Genetic Algorithms

Dr. Mahmoud Nabil Mahmoud
mnmahmoud@ncat.edu

North Carolina A & T State University

March 24, 2021

Outline

Variants of Genetic Algorithms

- Genetic Algorithms
- Evolution Strategies
- Evolutionary Programming
- Genetic Programming

Algorithm	**Chromosome Representation**	**Crossover**	**Mutation**
Genetic Algorithm (GA)	Array	X	X
Genetic Programming (GP)	Tree	X	X
Evolution Strategies (ES)	Array	(X)	X
Evolutionary Programming (EP)	No constraints	-	X

Outline

Evolution Strategies

- Developed: Germany in the 1960s by Rechenberg and Schwefel
- Typically applied to numerical optimisation
- Attributed features:
  - Fast
  - good optimizer for real-valued optimization
  - Referred as real valued GA
  - relatively much theory
- Special:
  - self-adaptation of (mutation) parameters standard

Example

A good example is the satellite dish holder boom.

- The design is encoded as a series of angles and spar lengths.
- All alleles are real values
- The resulting structure by GA is 20,000% (!) better than traditional shapes, but for humans it looks very strange: it exhibits no symmetry,

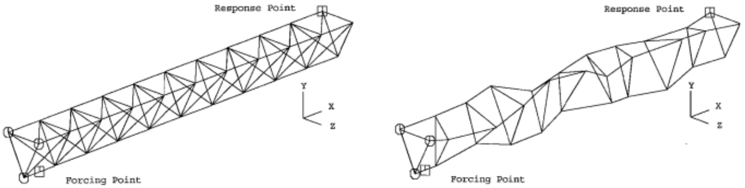


Fig. 2.4. The initial, regular design of the 3D boom (*left*) and the final design found by a genetic algorithm (*right*)

Real Valued Chromosome

How Chromosome j in the population look?

$$x^t(j) = \langle x_1^t, \dots, x_n^t \rangle$$

0.3	0.2	0.3	0.2	0.3	0.2	0.3	0.2	0.3

where x_i^t is gene i value at the t^{th} generetion

1. How each allele can mutate?

Real Valued Chromosome

How Chromosome j in the population look?

$$x^t(j) = \langle x_1^t, \ldots, x_n^t \rangle$$

0.3	0.2	0.3	0.2	0.3	0.2	0.3	0.2	0.3

where x_i^t is gene i value at the t^{th} generetion

1. How each allele can mutate?
    - **Uniform Mutation:** the mutated values are drawn uniformly randomly from $[L_i, U_i]$ naive

Real Valued Chromosome

How Chromosome j in the population look?

$$x^t(j) = \langle x_1^t, \dots, x_n^t \rangle$$

0.3	0.2	0.3	0.2	0.3	0.2	0.3	0.2	0.3

where x_i^t is gene i value at the t^{th} generetion

1. How each allele can mutate?
   - **Uniform Mutation:** the mutated values are drawn uniformly randomly from $[L_i, U_i]$ naive
   - **Nonuniform Mutation:** This is achieved by adding to the current gene value an amount drawn randomly from a Gaussian distribution $\mathcal{N}(0, \sigma)$

Real Valued Chromosome

How Chromosome j in the population look?

$$x^t(j) = \langle x_1^t, \ldots, x_n^t \rangle$$

0.3	0.2	0.3	0.2	0.3	0.2	0.3	0.2	0.3

where x_i^t is gene i value at the t^{th} generetion

1. How each allele can mutate?
   - **Uniform Mutation:** the mutated values are drawn uniformly randomly from $[L_i, U_i]$ naive
   - **Nonuniform Mutation:** This is achieved by adding to the current gene value an amount drawn randomly from a Gaussian distribution $\mathcal{N}(0, \sigma)$
   - How to select σ? called **mutation step size**

Real Valued Chromosome

How Chromosome j in the population look?

$$x^t(j) = \langle x_1^t, \ldots, x_n^t \rangle$$

0.3	0.2	0.3	0.2	0.3	0.2	0.3	0.2	0.3

where x_i^t is gene i value at the t^{th} generetion

1. How each allele can mutate?
   - **Uniform Mutation:** the mutated values are drawn uniformly randomly from $[L_i, U_i]$ naive
   - **Nonuniform Mutation:** This is achieved by adding to the current gene value an amount drawn randomly from a Gaussian distribution $\mathcal{N}(0, \sigma)$
   - How to select σ? called **mutation step size**
2. How crossover can be done?

Real Valued Chromosome

How Chromosome j in the population look?

$$x^t(j) = \langle x_1^t, \ldots, x_n^t \rangle$$

0.3	0.2	0.3	0.2	0.3	0.2	0.3	0.2	0.3

where x_i^t is gene i value at the t^{th} generetion

1. How each allele can mutate?
   - **Uniform Mutation:** the mutated values are drawn uniformly randomly from $[L_i, U_i]$ naive
   - **Nonuniform Mutation:** This is achieved by adding to the current gene value an amount drawn randomly from a Gaussian distribution $\mathcal{N}(0, \sigma)$
   - How to select σ? called **mutation step size**
2. How crossover can be done?
   - Can we do simple split crossover?

Real Valued Chromosome

How Chromosome j in the population look?

$$x^t(j) = \langle x_1^t, \ldots, x_n^t \rangle$$

0.3	0.2	0.3	0.2	0.3	0.2	0.3	0.2	0.3

where x_i^t is gene i value at the t^{th} generetion

1. How each allele can mutate?
   - **Uniform Mutation:** the mutated values are drawn uniformly randomly from $[L_i, U_i]$ naive
   - **Nonuniform Mutation:** This is achieved by adding to the current gene value an amount drawn randomly from a Gaussian distribution $\mathcal{N}(0, \sigma)$
   - How to select σ? called **mutation step size**
2. How crossover can be done?
   - Can we do simple split crossover?
     - No

$(\mu + \lambda)$ Evolution Strategy

1. Define μ: # parents, λ: # offspring

$(\mu + \lambda)$ Evolution Strategy

1. Define μ: # parents, λ: # offspring
2. Choose initial population $P = \{x(1), \ldots, x(\mu)\}$ and mutability $\sigma > 0$

$(\mu + \lambda)$ Evolution Strategy

1. Define μ: # parents, λ: # offspring
2. Choose initial population $P = \{x(1), \ldots, x(\mu)\}$ and mutability $\sigma > 0$
3. For every generation $k = 0, 1, \ldots$

$(\mu + \lambda)$ Evolution Strategy

1. Define μ: # parents, λ: # offspring
2. Choose initial population $P = \{x(1), \ldots, x(\mu)\}$ and mutability $\sigma > 0$
3. For every generation $k = 0, 1, \ldots$
   - Generate λ offsprings using mutation/recombination as follows:

$(\mu + \lambda)$ Evolution Strategy

1. Define μ: # parents, λ: # offspring
2. Choose initial population $P = \{x(1), \dots, x(\mu)\}$ and mutability $\sigma > 0$
3. For every generation $k = 0, 1, \dots$
   - Generate λ offsprings using mutation/recombination as follows:
     - Choose two parent randomly i.e., choose $j \in \{1, \dots, \mu\}$

$(\mu + \lambda)$ Evolution Strategy

1. Define μ: # parents, λ: # offspring
2. Choose initial population $P = \{x(1), \ldots, x(\mu)\}$ and mutability $\sigma > 0$
3. For every generation $k = 0, 1, \ldots$
   - Generate λ offsprings using mutation/recombination as follows:
     - Choose two parent randomly i.e., choose $j \in \{1, \ldots, \mu\}$
     - Mutate each parent $x(j)' = x(j) + \sigma z$ where $z \in \mathcal{N}(0, 1)^n$

$(\mu + \lambda)$ Evolution Strategy

1. Define μ: # parents, λ: # offspring
2. Choose initial population $P = \{x(1), \ldots, x(\mu)\}$ and mutability $\sigma > 0$
3. For every generation $k = 0, 1, \ldots$
    - Generate λ offsprings using mutation/recombination as follows:
        - Choose two parent randomly i.e., choose $j \in \{1, \ldots, \mu\}$
        - Mutate each parent $x(j)' = x(j) + \sigma z$ where $z \in \mathcal{N}(0, 1)^n$
        - Recombine the mutated parents to generate a child

$(\mu + \lambda)$ Evolution Strategy

1. Define μ: # parents, λ: # offspring
2. Choose initial population $P = \{x(1), \ldots, x(\mu)\}$ and mutability $\sigma > 0$
3. For every generation $k = 0, 1, \ldots$
   - Generate λ offsprings using mutation/recombination as follows:
     - Choose two parent randomly i.e., choose $j \in \{1, \ldots, \mu\}$
     - Mutate each parent $x(j)' = x(j) + \sigma z$ where $z \in \mathcal{N}(0,1)^n$
     - Recombine the mutated parents to generate a child
   - Selection: Choose $P \subset \hat{P} := \{x(1), \ldots, x(\mu), x'(1), \ldots, x'(\lambda)\}$, $|P| = \mu$ such that

$$min\{f(x): \quad x \in P\} >= max\{f(x): \quad x \in \hat{P} \backslash P\}$$

$(\mu + \lambda)$ Evolution Strategy

1. Define μ: # parents, λ: # offspring
2. Choose initial population $P = \{x(1), \ldots, x(\mu)\}$ and mutability $\sigma > 0$
3. For every generation $k = 0, 1, \ldots$
   - Generate λ offsprings using mutation/recombination as follows:
     - Choose two parent randomly i.e., choose $j \in \{1, \ldots, \mu\}$
     - Mutate each parent $x(j)' = x(j) + \sigma z$ where $z \in \mathcal{N}(0, 1)^n$
     - Recombine the mutated parents to generate a child
   - Selection: Choose $P \subset \hat{P} := \{x(1), \ldots, x(\mu), x'(1), \ldots, x'(\lambda)\}$, $|P| = \mu$ such that

     $$min\{f(x) : \quad x \in P\} >= max\{f(x) : \quad x \in \hat{P} \backslash P\}$$
   - Stop if criteria is met

$(\mu + \lambda)$ Evolution Strategy

1. Define μ: # parents, λ: # offspring
2. Choose initial population $P = \{x(1), \ldots, x(\mu)\}$ and mutability $\sigma > 0$
3. For every generation $k = 0, 1, \ldots$
   - Generate λ offsprings using mutation/recombination as follows:
     - Choose two parent randomly i.e., choose $j \in \{1, \ldots, \mu\}$
     - Mutate each parent $x(j)' = x(j) + \sigma z$ where $z \in \mathcal{N}(0, 1)^n$
     - Recombine the mutated parents to generate a child
   - Selection: Choose $P \subset \hat{P} := \{x(1), \ldots, x(\mu), x'(1), \ldots, x'(\lambda)\}$, $|P| = \mu$ such that

     $$min\{f(x): \quad x \in P\} >= max\{f(x): \quad x \in \hat{P} \backslash P\}$$

   - Stop if criteria is met

$(\mu + \lambda)$ Evolution Strategy

1. Define μ: # parents, λ: # offspring
2. Choose initial population $P = \{x(1), \ldots, x(\mu)\}$ and mutability $\sigma > 0$
3. For every generation $k = 0, 1, \ldots$
   - Generate λ offsprings using mutation/recombination as follows:
     - Choose two parent randomly i.e., choose $j \in \{1, \ldots, \mu\}$
     - Mutate each parent $x(j)' = x(j) + \sigma z$ where $z \in \mathcal{N}(0,1)^n$
     - Recombine the mutated parents to generate a child
   - Selection: Choose $P \subset \hat{P} := \{x(1), \ldots, x(\mu), x'(1), \ldots, x'(\lambda)\}$, $|P| = \mu$ such that

     $$min\{f(x): \quad x \in P\} >= max\{f(x): \quad x \in \hat{P} \backslash P\}$$

   - Stop if criteria is met

How to determine the step size? The highlighted step will be modified!

Notes

- In ES, operators are done in the reverse order.
- Mutation is essential.
- Very difficult to determine the mutation step size σ manually.
- In literature, this problem is called mutation strategy parameter control or self-adaption
- In addition, all alleles are assumed to have the same step size.

Variant (μ, λ) ES

Changes in red

1. Define μ: # parents, λ: # offspring
2. $\mu < \lambda$
3. For every generation $k = 0, 1, \ldots$
   - Generate λ offsprings using mutation/recombination as follows:
     - ...
   - Selection: Choose $P \subset \hat{P} := \{x'(1), \ldots, x'(\lambda)\}$, $|P| = \mu$ such that
     $$min\{f(x): \quad x \in P\} >= max\{f(x): \quad x \in \hat{P} \backslash P\}$$
   - Stop if criteria is met

Self-adaptive Mutation: Step Size variation in the litreature

- **Self-adaptation** intend to adapt the step-size automatically without users' intervention.

Self-adaptive Mutation: Step Size variation in the litreature

- **Self-adaptation** intend to adapt the step-size automatically without users' intervention.
- **Self-adaptation** has been successfully demonstrated in many domains.

Self-adaptive Mutation: Step Size variation in the litreature

- **Self-adaptation** intend to adapt the step-size automatically without users' intervention.
- **Self-adaptation** has been successfully demonstrated in many domains.
- The key concept is that the mutation step size are not set by the user rather the σ coevolves with the Chromosomes.

Self-adaptive Mutation: Step Size variation in the litreature

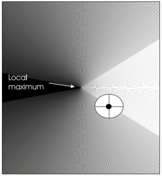
- **Self-adaptation** intend to adapt the step-size automatically without users' intervention.
- **Self-adaptation** has been successfully demonstrated in many domains.
- The key concept is that the mutation step size are not set by the user rather the σ coevolves with the Chromosomes.
- Three settings are considered

Self-adaptive Mutation: Step Size variation in the litreature

- **Self-adaptation** intend to adapt the step-size automatically without users' intervention.
- **Self-adaptation** has been successfully demonstrated in many domains.
- The key concept is that the mutation step size are not set by the user rather the σ coevolves with the Chromosomes.
- Three settings are considered

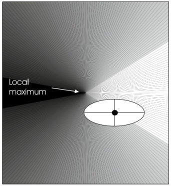
Self-adaptive Mutation: Step Size variation in the litreature

- **Self-adaptation** intend to adapt the step-size automatically without users' intervention.
- **Self-adaptation** has been successfully demonstrated in many domains.
- The key concept is that the mutation step size are not set by the user rather the σ coevolves with the Chromosomes.
- Three settings are considered



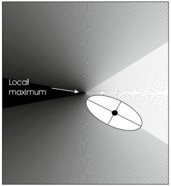
$x^t = \langle x_1^t, \ldots, x_n^t, \sigma \rangle$

Uncorrelated mutations, one step size



$x^t = \langle x_1^t, \ldots, x_n^t, \sigma_1^t, \ldots, \sigma_n^t \rangle$

Uncorrelated mutations, multiple step size



$x^t = \langle x_1^t, \ldots, x_n^t, \sigma_1^t, \ldots, \sigma_n^t, \alpha_1^t, \ldots, \alpha_{\frac{n(n-1)}{2}}^t \rangle$

Correlated mutations, multiple step size

Outline

Mutate σ first

- Net mutation effect: $x = \langle x_1, \ldots, x_n, \sigma \rangle \to x' = \langle x'_1, \ldots, x'_n, \sigma' \rangle$
- Order is important
  - first $\sigma \to \sigma'$
  - then $x \to x' = x + \sigma'.\mathcal{N}(0,1)$
- Rational is: two factors affect $\langle x', \sigma' \rangle$
  - x' is good if fitness $f(x')$
  - σ' is good if the created x' is good
- reversing mutation order this would not work

Mutation case 1: Uncorrelated mutation with one σ

- Chromosomes $x = \langle x_1, \ldots, x_n, \sigma \rangle$

Mutation case 1: Uncorrelated mutation with one σ

- Chromosomes $x = \langle x_1, \ldots, x_n, \sigma \rangle$
- $\sigma' = \sigma . e^{\tau . \mathcal{N}(0,1)}$

Mutation case 1: Uncorrelated mutation with one σ

- Chromosomes $x = \langle x_1, \ldots, x_n, \sigma \rangle$
- $\sigma' = \sigma . e^{\tau . \mathcal{N}(0,1)}$
- $x' = x + \sigma' . \mathcal{N}(0,1)^n$

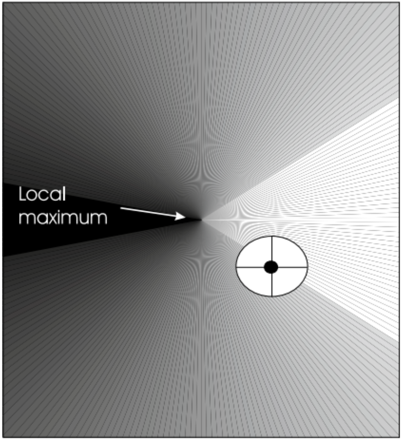
Mutation case 1: Uncorrelated mutation with one σ

- Chromosomes $x = \langle x_1, \ldots, x_n, \sigma \rangle$
- $\sigma' = \sigma.e^{\tau.\mathcal{N}(0,1)}$
- $x' = x + \sigma'.\mathcal{N}(0,1)^n$
- Typically the "learning rate" $\tau \propto \frac{1}{n^{0.5}}$

Mutation case 1: Uncorrelated mutation with one σ

- Chromosomes $x = \langle x_1, \ldots, x_n, \sigma \rangle$
- $\sigma' = \sigma.e^{\tau.\mathcal{N}(0,1)}$
- $x' = x + \sigma'.\mathcal{N}(0,1)^n$
- Typically the "learning rate" $\tau \propto \frac{1}{n^{0.5}}$
- And we have a boundary rule : if $\sigma' < \epsilon$ then $\sigma' = \epsilon$

Mutation case 1: Uncorrelated mutation with one σ



Mutation case 2: Uncorrelated mutation with n σ's

- Chromosomes $x^t = \langle x_1^t, \ldots, x_n^t, \sigma_1^t, \ldots, \sigma_n^t \rangle$

Mutation case 2: Uncorrelated mutation with n σ's

- Chromosomes $x^t = \langle x_1^t, \ldots, x_n^t, \sigma_1^t, \ldots, \sigma_n^t \rangle$
- $\sigma_i' = \sigma_i . e^{\tau' . \mathcal{N}(0,1) + \tau . \mathcal{N}(0,1)}$

Mutation case 2: Uncorrelated mutation with n σ's

- Chromosomes $x^t = \langle x_1^t, \ldots, x_n^t, \sigma_1^t, \ldots, \sigma_n^t \rangle$
- $\sigma_i' = \sigma_i . e^{\tau' . \mathcal{N}(0,1) + \tau . \mathcal{N}(0,1)}$
- $x_i' = x_i + \sigma_i' . \mathcal{N}(0, 1)$

Mutation case 2: Uncorrelated mutation with n σ's

- Chromosomes $x^t = \langle x_1^t, \ldots, x_n^t, \sigma_1^t, \ldots, \sigma_n^t \rangle$
- $\sigma_i' = \sigma_i . e^{\tau' . \mathcal{N}(0,1) + \tau . \mathcal{N}(0,1)}$
- $x_i' = x_i + \sigma_i' . \mathcal{N}(0,1)$
- Two learning rate parameters

Mutation case 2: Uncorrelated mutation with n σ's

- Chromosomes $x^t = \langle x_1^t, \ldots, x_n^t, \sigma_1^t, \ldots, \sigma_n^t \rangle$
- $\sigma_i' = \sigma_i . e^{\tau' . \mathcal{N}(0,1) + \tau . \mathcal{N}(0,1)}$
- $x_i' = x_i + \sigma_i' . \mathcal{N}(0, 1)$
- Two learning rate parameters
  - τ' overall learning rate

Mutation case 2: Uncorrelated mutation with n σ's

- Chromosomes $x^t = \langle x_1^t, \ldots, x_n^t, \sigma_1^t, \ldots, \sigma_n^t \rangle$
- $\sigma_i' = \sigma_i . e^{\tau' . \mathcal{N}(0,1) + \tau . \mathcal{N}(0,1)}$
- $x_i' = x_i + \sigma_i' . \mathcal{N}(0,1)$
- Two learning rate parameters
  - τ' overall learning rate
  - τ coordinate wise learning rate

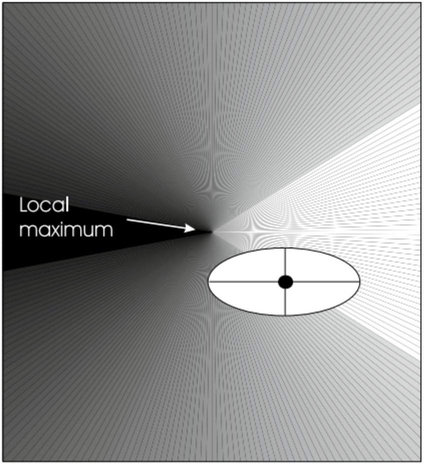
Mutation case 2: Uncorrelated mutation with n σ's

- Chromosomes $x^t = \langle x_1^t, \ldots, x_n^t, \sigma_1^t, \ldots, \sigma_n^t \rangle$
- $\sigma_i' = \sigma_i . e^{\tau' . \mathcal{N}(0,1) + \tau . \mathcal{N}(0,1)}$
- $x_i' = x_i + \sigma_i' . \mathcal{N}(0,1)$
- Two learning rate parameters
  - τ' overall learning rate
  - τ coordinate wise learning rate
- Typically $\tau' \propto \frac{1}{(2n)^{0.5}}$ and $\tau \propto \frac{1}{(2n^{0.5})^{0.5}}$

Mutation case 2: Uncorrelated mutation with n σ's

- Chromosomes $x^t = \langle x_1^t, \ldots, x_n^t, \sigma_1^t, \ldots, \sigma_n^t \rangle$
- $\sigma_i' = \sigma_i . e^{\tau' . \mathcal{N}(0,1) + \tau . \mathcal{N}(0,1)}$
- $x_i' = x_i + \sigma_i' . \mathcal{N}(0,1)$
- Two learning rate parameters
  - τ' overall learning rate
  - τ coordinate wise learning rate
- Typically $\tau' \propto \frac{1}{(2n)^{0.5}}$ and $\tau \propto \frac{1}{(2n^{0.5})^{0.5}}$
- And we have a boundary rule : if $\sigma_i' < \epsilon$ then $\sigma_i' = \epsilon$

Mutation case 2: Uncorrelated mutation with n σ's



Mutation case 3: Correlated mutations

- Chromosomes: $x^t = \langle x_1^t, \ldots, x_n^t, \sigma_1^t, \ldots, \sigma_n^t, \alpha_1^t, \ldots, \alpha_k^t \rangle$

Mutation case 3: Correlated mutations

- Chromosomes: $x^t = \langle x_1^t, \ldots, x_n^t, \sigma_1^t, \ldots, \sigma_n^t, \alpha_1^t, \ldots, \alpha_k^t \rangle$
- Where k $= \frac{n(n-1)}{2}$

Mutation case 3: Correlated mutations

- Chromosomes: $x^t = \langle x_1^t, \ldots, x_n^t, \sigma_1^t, \ldots, \sigma_n^t, \alpha_1^t, \ldots, \alpha_k^t \rangle$
- Where k $= \frac{n(n-1)}{2}$
- We define covariance matrix C as:

Mutation case 3: Correlated mutations

- Chromosomes: $x^t = \langle x_1^t, \ldots, x_n^t, \sigma_1^t, \ldots, \sigma_n^t, \alpha_1^t, \ldots, \alpha_k^t \rangle$
- Where k $= \frac{n(n-1)}{2}$
- We define covariance matrix C as:
  - $c_{ii} = \sigma_i^2$

Mutation case 3: Correlated mutations

- Chromosomes: $x^t = \langle x_1^t, \ldots, x_n^t, \sigma_1^t, \ldots, \sigma_n^t, \alpha_1^t, \ldots, \alpha_k^t \rangle$
- Where k $= \frac{n(n-1)}{2}$
- We define covariance matrix C as:
  - $c_{ii} = \sigma_i^2$
  - $c_{ij} = 0$ if i and j are not correlated.

Mutation case 3: Correlated mutations

- Chromosomes: $x^t = \langle x_1^t, \ldots, x_n^t, \sigma_1^t, \ldots, \sigma_n^t, \alpha_1^t, \ldots, \alpha_k^t \rangle$
- Where k $= \frac{n(n-1)}{2}$
- We define covariance matrix C as:
  - $c_{ii} = \sigma_i^2$
  - $c_{ij} = 0$ if i and j are not correlated.
  - $c_{ii} = 0.5(\sigma_i^2 - \sigma_j^2).\mathtt{tan}(2.\alpha_{ij})$ if i and j are correlated.

Mutation case 3: Correlated mutations

- Chromosomes: $x^t = \langle x_1^t, \ldots, x_n^t, \sigma_1^t, \ldots, \sigma_n^t, \alpha_1^t, \ldots, \alpha_k^t \rangle$
- Where k $= \frac{n(n-1)}{2}$
- We define covariance matrix C as:
  - $c_{ii} = \sigma_i^2$
  - $c_{ij} = 0$ if i and j are not correlated.
  - $c_{ii} = 0.5(\sigma_i^2 - \sigma_j^2).\mathtt{tan}(2.\alpha_{ij})$ if i and j are correlated.
- $\sigma_i' = \sigma_i.e^{\tau'.\mathcal{N}(0,1)+\tau.\mathcal{N}(0,1)}$

Mutation case 3: Correlated mutations

- Chromosomes: $x^t = \langle x_1^t, \ldots, x_n^t, \sigma_1^t, \ldots, \sigma_n^t, \alpha_1^t, \ldots, \alpha_k^t \rangle$
- Where k $= \frac{n(n-1)}{2}$
- We define covariance matrix C as:
  - $c_{ii} = \sigma_i^2$
  - $c_{ij} = 0$ if i and j are not correlated.
  - $c_{ii} = 0.5(\sigma_i^2 - \sigma_j^2).\mathtt{tan}(2.\alpha_{ij})$ if i and j are correlated.
- $\sigma_i' = \sigma_i.e^{\tau'.\mathcal{N}(0,1)+\tau.\mathcal{N}(0,1)}$
- $\alpha_j' = \alpha_j + \beta.\mathcal{N}(0,1)$

Mutation case 3: Correlated mutations

- Chromosomes: $x^t = \langle x_1^t, \ldots, x_n^t, \sigma_1^t, \ldots, \sigma_n^t, \alpha_1^t, \ldots, \alpha_k^t \rangle$
- Where k $= \frac{n(n-1)}{2}$
- We define covariance matrix C as:
  - $c_{ii} = \sigma_i^2$
  - $c_{ij} = 0$ if i and j are not correlated.
  - $c_{ii} = 0.5(\sigma_i^2 - \sigma_j^2).\mathtt{tan}(2.\alpha_{ij})$ if i and j are correlated.
- $\sigma_i' = \sigma_i.e^{\tau'.\mathcal{N}(0,1)+\tau.\mathcal{N}(0,1)}$
- $\alpha_j' = \alpha_j + \beta.\mathcal{N}(0,1)$
- $x' = x + \mathcal{N}(0, C')$

Mutation case 3: Correlated mutations

- Chromosomes: $x^t = \langle x_1^t, \ldots, x_n^t, \sigma_1^t, \ldots, \sigma_n^t, \alpha_1^t, \ldots, \alpha_k^t \rangle$
- Where k $= \frac{n(n-1)}{2}$
- We define covariance matrix C as:
  - $c_{ii} = \sigma_i^2$
  - $c_{ij} = 0$ if i and j are not correlated.
  - $c_{ii} = 0.5(\sigma_i^2 - \sigma_j^2).\mathtt{tan}(2.\alpha_{ij})$ if i and j are correlated.
- $\sigma_i' = \sigma_i.e^{\tau'.\mathcal{N}(0,1)+\tau.\mathcal{N}(0,1)}$
- $\alpha_j' = \alpha_j + \beta.\mathcal{N}(0,1)$
- $x' = x + \mathcal{N}(0, C')$
  - C' is the covariance matrix C after mutation.

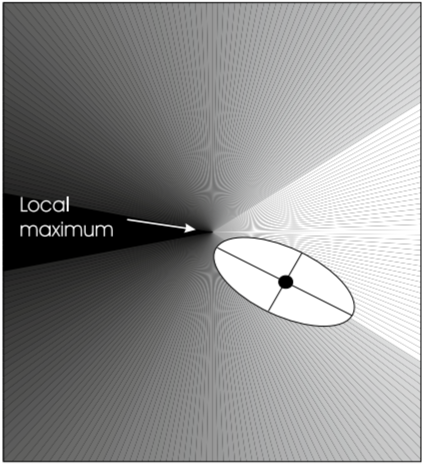
Mutation case 3: Correlated mutations

- Chromosomes: $x^t = \langle x_1^t, \ldots, x_n^t, \sigma_1^t, \ldots, \sigma_n^t, \alpha_1^t, \ldots, \alpha_k^t \rangle$
- Where k $= \frac{n(n-1)}{2}$
- We define covariance matrix C as:
  - $c_{ii} = \sigma_i^2$
  - $c_{ij} = 0$ if i and j are not correlated.
  - $c_{ii} = 0.5(\sigma_i^2 - \sigma_j^2).\mathtt{tan}(2.\alpha_{ij})$ if i and j are correlated.
- $\sigma_i' = \sigma_i.e^{\tau'.\mathcal{N}(0,1)+\tau.\mathcal{N}(0,1)}$
- $\alpha_j' = \alpha_j + \beta.\mathcal{N}(0,1)$
- $x' = x + \mathcal{N}(0, C')$
  - C' is the covariance matrix C after mutation.
- Typically $\tau' \propto \frac{1}{(2n)^{0.5}}$, $\tau \propto \frac{1}{(2n^{0.5})^{0.5}}$, and $\beta = 5°$.

Mutation case 3: Correlated mutations

- Chromosomes: $x^t = \langle x_1^t, \ldots, x_n^t, \sigma_1^t, \ldots, \sigma_n^t, \alpha_1^t, \ldots, \alpha_k^t \rangle$
- Where k $= \frac{n(n-1)}{2}$
- We define covariance matrix C as:
  - $c_{ii} = \sigma_i^2$
  - $c_{ij} = 0$ if i and j are not correlated.
  - $c_{ii} = 0.5(\sigma_i^2 - \sigma_j^2).\mathtt{tan}(2.\alpha_{ij})$ if i and j are correlated.
- $\sigma_i' = \sigma_i.e^{\tau'.\mathcal{N}(0,1)+\tau.\mathcal{N}(0,1)}$
- $\alpha_j' = \alpha_j + \beta.\mathcal{N}(0,1)$
- $x' = x + \mathcal{N}(0, C')$
  - C' is the covariance matrix C after mutation.
- Typically $\tau' \propto \frac{1}{(2n)^{0.5}}$, $\tau \propto \frac{1}{(2n^{0.5})^{0.5}}$, and $\beta = 5°$.
- And we have a boundary rule : if $\sigma_i' < \epsilon$ then $\sigma_i' = \epsilon$

Mutation case 3: Correlated mutations



Outline

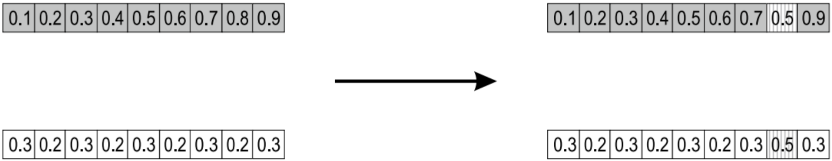
Recombination

- Creates one child $z = \langle z_1, \dots, z_n \rangle$
- Two parents can be selected randomly then recombine to generate a child
- OR, two parents can be selected randomly then recombine to generate a single gene value.

	Two fixed parents	**Two parents selected for each gene**
$z_i = (x_i + y_i)/2$	Local average	Global average
$z_i = (\alpha x_i + (1 - \alpha) y_i)/2$	Local arithmetic	Global arithmetic
$z_i =$ choose x_i or y_i randomly	Local discrete	Global discrete

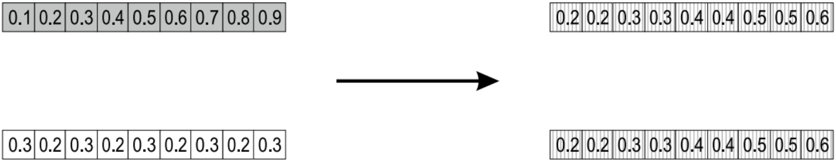
Types of arithmetic recombination

- **Single Arithmetic Recombination** Pick a random gene k. At that position, take the arithmetic average of the two parents.



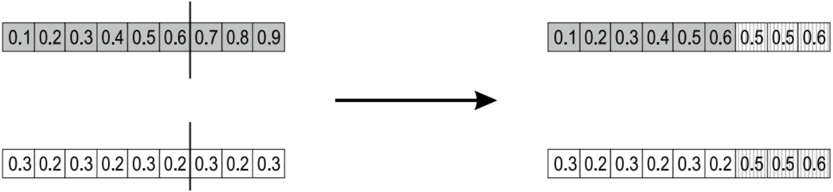
Types of arithmetic recombination

- **Whole Arithmetic Recombination** Take the weighted sum of the two parental values for each gene



Types of arithmetic recombination

- **Simple Arithmetic Recombination** First pick a recombination point k. Take the weighted sum of the two parental values for each gene starting from k.



Outline

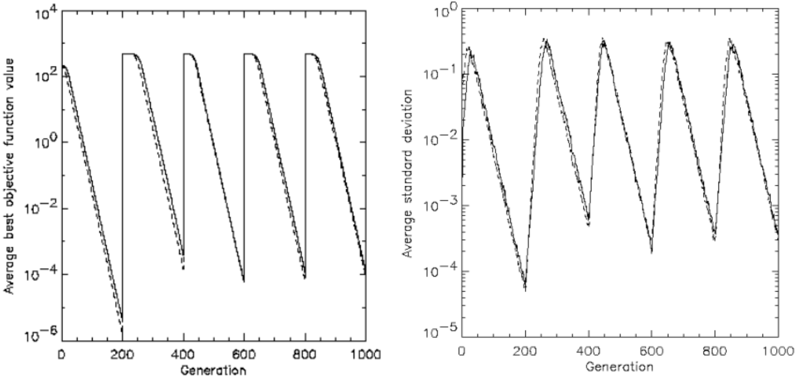
Self-adaptation illustrated

- Given a dynamically changing fitness landscape (optimum location shifted every 200 generations)
- Self-adaptive ES is able to
  - follow the optimum and
  - adjust the mutation step size after every shift !

Self-adaptation illustrated

- Given a dynamically changing fitness landscape (optimum location shifted every 200 generations)
- Self-adaptive ES is able to
  - follow the optimum and
  - adjust the mutation step size after every shift !

Self-adaptation illustrated cont.



Changes in the average best objective function values (left) and the mutation step sizes (right). The x-axis is the number of generations.

References

- Goldenberg, D.E., 1989. Genetic algorithms in search, optimization and machine learning.
- Michalewicz, Z., 2013. Genetic algorithms + data structures= evolution programs. Springer Science & Business Media

Questions