1. GA

Evolutionary Algorithm – optimization algorithm using the insights gained from evolution, including selection and reproduction

- Genetic Algorithm – the genome is a binary string (optimization variables).
- Evolutionary Strategy – genome is an array of floats, there are often strategy parameters that control the size of the mutation

Elements.

(1) representation: how are the optimization variables represented as a “genome”?
(2) How long does an individual survive (i.e. just 1 generation or more?)
(3) How is selection implemented?
(4) How is reproduction implemented?
   (a) How many parents per child? (and vice versa?)
   (b) Is there crossover? If so, what kind of crossover?
   (c) How is the mutation implemented?
Algorithm. For $\mu$ parents and $\lambda$ children

1. Generate $\mu$ viable individuals (i.e. individuals with finite cost). Store them in $P$ (which we will also call $G^0$). Set the generation counter $g$ to 1.
2. For $i$ in $1 : \lambda$:
   Create $G^g_i$:
   (a) Select 2 individuals from $P$ (possibly based on fitness of $P$)
   (b) Combine (with crossover) the genomes of the 2 parents to fill in the values of $G^g_i$.
   (c) Mutate $G^g_i$.
   (d) Evaluate fitness of $G^g_i$. If the fitness is infinite, repeat the process (i.e. goto (2a)) until we find a viable individual.
   (e) Store the fitness as $C^g_i$.
3. Assign $P$:
   - If $\mu == \lambda$, $P = G^g$
   - else $P = $ best members of $G^g$ (as a function of $C^g$)
4. $g = g + 1$
5. If not converged then go to (2)

Specifics for the Evolutionary Strategy.

- $\lambda$ should be $5\mu$.
- The parents of generation $g$ should be the $\mu$ fittest children from generation $g - 1$.
- Selection: randomly select two parents for each child.
- Uniform cross-over: When creating a child, we loop through the parameters and randomly choose which of our two parents to take the value from.
- Mutation: For each parameter, choose a value from a Gaussian distribution centered at its unmutated value, with a standard deviation of 5% of its unmutated value.