

# A Web-Based *Mathematica* Application

David Fowler  
Teachers College  
University of Nebraska-Lincoln  
dfowler@cantor.unl.edu

This report describes a World-Wide Web-based *Mathematica* application for demonstrating a genetic algorithm. The application was installed on a server maintained by UN-L Information Services, with the assistance of Donna Liss, Director of Campus-Wide Information Services, and Linda Hunt, Steve Hunt and Alex MacAuly from Wolfram Research Incorporated. The application links input from clients using a web browser [Netscape 2.2 or later] to a server-side implementation of *Mathematica* 3.0b2.

This system can be considered in "alpha-test" mode, but it already stands as a proof-of-concept test with significance for distance education. This system shows that students using a web browser with no special helper applications can access an interactive page and perform mathematical experiments in real time. The example of a genetic algorithm program was chosen because

- a. genetic algorithms are currently a high-interest area in computer science;
- b. these kinds of experiment are not readily done with a graphing calculator.

The author has created a number of *Mathematica*-based simulations and is interested in making them available on the Web as interactive lessons without translating them into a different language, such as Java.

One possibility for use of the system is to provide a demonstration for students with real-time prompting from a teacher connected through a video conferencing system. After students step through the on-line version, they could download a *Mathematica* notebook for extended experimentation on their home or school computer. This phase of the operation would, of course, require that they have the *Mathematica* application on their machine. Alternatively, the students could perform off-line experiments, including "pencil-assisted cognition" and then return to the on-line system to test out their ideas.

## ■ Sparse rulers

Sparse rulers [Golomb rulers] will measure all the consecutive integer lengths from 0 to a given number  $n$ , although the ruler has fewer than  $n$  marks. For example the ruler with marks at 0, 1, 4, and 6 will measure the lengths 2—the distance between 4 and 6—and the lengths 3, and 5 as well.

Applications of sparse ruler theory include topics to engineering, physics and coding theory. For example, the spacing of radio telescopes, called a "minimum base-line array" [Dewdney, 1987] is based on the theory of sparse rulers. There is no known formula for finding the minimal sparse ruler for a given value  $n$ .

## ■ Genetic algorithms

Genetic algorithms and, more generally genetic programming, operate by assigning fitness values to an evolving population of solutions. Considerable information on genetic algorithms is available as described in the **References** section.

The sparse ruler problem provides a simple demonstration of a genetic algorithm. The basic procedure is to take two "parent" rulers and combine segments of one ruler with segments of the second ruler, to produce a set of "offspring" rulers. The process is somewhat analogous to chromosomal crossing-over. The offspring rulers are evaluated for "fitness," with the best two rulers selected to be the new parents. Successive generations of rulers tend to have greater fitness, that is, they come closer to being minimal sparse rulers.

The crossing-over process is supplemented by a "mutation" process, in which new values may spontaneously occur. Here is an example.

```
r1 = {0, 1, 2, 12};
r2 = {0, 3, 7, 12};
offSpring[{r1, r2}, 5] =
{{0, 1, 2, 12}, {0, 1, 2, 3, 12}, {0, 7, 11, 12}, {0, 1, 4, 12}, {0, 1, 2, 7, 12}}
```

Each time the offSpring function is called, a different set of new rulers is produced. In the above example, the parent rulers r1 and r2 produced the five offspring rulers shown. The value 4 in the ruler {0,1,4,12} is the result of the mutation function. Two of the offspring then form the parents for the next generation. The selection of the two new parents depends upon the definition of "fitness." For the algorithm used in this example, the fitness function is defined as

$$\text{fitness}(s) := \frac{(p + 1) - \text{Length}[s]}{(p + 2) - \text{Length}[\text{span}(s)]};$$

The respective fitness values for the five rulers shown above are:

```
{1.125, 1.33333, 1.28571, 1.28571, 1.6}
```

## ■ The Web-based *Mathematica* application

The author uploaded a *Mathematica* 3.0 notebook to the appropriate directory, where it was converted to html format with appropriate cgi-bin programming. The intention of the WRI developers is to have the conversion system fully automatic. At present, some features of the conversion and the interface are under construction.

The interface will allow a student with no prior knowledge of *Mathematica* syntax a chance to perform experiments with sparse rulers. For example, the student can

- enter a ruler and automatically measure its fitness.
- enter two rulers and automatically generate offspring rulers.
- enter two rulers and automatically generate numerous generations with an associated graph that plots the evolving fitness of the generations.

Figure 1 shows a typical window in the current version of the system. The student can accept the default ruler {0, 1, 3, 8, 12} or define a new ruler, with standard mouse-and-click editing. Clicking the Evaluate function on the menu bar then enters that ruler into the program.

▼ **Defining a fitness function.**

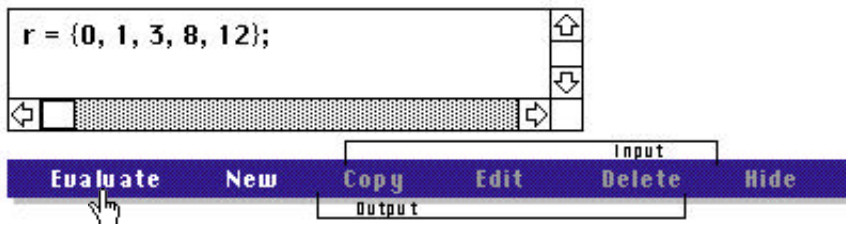


Figure 1

Figure 2 shows the result of evaluating one of the subroutines of the genetic algorithm. The function `Span` shows the set of lengths that can be measured by the given ruler `r`. Notice that the result of evaluation is displayed below the window.

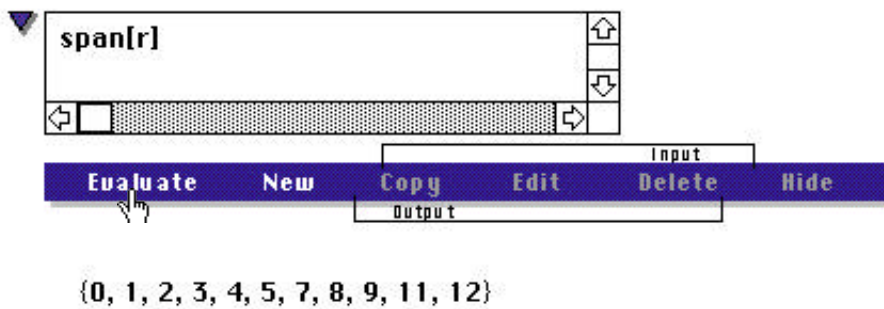


Figure 2

Figure 3 shows a block of *Mathematica* code for defining a section of the algorithm. The student can simply evaluate this code and skip the details, or she can select the code, and if desired, copy the code to a text editor on her computer [or to a blank *Mathematica* notebook page on her computer, if she has the application].

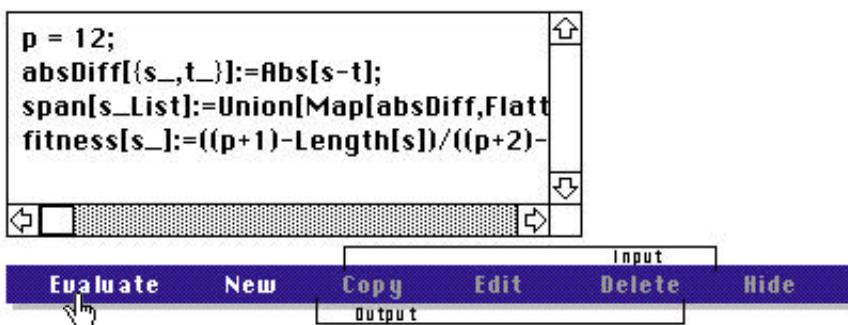


Figure 3

Figure 4 shows a plot of the fitness values of the best ruler in each of 15 generations. The evolution is clear, but not monotonic. Rulers of length greater than about 18 show interesting and complex behaviors, including an analog of the "punctuated equilibrium" phenomenon.

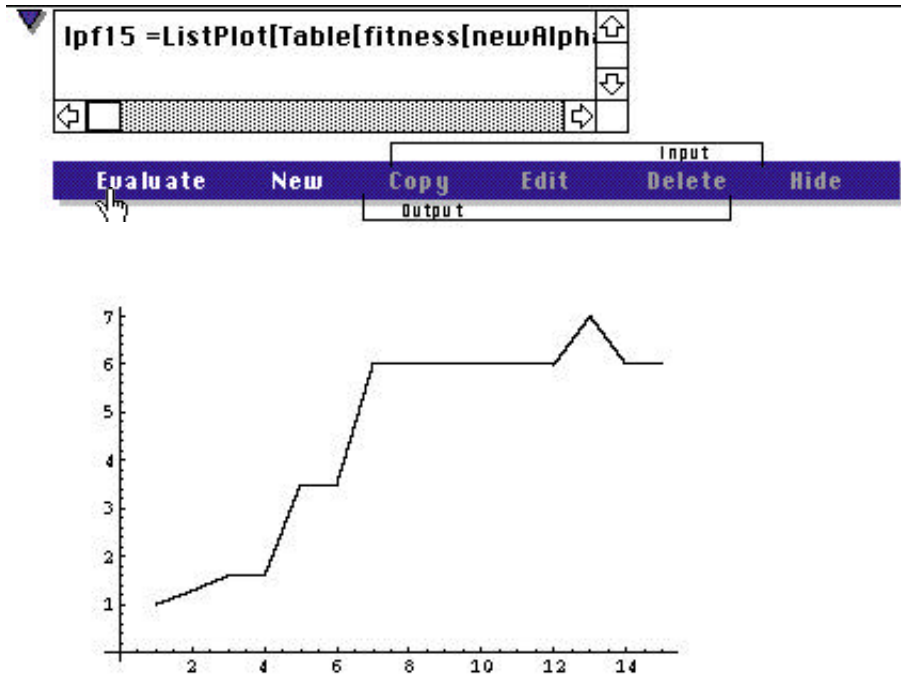


Figure 4.

### ■ Instructional design issues

Figure 5 shows a screen-capture of a *Mathematica* notebook designed for interactive study on a personal computer. In the Web-based version, most of the explanatory text has been removed, leaving the basic functions and some default values for the student to experiment with. The question of design for a Web-based application is not a simple one. Should the text be read in hard-copy format prior to the exploration? Should an on-line window contain optional text, or perhaps some additional instructional material, such as a short video clip or animated tutorial? As previously mentioned, a real-time video conferencing window is also a possibility.

## Genetically Seeking Sparse Rulers

David Fowler  
University of Nebraska-Lincoln

□note: in the following notebook, some of the *Mathematica* code is hidden in "initialization cells." This makes it easier to follow the general outline of the argument. Readers interested in using *Mathematica* as a programming language [it's very elegant] are encouraged to select the initialization cells and change their Attributes [under Style] to 'Open'.

■ Seeking sparse rulers to measure all the numbers from 0 to 12.

The sparse ruler problem is ideal for illustrating the process of genetic solution. In the following notebook, a ruler for the number  $p$  is simply represented as a list beginning with 0 and ending with  $p$ .

A genetic algorithm uses the analogy of biological evolution. Two possible solutions to a problem combine to produce "offspring solutions." The "fittest" of these solutions survive to produce new offspring. Mutations occasionally occur and may produce even fitter offspring.

□Defining a fitness function.

As an example, consider the ruler for  $p=12$  for the following arbitrarily selected values:

```
in[7]>p=12;  
    r = {0, 1, 3, 8, 12};
```

Certainly part of the fitness of a ruler is its ability to measure most of the numbers from 0 to 12. We will call the list of possible numbers that a ruler  $r$  can measure the span of  $r$ .

For the example  $r$ ,

```
span[r]  
{0, 1, 2, 3, 4, 5, 7, 8, 9, 11, 12}
```

Figure 5

The issues of Web-based presentation also include curricular issues related to uses of *Mathematica* itself. If students were expected to learn the details of the *Mathematica* code, then additional resources might be needed. For example, *Mathematica* contains an excellent Help Browser, as well as on-line help and error messages. Since the genetic algorithm includes many specially-defined functions, additional usage information can be built into the notebook.

Returning to the notion of Web-based experimentation, an obvious advantage of this experimental system is the capability for hyper-links to other Web sites dealing with *Mathematica*, genetic algorithms, or special pages devoted to the sparse ruler problem. In addition, if the Web-based server were being used in a credit course, then a list-serve could be linked to a discussion page. Special access privileges to students could allow them to upload their own notebooks to a subdirectory, from which point the notebooks would be automatically converted to html documents for exploration by other members of the class.

## ■ References

Holland, John H. *Adaptation in Natural and Artificial Systems*. MIT Press. (1992).

Kauffman, Stuart A. *The Origins of Order*. Oxford Press. (1993).

Koza, John R. *Genetic Programming*. MIT Press. (1992).

- To try the server: <http://crc-sybase.unl.edu/cgi-bin/Notebooks>

