Biologically-inspired algorithms and models

5. Controlling diversity

Maciej Komosinski

Standard selection techniques

- When do they work well?
  - When searching for local optima
  - In the presence of global convexity
  - In not-too-rugged fitness landscape
  - In non-deceptive fitness landscape

Standard selection techniques

- When do they work well?
  - When searching for local optima
  - In the presence of global convexity
  - In not-too-rugged fitness landscape
  - In non-deceptive fitness landscape
- When do they work not so well?

Standard selection techniques

- When do they work well?
  - When searching for local optima
  - In the presence of global convexity
  - In not-too-rugged fitness landscape
  - In non-deceptive fitness landscape
- When do they work not so well?
- How can they be improved?

Controlling diversity – a reminder

We will now learn about modern, efficient diversity control methods used in population algorithms. Their use is justified in problems where standard selection techniques are not effective enough (previous slide). This group of methods is usually called *quality–diversity*.

Recall the methods you have learned so far for enforcing/maintaining diversity in fitness values of solutions.

Controlling diversity – a reminder

We will now learn about modern, efficient diversity control methods used in population algorithms. Their use is justified in problems where standard selection techniques are not effective enough (previous slide). This group of methods is usually called *quality–diversity*.

Recall the methods you have learned so far for enforcing/maintaining diversity in fitness values of solutions.

- convection selection (presentation #1, meta-schemes of selection)

Recall the methods you have learned so far for enforcing/maintaining diversity in the contents of solutions.

Controlling diversity – a reminder

We will now learn about modern, efficient diversity control methods used in population algorithms. Their use is justified in problems where standard selection techniques are not effective enough (previous slide). This group of methods is usually called *quality–diversity*.

Recall the methods you have learned so far for enforcing/maintaining diversity in fitness values of solutions.

- convection selection (presentation #1, meta-schemes of selection)

Recall the methods you have learned so far for enforcing/maintaining diversity in the contents of solutions.

- crowding factor model (presentation #1, selection – additional properties)
- niching (presentation #4, algorithm logic and speciation)

Controlling the diversity in fitness values

In order of complexity:

- FUSS
- FUDS
- Convection selection – already discussed
- HFC

- Slightly different premises: MAP-Elites (requires auxiliary objective functions)

FUSS (Fitness Uniform Selection Scheme)

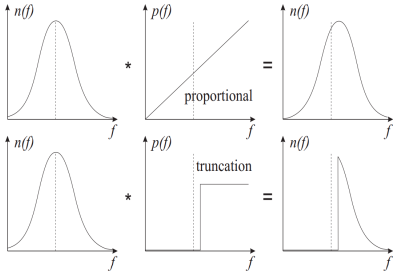
- Determine the lowest and the highest fitness values in the population: f_{min}, f_{max}.
- Select a fitness value f uniformly in the interval $[f_{min}, f_{max}]$.
- The individual with fitness nearest to f is selected [HL06].

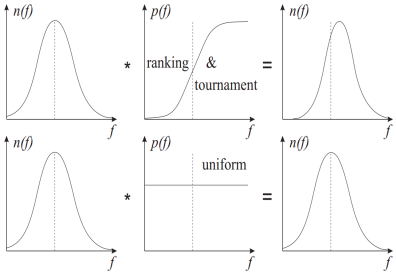
FUSS (Fitness Uniform Selection Scheme)

- Determine the lowest and the highest fitness values in the population: f_{min}, f_{max}.
- Select a fitness value f uniformly in the interval $[f_{min}, f_{max}]$.
- The individual with fitness nearest to f is selected [HL06].

- Uses no direct selective pressure.
- As better solutions are more difficult to find, it applies non-explicit selective pressure.

FUSS selection probability





FUSS selection probability

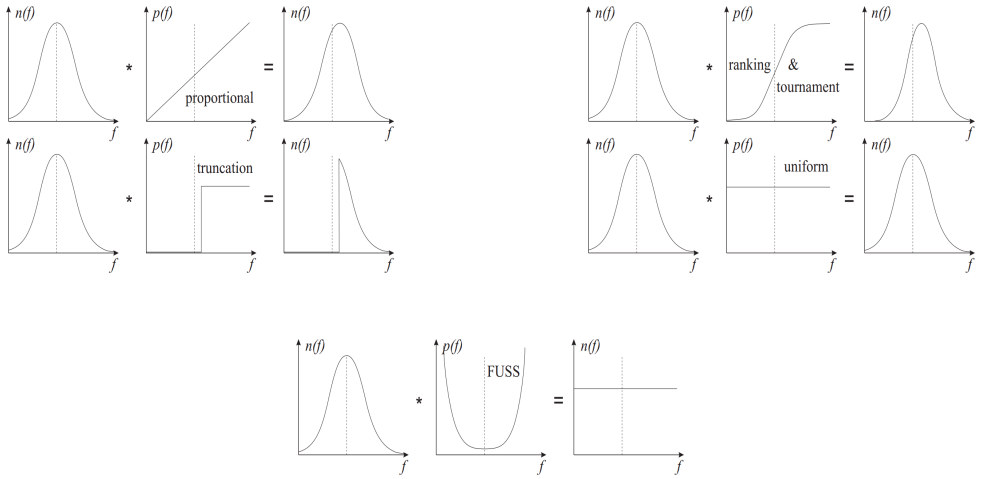


Figure: Fitness-dependent probability of selection $p(f)$ using different selection methods, and the expected fitness distribution in a new generation $n(f)$ [HL06].

FUSS behavior

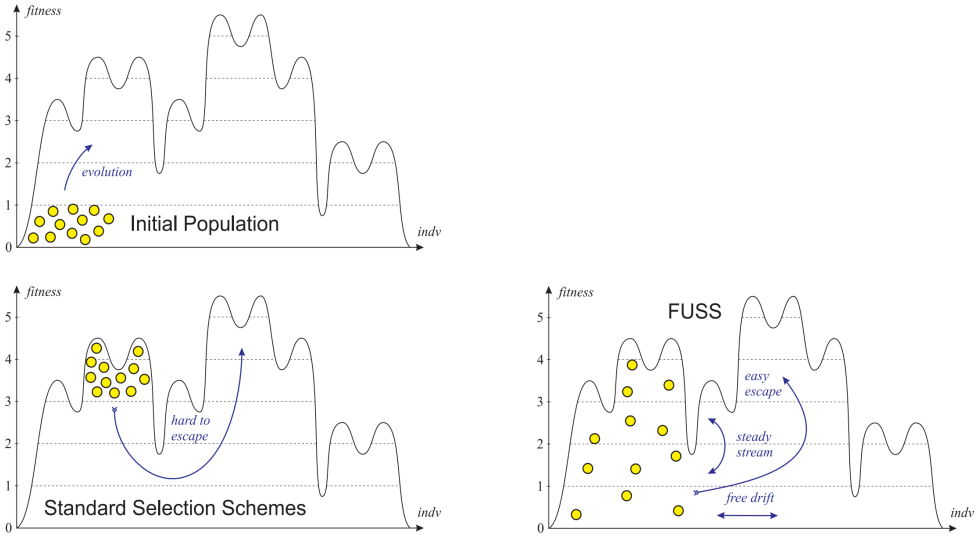


Figure: A forced "vertical" spread helps to continuously discover new optima [HL06].

Properties of FUSS

- FUSS effectively explores (also) fit individuals – the same reason as earlier discussed: why *EqualWidth* convection selection with random selection in subpopulations still manages to optimize?

* "Takeover" means that the best individual colonizes and occupies the entire population. "Takeover time": starting from a population with a single unique best individual (the remaining ones are worse), how many iterations of a given selection method we expect for the population to become homogenous and only consist of copies of the best individual? This single value can be used to compare selective pressures of different selection techniques.

Properties of FUSS

- FUSS effectively explores (also) fit individuals – the same reason as earlier discussed: why *EqualWidth* convection selection with random selection in subpopulations still manages to optimize?
- No takeover problem* – i.e., no premature convergence, or convergence at all!

* "Takeover" means that the best individual colonizes and occupies the entire population. "Takeover time": starting from a population with a single unique best individual (the remaining ones are worse), how many iterations of a given selection method we expect for the population to become homogenous and only consist of copies of the best individual? This single value can be used to compare selective pressures of different selection techniques.

Properties of FUSS

- FUSS effectively explores (also) fit individuals – the same reason as earlier discussed: why *EqualWidth* convection selection with random selection in subpopulations still manages to optimize?
- No takeover problem* – i.e., no premature convergence, or convergence at all!
- Favors individuals with rare fitness values (compared to common fitness values) – may favor unfit individuals, if they are rare.

* "Takeover" means that the best individual colonizes and occupies the entire population. "Takeover time": starting from a population with a single unique best individual (the remaining ones are worse), how many iterations of a given selection method we expect for the population to become homogenous and only consist of copies of the best individual? This single value can be used to compare selective pressures of different selection techniques.

Properties of FUSS

- FUSS effectively explores (also) fit individuals – the same reason as earlier discussed: why *EqualWidth* convection selection with random selection in subpopulations still manages to optimize?
- No takeover problem* – i.e., no premature convergence, or convergence at all!
- Favors individuals with rare fitness values (compared to common fitness values) – may favor unfit individuals, if they are rare.
- Free drift operates at different fitness levels and facilitates diversity.

* "Takeover" means that the best individual colonizes and occupies the entire population. "Takeover time": starting from a population with a single unique best individual (the remaining ones are worse), how many iterations of a given selection method we expect for the population to become homogenous and only consist of copies of the best individual? This single value can be used to compare selective pressures of different selection techniques.

Properties of FUSS

- FUSS effectively explores (also) fit individuals – the same reason as earlier discussed: why *EqualWidth* convection selection with random selection in subpopulations still manages to optimize?
- No takeover problem* – i.e., no premature convergence, or convergence at all!
- Favors individuals with rare fitness values (compared to common fitness values) – may favor unfit individuals, if they are rare.
- Free drift operates at different fitness levels and facilitates diversity.
- Self-balancing and parameter free.

* "Takeover" means that the best individual colonizes and occupies the entire population. "Takeover time": starting from a population with a single unique best individual (the remaining ones are worse), how many iterations of a given selection method we expect for the population to become homogenous and only consist of copies of the best individual? This single value can be used to compare selective pressures of different selection techniques.

Properties of FUSS

- FUSS effectively explores (also) fit individuals – the same reason as earlier discussed: why *EqualWidth* convection selection with random selection in subpopulations still manages to optimize?
- No takeover problem* – i.e., no premature convergence, or convergence at all!
- Favors individuals with rare fitness values (compared to common fitness values) – may favor unfit individuals, if they are rare.
- Free drift operates at different fitness levels and facilitates diversity.
- Self-balancing and parameter free.
- Simple implementation and low computational cost.

* "Takeover" means that the best individual colonizes and occupies the entire population. "Takeover time": starting from a population with a single unique best individual (the remaining ones are worse), how many iterations of a given selection method we expect for the population to become homogenous and only consist of copies of the best individual? This single value can be used to compare selective pressures of different selection techniques.

FUDS (Fitness Uniform Deletion Scheme)

- Deletion (i.e., negative selection) scheme. Any positive selection scheme can be used in conjunction with it.
- Divides the full fitness range of solutions into subintervals of equal width.
- Deletes a random solution from the most crowded subinterval.

FUDS (Fitness Uniform Deletion Scheme)

- Deletion (i.e., negative selection) scheme. Any positive selection scheme can be used in conjunction with it.
- Divides the full fitness range of solutions into subintervals of equal width.
- Deletes a random solution from the most crowded subinterval.

- Let f_{min} and f_{max} be the minimum and maximum fitness values possible for a problem, or reasonable upper and lower bounds.
- We divide the interval $[f_{min}, f_{max}]$ into a collection of subintervals of equal length $[f_{min}, f_{min} + \epsilon), [f_{min} + \epsilon, f_{min} + 2\epsilon), \ldots, [f_{max} - \epsilon, f_{max}]$.
- The subinterval with the largest number of individuals is determined, and then a random individual in this subinterval is deleted.
- In the case of multiple subintervals having the same number of individuals, the lowest subinterval is chosen [LH05].

Properties of FUDS

- No takeover problem in FUDS.

Properties of FUDS

- No takeover problem in FUDS.
- Facilitates continuous creation of individuals on every fitness level.

Properties of FUDS

- No takeover problem in FUDS.
- Facilitates continuous creation of individuals on every fitness level.
- Robust performance with respect to the intensity of positive selection pressure: even very low positive selection pressure (even random positive selection!) will not result in losing the best individuals – they will be preserved as long as they are rare.

Properties of FUDS

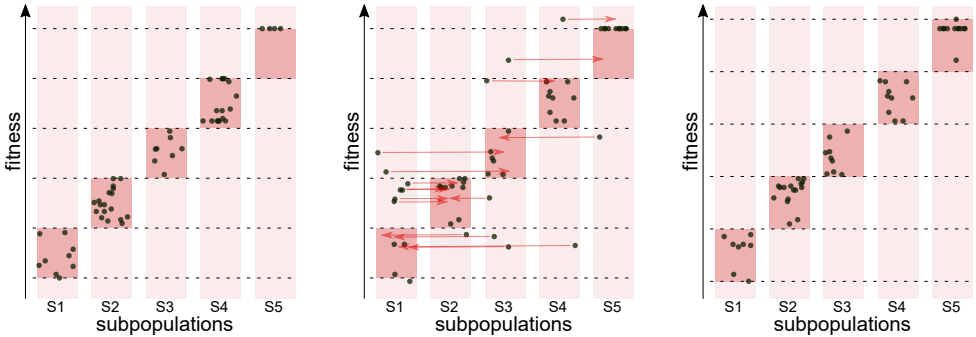
- No takeover problem in FUDS.
- Facilitates continuous creation of individuals on every fitness level.
- Robust performance with respect to the intensity of positive selection pressure: even very low positive selection pressure (even random positive selection!) will not result in losing the best individuals – they will be preserved as long as they are rare.
- Simple implementation and low computational cost.

Convection selection logic

ConvSel (Convection Selection)

- Already discussed earlier.
- A variant of the *Island Model*.
- Divides the population into M subpopulations, each of which is assigned a disjunctive fitness range of equal width.
- Subpopulations evolve independently for $N \cdot R$ evaluations, where N is the size of the full population, and R is a scaling parameter.
- Then, the fitness ranges of subpopulations are recomputed and solutions are reassigned (they migrate) to new subpopulations.
- During independent evolution in subpopulations, any EA can be used with any positive/negative selection.

Convection selection – the *convection*



After i-th migration | Before $(i+1)$-th migration | After $(i+1)$-th migration

Figure: Improved offspring of poor solutions can move up the subpopulation chain.
https://www.cs.put.poznan.pl/mkomosinski/convection-vpos.svg

HFC (Hierarchical Fair Competition)

- Population divided into subpopulations.
- Each subpopulation has admission/export levels of fitness associated with them; export level of i-th is admission level for $(i+1)$-th.
- Solutions exceeding the export fitness level of their subpopulation are moved to the admission buffer (this avoids the dominance of "champions" over each subpopulation).
- Broken solutions that may appear and fall below the admission level are not treated in any special way and are left untouched in subpopulations.
- In the event of migration, solutions move from the admission buffers upwards in the hierarchy of subpopulations.
- The worst subpopulation is refilled with random solutions.

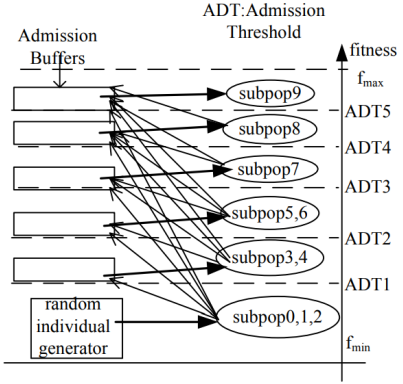


Figure: The HFC model [Hu+05].

HFC metaphor

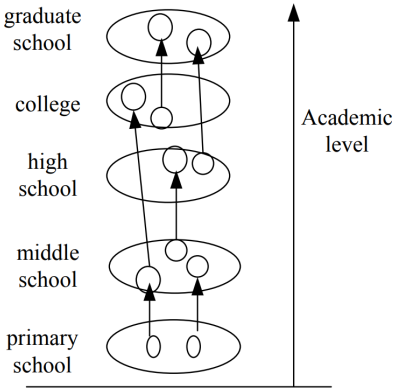


Figure: The metaphor: fair competition principle from societal and economic systems [Hu+05]. No falling down in education (?)

HFC properties

- Continuous supply and incorporation of new low-level genetic material.

HFC properties

- Continuous supply and incorporation of new low-level genetic material.
- Avoiding the founder effect.

- Continuous supply and incorporation of new low-level genetic material.
- Avoiding the founder effect.
- Broken solutions don't get a chance to spoil lower-level subpopulations...

HFC properties

- Continuous supply and incorporation of new low-level genetic material.
- Avoiding the founder effect.
- Broken solutions don't get a chance to spoil lower-level subpopulations...
- ...but they don't get a chance to be fixed either – the selective pressure in their subpopulation will kill them.

HFC properties

- Continuous supply and incorporation of new low-level genetic material.
- Avoiding the founder effect.
- Broken solutions don't get a chance to spoil lower-level subpopulations...
- ...but they don't get a chance to be fixed either – the selective pressure in their subpopulation will kill them.
- Just as in convection selection, during evolution in subpopulations, any EA can be used with any positive/negative selection.

Comparison: experiment setup

Four algorithms:

- a single population standard EA (StdEA)
- EA with a fitness uniform selection scheme (FUSS)
- EA with a fitness uniform deletion scheme (FUDS)
- multi-population EA with convection selection (ConvSel)
- no HFC :(Not enough time to implement [BKM22], but work in progress now!

Four benchmarks:

- Three simple math benchmarks: Drop-wave function, Shaffer function N.2, Shaffer function N.4.
- One complex evolutionary design problem (evolution of fast movement of simulated agents in a flat 3D environment).

All algorithms were using a steady-state evolution with a tournament selection and random deletion.

Parameter-agnostic comparison!

Parameter name	StdEA	FUSS	FUDS	ConvSel	Values
population size	✓	✓	✓	✓	[100, 200, 500]
tournament size	✓		✓	✓	[3, 5, 7]
crossover prob.	✓	✓	✓	✓	[0.5, 0.75]
mutation prob.	✓	✓	✓	✓	[0.25, 0.5]
M				✓	[5, 10]
R				✓	[10, 25]
α (math only)	✓	✓	✓	✓	[0.2, 0.4]

Table: The values of parameters used in the experiments. For each parameter, the methods using it are marked. Parameter α (mutation range) was only used in mathematical benchmarks.

Experiment parameters

- Testing every combination of the algorithm, parameter values and fitness function.
- 30 (mathematical benchmarks) or 10 (evolutionary design) independent evolutionary runs per combination.
- The mean and standard deviation were calculated for every combination of parameter values.
- $5 \cdot 10^5$ solutions (mathematical benchmarks) or 10^6 solutions (evolutionary design) per run.
- Parameter-agnostic comparison with regard to the parameters of the compared methods (at each point during evolution considers the best result from all combinations of parameter values).

Evolutionary design benchmark (land velocity)

Method	Mean	Std
StdEA	0.0143	0.0147
FUSS	0.0219	0.0141
FUDS	0.0177	0.0090
ConvSel	0.0216	0.0198

Table: Comparison of best average values obtained by fitness diversity methods for agent velocity maximization problem.

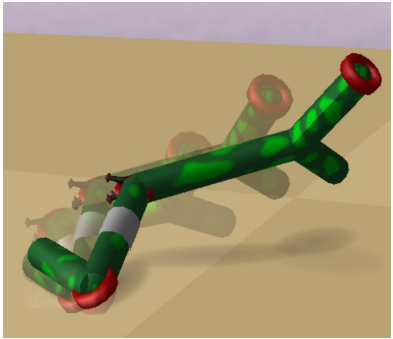


Figure: Example of an evolved agent in motion, as simulated in the Framsticks environment.

Results: evolutionary design benchmark (land velocity)

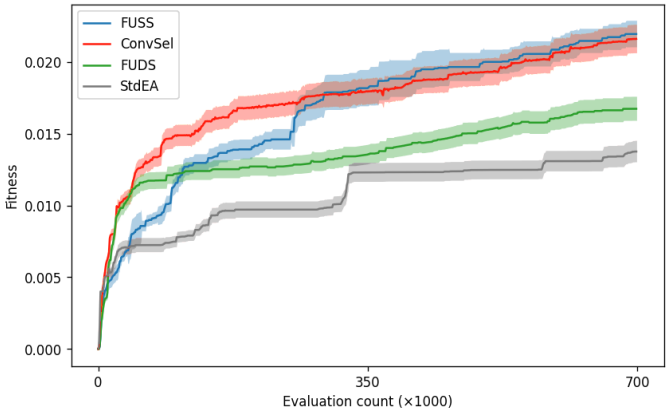


Figure: Comparison of the performance of fitness diversity methods. Each series shows the highest value of the average from the set of parameters (thus each point may originate from different parameter values). The band around each series represents 5% of the standard deviation.

Experiment: conclusions

- For a difficult problem of evolutionary design, all tested fitness diversity methods outperform steady-state single-population EA with tournament selection, with FUSS and ConvSel outperforming FUDS.
- For easier, mathematical benchmarks, FUDS and ConvSel outperform traditional evolutionary approaches.

MAP-Elites

A more complex diversification mechanism (albeit requiring the introduction of additional evaluation criteria) is MAP-Elites (Multi-dimensional Archive of Phenotypic Elites) [MC15].

- Define your fitness as usually (for example: maximize velocity of a robot).
- Define other features of each solution (for example: robot size, weight, energy consumption).
- Discretize ranges of these features, thus creating a multi-dimensional grid with "cells".
- During evolution, select an individual from a random cell, mutate/crossover it, evaluate its fitness and features, and put into the appropriate cell.
- In each cell keep only one solution with the best fitness found so far.

MAP-Elites

A more complex diversification mechanism (albeit requiring the introduction of additional evaluation criteria) is MAP-Elites (Multi-dimensional Archive of Phenotypic Elites) [MC15].

- Define your fitness as usually (for example: maximize velocity of a robot).
- Define other features of each solution (for example: robot size, weight, energy consumption).
- Discretize ranges of these features, thus creating a multi-dimensional grid with "cells".
- During evolution, select an individual from a random cell, mutate/crossover it, evaluate its fitness and features, and put into the appropriate cell.
- In each cell keep only one solution with the best fitness found so far.

Variants:

- Keep more than one solution in each cell.
- Start with a coarse discretization of features and gradually increase the number of intervals (resolution).

Niching

Already discussed earlier:

new fitness := original fitness/(the sum of similarities to other individuals)

. . . or equivalently:

new fitness := original fitness $\cdot$ (1 + the sum of distances to other individuals)

Novelty search

It seems an exotic idea for an optimization algorithm to completely ignore the objective function and base the logic of the algorithm on the very denominator of the formula for niching (and then "original fitness" becomes a constant). However, for certain optimization problems and an appropriate similarity measure (usually based on complex phenotypic features), this approach (*novelty search*) may suffice for successful optimization.

Enforcing just diversity is also useful wherever we want to evenly cover some complex space – e.g., to generate possibly diverse programming tests, evaluation functions, configurations in games, generative procedures, antigens in biology, etc.

Two criteria

In contrast to the formula for niching, which aggregates the quality of a solution and the dissimilarity of its contents from other known solutions, one can treat both of these factors as separate criteria (adequately to the name of the group of methods now being discussed: *quality–diversity*) and use a two-criteria optimization (e.g., NSGA-2).

The main advantage of this approach is the lack of trade-offs (typical when moving from aggregating criteria to treating them separately).

Global vs. local diversity

- *dist* – distance measure
- x_i – i-th solution
- n – population size
- k – the number of solutions, $k < n$, closest to the i-th solution according to *dist*

The global novelty metric for i-th solution is

$$distance\_global_i = \frac{1}{n} \sum_{\substack{j=1 \\ j \neq i}}^{n} dist(x_i, x_j)$$

The local novelty metric for i-th solution is

$$distance\_local_i = \frac{1}{k} \sum_{\substack{j=1 \\ j \neq i}}^{k} dist(x_i, x_j)$$

In novelty search, this is the only optimized criterion.
In niching, original fitness is multiplied by (1 + the global or local novelty metric).

NSLC: Novelty Search with Local Competition

Features [LS11]: archive (memory: track visited areas) and local competition (only diversify locally). Both features can also be (and are) employed in niching and in novelty search (these two methods have just been discussed).

From [CD17]:

≪
In NSLC, the exploration focuses on solutions that are both novel (according to the novelty score) and locally high-performing. The main insight consists of comparing the performance of a solution only to those that are close in the descriptor space. This is achieved with a "local quality score" that is defined as the number of the k-nearest neighboring solutions in the novelty archive with a lower performance (e.g., slower walking speed) than the considered solution. The exploration is then achieved with a multi-objective optimization algorithm (e.g., NSGA-II) that optimizes both the novelty and local quality scores of the solutions.

However, the local quality score does not influence the threshold used to select whether an individual is added to the novelty archive. The final result of NSLC is the population containing solutions that are both novel and high-performing compared to other local solutions. In other words, the population gathers solutions that are both different from those saved in the novelty archive, and high-performing when compared to similar types of solutions.
≫

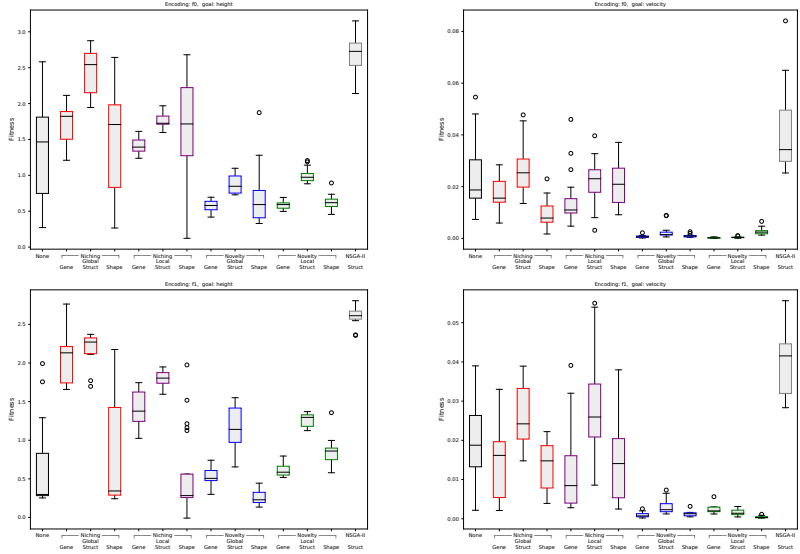


Figure: Raw fitness of the best evolved solutions for different diversity maintenance techniques. Columns are fitness criteria (left: height, right: velocity). Rows are genetic encodings (top: *f0*, bottom: *f1*). 14·10 solutions in each plot. Distance measure: genetic Levenshtein (*Gene*), phenetic graph structure (*Struct*), phenetic shape descriptors (*Shape*).

References I

[BKM22] Kamil Basiukajc, Maciej Komosinski, and Konrad Miazga. "Fitness Diversification in the Service of Fitness Optimization: a Comparison Study". In: *Genetic and Evolutionary Computation Conference Companion (GECCO '22)*. Boston, USA: ACM, 2022, pp. 471–474. DOI: 10.1145/3520304.3528949. URL: http://www.framsticks.com/files/common/FitnessDiversity.pdf.

[CD17] Antoine Cully and Yiannis Demiris. "Quality and diversity optimization: A unifying modular framework". In: *IEEE Transactions on Evolutionary Computation* 22.2 (2017), pp. 245–259. URL: https://arxiv.org/pdf/1708.09251.pdf.

[HL06] Marcus Hutter and Shane Legg. "Fitness uniform optimization". In: *IEEE Transactions on Evolutionary Computation* 10.5 (2006), pp. 568–589. URL: https://arxiv.org/pdf/cs/0610126.pdf.

[Hu+05] Jianjun Hu et al. "The Hierarchical Fair Competition (HFC) framework for sustainable evolutionary algorithms". In: *Evolutionary Computation* 13.2 (2005), pp. 241–277. URL: https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=9a9418ae55bdff2f1a988effe8554e8ec0050df0.

[LH05] Shane Legg and Marcus Hutter. "Fitness uniform deletion: A simple way to preserve diversity". In: *Proceedings of the 7th annual conference on Genetic and evolutionary computation*. 2005, pp. 1271–1278. URL: https://arxiv.org/pdf/cs/0504035.pdf.

[LS11] Joel Lehman and Kenneth O. Stanley. "Evolving a diversity of virtual creatures through novelty search and local competition". In: *Proceedings of the 13th annual conference on genetic and evolutionary computation*. 2011, pp. 211–218. URL: http://eplex.cs.ucf.edu/papers/lehman_gecco11.pdf.

[MC15] Jean-Baptiste Mouret and Jeff Clune. "Illuminating search spaces by mapping elites". In: *arXiv preprint arXiv:1504.04909* (2015). URL: https://arxiv.org/pdf/1504.04909.pdf.