

# Genetic Algorithm Approach for the Center Sequence Problem Arising in Post-transcriptional Gene Silencing

Holger Mauch  
University of Hawaii at Manoa  
Dept. of Information and Computer Science  
1680 East-West Road  
Honolulu, HI 96822

hmauch@hawaii.edu

March 30, 2003

## Abstract

A fundamental aspect of Post-transcriptional gene silencing (PTGS) is the requirement of homology between the transgene sequence and the virus or mRNA sequence being silenced. For example, virus-resistant plants are resistant only to viruses that are very closely related to the virus from which the transgene was derived. One idea is to devise an artificial sequence that is at least 90 – 95% homologous to all virus genes. This requires an algorithm which can determine an artificial sequence with an optimal homology (or at least a 90 – 95% homology) to all of the virus sequences. The genetic algorithm presented in this paper serves this purpose. It should be of great value to all researchers who utilize PTGS, gene quelling, or RNAi.

Mathematically, the task is to find the radius of a code  $S \subset \{A, C, G, T\}^n$ . Experimental results suggest that this NP-complete optimization problem can be approached well with a custom-built genetic algorithm (GA). Alternative approaches (exhaustive search, branch and bound, heuristics) are discussed and put in contrast to the GA approach.

*Keywords: Genetic Algorithm, Post-transcriptional Gene Silencing, Radius of Code*

# 1 Biological Motivation

Post-transcriptional gene silencing (PTGS) is a molecular mechanism in plants that is responsible for the degradation of specific mRNAs before they can be translated into protein [3, 15]. Plant virologists were among the first to utilize this mechanism when they transformed tobacco plants with a Tobacco mosaic virus (TMV) gene, and found that some of the transgenic plants were resistant to TMV [8]. Since then, several virus-resistant transgenic plants have been developed using a variety of virus genes demonstrating that it is not the function of the gene that is responsible for the resistant phenotype, but in fact a triggering of some inherent molecular pathway in the plant. In the case of virus-resistant plants, the PTGS mechanism degrades the viral RNAs before they can be replicated or translated into protein. In addition to invading viral genes, PTGS has the capability to silence endogenous genes as well [12, 14], making it a valuable tool to genomics researchers interested in gene function.

In recent years a great deal of research has been focussed on PTGS, resulting in a better understanding of its molecular pathway. It has been discovered that a fundamental aspect of PTGS is the requirement of homology between the transgene sequence and the virus or mRNA sequence being silenced [3, 5, 15]. For example, virus-resistant plants are resistant only to viruses that are identical or very closely related to the virus from which the transgene was derived. Studies suggest that this threshold sequence homology is about 95% [13], although effective silencing has been reported for homologies under 90% [4, 11].

In order to obtain resistance against a broader range of viruses, or for the genomics researcher who wishes to silence an entire gene family, simply transforming the plant with a gene from a single virus, or a single member of the gene family may not suffice. For example, a researcher wishes to develop a plant resistant to viruses  $A$ ,  $B$ , and  $C$  who have genes  $a$ ,  $b$ , and  $c$ , respectively. The sequence homology between  $a$  and  $b$ ,  $a$  and  $c$ , and  $b$  and  $c$ , are 98%, 93%, and 90%, respectively. Assuming a 95% homology threshold is required to induce the PTGS mechanism, plants transformed with  $a$  or  $b$  would be resistant to viruses  $A$  and  $B$ , but susceptible to virus  $C$ . Similarly, plants transformed with  $c$  would be resistant to virus  $C$ , but susceptible to viruses  $A$  and  $B$ . One solution would be to transform plants with  $a$  or  $b$ , and  $c$  [10]. However, as the number of viruses for which resistance is desired increases, the transgene would eventually become too large for current transformation protocols. Another solution would be to devise an artificial sequence that is at least 95% homologous to  $a$ ,  $b$ , and  $c$ . Most sequence analysis programs offer an algorithm that determines a *consensus sequence*, i.e. the most common nucleotide at each position in a collection of sequences. A consensus sequence would be very homologous to most of the sequences in the collection but still have low homology to divergent sequences in the collection which are less common. Therefore a plant containing an

artificial transgene derived from a consensus sequence may be resistant to most of the viruses “in the collection” but may still be susceptible to the less common, divergent viruses.

Therefore an algorithm which can determine an artificial sequence with an optimal homology (or at least a 95% homology) to all of the sequences in a collection would be of great value to all researchers who utilize PTGS, gene quelling, or RNAi. Such an artificial sequence could be used to silence a range of viral or endogenous genes much broader than any conventional “wild type” transgene.

## 2 Formal Description of the Problem

In the following subsections the input and output requirements of this optimization problem are described formally.

### 2.1 Input

The input is a set  $S = \{s_1, \dots, s_m\}$  of  $m$  nucleotide sequences, where  $s_i \in \{A, C, G, T\}^n$  for  $1 \leq i \leq m$ , and  $n, m$  are natural numbers (notation:  $n, m \in \mathbf{N}$ ). Note that  $S$  is not necessarily a set any more, if identical nucleotide sequences are allowed. Eliminating duplicates and considering  $S$  as a set does not change the solution to the center sequence problem as described below; however it could change other established characteristics of collections of nucleotide sequences such as the consensus sequence. If  $S$  is considered as a set it is also commonly called a *code* of length  $n$  (see [6]).

### 2.2 Optimal Output

An optimal output is a nucleotide sequence  $s^* \in \{A, C, G, T\}^n$ , such that

$$z = \max_{s \in S} \{d(s^*, s)\} \quad (1)$$

is minimal, where the distance  $d$  between two nucleotide sequences  $t, u \in \{A, C, G, T\}^n$  is defined as follows:

$$\begin{aligned} d: \quad & \{A, C, G, T\}^n \times \{A, C, G, T\}^n \rightarrow \{0, 1, \dots, n\} \\ & (t, u) \mapsto d(t, u) = \sum_{i=1}^n \delta_{t_i, u_i}, \end{aligned}$$

i.e. the number of loci, in which  $t$  and  $u$  differ, using the notation  $t_i$  to denote the  $i$ -th character of sequence  $t$ ,  $u_i$  to denote the  $i$ -th character of sequence  $u$ , and  $\delta$  being Kronecker’s delta.

Let’s call such a sequence  $s^*$  a *center sequence* and the optimization problem described above the *center sequence problem* (CSP). Note that  $s^*$  is not necessarily unique. We could try to find all such  $s^*$ , but since usually the biologist will be satisfied with any one we adopt a convention typical for optimization problems: any  $s^*$  found will be sufficient.

In the language of optimization problems, there are  $n$  decision variables  $(s_1^*, \dots, s_n^*)$  which we denote as the vector  $s^*$ . The only constraints in this optimization problem are that the decision variables take on only the discrete values  $\{A, C, G, T\}$ . The objective function is (1).

### 2.3 Alternative Terminology

Instead of describing the distance between two strings  $t, u \in \{A, C, G, T\}^n$  as an integer, we might as well describe the similarity  $g$  between these two strings  $t, u \in \{A, C, G, T\}^n$  as a rational number (or percentage) as follows:

$$\begin{aligned} g : \{A, C, G, T\}^n \times \{A, C, G, T\}^n &\rightarrow [0; 1] \\ (t, u) &\mapsto g(t, u) = \frac{n-d(t,u)}{n} \end{aligned}$$

This is merely a change in terminology. Our goal then becomes to maximize

$$\min_{s \in S} \{g(s^*, s)\}.$$

This terminology is often preferred in biology because the similarity  $g$  describes the homology between two nucleotide sequences.

### 2.4 Suboptimal Output

If we desire an optimal output as described so far, we have an optimization problem. However, in practice we can often relax the requirement of finding the global optimum in our search. E.g. we might already be pleased if we find any  $\hat{s}$ , such that

$$\min_{s \in S} \{g(\hat{s}, s)\} \geq 0.95.$$

Note that such an  $\hat{s}$  might not exist if we require too much similarity. If it does exist, then  $\hat{s}$  might not be unique, as in the optimization problem.

All solution approaches discussed in this paper allow for an “early exit” if during the computation a solution candidate  $\hat{s}$  is seen that meets the user’s similarity requirement.

### 2.5 More Definitions and Observations

#### Observation 2.1

*The search space  $\{A, C, G, T\}^n$  together with the metric  $d$  is an  $n$ -dimensional discrete metric space.*

**Proof:** Let  $s_1, s_2, s_3 \in \{A, C, G, T\}^n$ . The three metric axioms hold due to the way  $d$  is defined:

1.  $d(s_1, s_2) \geq 0$ , where  $d(s_1, s_2) = 0$  iff  $s_1 = s_2$  (Definiteness).
2.  $d(s_1, s_2) = d(s_2, s_1)$  (Symmetry).
3.  $d(s_1, s_2) \leq d(s_1, s_3) + d(s_3, s_2)$  (Triangular Inequality).

■

Nucleotide sequences are points in this space. Geometrically, we are looking for a center of the smallest  $n$ -dimensional sphere (or ball), which contains all  $s \in S$ . The radius of this smallest ball is commonly called the *Radius of code  $S$*  or briefly  $R(S)$ . The formal definitions follow.

**Definition 2.2 (Ball)**

The ball in  $\{A, C, G, T\}^n$  of radius  $r$  with center  $s_c \in \{A, C, G, T\}^n$  is

$$B_n(s_c, r) := \{s \in \{A, C, G, T\}^n \mid d(s_c, s) \leq r\}.$$

**Definition 2.3 (Radius of a Code)**

The radius of a code  $S$ , denoted  $R(S)$ , is the smallest integer  $r$  such that  $S \subset B_n(s, r)$  for some  $s \in \{A, C, G, T\}^n$ .

### 3 Classification of the Problem

Frances and Litman showed in [6], that the minimum radius (MR) decision problem of arbitrary binary codes is NP-complete. Based on this fact it will be shown that the task of finding a center sequence as described above is NP-complete.

The decision problem associated with the center sequence radius problem can be stated as follows:

**Definition 3.1 (CSRDP)**

*Center Sequence Radius Decision Problem (CSRDP)*

*Input: A code  $S \subset \{A, C, G, T\}^n$  and a positive integer  $k$ .*

*Output: Is  $R(S) \leq k$  ?*

**Theorem 3.2**

*The CSRDP problem is NP-complete.*

**Proof:** CSRDP is in NP, since a nondeterministic Turing machine can "guess" a center sequence and verify that  $R(S) \leq k$  in polynomial time.

To prove that CSRDP is NP-hard, we show that the NP-complete MR decision problem from [6] is polynomial time reducible to CSRDP. The MR decision problem is identical to CSRDP, except that strings are taken from the binary alphabet  $\{0, 1\}$  instead from the alphabet  $\{A, C, G, T\}$ . Therefore a simple mapping  $f : \{0, 1\} \rightarrow \{A, C, G, T\}$  with  $f(0) = A$  and  $f(1) = C$  transforms the input strings character by character and serves as the desired reduction and thus  $\text{MR} \leq_p \text{CSRDP}$ . ■

**Definition 3.3 (CSRP)**

*Center Sequence Radius Problem (CSRP)*

*Input: A code  $S \subset \{A, C, G, T\}^n$ .*

*Output:  $R(S)$ .*

**Definition 3.4 (CSP)***Center Sequence Problem (CSP)**Input:* A code  $S \subset \{A, C, G, T\}^n$ .*Output:* A sequence  $s^* \in \{A, C, G, T\}^n$  such that  $S \subset B_n(s^*, r)$  and  $r$  is minimal.

There is not much hope of finding a polynomial time algorithm for the CSP. If we could compute an  $s^*$  as stated in the CSP in polynomial time, we could, from  $s^*$ , compute the corresponding radius  $R(S)$  in polynomial time solving the CSRDP and thus the CSRDP in polynomial time, which would imply that  $P = NP$ .

## 4 Problem Characteristics

### 4.1 Elimination of Monomorphic Loci

A locus which takes on the same nucleotide base value  $x \in \{A, C, G, T\}$  across all sequences in  $S$  (i.e. a monomorphic locus) does not add complexity to the CSP, since a center sequence  $s^*$  necessarily also takes on the same value  $x$  at that locus. (If it would not take on the same value, the minimality requirement for the radius is violated.) This observation allows for a preprocessing of a CSP instance. Since only polymorphic loci are of interest, we store the values of all monomorphic loci along with the position of the loci in a table. Then all monomorphic loci can be eliminated from  $S$ , which reduces the sequence length  $n$  to an *essential sequence length*  $n'$ , where  $n'$  is the number of polymorphic loci. After solving the preprocessed instance of the CSP, it is easy to construct the solution to the original CSP.

### 4.2 Distance Matrix for All Pairs of Sequences

A useful tool for characterizing a problem instance is the  $m \times m$  distance matrix  $D = (d_{i,j})$ , which records the distances  $d(s_i, s_j)$  between pairs of virus sequences  $s_i$  and  $s_j$ , where  $1 \leq i, j \leq m$ . The distance matrix is symmetric and has zero entries on its diagonal.

An inspection of the distance matrix  $D$  leads to a lower bound for the radius  $R(S)$  as follows. Let  $d_{max}$  be the maximum distance entry in  $D$ . Then

$$R(S) \geq \lceil d_{max}/2 \rceil. \quad (2)$$

The proof of (2) follows from the triangular inequality established in the proof of observation 2.1. Say  $d_{max} = d(s_i, s_j)$ . Then  $d(s_i, s_j) \leq d(s_i, s^*) + d(s^*, s_j) \leq R(S) + R(S) = 2R(S)$ .

If, for practical purposes, we are happy with a suboptimal output, we will usually request a maximum distance (equivalent to a minimum similarity). If the requested maximum distance is less than the lower bound described above, then it can immediately be concluded that no sequence meets the requirements.

**Observation 4.1**

*Equality does not necessarily hold in (2).*

**Proof:** Look at the following example. Let  $S = \{s_1, s_2, s_3\}$  with  $s_1 = AAAA$ ,  $s_2 = CCCC$ , and  $s_3 = GGGG$ . The distance matrix is

$$D = \begin{bmatrix} 0 & 4 & 4 \\ 4 & 0 & 4 \\ 4 & 4 & 0 \end{bmatrix}.$$

The maximum entry is 4 and thus the lower bound is 2. But  $R(S) = 3$ . ■

## 5 Nonevolutionary Approaches

### 5.1 Exponential Time Approaches

An exhaustive search of the space  $\{A, C, G, T\}^n$  is easy to program but inefficient. Calculating the distance from each of the  $4^n$  points in the search space to all  $m$  points in  $S$  in order to find a center sequence takes  $mn4^n$  pairwise base comparisons, since it takes  $n$  base comparisons to calculate the distance between a pair of points. This method is not practical for large  $n$ .

Branch and bound techniques allow to prune the search space and to improve the brute force approach, but the improvements are not significant enough to implement an algorithm with sufficient efficiency for real world sized problems.

These approaches suffer from their exponential time requirements and are not suitable for real world problem sizes. Our customized GA on the other hand has acceptable time requirements and outputs results within a reasonable amount of time.

### 5.2 Heuristics

One heuristic is to calculate the consensus string  $x$ , then determine which  $s \in S$  is farthest away from  $x$ . The strategy is to iteratively correct  $x$  by increasing its similarity to the farthest outlier  $s$ . One way to accomplish this is to consider  $S$  as a multiset, then keep adding the most distant sequence  $s$  to  $S$  and recomputing the consensus sequence in every step.

A second heuristic builds a solution sequence from left to right, with a greedy selection at each locus. The obvious greedy choice is to select the allele from that  $s \in S$ , which is farthest away at the time of the choice. (Distances are computed only considering loci 1 to  $n_t$ , where  $n_t$  is the length of the solution sequence at step  $t$ .)

Other heuristics might employ a divide and conquer strategy by dividing the code  $S$  into smaller portions and later combining the partial solutions.

Or, starting with only few sequences in  $S$ , iteratively add more sequences to  $S$  until all  $m$  sequences are included. Note that divide and conquer strategies which divide a sequence of length  $n$  into two parts  $s'$  and  $s''$  with  $|s'|+|s''|=n$  are inherently flawed, for they eventually lead to the calculation of the consensus sequence. Also, approaches using properties that hold in Euclidean spaces only are generally not suitable for this metric space.

The problem with all these heuristics is that they might not converge to a center sequence. Most results are too far off to be of any use. GAs cannot guarantee to find a center sequence either, but they are more robust and have a good chance to produce a result that is good enough to be useful, as demonstrated by experimental results.

## 6 Genetic Algorithm Approach

### 6.1 Introduction to Genetic Algorithms

Genetic algorithms (GA) [1, 7, 9], inspired by biological systems, mimic the Darwinian evolution process. GAs tend to make more copies of individuals (fixed-length character strings) which exhibit higher fitness, as measured by a suitable fitness function. Over time individuals in the population evolve because of natural selection and because genetic operations (mutation, recombination) modify individuals.

After the random generation of an initial population, the GA enters an evaluation - selection - alteration - cycle until the termination criterion (e.g. maximum number of generations, perfect individual sighted, etc. ) is satisfied. GAs are a robust search technique and they are widely used in optimization. The discrete search space and the lack of further constraints indicate that the CSP should be a good application area for GAs.

### 6.2 Genetic Algorithm Design for the Center Sequence Problem

“A representation should always reflect fundamental facts about the problem at hand” [2, p.97]. For the CSP the most natural representation for candidate solutions are strings over the alphabet  $\{A, C, G, T\}$ . Therefore the population in our GA is a collection of strings from  $\{A, C, G, T\}^n$ .

The fitness function  $f$  used to evaluate an individual string  $\tilde{s}$  is simply the objective function (1), i.e.

$$f(\tilde{s}) = \max_{s \in S} \{d(\tilde{s}, s)\}$$

Note that a lower fitness value is considered better and that the “fittest” individuals that can ever evolve are center sequences — they have a fitness of  $R(S)$ .

The procedures we employ for random initialization, selection (tournament style), recombination (uniform crossover), and mutation (uniform mutation probability  $\theta_M$  for each locus) are widely used in the GA community.

## 7 Experimental Results of GA Approach

The actual data of the problem instance studied consists of  $m = 116$  virus sequences each having a sequence length of  $n = 626$ . Preprocessing revealed that only  $n' = 218$  out of 626 loci are polymorphic. According to section 4.1 we can simplify our studies by looking at sequences which have been reduced to their polymorphic loci. The essential sequence length is thus 218.

The calculation of the distance matrix reveals that the maximum entry is  $d_{max} = 66$ . Inequality (2) implies that no center sequence can have a radius of less than 33 (corresponding to a 94.73 % similarity).

Within 20 minutes on a Pentium 266 MHz the GA finds a sequence with a distance of 34 (corresponding to a 94.57 % homology), which comes very close to the theoretical upper bound of 33 (94.73 % homology) for this problem instance.

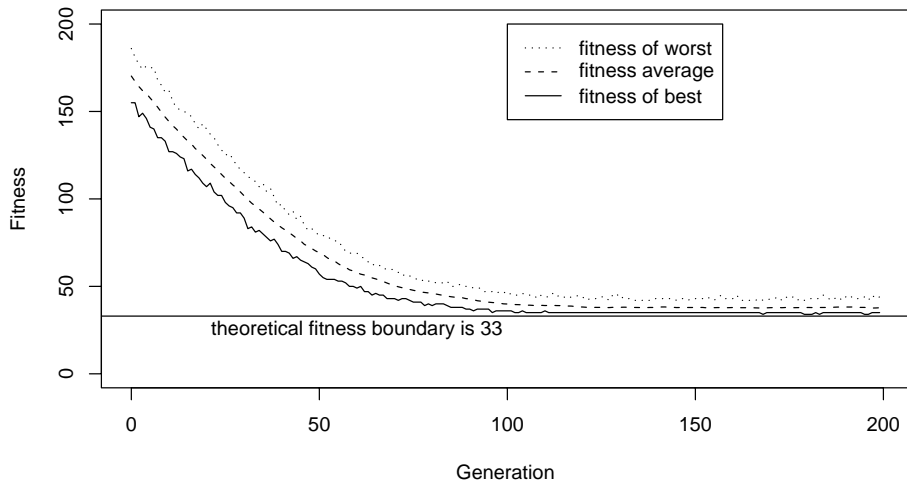


Figure 1: Fitness Statistics for a Sample GA Run

The statistics of a typical GA run are graphed in figure 1 and figure 2. Most of the optimization progress happens within the first 100 generations. After 100 generations the fitness variability stays fairly stable — an indication of decreased selection pressure due to less diversity in the population.

No GA run found a center sequence  $s^*$  with a distance of  $\lceil d_{max}/2 \rceil = 33$ , which would correspond to the lower bound for the radius according to

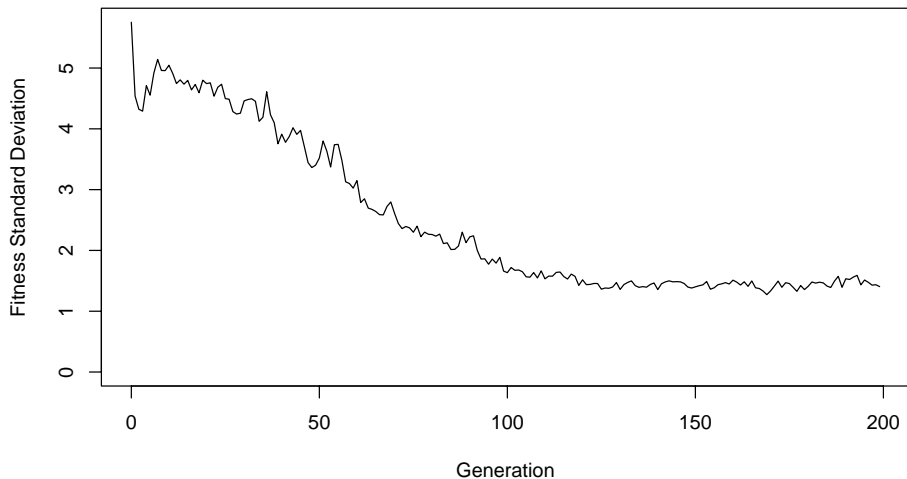


Figure 2: Fitness Standard Deviation for a Sample GA Run

inequality (2). However, potential center sequences with a distance of 34 have been found by 75% of all GA runs (see figure 3). Because randomness is intrinsic to the GA approach it remains unsolved whether there exist better sequences (establishing a radius of 33) or whether a sequence with distance 34 actually constitutes an optimal sequence, which could very well be the case according to observation 4.1. For practical purposes (PTGS), a sequence with distance 34 corresponds to a similarity of 94.57 % — sufficient homology for this instance.

For the creation of the success statistics (figure 3), a sequence with a distance (and therefore fitness) of 34 was considered a success (and the GA stopped), whereas a fitness of 35 was considered not good enough and evolution continued in that case. If after 250 generations no individual with fitness 34 evolved, the GA run was considered a failure. Note however that every GA run produced an individual with a fitness of 35, which still corresponds to a 94.41% similarity — for practical puposes still a very good result.

We performed 20 GA runs with the following parameters.

- Maximum number of generations: 250.
- Population size: 500.
- Recombination parameter (crossover rate)  $\theta_R = 0.6$ .
- Mutation parameter  $\theta_M = 0.005$ . This is close to the common recommendation to choose  $\theta_M$  as  $1/n' = 1/218$ .

- Selection parameter  $\theta_S = 1$ . Nonoverlapping generations. Tournament selection with tournament size 2.

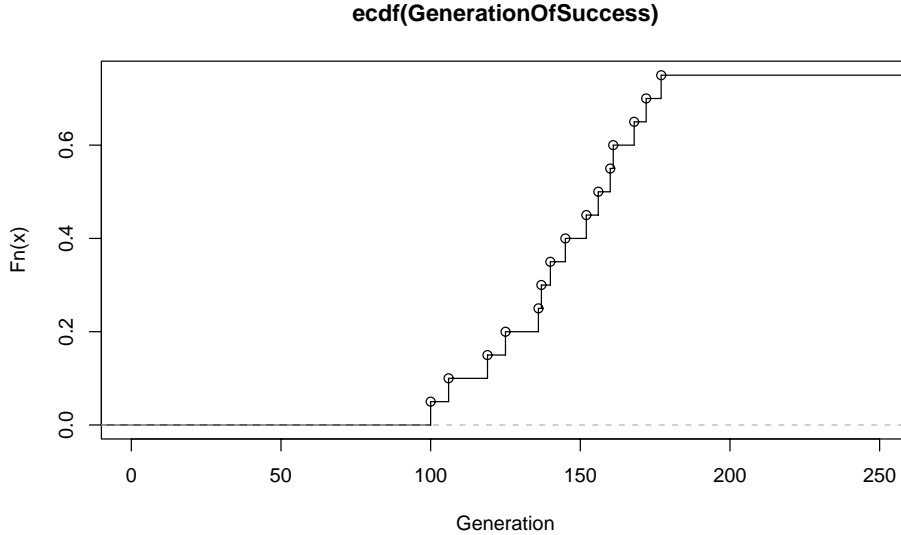


Figure 3: Success Statistics

Figure 3 shows the empirical cumulative distribution function which assigns to every generation the probability of success. We can see that the “fine tuning” — i.e. the discovery of an individual with fitness 34 — takes place between generation 100 and 180.

It has been determined that our custom-built GA operates more efficiently and produces better success rates than off-the-shelf GA software products, which cannot be adjusted easily to perform well on the CSP.

## References

- [1] Thomas Bäck, David B. Fogel, and Zbigniew Michalewicz, editors. *Evolutionary Computation 1 - Basic Algorithms and Operators*. Institute of Physics Publishing, Bristol, UK, 2000.
- [2] Wolfgang Banzhaf, Peter Nordin, Robert E. Keller, and Frank D. Francone. *Genetic Programming - An Introduction: On the Automatic Evolution of Computer Programs and its Applications*. Morgan Kaufmann Publishers, Inc., San Francisco, CA, 1998.
- [3] A. Depicker and M. V. Montagu. Post-transcriptional gene silencing in plants. *Current Opinion in Cell Biology*, 9:373–382, 1997.

- [4] S. W. Ding. RNA silencing. *Current Opinion in Biotechnology*, 11:152–156, 2000.
- [5] A. Fire. RNA-triggered gene silencing. *Trends in Genetics*, 15:358–363, 1999.
- [6] M. Frances and A. Litman. On covering problems of codes. *Theory of Computing Systems*, 30:113–119, 1997.
- [7] D. Goldberg. *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley, Reading, MA, 1989.
- [8] D. B. Golemboski, G. P. Lomonosoff, and M. Zaitlin. Plants transformed with a tobacco mosaic virus nonstructural gene sequence are resistant to the virus. *Proceedings of the National Academy of Sciences USA*, 87:6311–6315, 1990.
- [9] John H. Holland. *Adaptation in Natural and Artificial Systems*. University of Michigan Press, Ann Arbor, MI, 1975.
- [10] F. J. Jan, C. Fagoaga, S. Z. Pang, and D. Gonsalves. A single chimeric transgene derived from two distinct viruses confers multi-virus resistance in transgenic plants through homology-dependent gene silencing. *Journal of General Virology*, 81:2103–2109, 2000.
- [11] M. Prins and R. Goldbach. RNA-mediated virus resistance in transgenic plants. *Archives of Virology*, 141:2259–2276, 1996.
- [12] C. J. S. Smith, C. F. Watson, C. R. Bird, J. Ray, W. Schuch, and D. Grierson. Expression of a truncated tomato polygalacturonase gene inhibits expression of the endogenous gene in plants. *Molecular and General Genetics*, 224:477–481, 1990.
- [13] P. Tennant, G. Fermin, M. M. Fitch, R. M. Manshardt, J. L. Slightom, and D. Gonsalves. Papaya ringspot virus resistance of transgenic Rainbow and SunUp is affected by gene dosage, plant development, and coat protein homology. *European Journal of Plant Pathology*, 107:645–653, 2001.
- [14] A. R. Van der Krol, L. A. Mur, M. Beld, J. M. N. Mol, and A. R. Stuitje. Flavonoid genes in petunia: Addition of a limited number of gene copies may lead to a suppression of gene expression. *Plant Cell*, 2:291–299, 1990.
- [15] H. Vaucheret, C. Beclin, T. Elmayan, F. Feuerbach, C. Gordon, J. B. Morel, P. Mourrain, J. C. Palauqui, and S. Vernhettes. Transgene-induced gene silencing in plants. *Plant Journal*, 16:651–659, 1998.