The University of Vermont

CS352 Evolutionary Computation: Homework 2

Ayat Ospanov

September 26, 2018

Contents

1 Question 1 **1**

2 Question 2 **2**

3 Question 3 **2**

4 Question 4 **2**

5 Question 5 **3**

6 Question 6 **3**

7 Question 7 **6**

8 Question 8 **7**

1 Question 1

The equation for the basic Schema Theorem is the next:

$$E[m(H,t+1)] \geq m(H,t)\frac{f(H)}{\bar{f}}\Big(1-\frac{\delta(H)}{L-1}p_c - o(H)p_m\Big)$$

where

- $m(H,t)$ is the proportion of individuals representing schema H at time-step t
- $f(H)$ is the fitness of the schema H
- $\bar{f}$ is the mean fitness of the population
- $o(H)$ is the order of the schema H
- $\delta(H)$ is the defining length of the schema H
- L is the length of a genotype
- p_c is the probability of applying crossover
- p_m is the bitwise mutation probability

The first factor $m(H,t)\frac{f(H)}{f}$ is the probability of a schema being selected. It is obvious that it depends on the relative fitness of the schema (because the selection is fitness proportionate) and on the proportion of individuals.

The second factor $1 - \frac{\delta(H)}{L-1}p_c - o(H)p_m$ is the probability of surviving variation operators. $\frac{\delta(H)}{L-1}p_c$ is the probability of disrupting examples by crossover and $o(H)p_m$ is the probability of disrupting examples by mutation.

Thus, the theorem states "short, low-order (derives from the second factor) schemata with above-average fitness (the first factor) increase exponentially over generations"

2 Question 2

The fundamental message of the Schema Theorem, or Building Block Hypothesis, is that GAs can begin by selecting short, low-order schemata examples, and then combine them to create higher order schemata, repeating until a schema of length $L-1$ and order L is created and selected.

3 Question 3

There are 3 types of problem "difficulty".

- intra-BB
- inter-BB
- extra-BB

The most important is an intra-BB difficulty, or deceptiveness. This means at least one optimal schema is outcompeted by non-optimal schema.

4 Question 4

$$S1 = \texttt{*0**11***0**}, \qquad o(S1) = 4, \delta(S1) = 8, L = 12$$
$$S2 = \texttt{*****0*1****}, \qquad o(S1) = 2, \delta(S1) = 2, L = 12$$

a) The probability p_s of surviving both crossover and mutation is

$$p_s(H) = (1-p_m)^{o(H)}\Big(1 - p_c\frac{\delta(H)}{L-1}\Big)$$

Thus,

$$p_s(S1) = (1-p_m)^4(1-\frac{8p_c}{11})$$
$$p_s(S2) = (1-p_m)^2(1-\frac{2p_c}{11})$$

It is obvious, that $p_s(S1) \leq p_s(S2)$. And the probability of surviving of both of them is $p_s(S1) \cdot p_s(S2)$

b) As we know, "building blocks" are short (in terms of defining length) and low-order schematas. $S1$ cannot be called a "building block" as its order and length are high, while $S2$ can be called a "building block", because it has the length of 2 and the order of 2 which makes possible to describe $S2$ as short and low-order schemata.

5 Question 5

string	fitness
100	10
111	20
011	15
010	15

a)

$$f(1**) = \frac{N_{100}f_{100} + N_{111}f(111)}{N_{100} + N_{111}} = \\ = \frac{25 * 10 + 25 * 20}{25 + 25} = \frac{10 + 20}{2} = 15$$

b) The estimated number of samples of a schemata H at the next step is:

$$n(H, t+1) = n(H, t)\frac{f(H)}{\bar{f}}$$

Thus, for the schemata `1**` we can get the estimated number of samples on the next step:

$$n(\texttt{1**}, t+1) = 50 * \frac{15}{15} = 50$$

But we need the estimated fitness. To count, we need to estimate the number of survivors of guys from the schema `1**`. As we know those guys and can estimate numbers for a scheme, we can put $H_1 = 100$ and $H_2 = 111$. Now let's estimate the numbers:

$$n(100, t+1) = n(100, t)\frac{f(100)}{\bar{f}} = 25 \cdot \frac{10}{15} = 25 \cdot \frac{2}{3}$$
$$n(111, t+1) = n(111, t)\frac{f(111)}{\bar{f}} = 25 \cdot \frac{20}{15} = 25 \cdot \frac{4}{3}$$

Now we can estimate the fitness value for the `1**` scheme.

$$f_{t+1}(\texttt{1**}) = \frac{n(100, t+1) * f(100) + n(111, t+1) * f(111)}{n(100, t+1) + n(111, t+1)} = \\ = \frac{25 \cdot \frac{2}{3} \cdot 10 + 25 \cdot \frac{4}{3} \cdot 20}{50} = \frac{\frac{2}{3} \cdot 10 + \frac{4}{3} \cdot 20}{2} = \\ = \frac{10 + 2 \cdot 20}{3} = \frac{100}{6} \approx 16.7$$

6 Question 6

phenotype (integer)		0	1	2	3	4	5	6	7
genotype	binary	000	001	010	011	100	101	110	111
	gray	000	001	011	010	110	111	101	100
fitness		7	5	3	9	10	1	6	6

Table 1: Schema Analysis for binary code and gray code

	order	3	2			1			0
binary	schema	111	11*	1*1	*11	**1	*1*	1**	***
	fitness	**6**	**6**	3.5	7.5	5.25	**6**	5.75	**5.875**
	schema		01*	0*1	*01	**0	*0*	0**	
	fitness		**6**	7	3	**6.5**	5.75	**6**	
	schema		10*	1*0	*10				
	fitness		5.5	**8**	4.5				
	schema		00*	0*0	*00				
	fitness		**6**	5	**8.5**				
gray	schema	100	10*	1*0	*10	0**	*0*	**0	***
	fitness	**6**	**6**	**8**	**9.5**	**6**	**6**	**8**	**5.875**
	schema		11*	1*1	*11	1**	*1*	**1	
	fitness		5.5	3.5	2	5.75	5.75	3.75	
	schema		01*	0*1	*01				
	fitness		**6**	4	5.5				
	schema		00*	0*0	*00				
	fitness		**6**	**8**	6.5				

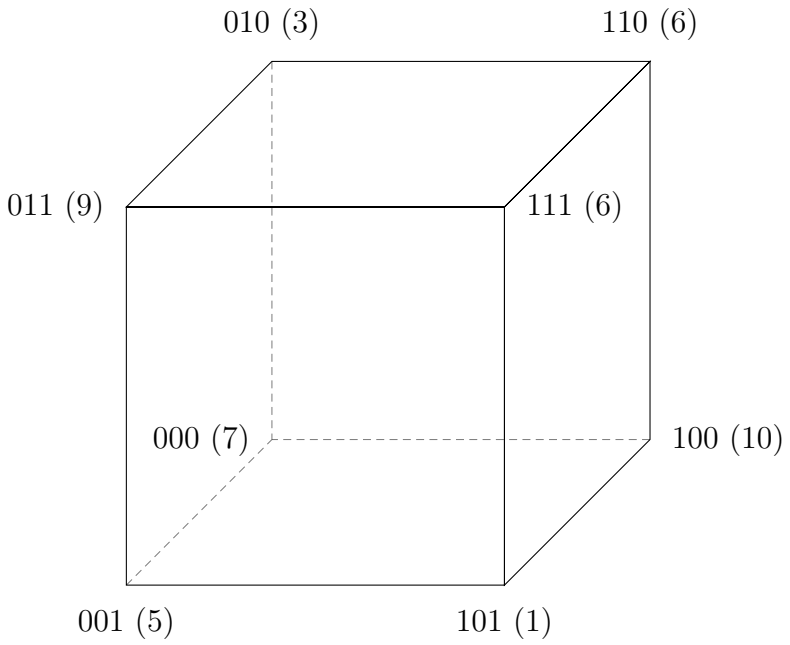

Figure 1: Hamming cube diagram for binary encoding

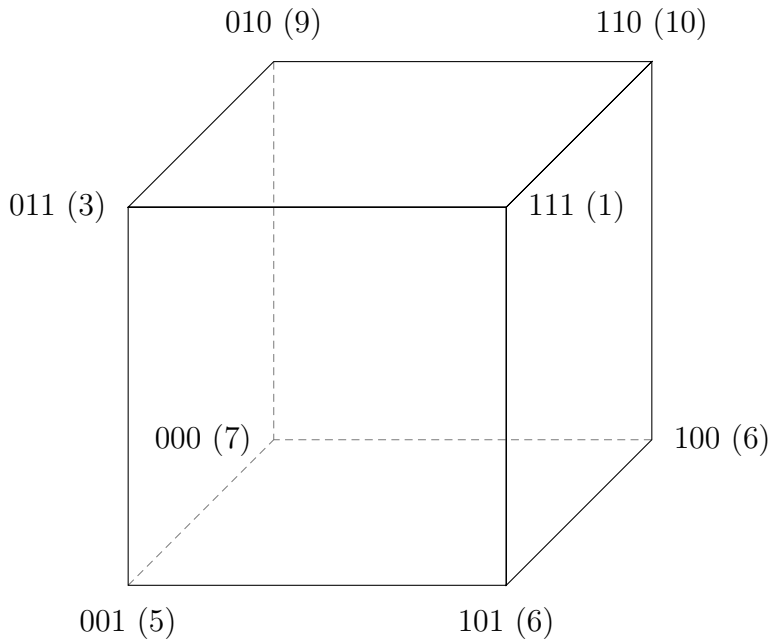

Figure 2: Hamming cube diagram for gray encoding

b) The problem is partially deceptive for both of encodings. For binary encoding the optimal schema 10* ($f(\texttt{10*}) = 5.5$) was outcompeted by non-optimal (e.g. $f(\texttt{11*}) = 6$), while the optimal schema 1*0 ($f(\texttt{1*0}) = 8$) outcompetes others. Thus, the problem for binary encoding is partially deceptive. For gray encoding the fitness value of the optimal schema 1*0 ($f(\texttt{1*0}) = 8$) is equal to the fitness value of the non-optimal schema 0*0 ($f(\texttt{0*0}) = 8$), which makes the problem partially deceptive.

Now, to determine a type of deceptiveness, let's look at the Hamming diagrams. For binary encoding, we have maximum distance between suboptima ($f(\texttt{011}) = 9$) and the optima ($f(\texttt{100}) = 10$). That is binary encoded problem is Type II deceptive. Gray encoding has the minimum distance between sub-optima and optima. Thus, the problem in this case is Type I deceptive.

c) They are not. If we look at the columns for order 1 on the Table 1, we can compute saliences for schemas as difference between corresponding rows. As in both encoding they have different differences, they are not uniformly scaled.

d) In case of non-uniformly scaled saliences, there is no selective pressure on lower order bits until the highest order bit gets fixed. Thus, it kills implicit parallelism, which makes it converge longer and it has so called domino convergence, when you drift from one suboptima to another step-by-step.

e) In terms of convergence, the gray encoding makes the problem easier as it becomes Type I deceptive. As we know, Type I deceptive problems are not GA-hard. On the other hand, Type II problem may converge faster, but it is not guaranteed it converges to the global optima. The convergence depends on parameters such as initial proportions of examples in population, etc.

f) The way to make an encoding to make the problem fully easy is arranging the fitness values ascending and assigning them to the Hamming cube's corners with the step of 1 in terms of the hamming distance. My custom encoding is given in the Table 2 and Fig. 3.

Table 2: Custom genotype

phenotype	0	1	2	3	4	5	6	7
custom genotype	100	011	101	010	110	001	000	111
fitness	7	5	3	9	10	1	6	6

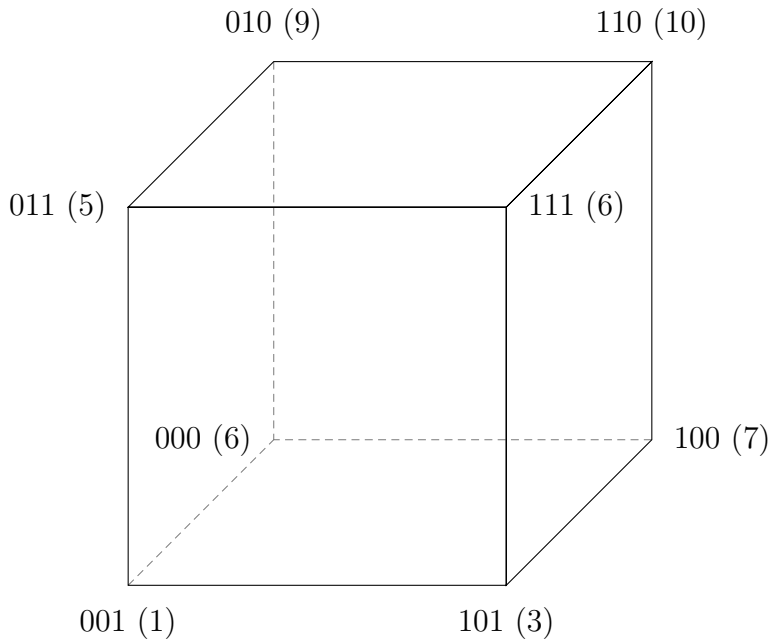

Figure 3: Hamming cube diagram for custom encoding

The proof that the problem is fully easy is given in the Table 3. Optimal schemas outcompete all non-optimal schemas. We have achieved it by constructing the encoding with no valleys.

Table 3: Schema Analysis for custom code. Optimal schemas are underscored.

	order	3	2			1			0
custom	schema	110	$\underline{11*}$	1*1	*11	**1	$\underline{*1*}$	$\underline{1**}$	***
	fitness	**10**	**8**	4.5	5.5	3.75	**7.5**	**6.5**	**5.875**
	schema		01*	0*1	*01	$\underline{**0}$	*0*	0**	
	fitness		7	3	2	**8**	4.25	5.25	
	schema		10*	$\underline{1*0}$	$\underline{*10}$				
	fitness		5	**8.5**	**9.5**				
	schema		00*	0*0	*00				
	fitness		3.5	7.5	6.5				

7 Question 7

Let's assume we have done crossover and have $P_{ij}^{t+1}, i, j \in 0, 1$. Now we do mutation for P_{11}: the number of samples that go to the next step is the number of samples which "survive" mutation. It is $(1 - p'_m)P_{11}^{t+1}$, where $p'_m = 1 - (1 - p_m)^{o(H)}$ is the probability of destruction. Now calculate the constructive part. We can get P_{11} from P_{10} by flipping the second bit, from P_{01} by flipping the first bit and from P_{00} by flipping both bits. The probability of mutating only one bit is $p_m(1 - p_m)$. The equation means we flip one and don't flip the second. The probability of flipping both bits is p_m^2. Thus, we have the next equation for P_{11}^{t+1}:

$$\mathbb{P}_{11}^{t+1} = (1 - p_m)^2 P_{11}^{t+1} + p_m(1 - p_m)[P_{01}^{t+1} + P_{10}^{t+1}] + p_m^2 P_{00}^{t+1}$$

I put $\mathbb{P}$ to point out that this is the new value.

Applying the same logic for others, we get the next set of equations:

$$\begin{aligned}
\mathbb{P}_{00}^{t+1} &= (1-p_m)^2 P_{00}^{t+1} + p_m(1-p_m)[P_{10}^{t+1} + P_{01}^{t+1}] + p_m^2 P_{11}^{t+1} \\
\mathbb{P}_{01}^{t+1} &= (1-p_m)^2 P_{01}^{t+1} + p_m(1-p_m)[P_{00}^{t+1} + P_{11}^{t+1}] + p_m^2 P_{10}^{t+1} \\
\mathbb{P}_{10}^{t+1} &= (1-p_m)^2 P_{10}^{t+1} + p_m(1-p_m)[P_{11}^{t+1} + P_{00}^{t+1}] + p_m^2 P_{01}^{t+1} \\
\mathbb{P}_{11}^{t+1} &= (1-p_m)^2 P_{11}^{t+1} + p_m(1-p_m)[P_{01}^{t+1} + P_{10}^{t+1}] + p_m^2 P_{00}^{t+1}
\end{aligned}$$

To check the correctness of the system, we can sum the equations up and they should equal to 1:

$$\begin{aligned}
Sum &= \mathbb{P}_{00}^{t+1} + \mathbb{P}_{01}^{t+1} + \mathbb{P}_{10}^{t+1} + \mathbb{P}_{11}^{t+1} = \\
&= (1-p_m)^2[P_{00}^{t+1} + P_{01}^{t+1} + P_{10}^{t+1} + P_{11}^{t+1}] + \\
&+ 2p_m(1-p_m)[P_{00}^{t+1} + P_{01}^{t+1} + P_{10}^{t+1} + P_{11}^{t+1}] + \\
&+ p_m^2[P_{00}^{t+1} + P_{01}^{t+1} + P_{10}^{t+1} + P_{11}^{t+1}] = \\
&= \{\text{as } P_{00}^{t+1} + P_{01}^{t+1} + P_{10}^{t+1} + P_{11}^{t+1} = 1\} = \\
&= (1-p_m)^2 + 2p_m(1-p_m) + p_m^2 = \\
&= [(1-p_m) + p_m]^2 = 1
\end{aligned}$$

8 Question 8

b) First, let's search for a case, when the problem doesn't converge without mutation, but converges with. From theory, we know, that Type II problem diverges when the proportion of genotypes are not equal. Thus, we simulate type II problem (a = 0.5, b = 0.9, c = 1.1) with disbalance in genotype frequencies ([3 3; 3 1]) (shown on Fig. 4).

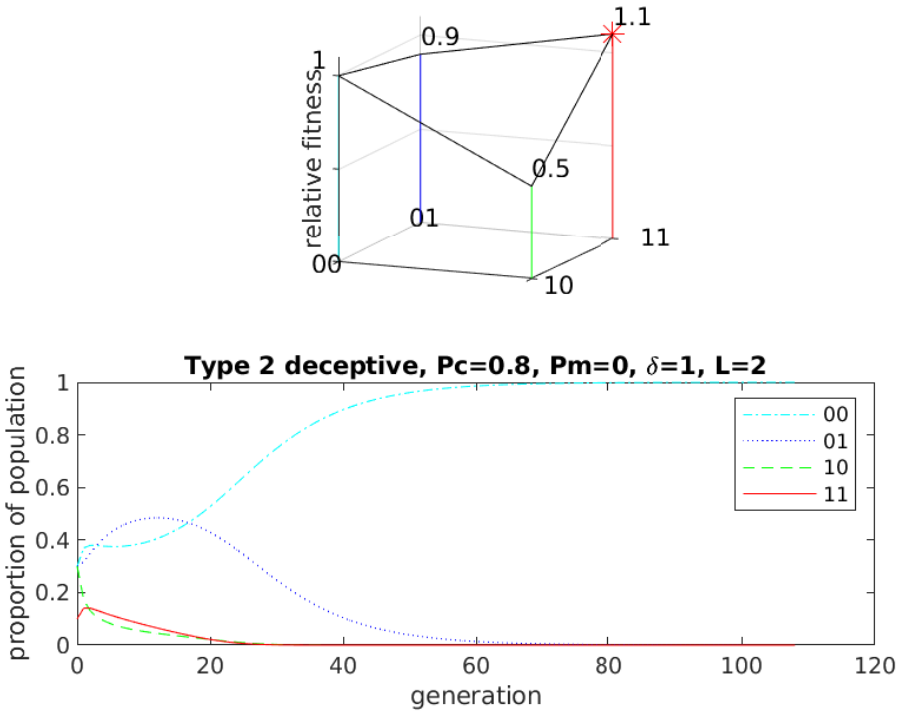

Figure 4: Type II problem without mutation (diverges)

But when we add mutation, the problem started converging (Fig. 5). I suppose it started converging, because mutation scatters genomes so that they try to balance by a transition to each other.

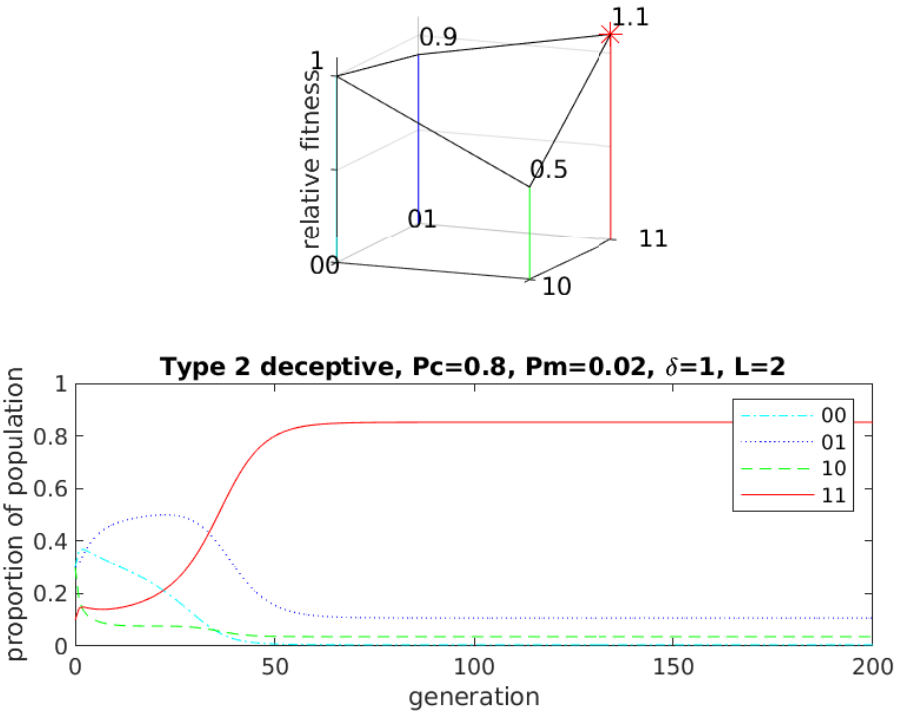

Figure 5: Type II problem with mutation (converges)

Now, let's search for a case, when the problem converges without mutation, but diverges with. It is a Type I problem.

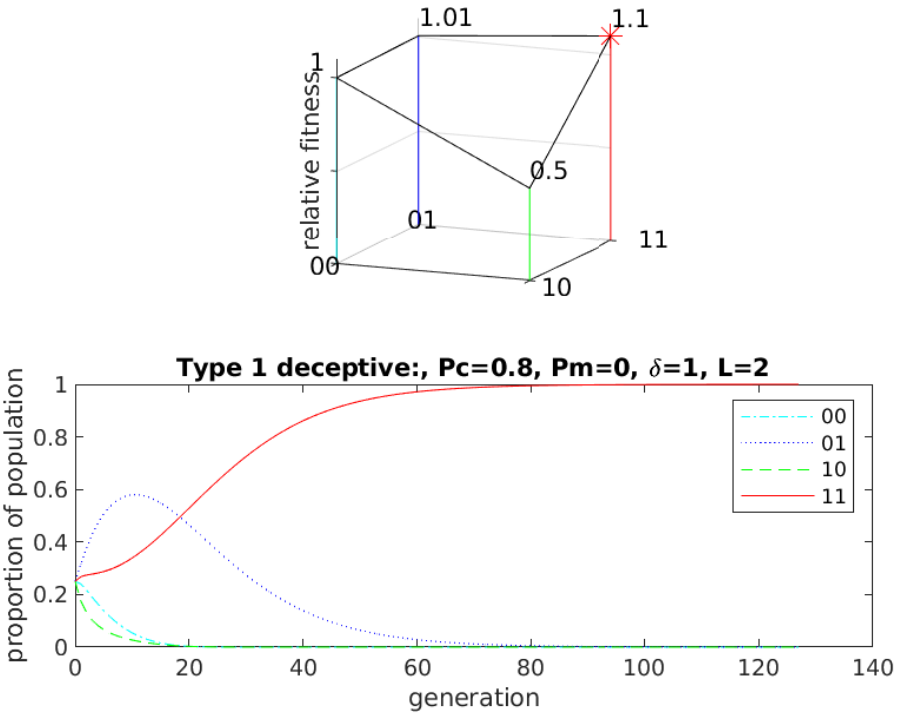

Figure 6: Type I problem without mutation (converges)

Parameters for the simulation are: a = 0.5, b = 1.01, c = 1.1, genome freqs = [1 1; 1 1]. If we run without mutation ($p_m = 0$), the problem converges (Fig. 6).

But if we run with mutation with probability of $p_m = 0.07$, the problem stops converging (Fig. 7). It will have constant proportion for sub-optimal and optimal values. I suppose this has happened, because as we shuffle samples by mutation of relatively high probability. Thus we are "pushing" genomes to both sub-optima and optima.

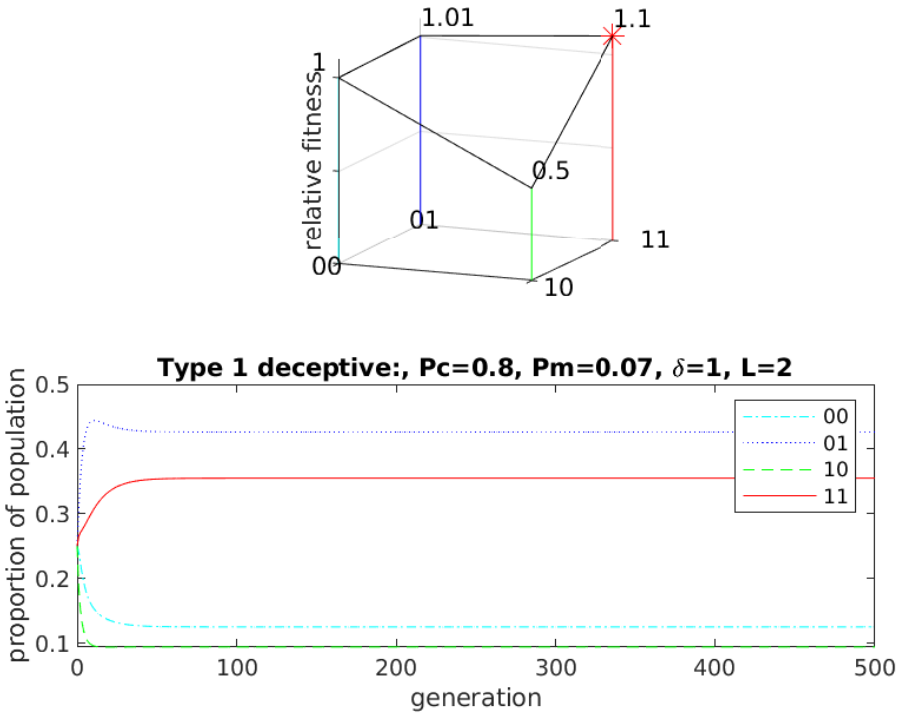

Figure 7: Type I problem with mutation (diverges)

c) In general, there is no common mutation rate selection procedure. But general suggestion is to put mutation is relatively low (< 0.1). For example, the problem of Type I discussed above (Fig. 7) converges until $p_m \approx 0.07$ and then diverges. But for the problem of Type II (Fig. 5), it diverges before $p_m \approx 0.01$ and after $p_m \approx 0.1$. So we have to choose experimentally, or theoretically the mutation rate for each type of problem we solve.