Proceedings of the 29th Annual International
Conference of the IEEE EMBS
Cité Internationale, Lyon, France
August 23-26, 2007.

Modified Genetic Algorithm for Parameter Selection of Compartmental Models

Neil A. Shah, Richard A. Moffitt, and May D. Wang

***Abstract*—A Modified Genetic Algorithm has been developed for the task of optimal parameter selection for compartmental models. As a case study, a predictive model of the emerging health threat of obesity in America was developed which incorporated varying levels of three treatment strategies in an attempt to decrease the amount of overweight Americans over a ten–year period. The Genetic Algorithm was then applied to the task of minimizing the number of overweight persons while minimizing the costs associated with implementing the chosen treatment plans. Throughout repeated trials, the GA was able to converge to consistent, high-scoring treatment strategies after only a few minutes of computation on a desktop PC. This result demonstrates the ability of the modified Genetic Algorithm to effectively perform multivariate, nonlinear, simulation-based optimization routines in a short time.**

I. Introduction

THE emergence of high-throughput data collection technologies such as mass spectrometry and microarrays has created the need for powerful parameter selection to solve the many emergent models of systems biology. Previously, robust global optimization routines such as Genetic Algorithms (GA) have been applied to this task with mixed results[1]. On one hand, the large number of parameters and complex nonlinear nature of systems biology problems make them ideal candidates for GA. Unfortunately, many of these parameters are highly constrained to a range of just a few orders of magnitude, making traditional bit-flipping mutations highly inefficient, and often nonsensical. We propose a modified GA which takes advantage of the relatively narrow range of acceptable parameters by replacing high-mortality bit-flipping mutation techniques with a more conservative multiplicative scheme. In this way, many favorable properties of the GA are retained, but expensive nonproductive mutations are avoided, allowing for more desirable local behavior.

Optimization techniques using evolutionary principles have been theoretically and practically developed since the early 1950s. Biologists first found these techniques to be useful for modeling evolution, but with the rapid increase in desktop computational power, many other practical applications for evolutionary optimization followed, including parameter selection for chemical kinetics, electronic circuit design, code-breaking, and scheduling.

The Genetic Algorithm, one of the first biologically-inspired optimization techniques, was first introduced and popularized by John Holland in the 1970s [2]. However, GA remained largely academic until sufficient computing power emerged and the First International Conference on Genetic Algorithms was held in 1985. As discussed further in the next section, the Genetic Algorithm incorporates Darwinian concepts of 'survival of the fittest' to evolve optimal solutions by combining successful solutions from previous generations and using biological phenomena such as crossover and mutation to maintain diversity.

In addition to the GA, many other biologically-inspired optimization techniques have been developed during the past few decades to solve a wide variety of problems. Many of these new evolutionary optimization techniques represent only minor variations of other techniques to better suit a particular problem. Some of these methods include Bacteriological Algorithms, Ant Colony Optimization, and Genetic Programming.

Bacteriological Algorithms (BA), like GAs, treat solutions as individual organisms in a population struggling for survival, relying on the assumption that, in a heterogeneous environment, a population is always better adapted than any one individual. In contrast to GA, BA solutions reproduce asexually, producing a distinctive difference in the way they converge for certain problems. BAs have been used to generate optimum test cases for testing software components [3].

Ant Colony Optimization (ACO) creates hordes of virtual ants that explore the local solution space and look for optimum areas of productivity. The ACO is known to be robust because it can be applied to different problems with minor changes. This algorithm is also population based, and includes positive feedback among the virtual ants as a search mechanism [4].

Genetic Programming (GP) mutates and evolves entire computer programs rather than their parameters. The domain space contains collections of functions and terminals of a program, rather than numerical parameters. GP is incredibly diverse, and has potential to solve unbounded problems in artificial intelligence and machine learning [5].

Previously, GA have been applied with success to many problems, including the evolution of virtual creatures [6], and automatic design and manufacture of locomotive robots

Manuscript received April 16, 2007.

N. A. Shah is with the Department of Biomedical Engineering at the Georgia Institute of Technology, GA 30332 USA (e-mail: neil.shah@gatech.edu).

R. A. Moffitt is with the Department of Biomedical Engineering at the Georgia Institute of Technology and Emory University (e-mail: R.Moffitt@gatech.edu)

M. D. Wang is with the Department of Biomedical Engineering at the Georgia Institute of Technology and Emory University (corresponding author. e-mail: maywang@bme.gatech.edu).

1-4244-0788-5/07/$20.00 ©2007 IEEE

[7]. In the next section, we describe a Genetic Algorithm designed for choosing optimal strategies for an obesity reduction simulation.

II. Methods

A. Model

For the purposes of this paper, a multivariate, nonlinear optimization problem similar to many systems biology applications has been developed to demonstrate the power of a GA.

The GA is used to determine the optimal combination of therapies to recommend for a hypothetical 10-year national weight loss plan. Currently, two-thirds of the American population have a BMI above healthy standards (BMI > 25), and the goal of the plan is to reduce the number of overweight individuals as much as possible by the end of 10 years. The model takes as input a comprehensive strategy, in the form of three parameters, and outputs the projected status of obesity after 10 years.

The American population is modeled as four compartments, and the number of Americans in each weight category is stored as the elements in a vector $\vec{x}$.

$$\vec{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} \qquad (1)$$

where the elements [x_1 ... x_4] represent the number of morbidly obese (MO), obese (OB), overweight (OW), and healthy (H) people, respectively.

The percentage of people that transition between weight categories is given by the parameters k_1, k_2, and k_3, and the constant proportions j_1, j_2, and j_3, according to Fig. 1 with j_1=0.1, j_2=0.1, j_3=0.5. The constraints on k are as follows: $k_1 \in [0,.42]$, $k_2 \in [0,.15]$, $k_3 \in [0,.17]$. The upper bounds for the three parameters and the constant values are theoretical standards.

$$MO \underset{j_3}{\overset{k_3}{\rightleftarrows}} OB \underset{j_2}{\overset{k_2}{\rightleftarrows}} OW \underset{j_1}{\overset{k_1}{\rightleftarrows}} H$$

Fig. 1. Four Compartment Model. Arrows represent first order rate constants for the transitions between compartments.

The number of people in each compartment at a given time point, *t* is given by the elements of *x(t)*, and is updated at each time step using simple forward-Euler integration as shown in (2).

$$\vec{x}(t+\Delta t) = \vec{x}(t) + \mathrm{A}\vec{x}(t)\,\Delta t \qquad (2)$$

where A is a matrix given by

$$\mathrm{A} = \begin{bmatrix} -k_3 & j_3 & 0 & 0 \\ k_3 & -k_2 - j_3 & j_2 & 0 \\ 0 & k_2 & -k_1 - j_2 & j_1 \\ 0 & 0 & k_1 & -j_1 \end{bmatrix} \qquad (3)$$

B. Solution Evaluation

A fitness function must be defined in order to assign strength to each individual solution. Two factors influence the strength of each treatment: the number of overweight persons at the end of simulation, and the cost of the treatment. The fitness function **Φ**(I) in (4) is comprised of summing the cost of a particular set of treatments (increasing quadratically with magnitude to represent diminishing return on investment) with the number of people who transitioned from an unhealthy state to a healthy one.

$$\Phi(I) = (k_1^{\,2} + 23k_2^{\,2} + 300k_3^{\,2}) + (\sum_{i=1}^{3} x_i(10) - 200) \qquad (4)$$

The coefficients of parameters k_1, k_2, and k_3 are standardized costs associated with implementing the different treatments which transition individuals between weight classes. It is important to note that for this model, a lower fitness value corresponds to a better solution.

With the fitness function developed to calculate the relative strengths of individuals in the solution space, a GA can now be coded to find an optimal treatment strategy.

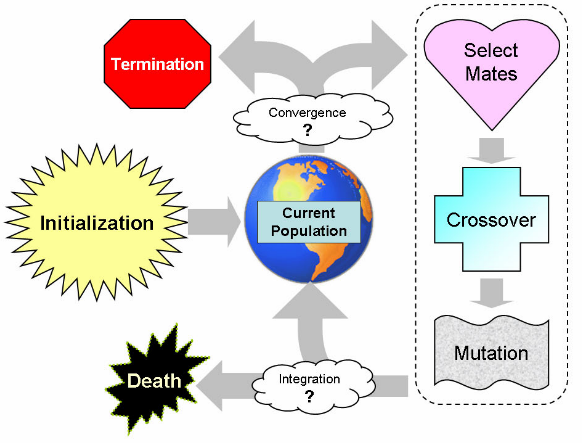

Fig. 2. Genetic Algorithm Workflow. First, the population is initialized, followed by multiple iterations of the selection, crossover, mutation, and integration steps. Finally, once convergence conditions are met, the algorithm terminates.

C. Genetic Algorithm Structure

A genetic algorithm generates an initial population of individual solutions using a predefined set of upper and lower bounds. These individuals are formatted as floating points with one leading digit, 4 digits in the mantissa, and 3 digits in the exponent. This population of individuals then undergoes selection, crossover, and mutation, and the population evolves from generation to generation until a termination condition is met (See Fig. 2). The optimal solution is then the individual with best fitness value when the termination condition is met.

After the initial population is created, two parents (I_M and I_F), are selected using a ranked selection method in which the probability of selecting an individual is determined by the relative rank of that individual's fitness value compared to other individuals in the population as shown in (5).

$$P(I_i) = \frac{i}{\sum_{j=1}^{L} j} \tag{5}$$

where *i* is the position of the individual after being ranked and L is the number of individuals in the population.

Once the probabilities are assigned to each individual, a random number $z \in [0,1]$ determines which individual I_M will be selected for mating. Likewise, another random number is generated to select the second parent I_F. These two parents will then mate, combining to create a new child.

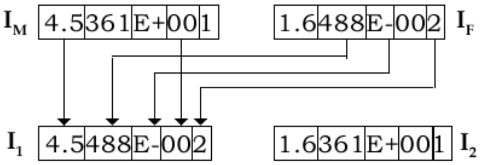

Fig. 3. Illustration of Crossover. Complementary children are formed by alternative splicing of parents at predetermined crossover points.

For crossover to occur, the parents must be spliced according to a predetermined set of crossover points. In Fig. 3, the set of crossover points is {2, 5, 6, 8} (in determining the crossover points, disregard the decimal point and exponent symbol 'E'). These slices are then alternatively exchanged to produce two new intermediate individuals, I_1 and I_2. One of these intermediates is randomly chosen as the intermediate child I_i, while the other dies off (analogous to a polar body during meiosis). The surviving child may then undergo mutation.

In nature, every cell that undergoes division does not mutate. There is a certain probability of mutation occurring during every cell division. Similarly, in a GA, a predetermined mutation rate $\mu \in [0,1]$ is set. A random number $z \in [0,1]$ is then generated, and under the condition that $z \leq \mu$, the intermediate who survived the crossover stage is mutated. A mutation occurs according to (6)

$$I_{mut} = I_i \cdot e^{\lambda} \tag{6}$$

where I_{mut} is the mutated individual and λ is a random number generated from a Gaussian distribution with unit variance and zero mean. This multiplicative mutation allows for diversity while preventing overly high variation in the individuals that occurs in bit-flipping mutation schemes.

Regardless of whether mutation occurs, the resulting individual is the offspring for this generation.

After the child is produced, it is tested to see whether it is strong enough to re-enter the population during the integration phase of a GA. The fitness value of this child is compared to the fitness value of the weakest individual in the population. If the child is stronger, it enters the population and the weakest member is eliminated. If, however, the child is not as strong as the weakest member, it does not survive.

D. Termination Conditions

One cycle of selection, crossover, mutation, and possible integration represents one generation in the life of this population. The cycle continues until termination conditions are met: either the predetermined maximum number of generations or convergence of the entire population to a stagnant solution, whichever comes first. The optimal solution is the individual with highest fitness at the termination point. In the following section, results of different initial conditions for the GA will be displayed.

III. Results

Simulations were run first without any treatment input to determine what might happen if steps are not taken to improve the American lifestyle. Fig. 4 shows the model's prediction for the number of people in each weight category over a 10 year span. The model predicts that the number of morbidly obese people will increase dramatically, and the number of healthy people will fall drastically as well.

The GA was then compared to a Monte Carlo (MC) stochastic optimization algorithm. This algorithm tests the fitness function at random locations in the search space without any memory or recombination [8]. Fig. 5 shows the convergence profile of a MC implementation for the weight loss therapy optimization problem.

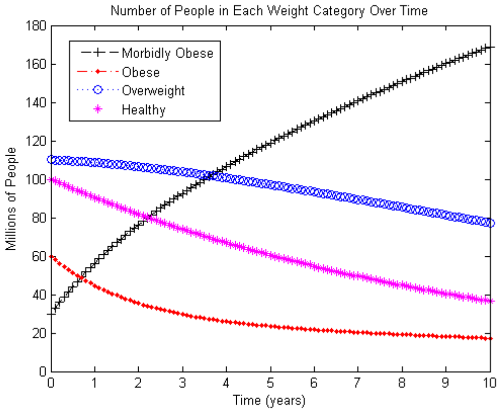

Fig. 4. If current conditions of American lifestyle remain untreated, the number of morbidly obese people will increase dramatically in the next 10 years. Consequently the number of healthy people will drop.

The MC in general achieves less optimal solutions than the GA for the same number of generations. (MC average solution: Φ = 1.846, GA average solution: Φ = 0.973 n = 5). Convergence of the GA, in contrast to MC, progresses more

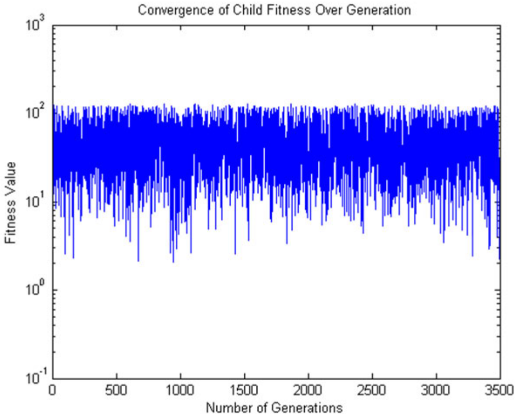

Fig. 5. A semi log convergence plot of the Monte Carlo Algorithm. The average converged fitness value of 5 trials was 1.846.

quickly, and is better at local optimization (Fig. 6).

Finally, Fig. 7 shows the predicted outcome of a 10 year plan under optimal conditions determined by the GA (Φ = 0.787). Notice the decrease in overweight and obese people over the 10 year period, and the increase in healthy people. The troubling increase in MO people is due to the relatively expensive (300 fold increase) cost of treatment for MO people versus that of OB and OW people. Therefore, in order to make the most Americans healthy at the lowest cost, it is best to treat MO people minimally, even if this is not a psychologically pleasant result.

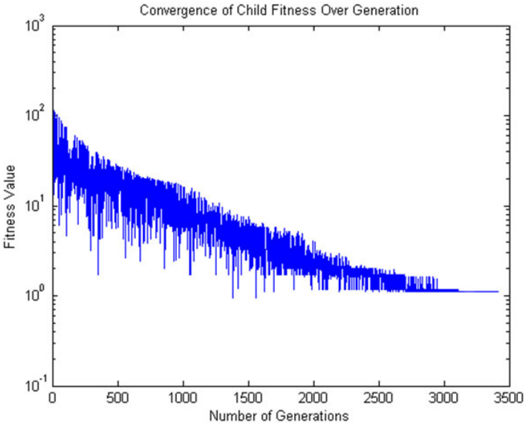

Fig. 6. A semi log convergence plot of the Genetic Algorithm. The average converged fitness value of 5 trials was 0.973.

IV. Conclusion

Here a Genetic Algorithm has been successfully applied to the problem of finding a treatment plan that will improve the health of most overweight Americans at the lowest cost. The algorithm provides an impressive result compared to the predicted progression of weight distribution without intervention. The Genetic Algorithm proves to have great potential for solving bigger problems with more complicated optimal solutions. Future work for this project includes running the Genetic Algorithm at different initial conditions (population size, mutation rates, terminating conditions, selection methods, etc.) and comparing convergence rates and solution reliabilities.

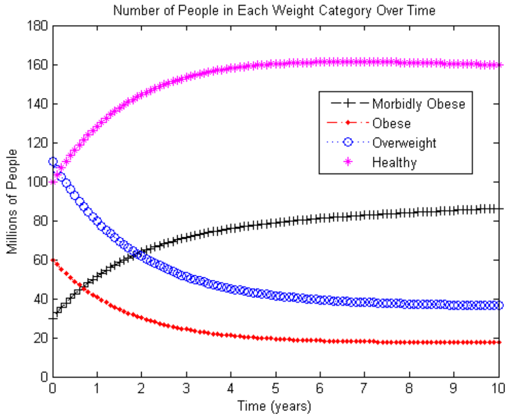

Fig. 7. Simulation result for optimal solution.

Acknowledgment

This research was supported by grants from Microsoft Research, the National Institutes of Health (Bioengineering Research Partnership R01CA108468, P20GM072069, Center for Cancer Nanotechnology Excellence U54CA119338), Georgia Cancer Coalition (Distinguished Cancer Scholar Award to M.D.W.), and the National Science Foundation (Graduate research fellowship to R.A.M). The authors would like to thank Dr. Wendy Newstetter for providing an interesting problem to work with.

References

[1] P. A. Henning, R. A. Moffitt, J. C. Allegood, E. Wang, A. H. Merrill, and M. D. Wang, "Computationally Predicting Rate Constants in Pathway Models," *Engineering in Medicine and Biology Society, 2005. IEEE-EMBS 2005. 27th Annual International Conference of the*, pp. 5093-5096, 2005.

[2] J. H. Holland, "GENETIC ALGORITHMS AND THE OPTIMAL ALLOCATION OF TRIALS," *SIAM J. COM'Ua*, vol. 2, 1973.

[3] B. Baudry, F. Fleurey, J. M. Jezequel, and Y. Le Traon, "Automatic test case optimization using a bacteriological adaptation model: application to .NET components," 2002.

[4] M. Dorigo, V. Maniezzo, and A. Colorni, "Ant system: optimization by a colony of cooperating agents," *Systems, Man and Cybernetics, Part B, IEEE Transactions on*, vol. 26, pp. 29-41, 1996.

[5] J. R. Koza, *Genetic Programming: On the Programming of Computers by Means of Natural Selection.* Cambridge, Mass.: MIT Press, 1992.

[6] K. Sims, "Evolving virtual creatures," *Proceedings of the 21st annual conference on Computer graphics and interactive techniques*, pp. 15-22, 1994.

[7] H. Lipson and J. B. Pollack, "Automatic design and manufacture of robotic lifeforms," *Nature*, vol. 406, pp. 974-8, 2000.

[8] J. E. Hirsch and R. M. Fye, "Monte Carlo Method for Magnetic Impurities in Metals," *Physical Review Letters*, vol. 56, pp. 2521, 1986.