HyperMu'NmGA - ODD Protocol

Hypermutations in a NetLogo minimal genetic algorithm

Cosimo Leuci

The description of the model follows the ODD (Overview, Design concepts, Details) protocol for individual and agent-based models[1].

1. Purpose

A minimal genetic algorithm (mGA) was preliminarily developed to search for the solution of an elementary arithmetic problem[2]. It has been modified to explore the effect of a mutator gene and the consequent entrance into a hypermutation state. The phenomenon is particularly important in some types of tumorigenesis[3], [4] and in a more general way, in cells and tissues submitted to chronic sublethal environmental or genomic stress[5, 6].

Since a long time, some scholars suppose that organisms speed up their own evolution by varying mutation rate, but evolutionary biologists are not convinced that evolution can select a mechanism promoting more (often harmful) mutations looking forward an environmental challenge[7,8].

---

[1]V. Grimm, S, F. Railsback, C. E. Vincenot, U. Berger, C. Gallagher, D. L. DeAngelis, Bruce Edmonds, J. Ge, J. Giske, J. Groeneveld, A. S. A. Johnston, A. Milles, J. Nabe-Nielsen, J. G. Polhill, V. Radchuk, M.-S. Rohwäder, R. A. Stillman, J. C. Thiele, D. Ayllón (2020, March 31) *TheODDProtocol for Describing Agent-Based and Other Simulation Models: A Second Update to Improve Clarity, Replication, and Structural Realism* Journal of Artificial Societies and Social Simulation 23(2) 7

(doi: 10.18564/jasss.4259 Url: http://jasss.soc.surrey.ac.uk/23/2/7.html)

[2]C. Leuci (2020, January 30). *MGA - Minimal Genetic Algorithm* (Version 1.1.0). CoMSES Computational Model Library. Retrieved from: https://doi.org/10.25937/db7w-zt41

[3] Loeb L. A. (2011) *Human cancers express mutator phenotypes: origin, consequences and targeting*. Nature Reviews Cancer 11: 450 – 457 (doi: 10.1038/nrc3063)

[4] A.E. Tijhuis, S. C. Johnson, S. E. McClelland (2019) *The emerging links between chromosomal instability (CIN), metastasis, inflammation and tumour immunity*. Molecular Cytogenetics 12:17 (doi: https://doi.org/10.1186/s13039-019-0429-1)

[5]McClintock B. (1984) *The significance of responses of the genome to challenge*. Science 226 (4676): 792 - 801 (doi: 10.1126/science.15739260)

[6] Mantovani A., Allavena P., Sica A., Balkwill F. (2008) *Cancer-related inflammation*. Nature 454: 436 – 444 (https://doi.org/10.1038/nature07205)

Hypermutations in NetLogo mGA (or *HyperMu'NmGA* or more briefly*HyperMu*) was mainly suggested by the studies of Taddei *et al.*[9] and Sniegowski *et al*.[10]. They led their researches on asexually reproducing cells (*E. coli*) engaged in an adaptation experiment. Our model allows the influence of homologous recombination in the control of mutators element diffusion to be simulated. This second feature is related to sexual reproduction maintenance, that is another debated issue in evolutionary biology[11,12].

The characterization of hypermutations in GAs show an increasing interest for applicative purpose, as well. Actually, GAs are powerful search algorithms that can be applied to a wide range of problems and can be exploited as engines in thinking machines. The choice of what values to assign to mutation rate is a vital factor in the success of any GA; generally, parameters setting is accomplished prior to running a GA. Since the best mutation rate depends on the specific problem and could be changed during evolution, the proposed strategy is to let the algorithm to search by itself the best mutation rate[13]. In other words, some parameters as mutation rate could be adjusted by self-adaptation via hypermutation cycles, as theorized in some biological systems[14].

2. Entities, state variables, and scales

The model is implemented in NetLogo programming environment, so the agents are typically named "turtles". In HyperMu, one of their main attributes is the "chromosome" that is a string representing one candidate solution to the problem:

"*find the greatest natural number composed of a given number of digits*".

The number of digits is set initially by the user thanks to the slider GENES NUMBER, indeed each digit can be seen as a "gene" of the chromosome.

---

[7]E. Pennisi (1998, August 21) *How genome readies itself for evolution* Science 281:1131 - 1134 (doi: 10.1126/science.281.5380.1131)

[8]M. Chicurel (2001, June 8) *Can organisms speed their own evolution?* Science 292: 1824 – 1827 (doi: 10.1126/science.292.5523.1824)

[9]Taddei, F., Radman, M., Maynard-Smith, J., Toupance, B., Gouyon, P. H., and Godelle, B. (1997, June) *Role of mutator alleles in adaptive evolution* Nature 387: 700–702. https://doi.org/10.1038/42696

[10]Sniegowski, P., Gerrish, P. and Lenski R. (1997, June) *Evolution of high mutation rates in experimental populations of* E. coli Nature 387: 703–705. https://doi.org/10.1038/42701

[11]Becks, L., Agrawal, A.F. (2012, May) *The Evolution of Sex Is Favoured During Adaptation to New Environments*. PLoS Biol. 10 (5): e1001317. https://doi.org/10.1371/journal.pbio.1001317

[12]Hoffmann, A.A., and Hercus, M.J. (2000, March) *Environmental Stress as an Evolutionary Force*. BioScience, 5 (3): 217-226. https://doi.org/10.1641/0006-3568(2000)050[0217:ESAAEF]2.3.CO;2

[13]Serpell, Mand Smith J. E. (2010) *Self-Adaptation of Mutation Operator and Probability for Permutation Representations in Genetic Algorithms*. Evolutionary Computation 18(3): 491–514

[14]Galhardo R. S. Hastings P. J. and Rosenberg S. M. (2007) *Mutation as a Stress Response and the Regulation of Evolvability*. Crit Rev Biochem Mol Biol. 42(5): 399–435

The candidate solution codified by the chromosome can be more or less close to the optimal solution: this value is quantified by the variable “fitness”; the evaluation methods are described in the submodels section.

The chromosomal genes could be considered as structural genes; turtles own three other variables that can be seen as regulatory genes: the most important one for our purpose is the *mutator* gene that quantifies the severity of mutational events. If planned, the model can acquire two other regulative genes: *mutator-switch* and *mate-switc*h.

The spatial position of turtles have no importance in this model, and no environmental inputs are simulated. The environmental stimuli are substituted by the computational challenge given by the problem submitted to the agents and influencing their reproductive success.

3. Process overview and scheduling

The process follows the general structure of a GA that is shown below:

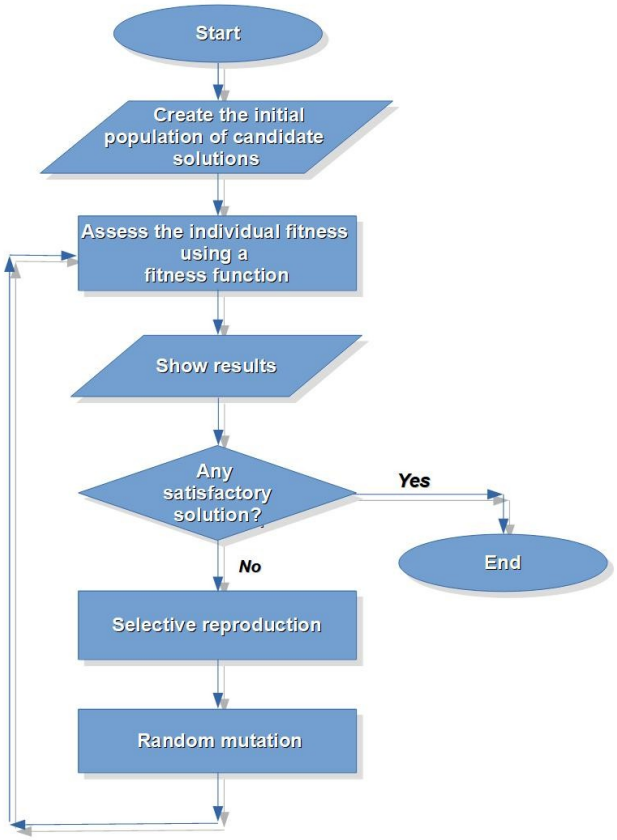


Figure 1. Flowchart of a generic GA

In more detail, for HyperMu we have:

1. A population of turtles is generated. The number of turtles can be previously set by the user.
2. The chromosome of each turtle is built converting the value 10^{n-1} into the corresponding string (*n* is the number of digits composing the wanted solution that is equal to the structural genes inserted into the turtles' chromosomes).
3. The fitness is evaluated according to the selected "method.
4. The worst and the best solution in the population are determined on the basis of turtles' fitness.
5. The two values are plotted
6. Stop conditions[15].
7. One of the best performing turtles reproduces a part of the genome by crossover (sexual reproduction) or entirely by cloning (asexual reproduction).
8. The fitness of the new chromosome is evaluated.
9. A mutational event happens with a frequency set by the user thanks to the slider BASIC_MUTATION-RATE.
10. The fitness of the mutant chromosome is evaluated.
11. Some details about mutators gene activity and distribution is displayed. Another diagram displays the number of asexually or sexually reproducing turtle
12. Return to the item (4).

[15] Some stop conditions are inserted here for our experiments (mainly executed under these parameters set: turtle number 85, number of structural genes 12, basic mutation-rate $5*10^{-6}$); they are useful when an experimental program is automated through the tool "BehaviorSpace" and are configured as follows.
- In presence of mutator genes, often their frequency and activity can increase, so it is interesting to follow their fate after the best solution is reached; in this case, the experiment is stopped when the average value of the mutator genes is reduced by 30% with respect to the default value (i.e. 1); but if this condition is not reached at 500000 ticks after the problem solution, muatators' value reduction is considered potentially endless and the experiment is stopped.
- If the mean of mutators genes value is under one and the ticks number is more than 750000 in the condition we have used in our experiments program the experiment execution is stopped because the evolutive potential is very low and is unlikely an ultimate solution could be found.
- If mutator genes are not active, the experiment ends when the best solution is reached.

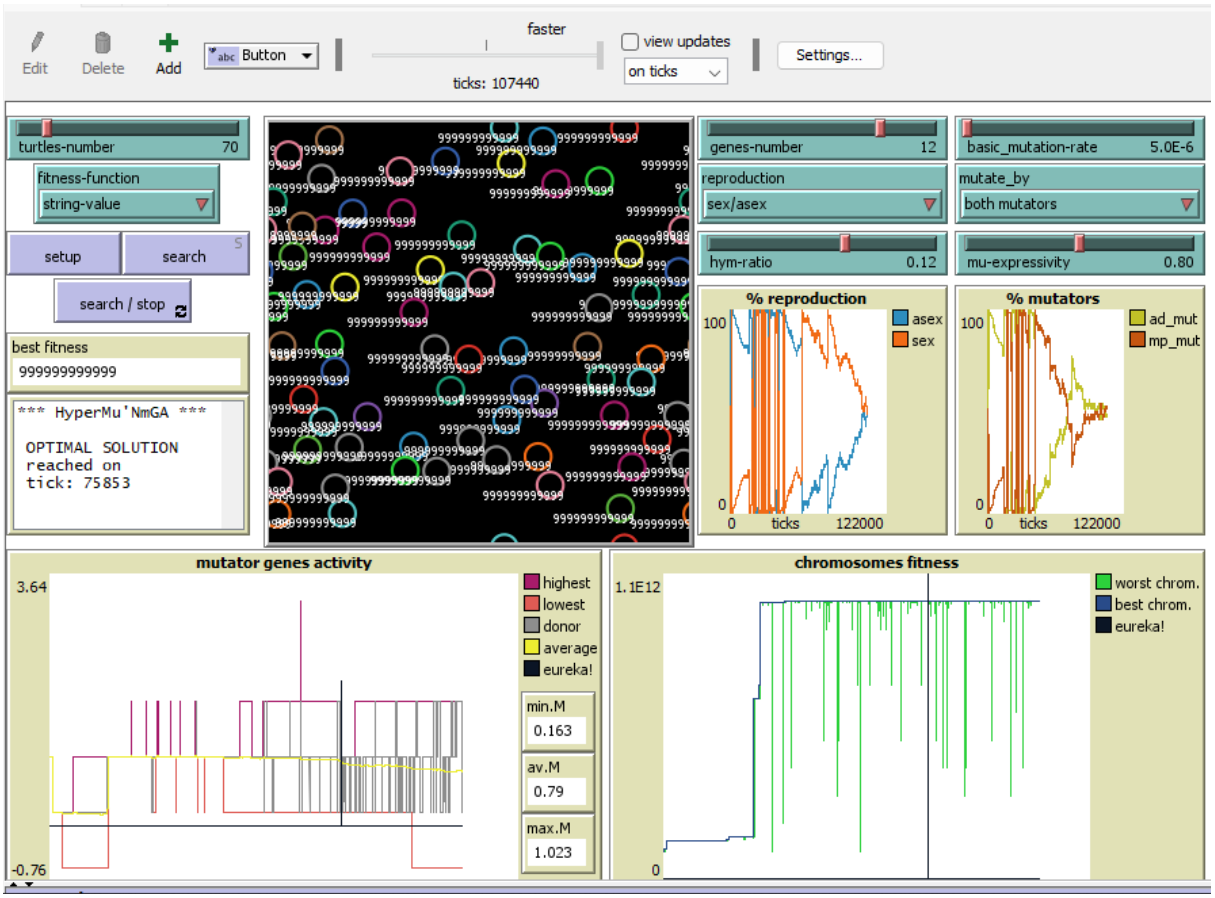


Figure 2. Part of the user interface of HyperMu'NmGA 1.2.0. The executable file is downloadable from the Comses.net library; it requires that the NetLogo 6.2 interpreter be downloaded and installed from the server at Northwestern[16]

[16]Wilensky, U. (1999). *NetLogo*. http://ccl.northwestern.edu/netlogo/. Center for Connected Learning and Computer-Based Modeling, Northwestern Institute on Complex Systems, Northwestern University, Evanston, IL.

4. Design concepts

The design of a GA assumes that optimization of a fitness function can be the consequence of two processes: casual mutation and selective reproduction, as occurs in breeders work; to select (one of) the best individuals, the algorithm must compute the fitness function for each of them. Generation after generation, the optimized trait emerges by artificial selection.

The adaptive process generated by the GA is then exploited as background to inspect the dynamics of mutator (or antimutator) alleles inside the turtles' population. To shed light about this process, we have designed the model assuming the following statements:

- Mutator genes come from an impairment process of hypothetical genes involved in genome stabilization; the effect is a consequence of the basic mutation rate acting on each genome.
- The phenotype related to the different mutator alleles is a quantitative trait and it can assume positive or negative decimal values. When it is negative, it protects against mutations, on the contrary, when positive it enhances the severity of mutation. In the first case they can act as some *proto-oncogenes*, that are able to protect against mutagenesis, repairing DNA mismatch. In the second case they can act as some *oncogenes* that enhance mutagenesis and tumorigenesis[17];
- mutator genes can feedback on themselves leading toward a more severe mutagenic activity or toward a reversion (triggering a hypermutations' cycle).

5. Initialization

The initialization of one experiment takes place after Setup button activation. It includes the creation of some turtles whose number is set by Turtles-Number slider. Initially, each turtle carries one chromosome codifying for a number representing a candidate solution to the problem: its value is initially assigned for all turtles by the formula:

$$10^{n-1}$$

where n is given by the slider GENES-NUMBER.

Before starting a search, there are other parameters to set: their importance will be discussed in the submodels sections.

6. Input data

The model does not use input data to represent time-varying processes

[17]Sherr, C. J. (2004, January 23) *Principles of Tumor Suppression* Cell, 116: 235–246. https://doi.org/10.1016/S0092-8674(03)01075-4

7. Submodels

HyperMu includes three submodels: fitness functions, reproduction, and mutation.

1. FITNESS FUNCTIONS. The fitness function can be evaluated by two different methods:

   A."string-value" converts the chromosome string into its numerical value;

   B. "hamming" evaluates the Hamming distance[18] complement (that is the proximity) to the best chromosomal string by counting the number of 9 digits.

The fitness range changes as a consequence of the selected fitness-function: it is between 10^{n-1} and 10^n-1 for string-value mode; it is between 0 and n for hamming mode (in this case n represents the GENES-NUMBER).

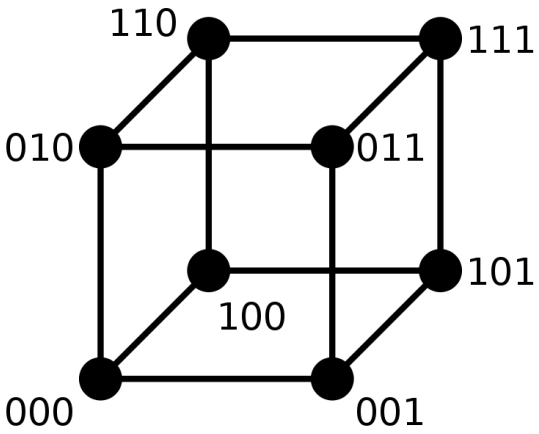


Figure 3. Comparison of the two fitness functions in a simplified system: the strings are binary and n is 3. In a "string-value" mode the string 001 has a lower numerical value and consequently also lower fitness than the string 100; in the "hamming" mode would not be any difference between the two strings because both require at least two mutations to reach the higher fitness string 111. The two functions give rise to different fitness landscapes: the more regular one is hamming, because it contains only one peak, while the string-value contains more irregularities, which creates a more complex fitness landscape. (Figure from Wikipedia)

2. REPRODUCTION. The user can choose to work with a population reproducing sexually, asexually or a mixed population of sexual and asexual turtles. In the last case, the reproduction style (determined by *mate-switch*) will be treated as a genetic character. Anyway, reproduction is a selective event because it involves always (one of) the best performing turtle called *donor*, and another one randomly chosen called *recipient*.

   A. ASEXUAL REPRODUCTION: CLONING. In asexually reproduction, donor turtle inserts its entire genome (structural genes and regulative genes) into the recipient turtle.

   B. SEXUAL REPRODUCTION: RECOMBINATION. Sexually reproducing turtles exhibits a kind of homologous recombination, similar to the bacterial process related to conjugation, more than the eukaryotic meiotic crossing-over: indeed just the recipient turtles varies its genome; on the contrary, the donor turtle replicates part of its chromosome into the corresponding loci of the second one (the recipient chromosome) so that no turtle is removed from the world, but one of them changes part of its chromosomal sequence by hybridization (see figure 4).

[18] „Hamming distance" from Wikipedia: https://en.wikipedia.org/wiki/Hamming_distance

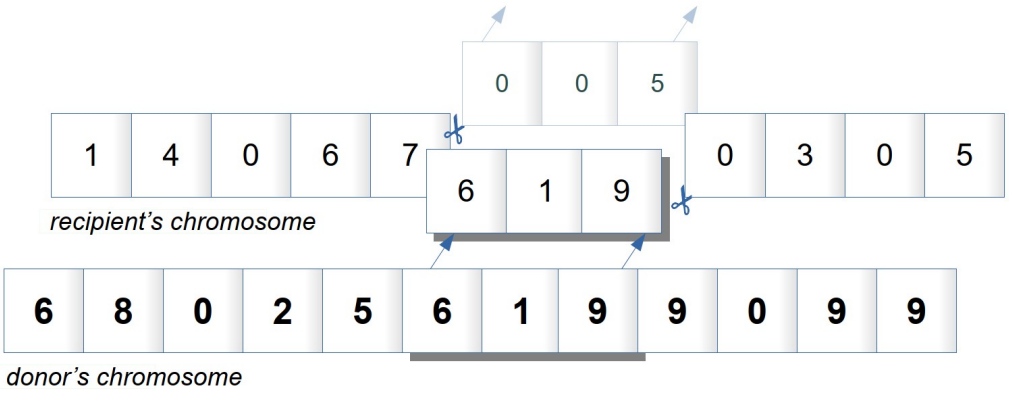


Figure 4. A representation of recombination between a donor's chromosome and a recipient chromosome; the two split points are chosen randomly, and the chromosomes are treated as circular.

3. MUTATION. Mutation events occur randomly in one turtle and one loci with a frequency given by the slider BASIC_MUTATION-RATE: it can be seen as the level of environmental mutagenicity acting on a single gene.

The mutations are always point-mutations, in the mode conventionally named "simple mutagenesis". If mutagenesis does not follow a "simple" mode, the mutations' severity varies for each turtle, according to its mutator gene that determines a multimutation process and/or a hypermutation process. Multimutation means that more point-mutations can hit the chromosome at the same time: the number of point-mutations is determined by mutator gene if it is greater than one, but if it is under one, no mutation will alter the structural genes. Nevertheless, it could be that the same mutator undergoes to mutagenesis: the frequency depends on the parameter HYM-RATIO whose value is set by a slider; if turtles' genes were physical entities, HYM-RATIO could be approximatively the ratio between the lengths' sum of regulative genes and the length of one turtles' entire genetic material.

If mutagenesis does not follow the "simple" way, it will be active the mutator gene that influences the number of mutation on the chromosome and is able to mutate itself starting a hypermutation cycle. The user can select one of the hypermutation modes thanks to the chooser MUTATE_BY; it allows to choose: "additive" (i.e. the weaker mode) and "multiplicative" (i.e. the stronger mode); it is possible to have a population where both modes coexist; in this case, the hypermutation stile will be treated as a genetic trait (determined by *mutator-switch*) that can mutate and recombine.

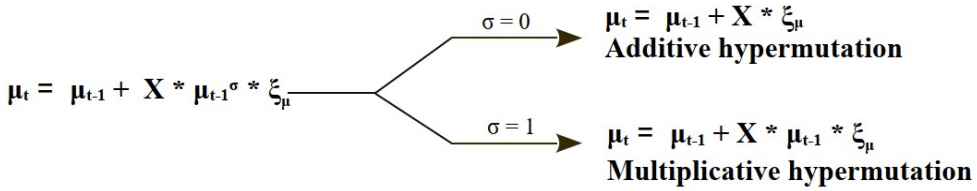
One of the main goals of the model is to detect if the GA select some particular alleles and if this leads to a variation in the performances of the GA. Our turtles are not provided with metabolism or physiological states, so we have used different algebraic equations to produce different processes of mutators' mutation as shown in the box below.

The proposed equations do not tend to replicate any empirical set of data about self-mutagenesis kinetics; it has been built to output two different dynamics (stronger/weaker) in order to characterize their behaviour when they are embodied into the adaptive process created by a GA, working under two different reproductive regimens. The equations are designed to increase or decrease the mutator value with the same probability; the GA's artificial selection

will choose the most suitable variants if they are caught by the selection filter, otherwise, they will undergo genetic drift.

Box 1

Mutating mutators



- σ: binary value of mutator-switch (agent variable);
- μ: decimal value of mutator alleles (agent variable);
- ξ_μ: decimal value of mutators expressivity (global variable);
- X: random discrete variable, taking on a value from the set {-1,0,+1}, each one with equal probability (local variable).