Test and real-world optimisation problems

Petr Bujok

University of Ostrava
Faculty of Science
Department of Informatics and Computers
Czech Republic

petr.bujok@osu.cz

Seminar of machine learning and modelling

Outline

Introduction: topics of my research

- Evolutionary Algorithms: Adaptation of parameters
- Evolutionary Algorithms: Cooperation (parallel) models
- Applied statistics methods

Global optimisation problem

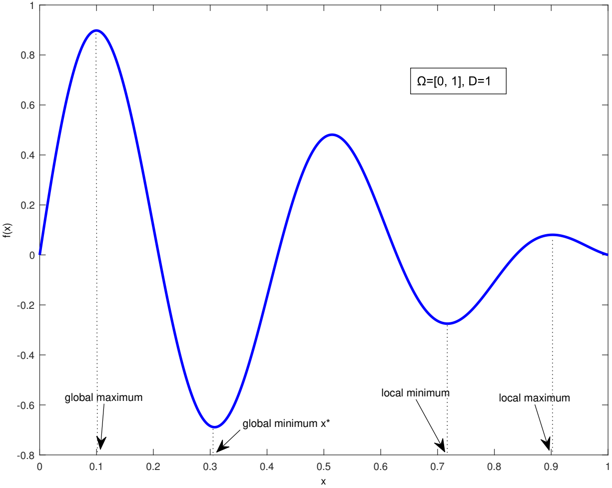
- objective function $f: \Omega \to \mathbb{R}, \ \Omega \subseteq \mathbb{R}^D$, D is the dimension of the task

- global minimum is a point $\boldsymbol{x}^*$, which satisfies:
  $\forall \boldsymbol{x} \in \Omega$: $f(\boldsymbol{x}^*) \leq f(\boldsymbol{x})$

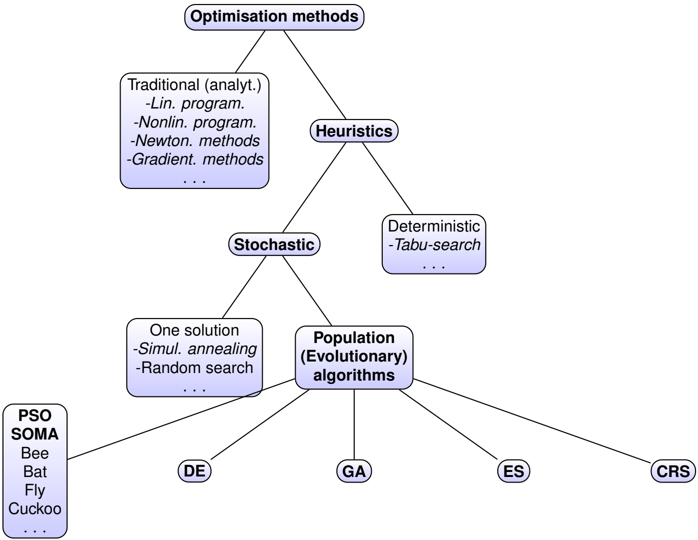
- search space is usually bounded Ω:
  $\Omega = [a_1, b_1] \times [a_2, b_2] \times \ldots \times [a_D, b_D], \ a_j < b_j, \ j = 1, 2, \ldots, D$

- goal of an optimisation process is to search for a solution $\boldsymbol{x}^*$

Global and local extrema



Methods used in optimisation



Variants of objective functions

- differentiable objective function
- separable objective function
- multimodal objective function
- continuous objective function
- constrained objective function

Bound-constrained continuous test functions

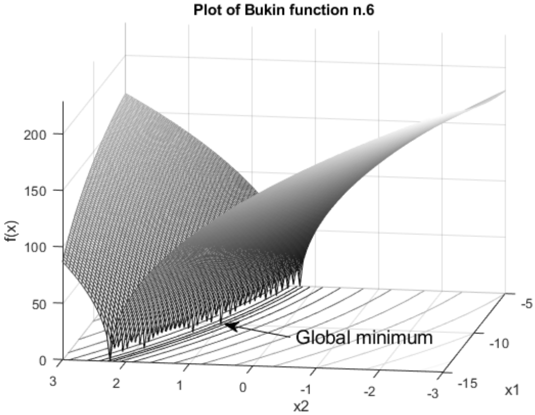
- test functions enable to **evaluate** and **compare** various optimisation methods
- the goal is also to indicate **poor** optimisation methods
- the difficulty of test problems is given by their **complexity**, **dimensionality** and unbounded search space
- dimensionality of the **scalable** problems is not restricted (typically from $D = 1$ to $D = 1000$)
- **known** true **solutions** of the test problems enable to evaluate the **success** or **reliability** of methods

Example: function Bukin n.6

Bukin function n.6 - multimodal, non-separable, non-scalable:

$f(\boldsymbol{x}) = 100\sqrt{|x_2 - 0.01x_1^2|} + 0.01|x_1 + 10|$,

$x_1 \in [-15, -5]$, $x_2 \in [-3, 3]$.



Test functions from competitions

- there are many sets of functions for **competitions**:
- **bound-constrained**, **unconstrained**, **feasible conditioned** (constrained), **multi-objective**, etc.
- some sets are **overlapped**
- beside '**artificial**' test functions, there is a set of 22 selected **real-world problems** (CEC 2011)
- true solutions of CEC 2011 functions (**minimisation**) are **not known**
- Q: '*Which set is the best to evaluate new methods?*'

Congress on evolutionary competitions (CEC)

CEC'05 - Real parameter single objective
CEC'06 - Constrained real parameter single objective
CEC'07 - Real-parameter MOEAs
CEC'08 - Large scale single objective with bound constraints
CEC'09 - Dynamic optimisation (composition functions)
CEC'09 - Real-parameter MOEAs
CEC'10 - Large-scale single objective with bound constraints
CEC'10 - Constrained real parameter single objective
CEC'10 - Niching scalable test problems
CEC'11 - Real-world numerical problems
CEC'13 - Real parameter single objective
CEC'14 - Real parameter single objective (2 scenarios)
CEC'14 - Dynamic MOEA benchmark problems
CEC'15 - Real parameter single objective (3 scenarios)
CEC'16 - Real parameter single objective (4 scenarios)
CEC'17 - Real parameter single objective (3 scenarios)
CEC'18 - Real parameter single objective (3 scenarios)
CEC'19 - 100-digit challenge on single objective

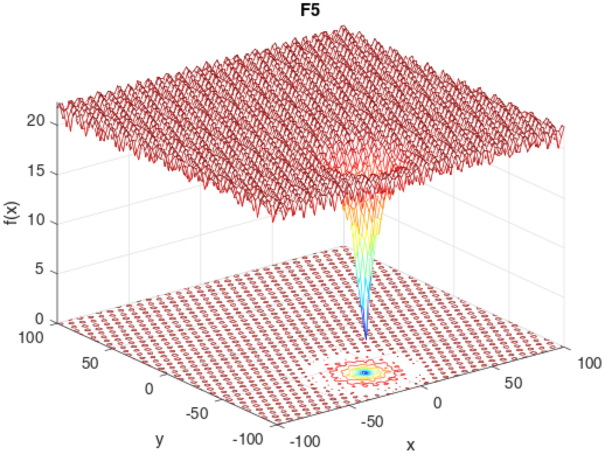
Black-box optimisation competition: *BBComp*

benchmark	D	problems	maxFES
BBComp2015: CEC	2 - 64	1000	100 D^2
GECCO	2 - 64	1000	10 D - 100 D
BBComp2016: 1OBJ	2 - 64	1000	100 D^2
1OBJ-expensive	2 - 64	1000	10 D - 100 D
2OBJ	2 - 64	1000	1000 D
2OBJ-expensive	2 - 64	1000	10 D - 100 D
3OBJ	2 - 64	1000	1000 D
BBComp2017: 1OBJ	2 - 64	1000	100 D^2
1OBJ-expensive	2 - 64	1000	10 D - 100 D
2OBJ	2 - 64	1000	1000 D
2OBJ-expensive	2 - 64	1000	10 D - 100 D
3OBJ	2 - 64	1000	1000 D
BBComp2018: 1OBJ	2 - 64	1000	100 D^2
1OBJ-expensive	2 - 64	1000	10 D - 100 D
2OBJ	2 - 64	1000	1000 D
2OBJ-expensive	2 - 64	1000	10 D - 100 D
3OBJ	2 - 64	1000	1000 D
BBComp2019: 1OBJ	2 - 64	1000	100 D^2
1OBJ-expensive	2 - 64	1000	10 D - 100 D
2OBJ	2 - 64	1000	1000 D
2OBJ-expensive	2 - 64	1000	10 D - 100 D
3OBJ	2 - 64	1000	1000 D

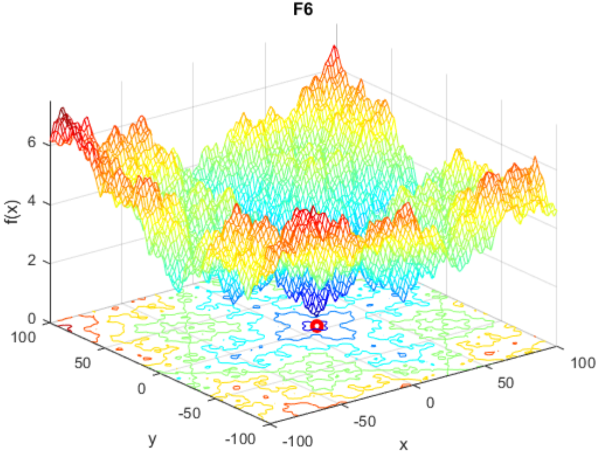
Example: CEC 2014 test problems

- 30 functions of four kinds of difficulty:
  - **unimodal** - simple functions with **one** extremum
  - **multimodal** - functions contain **many** extrema
  - **hybrid** - functions approximate of **real** problems
  - **composed** - **complex** functions composed of several functions
- the search space is **bounded** (box-constrained), $\Omega = [-100,\ 100]^D$
- the functions are **scalable** for $D = 2,\ 10,\ 30,\ 50,\ 100$ (restricted by rotation matrices)
- the search process is **limited**: $FES \leq 10000 \times D$
- the **accuracy** of found solution is evaluated by $error = f_{\min} - f^*$, $error < 1 \times 10^{-8}$ is a **good** solution

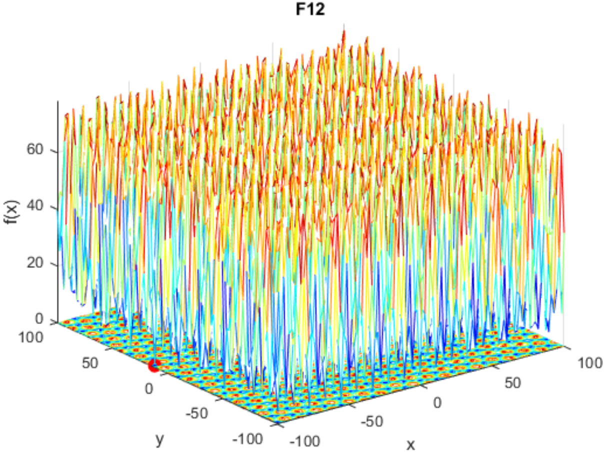
shifted and rotated Ackley function - multimodal, non-separable:



shifted and rotated Weierstrass function - multimodal, non-separable, differentiable only on a set of points:

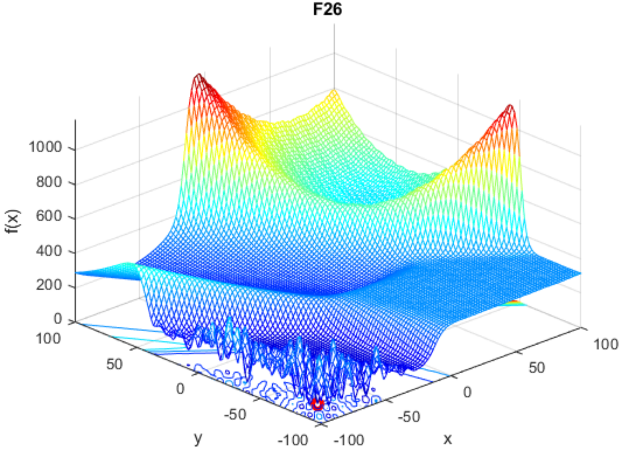


shifted and rotated Katsuura function - multimodal, non-separable, non-differentiable:



CEC 2014 composition test problem

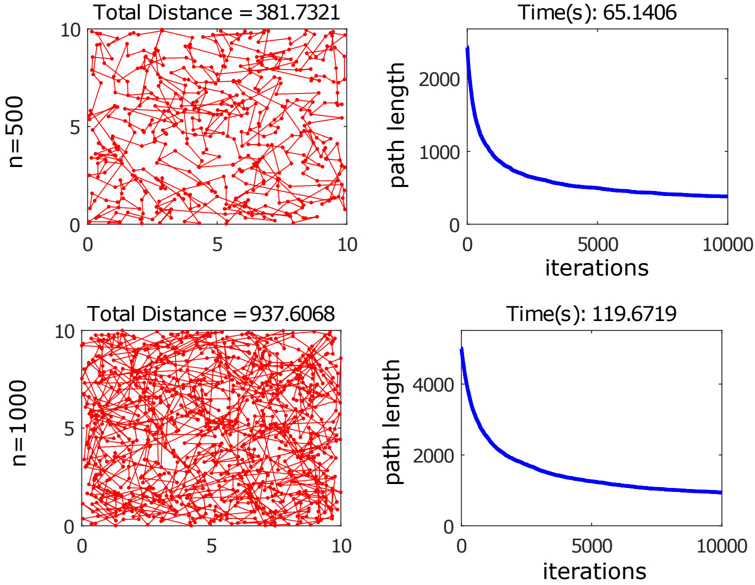
composed of five functions - multimodal, non-separable, asymmetrical:



Discrete (combinatorial) problems

- search space Ω is not continuous
- values of variables ($\boldsymbol{x}$) are from a **finite set**
- popular combinatorial problems are:
  - searching of the path in a graph
  - travelling salesman problem
  - time scheduling
  - knapsack problem, etc.

TSP problem solved by GA, $n = 500$ and $n = 1000$



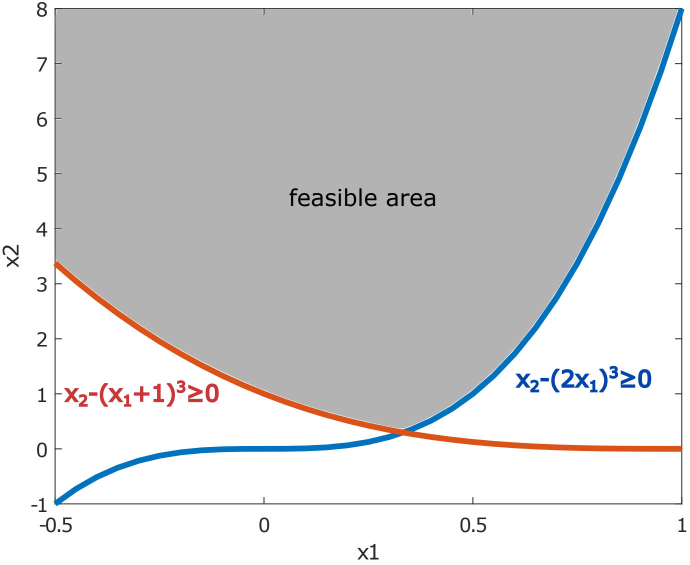
Constrained (continuous) optimisation problems

- objective function $f : \Omega \to \mathbb{R}^D$ is constrained into **feasible** area(s):
  a) $g_i(\boldsymbol{x}) \leq 0, \; i = 1, \; 2, \ldots p$
  b) $h_j(\boldsymbol{x}) = 0, \; j = p + 1, \; p + 2, \ldots m$
- global minimum (maximum) of the objective function, $\boldsymbol{x}^*$, is located in a **feasible** area defined by **constraints**
- possible criterion is average **violation of constraints**:

$$\bar{v} = \frac{\sum_{i=1}^{p} G_i(\boldsymbol{x}) + \sum_{j=p+1}^{m} H_j(\boldsymbol{x})}{m}$$

- $G_i(\boldsymbol{x}) = g_i(\boldsymbol{x}), \quad \text{for} \quad g_i(\boldsymbol{x}) > 0, \quad \text{otherwise} \quad G_i(\boldsymbol{x}) = 0$
- $H_j(\boldsymbol{x}) = |h_j(\boldsymbol{x})|, \quad \text{for} \quad |h_j(\boldsymbol{x})| - \varepsilon > 0,$ $\text{otherwise} \quad H_j(\boldsymbol{x}) = 0$

Constrained optimisation problems: example



Constrained optimisation problems

Possible ways to solve constraints problems:

- only information about **acceptance of the solution** is provided (located in feasible area)
- **combination** of the objective function value f and penalty criterion $\bar{v}$ (multi-objective approach)
- **penalty** criterion $\bar{v}$ and **objective function** value f are used **independently**
- **penalty** criterion $\bar{v}$ is used as an **objective** function

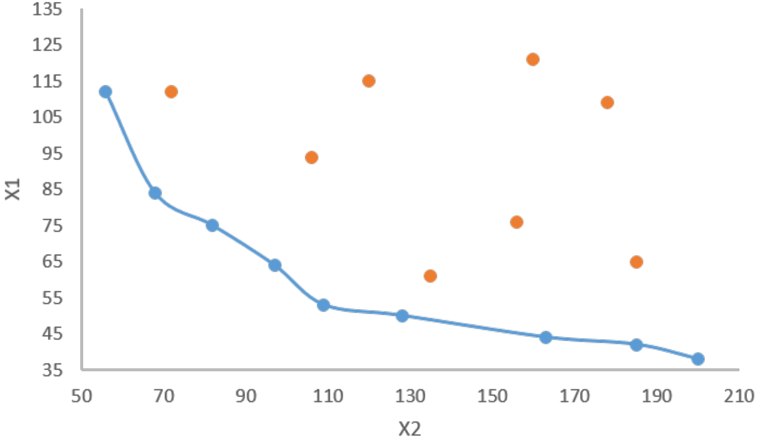
Multi-objective optimisation problems

- called '*multicriteria decision making*' or *Pareto front*
- **more than one** objective function that are to be minimized (or maximized)
- the solution is a **set of results** that define the best **compromise** between problem objectives
- for M objective functions:

$$f_m(\boldsymbol{x}), \quad m = 1,\ 2,\ \ldots,\ M$$

- Pareto front is defined by **non-dominated** (Pareto efficient) points

Pareto front: example



- the main goal of the development of optimisation methods is their application on a **real problem**
- each real problem is **represented** by an objective function (with restricted conditions)
- a set of 22 real-world problems of CEC 2011 competition enables to **identify a good optimisation method**
- the set includes:
  - 9 bound-constrained problems
  - 12 constrained problems (equality, inequality)
  - 1 unconstrained problem
- all problems are **minimisation**, true solution is **not known**

Estimation of parameters in nonlinear regression

- Tvrdík, J., Křivý, I., Mišík, L.: Adaptive population-based search: Application to estimation of nonlinear regression parameters, *Computational Statistics & Data Analysis* **52** (2007) 713-724
- additive nonlinear regression:

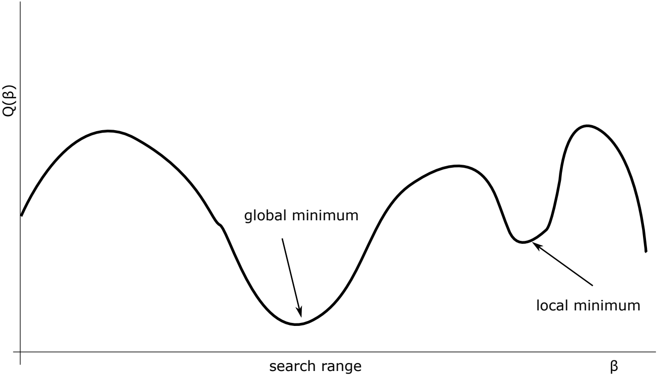
$$Y_i = g(\boldsymbol{x}_i,\ \boldsymbol{\beta}) + \varepsilon_i, \quad i = 1, . 2, \ldots, n$$

- estimation of parameters $\boldsymbol{\beta}$ by minimisation (**least squares**):

$$Q(\boldsymbol{\beta}) = \sum_{i=1}^{n} [Y_i - g(\boldsymbol{x}_i,\ \boldsymbol{\beta})]^2$$

- $Q(\boldsymbol{\beta})$ could be a **multimodal** optimisation problem

Estimation of parameters in nonlinear regression



Estimation of parameters in nonlinear regression

- 27 non-linear regression data sets (NIST)
- standard gradient-based methods are **compared** with proposed competitive CRS
- two proposed **competitive CRS** variants use four various strategies
- both CRS algorithms are **more reliable** compared with deterministic approach in all of 27 problems
- no CRS method needs tuning and they are **not dependent** on starting positions

- Tvrdík, J., Křivý, I.: Hybrid differential evolution algorithm for optimal clustering, *Applied Soft Computing* **35** (2015) 502-512
- **optimal partitioning** of n data objects (defined by p variables) to k clusters is solved:

i	v_1	v_2	$\ldots$	v_p	class
1	z_{11}	z_{12}	$\ldots$	z_{1p}	2
2	z_{21}	z_{22}	$\ldots$	z_{2p}	5
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
n	z_{n1}	z_{n2}	$\ldots$	z_{np}	2

Differential Evolution and optimal clustering

- count of possible partitions:

$$S(n,\ k) = \frac{1}{k!}\sum_{l=1}^{k}(-1)^{k-l}\binom{k}{l}l^{n}$$

n	k	$S(n,\ k)$
10	2	511
20	4	4.52E+010
100	5	7.89E+069
200	20	6.60E+241

- criterion to be **minimized** is $TRW = tr(\boldsymbol{W})$:

$$\boldsymbol{W} = \sum_{l=1}^{k}\boldsymbol{W}_l$$

- where $\boldsymbol{W}_l$ is variance matrix of attributes for the objects belonging to cluster $C_l,\ l = 1,\ 2,\ \ldots,\ k$

- three variants of **DE** algorithm (DE, jDE, b6e6rl) were applied and compared with k-means, ($N = 30$)
- eight various **real-world** data sets are used as a benchmark
- $n : 150 - 871, \quad p : 3 - 34, \quad k : 2 - 6$
- k-means algorithm is more **efficient** than DE-based methods in some easier problems of the benchmark
- the proposed approach is **applicable** in any arbitrary DE-based algorithm

Comparison of nature-inspired algorithms

- Bujok, P., Tvrdík, J., Poláková, R.: Comparison of nature-inspired population-based algorithms on continuous optimisation problems, *Swarm and Evolutionary Computation* **50** (2019) DOI:10.1016/j.swevo.2019.01.006

- many 'new' optimisation methods are **proposed** each year (especially nature-based ones)

- often, **poor** existing methods and **simple** test problems are used in the comparison

- **advanced adaptive methods** are developed to perform well on various optimisation problems

- the main **goal** of the study is to answer the question: '*How do very often applied Nature-inspired methods perform in comparison with advanced DE variants?*'

Nature-inspired algorithms in comparison

1. **ABC** (artificial bee colony, 2009) is controlled by *limit*$=N$

$$y(j) = P(i,j) + (P(i,j) - P(k,j)) \cdot U(-1,1)$$

2. **Bat algorithm** (2009), frequencies $f_{max} = 2,\ f_{max} = 0$, loudness $A_i = 1.2$ is reduced for unsuccessful individuals by $\alpha = 0.9$, emission rate $r_i = 0.1$ is for successful individuals increased by $\gamma = 0.9$

3. **Cuckoo search** (2009), probability to put a Cuckoo's egg into a host nest is $pa = 0.25$, Lévy flight parameter is $\lambda = 1.5$

4. **ACS-CS** (Adaptive Cuckoo Search, 2016) is an enhanced variant of CS with $pa = 0.25$

5. **DFO** (dispersive flies optimisation, 2014) is controlled by a disturbance threshold, $dt = 1 \times 10^{-3}$
6. **Firelfy** algorithm (2008), randomisation $\alpha = 0.5$, light absorption $\gamma = 1$, and attractiveness is updated between $\beta_0 = 1$ and $\beta_{min} = 0.2$
7. **Flower** Pollination Algorithm (2012) enables to switch between local and global optimisation ($p = 0.8$), Lé vy flight parameter is $\lambda = 1.5$
8. **MBO** (Monarch Butterfly Optimisation, 2015), elitism parameter $keep = 2$, $MaxStepSize = 1$, $seasonalPeriod = 1.2$, and proportion of the first sub-population $part = 5/12$

9. **PSO** (Particle Swarm Optimisation, 1998), variation *w* is linearly interpolated from $w_{max} = 1$ to $w_{min} = 0.3$, local and global velocity weight is $c = 1.05$

10. **HFPSO** (Hybrid Firefly and PSO algorithm, 2018) uses parameters for both original methods - $\alpha = 0.2$, $\beta_0 = 2$, $\gamma = 1$, and $c_1 = c_2 = 1.49$

11. **SOMA** (Self-Organising Migration Algorithm, 2000) has parameters *strategy* (all-to-one), and parameters *PathLength*$=2$, *Step*$=0.11$, and *Prt*$=0.1$

- (blind) Random Search (**RS**, 1963) was incorporated in the comparison as a **reference** to indicated poor methods

Differential Evolution algorithm variants

1. **classic DE** (1997) DE/randrl/1/bin, $F = 0.8$, $CR = 0.8$
2. **CoBiDE** (2014) uses covariance-matrix-based crossover and bimodal distribution of $\{F,\ CR\}$, *peig*$=$ 0.4 and *ps*$=$ 0.5
3. **L-SHADE** (2014) uses current-to-*p*best mutation, archive *A* and linearly decreased population size
4. **SHADE4** (2016) is SHADE with competition of four DE strategies (current-to-*p*best - randrl/1, binomial - exponential) based on their success
5. **jSO** (2017) is improved L-SHADE version with weighted mutation (second best algorithm in CEC 2017)

- **test** and **real-world** optimisation problems were used to indicate the **difference between the efficiency** of algorithms
- the set of **artificial** problems CEC 2014 contains **30 problems** in four classes of difficulty: **unimodal** (3), **multimodal** (13), **hybrid** (6), and **composition** (8)
- three **dimensions** of the search space were used: $D = 10,\ 30,\ 50$
- for each algorithm and problem, **51 independent** runs were carried out
- the **error** value of each run is computed, $error = f(\boldsymbol{x}^{*}) - f_{min}$

Experiments - CEC 2011

- 22 **real-world** optimisation problems from CEC 2011 competition were used as the second benchmark set
- **dimensionality** of the problems is $1 \leq D \leq 240$
- a true solution is **not known**, a lower function value is better
- for each method and problem, **25 independent** runs were carried out
- the minimal function value is recorded in **three stages** of each run, $FES =$50,000, 100,000, 150,000

Mean ranks from the Friedman tests (CEC 2014)

D, Alg.	jSO	LSHA	CoBi	SHA4	Firefly	ACS-CS	Cuckoo	ABC
D=10	**3**	4.1	**3.6**	3.7	7.4	7.6	6.8	8.4
D=30	**2.1**	**3**	3.9	4.2	6.7	7	7.4	8.3
D=50	**2.3**	**3.2**	4.2	4.1	5.9	7.3	8.6	7.1
avg	2.5	3.5	3.9	4	6.7	7.3	7.6	7.9
Flower	HFPSO	SOMA	DE08	PSO	MBO	DFO	RS	Bat
7.9	9.5	8.9	9.5	11.3	14	15.1	15.2	16.9
9	8.6	9.1	10.3	11.9	13.9	15.3	15.6	16.9
9.1	8	8.9	10.8	11.8	14	15.3	15.5	16.9
8.7	8.7	9	10.2	11.7	14	15.2	15.4	16.9

Counts of shared best positions from the Kruskal-Wallis tests (CEC 2014)

D	jSO	LSHA	CoBi	SHA4	Firefly	ACS-CS	Cuckoo	Flower
10	22	13	15	17	7	4	4	4
30	25	21	15	11	8	5	4	1
50	23	20	9	10	6	2	0	0
Σ	70	54	39	38	21	11	8	5
PSO	HFPSO	ABC	DE08	SOMA	MBO	DFO	RS	Bat
2	2	2	1	1	0	0	0	0
1	1	0	0	0	0	0	0	0
1	1	1	1	0	0	0	0	0
4	4	3	2	1	0	0	0	0

Counts of shared worst positions from the Friedman tests (CEC 2014)

D	jSO	LSHA	CoBi	SHA4	Firefly	ACS-CS	Cuckoo	Flower
10	0	0	0	0	0	0	0	0
30	0	0	0	0	0	0	0	0
50	0	0	0	0	0	0	0	0
Σ	0	0	0	0	0	0	0	0
PSO	HFPSO	ABC	DE08	SOMA	MBO	DFO	RS	Bat
0	0	1	3	0	4	23	25	30
0	0	0	2	0	4	20	24	30
0	0	0	1	0	4	15	26	30
0	0	1	6	0	12	58	75	90

Mean ranks from the Friedman tests (CEC 2011)

FES	jSO	LSHA	SHA4	CoBi	Cuckoo	HFPSO	Flower	SOMA
50,000	7.5	8.3	**<u>2.3</u>**	<u>5.9</u>	**5.5**	**5.5**	6.8	6.7
100,000	**4.8**	7.1	**<u>2.8</u>**	<u>5.9</u>	6.3	6.3	6.9	7.2
150,000	**<u>2.8</u>**	**3.5**	<u>3.8</u>	6.1	6.9	7.0	7.6	7.9
ACS-CS	ABC	DE08	PSO	MBO	DFO	RS	Firefly	Bat
7	8.9	10.6	8.7	11.9	13.9	14.6	14.1	14.7
7.6	8.9	9.8	8.8	12.6	13.8	14.5	14.7	15
8.3	9.1	9.6	9.7	12.7	13.9	14.1	14.9	15

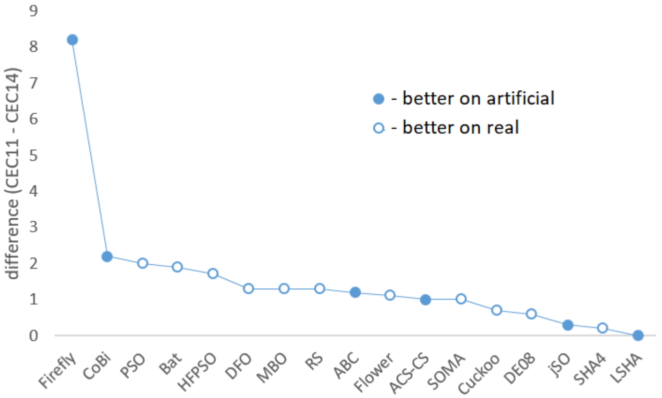
Counts of shared best and worst positions from the Friedman tests (CEC 2011)

posit	jSO	SHA4	LSHA	CoBi	HFPSO	PSO	Cuckoo	DE08
best	18	16	15	6	5	5	3	2
worst	0	0	0	0	1	1	0	2
Flower	ABC	SOMA	ACS-CS	MBO	DFO	RS	Bat	Firefly
1	1	0	0	0	0	0	0	0
0	1	0	1	7	15	16	17	17

Differences of mean ranks from the Friedman tests

Alg	Firefly	CoBi	PSO	Bat	HFPSO	DFO	MBO	RS
diff	8.2	2.2	-2	-1.9	-1.7	-1.3	-1.3	-1.3

ABC	Flower	ACS-CS	SOMA	Cuckoo	DE08	jSO	SHA4	LSHA
1.2	-1.1	1	-1	-0.7	-0.6	0.3	-0.2	0



Conclusion

- **different performance** of 17 stochastic optimisation methods on artificial and real-world problems (No-Free-Lunch theorem)
- **advanced** nature-inspired methods achieved acceptable results, mostly **simple** variants are used in applications
- the **difference** in mean ranks of Firefly algorithm is surprising
- the **worst** performing Bat algorithm is used in real applications
- published nature-inspired methods are often '**recycled**' versions of existing algorithms
- good results of a new method on artificial problems do **not guarantee** good results in real application

THANKS FOR YOUR ATTENTION

Separable functions

- Feoktistov, V.: Differential Evolution In Search of Solutions (2006)
- function f of D variables is separable when:

$$\frac{\partial f(X)}{\partial x_i} = g(x_i) \cdot h(X),\ X = (x_1,\ x_2, \ldots,\ x_D),$$

- function $f(X) = (x_1^2 + x_2^2)^2$ is not separable
- first derivation of $f(X)$ is $\dfrac{\partial f(X)}{\partial x_1} = 4x_1 \cdot (x_1^2 + x_2^2)$, $x_1 = 0$