8th International IEEE EMBS Conference on Neural Engineering
Shanghai, China, May 25 - 28 , 2017

Modified genetic crossover and mutation operators for sparse regressor selection in NARMAX brain connectivity modeling

Pedro C. Nariyoshi* J.R. Deller, Jr.,* *Fellow, IEEE*, and Jinyao Yan*

***Abstract*— In support of a method for nonlinear modeling of cortical connectivity, an innovation in evolutionary processing is reported. By a strategic modification to the crossover and mutation operators within the NSGA-II genetic algorithm, the number of generations required to achieve optimal nonlinear regressor models with 99.9% confidence is reduced by 52%.**

I. Introduction

This work is part of an ongoing effort to develop nonlinear system models for large-scale interconnectivity of cortical networks in the brain. The current work is centered on a biologically-motivated method for both the selection of the effective regressors and the estimation of the parameters of a general nonlinear autoregressive moving average with exogenous input (NARMAX) model (e.g. [1]). The approach integrates set-based parameter estimation and genetic algorithms for optimization over fitness measures derived from the set solutions [2]. A brief sketch of the overall approach appears in Section II. This paper is focused on an innovation in the evolutionary process by which the model regressor set is selected. The results have broad implications outside of the neural-modeling context in which they were developed.

As in any nonlinear identification solution, the evolutionary–set-theoretic framework described above is computationally-intensive as the number of regressors increase factorially with the order of the nonlinear expansion. In the broader view, the complexity of any credible brain modeling algorithm for large-scale cortical connectivity poses a formidable challenge to even state-of-the-art research computing infrastructures. The increasing amount of electroencephalogram (EEG) data that is publicly available to brain researchers renders insufficient processing power even more acute by limiting the utility of this information-rich resource [3].

In a general sense, this paper addresses the need to find more efficient data-processing algorithms for brain modeling. In terms of the brain-connectivity model, the objective of this work is to reduce the computational complexity of model-identification without incurring the cost of omitting observation data. A more efficient solution is based in the expected sparsity of the connectivity models in terms of the relatively low number of regressors that would be necessary to effectively characterize specific channels in the brain. This assumption has significant implications for the evolutionary search over the space of regressor combinations.

In particular, modified crossover and mutation operators are incorporated in a NSGA-II [4] framework to expedite feature (regressor) selection. By adjusting the mutation and crossover operators to account for sparsity, the number of generations needed to arrive at the solution is greatly reduced. Simulation studies show a 52% reduction in generations needed for optimization (with 99.9% confidence).

This work was supported in part by the U.S. National Science Foundation under Cooperative Agreement DBI-0939454. Any opinions, conclusions or recommendations expressed are those of the authors and do not necessarily reflect the views of the NSF.

*Department of Electrical and Computer Engineering, Statistical Signal Processing Lab & BEACON Center, Michigan State University, East Lansing, MI 48824-1226 USA nariyosh, deller, yanjinya egr.msu.edu

II. Methods

A. Model form and identification strategy

To provide context, we briefly sketch the general model to be created and identified. The critical concern for the present paper is the evolutionary process by which the regressor signals are determined. Further technical details of the operation of the model are found in previous papers [2], [5], [6]. The internal processing of the system is based on a subset of a ***candidate set*** of nonlinear regressor functions, $\Xi_\varphi = \{\varphi_q\}$, of size $|\Xi_\varphi|$. Each regressor is a mapping $\varphi_q : \mathbb{R}^{r_q+s_q} \to \mathbb{R}$, operating on a set of r_q past and present system inputs, and s_q past outputs. The linear, time-invariant in parameters (LTIiP) ***observation model***, $\mathbb{O}_{\theta_*,\varphi_*}$, for $t \in \mathbb{Z}$, is given by

$$\mathbb{O}_{\theta_*,\varphi_*} : \quad y[t] = \sum_{q=1}^{Q} \theta_{q*}\varphi_{q*}\left(x_{-\infty}^{t}, y_{-\infty}^{t-1}\right) + e_*[t] \qquad (1)$$

with $\theta_* \in$, and $e_* \in \mathbb{R}^{\mathbb{Z}}$ an error sequence representing uncertainties in the model. The "$*$" subscript indicates a "true," but unknown, quantity associated with the observation model.[1] The arguments, $x_{-\infty}^{t}$ and $y_{-\infty}^{t-1}$, of the regressor signals φ_q (or vector φ) indicate that a finite number of elements are selected from the subsequences $\{\ldots, x[t-1], x[t]\}$ and $\{\ldots, y[t-2], y[t-1]\}$ by each φ_q for processing at time t. For conservation of space, we define the vectors of $r_q + s_q$ signal samples used at time t by $u_{q*}[t]$, and the matrix $U_*[t] = \begin{bmatrix} u_{1*}[t] & u_{2*}[t] & \cdots & u_{Q*}[t] \end{bmatrix}$. Given observations of x and y sufficient to compute outputs on time interval $t = 1, 2, \ldots,$, we pose an ***estimation model*** as a function of the parameters and regressor signals,

$$\mathbb{M}_{\theta,\varphi} : \zeta^p\left(t, \theta, \varphi\right) = \sum_{q=1}^{Q} \theta_q \varphi_q\left(u_q[t]\right) \stackrel{\text{def}}{=} \theta^T \varphi\left(U[t]\right), \quad (2)$$

[1] It is to be understood that φ_{q*} is the q^{th} element *selected from* Ξ_φ, rather than element q of Ξ_φ.

in which each φ_q is drawn from the set Ξ_φ (see footnote 1), $\theta \in$, and the $u_q[t]$ and $U[t]$ are defined similarly to $u_{q*}[t]$ and $U_*[t]$. The superscript on ζ^p connotes "prediction", as this estimation model corresponds to the classical prediction-error method (e.g., [7]). This is true even though the regressor functions can be highly-nonlinear functions of the observations, because (when assumed fixed in the model) they appear in a model that is linear-time-invariant-in-parameters (LTIiP). Thus, the identification of the parameters using deterministic or stochastic mean squared error techniques is a well-known problem. Our approach, however, involves a distinctly different identification method which produces parameter solution *sets* rather than point estimates (e.g., [8], [9]). It is the properties of these sets that couple the model creation and parameter identification problems.

Models are treated as chromosomes. The LTIiP model is the phenotype of a chromosome, a binary sequence in which each bit indicates the presence or absence of a particular gene. Each gene codes for a particular regressor function in the model. The algorithm starts with a random population of chromosomes. The parameter sets result from the set-membership processing of the data and the genetic makeup of each chromosome. Unlike other estimation methods, the set-membership algorithms provide sets of feasible parameter vectors rather than a single point estimate. Measurable set properties are then used to assign fitness values to each chromosome, and the fitness value is used in the genetic algorithm selection process to evolve the population toward better solutions (e.g. [10]). This framework simultaneously addresses selection of the model structure and the parameter estimation. Moreover, a very significant advantage of the algorithm is the lack of need for assumptions about stationarity or distributional characteristics of the noise.

To reduce the computational complexity of this process, the search space of regressor models must be controlled, and the candidate and final models must use the fewest regressors that are consistent with an objective of prediction-error minimization, Since these objectives are conflicting, a multi-objective optimization approach is desired. For this work, the NSGA-II approach is adopted, since it generates set solutions (ideally the Pareto-front), providing the best solution for a given number of regressors and allowing the model with the best trade-off to be chosen.

B. NSGA-II

NSGA-II is a standard algorithm for solving multiobjective optimization problems. It requires a small number of parameters and is able to obtain solution sets with good spread. The basic NSGA-II algorithm is shown in Fig. 1. An initial random population of size N is generated and evaluated according to the two objectives: prediction accuracy and number of regressors. The population is then sorted, the best half is selected as parents, which go through selection, mutation and crossover to generate a new population of children. The parents and children of this generation become

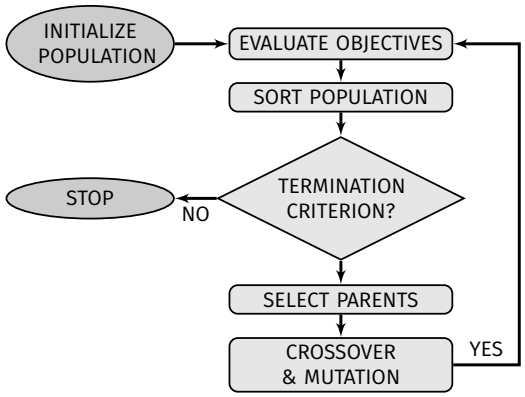

Fig. 1. NSGA-II algorithm summary

the parents of the following generation. The cycle is repeated until the termination criterion is reached.

In the original NSGA-II paper [4], the authors used binary tournament, bit-wise mutation and single-point crossover with probability of $p_c = 0.9$, and mutation probability $\mu = 1/\ell$ (where ℓ is the length of the chromosome). In this work, the same parameters are used, with the exception of the single-point crossover operator which is replaced by a two-point crossover operator.

C. Asymmetric sparse mutation operator

For sparse solutions, the mutation operator can be tuned to guide the population toward sparsity. Although judicious selection alone can effect sparse solutions, a properly tuned mutation operator can increase the convergence rate significantly. Classic mutation operators use a fixed probability to flip each chromosome regardless of its previous value. This is effective for blind exploration, but imposes pressure toward solutions with 50% active genes.

For a given μ probability of mutation, the expected number of active (N_1) and inactive genes (N_0) is given by

$$\begin{bmatrix} N_1^{n+1} \\ N_0^{n+1} \end{bmatrix} = \begin{bmatrix} (1-\mu) & + & \mu \\ \mu & + & (1-\mu) \end{bmatrix} \begin{bmatrix} N_1^{n} \\ N_0^{n} \end{bmatrix} \tag{3}$$

This matrix has eigenvalues 1 and (1-2μ). The eigenvector for 1 is $[\ 1 \quad 1\ ]^T$, which means that, in the absence of selection operators, the number of active and inactive genes tends to equality at a rate depending on μ .

An asymmetrical mutation operator can be used to achieve any desired rate of activation. Two distinct mutation operators are introduced to implement this effect: μ_{10} the probability of deactivating an active gene, and μ_{01} the probability of activating an inactive gene. The matrix system (3) becomes

$$\begin{bmatrix} N_1^{n+1} \\ N_0^{n+1} \end{bmatrix} = \begin{bmatrix} (1-\mu_{10}) & + & \mu_{01} \\ \mu_{10} & + & (1-\mu_{01}) \end{bmatrix} \begin{bmatrix} N_1^{n} \\ N_0^{n} \end{bmatrix} \tag{4}$$

The eigenvalues of this system are 1 and $1-\mu_{10}-\mu_{01}$ with corresponding eigenvectors $[\ \mu_{01} \quad \mu_{10}\ ]^T$ and $[\ 1 \quad -1\ ]^T$. The desired ratio of active to inactive genes is given by

$$r_d = \frac{\mu_{01}}{\mu_{10}+\mu_{01}} \tag{5}$$

For this scheme, the mutation rate is defined as

$$\mu = r_c\ \mu_{10} + (1-r_c)\mu_{01} \tag{6}$$

where r_c is the ratio of active to total genes (i.e. $N_1/(N_1+N_0)$). By combining eqs. (5) and (6), the following expressions for the mutation probabilities are obtained

$$\begin{aligned}\mu_{01} &= \frac{\mu\, r_d}{r_c + r_d - 2r_c r_d} \\ \mu_{10} &= \frac{\mu(1-r_d)}{r_c + r_d - 2r_c r_d}\end{aligned} \tag{7}$$

This extended solution reduces to that for the traditional mutation operator when $r_d = 0.5$. Decoupling the mutation probabilities yields a more flexible mutation operator with which pressure can be applied toward a desired sparsity level.

D. Reduced surrogate crossover

Evolution is improved by a crossover operator that generates novel individuals. A method to achieve novelty is to use reduced surrogate crossover (RSX) [11]. RSX crosses only non-matching alleles between individuals. This is especially important as the genetic diversity decreases with evolution.

A varying-minimum Hamming distance between chromosomes is suggested in [12]. In the present work, a fixed unity Hamming distance yielded small improvements in convergence speed. The fixed distance avoids the shortcomings of the minimum Hamming approach, but results in less efficient sampling of the search space.

III. Results

A randomly generated NARMAX model with five regressors was used to evaluate the modified operators. Three delayed outputs and two delayed inputs to the system were extracted as linear regressors and expanded to a 3rd order Volterra series, obtaining a total of 55 nonlinear regressors.

The estimated Pareto front is shown in Fig. 2. The ordinate shows the RMS error of the prediction error (dB scale) and the abscissa shows the number of regressors in each model. As expected, there is a knee located at five regressors, corresponding to the number in the generative model. There is some improvement in the RMS error for models with more regressors, but only due to overfitting.

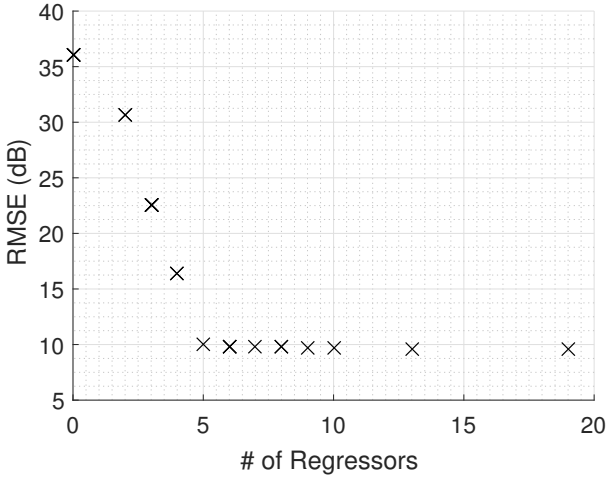

Fig. 2. Estimated pareto front

To assess the improvement relative to the unmodified NSGA-II method, simulations were used to estimate the number of generations required for each genetic algorithm (GA) to converge to the generative model. The number of generations follows a probability distribution with parameters depending on the GA and its internal parametrization.

Clearwater *et al.* [13] have shown that the number of generations required by a GA to find a solution follows a log-normal distribution. Due to the long-tailed nature, the mean and variance of this distribution are both significant. A lower-variance estimator can provide a more meaningful measure of number of generations to convergence, even at the expense of mild estimator bias.

To examine the how well the number of generations required to find the solution fits a log-normal distribution, 16384 runs of our algorithm were evaluated using the same parameters for each run. The number of generations that each run took was recorded and a log-normal distribution fitted to the data. Fig. 3 shows the histogram with "×" markers and the fitted distribution with the solid line. The fitted distribution tracks the histogram remarkably well, specially in the long tail of the distribution. For clarity, the histogram is omitted from further figures.

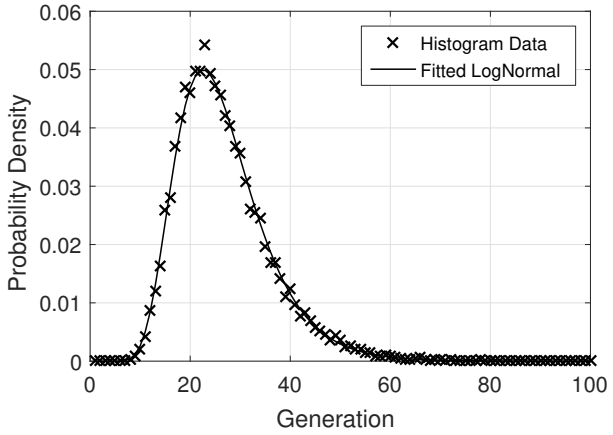

Fig. 3. Histogram vs Fitted distribution

Each modification to the original NSGA-II approach was evaluated by 16384 runs with the same parameters for each run. The original approach was implemented using NSGA-II with the original bit-wise mutation operator and 2-point crossover. The asymmetrical binary mutation with $r = 0.1$ and was applied to all remaining simulations. The modified domination criterion (unique sorting) was added to the third simulation onwards. The fourth simulation incorporated the RSX operator and the fifth added a minimum Hamming distance (HD) of 1 to the mutation operator. The results can be seen in Fig. 4. The parameters for the fitted models are compiled into table I. Since in the log-Normal distribution, μ and σ do not correspond to the mean and standard deviations, these values are also calculated and shown in separate columns.

The assymetrical mutation operator causes a drastic change in the simulation, reducing both the mean and standard deviation significantly. It reduces the number of gener-

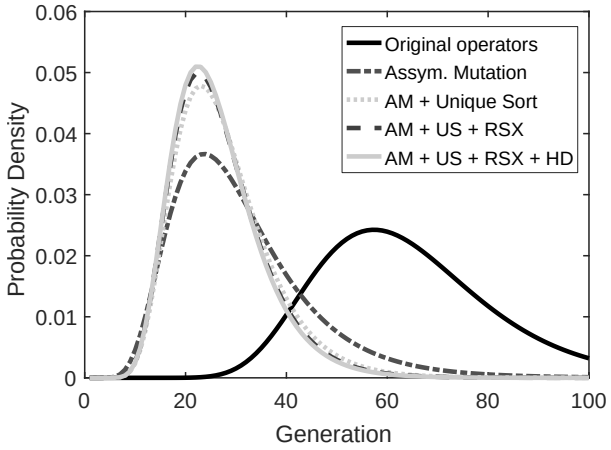

Fig. 4. Generations to arrive to desired model

TABLE I

FITTED PARAMETERS FOR DIFFERENT METHODS

Method	μ	σ	Mean	Std. Dev.
Original operators	4.13	0.28	64.3	18.1
Assymetrical Mutation	3.34	0.421	30.9	13.6
AM + Unique Sort	3.25	0.341	27.5	9.64
AM + US + RSX	3.23	0.335	26.7	9.19
AM + US + RSX + HD	3.22	0.329	26.4	8.95

ations to reach 99% confidence from 118 to 76 generations (reduction of 35%) and 99.9% from 145 to 104 generations (a reduction of 28%).

While the modified domination criterion caused a smaller reduction in μ, the σ is reduced more significantly. As can be seen in the graph, the modes are largely unchanged (ranging from 22 to 23 generations), but the reduction in σ greatly reduces number of generations to to reach high confidence. The old sorting algorithm passes 99% confidence at 76 generations and 99.9% confidence region at 104 generations, while the new sorting algorithm passes 99% confidence in 58 (a further reduction of 23%) and 99.9% in 75 generations (a further reduction of 27%).

The remaining improvements improve convergence, albeit in a smaller scale, with the reduced surrogate without minimum Hamming distance and the reduced surrogate with minimum distance of 1 needing 55 and 54 generations to reach 99% confidence respectively and needing 71 and 70 generations to reach 99.9% confidence.

IV. DISCUSSION

When a modeling problem provides little guidance on the selection of an effective model form, a GA must search a wide space of candidate features but determine a relaiable and consistent solution in a limited number of generations. The 99% and 99.9% confidence metrics resulting in the modified search methods provide a stronger measure of performance than the estimated mean and variance, even though the confidence information is theoretically inherent in the two statistics.

Due to the correlation between the linear regressor tokens used to compute nonlinear models, there is significant statistical dependence between nonlinear expansions of the regressors. This results in ambiguity between different regressors, this leads to many local minima.

Due to the large number of local minima, the genetic algorithm benefits from a more aggressive domination criterion, in which an individual must surpass another individual in at least one objective to not be dominated by it. This improves convergence speed by slightly prioritizing objective performance over diversity.

To complement the reduced pressure towards diversity, the modified domination criterion assigns duplicate individuals to the last front. The disadvantage of this approach is that it forcefully deemphasizes regions that attract large number of individuals, by reducing the population around a region if collisions happen. However, results show that this is relatively rare, as modifying the mutation operator to mutate identical individuals until no identical individuals exist did not improve convergence significantly.

V. CONCLUSION

In this work, simple modified crossover, mutation and domination operators were presented. These modifications yield significant performance improvements over the pure NSGA-II algorithm for this application. Such large improvements suggest that further optimizations to the algorithm are possible and will be investigated in the future. Additionally, at this point, the algorithm does not regard the relationships among regressors (e.g. $y(t-1)$ and $y(t-1)^3$). Accounting for such dependence will likely result in further improvement, especially in more complex models.

REFERENCES

[1] S. Chen and S. A. Billings, "Representations of non-linear systems: the NARMAX model," *International J. Control*, vol. 49, no. 3, pp. 1013–1032, 1989.

[2] J. Yan and J. Deller, Jr., "Narmax model identification using a set-theoretic evolutionary approach," *Signal Processing*, vol. 123, no. 5, pp. 30–41, 2016.

[3] H. V. Jagadish *et al.*, "Big data and its technical challenges," *Commun. ACM*, vol. 57, no. 7, pp. 86–94, Jul. 2014.

[4] K. Deb *et al.*, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.

[5] J. Yan, J. R. Deller Jr., M. Yao, and E. D. Goodman, "Evolutionary model selection for identification of nonlinear parametric systems," in *Proc. 2014 IEEE China Summit and International Conf. Signal and Information Processing*, 2014, pp. 693–697.

[6] J. Yan, J. R. Deller Jr., D. B. Fleet *et al.*, "Evolutionary identification of nonlinear parametric models with a set-theoretic fitness criterion," in *Proc. 2013 IEEE China Summit and International Conf. Signal and Information Processing*, vol. 1, 2013, pp. 44–48.

[7] L. Ljung, *System Identification: Theory for the User*, 2nd ed. Prentice-Hall PTR, 1999.

[8] J. R. Deller Jr., M. Nayeri, and M. S. Liu, "Unifying the landmark developments in optimal bounding ellipsoid identification," *International J. Adaptive Control and Signal Processing*, vol. 8, pp. 43–60, 1994.

[9] J. R. Deller Jr., M. Nayeri, and S. F. Odeh, "Least-square identification with error bounds for real-time signal processing and control," *Proc. IEEE*, vol. 81, pp. 815–849, 1993.

[10] C. Reeves and J. Rowe, *Genetic Algorithms: Principles and Perspectives: A Guide to GA Theory*. Springer, 2003, vol. 20.

[11] L. Booker, "Improving search in genetic algorithms," *Genetic Algorithms and Simulated Annealing*, pp. 61–73, 1987.

[12] M. L. Mauldin, "Maintaining diversity in genetic search." in *AAAI*, 1984, pp. 247–250.

[13] S. H. Clearwater, T. Hogg, and B. A. Huberman, "Cooperative problem solving," in *Computation: The Micro and the Macro View*. World Scientific, 1992, pp. 33–70.