Scalability of Population-based Stochastic Metaheuristics

Daniela Zaharie

Department of Computer Science
West University of Timisoara, Romania
e-mail:dzaharie@info.uvt.ro

Workshop on "Stochastic Geometry and Big Data" - 24.11.2015

Outline

Black box optimization

Example:

- given a parameterized module for image registration encapsulated in a proprietary software
- find the parameters values which:
  - ensure that the registration error is smaller than a given threshold
  - the running time is as small as possible

Black box optimization

Example:

- given a parameterized module for image registration encapsulated in a proprietary software
- find the parameters values which:
  - ensure that the registration error is smaller than a given threshold
  - the running time is as small as possible

Characteristics of the black box problems

- only partial/uncertain apriori knowledge on the influence of parameters on accuracy
- only objective function values are known, no gradient information

Large scale optimization

Example 1: non-rigid multi-modal image registration[1]

- find the the parameters of a free form deformation model
- which minimize a similarity measure (e.g. mutual information)
- Remarks:
  - problem size: for a $8 \times 8 \times 8$ mesh there are 1536 parameters; used optimizer
  - optimizer: cat swarm optimization

[1] Yang at al., Non-rigid Multi-modal Medical Image. Registration by Combining L-BFGS-B with Cat Swarm Optimization, Information Sciences 2015

[2] Zhang et al., PSO-EM: A Hyperspectral Unmixing Algorithm Based On Normal Compositional Model, TGRS 2014

Large scale optimization

Example 1: non-rigid multi-modal image registration[1]

- find the the parameters of a free form deformation model
- which minimize a similarity measure (e.g. mutual information)
- Remarks:
  - problem size: for a $8 \times 8 \times 8$ mesh there are 1536 parameters; used optimizer
  - optimizer: cat swarm optimization

Example 2: (hyper)spectral unmixing[2]

- find the abundancy values which maximizes the log-likelihood function from the E-step in an EM framework
- problem size: number of pixels $\times$ number of endmembers (750 for a 50×50 subimage and 3 endmembers)
- optimizer: particle swarm optimization

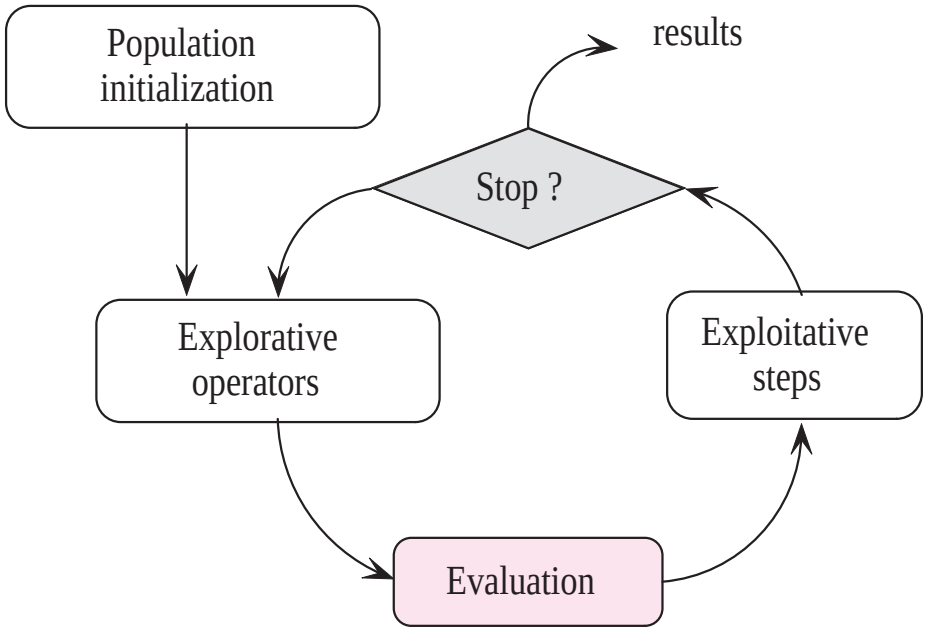
[1] Yang at al., Non-rigid Multi-modal Medical Image. Registration by Combining L-BFGS-B with Cat Swarm Optimization, Information Sciences 2015

[2] Zhang et al., PSO-EM: A Hyperspectral Unmixing Algorithm Based On Normal Compositional Model, TGRS 2014

Outline

Population-based search

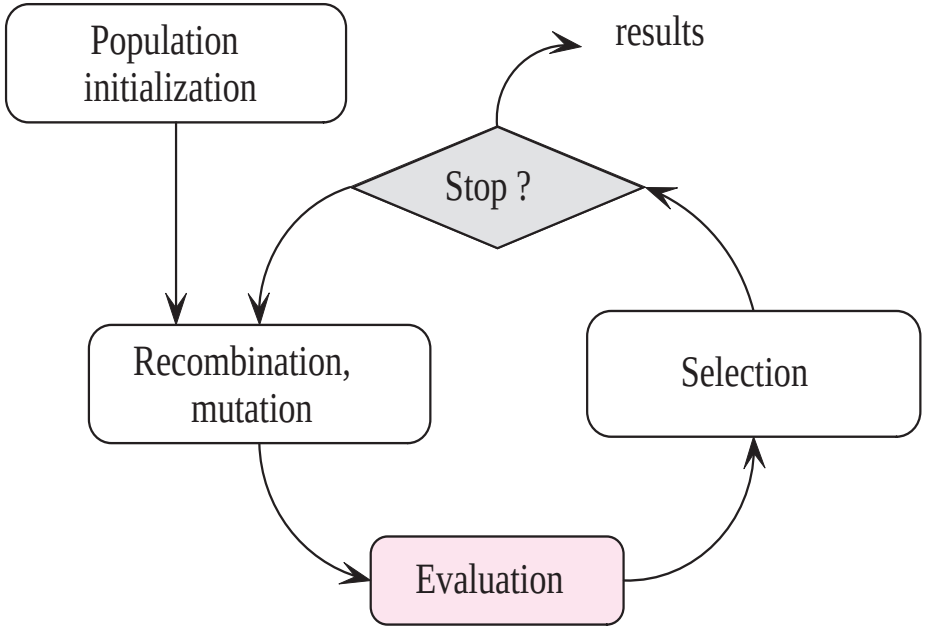
- an iterative optimization method which uses a population of candidates to search the solutions space
- it is based on two main mechanisms:
  - search space exploration
  - exploitation of the knowledge collected during the previous search steps



Population Based Iterative Search

Main Components

- Mutation: random perturbation of elements
  - distribution independent of the current population
  - distribution influenced by the current population
- Recombination: mixing information from several elements
  - discrete
  - arithmetical
- Selection: choice of promising elements
  - proportional
  - tournament



An Evolutionary Algorithm

Outline

Differential Evolution

... is a simple but rather powerful metaheuristic

- developed in 1995 by Rainer Storn and Kenneth Price as a continuous optimization method
  - starting problem: Chebyshev polynomials fitting (33 variables)
  - inital variant: genetic annealing algorithm developed by Kenneth Price (1994)
- main idea: use a mutation/recombination operator based on difference(s) between pairs of elements
- similarities with older direct search methods:
  - pattern search (Hooke-Jeeves, 1961)
  - simplex methods (Nelder-Mead, 1965)
- other population based methods involving differences:
  - Particle Swarm Optimization (Kennedy & Eberhart, 1995)

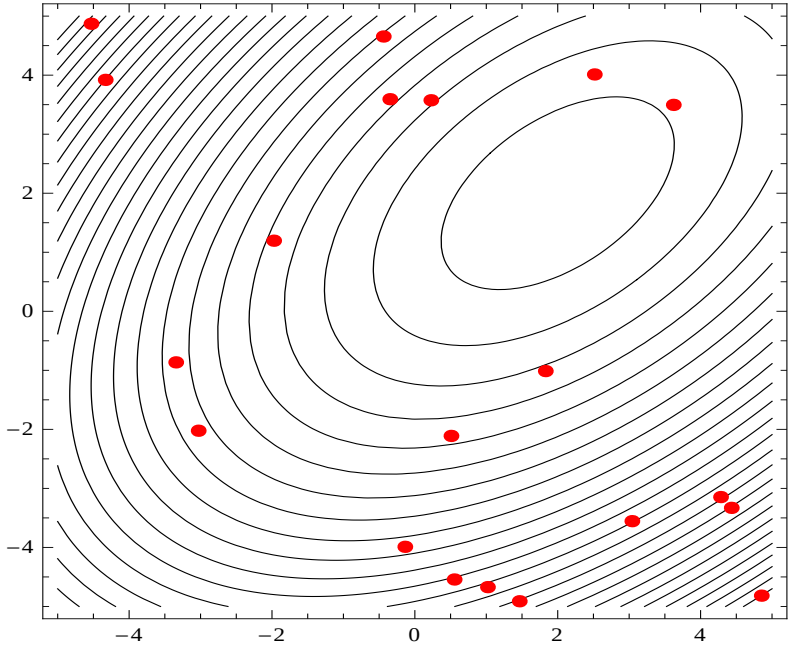
Standard Differential Evolution

Problem to be solved: minimize $f : [a_1, b_1] \times \ldots \times [a_n, b_n] \to \mathbb{R}$

DE in a phrase:

A population of m elements is randomly initialized in $[a_1, b_1] \times \ldots \times [a_n, b_n]$ and then iteratively transformed by applying difference based recombination and greedy-like selection

Initialization: $x_i = U(a_i, b_i), \qquad i = \overline{1, m}$



m - population size

Standard Differential Evolution

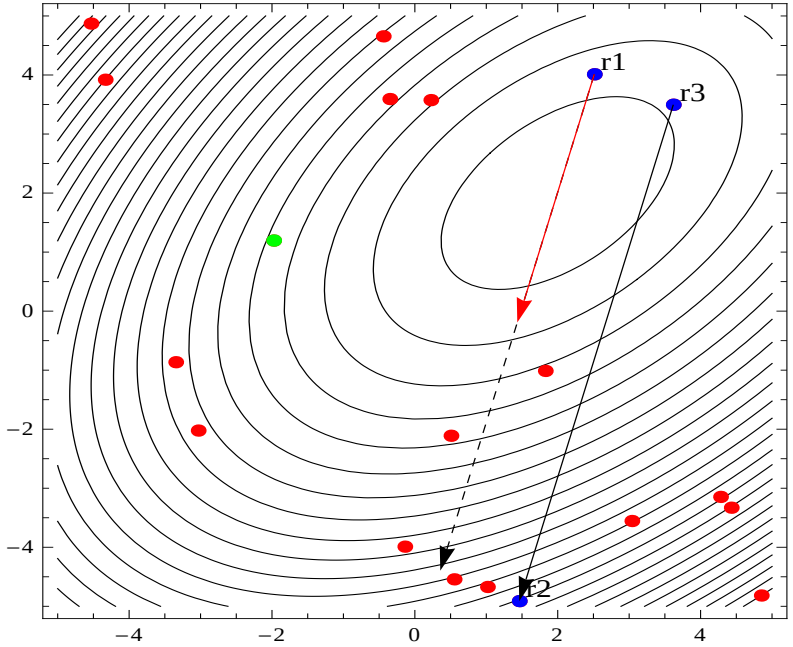
Problem to be solved: minimize $f : [a_1, b_1] \times \ldots \times [a_n, b_n] \to \mathbb{R}$

Initialization: $x_i = U(a_i, b_i), \qquad i = \overline{1, m}$

`while` ⟨ NOT termination ⟩ `do`

- Mutation - for each element x_i (target or current element) a mutant is constructed:

$$y_i = \underbrace{x_{r_1}}_{\text{base element}} + \underbrace{F \cdot (x_{r_2} - x_{r_3})}_{\text{scaled difference}}$$



m - population size

$F \in (0, 2)$ - scale factor

Standard Differential Evolution

Problem to be solved: minimize $f : [a_1, b_1] \times \ldots \times [a_n, b_n] \to \mathbb{R}$

Initialization: $x_i = U(a_i, b_i), \qquad i = \overline{1, m}$

`while` ⟨ NOT termination ⟩ `do`

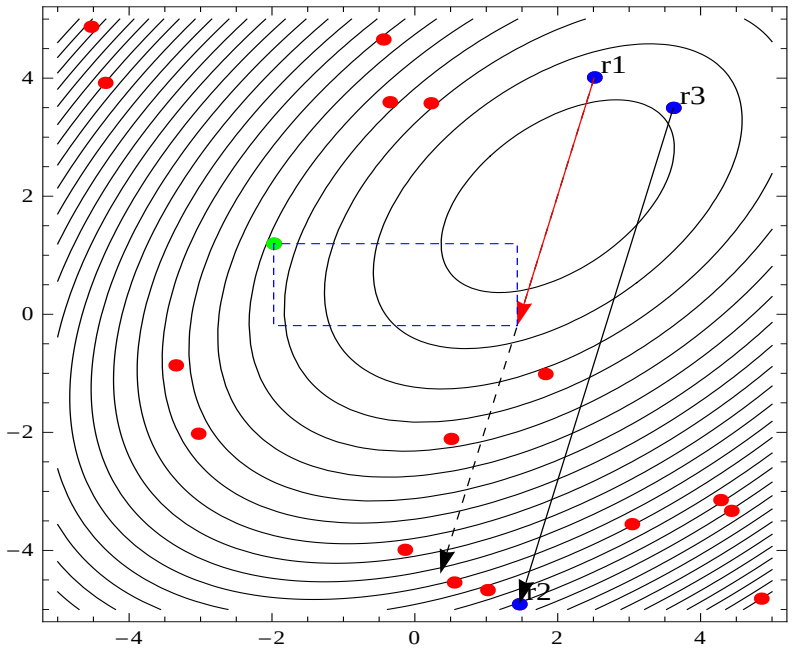
- Mutation:

$$y_i = x_{r_1} + F \cdot (x_{r_2} - x_{r_3}), \quad i = \overline{1, m}$$

- Crossover:

$$z_i^j = \begin{cases} y_i^j & \text{if } rand(0,1) < CR \text{ or } j = j_0 \\ x_i^j & \text{otherwise} \end{cases},$$

$$i = \overline{1, m}, j = \overline{1, n}$$



m - population size

$F \in (0, 2)$ - scale factor

$CR \in [0, 1]$ - crossover rate

j_0 - randomly selected component

Standard Differential Evolution

Problem to be solved: minimize $f : [a_1, b_1] \times \ldots \times [a_n, b_n] \to \mathbb{R}$

Initialization: $x_i = U(a_i, b_i), \qquad i = \overline{1, m}$

`while` ⟨ NOT termination ⟩ `do`

- Mutation:

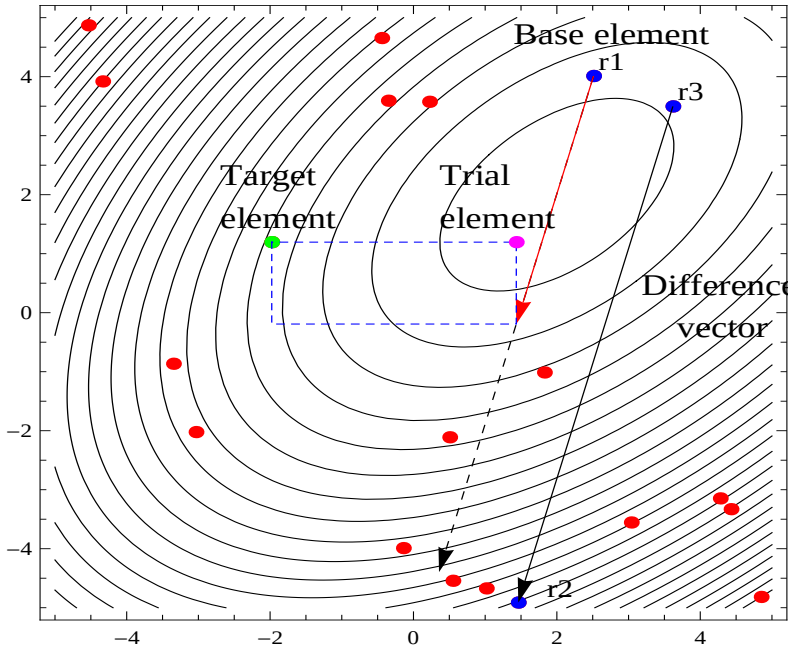
$$y_i = x_{r_1} + F \cdot (x_{r_2} - x_{r_3}), \quad i = \overline{1, m}$$

- Crossover:

$$z_i^j = \begin{cases} y_i^j & \text{if } rand(0,1) < CR \text{ or } j = j_0 \\ x_i^j & \text{otherwise} \end{cases},$$

- Selection:

$$x_i(g+1) = \begin{cases} z_i & \text{if } f(z_i) \le f(x_i(g)) \\ x_i^j & \text{if } f(z_i) > f(x_i(g)) \end{cases}$$



m - population size

$F \in (0, 2)$ - scale factor

$CR \in [0, 1]$ - crossover rate

j_0 - randomly selected component

Differential Evolution Variants

DE taxonomy: DE/ base element/ no. of differences/ crossover type

- Base element:
  - random(x_{r_1}): DE/rand/*/*
  - best (x_*): DE/best/*/*
  - combination of current and best elements ($\lambda x_* + (1-\lambda)x_i$): DE/current-to-best/*/*
  - combination of random and best elements ($\lambda x_* + (1-\lambda)x_{r_1}$): DE/rand-to-best/*/*
  - combination of current and random elements ($\lambda x_i + (1-\lambda)x_{r_1}$): DE/current-to-rand/*/*
- Number of differences: usually 1 (DE/*/1/*) or 2 (DE/*/2/*)
- Crossover type: binomial: DE/*/*/bin, exponential: DE/*/*/exp)

At least 20 DE variants ...

Mutation variants

- DE/rand/L/*

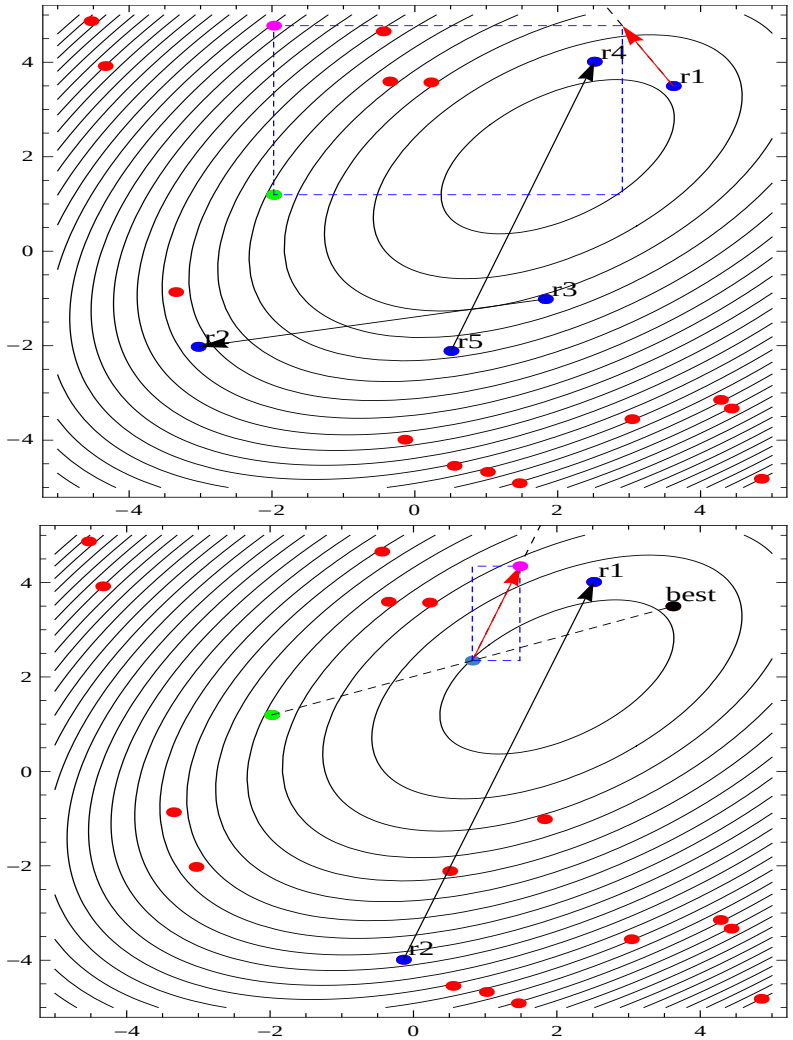
$$y_i = x_{r_1} + \sum_{l=1}^{L} F_l \cdot (x_{r_1(l)} - x_{r_2(l)})$$

- Typical variant: $L = 2$
- Allows to define new mutant directions ⇒ stimulates exploration

- DE/current-to-best/1

$$y_i = (1 - \lambda)x_i + \lambda x_* + F \cdot (x_{r_1} - x_{r_2})$$

- Introduces a bias toward the currently best element ⇒ stimulates exploitation



Crossover variants

- Binomial (DE/*/*/bin)

$$z_i^j = \begin{cases} y_i^j & \text{if } rand(0,1) < CR \text{ or } j = j_0 \\ x_i^j & \text{otherwise} \end{cases}, \qquad i = \overline{1,m}, j = \overline{1,n}$$

- Exponential (DE/*/*/exp)

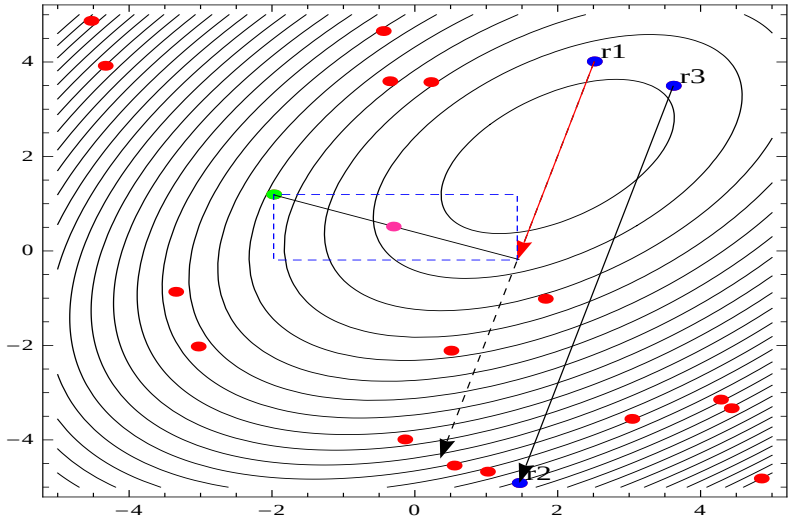
$$z_i^j = \begin{cases} y_i^j & \text{for } j \in \{j_0, \langle j_0+1\rangle_n, \ldots, \langle j_0+K-1\rangle_n\} \\ x_i^j & \text{otherwise} \end{cases}, i = \overline{1,m}, j = \overline{1,n}$$

$CR \in [0,1]$ - crossover rate, $j_0 \sim U(\{1,\ldots,m\})$, $K \sim Geom(CR)$

Other variants

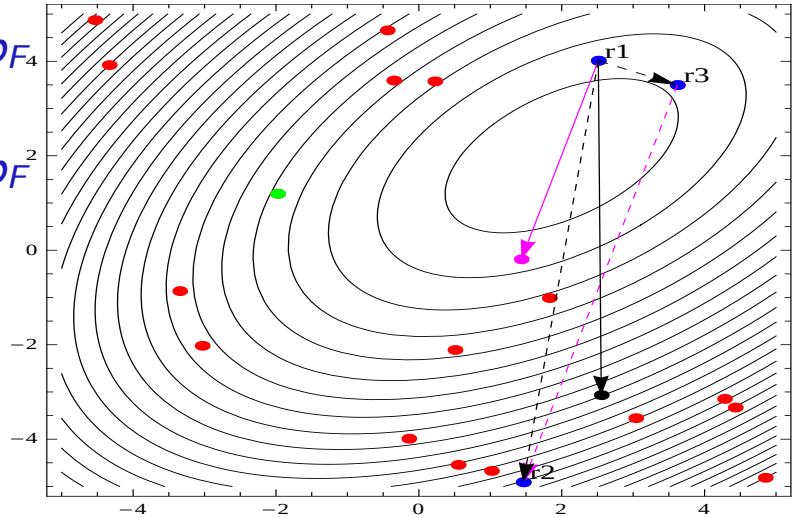
- Arithmetical (DE/*/*/arithmetical)

$$z_i = (1-q)x_i + q y_i, \qquad i = \overline{1, m}, \qquad q \in [0, 1]$$



- Either mutation or recombination (DE/either-or)

$$z_i = \begin{cases} x_{r_1} & +F \cdot (x_{r_2} - x_{r_3}) \quad \text{if } rand(0,1) \le p_F \\ x_{r_1} & +K \cdot (x_{r_2} - x_{r_1}) \\ & +K \cdot (x_{r_3} - x_{r_1}) \quad \text{if } rand(0,1) > p_F \end{cases}$$



- Remark: DE/either-or was created to compensate the lack of rotational invariance of DE involving binomial crossover

Which variant to choose ?

Recommendations

- no specific knowledge on the problem: use DE/rand/1/*
- need for an exploitative method: use DE/best/1/*
- need for a more explorative method: use DE/rand/2/*
- need for a rotationally invariant method: use DE/either-or

Remark: different variants could be appropriate in different stages of the optimization process $\Longrightarrow$ need for adaptation

(Self)adaptive variants

Self adaptation

- use a pool of variants and assign to each element one of these variants
- record the success/failure information of the variant attached to each element
- decide which variant to select based on the success/failure information (a probability distribution is usually constructed)
- self-adaptation of mutation/crossover is usually combined with self-adaptation of parameters
- Examples: SaDE [3], Competitive DE [4], EPSDE [5] etc.

[3] Qin et al., Differential Evolution Algorithm With Strategy Adaptation for Global Numerical Optimization, IEEE TEVC, 2009

[4] Tvrdik, Competitive Differential Evolution, Mendel 2006

[5] [Mallipeddi et al., Differential evolution algorithm with ensemble of parameters and mutation strategies, ASOC 2011]

Searching for Exploration - Exploitation Balance

Usage of differences ⇒ implicit adaptation of the amount of change in the population

- First stage of evolution: large differences ⇒ exploration
- Second stage of evolution: small differences ⇒ exploitation
- There is no clear separation between these stages

Searching for Exploration - Exploitation Balance

Usage of differences ⇒ implicit adaptation of the amount of change in the population

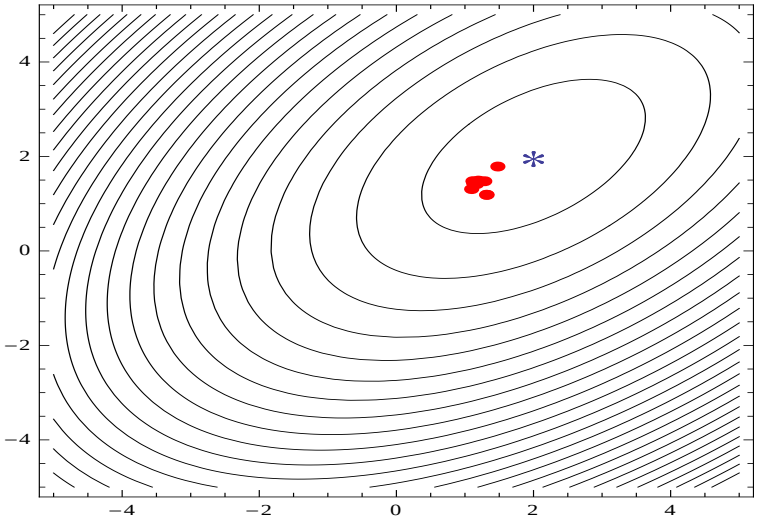
- First stage of evolution: large differences ⇒ exploration
- Second stage of evolution: small differences ⇒ exploitation
- There is no clear separation between these stages

Main difficulty: find the balance between exploration and exploitation:

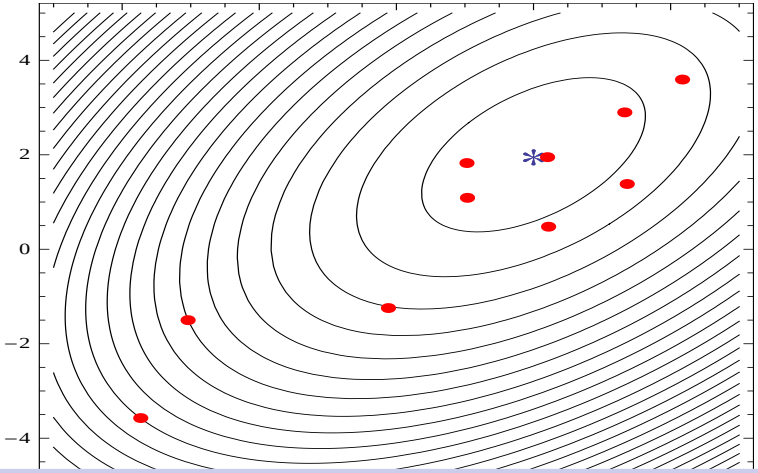
- less exploration, too much exploitation ⇒ premature convergence
- too much exploration ⇒ slow convergence

Searching for Exploration - Exploitation Balance

Population convergence:



Method convergence:

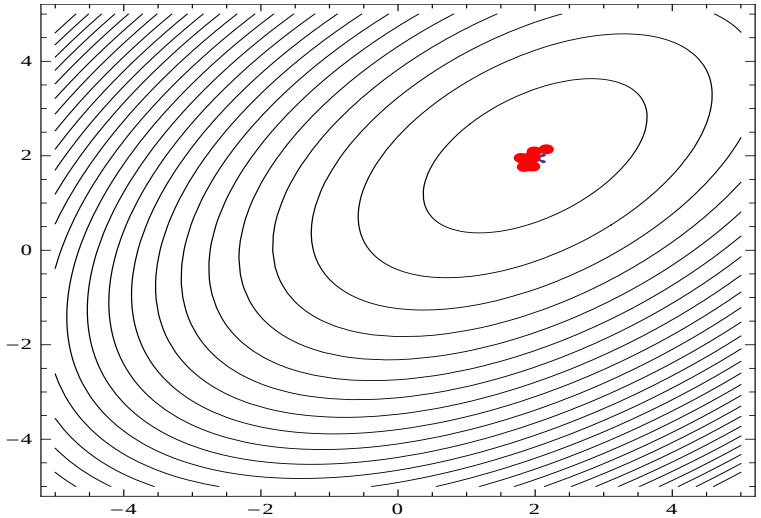


Ideal situation:

population convergence

is synchronized with

method convergence



Still unclear how to ensure this synchronization ...

Analysis of the DE behavior

Many empirical parameter studies led to statements as:

- for the same crossover rate (CR), the number of components taken from the mutant is highly depending on the crossover type (binomial vs. exponential) ... why ?
- the control parameters (m, F, CR) influence in an interrelated manner the population diversity ... how ?
- high values of the scale factor, F, are needed to avoid premature convergence ... does there exist a lower bound ?

Analysis of the DE behavior

Many empirical parameter studies led to statements as:

- for the same crossover rate (CR), the number of components taken from the mutant is highly depending on the crossover type (binomial vs. exponential) ... why ?
- the control parameters (m, F, CR) influence in an interrelated manner the population diversity ... how ?
- high values of the scale factor, F, are needed to avoid premature convergence ... does there exist a lower bound ?

It would be desirable to have theoretical results which explain such empirical remarks ...

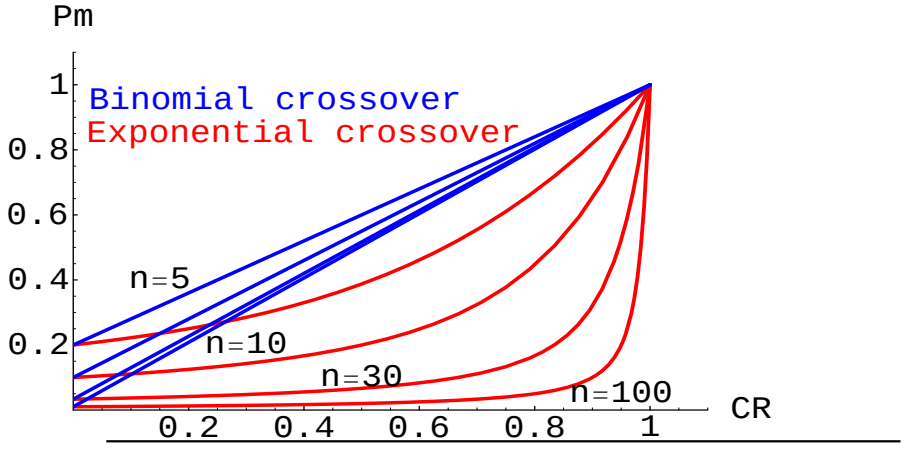
Binomial vs. Exponential Crossover[6]

Binomial crossover:

- the probability to take a component from the mutant vector is:

$$p_m = CR\left(1 - \frac{1}{n}\right) + \frac{1}{n}$$

- the number of mutated components: binomial distribution



Exponential crossover:

- the probability to take a component from the mutant vector is:

$$p_m = \frac{1 - CR^n}{n(1 - CR)}$$

- the number of mutated components: truncated geometric distribution

Remark: In the case of exponential crossover larger values of CR should be used in order to have the same number of mutated components as for binomial crossover .

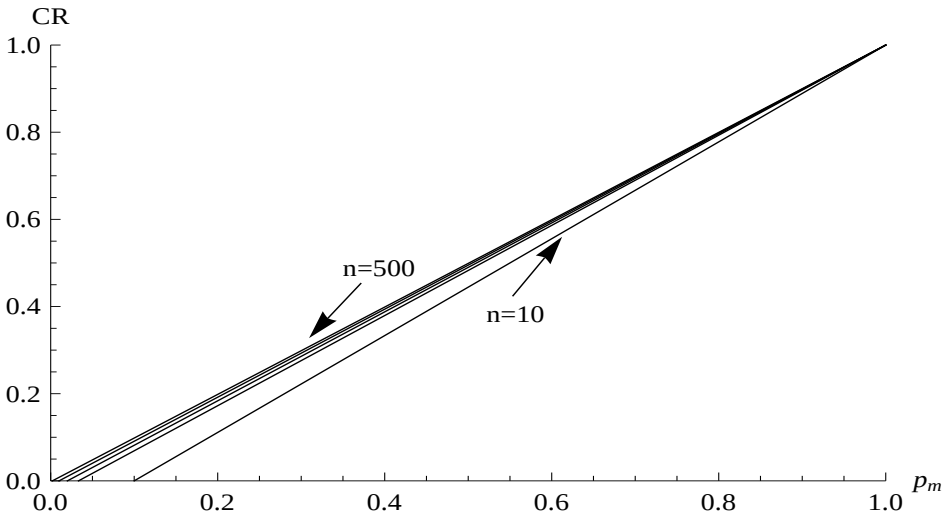
[6] D. Zaharie, Influence of crossover on the behavior of Differential Evolution Algorithms, ASOC 2009

Choice of crossover rate

- the DE behavior is influenced by the mutation probability, p_m, but the user provide a value for CR
- what value should have CR in order to ensure a given value for p_m?

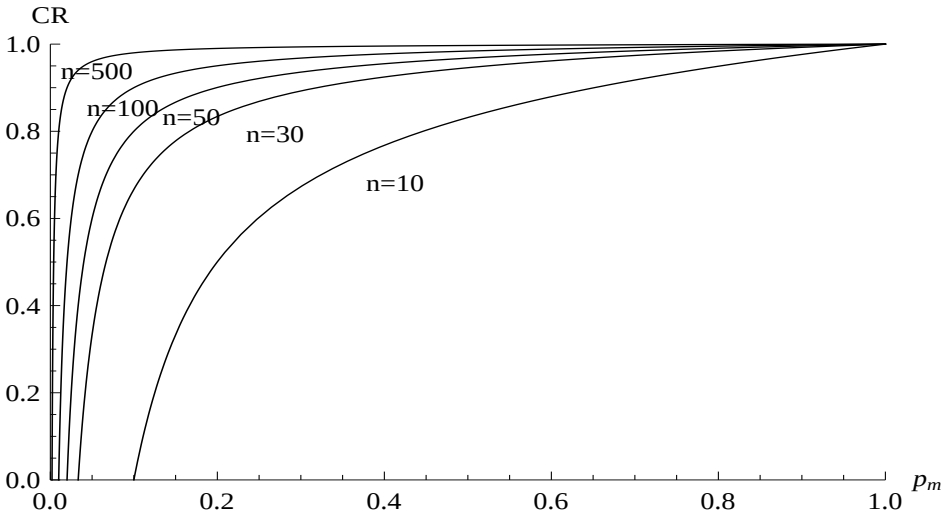
Binomial crossover

$CR = \frac{p_m - 1/n}{1 - 1/n}$



Exponential crossover

$CR^n - np_m CR + np_m - 1 = 0$

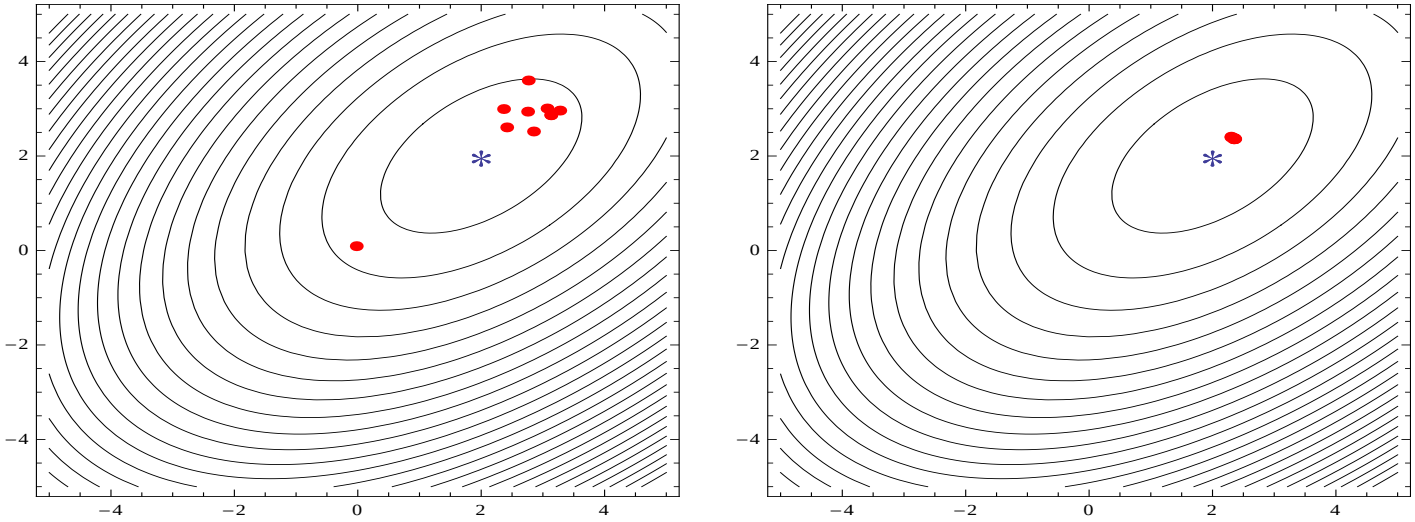


Practical remark: Exponential crossover is more sensitive to the problem size

Population diversity

Importance

- small diversity in the DE population ⇒ small values of the differences ⇒ limited progress ⇒ premature convergence (all population concentrated in a point which is NOT the optimum)



Question: What is the impact of mutation and crossover on the population diversity ?

Population diversity

Theoretical results [7]

- Diversity measure: population variance (component level)
  $Var(X) = \sum_{i=1}^{m}(x_i - \overline{x}_i)^2/m$
- Notations: $Var(X)$=variance of current population;
  $\mathbb{E}(Var(Z))$=expected variance of the trial population
- DE/rand/L/*

$$\mathbb{E}(Var(Z)) = \left(1 + 2p_m \sum_{l=1}^{L} F_l^2 - \frac{p_m(2-p_m)}{m}\right) Var(X)$$

[7] D. Zaharie, Critical values for the control parameters of differential evolution algorithms, Mendel 2002

Population diversity

Theoretical results[8]

- Diversity measure: population variance (component level)
  $Var(X) = \sum_{i=1}^{m}(x_i - \overline{x}_i)^2/m$
- Notations: $Var(X)$=variance of current population;
  $\mathbb{E}(Var(Z))$=expected variance of the trial population
- DE/random-to-best/1/*

$$\mathbb{E}(Var(Z)) = \left(1 + 2p_m F^2 - \frac{p_m(2-p_m)}{m} - \lambda p_m^2 \frac{m-1}{m}\right) Var(X)$$
$$+\lambda^2 \frac{p_m(1-p_m)}{m} \sum_{i=1}^{m}(x_* - x_i)^2$$

[8] D. Zaharie, Statistical Properties of Differential Evolution and Related Random Search Algorithms, CompStat 2008

Population diversity

Theoretical results

- Diversity measure: population variance (component level) $Var(X) = \sum_{i=1}^{m}(x_i - \overline{x}_i)^2/m$
- Notations: Var(X)=variance of current population; $\mathbb{E}(Var(Z))$=expected variance of the trial population
- DE/current-to-rand/1 (arithmetical recombination)

$$\mathbb{E}(Var(Z)) = \left(1 + 2F^2 - 2q + \frac{2m-1}{m}q^2\right) Var(X)$$

Population diversity

Theoretical results[9]

- Diversity measure: population variance (component level)
- Notations: $Var(X)$=variance of current population; $\mathbb{E}(Var(Z))$=expected variance of the trial population
- DE/either-or

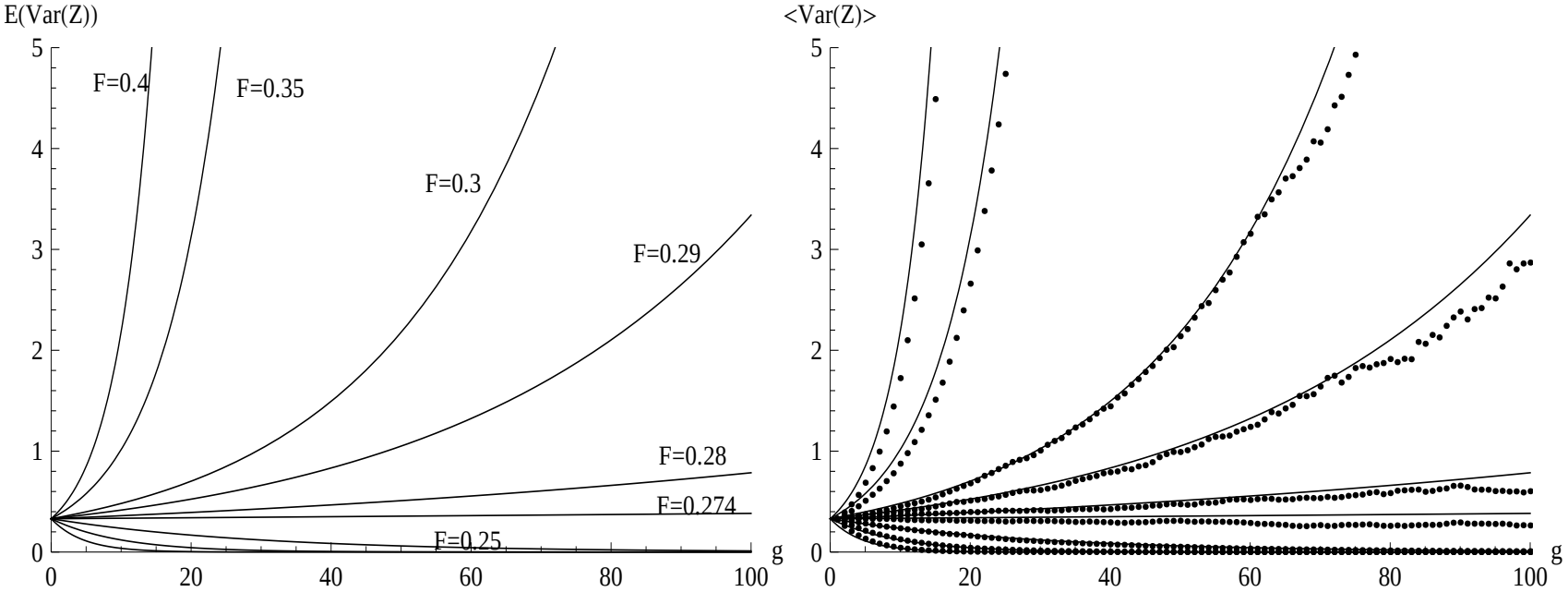
$$\begin{aligned}\mathbb{E}(Var(Z)) &= \left(p_F^2(1+2F^2-\tfrac{1}{m})+2p_F(1-p_F)(\tfrac{m-1}{m}+F^2+3K^2-2K)\right.\\ &\quad \left.+(1-p_F)^2\left(\tfrac{m-1}{m}+2\tfrac{m-2}{m}(3K^2-2K)\right)\right)Var(X)\end{aligned}$$

[9] D. Zaharie, Differential Evolution. From Theoretical Analysis to Practical Insights, Mendel 2012

Population diversity

Theoretical vs empirical evolution

- Evolution of population variance after mutation and crossover (no selection)
- Practical remark: the population variance can decrease even in the absence of selection pressure



DE/either-or

Population diversity

From theory to practical insights

$$\mathbb{E}(Var(Z)) = c(F, CR, p_F, q, m, n)Var(X)$$

Population diversity

From theory to practical insights

$$\mathbb{E}(Var(Z)) = c(F, CR, p_F, q, m, n)Var(X)$$

- if $c(F, CR, p_F, q, m, n) < 1$ the algorithm will probably prematurely converge

Population diversity

From theory to practical insights

$$\mathbb{E}(Var(Z)) = c(F, CR, p_F, q, m, n)Var(X)$$

- if $c(F, CR, p_F, q, m, n) < 1$ the algorithm will probably prematurely converge
- one can control the impact which mutation and crossover have on the population variance by changing the values of the parameters involved in the factor c

Population diversity

From theory to practical insights

$$\mathbb{E}(Var(Z)) = c(F, CR, p_F, q, m, n)Var(X)$$

- if $c(F, CR, p_F, q, m, n) < 1$ the algorithm will probably prematurely converge
- one can control the impact which mutation and crossover have on the population variance by changing the values of the parameters involved in the factor c
- this is a particularity of DE, as in EAs using mutation based on additive perturbation involving an arbitrary distribution:

$$\mathbb{E}(Var(Z)) = aVar(X) + b$$

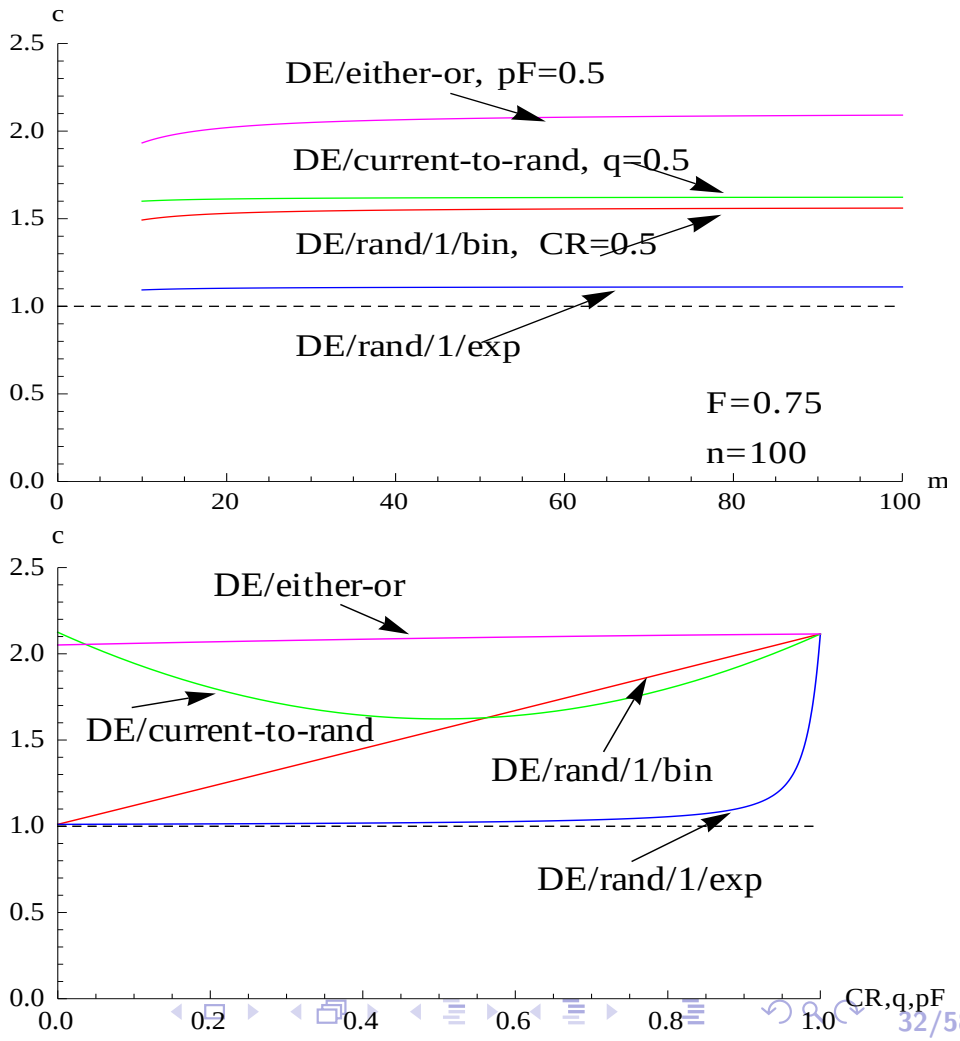
with b not necessarily zero

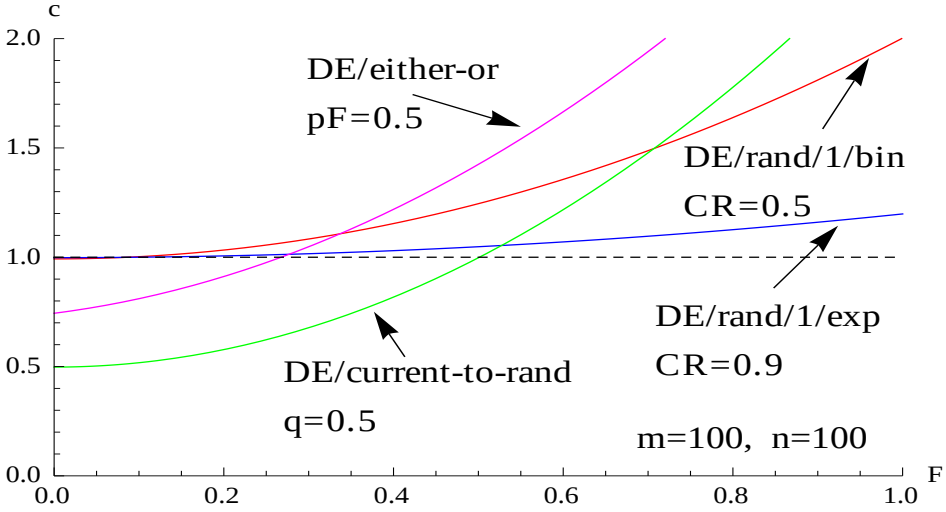
Population diversity

From theory to practical insights

$$\mathbb{E}(Var(Z)) = c(F, CR, p_F, q, m, n)Var(X)$$

- the value of $c(F, CR, p_F, q, m, n)$ is highly influenced by the type of mutation and crossover

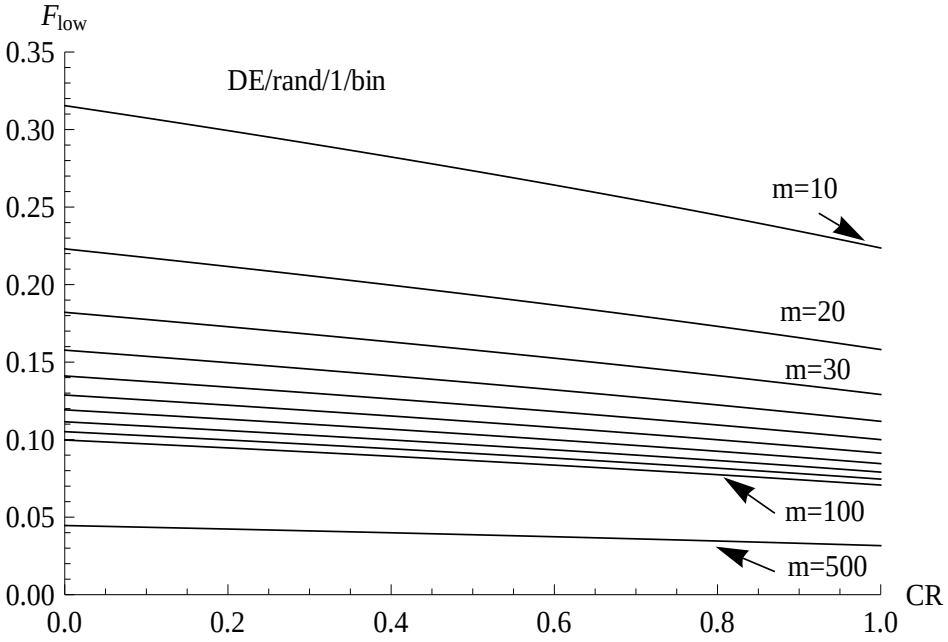




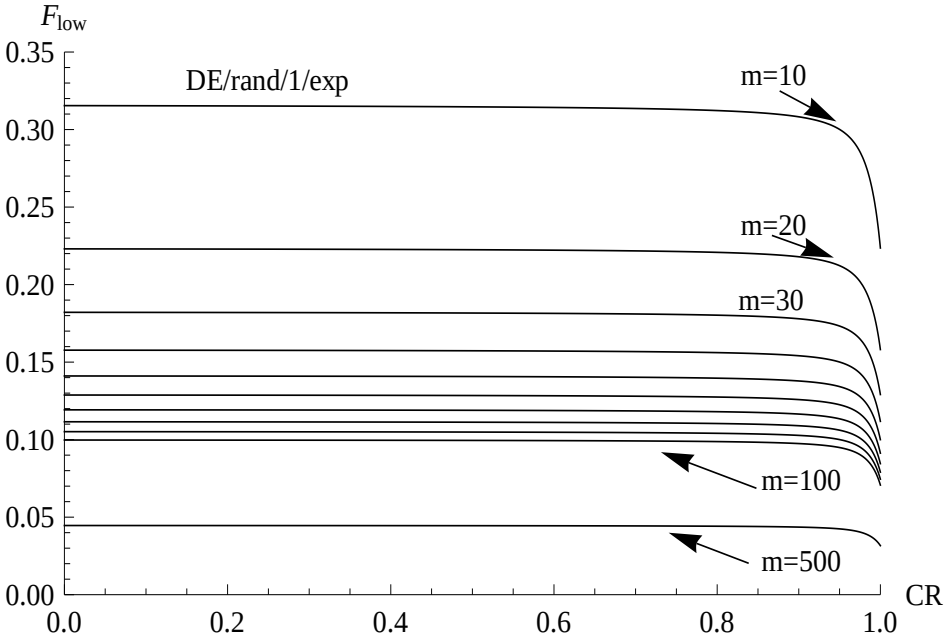
Population diversity

Avoiding premature convergence

- choose the DE control parameters (F, CR, m etc) such that the population diversity does not decrease too fast ($c(CR, F, q, m, n) > 1$)
- by solving $c(F, CR, p_F, q, m, n) = 1$ we can find a lower bound for F under which the population variance decreases even in the absence of selection



DE/rand/1/bin

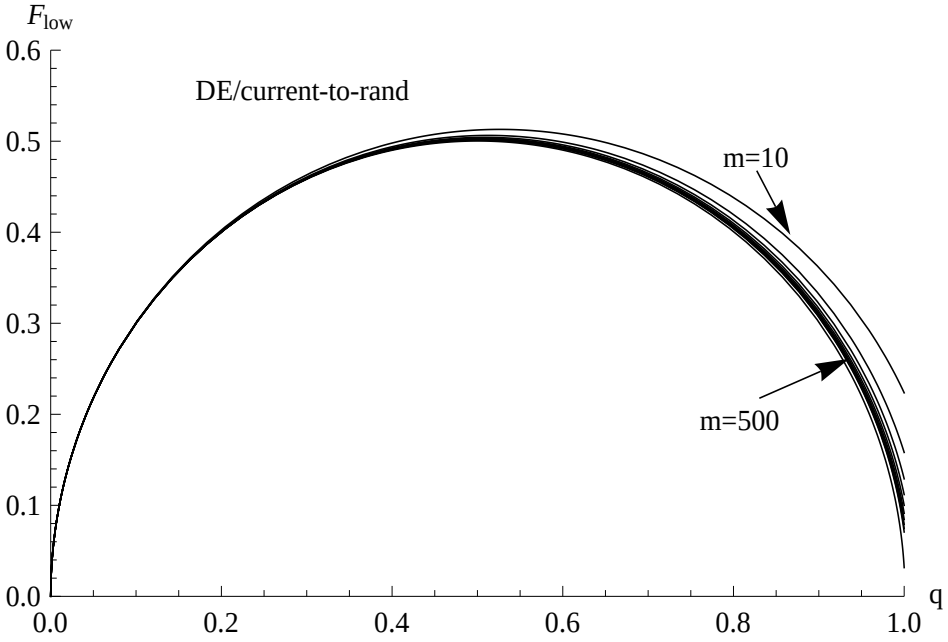


DE/rand/1/exp

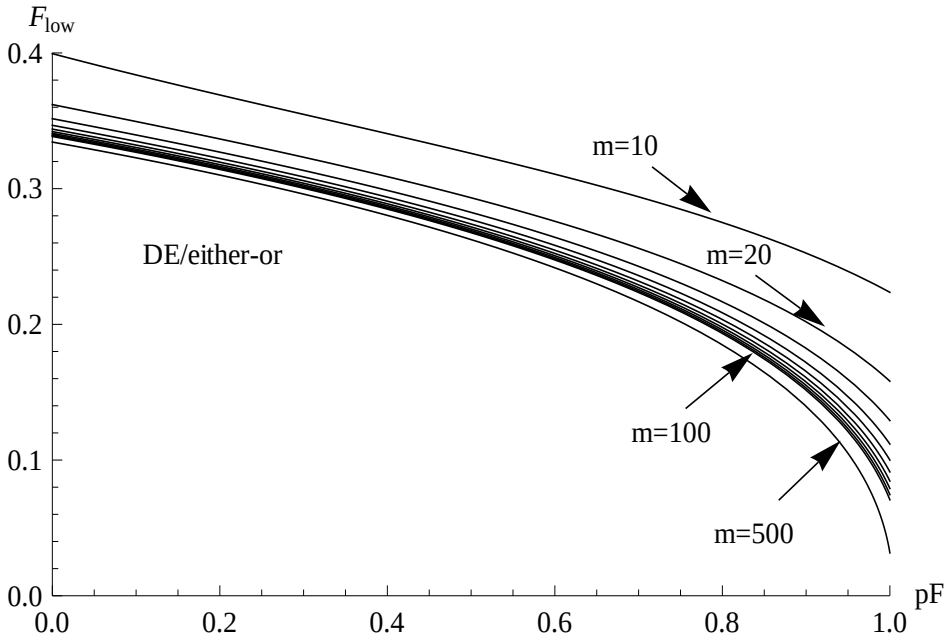
Population diversity

Avoiding premature convergence

- choose the DE control parameters (F, CR, m etc) such that the population diversity does not decrease too fast ($c(CR, F, q, m, n) > 1$)
- by solving $c(F, CR, p_F, q, m, n) = 1$ we can find a lower bound for F under which the population variance decreases even in the absence of selection



DE/current-to-rand



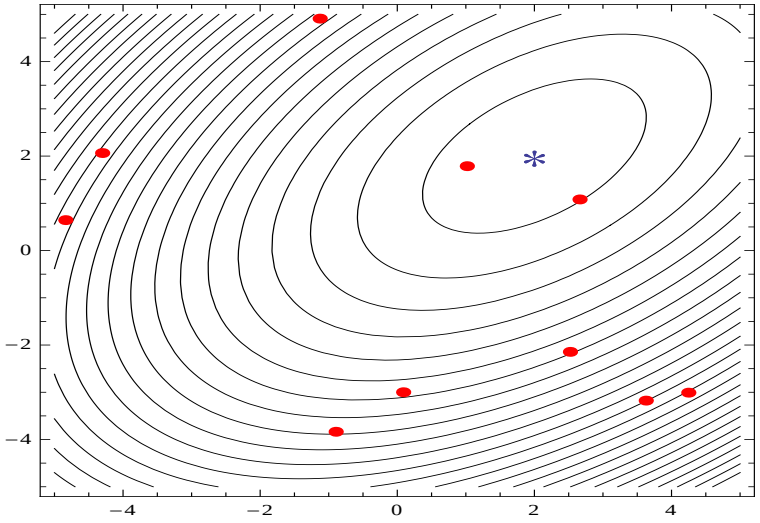
DE/either-or

Population diversity

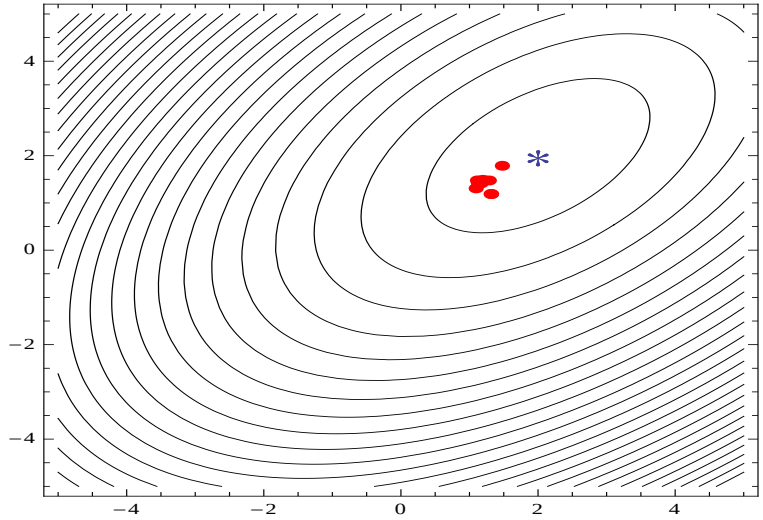
Avoiding premature convergence

Example:

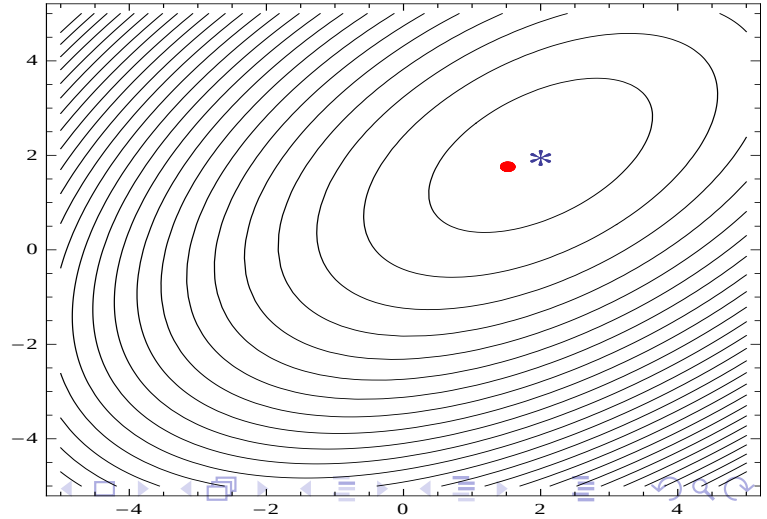
- DE/rand/1/bin for Neumaier fct, $n = 2$
- $m = 20$, $CR = 0.9$, $F = 0.2$
- lower bound $F_{low} = 0.23$



gen=1



gen=5



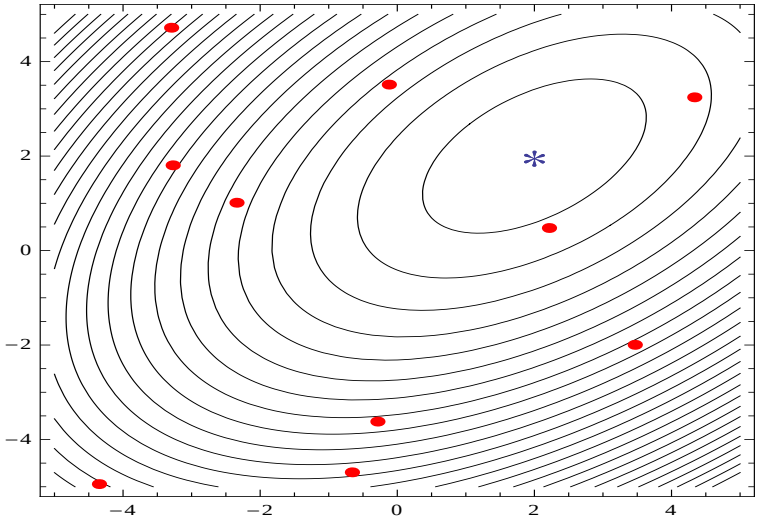
gen=15

Population diversity

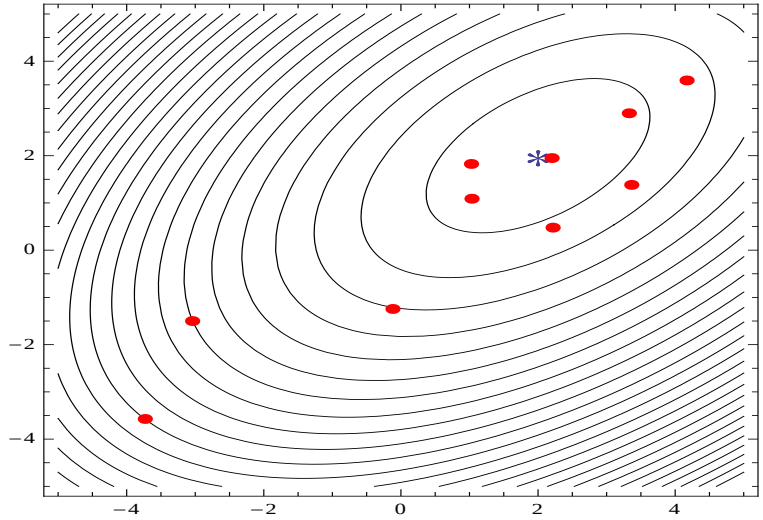
Avoiding premature convergence

Example:

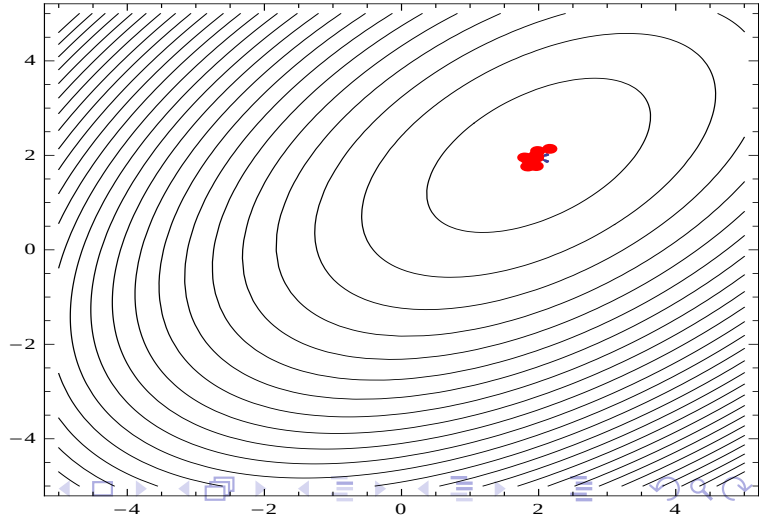
- DE/rand/1/bin for Neumaier fct, $n = 2$
- $m = 20$, $CR = 0.9$, $F = 0.5$
- lower bound $F_{low} = 0.23$



gen=1



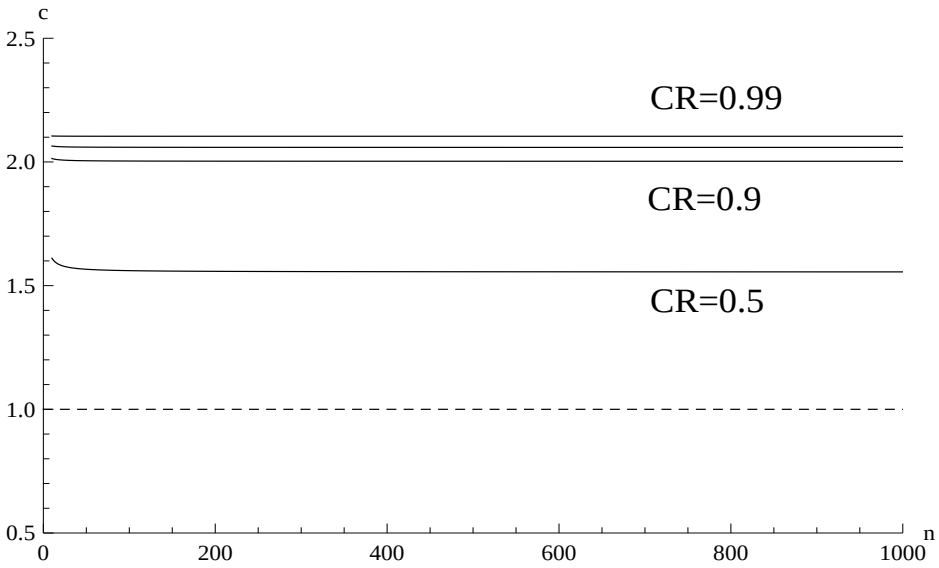
gen=5



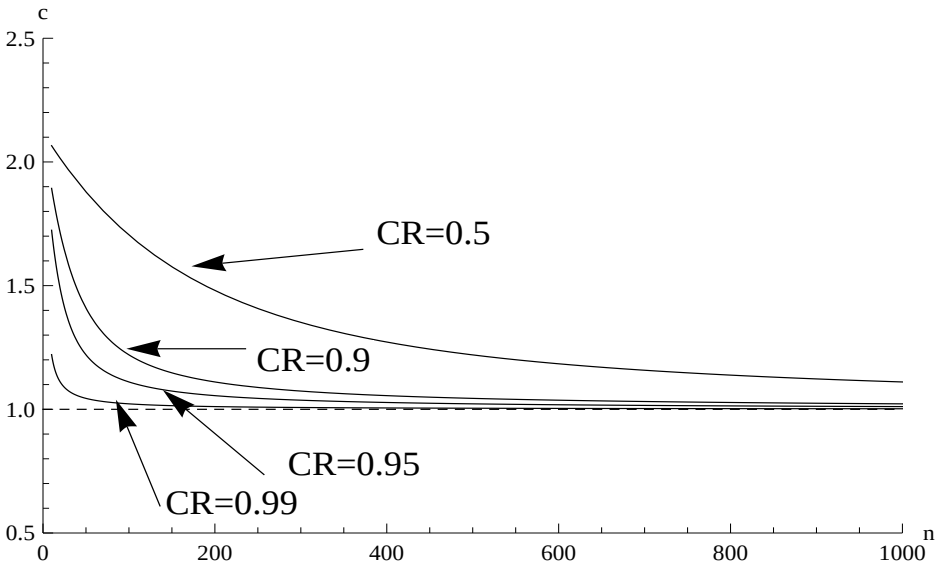
gen=15

High-dimensional problems

- The problem size influences directly the relationship between p_m and CR (especially for exponential crossover)
  - CR values tuned for small size problems are not necessarily good for large size problems



DE/rand/1/bin $F = 0.75$



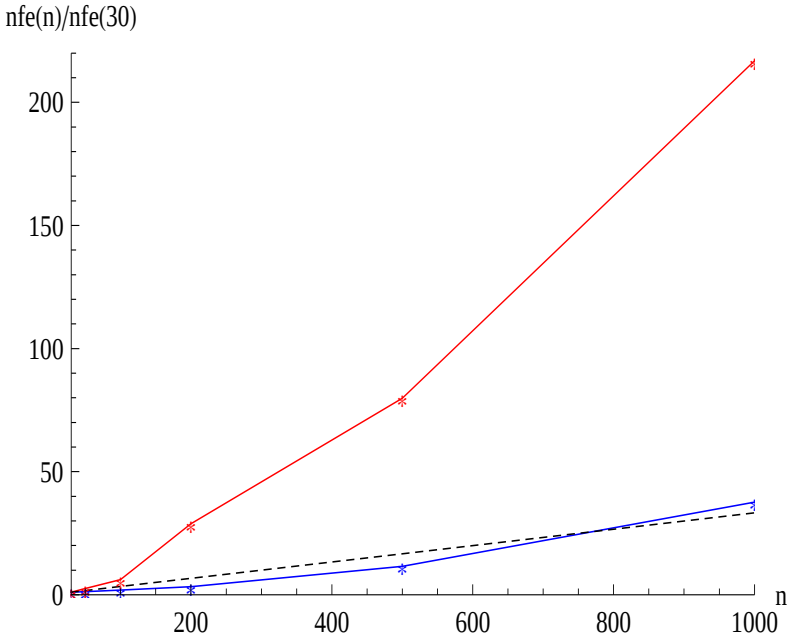
DE/rand/1/exp

Scalability issue

- a large number of variables means a larger search space
- asks for a larger volume of resources (larger populations and/or longer evolution time) to reach the same performance
- the performance of the search method may deteriorate as the problem size increases
- Scalable method: the volume of resources grows (almost) linearly with the problem size

Addressing the scalability issue:

- design new evolutionary operators and/or control parameter adaptation
- use a "divide and conquer" approach $\Longrightarrow$ cooperative coevolution



Measure of used resources: number of function evaluations (nfe)

Outline

Cooperative coevolution

Main idea: split the problem into smaller sub-problems

- a potential solution consists of several components
- evolve independently the population corresponding to each component (coevolution)
- each component is evaluated in the context of other components (cooperation)

Implementation issues

- Choosing the components
  - how many components?
  - how to assign a variable to a component?
- Components coevolution
  - how to construct the evaluation context for each component?
  - how long should be the evolution of a component in the same context?

Components

- each solution $x = (x_1, x_2, \ldots, x_n)$
- should be decomposed into components $x = \langle C_1(x), C_2(x), \ldots, C_K(x) \rangle$
- where
  - K is the number of components
  - $C(x)$ is a set of variables (not necessarily consecutive in the solution vector)
  - $\langle \cdot \rangle$ denotes the merging operation

Choosing the components

There are two main decisions to take:

- choose the number, K, of components
- define an assignment function of variables to components: $c : \{1, \ldots, n\} \rightarrow \{1, \ldots, K\}$

Variants:

- Simplest case [10]:
  $K = n, c(i) = i, C_i(x) = x_i$ (each component corresponds to one variable)
  - similar to line search techniques; adequate for separable problems
  - the behavior for nonseparable problems can be improved by choosing the adequate context

[10] Potter & deJong, A Cooperative Coevolutionary Approach to Function Optimization, PPSN 1994

Separability vs. nonseparability

- A problem is separable if its components do not interact (they are uncorrelated)
- The relative quality of two values of the same component does not depend on the context:
  - if $f(x_1, \ldots, x_i, \ldots, x_n) < f(x_1, \ldots, x_i', \ldots, x_n)$ then $f(y_1, \ldots, x_i, \ldots, y_n) < f(y_1, \ldots, x_i', \ldots, y_n)$ for any context y
  - i.e. if x_i is better than x_i' in a given context then it is better in any context, thus ...
  - in this case the choice of the context is not critical
- Example (additively separable): $f(x_1, x_2, \ldots, x_n) = \sum_{i=1}^{n} f_i(x_i)$

Separability vs. nonseparability

- Nonseparable functions: the variables are correlated
- Example: $f(x_1, x_2) = 100(x_1 - x_2)^2 + (1 - x_1)^2$
- Some simple computations lead to:
  - $f(1, x_2) < f(2, x_2)$ if $x_2 < 2.501$
  - $f(1, x_2) > f(2, x_2)$ if $x_2 > 2.501$
- In this case the relative quality of values 1 and 2 for x_1 depends on the context represented by x_2
- Thus, the context used to evaluate the quality of a component is important; for this example the variables should be evolved together.
- Fully nonseparable functions: each variable interacts with at least another variable
- Real world problems: partially separable

Choosing the components

- Ideal case: each component contains a group of highly interacting variables
  - difficult to identify the groups of highly correlated variables for black box optimization
- Compromise variant: assign the variables to components in a random manner[11]
  - K is randomly chosen
  - equally sized components: $c(i) = \lfloor \sigma^{-1}(i) \cdot \frac{K}{n} \rfloor + 1$, $\sigma = (\sigma(1), \ldots, \sigma(n))$ is a random permutation
  - components with variable size: $c(i) = rand(\{1, \ldots, K\})$

[11] Yang et al, Large scale evolutionary optimization using cooperative coevolution, Inf. Sci., 2008

Random choice of components (motivation[12])

The probability of assigning two interacting variables x_i and x_j to the same component in at least g cycles out of the total number of cycles, G:

$$P_g = \sum_{r=g}^{G} \binom{G}{r} \left(\frac{1}{K}\right)^r \left(1 - \frac{1}{K}\right)^{G-r}$$

Remarks.

- For $K = 10$ and $G = 50$: $P_1 = 0.9948$, $P_2 = 0.9662$
- any two variables (correlated or not) have a high chance to belong to the same component at least for a few number of cycles.

[12] Yang et al, Large scale evolutionary optimization using cooperative coevolution, Inf. Sci., 2008

Random choice of components (limits[13])

The probability of assigning ν interacting variables x_i and x_j to the same component in at least g cycles out of the total number of cycles, G:

$$P_g = \sum_{r=g}^{G} \binom{G}{r} \left(\frac{1}{K^{\nu-1}}\right)^r \left(1 - \frac{1}{K^{\nu-1}}\right)^{G-r}$$

Remarks.

- For $K = 10$, $G = 1000$, $\nu = 5$: $P_1 = 0.0951$, $P_2 = 0.0046$
- the chance of placing more than 4 interacting variables in the same component is small

[13] Omidvar et al., Cooperative Co-evolution for large scale optimization through more frequent random grouping, CEC 2010

Delta grouping[14]

Idea:

- the improvement interval decreases when there are interactions $\Longrightarrow$ variables with small changes are interacting
- compute the difference between the population centroids at two consecutive generations: $\Delta = (\delta_1, \delta_2, \ldots, \delta_n)$
- sort Δ and split the set of variables in equally sized components

Disadvantages:

- the number of components should be specified
- not appropriate if there is more than one subcomponent of interacting variables

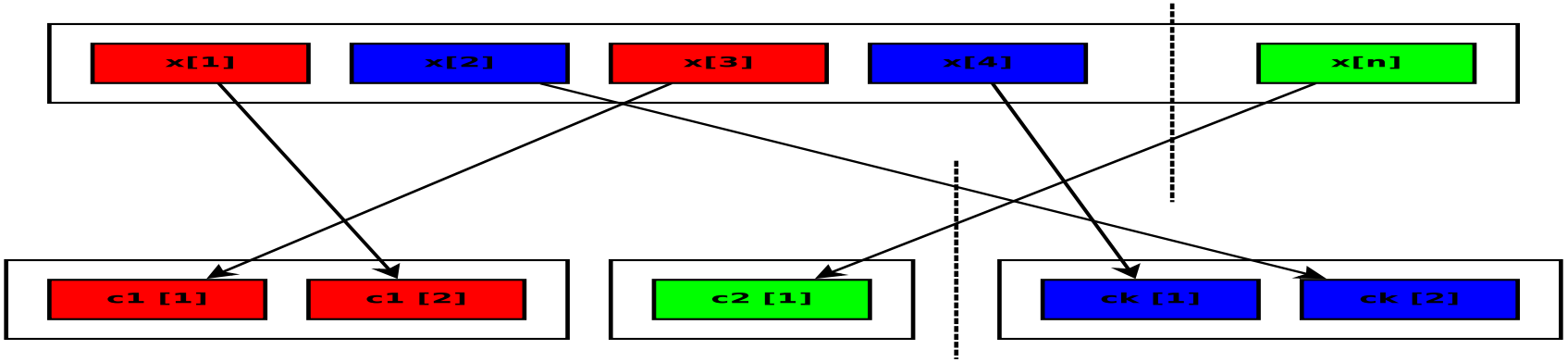
[14] M. Omidvar et al., Cooperative co-evolution with delta grouping for large scale for non-separable function optimization, CEC 2010

Differential grouping[15]

- incremental construction of components by identifying pairs of interacting variables
- two variables i and j satisfying (for an arbitrary selected vector x and arbitrary perturbations δ_i and δ_j)

$$f(\ldots, x_i + \delta_i, \ldots, x_j + \delta_j, \ldots) - f(\ldots, x_i + \delta_i, \ldots, x_j, \ldots) \neq$$
$$f(\ldots, x_i, \ldots, x_j + \delta_j, \ldots) - f(\ldots, x_i, \ldots, x_j, \ldots)$$

are considered interacting variables



[15] M. Omidvar et al., Cooperative Co-evolution with Differential Grouping for Large Scale Optimization, TEVC 2013

Choosing the evaluation context

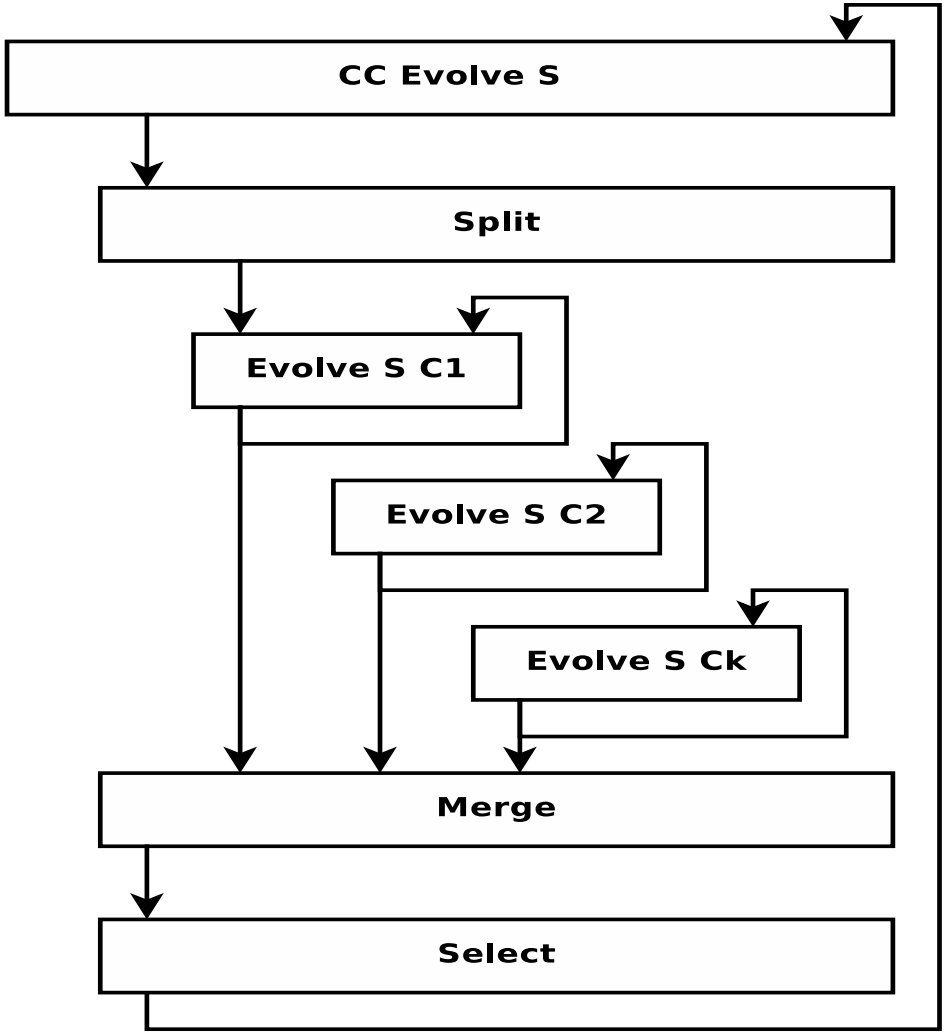
- to evaluate a component c_k one have to construct a virtual full solution, $\langle c_1, \ldots, c_k, \ldots, c_K \rangle$, by defining an evaluation context consisting of collaborators c_i provided by the subpopulations corresponding to each other components
- a collaborator, c_i, can be:
  - the current best value of component i (best individual of the subpopulation corresponding to ith component)
  - the ith component of the best individual in the entire population (it is not necessarily composed of the best components)
  - the ith component of a random individual
  - the ith component of the current individual (that from which the component c_k was evolved)

Implementation issues

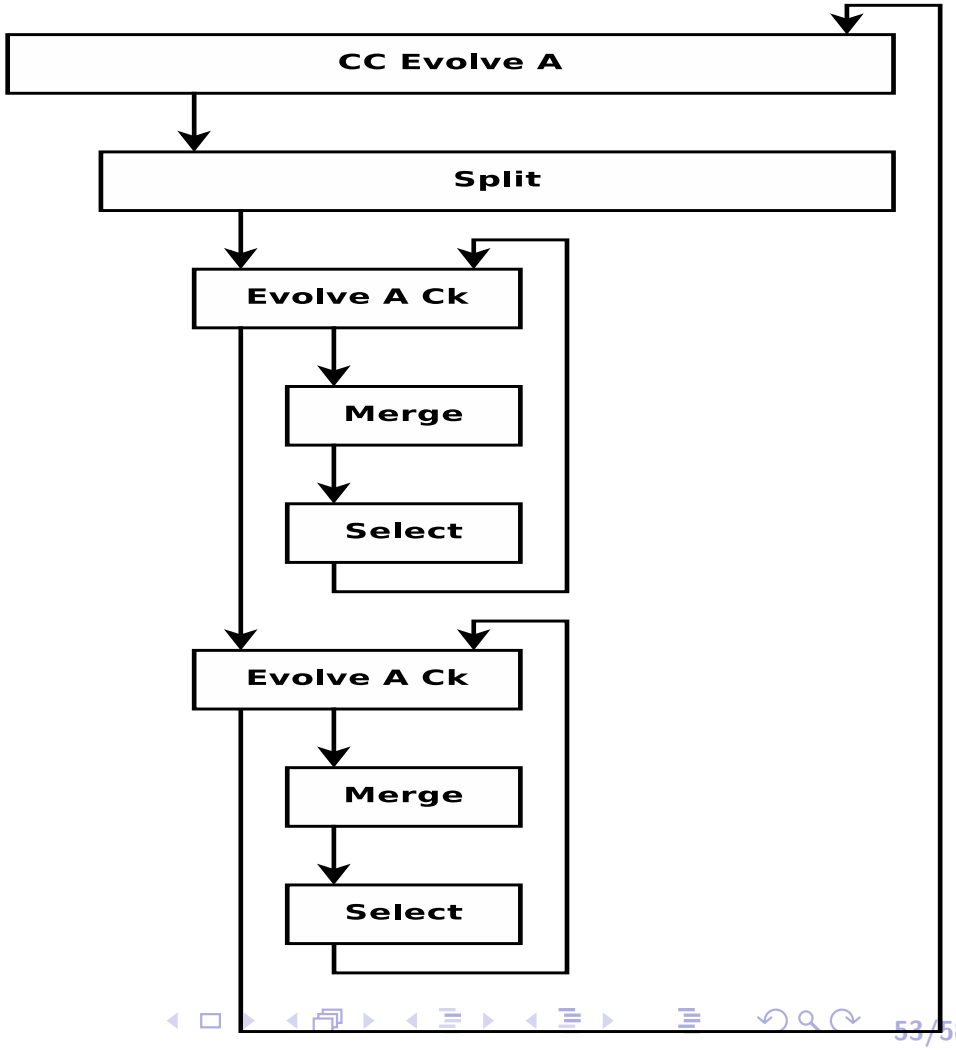
- most evolutionary algorithms can be
  - synchronous
  - asynchronous
- the same can be said about the Cooperative Coevolution framework
  - synchronous: the population used to provide the context is updated only after all components were evolved
  - asynchronous: the population used to provide the context is updated after the evolution of each component

sCC Synchronous vs Asynchronous

Synchronous



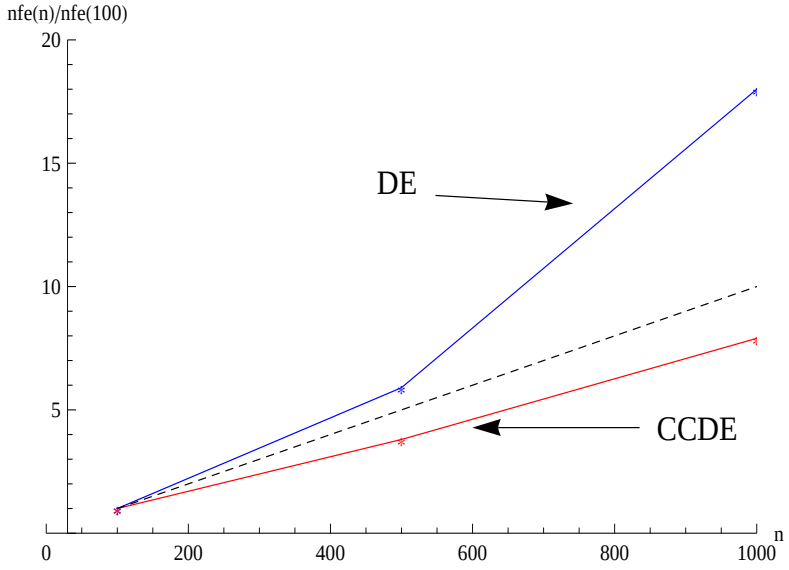
Asynchronous



Results[16]

- analysis approaches:
  - analyze approximation accuracy for increased computational budget $nfe(kn)/nfe(n) = k$
  - analyze the scalability factor $nfe(kn)/nfe(n)$ for a given approximation accuracy (e.g. $\epsilon = 10^{-10}$)

Scalability factor



Results:

- no significant differences between synchronous and asynchronous approach
- no critical impact of the context choice (current element context is slightly better)
- cooperative coevolution enhances the scalability

[16] C. Craciun, M.Nicoara, D. Zaharie, Enhancing the scalability of metaheuristics by cooperative coevolution, LSSC 2009

Parallelization models

- Objective function evaluation ⇒ master-slave model
  - the master process executes the iterative process
  - the slaves only evaluate the population elements (e.g. simulation of a process, costly computations)

Parallelization models

- Objective function evaluation ⇒ master-slave model
  - the master process executes the iterative process
  - the slaves only evaluate the population elements (e.g. simulation of a process, costly computations)
- Large size population ⇒ island model
  - the population is divided into several subpopulations on which the same or different algorithms are executed
  - the subpopulations communicate by transferring elements according to a given topology

Parallelization models

- Objective function evaluation ⇒ master-slave model
  - the master process executes the iterative process
  - the slaves only evaluate the population elements (e.g. simulation of a process, costly computations)
- Large size population ⇒ island model
  - the population is divided into several subpopulations on which the same or different algorithms are executed
  - the subpopulations communicate by transferring elements according to a given topology

Particularities in the case of coevolution:

- context broadcasting instead of a topology based migration
- the communication frequency depends on the sensitivity to the context

Summary

- DE is an easy to implement, (mostly) effective and an (almost) efficient algorithm for black-box optimization
  - important aspect: choice of the operators and parameters
- cooperative coevolution is a robust approach for enhancing scalability
  - key issue: choice of the components in accordance with the interaction between variables
- most of the current results are reported for problem sizes around 1000

Summary: DE-based image processing

- Image registration
  - Aim: Estimate the parameters of a rigid transformation
  - Optimization problem: minimizes the disimilarity between scene and model images
  - Method: DE/rand-to-best/1/exp [17], (DE/rand/1/bin + DE/rand/1/exp)+AIS [18]

[17] Salomon et al., 2005
[18] Santamaria et al., 2012
[19] Osuna-Enciso et al., 2013

Summary: DE-based image processing

- Image registration
  - Aim: Estimate the parameters of a rigid transformation
  - Optimization problem: minimizes the disimilarity between scene and model images
  - Method: DE/rand-to-best/1/exp [17], (DE/rand/1/bin + DE/rand/1/exp)+AIS [18]
- Segmentation through multi-level thresholding
  - Aim: estimate threshold values starting from of a mixture of Gaussians which matches the image histogram
  - Optimization problem: minimizes the Hellinger distance between the mixture of Gaussians and the image histogram
  - Population elements: parameters of each gaussian (mean and standard deviation) and apriori probability of each class [19]
  - Method: DE/best/1/bin

[17] Salomon et al., 2005
[18] Santamaria et al., 2012
[19] Osuna-Enciso et al., 2013

Summary: DE-based image processing

- Segmentation through deformable models
  - Aim: Localization of objects in an image
  - Optimization problem: maximizes the similarity between an object and the model or minimizes an energy function
  - Population elements: coordinates of an active net [20], parameters of a deformable model [21]
  - Method: DE/rand/1/bin

[20] Novo et al., 2012
[21] Mesejo et al., 2013
[22] Das& Kumar, 2009

Summary: DE-based image processing

- Segmentation through deformable models
  - Aim: Localization of objects in an image
  - Optimization problem: maximizes the similarity between an object and the model or minimizes an energy function
  - Population elements: coordinates of an active net [20], parameters of a deformable model [21]
  - Method: DE/rand/1/bin
- Segmentation through clustering
  - Aim: Identifying homogeneous regions in the image (the number of regions is not apriori known)
  - Optimization problem: maximizes a clustering validity index
  - Population elements: cluster centroids and cluster activation values [22]
  - Method: DE/rand/1/bin with adaptive parameters

[20] Novo et al., 2012
[21] Mesejo et al., 2013
[22] Das& Kumar, 2009

West University of Timisoara, Romania

www.uvt.ro

...in figures

- 11 faculties
- 82 undergradute programs
- 135 master programs
- 10 doctoral school
- 15000 students

Universitatea de Vest din Timişoara

Faculty of Mathematics and Informatics

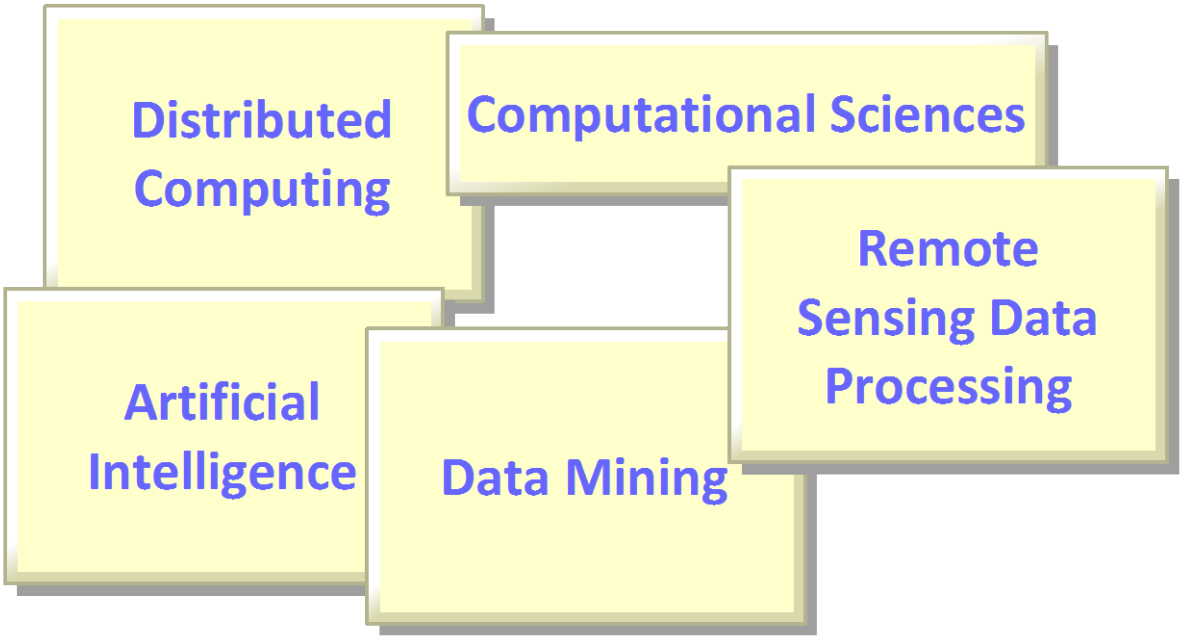
... in figures

- 1500 students
- 55 teachers
- 2 research centers (Mathematics, Informatics)
- 1 spin-off for research - Institute e-Austria (www.ieat.ro)

Research Center in Informatics (http:research.info.uvt.ro):

- Distributed and Parallel Computing
- Artificial Intelligence
- Theory of Computing
- Computational Mathematics

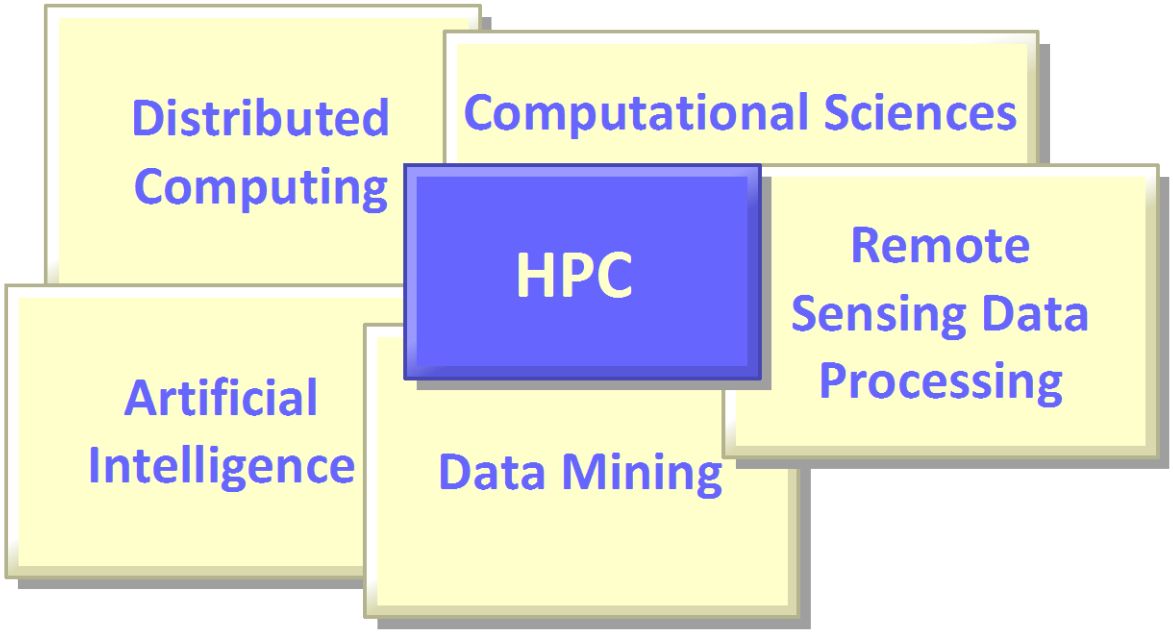
Research Topics Overview



Different topics but similar challenges:

- Large sets of data or large search spaces
- Computationally intensive tasks

Research Topics Overview



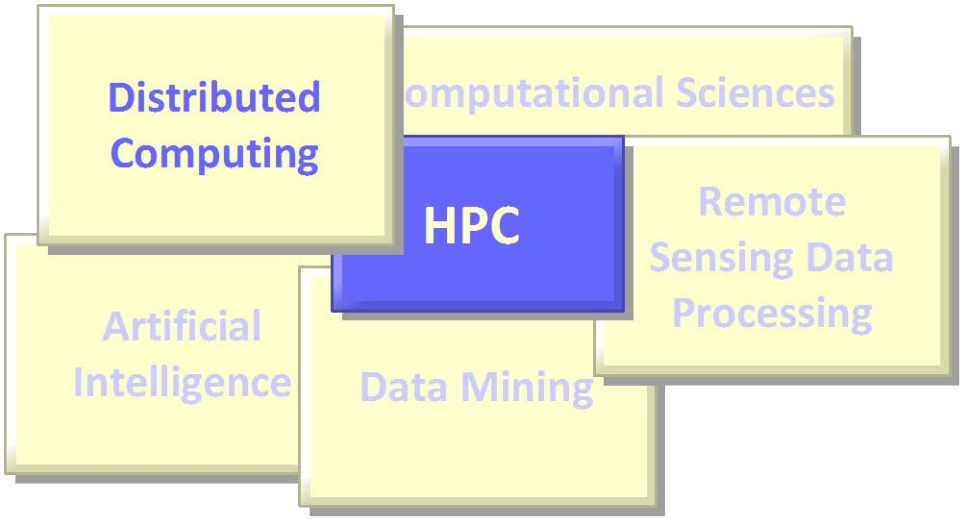
Different topics but similar challenges:

- Large sets of data or large search spaces
- Computationally intensive tasks

… which require high performance solutions

http://hpc.uvt.ro

Distributed Computing



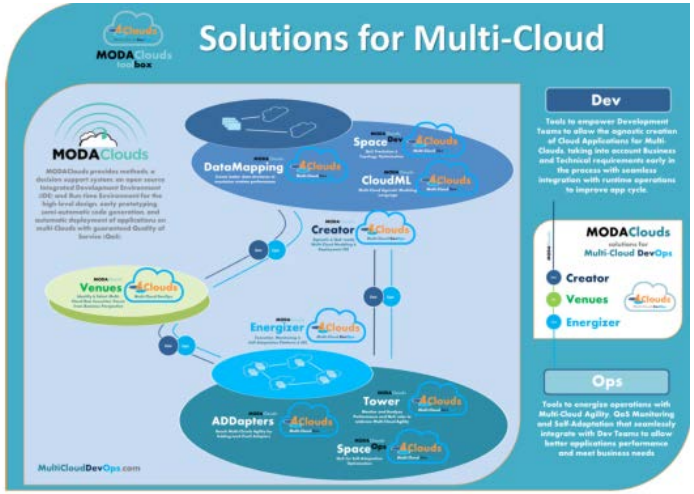
Current research topics:

- Tools for deployment of applications in multiple clouds
- Monitoring tools for data-inensive cloud applications
- Scheduling algorithms for resource provisioning

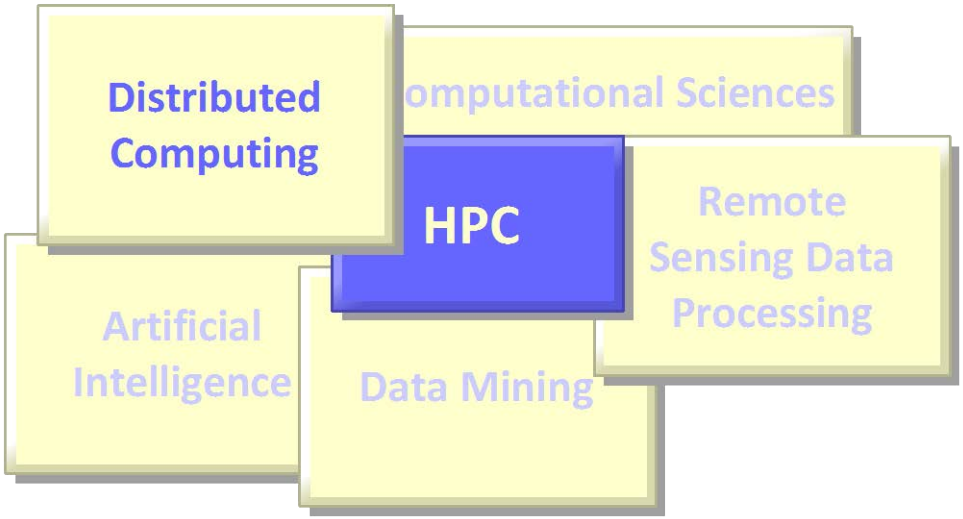
Related projects:

- MODAClouds (www.moda-clouds.eu, FP7 2012-2015)

Results: Multi-cloud toolbox for developers and operators of applications running on multi-clouds



Distributed Computing



Current research topics:

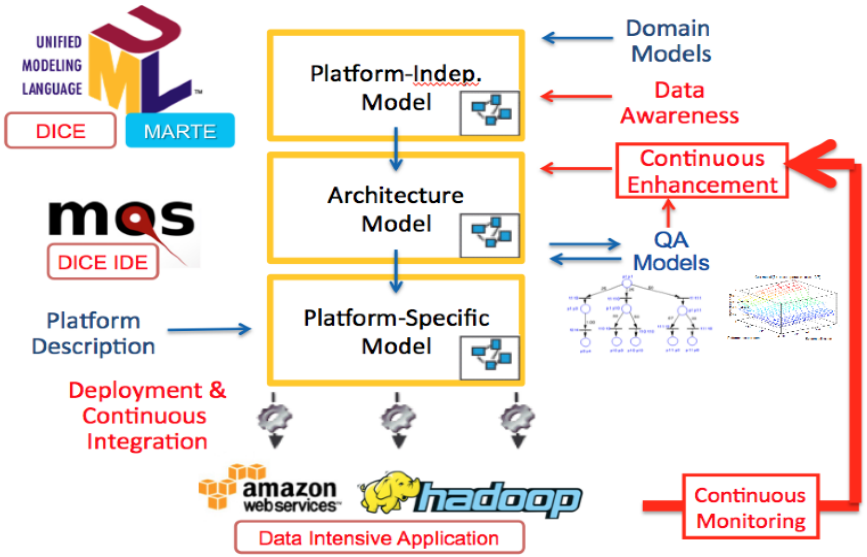
- Tools for deployment of applications in multiple clouds
- Tools for developing data-intensive cloud applications
- Scheduling algorithms for resource provisioning

Related projects:

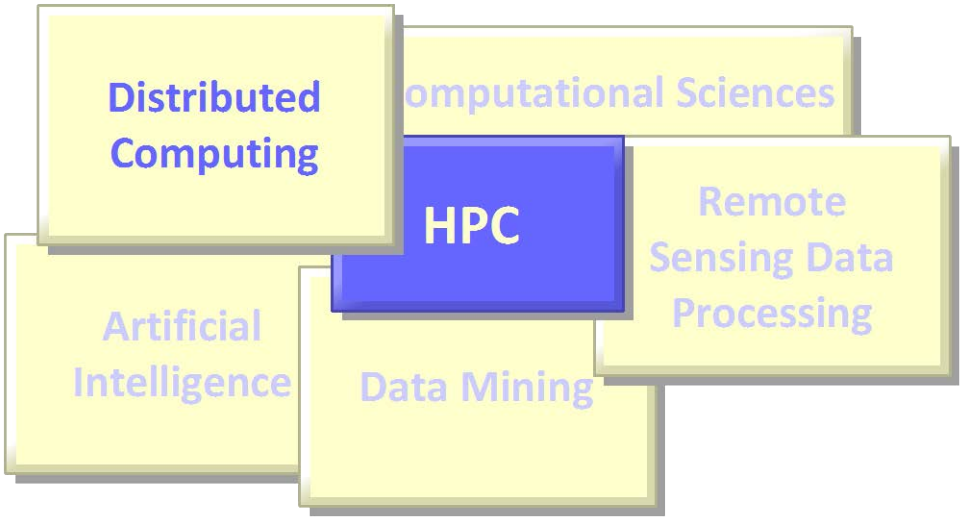
- DICE (www.dice-H2020.eu, 2015-2018)

Aim: develop tool chain containing

- IDE for data-intensive cloud applications
- Tools for quality analysis: monitoring + anomaly detection (using machine learning)



Distributed Computing



Current research topics:

- Tools for deployment of applications in multiple clouds
- Tools for developing data-intensive cloud applications
- Algorithms for scheduling and resource provisioning

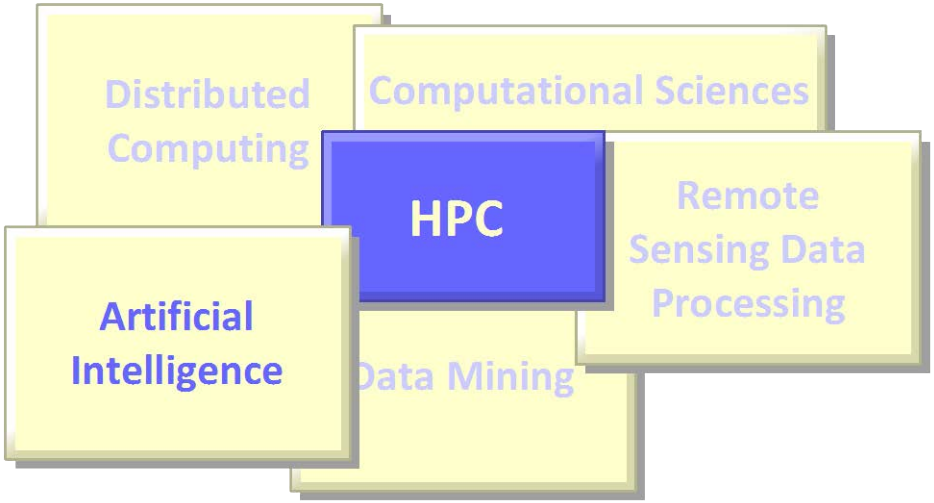
Related projects:

- AMICAS - Automated Management in Cloud and Sky Computing Environments (RO 2012-2016)

Challenges in solving scheduling problems:

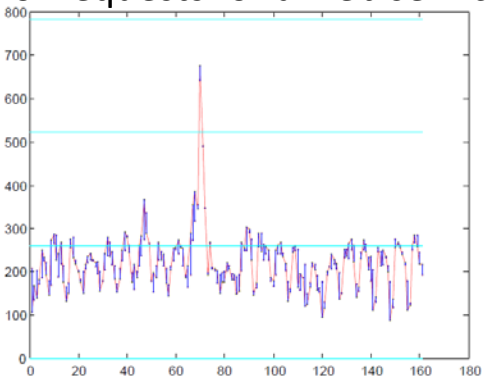
- Large search space
- Complex optimization problem (multiple objectives, constraints, dynamic)

Artificial Intelligence

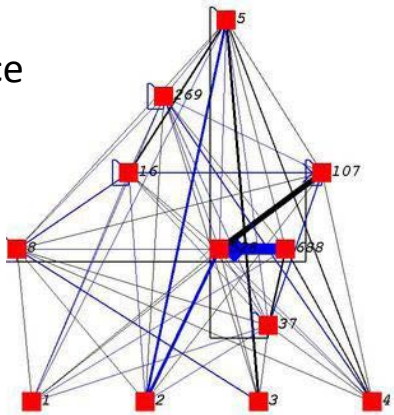


Current research topics:

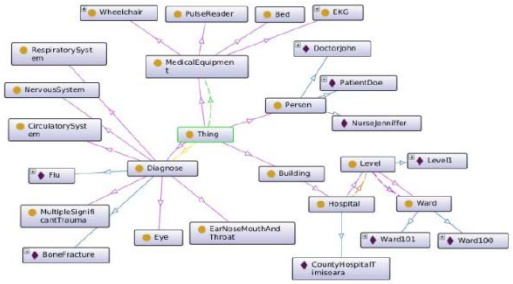
- Machine learning techniques in:
  - Prediction for auto-scaling of resources in distributed systems
  - Analysis of financial, meteorological, medical data
- Multi-agent approaches in:
  - Stock trading systems
  - Frameworks for strategy games
- Ontologies for
  - Context modelling in IoT
  - Semantic services



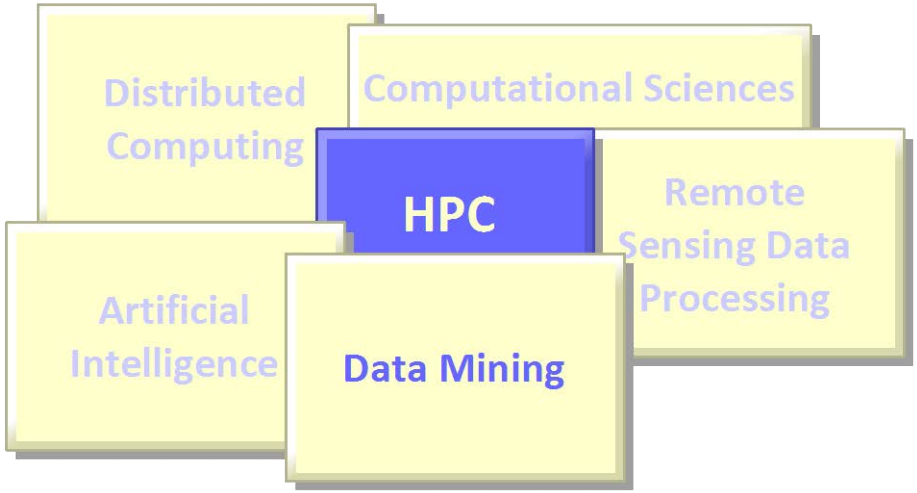
Predicting the number of requests for a web service



Neural networks for game strategy learning



Data Mining



Research topics:

- Distributed architectures for data mining
- Unsupervised classification of distributed data
- Extracting classification/prediction rules from data
- Taxonomy/ ontology based similarity measures for medical data
- Anomaly detection in data

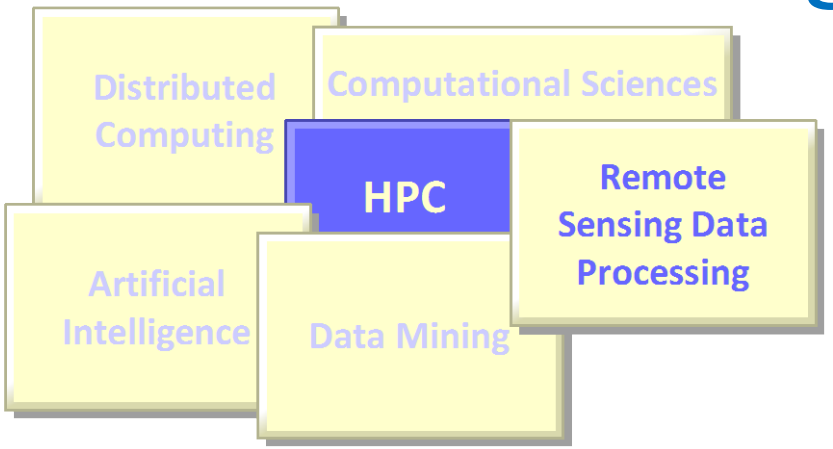
Data mining and machine learning as a Service:

- Access through lightweight web services (REST)
- Access to algorithms' parameters
- User experience enhancements
- Semantic modeling

Challenges in data mining:

- Large sets of data
- Response in real-time (prediction models)

Remote Sensing Data Processing

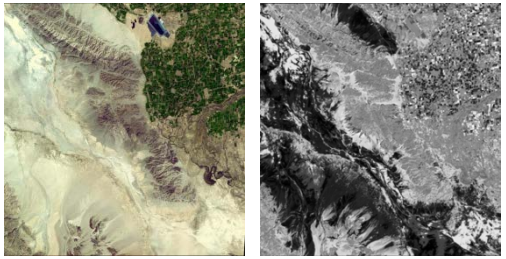


Challenges:

- Large images (many pixels, many spectral bands)
- Computational intensive image analysis algorithms

Research topics:

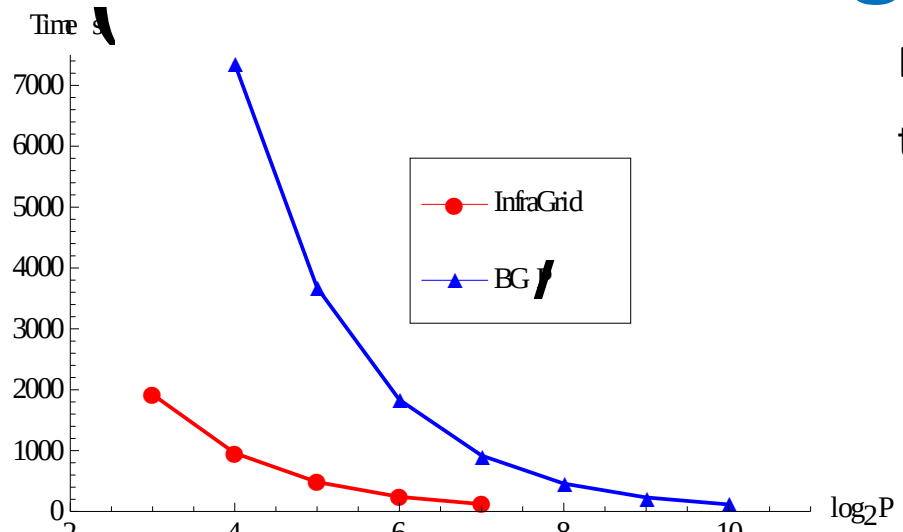
- Processing multi/hyper spectral images:
  - Identify homogeneous regions
  - Identify reference substances



Results:

- Parallel/ efficient implementations of
  - spatial variants of fuzzy clustering
  - algorithms for spectral mixture analysis (end-members extraction and abundances estimation)

Remote Sensing Data Processing



Efficiency of parallel fuzzy clustering tested on:

- InfraGrid cluster (400 cores)
- BG/P (1024 CPUs)

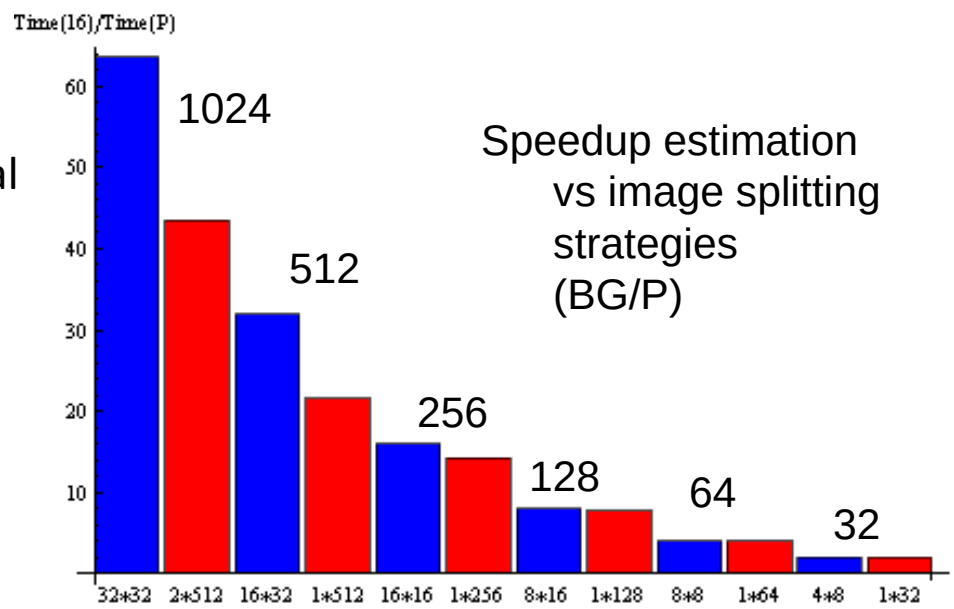
Algorithm: Spatial Fuzzy Cmeans

Implementation: C, MPI (MPICH-2)

Test images:

- LANDSAT (7856 x 8786 pixels, 4 spectral bands)
- AVIRIS (1087x614 pixels, 224 spectral bands)

[D.Petcu, D. Zaharie, S. Panica, A. Hussein, A. Sayed, H. El-Shishiny; Fuzzy Clustering of Large Satellite Images using HPC, Proc. SPIE 2011]



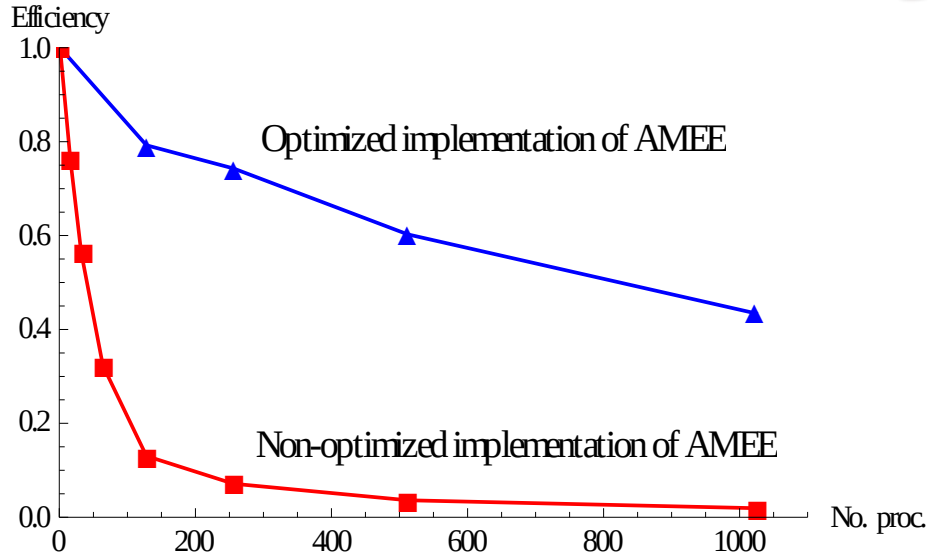
Speedup estimation vs image splitting strategies (BG/P)

Remote Sensing Data Processing

- Efficient implementation of endmembers extraction algorithms
  - AMEE (Automated Morphological Endmembers Extraction)

AMEE on BG/P

- Split the image + extract local information (local endmembers)
- exploit the structure of spectral angle metric to optimize the paired distances between local endmembers
- avoid a global computation by a particular procedure to merge local sets of endmembers
- control the synchronization among processes in the context of using collective communications (MPIBarrier)



Test image: AVIRIS Cuprite (224 bands, 614x2206 pixels)

[A. Toma, S. Panica, D. Zaharie, D. Petcu, Challenges in Processing Hyperspectral Images, Proc. RO-LCG, 2012]

Remote Sensing Data Processing

- Efficient implementation of endmembers extraction algorithms
  - MVSA (Minimum Volume Simplex Analysis)

MVSA on GPU

- Sequential implementation (Matlab): 3h
- Parallel implementation:
  - IBM blade with two Intel Xeon quad-core processors and one FermiTesla M2075 GPU, C+CUDA):
  - Time: 2 minutes

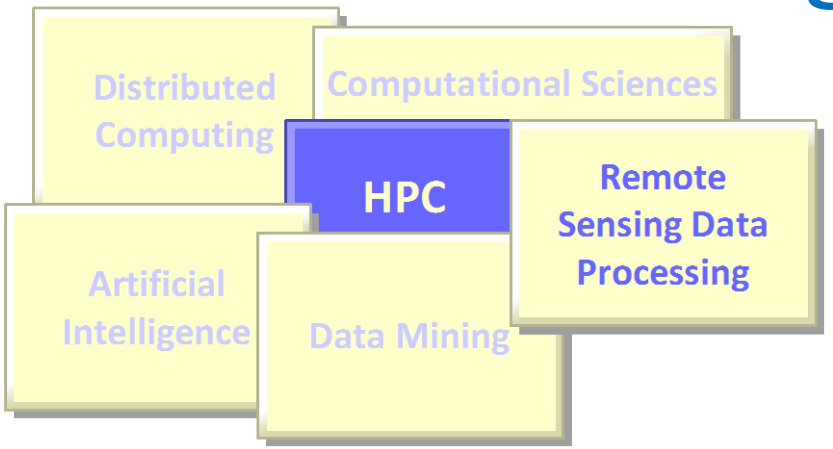
[A. Agathos, J.Li, D. Petcu, A. Plaza; MVSA: Multi-GPU Implementation of MVSA Algorithm for Hyperspectral Unmixing, IEEE JSTARS, 2014]

Fast MVSA

- Sequence of „approximate" constrained quadratic optimization problems solved by interior point method
- Time (for AVIRIS Cuprite data set, 250x190 pixels subscene):
  - Fast MVSA: 3 min
  - Other methods: 7 h (MVES) or 50 min (MVC-NMF)

[J.Li, A. Agathos, D. Zaharie, J. Bioucas, A. Plaza, X. Li; MVSA: A Fast Algorihtm for Linear Hyperspectral Unmixing, IEEE Trans. Geosci. Rem. Sensing - 2015]

Remote Sensing Data Processing

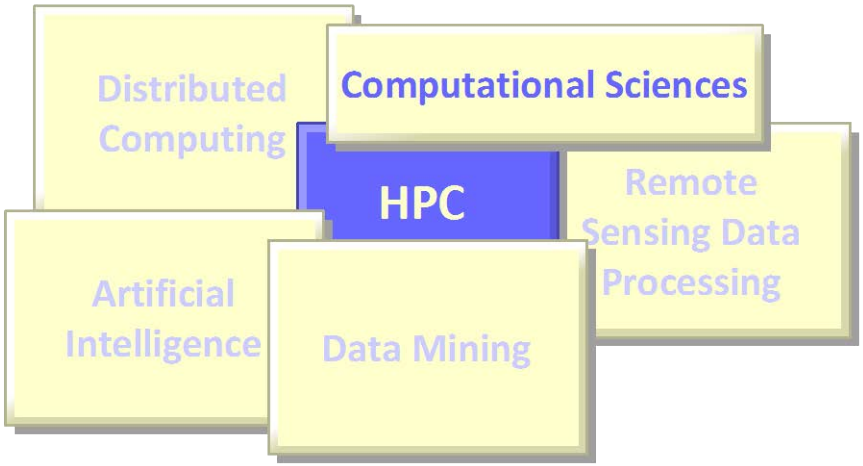


Collaborations

- IBM CAS Cairo, Egypt
- Ain Shams University, Cairo, Egypt
- University of Extremadura, Spain

- **Related projects:**
  - IBM OCR – High-Performance Satellite Multi/Hyperspectral Image Processing (2010-2011)
  - GiSHEO - On demand Grid services for high education and training in Earth observation (ESA-PECS, 2008-2010)
  - HPC-SEE - High-Performance Computing Infrastructure for South East Europe's Research Communities (FP7-Infrastructures, 2010-2013)
  - HOST – HPC Service Center (FP7-REGPOT, 2012-2014)

Computational Sciences



Related projects:

- Analysis of some mathematical physics problems occurring in the sound attenuation in an acoustically lined duct carrying gas flow (RO PN-II-ID-PCE, 2011-2013)
- SIMTIM -Modeling and simulation of the dynamics of thymocyte populations and cells of the thymus medulla under normal and pathological situations (RO PN-II-ID-PCE, 2012-2016)

Computational Mathematics:

- Efficient methods for solving large systems of equations
- Preconditioning through bandwidth reduction

Computational Physics:

- Efficient computational methods in nanoscale optics
- Simulation of crystallization processes, transport phenomena, acoustic lining

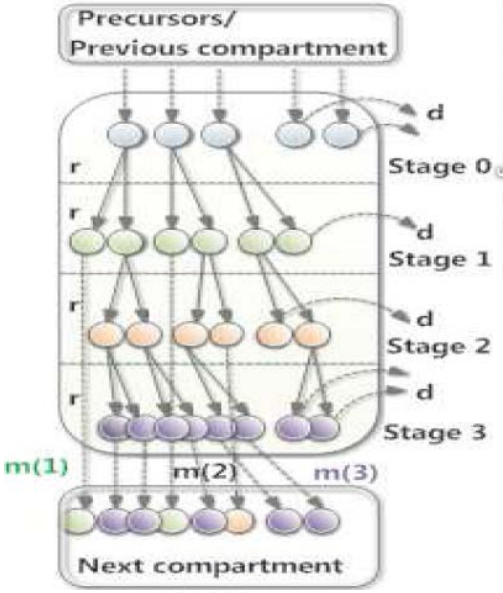
Computational Biology :

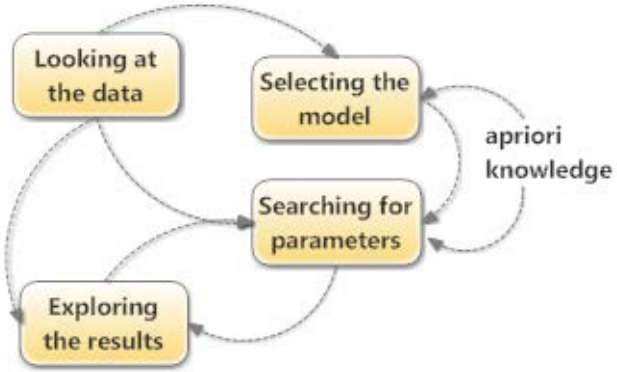
- Simulating the dynamics of thymus cells populations based on experimental data

Computational Sciences

Computational Biology :

- Multi-compartmental models :
  - Systems of (non)linear differential equations (from 4 to more than 20)
  - Various models for biological processes (proliferation, transfer, death)
  - Many parameters to estimate (from 30 to 55)
- Challenges for parameter estimation
  - no explicit relationship between objective function and parameters („semi-transparent" model)
  - parameter (non)identifiability
  - hard to check constraints
- Approach:
  - use of population-based metaheuristics





[D. Zaharie, L. Moleriu, V. Negru, Evolutionary Parameter Estimation in Multi-stage Compartmental Models of Thymocyte Dynamics, Proc. GECCO 2013]

[D. Moleriu, D. Zaharie et al, Insights into the mechanisms of thymus involution ... , J. Theor. Biol., 2014]