

A Fast Convergent Algorithm to Generate New Code String Sequences

Don Iglehart and S.S. Iyengar

Department of Computer Science

Louisiana State University

Baton Rouge, Louisiana 70803

(504) 388-1495, (FAX) (504) 388-1495

ABSTRACT

The past decade has seen an explosive growth in studies of genetic coding and encoding algorithms for various applications. The development of high performance algorithms which may speed the convergence of genetic algorithms in molecular biology is an important area of research in computer science.

This paper presents a new approach for efficiently generating new string sequences by retaining the best of the old generations. The principal components of this new algorithm are as follows:

- * At the end of each generation the worst of the new generation is replaced by the best of the old generation,
- * A larger sampling of the state space is obtained for the initial generation,
- * Avoiding the (sometimes wasteful) duplication of strings which tends to increase over the generations.

Key words and phrases:

Genetic Algorithm

Genetic encoding

Code string sequence

1.Introduction

The past decade has seen an explosive growth in studies of genetic coding and encoding algorithms for various applications. The development of high performance algorithms which may increase the convergence of genetic algorithms in molecular biology is an important area of research in computer science.

Artificial life, represented by genetic encodings within massively parallel computers, may some day furnish a more diverse variety of simple life forms than any jungle of the real world. Generating such theoretical life forms will require greater knowledge of such processes than as natural selection and genetic algorithms--a simulation of gene splicing and mutation using some objective function to simulate natural selection.

This paper presents a new approach for efficiently generating new string sequences. The major components of this new algorithm are:

- * At the end of each generation the worst of the new generation is replaced with the best of the old generation,
- * A larger sampling of the state space is obtained for the initial generation,
- * Avoiding the (sometimes wasteful) duplication of strings which tends to increase over generations.

2. Background

Goldberg [1] wrote the definitive book on genetic algorithms (see also Iyengar [3]) which are a biologically inspired paradigm useful in solving nonlinear problems. One starts with a gene pool which is a diverse set of strings. The strings are evaluated with respect to an objective function, the value of which is called fitness and represents how well the string would solve the given problem. A subsequent generation of strings is created by mating strings from the former generation--the selection of strings to be mated is random, but weighted in favor of strings with good objective function values (fitness). The mating of two strings, or cross over, is simply a matter of taking the first part of one string, cutting it off at some random point, and filling the rest from the latter part of the second string. An infrequent random mutation of a string is beneficial to the process.

Thus the three features of a genetic algorithm are:

- * Selection,
- * Crossover,
- * Mutation.

CROSSOVER

String 1 -	a b c d e f		g h i j k l m n
String 2 -	A B C D E F		G H I J K L M N
mate 1&2 -	a b c d e f		G H I J K L M N

The working of genetic algorithms can be found out in Goldberg's book. At this point we note that genetic algorithms work with a coding of a parameter set, not with parameter themselves; they work with a population of strings, not a single point; they use payoff information (the objective function), not derivatives or other auxiliary information; and they use probabilistic transition rules, not deterministic rules.

A chromosome is a string of bits representing the encoding of a set of parameters; for instance a string of five 16 bit binary unsigned integers contains 80 bits which could represent 40 parameters each ranging from 0 through 3 (2 bits each), it could also represent 10 parameters each ranging from 0 through 255 (8 bits each), etc.

An optional approach which one might consider is to leave the string in a decoded form--for example a string of 40 integers each allowed to take values 0 through 3. Crossover would require encoding the two parameters at which the crossover occurs and performing a local crossover on them then decoding the resulting parameter. Using this structure would require more space but would substantially reduce the number of decodings required.

The standard approach for programming a genetic algorithm is to initialize the first generation of N strings with random values, compute the value of the objective function for each string, then build a weighted "roulette wheel" from these values. Two strings are randomly selected from the wheel and mated using the crossover described. This selection and mating is done until a new generation has been built. The process is repeated by computing the value of the objective function for each string, etc.

In the early generations of a run a few good strings will tend to dominate the analysis and strangle out the diversity too quickly, producing premature convergence. In the later generations of a run the best objective function values k =may not be sufficiently better than the average values to receive full attention due to them. It is, therefore, usually helpful to apply a scaling factor to the objective function of a genetic algorithm. A linear fitness scaling can be obtained by

$$s = af + b$$

where f is the fitness and s is the scaled fitness. The values for a and b can be determined by

$$s_{avg} = f_{avg}$$

$$s_{max} = cf_{avg}$$

the value of the scaling factor, c , is determined empirically.

3.A New Computational Genetic String Sequence Generator

i. Initialize an array of $2*N$ strings with random values, compute the value of the objective function for each string, collect the best N strings into the first half of the array, reinitialize the second half of the array with random values and repeat this process several times in order to obtain a better survey of the state space terrain for the first generation.

ii. Now fill an ancillary array of $2*N$ size with the above strings. Quicksort the first half of this ancillary array on objective function values (from bad to good) of the corresponding strings (0 through $N-1$). Repeat this sorting for remaining strings (N through $2*N-1$). Now starting at the each end of ancillary array we are able to replace the strings with worst values in the first part of the array with the strings with best values until the objective function values from the first part are better than those from the second part.

iii. The new generation can be build as follows: Assign the objective function value of first string to first string, objective function value of second string plus preceding value to the second string, objective function value of third string plus the preceding sum to the third string, etc. The last string is assigned the sum of all the objective function values. Selecting a weighted random string is now a simple matter of picking a random number scaled to the sum of the objective function values and doing a binary search on the construct we just described.

The standard genetic algorithm retains good strings from generation to generation because they mate with themselves--causing more and more duplications of these strings.

One of the greatest strengths of genetic algorithms is their ability to tolerate fuzzy objective functions. But often the objective function is well defined and its value for a given string will not change. It seems reasonable to discourage duplication in such cases, but we could lose better results in previous generations as well as diminish the effectiveness of the selection feature.

We have implemented duplication avoidance in two ways. During initialization process the strings should compared with the previous strings, from this point onwards duplicates can be avoided when two halves of the (ancillary) array are sorted by objective function value. With this approach duplicates are

eliminated after being evaluated; but if we preclude any string from being mated with itself, the likelihood of duplication is very low.

In the second approach we use a hash table (accessed by hashing a string) which contains the index numbers for all currently existing strings. When a new string is created it is passed to the hash table routine which compares it to all strings with same hashing. If it is not a duplicate it is added else it may be mutated and passed to hash table routine to repeat the above process. We see both of these methods avoid duplication of strings between two consecutive generations, but not over all generations.

In summary, we propose the following changes to the standard genetic algorithm:

- i. enlarge the size of the initial generation,
- ii. replace the worst strings of new generation by the best strings of old generation, and
- iii. avoid duplicates by any of three methods:
 - a. prevent duplicates in step ii above,
 - b. use a hash table to identify duplicates, then
 1. mutate duplicate, or
 2. discard duplicate and reselect/remate.

4. Results and Discussion

Real life problems involving gene splicing involve some level of interactive repercussions; additionally, the vast number of organisms which exist indicate that there are many reasonably optimal solutions. The following experiment has both of these characteristics even though it was not devised with this in mind. We have 40 users each requiring 0 to 100 units from any of 4 suppliers. Our string contains 40 parameters of 2 bits each, these parameters point to one of the four suppliers (0 through 3), indicating the supplier who must furnish the units. We seek for a string for which the suppliers will furnish percentages of the total in a predetermined ratio: supplier 0 will furnish 30%, supplier 1 will furnish 10%, supplier 2 will furnish 5%, supplier 3 will furnish 55%.

One optimization function that maximizes this is

$$\text{obj} = 200 - \sum |pct_{actual} - pct_{desired}|$$

The best will be 200, the worst can approach zero.

We will have an initial generation of 200 with no duplicates and subsequent generations of 100. Table I shows the results of performing twenty tests on each of the five approaches as in the table.

TABLE I. Unscaled Comparative Tests

Test	Mean	Std.	Median
Standard with duplication	919.5	1008.3	491.5
Best of two generations	13.9	4.0	13.0
Dups eliminated after Eval.	13.2	3.3	13.5
Hash without mutation	12.2	2.0	12.0
Hash with mutation	11.1	2.2	11.5

The average number of generations required to converged to obj > 198 is depicted for each approach. We see most of the improvement attributable to obtained the best of both generations (the second line of table I) rather than to the elimination of the duplicates.

The standard algorithm in the above tests did not have the benefit of conventional scaling techniques. In table II we have applied a linear fitness scaling to two algorithms, with conventional factor of 2, in order to compare the obtainable convergence rates. We see that the convergence rates are low with no parametric manipulation at all--no a priori assumptions about the proper setting for parameters! It will be necessary to study these procedures as applied to the conventional genetic algorithms test functions.

TABLE II. Scaled Comparative Tests

	Mean	Std.	Median
Standard with duplication	32.4	11.2	31.0
Best of two generations	11.2	2.1	10.5

A rapid convergence rate may well be at the expense of missing global maxima, that is, premature convergence. If so our routine can be tempered in several ways: gathering the best strings using a probabilistic scheme, or restricting the

portion of old strings which may be retained, or some combination of these two techniques.

Finally in table III, we compare the best two generations technique starting with an initial generation of 200, as we have been doing, versus the same procedure starting with an initial generation of 100. An equitable comparison of the two requires adding a generation to the first set (the 200); even so, it appears that increasing the initial generation survey produce some tangible benefits.

TABLE III. Different Initial Generation Size

	Mean	Std.	Median
Best of two generations (200)	11.2	2.1	10.5
Best of two generations (100)	13.6	7.5	12.0

5. Conclusion

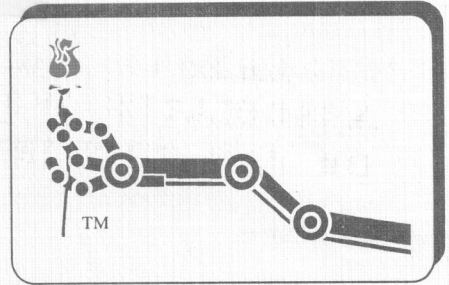
The problem used to test the new procedure has many reasonably optimal solutions and is subject to some interaction when any gene is changed--the previous supplier will be diminished and the new supplier will be increased. For this particular problem it was found that:

- i. Replacing unfit strings in the new generation with fit strings from the old generation enhances convergence and appears to be less sensitive to the parameterization required by scaling.
- ii. Increasing the size of the initial generation, providing a larger sampling of state space, appears to have some benefit. This increase need not require additional memory.
- iii. Avoiding duplicate strings has some benefit when the evaluation of the objective function is not fuzzy.

References

- [1] GOLDBERG, D.E., "Genetic Algorithms in Search, Optimization, and Machine learning", *Addison-Wesley*, Reading, Mass., 1989
- [2] IGLEHART, DON A., "Synergistic Control of N-Body Computer Generated Robots", PH.D. Dissertation, Louisiana State University, May 1993
- [3] IYENGAR, S.S., PRASAD, L., AND MORTON, D., "Structure of Genetic Algorithms in Biological Systems", *Structuring Biological Systems*, Ed. S.S Iyengar, pp. 1-16, 1991.

ISRAM[®]



INTERNATIONAL SYMPOSIUM ON ROBOTICS AND MANUFACTURING

URGENT MESSAGE

WAC '94 - World Automation Congress
ISRAM '94 - Maui, Hawaii, USA

Fax Number : + 505 291-0013 Alternative Fax : + (505) 277-1439
Telephone : + 505-277-5538

TO : ALL AUTHORS AND PARTICIPANTS OF MAUI Meeting Date : May 1, 1994

From : Mo Jamshidi, General Chair

Dear Friend :

As you may have already noted we have been negotiating to various airlines, car rental agencies, and travel brokers for the best possible prices for all of you. Here what we recommend :

Delta Airlines - American Express (Amex Canada, Inc.) 3030 Howard Avenue, Devonshire Mall at the Bay, Windsor, ONT. N8X 4T3, CANADA Phone : 519-969-8041 (Canada), 313-961-4443 (USA), FAX : (519) 969-9758 (Canada)

The prices listed below include a round trip airfare from specified city to Maui and includes a compact car rental for 7 days (worth about \$150 - based on 2 adults sharing a car). Dates are flexible and flights operate daily by DELTA Airlines, ISRAM's Official Carrier. Rates also include taxes on the flight and sales taxes on cars only, i.e. does not include car insurance.

city	Miweek Travel Sun-Wed (Outbound) Tue-Fri (Return)	Weekend Travel Thur-Sat (Outbound) Sat-Mon (Return)
Atlanta	\$805	\$ 921
Boston	\$ 787	\$873
Chicago	\$846	\$940
Detroit	\$767	\$854
Dallas	\$767	\$835
Los Angeles	\$480	\$527
Miami	\$816	\$979
Newark	\$787	\$873

P.O. BOX 14155, ALBUQUERQUE, NM 87191-4155, USA
FAX: (505) 291-0013 • E-MAIL: ISRAM@EECE.HOUDINI.UNM.EDU

Raleigh/Durham	\$883	\$979
San Francisco	\$480	\$527
Vancouver	\$835	\$921
Washington, DC	\$758	\$854

OTHER CITIES AVAILABLE FROM THE AGENT !

Inter-island air tickets are also available at about \$45 each between any two islands. Contact Debbie on the same.

HERE IS WHAT YOU HAVE TO DO :

1. Call Debbie (**collect**) at above number in US or Canada . Outside of US and Canada please fax in your needs to her.
2. Please give your **Delta** file no. (**I2241**) and **Alamo** file no. (**375171**) and ask for the above rates.

PLEASE ACT QUICKLY, BECAUSE THE SEAT ARE VERY LIMITED

Hotels for Other Isles :

ISRAM has managed to get special rates from three other islands - one hotel per island

1. **Oahu** (Honolulu & Waikiki Beach) - Royal Garden Hotel at \$76 per night excluding taxes.... Call 1-800-367-5666 or Fax 808-946-8777 and ask for Carmen Laasko and introduce yourself as an ISRAM Attendee in Maui.

2. **Hawaii** - The Big Island Hapuna Beach Prince Hotel (Available from August 18th on) at \$160 for 3 nights excluding taxes.... Call 1-800-882-7721 or Fax 808-882-1174 and ask for Lani Aubin and introduce yourself as an ISRAM Attendee in Maui.

3. **Kauai** : Islander on the Beach at \$76 - \$104 (depending on the ocean view, front, etc) per night excluding taxes.... Call 1-800-847-7417 or Fax 808-822-1947 and ask fro Vanessa Yatsuoka and introduce yourself as an ISRAM Attendee in Maui.

Best wishes & Aloha

mo

M. Jamshidi
WAC '94 & ISRAM '94 General Chair

PS: PLEASE INFORM ALL YOUR FRIENDS, COLLEAGUES, ASSOCIATES WHO PLAN TO COME TO MAUI OR SEND OR FAX THIS INFORMATION TO THEM.