**Deep Artificial Neural Network Optimization - Project Report**

Course: Computational Intelligence for Optimization

Authors: Joris Bertens, Antonio Curado, Lukas Früchtnicht, Alexandra Baur

1. **Introduction**
   1. **Objective of the project**

The objective is the optimization of the weights of the 64.000 connection in the given artificial neural network using optimization algorithms discussed during the semester. In the end, the artificial neural network should be able to recognize unseen handwritten digits.

This goal should be achieved by benchmarking different algorithms and parameters (especially variations of the Genetic Algorithm).

* 1. **Way of proceeding**

The procedure throughout the project was the following:

Test other algo-rithms

Research

Get best baseline para-meters

Fine tuning of best solution

Test selection methods

Test crossover methods

Test mutation methods

Test different GA´s

1. **Research**

During the research the following methods and algorithms were found and implemented and/or tested:

|  |  |  |
| --- | --- | --- |
| **Genetic Algorithm** | **Selection** | Tournament selection |
| Roulette-Wheel selection |
| Rank selection |
| Boltzmann selection |
| Random selection |
| Best selection |
| **Crossover** | One-point crossover |
| Two-point crossover |
| Uniform-swap |
| Arithmetic crossover |
| Random crossover |
| **Mutation** | Ball mutation |
| Ball mutation with boundaries |
| Random member mutation |
| Swap mutation |
| Decreasing mutation rate |
| **Other** | Elitism |
| Worst removal |
| Mating pool |
| Measuring phenotype/genotype |
| Two populations |
| **Particle Swarm Optimization** |  |  |
| **Simulated Annealing?** |  |  |
| **Hill Climbing?** |  |  |

1. **Exploration of Parameters of the given Algorithms**

The first step in order to optimize the network, is to explore and test different parameters on the existing algorithms.

Every algorithm was tested with 5 seeds and the unseen accuracy was measured by the average value over the 5 seeds:

* 1. **Hill Climbing**

The Hill Climbing algorithm was tested with various combinations of parameter settings to improve the performance:

* Neighborhood size: from 5 to 30 in steps of 5
* Generations: from 100 to 250 in steps of 50

The best unseen accuracy achieved using the Hill Climbing algorithm was … with the following settings:

* Neighborhood size:
* Generations:

Regarding the size of the search space, the Hill Climbing algorithm might not be the best solution due to the fact that it gets stuck in local optima.

* 1. **Simulated Annealing**

The Simulated Annealing algorithm was tested with various combinations of parameter settings to improve the performance. Generations were chosen no larger than 250 to maintain a useful neighborhood size to explore. The control parameter settings were chosen from 1 to 3 to explore how much probability of selecting individuals with a worse fitness will influence the results. The radius parameter is necessary for the ball mutation that was used for the neighborhood. The variance in the update rate parameter was chosen to explore if a slower decrease of the control parameter will influence the results. (Sources):

* Generations: from 100 to 250 in steps of 50
* Control: from 1 to 3 in steps of 1
* Radius: from 0.005 and 0.015 in steps of 0.005
* Update rate: from 0.7 to 0.9 in steps of 0.1

The best unseen accuracy achieved using the Simulated Annealing algorithm was … with the following settings:

* Generations:
* Control:
* Radius:
* Update rate:
  1. **Genetic Algorithm**

The Genetic algorithm was tested with various combinations of parameter settings to improve the performance and to get a benchmark baseline. The number of generations (and population size) was chosen to see if a higher number of generations or a bigger population size are better for the performance of the algorithm. For crossover probability all the possibilities were tested (from very low to always). Mutation probability was only tested to 0.5, because in the literature a low mutation rate is recommended and to reduce computational effort. For radius and pressure all possibilities were tested. (Sources)

The steps sizes were chosen to reduce the computational effort.

* Generations: from 100 to 260 in steps of 20
* Population size: 5000/number of generations (to keep up with the restrictions)
* Crossover probability: from 0.2 to 1.0 in steps of 0.2
* Mutation probability: from 0.1 to 0.5 in steps of 0.1
* Radius: from 0.2 to 1.0 in steps of 0.2
* Pressure (tournament): from 0.2 to 1.0 in steps of 0.2

The best unseen accuracy achieved using the Genetic algorithm was … with the following settings:

* Generations: 180
* Population size: 27
* Crossover probability: 1.0
* Mutation probability: 0.5
* Radius: 0.6
* Pressure (tournament): 0.8

The best parameter settings found were already improving the algorithm:

Fancy chart.

1. **Selection methods**

To improve the genetic algorithm with the best parameters found so far, various selection algorithms were applied and tested:

1. Best-or-random-selection:

Blabala…(Explanation/Sources)

1. The roulette-wheel-selection:

This selection algorithm selects individuals to be parents by adding up all the fitness values of the individuals in the population and then chooses a random number between 0 and the sum of all fitness values. One the one hand, individuals that have a higher fitness are more likely to be chosen, on the other hand, it is not impossible for the individual with the lowest fitness to become a parent. (Sources)

1. Tournament Selection:

The tournament selection chooses random individuals of the population into a tournament pool, where the fittest individual will be selected to be a parent. A single individual can be chosen into the tournament pool multiple times. The parameter for the tournament selection is the tournament size which is represented by the pressure. (Sources)

1. Best Selection:

Blabalabla… (Explanation/Sources)

1. Random Selection:

This selection method chooses random individuals of the population to be parents.(Sources)

1. Best-x-Selection:

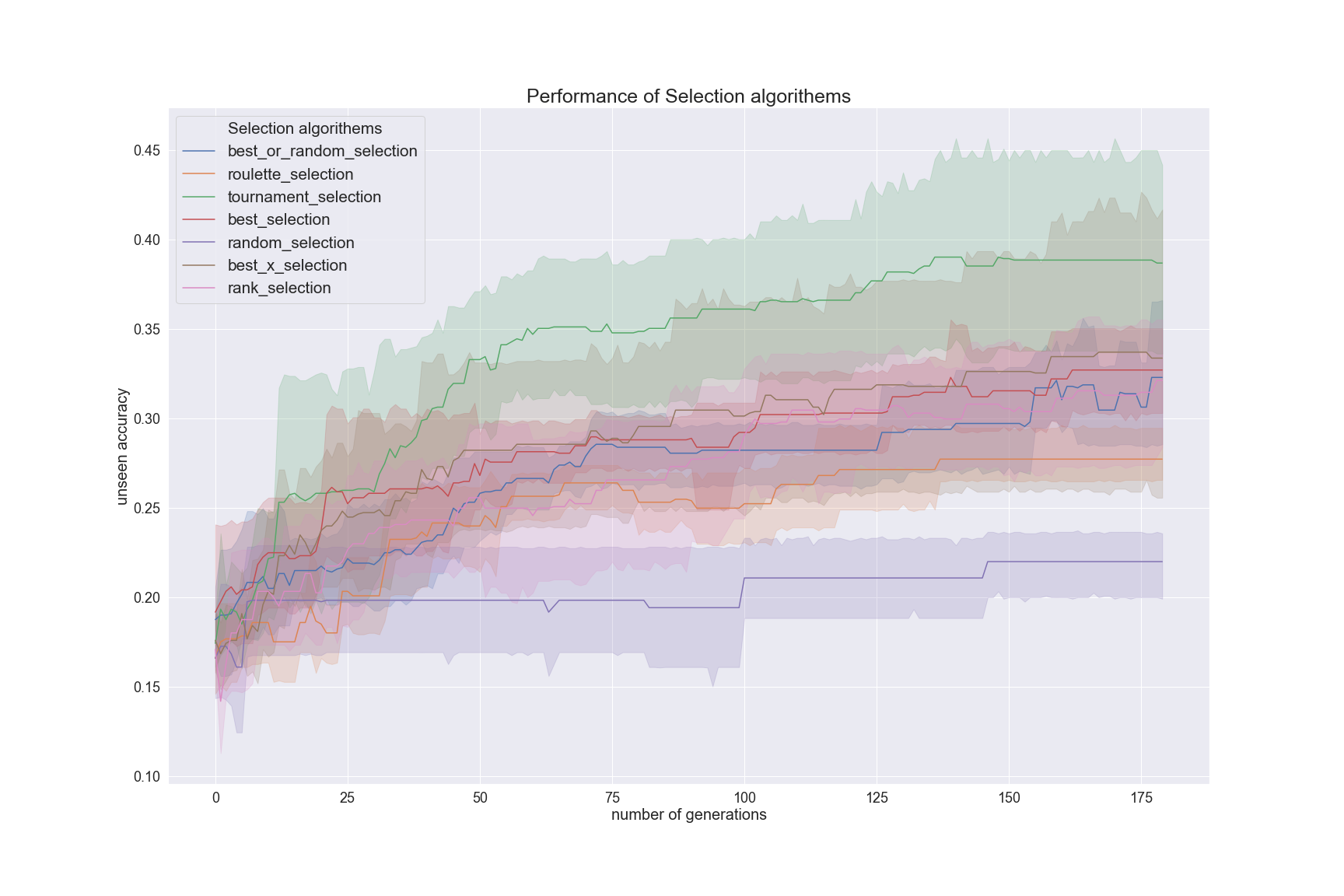
Blabla…(Explanation/Sources)

1. Rank Selection:

This selection algorithm sorts the individuals in the population by their fitness and assigns a rank to them starting with rank 1 for the worst fitness. It will calculate a new fitness proportion for the individual considering their rank. Afterwards, a roulette wheel selection with the new fitness values is performed, so the individuals with the highest fitness are more likely to be chosen, regardless their absolute distance to the ones with lower fitness. (Sources)

The following chart compares the impact of the different selection methods on the unseen accuracy. It can be said that random selection performes the worst and tournament selection performs best (0.38 unseen accuracy over 5 seeds).

The other algorithms show similar results between an average unseen accuracy of 0.27 and 0.33.

Figure 1: Comparision of impact of selection algorithms on unseen accuracy over generations 

1. **Crossover methods**

To improve the genetic algorithm with the best parameters found so far, various crossover methods were applied and tested:

1. Two-point-crossover:

Two random points are chosen to cut the individuals and the parts are shuffled to create the offsprings.

1. One-point-crossover:

One random point is chosen to cut the individuals and the parts are shuffled to create the offsprings.

1. Geometric Crossover:

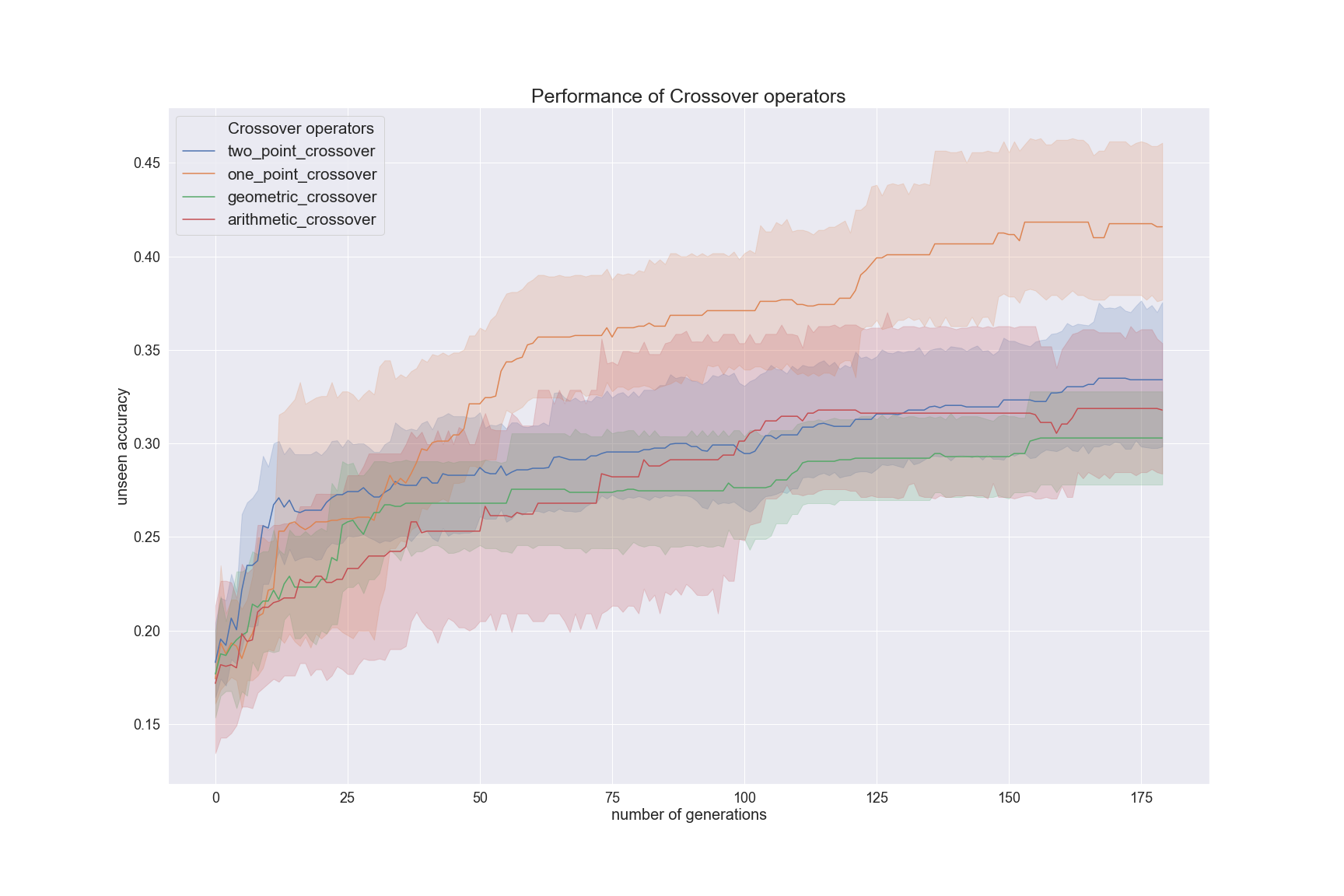
Blabla…

1. Arithmetic Crossover:

Blabla…

The chart shows that one-point-crossover performs best with an average unseen accuracy of 0.42. The other crossover methods are achieving a similar unseen accuracy between 0.3 and 0.34.

Figure 2: Comparision of impact of crossover methods on unseen accuracy over generations



1. **Mutation methods**

To improve the genetic algorithm with the best parameters found so far, various crossover methods were applied and tested:

1. Ball Mutation:

Blabla…

1. Random-member-mutation:

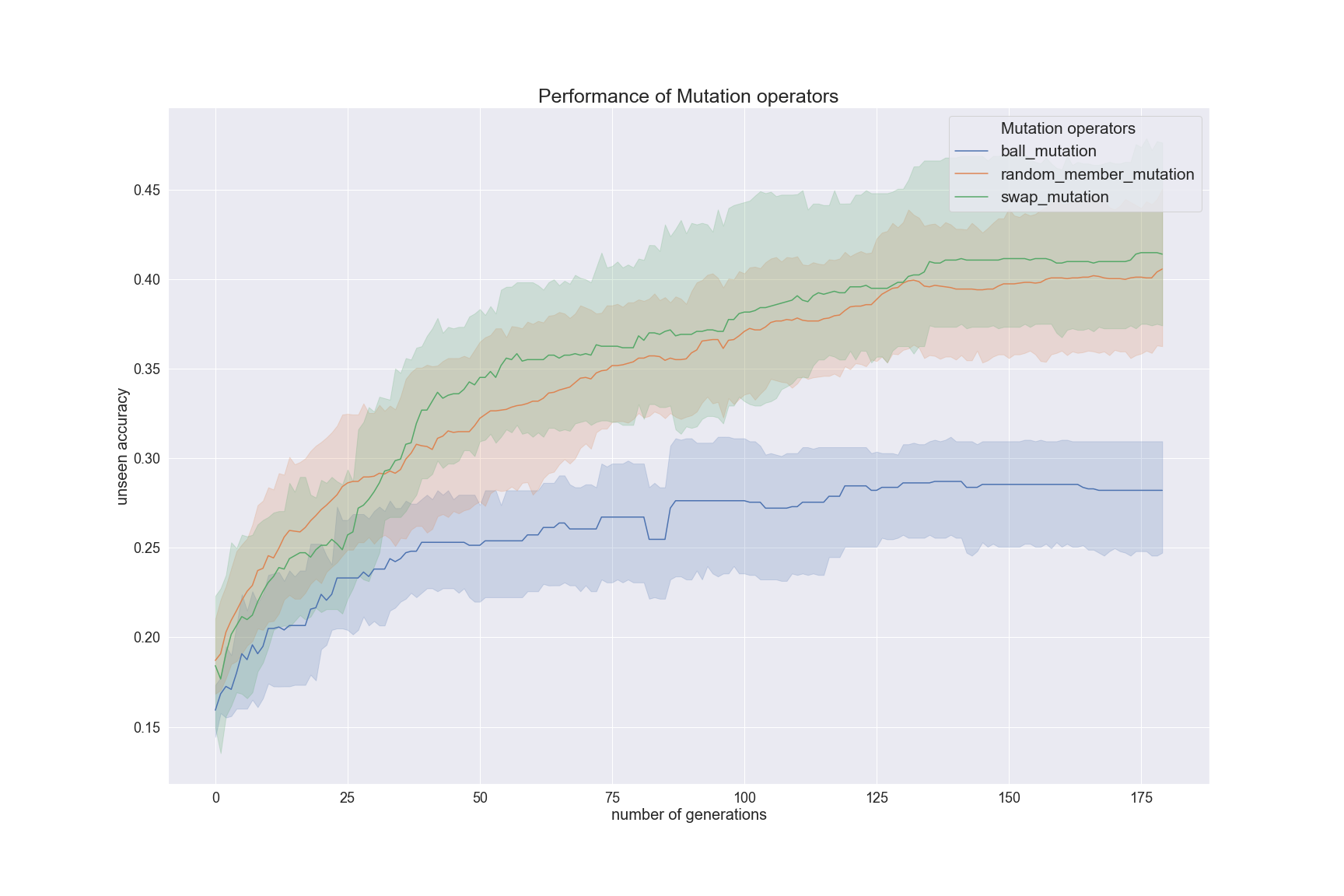
A random position of the individual is chosen and replaced by a random (possible) value.

1. Swap mutation:

Blbla…

The chart shows that ball mutation performs worst with an unseen accuracy of 0.28 over 180 generations. The random member mutation and the swap mutation are performing similar with achieving an unseen accuracy between 0.41 and 0.42.

Figure 3: Comparision of impact of mutation methods on unseen accuracy over generations



1. **Other**

Other genetic algorithms used

1. **Analysis of Combinations**

Best overall combination for genetic algorithm

1. **Other algorithms(PSO)**
   1. **Particle Swarm Optimization**

The Particle Swarm Optimization algorithm was tested with various combinations of parameter settings to improve the performance. The number of generations (and population size) was chosen to see if a higher number of generations or a bigger population size are better for the performance of the algorithm. The social and the cognitive component and inertia were chosen as recommended by literature (Leonardo´s book).

* Generations: from 100 to 250 in steps of 50
* Swarm size: 5000/number of generations (to keep up with the restrictions)
* Social component: from 1.0 to 2.0 in steps of 0.5
* Cognitive component: from 1.0 to 2.0 in steps of 0.5
* Inertia: from 0.8 to 1.2 in steps of 0.4

The best unseen accuracy achieved using the Simulated Annealing algorithm was 0.39 with the following settings:

* Generations: 150
* Swarm size: 33
* Social component: 1.0
* Cognitive component: 1.0
* Inertia: 0.8

1. **Fine Tuning**

Decision for best algorithm (with best combination and best parameters) because of best unseen accuracy.

Fine tuning applied in order to get even better results:

1. **Conclusion**
2. **References**