Problems in evolutionary computation

Mattias Wahde

Contents

Foreword

This document contains exercises on some of the topics covered in the course *Artificial Intelligence 2: Biological methods*. However, the use of evolutionary algorithms usually involves computer programming, and is thus not so well suited to pen-and-paper exercises. Thus, the problems given here cover mostly the basics of genetic algorithms.

Programming-oriented problems will mostly be given in the assignments, even though a few such problems are presented below. Those problems which require computer programming are marked with a [C]. Answers to some of the exercises are given at the end of the document.

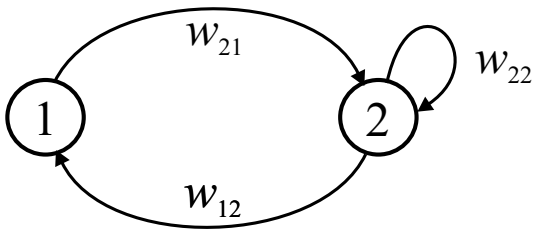

Figure E1: Genetic regulatory network for problem 1.2.

1. Biological basis of evolutionary algorithms

1.1 Consider a population of simple creatures, with a single chromosome consisting of $n = 1,000$ base pairs. Each entry in the chromosome can take four values (A, C, G, or T). Assume that the population size is equal to M.
a) How many possible chromosomes are there (express you answer in the form $c_1 10^{c_2}$?
b) Assuming that the chromosome length and the population size remain constant, what is the upper limit of the number of different chromosomes evaluated in the course of G generations?
c) If the population size is constant and equal to 10^{12}, how large a fraction q of the total number of chromosomes will be evaluated during 10^9 generations, assuming that all evaluated chromosomes are different?

1.2 In biological systems, some genes regulate the activity of other genes. A simplified model for the dynamics of genetic regulatory networks is given by

$$\tau_i \frac{\mathrm{d}x_i}{\mathrm{dt}} + x_i = \sigma\left(\sum_{j=1}^{n} w_{ij} x_j + b_i\right), \ i = 1, \ldots, n, \tag{E1}$$

where x_i is the activity of gene i, τ_i are time constants, and n is the number of genes in the network. The weights w_{ij} determine the influence of gene j on gene i, and the b_i are bias terms. The sigmoid function can be chosen as

$$\sigma(z) = \frac{1}{1 + \mathrm{e}^{-cz}}, \tag{E2}$$

where c is a constant. Consider now a network of two genes, as shown in Fig. E1. In this network, the only non-zero weights are w_{12}, w_{21}, and w_{22}. The two bias terms b_1 and b_2 are both equal to zero, and c is equal to 1.
a) Assuming that the self-inhibitory weight of the second gene (w_{22}) is equal to -1, and that the network reaches a fixed-point $x_1^{\mathrm{F}} = 0.1, x_2^{\mathrm{F}} = 0.5$ after a long time, determine the values of w_{12} and w_{21}.

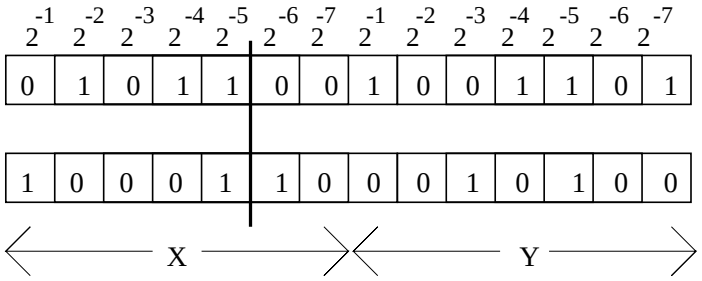

Figure E2: Chromosomes for two–dimensional function optimization (Problem 2.1).

b) Assuming that $\tau_1 = 1$ and $\tau_2 = 3$, sketch the dynamics for the first ten time units, starting from $x_1 = x_2 = 0$. What is the maximum value attained by x_1?

2. Basics of evolutionary algorithms

2.1 A genetic algorithm is used in order to find the optimum of the function $f(x, y) = \mathrm{e}^{-(x-0.5)^2-(y-0.5)^2}$. The chromosomes encode the values of x and y using 7–digit binary encoding, without rescaling, as shown in Fig. E2. The fitness is taken equal to the value of the function $f(x, y)$.
a) Decode the two chromosomes shown in of Fig. E2. What are the corresponding values of x and y, and what will be the fitness values for the two individuals?
b) Perform crossover between the two chromosomes considered in **a)**. Assume that the crossover point is chosen as shown in of Fig. E2. Again, decode the two resulting chromosomes and determine the values of x and y, as well as the fitness values.

2.2 **[C]** Write a "minimal" GA in which the selection operator simply picks two individuals at random, and where the reproduction operator is such that, with probability $p \approx 0.9$, the individual with the higher fitness is retained, and the other individual is replaced by a mutated copy (mutation rate: p_{mut}) of it, and vice versa with probability $1 - p$. (Before any selection is made, all individuals are evaluated once).

2.3 **[C]** Apply the simple GA in problem 2.2 to the problem of finding the minimum of the function $g(x, y) = 20 + \mathrm{e} - 20\mathrm{e}^{-0.1(x^2+y^2)} - \mathrm{e}^{\frac{1}{2}(\cos 2\pi x + \cos 2\pi y)}$. For the variables x and y, use 10-digit binary encoding where the first gene determines the sign of the variable (e.g. 0=positive, 1=negative),

and the remaining nine genes determine the absolute value of the variable, in the range $[0, 2[$. Use $f = 1/(1 + g(x, y))$ as the fitness measure. With $N = 30$ and $p_{\text{mut}} = 0.05$, how many individuals must be evaluated (average of 20 runs, say), in order to achieve a fitness greater than 0.9999?

2.4 Consider a case where an individual is to be chosen from a population with four individuals with fitness values $f_1 = 1, f_2 = 2, f_3 = 3$, and $f_4 = 4$. What is the probability that individual 2 (with fitness equal to 2) will be selected, if the selection is performed using
a) Roulette-wheel selection?
b) Tournament selection (with tournament size equal to 2, and the probability of selecting the best individual equal to 0.75)?

2.5 Consider a population consisting of five individuals with the fitness values (before ranking) $f_1 = 5, f_2 = 7, f_3 = 8, f_4 = 10$, and $f_5 = 15$. Compute the probability that individual 4 will be selected (in a single selection step) with (1) roulette wheel selection, (2) tournament selection, with tournament size equal to 2, and the probability of selecting the best individual (in a given tournament) equal to 0.75, (3) roulette wheel selection, based on linearly ranked fitness values, where the lowest fitness value is set to 1 and the highest fitness value set to 10.

3. Using evolutionary algorithms

3.1 Based on the Matlab program in Handout 3, write Matlab code for
a) Decoding m variables from a chromosome with n binary genes,
b) Decoding n variables from a chromosome with n real-valued genes,
c) Roulette-wheel selection,
d) Linear fitness ranking (hint: use the `sort` command in Matlab),
e) Steady-state reproduction, in which the two worst individuals in the parent population are replaced by the two offspring,
f) Creep mutations (for the case of real-numbered encoding, as in **b)**).

3.2 **[C]** Using the Matlab code from Handout 3, and the results from 3.1, write a standard GA as described in Handout 2.

3.3 **[C]** Apply the standard GA from problem 3.2 to the problem of finding the maximum of the benchmark function $\psi_5(x_1, \ldots, x_5)$ (see Handout 3), using various different parameter settings. For each parameter setting, make 10 runs, and list your results in a table similar to Tables 3.2 and 3.3 (see Handout 3).

3.4 **[C]** In his study of parameter selection for GAs, de Jong (See Handout 3), found that a population size of 200 gave better average performance than population sizes of 50 or 100 for the minimization of

$$f_1(x_1, x_2, x_3) = \sum_{i=1}^{3} x_i^2, \quad x_i \in [-5.12, 5.12], \tag{E3}$$

using a binary encoding scheme. Repeat de Jong's experiment for population sizes 10, 30, 50, 100, 200, 500, using 10 genes per variable, a crossover rate p_c of 1.0, and a mutation rate of $p_{\text{mut}} = 0.01$. What (if any) *statistically significant* conclusions can be drawn from the experiment?

3.5 **[C] a)** Write a standard GA using 10-digit decimal encoding for the variables x and y, and find the maximum of the function $g(x, y) = (\sin xy + \cos(x + \sqrt{3}y))/\sqrt{1 + x^2 + y^2}$ on the interval $x, y \in [-4, 4]$. Use a population size of $N = 30$. How does the performance of the GA vary with the values of the crossover probability and the mutation probability? (Note: Many runs are needed for each parameter setting in order to get a reliable average).
b) Use a population size of $N = 10$, and plot the locations in the xy–plane of the entire population at generations 1,2,5,10,20, and 50 for a single run.

4. Properties of evolutionary algorithms

4.1 A GA is used for finding the maximum of the (very simple) function $f(x, y) = x^2 + y^2$, in the interval $(x, y) \in [0, 0.9375]$. The fitness measure is taken simply as $f(x, y)$, without any rescaling or ranking. Assume that a binary encoding scheme is used, with 4 genes for the variable x (the four first genes) and four genes for the variable y. During decoding, the first gene of the variable x is multiplied by 2^{-1}, the second by 2^{-2} etc. The variable y is obtained in a similar way. In the formation of new chromosomes, the crossover probability $p_c = 0.10$ is used, and the mutation rate is set to 0.01. In generation g, the population consists of six individuals with chromosomes 10101101, 01100111, 01110101, 01011001, 10010001, and 10001001. Use the schema theorem to estimate the number of copies of the schema $S_1 = 1\text{xxx}1\text{xxx}$ in generation $g + 1$.

4.2 As in problem 4.1, a GA is used for finding the maximum of a function, namely $f(x, y) = 1 + xy + x^2 - y^2$, in the interval $(x, y) \in [0, 0.9375]$. The fitness measure is taken simply as $f(x, y)$, without any rescaling or ranking. Assume that a binary encoding scheme is used, with 5 genes for the

variable x (the five first genes) and five genes for the variable y. During decoding, the first gene of the variable x is multiplied by 2^{-1}, the second by 2^{-2} etc. The variable y is obtained in a similar way. In the formation of new chromosomes, the crossover probability $p_c = 0.50$ is used, and the mutation rate is set to 0.01. In generation g, the population consists of six individuals with chromosomes 1111101101, 0001100111, 0001110101, 1101110101, 0100110001, and 010011111. Use the schema theorem to estimate the number of copies of the schema $S_1 =$ 010xxxxxxx in generation $g + 1$.

4.3 Consider a population containing four individuals with chromosomes 101010, 000111, 010101, and 011011, and fitness values 1,2,3, and 4, respectively. In a given selection step, assume that individual 1 (with chromosome 101010) has been selected (using roulette-wheel selection) as the first parent. What is the probability that the schema 10xxxx will be represented in either of the two individuals resulting from the selection of a second parent, followed by crossover? (Crossover may occur, with equal probability, at any of the five available crossover points).

4.4 Consider a chromosome with n binary-valued genes, and assume that it is to be mutated using a mutation rate p. Determine the probability that the chromsome will undergo
a) No mutation.
b) Exactly one mutation.
c) Less than three mutations.

4.5 For the chromosome in problem 4.4, show that the average number of mutations equals np, if the mutation rate is equal to p.

4.6 For the chromosome in problem 4.4, derive an equation for the maximum mutation rate p, if it is required that the probability of the chromosome undergoing more than one mutation should be less than ϵ. Next, determine the numerical value for p in the case $n = 100$, $\epsilon = 0.001$.

4.7 A simple genetic algorithm, using only roulette-wheel selection, is applied to the counting-ones problem defined in Handout 4, using random initialization of the (infinite) population.

a) Compute analytically the probability distribution in generation 3 (after two selection steps), and show that the average fitness equals

$$\overline{f}_3 = \frac{n(n+3)}{2(1+n)}. \tag{E4}$$

b) Find the probability distribution for generation 4, and show that the average fitness equals

$$\overline{f}_4 = \frac{(n+1)(n^2+5n-2)}{2n(3+n)}. \tag{E5}$$

4.8 Consider again the counting-ones problem, but assume now that the population has been initialized in such a way that

$$p_1(k) = \frac{1}{n+1}, \tag{E6}$$

where n again denotes the chromosome length. Compute the average fitness value in the initial population, as well as the average fitness value after the first selection step (assuming that crossover and mutation do not occur).

4.9 For a simple genetic algorithm, using only roulette-wheel selection, applied to the counting-ones problem with $n = 50$, how large must the population be in order for the probability of finding an individual with at least 48 ones to reach approximately 10^{-5} in generation 3, i.e. after two selection steps, assuming that the probability distributions derived for infinite population size can be used also in a case where the population size is finite (and large)?

4.10 **a)** Using Eqs. (3.18) and (3.19) in Handout 3, compute the equilibrium point for the average fitness for a simple genetic algorithm (i.e one that uses only selection and single-gene mutations with $p_{\mathrm{m}} = 1$) applied to the counting-ones problem with $n = 200$. Assume an infinite population with random initialization, and that the distribution retains its shape (but is shifted towards higher average values) in the generations following initialization.

b) Write a computer program that implements the simple genetic algorithm as described in **a)**, and find the equilibrium point numerically.

4.11 Consider a case where a function optimization problem is to be solved using a GA with binary encoding and chromosome length $n = 4$. Thus, each chromosome consists of the four genes g_1, g_2, g_3, and g_4, and there are 16 different chromosomes. Assume that the fitness function is given by

$$f(g_1, g_2, g_3, g_4) = 3 + g_1 + g_2 - g_3 - g_4, \tag{E7}$$

so that the maximum fitness (=5) is obtained for the chromosome 1100. Furthermore, assume that the initial distribution of chromosomes is uniform, i.e.

$$p_1(g_1, g_2, g_3, g_4) = \frac{1}{2^n} = \frac{1}{16}, \tag{E8}$$

for all chromosomes.
a) Assuming that standard roulette-wheel selection is performed, determine the distribution $p_2(g_1, g_2, g_3, g_4)$, in the absence of mutations.
b) Again using standard roulette-wheel selection, determine $p_2(g_1, g_2, g_3, g_4)$ if the mutation rate is equal to $1/4$. Note that each chromosome may undergo anything from zero to four mutations.

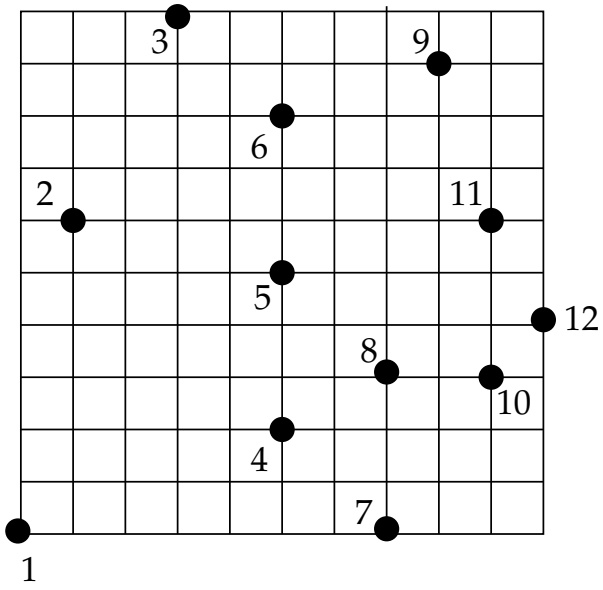

Figure E3: Map for Problem 5.4. The cities are numbered from 1 to 12, and their locations are given by two integer coordinates (x, y).

5. Advanced topics

5.1 Show that the Gray code is *not* unique for $n = 4$.

5.2 Write computer code (e.g using Matlab) for a representation that uses Gray coding for arbitrary chromosome lengths. Given an arbitrary chromosome, the decoding procedure should give the value of v variables, encoded by m bits each (n = m*v).

5.3 Consider the TSP defined in handout 5. Assuming that permutation encoding is used, so that a valid N-node path is encoded as some permutation of the numbers $1, \ldots, N$, e.g. $\{1, 4, 5, 3, 2\}$ in the case $N = 5$.
a) Define a mutation operator for the TSP that maps valid chromosomes (i.e. paths) onto other valid chromosomes.
b) Define a crossover operator for the TSP that maps valid chromosomes onto other valid chromosomes.

5.4 **[C]** Using the crossover and mutation operators from Problem 5.3, write an EA that solves the TSP, and apply it to the 12-city problem illustrated in Fig. E3. The location of each city in the figure is given by two integer coordinates. For example, city 4 is located at $(x, y) = (5, 2)$. What is the shortest path (and how long is it?).

5.5 Genetic algorithms can be used for optimizing polynomials, e.g. for function fitting. If the degree d of the desired polynomial is unknown, chromosomes of varying length can be used. Alternatively, one may simply set d to a high value. The latter approach has a significant drawback: The number of terms grows very rapidly with d, making the search space very large. For a polynomial u of n variables and degree d, use the notation

$$\begin{aligned} u(x_1, x_2, \ldots x_n) &= a_{00\cdots0} + a_{10\cdots0}x_1 + a_{01\cdots0}x_2 + \\ &+ \ldots + a_{110\cdots0}x_1x_2 + \ldots + a_{00\cdots d}x_n^d. \end{aligned} \tag{E9}$$

Thus, as a specific example, a polynomial of two variables and degree $d = 2$ would be given as

$$u(x_1, x_2) = a_{00} + a_{10}x_1 + a_{01}x_2 + a_{20}x_1^2 + a_{11}x_1x_2 + a_{02}x_2^2. \tag{E10}$$

Thus, in this case, there would be 6 parameters to determine. How many parameters would there be in a polynomial with
a) 3 variables and degree 2,
b) 4 variables and degree 3,
c) n variables and degree d. (Difficult).

6. Other EAs

6.1 Consider an LGP implementation with three calculation registers r_j and one constant register c_1, taking the value 1. Let A denote the union of the set of calculation registers and the set of constant registers, i.e. $A = \{r_1, r_2, r_3, c_1\}$. The elements of this set are denoted $A_j,\ j = 1, \ldots, 4$. The structure of the instructions in this LGP implementation is

$$r_i := A_j\ \mathrm{OP}_k\ A_m, \tag{E11}$$

where OP_k encodes an operator such that $\mathrm{OP}_1 = +$, $\mathrm{OP}_2 = -$, $\mathrm{OP}_3 = \times$, $\mathrm{OP}_4 = /$. Each instruction can thus be encoded by the four integers i, j, k, m, where $i \in \{1, 3\}$, $j, k, m \in \{1, 4\}$. Consider the chromosome $C = 11142111323212331243314221 33$.
a) Assuming that the calculation registers all contain the value 0 initially, what will be the values contained in these registers at the end of the evaluation of C.
b) Can you find a faster way (i.e. one represented by a shorter chromosome, using the same representation as above) of achieving the same result, i.e. the same final values in the calculation registers r_j?

Answers to selected exercises

1.1 a) $N_{\text{possible}} = 4^{1000} = 1.148 \times 10^{602}$
b) $N_{\text{evaluated}} = M \times G$. This is an upper limit, since it is likely that some individuals have identical chromosomes.
c) $q = \frac{N_{\text{evaluted}}}{N_{\text{possible}}} = \frac{10^{21}}{1.14810^{602}} = 8.7110^{-582}$.
1.2 a) $w_{12} = -2\ln 9, w_{21} = 5$.
2.1 $x_1 = 0.34375$, $y_1 = 0.601563$,$f_1 = 0.965867$, $x_2 = 0.546875$, $y_2 = 0.15625$, $f_2 = 0.8866$.
b) $x_1 = 0.359375$, $y_1 = 0.15625$, $f_1 = 0.871511$, $x_2 = 0.53125$, $y_2 = 0.601563$, $f_2 = 0.988772$.
2.3 The global minimum of $g(x, y)$ is equal to 0, corresponding to fitness $f = 1$.
2.5 (1) Roulette wheel selection gives $p_4 = \frac{2}{9} \approx 0.222$. (2) Tournament selection gives $p_4 = \frac{6}{25} = 0.240$. (3) Roulette wheel selection with fitness ranking gives $p_4 \approx 0.276$.
4.1 $D(S_1) = 4, \quad O(S_1) = 2, \quad \Gamma(S_1, g) = 2$
$\bar{f}(S_1) = 0.80859, \quad \bar{f} = 0.49544 \Rightarrow \Gamma(S_1, g + 1) \approx 3$
4.3 The probability of finding the schema 10xxxx in either of the two individuals equals $43/50 = 0.86$.
4.4a) p(no mutations) = $(1 - p)^n$.
b) p(one mutation) = $np(1 - p)^{n-1}$.
c) p(less than 3 mutations) = $(1 - p)^n + np(1 - p)^n + \frac{n(n-1)}{2}p^2(1 - p)^{n-2}$.
4.7a) The probability distribution in generation 3 is given by

$$p_3(k) = 2^{2-n}\frac{k^2}{n(n+1)}\binom{n}{k}. \tag{E12}$$

b) The probability distribution in generation 4 is given by

$$p_4(k) = 2^{3-n}\frac{k^3}{n^2(3+n)}\binom{n}{k}. \tag{E13}$$

4.9 A population size of around 2.44×10^6 is needed.
4.10a) The computed average fitness is around 136.
b) The average from the numerical simulation will be around 125-130.

5.3a) The mutation operator can be chosen such that it selects a gene at random, and swaps it with another randomly chosen gene. For example, if the second and fourth genes are chosen, the mutation operator would change the seven–city path $(5, 3, 7, 1, 4, 2, 6)$ to $(5, 1, 7, 3, 4, 2, 6)$.
5.3b) One possibility (there are others) is so called **order crossover**, in which substrings are exchanged between the two parents while keeping the order among those cities which are *not* part of the substring. For example, consider the two parents chromosomes $(7, 1, 2, 4, 6, 3, 5)$ and $(2, 3, 5, 1, 4, 7, 6)$. *Two* crossover points are chosen randomly, for instance between genes 3 and 4 and between genes 6 and 7. The corresponding substrings are $4, 6, 3$ and $1, 4, 7$. These are inserted in the (initially empty) offspring chromosomes which then take the form $(*, *, *, 1, 4, 7, *)$ and $(*, *, *, 4, 6, 3, *)$, where the $*$ indicates that the element is, as yet, unknown. The first offspring will now have its empty slots filled in using the remaining string of the first parent. Thus, the cities 1,4, and 7 are removed from the first string resulting in a substring $2, 6, 3, 5$. These elements are now inserted, in order, in the chromosome of the first offspring, starting from the first empty slot after the substring $1, 4, 7$. The resulting chromosome is $(6, 3, 5, 1, 4, 7, 2)$. Similarly, the second offspring takes the form $(5, 1, 7, 4, 6, 3, 2)$. Note that e.g. the strings $(1, 2, 3)$ and $(2, 3, 1)$ are identical. It is only the order of the elements that matter, not their absolute position.
5.4 The shortest path is given by $(1, 2, 3, 6, 9, 11, 12, 10, 8, 5, 4, 7)$. The length of this path is 41.015 units.
5.5 a) 10, **b)** 35, **c)** $\binom{n+d}{n} \equiv \binom{n+d}{d}$.
6.1 a) The final values are $r_1 = 1/2$, $r_2 = 1/8$, and $r_3 = 1/4$.