Evolution Strategy

Evolution Strategy (ES)

Generalities:

- *Developed*: Germany in the (late 1960's) early 1970 's by I. Rechenberg and H.-P. Schwefel [Schwefel, 1965][1] and [Rechenberg, 1971].
- *Attributed features*: meant for real-valued optimisation, relatively much theory, parameter-less (no need for tuning! ✓).
- *Special*: self-adaptation of (mutation) parameters standard; population is probabilistically described by a distribution, which is evolved to adapt to the fitness landscape.

Main idea:

The search for extrema is mainly mutation-driven through a multivariate Gaussian distribution $\mathcal{N}(\overline{\mathbf{x}}, \mathbf{C})$, $\overline{\mathbf{x}} \in \mathbb{R}$, $\mathbf{C} \in \mathbb{R}^2$. At each iteration variation operators first perturb/update strategy parameters (i.e. variances and angles for rotating the distribution).

[1] A version in English is given in [Schwefel, 1981].

ES: representation

- In the most general case an individual is encoded as: $\langle \mathbf{x}, \boldsymbol{\sigma}, \boldsymbol{\alpha} \rangle$
  - $\mathbf{x} = x_1, x_2, \ldots, x_n$: design variables;
  - $\boldsymbol{\sigma} = \sigma_1, \sigma_2, \ldots, \sigma_n$: step sizes vector (standard deviations[2]);
  - $\boldsymbol{\alpha} = \alpha_1, \alpha_2, \ldots, \alpha_{n\frac{n-1}{2}}$: angles vector (between two variables[3]).
- Not every component is always present!
  - simplest case: $\langle x_1, x_2, \ldots, x_n, \sigma \rangle$
  - in the general case the memory cost of $\mathbf{C}$ grows quadratically with n ✗

[2] Taken to the power of two, they form the main diagonal of $\mathbf{C}$.

[3] Needed to work out the cross-correlation between two variables in $\mathbf{C}$, see [Eiben and Smith, 2003].

ES: parent selection

Uniform random.

- Parents are selected by *uniform random distribution* whenever an operator needs one (or some)!
  - Thus, ES parent selection is unbiased: every individual has the same probability to be selected.

NOTE that in ES "parent" means a population member

no need for a mating pool, as in GA, where parents are population members selected to undergo variation.

ES: recombination
(discrete and intermediary recombination)

- Given two parents $\mathbf{x}$ and $\mathbf{y}$ a child $\mathbf{z}$ is generated according to
  - *discrete* recombination: $\mathbf{z}[i] = \mathbf{x}[i]$ or $\mathbf{y}[i]$ (fifty-fifty), $i = 1, 2, \ldots, n$
  - *intermediary* recombination: $\mathbf{z}[i] = \frac{\mathbf{x}[i]+\mathbf{y}[i]}{2}$, $i = 1, 2, \ldots, n$
- Both discrete and intermediary can follow a two-parent or multi-parent logic:
  - *local*: $\mathbf{x}$ and $\mathbf{y}$ are the same $\forall i = 1, 2, \ldots, n$;
  - *global*: $\mathbf{x}$ and $\mathbf{y}$ are randomly re-selected $\forall i = 1, 2, \ldots, n$.
- Thus, recombination plays a minor role but 4 combinations can be used:

	discrete	**intermediary**
local	local discrete recombination	local intermediary recombination
global	global discrete recombination	global intermediary recombination

ES: mutation
main idea

- Mutation is done via a Gaussian distribution centred in the offspring $\mathbf{x}$:

$$\mathbf{x}^{'} \sim \mathcal{N}(\mathbf{x}, \mathbf{C})$$

- it can be seen as adding "noise" to $\mathbf{x}$:

$$\mathbf{x}^{'} \leftarrow \mathbf{x} + \mathcal{N}(\emptyset, \mathbf{C})$$

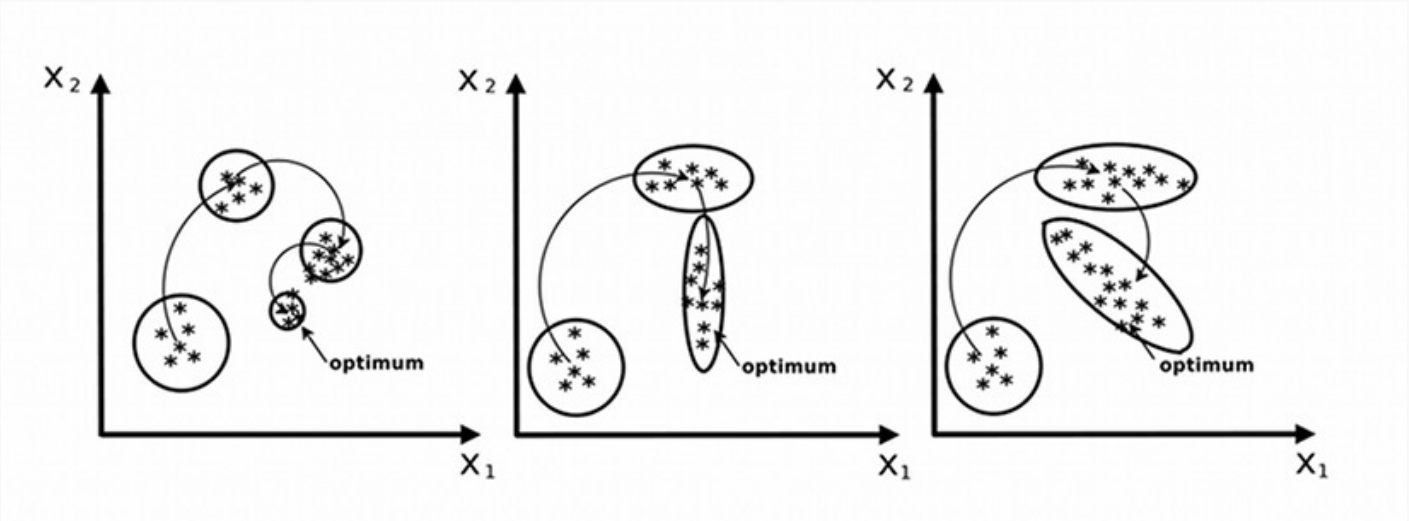
- $\mathbf{C}$ has to be updated to adapt it to the search space before the sampling!
  - parameters are evolved⇒good ones go ahead together with the good generated solution, and used for the next updated! ✓
- $\mathcal{C}$ contains variances (worked out from $\boldsymbol{\sigma}$) and correlations (worked out from $\boldsymbol{\alpha}$) (click-here):

$$\mathbf{C} = [c_{i,j}] = \begin{bmatrix} \sigma^2_{x_1} & \sigma_{x_1 x_2} & \sigma_{x_1 x_3} & \cdots & \sigma_{x_1 x_n} \\ \sigma_{x_1 x_2} & \sigma^2_{x_2} & \sigma_{x_2 x_3} & \cdots & \sigma_{x_2 x_n} \\ \sigma_{x_1 x_3} & \sigma_{x_2 x_3} & \sigma^2_{x_3} & \cdots & \sigma_{x_3 x_n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \sigma_{x_1 x_n} & \sigma_{x_2 x_n} & \sigma_{x_3 x_n} & \cdots & \sigma^2_{x_n} \end{bmatrix}$$

ES: mutation
the role of the covariance matrix.

The covariance matrix plays a major role in the mutation:

- Uncorrelated mutation: $\langle \mathbf{x}, \sigma \rangle \Rightarrow \mathbf{C} = \sigma^2 \mathbf{I}$
  - the Gaussian can only change size and move! (light ✓,less efficient ✗)
- Uncorrelated with n step sizes: $\langle \mathbf{x}, \boldsymbol{\sigma} \rangle \Rightarrow \mathbf{C} = diag\{\sigma_1, \sigma_1 \ldots, \sigma_n\}$
  - the Gaussian can change size, shape and move! (compromise)
- Correlated mutation: $\langle \mathbf{x}, \boldsymbol{\sigma}, \boldsymbol{\alpha} \rangle$
  - the Gaussian can change size, shape, orientation and move! (efficient ✓, space/time complexity ✗)



ES: uncorrelated mutations
implementation details

$$\forall i = 0, 1, 2 \ldots, n$$

- Uncorrelated mutation (with 1 step size):

$$\sigma^{'} = \sigma e^{\tau \mathcal{N}(0,1)}, \quad (\tau \propto \frac{1}{\sqrt{n}})$$

$$\mathbf{x}^{'}[i] = \mathcal{N}_i\left(\mathbf{x}[i], \sigma^{'2}\right) = \mathbf{x}[i] + \sigma^{'}\mathcal{N}_i(0,1)$$

- Uncorrelated mutation (with 1 step size):

$$\boldsymbol{\sigma}^{'}[i] = \boldsymbol{\sigma}[i] e^{\tau^{'}\mathcal{N}(0,1) + \tau \mathcal{N}_i(0,1)}, \quad (\tau^{'} \propto \frac{1}{\sqrt{2n}}, \tau \propto \frac{1}{\sqrt{2\sqrt{n}}})$$

$$\mathbf{x}^{'}[i] = \mathcal{N}_i\left(\mathbf{x}[i], \sigma^{'2}\right) = \mathbf{x}[i] + \sigma^{'}\mathcal{N}_i(0,1)$$

Always check $\sigma \neq 0$! (see [Eiben and Smith, 2003] for details)

ES: correlated mutation
implementation details

- The covariance matrix is defined as

$$\mathbf{C} = [c_{i,j}] = \begin{cases} c_{ii} = \sigma_i^2 \\ c_{i,j} = \frac{1}{2}\left(\sigma_i^2 - \sigma_j^2\right) tan\left(2\alpha_{i,j}\right) \end{cases}$$

- first we update the diagonal

$$\boldsymbol{\sigma}'[i] = \boldsymbol{\sigma}[i] e^{\tau' \mathcal{N}(0,1) + \tau \mathcal{N}_i(0,1)}, \quad (\tau' \propto \frac{1}{\sqrt{2n}}, \tau \propto \frac{1}{\sqrt{2\sqrt{n}}})$$

- then the correlations

$$\alpha_j' = \alpha_j + \beta \mathcal{N}(0,1), \quad (\beta \approx 5^\circ)$$

Always check σs and αs are acceptable (see [Eiben and Smith, 2003] for details).

ES: survivor selection

comma and plus strategies.

Survivor selection is deterministic:

- Comma selection $\Rightarrow (\mu, \lambda)$-ES:
  - generate $\lambda > \mu$ offspring, the fittest μ of them replace the old population.
- Plus strategy $\Rightarrow (\mu + \lambda)$:
  - generate λ offspring, the fittest μ out of a total of $\mu + \lambda$ points replace the old population.

Advantages of the comma strategy:

- better in leaving local optima;
- better in following moving optima;
- unlike the $(\mu + \lambda)$, prevents from retaining mediocre solutions for many generations;
- unlike the $(\mu + \lambda)$, the scheme is not too exploitative.

Advantages of the plus strategy:

- unlike (μ, λ), useful solutions are not cancelled.
- the $(\mu + \lambda)$-selection can be preferred when are looking for "marginal"; enhancements and when it is important not to lose the previous enhancements;
- the $(\mu + \lambda)$ strategy can be preferable in large scale problems when exploitation is needed;

(1+1)-ES with One-Fifth Success Rule

- A very simple (single solution) ES has been originally proposed.
- This variant employed a basic uncorrelated mutation with single step size σ.
- Theoretical studies motivated an update on-the-fly for σ according to the popular $\frac{1}{5}$ success rule [Rechenberg, 1971].
- Two variants are described in this document (click-here).

Important Consideration: Premature Convergence and Stagnation

Collective behavior in a population

- In Unlike single-solution algorithms, a population-based algorithm has many solutions that somehow “interact”;
- how some solutions are with respect to the others, is very important in terms of algorithmic success or failure!

Population diversity

- In this context, the population diversity is the average distance among the individuals of the population.
- This quantity can be interpreted also as the width of decision space covered by the population A focused population has low diversity, a spread population has a high diversity.

Premature convergence

- While during the optimisation, the diversity may become very low, i.e. all the individuals are very similar (very close to each other).
- If this diversity loss occurs in a suboptimal are of the decision space (before achieving the neighborhood of the optimum), this condition is said premature convergence

NOTE: It is not the convergence to a local optimum!

Stagnation

- The diversity can still be high but there is no improvement in the objective function.
- The move operators generate trial solutions that are not better than the existing population individuals.
- The search does not lead to successful improvements (as if the convergence is excessively slow).

Exploration and exploitation balance
(again)

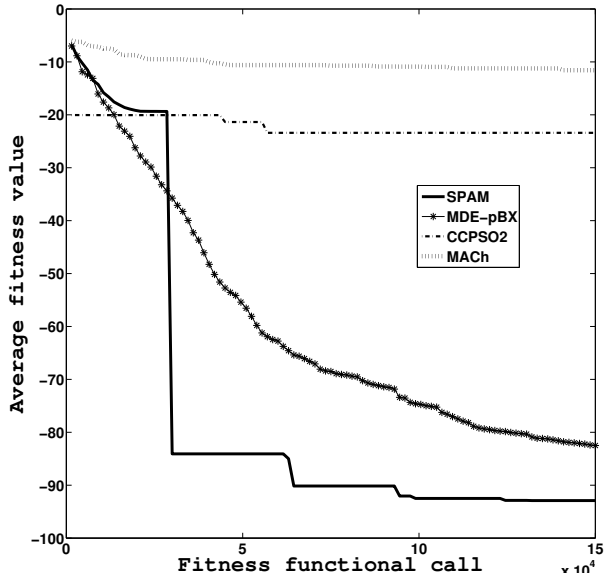
- An excessive exploitation leads to premature convergence: the present search directions are so excessively used that new promising points far away from the current best/elite cannot be generated/computed.
- An excessive exploration leads to stagnation: the algorithm keeps on discarding possible search directions in order to find new areas of the decision space without having carefully used the present search directions.

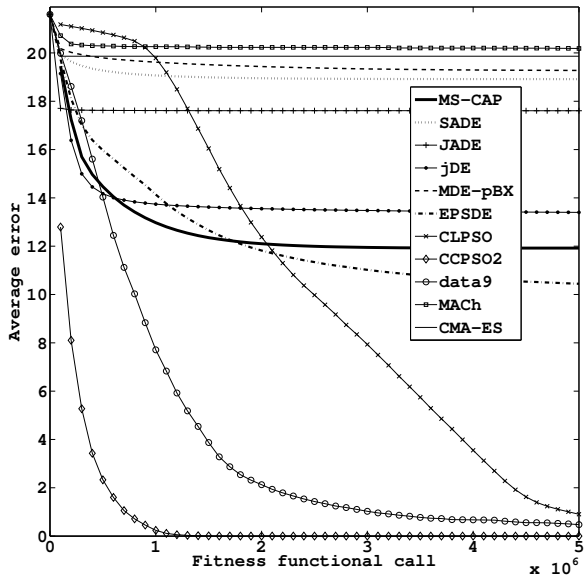
For GAs we talked about selection pressure:

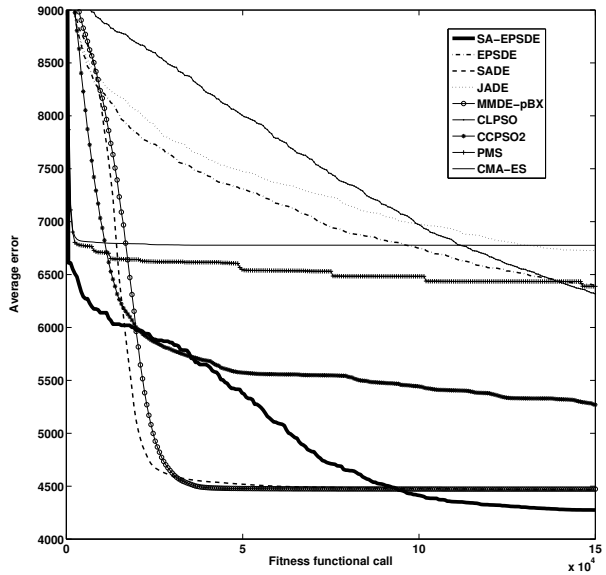
- low diversity leads to low selection pressure $\Rightarrow$ similar behaviour to stagnation but it happens with low diversity! e.g. in a plateau.
- high diversity leads implicates high selection pressure leading to premature convergence: EAs (and other optimisers) do not have a proper selection mechanism but can still suffer of this phenomenon!

Performance trend

- Graph average performance vs fitness evaluations/function calls.
- In presence of stagnation or premature convergence it is flat in correspondence to a mediocre fitness value.







Examples

- An ES with plus strategy is likely to suffer from premature convergence, especially if the population size is small as it will quickly focus on an area of the decision space and then is unlikely to be able to find other points far away from this area.
- A GA with tournament selection, small tournament size, and high mutation rate can suffer from stagnation as will tend to generate very different solutions from the parents.

A Modern And Efficient Evolution Strategy: CMA-ES

Covariance Matrix Adaptation ES
CMA-ES

- one of the most elegant and efficient meta-heuristic for real-valued optimisation!
  - designed on solid mathematical considerations [Hansen and Ostermeier, 1996] ✓
  - lacks of problem dependent parameters and is invariant to many transformations ✓
  - handles ill-conditioned problems ✓
  - nowadays treated as a benchmark for real-valued optimisation;
- many variants exist (e.g. sep-CAME-ES, g-CMA-ES, (1+1)-CMA-ES etc.);
- complex and memory expensive! $\Rightarrow$ not very suitable for real-time and large sale optimisation ✗
- preferable in mono-modal landscape rather than multi-modal.

Standard CMA-ES

(namely rank-μ-update CMA-ES with weighted global recombination [Nikolaus and Stefan, 2004])

- Standard CMA-ES is self-tuning: optimal parameters setting is given and reported in Table 1.
  - N:B. Also the optimal population size is self-tuned! ✓

Table: Description of parameters in CMA-ES

Parameter	Description	Value
λ	offspring number, population size	$\lambda = 4 + \lfloor 3ln(n) \rfloor$
μ	parents number, used to update mean value, covariance and step-size	$\mu = \lfloor \frac{\lambda}{2} \rfloor$
$w_{i=1\dots\mu}$	recombination weights	$w_{i=1\dots\mu} = \frac{ln(\mu+1)-ln(i)}{\sum_{j=1}^{\mu} ln(\mu+1)-ln(j)}$
μ_{eff}	variance effective selection mass	$\mu_{eff} = \frac{1}{\sum_{1}^{\mu} w_i^2}$
		($w_{i=1\dots\mu}$ are chosen so that $\mu_{eff} \approx \frac{\lambda}{4}$)
c_c	decay rate for the evolution path	$c_c = \frac{4}{n+4}$
μ_{cov}	learning coefficient for the covariance matrix	$\mu_{cov} = \mu_{eff}$
c_{cov}	learning rate for the covariance matrix	$c_{cov} = \frac{1}{\mu_{cov}} \frac{2}{\left(n+\sqrt{2}\right)^2} + \left(1 - \frac{1}{\mu_{cov}}\right) min\left(1, \frac{2\mu_{eff}-1}{(n+2)^2+\mu_{eff}}\right)$
c_σ	decay rate for the conjugate evolution path	$c_\sigma = \frac{\mu_{eff}+2}{n+\mu_{eff}+3}$
d_σ	damping parameter for σ-change	$d_\sigma = 1 + 2max\left(0, \sqrt{\frac{\mu_{eff}-1}{1+n}} - 1\right) + c_\sigma$

Standard CMA-ES
working logic:

CMA-ES pseudo-code

–problem dependent user defined input parameters–

initialise n, $\langle \mathbf{x} \rangle_w^{(\mathbf{0})}$ and $\sigma^{(\mathbf{0})}$ ▷ $\sigma^{(\mathbf{0})} = 0.5$, $\langle \mathbf{x} \rangle_w^{(\mathbf{0})} \sim \mathcal{U}(0,1)$ in [Nikolaus and Stefan, 2004]

–strategy parameter setting: selection–

initialise λ, μ, μ_{eff} and $w_{i=\mathbf{1}\dots,\mu}^{(\mathbf{0})}$ as in Table 1 ▷ λ can be modified if needed

–strategy parameter setting: adaptation–

initialise c_c, μ_{cov}, c_σ, d_σ and d_σ as in Table 1

–initialise dynamic (internal) matrices parameters and constants–

$\mathbf{P_c} = \mathbf{P_\sigma} = [\emptyset]$, $\mathbf{B} = \mathbf{D} = \mathbf{I}$, $\mathbf{C} = \mathbf{B} * \mathbf{D}\,(\mathbf{B} * \mathbf{D})^T$

–main loop–

while condition on budget **do**
- sample λ new individuals from distribution (*mutation*) ▷ Formula 1
- the new λ points replace the old population ((λ, λ) *strategy*)
- evaluate individuals and sort them based on their fitness (*recombination*)
- update $\langle \mathbf{x} \rangle$ based on a weighted sum of the best μ individuals ▷ Formula 2
- update the evolution paths $\mathbf{P}_\sigma$ and $\mathbf{P_c}$ ▷ Fomula 5 and 3
- update covariance matrix $\mathbf{C}$ and step-size σ consequently ▷ Formula 4 and 6

end while

Output best individual ever found $\mathbf{x_e}$

Frame Title
the sampling

- Mathematically, at a generic generation g, λ individuals are sampled from:

$$\mathbf{x}_{\mathbf{k}}^{(\mathbf{g}+\mathbf{1})} \sim \mathcal{N}\left(\langle \mathbf{x} \rangle_{w}^{(g)}, \left(\sigma^{g}\right)^{\mathbf{2}} \mathbf{C}^{(g)}\right) \tag{1}$$

which is centred in the on the point obtained as weighted sum of the best $\mu \leq \lambda$ candidate solutions:

$$\langle \mathbf{x} \rangle_{w}^{(g)} = \sum_{i=\mathbf{1}}^{\mu} w_i \mathbf{x}_{i \in \{\textit{indexes of the best } \mu \textit{ points}\}}^{g} \tag{2}$$

- Implementation-wise, we decompose $\mathbf{C}$ into its diagonal ($\mathbf{D}$) and orthonormal ($\mathbf{B}$) components via PCA:

$$\mathbf{C}^{(\mathbf{g})} = \mathbf{B}^{(\mathbf{g})} \mathbf{D}^{(\mathbf{g})^{\mathbf{2}}} \mathbf{B}^{(\mathbf{g})^{T}}$$

such that Formula 1 can be implemented as:

$$\mathbf{x}_{\mathbf{k}}^{(\mathbf{g}+\mathbf{1})} = \langle \mathbf{x} \rangle_{w}^{(g)} + \sigma^{(g)} \mathbf{B}^{(\mathbf{g})} \mathbf{D}^{(\mathbf{g})} \mathcal{N}(\emptyset, \mathbf{I})$$

Standard CMA-ES
the covariance matrix

- C is updated to approximate the local behaviour of the fitness landscape (Hessian matrix) in a nighbouroigh of the optimal solution!
- first, we update the evolution path $\mathbf{P_c}$:

$$\mathbf{p_c}^{(g+1)} = (1 - c_c)\mathbf{p_c}^{(g)} + H_\sigma^{(g+1)} \sqrt{c_c \cdot (2 - c_c)} \cdot \frac{\sqrt{\mu_{eff}}}{\sigma^{(g)}} \left(\langle \mathbf{x} \rangle_w^{(g+1)} - \langle \mathbf{x} \rangle_w^{(g)} \right) \tag{3}$$

and then:

$$\mathbf{C}^{(g+1)} = (1 - c_{cov})\mathbf{C}^{(g)} + c_{cov} \cdot \frac{1}{\mu_{cov}} \mathbf{p_c}^{(g+1)} \left(\mathbf{p_c}^{(g+1)} \right)^T$$

$$+ c_{cov} \cdot \left(1 - \frac{1}{\mu_{cov}} \right) \sum_{i=1}^{\mu} \left(\mathbf{x}_{i:\lambda}^{(g+1)} - \langle \mathbf{x} \rangle_w^{(g)} \right) \left(\mathbf{x}_{i:\lambda}^{(g+1)} - \langle \mathbf{x} \rangle_w^{(g)} \right)^T \tag{4}$$

with

$$H_\sigma^{(g+1)} \begin{cases} 1 & \text{if } \frac{||\mathbf{P}_\sigma^{(g+1)}||}{\sqrt{1 - \left(1 - c_\sigma^{(g+1)2}\right)}} < \left(1.5 + \frac{1}{n - 0.5}\right) E\left(||\mathcal{N}(\emptyset, \mathbf{I})||\right). \\ 0 & \text{otherwise} \end{cases}$$

and "conjugate evolution path" $\mathbf{P}_\sigma$ equal to:

$$\mathbf{P}_\sigma^{(g+1)} = (1 - c_\sigma) + \sqrt{c_\sigma \cdot (2 - c_\sigma)} \underbrace{\mathbf{D}^{(g)} \mathbf{B}^{(g)-1} \mathbf{D}^{(g)T}}_{\mathbf{C}^{(g)-\frac{1}{2}}} \frac{\sqrt{\mu_{eff}}}{\sigma^{(g)}} \left(\langle \mathbf{x} \rangle_w^{(g+1)} - \langle \mathbf{x} \rangle_w^{(g)} \right) \tag{5}$$

- finally, the step size is updated:

$$\sigma^{(g+1)} = \sigma^{(g)} e^{\frac{c_c}{d_\sigma} \left(\frac{||\mathbf{P_c}||}{E||N(\emptyset, I)||} - 1 \right)} \tag{6}$$

Swarm Intelligence

Swarm intelligence

general metaphor:

- Some animals live and operate in groups;
- when these animals (e.g. swarms of birds, school of fish, ants, bees, monkeys, etc.) are all together can make things that would not be able to do when they are alone:
  - "collective intelligence".
- Swarm Intelligence algorithms are inspired by the collective behavior, i.e. solutions share information with other close solutions (neighbourhood).

Swarm intelligence optimisation
generalities:

- population-based;
- each individual "move" thanks to a moving operator that makes use of the other individuals;
- One-to-one replacement (one-to-one spawning) right after the move.

Swarm Intelligence optimization

Initialisation ▷ Randomly sample initial swarm
while *Condition on budget* **do**
  for each x $\in$ *Swarm* **do**
    Update moving parameters
    Perform move
    Evaluate $f(x)$ *and eventually perform replacement* ▷ one-to-one spawning
  end for
end while
Output *Best Individual*

SI and EA

- Swarm intelligence and EAs are clearly similar;
- the way of generating a trial individual is of the same type but one parent takes a special role (individual before the move)
- swarm intelligence has a peculiar survivor selection: the individual after the move (offspring) replaces the individual before the move (parent that generated it).

Laboratory and participation work:

- For the usual four problems in $10D$ and $50D$, and with the usual experimental setting (i.e. $5000 \times n$ fitness evaluations, 30 runs), implement the two variants of (1+1)-ES with One-Fifth Success Rule described in the paper.
  - compare between them.
- As they can be seen as single solution variants, compare the results against those of S and SA.

References I

Eiben, A. E. and Smith, J. E. (2003).
Introduction to Evolutionary Computation.
Springer-verlag, Berlin.

Hansen, N. and Ostermeier, A. (1996).
Adapting arbitrary normal mutation distributions in evolution strategies: The covariance matrix adaptation.
In *Proceedings of the IEEE International Conference on Evolutionary Computation*, pages 312–317.

Nikolaus, H. and Stefan, K. (2004).
Evaluating the cma evolution strategy on multimodal test functions.
In *Parallel Problem Solving from Nature-PPSN VIII*, pages 282–291. Springer.

Rechenberg, I. (1971).
Evolutionsstrategie – Optimierung technischer Systeme nach Prinzipien der biologischen Evolution.
PhD thesis, Technical University of Berlin.

References II

Schwefel, H. (1981).
Numerical Optimization of Computer Models.
Wiley, Chichester, England, UK.

Schwefel, H.-P. (1965).
Kybernetische Evolutioals Strategie der experimentellen Forschung in der Strömungstechnik.
PhD thesis, Technical University of Berlin, Hermann Föttinger–Institute for Fluid Dynamics.