Multi-Population Adaptive Inflationary Differential Evolution

Marilena Di Carlo, Massimiliano Vasile, Edmondo Minisci

Department of Mechanical and Aerospace Engineering
University of Strathclyde

marilena.di-carlo@strath.ac.uk

PPSN BIOMA - Bioinspired Optimization Methods and their Applications
Ljubljana, 13 September 2014

Introduction

- Differential Evolution (DE) is a very efficient population-based stochastic algorithm for global numerical optimization problems

Introduction

- Differential Evolution (DE) is a very efficient population-based stochastic algorithm for global numerical optimization problems
- Its performance can be enhanced by combining it with others optimizer:

Introduction

- Differential Evolution (DE) is a very efficient population-based stochastic algorithm for global numerical optimization problems
- Its performance can be enhanced by combining it with others optimizer:

  Inflationary Differential Evolution Algorithm, IDEA

Introduction

- Differential Evolution (DE) is a very efficient population-based stochastic algorithm for global numerical optimization problems
- Its performance can be enhanced by combining it with others optimizer:

  Inflationary Differential Evolution Algorithm, IDEA
- DE performance are strongly influenced by setting of the algorithm parameter:

Introduction

- Differential Evolution (DE) is a very efficient population-based stochastic algorithm for global numerical optimization problems
- Its performance can be enhanced by combining it with others optimizer:

  Inflationary Differential Evolution Algorithm, IDEA
- DE performance are strongly influenced by setting of the algorithm parameter:

  Adaptive Inflationary Differential Evolution Algorithm, AIDEA

Introduction

- Differential Evolution (DE) is a very efficient population-based stochastic algorithm for global numerical optimization problems
- Its performance can be enhanced by combining it with others optimizer:

  Inflationary Differential Evolution Algorithm, IDEA
- DE performance are strongly influenced by setting of the algorithm parameter:

  Adaptive Inflationary Differential Evolution Algorithm, AIDEA
- **Multi-population version of AIDEA (MP-AIDEA)**

Contents

Contents

- Differential Evolution

Contents

- Differential Evolution
- IDEA

Contents

- Differential Evolution
- IDEA
- AIDEA

Contents

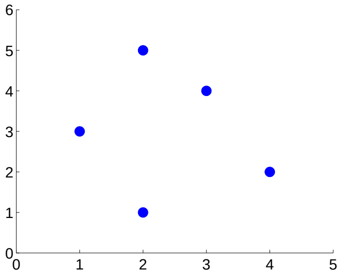
- Differential Evolution
- IDEA
- AIDEA
- Multi-Population AIDEA

Contents

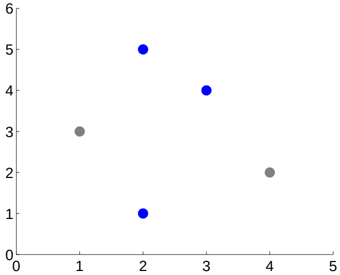
- Differential Evolution
- IDEA
- AIDEA
- Multi-Population AIDEA
- Test Results

Differential Evolution

- Initialize population in the search space

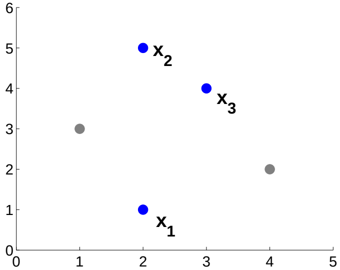


Differential Evolution



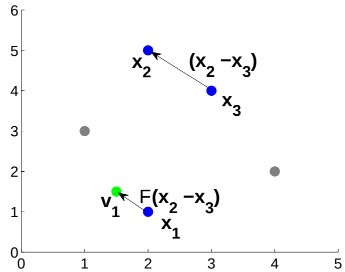
- Initialize population in the search space
- Select three individuals $\mathbf{x}_1$, $\mathbf{x}_2$ and $\mathbf{x}_3$

Differential Evolution



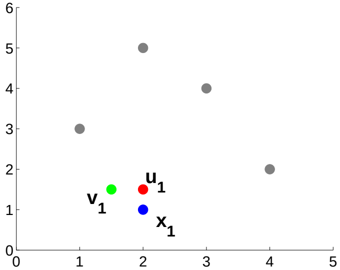
- Initialize population in the search space
- Select three individuals $\mathbf{x}_1$, $\mathbf{x}_2$ and $\mathbf{x}_3$

Differential Evolution



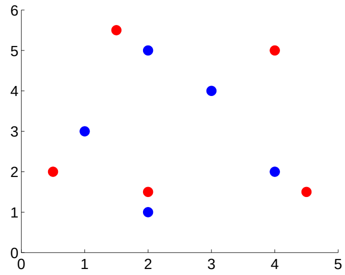
- Initialize population in the search space
- Select three individuals $\mathbf{x}_1$, $\mathbf{x}_2$ and $\mathbf{x}_3$
- Apply mutation:
  $\mathbf{v}_1 = \mathbf{x}_1 + F \cdot (\mathbf{x}_2 - \mathbf{x}_3)$

Differential Evolution



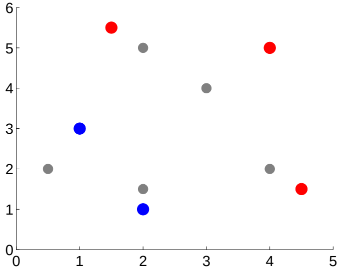
- Initialize population in the search space
- Select three individuals $\mathbf{x}_1$, $\mathbf{x}_2$ and $\mathbf{x}_3$
- Apply mutation:
  $\mathbf{v}_1 = \mathbf{x}_1 + F \cdot (\mathbf{x}_2 - \mathbf{x}_3)$
- Apply crossover to obtain trial vector $\mathbf{u}_1$:
  $$u_1^j = \begin{cases} v_1^j, & \text{if } \text{rand}(0,1) \leq CR \text{ or } j = j_{rand} \\ x_1^j, & \text{otherwise} \end{cases}$$

Differential Evolution



- Initialize population in the search space
- Select three individuals $\mathbf{x}_1$, $\mathbf{x}_2$ and $\mathbf{x}_3$
- Apply mutation:
  $\mathbf{v}_1 = \mathbf{x}_1 + F \cdot (\mathbf{x}_2 - \mathbf{x}_3)$
- Apply crossover to obtain trial vector $\mathbf{u}_1$:
  $$u_1^j = \begin{cases} v_1^j, & \text{if } \text{rand}(0,1) \leq CR \text{ or } j = j_{rand} \\ x_1^j, & \text{otherwise} \end{cases}$$
- Repeat operation for all the individuals

Differential Evolution



- Initialize population in the search space
- Select three individuals $\mathbf{x}_1$, $\mathbf{x}_2$ and $\mathbf{x}_3$
- Apply mutation:
  $\mathbf{v}_1 = \mathbf{x}_1 + F \cdot (\mathbf{x}_2 - \mathbf{x}_3)$
- Apply crossover to obtain trial vector $\mathbf{u}_1$:
  $$u_1^j = \begin{cases} v_1^j, & \text{if } \text{rand}(0,1) \leq CR \text{ or } j = j_{rand} \\ x_1^j, & \text{otherwise} \end{cases}$$
- Repeat operation for all the individuals
- Survival selection:
  $$\mathbf{x}_i' = \begin{cases} \mathbf{u}_i, & \text{if } f(\mathbf{u}_i) \leq f(\mathbf{x}_i) \\ \mathbf{x}_i, & \text{otherwise} \end{cases}$$

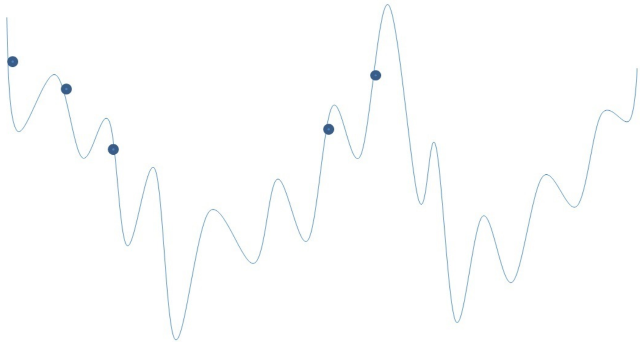
IDEA

- DE drawbacks:
  - Stagnation of the optimization process
  - CR and F difficult to tune and heavily problem dependent

IDEA

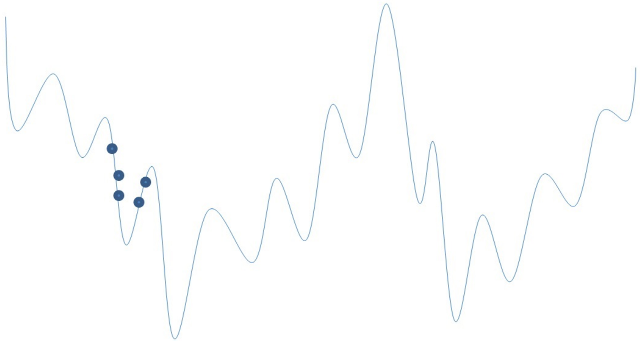
- DE drawbacks:
  - Stagnation of the optimization process
  - CR and F difficult to tune and heavily problem dependent
- IDEA (Inflationary Differential Evolution Algorithm)
  M. Vasile, E. Minisci, M. Locatelli, 2011
  - Hybridization of DE with the restarting procedure of Monotonic Basin Hopping (MBH) algorithm

IDEA



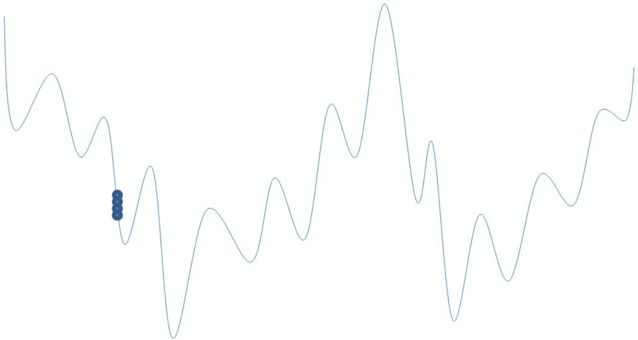
1. Initialize population in the search space and run Differential Evolution

IDEA



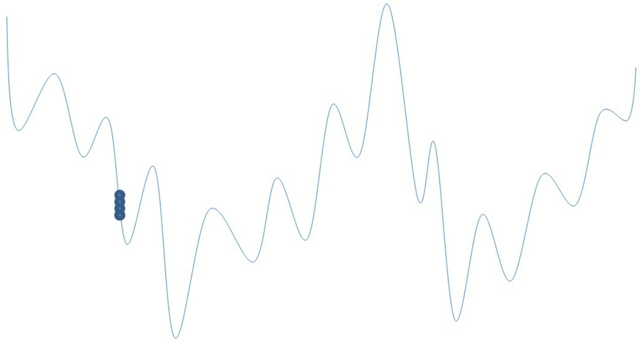
1. Initialize population in the search space and run Differential Evolution

IDEA



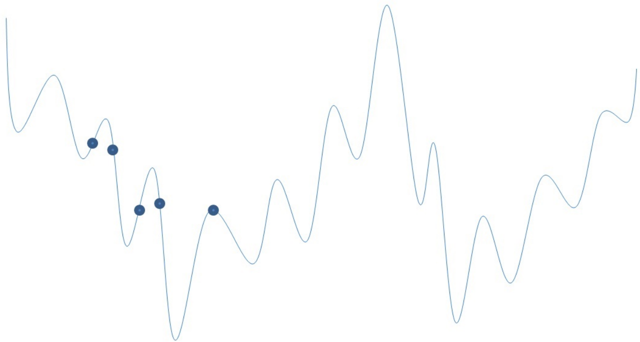
2. Population contraction: $r \leq \rho \cdot r_{max}$
   $r = \max\left(\|\mathbf{x}_i - \mathbf{x}_j\|\right)$ and r_{max} is the maximum value of r recorded during the convergence

IDEA



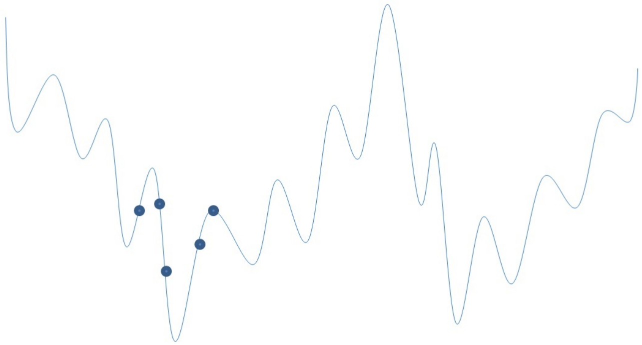
3. Perform **local search** from the best individual in the population and locate local minimum

IDEA



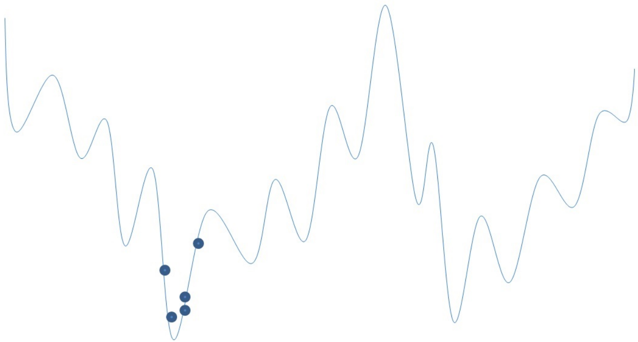
4. **Restart population in a bubble** of dimension δ_{local} around the local minimum

IDEA



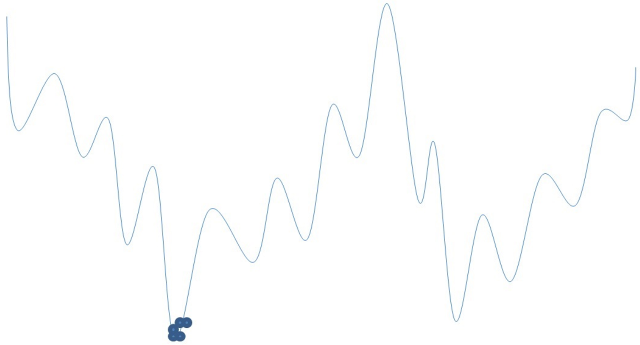
5. Repeat DE until convergence

IDEA



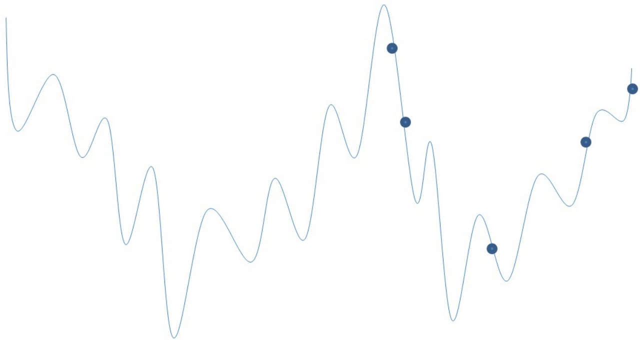
5. Repeat DE until convergence

IDEA



5. Repeat DE until convergence

IDEA



6. When more than n_{LR} local restarts have been performed **globally restart the population** at a distance δ_{global} from the centers of the clusters of local minima

AIDEA

- DE drawbacks:
  - Stagnation of the optimization process
  - CR and F difficult to tune and heavily problem dependent

AIDEA

- DE drawbacks:
  - Stagnation of the optimization process
  - CR and F difficult to tune and heavily problem dependent
- AIDEA (Adaptive Inflationary Differential Evolution Algorithm)
  E. Minisci, M. Vasile, 2014
  - Hybridization of DE with the restaring procedure of Monotonic Basin Hopping (MBH) algorithm
  - Adaptation of the DE parameters *CR* and *F*

AIDEA

CR and F adaptation

- Initialization of CRF, regular mesh in the space:

  $CR \in [0.1, 0.99]$
  $F \in [-0.5, 1]$

AIDEA

CR and *F* adaptation

- Initialization of *CRF*, regular mesh in the space:
  $CR \in [0.1, 0.99]$
  $F \in [-0.5, 1]$
- Sample of *CR* and *F* values from the Parzen distribution associated to *CRF* and association of each (CR, F) couple to an individual of the population

AIDEA

CR and *F* adaptation

- Initialization of *CRF*, regular mesh in the space:
  $CR \in [0.1, 0.99]$
  $F \in [-0.5, 1]$
- Sample of *CR* and *F* values from the Parzen distribution associated to *CRF* and association of each (CR, F) couple to an individual of the population
- Computation for each individual $\mathbf{x}$ and its child $\mathbf{x}'$ of the difference:

$$d = f(\mathbf{x}') - f(\mathbf{x})$$

AIDEA

CR and *F* adaptation

- Initialization of *CRF*, regular mesh in the space:

  $CR \in [0.1, 0.99]$
  $F \in [-0.5, 1]$

- Sample of *CR* and *F* values from the Parzen distribution associated to *CRF* and association of each (CR, F) couple to an individual of the population
- Computation for each individual $\mathbf{x}$ and its child $\mathbf{x}'$ of the difference:

  $$d = f(\mathbf{x}') - f(\mathbf{x})$$

- *CRF* update: (CR, F) couples with lower d are substituted by (CR, F) couples with higher d

MP-AIDEA

MP-AIDEA

- Adaptation of other parameters: multi-population AIDEA

MP-AIDEA

- Adaptation of other parameters: multi-population AIDEA
- Adaptation of the dimension of the bubble for the local restart δ_{local}

MP-AIDEA

- Adaptation of other parameters: multi-population AIDEA
- Adaptation of the dimension of the bubble for the local restart δ_{local}
- Strategies for the generation of the mutant vector:
  - DE/best-DE/rand
  - DE/arch-DE/rand
  - DE/arch-DE/best

MP-AIDEA

- Adaptation of other parameters: multi-population AIDEA
- Adaptation of the dimension of the bubble for the local restart δ_{local}
- Strategies for the generation of the mutant vector:
  - DE/best-DE/rand
  - DE/arch-DE/rand
  - DE/arch-DE/best
- *CR* and *F* adaptation:
  - MP-AIDEA-CRF1
    Same *CR* and *F* values for every individual of the same population
  - MP-AIDEA-CRF2
    Different *CR* and *F* values for each element of each population

MP-AIDEA

δ_{local} adaptation

- Computation of the minimum and maximum distance between all local minima, d_{minMIN} and d_{minMAX}

MP-AIDEA

δ_{local} adaptation

- Computation of the minimum and maximum distance between all local minima, d_{minMIN} and d_{minMAX}
- Creation of a regular mesh B in the space $[d_{minMIN}, d_{minMAX}]$

MP-AIDEA

δ_{local} adaptation

- Computation of the minimum and maximum distance between all local minima, d_{minMIN} and d_{minMAX}
- Creation of a regular mesh B in the space $[d_{minMIN}, d_{minMAX}]$
- Sample of δ_{local} from the Parzen distribution associated to B for each population

MP-AIDEA

δ_{local} adaptation

- Computation of the minimum and maximum distance between all local minima, d_{minMIN} and d_{minMAX}
- Creation of a regular mesh B in the space $[d_{minMIN}, d_{minMAX}]$
- Sample of δ_{local} from the Parzen distribution associated to B for each population
- Computation for each population of the distance between consecutive local minima:
  $p = \left\| \mathbf{x}_{min}^{k+1} - \mathbf{x}_{min}^{k} \right\|$

MP-AIDEA

δ_{local} adaptation

- Computation of the minimum and maximum distance between all local minima, d_{minMIN} and d_{minMAX}
- Creation of a regular mesh B in the space $[d_{minMIN}, d_{minMAX}]$
- Sample of δ_{local} from the Parzen distribution associated to B for each population
- Computation for each population of the distance between consecutive local minima: $p = \left\| \mathbf{x}_{min}^{k+1} - \mathbf{x}_{min}^{k} \right\|$
- Update of B: population with higher values of p are characterized by a better value of δ_{local}

MP-AIDEA versions

	CRF1	CRF2	DE-mut1	DE-mut2	DE-mut3	δ_{local}
MP-AIDEA 1	✓		✓			
MP-AIDEA 2	✓			✓		
MP-AIDEA 2	✓				✓	
MP-AIDEA 4	✓		✓			✓
MP-AIDEA 5	✓			✓		✓
MP-AIDEA 6	✓				✓	✓
MP-AIDEA 7		✓	✓			
MP-AIDEA 8		✓		✓		
MP-AIDEA 9		✓			✓	
MP-AIDEA 10		✓	✓			✓
MP-AIDEA 11		✓		✓		✓
MP-AIDEA 12		✓			✓	✓

DE-mut1: DE/best-DE/rand

DE-mut2: DE/arch-DE/rand

DE-mut3: DE/arch-DE/best

MP-AIDEA versions

	CRF1	CRF2	DE-mut1	DE-mut2	DE-mut3	δ_{local}
MP-AIDEA 1	✓		✓			
MP-AIDEA 2	✓			✓		
MP-AIDEA 2	✓				✓	
MP-AIDEA 4	✓		✓			✓
MP-AIDEA 5	✓			✓		✓
MP-AIDEA 6	✓				✓	✓
MP-AIDEA 7		✓	✓			
MP-AIDEA 8		✓		✓		
MP-AIDEA 9		✓			✓	
MP-AIDEA 10		✓	✓			✓
MP-AIDEA 11		✓		✓		✓
MP-AIDEA 12		✓			✓	✓

DE-mut1: DE/best-DE/rand

DE-mut2: DE/arch-DE/rand

DE-mut3: DE/arch-DE/best

Test Cases

Competition of the Congress on Evolutionary Computation (CEC)

Test Cases

Competition of the Congress on Evolutionary Computation (CEC)

1. Spread Spectrum Radar Polyphase Code Design, CEC 2011
2. Tersoff Radar Function Minimization Problem, CEC 2011
3. Schwefel's Problem, CEC 2005
4. Rotated Version of Hybrid Composition Function, CEC 2005

Test Cases

Competition of the Congress on Evolutionary Computation (CEC)

1. Spread Spectrum Radar Polyphase Code Design, CEC 2011
2. Tersoff Radar Function Minimization Problem, CEC 2011
3. Schwefel's Problem, CEC 2005
4. Rotated Version of Hybrid Composition Function, CEC 2005

Algorithm performance:

- Success rate: number of successful runs over 100 total runs
- Successful run: $f(\mathbf{x}_{min}) < f_{min} + \epsilon$

  CEC 2011: $\epsilon = 0.001$

  CEC 2005: $\epsilon = 0.01$

Spread Spectrum Radar Polyphase Code Design

Spread Spectrum Radar Polyphase Code Design

- Problem and algorithm parameters

D	f_{min}	FEs	δ_{local}*	δ_{global}	ρ	n_{LR}
20	0.5	$1.5 \cdot 10^5$	0.1	0.1	0.2	10

* for AIDEA and MP-AIDEA versions which do not adapt δ_{local}

Spread Spectrum Radar Polyphase Code Design

- Problem and algorithm parameters

D	f_{min}	FEs	δ_{local}*	δ_{global}	ρ	n_{LR}
20	0.5	$1.5 \cdot 10^5$	0.1	0.1	0.2	10

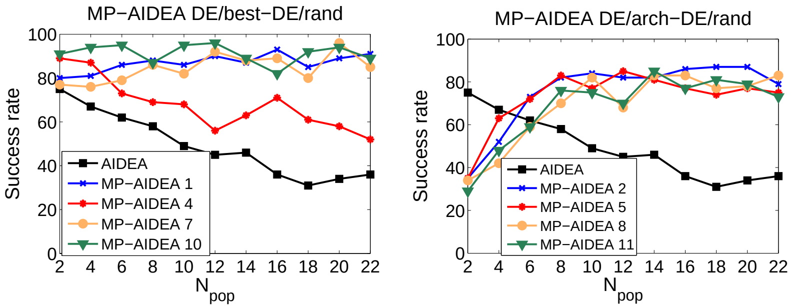
* for AIDEA and MP-AIDEA versions which do not adapt δ_{local}

- Algorithm comparison
  - AIDEA
  - Best performing algorithms of CEC 2011 competition:

    GA-MPC (Genetic Algorithm with Multi-Parent Crossover)
    WI-DE (Weed Inspired Differential Evolution)

Spread Spectrum Radar Polyphase Code Design

- MP-AIDEA and AIDEA success rate



Spread Spectrum Radar Polyphase Code Design

- Statistics of the results

Algorithm	Min	Mean	Max	Str.Dev.
MP-AIDEA 10	0.5000	0.5045	0.5690	0.0135
AIDEA	0.5000	0.5130	0.6422	0.0263
GA-MPC	0.5000	0.7484	0.9334	0.1249
WI-DE	0.5000	0.6560	0.9931	0.1160

Tersoff Potential Function Minimization Problem

Tersoff Potential Function Minimization Problem

- Problem and algorithm parameters

D	f_{min}	FEs
30	-36.9286	$1.5 \cdot 10^5$

	δ_{local}*	δ_{global}	ρ	n_{LR}
Case 1	**0.1**	0.1	0.2	10
Case 2	**0.3**	0.1	0.2	10

* for AIDEA and MP-AIDEA versions which do not adapt δ_{local}

Tersoff Potential Function Minimization Problem

- Problem and algorithm parameters

D	f_{min}	FEs
30	-36.9286	$1.5 \cdot 10^5$

	δ_{local}*	δ_{global}	ρ	n_{LR}
Case 1	**0.1**	0.1	0.2	10
Case 2	**0.3**	0.1	0.2	10

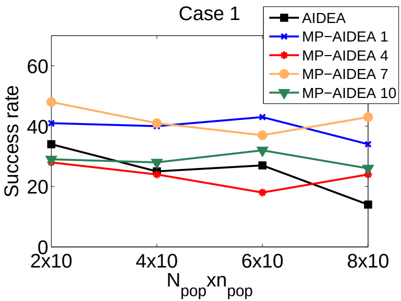
* for AIDEA and MP-AIDEA versions which do not adapt δ_{local}

- Algorithm comparison
  - AIDEA
  - Best performing algorithms of CEC 2011 competition:

    GA-MPC (Genetic Algorithm with Multi-Parent Crossover)
    WI-DE (Weed Inspired Differential Evolution)

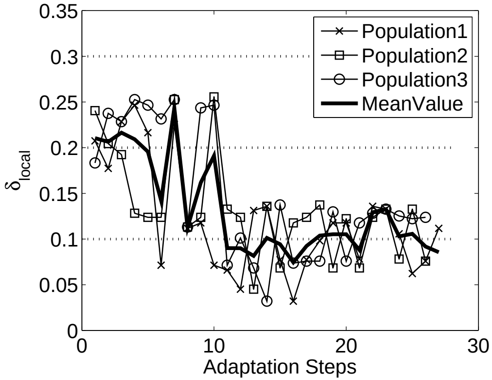
Tersoff Potential Function Minimization Problem

- MP-AIDEA and AIDEA success rate



Tersoff Potential Function Minimization Problem

- δ_{local} adaptation

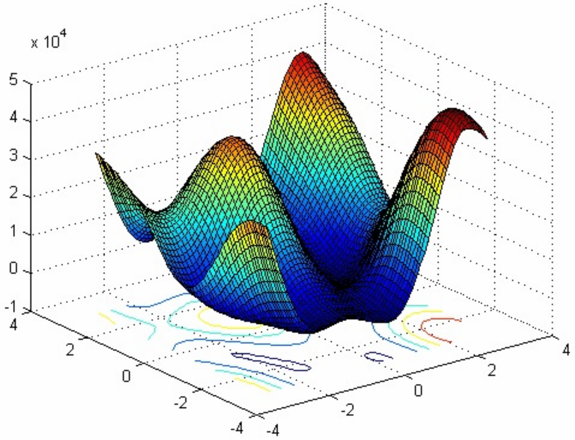


Tersoff Potential Function Minimization Problem

- Statistics of the results

Algorithm	Min	Mean	Max	Str.Dev.
Case 1				
MP-AIDEA 7	-36.9286	-36.7120	-34.3504	0.3835
AIDEA	-36.9286	-36.8046	-35.9700	0.2483
Case 2				
MP-AIDEA 10	-36.9286	-36.6689	-34.1647	0.4399
AIDEA	-36.9286	-36.6219	-35.4467	0.4694
GA-MPC	-36.8457	-35.03883	-34.1076	0.8329
WI-DE	-36.8	-35.6	-34.2	0.904

Schwefel's Problem



Schwefel's Problem

- Problem and algorithm parameters

D	f_{min}	FEs	δ_{local}*	δ_{global}	ρ	n_{LR}
30/50	-460	$3 \cdot 10^5$ / $5 \cdot 10^5$	0.1	0.1	0.2	5

* for AIDEA and MP-AIDEA versions which do not adapt δ_{local}

Schwefel's Problem

- Problem and algorithm parameters

D	f_{min}	FEs	δ_{local}*	δ_{global}	ρ	n_{LR}
30/50	-460	$3{\cdot}10^5$ / $5{\cdot}10^5$	0.1	0.1	0.2	5

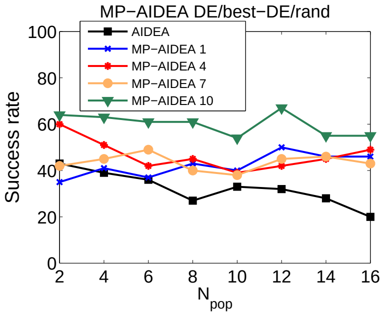
* for AIDEA and MP-AIDEA versions which do not adapt δ_{local}

- Algorithm comparison
  - AIDEA
  - Best performing algorithms of CEC 2005 competition:

    IPOP-CMA-ES (Increasing Population Size Covariance Matrix Adaptation Evolution Strategy)

Schwefel's Problem

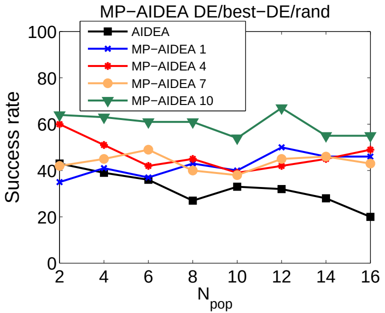
- MP-AIDEA and AIDEA success rate



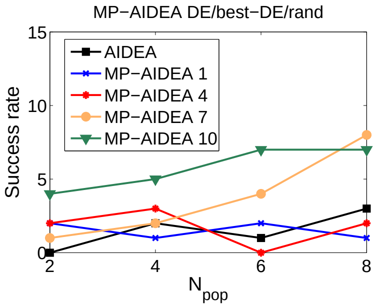
$$D = 30$$
$$n_{pop} = 10$$

Schwefel's Problem

- MP-AIDEA and AIDEA success rate



$D = 30$
$n_{pop} = 10$



$D = 50$
$n_{pop} = 20$

Schwefel's Problem

- Statistics of the results (error values w.r.t. f_{min})

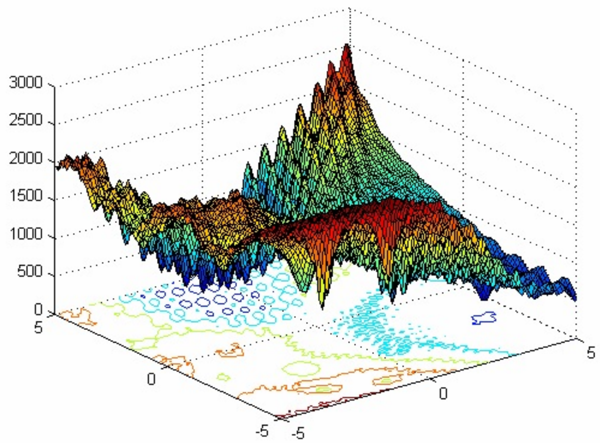
Algorithm	Min	Mean	Max	Str.Dev.
D = 30				
MP-AIDEA 10	1.39e-9	2.45e+1	4.77e+2	7.25e+1
AIDEA	2.01e-9	1.03e+2	1.00e+3	1.97e+2
IPOP-CMA-ES	3.79e-9	4.43e+4	1.10e+6	2.19e+5
D = 50				
MP-AIDEA 10	2.48e-8	8.91e+2	9.54e+3	1.27e+3
AIDEA	5.61e-8	2.22e+3	1.33e+4	2.69e+3
IPOP-CMA-ES	9.67e+0	2.27e+5	5.57e+6	1.11e+6

Schwefel's Problem

- Statistics of the results (error values w.r.t. f_{min})

Algorithm	Min	Mean	Max	Str.Dev.
D = 30				
MP-AIDEA 10	1.39e-9	2.45e+1	4.77e+2	7.25e+1
AIDEA	2.01e-9	1.03e+2	1.00e+3	1.97e+2
IPOP-CMA-ES	3.79e-9	4.43e+4	1.10e+6	2.19e+5
D = 50				
MP-AIDEA 10	**2.48e-8**	8.91e+2	9.54e+3	1.27e+3
AIDEA	**5.61e-8**	2.22e+3	1.33e+4	2.69e+3
IPOP-CMA-ES	**9.67e+0**	2.27e+5	5.57e+6	1.11e+6

Rotated Version of Hybrid Composition Function



Rotated Version of Hybrid Composition Function

- Problem and algorithm parameters

D	f_{min}	FEs	δ_{local}*	δ_{global}	ρ	n_{LR}
10	120	$1 \cdot 10^5$	0.1	0.1	0.2	5

* for AIDEA and MP-AIDEA versions which do not adapt δ_{local}

Rotated Version of Hybrid Composition Function

- Problem and algorithm parameters

D	f_{min}	FEs	δ_{local}*	δ_{global}	ρ	n_{LR}
10	120	$1 \cdot 10^5$	0.1	0.1	0.2	5

* for AIDEA and MP-AIDEA versions which do not adapt δ_{local}

- Algorithm comparison
  - AIDEA
  - Best performing algorithms of CEC 2005 competition:

    IPOP-CMA-ES (Increasing Population Size Covariance Matrix Adaptation Evolution Strategy)

Rotated Version of Hybrid Composition Function

- MP-AIDEA success rate

Algorithm	2x20	4x20	6x20	8x20
MP-AIDEA 4	**2**	**1**	**1**	0
MP-AIDEA 10	**1**	0	**1**	**3**

Rotated Version of Hybrid Composition Function

- MP-AIDEA success rate

Algorithm	2x20	4x20	6x20	8x20
MP-AIDEA 4	**2**	**1**	**1**	0
MP-AIDEA 10	**1**	0	**1**	**3**

- Statistics of the results (error values w.r.t. f_{min})

Algorithm	Min	Mean	Max	Str.Dev.
MP-AIDEA 10	**7.44e-11**	1.05e+2	1.32e+2	2.35e+1
AIDEA	**5.38e+1**	1.02e+2	1.14e+2	8.42e+0
IPOP-CMA-ES	**7.92e+1**	9.13e+1	9.68e+1	3.49e+0

Conclusions

Conclusions

- Multi-population version of AIDEA

Conclusions

- Multi-population version of AIDEA
  - Hybridization of DE and MBH
  - Adaptation of CR, F and δ_{local}

Conclusions

- Multi-population version of AIDEA
  - Hybridization of DE and MBH
  - Adaptation of CR, F and δ_{local}
- Test results:

Conclusions

- Multi-population version of AIDEA
  - Hybridization of DE and MBH
  - Adaptation of CR, F and δ_{local}
- Test results:
  - MP-AIDEA with adaptation of δ_{local} give good results and do not require the setting of this parameter

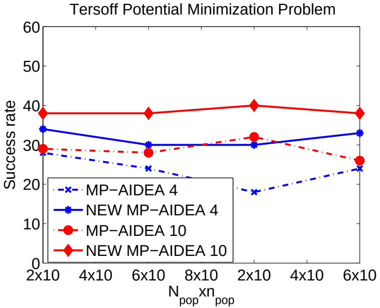
Conclusions

- Multi-population version of AIDEA
  - Hybridization of DE and MBH
  - Adaptation of CR, F and δ_{local}
- Test results:
  - MP-AIDEA with adaptation of δ_{local} give good results and do not require the setting of this parameter
  - **Most successful versions of MP-AIDEA were able to locate for the first time the global minima of two difficult academic test functions**

Future work

Future work

- Adaptation of other parameters:
  maximum number of local restart n_{LR}



Thank you for your attention