*Available online at* www.joac.info

ISSN: 2278-1862

# Journal of Applicable Chemistry

**2015, 4(6): 1597-1690**

(International Peer Reviewed Journal)

**Cutting edge technology algorithms (Ceta)**

# Evolution of Mimics of Algorithms of Nature (E-man) Part 6: Research Tutorial on bat and Mosquito algorithms

**K RamaKrishna[1] andR Sambasiva Rao[2*]**

1. Department of Chemistry, Gitam Institute of Science, Gitam University, Visakhapatnam, 530 017, **INDIA**
2. School of Chemistry, Andhra University, Visakhapatnam 530 003, **INDIA**

Email: karipeddirk@gmail.com, rsr.chem@gmail.com

Accepted on 9thNovember2015

(Dedicated to Dr L S A Dikshitulu, former professor of chemistry, Andhra University on his 80th birth anniversary)

---

## CONSPECTUS

*Background:The processes in nature are comprehended mainly as those in living (animate) creatures and lifeless (non-living) entities (objects). The latter are broadly sub classified into physical-, chemical-, and geological- astronomical-interactions. The broad characteristics of life are foraging for food, digestion, reproduction, shelter seeking, off-spring care, defense against predators etc.Amidst the life threats, harsh surroundings, calamities and eco-imbalance, species adapt to the changing natural/man-made scenarios and survive. However, when it is out-of-proportion, the population of the species diminish slowly and extinct finally.*

***Nature inspired algorithms:*** *Computations with numerical data is familiar in formal education, but computing with words is of recent thrust area. From the stand point of a deeper level of comprehension, nature does perform computations; need not be in the same fashion we do with numbers. The miracles of human brain and animal behavior inspired invocation of neural networks in 1940s. The mapping of Darwin's postulates of genetics into genetic algorithm is a mile stone in nature inspired computations. PSO (particle swarm optimization, ACO (ant colony optimization), honey bee foraging, group of lions hunting prey are cornerstones of swarm intelligent approaches. SAA (simulated annealing algorithm) speaks of impact of annealing of glass, a chemical process and intelligent water drop/gravity/ big bang big crunch algorithm were invoked from keen research of physical processes. Now there are around fifty nature-inspired mathematical optimization tools. The different types of classifications and a bird's eye view of them are described. The frame consisting of heuristics, meta-heuristics and hyper-heuristics is sought after in this domain.*

*E-man_ToolBox: It operates in classical and user chosen and research modes. Each of NI (nature inspired) - algorithms in basic/advanced modes can be run using built-in-prefixed sequences of operations. In the research mode, a new approach, an existing one with untried options is available for an exploratory study. An exhaustive search for all possibilities is time consuming, but is easily possible with smaller test data. The structure of the toolbox includes algorithm specific parameters in default, user chosen or those available in reported literature database. The NI_algorithm independent operations like iteration, fitness function, convergence, solution evaluation are kept separate. The initiation, movement in m-D space,*

*fitness evaluation, and best position selection are programmed as functions using IF-Then-else logic and operators for an instance. The modifications and advances are developed as functions/operators.*

***Bats in nature**: Bats live in groups in their roosting crevices. It is the only mammal with wings. Microbats exhibit echolocation to detect prey for food and also way back to their home. Although a large number of bats move, apparently there is no swarm intelligence in their activities. Evolution made them to transmit ultra sound of 20-250 KHz frequency with an amplitude (loudness) using a definite emission rate and the bursts last for a very small time. It waits for the reflected echo from a prey/obstacle and analyzes the wave. The bat has the ability to distinguish the prey from background even in darkness. The microbats use not only echolocation technique, but also vision and other senses to locate food. It changes the loudness and emission rate when it approaches the prey. Even frequency is a variable and it matches with the size of the food preys. The frequency decreases soon after they find a prey and subsequently approach them slowly.*

*Artificial bats:The echolocation process is translated into mathematical model. The fixed frequency of bat emission is assumed. Each virtual bat flies randomly with a velocity vi at position xi. As it searches and finds its prey, the emitted frequency, loudness and pulse emission rate of ultrasound are changed. Selection of the best continues until certain stopping criteria are met. The translation of biological features into mathematical domain is only partial and some details are ignored. This basic bat algorithm is improved in these five years to a noteworthy state by hybridizing with fuzzy logic, PSO, GA, mutation operators, chaos etc. The data representation ranged from binary to floating point and quaternion formats. The exploration and exploitation of search space is imbedded in this model in right proportion. The bat algorithm is designed for classification, single/multiple optimization criteria and refinement of weights of neural networks as well as the architecture. Bat algorithm is added to E-man tool kit and the operators for velocity, frequency, pulse emission rate, loudness are developed for textual display and numerical computation.The noteworthy hyper heuristic is COBRA (Co-Operation of Biology Related Algorithms) making best use of positive features of algorithmsParticle Swarm Optimization (PSO), Wolf Pack Search Algorithm (WPS), Firefly Algorithm (FFA), Cuckoo Search Algorithm (CSA) and Bat Algorithm (BA).*

***Mosquitoes in nature:** The natural evolution enhanced the survival of mosquitoes in widely varying environments and under harsh surroundings. As one of the measures of mosquito control, the sperm of male mosquitoes is deactivated through chemicals in recent advanced man-made intervention. A sterilized male mosquito produces only semen, but not the sperm. Thus, even if it mates with a female mosquito, the latter does not become pregnant.*

*Mosquito-host-seeking-process in nature: Natural mosquitoes detect a host by heat/ odorand move in radical orbits towards the host for a blood meal.*

*MosquitoHostSeekAlg:The metaheuristic, 'mosquito-host-seeking-algorithm', is through the inspiration of local interactivity, parallelism and self-organization of movement of natural mosquitoes towards the human target. The terminology of translated biological processes into mathematical paradigm include binary values for sex of mosquito, attraction/ radial distance/ gray values for artificial mosquitoes from host, computing cell to perform parallel computations, traversing path, shortest distance etc. The pseudo code, matlab-modules and heuristics in if-then-else format are critically discussed. MosquitoHostSeekAlg is used to solve travelling sales man task with 110 to 510 cities using one, eight and sixteen parallel nodes. It is also used in quality assurance task. The efficiency of algorithm is compared with ACO, SAA, SOM and EN. The future scope of this coveted algorithm is incorporated.*

*Egg laying of female mosquitoes in nature:Female mosquito exhibits intelligence to locate a site to lay eggs.*

*Delay of egg hatching in unfavorable surrounding conditions:*
*Astonishingly, the in situ eggs are blessed with built in knowledge to detect favorable surroundings, otherwise they delay the hatching process and await for suitable environment.*

*Mosquito-oviX- optimization algorithm:With inspiration from the highly selective behavior of female mosquitoes in choosing a habitat to lay their eggs and the inhibition of those eggs to hatch into the next stage, Mosquito-oviX- optimization metaheuristic algorithm is proposed. It is used to find global optima of multi-dimensional test functions.MosquitoHostSeekAlg and Mosquito-oviX are also now an integral part of E-man_ToolBox.*

**Keywords:** Multi-object-functions, Nature-inspiring algorithms, E-man, Bat algorithm, Echolocation, Mosquito-oviposition-site/egg-hatching (Mosq-oviX-eggHatch), bio-process-knowledge, swarm intelligence, bio-nspiration.

---

## Bat and mosquito algorithms in OmniMetricS

## II. Mosquito algorithm (6)

## INTRODUCTION

The macro- and micro- processes in biology, physics, geology, chemistry, psychology or social behavior is a perineal inspiration to mathematicians in particular in the recent past and scientists in general. In a nut shell, origin of universe, its evolution and future plight are imbibed in all these processes. Biology continues to be a brilliant preacher with proficiency even in construction of efficient structural systems. They surpass man-made construction for thousands of years, and the recent trend is to mimic nature in unique civil constructions.

The fact base accumulated/validated by biologists during last one century,inspired mathematicians to at least partially translate major concepts into mathematical heuristics. This is an adventurous choice to resort to adapt a new unknown world in solving real life numerical tasks leaving aside the firm base of theoretically sound and also trodden path of mathematics and/or statistics. It started with translation of Darwin's genetics to GA (genetic algorithm) by Goldberg (1975) and functioning of brain into neural networks (1943). The stochastic simulating annealing algorithm is a mapping of annealing of glass/molten metals (1963). The basic format these algorithms are reshaped astoundingly and they guide the development of next generation algorithms and are reviewed under the category of E-man[223-226]. The swarm i.e. large number of species is the basic philosophy of multi-agent approaches enabling parallel search against conventional single agent iterative solution methods. The human senses, instruments extract information from thephysical systems under noisy and partially observable environment. It enables to build an internal representation of external world varying in time and space.

Bio-inspired algorithms are modern optimization tools developed to solve complicated design problems in science, engineering and technology. This new tool aims to speed up the multi-objective and NP-complete optimization processes expanding the scope into tougher optimization regions.

Nature does computing also in addition to many other chores. It evolved in time-space continuum under extreme energy (temperature)/density and at present, the entire scenario reflected in every species. Thus, computing is now realized as a characteristic property of nature. But, it is a consequence of trillions of generations of species over millions of years combating the odds and optimizing resources at natures' pace. Humans, being higher order species of nature, they also perform computation in multitude of ways. An in depth as well as breadth wise extensive research for over a century in physic-chemical-, biological sciences resulted in heuristics, laws, theorems of micro-/macroscopic processes. The application of microscopic laws to macro-systems and vice versa promoted a march towards a unified paradigm and evolving hybrid forms. This experimental science used the probes of instruments and rigor of mathematical statistical approaches in extraction of knowledge/intelligence.Solution of Mathematical/statistical /fuzzy/chaotic equations involve computing with paper and pencil in yesteryears even for a toy task. Now, software, hardware, multi-node/blade architectures, parallel algorithms/languages, distributed /cloud-computing relieved the drudgery of manual calculations and paved way to develop intelligent algorithms for solving computationally difficult tasks. In an attempt to understand moving nearer to Mother Nature, the-state-of-technology the subsection of rat brain is simulated on a computer system with silicon chips mimicking biological neurons.

The traditions in model development from physico-chemical-biological processes, solution methods, validation protocols, interpretations although change at a slow pace, most of them have gone to the stage of mind-set or firm(-hard)wired in the annals of science. The present review deals with state-of-knowledge of bat, mosquito algorithms and bird's eye view of nature-inspired algorithms [1-229] in general.

Nature inspired mathematical tools: The nature inspired mathematical methods started over half a century ago. Now, they are classified as heuristic, meta-heuristic and hyper-heuristic approaches. These channels (approaches) do not require gradient/ Hessian information or error distributions of measurement. Further, they are devoid of assumptions of hard statistical models. Yet, they arrive at global solutions efficiently even for NP-complete multi-object-optimization tasks [1-90, 179-229, Appendix_NIA-1]. The one limitation however, is that the solution may not be true (in true sense), but serves the purposes in many real life micro- to mega- tasks. The recent view is that who want and in what context the really true solution of a complicated issue, when it is impossible to arrive at exact solution even after many years of CPU time. One should also bear in mind that since last two centuries, the solutions for (theoretical, empirical) models are with lot many assumptions regarding model parameters, noise/error in variables. Many approaches like exact solution of approximate equation and/or approximate solution of exact equation are implicit for experts and are not even on the back of the brain for routine practitioners

The impetus for nature inspiring mathematical heuristics is from biological species and physical/ chemical/ geological/ astronomical systems in nature. Does this sought after inspiration has a worth noting consequence or just a passing phase? Does the end-product --algorithms, software, robots -- is panacea or become complimentary/supplementary/ essential add-ons to the existing long cherished mathematical tools? Will the loose-coupling at the moment evolve into an integrated close woven paradigm with hitherto unforeseen and unknown features? The first level answer is basic version of each algorithm in this bandwagon surpassed time tested mathematical/ statistical procedures. A natural question arises;is it possible to combine major advantages of these algorithms and try to develop a potentially better set of algorithms with sequential, parallel, fused, hierarchical structure enabling adaptive distributed auto-expert-mode cloud computing independent of software/hardware barriers.

The comparison between algorithms is thus more complicated than ever in practice.Generally, a new algorithm is compared with others mostly basic versions and reports that proposed one is superior. Even in the context of inter-comparison of a set of programs, a mindset dominates over reality. Unless the latest version or hybrid ones are used for that algorithm the real scenario is masked. This is only a caution for software professionals developing machine learning techniques and for a healthy sustenance of the best of best methods for the next decade.

One more point worth not forgetting is that no model is final and continuous evolution is the law of science rather than exception. This is akin to the popular slogan 'change is law of nature'. With the experimental evidence of theoretically proposed of boson (Nobel Prize 2013), Standard Model of particle physics is complete. But, the mass of neutrino (Nobel Prize 2015), dark matter/dark energy now open new vistas to upgrade it. The futuristic science doctrines will be rewritten with these innovations making it more bugs free.

Heuristics: A rule of thumb arrived at during inductive solution of a task. It is a term popular in computerscience, artificial intelligence, and mathematical optimization. The levo and dextro rotation behavior of asymmetric carbon atom is a classic example in organic chemistry. The heuristics learnt empirically or invoked through intuition are susceptible for pitfalls(chart 0.1).

Chart 0.1: High lights and limitations of Heuristics

- Heuristics
- Only alternative for NP-Hard optimization/classification problems
- Results in good approximate solution for refinement
- used along with optimization algorithms for a fast solution

+ Fast compared to classical methods

− classic methods fail to find any exact solution for complex tasks
Remedy : Heuristic approach

− Partial sacrifice of optimality, completeness, accuracy, and/or precision for speed.

− Solution may be sub-optimal

+ The only way to find a solution in viable CPU time

Remedy: Hybrid methods

| Solutions | Heuristic | Goal |
|---|---|---|
| Many | Does it guarantee the best? | Is the best is required always? |
| Confidence interval | Does it give confidence interval?<br>Is the error bar very high? | Will it perturb the entire project? |
| CPU time | Which is the best heuristic? | Inferior heuristic with low CPU time?<br>Superior heuristic with very large CPU time?<br>Option for the task on hand |

TSP is a NP-complete task. The optimal solution is not tractable for even a small number of cities. The heuristic approach like greedy search is used for a good start and this may not be optimal one at all. This method picks up whatever is best in the current step. It may preclude good solutions at a later stage of iteration. The numerical experience votes for a good enough solution, while theoretical probe predicts better solutions even on quantitative scale.

Metaheuristics: A higher level heuristic developed to select set of heuristics which enable solution of a taskwith limited/incomplete/imperfect information. The functional domain of metaheuristics is the problem search space. In other words, meta heuristics strategies (viz. simple search procedure to complicating learning procedure) consider samples of sets of solutions. Glover, in 1986, used the term metaheuristics for the first time. There is a functional similarity between biological processes and nature inspired meta-heuristic algorithms.

The solution process with metaheuristics is an iterative procedure guided by child (sub ordinate) heuristic rules. Here, exploration and exploitation segments in search space, learning strategies are sewed to arrive at near-optimal solutions. The positive features are avoidance of getting stuck in local optima and if at all trapped, they escape and proceed further. The range of techniques differs widely starting with simple local searches to complicated learning strategies (chart 0.2).

| Chart 0.2: Limitations of Meta heuristics |
|---|
| Meta heuristics : [tabu search, simulated annealing, ant algorithms ; genetic algorithms] |
| + Avoids local minima |
| – Cannot probe into heuristic search space<br>Remedy: Hyper-heuristics |
| – No promise to arrive at globally optimum solution |
| – Many metaheuristics involve stochastic procedures |
| – Solutions depends upon random variables at that instance |

Hyper-heuristics: It is an off-the-shelf iterative heuristic search method controlling/ implementing heuristics with cited noteworthy characteristics (chart 0.3). It couples machine learning procedures into algorithms enabling automation of adaptive selection/ combination/ generation of heuristics and meta heuristics. This frame is broad permitting to solve sets if classes of tasks rather than a single targeted task. It is a hierarchical paradigm wherein high-level method selects lower level implementable heuristics.

| Chart 0.3: Positive features of Hyper heuristics |
|---|
| Hyper-heuristics [tabu-search hyperheuristic; Case Based Heuristic Selection Method |
| ✈ Keeps track of the non-problem-specific data [fitnesFnValue change, execution time] |
| ✈ Operate on the search space of heuristics instead of candidate solutions. |
| + Retain strengths of each heuristic and compensates weaknesses |
| + Select appropriate/ adequate method/ set of heuristics for a task on hand |
| + Developed with minimum domain knowledge for search process |
| + Increase generalizability |
| + Knowledge poor heuristics |
| + Searches for most suitable lower level heuristics |
| + Hyper-heuristic acts as a heuristic scheduler over a set of heuristics by deterministic or a non-deterministic mode |

| modifications by hill climbers | | |
|---|---|---|
| $$ $ Hill Climbing Algorithm | | Set of modifications |
| DBHC | Davis' Bit | |
| NDHC | Next Descent | Inversion of a bit |
| RBHC | Random Bit | A bit is selected randomly<br>Inverted for a number of |

| Mutational heuristics | | |
|---|---|---|
| SWPD | Swap Dimension | randomly choosestwo different dimensions in a candidate solution<br>swaping operation |
| DIMM | Dimensional Mutation | randomly chooses a dimension<br>inverts all bits in this |

| | | | | | |
|---|---|---|---|---|---|
| | | iterations | | | dimension (prob: 0.5) |
| SDHC | Steepest Descent | Checks each single bit inversion variant of input candidate; Accepts one with best improvement. | HYPM | Hyper-mutation | inverts each bit in the candidate solution (prob: 0.5) |

Nature inspired mathematical tools

## 1. Bats in nature

There are about one thousandand odd different species in bats and they account for up to 20% of all mammals. The bats are exceptional in having wings.

Size: The size and weight of bats range widely. The tiny bumblebee bat weights around 1.5 to 2g, while giant bats are of 1 kg with wingspan of about 2m. Most microbats are insectivores and the length of forearm is about 2.2 to 11cm.

Sense organs:Echolocation is a natures' boon to bats' life to sense (in our perception measure) distance and also to differentiate a prey to hunt for its food from background barriers even in darkness.But, how they acquired the skill/knowledge in biological evolution is a still a potential area of research. Most of species of bats are very sensitive to smell and some have good eyesight. In navigation and prey detection, they use all the senses in combination resulting in maximized efficiency.Microbats are endowed with bliss in making use of time difference between their two ears to map 3D-information. Thus, they detect and identify the type of moving insect against background disturbance even in dark.

Echolocation:Microbats emit a very loud sound pulse and waits to listen echo coming from surrounding objects after bouncing. The properties of pulses depend upon hunting strategies. The variation in band width of signal is species dependent. Further, they avoid obstacles during navigation and locate their roosting crevices in the dark. Most bats use short, frequency-modulated signals covering an octave. The others employ constant-frequency signals in echolocation.

Higher frequencies of sound travel shorter distances as the corresponding wavelengths are short. A frequency range of 20 kHz to 500 kHz corresponds to 0.7mm to 17mm in wavelength scale frequency wavelength. The typical ranges in microbats bats are around few meters. The size of prey is around the frequency coverage of bat produced ultrasonic sound. The audible frequencies to human being, animals, insects and the range of frequencies employed in medical diagnosis and treatment are briefly described from literature reports in appendix-Bat-1.

Movement: The bats fly randomly from the position (x) with a velocity vel. The frequency of sound is a fixed value initially at minimum frequency (freqmin). But, they can automatically adjust frequency, loudness and pulse emission rate depending upon proximity of prey during detection and hunting of preys.

## 2. Translation of (natural) bat echolocation processes into intelligent (artificial) bat-mimicking algorithm for mathematical optimization

Xin-She Yang proposed bat algorithm in 2010 inspired by echolocation of microbats in nature to locate small insects for food. It is another metaheuristic method from this school and expanded the band wagon of nature-inspired mathematical algorithms, a new discipline entering into golden jubilee year. Bat procedure is a swarm intelligence optimization. The implicit biological echolocation process in bats is translated into a mathematical form based on laws of echo-experiments. The movement of bats when they are far off from prey maps to exploration (global search) in mathematical search space. The lowering of loudness and increasing pulse emission rate as bat approaches the prey corresponds to exploitation or local search.

### 2.1 Artificial (virtual) bat (Art.Bat, AB; or Virt.Bat, VB)

The simplification of artificial bat is that no ray tracing is used in estimating time, as it is computationally intensive in multi-dimensional optimization. Of course, this concept is a beneficial feature in computational geometry. Further, the variation of loudness is from a large positive loudness0 to a minimum constant value of loudnessMin. An artificial bat uses a frequency-tuning technique to control the dynamic behavior of a swarm of bats. The balance between exploration and exploitation is controlled by tuning algorithm-dependent parameters viz. pulse emission rate, frequency and loudness.Chart 2.1 depicts one-to-one correspondence of activity of natural bat versus algorithmic steps in nature inspired bat algorithm.

**Chart 2.1: Panoramic view of echolocation of bats in natures versus model bats**

| Bats in nature | | Artificial bats | |
|---|---|---|---|
| | | | |
| **Foraging** | **Prey hunting** | Optimization | Finding global optimum |
| **Locating home in dark** | **Roosting crevices** | Classification | Number of classes |
| | | | |
| | | | |
| **Bat flying** | **In real environment in dark** | Movement | In search space |
| Echolocation | | Simplified (artificial) mathematically translated echo-driven equations | |
| High loudness &<br>Small pulse emission rate | | Moving in large steps | Global search |
| Low loudness &<br>Large pulse emission rate | | Moving in small steps | Local search |
| | | | |
| Judgement whether prey is far away or very nearer | | Intelligent (knowledge based) process<br>🕮 detection of how far current solution is from optimum &<br>🕮 switching of global to local search | |

### 2.2 Position of bat

**Chart 2.2: Positions of bats in search space**

| | User given data | |
|---|---|---|
| X = randU ([LLT,ULT],nbats) | #bats (solns) | nbats (or nsolns) |
| | LLT | Lower limit |
| | ULT | Upper bound |

The co-ordinates of bats on the search space refer to position of the $i^{th}$ solution in $j^{th}$ dimensionat current ($it^{th}$) iteration (chart 2.2). The tracing of third order X tensor of floating point values during iteration maps the footsteps of the algorithm towards the optimum. Taha et al. [153] represented the position of bat in a binary string of length equal to number of features (Nfeatures). The binary 1 corresponds to the selected while 0 to that unselected feature. Fister et al. used quaternion $\left(a+i*b+j*c+k*d\right)$ for position of bats and quaternion algebra for navigation.The heuristics promote refinement of position of bats. A global best bat (or global best position) is calculated at each iteration. The number of bats is a user given integer and the X is initiated by random number generator. The upgradation of positions is discussed in 2.14 under movement of bat.

### 2.3 Object (Fitness) function (objFn, FitnessFn)

An object function is task dependent i.e. different for classification, optimization etc. It is also based on type of error minimization / outcome maximization and approaches like Bayesian. Mostly many tasks solved in the last century are single object function based. Later multiple objectives are transformed into a single function. However, recent interest is in multiple object function optimizations (MOOs) with conflicting objectives. Alihodzic and Tuba [130]used entropy based object functions in bat algorithm (Eqns. 2.A).

The numerical value obtained by substituting X and algorithm specific parameters at current iteration in the object function is called object function value (Eqn. 2.A). It is the heart of optimization /classification algorithm to arrive at global/local extrema within the desired level of accuracy. The vector is sorted in ascending order and sets of best and worst objFnValue vectors with respect to a single object function and corresponding X tensor are stored.

**Eqn. 2.A: Single and multiple object functions**

$$objFn = fn\left(\left[x1,...,xd\right]^T;\left[par;\text{constants}\right]\right) \quad \text{Eqn. 2.1}$$

$$MultiObjFn = \left[objFn_i\right] = \left[objFn_1, objFn_2, \ldots\ objFn_{no\_objFns}\right] \quad \text{Eqn. 2.2}$$

| | | |
|---|---|---|
| $SOF$ | : | single *object function* |
| $objFn_i$ | : | $i^{th}$ $SSO$ |
| $no\_objFns$ | : | *number of objFns* |

| Conversion of MultiObjFn into SOFn | Notation |
|---|---|
| $objFn\_mul2\text{sing}$ | |
| $= MultiObjFn^T * wt$ | Tensor |
| $= \left[objFn_1, objFn_2, objFn_3\right] * \begin{bmatrix} wt_1 \\ wt_2 \\ wt_3 \end{bmatrix}$ | Vector product |
| $= \begin{bmatrix} wt_1 * objFn_1 + wt_2 * objFn_2 + \\ wt_3 * objFn_3 \end{bmatrix}$ Eqn. 2.3 | Algebraic |

$$wt = \left[wt_1, wt_2, \ldots\ wt_{no\_objFns}\right]^T$$
$$sum(wt) = 1; \quad \text{Eqn. 2.4}$$

$$wtClasAccu = randu([0,1]) \quad ;default = 0.9$$
$$subsetLength = 1 - wtClasAccu$$
$$FitnessFn\_class(i,it) = wtClassAccu * prob\left(Yj \middle| X\right) + subsetLength * \frac{NoFeat - NoFeatSelect}{NoFeat}$$

| | | |
|---|---|---|
| | : | Number of solutions |
| $P(YJ\|X)$ | : | classification accuracy |
| *NoFeat* | : | total number of all features |
| *NoFeatSelect* | : | number of selected features |

$$objFnValue(m,it) = objFn\left(\left[X(i,j,it); bat\_par(k,it)\right]\right)$$

$$objFnValueAsc = sortAsc\left(objFnValue(1,it)\right)$$
$$[row,col] = size\left(objFnValueAsc\right)$$
$$bestSet = objFnValueAsc\left(col : col - p, 1\right)$$
$$worstSet = objFnValueAsc\left(1 : q, 1\right)$$

2.4 Frequency of ultrasound wave: The initial values of frequencies are bats are from uniform random number (randu) generator. The linear scale has chances of better feedback information. Yet, different patterns of variations in frequency are in vogue (chart 2.3). It controls the pace and range of the movement. It is similar to particles in PSO.Xie et al. [151] proposed calculation of frequency adapting DE/best/2 strategy. The frequency outside the set limits is autocorrected (KB. 2.1, chart 2.3).

**Eqn. 2.B:** Frequency of emitted sound by artificial bats

$$Frequency(i,it) = \text{randu}\left(\left[fmin, fmax\right]\right)$$

| | |
|---|---|
| $\beta_1, \beta_2$ | $randu \in [0,1]$ |

Chart 2.3: Different numerical limits in literature for frequencies of bat emitted ultrasound

| | Freqmin | freqMax |
|---|---|---|
| 13-abs | 0 | 2 |

KB. 2.1 : Auto correction of frequencies outside the user specified limits

| | | | | |
|---|---|---|---|---|
| | | | If | Freq i >freqMax |
| If | Value of freq outside allowed limits | | Then | Freqi = freqMax +delta |
| Then | Replace with a limit value | | If | Freq i < freqMin |
| | | | Then | Freqi = freqMin +delta |

## 2.5Loudness

The scale of loudness is either 1 to 100 or 0 to 1. In each iteration, it is changed as in Eqn. 2.C. The initial values are sometimes taken in the range [1 to 2].

**Eqn. 2.C: Loudness in artificial bat sound**

$$loudness(i,it+1) = \alpha * loudness(i,it)$$

| | |
|---|---|
| $0 < \alpha < 1$ | $\gamma > 0$ |

| | | | |
|---|---|---|---|
| | | Assumption | $\alpha = \gamma$ |
| | | Default | $\alpha = \gamma = 0.9$ |

| Loudness parameter- Evolution | | |
|---|---|---|
| $$A^{(t+1)} = \begin{cases} A_{lb}^{(t)} + \text{rand}_0 \left( A_{ub}^{(t)} - A_{lb}^{(t)} \right) & \text{if } \text{rand}_1 < \tau_1, \\ A^{(t)} & \text{otherwise,} \end{cases}$$ $$r^{(t+1)} = \begin{cases} r_{lb}^{(t)} + \text{rand}_2 \left( r_{ub}^{(t)} - r_{lb}^{(t)} \right) & \text{if } \text{rand}_3 < \tau_2, \\ r^{(t)} & \text{otherwise.} \end{cases}$$ | | |

| | |
|---|---|
| $\tau 0$ and $\tau 1$ | learning rates |
| $\tau 0 = \tau 1 = 0.1$, | |
| rand$ii$ = 1 ⋯ 4 | randu[0, 1] |

Classification: In a classification task employing Naive Bayes algorithm, loudness is calculated empirically as number of features divided by five. The Value for maximum loudness is dynamic function of number of features for some datasets (KB. 2.2).

| KB. 2.2 : Reduction of features in classification by artificial bat | | | | |
|---|---|---|---|---|
| If | LoudnessMax = 3 & LoudnessMin =1 | | If | Bat is closer to the prey |
| Then | # features reduced from 3 to 2 | | Then | #features = 1 |

2.6 Pulse emission rate: This parameter dictates the onset of a local search operation around the global best bat solution (KB. 2.3). The rate of emission of pulses changes as the bats fly towards the prey. The initial value is around zero or from a uniform random number in the range [0 to 1]. The number zero means no pulses at all, and one corresponds to the maximum rate of pulse emission. The different proposals in literature deserve intensive research in overall performance of bat algorithm.

Thus, loudness and pulse emission rates automatically control and auto-zoom the region with promising solutions or successful prey hunt.

**KB. 2.3: Variation of pulse emission rate with progress of prey hunting**

| Meta rule | |
|---|---|
| If | Bat is moving towards prey (i.e. position of bat in current iteration improved) |
| Then | Refine loudness & Pulse emission rate |

| | |
|---|---|
| If | Higher pulse rate |
| Then | Prob (conducting a local search around the global best ) is low |
| Else | Prob (conducting a local search around the global best ) is high |

| | |
|---|---|
| If | Bat found prey |
| Then | decreases loudness & rate of pulse emission increases |
| | Amin = 0 means that a bat has just found the prey and temporarily stop emitting any sound |
| If | $it \to \text{large}(i.e.\infty)$ |
| Then | $PulseEmis\_rate\ (i,it) \to PulseEmis\_rate0$ & |
| | $loudness(i,it) \to 0$ |

| Eqn. 2.D: Pulse emission rate - Evolution | |
|---|---|
| $PulseEmis\_rate\ (i,it) = PulseEmis\_rate0 * [1 - \exp(-const([0,1]) * it)]$ | $PulseEmis\_rate\ (i,it) = PulseEmis\_rate(i,it-1) * \left(\frac{it}{fixPar}\right)^3$ |

2.7 Velocity of bat: The increment in velocity of i[th] bat is related to frequency (**KB. 2.4**).

$$\boxed{vel\_inc(i,it)) = \lambda_i * freq_i}$$

**KB. 2.4 : Heuristics in refining velocity, loudness, pulse rate emission rate, and position during**

| (a) | |
|---|---|
| if | vel(i) > velMax |
| Then | vel(i) = velMax |
| | |
| if | vel(i) < velMin |
| Then | vel(i) = velMin |

| (b) | |
|---|---|
| if | epsi * loudness(i,it) > loudnessMax |
| Then | epsi * loudness(i,it) = loudnessMax |
| | |
| if | epsi * loudness(i,it) < loudnessMin |
| Then | epsi * loudness(i,it) = loudnessMin |

| (c) | |
|---|---|
| if | x(i,it) > = xBestGlob(it) |
| Then | xc = x(i,it) |
| Else | xc = xBestGlob(it) |
| | |
| if | Rand < loudness(i,it) & objFnValue( xc ) < objFnValue(xBestGlob(it)) |
| Then | Accept soln xc |
| | |
| if | Accept |
| Then | Increase pulse emission rate & Decrease loudness |

| (d) |
|---|
| pulseEmsRate = pulseEmsRateInit * [ 1 - exp(-gama *it)] |
| loudness(i,it) = alpha * loudness(i,it-1) |

2.8 Movement of bat:

Artificial bats navigate by using time delay from emission to the reflection.

**Eqn. 2.E: Refinement of frequency, velocity and position of bat**

| Frequency calculation | Updating velocity |
|---|---|
| $freq(i,it) = freqMin + ( freqMax + freqMin ) * \beta$ | $vel(i,it) = vel(i,it-1) + [x(i,it-1) + xBestGlob(it)] * freq(i,it)$ |

Updating position X

| Updating position X | | | |
|---|---|---|---|
| $x(i,it) = x(i,it-1) + vel(i,it)$ | | | |
| Nsol | : | Number of solutions | |
| beta | : | Uniform random vector [0, 1] or Normal random vector [0, 1] | $\beta$ |
| *xBestGlob* | : | Current global best location (solution) which is located after comparing all the solutions among all the n bats | $X_*$ |

| | |
|---|---|
| $\left[x(i,it-1) + xBestGlob(it)\right]$ | Difference between position of global best bat and that of ith bat |

| Refinement of position of solutions in bat-algorithm | |
|---|---|
| $xnew = xold + rand * freq(i,it)$ | $randu \in [-1,1]$ |
| | $\langle Freq(i,it) \rangle$: Average loudness of all bats at current iteration |
| | $Freq(i,it) = \langle Freq(i,it) \rangle$ |

| **Position X - Evolution** | |
|---|---|
| $x(i,it) = xbest + ScFac * Loudness(i,it) * randn([0,1])$ | $ScFac$ : Scaling factor for Loudness |

| | |
|---|---|
| If | Randu([0,1]) > pulseEmisRate(i,it) |
| Then | $x(i,it) = xbest + \langle Loudness(i,it) \rangle * randn([-1,1])$ |

2.9 Current best solution: The refined solutions in that iteration (it < itMax) for all bats are calculated and sorted in the order of magnitudes of objFnValue tensor. The sets of best and worst solutions (or swarm of bats) are stored for further use or for expert system based monitoring of the progress during iterations for a deeper levelpostmortem of failure or passive natureof algorithm.

2.10 Local search: One of the current best solutions is chosen for carrying out local search. A random walk is performed for an intensive local search with appropriately tuned loudness and pulse rate (KB)of bats. It results in generating a new solution for each bat based their positions.

2.11 Acceptance of current solution: The heuristics for acceptance or rejection of the refined solution are in KB. 2.5.
It is a quality control of iteration procedure checking the progress in the right direction.

| | **KB. 2.5: Heuristics for acceptance of refined solution in the iteration process** |
|---|---|
| if | (randn([0,1]< Loudness(i,it) & |
| | objFn (xi) < objFn (xbestGlob)) |

Then Accept new solutions as current values

$$x(i, j, it)$$

$$objFnValue\left(x(i, it)\right)$$

Else

$$reject\ current\ values\ and$$

$$retain\ previous\ iteration\ best\ ones$$

$$x(i, j, it-1)$$

$$objFnValue = objFn\left(x(i, it-1)\right)$$

end if

2.12 Exploration and exploitation: A simple way is to examine the search exhaustively with in grid of desired accuracy of solution. It is practicable for linear and quadratic equations in one variable and that too when integer solutions are the target within a small range (-5 to +5 or at the most -10 to +10). In all other even tiny problems, real life tasks in science, engineering and commerce, well established root finding, optimization procedures are in practice. Gradient based approaches have tremendous success over one century in research. When the gradient is not easy to calculate, direct search methods were proposed. In statistical experimental exploratory analysis of optimum operating conditions, even function relating response and assumed causal factors are not known. Here empirical models like full quadratic model, neural networks are the choice. Simplex optimization and its advances also find a place in this pursuit. In global and/or multi-objective optimization with multiple constraints with pathological (in mathematical sense) profiles of response, error function/object function, nature inspired algorithms found a niche, which superseded most of yester years mathematical procedures. Appendix NIA-2depicts a brief roadmap of the evolution of these procedures.

Exploration refers rough search in large intervals in each dimension to search for the presence of an extremum and is also known as global search/haunt/wide-scale or telescopic view. On the other hand, either from a priori knowledge, results of exploration, intensive/ small scale/microscopic inspection is the population local search or referred as exploitation in the sense that eagle' view of presence of prey is hunted in close area. The simile is going around bush.

In bat algorithm, loudness Ai and the rate ri of pulse emission are heuristically changed during the iterations to provide an effective mechanism to control the exploration and exploitation and switch to exploitation stage at appropriate stage.

2.13 Input data: The input (chart 2.4) to the bat algorithm is through an ASCII file, GUI or in the interactive expert system driven mode. The object function, algorithm specific parameters, general optimization constants and task (optimisation, classification etc.) specific data are inputted to the algorithm.

Chart 2.4(a) Input & parameters

| | |
|---|---|
| ○ Input data | ☏ Task specific [optimization, classification, image analysis ] |
| ○ Optimization | ☏ Maxit, Convergence criteria |

Chart 2.4 (b) : Bat echolocation specific parameters

| | |
|---|---|
| pulse emission rates at xi | PulseEmis_rate (i,it) |
| loudness | loudnss Ai |
| pulse frequency | freqi at xi |

| | | | | |
|---|---|---|---|---|
| | relevant parameters | | | |
| ○ | Objective function | ☎ | Built in, user chosen | |
| ○ | Initialization of algorithm specific free parameters | ☎ | Frequency, Loudness, pulse emission rate, | |

| | |
|---|---|
| Maximum no of iterations | MaxIt |
| Convergence | Accracy_X |
| CPU time | MaxCPU_minutes |

| **Alg. 2.1: Initialization** | |
|---|---|
| Position of bats | x |
| velocity | vel |

- User choice or default values: The number of bats, (approximate solutions), maximum number of iterations are user's choice. But, otherwise default values are software driven.
- Algorithm specific parameters: The frequency, loudness, pulse emission rates and their variation during the entire process are also user chosen depending upon the task and desired accuracy. The values used in different published reports serve as a guideline.
- Initialization:The positions of bats (i.e. values of approximate solutions in d-dimensional search space), frequency and velocity are initiated (Alg.2.1).

2.14 Bird's eye view of basic Bat algorithm: To start with,the fitness function (objFn) is calculated with the approximate values of solution and algorithm specific parameters. It follows the refining frequency of each bat, velocity and position, pulse emission rate etc. The position best bat among them is selected. This corresponds to exploration or global search. The prospects on the finer grid around the global best position are carried out and it is referred as local search or exploitation. The results of test for acceptance of this solution is performed indicate whether the current best is in the correct direction to approach the optimum solution. If so, the loudness is decreased and pulse emission rate is increased. Now the bats are ranked and position of current best bat is calculated. The iterative process is continued until convergence criteria are met or time out condition (maximum number of iterations and /or CPU hours) is reached. The Alg. 2.2 incorporates stepwise operations of basic bat algorithm.

| Alg. 2.2: Basic bat algorithm | |
|---|---|
| Input & parameters | Chart 2.4 |
| Cal ObjFnvalue with initial conditions | Eqn. 2.B |
| Initialization | Alg. 2.1 |

| |
|---|
| % Algorithm detects the most successful solution as *x*best before starting iterative search process |
| |
| eval = evaluate the new population<br>*f*min = find best solution(xbest); |

(a) Major steps

| | |
|---|---|
| Iteration | Begin |
| | %% Bat alg. Begin %% |
| | % Exploitation OR Global search |

| | |
|---|---|
| (c ) Pseudo code | |
| while | it <maxIt \| ~converged |

| | |
|---|---|
| Variation operation<br>Best solution<br>% Exploration OR Local search<br>Best solution<br>Convergence test<br>% Movement of bat in the search space<br>Replacement operator<br>%% End Bat alg. %%<br>End Iteration | Adjust frequency<br>Generate new solutions<br>Updating<br>Velocities<br>Locations/solutions<br>if Rand([0,1])> ri<br>%improve the best solution(xbest)<br>Cal best set of solutions<br>Select one among them<br>Generate a local solution around selected best solution<br>end if<br>random flight of bat<br>calculate new solution<br>%simulated annealing<br>if (randn([0,1]< Loudness(i,it) & freq(xi)< freq(xbestGlob))<br>Accept new solutions<br>Increase pulseEmissRate(ri) and reduce Freqi (Ai)<br>end if<br>$objFn_{min}$ = find the best solution($\mathbf{x}_{best}$);<br>Rank bats<br>calculate current bestxbestGlob<br>end while |
| | Output |

2.15 Positive features and Limitations of bat algorithm:Since frequency of sound transmitted, pulse emission rate dictate movement of bat to reach the optimum location, it can be thought of as a frequency-tuning algorithm. The variation or tuning of frequency is similar to a parameter in PSO/ harmony search or temperature in SAA. They drive towards regions of promising solutions and also help in resorting to exploitation from exploration. This is one positive feature of bat metaheuristic. The one-one mapping of processes in nature and artificial bat are briefly depicted in chart 2.5. The inherent advantages and shortcomings with remedial measures of basic bat algorithm are given inchart 2.6.

**Chart 2.5: One-to-one mapping of bat algorithm to mathematical optimization**

| Nature Inspired (NI). Bat alg. | | Math. Optimization alg. | |
|---|---|---|---|
| | Distribution | | Random |

**Chart 2.6: Remedial measures for limitations and inherent + points of bat alg.**

| | |
|---|---|
| Positive characteristics | + Balanced combination of exploration and exploitation.<br>+ Quick convergence at a very initial stage by switching from exploration to exploitation |

| | | | |
|---|---|---|---|
| | of bats | | positions of approximate solutions |
| flight | | Iterate | |
| | | | Movement to a new position |
| | Transmission of ultra-sonic sound<br>Analysis of echo from the prey | | Exploration |
| | | | Finding global best |
| | | | Refined new position |
| | Position Velocity of bat | | Exploitation |
| end | flight | | Local search |
| | | | New global best position |
| End | | End | iteration |

| | |
|---|---|
| | + Acceptable results for lower dimensional problems<br>+ Pulse emission rate controls exploration and exploitation<br>+ Exploitation capability results in good intensification |
| **Limitations** | − Varying frequency and pulse emission ratetoo quickly (i.e. fast switching to exploitation stage) leads to stagnation after some initial stage<br>Remedy: [Genetic operators, PSO] |
| | − Difficult for multimodal optimization problems<br>Remedy:Pareto front |
| | − Poor results for high dimensional and/or hard problems<br>Remedy: Self adaptation of control parameters<br>⊥ Injection of problem specific knowledge in the form of local search |
| | − algorithm loses exploration capability with progressof iterations<br>Remedy: inserting differential operators for crossover andmutation |
| | − A few solutions get stuck in some local minima<br>Remedy:ABC algorithm |
| | − Poor in balancing exploitation and exploration in certain tasks/datasets<br>⇒ Local optima<br>Remedy: Bat with double-subpopulation |

## 3. Applicationsof bat algorithm

Bat algorithm and its advanced versions find applications in science, engineering and mathematics. Typical diverse tasks studied include classification, fuzzy clustering, function approximation, parameter estimation in dynamicbiological systems, multi-objective optimization, image matching, economic load and emission dispatchproblems, data mining, scheduling problems, neural networks, and phishing website detection.

### Chemistry

Nuclear reactor core: Kashi et al. [143]developed bat Algorithm Nodal Expansion Code (BANEC) and used for optimizing loading of fuel in nuclear reactor core. The multiple object functions viz. maximizing multiplication factor and minimizing power peaking factor are simultaneously solved and the software system is tested with two PWR test cases including KWU and BIBLIS reactors.

Fuel cell: The high accuracy in performance of simulated proton exchange membrane fuel cell was hampered due to error in parameter estimation. Turgut and Coban [94]modelled optimal proton exchange membrane fuel cell is with hybrid DE and Teaching Learning algorithm (chart 3.1).

**Chart 3.1: Methods of comparison for Fuel cell performance**

- ✓ DE + Teaching Leaning >>
  - Melody Search (MS)
  - Backtracking Search (BS)
  - Artificial Cooperative Search (ACS)
  - Quantum behaved PSO (QPSO)
  - Bat algorithm (BAT)
  - Intelligent Tuned Harmony Search (ITHS)
  - Cuckoo Search (CS)

## Chemical biology

Gene expression dataset: Mishra and Mishra [110] analyzed gene expression data with hybrid bat and functional link NN method (chart 3.2, Alg. 3.1). It excelled recent similar procedures in accuracy. Recent interest in classification/pattern recognition is around ensemble of learning techniques which are also known as classifier fusion.

**Chart 3.2: Classification of gene expression data**

| Classifiers | Methods of Comparison |
|---|---|
| ▶ Particle Swarm Optimization-Functional Link Neural Network (PSO-FLANN),<br>▶ Bat inspired-Functional Link Artificial Neural Network (BAT-FLANN)<br>▶ Support Vector Machine (SVM), | → UniformVoting<br>→ Distribution Summation<br>→ Dempster–Shafer<br>→ Entropy Weighting<br>→ Density based Weighting |

*Alg. 3.1 : Bat*-Functional Link Artificial Neural Network

| | | |
|---|---|---|
| for | *i*=1:NoOfIteration | |
| | for | *j*=1: *n* |
| | | read *Dj* |
| | | compute each *batj* frequency as *fj* |
| | | compute class(object) distance from *batj* as *S objectj* |
| | | compute for each class update |
| | | compute the new position *Pj* and change pulse rate controller *c*1 of *Batj* |
| | end | |
| | | calculate the error and update *wt* |
| end | | |

| | |
|---|---|
| *delwt = wt* | |
| *fj= c*1 * Mean(*Dj*) | |
| *Sobject j =fj * Dj * delwt* | |
| *Ej= Sobject j –* 1 | |
| *wt = wt-2*μ*Ei* | |
| if | *Ej<Ej*-1 |
| | *Pj = Pj + Ej*<br>*c*1 = *fj + c*2 * *Pj* * *Ej*2 |
| end | |
| err(*i*)=Mean(*E*) | |
| *wt = wt- delwt* | |

## Chemical technology

Ethanol Production: Farias Jr. et al. [136]modeled the data the fermentation process. The extreme learning machine combined with bat algorithm is a viable alternative to standard soft sensor approach using MLP_NN (chart 3.3).

**Chart 3.3: Bat +Extreme learning machine for fermentation in ethanol manufacture**

| Chemical processes | Prediction |
|---|---|
| ☞ Alkaline hydrogen peroxide pre-treatment<br>☞ Hydrolyzed sugarcane bagasse<br>☞ Fermentation of a mixture of molasses | • Concentration of ethanol,<br>• Substrate cells from secondary measurements<br>○ pH, turbidity,<br>○ $CO_2$, temperature |
| | [Bat + ExtLrnMach] > NN |

## Energy

### Fuzzy + bat alg.

Using fuzzy systems and bat algorithm has shown a reliable match between prediction and actual data for energy modelling.

## Pattern recognition

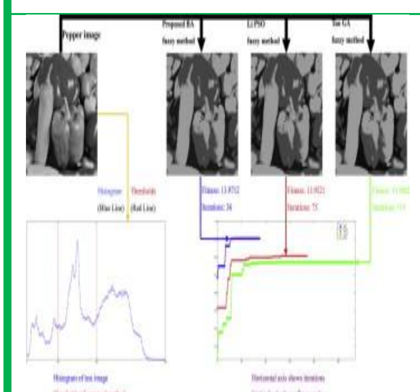
### Classification/Discrimination

Feature selection: Rodrigues et al. [137] solved feature selection as an binary-valued optimization problem. The wrapper feature selection approach with maximization of Optimum-Path Forest (OPF) accuracy over a validating set as the fitness function is used employing bat algorithm. The quality of reduced feature set is assessed with six public datasets. The feature selection approach leads to classification model as simple as possible, optimizes classifier's predictive capability and looks after curse of dimensionality hurdle.

### Image analysis:

Maximizing fuzzy entropy with bat algorithm:Ye et al. [101]applied bat algorithm to maximize fuzzy entropy in modeling natural and infrared images (Fig. 3.1). This method surpassed when ABC, GA, PSO, ACO are used instead of BAT metaheuristic. Image threshold is formulated as a constrained optimization task.

Fig. 3.1: Image segmentation with entropy maximization using bat alg.



### Minimum path distance or shortest route

**TSP:** Travelling sales man problem belongs to minimum distance task. The distance between two adjacent nodes/cities can be coded in terms of time delay. The microbats use time difference between their two ears to obtain three-dimensional information. Thus, bat algorithm is a apt in solving TSP for large number of cities.

### Medical diagnosis

Breast cancer data: Mishra et al. [162, 110] used different microarray datasets for classification of cancer using bat-NN, PSO and NN models (chart 3.4).The hybrid bat algorithm classified with more than 90%accuracy while other methods performance is less than 70%.

| **Chart 3.4: Microarray assay of breast cancer** | | |
|---|---|---|
| Data set | NP x Nfeatures | |
| | Raw data | PCA |
| Breast Cancer | 98 * 1213 | 98 * 97 |
| Lung Cancer | 197 * 581 | 197 * 81 |
| *StJudel Leukemia* | *248 * 985* | *248 * 91* |
| | | |
| 1-11 | *1* | |
| 12-62 | *2* | |
| *others* | *3* | |

Artificial pancreas design patient-specific: Kirubakaran et al. [138] estimated Hovorka–Wilinsk (H–W) model parameters from virtual patient data employing chaotic bat metaheuristic. The designed artificial pancreas is tested for efficient elimination of hypoglycemic danger. The multiple empirical (second-order plus delay time) models for glucose–insulin dynamics are analyzed by k-means clustering and implicit ones are used in predictive controllers (mpMPCs). The design of on board insulin safety trigger is designed using estimated H–W model results with multiple full-order linearized chaotic bat algorithms in fuzzy logic domain.

### Engineering

Bat algorithm has been used for engineering design,[10]

### Electric power systems

Electrical engineering: Guerraiche et al. [120] applied 'directed bat algorithm' for optimizing non-linear mathematical model of multi-state series–parallel power system with constraints. The minimization of investment cost for selecting appropriate components from power system is mapped as resolving redundancy optimization problem. The modified bat algorithm is 5-10% more efficient compared to ABC algorithm.

Design of passive power filters: Yang and Le [103]proposed bat algorithm with Pareto front for the design of passive power filters(PPFs). Introducing common characteristics like single-tuning, second-order, third-order, and C-type damping in passive filters suppresses critical harmonics and improve power factor.

Electric power generating company's activity: proposed a model for bidding price against bidding quantity in constrained electricity market. The information about rivals here is incomplete. The task is modelled as a bi-level optimization challenge. The higher level component maximizes GENCO's payoff and lower level task finds solution for market clearing based on maximization of social welfare. The nature inspiring algorithm is enhanced bat echolocation procedure.

Optimal placement of capacitors in radial distribution systems: Injeti et al. [118]achieved optimal placement of capacitors on standard 34 and 85 bus in radial distribution systems with bat algorithm [chart 3.5].

| Chart 3.5: Placement of capacitors in distribution systems | |
|---|---|
| Objectives<br>**1)** Minimization ( real power loss )<br>**2)** Maximization ( network savings ) | Methods compared<br>⊥ Particle Swarm Optimization (PSO)<br>⊥ Harmonic Search (HS),<br>⊥ Genetic Algorithm (GA),<br>⊥ Artificial Bee Colony (ABC),<br>⊥ Teaching Learning Based Optimization (TLBO)<br>⊥ Plant Growth Simulation Algorithm (PGSA), |
| optimal placement of fixed size of capacitor banks<br>(Variable Locations Fixed Capacitor banks-VLFQ)<br><br>optimal sizing and placement of capacitors<br>(Variable Locations Variable sizing of Capacitors-VLVQ) | |

Batteries and thermal energy storage: Ikeda and Ooka [122]resorted to nature inspired metaheuristic tools for peak load shifting energy systems with an objective of less computational time consumption (chart 3.6).

| **Chart 3.6: Storage of batteries and thermal energy** | |
|---|---|
| Meta heuristics | |
| Genetic algorithms<br>Particle swarm optimization<br>Cuckoo search<br>Differential evolution<br>Mutation-pso<br>Cuckoo search<br>Bat_self-adaptive learning | + m-PSO and cuckoo search advantageous over Dynamic programming<br><br>Mathematical programming methods<br>+ Theoretically optimal solution<br>− Computationally time consuming |
| Optimization of operating schedules of energy systems<br>Battery<br>TES<br>Air-source heat pump | + m-PSO was the fastest<br>+ cuckoo search most accura |
| | Cuckoo search<br>Semi-optimal solution<br>135 times faster than Dynamic programming |

**Structural Engineering**

Structural optimization: Hasançebi et al. [160] investigated the performance of bat algorithm in truss structure design for minimum weight with constraints like stress and stability based on AISC-ASD (American Institute of Steel Construction-Allowable Stress Design).

Taper cutting in WEDM process: Nayak and Mahapatra [92] applied multiple tools for taper cutting in WEDM (wire electrical discharge machining process) process. Taguchi experimental design with factors viz. part thickness, taper angle, pulse duration, discharge current, wire speed and wire tension is used to obtain optimum values. The multiple performance characteristics were transformed into an equivalent single performance by maximum deviation theory. Thus, overcomes inherent limitation of Taguchi design in simultaneous optimization of performance characteristics.

Makespan in a Flow Shop: Pugazhenthi and Xavior [147]modeled flow shop scheduling in modern manufacturing unit with bat algorithm. Using reverse engineering procedure, minimal make span is achieved (chart 3.7).

Renewable energy: Kavousi-Fard and Niknam [144] reported the application renewable energy systems with self-adaptive modified bat optimization algorithm (chart 3.8). The reliability of the system is studied by effect of renewable energy sources on the reliability of the power system and wind power.

Multilevel thresholding task: Kiran [109]investigated the relative efficiency of bat algorithm and several other metaheuristic procedures with large (twenty ) set of bench mark functions in two to five dimensions and multilevel thresholding tasks (chart 3.9).

Scheduling of high speed trains: Zheng [112][BAT-59] reported application of a newly developed water wave optimization for real life scheduling of high speed trains in China and diverse set global optima for diverse bench mark functions. The computational results are compared with bat and other nature inspiring algorithms (chart 3.10).

Typical functionstestedwith Bat algorithm are incorporated in chart 3.11. Miscellaneous applications in of ba heuristic in diverse tasks are tabulated (table 3.1).

Chart 3.7: Flow shop design with GA+bat

→ Task
→ Permutation flow shop design
→ Model
ॐ GA + Bat >> [ CDS NEH heuristics];

Chart 3.8: Multi-object renewable energy task

| Task : | Multi-object Function |
|---|---|
| ✡ optimization<br>✓ discrete,<br>✓ nonlinear<br>✓ non-convex | • System Average Interruption Frequency Index<br>• Average Energy Not Supplied<br>• Total active power losses<br>• Total network cost |

**Chart 3.9: Comparison of bat with tree-seed and other metaheuristics**

ॐ Tree-seed algorithm
ॐ Artificial bee colony (ABC)
ॐ Particle swarm optimization (PSO),
ॐ Harmony search (HS)
ॐ firefly algorithm (FA)
ॐ bat algorithm (BA)

**Chart 3.10: High speed train scheduling efficiency comparison of bat, invasive weed etc. alg.**

ॐ Invasive weed optimization
ॐ Biogeography-based optimization
ॐ Bat algorithm
ॐ ...

| Water wave optimization | | |
|---|---|---|
| Water waves in nature<br>Propagation<br>Refraction<br>Breaking | → | Nature Inspired Water waves alg.<br>+ Easy to implement<br>+ Small-size population<br>+ Few control parameters<br>+ High-dimensional solution space |

| Chart 3.11: Test functions |
|---|
| Griewangk |
| Rastringin |
| Rosenbrock |
| Ackley |
| Schwefel |
| De Jong's sphere |
| Easom |
| Michalewicz |
| Xin-She Yang |
| Zakharov |

Table 3.1: Applications of Bat metaheuristic in diverse tasks

| Task | Ref |
|---|---|
| Classification using bat algorithm to update the weights of a Functional Link Artificial Neural Network (FLANN)<br>+ Faster than FLANN PSO-FLANN | 162 |
| Wireless sensor networks<br>CModified Cuckoo Search with MAP (MCS-MAP) algorithm<br>Firefly Optimization Algorithm with MAP (FOA-MAP) | 104 |
| Constrained economic interline power-load dispatch | 91 |
| Enhanced Bat Algorithm | |
| Design of Maximum Power Point Tracking (MPPT) control in Photovoltaic (PV) systems --> mapped as optimization problem | 95 |
| Minimization of real power losses in a power system in presence of unified power flow controller (UPFC)<br>+ Compared with GA | 117 |
| automatic generation control (AGC) of an interconnected multi area thermal system | 116 |
| Fault diagnosis (binary bat algorithm)<br>Low-speed rolling element bearing failures | 125 |
| Planning of sports training sessions | 121 |
| Load frequency control<br>Dual mode Bat algorithm based scheduling of PI controllers for interconnected power systems.<br>Applied tomulti-area interconnected thermal power system | 114 |
| Tuning of stabilizers of New England test system | 115 |

| | |
|---|---|
| (ACO, GA,Bat) | |
| Telecommunications. | 113 |
| Process planning | 111 |
| Optimal spot pricing in electricity market<br>Bat >> LP, GA | 142 |
| optimal design of Power System Stabilizers (PSSs) in a multi-machine environment<br>Bat >>GA | 141 |
| Robust tuning of power system stabilizer<br>BatAlg_ > PSO_ CPSS | 140 |
| Bi-objective inventory model<br>- single manufacturer-single vendor multi-retailer (SM-SV-MR) supply chain | 146 |
| Micro-grid operation management<br>Optimal sizing of battery energy storage | 139 |
| Real-size large steel frames under actual load and design conditions | 145 |
| Unconfined compressive strength of cement-based bricks<br>(Bat +SVR)<br>(Single objective& multi-objective) multiprocessor scheduling | 124 |

## Commerce

Stock price prediction: Hafeziet al.[100] found bat-neural network multi-agent system (BNNMAS) is superior to genetic_NN and generalized regression_NN in predicting long term DAX stock-market price. The data comprised of quarterly values for a period of eight years. It is a four layer multi-agent frame work.

## 4. Recent advances in bat algorithm research

Binary bat algorithm: A binary version of bat was reported for feature selectionin classification.

Quaternion bat alg.: Fister et al. [149] represented data in quaternion form in bat algorithm.It is applied to computational geometry and large-scale optimization problems wherein extensive geometric rotations play a key role.

Multi-objective bat algorithm:Compared to single objective optimization (SOO), multi-objective optimization (MOO) problems are more complicated and rarely unique solution is arrived at. Pareto optimality front is a sought after approach. The details of Pareto-optimality and its application in chemical technology will be published separately [228].But, often, MOO task is transformed into a single objective function and solved by traditional procedures. Xin-She Yang [37] extended bat algorithm for multi-object optimization real life problems (Alg. 4.1).

| **Eqns. 4.A: Multi-objective Bat alg.** | |
|---|---|
| Pareto optimal set in the search space | **Alg. 4.1 : Multi-objective bat algorithm**<br>for j = 1 to NP_ParetoFronts |

| | |
|---|---|
| weights | Randu([0,1]) |
| | |

+ Weights with sufficient diversity -->

+ Approximate Pareto front

| | |
|---|---|
| | Generate K weights wk ≥ 0 |
| | Form a single objective |
| | Iteraive procedure |
| | Record xbestGlob as a non-dominated solution |
| end | for |

Cloud computing: This paradigm employs shared pool of resources available on internet and exploits the benefits of high performance distributed computing. Here, scheduling is a key factor and belongs to a category of NP-hard problems. In fact, there are no algorithms giving optimal solutions with in polynomial time. Thus, one should resort to suboptimal solution within reasonable CPU time. Bat algorithm is compared for efficiency with ACO, GA, PSO and League Championship Algorithms.

Single sonar unit (SSU) alg.:Tawfeeq [02-bat] [164] mimicked natural bat sonar echolocation phenomenon in proposing an intelligent optimization algorithm with different strategies. In the first strategy, a single sonar unit with a single starting point and fixed beam length is used (Alg. 4.2).

Limitation of SSU: Random selection of algorithm specific parameters like beam length, state space or specific nature of task vitiate the progress of approximate solution towards global optimum. It is because, the selected length of the transmitted beam is too long or too short (in one or more directions). Thus, it cannot probe in to the area in where global minimum or maximum of object function profile exists.

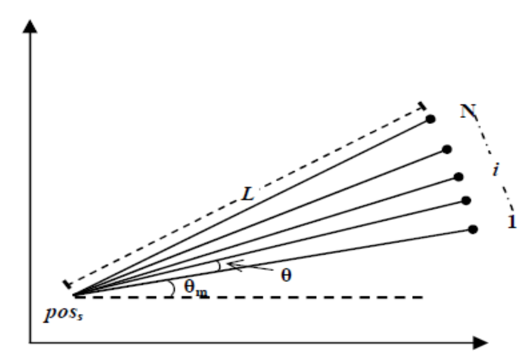
**Remedy**: Single Sonar Unit with a Momentum (SSM)

**Alg. 4.2 : Single sonar alg.**

| | |
|---|---|
| (a) | Major steps |
| | Input & parameters |
| | Cal objFn at the approximate point |
| Iteration | Until not converged |
| | Position refinement |
| | Test for convergence |
| End | Iteration |

| | |
|---|---|
| Number of beams in each single transmitted signal. randInt (smallnumber) | nbeams |
| any point in the search space selected randomly. | Starting point poss |
| Angle between beams | θ |

| | |
|---|---|
| | Method = "Fixedθ" |
| Then | randomly select a small fixed value θ between any two successive beams |
| | |
| if | Method = " Randθ " |
| Then | randomly select a different angle θi between any two successive beams, where (i=1, …, N-1). |




Courtesy of 02-bat

| |
|---|
| Initialization |
| Solution range: min, max values of the search space |
| - Beam length L: random value not exceeding half the solution range |
| Methods: [techniques "Fixedθ" and "Randθ"] |

| (c ) Pseudo code | |
|---|---|
| while | it <maxIt \| ~converged |
| | Select random value representing the main beam direction θm starting from poss |
| | Transmit N beams starting from poss with main beam direction of θm and angle θ between any two successive beams |
| | Determine the coordinates of the remote end point posi for each transmitted beam (i=1,…,N) |
| | evaluate the fitness function Fi at these ends |
| end while | |

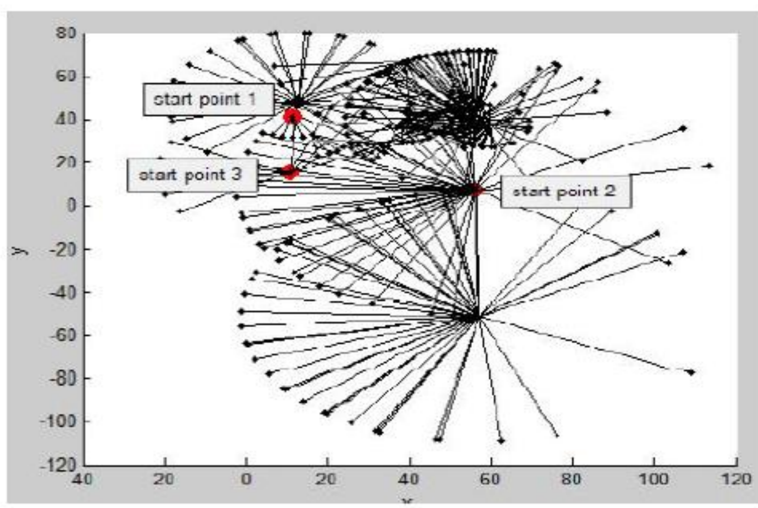
| (d) Comparision of objFn values | |
|---|---|
| | Cal Stopping criteria |
| if | Fs is the optimum value (i.e., for maximizing Fs ≥ Fi, and for minimizing Fs ≤ Fi) then go to step 3 |
| Then | Not converged |
| else | Replace the coordinates of poss with the coordinates of the optimum point of Fi and replace Fs with the optimum Fi:<br>poss = posi of optimum Fi<br>Fs = optimum Fi,<br>converged |

**Alg. 4. 2b: Single Sonar Units with A Momentum**

- %% Single sonar unit algorithm %%
- Solution is checked for local optimum
- Calculate length of transmitted beam with momentum
- $$Leng\_Beam(it+1) = Leng\_Beam(it) * (1 \pm Momentum)$$
$$0 < Momentum < 1$$

+ Reduces probability of convergence towards a local optimum.

Multi_sonar search alg.: Using core structure of single sonar search strategy, a multi-sonar heuristic is proposed (Alg. 4.3).

| **Alg. 4.3: Multiple sonar search alg.** | | |
|---|---|---|
| | Major steps | |
| For | i : number_sonar_units |  |
| | Initiate a starting position | |
| end | for | |
| | %% Search for optimum position tensor<br>Parallel search %% | |
| | Advantages<br>+ Converges for<br>o larger search space<br>o More complicated surfaces<br>+ Low CPU time | Multi(3-) sonar units<br>Courtesy from 02-bat |

Intermittent search procedure:Thebat algorithm was found to perform better than intermittent search procedure using suite of test functions and tasks from engineering literature.

Bat algorithm_Meng: Menget al. [108] improved the performance of bat algorithm by adding new features viz. habitat selection and self-adaptive compensation of natural bats. The proper selection between quantum and mechanical behavior could model habitat selection in bats. The contribution of Doppler effect in echoes is taken care by self-adaptive compensation in bats. In addition to it, self-adaptive local search is also incorporated. These new phenomenon improved the echolocation and yielded superior results for twenty bench mark test sets and four real-life engineering design problems.

Double-subpopulation variant of the bat algorithm: Jun et al. [123] introduced double subpopulation sets and Levy light in bat algorithm to combat with chances of getting trapped in local minima.

Bat algorithm with multi-population cooperation: Jaddi et al. [126] improved bat algorithm (chart 4.1) and applied to simultaneous optimization of NN architecture and connecting weights. The test data used is from classification and time series tasks.

**Chart 4.1: Bat alg + multi-population + coevolution of slave subpopulation**

- velocity equation
- Two topologies for subpopulations
  - + Increases cooperation
  - + Maintains diversity of bats in population
- Topology-1
  Combination of Ring + Master-slave mechanisms
- Topology-2
- Master–Slave strategy +
  Insertion of coevolving strategy of
  slave subpopulations

**Chart 4.2: Bat algorithm for refining ANFIT parameters**

| ANFIS parameters for refinement | Goodness of model statistics |
|---|---|
| Learning Rate (η)<br>Forgetting Factor (λ)<br>Steepest Descent Momentum Constant (α) | Root Mean Squared Error<br>Integral of Absolute Error<br>Integral of Time Multiplied Absolute Error<br>Integral of Squared Error |

**Alg. 4.4: Algorithm for bat algorithm with habitat selection**

- Add-ons-to basic bat alg.
- Model for bats' habitat selection
- Cal selection between their quantum behaviors &
- mechanical behaviors
- Bats' self-adaptive compensation for Doppler effect in echoes
-

```
For bats = 1: nbas
    Individual's difference in the compensation rate
    Echolocation characteristics
End
```

-
- Self-adaptive local search strategy

## 5. Hybrid-bat algorithm

In an attempt to enhance the capabilities of bat algorithm and render it robust and high efficient, many modules from other nature-mimicking or mathematical/statistical algorithms with positive functional advantages are incorporated in hybrid systems. This opens a futuristic fusion mode, intelligent choice of components for subtasks. In most of nature inspired algorithms, some of basic characteristics are translated into NIAs. Incorporating some more details of bio-processes in each step will enhance efficacy and may even accuracy of solutions.

Fuzzy logic + bat:Khan et al. [168] hybridized fuzzy method with bat algorithm for a clustering task. The application to ergonomic workplace problems proved the good predictive capability.

Fuzzy logic + Bat alg.: Khooban and Niknam [119] applied the hybrid Self-Adaptive Modified Bat (SAMBA) and the Fuzzy Logic (FL) algorithms for control of multi-area electric load frequency. It optimizes input/output membership functions and parameters of controller simultaneously imparting stability and robustness against extregenous disturbances and impermanent dynamics. The application to four-area interconnected power system shows the present system is superior to proportional Integral Derivative (PID) controller and Optimal Fuzzy PID (OFPID) controllers.

## Neural Networks

NN training and optimization of architecture: Svečko and Kusić [107]employed bat algorithm for optimization of number of neurons and training weights of a NN model for precise positional controls of piezoelectric actuators (PEA).

Simultaneous optimization of NN architecture and weight refining: Jaddi et al. et al. [105,126]applied modified bat algorithm for optimum architecture of NN and refinement of weights and biases. The changes in bat algorithm are introduction of personal best solution in changing velocity and three different chaotic maps. Taguchi refinement of parameters of algorithm imparted best characteristics to solution. Six classification and two time series benchmark datasets along with a real life task of prediction of rain fall data are used in this study.

Fine tuning of learning parameters of ANFIS:Premkumar and Manikandan [102] reported speed control of brushless DC motor with ANFIS (Adaptive Neuro-Fuzzy Inference System) model. GA, PSO and bat algorithm are employed in tuning of the gains of the Proportional Integral Derivative (PID), Fuzzy PID and Adaptive Fuzzy Logic Controller. The online ANFIS controller optimized by bat algorithm has superior performance compared to the other controllers.

Refining weights of Neural Network (FLANN) classifier: Mishra et al. [110,162] applied bat algorithm to refine weights of NN classifier and cancer data sets are analyzed with more than 90% accuracy.

k-means + bat alg.: The combination of k-means, a popular procedure with bat algorithm is used for efficient clustering.

PSO + MultiObjective Bat alg.:George [97] proposed hybridization of PSO with bat alg. for multi-objective optimization tasks (Alg.4.zz). In the local search, PSO is used for better accuracy and multi-objective bat algorithm is in the global updating process.

| Alg. 5.1. (a) **Multiple-objective Bat- algorithm** | | |
|---|---|---|
| For | i= 1:noObjFns | |
| | Generate K weights wk>=0 so that ΣK k=1 wk=1 | |
| end | for | |
| | Form a single objective f=ΣKK=1wkfk | |
| while | it <maxIt \| ~converged | |
| | Generate new solutions<br>Updating position,velocity, frequency,<br>pulse emission rate (Bat-algiorithm) | |
| | If | Rand([0,1])> pulseEmissionRate |
| | | Local search around global best |
| | end | if |
| | If | (randn([0,1]< Loudness(i,it) & f(xi)< f(xbestGlob)) |
| | | Acceptance of solution<br>Adaptation of frequency (KB 2.1)<br>pulse emission rate |
| | end | if |
| | | Ranking of bats<br>Current best solution xbest(i,j,it) |
| end | while | |
| | Non_dominated-Soln <-- xbest(i,j,it) | |

| **Alg. 5.1 (b ):PSO algorithm** | | |
|---|---|---|
| For | i= 1:noSolns | |
| | Cal objFn | |
| | If | current value > pbesti, |
| | Then | Current value <-- pbesti current location<----XBestGlob |
| | A particle (solution) in the neighborhood of hitherto best success is indexed as gbestGlob | |
| end | for | |

| **Alg. 5.1 (c):Multiple-objective-PSO- Bat- algorithm** | | | |
|---|---|---|---|
| Iterate | | | |
| | Initialization | | |
| | **%% PSO_operation %%** | | |
| | Form a single objective function **Eqn. 2.3** | | |
| | Repeat | | |
| | | Updatting position, velocity | |
| | | *vi ⟵ _vi + _ U(0,φ1)⊗( _pi − _xi )+ _ U(0,φ2)⊗( _pg − _xi ),* | |
| | | *_xi ⟵ _xi + _vi.* | |
| | | Until | sufficient good fitness or maxIt |
| | | | **%% MOBA search process%%** |
| | | | **%% global updating rule %%** |
| | | End | until |
| end | Iterate | | |
| | Non_dominated-Soln <-- xbest(i,j,it) | | |

Harmony search + Bat alg.: Wang and Guo [152] proposed hybridization of pitch adjustment operation of HarmSerch in bat updating heuristics (Alg. 5.2). It serves as a mutation operator and speeds up convergence. This method is tested with fourteen standard benchmark functions and the results are competitive with basic bat and other nature inspired procedures viz. ACO, BA, BBO, DE, ES, GA, HS, PSO, and SGA.

| Alg. 5.2a: Bat-Harmony-Elitism algorithm -- pseudo code |
|---|
| Parameters & Initiation |

Limitations of Basic bat alg.

- search relies entirely on random walks -->
- fast convergence cannot be guaranteed

```
while          it < maxIt  | ~converged
      For      all bats
               %%  Bat alg.          %%
               %% Harmony search %%
               %%  Elitism of GA%%
      end for    bats

end while
```

**Remedial Measures**

- Injection of problem specific knowledge in the form of local search
- Adding mutation operator

Alg, 5.2 b. Parameters & Initiation

- **Iteration dependent parameters**
- **Algorithm specific parameters**
  - **Bat echolocation**
  - **Harmony search**
  - **Elitism**

Alg, 5.2 (c ) %% Elitism algorithm %%

```
        Evaluate the fitness for the offsprings
        xtu,xtt,xvt
If       RandU < Loudness
        Xtr1 = xtk
end     if
        Replace Keep worst bats with
        KEEP_best bats stored
```

| Modifications | | Values used | |
|---|---|---|---|
| I | Fixed frequency | 0.5 | |
| | Fixed loudness | 0.95 | |
| | Pulse emission rate | 0.6 | |
| | Mutation operator | | Increase diversity of the population --> improved search efficiency fast convergence |
| | | | |
| II | Harmony memory consideration Rate | 0.95 | If randU[0.0 to 1.0]<= pulseEmisRate |

| Alg. 5.2 (d): Harmony search specific parameters | |
|---|---|
| Harmony memory consideration Rate | Harm.Mem. Consid.Rate |
| Pitch adjustment rate | Pitch.Adjust. rate |
| Bandwidth. | Bandwidth |
| Elitism specific parameters | |
| Maximum of elite individuals retained | Elite.individ. retained.Max |

| | | | |
|---|---|---|---|
| | | | Then pitch adjustment operation in HS i.e. serving as a mutation operator<br>+ Increases diversity of population<br>+ Improved search efficiency |
| | ✓ Pitch adjustment rate | 0.1 | |
| III | ✓ Elitism | | Retaining best solutions of population --> prevents corruption of best solutions by pitch adjustment operator |

$$vel(i,it) = vel(i,it-1) + \left(vel(i,it) - xbestGlob\right) * freqFix$$

$$x(i,it) = x(i,it-1) + vel(i,it)$$

Alg. 5.2 (e) %% Harmony algorithm %%

```
Harmony search algorithm parameters
Cal fitnessFnvalues for NSolns
while   it <maxIt | ~converged

    For   dim = 1:NDim%% Harmony Search
          If      randU < HMCR
          Then    Xnew(dim) = xa(dim)
                  a ∈[1,2,....,HMS]
                  If      randU < PAR
                  Then    xnew(dim) = xold(dim)  + bandWidth * (2 *
                          randu -1)
                  End     If
          else    xew(dim) = xmin(dim)  + randU* (xmax(dim) -
                  xmin(dim)  )
          endif
    End   for dim
    end   For%% Harmony search

end while
```

Naive Bayes + bat algorithm (NaiveBayes_BatAlg): Taha et al. [153] hybridized Naive Bayes algorithm with bat method for feature selection in classification task (Alg. 5.3). Here, maximum velocity is taken as equal to one third of number of features.

**Alg. 5.3 : Hybrid Self adaptive Bat alg. (Hyb.selfAdap.Bat)**

| | |
|---|---|
| Features & limitations of binary bat algorithm | **KB. 2.5: Retention of features of global best bat** |

| | | |
|---|---|---|
| Binary bat algorithm<br>− Velocity is calculated for each feature --><br>− Large CPU time<br>− Departs from philosophy of Bat alg.<br><br>**Remedy : NB-BA**<br>+ Calculates velocity once for each solution<br>+ Adaptive for feature selection task<br>+ Mimics Bat algorithm behavior<br>+ Velocity increment of ith bat $\lambda_i * freq_i$ | If | velocity of $i$th bat < = number of different bits, |
| | Then | $i$th bat will copy some features from global best bat<br>---> moving<br>towards global best bat while still exploring<br>new search space |
| | If | velocities of $i$th bat > velocity of global best bat |
| | Then | $i$th bat will import all features from the global best bat<br>[i.e. becomes same as global best bat]<br>a few different bits further facilitate exploitation |
| | | Current global best location (solution) in whole bats swarm. |
| | If | $x(i,it-1) > xBestGlob(it)$ |
| | Then | $i$th bat has less features than those of global best bat |
| | | $vel(i,it-1)+\left[x(i,it-1)-xBestGlob(it)\right]$<br>+ accelerates ith bat to move towards global best bat |
| | If | $x(i,it-1) < xBestGlob(it)$ |
| | Then | $i$th bat has more features than those of the global best bat |
| | | $vel(i,it-1)+\left[x(i,it-1)-xBestGlob(it)\right]$<br>+ *will decrease the velocity of ith bat and help to attract it closer to global best bat* |

**DE + Levy flights + bat alg.**

Xie et al. [151] proposed injecting Levy flights and differential operators into bat algorithm at different stages of foraging flight for its prey (alg. 5.4).

**Alg. 5.4: Bat + Levy flights + double**

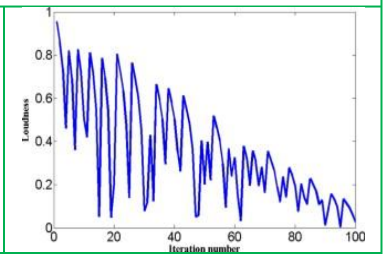
- ✡ Lévy flight bat algorithm
- ✡ Dynamic transition strategy double subgroup
- ✓ external exploration subgroup
- ✓ internal exploitation subgroup)
  - + Increased local exploiting ability of BA
  - + Improve the global exploring ability

SVR +Bat algs: Ansari and Gholami [128] found Bat alg. is superior in terms of correlation and mean square error to arrive at free parameters of SVR compared to other nature inspired methods like GA, PSO, Cuckoo search and imperialist competitive algorithms. This approach is used to develop a fused model for establishing the relation between the saturation pressure and compositional data in crude oil reservoir calculations.

SVR + Bat alg: The bat algorithm is used to refine parameters of SVM. The optimized model was applied to investigate relationships between saturated pressure and compositional data (viz. temperature, hydrocarbon and non-hydrocarbon compositions of crudes, and heptane-plus specifications of crude oils. The other optimization methods used along with SVR aregenetic algorithm (GA), imperialist competitive algorithm (ICA), particle swarm optimization algorithm (PSO), cuckoo search algorithm (CS). A thorough comparision showed that SVR_Bat alg is more robust with high performance evident from high correlation coefficient and lower MSE.

Chaotic alg + Bat: Jordehi [98] showed that chaotic functions mitigate the hurdle of convergence into local optima by basic bat algorithm by analyzing bench mark data sets (chart 5.1).

**Chart 5.1: Bat + Chaotic strategy**

| | | |
|---|---|---|
| Eleven chaotic map functions<br>Ergodicity<br>Non-repetitious profile<br>+ Diversify bats<br>+ Mitigate premature convergence problem | Bat + best_Chaotic Fn >><br>Conventional bat<br>Cuckoo search<br>Big bang-big crunch<br>Gravitational search<br>Genetic algorithm (GA) |  |



Chaotic maps +bat alg.:Gandomi and Yang [155] proposed chaotic hybrid bat algorithm.It enhanced the global search aspect and is robust.

Chaotic bat algorithm: Gandomi, Xin-She Yang et al. [134] hybridized bat algorithm with chaos and the consequence is increased global search mobility and robustness. Thirteen chaotic maps are used in four different variants of chaotic bat algorithm and the model is tested with standard bench mark test data.

Optimum-Path Forest (OPF) +bat alg.: Rodrigueset al. [156] applied a hybrid algorithm with Optimum-Path Forest (OPF) and binary bat alg. as components for feature selection task. This type of classification model optimizes performance and addresses curse of dimensionality. The results are compared by changing OPF with firefly, gravitational seach, harmony search and PSO.

Co-Operation of Biology Related Algorithms (COBRA): Shakhnaz and Eugene [150]putforward COBRA(Co-Operation of Biology Related Algorithms), a meta-heuristic algorithm which functions with cooperation of several nature inspired procedures viz. Particle Swarm Optimization (PSO), Wolf Pack Search Algorithm (WPS), Firefly Algorithm (FFA), Cuckoo Search Algorithm (CSA) and Bat Algorithm (BA). The results for 28test functions and scope for improvement are reported.

Random walk direct exploitation heuristics (RWDE) + Self adaptive Bat:Thenoteworthy modification of basic bat procedure was put forward by hybridizing with self-adaption scheme and differential operators (chart 5.5).

Self-adaptive bat alg (SelfAdaptBatAlg): The local search and also global search by tuning loudness and pulse emission rate are the two phases of a basic version of bat algorithm. The diversity of population is advantageously used in RWDE and SAA. Domain specific knowledge is of intense help to solve complicated tasks and it can be incorporated in differential evolution strategies. The local search helps to improve the best solution found by global search. Based on inspiration of functioning of self-adapting function, it is hybridized with DE resulting in jDE. It has positive operative features in continuous optimization. After the preliminary operation of finding best solution with initial population, improved bat alg. detects the most successful solution as xbest and the process proceeds for iterative search cycle.

**Chart 5.5: Features & limitations of self-adaptive bat algorithm**

SelfAdaptBatAlg

− No incorporated domain-specific knowledge

**Remedy :** local search heuristics that

+ Better exploits the self-adaptation mechanism

Differential Evolution strategy: The standard "rand/1/bin" DE strategies along with other ones employed widely in literature are depicted (Eqn. 5.A, Alg. 5.6, KB. 5.1).

**Eqn. 5.A: DE strategies employed in Self adaptive Bat alg.**

| | | | | |
|---|---|---|---|---|
| 1 | "rand/best/1/bin," | DE/rand/1/bin | $y_j = x_{r1,j} + F \cdot (x_{r2,j} - x_{r3,j})$ | + Directs towards current best solution |
| 2 | rand-ToBest/1/bin," | DE/randToBest/1/bin | $y_j = x_{i,j} + F \cdot (\text{best}_j - x_{i,j}) - F \cdot (x_{r1,j} - x_{r2,j})$ | + Introduces an additional randomness into a search process |
| 3 | "best/2/bin," | DE/best/2/bin | $y_j = \text{best}_j + F \cdot (x_{r1,j} + x_{r2,j} - x_{r3,j} - x_{r4,j})$ | |
| 4 | "best/1/bin," | DE/best/1/bin | $y_j = \text{best}_j + F \cdot (x_{r1,j} - x_{r2,j})$ | |

Alg. 5.6: Hybrid Self adaptive Bat alg. (Hyb.selfAdap.Bat)

**Input : X, A, r**

- Cal. current global best location (solution) in whole bats swarm.
- Four bats are chosen randomly from current population
- Selection of strategy : [ 1:4]

| | |
|---|---|
| diff.Wt. | differential weight scales the rate ofmodification |
| Prob(cross over) | crossover probability [0,1] |

| KB. 5.1 : Knowledge Base for escaping from local optima | |
|---|---|
| If | Solution is trapped in local optimum |
| Then | Redirect into new region with random walk (ABC alg.) |
| If | Solution is not changed significantly within Limit number of Iterations & Not converged |
| Then | Redirect to a new region with random walk |

## 6. Mosquitoes in nature

The evolution of mosquitoes and ants over one hundred millions of years is a cumulative consequence of trillions of micro- cosmic/terrestrial/ physico-chemical-biological processes. The natural evolution enhanced the survival of mosquitoes in widely varying environments including harsh surroundings. The memory, observation, adaptation, modification, radical (mutational) changes are retention, consolidation, up gradation of so called knowledge/intelligence is amazing even in these tiny living creatures compared to human beings and dolphins. Microscopic processes with shortcuts, metaheuristics result in macroscopic wonders which are respected/ boosted as intelligence.

Swarm intelligence is similar amazing macro-processes/groups of micro-processes exhibited by a large number of tiny unintelligent creatures without a leader. A few happenings of this sort in nature are foraging ants/honey bees, migratory birds, honey bees in site selection of honey comb, ants shifting colony, group of lions hunting by reducing the radius of encircled a lamb, female fireflies intentionally sending wrong signals to males for mating but eating them when approached etc.

### 6.1 Natural intelligence of mosquitoes to combat with life threats in the life cycle

The threats for mosquitoes in all stages of life cycle are multifold starting with eggs. The harsh environments, dynamic water/gutter bodies, insufficient nutrition sources, younger larvae becoming a prey/food to larger larvae are all prevalent in natural course of life. During evolution over more than 150 millions of years of time, mosquitoes learned and consolidated several counter methods to combat with odds and sustain their progeny from extinct of species.

**Movement of (Artificial) natural mosquitoes towards the host**

A female mosquito moves towards the host by sensing carbon dioxide, odors, and/or radiated heat. The factors influencing and consequent hybrid attraction forces for the journey of mosquitoes (agents) for food (blood meal) along their own radial orbit towards the host are

- attraction of the host
- personal/ aggregate host-seeking behavior and
- social coordination in the swarm of artificial mosquitoes

Man-made threats for mosquito life cycle: In addition to natural obstacles, man promoted intentional chemical threat for mosquito growth/spreading to prevent/eradicate infectious diseases like malaria among humans. This ranges from mosquito repellents to mosquito killers. Recent technology includes immobilizing mosquito sperm to diminish population, although they participate in mating with reproductive females. Nature is impartial to any species. The natural evolution/knowledge/intelligence tries to combat with all these threats trying to keep optimum of the survival to its span life and passage of genes to the succeeding generations and continuation of lineage.

### 6.2. Translation of mosquito-host-seeking-process-in_nature into nature_inspired-(artificial)_mosquito-host-seeking-algorithm

Feng et al. [171] proposedmosquito host-seeking algorithm (Mosq-host-seek, MHSA) inspired by the unique features viz. parallelism, local interactivity and self-organization of real mosquitoes in their kinematics and dynamics. The authors report that it is diverse from other similar nature inspired algorithms and has an edge over similar metaheuristics. The knowledge base can be represented as first order predicate (If-then-else) logic in prolog (AI language) style.

6.3. Artificial mosquito: In the proposed Mosquito-host-seeking algorithm, all the artificial (swarm of) mosquitoes are evenly distributed surrounding a host. The radial distances between them are even(Fig.6.1). In fact, each artificial mosquito (mosqij) is a computing cell, with the sex attribute xij. But, in the algorithm, all artificial mosquitoes are females.

- Sex of mosquitoes: The sex of artificial mosquitoes is a logical/binary variable and a value of one corresponds to a female (KB. 6.1). A male mosquito does not contribute to computation and thus the corresponding computing cell is dead.Each female artificial mosquito corresponds to a living computing cell.

- Gray values of mosquitoes: The grayscale value of an artificial (female) mosquito (greyVal) changes between 0 and 1 as it moves. The grey values of male mosquitos are always zero (KB-6.1c). When an equilibrium state is reached, greyVal converges to 0 or 1 and the approximate solution is deemed as converged.

**KB. 6.1: Characteristics of artificial mosquitoes**

**(a) Sex of mosquito**

**If** X(i,j) =1
**Then** Mosq(i,j) = female
ComputingCell = alive

**If** X(i,j) =0
**Then** Mosq(i,j) = male
ComputingCell = dead

**(c) greyVal and inferences for female mosquitoes**

**At equilibrium**

**If** greyVal(i,j) = 1
**Then** artificial female mosquito attacked the host &
path(i,j) is black &
short_ path passes through the path

**If** greyVal(i,j) = 0
**Then** Female mosq(i,j) did not attack the host &
path(i,j) is white &
short_ path does not pass through the path(i,j)

**(d) : greyVal at equilibrium state and inferences**

**If** Dist.rad <= small
**Then** artificial mosquitoes are closer to the host &
$utility\_sum(it)$
is larger

**(b) Sex based attributes of mosquitoes**

**If** Mosq is male
**Then** mosqSex = 0
Computing cell = dead
greyValue fixed at 0
path is white

**If** Mosq is female
**Then** mosqSex = 1
Computing cell = alive

**If** Mosq is female &
It is in motion
**Then** greyValue changes

greyValue = [0 to 1]

xij does not evolve with time

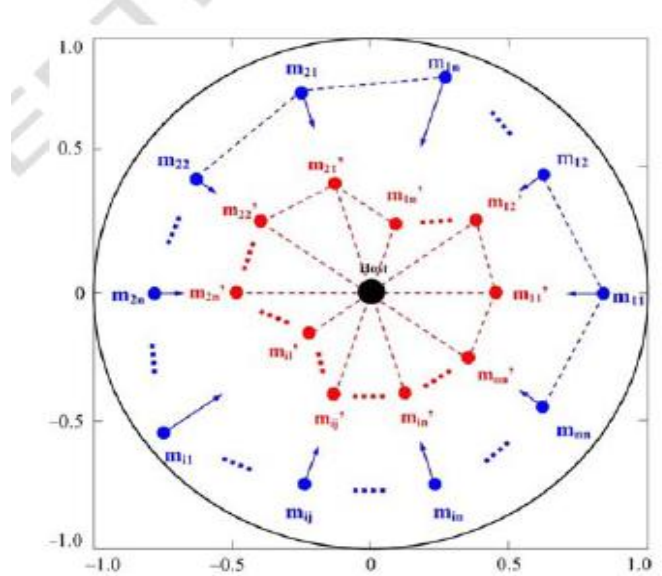


Fig.6.1 The biological model of MHSA base on theoretical biology

Initiation of grayscale values: ThegreyVal of artificial mosquitoes [mosq(i,j)] are initialized as average values with the constraints shown in Formula. 6.1A.

Formula. 6.1A: Initialization of greyVal (rij) **(Un-influential process)**

| $r_{ij} = \left[ones(n,n)\right] * \frac{2}{n}$ | $r_{ij} = rand(n,n)$ |
|---|---|

| | Output |
|---|---|
| %<br>%<br>[Greyvalues_init_Fr, Greyvalues_init_rand] = function init_greyvalues(n)<br>$r_{ij} = \left[ones(10,10)\right]*0.2$<br>[Greyvalues_init_Fr] = [ones(n, n)] * 0.2<br>[Greyvalues_init_rand] = randu(n, n) | n= 3;<br>Greyvalues_init_F r=<br>Greyvalues_init_rand = |

| **Constraints** | | |
|---|---|---|
| Equality | $r_{ij} = 0 \ \forall x_{ij} = 0$ | If $x_{ij}$ =0<br>Then greyVal(i,j)=0 |
| Nonnegativity | $r_{ij} = r_{ij} - \min\left(r_{i,j}\right)$ | If $\min\left(r_{i,j}\right) < 0$<br>Then $r_{ij} = r_{ij} - \min\left(r_{i,j}\right)$ |
| Normalisation | $r_{ij} = \frac{2 * r_{ij}}{\sum_{i=1}^{10} r_{ij}} \ \ and \ \ \frac{2 * r_{ij}}{\sum_{j=1}^{10} r_{ij}}$ | |

Weight of connection between a pair of points WtC(cij): The weights of the points (mosquitoes) at initiation and during iteration process are calculated vide KB. 6.2.

KB. 6.2: Weights of points

| | | |
|---|---|---|
| **If** | Iter = 1 | %Initial |
| **Then** | WtPt (i,j, it = 0) = [max (dij)] - dij ; | |
| **If** | Iter >0 & | %Iterations |
| | Not converged | |
| **Then** | $WtPt(i, j, it > 0) \subset [0,1]$ | |

KB. 62b: Path of mosquitoes

| | |
|---|---|
| **If** | path(i,j)passes through X |
| **Then** | xij : xij = 1 |
| **If** | convergence & |
| | path_short passes through X |
| **Then** | XOpt =1 |
| | greyVal(i,j) = 1 |
| **If** | convergence & |
| | path_short does not passes through X |
| **Then** | XOpt = 0 greyVal(i,j) = 0 |
| **If** | Not convergence |
| **Then** | XOpt = [0,1] |

Computing cells: The data structure for minimum path finding with np points is np x np computing cells. The state of computing cell depends upon a logical variable (xij). It consists of four computing cell arrays, C, Crow, Ccol, and Cglobal (Chart 6.1) with a total number of $n^2$+2n+1 cells. There is no interconnection among computing cells in the same array.

**Chart 6.1: Np x np grid of computing cells**

| | | |
|---|---|---|
| # | : | Array |
| n × n | : | C |
| n | : | Crow |
| n | : | Ccol |
| 1 | : | Cgloble |

| Local Conne-ctions | | Con-nection order |
|---|---|---|
| Cij | Ci | 2 |
| Cij | C_j | 2 |
| Ci_ | C__; | 2 |
| C_j | C__. | 2 |

| total number of computing cells: |
|---|
| $n^2+2n+1$ |

| Cgloble computing cell | n + n |
|---|---|

connection degree of each computing cell in array C of n × n computing cell is equal to at most 2,

Distance:This algorithm uses Euclidean distance between two points (cities in the case of travelling sales man task) (om. 6.1) and radial distance between female mosquitoes and the host.

```
Oject matlabFile (Om. 6.1):distEucl
%
%   om_distEucl.m    12/6/13
%
function [dist_Eucl] =om_distEucl(X)
%(xi, yi) is the coordinates of a point (or city Ci)
%
[nsol,ndim] = size(X) ;
dist_Eucl   = zeros(nsol,nsol);

for  i = 1:nsol
%%%
for j = 1:nsol
%%%
if i ~= j
        %%
           Xi = X(i,:); Xj = X(j,:);
           dist_Eucl(i,j) = norm (Xi - Xj) ;
%
end%%if
end%%%j
end%%%i
```

$i \neq j,\ d_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2},$

*Radial distance of mosquitoes from host:* The success value of artificial mosquito in host-seeking (or also called personal utility) is equal to the radial distance between an artificial mosquito and the host (Chart 6.2). It measures how close a female mosquito is to the host.

**Chart 6.2: Mosq_Host_Seek alg.**

$$dist\_radial(it) = \exp\left(\left[-c(:,:,it) * r(:,:,it) * X(:,:,it)\right]\right)$$

$$utility\_sum(it) = sum(sum(dist\_radial(it)))$$

| | | |
|---|---|---|
| $dist\_radial(it)$ | : | radial distance between artificial mosquito mosq(i,j) and host at iteration it |
| $utility\_sum(it)$ | : | utility sum of all artificial mosquitoes |

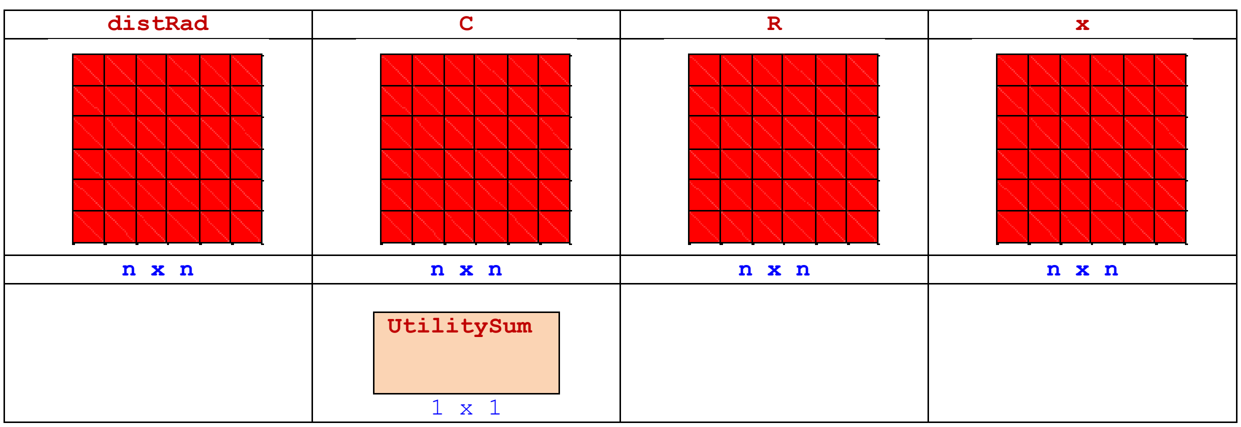
**Input**

| | | | |
|---|---|---|---|
| c = | 1 | 1 | 1 |
| | 1 | 1 | 1 |
| | 1 | 1 | 1 |
| r = | 2 | 2 | 2 |
| | 2 | 2 | 2 |
| | 2 | 2 | 2 |
| X = | 3 | 3 | 3 |
| | 3 | 3 | 3 |
| | 3 | 3 | 3 |

```
%
%  om_utility.m R S Rao 07-01-2014
%
function [dist_radial,utility_sum] =
om_utility(c,r,X)
dist_radial = c.*r.*X;
utility_sum = sum(sum(dist_radial));
```

**Output**

```
dist_radial =

 6        6        6
 6        6        6
 6        6        6

utility_sum = 54
```




Path:The line joining between any two points (or cities in the case TSP) called path.
Shortest path (Z): The route with minimum length passing through all, but touching each point only once is the shortest path (path_short). Of course, path_short begins at the starting point and again ends at the same one i.e. like a closed circuit. It is the solution of a minimum path task at the end of iteration process.

The attraction and utility functions are incorporated in formula. 6.2, om. 6.2 and om. 6.3.

**Formula. 6.2 Attraction Function between host and mosquitoes**

$$attractFn(it) = has^2 * \ln\left(\sum\sum \exp\left\{\frac{distRad_{i,j}^2}{2 * has^2}\right\}\right) - has^2 * \ln$$

| | | |
|---|---|---|
| P(t) | : | attraction function caused by the host at time t |
| has | : | strength of a host's attraction host_attract_strength |

Om. 6.2: Attraction function

```
%
%      om_attractFn.m     R S Rao 07-01-2014
%
function [attractFn] = om_attractFn(distRad)
[row,col] = size(distRad); n = row;
randu = rand(1,1), randu2= randu*randu
T1 = exp(-[ [distRad.^2]./[randu2/2] ])
T2 = sum(sum(T1))

attractFn = randu2 * log(T2)-randu2.^2*log(n*n);
```

| | |
|---|---|
| If | Mosquito is extremely weak |
| Then | they do not attack the host |

| Constraints | |
|---|---|
| Equality | $c_{ij} = 0 \ \forall x_{ij}$ |
| Nonnegativity | $c_{ij} = c_{ij} - \min\left(c_{i,j}\right)$ |
| Normalisation | $c_{ij} = \frac{2 * c_{ij}}{\sum_{i=1}^{10} c_{ij}} \quad and \quad \frac{2 * c_{i}}{\sum_{j=1}^{10} c_{i}}$ |

Interaction behavior function =

penalty function (or related to the constraints on TSP )

+ monotone decreasing sigmoid function (or social coordinations among the artificial mosquitoes)

Om. 6.3: Utility function

```
[distRad,utility_sum] = om_utility(wtC,Xopt,X);
 %
randu = rand(1,1), randu2 = randu*randu
T1 = exp(-[ [distRad.^2]./[randu2/2] ])
T2 = sum(sum(T1))
%
dpathByXopt = -distRad.* T1./T2.*ddistRadByXopt

sumij = 0
for i = 1:row
   sumj = 0
   t11(i) = 0
   for j = 1:col
      t12(i,j) = 1./(1+exp(-10*distRad(i,j)))-0.5
      t11(i)= t11(i) +x(i,j)
      sumj = sumj + Xopt(i,j).*X(i,j)
   end %j
   sumij = sumij+sumj-1
end  %i

 for i = 1:row
    for j = 1:col
    dinteractFnByXopt(i,j) = 2* t11(i) * sumij *t12(i,j) *ddistRadByXopt(i,j)
    end  %j
 end  %i
```

**Equations and pseudo code ofMosquitoHostSeekAlgmetaheuristic** (MosqHostSeekAlg)

The equations for movement and motion of mosquitoes towards host are depicted in pseudocode. 6.1.

**Pseudocode. 6.1: Mosquito_host_seek_Alg**

Iterate until all $\left( \frac{d\left\{distRad(i,j,it)\right\}}{dt} = 0 \right)$ % finish successful

For i = 1: computing cell Cij (artificial mosquito mosq(i,j))

Calculate $distRad(i,j,it)$ of each mosq(i,j)

Cal $\frac{d\left\{distRad(i,j,it)\right\}}{dt}$

Cal $\frac{d\left\{greyVal(i,j,it)\right\}}{dt}$

Cal $\frac{d\left\{c(i,j,it)\right\}}{dt}$

% upgrade $greyVal(i,j,it)$

$$greyVal(i,j,it+1) = greyVal(i,j,it) + \frac{d\left\{greyVal(i,j,it)\right\}}{dt}$$

% refine $c(i,j,it)$

$$c(i,j,it+1) = c(i,j,it) + \frac{d\left\{c(i,j,it)\right\}}{dt}$$

endFor

endIterate

**Initialisation**

**Constants specific to mosquito host seeking alg**

The magnitudes of const (Formula ) vector specific to this algorithm are taken as [0.8, 0.05, 0.05, 0.9, 0.9].

*Parameters :*

Chart :Knowledge of choice of constants $\left\{[const_1, const_2, const_3, const_4\ ],[\varepsilon]\right\}$ for sure and fast convergence

| | |
|---|---|
| If | $\frac{const_4}{const_1 + const_2} >$ larger |
| Then | MHSA converges faster |

| | |
|---|---|
| If | $(const_1 + const_2) \leq 0.45\&$ <br> $const_4 \geq 0.9$ |
| Then | MHSA converges to a stable equilibrium state |
| If | $const_1, const_2, const_3, const_4$ *are chosen such that* <br> $(const_1 + const_2) * \frac{\partial J(t)}{\partial U_{ij}(t)} + const_3 * \frac{\partial P(t)}{\partial U_{ij}(t)} + const_4 * \frac{\partial Q(t)}{\partial U_{ij}(t)} < 0.0 \quad \text{Eqn.7.14}$ |
| Then | convergence and stability can be guaranteed i.e. <br> $t \to \infty, as\ R(t) \to R$ |

**Uninfluential parameter:** $\lambda_3$ *magnitude hardly* inf *luences convergence of the MHS*

| | |
|---|---|
| If | $const_4 >> [const_1 \ and \ const_2]$ *based on Eqn.7.17* |
| Then | MHS algorithm converges |
| If | $const_1, const_2, const_3, const_4$ change in direct proportion |
| Then | results of the MHS algorithm will hardly be influenced <br> $const_1, const_2, const_3, const_4 \in [0,1]$ |

**Change in WtC and path ($\Delta wtC, \Delta path$)**

The dynamic equations of computing cell Cij for solution variable greyVal(i,j,iter) and weight variable cij(t) are computed in parallel (om. 6.4, om. 6.5 om. 6.6).

**Om. 6.4: calculaation of delta_pw**

```
%
%      om_delta_pw.m     R S Rao 07-01-2014
%
function [delpath,delwtC] = om_delta_pw(pdfn)

dpathByXopt= pdfn.pathByXopt
dJBywtC   =pdfn.JBywtC
dpathBywtC = pdfn.pathBywtC

dinteractFnBywtC= pdfn.interactFnBywtC

  delpath = -const1* ddistRadByXopt - const2 * dJByXopt -const3 * dpathByXopt..
            -const4 *dinteractFnByXopt

  delwtC = -const1* ddistRadBywtC - const2 * dJBywtC -const3 * dpathBywtC..
            -const4 *dinteractFnBywtC
```

| | | | |
|---|---|---|---|
| Const1 +const2 | ✌ | Personal utility | ∞ |
| | ✌ | Increases | |

| | | | |
|---|---|---|---|
| const2 | ✌ | Monotonic increase of the whole utility of all the artificial mosquitoes | ∞ |
| Const4 | ✌ | Monotonic decrease of the artificial mosquitoes' behavior interaction function Q(t) | ∞ |
| Const3 | ✌ | Increasing the minimal utility | ∞ |
| | ✌ | Decrease of P(t) will result in the increase of the minimal utility, in direct proportion to the value of λ3. | |

**Om. 6.5: calculation of v based on radial distance**

Input
distRad =

| | | |
|---|---|---|
| -2 | 0 | 1 |
| -0.5 | 0.5 | 11 |
| -0.002 | 0.004 | 1.006 |

```
%
%       om_v.m      R S Rao 07-01-2014
%
function [plfu] = om_v(distRad)
[row,col] = size(distRad)
for i = 1:row
for j = 1:col
if distRad(i,j)<0
            v(i,j) = 0;
elseif distRad(i,j)> 1;
            v(i,j) = 1;
else
            v(i,j) =  distRad(i,j);
end
%i,j,v(i,j)
end
end
plfu = v;
```

$$v_{i,j}(it)\begin{cases} 0 & if & distRad_{ij} < 0 \\ distRad_{ij} & if & 0 \le distRad_{ij} \le 1 \\ 1 & if & distRad_{ij} > 1 \end{cases}$$

Output
plfu =

| | | |
|---|---|---|
| 0 | 0 | 1 |
| 0 | 0.5 | 1 |
| 0 | 0.004 | 1 |

**Refinement (updating/iteration) of approximate set of solutions**

$$pathShort = \sum_{i=1}^{nmosq}\sum_{j=1}^{nmosq} d(i,j)*r(i,j)*x(i,j)$$
$$= 28.5866$$

**Om. 6.6: Refinement of WtC and path:**

| | |
|---|---|
| Xapp = | 0.11 0.12<br>0.21 0.22 |
| deltaX = | 0.09 0.08<br>0.09 0.08 |
| iter = | 2 |

```
%
%   om_refineX.m    12/6/13
%
function [X,Xiter] = om_refineX(Xapp,deltaX,iter)

X= Xapp + deltaX;
Xiter(:,:,iter)  = X ;
```

| | |
|---|---|
| Xrefined = | 0.2 0.2<br>0.3 0.3 |
| Xiter(:,:,2) = | 0.2 0.2<br>0.3 0.3 |

The features and positive characteristics of Mosq_Host_Seek alg are briefed in chart 6.3.

**Chart 6.3(a): Unique Characteristics of mosquito algorithm**

- The artificial host-seeking behaviors of mosquitoes are distinct compared to other nature inspired algorithms in its elements/ their states/ biological model/ translated mathematical model and the theoretical foundation.
- Inherent parallelism
- Good scalability
    - The cellular structure, the cellular dynamics and algorithm are all independent of the problem scale
- Optimization of multiple objectives which could include aggregate utility, personal utility, minimal personal utility, etc.

Parallelism inMosq_Host_Seek alg.

- The calculation of grayscale values, weights, positions of points etc. are computable and upgradable in parallel
    - Since they do not require any information exchange
- Computations of cellular dynamics both in the same array and in differentarrays are concurrently carried out
- No direct interconnection among computing cells in the same array

**Chart 6.3(b ): Advantages of Mosq_Host_Seek alg**

- Converges to true optima
- Robust
    - Parameter setting does not depend upon
        - Learning algorithm
        - Prior knowledge
        - Initial conditions
        - Problem size
        - Small-range parameters changes
- Comprehensive optimization ability for multiple objectives
- Distributed optimization of large-scale intricate and dynamic real-time changing environment complicated problems
- Low execution times due to parallel nature
- Models complicated, high-dimensional, highly non-linear, micro-evolutionary and random dynamic behavior patterns
    - This is due to the introduction of the aggregate intention strength factor in the definition of the artificial mosquitoes' interactionbehavior function Q(t).

**Stable equilibrium state and Lyapunov function:** In this algorithm, if the host disappears permanently along with its attraction, it is deemed that a final stable state is reached. Then the swarm of artificial mosquitoes stops moving. Lyapunov second theorem on stability is proved for MHS using hybrid attraction function (Chart 6.4). Thus it is better in performance compared to other nature inspired procedures. $$$

| | |
|---|---|
| If | Artificial mosquitoes stop moving |
| Then | Equilibrium state |

**Chart 6.4: Lyapunov theorem on stability**

Lyapunov second theorem on stability: Consider a function L(X) such that

- L(X) > 0 (positive definite)
- $\frac{dL(X(t))}{dt} < 0\,(Negative\,definite)$

Then L(X(t)) is called a Lyapunov function candidate and X is asymptotically stable in the sense of Lyapunov

Hardware and Software:Feng et al. [171] implemented the algorithm on sequential to 16-node computer.

### 6.6 Applications of Mosquito algorithm

**Quality assurance:**Andras [178] used mosquito algorithm in call admission control with real time guarantee on Quality of Service (QoS) parameters. The earlier procedures reported calculated the loss probability accounting for statistical behavior of the sources and measurement errors. Further, too many sources admitted into the system violate QoS. The effect of bound on the probability of a bad Call Admission Control (CAC) decision due to measurement uncertainty is not accounted for. But, the probability of a buffer overflow is a cumulative effect of all these factors. The results of mosq.host.seek.alg are superior.

## 7. Selection of sites for egg laying by female mosquitoes in nature

The mother mosquito is intelligent in selection of sites for laying eggs by spreading them in different sites (called oviposition). Astonishingly, the in situ eggs delay hatching themselves till the surroundings are favorable. This is another amazing nature's trait for adaptation of species to surroundings.

| | |
|---|---|
| | mosquitoes in nature sense and move towards host |
| sensing | carbon dioxide in exhaled breath, odorous compounds in sweat and heat of body |

The slow phenotype changes over generations at first level and consequent genotype alterations with knowledge/intelligence over a very long time period is worth noting. Here, time scale depends upon their life span. It is also related to number of eggs laid each time and total number during the life cycle of a female mosquito, with a consequence of number of adults in next generation. It is a miracle to the human brain to understand natures' tech-knowledge even at functional levelat the moment, leave alone probing into single cellular/bimolecular stage.

Mating of mosquitoes: A female mosquito on mating with a male mosquito (with active sperm) becomes pregnant.
Further, the delay of the egg in hatching is another wonder as even mother mosquito has no role here.

Egg laying of mosquitoes in nature: In nature, female mosquitoes lay eggs in left over containers or in water in guts, after mating with male mosquitoes. The water environment is dynamic and uncertain increasing the chances for destruction of eggs,the potential off spring.

Intelligence in site selection for egg laying: The site selection for egg deposition (called oviposition-sites) is based on multiple environmental parameters like temperature, time in the year, moisture and nutrient content of the water and number of mosquito larvae already present at the site. The better, if not worse compared to the current location is searched by the female mosquito in and around the vicinities. It is astonishing how female mosquitoes learned the process of monitoring and deciding the best breeding habitat. Thus, the intelligence lies in selection of sites.It lays eggs at every potential site spreading over congenial areascattering to different sites [176,177]. And, this intelligent oviposition strategy maximizes the survival of eggs growing into larvae. This knowledge of bioprocess and monitoring is conceived even now as intelligence of a female mosquito. It reminds one, the episodes of the honey bees searching/selection of a site for building a honeycomb and also ants shifting of the colony.

Built in (intelligent) knowledge in eggs to combat with life threats ( Ovipause): The hatching produces larvae which are transformed into adult mosquitoes subsequently.Mosquito eggs themselves exhibit an intelligent process knowledge named Ovipause. This intelligent defense operation delays their hatching till favorable conditions exist around surroundings (KB. 7.1).

| KB.7.1 : Knowledge bits of ovipause in mosquito eggs | |
|---|---|
| If | Pregnant female mosquito laid eggs in intelligently chosen sites & Environment is unfavorable |
| Then | Eggs themselves delay hatching process until conditions become favorable |
| if | Competition for nutrient resources is tough in breeding grounds |
| Then | Ovipause promotes survival of mosquito eggs and larvae production |
| **Consequences:** enhances survival of eggs → larvae production → → increase in adult mosquito population | |

## 7.2 Translation of natural female mosquitooviposition-site selection and (self) egg-hatching activity into artificial mosquito-oviX- optimization algorithm

The selection of male/female mosquito for mating, egg laying and eggs themselves delaying hatching are all highly knowledgeable processes. These natural processes are the inspiration for Minhas and Arif [172] to propose mosq.oviposition.site_selectionalgorithm (Mosq-oviX-alg) for global optimization of multidimensional mathematical functions. The test results of several standard multimodal non-separable functions endorse the prospects of this approach.

## 7.3 Artificial mosquito OX

Each adult mosquito is represented by X-matrix denoting its position in 3D-space. The value of X is deemed as its fitness value of that mosquito. A mosquito (male or female) flies to approach a selected mate or just performs random jumping/shift (chart.7.1).

Object (Fitness) function: A random number of position vector components (determined by probability) of female and its partner are selected. If object (fitness) functions (Alg. 7.1) are favorable, then this change is accepted.

**Alg. 7.1: Fitness calculation and analysis**

- Rank the fitness of all females in the adult set
- Calculate prob(female to attract a male) with a roulette wheel

| | |
|---|---|
| [xit] = | fitnessFnValueCheck(fitnessFn,xit, xtemp) |
| | fitnessFnValue = fitnessFn(Xtemp) |
| If | fitnessFnValue > Xf |
| Then | xf = xtemp<br>update saved fitness for the female |

| | |
|---|---|
| If | fitness of ***xtemp*** is not worse than that of ***xf*** then |
| Then | ***xf* ←*xtemp***<br>update the saved fitness for the female<br>#eggs to be carried by female (f)← randomInteht(1:Ne) |

Random shifting in adult mosquitoes: In male mosquitoes, random shifting of the current position is determined by a probability prob_c. An adult female mosquito selects two random position vectors xb, and xa with the constraint objFnValue(xb) > objFnValue(xa) (chart 7.2). The value (xb-xa) denotes direction of improvement. The current best location (xbest) for a randomly chosen mosquito is used to find a new location for the female.

**Chart 7.2: random jumping/shift**

**Random Shift**

- Random selection with prob ( )
- Find Components to be changed from xm of xDim

**Random jumping**

- Select (randomly) three individuals from the set of adults
- Cal fitnessValue; sort (fitnessValues)
- ([*xa*<*xb xc*)

| | |
|---|---|
| xmChanged = Add randN([0,1])<br>fitnessValue (xmChanged) = fitnessFn(xmChanged) | Find the best individual, *xbest* in the current adult set<br>Xtemp = diverse * xbest + (1-diverse) *xc +<br>varPos *(xb-xa)<br>FitnessValue = fitnessFn (Xtemp) |
| If FitnessValueXtemp > = xf<br>Then Xf = xtemp<br>save improvement | If gama is close to 1<br>Then new location is (in step 5.2.2) closer to the current best<br>If gama low value<br>Then more diversity i.e. Location near any randomly chosen |

### Mating of artificial mosquitoes

The positive benefit of matingprocess in natural mosquitoesis exchanging genetic information for beneficiary aspects of off sprigs. The typical steps adapted in artificial mosquito mating algorithm are briefed in chart 7.3.The gender of the new born mosquito is chosen at random.

**Chart 7.3: Artificial mosquitoes**

| | | | |
|---|---|---|---|
| **FitnessFnValue** | **Male** | | |
| | **Female** | | |
| **Attraction** | **Prob (Male vs female)** | | |
| | **Prob (Female vs male)** | | |
| **Movement** | **Male** | **Moves towards** → | sleeted female |
| | **Female** | **Moves towards** → | selected male greedily |
| **Mating** | **Female** | **Mating with male at a probability level** | |
| | | | |
| **Female** | **Produces eggs** | | |
| | | | |
| **Female** | If carrying eggs<br>Searches a suitable site for egg laying<br>Else Random jump | | |

Movements of mosquitoes toward mates or for mating

Movement of a male or female species towards a mate in fact knows information about "preferable" locations among the two partners (chart 7.4, KB. 7.2).

Movement of male mosquito: Each male mosquito selects a female partner based on her rank of fitness and moves a random distance towards a female for mating or randomly around its neighborhood.

Female mosquito selecting a male partner: For mating, a female agent selects its male partner on the basis of the fitness rank of the male. Then, the female mosquito moves towards the male by changing its position as a function of the position of the male.The female mosquito in the beginning (initial iterations) performs hill climbing search.

**KB. 7.2(a): Female mosquito for mating**

| | |
|---|---|
| If | A female mosquito is not carrying any eggs (or pregnant) |
| Then | it is available for mating<br>mosq.female.mating.eligible =.yes. |
| | |
| If | mosq.female.mating.available |
| Then | Participates in mating with a probability (prob_) |
| | random jumping -- moves off to a random location |

**KB. 7.2b:Female mosquito movement**

| | |
|---|---|
| If | Old position of female is better than the new one, |
| Then | It flies back to its old position after mating |

| | | |
|---|---|---|
| For | each male *m* in the adult set | |
| | **If** | male either selects a female |
| | **Then** | moves towards it with prob(moveMosq) |
| | | performs a random shift |
| | **End** | if |
| **end** | | |

**Chart 7.4 (a ):Movement of male mosquito towards a female**

- ✓ Prob( femalesTo AttractAmale (calculated in step-7)
- ✓ select a female partner ***xf*** for this male
- ✓ Move the male towards the selected female
- ✓ *xm=xm +randU([0,1]) * (xf -xm)*

**Chart 7.4b: Movement of female mosquito towards a male**

- ✓ Prob( malesTo AttractAfemale
- ✓ Random selection of a male partner ***xm***for this female
- ✓ Female Moves towards the selected male greedly
- ✓ Female produce eggs

**Chart 7.4(c )Mating**

**Select randomly male partner xm**
**Xtemp = xf;**

**[ (Xtemp, xf)**
fitnessFnValue = fitnessFn(Xtemp)
if fitnessFnValue > Xf, xf = xtemp
fitnessFnValue(xf) stored

**Mosq.OviX.alg**

→ Xtemp = Xfemale(i)
→ Select random number of components of Xtemp
→ Add noise to Xtemp
→ Xtemp(i) = Xtemp + randN(0,beta)
→ female lay an with prob(e) at xf
→ % fitness of egg is same as that of xf at same location

### Female mosquito selecting site (X) for laying eggs

MOX for MOO:The adult mosquito is represented by a position vector. Here, an egg is also considered by the location where it is laid. The fitness of each agent, mosquito as well as egg, is calculated from the coordinates of their positions.

### Pseudo code of Mosquito_oviX_Alg:

This algorithm uses a model of distribution of laying of eggs by female mosquito and also inhibition in hatching process later by eggs themselves (alg. 7.2).This algorithm gives a chance for each individual in the adult population to change its location either through mating or by random movements

**Alg. 7.2: Bird's eye-view of Mosquito_oviX_Alg**

☞ `Selection of N best performing mosquitoes`

```
    ☞ Eggs selected will be hatched
    ☞ Ranking  males
    ☞ Cal prob(attracting_a_female);
    ☞ Method:Roulette wheel selection
%
For     each female mosquito
        If      Carrying eggs
        Then    Selection of site to lay eggs

        If      not pregant
        Then    Mating
        Else    Random Jumping
End     for


If      Size_egg_set > egg_set_Max
Then    Eggs with lowest fitness are removed
%
Ranking females
Cal prob(attracting_a_male);
Method:Roulette wheel selection

%
For     each male mosquito
        If      Selects a female mosquito
        Then    Moves towards it
        Else    Random shift
End     for

    ✍ Elitism: Replace worst mosquitos by best
        o  individuals already stored
```

| | |
|---|---|
| If | Number of eggs laid > eggsMax |
| Then | Discard eggs with low fitness |
| If | Fitness (i,it) > Fitness(worst adult which is replaced ) |
| Then | Egg is hatched (to be added to the adult population) |

Initialization: Approximately equal number of male and female adult mosquitoes spread over fitness landscape (alg. 7.3) is initiated with uniform random number generator. The total number of mosquitoes is approximately twice the dimensions of X-search space.

The major steps of OVIX are in alg. 7.3 to alg. 7.5. The advantages of Mosq.OviX.alg are described in chart 7.6.

| **Alg. 7.3Initialization. OVIX** | |
|---|---|
| Random generation of Nmosq | Mosq.adult = rand(X(i,dim)) |
| choice of gender randomly and with equal probability $nfemale\_mosq \cong nmale\_mosq \cong \frac{nmosq}{2}$ | Mosq.adult(prob(p,gender)); gender(i) = {male,female} |
| If: xDim is dimension of X; Then: nmosq = 2 * xDim | |

| | |
|---|---|
| Number of eggs for each adult to zero | Number of eggs in egg set = 0 |
| Beta = 1 | Beta = 1 |
| Calculate fitness value | FitnessFnValue = Fitness(i) |
| Evaluate mosquitos using fitness criteria and store | Fit.mosq (k) = {FitnessFnValue(i)) > FitnessCriteria} |

| | | |
|---|---|---|
| Nmosq | : | Number of mosquitoes (solutions) |
| Nmales | : | Number of adult male mosquitoes |
| Nfemales | : | Number of adult female mosquitoes |

| | |
|---|---|
| pm | 0.4 and 0.7. |
| Emax | 10 and 18 |
| Ne | 2 and 6 |
| pe | 0.8 and 1.0 |
| pr | 0.4 and 0.6. |
| pc | 0.05 and 0.2 |
| | |

If Gama is very small
Then Decreasing attraction forces of host leads to increasing the minimal utility of artificial mosquitoes

| Algorithm Parameters | | |
|---|---|---|
| Maximum number of adults | : | *mosq_adults* |
| Number of components in position vector of each individual = number of variables | : | *Xdim* |
| Maximum number of eggs | : | egg_max |
| Maximum number of eggs a female mosquito can carry | : | egg_max_carry |
| Probability of mating | : | prob_mating |
| Probability of each component of position vector of a maleto be changed in mating with a female | : | prob_X_change_in_ male_mating_with_female |
| Probability of Laying an egg | : | prob_egg_laying |
| Probability of each component of position vector of a male to be changed in Random Shifting | : | prob_male_randShift |
| Gamma | : | Gamma |
| Minimum value of $\beta$ | : | betaMin |

**KB. 7.3: Female mosquito intelligence in increasing life of its progeny**

**Egg laying**
If a female is carrying &
Many (>1) eggs available in womb
Then It deposits only one egg in each iteration

Spreading egg laying locations
If Vicinity locations not worse than Current location
Then Female mosquito distributes/scatters eggs in vicinity also

| Alg. 7.4:Egg production by female | 5.2.1 |
|---|---|
| ✓ Make a copy of position of female | xtemp = xf |

| | |
|---|---|
| ✓ Select randomly co-ordinates of d-dimensional Xf<br>✓ Copy to xtemp | Xtemp |
| ✓ Cal fitness value & check | fitnessFnValueCheck |
| ✓ Random integer between 1 and EggsMax | randInt([0 to maxEggs |
| ✓ Number of eggs carried by female(i,it) | |

**Alg. 7.5:Mosq-oviX-alg**

- Input X
- Random generation of mosqSex with equal prob([male,female])
- Beta ← 1
- Cal fitFn(X)
- Choose N best fitness mosq
- Remove others from the set

| | |
|---|---|
| If | Size of egg set > Emax |
| Then | Remove eggs with worst fitness |

| | |
|---|---|
| If | Fitness of eggs > fitness of adults |
| Then | Adults with less fitness deleted<br>Eggs are added to adults list |

Consequence → increase in number of adult mosquitoes with high fitness in different areas

- Select eggs and consider them as adults
- eggs = []
- Fitness(mosq.select)
- Pickup best performing → adult.best
- Cal beta(iter)
- fitnessRank ← sort(fitness.males)
- Roulette wheel (fitnessRank)
- Prob(maleAttractingFemale)

| | | | |
|---|---|---|---|
| Iterate | until convergence OR stopping criteria true | | |
| | For | i = 1: #mosq_female (computing cell Cij) | |
| | | | **egg_laying_site_selection** |
| | | if | female is carrying eggs (egg_female >0) |
| | | Then | female_mosq greadily searches for a site to lay its eggs |
| | | | |
| | | if | female is not carrying eggs (egg_female = 0) |
| | | Then | mates with probability (prob_mating) |
| | | else | Jumps to a random location |
| | end For | | |
| endIterate | | | |

Variation in position of solution (X) (var_positX) beta:With progress in optimization, the distances between different solutions (X) change. The adaptive constant var_positX controls these changes helping convergence to global optima. It uses information of fitness improvement in the preceding iterations.

**Chart 7.5: Adaptive variation in position of solution (X)**

| Formula | KB |
|---|---|
| $var\_positX(i,it+1) = RandN([0,0.1]) * \frac{1}{H} * \sum_{i=1}^{H} var\_positX(i,it) * \mathrm{xDim(i,,f)}$ | If: $var\_positX(i,it+1) < var\_positX\_min$<br>Then: $var\_positX(i,it+1) = var\_positX\_min$<br><br>If: $var\_positX(i,it+1) > 1$ |

| | | |
|---|---|---|
| | Then | $var\_positX(i,it+1)=1$ |

$$var\_positX(i,it+1) > 0.3$$
$$range\big(var\_positX(i,it+1)\big) = [20\ to\ 40]$$

**Chart 7.6**

**Positive features**

- Efficient with reduced number of function evaluations
- Global optima of a multidimensional function at a high success rate
- Have implicit parallelism in search -->
- Less susceptible for settling in local optima
- Easily implementable on parallel hardware architectures

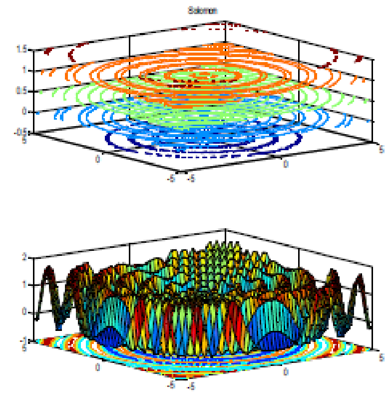
## 7.4 Applications

Mosq.OviX algorithm has been used to find global optima of multi-dimensional test functions (chart 7.7). Matlab graphic output for a few select functions is given in fig. 7.1.Mosq.OviX can be extended to dynamic optimization task because inhibition of egg hatching when conditions are not appropriate helps to find and keep track of optima of a dynamic optimization problem.

**Chart 7.7: Test functions used for Mosq.OviX**

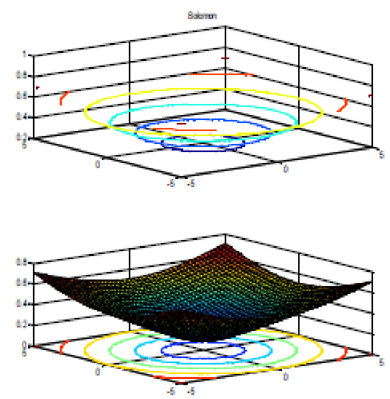
| Modality | Separabiliy | Function Name |
|---|---|---|
| Unimodal | Separable | Sphere |
| | | Step |
| | | |
| | Non-separable | Schwefel |
| | | Rosenbrock |
| | | |
| Multi-modal | Separable | Quartic |
| | | Rastrigin |
| | | |
| | Non-separable | Ackley |
| | | Grewank |

**Fig. 7.1: Iso-contours and 3D-surfaces of select function used in nature inspired algorithms**

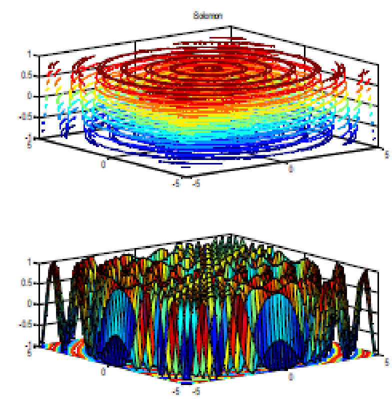



```
f1f2= f1+f2;
```

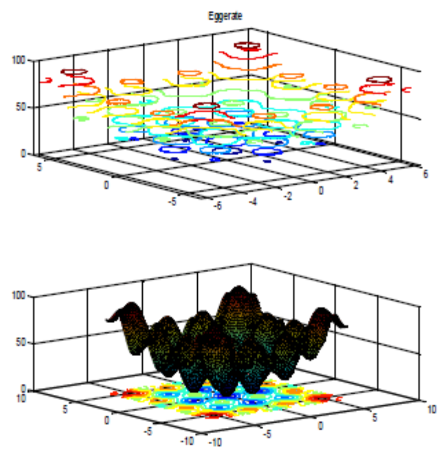
```
%
% Solomon
%
c =5;ll = -c ; ul = c; inc =
.2;
[x,y,one] = xygrid(ll,ul,inc);
fnName ='Solomon'
[row,col] = size(x);
f1 =  0.1 * [sqrt(x.^2 +y.^2 +
one)];
f2 =  -cos([2*pi* sqrt(x.^2
+y.^2)]);
f1f2= f1+f2;

om_xyz_surfcont(x,y,f1f2);
```




```
f1 =  0.1 * [sqrt(x.^2 +y.^2 +
one)];
```

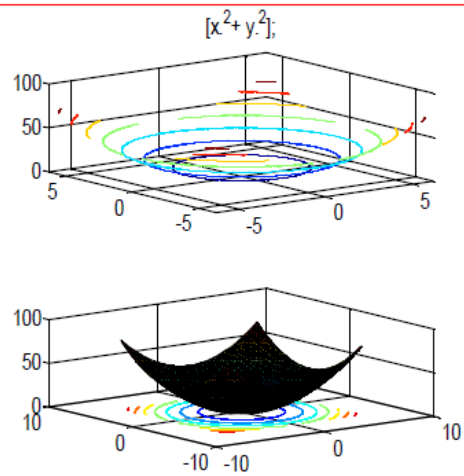



```
f2 =  -cos([2*pi* sqrt(x.^2
+y.^2)]);
```

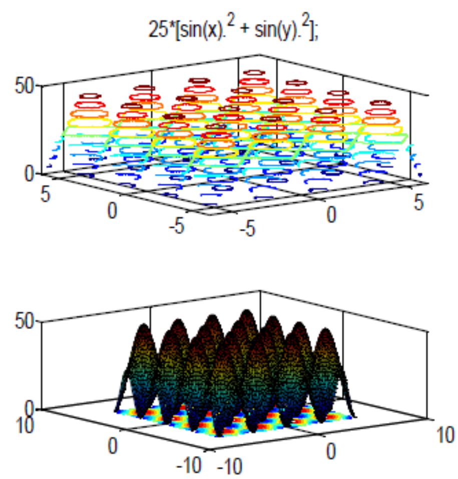



```
f1f2 = f1 + f2;
```

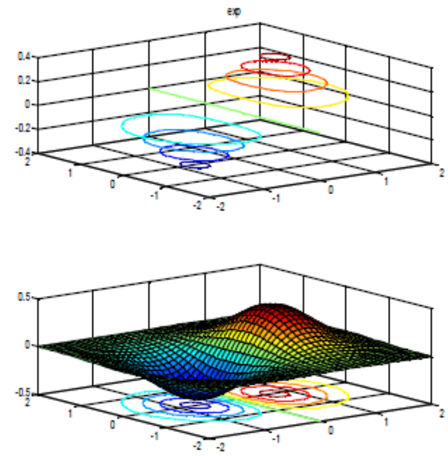
```
%
% Eggerate
%
c =2*pi;ll = -c ; ul = c; inc =
.1;
[x,y,one] = xygrid(ll,ul,inc);
fnName ='Eggerate'
[row,col] = size(x);
f1st = '[x.^2+ y.^2];'
f2st = '25*[sin(x).^2 +
sin(y).^2];';
f1 = eval(f1st);
f2 = eval(f2st);
f1f2 = f1 + f2;
om_xyz_surfcont(x,y,f1,f1st);
om_xyz_surfcont(x,y,f2,f2st);
om_xyz_surfcont(x,y,f1f2,fnName);


min.globalFnValue = 0
min.global.x      = [0,0]
```




```
f1 = [x.^2+ y.^2];
```

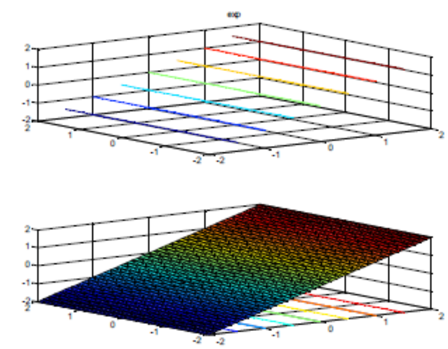



```
f2 = 25*[sin(x).^2 + sin(y).^2];
```

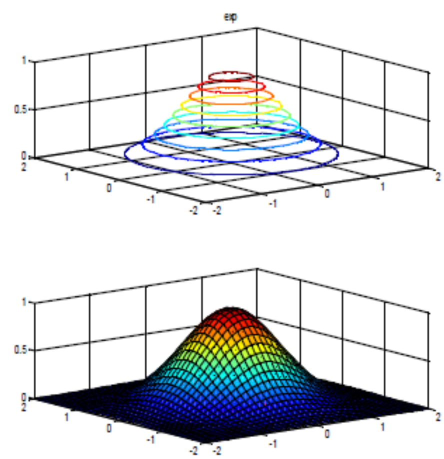


f1f2 = f1.*f2;

```
%
% exp
%
c =2;ll = -c ; ul = c; inc =  .1;
[x,y,one] = xygrid(ll,ul,inc);
fnName ='exp'
[row,col] = size(x);
f1 = x;
f2 = exp([-x.^2-y.^2]);
f1f2 = f1.*f2;


om_xyz_surfcont(x,y,f1f2);
max.globalFnValue = 0.4289
max.global.x      = [0.7072,0]
```



f1 = x;



f2 = exp([-x.^2-y.^2]);

```
%
% om_xyz_surfcont.m (R S Rao) 30-5-93; Modified 1-11-15
%
function   om_xyz_surfcont(x,y,z,fnName)
if nargin <4, fnName = '   ';end
figure, axis equal
%
subplot(2,1,1),contour3(x,y,z),title(fnName)
subplot(2,1,2),surfc(x,y,z)
```

**8. State-of-knowledge of**

Nature inspired algorithms (NIA): Each nature inspiring algorithm is a current expert scientists' brain child. They are successors of classical mathematical tools. These metaheuristics function better for NP-complete tasks in isolation or in hybrid mode with another NIA or statistical/ mathematical procedure. The chemists in particular, and scientists/ technocrats/ management personnel are at the intelligent cross roads with matured sets of Chemical toolbox of nature, synthetic organic chemists' skills/knowledge/Computational toolbox of nature and bandwagon of mathematical methods. The blue prints of futuristic highways and bridges of what to do and what not to do for health/ environment/ comfort will be visualized in the backdrop of these hyper intelligent tools.

Although all swarms of say a particular biological species need not exhibit intelligence (in the sense of computer science terminology or child prodigy from common man point of view). This happened mostly for reasons of self-defense from becoming prey and immigration. However, pheromone trails of ants in foraging activity, hive selection/waggle dance of honey bees, migrating birds to unknown places etc. are accepted as swarm-intelligence. Chart 8.1 incorporates the bat algorithm in research mode.

**Chart 8.1: Bat algorithm in research mode**

| Input Data structure | |
|---|---|
| | |
| Real | Discret |
| | Floating point |
| Quarternion | |

| # ObjFn | | |
|---|---|---|
| | | |
| Single | One-ObjFn | SOF |
| Multi- | Mul-ObjFn | MOF |

| Pareto front shape |
|---|
| |
| Convex |
| Non-convex |
| Discontinuous |
| Complicated |
| |

| Intermediate Data structure |
|---|
| |
| **Floating point** |
| Quaternion |
| Binary |
| Discrete |

| **Advances bat alg.** |
|---|
| Bat + |
| |
| Self-adaptive |
| Improved |
| Random walk direct exploitation |
| Pareto front |
| Bat inspired-Functional Link Artificial Neural Network |
| |

| Hybrid algorithms Binary |
|---|
| Bat + |
| • Quadratic programing |
| • Force fields |
| • k-means |
| • Naive Bayes |
| • Fuzzy logic |
| • Chaos |

| Hybrid algorithms Binary |
|---|
| Bat + |
| • Levy flights |
| • PSO |
| • Harmony search |
| • Differential Evolution |
| • Genetic operators |
| • NSGA-II |

| Comparison of bat alg. with |
|---|
| |
| • Firefly |
| • DE |
| • ABC |
| • ACO |
| • PSO |
| • GA |

| Comparison of bat alg. with |
|---|
| |

| Comparison of bat alg. with . |
|---|
| |
| • Invasive weed optimization |
| • Water wave optimization |
| • ⊥ Teaching Learning Based Optimization (TLBO |
| • ⊥ Plant Growth Simulation Algorithm (PGSA |

- Melody Search (MS)
- Backtracking Search (BS)
- Artificial Cooperative Search (ACS)
- Quantum behaved PSO (QPSO)
- Bat algorithm (BAT)
- Intelligent Tuned Harmony Search (ITHS)
- Cuckoo Search (CS)

| Application fields of Bat alg. |
|---|
| |
| Classificatiion |
| Design |
| Training NNs |

| Application fieldsof Bat alg. |
|---|
| Medical |
| Breast cancer |
| Artificial pancreas |
| Gene expression |

| Application fieldsof Bat alg. |
|---|
| |
| loading of fuel nuclear reactor core |
| Fuel cells |

| Application fields |
|---|
| Engineering |
| Electrical |
| Chemical |

| Application fields |
|---|
| |
| Stock price prediction |

## 9. Future scope

Bat algorithm:The directional echolocation and Doppler effect will be add onto the efficiency of the algorithm. The sensitivity analysis, rate of convergence of algorithm, optimality and existence of solution will strengthen the procedure as in any other case.

**MosqHoSA:** Due to inherent parallelism in the algorithm, it is a good start for software on chip using VLSI technology.

Mosq.OviX.alg: Mosq.oviposition.site_selection can be extended to MOO with constraints and developing a parallel algorithm to speed up calculations for large datasets

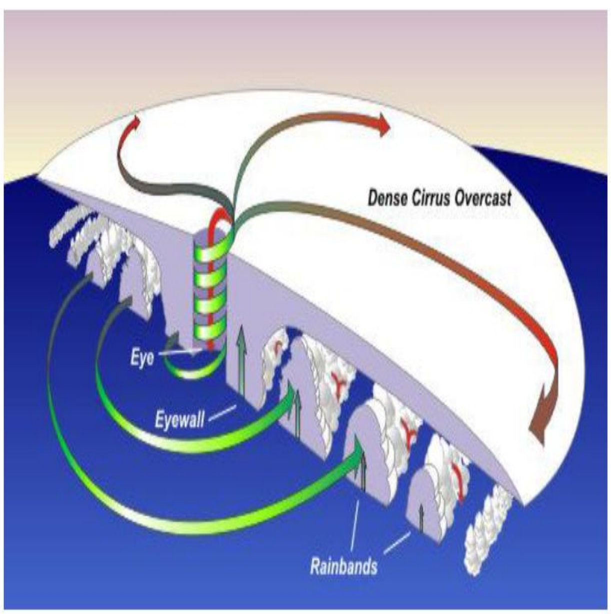
Scientists are already on the job of in planning for the predicted global warming by 2100, pollution level, unraveling secrets of genes and their relation to lifespan/ personality/ engineering to surmount diseases at fetes level, living/pleasure trips to other planets (moon, mars etc.) and synthesizing brain, understanding consciousness/mind to combat with dreaded diseases like cancer/ HIV/ mental disorders. The computational tools (neural networks, statistical probes, nature-mimicking algorithms, quantum mechanics/chemistry/physics) reached the status of instrumental probes and simulations/emulations/computations are also now experiments. The traditional experiments, brain storming computations, exciting simulations/retina quality virtual realities are interwoven even now, but will undergo renaissance to do science, to exploit technology and reap prospects of inexpensive but high ended prototype products.

In the future, a study of the complicated problems from a variety of real time scenario of social interactions and autonomous behaviors will open new vistas in computational science. The translation of such social behaviorsand not attempted chores of nature into a mathematical model would be beneficial. By 2050, if not by 2025, CPU time, memory, computer-chips will not be a concern except looking at it as functional tool, just as now we don't bother about how many cells/molecules are there in the body, brain etc. For small (yesteryears' large) tasks, accuracy and reproducibility will only matter. The available computational tools merge/gives birth to newer ones/evolve (whatever it be) into altogether a new phase. The user executing through cloud computing or (universe-computing !!), and becomes conversant tools at functional level only. A similar scenario now is that we don't need to pay attention at all about micro-details of internet/intranet technology, but it is suffice to have hands on experience of constituent black box modules.

**Appendix - NIA-1: Typical phenomenon and operators in nature**

Happenings (consequence of consequences) in nature

**Intelligent water drops**



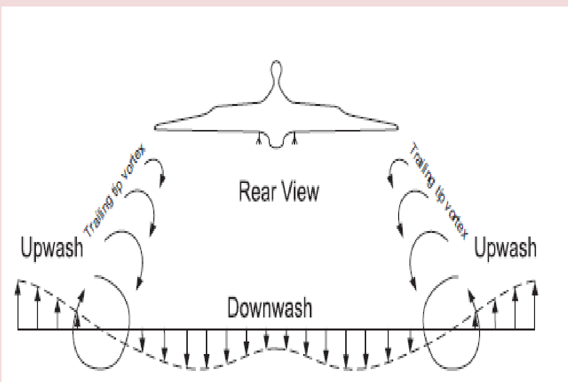

courtesy of Ref 94

Eye of Hurricane (courtesy of Ref 95 )S

| Chart NIA-1: Evolution profile of life | |
|---|---|
| Prokaryotes cells --> Homo erectus | million years ago |

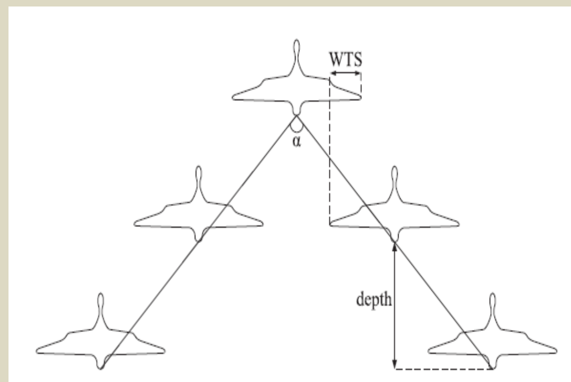
| | |
|---|---|
| Prokaryotes cells | 4,000 |
| Eukaryotes | 2,100 |
| Sponges | 600 |
| Vertebrates | 500 |
| Tetrapods | 390 |
| Synapsida | 256 |
| Reptiles | 250 |
| Placental mammal | 160 |
| Primates | 75 |
| Hominidae | 15 |
| Australopithecus Afarensis | 3.6 |
| Homo erectus | 1.8 |

Regions of upwash and downwash created by trailing vortices. migrating birds forming a effective structure with energy saving

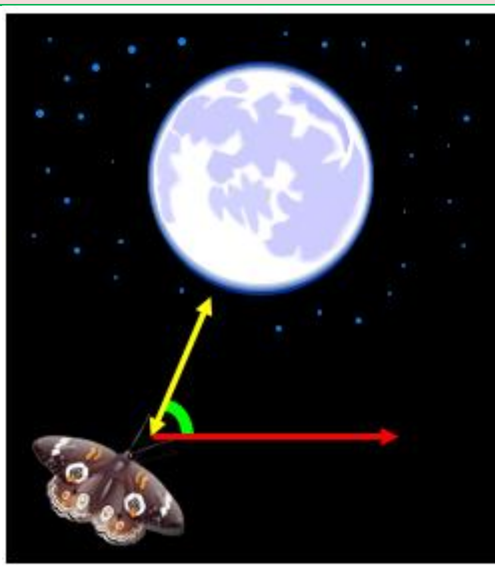



**Moth-flame Optimization Algorithm**

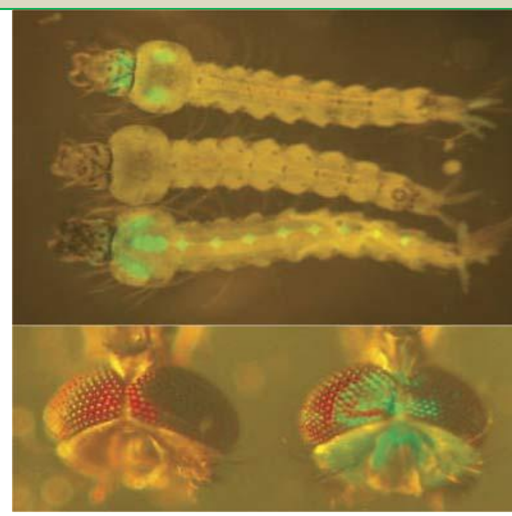
The V formation- Generation of neighbors-- leader move to end



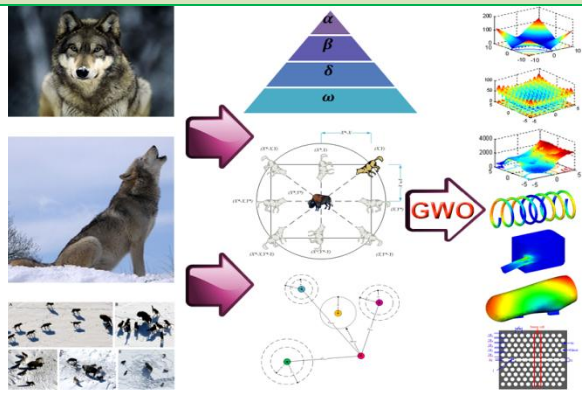
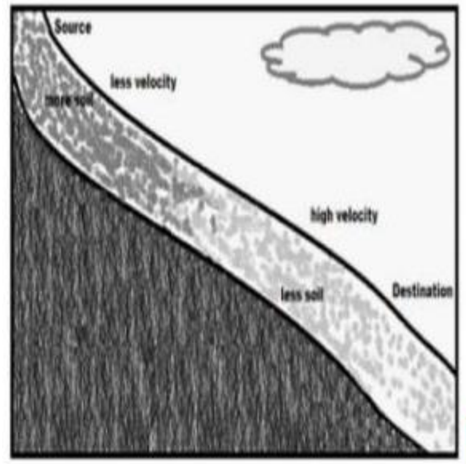
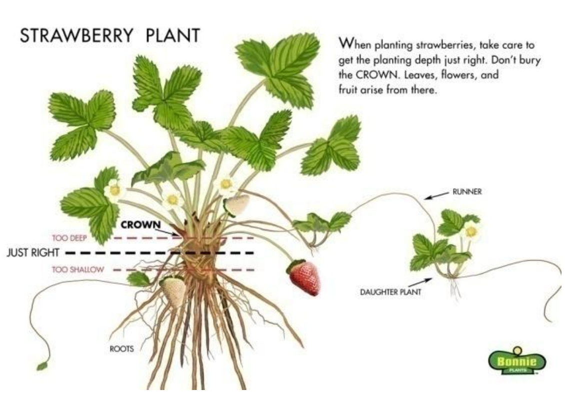
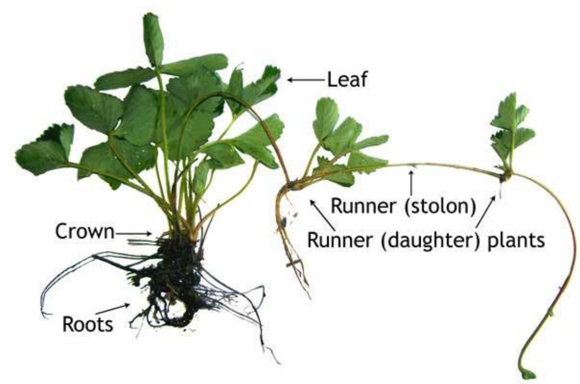

**Mosquito -eye**



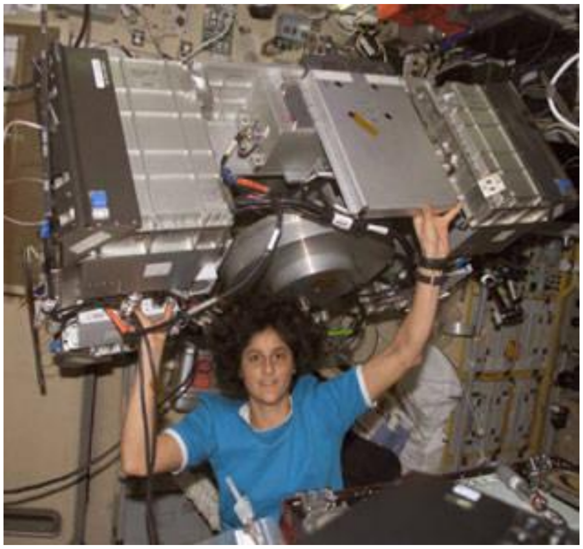
Wolf hunting prey



Water drops moving under gravitational force

<table>
<tr><td></td><td>in shortest path to destination</td></tr>
<tr><td>
</td><td>
</td></tr>
<tr><td colspan="2">**Runner-root**</td></tr>
<tr><td>
</td><td>
</td></tr>
</table>

Weight of equipment on Earth > 700 pounds on



Astronaut Suni Williams could easily move equipment in in microgravity environment

***Credit: NASA***


"Up" and "down" mean different things in space

***Credit: NASA***

## Appendix - NIA-2: Categories of Nature inspired algorithms

Scientists inspired by natural phenomena formulated physico-chemical-biological-geological laws over the last few centuries. Mathematics is the precise language based on firm theoretical ground for representation of facts, relationships in a discipline independent format. The earliest nature-inspired algorithm to mimic human brain crossed half-a-century-of-age and is in use in multiple disciplines. The inseparable-natural link between computational and experimental science is the base of innovation [1-170,179-226]. The promising recent E-man-methods of this decade require knowledge-transfer in a phase wise mode at different levels to bridge the gap. The progress in invocation of NIAs is divided into different waves.

First wave: McCulloch-Pitts [90] innovative contribution (in 1943) is in proposing a neural network to mimic Boolean (and, or) gates with fixed connection weights. The exponential number of publications in theoretical and applications is to find a way to create artificial brain [222]. In early sixties Zadeh's seminal contribution of fuzzy logic, going away from century old theory of probability is through inspiration of human thinking process. The theory of possibility and fuzzy calculations had tremendous impact on industrial to domestic products and it is not at all a dispensable tool. Simulating annealing algorithm is a powerful search technique based on annealing of glass and metals, popular chemical processes.

Second wave: Goldberg revolutionized computational science with genetic algorithm by translating survival of fittest phenomenon, the spirit of Darwinism. The crossover, mutation operators and other operators opened new vistas in facing complicated mathematical tasks. Genetic programming, evolutionary algorithm, evolutionary programming are cousins of GA and widened the scope of application. Taboo search helped to conserve energy in not repeating search in known/arrived at unfruitful regions of search space.

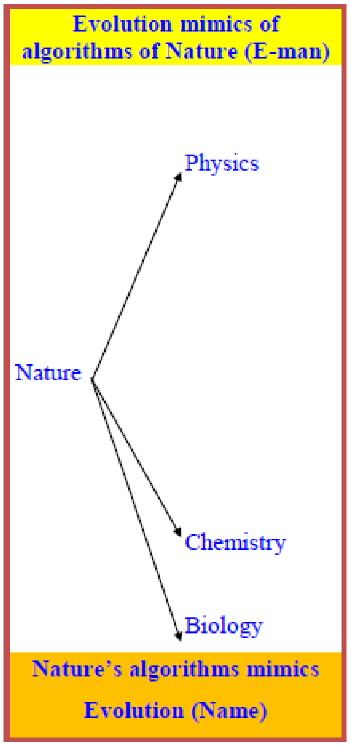
Third wave: The depth wise research from 1980s in theoretical postulates, different architectures, training algorithms of neural network reserved a niche for it. It enabled a solution for different kinds of complicated application disciplines. The unique feature is solution may be inferior, but the algorithm does not fail, as was the case with many classical chores. In 1990s, Dorigo proposed ant colony algorithm amazed at natures spirit in foraging and colony shifts of ants. The pheromone trails gave way to find shortest route between colony and food source by a swarm of ants even when faced with obstacles on their way. The unintelligent species (agents) with no leader or a priori-information could do the job routinely, definitely not by chance. It is the start of new era of swarm intelligence, a new add-on to artificial intelligence tools of 1960s. Kennedy and Eberhart came out with particle swarm optimization imbibing the smartness of flock migrating birds and fish schools. This metaheuristic had been applied extensively and many noteworthy modifications are reported to enhance its scope and speed.

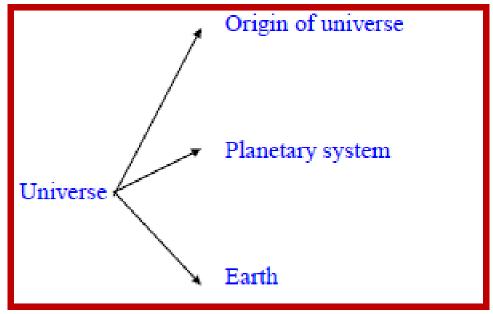
Fourth wave: Honey bee foraging is another swarm intelligent algorithm proposed by Karaboga in 2005 understanding multiple phenomena involved in search for flowers, conversion in theirs guts to honey, storing in comb by bees is no simple job. The site selection for honey comb is an intelligent decision making task and its development is optimum structure development. The foraging of bats, hunting of lions, foxes involve different modes of operation. The host seeking of mosquito for its blood meal involves self-organization. The mating of honey bee, mosquitoes brought new ideas into nature inspiring computations. The site selection by female mosquitos for egg-laying and eggs postponing their hatching under unfavorable surroundings are utilized in invoking mathematical algorithms. The tree-seed, runner-root metaheuristics are outcome of intelligent botanical phenomena.

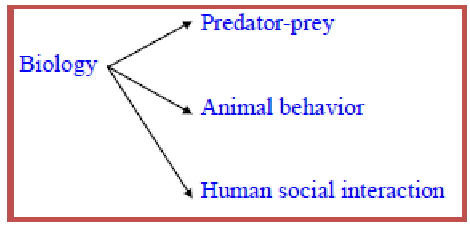
Fifth wave: The modifications and advances of each of the basic NI algorithmincreased the unique characteristics, positive features and diminished limitations. The binary-, ternary- hybridization of NIA with another NIA or with statistical/mathematical procedures brought a facelift in the intelligent computational paradigm.

Our main focus is to develop hierarchical knowledge base of equations, illustrative dictionary of variables, constants, their default values, literature reported values with success flag, developments in each method, hybrid systems, synergetic advantages, limitations, finding hole, bug fix suggestions and prospective explorative steps worth pursuing. For application scientists, it serves as a ready reckoner for the basics of each algorithm, necessary conditions, main mathematical features, failure conditions, remedial measures, unsolved issues etc. The equation base for instance is useful for display as well as to use the same string for numerical evaluation. The KBs serve for expert mode display and to write active files to uplift the number crunching into intelligent computational level. The research mode is for exploration and to answer what if? If not what is the consequence like brain storming queries. In the end-user-friendly mode, a trodden path, programs run faithfully serving within the frame of its scope. The modular comparison for a set of algorithms, at conceptual level, mathematical equations for each sub-task of algorithm will be published separately [228]. The algorithms chosen for this goal are charge system search, BBBC, firefly, mosquito, honeybee, bat, eagle, lion, PSO, SAA at the first instance.

In this review, only a gist of basic reports of NIAs is given. An object oriented knowledge base with relevant data is developed in this laboratory for main algorithms and also for most of the improvements (unpublished) and will be reported elsewhere. Many modifications are suggested and will be discussed in a separate communication. The MethodBase was developed and is in continuous upgradation. Here the major categories based on disciplines of science are depicted (vide infra). The series of papers focus the prospects of feasibility in chemical/medical sciences including diagnosis [217-228].

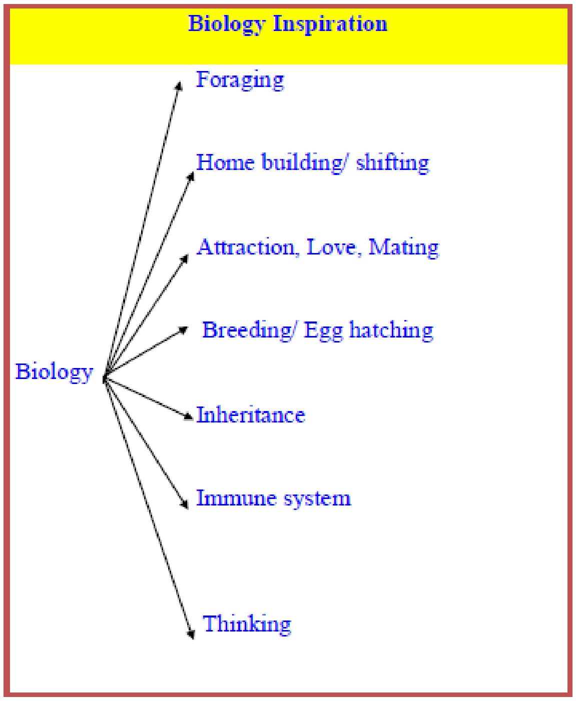


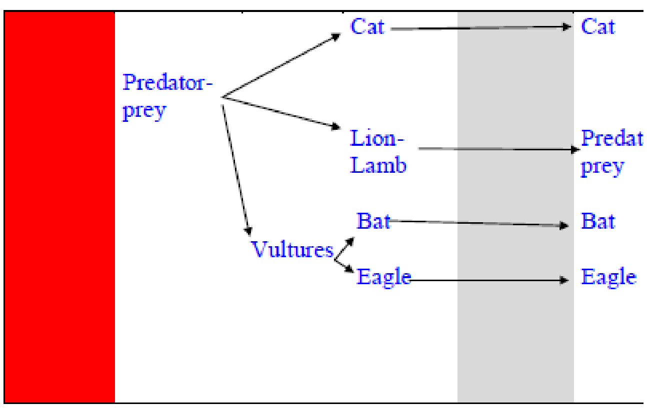

```
Nature =
Biology: [1x1 struct]
     Physics: {6x1 cell}
   Chemistry: {'Glass annealing'}
     Geology: {2x1 cell}
```

```
Nature.Physics
 =
   'Gravitational'
   'Charged system'
   'Magentic system'
   'waterdrop'
   'Cuckoo'
   'Harmony search'
```




```
Nature.Biology
=
     foraging: [1x1 struct]
  HomeBuilding: {2x1 cell}
       Mating: {'Honeybee'}
   EggHatching: {'Mosquito'}
   Inheritance: {'Human beings'}
  ImmuneSystem: {'Human beings'}
     Thinking: {'Human beings'}
```

| | |
|---|---|
| 'Biology.foraging.food' | 'Ant'<br>'Honey bee' |
| 'Biology.foraging.hunting' | 'Fire fly'<br>'Bat'<br>'Vulture'<br>'toothed Whales' |
| 'Biology.HomeBuilding' | 'Honeybee'<br>'Ant' |
| 'Biology.Mating' | 'Honeybee' |
| 'Biology.EggHatching' | 'Mosquito' |
| 'Biology.Inheritance' | 'Human beings' |
| 'Biology.ImmuneSystem' | 'Human beings' |
| 'Biology.Thinking' | 'Human beings' |




```
%
%     nialg.m   [RSRao 24-10-15] [15-5-13; 6-12-09]
%
 clean

 foraging = {'food'; 'Hunting'};
 food = {'Ant' ; 'Honey bee'};
 hunting = {'Fire fly';'Bat' ; 'Vulture' ; 'toothed Whales'};
 foraging.food = food;
 foraging.hunting = hunting;

 HomeBuilding = {'Honeybee'; 'Ant'};
 Mating = {'Honeybee'};
 EggHatching = {'Mosquito'};
 Inheritance = {'Human beings'};
```

```
Nature.Biology = Biology;
Nature.Physics = Physics;
Nature.Chemistry = Chemistry;
disp('+++++++++++++++++++++++++++++++=== ')
Nature
disp('Nature.Physics')
Nature.Physics
disp('Nature.Chemistry')
Nature.Chemistry
disp('Nature.Biology')
Biology
disp(' ====================================== ')
stf= {
'Biology.foraging.food'
```

| | |
|---|---|
| Thinking = {'Human beings'};<br>ImmuneSystem= {'Human beings'};<br><br>Biology.foraging = foraging ;<br>Biology.HomeBuilding = HomeBuilding;<br>Biology.Mating = Mating;<br>Biology.EggHatching = EggHatching;<br>Biology.Inheritance = Inheritance ;<br>Biology.ImmuneSystem = ImmuneSystem;<br>Biology.Thinking = Thinking;<br><br>Physics = {'Gravitational'; 'Charged system'; 'Magentic system';<br>'waterdrop'; 'Cuckoo'; 'Harmony search'};<br><br>Chemistry = {'Glass annealing'}; | 'Biology.foraging.hunting'<br>'Biology.HomeBuilding'<br>'Biology.Mating'<br>'Biology.EggHatching'<br>'Biology.Inheritance'<br>'Biology.ImmuneSystem'<br>'Biology.Thinking'<br>}<br><br>Biology.foraging.food<br>Biology.foraging.hunting<br>Biology.HomeBuilding<br>Biology.Mating<br>Biology.EggHatching<br>Biology.Inheritance<br>Biology.ImmuneSystem<br>Biology.Thinking<br>disp('\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\') |

## Appendix - NIA-3: Typical subsets of E-man

| | water |
|---|---|
| | |
| Water drop | → Intelligent water drop |
| Water flow | → River formation dynamics |
| River | → River course<br>→ Water cycle algorithm |
| Water flow | → Parallel physics-inspired water flow particle mechanics |
| Ocean | → Ocean wave |
| Hurricane Search | → Wind parcels move in a spiral path outward from a lower pressure warm Zone (called eye) |

| Universe |
|---|
| |
| BBBC |
| Ruin and Recreate |
| Black hole |
| Galaxy-based search |

| Sound |
|---|
| |
| Harmony search |
| Cockoo |

| Explosion |
|---|
| |
| Grenade explosion |

| Attraction forces |
|---|
| |
| Gravitational |

| |
|---|
| *Charged system search* |

- Central force optimization
- *Electro-magnetism optimization*
- optics inspired optimization

- Biogeography
- Oppositional biogeography
- Anarchic society optimization

Output
Nature.Geology:
'BioGeography'
'OppositioalBioGeography'

- Clouds (Atmospheric)

**Chemistry**

Particles diffusion
- Stochastic difusion search
- Self-propelled particles
- Good lattice swarm algorithm
- Weightless Swarm Algorithm

- Simulated annealing (glass annealing)
- Phase transition in a system of self-driven particles
- Threshold Acceptance (TA)

Shape adaptation
- Snake algorithm
- Spiral optimization

- Artificial life

- Molecular algorithm

- Cellular automaton

**Foraging:** Food is not distributed evenly all over the world and it accumulates in lumps. Thus the foraging requires different methods to find and grab amidst many seekers. The skills and methods employed by ants honey bees are diverse from those of groups of lions wolves bats etc. In the group hunting the prey is encircled first. Then group moves slowly towards it and finally catch the animal. Wolves for example do not stand in the direction of wind to avoid prey senses the smell.

**Life**

**Foraging**

| Bacteria | |
|---|---|
| | |
| Bacterial foraging | Ant |
| Bacterial-GA Foraging | Honey bee |
| Fast bacterial swarming algorithm | E-coli |
| bacterial foraging-biomimicry | |
| Appetitive Reward-Based | Feeding Behavior in Aplysia |

| | |
|---|---|
| | |
| Predator-prey | Wolf search |
| Bat | Grey wolf |
| Vulture (Egyptian) | Binary Grey wolf |
| Eagle strategy using levy walk firefly algorithms | Lion and Lamb |

| Echolation to detect position of objects such as prey |
|---|
| |
| Dolphin echolocation |
| Odontocetes (toothed whales |

| Honey bee |
|---|
| |
| ABC |
| Bee system |
| Bumble bees |
| BeeHive |
| Bees swarm optimization |
| Queen-bee evolution |

## Locomotion

| | |
|---|---|
| Hopping Kangaroo | Migrating Birds (PSO) |

## Reproduction

### Courting

| | |
|---|---|
| Firefly | Sending signals |
| Lion | Promising Security |
| Love | |

### Mating

| Reproduction(Sexual) |
|---|
| |
| Honey bee mating algorithm |
| Mosquito |
| Lion pride |

| Reproduction | Eggs care |
|---|---|

Reproduction (Asexual)

- Asexual reproduction Optimization (ARO)
- Sexual reproduction

| | |
|---|---|
| • Mosquitoes | • Oviposition |
| • Honeybess | • Broods care |

- Moth-flame
- Dragonfly
- Termite colony optimization
- Wasp

- Roach infestation algorithm

## Human intellect

Thinking

- NN
- Fuzzy
- Brainstorming process

Learning

- Teaching -learning

Language

- Grammatical evolution
- Grammatical inference on strings representing chemical compounds

- Cooperative search
- Human-inspired

- Consultant-guided search

- Anarchic society
-

- Cultural
- Group search optimizer
- Imperialist Competitive Algorithm
- League championship algorithm
- Social emotional optimization

| | |
|---|---|
| • Seeker optimization | • human understanding and searching capability for finding an optimum solution<br>• 2009 |

| |
|---|
| |
| Brain Storm Optimization |
| Bumblebees |
| Neural networks |

| |
|---|
| |
| Cat |

## Communication

| Purpose |
|---|
| |
| Foraging |
| Predator detection |
| Courting |
| Endanger |

| | |
|---|---|
| Waggle dance | Honey bees |
| Ultrasound | Bat |
| Dipole | |
| Odor | Bat, |
| Phermone | Ants |

## Evolution

| |
|---|
| |
| Evolution |
| Genetic |
| Differential evolution |
| Evolutionary ( eco-inspired) |

| Genetic |
|---|
| |
| Genetic Algorithm |
| Genetic Programming |
| |

| Genetic |
|---|
| |
| Evolutionary Algorithm |
| EP |
| Eco-inspired evolutionary algorithm |

| |
|---|
| |
| DNA computing |
| Gene expression |

| |
|---|
| |
| Immune |
| Stem Cells Optimization |

| |
|---|
| |
| Taboo search |

| | |
|---|---|
| Firefly | Glow worm |
| Furitfly | |

| |
|---|
| |
| Fish swarm/school |
| Fish-school Search |
| Great salmon run |

- Monkey
- Spider monkey

- Cellular particle swarm optimization
- Good lattice swarm optimization

- Frog
- Frog Shuffled
- Frog calling Japanese tree

- Krill Herd
- Oppositional krill herd

Flora

- Flower pollination algorithm
- Flower algorithm
- 
- Paddy Field Algorithm

- Invasive weed
- Weed colonization
- Tree seed
- runner-root

**Quantum computation**

- Quantum
- Membrane-inspired algorithms
- Membrane algorithms

- Flow field designs
- The Great Deluge Algorithm

```
%%
% iamy_2015.m   28/9/2015;   10-2-08
% Data base upgraded during 2008-2015  %%


function [xAscznia] = inp_line
%
fid = fopen ('sahasra.txt''r');
%
%
  Nmethods = 53;
for jj = 1: Nmethods
     tline1 = fgetl(fid);
     disp(' ')disp(['..........    '
int2str(jj)])
     Number = {jj};
     [nia] = line2keys(tline1);
     nia
```

```
%
%
function [nia] = line2keys(stline)
     len= length(stline);
     word = ' '; oodb= ''; nowords = 0;

for i = 1:len
     ch= stline(i);
if (ch == '' | ch=='$')
         nowords = nowords +1;
%
if nowords ==1;
               method = {word};
end
if nowords ==2
               year= {word};
end
if nowords ==3
```

| | |
|---|---|
| <pre>     znia(jj) = nia;<br>end<br>   znia<br>for i = 1:Nmethods<br>    method2(i:)      = znia(i).method;<br>    year2(i:)    = znia(i).year;<br>    authors2(i:) = znia(i).authors;<br>    insp2(i:) = znia(i).insp;<br>    xAsc(i:) = [year2(i:)method2(i:)<br>insp2(i:) authors2(i:)];<br>end<br>  fclose('all')</pre> | <pre>              authors= {word};<br>end<br>if nowords ==4<br>              insp= {word};<br>end<br>%<br><br>         oodb = {oodbword};<br>         word = '  ';<br>else<br>         word = [word ch];<br>end%if (ch)<br>end%for<br>%<br>  nia.method = method;<br>  nia.year = year;<br>  nia.authors = authors;<br>  nia.insp = insp;<br>%</pre> |

| | |
|---|---|
| function om123 [zni] = inp_line | Sahasra.txt<br>Runner-root 2015 F. Merrikh-Bayat runners and roots of some plants in nature<br>Moth-Flame 2015Seyedali Mirjalili navigation method of moths in nature called transverse orientation<br>Big bang-big Crunch  9999$ Zandi et al.  origin of universe $ $<br>Genetic Algorithm1975 Holland Darwinism<br>Transgenic Algorithm$2010$ Ruiz-Vanoye and Díaz-Parra$$$$$<br>Firefly Algorithm$2008$Yang $$$$$<br>...<br>Simulations of the evolution process$1951$Robbins and Monro $$$$$ |

Output

```
..........   1
nia =
     method: {' Runner-root'}
       year: {'   2015'}
    authors: {'   F. Merrikh-Bayat'}
       insp: {'   runners and roots of some plants in nature'}

..........   2
nia =
     method: {' Moth-Flame'}
       year: {'   2015'}
    authors: {'  Seyedali Mirjalili'}
       insp: {'   navigation method of moths in nature called transverse orientation'}

..........   3
nia =
     method: {' Big bang-big Crunch '}
       year: {'   9999'}
    authors: {'   Zandi et al. '}
       insp: {'   origin of universe '}
..........   19
nia =
     method: {' optics inspired optimization '}
       year: {'   2015'}
```

```
    authors: {'  '}
       insp: {'    '}
nia =
     method: {' Genetic Algorithm'}
       year: {'  1975'}
    authors: {'   Holland'}
       insp: {'   Darwinism '}
```

## Appendix - NIA-4: Year wise list of Nature inspired algorithms

| | | |
|---|---|---|
| 2015<br>• Moth-flame<br>• Pollination Flower<br>• optics inspired optimization | 2015<br>• Dragonfly<br>• Tree seed<br>• Runner-root | |
| 2011<br>• Cuckoo Modified | 2010<br>• Bat<br>• Transgenic Algorithm<br>• Grenade explosion | 2009<br>• Cuckoo Search<br>• Cuckoo search via l´evy flights |
| 2008<br>• Firefly algorithm | 2005<br>• Honey Bee Algorithm | 2001<br>• Harmony Search |
| 1999<br>• Extremal optimization | 1997<br>• Cross entropy method | 1995<br>• Ant colony algorithms |
| 1989<br>• Memetic | 1986<br>• Tabu Search' | 1983<br>• SAA |
| 1980<br>• Genetic programming' | 1975<br>• Genetic Algorithm | 1965<br>• Evolution Strategies |
| 1966<br>• Evolutionary programming | 1965<br>• Evolution Strategies | 1960<br>• Fuzzy logic |
| | 1943 | |

| |
|---|
| |
| • Neural network |

```
%
% iamy_2015.m   28/9/2015;   10-2-08
% Data base upgraded during 2008-2015
%
%!!!!!!!!!!! Inspiration-authors-method-year-upto-2015
%
function iamy
clean
[xAsczz] = inp_line;
 x = xAsc;
%
%
disp('!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!   Inspiration ++++++++++++++++++')
      [~Ind3] = sort((x(:3)));
      x(Ind3:)
disp('    000000000000000000000000000000000000 yearwise
999999999999999999999999999999')
%
[~Ind] = sort(str2double(x(:1)));
x(Ind:)
     disp('------------------------------           Methodwise ++++++++++++++++++')
     [~Ind2] = sort((x(:2)));
     x(Ind2:)
```

| | |
|---|---|
| '  1965' | ' Evolution Strategies' |
| '  2001' | ' Harmony Search' |
| '  2012 ' | ' Grenade explosion  ' |
| '   2015' | ' Runner-root' |
| '   2015' | ' Moth-Flame' |
| '   2015' | ' Pollination Flower |
| '   2015' | ' Dragonfly ' |
| '   2015' | ' Tree seed  ' |
| ..... | |

## Appendix - NIA-5: Hybrid-Nature inspired algorithms

In this decade hybridization of two E-man components or one nature inspired algorithm with one mathematical (or statistical) procedure excelled the performance of individual methods. This is similar to popular synergistic chemical extraction of a compound with two solvents. The binary hybridization gave birth to ternary and quaternary hybrid systems with astounding end results. The field of nature inspired computation (computational intelligence E-man knowledge based number crunching) is a cutting edge interdisciplinary field of research in mathematical science.

The hybrid algorithms with two and three components with firefly CSS gravity BBBC and neural network (SOM RecNN RBF MLP_NN ARTX ARTMAP) as the first NIA are described in our earlier publications [223-227].

| | Hybrid algorithms | | | |
|---|---|---|---|---|
| Algorithm-1 | Algorithm-2 | Algorithm-3 | | |
| ACO | DE | | | |
| ACO | | | | function om9 |
| ACO | GA | | | clean |
| ACO | SAA | | | [zzNo] = om99; |

| | | |
|---|---|---|
| BACTERIAL FORAGING | GA | |
| BBO | DE | |
| BBO | PSO | |
| DE | GA | |
| DE | PSO | |
| DE | Random walk | |
| DE | Taguchi | |
| Fuzzy adaptive | SIMPLEX | |
| Immune | Steepest ascent | |
| NN | GA | |
| PSO | AUGMENTED LAGRANGIAN | |
| PSO | Local search | |
| PSO | SA | |
| PSO | SIMPLEX | |
| PSO | Steepest descent | |
| PSO | Tabu search | |
| SAA | GA | |
| SIMPLEX | GA | |
| Taguchi | GA | |
| ACO | PSO-Fuzzy adaptive | k-means |
| ACO | SAA | Variable neighborhood search |
| DE | CovMatEvolution | Backward ray tracing |
| PSO | GA | Fuzzy logic |
| ABC | GA | |
| ABC | | |
| ABC | DE | |
| ABC | BBO | |
| GA | | |
| NN | | |

```
function om9
clean
[zzNo] = om99;

for i = 1:No

   alg1(i:)     = zz(i).alg1;
   alg2(i:)   = zz(i).alg2;
   alg3(i:)   = zz(i).alg3;

end

x
 [~Ind] = sort(str2double(x(:1)))
   x(Ind:)
return
```

```
for i = 1:No

   alg1(i:)     = zz(i).alg1;
   alg2(i:)   = zz(i).alg2;
   alg3(i:)   = zz(i).alg3;


end

x = [alg1 alg2  alg3 ];
 [~Ind] = sort(str2double(x(:1)));
 No
   x(Ind:)
return

function [zniNo] =  om99

fid = fopen ('hyb2eman.txt''r');

  No = 0;
for jj = 1: 33
     tline1 = fgetl(fid);
     disp(' ')disp(['..........   ' int2str(jj)])
        Number = {jj}
        No = No+1;
      disp([int2str(jj) '   'tline1])disp(' ')
     [ ni] = st99(tline1);
        zni(jj) = ni;
end

function [ni] = st99(tline)
     len= length(tline);
     word = ' '; nowords = 0;
     alg2 = {' '};
     alg3 = {' '};
     alg4 = {' '};
for i = 1:len

     ch= tline(i);
if (ch == '' | ch=='$')
        nowords = nowords +1;

if nowords ==1;
           alg1= {word}
end
if nowords ==2
              alg2= {word}
end

if nowords ==3
        alg3= {word}
end
         word = '  ';
```

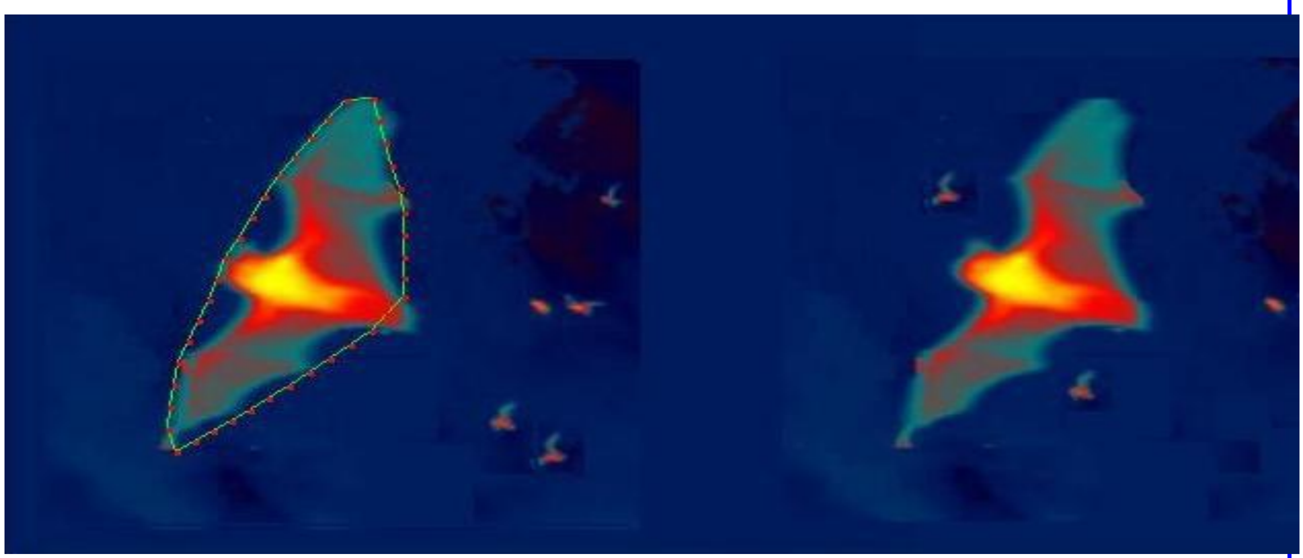
```
else
      word = [word ch];
end
end

  ni.alg1 = alg1;
  ni.alg2 = alg2;
  ni.alg3 = alg3;
```

**Search**

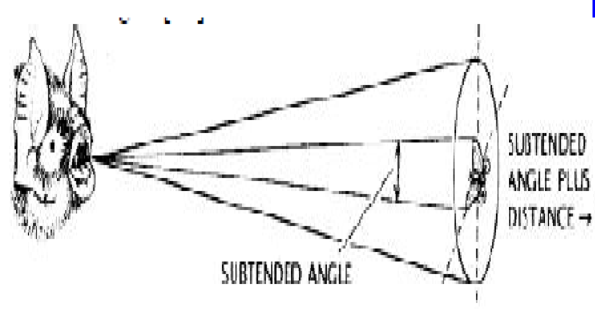
| | | |
|---|---|---|
| • Global<br>• Local<br>• Scatter | • Guided Local Search (GLS)<br>• Iterative Local Search (ILS) | • Variable Neighborhood Search (VNS)<br>• Greedy Randomized Adaptive Search Procedure (GRASP) |

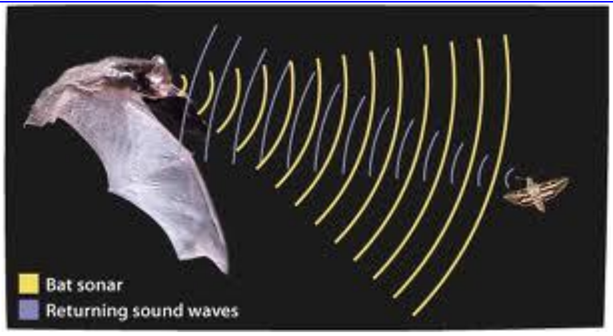
## Appendix - Bat-1: Acoustics of Echolocation in microbats

**Fig. A1_bat: Biological bliss for Bats in nature in searching their food**



**Bats hunting their preys**






**Bat transmitting sonar signal (yellow)and waiting to receive the echo (blue) from the prey (Yellow)**

**The angles to compute distance size of prey**

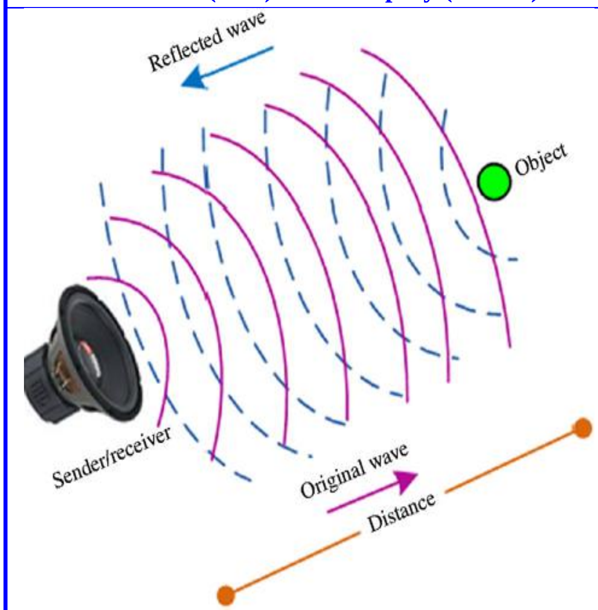
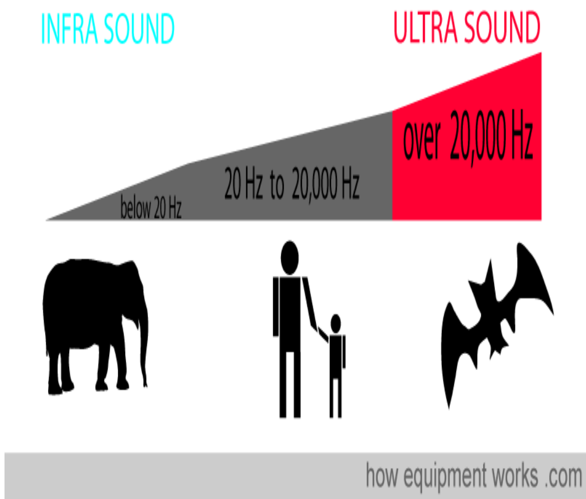
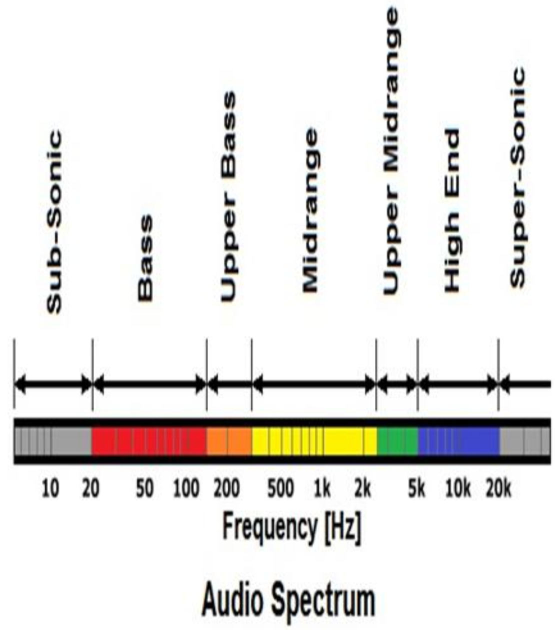
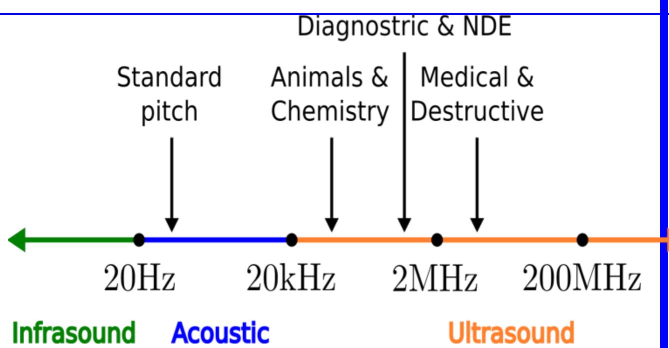


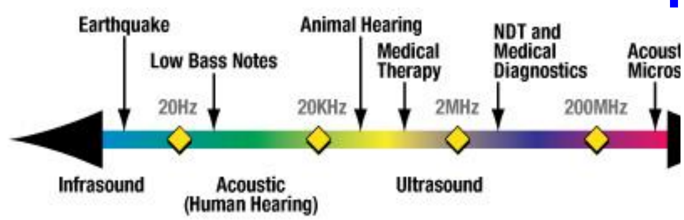

**Table A1_bat: Audible range in different species**

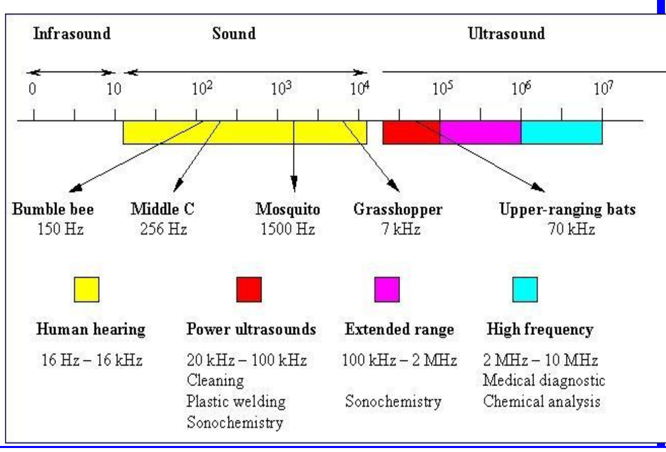
| species | Range | | | Information |
|---|---|---|---|---|
| | Hz | to | Hz. | |
| Bat | 1 k | to | 200k | ❖ Sensitive range<br>❖ 15 kHz to 90 kHz |
| Mice | 1 | to | 70 k | ❖ Large ears compared to body size |
| Dogs | 40 | to | 60k | ○ Ears<br>☞ Upright<br>☞ Curved<br>○ Tiltable<br>○ Rotatable |
| Birds | 1 k | to | 4 k | ✈ Ears are funnel-shaped<br>✈ Located slightly behind and below the eyes<br>✈ Covered with soft feathers for protection |
| Humans | 64 | | 23k | |

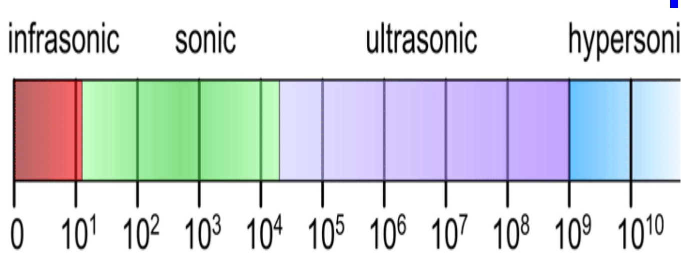

INFRA SOUND
ULTRA SOUND
over 20,000 Hz
20 Hz to 20,000 Hz
below 20 Hz
how equipment works .com


Sub-Sonic
Bass
Upper Bass
Midrange
Upper Midrange
High End
Super-Sonic
10 20 50 100 200 500 1k 2k 5k 10k 20k
Frequency [Hz]
Audio Spectrum


Diagnostric & NDE
Standard pitch
Animals & Chemistry
Medical & Destructive
20Hz
20kHz
2MHz
200MHz
Infrasound
Acoustic
Ultrasound


Earthquake
Low Bass Notes
Animal Hearing
Medical Therapy
NDT and Medical Diagnostics
Acoust Micros
20Hz
20KHz
2MHz
200MHz
Infrasound
Acoustic (Human Hearing)
Ultrasound


Infrasound
Sound
Ultrasound
0 10 $10^2$ $10^3$ $10^4$ $10^5$ $10^6$ $10^7$
Bumble bee 150 Hz
Middle C 256 Hz
Mosquito 1500 Hz
Grasshopper 7 kHz
Upper-ranging bats 70 kHz
Human hearing 16 Hz – 16 kHz
Power ultrasounds 20 kHz – 100 kHz Cleaning Plastic welding Sonochemistry
Extended range 100 kHz – 2 MHz Sonochemistry
High frequency 2 MHz – 10 MHz Medical diagnostic Chemical analysis


infrasonic
sonic
ultrasonic
hypersoni
0 $10^1$ $10^2$ $10^3$ $10^4$ $10^5$ $10^6$ $10^7$ $10^8$ $10^9$ $10^{10}$

## REFERENCES

**Typical Nature inspired algorithms**
**1943-2015**


[1] ABC,D. Karaboga, B. Basturk, J. of global optimization,**2007,** 39, 3,459–471.
A powerful and efficient algorithm for numerical function optimization: artificial bee colony (abc) algorithm.

[2] Anarchic society, H. Shayeghi,J. Dadashpour, Electrical and Electronic Eng.,**2012,**2,4,199–207.
Anarchic society optimization based pid control of an automatic voltage regulator (avr) system.

[3] Ant virtual, Xin-She Yang, J. M. Lees,C. T. Morley, Springer,**2006,**834–837.
Application of virtual ant algorithms in the optimization of cfrp shear Strengthened pre cracked structures.

[4] Backtracking,P. Civicioglu, Applied Mathematics and Computation,**2013,** 219, 15,8121–8144.
Backtracking search optimization algorithm for numerical optimization problems.

[5] Bacterial Biomimicry foraging, K. M. Passino, *Control Syst., IEEE*, **2002,**22, *3*, 52–67.
Biomimicry of bacterial foraging for distributed optimization and control.

[6] Bacterial swarming, Y. Chu, H. Mi, H. Liao, Z. JiQH Wu,*Evolutionary Computation*,***2008,***3135–3140.
A fast bacterial swarming algorithm for high-dimensional function optimization.

[7] Bacterial-GA foraging, Tai-Chen Chen, Pei-Wei Tsai, Shu-Chuan Chu,Jeng-ShyangPan, *Innovative Computing, Information and Control, 2007****,****391*–391.
A novel optimization approach: bacterial-ga foraging.

[8] Bat, X.S. Yang,*Nature Inspired Cooperative Strategies for Optimization*,**2010,**65–74.
A new metaheuristic bat-inspired algorithm.

[9] Bee colonies,R. D. Maia, L. N. de Castro,W. M. Caminhas, Evolutionary Computation (CEC),**2012,** 1–8.
Bee colonies as model for multimodal continuous optimization: The opt bees algorithm.

[10] Bee- Queen evolution,S. H. Jung, Electronics letters,**2003.** 39,6, *575*–576.
Queen-bee evolution for genetic algorithms.

[11] Bee system,P LucicD. Teodorovic, Preprints of the TRISTAN IV triennial symposium on transportation analysis,**2001,**441–445.
Bee system: modeling combinatorial optimization transportation engineering problems by swarm intelligence.

[12] Bees Cooperative swarm,H. Drias, S. Sadeg,S. Yahi, Computational Intelligence and Bioinspired Syst., Springer,**2005,** 318–325.
Cooperative bees swarm for solving the maximum weighted satisfiability problem.

[13] Big bang-big crunch optimization, Z. Zandi, E. AfjeiM. Sedighizadeh,*Power and Energy (PECon)*, **2012,**239–244.
Reactive power dispatch using big bang-big crunch optimization algorithm for voltage stability enhancement.

[14] Biogeography-based, D. Simon, Evolutionary Computation,**2008,**12,6,702–713.
Biogeography-based optimization.

[15] Biologically inspired,U. Premaratne, J. Samarabandu,T. Sidhu. Industrial and Information Syst., (ICIIS),**2009,** 279–284.

A new biologically inspired optimization algorithm.
[16] Black hole, A. Hatamlou, Information *Sciences*,**2012.**
Black hole: A new heuristic optimization approach for data clustering.
[17] Brainstorming process, Y. Shi, Int. J. of Swarm Intelligence Research,(IJSIR),**2011,**2, *4,* 35–62.
An optimization algorithm based on brainstorming process.
[18] Cat swarm optimization, S.A. Chu, P.W. Tsai,J. S. Pan,*Lecture Notes in Comput. Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, **2006,**854–858.
Cat swarm optimization.
[19] Central force, R. A. Formato,*electromagnetics Research*,**2007,**77, 425–491.
Central force optimization: A new metaheuristic with applications in applied electromagnetics.
[20] Charged system search, A. Kaveh, S. Talatahari,*Acta Mechanica*,**2010,**213,3-4,267–289.
A novel heuristic optimization method: charged system search.
[21] Clouds (atmosphere) model, Gao-Wei Yan, Zhan-Ju Hao,*International J. of Computational Intelligence and Appl.* **2013.***12*, 1350002.
A novel optimization algorithm based on atmosphere clouds model.
[22] Consultant-guided search, S. Iordache, *Proceedings of the12th annual conference on Genetic and evolutionary computation*, **2010,**225–232.
Consultant-guided search: a new metaheuristic for combinatorialoptimization problems.
[23] Cooperative search algorithm, P. Civicioglu,*Information Sciences*,**2013,**229, 58–76.
Artificial cooperative search algorithm for numerical optimization problems.
[24] Cuckoo search via levy flights, Xin-She YangS. Deb, *Nature & Biologically Inspired Computing*,***2009,***210–214.
Cuckoo search via l´evy flights.
[25] Differential evolution, R. StornK. Price, J. of global optimization,**1997,**11,4,341–359.
Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces.
[26] Differential search,P. Civicioglu, Comput.s & Geosciences,**2012,** 46,229–247.
Transforming geocentric cartesian coordinates to geodetic coordinates by using algorithm.
[27] Dolphin echolocation, A. Kaveh,N. Farhoudi, *Adv. in Eng. Software*,**2013,** 59,53–70.
A new optimization method: Dolphin echolocation.
[28] Eagle strategy + levy walk+ firefly algorithms, Xin-She YangS. Deb, *Nature Inspired Cooperative Strategies for Optimization***2010,**101–111.
Eagle strategy using levy walk and firefly algorithms for stochastic optimization.
[29] Electromagnetism optimization, Erik Cuevas, Diego Oliva, Daniel Zaldivar, Marco P´erez-Cisneros,and Humberto Sossa.*Information Sciences*,**2012,** 182(1),40–55.
Circle detection using electromagnetism optimization.
[30] Evolutionary (eco-inspired) algorithm, R.S. ParpinelliH.S. Lopes,*Nature andBiologically Inspired Computing (NaBIC)*,466–471.
An eco-inspired evolutionary algorithm applied to numerical optimization.
[31] Firefly,I. Fister, I. Fister Jr., Xin-She Yang, J. Brest, Swarm and Evolutionary Computation,**2013.**
A comprehensive review of firefly algorithms.
[32] Firefly,J. Nayak, B. Naik, H.S. Behera, Eng. Science and Technology, **2015.**
A novel nature inspired firefly algorithm with higher order neural network: Performance analysis

[33] Firefly algorithm, Xin-She Yang, *Int. J. of Bio-Inspired Computation*,**2010,**2, 2, 78–84.
Firefly algorithm, stochastic test functions and design optimisation.

[34] Fish school behaviour, F. B de Lima Neto, A.J.C.C. Lins, A. I. S. Nascimento, M. P. Lima et al.*Syst., Man and Cybernetics*,***2008,***2646–2651.
A novel search algorithm based on fish school behavior.

[35] Fish school search, C. J. A B. Filho, F. B de L. Neto, A. J. C. C. Lins, A. I. S. Nascimento,M. P. Lima, *Nature-Inspired Algorithms for Optimisation*,**2009,**261–277.
Fish school search.

[36] Fish-swarm algorithm, X.L. Li, Z.J. Shao,J.X. Qian,*Xitong Gongcheng Lilun yu Shijian/System Eng. Theory and Practice*,**2002,**22,11,32.
Optimizing method based onautonomous animats: Fish-swarm algorithm.

[37] Flower algorithm, Xin-She Yang, M. Karamanoglu,X. He,*Procedia Comput.Science*, **2013,**861–868.
Multiobjective flower algorithm for optimization.

[38] Frog calling behavior of japanese tree frogs, H. H. andez, C. Blum,*Swarm Intelligence*,**2012,** 6, 2,117–150.
Distributed graph coloring: an approach based on the calling behavior of japanese tree frogs.

[39] Frog shuffled,M. M. EusuffK.E. Lansey, J. of Water Resources Planning and Manag., **2003,** 129,3,210–225.
Optimization of water distribution network design using the shuffled frog leaping algorithm.

[40] Galaxy-based, H. Shah-Hosseini, Int. J. of Computational Science and Eng.,**2011,**6,*1*,132–140.
Principal components analysis by the galaxy-based search algorithm: a novel metaheuristic for continuous optimisation.

[41] Glow-worm,K. N. Krishnanand, D. Ghose. Swarm Intelligence Symposium,**2005,** 84–91.
Detection of multiple source locations using a glowworm metaphor with applications to collective robotics.

[42] Glow-worm, K. N. Krishnanand, D. Ghose,*International J. of Computational Intelligence Studies*, **2009,**1,93–119.
Glowworm swarm optimisation: a new method for optimising multi-modal functions.

[43] Good lattice swarm algorithm, S. Su, J. Wang, W. Fan,X. Yin,Wireless Communications, Networking and Mobile Computing,**2007,**6421–6424.
Good lattice swarm algorithm for constrained engineering design optimization.

[44] Grammatical evolution, C. Ryan, J.J Collins,M. O. Neill,Genetic Programming,**1998,**83–96.
Grammatical evolution: Evolving programs for an arbitrary language.

[45] Gravitational, E. Rashedi, H. Nezamabadi-Pour,S. Saryazdi, *Information sciences*, **2009,**179, 13,2232–2248.
Gsa: a gravitational search algorithm.

[46] Gravity rule,A. Bahrololoum, H. Nezamabadi-pour, S. Saryazdi, Eng. Appli., of Artificial Intelligence,**2015,** 45,415-428.
A data clustering approach based on universal gravity rule

[47] Great salmon run, A. Mozaffari, A. Fathi,S. Behzadipour, *Int. J. of Bio-Inspired Computation*,**2012,**45,286–301.
The great salmon run: a novel bio–inspired algorithm for artificial system design and optimisation.

[48] Grenade Explosion (GEM), A. Ahrari, A.A. Atai,Applied Soft Computing,**2010,**10,4,1132–1140.

Grenade explosion method – a novel tool for optimization of multimodal functions

[49] Group search optimizer, S. He, Q.H Wu,J.R. Saunders, *Evolutionary Computation*,**2009,**13,5,973–990.

Group search optimizer: an optimization algorithm inspired by animal searching behavior.

[50] Harmony search, Z.W. Geem, J. H. Kim,G. V. Loganathan, *Simulation*,**2001,**76,2,60–68.

A new heuristic optimization algorithm: harmony search.

[51] Honey bee, H. F. Wedde, M. Farooq,Y. Zhang,Lecture Notes in Comput. Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), **2004,**83–94.

Beehive: An efficient fault-tolerant routing algorithm inspired by honey bee behavior.

[52] Honey bees, D.T Pham, A. Ghanbarzadeh, E. Koc, S. Otri, S. Rahim, and M. Zaidi,*Proceedings of the 2nd Virtual InternationalConference on Intelligent Production Machines and Syst., (IPROMS 2006)*,**2006,**454–459.

The bees algorithm-a novel tool for complex optimisation problems.

[53] Honey fee, D. TeodorovicM. D. Orco, Advanced OR and AI Methods in Transportation**2005,**51–60.

Bee colony optimization–a cooperative learning approach to complex transportation problems.

[54] Hurricane, J. M. Edson Correia, Dissertation,.Heriot Watt Uninv, August **2015**, 64 pages.

Experimental Evaluation of the.

[55] Hurricane Search,I. Rbouh, A.A.El Imrani, *Information and Communication Systems (ICICS), 5th International Conference*, **2014,** 1–5.Hurricane Search algorithm a new model for function optimization

(b) I. Rbouh, A.A.El Imrani, *AASRI Procedia*, **2014,** 6**,** 26 – 33.

Hurricane based Optimization Algorithm

[56] Imperialist competitive, E. Atashpaz-GargariC. Lucas, *Evolutionary Computation*, ***2007,*** 4661–4667.

Imperialist competitive algorithm: an algorithm for optimization inspired by imperialistic competition.

[57] Intelligent water drops, H. Shah-Hosseini, Evolutionary Computation, **2007,** 3226–3231.

Problem solving by intelligent water drops.

[58] Kangaroo Hopping, Guan-Horng Liu, Hou-Yi Lin, Huai-Yu Lin, Shao-Tuan Chen, Pei-Chun Lin J. of Bionic Eng., **2014,** 11, 4,541-555.

A Bio-Inspired Hopping KangarooRobot with an Active Tail

[59] Krill herd, A. H. Gandomi, A. H. Alavi, *Communications in Nonlinear Science and Numerical Simulation*, **2012,** 17, 4831-4845.

Krill herd: a new bio-inspired optimization algorithm.

[60] Krill herd Oppositional,Sultana, Sneha Roy, Provas Kumar Int. J. of Electrical Power and Energy Syst.,**2015,** 73,182-191.

Oppositional krill herd algorithm for optimal location of distributed generator in radial distribution system

[61] League championship, A. H. Kashan,*Soft Computing and Pattern Recognition*,***2009,***43–48.

League championship algorithm: anew algorithm for numerical function optimization.

[62] Lion,B.R. Rajakumar, Procedia Technology,**2012,** 6,126-135.

The Lion's Algorithm: A New Nature-Inspired Search Algorithm

[63] Lion pride,Bo Wang, XiaoPing Jin, Bo Cheng, Science China Information Sciences **2012,** 55,10,2369-2389.
Optimizer: An optimization algorithm inspired by lion pride behavior

[64] Lion pride,B. R. Rajakumar, Evolutionary Computation, **2014,** 2116 – 2123.
Animal behavior evolutionary algorithm lion pride optimization swarm intelligence

[65] Monkey search, A. Mucherino, O. Seref,*Data Mining*, *Syst.*, *Analysis and Optimization in Biomedicine*,**2007,** 953, *162*–173.
Monkey search: a novelmetaheuristic search for global optimization.

[66] Monkey Spider,S. Kumar, R. Kumari, V. K. Sharma, *Procedia Comput. Science*,**2015,** 62,442-449.
Fitness Based Position Update in Spider Monkey Optimization Algorithm.

[67] P S O,J. KennedyR. Eberhart, *Neural Networks*,**1995,** 4,1942–1948.
Particle swarm optimization.

[68] Phase transition in a system of self-driven particles, T. Vicsek, A. Czirok, E. Ben-Jacob, I. Cohen, andO. Shochet, *Physical Review Letters*,**1995,**75,6,1226–1229.
Novel type of phase transition in a system of self-driven particles.

[69] Plant growth algorithm,W. Cai, W. Yang, X. Chen Intelligent Computation Technology and Automation (ICICTA), 2008 International Conference on, Volume: 1 **2008,** 1.
A Global Optimization Algorithm Based on Plant Growth Theory: Plant Growth Optimization.

[70] Pollination Flower algorithm, Xin-She Yang,*Unconventional Computation and Natural Computation*, **2012,**240–249.
Flower pollination algorithm for global optimization.

[71] River formation dynamics, P. Rabanal, I. Rodrıguez, F. Rubio,*Unconventional Computation*,**2007,**163–177.
Using river formation dynamics to design heuristic algorithms.

[72] Roach infestation optimization, T. C Havens, C. J. Spain, N. G. Salmon,J. M. Keller, *Swarm Intelligence Symposium*,***2008,***1–7.
Roach infestation optimization.

[73] Searching Stochastic,J. M. Bishop, *Artificial NeuralNetworks*, **1989,** 329–331.
Stochastic searching networks.

[74] Seeker optimization, C. Dai, W. Chen, Y. Zhu, & X. Zhang,*IEEE Transactions on Power Syst*.,**2009a,** 24, 1218–1231.
Seeker optimization algorithm for optimal reactive power dispatch.

[75] Seeker optimization, C. Dai, W. Chen, Y. Zhu, & X. Zhang,*Electric Power Syst., Research*,**2009b,**79, 10,1462–1471.
Reactive power dispatch considering voltage stability with algorithm.

[76] Simulated annealing, S. Kirkpatrick, D. Gelatt Jr., M. P Vecchi, *Science*,**1983,** 220,4598,671–680.
Optimization by simulated annealing.

[77] Smart structural syst.,R. Wang, R. C. Purshouse, P. J. Fleming, *European J. of Operational Research*,**2015,** 243,2,423-441.
Sustainable and for bridges in the 21st century. Preference-inspired co-evolutionary algorithms using weight vectors

[78] Social emotional optimization, Y. Xu, Z. Cui,J. Zeng, Swarm, *Evolutionary, and Memetic Computing*,**2010,**583–590.

Social emotional optimization algorithm for nonlinear constrained optimization problems.

[79] Spiral dynamics inspired, K. TamuraK. Yasuda,J. Adv. Computational Intelligence and Intelligent Informatics,**2011,**15, 8, 1116–1122.
Spiral dynamics inspired Optimization.

[80] Stem Cells, M. Taherdangkoo, M. Yazdi,M. H. Bagheri, D. S. Huang et al.*ICIC 2011*, *LNBI* 6840,**2012,**394–403.
Stem Cells Optimization Algorithm

[81] Teaching–learning, R.V. Rao, V.J. Savsani, D.P. Vakharia,*Information Sciences,***2012,**183,1,1–15.
Based optimization: an optimization method for continuous non-linear large scale problems,

[82] Teaching–learning-based, R.V. Rao, V.J. Savsani, D.P. Vakharia,*Comput.Aided Design*,**2011,**43, 3,303–315.
Teaching–learning-based optimization: a novel method for constrained mechanical design optimization problems,

[83] Termite colony optimization, R. Hedayatzadeh, F. A. Salmassi, M. Keshtgari, R. Akbari, and K. Ziarati, *Electrical Eng. (ICEE), 2010 18th Iranian Conference*, **2010,**553–558.
Termite colony optimization:A novel approach for optimizing continuous problems.

[84] Vulture (Egyptian) optimization, C. Sur, S. Sharma,A. Shukla,*The 9th International Conference onComputing and InformationTechnology (IC2IT2013)*, **2013,**227–237.
Algorithm–a new nature inspired metaheuristics for knapsack problem

[85] Water cycle algorithm, H. Eskandar, Ali Sadollah, A. Bahreininejad, M. Hamdi.*Comput.s & Structures*, **2012.**
Water cycle algorithm–a novel metaheuristic optimization method for solving constrained engineering optimization problems.

[86] Water drops,H. Mokhtari, Applied Soft Computing,**2015,** 26,166-179.
A nature inspired intelligent water drops evolutionary algorithm for parallel processor scheduling with rejection

[87] Weed colonization, A. R. Mehrabian, C. Lucas,*Ecological Informatics*, **2006, 1**, 4,355–366.
A novel numerical optimization algorithm inspired from weed colonization.

[88] Weightless swarm algorithm, TO Ting, Ka Lok Man, Sheng-Uei Guan, M. Nayel, and K. Wan,*Network and Parallel Computing*,**2012,**508–515.
Weightless swarm algorithm (wsa) for dynamic optimization problems.

[89] Wolf search, R. Tang, S. Fong, Xin-She Yang, S. Deb,*Digital Information Manag. (ICDIM)*,**2012.**165–172.
Wolf search algorithm with ephemeral memory.

[90] NeuralNetworks W.S. McCulloch, W. Pitts, *Bull. Math. Biophy.*, **1943,** 5, 115–133.
A logical calculus of the ideas immanent in nervous activity.

**Bat algorithm**

[91] Y.N. Vijay Kumar, S. Sivanagaraju, C. V. Suresh *Ain Shams Eng. J.,* **2015,** In Press.
Security constrained economic load dispatch in the presence of interline power flow controller using modified BAT algorithm

[92] B. B. Nayak, S. S. Mahapatra, *Eng. Science and Technology, an Int. J.,* **2015,**In Press.
Optimization of WEDM process parameters using deep cryo-treated Inconel 718 as work material

[93] M. Kalra, S. Singh, *Egyptian Informatics J.,* **2015,** In Press.

A review of metaheuristic scheduling techniques in cloud computing

[94] O. E. Turgut, M. T. Coban, *Ain Shams Eng. J.,* **2015,**In Press.
Optimal proton exchange membrane fuel cell modelling based on hybrid Teaching Learning Based Optimization – Differential Evolution algorithm.

[95] A.S. Oshaba, E.S. Ali, S.M. Abd Elazim, *Sustainable Energy, Grids and Networks,* **2015,** 2,51-60.
MPPT control design of PV system supplied SRM using BAT search algorithm.

[96] O. Hegazy, O. S. Soliman, M. Abdul Salam, *Int. J. Adv. Comput. Research,* **2015,** 5, 8, 2249.
Comparative Study between FPA, BA, MCS, ABC, and PSO Algorithms in Training and Optimizing of LS-SVM for Stock Market Prediction.

[97] S. George, *Int. J. Adv. Comput. Science and Appl,***2015,**6, 2, 159-163.
Hybrid PSO-MOBA for Profit Maximization in Cloud Computing.

[98] A. Rezaee Jordehi, *Applied Soft Computing,* **2015,** 26, 523-530.
Chaotic bat swarm optimisation (CBSO).

[99] S. Yılmaz, E. U. Küçüksille, *Applied Soft Computing,* **2015,** 28,259-275.
A new modification approach on bat algorithm for solving optimization problems.

[100] R. Hafezi, J. Shahrabi, E. Hadavandi, *Applied Soft Computing,* **2015,** 29, 196-210.
A bat-neural network multi-agent system (BNNMAS) for stock price prediction: Case study of DAX stock price.

[101] Zhi-Wei Ye, Ming-Wei Wang, W. Liu, Shao-Bin Chen, *Applied Soft Computing,* **2015,** 31,381-395.
Fuzzy entropy based optimal thresholding using bat algorithm.

[102] K. Premkumar, B.V. Manikandan, *Applied Soft Computing,* **2015,** 32,403-419.
Speed control of Brushless DC motor using bat algorithm optimized Adaptive Neuro-Fuzzy Inference System.

[103] Nien-Che Yang, Minh-Duy Le, *Applied Soft Computing,* **2015,** 35, 257-266.
Optimal design of passive power filters based on multi-objective bat algorithm and pareto front.

[104] S. Sivakumar, R. Venkatesan, *Applied Soft Computing,* **2015,** 36,506-518.
Meta-heuristic approaches for minimizing error in localization of wireless sensor networks.

[105] N. S. Jaddi, S. Abdullah, A. R. Hamdan, *Applied Soft Computing,* **2015,** 37,71-86.
Optimization of neural network model using modified bat-inspired algorithm.

[106] M. Mitić, Z. Miljković, *Expert Syst., with Appl.,* **2015,** 42,5,2624-2637.
Bio-inspired approach to learning robot motion trajectories and visual control commands.

[107] R. Svečko, D. Kusić, *Expert Syst., with Appl.* **2015,** 42,13,5416-5423.
Feedforward neural network position control of a piezoelectric actuator based on a BAT search algorithm.

[108] Xian-Bing Meng, X.Z. Gao, Yu Liu, H. Zhang, *Expert Syst., with Appl,***2015,**42,17–18,6350-6364.
A novel bat algorithm with habitat selection and Doppler effect in echoes for optimization.

[109] M. S. Kiran, *Expert Syst., with Appl.* **2015,** 42,19,6686-6698.
TSA: Tree-seed algorithm for continuous optimization

[110] S. Mishra, D. Mishra, *J. of the Korean Statistical Society,* **2015,** 44,2,247-260.
Adaptive multi-classifier fusion approach for gene expression dataset based on probabilistic theory.

[111] J. Wang, X. Fan, A. Zhao, M. Yang, *IFAC-Papers Online,* **2015,** 48,3,1708-1713.
A Hybrid Bat Algorithm for Process Planning Problem.

[112] Yu-Jun Zheng, *Comput.s & Operations Research*, **2015,** 55,1-11.
Water wave optimization: A new nature-inspired metaheuristic.

[113] K. Singh, P. Vasant, I. Elamvazuthi, R. Kannan, *Procedia Comput. Science*, **2015,** 60,1798-1808.
PID Tuning of Servo Motor Using Bat Algorithm

[114] M.R. Sathya, M. M. Thameem, Ansari, Int. *J. of Electrical Power & Energy Syst.*, **2015,** 64,365-374.
Load frequency control using Bat inspired algorithm based dual mode gain scheduling of PI controllers for interconnected power system.

[115] W. Peres, E. José de Oliveira, J. A. P. Filho, Ivo Chaves da Silva Junior, *Int. J. of Electrical Power & Energy Syst.*, **2015,** 64,419-428.
Coordinated tuning of power system stabilizers using bio-inspired algorithms.

[116] P. Dash, L. C. Saikia, N. Sinha, *Int. J. of Electrical Power & Energy Syst.*, **2015,** 68,364-372.
Automatic generation control of multi area thermal system using Bat algorithm optimized PD–PID cascade controller.

[117] B. Venkateswara Rao, G.V. Nagesh Kumar, *Int. J. of Electrical Power & Energy Syst.*, **2015,** 68,81-88.
Optimal power flow by BAT search algorithm for generation reallocation with unified power flow controller.

[118] S. Kumar Injeti, V. Kumar Thunuguntla, M. Shareef, *Int. J. of Electrical Power & Energy Syst.*, **2015,** 69,441-455.

[119] M. H. Khooban, T. Niknam, *Int. J. of Electrical Power & Energy Syst.*,**2015,**71,254-261.
A new intelligent online fuzzy tuning approach for multi-area load frequency control: Self Adaptive Modified Bat Algorithm.

[120] K. Guerraiche, M. Rahli, A. Zeblah, L. Dekhici, *Energy Procedia*, **2015,** 74,547-554.
Budgetary and Redundancy Optimization of Heterogeneous Series-Parallel Power Systems under Availability Constraints.

[121] I. Fister, S. Rauter, Xin-She Yang, K. Ljubič, I. Fister Jr *Neurocomputing*, **2015,** 149, Part B,993-1002.
Planning the sports training sessions with the bat algorithm.

[122] S. Ikeda, R. Ooka, *Applied Energy*, **2015,** 151,192-205.
Metaheuristic optimization methods for a comprehensive operating schedule of battery, thermal energy storage, and heat source in a building energy system.

[123] L. Jun, L. Liheng, Wu Xianyi, *Applied Mathematics and Computation*, **2015,** 263,361-377.
A double-subpopulation variant of the bat algorithm.

[124] S. Shamshirband, A. Tavakkoli, C. B.Roy, S. Motamedi, Song KI-IL, R. Hashim, S. M. Islam, *Powder Technology*, **2015,** 284,560-570.
Hybrid intelligent model for approximating unconfined compressive strength of cement-based bricks with odd-valued array of peat content (0–29%).

[125] M. Kang, J. Kim, Jong-Myon Kim, *Information Sciences*, **2015,** 294,423-438.
Reliable fault diagnosis for incipient low-speed bearings using fault feature analysis based on a binary bat algorithm.

[126] N. S. Jaddi, S. Abdullah, A. R. Hamdan, *Information Sciences*, **2015,** 294,628-644.
Multi-population cooperative bat algorithm-based optimization of artificial neural network model.

[127] M. Kang, J. Kim, Jong-Myon Kim, A. C.C. Tan, Eric Y. Kim, Byeong-Keun Choi, *Information Sciences,* **2015,** 311,205-206.
Corrigendum to "Reliable fault diagnosis for incipient low-speed bearings using feature analysis based on a binary bat algorithm".

[128] H. R. Ansari, A. Gholami, *Fluid Phase Equilibria,* **2015,** 402,124-132.
An improved support vector regression model for estimation of saturation pressure of crude oils.

[129] Xin-She Yang, *Nature-Inspired Optimization Algorithms,* **2014,** 141-154.
Chapter 10 - Bat Algorithms

[130] A. Alihodzic, M. Tuba, *The Scientific World J. Volume,* **2014,** Article ID 176718.
Improved Bat Algorithm Applied to Multilevel Image Thresholding

[131] I. Fister Jr., S. Fong, J. Brest, I. Fister, *The Scientific World J.* **2014,** Article ID 709738.
A Novel Hybrid Self-Adaptive Bat Algorithm.

[132] Y. Zhou, J. Xie, L. Li, M. Ma, *The Scientific World J.*, **2014,** ID 237102.
Cloud Model Bat Algorithm

[133] R. Kheirollahi, F. Namdari, *International Electrical Engineering J.* **2014,**521273-1279.
Optimal Coordination of Overcurrent Relays Based on Modified Bat Optimization Algorithm.

[134] A. H. Gandomi, Xin-She Yang, *J. of Computational Science,* **2014,** 5,2,224-232.
Chaotic bat algorithm

[135] Xin-She Yang, S. Deb, S. Fong, *Multiple-Valued Logic and Soft Computing,* **2014,**223223-237.
Bat Algorithm is better than Intermittent Search Strategy.

[136] F. S. Farias Jr., R. A. Azevedo, E. C. Rivera, W. E. Herrera, R. M. Filho, L. P. Lima Jr., *Comput. Aided Chemical Eng.,* **2014,** 33,955-960.
Product Quality Monitoring Using Extreme Learning Machines and Bat algorithms: A Case Study in Second-Generation Ethanol Production.

[137] D. Rodrigues, L. A.M. Pereira, R. Y.M. Nakamura, K. A.P. Costa, Xin-She Yang, A. N. Souza, J. P. Papa, *Expert Syst., with Appl.* **2014,** 41,5,2250-2258.
A wrapper approach for feature selection based on Bat Algorithm and Optimum-Path Forest.

[138] V. Kirubakaran, T. K. Radhakrishnan, N. Sivakumaran, *Ind. Eng. Chem. Res.,* **2014,** 53,39, 15052–15070.
Metaheuristic Patient Estimation Based Patient-Specific Fuzzy Aggregated Artificial Pancreas Design.

[139] B. Bahmani-Firouzi, R. A. Abarghooee, *Int. J. of Electrical Power & Energy Syst.,* **2014,** 56,42-54.
Optimal sizing of battery energy storage for micro-grid operation management using a new improved bat algorithm.

[140] D.K. Sambariya, R. Prasad *Int. J. of Electrical Power & Energy Syst.,* **2014,** 61, 229-238.
Robust tuning of power system stabilizer for small signal stability enhancement using metaheuristic bat algorithm.

[141] E.S. Ali *Int. J. of Electrical Power & Energy Syst.,* **2014,** 61,683-690.
Optimization of Power System Stabilizers using BAT search algorithm.

[142] M. Murali, M. Sailaja Kumari, M. Sydulu, *Int. J. of Electrical Power & Energy Syst.,* **2014,** 62,897-911.
Optimal spot pricing in electricity market with inelastic load using constrained bat algorithm.

[143] S. Kashi, A. Minuchehr, N. Poursalehi, A. Zolfaghari, *Annals of Nuclear Energy*, **2014,** 64,144-151.
Bat algorithm for the fuel arrangement optimization of reactor core.

[144] A. Kavousi-Fard, T. Niknam, *Energy*, **2014,** 64,342-354.
Multi-objective stochastic Distribution Feeder Reconfiguration from the reliability point of view

[145] O. Hasançebi, S. Carbas, *Adv. in Eng. Software*, **2014,** 67,173-185.
Bat inspired algorithm for discrete size optimization of steel frames

[146] J. Sadeghi, S. M. Mousavi, S. T. A. Niaki, S. Sadeghi, *Transportation Research Part E: Logistics and Transportation Review*, **2014,** 70,274-292.
Optimizing a bi-objective inventory model of a three-echelon supply chain using a tuned hybrid bat algorithm

[147] R. Pugazhenthi, M. A. Xavior, *Procedia Eng.*, **2014,** 97,1735-1744.
A Genetic Algorithm Applied Heuristic to Minimize the Makespan in a Flow Shop.

[148] V.P. T. Ijyas, S.M. Sameer, *Applied Mathematics and Computation*, **2014,** 230,342-358.
Low complexity metaheuristics for joint ML estimation problems.

[149] I. Fister, I. Fister Jr. *Technical Report on Faculty of Electrical Eng. and Comput. Science, Maribor, Slovenia*, **2013.**
Using the quaternion's representation of individuals in swarm intelligence and evolutionary computation.

[150] A. Shakhnaz, S. Eugene, *IEEE Congress on Evolutionary Computation*, *2013***,** 2207-2214.
Co-Operation of Biology Related Algorithms.

[151] J. Xie, Y. Zhou, H. Chen, *Computational Intelligence and Neuroscience* **2013,** Article ID 453812.
A Novel Bat Algorithm Based on Differential Operator and Lévy Flights Trajectory

[152] G. Wang, L. Guo, *Hindawi Publishing Corporatio, J. Applied Math.*, ***2013,*** *Article ID 69649.*
A Novel Hybrid Bat Algorithm with Harmony Search for Global Numerical Optimization.

[153] A.M. Taha, A. Mustapha, S. Chen, *The ScientificWorld J.* **2013,** ID 325973.
Naive Bayes-Guided Bat Algorithm for Feature Selection.

[154] I. Fister Jr., D. Fister,Xin-She Yang *Elek-trotehniski vestnik*, **2013,**80(1-2): 1–7.
A Hybrid Bat Algorithm.

[155] A. H. Gandomi, Xin-She Yang, *J. of Computational Science,* **2013,** DOI: 10.1016/j.jocs.2013.10.002.
Chaotic bat algorithm

[156] D. Rodrigues, L. A.M. Pereira, R. Y.M. Nakamura, K. A.P. Costa, Xin-She Yang, A. N. Souza, J. P. Papa, *Expert Syst., with Appl*,**2013,** In Press.
A wrapper approach for feature selection based on Bat Algorithm and Optimum-Path Forest.

[157] Xin-She Yang, *Int. J. Bio-Inspired Computation*, **2013,** 5, 3, 141–149.
Bat Algorithm: Literature Review and Applications.

[158] O. Nikolaeva Roeva,S. S. Fidanova *Biotechnology & Biotechnological Equipment*,**2013,** 27 6, 4323-4326.
Hybrid Bat Algorithm for Parameter Identification of an E. Coli Cultivation Process Model.

[159] T. Niknam, S. Sharifinia, R. Azizipanah-Abarghooee, *Energy Conversion and Manag.*, **2013,** 76,1015-1028.
A new enhanced bat-inspired algorithm for finding linear supply function equilibrium of GENCOs in the competitiveelectricity market.

[160] O. Hasançebi, T. Teke, O. Pekcan *Comput.s & Structures*, **2013,** 128,77-90.
A bat-inspired algorithm for structural optimization

[161] G. Wang, L. Guo, H. Duan, L. Liu, H. Wang, *The ScientificWorld J.*, **2012,**Article ID 418946.
A Bat Algorithm with Mutation for UCAV Path Planning

[162] S. Mishra, K. Shaw, D. Mishra, *Procedia Technology*,**2012,** 4,802-806.
A New Meta-heuristic Bat Inspired Classification Approach for Microarray Data.

[163] K. Khan, A. Sahai *I.J. Intelligent Syst., and Appl*, **2012,** 7, 23-29.
A Comparison of BA, GA, PSO, BP and LM for Training Feed forward Neural Networks in e-Learning Context.

[164] M. Ali Tawfeeq *Int. J. of Comput. Science and Information Security*,**2012,**10, 10.
Intelligent Algorithm for Optimum Solutions Based on the Principles of Bat Sonar.

[165] Xin-She Yang, A. H. Gandomi *Eng.Computations*,**2012,** 29 5 464 – 483.
Bat algorithm: a novel approach for global engineering optimization.

[166] M. Behnam, K. Hyun, S. Shaya, *The Int. J. Adv. Manufacturing Technology*, **2012,** 60,9, 1071-1086.
Bat intelligence search with application to multi-objective multiprocessor scheduling optimization.

[167] Xin-She Yang *Int. J. Bio-Inspired Computation*, **2011,** 3, 5, 267-274.
Bat Algorithm for Multi-objective Optimisation.

[168] K. Khan, A. Nikov, A. Sahai,*Advances in Intelligent and Soft Computing* **2011,** 101, 59-66.
A Fuzzy Bat Clustering Method for Ergonomic Screening of Office Workplaces.

[169] Xin-She Yang, *Nature Inspired Coop-erative Strategies for Optimization (NISCO 2010) (Eds. J. R. Gonzalez et al.), Studiesin Computational Intelligence, Springer Berlin*, **2010,** 65-74.
A New Metaheuristic Bat-Inspired Algorithm.

[170] Xian-Bing Meng, X.Z. Gao, Liu.Yu, Z. Hengzhen, *Expert Syst., With Appli.*, **2015,** 42(15), 6350-6364.
A novel bat algorithm with habitat selection and Doppler effect in echoes for optimization.

**Mosquito**algorithm

[171] X. Feng, F. C.M. Lau, H. Yu, Inf. Sci. 233: 87-108 (2013).
A novel bio-inspired approach based on the behavior of mosquitoes.

[172] F. A. A. Minhasa, M. Arif, Applied Soft Computing 11 (2011) 4614–4625. Mosq-101 MOX:
A novel global optimization algorithm inspired from Oviposition site selection and egg hatching inhibition in mosquitoes.

[173] N. Gopika, Int. J. Resear. In Computer Applications And Robotics, ISSN 2320-7345.
Optimal route queries with mosquito swarm algorithm in the road networks.

[174] W. R. Shawa, E. Teodori, S. N. Mitchell, F. Baldinib, P. Gabrielib, D. W. Rogersa, F. Catterucciaa,
Mating activates the heme peroxidase HPX15 in the sperm storage organ to ensure fertility in Anopheles gambiae.

[175] M. Buchanan, *Nature*, **2008**, 453(5), 714-716.
The mathematical mirror to animal nature.

[176] W. Gu, et al., Proc. Nat. Acad. Sci.,USA, 103 (46) (2006) 17560–17563.
Source reduction of mosquito larval habitats has unexpected consequences on malaria transmission.

[177] L. Ponnusamy, et al., Aedes aegypti, Proc. Nat. Acad. Sci.,USA, 105 (27) (2008) 9262–9267. Identification of bacteria and bacteria-associated chemical cues that mediate oviposition site preferences

[178] Racz, Andras, IEEE Global Telecomm. Conference, 2, **1999**, 1634-1640. How to build robust Call Admission Control based on on-line measurements?

**Monographs in Nature inspired algorithm N**

[179] Xin-She Yang, Z. Cui, R. Xiao, A. H. Gandom, M. Karamanoglu, *Swarm Intelligence and Bio-inspired Computation: Theory and Applications*,**2013**.

[180] B. K. Panigrahi, Meng-Hiot Lim, Y. Shi (Eds.), *Handbook of Swarm Intelligence Concepts, Principles and Applications*, Springer-Verlag Berlin Heidelberg, **2011**.

[181] V. GaziK, M. Passino, *Swarm Stability and Optimization*, Springer, **2011**.

[182] Xin-She Yang, *Nature-Inspired Metaheuristic Algorithms, 2nd Edition.*, Luniver Press, **2010.**

[183] H. Chen, Y. Zhu, K. Hu,X. He, *Hierarchical swarm model: a new approach to optimization, Discrete Dynamics in Nature and Society*, **2010.**

[184] R. Riolo (Eds.), *Genetic Programming, Trent McConaghy Una-May O'Reilly Theory and Practice VII*, Springer Science+Business Media, LLC, **2010**.

[185] S. sumathi surekha, *Computational Intelligence Paradigms Theory and Applications using MATLAB*, Taylor and Francis Group, LLC, CRC Press is an imprint of Taylor & Francis Group, an Informa business, **2010**.

[186] Xin-She Yang, *Nature-Inspired Metaheuristic Algorithms*, Luniver Press, UK, **2008**.

[187] S. Sumathi, T. Hamsapriya, P. Surekha, *Evolutionary Intelligence An Introduction to Theory and Applications with Matlab*, Springer-Verlag Berlin Heidelberg, **2008**.

[188] D. E. Goldberg, J. R. Koza, Evolutionary Algorithms for Solving Multi-Objective Problems Genetic and Evolutionary Computation Series,Springer Science+Business Media, LLC, **2007**.

[189] M. Clerc, Particle Swarm Optimization, ISTE USA 4308 Patrice Road, Newport Beach, CA 92663, USA, **2006**.

[190] D. Ashlock, Evolutionary Computation for Modeling and Optimization, Springer Science+Business Media, Inc, **2006**.

[191] M. D. T. Stutzle, Ant Colony Optimization, A Bradford Book, The MIT Press, Cambridge, Massachusetts,London, England, 2**004.**

[192] Edited by Anil Menon, Frontiers of evolutionary computation, ProductSoft, Inc. Pittsburgh, Pennsylvania, USA, Kluwer Academic Publishers, 2**004**.

[193] J. Hromkovic, Algorithmics for Hard Problems, 2nd ed. Berlin, Springer-Verlag., **2003**.

[194] J. Dréo, A. Pétrowski, P. Siarry, E. Taillard, Metaheuristiques for difficult optimization, Eyrolles, Paris, 2003.

[195] F. Glover, & G. Kochenberger, (Eds.), Handbook of Metaheuristics, Norwell, MA, Kluwer Academic Publishers, **2002**.

[196] V. Miranda, N. Fonseca, "EPSO – Evolutionary self-adapting Particle Swarm Optimization", INESC, Oporto, Portugal, **2002**.

[197] M. L. Wong, K. S. Leung, Data mining using grammar based genetic programming and applications, Kluwer Academic Publishers, New York, Boston, Dordrecht, London, Moscow,**2002**.

[198] P. Hansen, & C. Ribeiro, (Eds.), Essays and Surveys on Metaheuristics, Boston, Kluwer Academic Publishers, **2001**.

[199] J. Kennedy, R. Eberhar, Y.Shi, Swarm Intelligence, Morgan Kaufmann, Academic Press, San Fransisco, **2001**.

[200] D. Karaboga, & D. T. Pham, Intelligent Optimisation Techniques, Berlin, Springer-Verlag., **2000**.

[201] Z. Michalewicz, & D. B. Fogel, How to Solve It: Modern Heuristics, Berlin, Springer-Verlag. **2000**.

[202] R. Koza, H. B. Bennett, D. Andre, & M. A. Keane, Genetic Programming III: Darwinian Invention & Problem Solving, organ Kaufmann Publishers., **1999**.

[203] F. Glover, & M. Laguna, Tabu Search, Boston, Kluwer Academic Publishers, **1997**.

[204] D. S. Hochbaum, (Ed.), Approximation Algorithms for NP-Hard Problems, Boston, PWS Publishing Company. **1997**.

[205] T. Mitchell, Machine Learning, Boston, McGraw Hill. **1997**.

[206] J. Mockus, E. Eddy, A. Mockus, L. Mockus, & G.V. Reklaitis, Bayesian Heuristic Approach to Discrete and Global Optimization, Dordrecht, The Netherlands, Kluwer Academic Publishers, **1997**.

[207] M. Mitchell, An Introduction to Genetic Algorithms, Cambridge, MA, MIT Press, **1996**.

[208] C. Reeves, (Ed.)Modern Heuristic Techniques for Combinatorial Problems, London, McGraw Hill, **1995.**

[209] Z. Michalewicz, Genetic Algorithms, Data Structures, Evolution Programs,Berlin, SpringerVerlag, **1994.**

[210] M. Resnick, Turtles, Termites, and Traffic Jams,Cambridge, MA, MIT Press **1994.**

[211] J. R. Koza, Genetic Programming II: Automatic Discovery of Reusable Programs,The MIT Press, **1994.**

[212] M. Dorigo, Optimization, learning and natural algorithms,*Ph, D, Thesis, Politecnico di Milano, Italy*, **1992.**

[213] Martello, & P. Toth, Knapsack Problems, Algorithms and Computer Implementations, Chichester, John Wiley & Sons, **1990.**

[214] Goldberg, D. E. Genetic Algorithms in Search, Optimization and Machine Learning, Reading, MA, Addison-Wesley, **1989.**

[215] E. L. Lawler, J. K. Lenstra, A. H. G. Rinnooy Kan, & D. B.Shmoys, S. The Travelling Salesman Problem,Chichester, UK, John Wiley & Sons, **1985.**

[216] J. Holland, Adaptation in Natural and Artificial Syst, Ann Arbor, University of Michigan Press, **1975.**

**Neural networks**

[217] K RamaKrishna, Ch. V. Kameswara Rao, V. Anantha Ramam, R. Sambasiva Rao, *J. Applicable Chem.*, **2014,**3,6,2209-2311.
Mathematical Neural Network (MaNN) Models, Part VI: Single-layer perceptron [SLP] and Multi-layer perceptron [MLP] Neural networks in ChEM- Lab.

[218] K RamaKrishna, V.Anantha Ramam, R. Sambasiva Rao, *J. Applicable Chem.*, **2014,** 3,5, 1807-1893**.**
Mathematical Neural Network (MaNN) Models, Part V: Radial basis function (RBF) neural networks (NNs) in Chemometrics, Envirometrics and Medicinometrics 3D-fns.

[219] K RamaKrishna, V. Anantha Ramam, R. Sambasiva Rao, *J. Applicable Chem*., **2014**, 3 (4) 1337-1422**.**
Mathematical Neural Network (MaNN) Models Part IV: Recurrent Neural networks (RecNN) in bio-/chemical- tasks.

[220] K RamaKrishna, V. Anantha Ramam, R. Sambasiva Rao, *J. Applicable Chem.*,**2014**, 3 (3) 919-989.
Mathematical Neural Network (MaNN) Models, Part III: ART and ARTMAP in OMNI _METRICS.

[221] K RamaKrishna, V. Anantha Ramam, R. Sambasiva Rao, *J. Applicable Chem.*,**2014**, 3 (2) 834-884.
Mathematical Neural Network (MaNN) Models, Part II: Self Organizing Maps. (SOMs) in chemical sciences.

[222] I. Suryanarayana, A. Braibanti, R. Sambasiva Rao, V. AnantaRamam, D. Sudarsan, G. Nageswara Rao, *Fisheries Research*, **2008**, 92, 115-139.
Neural networks in fisheries research

**E-man**

[223] K. RamaKrishna, G. Ramkumar and R. Sambasiva Rao,*J. Applicable Chem.*, **2013,**2, 6, 1413-1458.
Evolution of Mimics of Algorithms of Nature (E-man) Part 5: Tutorial on Big_Bang–Big_Crunch algorithm.

[224] K. RamaKrishna, Ch. V. Kameswara Rao and R. Sambasiva Rao,*J. Applicable Chem.*, **2013,**2,5, 1007-1034.
E-man Part 4: Tutorial on prospects of charged system search (CSS) algorithm in chemical sciences.

[225] K. RamaKrishna, Ch. V. Kameswara Rao, R. Sambasiva Rao, *J. Applicable Chem.*, **2013,**2,4,698-713.
Eman-Part III: Tutorial on gravitational algorithm in Structure activity relationships (SXR).

[226] K. Viswanath, R. Sambasiva Rao, Ch. V. Kameswara Rao, K. Rama Krishna,B. Rama Krishna and G. E. G. Santhosh, *J. Applicable Chem.*, **2012,**1, *1*, 109-124.
Eman (Evolution of Mimics of Algorithms of Nature)-Part II: Application of neural networks for classification of bauxite.

**Swarm Intelligence**

[227] K. RamaKrishna, R. Sambasiva Rao,*J. Applicable Chem*., 2014, 3 (2) 449-492.
Swarm_Intelligence (SI)-State-of-Art (SI-SA), Part I: Tutorial on Firefly algorithm.

[228] R. Sambasiva Rao, (unpublished and private communications)

[229] E. Duman, M. Uysal, A. Alkaya, Information Sciences 217 (2012) 65–77.
Migrating Birds Optimization: A new metaheuristic approach and its performance on quadratic assignment problem

## AUTHORS' ADDRESSES

1.**K RamaKrishna**
Department of Chemistry,
Gitam Institute of Science,
Gitam University,
Visakhapatnam, A.P

2. **R. Sambasiva Rao**
School of Chemistry,
Andhra University,
Visakhapatnam 530 003, A.P