EUROPEAN UNION
European Structural and Investment Funds
Operational Programme Research,
Development and Education

MINISTRY OF EDUCATION,
YOUTH AND SPORTS

Evolutionary Computation Techniques

Swarm intelligence

Algorithms PSO a SOMA

Strategic project of TBU in Zlín, reg. no. CZ.02.2.69/0.0/0.0/16_015/0002204

Doc. Ing. Roman Šenkeřík, Ph.D.
FAI, Ústav informatiky a umělé inteligence

Content of lecture

- Swarm inteligence.
- PSO algorithm.
- Modern variants of PSO.
- SOMA algorithm
- Modern variants of SOMA
- Other useful information

All graphic content here is used either under a Creative Commons license for free distribution, or as the presentation Author's own creation, or with the consent of the graphic Author, in other cases a reference to the source is given.

1. Swarm intelligence - Basic principles

Swarm intelligence

- Modern methods of global optimization include the most special techniques and methods inspired by the behavior of animals in swarms/flocks [1], [2], [3], [4].

- Many animal groups are characterized by the phenomenon of collective intelligence, where a swarm/flock acts as a whole without having central control or being controlled by someone else.

- Animal swarms/flocks can effectively find food and defend themselves against predators, even though they do not have central control.

Swarm intelligence

- Although individuals are guided solely by information from their immediate surroundings, a sophisticated way of communicating with each other allows the system to function properly without individuals knowing the group's goals.

- Swarm algorithms mimic the observed behavior of flocks and swarms of animals from termite colonies to bats. They successfully use this behavior to optimize a variety of real-world problems.

- The search for further inspiration and rapid development was accelerated by the success of PSO algorithms and ant colony optimization (ACO).

Swarm Intelligence

- Each currently available swarm algorithm has its own original pattern of behavior inspired by nature.

- The observed behavior has already gone through an imaginary optimization by means of natural selection, and therefore it can be considered as a successful solution.

- The subject of swarm algorithms is a suitable simulation of this exemplary behavior. BUT! in many cases it is necessary to define the necessary simplifications of the given system.

Swarm Intelligence – An overview

Swarm Intelligence

Biology

- Insect / animal groups
- Cooperative behavior
- Solving tasks no-individual can solve alone
- Global intelligence driven by local interactions

Computer Science

- Branch of AI (CI)
- Bio-inspired algorithms
- Metaheuristic optimizers
- Population-based
- Communication
- Iterative process

Robotics

- Cooperative problem solving
- Decentralization
- Self-organization
- Autonomous systems

2. Particle Swarm Optimizer Algorithm (PSO)

Particle Swarm Optimization – An Inspiration

Source: https://www.biographic.com/posts/sto/lens-of-time-secrets-of-schooling

PSO Algorithm

- One of the most popular metaheuristic optimizer.

- The PSO algorithm is a simple optimization tool that was created in 1995 by Russell Eberhart and James Kennedy [5].

- Imitation of "intelligent behavior" of a flock of e.g. birds / fish.

- Population algorithm: it is based on working with a population of individuals whose position in the space of possible solutions is changed using the so-called velocity vector.

- It does not contain "classical" evolutionary functions such as crossover and mutations.

PSO Algorithm

- It does not contain the classic mechanism of exchanging the old generation for a new one based on evolutionary operators. The entire population is constantly "on the move" and in each iteration there is an update of positions, and an "evolution" of global / local (personnel) knowledge about the quality of the solution.

- Each individual (particle) is influenced by a shared global best solution (*gBest*) and his own personal best solution (*pBest*) as it moves over the hyperplane of the objective function.

- Given time, the swarm converges on location of agreed global best solution.

- Very successful algorithm in practice, there are many versions and modifications…

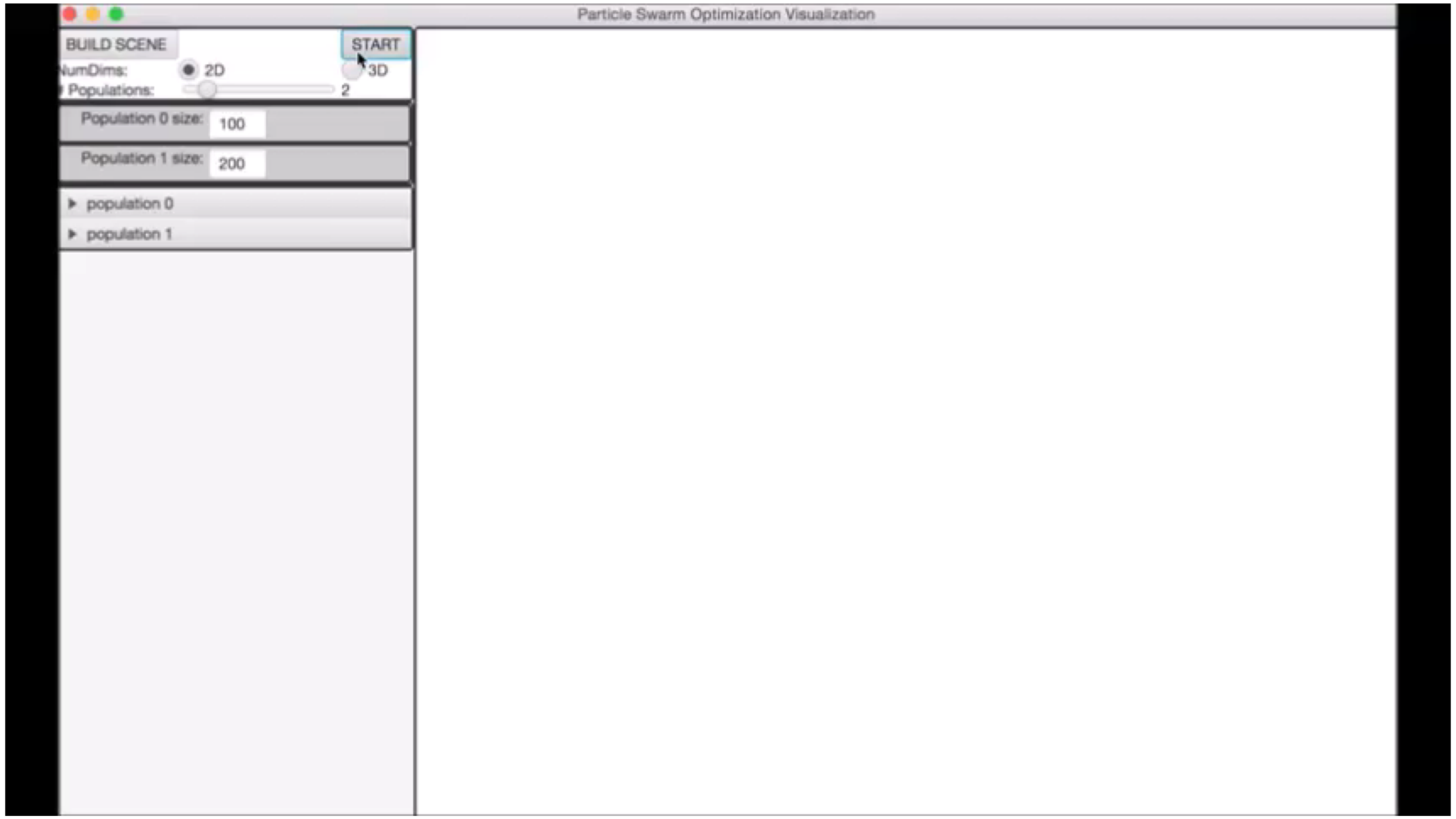
PSO Algorithm

- The PSO algorithm is initialized by a population of randomly placed particles (solutions / individuals). The initial speed is zero.

- In subsequent iterations, a speed vector (so-called "velocity") is calculated for each individual, which indicates the direction in which the individual wants to “fly” in the next iteration. The value of the objective function is calculated from the position of the individuals. The individual with the lowest / highest value stores his current position in the common memory of the population (*gBest*), so that each individual knows where the best solution found is located.

- At the same time, each individual will find out if his current position is not better than his current best position. If so, it will store this new best position in its memory (pBest). Once each particle knows gBest and pBest, the particle adjusts its velocity and position according to the PSO equations of motion.

Particle Swarm Optimization - Simulation



Source: https://github.com/0la0/psoViz

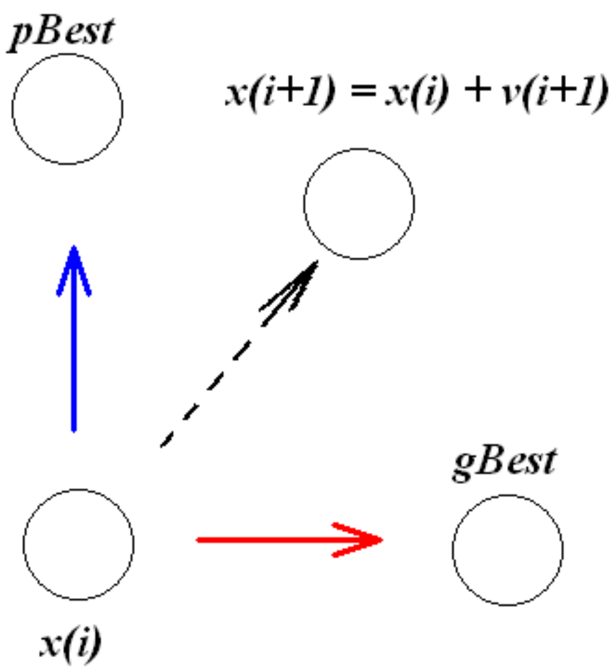
PSO Algorithm

PSO formula

$$v_{ij}^{t+1} = w \cdot v_{ij}^{t} + c_1 \cdot r_1 \cdot \left(pBest_{ij} - x_{ij}^{t}\right) + c_2 \cdot r_2 \cdot \left(gBest_j - x_{ij}^{t}\right)$$

Position update

$$x_{ij}^{t+1} = x_{ij}^{t} + v_{ij}^{t+1}$$



Where:

$v(t+1)$ – New velocity of particle

$v(t)$ – Current velocity of particle.

c_1, c_2 – Priority factors

pBest – Best solution found by particle

gBest – Best solution found in population

$x(t)$ – Current position of particle

Rand – Random number, from the interval <0,1>.

3. Modern variants of PSO

Heterogeneous particle swarm optimizers (HPSO)

- It allow particles to use different update equations, representing different types of behaviors, within the swarm [6].
- Dynamic HPSO allows the particles to change their behaviors during the search.
- This method alter the exploration/exploitation balance during the search based on the simple learning mechanism (f_k-HPSO).
- Usually pool of six behaviors is implemented.

Heterogeneous particle swarm optimizers (HPSO)

TVIW-PSO, time variant inertia weight with constant acceleration factors,

TVAC-PSO, in this model the value of c_1 and c_2 changes linearly. The c_1 is decreasing and c_2 increasing therefore the algorithm has good exploration and exploitation ability,

sPSO, social behavior favoring model, $v_{ij}^{t+1} = v_{ij}^{t} + c_1 \cdot Rand(\ ) \cdot (gBest_j - x_{ij}^{t})$

cPSO, cognitive behavior favoring model, $v_{ij}^{t+1} = v_{ij}^{t} + c_1 \cdot Rand(\ ) \cdot (pBest_{ij} - x_{ij}^{t})$

modBB-PSO, is able to escape from local minima as well as explore the *pBest* areas. The algorithm is more likely to select *pBest* area for exploration in the end phase,

QSO uses a linear decrease of "cloud size" around the *gBest*. Therefore, the particles are forced to move in shrinking space in every dimension

Orthogonal Learning PSO (OLPSO)

- The orthogonal learning method consists of following steps [7]:
  - Creating an orthogonal matrix according to a set of rules.
  - Creating trial solutions by crossover of *pBest* and *gBest* according to data in the matrix.
  - Evaluating trial vectors and storing the best as x_b.
  - Evaluating the most beneficial dimensional components and creating trial vector x_p according to the results.
  - Comparing the fitness of x_b and x_p. The winner is used as a guidance vector P in:

$$v_{ij}^{t+1} = w \cdot v_{ij}^{t} + c \cdot Rand() \cdot (P_{ij} - x_{ij})$$

Comprehensive learning particle swarm optimization (CLPSO)

- CLPSO utilizes the idea of using a personal best solution of other members of the swarm as a way to improve the search quality.
- Based on a learning probability, the movement of a particle in each dimension might be based on *pBest* location (dimension component) of another particle.
- At least one-dimensional component is always used from another particle than self.
- More in research paper: [8].

Heterogenous comprehensive learning particle swarm optimization (HCLPSO)

- Update of CLPSO version.
- The swarm (population) is divided into two subpopulations.
- One is enhanced for exploration and the second one is enhanced for exploitation.
- For both subpopulations, a comprehensive learning (CL) strategy [8] is used to generate example particle $pBest_{\mathrm{fi(j)}}$.
- For detailed information about selecting example particle, see the full paper [9].

Attractive and Repulsive PSO (ARPSO)

- This modification was created due to the premature convergence of generic PSO on complex multimodal optimization problems [10].
- This phenomenon arises due to the reduction of population diversity in the searched area, which leads to the absolute stagnation of the value of the objective function of the entire swarm.
- A mechanism for controlling the diversity of individuals (population of solutions) was introduced and the equation of calculating the new velocity for the next iteration was modified.
- A simple variable (*dir*) indicating the direction of particle motion was introduced.
- This can only take the values 1 (normal behavior of the individual as in a generic PSO - attractive force), or -1 (repulsive direction of movement of the individual).

Diversity guided ARPSO

- Measuring the diversity of the population:
- $diversity = \frac{1}{NP \cdot |L|} \cdot \sum_{i=1}^{NP} \sqrt{\sum_{j=1}^{D} \left(x_{i,j} - \overline{x_j}\right)^2}$
- If the diversity drops below a threshold:

$$v_{ij}^{t+1} = w \cdot v_{ij}^{t} + c_1 \cdot r_1 \cdot \left(pBest_{ij} - x_{ij}^{t}\right) - c_2 \cdot r_2 \cdot \left(gBest_j - x_{ij}^{t}\right)$$

- More in research paper [10].

4. SOMA Algorithm

SOMA algorithm

- Model of social cooperation between individuals (when solving optimization task) [12].
- Movement in series of „jumps“ across the search space.
- Stochastic component of movement – perturbation
- Swarm-based algorithm (with mutation functionality).

SOMA algorithm

- Discrete perturbation vector - mimics the mutation process known from the classical evolutionary computing techniques.

- Self-adaptation of movement over the search space (relative step sizes and path lengths for individuals).

- Several different strategies:
  - All to One
  - All to All
  - All to All Adaptive
  - All to Random

SOMA - Animation

Before we focus on the equations, let's look at a classic animation from the beginning of the millennium :-)

SOMA - Animation

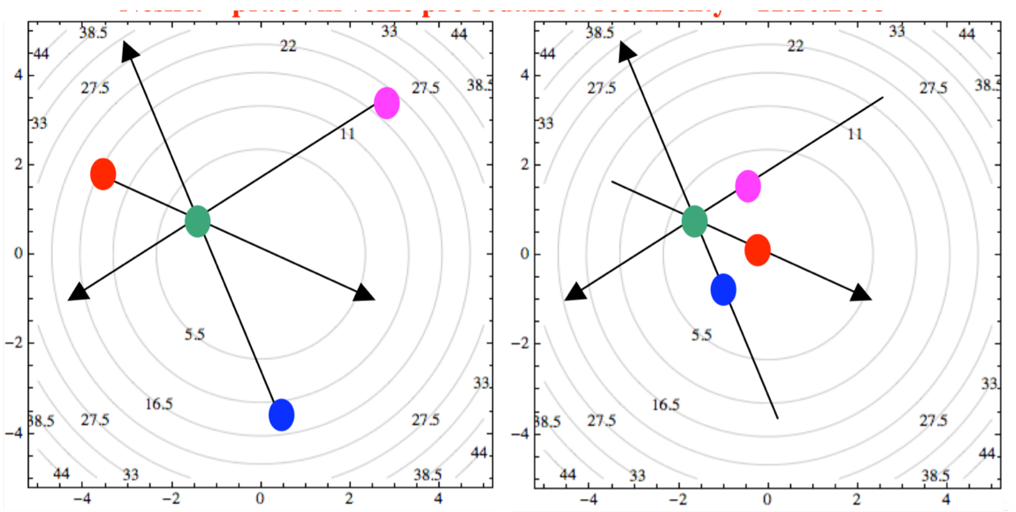
https://youtu.be/cewyVJ1Xjjw

SOMA algorithm: Strategies

- **All To One:** All individuals from the population are migrating (moving) toward one selected leader. This leader is selected based on its objective function value, the individual with the best objective function value is selected in each iteration of an algorithm.

- **All To Random:** The leader is selected randomly at the beginning of each iteration from all individuals in the population.

Source: BIA lectures http://navy.cs.vsb.cz/

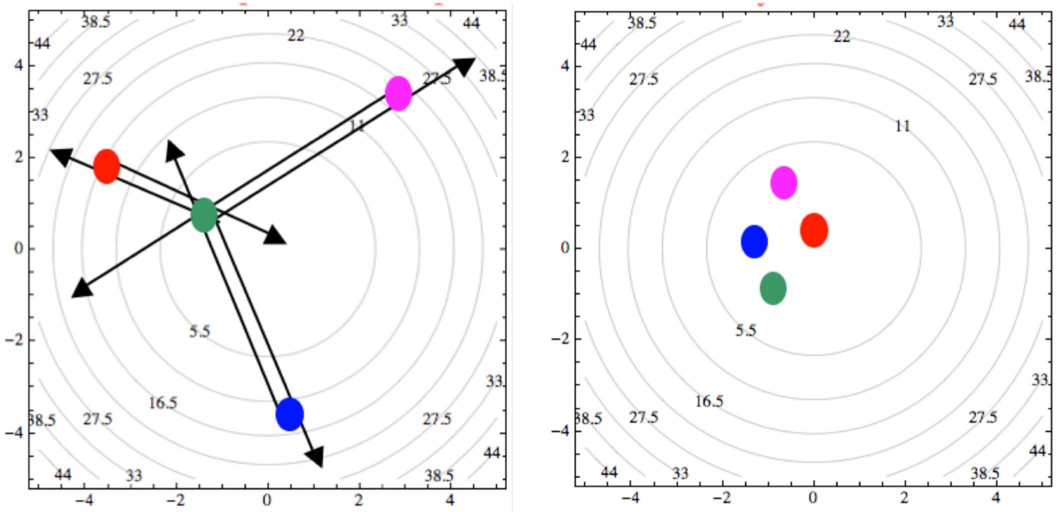


SOMA algorithm: Strategies

- **All To All:** All individuals migrate towards all others. After the end of migration of a particular individual, this individual returns to its original position. All individuals update their positions after all individuals completely perform their migrations.

- **All To All Adaptive:** This strategy is identical with the AllToAll with the difference that the currently migrating individual does not update its position after all migrations is finished (to each other individuals), but the update process is immediate. From this "new" position, it then migrates to the other remaining individuals.

Source: BIA lectures http://navy.cs.vsb.cz/



SOMA algorithm: Pseudocode

$$\begin{aligned}&\mathit{Input}: N, \mathit{Migrations}, \mathit{PopSize} \ge 2, \mathit{PRT} \in [0,1], \mathit{Step} \in (0,1],\ \mathrm{MinDiv} \in (0,1],\\ &\qquad \mathrm{PathLength} \in (0,5],\ \mathit{Specimen} \text{ for all parameters } x_j^{(hi)}, x_j^{(lo)}\end{aligned}$$

$$\text{Inicialization: } \begin{cases} \forall i \le \mathit{PopSize} \wedge \forall j \le N : x_{i,j,\mathit{Migrations}=0} = x_j^{(lo)} + \mathit{rand}_j[0,1] \bullet \left(x_j^{(hi)} - x_j^{(lo)}\right) \\ i = \{1,2,\ldots,\mathit{Migrations}\},\ \ j = \{1,2,\ldots,N\},\ \mathrm{Migrations} = 0,\ \mathit{rand}_j[0,1] \in [0,1] \end{cases}$$

$$\begin{cases} \text{While } \ \mathrm{Migrations} < \mathrm{Migrations}_{\max} \\ \forall i \le \mathit{PopSize} \begin{cases} \text{While } \ t \le \mathit{PathLength} \\ \mathit{if}\ \ \mathit{rnd}_j < \mathit{PRT}\ \ \mathit{pak}\ \mathit{PRTVector}_j = 1\ \ \mathit{else}\ 0\,, \quad j = 1,\ldots,N \\ x_{i,j}^{ML+1} = x_{i,j,start}^{ML} + (x_{L,j}^{ML} - x_{i,j,start}^{ML})\, t\, \mathit{PRTVector}_j \\ f\left(x_{i,j}^{ML+1}\right) = \ \text{ if } f\left(x_{i,j}^{ML}\right) \le f\left(x_{i,j,start}^{ML}\right) \ \ \text{ else } f\left(x_{i,j,start}^{ML}\right) \\ t = t + \mathit{Step} \end{cases} \\ \mathrm{Migrations} = \mathrm{Migrations} + 1 \end{cases}$$

SOMA algorithm

Cooperation between individuals is defined as a migration (movement).

$$x_{i,j}^{k+1} = x_{i,j}^{k} + \left(x_{L,j}^{k} - x_{i,j}^{k}\right) \cdot t \cdot PRTVector_j \qquad PRTVector_{i,j} = \begin{cases} if\ rand_j < prt_i, & 1 \\ otherwise, & 0 \end{cases}$$

Where:

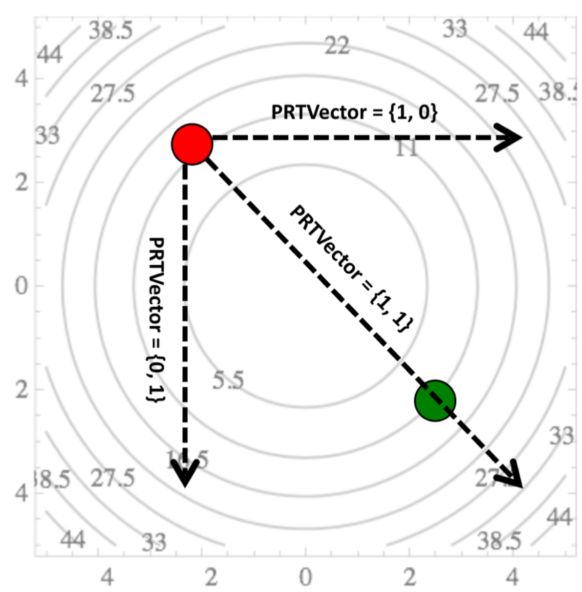
$x_{i,j}^{k+1}$ is a new position of i-th solution (for iteration k+1) for dimension j.

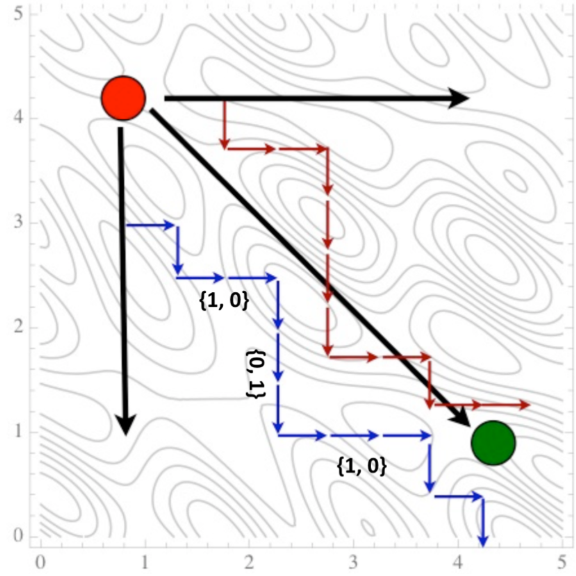
The current position of a solution x_i

The x_L is a position of a so called leader, which is selected based on the chosen SOMA strategy.

Parameter t represents steps from i-th solution to the leader.

prt is a user defined parameter (threshold for "mutation")





Source: BIA lectures http://navy.cs.vsb.cz/

SOMA algorithm: Parameters

Recommended parameter values:

Parameter	Value	Comment
Population Size	10 - 25 - 50 - 100	Depends on strategy, min 25 for ATO/ATR, lower values for ATA/ATAA, to keep CFE budget
PathLength	1.1 - 5.0	Recommended value 2.0 - 3.0 Lower values - less exploration
Step size	0.11 - Pathlength	Recommended 0.11, 0.22, 0.33… NOT (0.1, 0.25, 0.5) - NOT to step at "Leader"
prt value	0.1 - 0.9	0.3 seems to be a good choice, but it is clearly problem dependent

SOMA algorithm: FES Budget

- FES/CFE Calculations for ATO/ATR Strategy:

$$CFE = \frac{(PopSize - 1) * PathLength * \mathrm{Migrations}}{Step}$$

- FES/CFE Calculations for ATA/ATAA Strategy:

$$CFE = \frac{PopSize * (PopSize - 1) * PathLength * \mathrm{Migrations}}{Step}$$

5. Modern Variants of SOMA

ESP-SOMA Algorithm

- **ESP**: The **E**nsemble of **S**trategies and **P**erturbation Parameter in Self-organizing Migrating Algorithm [13].

- User-predefined parameters are adaptive based on the actual performance of a running algorithm.

- Each individual from a population has its parameter values (*prt* value and *Strategy*) assigned.

ESP-SOMA Algorithm

- Each individual $\boldsymbol{x}$ randomly chooses its own SOMA strategy from a given set: {AllToOne, AllToAll, AllToRandom}

- **Optionally**, each individual $\boldsymbol{x}$ randomly chooses its own *prt* value from given set: {0.1, 0.3, 0.5, 0.7, 0.9}, based on the user settings of *adaptivePrt* Boolean parameter.

ESP-SOMA Algorithm

- ***Refreshing gap***
  - Maximum number of unsuccessful migrations of ***x***.
  - If a solution reaches this maximum level, it is forced to adapt new *prt* value (optionally) and new *Strategy*.
  - Currently no recommended setting is available (problem dependent).

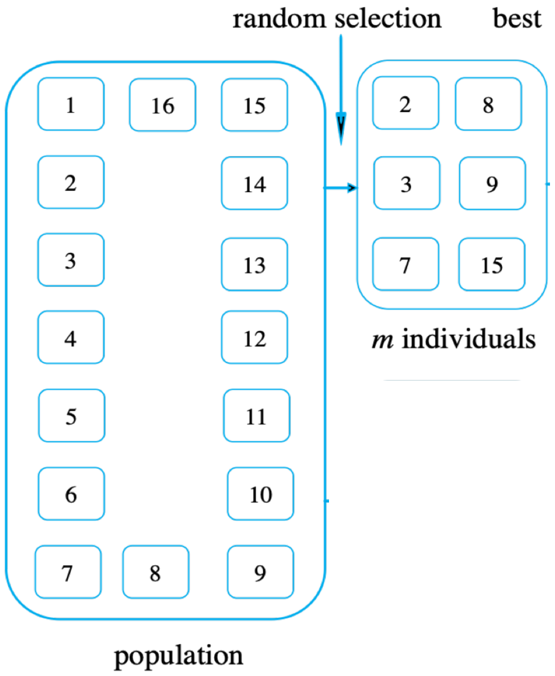
ESP-SOMA Algorithm

- ***AdaptivePrt***
  - Boolean parameter, that is used to alternate between fixed *prt* value or adaptive version (assigned ensemble).
  - If *adaptivePrt* == 1, then each individual ***x*** can adapt the assigned *prt* value from the pool of *prt* values. ***ESP-SOMA is adapting both prt and Strategy.***
  - If the *adaptivePrt* == 0, then all individuals ***x*** have the fixed value prt = 0.3. ***ESP-SOMA is adapting ONLY Strategy.***

T3A SOMA

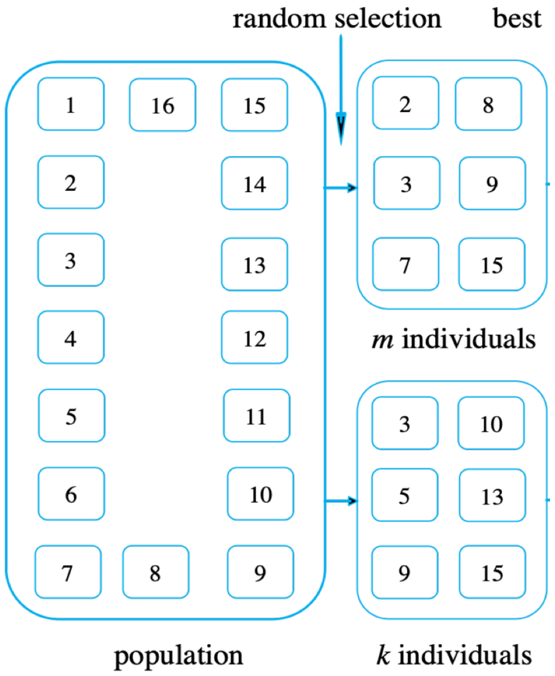
- T3A = Team to Team Adaptive SOMA [14].
- Introducing new 2-step organization process of random/best driven selection from subpopulations.
- New control mechanisms for "*prt*" and "*step*" parameter.

T3A SOMA: Organization Process



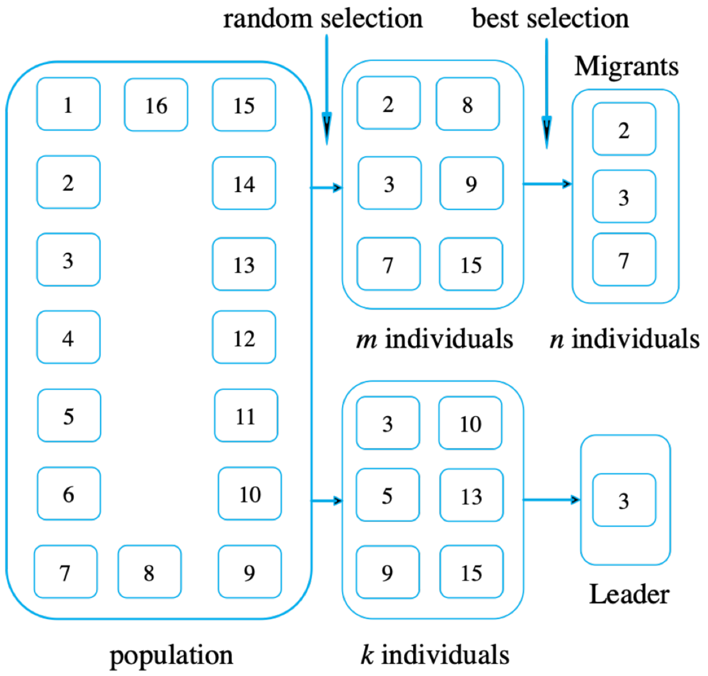
- Random selection of a small group containing *m* individuals in the population to become candidates for the ***migrating individuals***.

T3A SOMA: Organization Process



- To select the ***“Leader subgroup”***, the algorithm randomly select a small group containing *k* individuals from the population

T3A SOMA: Organization Process



- Selection of the best *n* out of *m* individuals to become ***migrating individuals***.
- The best individual from *k*-size subgroup is the ***Leader***.

T3A SOMA: Organization Process

Algorithm 1 SOMA T3A

Create the initial population
Evaluate the initial population
while stopping condition not reached **do**
    Update PRT and $Step$ values
    Choose randomly m individuals from the population
    Choose the best n Migrants out of m individuals
    for $i = 1$ to n Migrants **do**
        Choose randomly k individuals from the population
        Choose the Leader from k individuals
        if the Migrant is not the Leader **then**
            The Migrant moves to the Leader
            Checking boundary
            Re-evaluate fitness function
            Updated the better position of the Migrant
        end if
    end for
end while
return

T3A SOMA: Organization Process

- Individuals in the *k* group that includes the Leader may coincident with one of the *n* migrating individuals group, in this case, the movement is not performed (ignored).

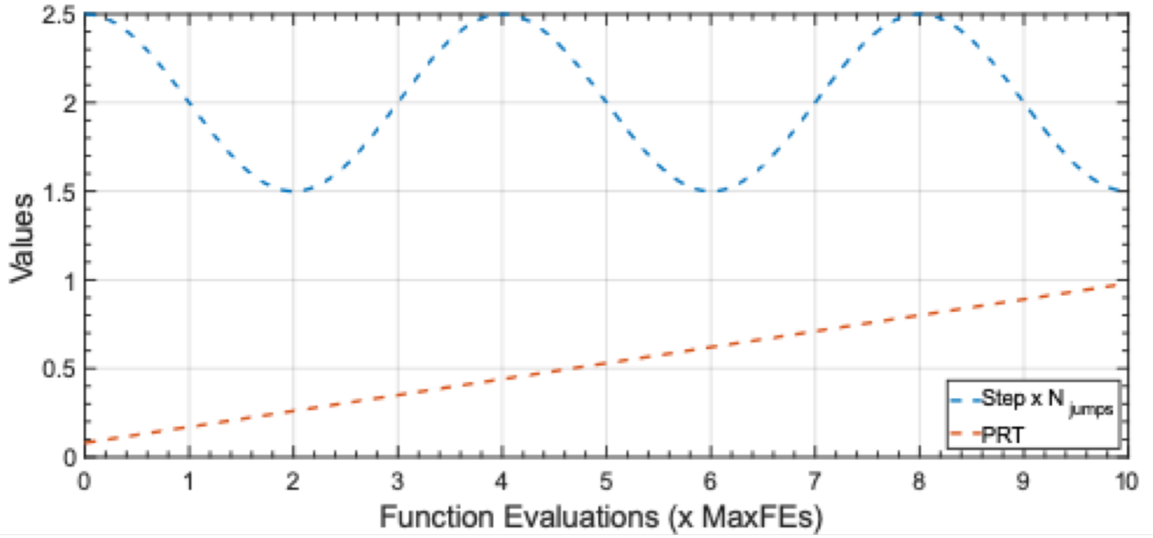
- Leader selection is performed for each migrating individual. That is, corresponding to *n* individuals, there are *n* times generated the *k* group, thus we have *n* “original” Leaders, instead of just choosing the only Leader for *n* migrating individuals.

- Currently, there are not known best values of *m*, *n*, and *k* for all optimization functions. Their values depend on the complexity and number of dimensions of functions, as well as the population size.

T3A SOMA: Adaptive Parameters

- Control mechanism for *prt* and *step*

- Self-adaptation in granularity of steps and transition between exploration and exploitation.



$$PRT = 0.08 + 0.90\,\frac{FEs}{MaxFEs} \qquad Step = 0.02 + 0.005\,cos(0.5\pi 10^{-7} FEs)$$

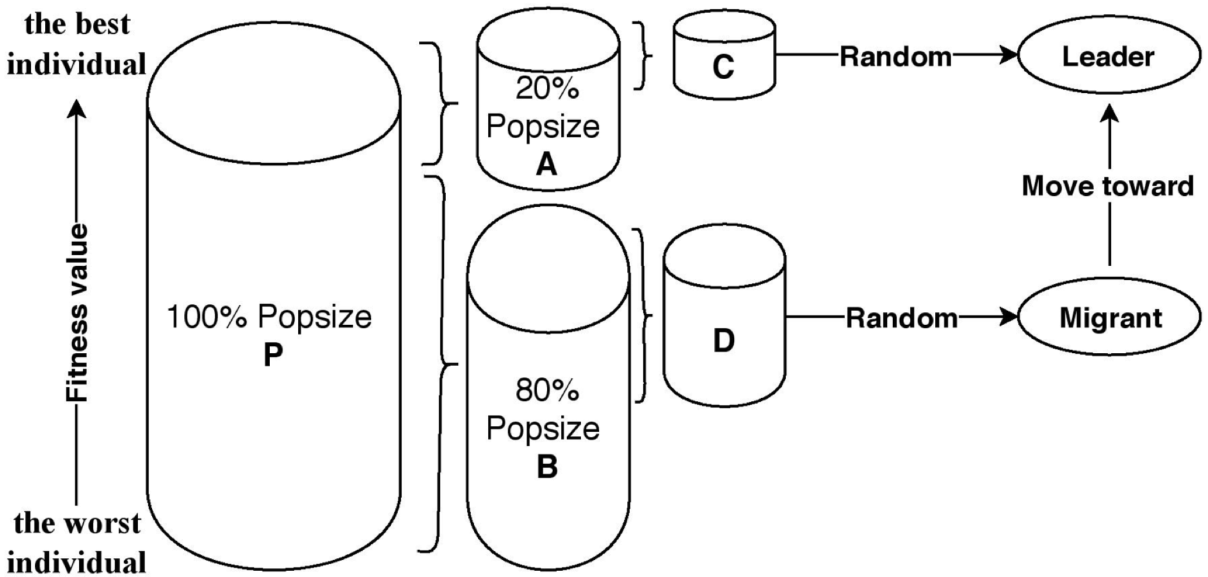
Pareto SOMA

- ***T3A version reveals important fact:*** The organization process plays an essential role in exploring promising subspaces and then focusing on exploiting these promising subspaces.

- Therefore, an individual selected to become the Leader is an individual with good fitness value, but should not always be the best.

- To choose the migrating individual and Leader, another SOMA version is applying the “Pareto" principle [15].

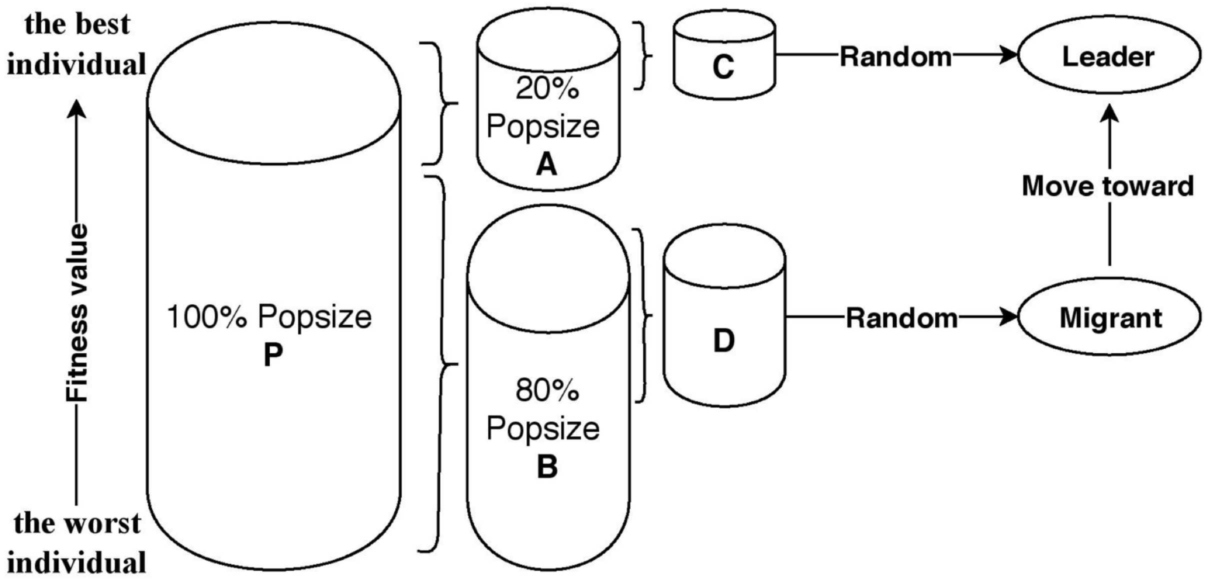
Pareto SOMA: Organization Process



- We choose 20% of the population with the best fitness value to form the group **A**, while the remaining 80% of the population will put into group **B**.

- The Leader should be in group **A** and the migrating individual should be in group **B**.

Pareto SOMA: Organization Process



- Next step: Randomly select an individual within the best 20% of group A (marked as group **C**) as the Leader and randomly select an individual within the best 20% of group B (marked as group **D**) as the migrating individual…

- Controlling the *prt* and *step* is similar to T3A version

SOMA-CL

- ***CL - Self-organizing Migrating Algorithm with Clustering-aided Migration.*** [17]

- The new variant of SOMA utilizes the natural abilities of exploration and exploitation phases of its ancestor. These phases are influenced by migration strategies.

- SOMA-CL in each iteration migrates the population by two separate strategies:
  - All-To-Random (Exploration)
  - All-To-Cluster-Leaders (Exploitation)

SOMA-CL

- The first strategy *All-To-Random* serves mainly as an explorer, which maps the search space.
- The second strategy is named *All-ToCluster-Leaders* and is used to exploit areas of potential positions of a global optima.
- Each migration strategy utilizes a different set of control parameters, known from canonical SOMA (*PopSize, step, pathLength, prt*).

SOMA-CL

- Initialization

1. All-To-Random Migration Strategy

2. Identification of Cluster Leaders

3. All-To-Cluster-Leaders Migration Strategy

Algorithm 1 SOMA-CL

Set D, NP, NP_L, and $MAXFES$
Set $step$, $pathLength$, and prt
Set $step_L$, $pathLength_L$, and prt_L
while Stopping criterion not met **do**
    $M = \emptyset$
    for $i = 1$ **to** NP **do**
        x_L = pick random solution x
        for $t = 0$ **to** $pathLength$ **with** $t+ = step$ **do**
            generate $\boldsymbol{PRTVector}$ by eq. (2)
            migrate x_i to x_L by eq. (1)
            save each evaluated solution into $\boldsymbol{M}$
        end for
    end for
    k-means clustering method for solutions stored in $\boldsymbol{M}$
    keep only best-solution from each cluster
    sort the remaining solutions in $\boldsymbol{M}$
    for $i = 1$ **to** NP **do**
        x_L = Rank Selection from cluster leaders
        for $t = 0$ **to** $pathLength_L$ **with** $t+ = step_L$ **do**
            generate $\boldsymbol{PRTVector}$ by eq. (2)
            migrate x_i to x_L by eq. (1)
        end for
    end for
    record the best solution
end while

SOMA: Applications

- Non-linear controller: SOMA showed interesting performance in comparison with DE
- AP/GE (symbolic regression): SOMA performed very well
- DSOMA: very good performance in OR applications
- Computer games: SOMA Tic-Tac-Toe, GameSourcing powered by SOMA
- Many more…

SOMA: Applications

- Many different versions of SOMA were introduced in recent year(s): CSOMA, C-SOMGA, M-SOMAQ, M-NM-SOMA, DSOMA, MOSOMA, QSOMA and others.

- Many open possibilities…

- Novel ESP-SOMA/T3A/CN-SOMA has manifested fine exploration and exploitation ratio securing stable searching in highly constrained search space

- Finally, in many applications best results were obtained through the ***utilization of simpler adaptation strategies applied to SOMA. This can be very positive regarding the understandability, applicability, and effectivity of the optimization processes in the real environment.***
- Available in JAVA, Mathematica, MATLAB code, C/C++ and Python.

5. SOMA/PSO Additional Info

Recent Results for SOMA

- Best performing swarm algorithm in 2019
- Soma T3A: 4th place
- SOMA Pareto: 6th place
- ESP SOMA: 18th place
- Out of 36 algorithms in competition

The 2019 100-Digit Challenge on Real-Parameter, Single Objective Optimization: Analysis of Results

K. V. Price[1], N. H. Awad[2], M. Z. Ali[3], P. N. Suganthan[4]

TABLE VII.
SCORES FOR 18 PRIMARY ALGORITHMS

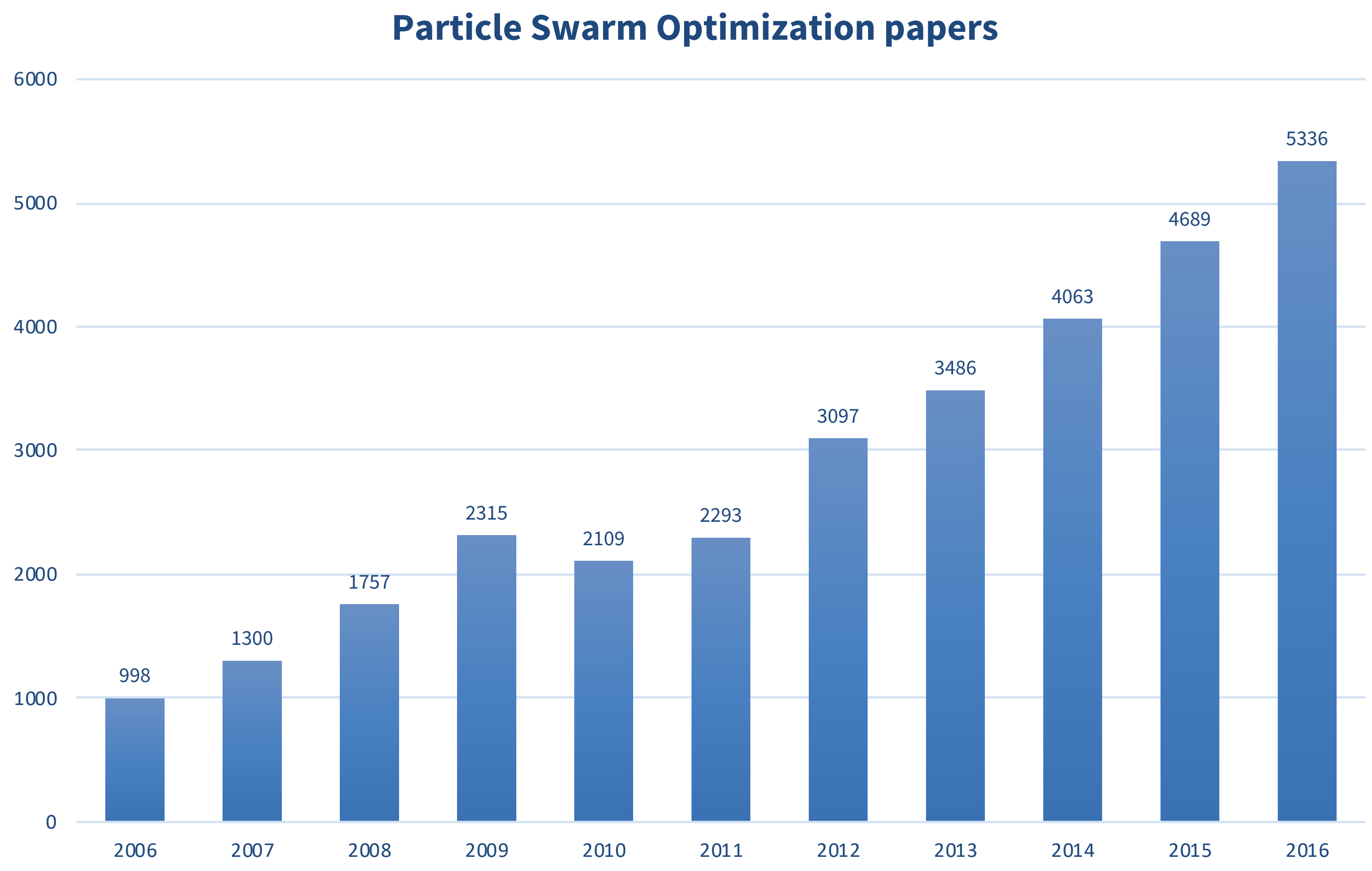
Primary Algorithm[REF]	Score by Function										Total Score
	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	
jDE100[10]	10	10	10	10	10	10	10	10	10	10	**100**
DISHchain1e+12[21]	10	10	10	10	10	10	10	10	7.12	10	**97.12**
HyDE-DF[19]	10	10	10	10	10	10	10	10	3	10	**93**
SOMA T3A[23]	10	10	10	10	10	10	10	10	3	10	**93**
ESHADE-USM[12]	10	10	10	10	10	10	10	2	3.52	10	**85.52**
SOMA Pareto[22]†	10	10	10	10	10	10	10	2	3.04	10	**85.04**
rCIPDE[9]	10	10	10	10	10	10	10	2	3	10	**85**
Co-Op[15]	10	10	10	10	10	10	9.56	1	4	10	**84.56**
DISH[7]	10	10	10	10	10	10	10	1.92	2	10	**83.92**
rjDE[24]	10	10	10	10	10	10	10	1.52	2	10	**83.52**
mL-SHADE[16]	10	10	10	10	10	10	5.04	1	2.16	10	**78.2**
GADE[18]	10	10	10	10	9.44	10	1	2	3	10	**75.44**
CMEAL[13]	10	10	10	10	10	10	1.4	1	1.04	10	**73.44**
HTPC[11]	10	10	10	10	10	10	0.28	0.04	3.04	10	**73.36**
UMDE-MS[6]	10	10	10	6.4	10	10	1	1	2	10	**70.4**
DLABC[17]	10	0	3.88	10	10	10	10	2	2	10	**67.88**
MiLSHADE-LSP[8]	10	10	10	2.92	10	5.2	0	0.6	2	10	**60.72**
ESP-SOMA[14]	9.68	0	4.12	10	4.08	10	1	1.04	2	10	**51.92**
AVERAGE	**9.9822**	**8.8888**	**9.3333**	**9.4067**	**9.64**	**9.7333**	**6.6267**	**3.2844**	**3.2178**	**10**	**80.1133**

† Data were extracted from multiple tables to show the results when population sizes are tuned.

PSO application areas (WoS)

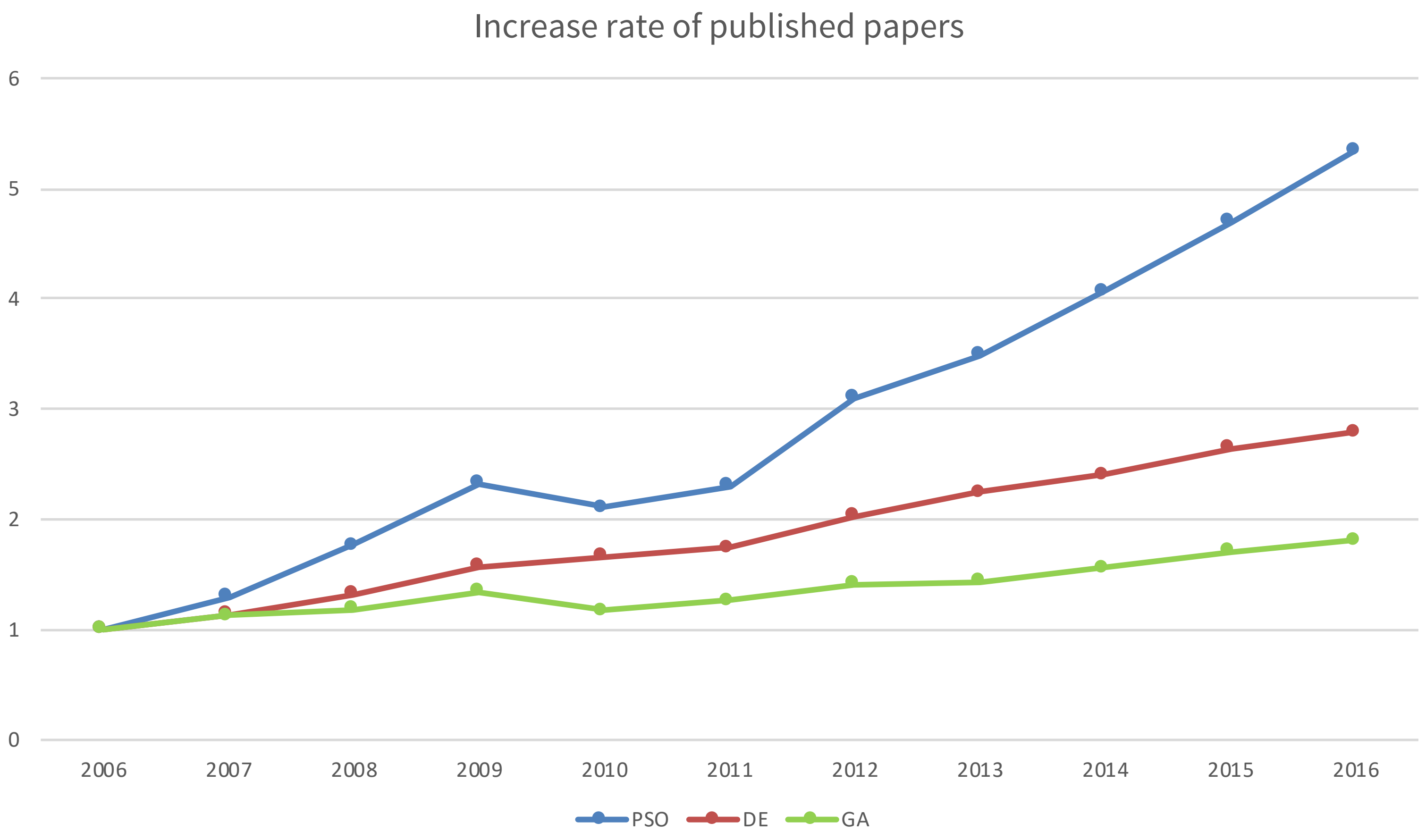
- ENGINEERING (18,723)
- COMPUTER SCIENCE (16,568)
- AUTOMATION CONTROL SYSTEMS (3,804)
- ENERGY FUELS (2,613)
- TELECOMMUNICATIONS (2,387)
- OPERATIONS RESEARCH MANAGEMENT SCIENCE (1,889)
- MATHEMATICS (1,874)
- PHYSICS (1,276)
- MATERIALS SCIENCE (1,238)
- SCIENCE TECHNOLOGY OTHER TOPICS (1,121)
- MECHANICS (969)
- ROBOTICS (933)
- INSTRUMENTS INSTRUMENTATION (783)
- MATHEMATICAL COMPUTATIONAL BIOLOGY (769)
- THERMODYNAMICS (641)
- CHEMISTRY (601)
- OPTICS (600)
- IMAGING SCIENCE PHOTOGRAPHIC TECHNOLOGY (491)
- ENVIRONMENTAL SCIENCES ECOLOGY (382)
- BUSINESS ECONOMICS (344)
- WATER RESOURCES (327)
- REMOTE SENSING (286)
- TRANSPORTATION (267)
- CONSTRUCTION BUILDING TECHNOLOGY (254)
- GEOLOGY (242)
- ELECTROCHEMISTRY (155)
- BIOCHEMISTRY MOLECULAR BIOLOGY (139)
- METALLURGY METALLURGICAL ENGINEERING (130)
- MEDICAL INFORMATICS (120)
- ACOUSTICS (112)
- NEUROSCIENCES NEUROLOGY (111)
- GEOCHEMISTRY GEOPHYSICS (106)
- NUCLEAR SCIENCE TECHNOLOGY (105)
- BIOTECHNOLOGY APPLIED MICROBIOLOGY (97)
- RADIOLOGY NUCLEAR MEDICINE MEDICAL IMAGING (96)
- EDUCATION EDUCATIONAL RESEARCH (93)
- AGRICULTURE (82)
- PHYSICAL GEOGRAPHY (64)
- LIFE SCIENCES BIOMEDICINE OTHER TOPICS (62)
- SPECTROSCOPY (61)
- PHARMACOLOGY PHARMACY (56)
- INFORMATION SCIENCE LIBRARY SCIENCE (54)
- OCEANOGRAPHY (51)
- MARINE FRESHWATER BIOLOGY (50)
- METEOROLOGY ATMOSPHERIC SCIENCES (47)
- MINING MINERAL PROCESSING (43)
- FOOD SCIENCE TECHNOLOGY (39)
- ASTRONOMY ASTROPHYSICS (37)
- POLYMER SCIENCE (30)
- RESEARCH EXPERIMENTAL MEDICINE (30)
- HEALTH CARE SCIENCES SERVICES (27)
- MATHEMATICAL METHODS IN SOCIAL SCIENCES (23)
- BIOPHYSICS (22)
- GEOGRAPHY (21)
- SOCIAL SCIENCES OTHER TOPICS (21)
- GENETICS HEREDITY (16)
- PSYCHOLOGY (12)
- PUBLIC ADMINISTRATION (12)
- PUBLIC ENVIRONMENTAL OCCUPATIONAL HEALTH (12)
- CRYSTALLOGRAPHY (10)
- FORESTRY (10)
- MEDICAL LABORATORY TECHNOLOGY (10)
- CELL BIOLOGY (9)
- TOXICOLOGY (9)
- MINERALOGY (7)
- ONCOLOGY (6)
- MICROBIOLOGY (5)
- MICROSCOPY (5)
- SURGERY (5)
- URBAN STUDIES (5)
- GENERAL INTERNAL MEDICINE (4)
- REHABILITATION (4)
- SOCIAL ISSUES (4)
- BIODIVERSITY CONSERVATION (3)
- COMMUNICATION (3)
- ENDOCRINOLOGY METABOLISM (3)
- FISHERIES (3)
- DENTISTRY ORAL SURGERY MEDICINE (2)
- EVOLUTIONARY BIOLOGY (2)
- HISTORY PHILOSOPHY OF SCIENCE (2)
- ANATOMY MORPHOLOGY (1)
- ANESTHESIOLOGY (1)
- ARCHITECTURE (1)
- ART (1)
- ARTS HUMANITIES OTHER TOPICS (1)
- CARDIOVASCULAR SYSTEM CARDIOLOGY (1)
- GERIATRICS GERONTOLOGY (1)
- OBSTETRICS GYNECOLOGY (1)
- OPHTHALMOLOGY (1)
- PHILOSOPHY (1)
- PHYSIOLOGY (1)
- PLANT SCIENCES (1)
- PSYCHIATRY (1)
- SPORT SCIENCES (1)
- VIROLOGY (1)

PSO – the popularity



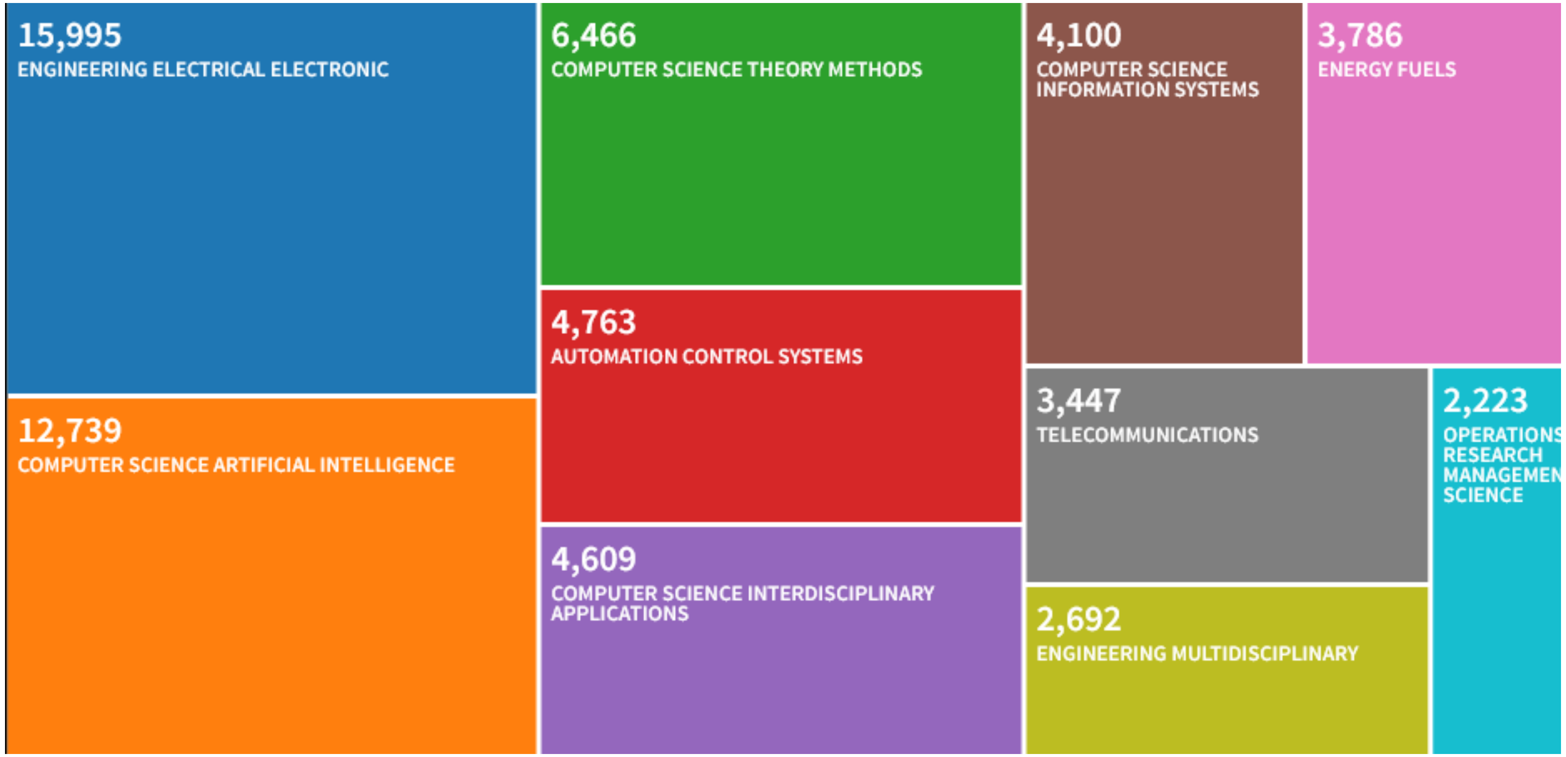
*Web of Science Core collection

Comparison of publication increase rate: PSO vs. "Classic"



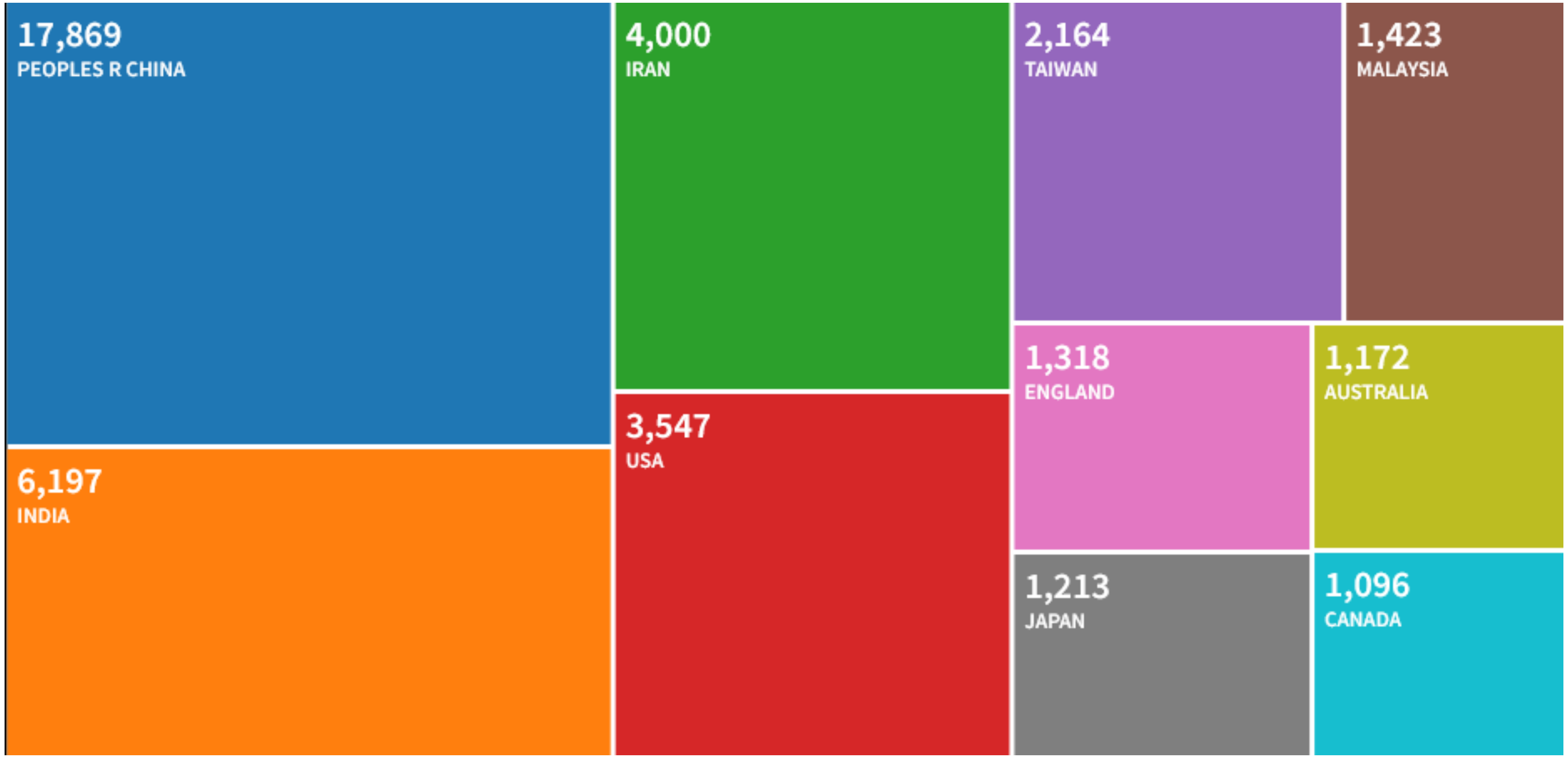
*Web of Science Core collection

PSO WoS categories (total: 46,378)



*Web of Science Core collection

PSO – by region (total: 46,378)



*Web of Science Core collection

References

[1] Eberhart, R. C., Shi, Y., & Kennedy, J. (2001). Swarm intelligence. Elsevier.

[2] Bonabeau, E., Dorigo, M., Marco, D. D. R. D. F., Theraulaz, G., & Théraulaz, G. (1999). Swarm intelligence: from natural to artificial systems (No. 1). Oxford university press.

[3] Kennedy, J. (2006). Swarm intelligence. In Handbook of nature-inspired and innovative computing (pp. 187-219). Springer, Boston, MA.

[4] Blum, C., & Merkle, D. (2008). Swarm intelligence. Swarm Intelligence in Optimization; Blum, C., Merkle, D., Eds, 43-85.

[5] Kennedy, J., & Eberhart, R. (1995, November). Particle swarm optimization. In Proceedings of ICNN'95-International Conference on Neural Networks (Vol. 4, pp. 1942-1948). IEEE.

[6] Nepomuceno, F. V., & Engelbrecht, A. P. (2013, June). A self-adaptive heterogeneous pso for real-parameter optimization. In 2013 IEEE congress on evolutionary computation (pp. 361-368). IEEE.

[7] Zhan, Z. H., Zhang, J., Li, Y., & Shi, Y. H. (2010). Orthogonal learning particle swarm optimization. IEEE transactions on evolutionary computation, 15(6), 832-847.

[8] Liang, J. J., Qin, A. K., Suganthan, P. N., & Baskar, S. (2006). Comprehensive learning particle swarm optimizer for global optimization of multimodal functions. IEEE transactions on evolutionary computation, 10(3), 281-295.

References

[9] Lynn, N., & Suganthan, P. N. (2015). Heterogeneous comprehensive learning particle swarm optimization with enhanced exploration and exploitation. Swarm and Evolutionary Computation, 24, 11-24..

[10] Riget, J., & Vesterstrøm, J. S. (2002). A diversity-guided particle swarm optimizer-the ARPSO. Dept. Comput. Sci., Univ. of Aarhus, Aarhus, Denmark, Tech. Rep, 2, 2002.

[11]

[12] Zelinka, I. (2016). SOMA—self-organizing migrating algorithm. In Self-Organizing Migrating Algorithm (pp. 3-49). Springer, Cham.

[13] Kadavy, T., Pluhacek, M., Senkerik, R., & Viktorin, A. (2019, June). The Ensemble of Strategies and Perturbation Parameter in Self-organizing Migrating Algorithm Solving CEC 2019 100-Digit Challenge. In 2019 IEEE Congress on Evolutionary Computation (CEC) (pp. 372-375). IEEE.

[14] Diep, Q. B., Zelinka, I., Das, S. & Senkerik, R. (2019, July). SOMA T3A for Solving the 100-Digit Challenge. In 7-th Joint International Conferences on Swarm, Evolutionary and Memetic Computing Conference (SEMCCO 2019) & Fuzzy And Neural Computing Conference (FANCCO 2019), Maribor, 10-12 July 2019.

References

[15] T.C., Diep, Q. B., Zelinka, I. & Senkerik, R. (2019, July). Pareto-based Self-Organizing Migrating Algorithm solving 100-Digit Challenge. In 7-th Joint International Conferences on Swarm, Evolutionary and Memetic Computing Conference (SEMCCO 2019) & Fuzzy And Neural Computing Conference (FANCCO 2019), Maribor, 10-12 July 2019.

[16] Kadavy, T., Pluhacek, M., Viktorin, A., & Senkerik, R. (2020, July). Self-organizing migrating algorithm with clustering-aided migration. In Proceedings of the 2020 Genetic and Evolutionary Computation Conference Companion (pp. 1441-1447).

EUROPEAN UNION
European Structural and Investment Funds
Operational Programme Research,
Development and Education

MINISTRY OF EDUCATION,
YOUTH AND SPORTS

Thank you for attention

Strategic project of TBU in Zlín, reg. no. CZ.02.2.69/0.0/0.0/16_015/0002204