American Journal of Information Science and Computer Engineering
Vol. 1, No. 3, 2015, pp. 94-106
http://www.aiscience.org/journal/ajisce

Physics Based Metaheuristic Algorithms for Global Optimization

Umit Can[1, *], Bilal Alatas[2]

[1]Tunceli University, Department of Computer Engineering, Tunceli, Turkey
[2]Firat University, Department of Software Engineering, Elazig, Turkey

Abstract

In recent years, several optimization methods especially metaheuristic optimization methods have been developed by scientists. People have utilized power of nature to solve problems. Therefore, those metaheuristic methods have imitated physical and biological processes of nature. In 2007, Big Bang Big Crunch optimization algorithm based on evolution of universe and in 2009, Gravitational Search Algorithm based on gravity law have been proposed and have been applied to solve complex problems. However, there have been proposed many physics based algorithms afterwards. Although, many of the proposed metaheuristic optimization algorithms are known as biology based, in fact, the number of the proposed physics based metaheuristic algorithms is not less than that of algorithms based on biology. In this paper; all of the current physics based metaheuristic optimization algorithms have been searched, collected, and introduced with the performed studies.

Keywords

Metaheuristic Optimization, Physics Based Metaheuristic Optimization, Artificial Intelligence Optimization Algorithms

Received: May 5, 2015 / Accepted: May 24, 2015 / Published online: July 7, 2015

1. Introduction

In solving of complex, multimodal, high dimensional and nonlinear problems; the metaheuristic optimization methods are used. Generally, these problems can be seen in engineering, industry, business and many other areas. Scientists utilize several physical, chemical, biological laws which are helped them to improve new optimization methods.

Although there are many metaheuristic optimization methods that are based on physics, many of them are not known by scientist and there are very limited works about these methods. In this paper, physics based algorithms proposed by different scientists have been introduced by deeply exploring the related literature.

2. Gravitational Search Algorithm

Rashedi et al. proposed Gravitational Search Algorithm (GSA) in 2009 [1]. GSA is based on law of gravity and law of motion. According to the gravity law, every particle in universe attracts each other with a force and this force is proportional to their masses and inversely proportional to the square of the distance between them. In this algorithm, every agent is considered as objects. These objects' performances are measured by their masses. In this algorithm, it is expecting that at the end of the GSA run, position of the object with the heaviest mass will show the global solution. Main steps of the algorithm are shown in [Fig-1]

Studies based on GSA can be summarized as follows:

Duman et al. applied GSA to the Optimal Reactive Power

* Corresponding author
E-mail address: ucan@tunceli.edu.tr (U. Can), balatas@firat.edu.tr (B. Alatas)

Dispatch Problem (ORPD) and they showed that GSA approach indicated higher quality solution for the different objective functions [2]. Rashedi et al. applied GSA and Particle Swarm Optimization (PSO) to place Static Var Compensator (SVC) in a power system [3]. Although GSA and PSO gave the same bus and same level of compensation for the SVC placement, GSA quickly found the high-quality optimal solution in finding the location and size of SVC. GSA gives good results to solve complex power system problems. Chatterjee and Mahanti compared GSA and modified PSO for synthesis of thinned scanned concentric ring array antenna [4]. In terms of computed final fitness value and computational time GSA was better than PSO. Duman et al. applied GSA to Economic Dispatch Problem and they indicated that GSA had better effectiveness and robustness than the other used methods before [5].

Step 1: Identify the search space
Step 2: Initialize.
Step 3: Evaluate Fitness
Step 4: Update best(t), worst(t) and M(t) and G(t).
Step 5: Calculate total force in different directions.
Step 6: Calculate acceleration and velocity.
Step 7: Update agent's position.
Step 8: Repeat step 3 to 6 until the meeting stop criteria.

Fig. 1. Main steps of GSA.

3. Electromagnetism-Like Algorithm

Electromagnetism-like algorithm (EMA) has been inspired by basic electromagnetism law [6]. In this algorithm, every sample point is considered as a charged particle. In this approach, each point's charge is related to the objective function. At the same time this charge decides the magnitude of attraction or repulsion of the point which is in the sample population. To make the objective function value better means that attraction will be high. This attraction and repulsion force moves the points toward the optimality. The force which acts on each point determines the direction of movement for each point.

The EMA consist of four phases: initialization, calculation of total force acting on each particle, movement along the force direction and to finding the local minima using the neighborhood search. [Fig-2] shows the main steps of EMA.

Studies based on EMA are as follows:

Wang et al. showed that EMA was successful in training neural networks for classification problem [7]. In the comparison with back-propagation (BP) and Genetic Algorithms (GA), EMA was more efficient in terms of classification accuracy and expense of computation. Abdullah et al. applied EMA with force decay rate great deluge for course timetabling problems [8]. EMA was able to produce both feasible and good quality timetables that are consistently high quality across all the benchmark problems. Rocha and Fernandes proposed a new local search procedure based on pattern search and a population shrinking strategy to improve the EMA [9]. They compared with the original EMA. The experiments showed that the developed EMA was efficient despite the important reduction on the function evaluations. An improved electromagnetism-like algorithm (IEMA) was applied for recurrent fuzzy neural controller design by Lee et al [10]. IEMA was a multi-point searching, had faster convergence and less computational effort than EMA.

m: the number of sample points
MAXITER: maxsimum number of iterations
LSITER: maxsimum number of local search iterations
δ : local search parameter, $\delta=[0.1]$
Step 1: Initialize()
Step 2: iteration ←1
Step 3: **while** iteration < *MAXITER* **do**
Step 4: Local(*LSITER*, δ)
Step 5: **F** ←CalcF()
Step 6: Move(**F**)
Step 7: iteration ←iteration + 1
Step 8: **end while**

Fig. 2. Main steps of EMA.

4. Central Force Optimization Algorithm

Central Force Optimization (CFO) Algorithm was proposed by Formato in 2007 [11]. CFO is a metaheuristic algorithm based on metaphor of gravitational kinematics. Unlike the many other stochastic algorithms, CFO is a deterministic method. CFO doesn't require randomness in any of its calculation. CFO searches a multi-dimensional decision space. In this algorithm, probes fly along the decision space under the impression of gravity and they change their positions according to the equations of motion. In process of time, it is expecting that all probes will be trapped in close orbits of big masses with largest gravitational field. That means they slowly move towards the probe that has achieved the highest mass or fitness.

Main steps of CFO are shown in [Fig-3].

Studies based on CFO are as follows:

This algorithm has given good results in some studies. Formato reported the result of testing CFO against the PBM suite of antenna benchmarks and he showed CFO performs quite well in many respects better than the more established algorithms analyzed in PBM [12]. Green et al. contributed a ring based topology to CFO algorithm and applied on a massively threaded platform using CUDA (Compute Unified Device Architecture). The results were very good. Asi and Dib implemented a modified CFO to the optimal design of multilayer microwave absorbers in a specific frequency range [13]. The experimental results indicated that CFO was better than PSO and GSA.

Step 1: Compute the initial probe positions, the corresponding fitnesses, and assign the initial accelerations.
Step 2: Compute each probe's new position based on previously computed accelerations.
Step 3: Verify that each probe is located inside the decision space, making correction as required.
Step 4: Update the fitness at each new probe position.
Step 5: Compute the accelerations for the next time step based on new positions.
Step 6: Loop over all time steps.

Fig. 3. Main steps of CFO.

Problem is expressed by a set of node and edges in graph (N, E).
Step 1: Initialize static parameters.
Step 2: Spread the IWDs randomly on the nodes of the graph as their first visited nodes.
Step 3: Update the each IWD's visited nodes list that include the nodes just visited.
Step 4: For those IWDs repeat the step 5.1 to 5.4 with partial solutions.
Step 5.1: Choose the node *j* that will be visited after node *i*.
Step 5.2: Update the velocity of IWD which moves to node j from node i.
Step 5.3: Calculate the soil loaded by IWD on the path when it moves to node j from node i.
Step 5.4: Update the soil of the path that IWD moved to node j from node i and update soil of IWD.
Step 6: Find the iteration best solution from all solutions that found by IWDs.
Step 7: Update the soil of path that form the current best solution.
Step 8: Use the current best iteration solution to update the total best solution.
Step 9: Increase the number of iteration and go to Step 2.
Step 10: The algorithm will stop when it finds the total best solution.

Fig. 4. Main steps of IWD algorithm.

5. Intelligent Water Drops Algorithm

Intelligent Water Drops (IWD) algorithm has been suggested by Shah-Hosseini [14, 15]. This algorithm was inspired by water drops moving in a river as a big swarm. Assume that a fictional water drop flow from one point of a river to the next point of river, there will be three important changes. Velocity of water drop and soil of water drop will increase and soil of river's bed will decreased. Therefore an IWD has two important properties; velocity of IWD and the soil it carries. Consequently, a river is consisting of water drops that search a proper path for a given problem and the path which is converged by water drops is solution. The basic steps of IWD algorithm are shown in [Fig-4].

Some studies based on IWDA are as follows:

Shah-Hosseini tested IWD algorithm with the multidimensional knapsack problem, n-queen puzzle, travelling salesman problem, and automatic multilevel thresholding [15]. The experiments specified that the IWD algorithm was good at finding optimal or near optimal solutions. Hain Duan et al. applied an improved IWD algorithm for air robot path planning in complex environments [16]. The experiments indicated that this method was a useful way in air robot path planning. Kamkar et al. used IWD algorithm for solving vehicle routing problem (VRP) and they tried on 14 benchmark VRP problems [17]. As a result, they indicated that IWD algorithm converged fast to optimal solutions and found promising results.

6. River Formation Dynamics Algorithm

River Formation Dynamics Algorithm (RFDA) was based on how water forms rivers by eroding the ground and depositing sediments [18]. The water drops change the environment by increasing or decreasing the altitude of places so all solutions are given in the path of decreasing altitudes. Decreasing gradients are built and other water drops follow these gradients for construct new gradients and the new ones are reinforced. After some steps, the paths which are better from origin to target are found. Basic steps of RFDA algorithm are shown in [Fig-5].

Studies based on RFDA are as follows:

Rabanal et al. implemented RFDA to solve the dynamic Traveling Salesman Problem and they compared results with Ant Colony Optimization (ACO) [18]. The experimental results showed that (1) In both static and dynamic graphs, ACO works faster than RFDA, but in the long term RFDA obtains better solutions in both cases; (2) RFDA works faster in the dynamic case than in the static case; and (3) RFDA always obtains a solution after a modification is introduced, while sometimes ACO cannot adapt the solution constructed before changing the graph to the new scenario. These features are a consequence of the fact that the exploration of the graph is deeper in RFDA than in ACO, which in turn is due to the differences between both methods. Rabanal et al. applied RFDA to solve the minimum Steiner Tree Problem (STP) and they showed that the gradient orientation of RFDA made it suitable for solving STP [19]. They applied RFDA to benchmark graphs from the SteinLib Testdata Library. The results showed that, in 84.2% of all conducted experiments, the difference to the optimal solution was fewer than 5%. This result was better than the best performance ratio reached so far by a polynomial time algorithm.

7. Space Gravitational Algorithm

Space Gravitational Algorithm (SGA) was proposed by Hsiao et al [20]. This algorithm was inspired by simulation of several asteroids that shifting in universe to search for the heaviest mass (optimal solution) body. SGA utilizes Einstein's general theory of relativity and Newton's gravity. SGA uses the gravitational field to search the global solution. Asteroids change their positions independently so the computational complexity of algorithm is very small and the probability of searching agent to be trapped in local optima is very small. [Fig-6] shows the basic steps of SGA.

Hsiao et al. implemented SGA on an application of designing of PID controller [20]. The results showed that with small amount of searching agents SGA had better performance and lesser epoch cycles than other known methods.

Step 1: initialize Drops()
Step 2: initialize Nodes()
Step 3: **while** (not all Drops Follow The Same Path ()) and (not other Ending Condition())
Step 4: move Drops()
Step 5: erode Paths()
Step 6: deposit Sediments()
Step 7: analyze Paths()
end while

Fig. 5. Main steps of RFDA.

Step 1: A group of asteroids initialized randomly with their positions and velocities.
Step 2: In order to determine the trajectories for the asteroids detect the variation in geometry of space-time.
Step 3: After obtaining the acceleration rate update the speed of the asteroids and update the position of the asteroids.
Step 4: Update Optimal Solution and Check Stop Criterion

Fig. 6. Main steps of SGA.

8. Particle Collision Algorithm

Particle Collision Algorithm (PCA) was proposed by Wagner et al. in 2005 [21]. For solving complicated optimization problems, many optimization methods have some disadvantages. Simulated Annealing (SA) is too fragile choosing of parameter and the canonical algorithm is liable to suboptimal convergence. In order to overcome these problems PCA has been developed. PCA was inspired by nuclear collision reactions, especially scattering and absorption. The structure of PCA resembles a SA structure but it does not rely on user-defined parameters and it does not have cooling schedule. Steps of the algorithm are shown in [Fig-7].

Studies based on PCA are as follows:

Abuhamdah and Ayop applied PCA to course timetabling problems and they tested PCA on standard test benchmark course timetabling datasets of Socha [22]. Results indicated that PCA outperformed SA and Great Deluge approach in some instances and also indicated that PCA produced good quality solutions. Unlike PCA implements search for one single particle, da Luz et al. improved Multi-Particle Collision Algorithm (M-PCA) that used many particle to improve the searching [23]. In that way, the algorithm explored the search space better than PCA and did not trap in a local optimum. The test results indicated that M-PCA produced better results (computational time) in all cases analyzed.

```
Step 1: For n = 0 to # of iterations
Step 2:       Generate a stochastic perturbation of the solution
Step 3:       If Fitness(New_Config) > Fitness(Old_Config)
Step 4:             Old_Config := New_Config
Step 5:             Exploration ()
              Else
Step 6:             Scattering ()
              End If
        End For
Step 7: Exploration ()
        For n = 0 to # of iterations
Step 9:          Generate a small stochastic perturbation of the
solution
Step 10:      If Fitness(New_Config) > Fitness(Old_Config)
Step 11:            Old_Config := New_Config
              End If
        End For
        return
Step 12: Scattering ()
Step 13: If > random (0, 1)
Step 14       Old_Config := random solution
        Else
Step 15:      Exploration ();
        End if
        return
```

Fig. 7. Main steps of PCA.

```
Step 1: Form an initial generation of N candidates in a
random manner. Respect the limits of the search space.
Step 2: Calculate the fitness function values of all the
candidate solutions.
Step 3: Find the center of mass.
Step 4: Calculate new candidates around the center of
mass by adding or subtracting a normal random number
whose values decreases as the iteration eclapse.
```

Fig. 8. Main steps of BB-BC algorithm.

```
Step 1: SG Generate Initial Solution
Step 2: SG Local Search (SG)
Step 3: While (termination condition is not met) do
Step 4:   Flag False
Step 5:   Spiral Chaotic Move (SG, Flag)
Step 6:   If (Flag) then
Step 7:      SG Local Search (SG)
          End if
        End While
        Return SG
Step 8: End procedure
```

Fig. 9. Main steps of GbSA.

9. Big Bang-Big Crunch Algorithm

Erol and Eksin proposed Big Bang-Big Crunch (BB-BC) algorithm in 2005 [24]. This algorithm based on Big Bang and Big Crunch theory that is about evolution of universe. BB-BC consists of two phases: big bang and big crunch. In the big bang phase, algorithm creates an initial random population that is used in big crunch phase. In the big crunch phase, the population shrinks in to singularity. This might be a single quality solution that specified by centre of mass. Basic steps of BB-BC algorithm are shown in [Fig-8].

Studies based on BB-BC algorithm are as follows:

Genc and Hocaoglu applied BB-BC algorithm on Bearing Only Target Motion Analysis (BO-TMA) [25]. Result showed that in BO-TMA problem, BB-BC algorithm converged faster than the other common evolutionary computing methods. Alatas integrated chaos and uniform population method to BB-BC algorithm and reported promising results [26]. The reaching time of the global minima was decreased by this modification. Jaradat and Ayob implemented BB-BC algorithm to course timetabling problem [27]. Experiments showed that the algorithm could generate good quality results.

10. Galaxy Based Algorithm

Galaxy Based Search Algorithm (GbSA) has been proposed by Shah-Hosseni in 2011 [28, 29]. GbSA was inspired by spiral arms of spiral galaxies to search its nearing. It means that GbSA searches the solution space to find a better solution by using the spiral-like arm. In order to escape from local optima spiral movement is improved by chaos. In addition to that GbSA can be classified in variable neighborhood search algorithms. Basic steps of GbSA are shown in [Fig-9].

Studies based on GbSA are as follows:

Shah-Hosseini applied GbSA to the principle components analysis (PCA) problem [29]. The experimental results showed that GbSA obtain promising results for PCA estimation. Shah-Hosseini used GbSA for multilevel thresholding [28]. GbSA applied to optimize the Otsu's criterion for multilevel thresholding of gray-level images. The results showed that GbSA was so promising in this application.

```
Step 1: Creation of Universe
Step 2: Calculation of Each Body Mass
Step 3: Calculation of Gravitational Force
Step 4: Decreasing of Number of Body
Step 5: Searching for Local Improvement
Step 6: Stopping Criteria
```

Fig. 10. Main steps of BCA.

Step 1: Randomly initialize a group of search agents. Then calculate the cost of each agent and plant them into their corresponding position in search space.
Step 2: Using the equations in astrophysics and estimating the geometry of gravity field and the strength of gravitational radiation.
Step 3: The displacement of a search agent is governed by the gravitational radiation emitted from other search agents. Once the positions of all search agents are updated, an iteration of optimization is then completed. The algorithm will stop when maximum number of iteration is achieved.

Fig. 11. Main steps of IRA.

Step 1: Initialize an array of CPs with random positions and their associated velocities.
Step 2: Evaluate the values of the fitness function for the CPs, compare with each other and sort increasingly.
Step 3: Store CMS number of the first CPs and their related values of the objective function in the CM (Charged Memory).
Step 4: Determine the probability of moving each CP toward others (Rule 3), and calculate the attracting force vector for each CP (Rule 4).
Step 5: Move each CP to the new position and find the velocities (Rule 5).
Step 6: If each CP exits from the allowable search space, correct its position using
Step 7: Evaluate and compare the values of the objective function for the new CPs, and sort them increasingly.
Step 8: If some new CP vectors are better than the worst ones in the CM, include the better vectors in the CM and exclude the worst ones from the CM (Rule 6).
Step 9: Repeat search level steps until a terminating criterion is satisfied

Fig. 12. Main steps of CSSA.

11. Big Crunch Algorithm

Big Crunch Algorithm (BCA) has been proposed by Kripka and Kripka in 2008 and BCA was based on Closed Universe Theory [30]. The kinetic energy that is generated by first universe explosion (Big Bang) and it overcome the energy of attraction of bodies. Than the explosion will stop and subsequent to this shrinking or collapsing will be done. As it done initial of universe this will be end with infinite heat and intensity (Big Crunch). This process will terminate until one body remains in the universe and this will cause the fitness result. [Fig-10] shows the basic steps of the BCA:

Kripka and Kripka introduced BCA in their paper in 2008 [30]. They indicated that due to the few control parameters BCA had easy computation when compared to other popular metaheuristics and with the few modifications the adaptation of the method to the treatment of discrete variables could be done.

12. Integrated Radiation Algorithm

Chuang et al. proposed Integrated Radiation Algorithm (IRA) for solving nonlinear multidimensional optimization problems in 2007 [31]. IRA was based on the concept of gravitational radiation in Einstein's theory of general relativity. Chuang et al. used this important theory for searching optimal solution in searching space. In IRA, search space is considered as a hyperspace that has search agents and these agents distribute randomly inside the search space. The solution that is found to have better objective value is assumed to have a supernova with imperfect symmetrical shape expanding at the place. [Fig-11] shows the basic steps of IRA.

Chuang et al. applied IRA to find the minimum value of polynomial function and to optimize the design of PID controller [31]. The experimental results indicated that IRA was better than other methods and IRA solved complex optimization problems with minimum computational cost.

13. Charged System Search Algorithm

Kaveh and Talatahari proposed Charged System Search Algorithm (CSSA) in 2010 [32]. CSS was based on the Gauss and the Coulomb laws. In this algorithm each agent is represented as a Charged Particle (CP). Each particle is considered as a candidate of solution. In addition to that the law of motion is used to conduct the movement of CPs. CP is affected by other charged particles with their fitness values and separation distances. The force that acted on each CP determines its new position, velocity and acceleration. [Fig-12] shows the basic steps of CSSA.

Studies based on CSSA are as follows:

Kaveh and Talatahari applied CSSA to find the optimal design of grillage system [33]. They compared CSSA with the harmony search and the GA in two grillage systems and CSSA showed a good balance between the exploration and exploitation. Talatahari et al. added chaos to the CSS to solve mathematical global optimization problem [34]. They improved nine chaos-based CSSAs and for each of them ten different chaotic maps were used for searching the most powerful of them. The results showed that some of them had good results. Kaveh and Talatahari improved a discrete version of the CSSA to optimize truss structures with discrete variables [35]. They used several standard examples to compare with the other metaheuristics and the results indicated that CSSA had better performance than the others. In addition to that CSSA had better convergence than the others proposed methods. Kaveh and Talatahari applied CSSA to design of three frame structures [36]. They compared to some others popular metaheuristic algorithms. CSSA had better solutions in a less iteration numbers.

14. Artificial Physics Algorithm

Artificial Physics Algorithm (APA) has been proposed by Xie et al. in 2009 [37]. APA was based on Physicomimetics framework and inspired by physical forces. In this algorithm, each agent is considered as a physical particle which has a mass, velocity and position. By the virtual forces agents move toward the better fitness areas and the fitness value is correspond to the mass of agent which is user-defined. With these physical laws agents search a solution space. [Fig-13] shows the basic steps of APA.

Studies based on APA are as follows:

Xie et al. developed a vector model of APA for easily analyze the algorithm [38]. Then they came in to a way that improved the APA performance in diversity. Xie and Zeng built three force laws: negative exponential force law, unimodal force law and linear force law [39]. After they used different version of APA algorithms to solve multimodal and high dimensional problems the experimental results showed that linear force law was more effective and efficient than negative exponential force law and unimodal force law.

15. Magnetic Optimization Algorithm

Studies based on MOA are as follows:

Mirjalili and Sadiq applied MOA as a new training method for Multi Layer Perceptron (MLP) in order to improve the weakness of MLP [41]. The proposed learning method was compared with PSO and GA-based learning algorithms using 3-bit XOR and function approximation benchmark problems. The results proved the high performance of this new learning algorithm for large numbers of training samples. Mirjalili and Hashim applied the binary version of MOA named BMOA to solve problems with having discrete search space [42]. In order to reveal the performance of BMOA, four benchmark functions were applied then compared with PSO and GA. The results specified that BMOA is capable of finding global minima more accurate and faster than PSO and GA. Abidin applied MOA solving of Traveling Salesman Problem (TSP) [43]. Result obtained from the case study showed that the proposed approach managed to find a better solution compared to the solution suggested by TSPLIB.

Step 1: Initialize.
Step 2: Evaluate Fitness of Agents.
Step 3: Update the Global Best Position.
Step 4: Calculate the Mass, Component Force and Total Force.
Step 5: Update the Velocity and Position.
Step 6: Stop with termination criteria.

Fig. 13. Main steps of APA.

Step 1: All of the particles in the population are initialized randomly.
Step 2: When the termination condition is satisfied the loop will end.
Step 3: The objective of each particles is calculated and stored in the magnetic field.
Step 4: Next the normalization is performed
Step 5: The mass of all particles is calculated and stored.
Step 6: In the "for" loop, the resultant force of all forces on each particle is calculated
Step 7: At first the resultant force which is applied to a particle is set zero.
Step 8: Finding neighbors of particles.
Step 9: The force which is applied to the particle from its neighbor's is calculated.
Step 10: The force which is applied from neighbors to particles is related to distance between two particles and is calculated

Fig. 14. Main steps of MOA.

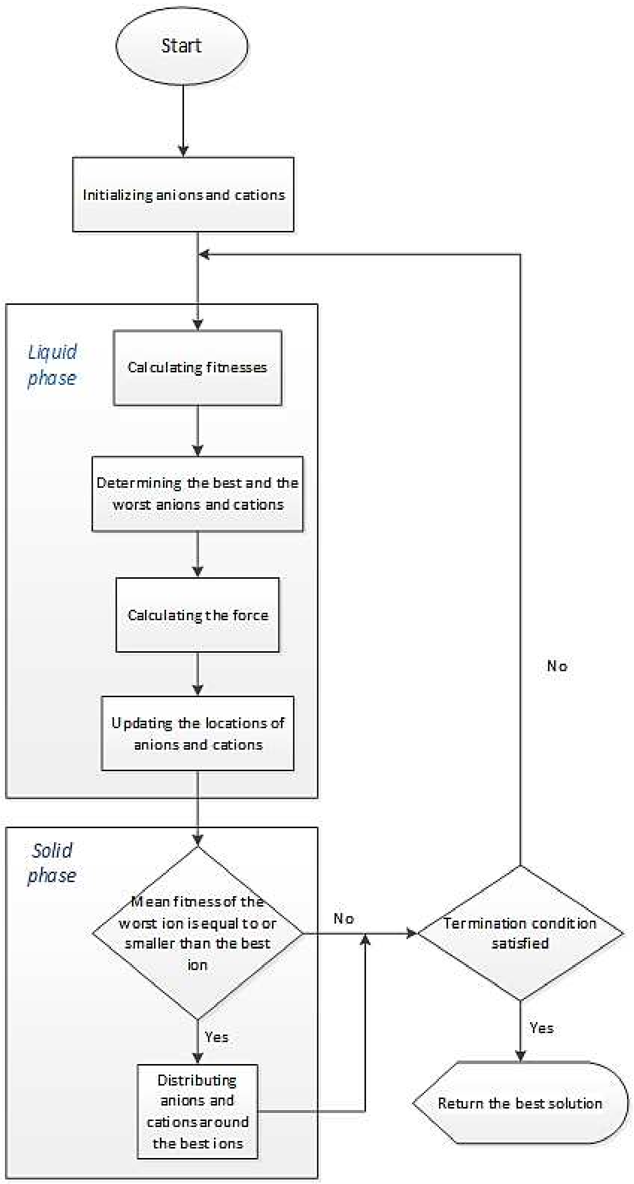

Fig. 15. Main steps of IMO.

16. Ion Motion Algorithm

Ion motion algorithm (IMO) has been recently developed inspired by the nature of the ion motion [44]. The algorithm mimics the push and pull forces of anions and cations. In the algorithm, candidate solutions of the optimization problem are divided into two groups as anions (negative ions) and cations (positive ions). Ions define candidate solutions to a particular problem and pull / push force allows the movement of ions in the search space.

Ions are calculated by calculating the fitness, so fitnesses are proportional to the value of the objective function value of the ions. Cations move towards the optimal anion while the anions move toward the optimum cation. The amount of movement depends on the size of pull / push force. The magnitude of this force indicates the acceleration of each ion. This algorithm is the most novel physics based artificial intelligence algorithm and there is not any studies except the original article. [Fig-15] shows the basic steps of IMO.

17. Gravitation Field Algorithm

Gravitational Field Algorithm (GFA) is derived from Solar Nebula Disk Model, the idea of planet formation theory widely accepted in the astronomy. The complex astronomical theory can simply be summarized as follows: A few billion years ago there has been no planet in the solar system; the world had been covered with dust. Then the dust was combined with their shots. After a long time the rocks were formed. After this time, the rocks moved quickly to meld together. Large boulders drew smaller rocks and rocks became larger. Finally planets appeared and suck the rocks. GFA has been proposed inspired from the viewpoint of this thesis [45].

First powder known as potential solutions are randomly initialized, or depending on prior knowledge weights according to the mass function space are assigned to each powder and then GFA starts. Attraction force between powders pulls the other powders by the same effect on the other powders. Thus, powders are combined, and eventually planets are formed and these are the optimal points. If global optimum solution is desired, planets are combined again the biggest planet appears.

GFA has been used for gene clustering in [45]. Parallel version of GFA based on CUDA platform has been proposed in [46].

18. Gravitational Interactions Optimization

Gravitational Interactions Optimization (GIO) [47] is very similar to the CSSA and GSA. It has been proposed by independent researchers at about the same time. This algorithm assigns mass and load to the bodies depending on the fitness function in the space inhabited by the bodies to decide on development. [Fig-16] shows the basic steps of GFA.

19. Hysteretic Optimization

Hysteretic optimization (HO) is another physics based metaheuristic optimization algorithm based on the well-known demagnetization process of magnetic materials in magnetism [48]. It inspired by the demagnetization of magnetic materials by an alternating external field of decreasing amplitude.

Studies based on HO are as follows:

HO has been proposed for well-known travelling salesman problem in [49] and promising results have been obtained. Pál has proposed the faster and simpler version of HO [50]. HO has also been used for the capacitated vehicle routing problem [51] ant it has been experimentally shown that proposed method is competitive with other popular algorithms, such as particle swarm optimization, genetic algorithms. HO is also proposed for finding ground states of Sherrington–Kirkpatrick spin glass systems [52].

20. Light Ray Optimization Algorithm

Light Ray Optimization (LRO) is inspired from the optical refraction and reflection of light rays which make the light rays travel in least time [53]. First, the search space is divided into small divisions which are full of different kinds of media, where light rays go at different velocities. Then let the velocity of light rays in each medium be the value of objective function at some point in the division. With the laws of refraction and reflection, a beam of light travels in different media to seek the optimum.

Studies based on LRO are as follows:

LRO algorithm based on annealing strategy has been introduced in [54]. The principle analysis of LRO has been performed in [55]. In [56], LRO has been used for parameter identification of ship vertical motions. Multi-objective version of LRO has been proposed in [57]. The relationship between the Euler method of ray equations of two-dimensional continuous derivable media and the iterative formula of light ray optimization were studied using the variation method and differential equation theory to obtain high convergence with high accuracy for the complex problems [58].

21. Ray Optimization

Ray Optimization (RO) has been conceptualized using the relationship between the angles of incidence and fraction based on Snell's law. In RO, each agent is modelled as a ray of light that moves in the search space in order to find the optimum solution of the complex problems [59].

Determine ranges, number of Bodies, maxIter
Step 1: bodies= initialize Particles (number of Bodies, ranges)
Step 2: **For t = 0 to maxIter do**
Step 3: compute Velocities (bodies)
Step 4: limit Velocity()
Step 5: update Position()
Step 6: limit Position()
Step 7: update Fitness()
Step 8: update Best position()
Step 9: **End For**

Fig. 16. Main steps of GFA.

Step 1 (Preparation): Select the number of search points m > 2, the parameters θ and r and the maximum number of iterations k_{max}.
Step 2 (Initialization): Initialize randomly the points; $x_i(0)$ i = 1..m; in the feasible search region and the center x^* as the point with the least fitness value.
Step 3 (Updating x_i): $x_i(k+1) = rR_2(\theta)x_i(k) - (rR_2(\theta) - I_2)x^*$ for $i = 1$...m.
Step 4 (Updating x^*): Select x^* as the point with the least fitness function in the updated set of points.
Step 5 (Check for termination criterion): If $k = k_{max}$ then stop. Otherwise, continue iteration

Fig. 17. Main steps of SOA.

Step 1: Choose the initial parameters of the WCA: N_{sr}, d_{max}, N_{pop}, max_iteration.
Step 2: Generate random initial population and form the initial streams (raindrops), rivers, and sea
Step 3: Calculate the value (cost) of each raindrops
Step 4: Determine the intensity of flow for rivers and sea
Step 5: The streams flow to the rivers
Step 6: The rivers flow to the sea which is the most downhill place
Step 7: Exchange positions of river with a stream which gives the best solution
Step 8: Similar to Step 7, if a river finds better solution than the sea, the position of river is exchanged with the sea.
Step 9: Check the evaporation condition
Step 10: If the evaporation condition is satisfied, the raining process will occur
Step 11: Reduce the value of d_{max} which is user defined parameter
Step 12: Check the convergence criteria. If the stopping criterion is satisfied, the algorithm will be stopped, otherwise return to Step 5.

Fig. 18. Main steps of WCA.

22. Spiral Optimization Algorithm

Spiral Optimization Algorithm (SOA) is a novel and current metaheuristic based on an analogy of spiral phenomena in nature [60, 61]. Main steps of the algorithm is shown in [Fig-17].

Studies based on SOA are as follows:

SOA has been used for solving combined economic and emission dispatch problem and encouraging results have been obtained [62]. SOA has also been used for minimizing the active power loss along with partial compensation of inter bus voltage drop [63]. The algorithm has been effectively used for optimal multi-objective design of digital filters in [64]. Improved version of SOA has been proposed and used in many engineering problems [65].

23. Water Cycle Algorithm

Water Cycle Algorithm (WCA) is a recently proposed physics based metaheuristic optimization algorithm inspired from nature and based on the observation of water cycle process and how rivers and streams flow to the sea in the real world [66, 67]. [Fig-18] shows the basic steps of WCA.

WCA has been used to find optimal operation strategies for the Karon-4 reservoir and a four-reservoir system in Iran [68]. Multi-objective versions of WCA has been introduced in [69, 70]. WCA has also been used for weight minimization of truss structures including discrete sizing variables [71].

24. Water Flow Algorithm

Water Flow Algorithm (WFA) is another interesting physics based algorithm proposed by simulating the hydrological cycle in meteorology and the erosion phenomenon in nature. The proposed algorithm is based on the simulation of spreading raindrops into many places on the ground, the property of water flow always moving from higher positions to lower positions, and the erosion capability of water flow on the ground. Basic operators of this algorithm are based on the raindrop distribution simulation, the property of water flow always moving to lower positions, and the erosion process to overcome obstacles [72].

WFA has been used for multi-skewed handwritten text line segmentation and promising results have been obtained [73]. In [74], flexible flow shop scheduling problem with limited or unlimited intermediate buffers has been solved with WFA and efficient results have been demonstrated.

25. Conclusions

Physics based metaheuristic optimization algorithms are efficient and robust in solving of high-dimensional and hard problems. Although there are twenty three metaheuristic algorithms originated from physics, they are not known many researchers of the related area. In this paper, all of the physics based metaheuristic algorithms are searched, collected, and their properties are introduced in a smooth way. In this way, first the creator of algorithm is told and source of inspiration is introduced. In addition to that, the basic steps of algorithms are showed. Consequently, studies about related algorithm are shown to reveal how those algorithms helped solving of complex problems.

This paper will help researchers to see the popular physics based algorithms in a simple way and give some ideas that can help in their studies.

Acknowledgements

This work is supported by the Scientific Research Project Fund of Tunceli University under the project number YLTUB013-14.

References

[1] Rashedi, E., Nezamabadi, H., Saryazdi, S., (2009). Gravitational Search Algorithm, Information Sciences, 179, 2232-2248.

[2] Duman, S. Sonmez, Y. Guvenc, U. Yorukeren, N., (2011). Application of Gravitational Search Algorithm for Optimal Reactive Power Dispatch Problem, 2011 International Symposium on Innovations in Intelligent Systems and Applications (INISTA), Istanbul, 519-523.

[3] Rashedi, E. Nezamabadi-pour, H. Saryazdi, S. M.Farsangi, M., (2007). Allocation of Static Var Compensator Using Gravitational Search Algorithm, First Joint Congress on Fuzzy and Intelligent Systems Ferdowsi University of Mashhad, Iran

[4] Chatterjee, A. and Mahanti, G.K., (2010). Comparative Performance of Gravitational Search Algorithm and Modified Particle Swarm Optimization for Synthesis of Thinned Scanned Concentric Ring Array Antenna, Progress In Electromagnetics Research B, 25, 331-348.

[5] Duman, S. Guvenc, U. Yorukeren, N., (2010). Gravitational Search Algorithm for Economic Dispatch with Valve-Point Effects, International Review of Electrical Engineering (I.R.E.E.), 5, 6

[6] Birbil, S.I., Fang, S.C., (2003). An Electromagnetism-like Mechanism for Global Optimization Journal of Global Optimization 25, 263-282.

[7] Wang, X.Y. Gao, L. Zhang, C.Y., (2008). Electromagnetism-Like Mechanism Based Algorithm for Neural Network Training, Advanced Intelligent Computing Theories and Applications. With Aspects of Artificial Intelligence Lecture Notes in Computer Science, Volume 5227/2008, 40-45.

[8] Abdullah, S. Turabeih, H. McCollum, B., (2009). Electromagnetism-like Mechanism with Force Decay Rate Great Deluge for the Course Timetabling Problem, Lecture Notes in Computer Science, Volume 5589/2009, 497-504.

[9] Ana Maria A.C. Rocha, Edite M.G.P. Fernandes, (2007). A new Electromagnetism-like Algorithm with a Population Shrinking Strategy, 6th WSEAS International Conference on System Science and Simulation in Engineering, Venice, 21-23.

[10] Lee, C.H. Chang, F.K. Lee, Y.C., (2010). An Improved Electromagnetism-like Algorithm for Recurrent Neural Fuzzy Controller Design, International Journal of Fuzzy Systems, Vol. 12, 280-290.

[11] Formato, R. A., (2007). Central force optimization: a new metaheuristic with applications in applied electromagnetics, Progress In Electronagnetics Research, PIER 77, 425-491.

[12] Formato, R.A., (2010). Central Force Optimization Applied to the PBM Suite of Antenna, The Computing Research Repository, vol. 1003, 1-89.

[13] Asi, M. J. Dib, N. I. (2010). Design of multilayer microwave broadband absorbers using central force optimization, Progress In Electromagnetics Research B, Vol. 26, 101-113.

[14] Shah-Hosseini, H., (2009a). Optimization with the Nature-Inspired Intelligent Water Drops Algorithm, In W. P. Dos Santos (Eds.), Evolutionary computation, 297-320, Austria: I-Tech, Vienna.

[15] Shah-Hosseini, H., (2009b). The intelligent water drops algorithm: a nature-inspired swarm-based optimization algorithm, Int.J Bio-Inspired Computation, Vol. 1, No. 2

[16] Duan, H. Liu, S. Lei, X., (2008). Air robot path planning based on Intelligent Water Drops optimization, IEEE International Joint Conference on Neural Networks, Hong Kong, 1397-1401.

[17] Kamkar, I. Akbarzadeh-T, M.-R. Yaghoobi, M., (2010). Intelligent Water Drops: a new optimization algorithm for solving the Vehicle Routing Problem, 2010 IEEE International Conference on Systems Man and Cybernetics (SMC), 4142-4146.

[18] Rabanal, P. Rodriguez, I. Rubio, F., (2008). Solving Dynamic TSP by using River Formation Dynamics, Fourth International Conference on Natural Computation, Jinan, 246-250.

[19] Rabanal, P. Rodriguez, I. Rubio, F., (2010). Applying River Formation Dynamics to the Steiner Tree Problem, 9th IEEE International Conference on Cognitive Informatics (ICCI), Beijing, 704-711.

[20] Hsiao, Y. T., Chuang, C., L., Jiang, J. A., Chien, C. C., (2005). A Novel Optimization Algorithm: Space Gravitational Optimization, 2005 IEEE International Conference on Systems, Man and Cybernetics, Vol. 3, 2323-2328.

[21] Wagner F. Sacco, Cassiano R.E. de Oliveria, (2005). A New Stochastic Optimization Algorithm based on a Particle Collision Metaheuristic, 6th World Congresses of Structural and Multidisciplinary Optimization. Rio de Janerio, Brazil

[22] Abuhamdah, A. Ayob, M., (2009). Experimental Result of Particle Collision Algorithm for Solving Course Timetabling Problems, IJCSNS International Journal of Computer Science and Network Security, vol 9, 134-142

[23] da Luz, E. F. P., Becceneri, J. C., de Campos Velho H., F., (2008). A new multi-particle collision algorithm for optimization in a high performance environment, Journal of Computational Interdisciplinary Sciences, 1(1): 3-10.

[24] Genç, H.M. Eksin, I. Erol, O.K., (2010). Big Bang - Big Crunch Optimization Algorithm Hybridized With Local Directional Moves and Application to Target Motion Analysis Problem, 2010 IEEE International Conference on Systems Man and Cybernetics (SMC), 881 – 887.

[25] Genc, H.M. Hocaoglu, A.K., (2008). Bearing-Only Target Tracking Based on Big Bang – Big Crunch Algorithm, The Third International Multi-Conference on Computing in the Global Information Technology, Athens, 229-233.

[26] Alatas, B., (2011). Uniform Big Bang–Chaotic Big Crunch Optimization, Communication Nonlinear Science and Numerical Simulation, 16(9), 3696-3703.

[27] Jaradat, G.M. Ayob, M., (2010). Big Bang-Big Crunch Optimization Algorithm to Solve the Course Timetabling Problem, 10th International Conference on Intelligent Systems Design and Applications (ISDA), Cairo, 1448-1452.

[28] Shah-Hosseini, H. (2011a). Otsu's Criterion-based Multilevel Thresholding by a Nature-inspired Metaheuristic called Galaxy-based Search Algorithm, 2011 Third World Congress on Nature and Biologically Inspired Computing (NaBIC), Salamanca, 383-388.

[29] Shah-Hosseini, H., (2011b). Principal components analysis by the galaxy-based search algorithm: a novel metaheuristic for continuous optimisation, Int. J. Computational Science and Engineering, Vol. 6, 132-140.

[30] Kripta, M. M.L. Kripta. R., (2008). Big Crunch Optimization Method, International Conference on Engineering Optimization, Rio de Janeiro

[31] Chuang, C. Cl., Jiang, J. A., (2007). Integrated radiation optimization: inspired by the gravitational radiation in the curvature of space-time, IEEE Congress on Evolutionary Computation, Singapore, 3157-3164.

[32] Kaveh, A., Talatahari, S., (2010a). A novel heuristic optimization method: charged system Search, Acta Mechanica, Volume 213, 267-289

[33] Kaveh, A., Talatahari, (2010b). Charged system search for optimum grillage system design using the LRFD-AISC code, Journal of Constructional Steel Research, Volume 66, 767-771.

[34] Talatahari, S., Kaveh, A., (2011). Sheikholeslami, R., An efficient charged system search using chaos for global optimization problems, International Journal of Optimization in Civil Engineering, No. 2, 1, 305-325.

[35] Kaveh, A., Talatahari, (2010c). A Charged System Search with a Fly to Boundary Method for Discrete Optimum Design of Truss Structures, Asian Journal of Civil Engineering, Vol. 11, 277-293.

[36] Kaveh, A., Talatahari, (2012). Charged system search for optimal design of frame structures, Applied Soft Computing, Volume 12, January 2012, 382-393.

[37] Xie, L., Zeng J., Cui, Z., (2009). General framework of Artificial Physics Optimization Algorithm, World Congress on Nature & Biologically Inspired Computing, Coimbatore, 1321-1326.

[38] Xie, L., Zeng, J., Cui, Z., (2009). The Vector Model of Artificial Physics Optimization Algorithm for Global Optimization Problems, Intelligent Data Engineering and Automated Learning, Volume 5788, 610-617.

[39] Xie, LP, Zeng, JC, (2010). The Performance Analysis of Artificial Physics Optimization Algorithm Driven by Different Virtual Forces, ICIC Express Letters (ICIC-EL).

[40] Tayarani M. H., Akbarzadeh, M.R., (2008). Magnetic Optimization Algorithms a new synthesis, in IEEE Congress on Evolutionary Computation, 2659-2664.

[41] Mirjalili, S., Sadiq, A. S., (2011). Magnetic Optimization Algorithm for training Multi Layer Perceptron IEEE conference on Communication Software and Networks (ICCSN), 42-46.

[42] Mirjalili, S., Hashim, S. Z. M., (2012). BMOA: Binary Magnetic Optimization Algorithm, International Journal of Machine Learning and Computing, Vol. 2, No. 3, 204-208.

[43] Abidin, A. F. Z., (2012). Magnetic Optimization Algorithm Approach For Travelling Salesman Problem, World Academy of Science, Engineering and Technology 62, 1393-1397.

[44] Javidy B. et al., (2015). Ions motion algorithm for solving optimization problems, Applied Soft Computing, 32, 72-79.

[45] Zheng, M. Et al.,(2010). Gravitation Field Algorithm And Its Application in Gene Cluster, Algorithms for Molecular Biology, 5:32, 1-11.

[46] Rong, G., et al., (2013). Parallel Gravitation Field Algorithm Based on the CUDA Platform, Journal of Information & Computational Science, 10:12, 3635-3644.

[47] Flores, J. J., López, R., Barrera, J. (2011). Gravitational Interactions Optimization, Learning and Intelligent Optimization Lecture Notes in Computer Science, 6683, 226-237.

[48] Zaránd, G., Pázmándi, F., Pál, K. F., and Zimányi, G. T., (2002.) Using Hysteresis for Optimization, Phys. Rev. Lett. 89, 150201.

[49] Pál, K. F., (2003). Hysteretic optimization for the traveling salesman problem, Physica A: Statistical Mechanics and its Applications, Volume 329, Issues 1–2, 1 287–297.

[50] Pál, K. F., (2006). Hysteretic optimization, faster and simpler, Physica A: Statistical Mechanics and its Applications, Volume 360, Issue 2, 525–533.

[51] Yan, X., Wu, W., (2012). Hysteretic optimization for the capacitated vehicle routing problem, 9th IEEE International Conference on Networking, Sensing and Control (ICNSC), 12-15

[52] Pál, K. F., (2006). Hysteretic optimization for the Sherrington–Kirkpatrick spin glass, Physica A: Statistical Mechanics and its Applications, Volume 367, 261-268.

[53] Shen J H, Li Y. (2009). Light Ray Optimization and its parameter analysis, International Joint Conference on Computational Science and Optimization, 918-922.

[54] Shen, J.H. ; Li, J.L. (2012). Light Ray Optimization Algorithm Based on Annealing Strategy, Advanced Materials Research, 461, 435-439.

[55] Shen, J., Li, J., (2010). The principle analysis of Light Ray Optimization Algorithm, IEEE Second International Conference on Computational Intelligence and Natural Computing Proceedings (CINC), 154-157.

[56] Yuan, G. N., Zhang, L. N., Wang, D. D., Harbin T. L., (2013). Parameter Identification of Ship Vertical Motions using Light Ray Optimization Algorithm, Pacific PNT Conference.

[57] Kan, W., Jihong, S., (2012). Multi-Objective Light Ray Optimization, IEEE Fifth International Joint Conference on Computational Sciences and Optimization (CSO), 822-825.

[58] Shen, J., Li, J-F., Li, Y., (2012). Eulerian Light ray optimization algorithm, Journal of Harbin Engineering University, 33, 7.

[59] Kaveh, A., Ghazaan, M. I., Bakhshpoori, T., (2013). An Improved Ray Optimization Algorithm for Design of Truss Structures, Civil Engineering, 57(2), 97–112

[60] Tamura, K., Yasuda, K., (2011), Primary study of spiral dynamics inspired optimization, IEEJ Trans Electrical Electronic Eng, 6 (S1), 98–100.

[61] Tamura, K., Yasuda, K., (2011), Spiral dynamics inspired optimization, J Adv Comput Intell Intell Inform, 15 (8), 1116–1122.

[62] Benasla, L, Belmadani, A., Rahli, M., (2014). Spiral Optimization Algorithm for solving Combined Economic and Emission Dispatch, International Journal of Electrical Power & Energy Systems, Volume 62, 163–174.

[63] Phukan, R., (2013). Reactive Power Management using Firefly and Spiral Optimization under Static and Dynamic Loading Conditions, J Electr Electron Syst 2: 114.

[64] Ouadi, A., Bentarzi, H., Reciou, A., (2013). Optimal multiobjective design of digital filters using spiral optimization technique, 2, 461.

[65] Nasir, A. N. K., (2015). An Improved Spiral Dynamic Optimization Algorithm with Engineering Application, IEEE Transactions on Systems, Man, and Cybernetics: Systems, PP, 99, 1-12.

[66] Eskandar, H., Sadollah, A., Bahreininejad, A., Hamd, M., (2012). Water cycle algorithm – a novel metaheuristic optimization method for solving constrained engineering optimization problems", Computers & Structures, 110-111, 151-166.

[67] Sadollah, A., Eskandar, H., Bahreininejad, A., Kima, J. H., (2015). Water cycle algorithm with evaporation rate for solving constrained and unconstrained optimization problems, Applied Soft Computing, 30, 58-71.

[68] Haddad, O. B., Moravej, M., Loáiciga, H. A., Application of the Water Cycle Algorithm to the Optimal Operation of Reservoir System, J. Irrig. Drain Eng., 141(5), 04014064.

[69] Sadollah, A., Eskandar, H., Bahreininejad, A., Kim, J. H., (2014). Water cycle algorithm for solving multi-objective optimization problems, Soft Computing.

[70] Sadollah, A., Eskandar, H., Kim, J. H., (2015). Water cycle algorithm for solving constrained multi-objective optimization problems, Applied Soft Computing, 27, 279-298.

[71] Sadollah, A., Eskandar, H., Bahreininejad, A., Kim, J. H., (2015). Water cycle, mine blast and improved mine blast algorithms for discrete sizing optimization of truss structures, Computers & Structures, 149, 1-16.

[72] Hieu, T. T., (2011). A water flow algorithm for optimization problems, PhD thesis, National University of Singapore.

[73] Brodić, D., (2011). Advantages of the Extended Water Flow Algorithm for Handwritten Text Segmentation, Lecture Notes in Computer Science Volume 6744, 418-423.

[74] Tran, T. H. and Ng, K. M. (2011). A Water Flow Algorithm for Flexible Flow Shop Scheduling with Intermediate Buffers. Journal of Scheduling, 14 (5), 483-500.