the k^{th} bit is 1 and other bits are 0. L_i and U_i denote n-dimensional vectors respectively similar to L and U. In this way, the feasible region of the problem can be divided into B subspaces, namely $\left[L_1, U_1\right], \left[L_2, U_2\right], \cdots, \left[L_B, U_B\right]$.

(2) Discretize each independent variable in each subspace according to the following equation. Suppose the domain of the independent variable x_i is $\left[l_i, u_i\right]$, then x_i can be quantized into Q_1 (the design parameter) levels $a_{i1}, a_{i2}, \cdots, a_{iQ_1}$, and the detailed computation method for a_{ij} is

$$a_{ij} = \begin{cases} l_i & j = 1 \\ l_i + (j-i)\left[(u_i - l_i)/(Q_1 - 1)\right] & 2 \le j < Q_1 \\ u_i & j = Q_1 \end{cases}$$

(3) Select M_1 chromosomes from each subspace.

First construct the orthogonal table $L_{M_1}(Q_1^N) = \left[a_{ij}\right]_{M_1 \times N}$ in which N denotes the dimensionality of the problem, $M_1 = Q_1^{J_1}$, and J_1 denote the positive integers that satisfy the condition $\frac{Q_1^{J_1} - 1}{Q_1 - 1} \ge N$. Then select M_1 combinations from the Q_1^N ones to form M_1 chromosomes.

(4) Select the best G (the size of the initial population) chromosomes from the $M_1 B$ potential ones to form the initial population according to their fitness value.

Crossover operation: Select two parent chromosomes for crossover operation according to the crossover probability. Suppose that the two selected parents are $\begin{cases} p_1 = (p_{11}, p_{12}, \cdots, p_{1N}) \\ p_2 = (p_{21}, p_{22}, \cdots, p_{2N}) \end{cases}$ and the defined solution space $\left[l_{parent}, u_{parent}\right]$ is

$$\begin{cases} l_{parent} = \left[\min(p_{11}, p_{21}), \min(p_{12}, p_{22}), \cdots, \min(p_{1N}, p_{2N}) \right] \\ u_{parent} = \left[\max(p_{11}, p_{21}), \max(p_{12}, p_{22}), \cdots, \max(p_{1N}, p_{2N}) \right] \end{cases}$$

(2) Discetize the solution space $\left[l_{parent}, u_{parent} \right]$ of two parent individuals for crossover operation into Q_2 parts.

(3) Select some independent variables that will suffer crossover operations. The number of potential child individuals generated by each pair of parent individuals should be controlled to avoid a huge evaluation on the populations during the crossover operation. In this paper, the crossover only operates on F genes of the parent chromosome. Discretize these F independent variables in each subspace.

(4) Select potential child points from the solution space of parents according to the orthogonal table. First generate the orthogonal table $L_{M_2}(Q_2^F) = \left[b_{ij} \right]_{M_2 \times F}$ in which Q_2 is an odd number, $M_2 = Q_2^{J_2}$, and J_2 is the smallest positive integer that satisfies the condition $\frac{Q_2^{J_2} - 1}{Q_2 - 1} \geq F$. Then select M_2 combinations from these Q_2^F ones to form M_2 potential child individuals.

(5) Select two with the best fitness value from the M_2 potential child individuals and two parents as the result of this crossover operation.

(6) If the number of the implemented crossover operations has reached the preset value, stop crossover immediately, or turn to (1).

IV Experimental Results

XX system exerts its damage on the target through target search, target identification and target attack. The system owns multiple factors and its exertion process is complicated, so we can make full use of its high cost-effectiveness only by optimizing structure parameters and performance parameters and coordinating their relationships. Performance factors selected in this paper for the system mainly include: the fall velocity V_y, rotating speed ω, scanning angle θ_s, operating distance H, position error of sensors and dispersion error of warhead E, ambient wind velocity F. The hit probability is selected as the target function.

According to the practical simulation model of XX system, the obtained simulation hit probability corresponding to the above-mentioned optimization parameters is 0.812. Compare the optimal hit probability with the simulation hit probability and we can find that the simulation model based on the performance factors of neural network is feasible and accurate. The author has made a comparative analysis between the result obtained in this paper and that in literature (Liu, 1996; Huang et al., 2001; Huang et al., 1999; Huang et al., 2004), shown in Table 1. It shows that the two are in close proximity to each other, which further indicates that it is totally feasible to employ genetic neural networks for optimal design.

TABLE 1. Comparative analysis on experimental results

	Method in (Liu, 1996)	Method in (Huang et al., 2001)	Method in (Huang et al., 1999)	Method in (Huang et al., 2004)	Method in this paper
$V_y\left(\mathrm{ms}^{-1}\right)$	8	8	8	8	8
$\omega\left(\mathrm{rs}^{-1}\right)$	3.000	3.500	4.000	4.965	4.876
$\theta_s\left(^{\circ}\right)$	26.00	28.00	30.00	30.54	31.02
$H(\mathrm{m})$	/	110	/	140	136
$E(\mathrm{m})$	/	0.1	/	0.1	0.1
$F\left(\mathrm{ms}^{-1}\right)$	0	0	0	0	0
P_h (hit probability)	0.529	0.614	0.501	0.749	0.812

Notes: "/" means that no researches have been made.

V Conclusions

Main contributions of this paper include: employing the structure-based neural networks with highly-nonlinear mapping capability to fit the input-output relationship of the weapon system, using genetic algorithms with global searching ability to train parameters of the structure-based network and optimize neural network models. This approach extracts some knowledge from the optimization process of genetic algorithms and then uses it to guide the subsequent optimization process. The experimental result indicates that the approach outperforms several other existing methods.

VI References

[1] Camara, M., J. Ortega and F. De-Toro, 2009. A single front genetic algorithm for parallel multi-objective optimization in dynamic environments. Neurocomputing, 72(16-18): 3570-3579.

[2] Yuan, Q., Z.Q. He and H.N. Leng, 2008. A hybrid genetic algorithm for a class of global optimization problems with box constraints. Applied Mathematics and Computation, 197(2): 924-929.

[3] Mukhopadhyay, A., U. Maulik and S. Bandyopadhyay, 2009. Multiobjective genetic algorithm-based fuzzy clustering of categorical attributes. IEEE Transaction on Evolutionary Computation, 13(5): 991-1005.

[4] Zhou, Y.R., 2009. Runtime analysis of an ant colony optimization algorithm for TSP Instances. IEEE Transaction on Evolutionary Computation, 13(5): 1083-1092.

[5] Kim, K. and I.J. Jeong, 2009. Flow shop scheduling with no-wait flexible lot streaming using an adaptive genetic algorithm. The International Journal of Advanced Manufacturing Technology, 44(11-12): 1181-1190.

[6] Wang, Y.M., H.L. Yin and J. Wang, 2009. Genetic algorithm with new encoding scheme for job shop scheduling. The International Journal of Advanced Manufacturing Technology, 44(9-10): 977-984.

[7] Marano, G.C., G. Quaranta and R. Greco, 2009. Multi-objective optimization by genetic algorithm of structural systems subject to random vibrations. Structural and Multidisciplinary Optimization, 39(4): 385-399.

[8] Zhou, H., W. Cheung and L.C. Leung, 2009. Minimizing weighted tardiness of job-shop scheduling using a hybrid genetic algorithm. European Journal of Operational Research, 194(3): 637-649.

[9] Whittaker, G., R. Confesor and S.M. Griffith et al., 2009. A hybrid genetic algorithm for multiobjective problems with activity analysis-based local search. European Journal of Operational Research, 193(1): 195-203.

[10] Chung, J.W., S.M. Oh and I.C. Choi, 2009. A hybrid genetic algorithm for train sequencing in the korean railway. OMEGA - The International Journal of Management Science, 37(3): 555-565.

[11] Martin, C.H., 2009. A hybrid genetic algorithm / mathematical programming approach to the multi-family flow shop scheduling problem with lot streaming. OMEGA - The International Journal of Management Science, 37(1): 126-137.

[12] Kirnbrough, S.O., G.J. Koehler and M. Lu et al., 2008. On a feasible-infeasible two-population (FI-2Pop) genetic algorithm for constrained optimization: distance tracing and no free lunch. European Journal of Operational Research, 190(2): 310-327.

[18]

[13] Vilcot, G. and J.C. Billaut, 2008. A tabu search and a genetic algorithm for solving a bicriteria general job shop scheduling problem. European Journal of Operational Research, 190(2): 398-411.

[14] Liu, R.Z., 1996. Study of the structural dynamic responses and efficacy analysis about terminal-sensitive projectile. Nanjing University of Science and Technology.

[15] Huang, K. and R.Z. Liu, 2001. Sensitivity analysis of system efficacy about terminal-sensitive projectile. Acta Armamentarii, 22(3): 412-415.

[16] Huang, K. and R.Z. Liu, 1999. Injury probability calculation and influence factor analysis about terminal-sensitive projectile. Journal of Ballistics, 11(2): 37-42.

[17] Huang, K., S.F. Chen and R.Z. Liu, 2004. Optimal design of system efficiency parameters about terminal-sensitive projectiles based on neural network and genetic algorithm. Acta Armamentarii, 25(3): 257-260.