Bloat occurs in evolutionary algorithms when large increases in genetic material are observed without an observed increase in fitness. There are currently three methods implemented to control genetic bloat in PonyGE2:

1. Limiting the maximum derivation tree depth
2. Limiting the number of nodes in a derivation tree
3. Limiting the maximum length of the genome.

Any combination of these three methods can be used with PonyGE2. Furthermore, these bloat control measures are **not** limited to specific representation types; it is possible to use genome length limitation with derivation tree based operators, and vice versa.

***NOTE** that these techniques may not be necessary with all setups. These should be implemented if and when genetic bloat is observed.*

Max Tree Depth

By default the maximum depth limit for a derivation tree is set to 90. This limit is due to Python's `eval()` stack limit which is set from 92-99 (depending on the Python version used), but grammar design also plays a large part in stack and memory overflow errors. Such a high depth limit can lead to genetic bloat, dramatically slowing down the overall evolutionary process. One way to prevent this is to specify a global maximum tree depth with the argument:

```
--max_tree_depth [INT]
```

or by setting the parameter `MAX_TREE_DEPTH` to `[INT]` in either a parameters file or in the params dictionary, where `[INT]` is an integer which specifies the desired maximum depth limit for derivation trees.

***NOTE** that setting the parameter `MAX_TREE_DEPTH` or argument `--max_tree_depth` to 0 is the same as setting no maximum tree depth, i.e. trees will be allowed to grow in an un-controlled manner. This may cause memory errors and cause PonyGE2 to crash.*

***NOTE** that the parameter `MAX_TREE_DEPTH` is distinct from the parameter `MAX_INIT_TREE_DEPTH` The former sets the global limit across the entire evolutionary process, while the latter is used solely to control derivation tree depth during derivation tree-based initialisation. **NOTE** also that `MAX_TREE_DEPTH` >= `MAX_INIT_TREE_DEPTH`.*

Max Tree Nodes

By default there are no limits to the maximum number of nodes a derivation tree can have. This can lead to genetic bloat, dramatically slowing down the overall evolutionary process. One way to prevent this is to specify a global maximum number of derivation tree nodes with the argument:

```
--max_tree_nodes [INT]
```

or by setting the parameter `MAX_TREE_NODES` to `[INT]` in either a parameters file or in the params dictionary, where `[INT]` is an integer which specifies the desired maximum number of nodes for derivation trees.

***NOTE** that setting the parameter `MAX_TREE_NODES` or argument `--max_tree_nodes` to 0 is the same as setting no limit on the maximum number of nodes a derivation tree can have, i.e. trees will be allowed to grow in an un-controlled manner.*

Max Genome Length

By default there are no limits to the maximum length a genome can take. This can lead to genetic bloat, dramatically slowing down the overall evolutionary process. One way to prevent this is to specify a global maximum genome length with the argument:

```
--max_genome_length [INT]
```

or by setting the parameter `MAX_GENOME_LENGTH` to `[INT]` in either a parameters file or in the params dictionary, where `[INT]` is an integer which specifies the desired maximum global genome length.

***NOTE** that setting the parameter `MAX_GENOME_LENGTH` or argument `--max_genome_length` to 0 is the same as setting no limit to the lengths of genomes, i.e. genomes will be allowed to grow in an un-controlled manner.*

***NOTE** that the parameter `MAX_GENOME_LENGTH` is distinct from the parameter `MAX_INIT_GENOME_LENGTH`, which is used solely to control genome size during genome-based initialisation.*