

The replacement strategy for an Evolutionary Algorithm defines which parents and children survive into the next generation.

## Generational

Generational replacement replaces the entire parent population with the newly generated child population at every generation. Generational replacement can be activated with the argument:

```
--replacement generational
```

or by setting the parameter **REPLACEMENT** to **generational** in either a parameters file or in the params dictionary.

## Elitism

Generational replacement is most commonly used in conjunction with elitism. With elitism, the best **[ELITE\_SIZE]** individuals in the parent population are copied over unchanged to the next generation. Elitism ensures continuity of the best ever solution at all stages through the evolutionary process, and allows for the best solution to be updated at each generation.

The number of children created at each generation is known as the **GENERATION\_SIZE**, and is equal to **[POPULATION\_SIZE]** - **[ELITE\_SIZE]**. This parameter is set automatically by PonyGE2.

The default number of elites is 1 percent of the population size. This value can be changed with the argument:

```
--elite_size [INT]
```

or by setting the parameter **ELITE\_SIZE** to **[INT]** in either a parameters file or in the params dictionary, where **[INT]** is an integer which specifies the number of elites to be saved between generations.

## Steady State

With steady state replacement, only 2 children are created by the evolutionary process at each evolutionary step (i.e. the **GENERATION\_SIZE** is automatically set to 2). Steady state replacement first selects two parents, performs crossover on them to produce two children, mutates and evaluates these children, and then replaces the two worst individuals in the original population with the new children, regardless of whether or not these children are fitter than the individuals they replace. As such, steady state replacement implements its own specialised **step** loop.

At each generation, steady state replacement continues until **POPULATION\_SIZE** children have been created and inserted into the original population. Adopting a steady state replacement strategy ensures that successive populations overlap to a significant degree (i.e. parents and their children can co-exist). This requires less memory as only one population of individuals needs to be maintained at any given point in time. This strategy also allows the evolutionary process to exploit good solutions as soon as they appear.

Steady state replacement can be activated with the argument:

```
--replacement steady_state
```

or by setting the parameter **REPLACEMENT** to **steady\_state** in either a parameters file or in the params dictionary.

## NSGA2

NSGA2 replacement replaces the old population with the new population based on crowding distance per pareto front. The crowding distance comparison of the individuals ensures dispersion on the pareto fronts. Elitism is implicitly maintained by the existance of the first pareto front. For more details on the NSGA-II operator see [[[Deb et al., 2002|References]]].

Activate with:

```
--replacement nsga2_replacement
```

or by setting the parameter **REPLACEMENT** to **nsga2\_replacement** in either a parameters file or in the params dictionary.

**NOTE** that NSGA2 replacement can only be used in conjunction with [[multiple objective optimisation|Evaluation#multiple-objective-optimisation]] problems.

**NOTE** that multi-objective optimisation compatible operators require a **multi\_objective = True** attribute.