Advances in compact Differential Evolution

Postgraduate Seminar

Giovanni Iacca

University of Jyväskylä, Faculty of Information Technology
Department of Mathematical Information Technology

December 9th 2010

My background

- **2006: Master of Science in Software Engineering & Automation at Technical University of Bari, Italy**
- **2006 - 2009: Researcher and Software Developer for a spinoff of CNR (Italian National Research Council)**
  - *Real Time Control Systems*
  - *Robotics and Machines Tools*
  - *Intelligent Systems*
  - *Rapid prototyping & Automatic Code Generation with Matlab/Simulink*
  - *Field Buses, Servo drives, Sensors & Actuators*
  - *SW Engineering: Linux drivers, Human Machine Interfaces (HMI), C/C++ and Java distributed real time applications*
- **2010 - nowadays: PhD student at University of Jyväskylä, Faculty of IT, Department of MIT**

My PhD (1/2)

"Usability and Commercialization of Advanced Computational Intelligence Optimization"

(collection of papers)

- **Two research fields**
  - *Computational Intelligence Optimization*
  - *Software Engineering*

- **Supervisors**
  - *Ferrante Neri, Adj. Prof - Ernesto Mininno, PhD*
  - *Tuomo Rossi, Prof. - Raino Mäkinen, Prof.*

- **International collaborations**
  - *Ponnuthurai Nagaratnam Suganthan, Ass. Prof.*
  - *Rammohan Mallipeddi, PhD*
    (Nanyang Technological University, Singapore)

My PhD (2/2)

"Usability and Commercialization of Advanced Computational Intelligence Optimization"

(collection of papers)

- **Status of research activities**
  - *1 journal paper to appear*
  - *2 journal papers submitted (under review)*
  - *2 conference papers submitted to* ***IEEE Symposium Series on Computational Intelligence - SSCI 2011, Paris*** *(under review)*
  - *1 conference paper submitted to* ***evo* 2011, Turin*** *(under review)*
  - *1 more journal paper to be submitted (hopefully) by the end of 2010*
- **Status of study plan**

  45/60 credits
- **Expected finishing time**

  end of 2011 – beginning of 2012

Introduction

- What is Computational Intelligence (CI)?
- A popular CI Algorithm: Differential Evolution (DE)
- A brief survey of DE-based Algorithms
- compact Differential Evolution (cDE)
- cDE-based algorithms proposed during my first works
- Case study: space robotic arm
- Conclusions and future works

Computational Intelligence

Computational Intelligence (Optimization)

When the problem cannot be solved by means of an exact method due to the lack of differentiability or even analytic expression an alternative way must be found

Methodologies

- **Memetic Computing**

  encoding of culture into optimization algorithms, e.g. hybrid approaches, integration of knowledge

- **Differential Evolution**

  specific Optimization Algorithm for continuous problems

Computational Intelligence

- Global optimization is necessary in fields such as engineering, statistics and finance
- But many practical problems have objective functions that are non differentiable, non-continuous, non-linear, noisy, multi-dimensional
- Such problems are difficult if not impossible to solve analytically
- Computational Intelligence Optimization Algorithms can be used to find approximate solutions to such problems
- Evolutionary Optimization in the Presence of Uncertainties
- Large Scale and Computationally Expensive Optimization Problems

Differential Evolution (DE)

- ***Differential Evolution*** (Storn and Price in 1995) is a stochastic population based evolutionary algorithm fairly fast and reasonably robust
- A population of potential solutions, within an n-dimensional search space, is randomly initialized, then evolves over time to explore the search space and to locate the minima of the objective function
- At each iteration new vectors are generated by the combination of vectors randomly chosen from the current population (***mutation***)
- The new vectors are then mixed (***recombination***, or ***crossover***) with a predetermined target vector to get a trial vector
- Finally, the trial vector is accepted (***selection***) for the next generation if and only if it yields a reduction in the value of the objective function

Differential Evolution *rand/1/bin*

generate S_{pop} individuals of the initial population pseudo-randomly;
while budget condition
  for $i = 1 : S_{pop}$
    compute $f(x_i)$;
  end-for
  for $i = 1 : S_{pop}$
    mutation
    select three individuals x_r, x_s, and x_t;
    compute $x'_{off} = x_t + F(x_r - x_s)$;
    crossover
    $x_{off} = x'_{off}$;
    for $j = 1 : n$
      generate $rand(0, 1)$;
      if $rand(0, 1) < CR$
        $x_{off,j} = x_{i,j}$;
      end-if
    end-for
    selection
    if $f(x_{off}) \leq f(x_i)$
      save index for replacement $x_i = x_{off}$;
    end-if
  end-for
  perform replacements;
end-while

mutation

crossover

selection

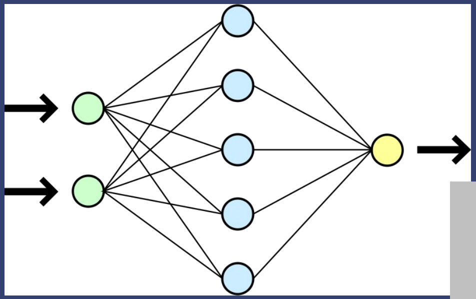
Other mutation rules:
DE/best/1
DE/cur-to-best/1
DE/best/2
DE/rand/2
DE/rand-to-best/2
...

Other crossover rules:
Exponential
SPX
BLX-α
...

Only three parameters
F (scale factor)
CR (crossover ratio)
S_{pop} (population size)

Applications of DE

- Robotics
- Multiprocessors synthesis
- Neural networks learning
- Calibration of financial models
- Optimal portfolio selection
- Crystallographic characterization
- Synthesis of modulators
- Optimal design of heat exchangers
- Non-linear chemical processes
- Planning of cropping patterns
- Water pumping systems
- Design of gas transmission network
- Physiochemistry of Carbon materials
- Radio network design
- … and many more!



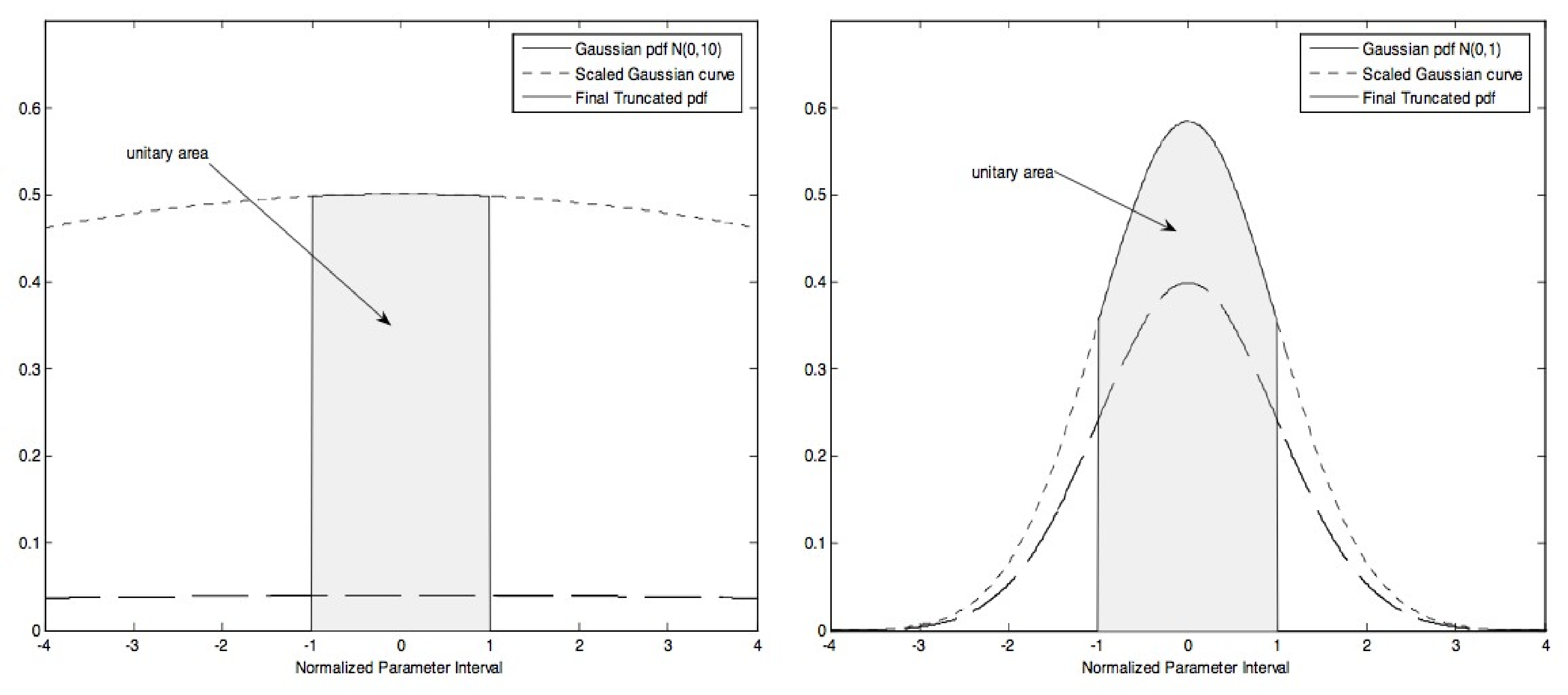
Differential Evolution Variants

- **Population based Differential Evolution**
  - ***Additional components to standard DE framework***
    - *DE with Trigonometric Mutation (TDE)*
    - *DE with Adaptive Crossover Local Search (DEahcSPX)*
    - *DE with Population Size Reduction (DEPSR)*
    - *DE with Scale Factor Local Search (DESFLS)*
  - ***Modified structures of Differential Evolution***
    - *Self-Adapting Parameter Setting in Differential Evolution (jDE)*
    - *Opposition Based Differential Evolution (OBDE)*
    - *DE with Global and Local Neighborhoods (DEGL)*
    - *Self Adaptive Differential Evolution (SADE)*

- **compact Differential Evolution (cDE)**
  - ***Part of my PhD research: investigating different novel structures of cDE-based algorithms***

compact DE (cDE)

- belongs to the class of *Estimation Distribution Algorithms* (EDA)
- does not use a population of individuals
- makes use of a statistic representation of the population
- this approach is necessary to solve complex optimization problems despite the **absence of a full performance computer (memory issues)**



(a) beginning of the optimization

(b) late stage of the optimization

compact DE (cDE)

Probability Vector (PV)

$$PV_m^t = [\mu^t, \sigma^t]$$

$$PDF(\mu[i], \sigma[i]) = \frac{e^{-\frac{(x-\mu[i])^2}{2\sigma[i]^2}}\sqrt{\frac{2}{\pi}}}{\sigma[i]\left(\mathrm{erf}\left(\frac{\mu[i]+1}{\sqrt{2}\sigma[i]}\right)-\mathrm{erf}\left(\frac{\mu[i]-1}{\sqrt{2}\sigma[i]}\right)\right)}$$

- Survivor selection scheme (one-to-one spawning logic)
- DE can be straightforwardly encoded into a compact algorithm without losing the basic working principles
- Sampling introduces beneficial extra randomness
- Convergence: shrinkage of (truncated) Gaussian bell-shaped curve over the (global) best

counter $t = 0$

initialize PV

for $i = 1 : n$ **do**
  {** *PV* initialization **}
  initialize $\mu[i] = 0$
  initialize $\sigma[i] = \lambda$
end for
generate *elite* by means of *PV*
while budget condition **do**
  {** Mutation **}

sampling from PV

  generate 3 individuals x_r, x_s, and x_t by means of *PV*

mutation

  compute $x'_{off} = x_t + F(x_r - x_s)$
  {** Crossover **}

crossover

  $x'_{off} = x_{off}$
  for $i = 1 : n$ **do**
    generate $rand(0, 1)$
    if $rand(0, 1) < Cr$ **then**
      $x_{off}[i] = elite[i]$
    end if
  end for
  {** Elite Selection **}

PV update

  $[winner, loser] = compete(a, elite)$
  if $a == winner$ **then**
    $elite = a$
  end if
  {** *PV* Update **}
  $\mu^{t+1} = \mu^t + \frac{1}{N_p}(winner - loser)$
  $\sigma^{t+1} = \sqrt{(\sigma^t)^2 + (\mu^t)^2 - (\mu^{t+1})^2 + \frac{1}{N_p}(winner^2 - loser^2)}$
  $t = t + 1$
end while

cDE-based algorithms proposed

- **An unconventional memetic approach:**
  - ***Disturbed Exploitation cDE (DecDE)***
- **Combining distributed compact units:**
  - ***Composed cDE (CcDE)***
  - ***Supervised cDE (ScDE)***
- **Using domain knowledge to support optimization:**
  - ***Super-Fit and Population Size Reduction cDE (SfcDE-PSR)***
  - ***Compact Opposition DE (cODE)***
- **Noisy optimization:**
  - ***Noise Analysis cDE (NAcDE)***

Disturbed Exploitation cDE (DEcDE)

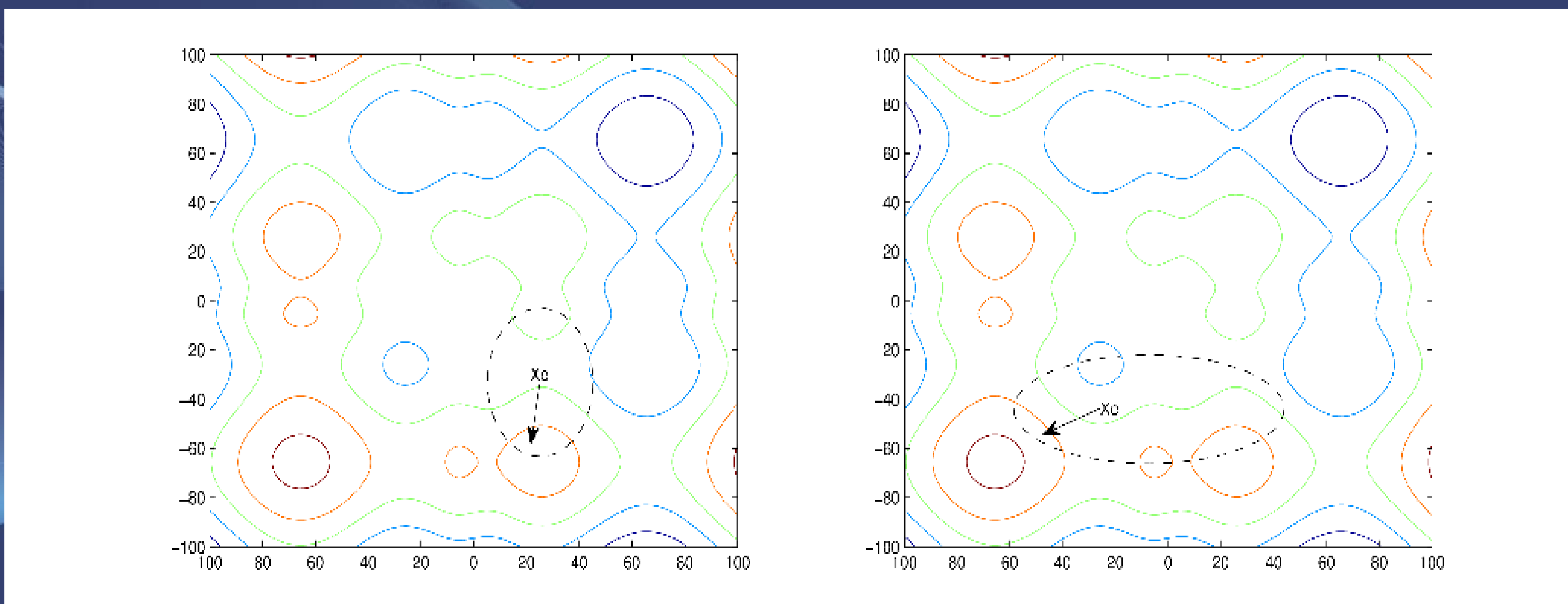
- Rand/1/Exp or Trigonometric Mutation

$$x_{off} = \frac{(x_r + x_s + x_t)}{3} + (p_s - p_r)(x_r - x_s) + (p_t - p_s)(x_s - x_t) + (p_r - p_t)(x_t - x_r)$$

$$p_k = \frac{|f(x_k)|}{|f(x_r)| + |f(x_s)| + |f(x_t)|}$$

- PV perturbation (~ *replacement of part of the population in a DE*)

$$\mu^{t+1} = \mu^{t+1} + 2\tau \cdot rand(0,1) - \tau \qquad \left(\sigma^{t+1}\right)^2 = \left(\sigma^{t+1}\right)^2 + \tau \cdot rand(0,1)$$



DEcDE – Results (1/2)

Table 1: Average final fitness values ± standard deviations for compact algorithms

Test Problem	cGA	rcGA	cDE	McDE	DEcDE
f_1	1.446e+04 ± 4.63e+03	1.906e+04 ± 9.62e+03	**4.520e-28 ± 1.74e-27**	6.526e-25 ± 8.34e-25	1.681e+01 ± 1.01e+01
f_2	1.628e+06 ± 7.00e+05	2.677e+04 ± 4.78e+03	9.865e+03 ± 2.52e+03	2.322e+03 ± 9.48e+02	**1.799e+03 ± 1.79e+03**
f_3	2.432e+09 ± 1.62e+09	1.803e+09 ± 2.02e+09	9.898e+01 ± 1.41e+02	1.479e+04 ± 6.89e+04	**8.473e+03 ± 6.51e+03**
f_4	1.681e+01 ± 9.45e-01	1.859e+01 ± 4.15e-01	1.074e+01 ± 1.75e+00	1.887e+00 ± 1.70e+00	**1.673e+00 ± 4.74e-01**
f_5	1.721e+01 ± 1.41e+00	1.880e+01 ± 4.54e-01	1.028e+01 ± 1.83e+00	3.505e+00 ± 1.29e+00	**1.540e+00 ± 4.62e-01**
f_6	8.840e+02 ± 3.08e+01	**2.259e-03 ± 4.11e-03**	1.883e-01 ± 2.03e-01	7.533e-03 ± 1.89e-02	6.081e-01 ± 2.01e-01
f_7	8.778e+02 ± 3.34e+01	**3.403e-02 ± 9.71e-02**	1.891e-01 ± 2.06e-01	2.497e-01 ± 2.12e-01	7.258e-01 ± 1.30e-01
f_8	2.265e+02 ± 4.06e+01	2.037e+02 ± 2.74e+01	5.959e+01 ± 1.33e+01	6.586e+01 ± 1.36e+01	**1.651e+01 ± 3.31e+00**
f_9	3.013e+02 ± 4.72e+01	1.985e+02 ± 3.06e+01	1.219e+02 ± 2.58e+01	**1.210e+02 ± 2.61e+01**	2.005e+02 ± 2.09e+01
f_{10}	1.307e+05 ± 4.96e+04	2.900e+03 ± 3.07e+03	6.448e+03 ± 2.75e+03	6.984e+03 ± 3.39e+03	**1.648e+03 ± 3.71e+02**
f_{11}	4.947e+03 ± 7.03e+02	3.156e+03 ± 7.54e+02	9.972e+02 ± 3.25e+02	1.090e+03 ± 3.02e+02	**1.626e+02 ± 4.32e+01**
f_{12}	5.614e+00 ± 2.91e+00	4.127e+00 ± 4.90e+00	2.558e-02 ± 7.10e-03	8.208e-01 ± 1.74e-01	**4.402e-01 ± 1.19e-01**
f_{13}	3.309e+01 ± 1.13e+01	**-1.000e+02 ± 5.06e-09**	**-1.000e+02 ± 1.73e-06**	**-1.000e+02 ± 1.21e-05**	-9.962e+01 ± 1.45e-01
f_{14}	4.472e+04 ± 1.37e+05	1.401e+00 ± 1.91e+00	**1.982e-04 ± 1.74e-04**	1.468e-02 ± 2.57e-02	3.921e-02 ± 2.10e-02
f_{15}	1.121e+06 ± 3.66e+06	-7.869e-01 ± 8.94e-01	**-1.148e+00 ± 1.67e-03**	-1.047e+00 ± 7.35e-02	-9.811e-01 ± 1.25e-01
f_{16}	1.172e+04 ± 2.57e+03	8.975e+03 ± 2.38e+03	8.023e+03 ± 3.42e+03	7.600e+03 ± 2.53e+03	**3.044e+03 ± 1.91e+03**
f_{17}	1.307e+02 ± 2.03e+00	1.226e+02 ± 3.21e+00	1.242e+02 ± 3.21e+00	**1.223e+02 ± 4.11e+00**	1.302e+02 ± 1.46e+00
f_{18}	2.958e+05 ± 1.05e+05	3.089e+05 ± 1.38e+05	5.480e+04 ± 3.21e+04	**3.661e+03 ± 6.25e+03**	8.517e+04 ± 1.40e+04
f_{19}	5.296e-02 ± 9.80e-09	**5.296e-02 ± 5.22e-18**	**5.296e-02 ± 3.28e-11**	**5.296e-02 ± 4.63e-10**	**5.296e-02 ± 9.40e-09**
f_{20}	-9.632e-01 ± 4.22e-02	**-1.067e+00 ± 4.09e-16**	**-1.067e+00 ± 1.50e-05**	**-1.067e+00 ± 2.82e-05**	**-1.067e+00 ± 1.13e-05**
f_{21}	2.337e+01 ± 1.14e+00	**3.979e-01 ± 9.70e-13**	**3.979e-01 ± 1.71e-05**	**3.979e-01 ± 1.45e-05**	**3.979e-01 ± 1.44e-05**
f_{22}	-3.760e+00 ± 1.95e-02	**-3.863e+00 ± 1.94e-15**	**-3.863e+00 ± 1.21e-06**	**-3.863e+00 ± 5.89e-06**	**-3.863e+00 ± 6.59e-06**
f_{23}	-4.819e-01 ± 4.27e-02	-3.238e+00 ± 5.53e-02	**-3.288e+00 ± 5.54e-02**	**-3.288e+00 ± 5.53e-02**	-3.283e+00 ± 4.61e-02
f_{24}	-2.773e+00 ± 1.13e+00	-6.458e+00 ± 2.47e+00	-5.451e+00 ± 3.24e+00	-4.927e+00 ± 2.94e+00	**-9.955e+00 ± 1.66e+00**
f_{25}	-2.628e+00 ± 9.63e-01	-7.258e+00 ± 3.01e+00	-5.504e+00 ± 3.33e+00	-5.738e+00 ± 3.18e+00	**-9.891e+00 ± 1.96e+00**
f_{26}	-2.764e+00 ± 9.00e-01	-6.940e+00 ± 3.19e+00	-6.239e+00 ± 3.75e+00	-4.912e+00 ± 3.07e+00	**-9.916e+00 ± 1.53e+00**

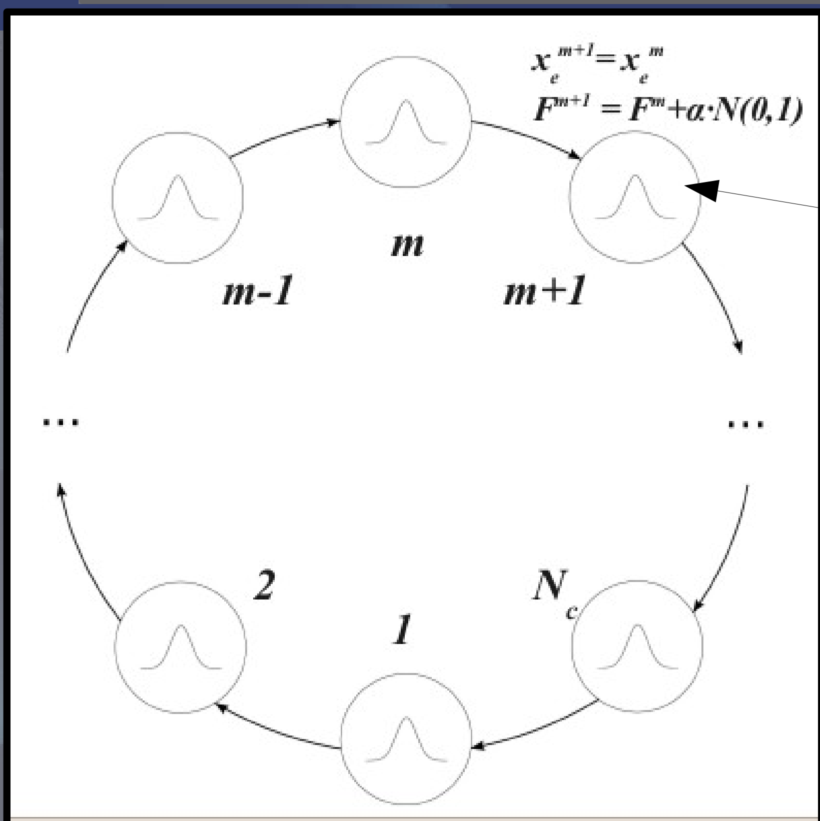
DEcDE – Results (2/2)

Table 3: Average final fitness values ± standard deviations for population-based algorithms

Test Problem	EDA_{mvg}	RCMA	DEahcSPX	DEcDE
f_1	6.955e+01 ± 1.47e+02	**3.411e-16 ± 6.85e-16**	9.690e+01 ± 1.50e+01	1.681e+01 ± 1.01e+01
f_2	3.145e+02 ± 3.01e+02	**1.255e-13 ± 2.95e-13**	1.654e-02 ± 2.30e-02	1.799e+03 ± 1.79e+03
f_3	2.319e+06 ± 5.55e+06	**2.827e+01 ± 1.28e-01**	1.110e+05 ± 3.11e+04	8.473e+03 ± 6.51e+03
f_4	3.271e+00 ± **4.71e-01**	**7.012e-09** ± 1.04e-08	2.567e+00 ± 4.64e-01	1.673e+00 ± 4.74e-01
f_5	3.406e+00 ± 1.07e+00	**8.119e-09 ± 8.45e-09**	2.509e+00 ± 4.09e-01	1.540e+00 ± 4.62e-01
f_6	2.638e+02 ± 3.90e+01	6.575e+00 ± 3.85e+00	4.886e+00 ± 3.60e-01	**6.081e-01 ± 2.01e-01**
f_7	2.735e+02 ± 4.50e+01	6.063e+00 ± 3.50e+00	4.824e+00 ± 3.50e-01	**7.258e-01 ± 1.30e-01**
f_8	1.770e+02 ± 1.35e+01	**7.105e-15 ± 2.55e-14**	3.405e+01 ± 3.48e+00	1.651e+01 ± 3.31e+00
f_9	1.816e+02 ± 1.26e+01	**4.737e-15 ± 2.32e-14**	1.172e+02 ± 2.56e+01	2.005e+02 ± 2.09e+01
f_{10}	1.402e+04 ± 5.35e+04	1.003e+04 ±1.78e+04	3.026e+03 ± 3.75e+02	**1.648e+03 ± 3.71e+02**
f_{11}	1.015e+04 ± 4.23e+02	2.989e+03 ± 5.98e+02	8.258e+02 ± 8.20e+01	**1.626e+02 ± 4.32e+01**
f_{12}	1.294e+00 ± 6.90e-01	**6.329e-10 ± 2.63e-09**	1.333e-01 ± 3.16e-02	4.402e-01 ± 1.19e-01
f_{13}	-9.475e+01 ± 2.21e+01	-6.845e+01 ± 1.17e+01	-8.561e+01 ± 1.39e+00	**-9.962e+01 ± 1.45e-01**
f_{14}	5.928e-01 ± 1.34e+00	**1.030e-06 ± 1.29e-06**	1.157e-02 ± 8.74e-03	3.921e-02 ± 2.10e-02
f_{15}	-1.290e-01 ± 1.90e+00	**-1.149e+00 ± 3.70e-03**	-8.535e-01 ± 1.08e-01	-9.811e-01 ± 1.25e-01
f_{16}	1.007e+04 ± 3.36e+03	9.318e+03 ± 2.45e+03	9.314e+03 ± 1.02e+03	**3.044e+03 ± 1.91e+03**
f_{17}	1.303e+02 ± 7.51e-01	**1.256e+02 ± 3.65e+00**	1.298e+02 ± 1.15e+00	1.302e+02 ± 1.46e+00
f_{18}	5.272e+05 ± 2.93e+05	1.298e+05 ± 4.67e+04	**7.203e+04 ± 1.03e+04**	8.517e+04 ± 1.40e+04
f_{19}	5.296e-02 ± 6.61e-09	**5.296e-02 ± 4.34e-18**	5.296e-02 ± 3.64e-09	5.296e-02 ± 9.40e-09
f_{20}	**-1.067e+00 ± 4.54e-16**	-1.067e+00 ± 6.64e-06	-1.067e+00 ± 2.16e-05	-1.067e+00 ± 1.13e-05
f_{21}	3.987e-01 ± 3.42e-03	3.980e-01 ±1.95e-04	3.980e-01 ± 1.97e-04	**3.979e-01 ± 1.44e-05**
f_{22}	-3.847e+00 ± 3.37e-02	-3.863e+00 ± 5.91e-06	-3.863e+00 ± 5.56e-06	**-3.863e+00 ± 6.59e-16**
f_{23}	-3.126e+00 ± 1.65e-01	-3.268e+00 ± 6.07e-02	**-3.322e+00 ± 1.33e-04**	-3.283e+00 ± 4.61e-02
f_{24}	-5.633e+00 ± 3.50e+00	-5.302e+00 ± 3.20e+00	**-1.004e+01 ± 1.19e-01**	-9.955e+00 ± 1.66e+00
f_{25}	-9.252e+00 ± 2.42e+00	-4.648e+00 ± 3.68e+00	**-1.024e+01 ± 1.06e-01**	-9.891e+00 ± 1.96e+00
f_{26}	-8.449e+00 ± 3.01e+00	-5.482e+00 ± 3.33e+00	**-1.038e+01 ± 1.93e-01**	-9.916e+00 ± 1.53e+00

Composed cDE (CcDE)

for each generation **do**
  perform a cDE generation (offspring, comparison, replacement)
  if $rand(0,1) < M_e$ AND $f(x_e^m) < f(x_e^{m+1})$ **then**
    send a copy of the elite individual to the neighbour unit
    replace the elite individual: $x_e^{m+1} = x_e^m$
    apply the scale factor inheritance mechanism
    replace the scale factor: $F^{m+1} = F^m + \alpha \mathcal{N}(0,1)$
  end if
end for



- Each **cDE unit** performs one offspring generation and possible elite replacement
- Migration of scale factor (perturbed) and elites among neighbour compact units, according to a ring topology
- Global knowledge & local knowledge of the fitness landscape
- Self-adaption of the ring

CcDE – Results (1/2)

Table 2: Average final fitness values ± standard deviations for compact algorithms

Test Problem	cGA	rcGA	cDE	McDE	(1 + 1)-CMA-ES	CcDE
f_1	1.446e+04 ± 4.63e+03	1.906e+04 ± 9.62e+03	**4.520e-28 ± 1.74e-27**	3.235e-22 ± 3.52e-22	1.961e-27 ± 1.47e-27	2.380e-02 ± 3.47e-02
f_2	1.628e+06 ± 7.00e+05	2.677e+04 ± 4.78e+03	9.865e+03 ± 2.52e+03	3.896e+03 ± 1.66e+03	**6.430e-26 ± 7.81e-26**	2.062e+02 ± 2.74e+02
f_3	2.432e+09 ± 1.62e+09	1.803e+09 ± 2.02e+09	9.898e+01 ± 1.41e+02	1.795e+02 ± 2.17e+02	**1.017e+00 ± 1.80e+00**	9.967e+01 ± 7.48e+01
f_4	1.681e+01 ± 9.45e-01	1.859e+01 ± 4.15e-01	1.074e+01 ± 1.75e+00	7.761e-02 ± 2.63e-01	1.946e+01 ± 1.77e-01	**1.500e-02 ± 1.67e-02**
f_5	1.721e+01 ± 1.41e+00	1.880e+01 ± 4.54e-01	1.028e+01 ± 1.83e+00	**1.084e+00 ± 9.46e-01**	1.942e+01 ± 1.99e-01	2.586e+00 ± 1.07e+00
f_6	8.840e+02 ± 3.08e+01	**2.259e-03 ± 4.11e-03**	1.883e-01 ± 2.03e-01	2.630e-02 ± 6.65e-02	9.842e-03 ± 1.05e-02	2.188e-02 ± 2.46e-02
f_7	8.778e+02 ± 3.34e+01	**3.403e-02 ± 9.71e-02**	1.891e-01 ± 2.06e-01	2.050e-01 ± 2.39e-01	2.843e-01 ± 2.34e-01	1.292e-01 ± 1.85e-01
f_8	2.265e+02 ± 4.06e+01	2.037e+02 ± 2.74e+01	5.959e+01 ± 1.33e+01	1.390e+02 ± 2.22e+01	1.912e+02 ± 3.13e+01	**1.433e-02 ± 2.83e-02**
f_9	3.013e+02 ± 4.72e+01	1.985e+02 ± 3.06e+01	1.219e+02 ± 2.58e+01	1.805e+02 ± 3.06e+01	1.892e+02 ± 4.33e+01	**7.212e+01 ± 5.20e+01**
f_{10}	1.307e+05 ± 4.96e+04	2.900e+03 ± 3.07e+03	6.448e+03 ± 2.75e+03	9.842e+03 ± 4.21e+03	3.980e+02 ± 1.14e+02	**8.339e+00 ± 1.04e+01**
f_{11}	4.947e+03 ± 7.03e+02	3.156e+03 ± 7.54e+02	9.972e+02 ± 3.25e+02	1.138e+03 ± 5.10e+02	5.815e+03 ± 8.42e+02	**5.546e-02 ± 9.79e-02**
f_{12}	5.614e+00 ± 2.91e+00	4.127e+00 ± 4.90e+00	2.558e-02 ± 7.10e-03	8.208e-01 ± 1.74e-01	**6.646e-04 ± 1.81e-03**	1.694e-02 ± 1.82e-02
f_{13}	3.309e+01 ± 1.13e+01	**-1.000e+02 ± 5.06e-09**	-1.000e+02 ± 1.73e-06	-1.000e+02 ± 4.92e-04	-1.000e+02 ± 1.20e-03	-9.999e+01 ± 2.93e-02
f_{14}	4.472e+04 ± 1.37e+05	1.401e+00 ± 1.91e+00	1.982e-04 ± 1.74e-04	3.132e-01 ± 2.15e-01	4.295e+00 ± 5.06e+00	**1.277e-04 ± 2.39e-04**
f_{15}	1.121e+06 ± 3.66e+06	-7.869e-01 ± 8.94e-01	**-1.148e+00 ± 1.67e-03**	-2.771e-01 ± 2.63e-01	1.534e+00 ± 4.07e+00	-1.141e+00 ± 1.78e-02
f_{16}	1.172e+04 ± 2.57e+03	8.975e+03 ± 2.38e+03	8.023e+03 ± 3.42e+03	6.918e+03 ± 2.97e+03	**4.066e+03 ± 1.14e+03**	5.638e+03 ± 2.28e+03
f_{17}	1.307e+02 ± 2.03e+00	1.226e+02 ± 3.21e+00	1.242e+02 ± 3.21e+00	1.300e+02 ± 1.22e+00	**1.223e+02 ± 4.11e+00**	1.223e+02 ± 3.07e+00
f_{18}	2.958e+05 ± 1.05e+05	3.089e+05 ± 1.38e+05	5.480e+04 ± 3.21e+04	1.280e+05 ± 5.48e+04	**3.661e+03 ± 6.25e+03**	1.041e+04 ± 5.98e+03
f_{19}	5.296e-02 ± 9.80e-09	**5.296e-02 ± 5.22e-18**	5.296e-02 ± 3.28e-11	5.296e-02 ± 4.63e-10	5.296e-02 ± 4.34e-18	5.296e-02 ± 2.20e-16
f_{20}	-9.632e-01 ± 4.22e-02	**-1.067e+00 ± 4.09e-16**	-1.067e+00 ± 1.50e-05	-1.067e+00 ± 2.82e-05	-1.067e+00 ± 3.37e-16	-1.067e+00 ± 1.31e-09
f_{21}	2.337e+01 ± 1.14e+00	3.979e-01 ± 9.70e-13	3.979e-01 ± 1.71e-05	3.979e-01 ± 5.64e-05	**3.979e-01 ± 0.00e+00**	3.979e-01 ± 9.67e-11
f_{22}	-3.760e+00 ± 1.95e-02	**-3.863e+00 ± 1.94e-15**	-3.863e+00 ± 1.21e-06	-3.863e+00 ± 5.89e-06	-3.863e+00 ± 2.04e-15	-3.863e+00 ± 5.50e-07
f_{23}	-4.819e-01 ± 4.27e-02	-3.238e+00 ± 5.53e-02	-3.288e+00 ± 5.54e-02	-3.317e+00 ± 2.43e-02	-3.293e+00 ± 5.27e-02	**-3.322e+00 ± 2.99e-06**
f_{24}	-2.773e+00 ± 1.13e+00	-6.458e+00 ± 2.47e+00	-5.451e+00 ± 3.24e+00	-8.905e+00 ± 2.30e+00	-6.082e+00 ± 3.36e+00	**-9.943e+00 ± 1.03e+00**
f_{25}	-2.628e+00 ± 9.63e-01	-7.258e+00 ± 3.01e+00	-5.504e+00 ± 3.33e+00	-1.011e+01 ± 1.07e+00	-5.557e+00 ± 3.30e+00	**-1.018e+01 ± 1.08e+00**
f_{26}	-2.764e+00 ± 9.00e-01	-6.940e+00 ± 3.19e+00	-6.239e+00 ± 3.75e+00	**-1.014e+01 ± 1.56e+00**	-4.572e+00 ± 2.96e+00	-9.910e+00 ± 2.03e+00

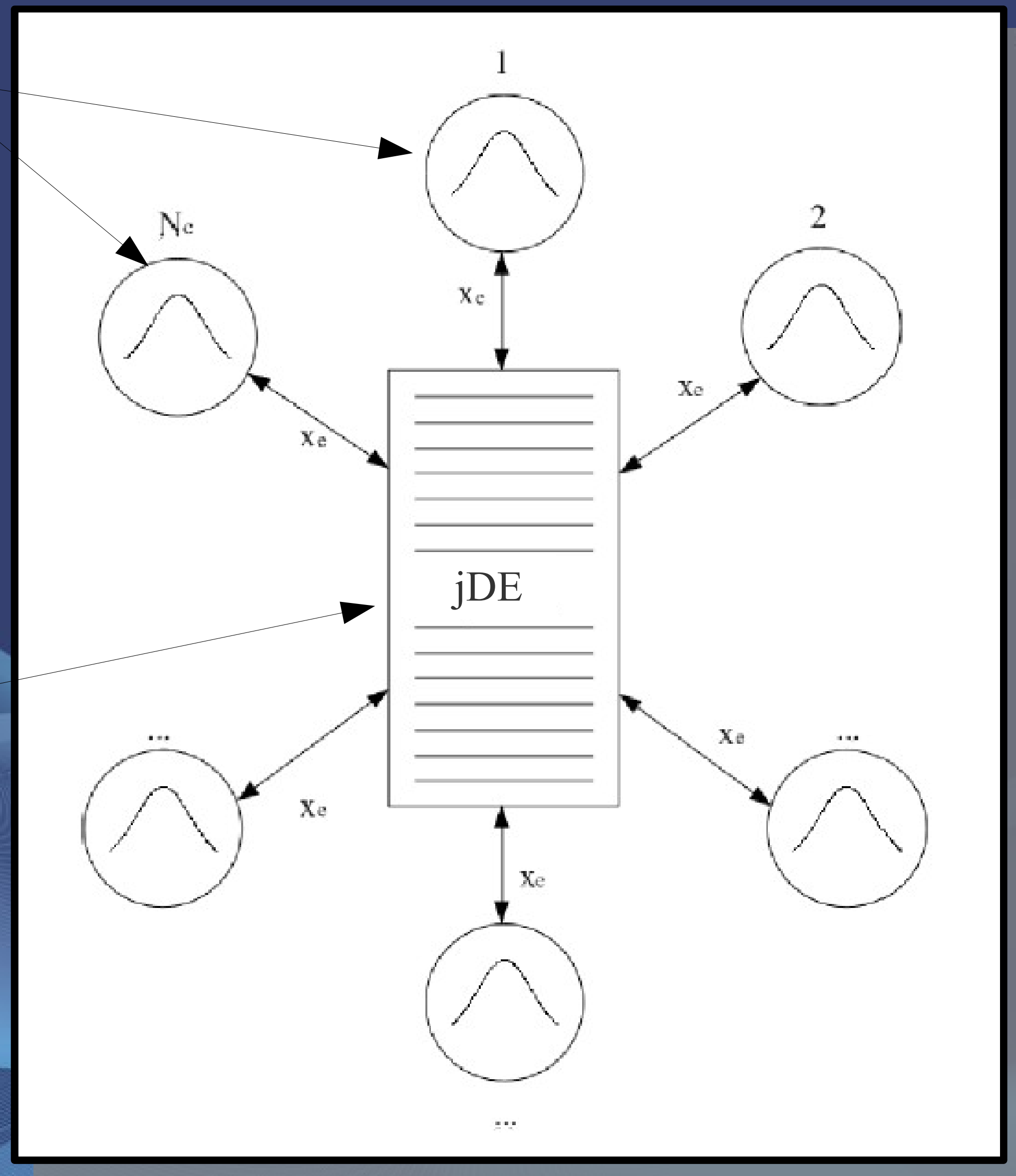
CcDE – Results (2/2)

Table 4: Average final fitness values ± standard deviations for population-based algorithms

Test Problem	$RCEDA_{mvg}$	RCMA	DEahcSPX	CcDE
f_1	6.955e+01 ± 1.47e+02	**3.411e-16 ± 6.85e-16**	9.690e+01 ± 1.50e+01	2.380e-02 ± 3.47e-02
f_2	3.145e+02 ± 3.01e+02	**1.255e-13 ± 2.95e-13**	1.805e+03 ± 2.36e+02	2.062e+02 ± 2.74e+02
f_3	2.319e+06 ± 5.55e+06	**2.827e+01 ± 1.28e-01**	1.110e+05 ± 3.11e+04	9.967e+01 ± 7.48e+01
f_4	3.271e+00 ± 4.71e-01	**7.012e-09 ± 1.04e-08**	2.567e+00 ± 4.64e-01	1.500e-02 ± 1.67e-02
f_5	3.406e+00 ± 1.07e+00	**8.119e-09 ± 8.45e-09**	2.509e+00 ± 4.09e-01	2.586e+00 ± 1.07e+00
f_6	2.638e+02 ± 3.90e+01	6.575e+00 ± 3.85e+00	4.886e+00 ± 3.60e-01	**2.188e-02 ± 2.46e-02**
f_7	2.735e+02 ± 4.50e+01	6.063e+00 ± 3.50e+00	4.824e+00 ± 3.50e-01	**1.292e-01 ± 1.85e-01**
f_8	1.770e+02 ± 1.35e+01	**7.105e-15 ± 2.55e-14**	3.405e+01 ± 3.48e+00	1.433e-02 ± 2.83e-02
f_9	1.816e+02 ± 1.26e+01	**4.737e-15 ± 2.32e-14**	1.172e+02 ± 2.56e+01	7.212e+01 ± 5.20e+01
f_{10}	1.402e+04 ± 5.35e+04	1.003e+04 ± 1.78e+04	3.026e+03 ± 3.75e+02	**8.339e+00 ± 1.04e+01**
f_{11}	1.015e+04 ± 4.23e+02	2.989e+03 ± 5.98e+02	8.258e+02 ± 8.20e+01	**5.546e-02 ± 9.79e-02**
f_{12}	1.294e+00 ± 6.90e-01	**6.329e-10 ± 2.63e-09**	1.333e-01 ± 3.16e-02	1.694e-02 ± 1.82e-02
f_{13}	-9.475e+01 ± 2.21e+01	-6.845e+01 ± 1.17e+01	-8.561e+01 ± 1.39e+00	**-9.999e+01 ± 2.93e-02**
f_{14}	5.928e-01 ± 1.34e+00	**1.030e-06 ± 1.29e-06**	1.157e-02 ± 8.74e-03	1.277e-04 ± 2.39e-04
f_{15}	-1.290e-01 ± 1.90e+00	**-1.149e+00 ± 3.70e-03**	-8.535e-01 ± 1.08e-01	-1.141e+00 ± 1.78e-02
f_{16}	1.007e+04 ± 3.36e+03	9.318e+03 ± 2.45e+03	9.314e+03 ± 1.02e+03	**5.638e+03 ± 2.28e+03**
f_{17}	1.303e+02 ± 7.51e-01	1.256e+02 ± 3.65e+00	1.298e+02 ± 1.15e+00	**1.223e+02 ± 3.07e+00**
f_{18}	5.272e+05 ± 2.93e+05	1.298e+05 ± 4.67e+04	7.203e+04 ± 1.03e+04	**1.041e+04 ± 5.98e+03**
f_{19}	5.296e-02 ± 6.61e-09	**5.296e-02 ± 4.34e-18**	5.296e-02 ± 3.64e-09	5.296e-02 ± 2.20e-16
f_{20}	**-1.067e+00 ± 4.54e-16**	-1.067e+00 ± 6.64e-06	-1.067e+00 ± 2.16e-05	-1.067e+00 ± 1.31e-09
f_{21}	3.987e-01 ± 3.42e-03	3.980e-01 ± 1.95e-04	3.980e-01 ± 1.97e-04	**3.979e-01 ± 9.67e-11**
f_{22}	-3.847e+00 ± 3.37e-02	-3.863e+00 ± 5.91e-06	-3.863e+00 ± 5.56e-06	**-3.863e+00 ± 5.50e-07**
f_{23}	-3.126e+00 ± 1.65e-01	-3.268e+00 ± 6.07e-02	-3.322e+00 ± 1.33e-04	**-3.322e+00 ± 2.99e-06**
f_{24}	-5.633e+00 ± 3.50e+00	-5.302e+00 ± 3.20e+00	**-1.004e+01 ± 1.19e-01**	-9.943e+00 ± 1.03e+00
f_{25}	-9.252e+00 ± 2.42e+00	-4.648e+00 ± 3.68e+00	**-1.024e+01 ± 1.06e-01**	-1.018e+01 ± 1.08e+00
f_{26}	-8.449e+00 ± 3.01e+00	-5.482e+00 ± 3.33e+00	**-1.038e+01 ± 1.93e-01**	-9.910e+00 ± 2.03e+00

Supervised cDE (ScDE)

- Each **cDE unit** performs one offspring generation and possible elite replacement
- When all the compact units performed one step, all the elites are inserted into an auxiliary population
- Within the auxiliary population, the candidate solutions (the elites) are processed by means of one generation of a **global optimizer** (jDE)
- After one generation, a new population of elite solutions is produced. The elite solutions are then injected into the corresponding compact units



Self-Adapting DE (jDE)

generate S_{pop} individuals of the initial population with related parameters pseudo-randomly;

while budget condition

  for $i = 1 : S_{pop}$

    compute $f(x_i)$;

  end-for

  for $i = 1 : S_{pop}$

    $**F_i$ update$**$

    generate $rand_1$ and $rand_2$;

mutation

$$F_i = \begin{cases} F_l + F_u rand_1, & \text{if } rand_2 < \tau_1 \\ F_i, & \text{otherwise} \end{cases};$$

    mutation

    perform standard DE mutation by means of $x'_{off} = x_t + F_i(x_r - x_s)$;

    $**CR_i$ update$**$

    generate $rand_3$ and $rand_4$;

$$CR_i = \begin{cases} rand_3, & \text{if } rand_4 < \tau_2 \\ CR_i, & \text{otherwise} \end{cases}$$

crossover

    crossover

    perform standard DE crossover with the CR_i calculated;

    selection

    perform the standard DE one-to-one spawning selection;

  end-for

end-while

Scale Factor Update Rule (no fixed F value, one value for each individual)

Crossover Ratio Update Rule (no fixed CR value, one value for each individual)

Only four parameters

F_l, F_u *(scale factor bounds)*

τ_1, τ_2 *(update thresholds)*

ScDE - Results

TABLE I
AVERAGE FINAL FITNESS ± STANDARD DEVIATION FOR 30D PROBLEMS

Test Problem	jDE	JADE	SADE	ScDE
f_1	**0.000e+00 ±0.000e+00**	**0.000e+00 ± 0.000e+00**	**0.000e+00 ± 0.000e+00**	**0.000e+00 ± 0.000e+00**
f_2	4.219e+00 ± 4.91e+00	**2.842e-13 ± 2.99e-13**	1.131e-01 ± 2.42e-01	8.795e-02 ± 8.38e-02
f_3	1.656e+06 ± 6.91e+05	**4.408e+04 ± 3.15e+04**	3.393e+05 ± 1.73e+05	1.286e+06 ± 9.94e+05
f_4	2.708e+02 ± 2.86e+02	**3.830e+00 ± 5.91e+00**	3.885e+02 ± 4.47e+02	9.539e+02 ± 5.97e+02
f_5	1.437e+03 ± 6.78e+02	**6.276e+02 ± 4.56e+02**	1.997e+03 ± 7.35e+02	2.890e+03 ± 6.55e+02
f_6	3.438e+01 ± 3.00e+01	**7.489e+00 ± 3.00e+01**	2.209e+01 ± 2.36e+01	3.108e+01 ± 3.35e+01
f_7	**1.139e-02 ± 6.87e-03**	1.498e-02 ± 9.99e-03	4.696e+03 ± 2.08e-07	1.764e-02 ± 1.29e-02
f_8	2.095e+01 ± 6.10e-02	2.098e+01 ± 3.72e-02	2.060e+01 ± 3.78e-01	**2.024e+01 ± 2.16e-01**
f_9	8.291e-01 ± 6.98e-01	**0.000e+00 ± 0.000e+00**	1.526e+01 ± 1.43e+01	7.299e-01 ± 3.58e+00
f_{10}	6.218e-01 ± 6.44e-01	**0.000e+00 ± 0.000e+00**	1.884e+01 ± 1.52e+01	**0.000e+00 ± 0.000e+00**
f_{11}	**1.978e+01 ± 4.58e+00**	2.690e+01 ± 1.73e+00	2.219e+01 ± 3.21e+00	2.314e+01 ± 4.49e+00
f_{12}	7.917e+04 ± 2.54e+04	3.470e+04 ± 5.66e+03	3.314e+04 ± 2.95e+04	**8.943e+03 ± 7.54e+03**
f_{13}	2.803e+00 ± 1.03e+00	1.495e+00 ± 1.01e-01	3.246e+00 ± 9.55e-01	**1.177e+00 ± 2.08e-01**
f_{14}	1.293e+01 ± 4.05e-01	**1.239e+01 ± 2.42e-01**	1.243e+01 ± 5.82e-01	1.250e+01 ± 3.46e-01

TABLE III
AVERAGE FINAL FITNESS ± STANDARD DEVIATION FOR 100D PROBLEMS

Test Problem	jDE	JADE	SADE	ScDE
f_1	**5.684e-14 ± 8.13e-14**	2.274e-13 ± 3.25e-13	2.842e-13 ± 3.04e-13	4.547e-13 ± 5.14e-13
f_2	6.957e+03 ± 2.09e+03	**4.890e+00 ± 2.34e+01**	1.716e+03 ± 7.42e+02	2.630e+02 ± 8.35e+01
f_3	8.799e+06 ± 2.46e+06	**1.844e+06 ± 3.67e+05**	5.825e+06 ± 1.82e+06	9.972e+06 ± 3.43e+06
f_4	1.006e+05 ± 3.05e+04	**5.000e+04 ± 8.68e+03**	9.405e+04 ± 2.25e+04	8.773e+04 ± 2.56e+04
f_5	**9.694e+03 ± 1.20e+03**	1.072e+04 ± 1.77e+03	1.682e+04 ± 2.28e+03	2.217e+04 ± 3.07e+03
f_6	1.335e+02 ± 4.75e+01	1.292e+02 ± 4.52e+01	1.698e+02 ± 4.92e+01	**1.109e+02 ± 2.38e+01**
f_7	**4.181e-03 ± 5.73e-03**	6.466e-03 ± 6.73e-03	1.398e+04 ± 1.10e+03	1.686e-02 ± 1.06e-02
f_8	2.126e+01 ± 2.12e-01	2.126e+01 ± 2.44e-01	2.037e+01 ± 2.90e-01	**2.026e+01 ± 2.25e-01**
f_9	3.384e+01 ± 7.27e+00	**5.684e-14 ± 6.91e-14**	1.661e+02 ± 4.85e+01	5.671e+00 ± 2.78e+01
f_{10}	3.826e+01 ± 7.34e+00	**4.146e-02 ± 2.03e-01**	1.660e+02 ± 5.62e+01	1.513e+01 ± 5.29e+01
f_{11}	**8.800e+01 ± 8.65e+00**	1.316e+02 ± 3.08e+00	1.220e+02 ± 8.64e+00	1.136e+02 ± 1.11e+01
f_{12}	1.732e+06 ± 4.14e+05	7.326e+05 ± 7.55e+04	3.299e+05 ± 1.50e+05	**1.870e+05 ± 2.49e+05**
f_{13}	1.007e+01 ± 1.90e+00	7.625e+00 ± 5.60e-01	1.814e+01 ± 2.74e+00	**4.100e+00 ± 6.44e-01**
f_{14}	4.679e+01 ± 5.41e-01	**4.576e+01 ± 5.46e-01**	4.649e+01 ± 8.55e-01	4.601e+01 ± 7.53e-01

Super-Fit and Population Size Reduction cDE (SFcDE-PSR)

```
counter t = 0
for i = 1 : n do
    {** PV initialization **}
    initialize μ[i] = 0
    initialize σ[i] = λ = 10
end for
generate elite x_e by means of PV
{**Super-fit generation**}
while budget condition OR tolerance condition do
    apply Rosenbrock Algorithm to x_e
end while
replace the original elite with the solution improved by Rosenbrock Algorithm
calculate remaining budget and Population Size Reduction conditions
while budget condition do
    {** Mutation **}
    generate 3 individuals x_r, x_s, and x_t by means of PV
    compute x'_off = x_t + F(x_r − x_s)
    {** Crossover **}
    x_off = x'_off
    for i = 1 : n do
        generate rand(0, 1)
        if rand(0, 1) > Cr then
            x_off[i] = elite[i]
        end if
    end for
    {** Elite Selection **}
    [winner, loser] = compete(x_off, elite)
    if x_off == winner then
        elite = x_off
    end if
    {** PV Update **}
    for i = 1 : n do
        μ^{t+1}[i] = μ^t[i] + (1/N_p)(winner[i] − loser[i])
        σ^{t+1} = sqrt((σ^t[i])^2 + (μ^t[i])^2 − (μ^{t+1}[i])^2 + (1/N_p)(winner^2[i] − loser^2[i]))
    end for
    t = t + 1
    {**Virtual Population Size Reduction**}
    if t == Population Size Reduction condition then
        N_p = N_p / 2
    end if
end while
```

Super-Fit generation

Virtual Population Size Reduction

- Part of the total budget (1/5 of the maximum number of FEs) is reserved to Super-Fit generation (Rosenbrock Algorithm)
- Virtual Population Size very large at the beginning (exploration), then progressively halved (increasing exploitation)

SFcDE-PSR - Results

TABLE I
AVERAGE FINAL FITNESS VALUES ± STANDARD DEVIATIONS

Test Problem	cDE	cDE-PSR	SFcDE	SFcDE-PSR
f_1	-4.048e+02 ± 1.30e+02	-4.278e+02 ± 7.40e+01	**-4.500e+02 ± 3.20e-11**	**-4.500e+02 ± 3.20e-11**
f_2	4.382e+03 ± 3.89e+03	3.139e+03 ± 1.67e+03	**-4.497e+02 ± 2.36e-01**	**-4.496e+02 ± 2.90e-01**
f_3	1.188e+07 ± 6.54e+06	1.597e+07 ± 9.80e+06	**8.290e+05 ± 3.56e+05**	**8.493e+05 ± 3.57e+05**
f_4	3.456e+04 ± 1.12e+04	2.391e+04 ± 8.46e+03	3.961e+04 ± 2.23e+04	**2.350e+04 ± 9.14e+03**
f_5	7.327e+03 ± 1.85e+03	**6.561e+03 ± 1.48e+03**	9.476e+03 ± 3.96e+03	7.517e+03 ± 2.67e+03
f_6	1.586e+03 ± 3.39e+03	3.279e+04 ± 1.45e+05	**4.910e+02 ± 1.27e+02**	5.624e+02 ± 3.16e+02
f_7	4.516e+03 ± 9.85e-08	4.516e+03 ± 2.69e-02	**-1.800e+02 ± 1.37e-02**	**-1.800e+02 ± 1.37e-02**
f_8	-1.194e+02 ± 1.51e-01	-1.194e+02 ± 1.28e-01	**-1.200e+02 ± 2.41e-02**	**-1.200e+02 ± 2.54e-02**
f_9	-2.915e+02 ± 9.48e+00	-2.989e+02 ± 8.25e+00	-2.885e+02 ± 1.08e+01	**-2.992e+02 ± 7.92e+00**
f_{10}	-2.937e+02 ± 1.01e+01	**-3.002e+02 ± 9.67e+00**	-2.901e+02 ± 1.12e+01	-2.978e+02 ± 6.90e+00
f_{11}	1.211e+02 ± 2.58e+00	**1.190e+02 ± 3.27e+00**	1.250e+02 ± 5.04e+00	1.235e+02 ± 5.27e+00
f_{12}	5.319e+04 ± 2.57e+04	5.346e+04 ± 3.51e+04	**2.241e+03 ± 3.26e+03**	**2.241e+03 ± 3.26e+03**
f_{13}	-1.250e+02 ± 1.47e+00	**-1.251e+02 ± 1.39e+00**	-1.240e+02 ± 1.92e+00	-1.245e+02 ± 1.94e+00
f_{14}	-2.870e+02 ± 3.95e-01	**-2.871e+02 ± 4.05e-01**	-2.869e+02 ± 3.48e-01	**-2.871e+02 ± 3.66e-01**

Compact Opposition DE(cODE)

- Opposition Based Learning (OBL)
- Beneficial on non-separable functions

  (diagonal moves in an hyperspace), detrimental otherwise
- OBL is more effective if combined with rand/1/bin DE scheme

```
counter t = 0
for i = 1 : n do
    {** PV initialization **}
    initialize μ[i] = 0
    initialize σ[i] = λ
end for
generate elite x_e by means of PV
while budget condition do
    {** Mutation **}
    generate 3 individuals x_r, x_s, and x_t by means of PV
    compute x'_off = x_t + F(x_r − x_s)
    {** Crossover **}
    apply crossover (binomial or exponential) and generate x_off
    {** Generalized Opposition-Based Learning**}
    if rand(0, 1) < jr then
        compute a = μ − 0.5 · σ and b = μ + 0.5 · σ
        compute k = rand(0, 1)
        compute x̃_off = k(a + b) − x_off
        if f(x̃_off) ≤ f(x_off) then
            x_off = x̃_off
        end if
    end if
    {** Elite Selection **}
    [winner, loser] = compete(x_off, x_e)
    if x_off == winner then
        x_e = x_off
    end if
    {** PV Update **}
    μ^{t+1} = μ^t + (1/N_p)(winner − loser)
    σ^{t+1} = sqrt((σ^t)^2 + (μ^t)^2 − (μ^{t+1})^2 + (1/N_p)(winner^2 − loser^2))
    t = t + 1
end while
```

OBL

cODE - Results

	binomial crossover			exponential crossover		
Test Problem	cDE/rand/1/bin	cODE/rand/1/bin		cDE/rand/1/exp	cODE/rand/1/exp	
f_{1}	**4.520e-28 ± 1.74e-27**	7.877e-26 ± 1.22e-25	-	**0.000e+00 ± 0.00e+00**	2.234e-27 ± 6.48e-27	-
f_{2}	9.865e+03 ± 2.52e+03	**2.814e-27 ± 8.40e-27**	+	1.204e+03 ± 7.58e+02	**1.416e-26 ± 5.40e-26**	+
f_{3}	9.898e+01 ± 1.41e+02	**2.438e+01 ± 9.04e-01**	=	1.099e+02 ± 9.29e+01	**2.549e+01 ± 9.43e-01**	+
f_{4}	1.074e+01 ± 1.75e+00	**4.604e-14 ± 1.07e-13**	+	2.931e-14 ± 6.95e-15	**1.155e-14 ± 1.73e-14**	+
f_{5}	1.028e+01 ± 1.83e+00	**3.049e-14 ± 4.24e-14**	+	4.466e+00 ± 1.34e+00	**7.253e-15 ± 1.80e-14**	+
f_{6}	1.883e-01 ± 2.03e-01	**1.309e-15 ± 4.95e-16**	+	1.947e-03 ± 7.35e-03	**0.000e+00 ± 0.00e+00**	+
f_{7}	1.891e-01 ± 2.06e-01	**2.558e-15 ± 1.02e-15**	+	2.360e-01 ± 2.27e-01	**1.184e-15 ± 9.85e-16**	+
f_{8}	5.959e+01 ± 1.33e+01	**0.000e+00 ± 0.00e+00**	+	1.053e+01 ± 3.60e+00	**0.000e+00 ± 0.00e+00**	+
f_{9}	1.219e+02 ± 2.58e+01	**1.231e+01 ± 4.19e+01**	+	1.498e+02 ± 2.46e+01	**0.000e+00 ± 0.00e+00**	+
f_{10}	6.448e+03 ± 2.75e+03	**3.688e+02 ± 7.30e+02**	+	1.287e+02 ± 2.14e+02	**2.812e+01 ± 1.31e+01**	+
f_{11}	**9.972e+02 ± 3.25e+02**	1.121e+03 ± 3.73e+02	=	**1.439e+02 ± 1.15e+02**	2.370e+02 ± 1.40e+02	-
f_{12}	2.558e-02 ± 7.10e-03	**4.158e-03 ± 1.44e-03**	+	**9.252e-17 ± 4.53e-16**	2.062e-09 ± 2.03e-09	-
f_{13}	-1.000e+02 ± 1.73e-06	**-1.000e+02 ± 3.73e-08**	+	**-1.000e+02 ± 1.66e-09**	-1.000e+02 ± 1.29e-08	-
f_{14}	1.982e-04 ± 1.74e-04	**5.570e-06 ± 6.30e-06**	+	**1.262e-23 ± 1.15e-23**	4.334e-18 ± 1.66e-17	-
f_{15}	**-1.148e+00 ± 1.67e-03**	-1.059e+00 ± 5.13e-02	-	**-1.150e+00 ± 2.24e-03**	-1.067e+00 ± 5.92e-02	-
f_{16}	**8.023e+03 ± 3.42e+03**	8.819e+03 ± 1.80e+03	=	**9.773e+03 ± 3.30e+03**	9.983e+03 ± 2.90e+03	=
f_{17}	**1.242e+02 ± 3.21e+00**	1.301e+02 ± 8.67e-01	-	**1.246e+02 ± 4.19e+00**	1.288e+02 ± 1.68e+00	-
f_{18}	5.480e+04 ± 3.21e+04	**4.373e+04 ± 2.78e+04**	=	**3.507e+04 ± 2.01e+04**	4.566e+04 ± 2.30e+04	=
f_{19}	**5.296e-02 ± 3.28e-11**	5.296e-02 ± 1.83e-09	-	**5.296e-02 ± 4.80e-18**	5.296e-02 ± 1.70e-09	-
f_{20}	**-1.067e+00 ± 1.50e-05**	-1.067e+00 ± 4.10e-05	=	**-1.067e+00 ± 3.93e-16**	-1.067e+00 ± 1.32e-04	-
f_{21}	**3.979e-01 ± 1.71e-05**	3.980e-01 ± 1.56e-04	-	**3.979e-01 ± 1.52e-07**	3.979e-01 ± 4.30e-05	-
f_{22}	**-3.863e+00 ± 1.21e-06**	-3.863e+00 ± 6.74e-06	-	**-3.863e+00 ± 1.86e-15**	-3.863e+00 ± 1.37e-08	=
f_{23}	**-3.288e+00 ± 5.54e-02**	-3.288e+00 ± 5.54e-02	=	**-3.268e+00 ± 6.07e-02**	-3.258e+00 ± 6.07e-02	=
f_{24}	-5.451e+00 ± 3.24e+00	**-9.790e+00 ± 1.54e+00**	+	-5.965e+00 ± 3.19e+00	**-8.477e+00 ± 2.71e+00**	=
f_{25}	-5.504e+00 ± 3.33e+00	**-8.886e+00 ± 3.01e+00**	+	-6.137e+00 ± 3.19e+00	**-9.729e+00 ± 2.15e+00**	=
f_{26}	-6.239e+00 ± 3.75e+00	**-1.025e+01 ± 1.37e+00**	=	-6.622e+00 ± 3.48e+00	**-8.865e+00 ± 3.08e+00**	=

Noise Analysis cDE (NAcDE)

```
counter t = 0
for i = 1 : n do
    {** PV initialization **}
    initialize μ[i] = 0
    initialize σ[i] = λ = 10
end for
generate elite x_e by means of PV
θ = 0
while budget condition do
    {** Mutation **}
    generate 3 individuals x_r, x_s, and x_t by means of PV
    compute x'_off = x_t + F(x_r − x_s)
    {** Crossover **}
    x_off = x'_off
    for i = 1 : n do
        generate rand(0, 1)
        if rand(0, 1) > Cr then
            x_off[i] = elite[i]
        end if
    end for
    {** Elite Selection **}
    [winner, loser] = compete(x_off, x_e)
    θ = θ + 1
    if x_off == winner OR θ ≥ η then
        elite = x_off
        θ = 0
    end if
    {** PV Update **}
    for i = 1 : n do
        μ^{t+1}[i] = μ^t[i] + (1/N_p)(winner[i] − loser[i])
        σ^{t+1} = sqrt((σ^t[i])^2 + (μ^t[i])^2 − (μ^{t+1}[i])^2 + (1/N_p)(winner^2[i] − loser^2[i]))
    end for
    t = t + 1
end while
```

```
{**Compete function with Noise Analysis**}
[winner, loser] = compete(x_off, x_e)
{********}
winner = x_e and loser = x_off
if |f̄(x_e) − f̄(x_off)| > 2σ then
    if f̄(x_off) ≤ f̄(x_e) then
        winner = x_off and loser = x_e
    end if
else
    α = min{f̄(x_i), f̄(x_off)}
    β = max{f̄(x_e), f̄(x_off)}
    compute υ = (α+2σ−(β−2σ)) / (β+2σ−(α−2σ))
    compute n_s = ⌈(1.96 / (2·(1−υ)))^2⌉
    perform re-sampling
    update f̄(x_e) and f̄(x_off)
    if f̄(x_off) ≤ f̄(x_e) then
        winner = x_off and loser = x_e
    end if
end if
```

- Gaussian noise on fitness function, (zero mean, different levels of std. Deviation)
- Re-sampling and filtering used to perform selection in noisy environment

NAcDE – Results (1/3)

Table 1. Final average fitness ± standard deviation for $n = 10$

Test Problem		DE-RSF-TS	ODE	NADE	NAcGA	NAcDE
f_1	0	**1.17e-09±5.13e-10**	1.05e+02±1.63e+02	1.23e-09±5.32e-10	5.00e+00±4.36e+00	1.86e-08±1.45e-08
	0.05	4.00e+03±9.96e+02	5.74e+03±1.86e+03	2.04e+03±1.26e+03	2.12e+03±6.03e+02	**1.86e+03±6.85e+02**
	0.1	7.05e+03±2.45e+03	9.62e+03±3.72e+03	3.42e+03±1.36e+03	4.18e+03±1.02e+03	**2.84e+03±1.03e+03**
	0.2	1.30e+04±4.30e+03	1.41e+04±4.82e+03	5.99e+03±1.75e+03	6.39e+03±2.10e+03	**4.63e+03±1.19e+03**
f_2	0	1.44e+00±5.41e-01	2.53e+03±1.80e+03	1.44e+00±6.80e-01	5.12e+01±3.40e+01	**3.43e-02±9.26e-02**
	0.05	1.76e+04±5.96e+03	1.92e+04±6.86e+03	1.43e+04±6.62e+03	1.08e+04±3.37e+03	**9.80e+03±3.18e+03**
	0.1	1.82e+04±6.86e+03	1.93e+04±6.88e+03	1.68e+04±6.71e+03	1.64e+04±5.17e+03	**1.39e+04±4.20e+03**
	0.2	1.92e+04±6.79e+03	1.81e+04±6.79e+03	1.73e+04±5.94e+03	1.57e+04±5.13e+03	**1.44e+04±3.95e+03**
f_3	0	1.32e+06±4.21e+05	2.56e+07±7.64e+06	**1.28e+06±4.55e+05**	1.68e+06±7.35e+05	1.42e+06±8.39e+05
	0.05	1.62e+08±8.15e+07	2.33e+08±1.45e+08	1.24e+08±9.08e+07	8.17e+07±4.10e+07	**6.04e+07±3.25e+07**
	0.1	2.26e+08±1.37e+08	1.99e+08±1.14e+08	2.34e+08±1.43e+08	1.22e+08±7.22e+07	**9.68e+07±5.75e+07**
	0.2	2.48e+08±1.37e+08	2.68e+08±1.74e+08	2.57e+08±1.70e+08	1.70e+08±7.64e+07	**1.36e+08±7.56e+07**
f_4	0	1.07e+04±5.43e+03	3.03e+03±1.60e+03	**1.36e+03±7.20e+02**	5.67e+03±3.51e+03	1.53e+04±5.60e+03
	0.05	2.32e+04±8.64e+03	2.66e+04±8.76e+03	**2.12e+04±7.73e+03**	5.48e+04±3.17e+04	3.82e+04±2.00e+04
	0.1	2.44e+04±5.81e+03	2.48e+04±6.76e+03	**2.42e+04±6.79e+03**	5.00e+04±3.26e+04	4.00e+04±2.56e+04
	0.2	2.54e+04±7.91e+03	2.49e+04±6.87e+03	**2.43e+04±6.30e+03**	6.30e+04±4.37e+04	4.27e+04±1.89e+04
f_5	0	**7.06e-07±4.00e-07**	4.69e+03±2.33e+03	7.11e-07±4.31e-07	5.87e+01±2.90e+01	2.76e+01±6.26e+01
	0.05	1.75e+03±6.97e+02	6.25e+03±1.91e+03	1.19e+03±3.25e+02	3.55e+03±1.22e+03	**1.04e+03±4.04e+02**
	0.1	4.25e+03±1.48e+03	7.18e+03±1.75e+03	2.74e+03±8.59e+02	6.79e+03±1.10e+03	**1.92e+03±6.59e+02**
	0.2	1.01e+04±2.48e+03	8.28e+03±8.48e+02	4.44e+03±1.54e+03	1.03e+04±1.91e+03	**4.29e+03±1.41e+03**
f_6	0	**5.19e+00±5.41e-01**	5.30e+06±1.79e+07	5.30e+00±4.81e-01	2.35e+03±2.08e+03	1.01e+03±1.01e+03
	0.05	1.65e+09±8.44e+08	2.27e+09±1.36e+09	9.90e+08±6.64e+08	5.87e+08±4.41e+08	**2.89e+08±1.36e+08**
	0.1	2.33e+09±1.20e+09	3.24e+09±1.77e+09	1.34e+09±6.98e+08	9.96e+08±6.67e+08	**3.43e+08±1.97e+08**
	0.2	3.79e+09±2.33e+09	3.82e+09±1.57e+09	2.19e+09±1.26e+09	1.70e+09±1.04e+09	**7.66e+08±6.32e+08**
f_7	0	1.27e+03±5.08e-02	1.27e+03±2.27e-01	**1.27e+03±4.65e-13**	1.27e+03±4.35e-01	1.27e+03±8.73e-07
	0.05	1.36e+03±5.48e+01	**1.27e+03±4.65e-13**	1.32e+03±2.65e+01	1.63e+03±6.71e+01	1.32e+03±2.44e+01
	0.1	1.49e+03±1.04e+02	**1.27e+03±1.45e-01**	1.37e+03±7.80e+01	1.88e+03±1.24e+02	1.38e+03±6.26e+01
	0.2	1.99e+03±3.89e+02	**1.28e+03±3.39e+01**	1.47e+03±1.08e+02	2.22e+03±1.55e+02	1.52e+03±7.10e+01
f_8	0	2.04e+01±7.81e-02	2.06e+01±1.20e-01	2.04e+01±7.84e-02	2.04e+01±6.63e-02	**2.04e+01±8.64e-02**
	0.05	2.04e+01±1.12e-01	2.06e+01±1.14e-01	2.04e+01±1.04e-01	2.05e+01±7.56e-02	**2.04e+01±8.37e-02**
	0.1	2.05e+01±1.30e-01	2.06e+01±1.56e-01	2.05e+01±1.05e-01	2.05e+01±9.11e-02	**2.04e+01±9.12e-02**
	0.2	2.07e+01±1.72e-01	2.07e+01±1.08e-01	2.06e+01±1.75e-01	2.06e+01±1.04e-01	**2.05e+01±9.70e-02**
f_9	0	7.41e+00±1.58e+00	1.87e+01±6.45e+00	7.51e+00±1.35e+00	**4.59e+00±1.09e+00**	1.02e+01±5.08e+00
	0.05	5.01e+01±1.07e+01	3.90e+01±9.72e+00	3.22e+01±8.37e+00	2.52e+01±5.55e+00	**2.11e+01±5.80e+00**
	0.1	7.35e+01±1.74e+01	7.57e+01±2.05e+01	6.20e+01±1.37e+01	4.29e+01±1.01e+01	**4.26e+01±1.07e+01**
	0.2	1.04e+02±1.99e+01	1.08e+02±1.58e+01	9.05e+01±1.35e+01	6.69e+01±1.04e+01	**6.43e+01±1.33e+01**
f_{10}	0	7.99e+00±1.78e+00	1.63e+01±6.28e+00	7.84e+00±1.27e+00	**4.19e+00±1.09e+00**	1.02e+01±4.42e+00
	0.05	4.52e+01±5.82e+00	3.79e+01±8.61e+00	3.43e+01±7.73e+00	2.29e+01±4.32e+00	**2.15e+01±8.12e+00**
	0.1	7.90e+01±1.59e+01	7.20e+01±1.53e+01	5.23e+01±1.30e+01	**4.20e+01±9.69e+00**	4.34e+01±8.94e+00
	0.2	1.00e+02±1.75e+01	1.05e+02±1.47e+01	8.38e+01±1.70e+01	**6.52e+01±1.15e+01**	6.58e+01±6.47e+00
f_{11}	0	9.30e+00±5.75e-01	1.02e+01±8.17e-01	9.17e+00±7.44e-01	6.48e+00±6.19e-01	**5.63e+00±1.61e+00**
	0.05	9.93e+00±1.01e+00	1.02e+01±8.41e-01	9.47e+00±5.45e-01	**7.45e+00±7.29e-01**	7.81e+00±1.56e+00
	0.1	1.05e+01±9.29e-01	1.11e+01±1.07e+00	1.03e+01±1.11e+00	**7.89e+00±7.66e-01**	9.66e+00±5.48e-01
	0.2	1.18e+01±1.47e+00	1.17e+01±1.24e+00	1.16e+01±1.53e+00	**9.45e+00±6.45e-01**	1.03e+01±8.40e-01
f_{12}	0	7.94e+03±2.24e+03	1.24e+04±6.21e+03	7.86e+03±3.00e+03	**2.10e+03±7.40e+02**	9.30e+03±6.23e+03
	0.05	5.89e+04±1.77e+04	4.98e+04±1.57e+04	4.85e+04±1.22e+04	**2.26e+04±5.77e+03**	3.45e+04±1.26e+04
	0.1	9.41e+04±2.75e+04	9.36e+04±3.22e+04	6.59e+04±3.17e+04	**3.68e+04±1.07e+04**	4.41e+04±1.50e+04
	0.2	1.17e+05±4.64e+04	1.30e+05±4.66e+04	1.16e+05±4.28e+04	**5.71e+04±2.52e+04**	5.90e+04±1.72e+04
f_{13}	0	2.18e+00±3.35e-01	2.56e+00±8.73e-01	2.13e+00±3.42e-01	**7.97e-01±3.83e-01**	9.00e-01±3.19e-01
	0.05	4.90e+01±3.74e+01	4.05e+01±2.69e+01	3.62e+01±1.96e+01	2.71e+01±1.42e+01	**1.90e+01±4.86e+00**
	0.1	6.05e+01±4.67e+01	4.83e+01±3.76e+01	4.91e+01±3.71e+01	2.85e+01±2.02e+01	**2.59e+01±1.08e+01**
	0.2	6.10e+01±4.82e+01	6.07e+01±4.83e+01	5.61e+01±4.20e+01	3.79e+01±1.82e+01	**2.73e+01±1.35e+01**
f_{14}	0	3.86e+00±2.02e-01	4.17e+00±2.27e-01	3.89e+00±1.51e-01	3.46e+00±2.43e-01	**3.32e+00±4.04e-01**
	0.05	3.94e+00±1.39e-01	4.27e+00±8.76e-02	3.99e+00±9.06e-02	3.70e+00±1.49e-01	**3.33e+00±3.90e-01**
	0.1	4.01e+00±1.95e-01	4.28e+00±8.99e-02	4.06e+00±1.48e-01	**3.75e+00±2.36e-01**	3.87e+00±1.93e-01
	0.2	4.19e+00±2.77e-01	4.33e+00±1.34e-01	4.16e+00±2.27e-01	**4.00e+00±1.60e-01**	4.15e+00±1.13e-01

NAcDE – Results (2/3)

Table 3. Final average fitness ± standard deviation for $n = 30$

Test	Problem	DE-RSF-TS	OBDE	NADE	NArCGA	NAcDE
f_1	0	2.05e-05±8.89e-06	7.02e+03±3.24e+03	**1.80e-05±7.38e-06**	1.88e-05±1.92e-05	3.34e+02±4.08e+02
	0.05	1.82e+04±4.86e+03	3.28e+04±9.04e+03	**9.01e+03±2.73e+03**	1.54e+04±2.42e+03	1.61e+04±2.62e+03
	0.1	3.99e+04±8.11e+03	6.56e+04±2.07e+04	**1.63e+04±4.30e+03**	2.03e+04±2.99e+03	2.16e+04±3.16e+03
	0.2	6.83e+04±1.71e+04	9.38e+04±1.36e+04	3.11e+04±7.54e+03	4.41e+04±7.27e+03	**2.85e+04±5.32e+03**
f_2	0	1.36e+04±2.51e+03	6.98e+04±9.27e+03	1.43e+04±2.90e+03	**1.15e+03±1.08e+03**	1.47e+04±5.98e+03
	0.05	1.56e+05±3.92e+04	1.64e+05±4.24e+04	1.51e+05±3.34e+04	**9.90e+04±1.27e+04**	1.00e+05±1.67e+04
	0.1	1.62e+05±3.96e+04	1.63e+05±4.04e+04	1.52e+05±3.69e+04	**1.04e+05±1.19e+04**	1.13e+05±2.61e+04
	0.2	1.58e+05±3.48e+04	1.64e+05±4.24e+04	1.55e+05±4.41e+04	**9.77e+04±2.10e+04**	1.24e+05±2.28e+04
f_3	0	1.45e+08±3.50e+07	4.25e+08±1.36e+08	1.50e+08±2.24e+07	**9.58e+06±5.18e+06**	5.06e+07±2.04e+07
	0.05	1.42e+09±4.50e+08	1.74e+09±4.83e+08	7.13e+08±2.61e+08	5.59e+08±7.71e+07	**3.96e+08±1.48e+08**
	0.1	1.79e+09±5.69e+08	2.18e+09±5.42e+08	1.24e+09±4.49e+08	9.74e+08±1.97e+08	**6.14e+08±1.60e+08**
	0.2	2.32e+09±6.03e+08	2.28e+09±5.08e+08	1.29e+09±5.90e+08	1.15e+09±2.90e+08	**1.01e+09±3.22e+08**
f_4	0	1.01e+05±3.59e+04	7.59e+04±1.17e+04	**3.96e+04±1.31e+04**	9.23e+04±3.34e+04	1.27e+05±3.10e+04
	0.05	1.99e+05±4.72e+04	2.08e+05±4.52e+04	**1.87e+05±5.05e+04**	2.94e+05±1.69e+05	2.58e+05±1.56e+05
	0.1	**1.96e+05±3.85e+04**	2.07e+05±4.38e+04	2.04e+05±3.78e+04	2.97e+05±1.48e+05	3.09e+05±1.84e+05
	0.2	**2.01e+05±3.94e+04**	2.09e+05±4.79e+04	2.06e+05±4.76e+04	3.29e+05±1.55e+05	2.75e+05±1.72e+05
f_5	0	2.07e+03±7.48e+02	1.85e+04±2.40e+03	**1.95e+03±4.25e+02**	3.71e+03±1.51e+03	6.31e+03±1.37e+03
	0.05	1.81e+04±1.89e+03	2.10e+04±1.70e+03	**1.15e+04±1.16e+03**	1.19e+04±1.26e+03	1.16e+04±1.41e+03
	0.1	2.74e+04±4.01e+03	2.72e+04±5.80e+03	1.60e+04±2.61e+03	1.68e+04±2.58e+03	**1.45e+04±1.79e+03**
	0.2	3.64e+04±4.91e+03	3.76e+04±6.82e+03	2.23e+04±2.17e+03	2.81e+04±3.09e+03	**1.84e+04±3.34e+03**
f_6	0	3.82e+01±3.64e+01	5.70e+08±3.96e+08	**3.64e+01±3.86e+01**	4.82e+03±4.61e+03	4.32e+06±1.05e+07
	0.05	1.61e+10±5.76e+09	2.96e+10±9.28e+09	5.80e+09±2.70e+09	4.62e+09±1.05e+09	**4.16e+09±1.40e+09**
	0.1	2.75e+10±9.39e+09	4.90e+10±1.76e+10	1.20e+10±5.82e+09	9.90e+09±3.80e+09	**6.31e+09±2.43e+09**
	0.2	5.56e+10±1.88e+10	7.19e+10±2.03e+10	1.94e+10±7.23e+09	2.70e+10±6.77e+09	**1.07e+10±4.33e+09**
f_7	0	4.70e+03±7.81e-12	4.78e+03±2.36e+01	**4.70e+03±1.92e-12**	4.70e+03±5.54e-01	4.70e+03±1.99e-06
	0.05	5.50e+03±1.59e+02	**4.79e+03±0.00e+00**	5.04e+03±9.11e+01	5.27e+03±8.31e+01	5.00e+03±1.07e+02
	0.1	6.39e+03±4.41e+02	**4.82e+03±9.07e+01**	5.33e+03±1.26e+02	6.11e+03±3.35e+02	5.28e+03±1.20e+02
	0.2	8.64e+03±9.03e+02	**5.38e+03±3.57e+02**	5.83e+03±3.37e+02	8.48e+03±5.01e+02	5.83e+03±3.55e+02
f_8	0	2.10e+01±4.81e-02	2.10e+01±5.89e-02	2.10e+01±6.75e-02	2.10e+01±6.33e-02	**2.09e+01±1.16e-01**
	0.05	**2.10e+01±8.67e-02**	2.10e+01±5.47e-02	2.10e+01±5.37e-02	2.10e+01±6.48e-02	2.10e+01±7.24e-02
	0.1	2.11e+01±9.23e-02	2.10e+01±8.92e-02	2.10e+01±1.12e-01	2.10e+01±4.65e-02	**2.10e+01±6.13e-02**
	0.2	2.11e+01±1.18e-01	2.11e+01±1.33e-01	2.11e+01±9.13e-02	2.10e+01±5.22e-02	**2.10e+01±6.94e-02**
f_9	0	**4.62e+01±6.93e+00**	1.61e+02±2.59e+01	4.69e+01±5.63e+00	7.21e+01±1.34e+01	6.26e+01±1.38e+01
	0.05	2.47e+02±2.50e+01	2.42e+02±3.13e+01	1.67e+02±1.97e+01	1.93e+02±1.61e+01	**1.36e+02±1.73e+01**
	0.1	3.49e+02±3.05e+01	3.78e+02±6.10e+01	2.41e+02±3.37e+01	2.52e+02±2.03e+01	**2.08e+02±3.66e+01**
	0.2	4.84e+02±5.38e+01	5.09e+02±4.44e+01	3.22e+02±3.49e+01	3.36e+02±3.36e+01	**2.82e+02±2.72e+01**
f_{10}	0	**4.73e+01±4.65e+00**	1.57e+02±2.43e+01	4.79e+01±5.18e+00	6.99e+01±1.24e+01	6.39e+01±1.20e+01
	0.05	2.16e+02±2.03e+01	2.08e+02±2.92e+01	**1.44e+02±2.42e+01**	2.03e+02±1.24e+01	1.47e+02±2.41e+01
	0.1	3.27e+02±3.38e+01	3.61e+02±3.48e+01	2.43e+02±3.19e+01	2.57e+02±2.07e+01	**2.41e+02±2.84e+01**
	0.2	4.52e+02±4.89e+01	4.82e+02±5.86e+01	3.01e+02±4.31e+01	3.57e+02±3.00e+01	**2.92e+02±2.68e+01**
f_{11}	0	4.00e+01±1.18e+00	4.10e+01±1.08e+00	4.03e+01±1.49e+00	3.37e+01±2.25e+00	**2.94e+01±3.05e+00**
	0.05	4.11e+01±1.11e+00	4.15e+01±1.76e+00	4.14e+01±1.60e+00	3.90e+01±1.22e+00	**3.31e+01±2.77e+00**
	0.1	4.27e+01±2.05e+00	4.29e+01±1.98e+00	4.21e+01±2.01e+00	3.99e+01±1.19e+00	**3.53e+01±3.33e+00**
	0.2	4.55e+01±2.12e+00	4.52e+01±2.28e+00	4.39e+01±2.22e+00	4.13e+01±1.70e+00	**4.01e+01±2.07e+00**
f_{12}	0	4.02e+05±5.99e+04	3.51e+05±8.64e+04	4.10e+05±6.06e+04	**2.46e+05±7.47e+04**	3.21e+05±7.99e+04
	0.05	9.72e+05±1.22e+05	8.38e+05±1.05e+05	7.49e+05±1.53e+05	6.78e+05±1.30e+05	**5.70e+05±1.52e+05**
	0.1	1.34e+06±2.67e+05	1.22e+06±1.99e+05	9.91e+05±1.87e+05	9.91e+05±1.31e+05	**9.24e+05±1.71e+05**
	0.2	1.78e+06±2.96e+05	1.62e+06±2.27e+05	**1.28e+06±2.40e+05**	1.32e+06±1.67e+05	1.37e+06±2.06e+05
f_{13}	0	1.45e+01±1.08e+00	2.87e+01±9.19e+00	1.45e+01±1.07e+00	**9.93e+00±2.34e+00**	1.80e+01±1.17e+01
	0.05	7.64e+02±2.95e+02	9.97e+02±4.87e+02	3.04e+02±1.11e+02	**2.17e+02±7.89e+01**	2.95e+02±8.90e+01
	0.1	1.15e+03±4.80e+02	1.28e+03±6.05e+02	7.30e+02±4.38e+02	**3.04e+02±7.99e+01**	3.90e+02±1.46e+02
	0.2	1.42e+03±5.40e+02	1.31e+03±5.59e+02	1.06e+03±4.98e+02	**5.06e+02±2.20e+02**	5.59e+02±2.19e+02
f_{14}	0	1.36e+01±1.63e-01	1.39e+01±1.60e-01	1.37e+01±1.26e-01	1.32e+01±2.50e-01	**1.25e+01±4.88e-01**
	0.05	1.37e+01±1.51e-01	1.39e+01±1.95e-01	1.37e+01±1.41e-01	1.34e+01±1.84e-01	**1.26e+01±4.96e-01**
	0.1	1.38e+01±2.14e-01	1.40e+01±1.86e-01	1.38e+01±2.00e-01	1.35e+01±2.34e-01	**1.29e+01±4.40e-01**
	0.2	1.41e+01±2.13e-01	1.42e+01±1.82e-01	1.40e+01±2.04e-01	1.36e+01±1.72e-01	**1.35e+01±2.98e-01**

NAcDE – Results (3/3)

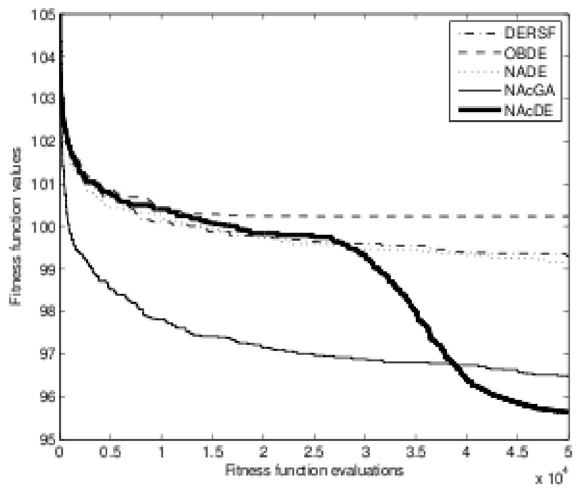


Figure 4. f_{11} for $n = 10$ and no noise

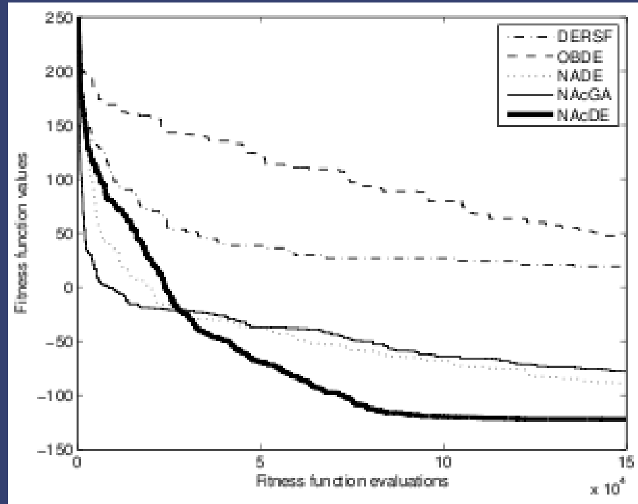


Figure 6. f_9 for $n = 30$ and 10% of noise

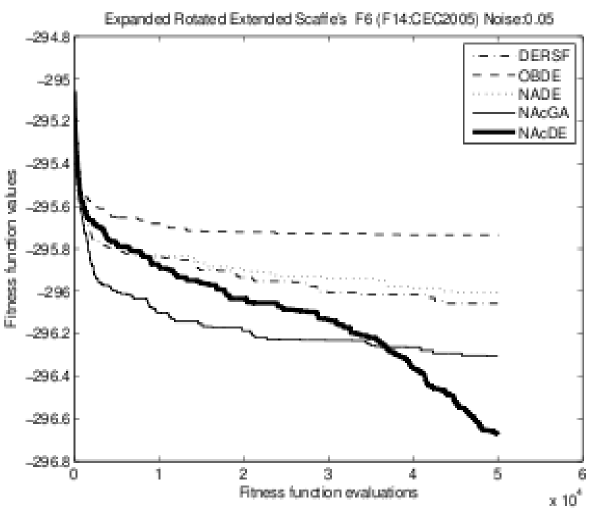


Figure 5. f_{14} for $n = 10$ and 5% of noise

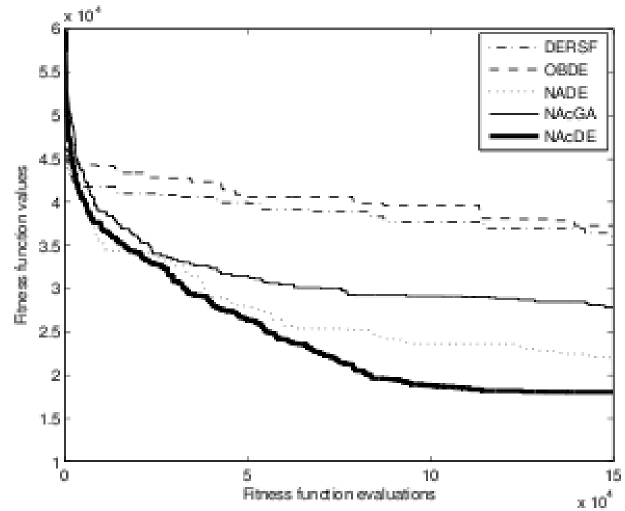
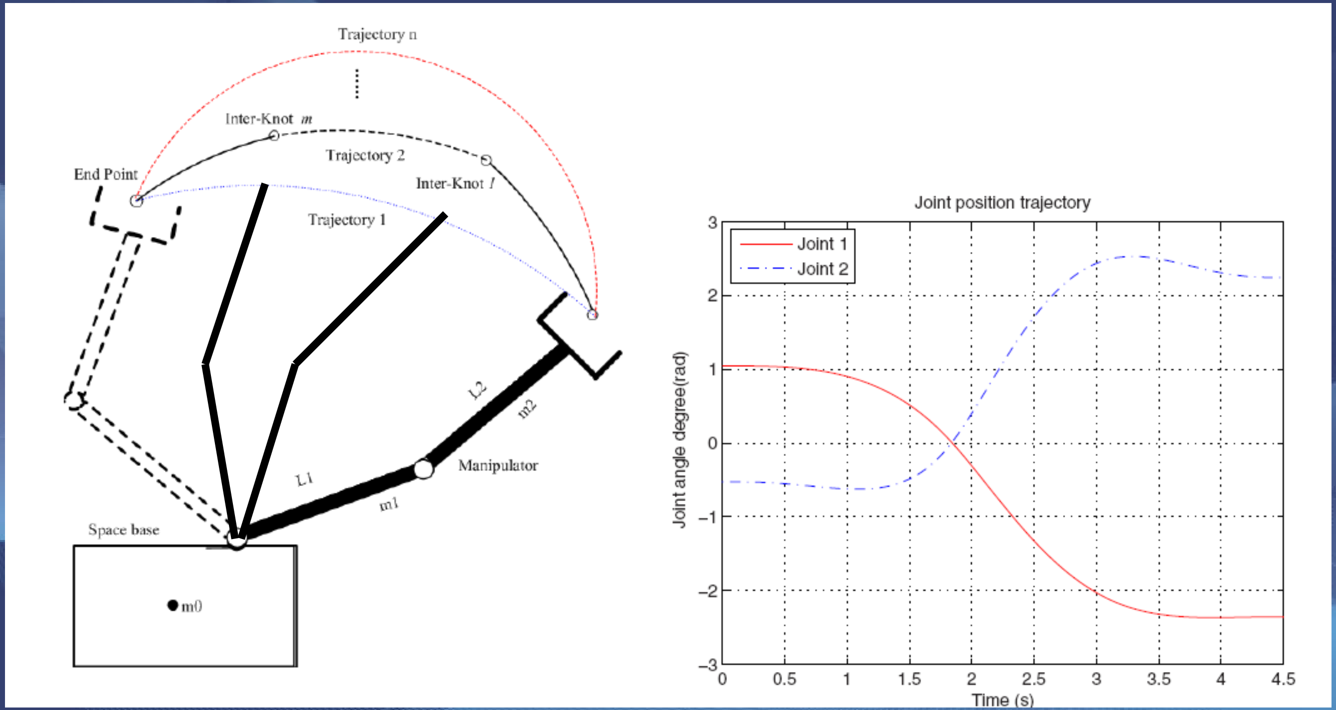


Figure 7. f_5 for $n = 30$ and 20% of noise

Case study: space robotic arm (1/3)

- Trajectory planning:
  - Given a specific task p(t), evaluate θ(t), θ'(t), θ''(t) so that:
    - the end effector follows the desired trajectory
    - the trajectory is smooth (without discontinuities)
  - Point-to-point problem: define inter-knot points and interpolate (linear interpolation, spline, etc.)
- Motion control: define torques to be applied, s.t. dynamic/kinematic constraints



Case study: space robotic arm (2/3)

Free-floating environment

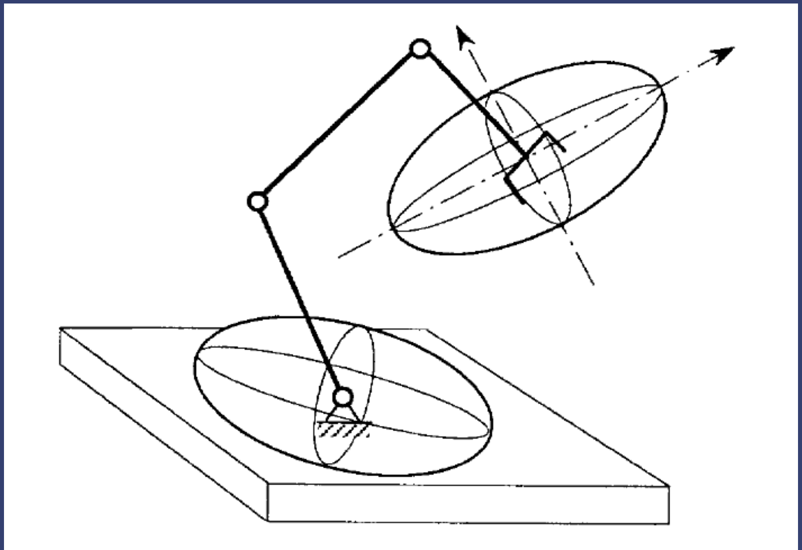
↓

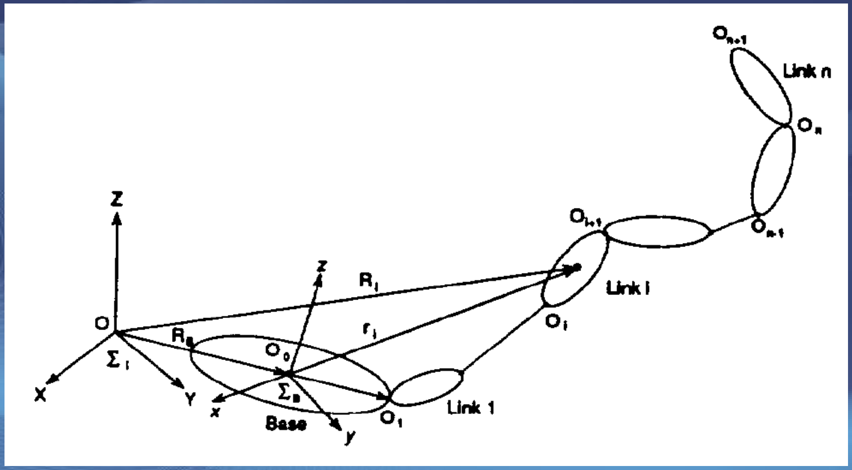
Mutual disturbance between base and end-effector:

$$F_B = \mathbf{N}^{-1}F_E \leftrightarrow F_E = \mathbf{N}F_B$$

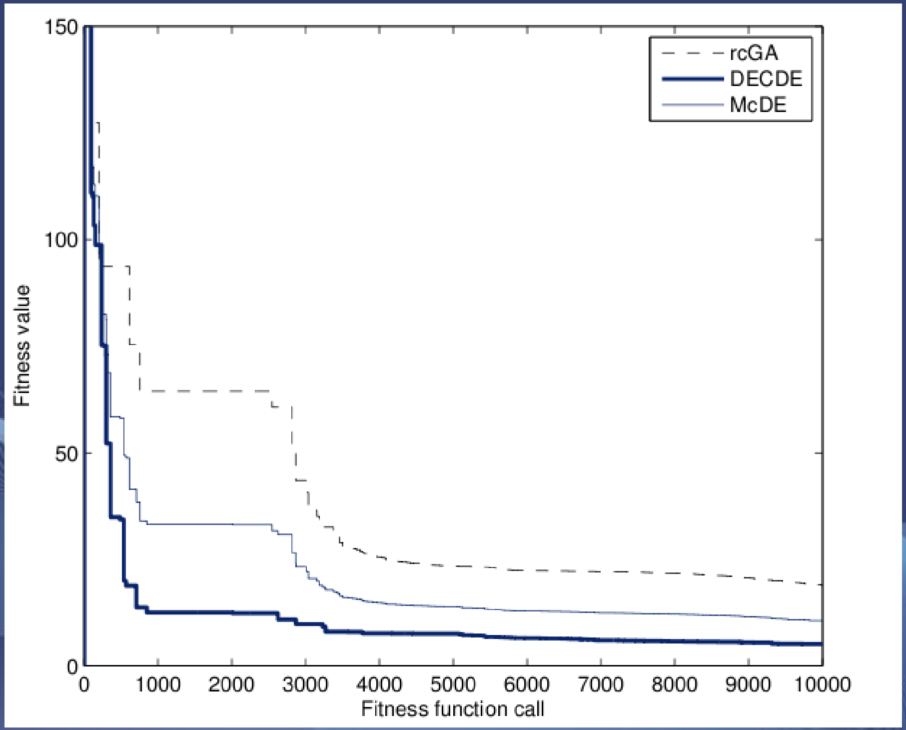
N (dynamic coupling matrix) *“is a function of the robot configuration* [<u>θ</u>]*, the geometric and inertia parameters of the robot and the spacecraft, and the position of the robot base with respect to the the spacecraft”* [29]

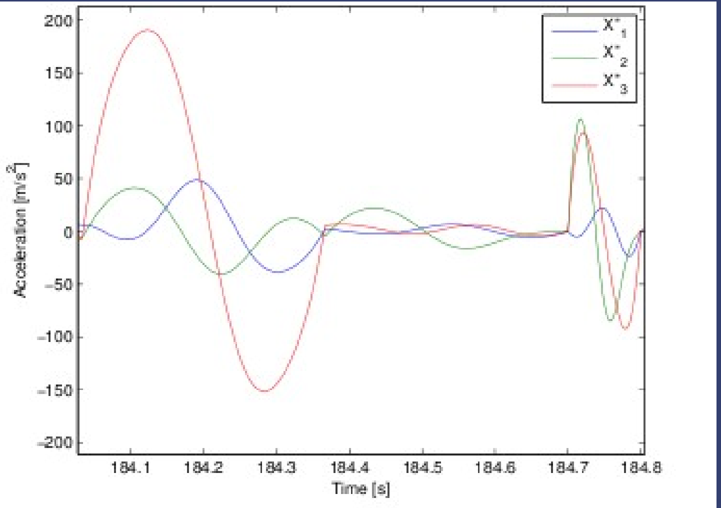
<u>The shorter the motion time is, the greater the disturbance to the base will be</u>



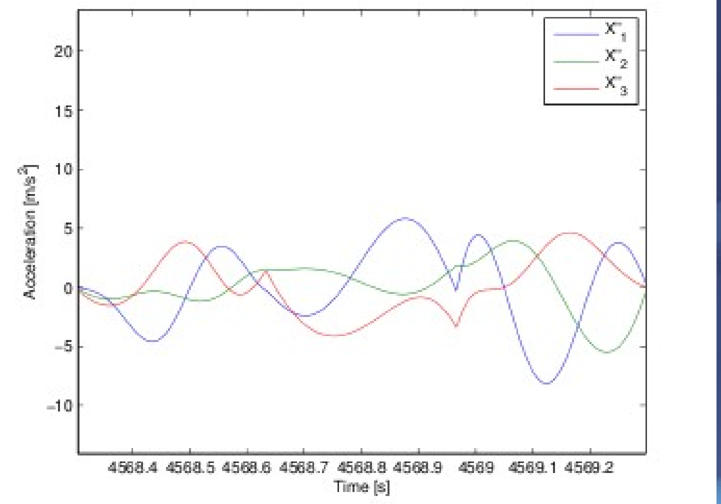


Case study: space robotic arm (3/3)





(a) without optimization (beginning of learning period)



(b) with optimization (end of learning period)

Conclusions and future works

- **Computational Intelligence Optimization is an active research field, with many different real word applications**
- **Compact Algorithms (CAs), especially cDE-based Algorithms, due to their characteristics of compactness and robustness, can become very popular in the future**
- **Future works: investigate other CAs (e.g. compact Memetic Algorithms - cMAs) and possible applications**
- **Second part of my PhD: proposing an advanced SW environment for designing novel optimization algorithms and integrating them with external models and software**

References

[1] F. Neri and V. Tirronen, “Recent advances in differential evolution: A review and experimental analysis,” Artificial Intelligence Review, vol. 33, no. 1, pp. 61–106, 2010.

[2] H.-Y. Fan and J. Lampinen, “A trigonometric mutation operation to differential evolution,” Journal of Global Optimization, vol. 27, no. 1, pp. 105–129, 2003.

[3] A. K. Qin, V. L. Huang, and P. N. Suganthan, “Differential evolution algorithm with strategy adaptation for global numerical optimization,” IEEE Transactions on Evolutionary Computation, vol. 13, pp. 398–417, 2009.

[4] S. Rahnamayan, H. R. Tizhoosh, and M. M. Salama, “Opposition-based differential evolution,” IEEE Transactions on Evolutionary Computation, vol. 12, no. 1, pp. 64–79, 2008.

[5] S. Das, A. Abraham, U. K. Chakraborty, and A. Konar, “Differential evolution with a neighborhood-based mutation operator,” IEEE Transactions on Evolutionary Computation, vol. 13, no. 3, pp. 526–553, 2009.

[6] J. Brest, B. Boˇskovi´c, S. Greiner, V. ˇZumer, and M. S. Mauˇcec, “Performance comparison of self-adaptive and adaptive differential evolution algorithms,” Soft Computing, vol. 11, no. 7, pp. 617–629, 2007.

[7] J. Brest, A. Zamuda, B. Boˇskovi´c, M. S. Maucec, and V. ˇZumer, “Highdimensional real-parameter optimization using self-adaptive differential evolution algorithm with population size reduction,” in Proceedings of the IEEE World Congress on Computational Intelligence, 2008, pp. 2032–2039.

[8] F. Neri and V. Tirronen, “Scale factor local search in differential evolution,” Memetic Computing Journal, vol. 1, no. 2, pp. 153–171, 2009.

[9] V. Tirronen, F. Neri, T. K¨arkk¨ainen, K. Majava, and T. Rossi, “An enhanced memetic differential evolution in filter design for defect detection in paper production,” Evolutionary Computation, vol. 16, pp. 529–555, 2008.

[10] A. Caponio, F. Neri, and V. Tirronen, “Super-fit control adaptation in memetic differential evolution frameworks,” Soft Computing-A Fusion of Foundations, Methodologies and Applications, vol. 13, no. 8, pp. 811–831, 2009.

[11] Y. Gao and Y.-J. Wang, “A memetic differential evolutionary algorithm for high dimensional functions’ optimization,” in Proceedings of the Third International Conference on Natural Computation, 2007, pp. 188–192.

[12] N. Noman and H. Iba, “Accelerating differential evolution using an adaptive local search,” IEEE Transactions on Evolutionary Computation, vol. 12, no. 1, pp. 107–125, 2008.

[13] D. K. Tasoulis, N. G. Pavlidis, V. P. Plagianakos, and M. N. Vrahatis, “Parallel differential evolution,” in Proceedings of the IEEE Congress on Evolutionary Computation, 2004, pp. 2023–2029.

[14] M. Weber, V. Tirronen, and F. Neri, “Scale factor inheritance mechanism in distributed differential evolution,” Soft Computin - A Fusion of Foundations, Methodologies and Applications, vol. 14, no. 11, pp. 1187–1207, 2010.

[15] M. Weber, F. Neri, and V. Tirronen, “Distributed differential evolution with explorative-exploitative population families,” Genetic Programming and Evolvable Machines, vol. 10, no. 4, pp. 343–371, 2009.

References

[16] S. Das, A. Konar, and U. K. Chakraborty, “Two improved differential evolution schemes for faster global search,” in Proceedings of the 2005 conference on Genetic and evolutionary computation. ACM, 2005, pp. 991–998.

[17] J. Brest, S. Greiner, B. Boˇskovi´c, M. Mernik, and V. ˇZumer, “Selfadapting control parameters in differential evolution: A comparative study on numerical benchmark problems,” IEEE Transactions on Evolutionary Computation, vol. 10, no. 6, pp. 646–657, 2006.

[18] J. Zhang and A. C. Sanderson, “Jade: Adaptive differential evolution with optional external archive,” IEEE Transactions on Evolutionary Computation, vol. 13, no. 5, pp. 945–958, 2009.

[19] E. Mininno, F. Neri, F. Cupertino, and D. Naso, “Compact differential evolution,” IEEE Transactions on Evolutionary Computation, 2011, to appear.

[20] E. Mininno, F. Cupertino, and D. Naso, “Real-valued compact genetic algorithms for embedded microcontroller optimization,” IEEE Transactions on Evolutionary Computation, vol. 12, no. 2, pp. 203–219, 2008.

[21] F. Neri and E. Mininno, “Memetic compact differential evolution for cartesian robot control,” IEEE Computational Intelligence Magazine, vol. 5, no. 2, pp. 54–65, 2010.

[22] W. Gautschi, “Error function and fresnel integrals,” in Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables, M. Abramowitz and I. A. Stegun, Eds., 1972, ch. 7, pp. 297–309.

[23] W. J. Cody, “Rational chebyshev approximations for the error function,” vol. 23, no. 107, pp. 631–637, 1969.

[24] K. V. Price, R. Storn, and J. Lampinen, Differential Evolution: A Practical Approach to Global Optimization. Springer, 2005.

[25] I. Rechemberg, Evolutionstrategie: Optimierung Technisher Systeme nach prinzipien des Biologishen Evolution. Fromman-Hozlboog Verlag, 1973.

[26] K. Ohkura, Y. Matsumura, and K. Ueda, “Robust evolution strategies,” Applied Intelligence, vol. 15, no. 3, pp. 153–169, 2001.

[27] P. N. Suganthan, N. Hansen, J. J. Liang, K. Deb, Y.-P. Chen, A. Auger, and S. Tiwari, “Problem definitions and evaluation criteria for the CEC 2005 special session on real-parameter optimization,” Nanyang Technological University and KanGAL, Singapore and IIT Kanpur, India, Tech. Rep. 2005005, 2005.

[28] F. Wilcoxon, “Individual comparisons by ranking methods,” Biometrics Bulletin, vol. 1, no. 6, pp. 80–83, 1945.

[29] Y. Xu, “The Measure of Dynamic Coupling of Space Robot Systems”, Proceedings of the IEEE International. Conference on Robotics and Automation, 1993