Seeding GE Runs with target solutions

Combining the [[GE LR Parser|Scripts#ge-lr-parser]] with the full PonyGE2 library, it is possible to parse a target string into a GE individual and then to seed an evolutionary run of PonyGE2 with that individual. Provision is made in PonyGE2 to allow for the seeding of as many target individuals as desired into an evolutionary run.

There are two ways to seed individuals into a PonyGE2 run:

1. Seeding runs with a single target solution

If a single target phenotype string is to be included into the initial population, users can specify the argument:

```
--reverse_mapping_target [TARGET_STRING]
```

or set the parameter `REVERSE_MAPPING_TARGET` to `[TARGET_STRING]` in either a parameters file or in the params dictionary, where `[TARGET_STRING]` is the phenoytpe string specifying the target string to be parsed by the GE LR Parser into a GE individual.

NOTE *that as with the GE LR Parser described above, a compatible grammar file needs to be specified along with the target string. If the target string cannot be parsed using the specified grammar, an error will occur.*

2. Seeding runs with one or more target solutions

Alternatively, if one or more target individuals are to be seeded into a GE population, a folder has been made available for saving populations of desired individuals for seeding. The root directory contains a `seeds` folder. Any number of desired target individuals for seeding can be saved in ***separate text files*** within a unique folder in the scripts directory. This target seed folder can then be specified with the argument:

```
--target_seed_folder [TARGET_SEED_FOLDER]
```

or by setting the parameter `TARGET_SEED_FOLDER` to `[TARGET_SEED_FOLDER]` in either a parameters file or in the params dictionary, where `[TARGET_SEED_FOLDER]` is the name of the target folder within the `scripts` directory which contains target seed individuals.

PonyGE2 currently supports four formats for saving and re-loading of such individuals (examples of each are given in the `seeds/example_pop` folder):

1. (`example_1.txt` in `seeds/example_pop`) PonyGE2 can re-load "best.txt" outputs from previous PonyGE2 runs. These files contain the saved genotypes and phenotypes of the best solution evolved over the course of an evolutionary run. Re-using these output files greatly improves the seeding process, as the genotypes can be quickly used to re-map the exact identical individual evolved by

PonyGE2. If possible, this is the preferred option for seeding populations as the use of genomes to re-build previous individuals guarantees the same genetic information will be retained.

2. (`example_2.txt` in `seeds/example_pop`) Target phenotypes can be saved as a simple text file with a single header of "`Phenotype:`", followed by the phenotype string itself on the following line. The phenotype will then be parsed into a PonyGE2 individual using the GE LR Parser.
3. (`example_3.txt` in `seeds/example_pop`) Target genotypes can be saved as a simple text file with a single header of "`Genotype:`", followed by the genotype itself on the following line. The genotype will then be mapped into a PonyGE2 individual using the normal GE mapping process. As with option 1 above, this will result in an identical PonyGE2 individual being re-created from the specified genome.
4. (`example_4.txt` in `seeds/example_pop`) Target phenotypes can be saved as a simple text file where the ***only*** content of the file is the phenotype string itself (i.e. no descriptive text, headers, comments, etc). The content of these files will then be parsed into PonyGE2 individuals using the GE LR Parser.

***NOTE** that the names of individual files contained in a specified target population folder in the `seeds` directory **do not matter**. These files can be named however so desired.*

***NOTE** that as with the GE LR Parser described above, a compatible grammar file needs to be specified along with the target `seeds` folder. If the target string cannot be parsed using the specified grammar, an error will occur. If the target genotype results in a different phenotype to that specified, an error will occur.*

***NOTE** that at present, phenotypes spanning multiple lines can only be parsed correctly using file format 4 above, i.e. the phenotype string constitutes the sole information in the file. If a genotype exists for such phenotypes, best practice is to use the genotype to seed the solution using file format 3 above, i.e. discard the phenotype string and allow the genotype to re-produce it.*

Initialisation

All standard [[initialisation|Initialisation]] techniques in PonyGE2 are compatible with seeding evolutionary runs with target individuals. When seeding a run with target solutions in conjunction with a known initialisation technique (e.g. [[Ramped Half-Half|Initialisation#ramped-half-half]]), the initial population size generated by the specified initialisation technique is reduced from the original specified `POPULATION_SIZE` parameter by the total number of seed individuals. This ensures that the total size of the initial generation matches the specified `POPULATION_SIZE` parameter.

An additional initialisation option is also included in PonyGE2 which may be of some use in the case of Genetic Improvement. An option is available to initialise the entire population with only identical copies of the specified seed individual (or individuals). If only one target seed is specified, the initial population will consist of `POPULATION_SIZE` copies of that individual. If multiple target seeds are specified, the initial population will consist of equal amounts of copies of each specified seed. This option can be specified with the argument:

```
--initialisation seed_individuals
```

or by setting the parameter `INITIALISATION` to `seed_individuals` in either a parameters file or in the params dictionary.

Examples

An example parameters file for seeding runs with a number of individuals has been included in the parameters folder under `seed_run_target.txt`. An example folder named `example_pop` with a range of compatible formatting types for seeding target solutions is included in the `seeds` directory.

Re-creating PonyGE2 runs

It is possible to set the random seeds for the various random number generators (RNGs) used by PonyGE2 in order to exactly re-create any given evolutionary run. All PonyGE2 runs by default save their random seeds. By simply specifying the argument:

```
--random_seed [RANDOM_SEED]
```

or by setting the parameter `RANDOM_SEED` to `[RANDOM_SEED]` in either a parameters file or in the params dictionary, where `[RANDOM_SEED]` is an integer which specifies the desired random seed.

At present, the main branch of PonyGE2 only uses two RNGs:

1. The core Python `random` module, and
2. The numpy `np.random` module.

Both of these RNGs are set using the same seed. When the `RANDOM_SEED` parameter is set, and provided the grammar, fitness function, and all parameters remain unchanged, then PonyGE2 will produce identical results to any previous run executed using this random seed.