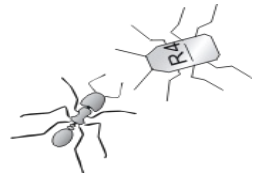


Evolutionary Strategies



Companion slides for the book *Bio-Inspired Artificial Intelligence: Theories, Methods, and Technologies* by Dario Floreano and Claudio Mattiussi, MIT Press

Evolutionary Strategies (ES)

Rechenberg, 1973

Genetic representation = Vector of real-valued n numbers

Population = fixed size

μ = number of selected parents

λ = number of individuals in the population

Selection = truncated rank selection

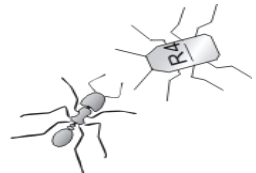
Two variants:

(μ, λ) = selected parents are replaced by their offspring

$(\mu + \lambda)$ = selected parents coexist with their offspring

Mutation = perturbations of all genes with normal probability density function

Crossover = not used

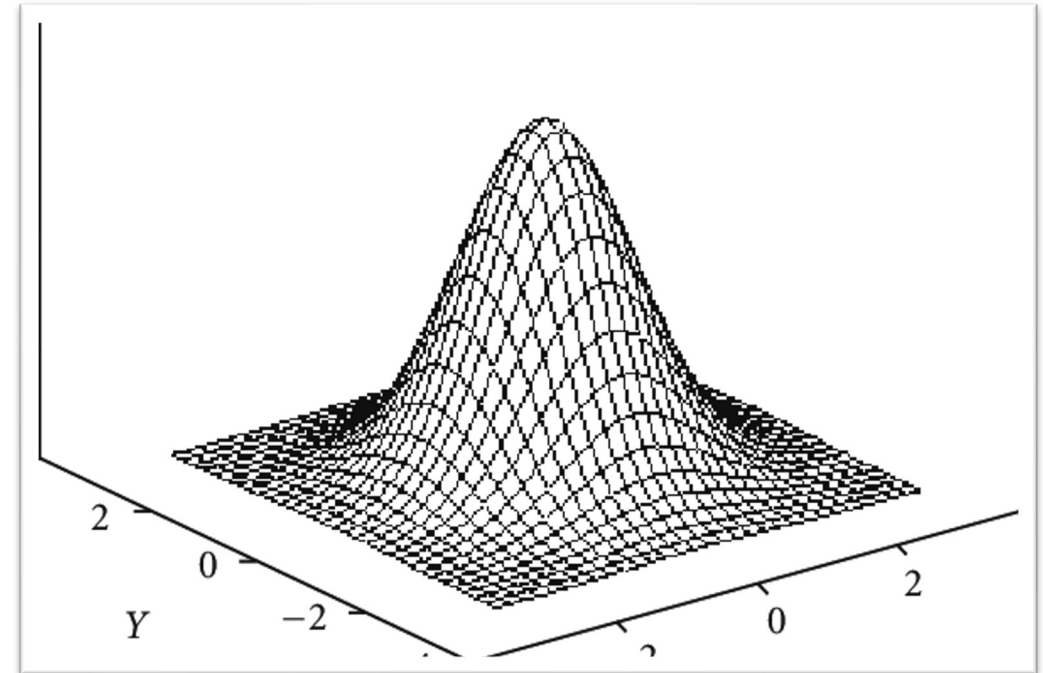
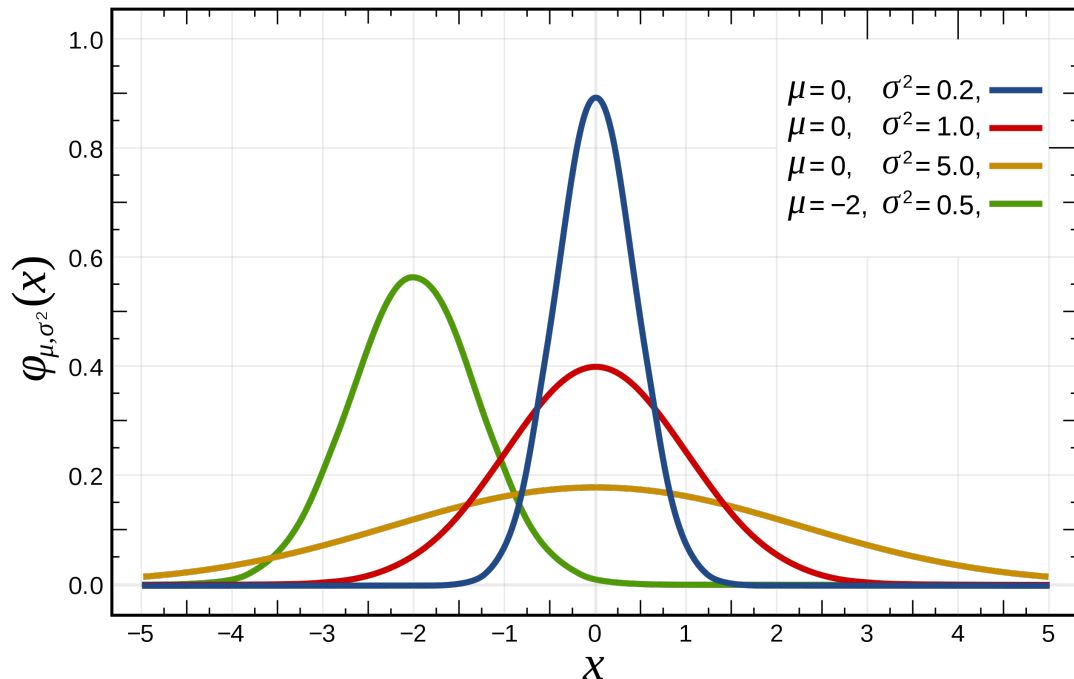


Mutations are Gaussian perturbations

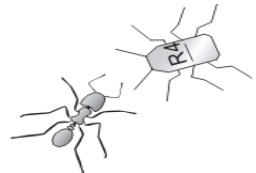
Mutate each gene x by adding a number sampled from a Normal distribution $N(0, \sigma^2)$

$$x' = x + N(0, \sigma^2) = x + \sigma N(0, 1)$$

For genetic strings of length $n > 1$ (e.g. $\langle x_1, x_2 \rangle$), we sample the Normal distribution $N(0, I)$, where I is the **Identity Matrix**



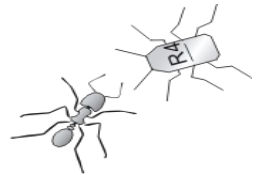
Standard deviation σ , *mutation step*, controls amount of change



ES (μ, λ) algorithm: Initialisation

1. Set up population size λ , number of parents μ , mutation step size σ
2. Set up individual \mathbf{m} (e.g., solution guess or at the distribution center)
3. Create population: generate λ offspring by adding mutations to \mathbf{m}

$$\mathbf{x}_i = \mathbf{m} + \sigma \mathbf{N}_i(0, \mathbf{I}) \quad \text{for } 0 < i \leq \lambda$$



ES (μ, λ) Algorithm: Selection and Reproduction

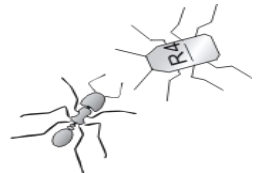
4. Evaluate λ offspring
5. Select μ parents with Truncated Rank Selection, e.g. top 25%
6. Update population mean using *fitness-weighted* values of μ parents

$$\mathbf{m} = \sum_{i=1}^{\mu} \mathbf{w}_i \mathbf{x}_i$$

where $w_1 \geq w_2 \geq w_3 \geq \dots \geq w_{\mu} \geq 0$
and $\sum_{i=1}^{\mu} w_i = 1$

7. Go to step 3 ($\mathbf{x}_i = \mathbf{m} + \sigma \mathbf{N}_i(0, \mathbf{I})$ for $0 < i \leq \lambda$)

Exercise: <https://machinelearningmastery.com/evolution-strategies-from-scratch-in-python/>



Co-evolution of mutation step

Mutation size σ can be added to genome and co-evolved $\langle x_1, \dots, x_n, \sigma \rangle$

$$\langle x, \sigma \rangle \rightarrow \langle x', \sigma' \rangle$$

Mutation order is important:

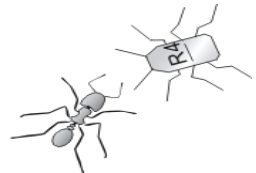
1. $\sigma \rightarrow \sigma'$
2. $x \rightarrow x' = x + \sigma' N(0,1)$

Rationale: quality of children $\langle x', \sigma' \rangle$ is evaluated twice

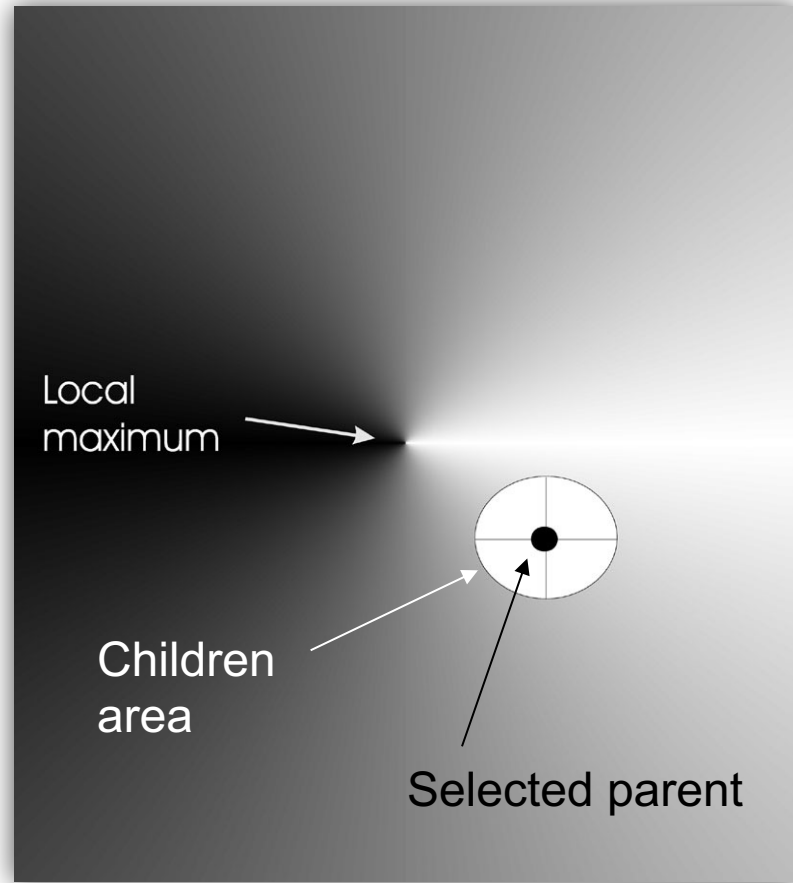
- Primary evaluation: x' is good if $f(x')$ is good
- Secondary evaluation: σ' is good if the x' it created is good
(reversing mutation order this would not work)

Mutations:

- $\sigma' = \sigma \cdot \exp(\tau N(0,1))$
where $\tau \propto 1/n^{1/2}$ where n = number of genes; boundary rule: if $\sigma' < \varepsilon_0 \Rightarrow \sigma' = \varepsilon_0$
- $x'_i = x_i + \sigma' N_i(0,1)$



One mutation step for all genes



$$\langle x_1, x_2, \sigma \rangle$$

$$\sigma' = \sigma \cdot \exp(\tau N(0,1))$$

$$x'_i = x_i + \sigma' N_i(0,1)$$

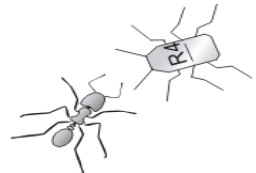
where I is the Identity Matrix

where $\tau \propto 1/n^{1/2}$

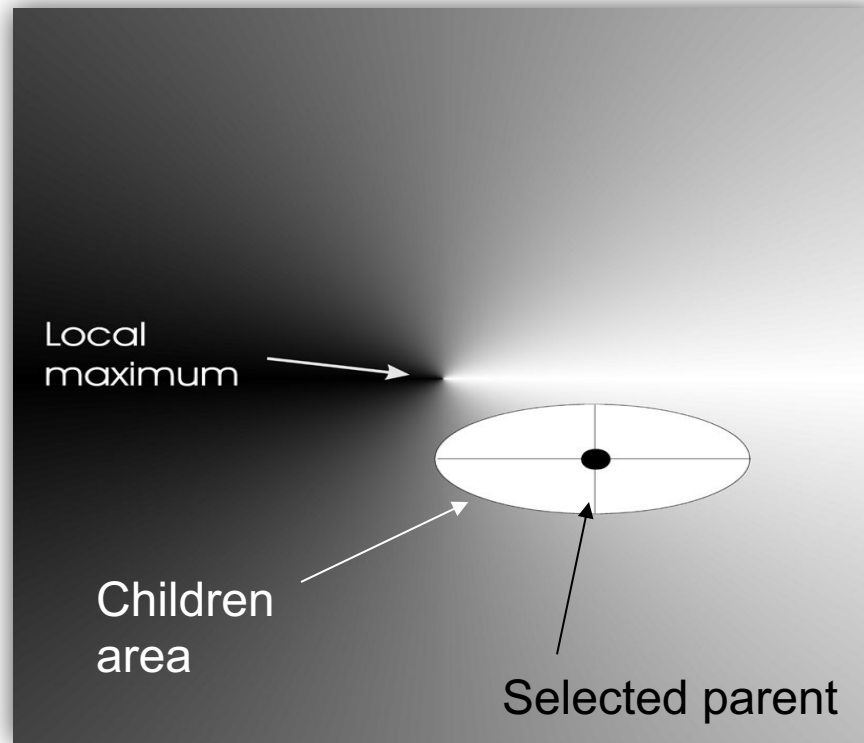
boundary rule $\sigma' < \varepsilon_0 \Rightarrow \sigma' = \varepsilon_0$

Adapted from: Eiben & Smith: <http://www.evolutionarycomputation.org/slides/>

Companion slides for the book *Bio-Inspired Artificial Intelligence: Theories, Methods, and Technologies* by Dario Floreano and Claudio Mattiussi, MIT Press



Individual and uncorrelated mutation steps



$$\langle x_1, x_2, \sigma_1, \sigma_2 \rangle$$

$$\sigma'_i = \sigma_i \cdot \exp(\tau' N(0,1) + \tau N_i(0,1))$$
$$x'_i = x_i + \sigma'_i N_i(0,1)$$

Two learning rate parameters:

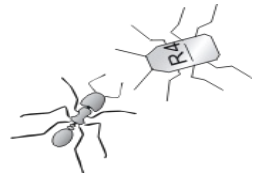
τ' overall learning rate

τ coordinate wise learning rate

$\tau' \propto 1/(2n)^{1/2}$ and $\tau \propto 1/(2n^{1/2})^{1/2}$

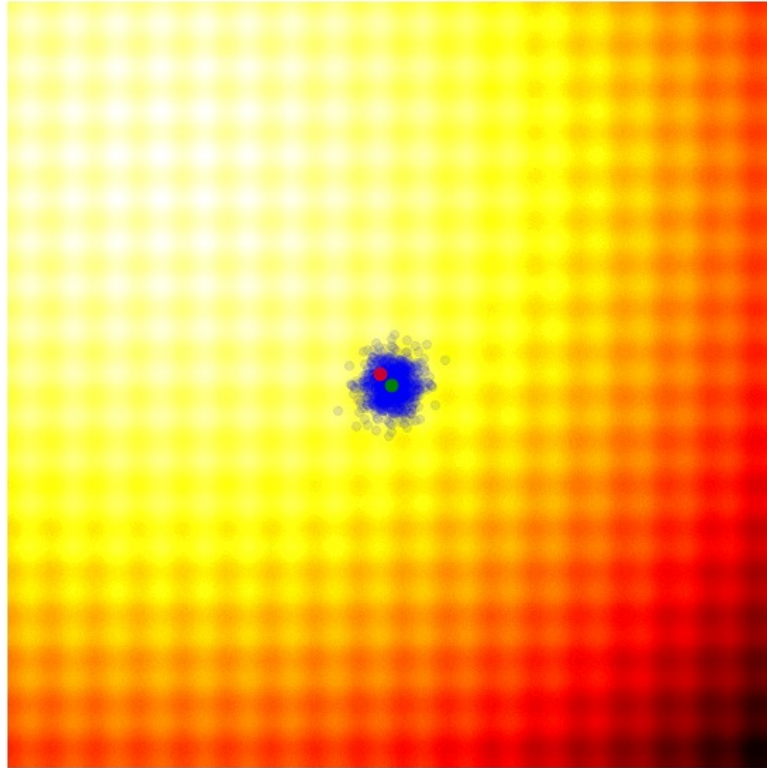
Boundary rule: $\sigma'_i < \varepsilon_0 \Rightarrow \sigma'_i = \varepsilon_0$

Adapted from: Eiben & Smith: <http://www.evolutionarycomputation.org/slides/>

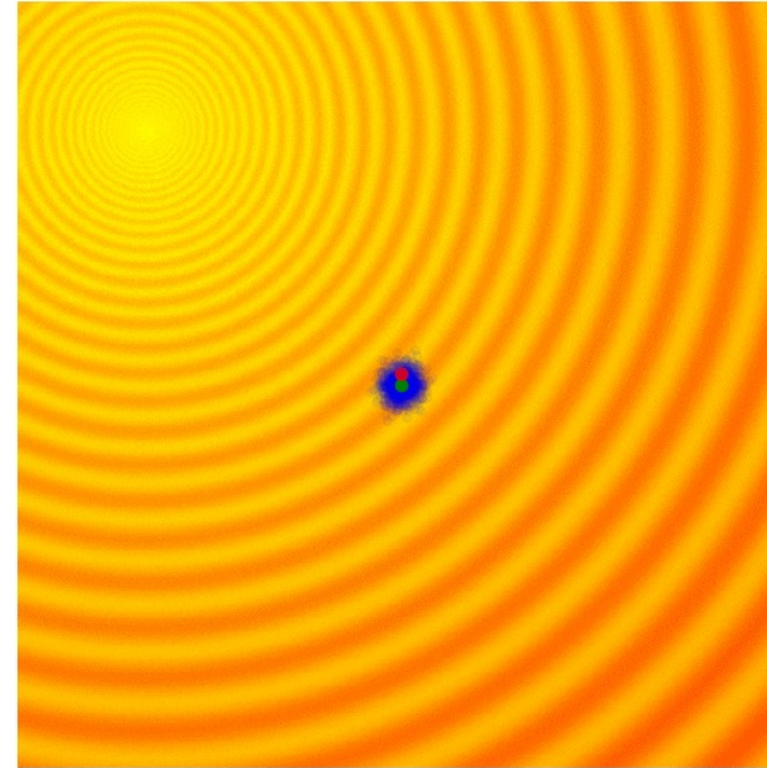


ES (μ, λ) algorithm with adaptive independent mutations

Shifted Schaffer-2D function



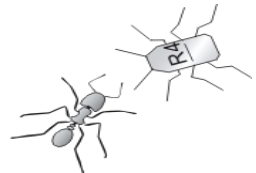
Shifted Rastrigin-2D function



Green dot = mean solution

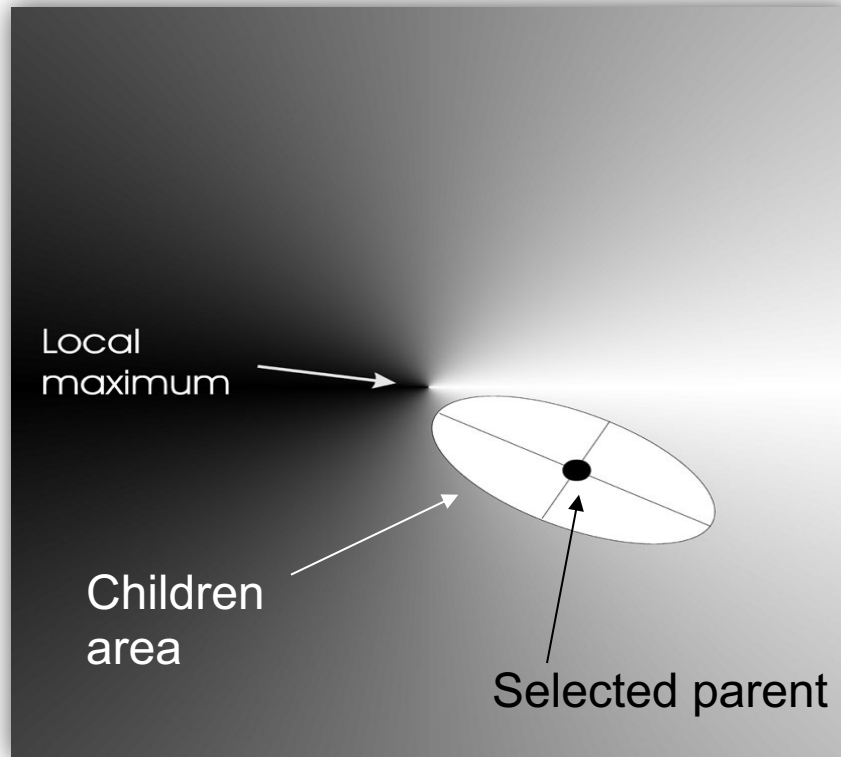
Blue dots = sampled solutions

Red dot = best solution



Correlated mutation steps

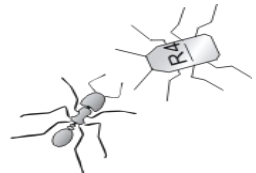
We want to tailor the size of each mutation in order to move each individual in the direction of the estimated gradient of the fitness distribution of the population



This can be done by using the **Covariance matrix** of the population instead of the Identity matrix to sample the mutation vector of each individual

Adapted from: Eiben & Smith: <http://www.evolutionarycomputation.org/slides/>

Companion slides for the book *Bio-Inspired Artificial Intelligence: Theories, Methods, and Technologies* by Dario Floreano and Claudio Mattiussi, MIT Press

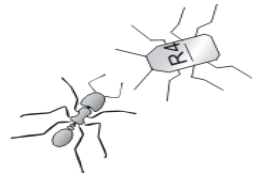


A reminder: Variance and Covariance

$$\text{var}(X) = \frac{1}{n} \sum_{i=1}^n (X_i - \bar{X})^2, \bar{X} \text{ is the mean of the samples of } X$$

$$\text{covar}(X, Y) = \frac{1}{n} \sum_{i=1}^n (X_i - \bar{X})(Y_i - \bar{Y})$$

$$\text{covar}(X, X) = \text{var}(X)$$

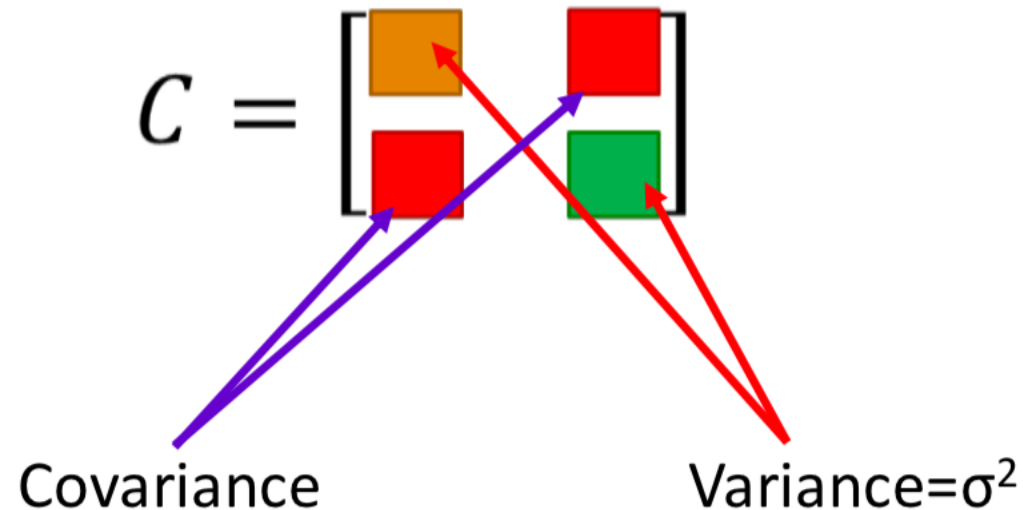


Covariance Matrix

A Covariance Matrix is the matrix whose (ij) element is the covariance between the i and the j element of the data distribution.

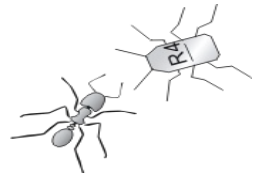
For a distribution with two dimensions **A** (gene x) and **B** (gene y):

$$\begin{pmatrix} cov(A, A) & cov(A, B) \\ cov(B, A) & cov(B, B) \end{pmatrix} = \begin{pmatrix} var(A) & cov(A, B) \\ cov(B, A) & var(B) \end{pmatrix}$$



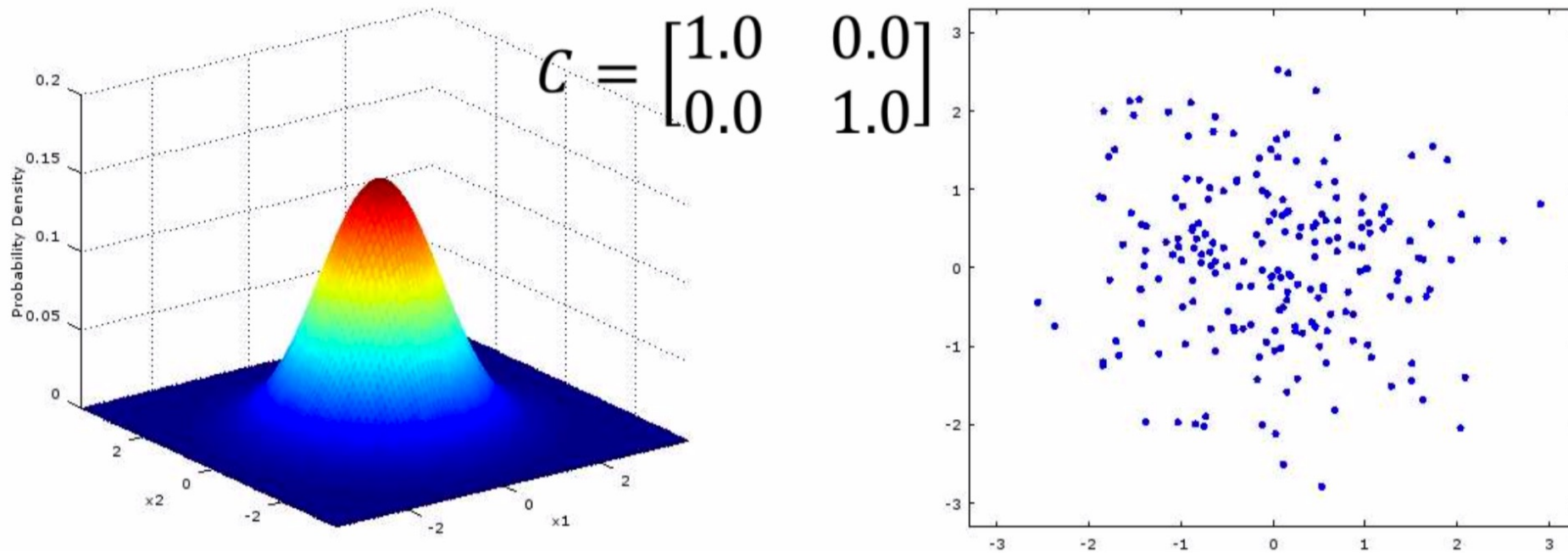
Adapted from: <https://www.slideshare.net/OsamaSalaheldin2/cmaes-presentation>

Companion slides for the book *Bio-Inspired Artificial Intelligence: Theories, Methods, and Technologies* by Dario Floreano and Claudio Mattiussi, MIT Press



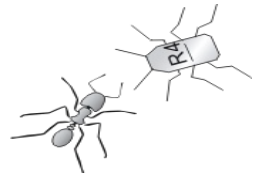
Covariance Matrix of variables that don't covary with equal variance

If two elements x , y do not covary and normally distributed, the covariance matrix is equivalent to $\sigma^2 N(0, I)$



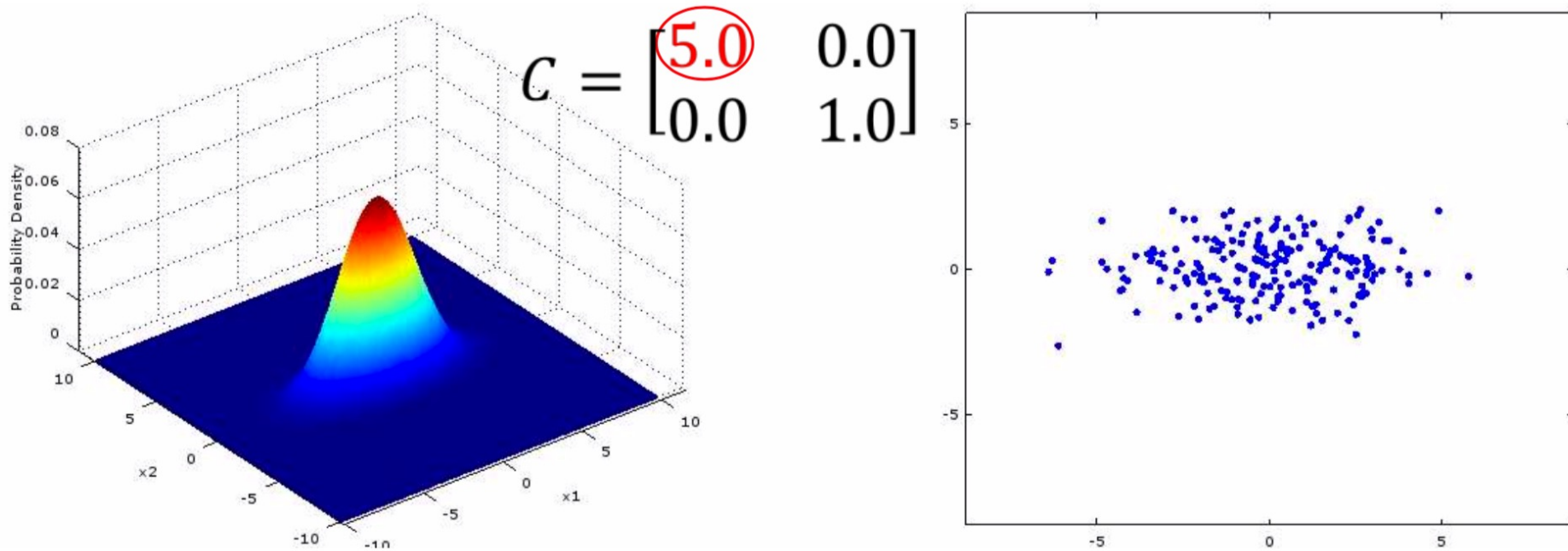
Adapted from: <https://www.slideshare.net/OsamaSalaheldin2/cmaes-presentation>

Companion slides for the book *Bio-Inspired Artificial Intelligence: Theories, Methods, and Technologies* by Dario Floreano and Claudio Mattiussi, MIT Press

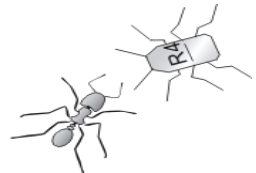


Covariance Matrix of variables that don't covary with different variance

If the two elements x , y do not covary, but x has larger variance

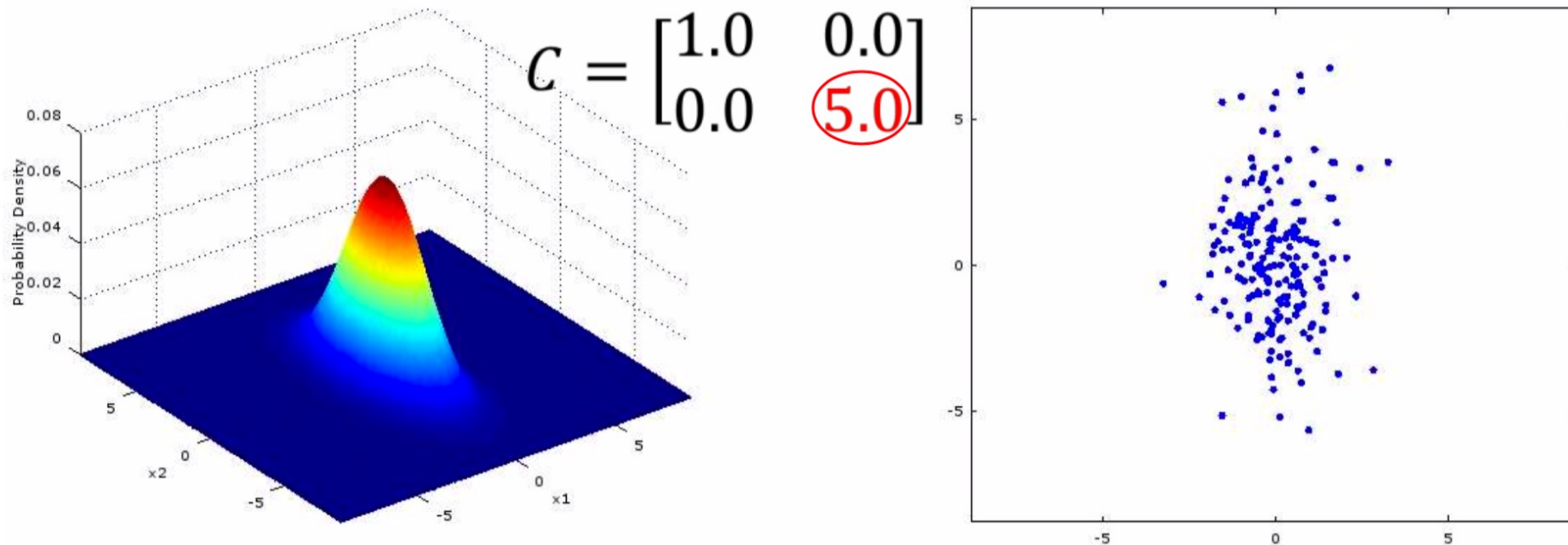


Adapted from: <https://www.slideshare.net/OsamaSalaheldin2/cmaes-presentation>

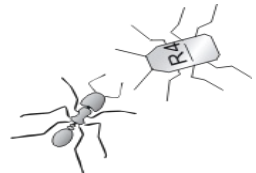


Covariance Matrix of variables that don't covary with different variance

If the two elements x , y do not covary, but y has larger variance

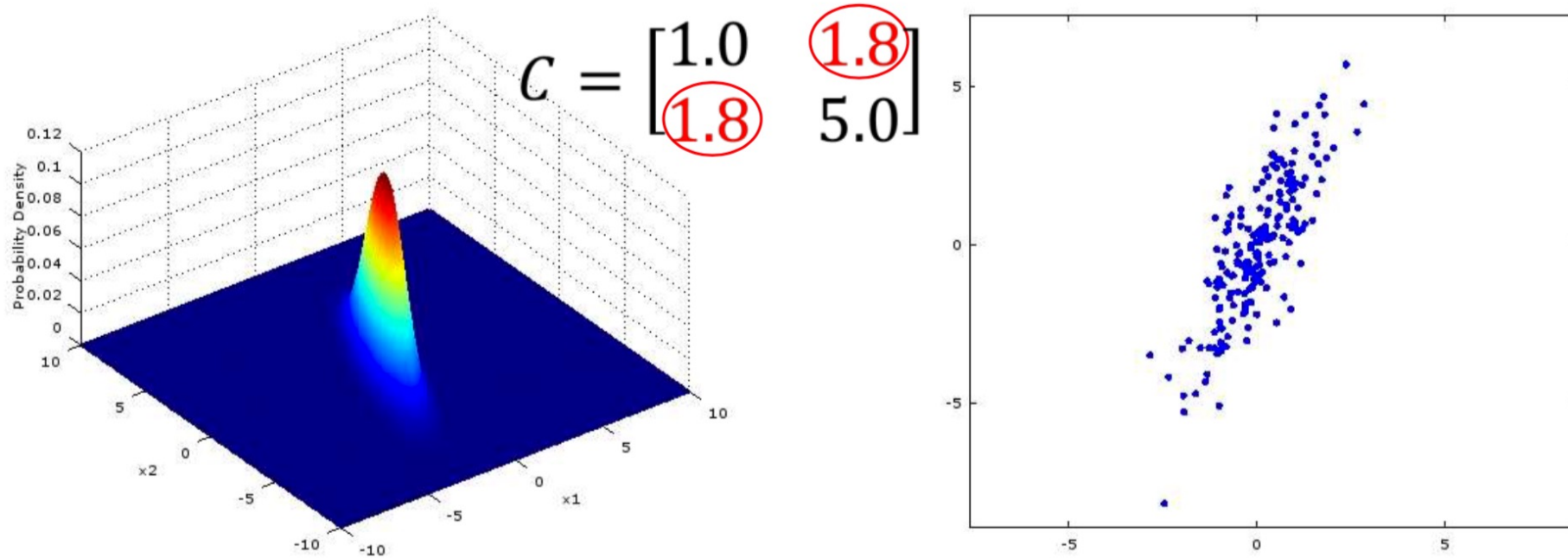


Adapted from: <https://www.slideshare.net/OsamaSalaheldin2/cmaes-presentation>



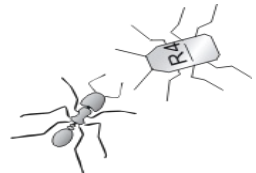
Covariance Matrix of variables that covary with different variance

If the two elements x , y covary, and y has larger variance



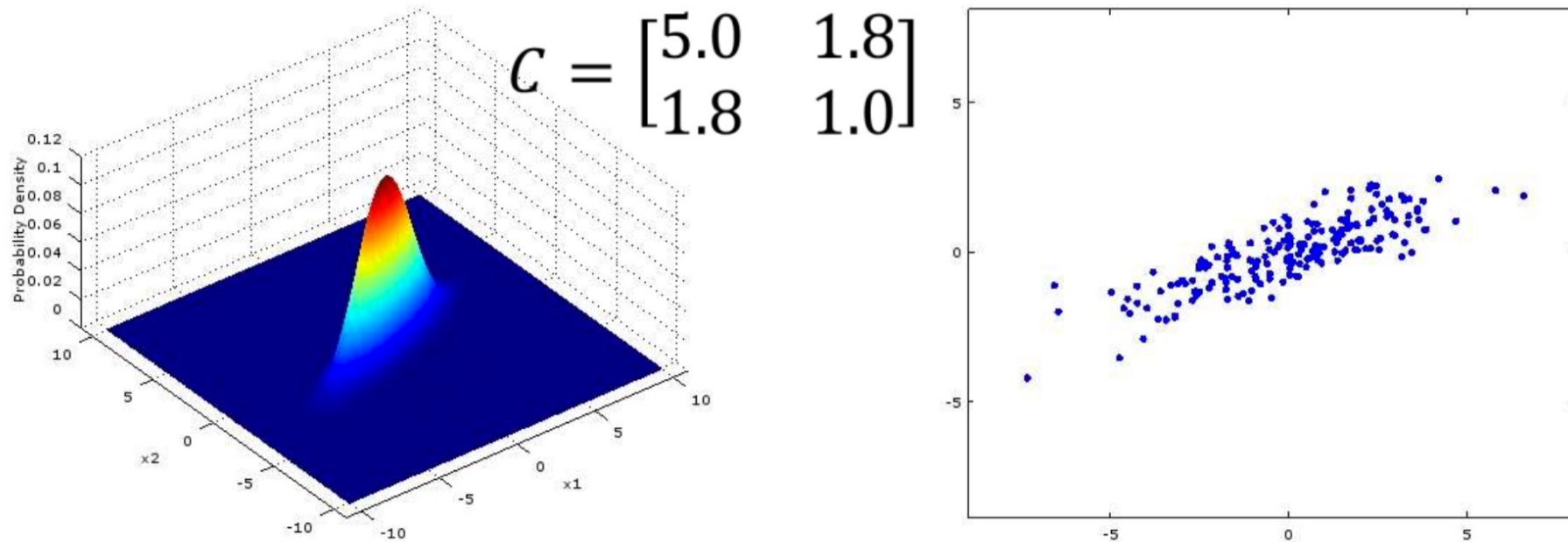
Adapted from: <https://www.slideshare.net/OsamaSalaheldin2/cmaes-presentation>

Companion slides for the book *Bio-Inspired Artificial Intelligence: Theories, Methods, and Technologies* by Dario Floreano and Claudio Mattiussi, MIT Press



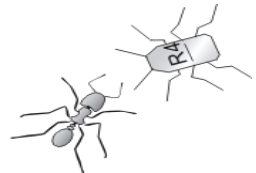
Covariance Matrix of variables that covary with different variance

If the two elements x , y covary, and x has larger variance



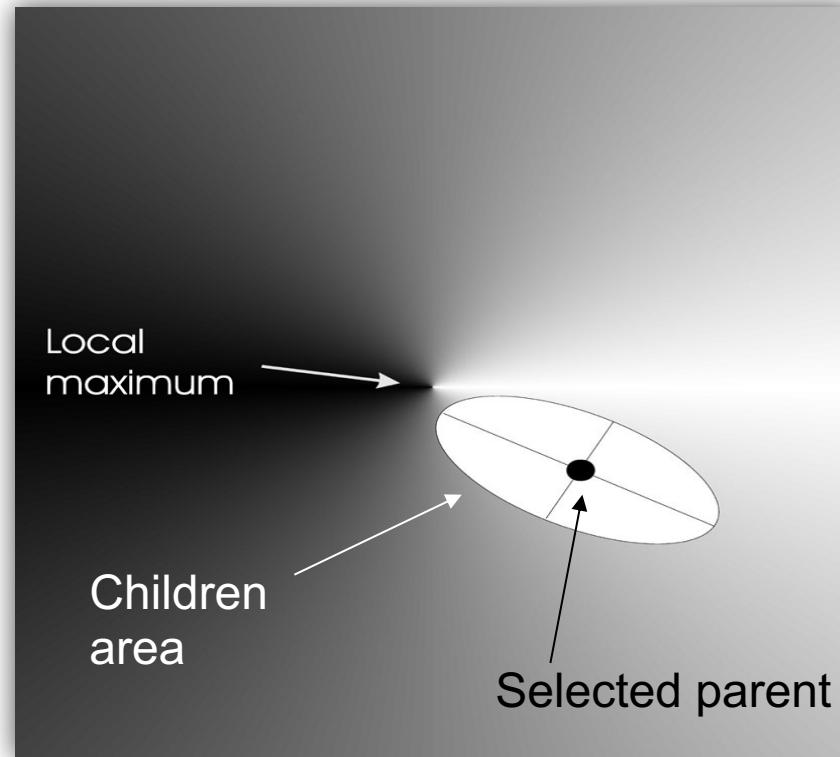
Adapted from: <https://www.slideshare.net/OsamaSalaheldin2/cmaes-presentation>

Companion slides for the book *Bio-Inspired Artificial Intelligence: Theories, Methods, and Technologies* by Dario Floreano and Claudio Mattiussi, MIT Press



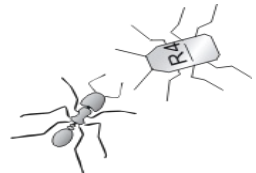
Covariance Matrix Adaptation ES (CMA-ES)

Take larger steps in the direction of highest variance, i.e. the population should move faster in the direction of the eigenvalue corresponding to the largest eigenvector of the covariance matrix of the fitness distribution of the current population



Hansen N, Ostermeier A (2001). [Evolutionary Computation](#), **9(2)**, 159–195.

Companion slides for the book *Bio-Inspired Artificial Intelligence: Theories, Methods, and Technologies* by Dario Floreano and Claudio Mattiussi, MIT Press

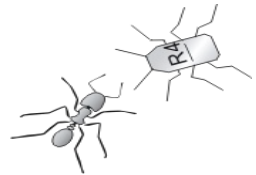


CMA-ES Algorithm: Initialise population

1. Initialise Covariance Matrix **C** as n x n Identity Matrix
2. Set up individual **m** (e.g., solution guess or at the distribution center)
3. Set up initial mutation step size vector σ
4. Generate λ offspring from **m**

$$\mathbf{x}_i = \mathbf{m} + \mathbf{N}_i(\sigma^2, \mathbf{C}) \quad \text{for } 0 < i \leq \lambda$$

notice that **m** represents an estimate of the population mean



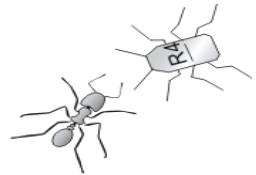
CMA-ES Algorithm: Selection and Reproduction

5. Evaluate λ offspring
6. Identify μ parents with Truncated Rank Selection, e.g. top 25%
7. Update population mean using weighted values of μ parents

$$\mathbf{m} = \sum_{i=1}^{\mu} \mathbf{w}_i \mathbf{x}_i$$

where $w_1 \geq w_2 \geq w_3 \geq w_{\mu} \geq 0$
and $\sum_{i=1}^{\mu} \mathbf{w}_i = 1$

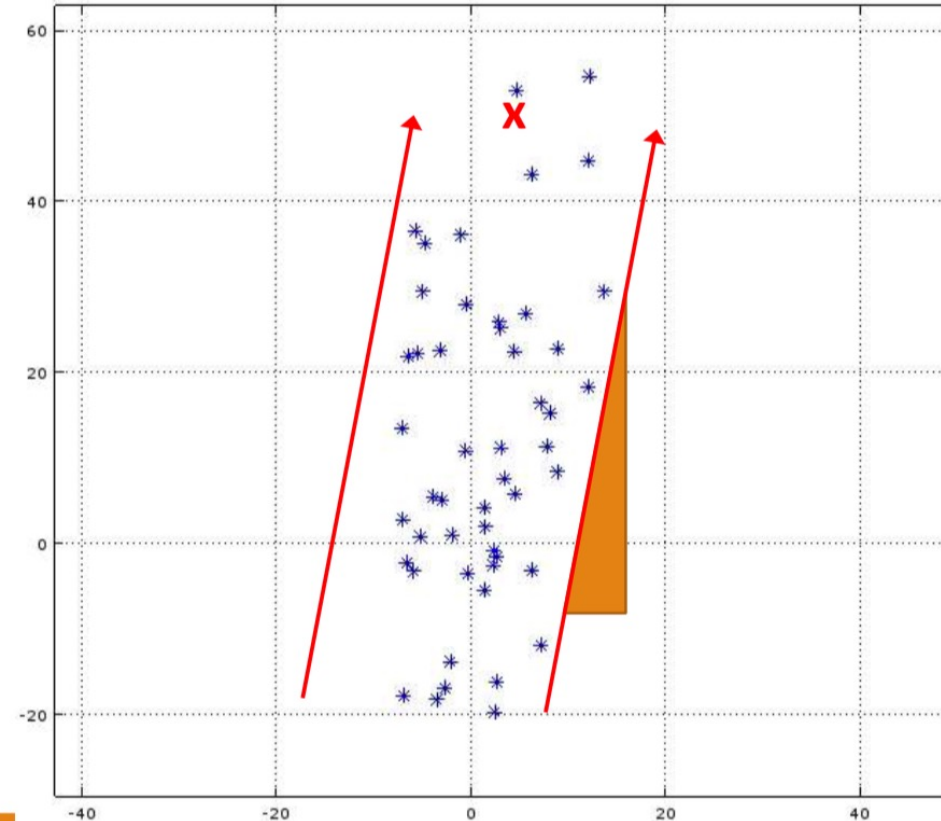
5. Update \mathbf{C} covariance matrix with new \mathbf{C} computed for \mathbf{m} distribution
6. Adapt mutation step size vector σ
7. Go to step 4 ($\mathbf{x}_i = \mathbf{m} + \mathbf{N}_i(\sigma^2, \mathbf{C})$ for $0 < i \leq \lambda$)



An example of CMA-ES at work

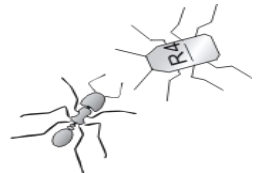
- A practical run of CMA-ES
 - The optimum solution is (5, 50)
 - The initial guess is (0, 0)
 - The population moves faster towards the direction of the second component (50)

$$C = \begin{bmatrix} 1.4011 & 2.0368 \\ 2.0368 & 11.8843 \end{bmatrix}$$

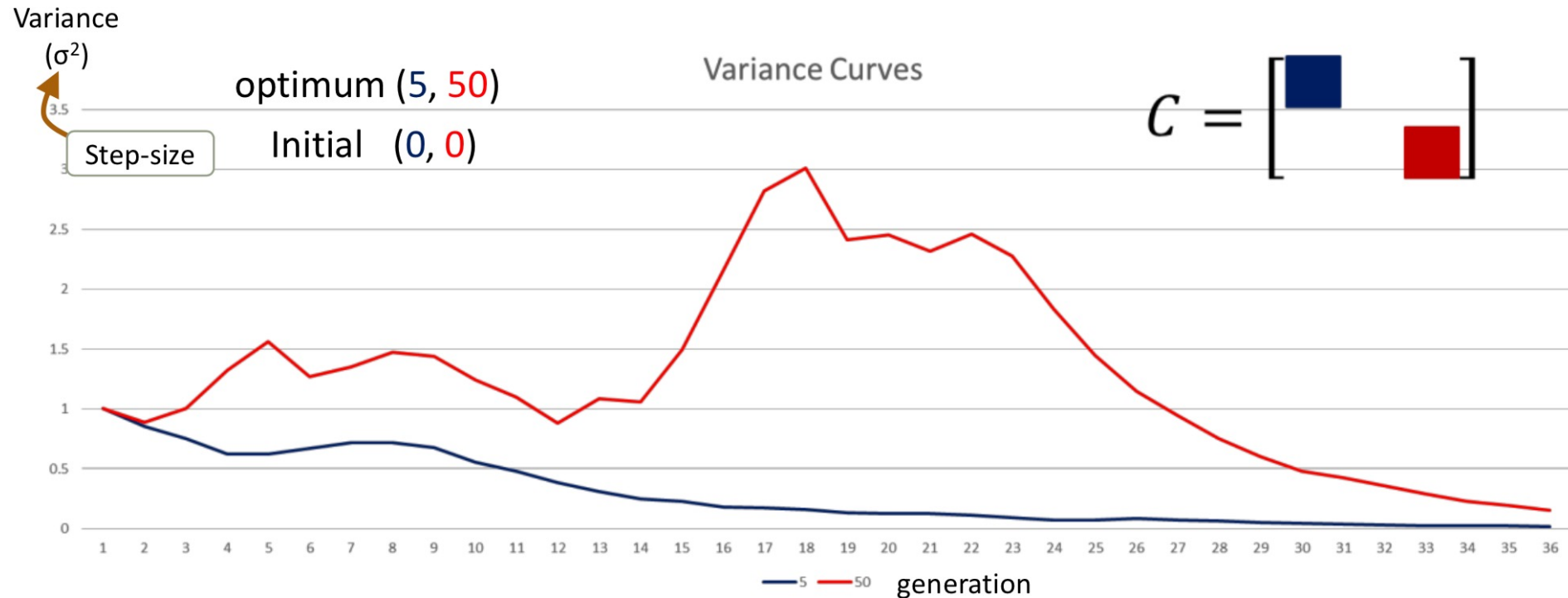


Source: <https://www.slideshare.net/OsamaSalaheldin2/cmaes-presentation>

Companion slides for the book *Bio-Inspired Artificial Intelligence: Theories, Methods, and Technologies* by Dario Floreano and Claudio Mattiussi, MIT Press

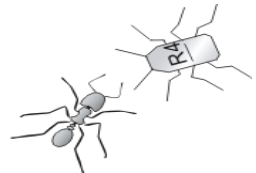


An example of CMA-ES at work

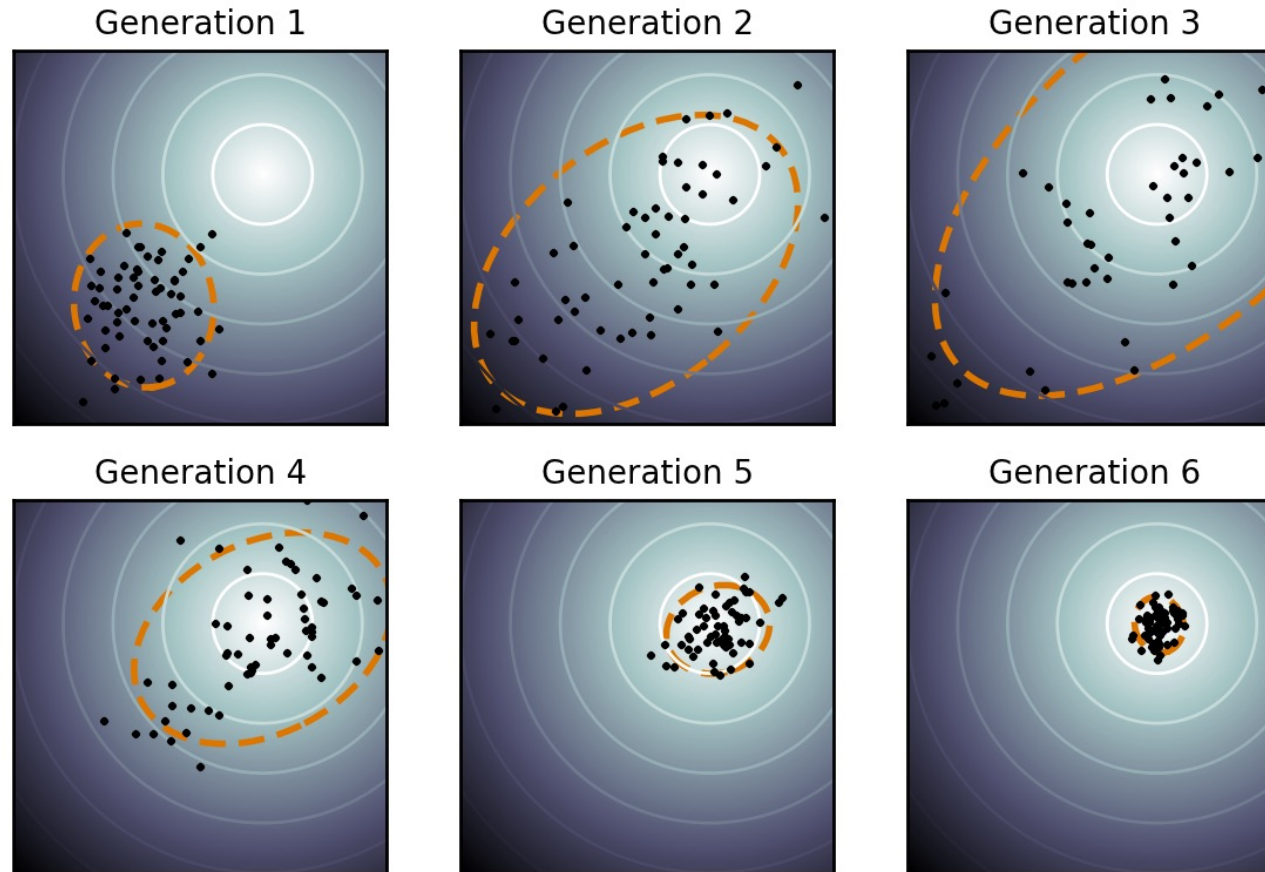


Source: <https://www.slideshare.net/OsamaSalaheldin2/cmaes-presentation>

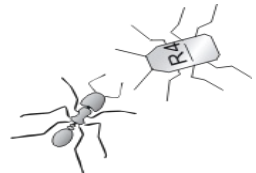
Companion slides for the book *Bio-Inspired Artificial Intelligence: Theories, Methods, and Technologies* by Dario Floreano and Claudio Mattiussi, MIT Press



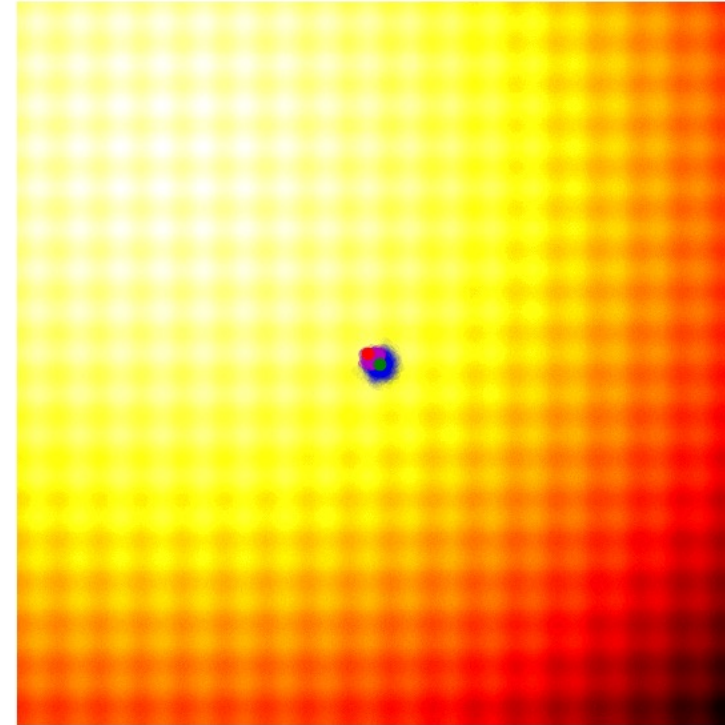
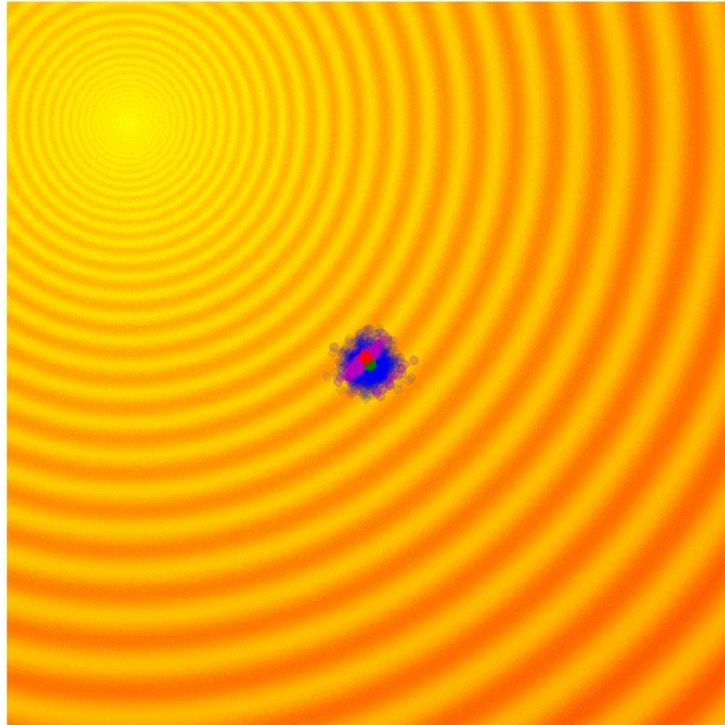
CMA-ES at work



- Full details of \mathbf{C} and σ updates: <https://arxiv.org/abs/1604.00772>
- Computer code: <https://github.com/CMA-ES>



CMA-ES adapts direction and spread



CMA-ES is currently the most powerful evolutionary algorithm for real-value optimization, but for $>10K$ variables it becomes computationally very expensive

