Nature Inspired Computing: An Overview and Some Future Directions

Nazmul Siddique[1] · Hojjat Adeli[2]

Published online: 30 November 2015

Abstract This paper presents an overview of significant advances made in the emerging field of nature-inspired computing (NIC) with a focus on the physics- and biology-based approaches and algorithms. A parallel development in the past two decades has been the emergence of the field of computational intelligence (CI) consisting primarily of the three fields of neural networks, evolutionary computing and fuzzy logic. It is observed that NIC and CI intersect. The authors advocate and foresee more cross-fertilisation of the two emerging fields.

Keywords Nature-inspired computing · Physics-based algorithms · Biology-based algorithms · Meta-heuristic algorithms · Search and optimisation

Inspiration from the Nature

Nature does things in an amazing way. Behind the visible phenomena, there are innumerable invisible causes hidden at times. Philosophers and scientists have been observing these phenomena in the nature for centuries and trying to understand, explain, adapt and replicate the artificial systems. There are innumerable agents and forces within the living and non-living world, most of which are unknown and the underlying complexity is beyond human comprehension as a whole. These agents act in parallel and very often against each other giving form and feature to nature, and regulating the harmony, beauty and vigour of life. This is seen as the dialectics of nature which lies in the concept of the evolution of the natural world. The evolution of complexity in nature follows a distinctive order. There is also information processing in nature performed in a distributed, self-organised and optimal manner without any central control [1]. This whole series of forms, mechanical, physical, chemical, biological and social, is distributed according to complexity from lower to higher. This sequence expresses its mutual dependence and relationship in terms of structure and history. The activities change due to changed circumstances. All these phenomena known or partially known so far are emerging as new fields of science and technology, and computing that study problem-solving techniques inspired by nature as well as attempts to understand the underlying principles and mechanisms of natural, physical, chemical and biological organisms that perform complex tasks in a befitting manner with limited resources and capability.

Science is a dialogue between the scientists and the nature [2] which has evolved over the centuries enriching with new concepts, methods and tools and developed into well-defined disciplines of scientific endeavour. Mankind has been trying to understand the nature ever since by developing new tools and techniques. The field of nature-inspired computing (NIC) is interdisciplinary in nature combining computing science with knowledge from different branches of sciences, e.g. physics, chemistry,

This is an Invited paper.

✉ Nazmul Siddique
nh.siddique@ulster.ac.uk

Hojjat Adeli
Adeli.1@osu.edu

[1] School of Computing and Intelligent Systems, Ulster University, Northland Road, Londonderry BT48 7JL, UK

[2] Departments of Biomedical Engineering, Biomedical Informatics, Civil, Environmental, and Geodetic Engineering, Electrical and Computer Engineering, Neuroscience, and Neurology, The Ohio State University, 470 Hitchcock Hall, 2070 Neil Avenue, Columbus, OH 43210, USA

biology, mathematics and engineering, that allows development of new computational tools such as algorithms, hardware, or wetware for problem-solving, synthesis of patterns, behaviours and organisms [3, 4]. This Keynote paper presents an overview of significant advances made in the emerging field of nature-inspired computing (NIC) with a focus on the physics- and biology-based approaches and algorithms.

Search and Optimisation

All the living and non-living world, the planetary, galactic, stellar system and the heavenly bodies in the universe belong to nature. One common aspect can be observed in nature, be it physical, chemical or biological, that the nature maintains its equilibrium by any means known or unknown to us. A simplified explanation of the state of equilibrium is the idea of optimum seeking in nature. There is optimum seeking in all spheres of life and nature [5–7]. In all optimum seeking, there are goals or objectives to be achieved and constraints to be satisfied within which the optimum has to be found [8–11]. This optimum seeking can be formulated as an optimisation problem [12–15]. That is, it is reduced to finding the best solution measured by a performance index often known as objective function in many areas of computing and engineering which varies from problem to problem [16–19].

Many methods have emerged for the solution of optimisation problems which can be divided into two categories based on the produced solutions [20], namely deterministic and nondeterministic (stochastic) algorithms as shown in Fig. 1. Deterministic algorithms in general follow more rigorous procedures repeating the same path every time and providing the same solution in different runs. Most conventional or classic algorithms are deterministic and based on mathematical programming. Many different mathematical programming methods have been developed in the past few decades. Examples of deterministic algorithms are linear programming (LP), convex programming, integer programming, quadratic programming, dynamic programming, nonlinear programming (NLP), and gradient-based (GB) and gradient-free (GF) methods. These methods usually provide accurate solutions for problems in a continuous space. Most of these methods, however, need the gradient information of the objective function and constraints and a suitable initial point.

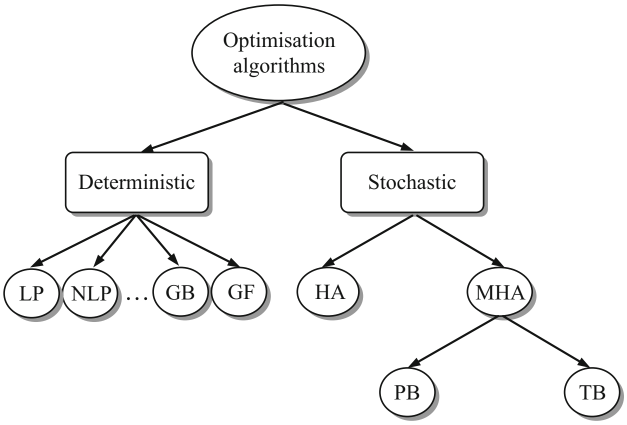

Fig. 1 Classification of optimisation algorithms

On the other hand, nondeterministic or stochastic methods exhibit some randomness and produce different solutions in different runs. The advantage is that these methods explore several regions of the search space at the same time and have the ability to escape from local optima and reach the global optimum. Therefore, these methods are more capable of handling NP-hard problems (i.e. problems that have no known solutions in polynomial time) [21]. There are a variety of derivative-free stochastic optimisation algorithms which are of two types: heuristic algorithms (HA) and meta-heuristic algorithms (MHA) (Fig. 1).

Heuristic means to find or discover by means of trial and error. Alan Turning was one of the first to use heuristic algorithms during the Second World War and called his search methods heuristic search. Glover [22] possibly revived the use of heuristic algorithms in 1970s. The general problem with heuristic algorithms (e.g. scatter search) is that there is no guarantee that optimal solutions are reached though quality solutions are found in a reasonable amount of time. The second generation of the optimisation methods is meta-heuristic proposed to solve more complex problems and very often provides better solutions than heuristic algorithms. The 1980s and 1990s saw a proliferation of meta-heuristic algorithms. The recent trends in meta-heuristic algorithms are stochastic algorithms with certain trade-off of random and local search. Every meta-heuristic method consists of a group of search agents that explore the feasible region based on both randomisation and some specified rules. These methods rely extensively on repeated evaluations of the objective function and use heuristic guidelines for estimating the next search direction. The guidelines used are often simple, and the rules are usually inspired by natural phenomena or laws. Glover and Kochenberger [23] present a review of the field of meta-heuristics up to 2003.

There are different classifications of meta-heuristic algorithms reported in the literature [24, 25]. They can be classified as population based (PB) and neighbourhood or trajectory based (TB) (Fig. 1). Neighbourhood-based meta-heuristics such as simulated annealing [26] and tabu search [27] evaluate only one potential solution at a time and the solution moves through a trajectory in the solution space.

The steps or moves trace a trajectory in the search space, with nonzero probability that this trajectory can reach the global optimum. In the population-based meta-heuristics, a set of potential solutions move towards goals simultaneously. For example, genetic algorithm (GA) [28, 29] and particle swarm optimisation (PSO) [30, 31] are population-based algorithms and use a population of solutions.

Nature-Inspired Computing Paradigm

The nature-inspired computing paradigm is fairly vast. Even though science and engineering have evolved over many hundred years with many clever tools and methods available for their solution, there is still a diverse range of problems to be solved, phenomena to be synthesised and questions to be answered. In general, natural computing approaches should be considered when:

- The problem is complex and nonlinear and involves a large number of variables or potential solutions or has multiple objectives.
- The problem to be solved cannot be suitably modelled using conventional approaches such as complex pattern recognition and classification tasks.
- Finding an optimal solution using traditional approaches is not possible, difficult to obtain or cannot be guaranteed, but a quality measure exists that allows comparison of various solutions.
- The problem lends itself to a diversity of solutions or a diversity of solutions is desirable.

Nature-inspired computing (NIC) refers to a class of meta-heuristic algorithms that imitate or are inspired by some natural phenomena explained by natural sciences discussed earlier. A common feature shared by all nature-inspired meta-heuristic algorithms is that they combine rules and randomness to imitate some natural phenomena. Many nature-inspired computing paradigms have emerged in recent years. They can be grouped into three broad classes: physics-based algorithms (PBA), chemistry-based algorithms (CBA) [32] and biology-based algorithms (BBA) (Fig. 2).

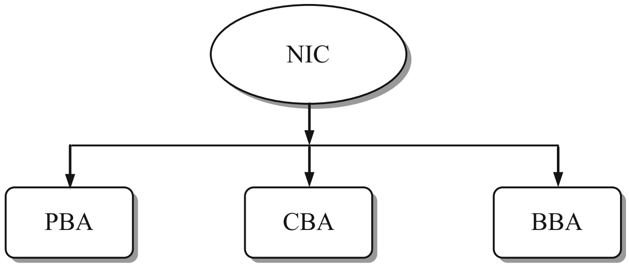

Fig. 2 Broad classification of NIC

Physics-Based Algorithms

Physics-inspired algorithms employ basic principles of physics, for example, Newton's laws of gravitation, laws of motion and Coulomb's force law of electrical charge discussed earlier in the paper. They are all based on deterministic physical principles. These algorithms can be categorised broadly as follows:

(a) Inspired by Newton's laws of motion, e.g. Colliding Bodies Optimisation (CBO),
(b) Inspired by Newton's gravitational force, e.g. Gravitational Search Algorithm (GSA), Central Force Optimisation (CFO), Space Gravitation Optimisation (SGO) and Gravitational Interaction Optimisation (GIO)
(c) Inspired by celestial mechanics and astronomy, e.g. Big Bang–Big Crunch search (BB–BC), Black Hole Search (BHS), Galaxy-based Search Algorithm (GbSA), Artificial Physics-based Optimisation (APO) and Integrated Radiation Search (IRS),
(d) Inspired by electromagnetism, e.g. Electromagnetism-like Optimisation (EMO), Charged System Search (CSS) and Hysteretic Optimisation (HO),
(e) Inspired by optics, e.g. Ray Optimisation (RO),
(f) Inspired by acoustics, e.g. Harmony Search Algorithm (HSA),
(g) Inspired by thermodynamics, e.g. Simulated Annealing (SA),
(h) Inspired by hydrology and hydrodynamics, e.g. Water Drop Algorithm (WDA), River Formation Dynamics Algorithm (RFDA) and Water Cycle Algorithm (WCA).

The earliest of all these algorithms was the Simulated Annealing (SA) algorithm based on the principle of thermo-dynamics [26]. The algorithm simulates the cooling process by gradually lowering the temperature of the system until it converges to a steady state. The idea to use simulated annealing to search for feasible solutions and converge to an optimal solution was very stimulating and led researchers to explore other areas of physics.

An idea from the field of sound and acoustics led to the development of HSA inspired by a phenomenon commonly observed in music. The concept behind the HSA is to find a perfect state of harmony determined by aesthetic estimation [33]. A review of harmony search algorithms and its variants is provided by Siddique and Adeli [34]. Hybrid harmony search algorithms are presented by Siddique and Adeli [35]. Applications of HSA are reviewed in Siddique and Adeli [36].

Zaránd et al. [37] proposed a method of optimisation inspired by demagnetisation, called hysteretic optimisation (HO). This is a process similar to simulated annealing where the material achieves a stable state by slowly decreasing the temperature. That is, finding the ground states of magnetic samples is similar to finding the optimal point in the search process. Based on the principles of electromagnetism, Birbil and Fang [38] introduced the electromagnetism-based optimisation. The EM-based algorithm imitates the attraction–repulsion mechanism of the electromagnetism theory in order to solve unconstrained or bound constrained global optimisation problems. It is called electromagnetism-like optimisation (EMO) algorithm. A solution in EMO algorithm is seen as a charged particle in the search space and its charge relates to the objective function value.

Motivated by natural physical forces, Spears et al. [39] introduced the Artificial Physics Optimisation (APO) where particles are seen as solutions sampled from the feasible region of the problem space. Particles move towards higher fitness regions and cluster to optimal region over time. Heavier mass represents higher fitness value and attracts other masses of lower fitness values. The individual with the best fitness attracts all other individuals with lower fitness values. The individuals with lower fitness values repel each other. That means the individual with best fitness has the biggest mass and move with lower velocity than others. Thus, the attractive–repulsive rule can be treated as the search strategy in the optimisation algorithm which ultimately leads the population to search the better fitness region of the problem. In the initial state, individuals are randomly generated within the feasible region. In APO, mass is defined as the fitness function for the optimisation problem in question. A suitable definition of mass of the individuals is necessary.

Central Force Optimisation (CFO) uses a population of probes that are distributed across a search space [40]. The basic concept of the CFO is the search for the biggest mass that has the strongest force to attract all other masses distributed within a decision space towards it considered as the global optimum of the problem at hand. A review of articles on CFO and its applications to various problems is presented in a recent article by Siddique and Adeli [41].

Gravitational Search Algorithm (GSA) is a population-based search algorithm inspired by the law of gravity and mass interaction [42]. The algorithm considers agents as objects consisting of different masses. The entire agents move due to the gravitational attraction force acting between them, and the progress of the algorithm directs the movements of all agents globally towards the agents with heavier masses [42]. Gravitational Interactions Optimisation (GIO) is inspired by Newton's law [43]. It has some similarities with GSA and was introduced around the same time independently of GSA. The gravitational constant G in GSA decreases linearly with time, whereas GIO uses a hypothetical gravitational constant G as constant. GSA uses a set of best individuals to reduce computation time, while GIO allows all masses to interact with each other.

Based on the simple principle of continuous collision between bodies, Kaveh and Mahdavi [44] proposed the Colliding Bodies Optimisation (CBO). Hsiao et al. [45] proposed an optimal searching approach, called Space Gravitational Optimisation (SGO) using the notion of space gravitational curvature inspired by the concept of Einstein equivalence principle. SGO is an embryonic form of CFO [46]. Based on the notion of Big Bang and shrinking phenomenon of Big Crunch, Erol and Eksin [47] proposed Big Bang and Big Crunch (BB–BC) algorithm. In the Big Bang phase, a population of masses is generated with respect to centre of mass. In the Big Crunch phase, all masses collapse into one centre of mass. Thus, the Big Bang phase explores the solution space, while Big Crunch phase performs necessary exploitation as well as convergence. Chuang and Jiang [48] proposed Integrated Radiation Optimisation (IRO) inspired by the gravitational radiation in the curvature of space–time. Hosseini [49] proposed Galaxy-based Search Algorithm (GbSA) inspired by the spiral arm of spiral galaxies to search its surrounding. GbSA uses a spiral-like movement in each dimension of the search space with the help of chaotic steps and constant rotation around the initial solution. The spiral optimisation (SpO) is a multipoint search for continuous optimisation problems. The SpO model is composed of plural logarithmic spiral models and their common centre [50].

Inspired by the phenomenon of the black hole, Hatamlou [51] proposed the Black Hole (BH) algorithm where candidate solutions are considered as stars and the solution is selected to be black hole. At each iteration, the black hole starts attracting other stars around it. If a star gets too close to the black hole, it will be swallowed and a new star (candidate solution) is randomly generated and placed in the search space to start a new search.

The basic idea of Snell's law is utilised in Ray Optimisation (RO) proposed by Kaveh and Khayatazad [52] where a solution consisting of a vector of variables is simulated by a ray of light passing through space treated as media with different refractive indices. Based on the principles of hydrodynamics and water cycles, Intelligent Water Drop (IWD) was proposed by Shah-Hosseini [53]. Considering the natural phenomenon of river formations through land erosion and sediment deposits, Rabanal et al. [54] proposed River Formation Dynamics (RFD). Eskandar et al. [55] proposed Water Cycle Algorithm (WCA) based on the principle of water cycle that forms streams and rivers where all rivers flow to the sea which is the ultimate destination and optimal solution in terms of optimisation.

Biology-Based Algorithms

Biology-based algorithms can be classified into three groups: Evolutionary Algorithms (EA), Bio-inspired Algorithms (BIA) and Swarm Intelligence-based Algorithms (SIA) (Fig. 3).

The fundamental idea of evolutionary algorithms is based on Darwin's theory of evolution, which gained momentum in the late 1950s nearly a century after publication of the book 'Origin of Species'. Fraser [56] first conducted a simulation of genetic systems representing organisms by binary strings. Box [57] proposed an evolutionary operation to optimising industrial production. Friedberg [58] proposed an approach to evolve computer programs. The fundamental works of Lowrence Fogel [59] in evolutionary programming, John Holland [60] in genetic algorithms, Ingo Rechenberg [61] and Hans-Paul Schwefel [62] in evolution strategies had great influences on the development of evolutionary algorithms and computation as a general concept for problem-solving and as a powerful tool for optimisation. Since the development years of 1960s, the field evolved into three main branches [63]: evolution strategies [64], evolutionary programming and genetic algorithms. In the 1990s there was another set of development in the evolutionary algorithms such as Koza [65] developed genetic programming, Reynolds [66] developed cultural algorithms and Storn and Price [67] developed differential evolution. Evolutionary algorithms have now found wide spread applications in almost all branches of science and engineering [68–70]. Different variants of EAs such as Evolutionary Programming (EP) [71], Evolution Strategies (ES) [72, 73], Genetic Algorithm (GA) [74–76], Genetic Programming (GP), Differential Evolution (DE) and Cultural Algorithm (CA) are discussed in the book by Siddique and Adeli [77].

The BIA are based on the notion of commonly observed phenomenon in some animal species and movement of organisms. Flocks of birds, herds of quadrupeds and schools of fish are often shown as fascinating examples of self-organised coordination [1, 78]. Particle Swarm Optimisation (PSO) simulates social behaviour of swarms such as birds flocking and fish schooling in nature [79–81]. Particles make use of the best positions encountered and the best position of their neighbours to position themselves towards an optimum solution [82]. There are now as many as about 20 different variants of PSO [83–86].

Bird Flocking (BF) is seen as feature of coherent manoeuvring of a group of individuals due to advantages for protecting and defending from predators, searching for food, and social and mating activities [87]. Natural flocks maintain two balanced behaviours: a desire to stay close to the flock and a desire to avoid collisions within the flock [88]. Reynolds [89] developed a model to mimic the flocking behaviour of birds using three simple rules: collision avoidance with flockmates, velocity matching with nearby flockmates and flock centring to stay close to the flock [90, 91]. Fish School (FS) shows very interesting features in their behaviour. About half the fish species are known to form fish schools at some stage in their lives. FS is observed as self-organised systems consisting of individual autonomous agents [92, 93] and come in many different shapes and sizes [87, 94, 95].

MacArthur and Wilson [96] developed mathematical models of biogeography that describe how species migrate from one island to another, how new species arise and how species become extinct. Since 1960s biogeography has

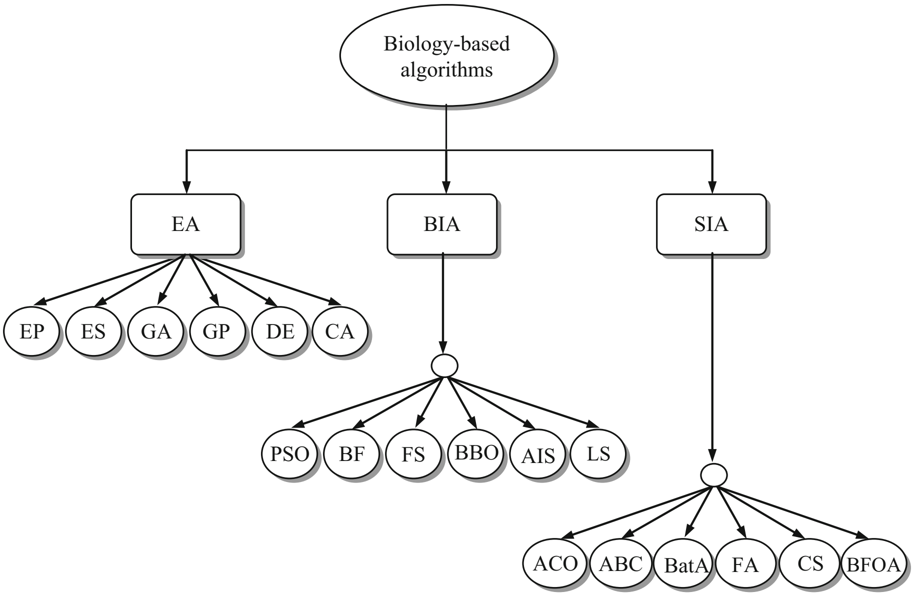

Fig. 3 Classification of biology-based algorithms

become a major area of research that studies the geographical distribution of biological species. Based on the concept of biogeography, Simon [97] proposed Biogeography Based Optimisation (BBO). Based on the principles of biological immune systems, models of Artificial Immune Systems (AIS) were proposed by Farmer et al. [98] in the 1980s that stipulated the interaction between antibodies mathematically. In 1968, Lindenmayer [99] introduced formalism for simulating the development of multi-cellular organisms, initially known as Lindenmayer systems and subsequently named L-systems which attracted the interest of theoretical computer scientists. Aono and Kunii [100] and Smith [101] used L-systems to create realistic-looking images of trees and plants. There are other bio-inspired search and optimisation algorithms reported in the literature which haven't attract much attention in the research community such as atmosphere clouds model [102], dolphin echolocation, Japanese tree frogs calling, Egyptian vulture, flower pollination algorithm, great salmon run, invasive weed optimisation, paddy field algorithm, roach infestation algorithm and shuffle frog leaping algorithm.

The SIA are based on the idea of collective behaviours of insects living in colonies such as ants, bees, wasps and termites. Researchers are interested in the new way of achieving a form of collective intelligence called swarm intelligence. SIAs are also advanced as a computational intelligence technique based around the study of collective behaviour in decentralised and self-organised systems. The inspiring source of Ant Colony Optimisation (ACO) is based on the foraging behaviour of real ant colonies [103, 104]. While moving, ants leave a chemical pheromone trail on the ground. When choosing their way, they tend to choose paths marked by strong pheromone concentrations. The pheromone trails will guide other ants to the food source. It has been shown that the indirect communication between the ants via pheromone trails enables them to find the shortest paths between their nest and food sources.

Honey bees search for food sources and collect by foraging in promising flower patches. The simple mechanism of the honey bees inspired researchers to develop a new search algorithm, called Bee Algorithm [105, 106]. Similarly, Artificial Bee Colony (ABC) algorithm was proposed by Karaboga [107] and virtual bee algorithm was proposed by Yang [108]. Bat Algorithm (BatA) is based on the echolocation behaviour of bats. The capability of microbats is fascinating as they use a type of sonar, called echolocation, to detect prey, avoid obstacles and locate their roosting crevices in the dark. Yang [109] simulated echolocation behaviour of bats. Quite a number of cuckoo species engage the obligate brood parasitism by laying their eggs in the nests of host birds of different species. Yang and Deb [110] describe the Cuckoo Search (CS) algorithm based on the breeding behaviour of certain cuckoo species. The flashing of fireflies in the summer sky in the tropical regions has been attracting the naturalists and researchers for many years. The rhythm, the rate and the duration of flashing form part of the signalling system that brings two fireflies together. Based on some idealised rules, Yang [111] proposed the Firefly Algorithm (FA).

Individual and groups of bacteria forage for nutrients, e.g. chemotactic (foraging) behaviour of *E. coli* bacteria. Based on this concept, Passino [112] proposed Bacterial Foraging Optimisation Algorithm (BFOA). There are many swarm intelligence-based search and optimisation algorithms reported in the literature which haven't attract much attention in the research community such as wolf search, cat swarm optimisation, fish swarm optimisation, eagle strategy, krill herd, monkey search and weightless swarm algorithms.

Conclusion

It is obvious from this review that the field of nature-inspired computing is large and expanding. This invited paper provided a brief summary of significant advances made in this exciting area of research with a focus on the physics- and biology-based approaches and algorithms.

A parallel development has been the emergence of the field of computational intelligence (CI) mainly consisting of neural networks [113–120], evolutionary computing [121] and fuzzy logic [122–125] in the past twenty years starting with the seminal book of Adeli and Hung [126] which demonstrated how a multi-paradigm approach and integration of the three CI computing paradigms can lead to more effective solutions of complicated and intractable pattern recognition and learning problems. It is observed that NIC and CI intersect. Some researchers have argued that swarm intelligence provides computational intelligence. The authors advocate and foresee more cross-fertilisation of the two emerging fields. Evolving neural networks is an example of such cross-fertilisation of two domains [127, 128].

References

1. Lopez-Rubio E, Palomo EJ, Dominguez E. Bregman divergences for growing hierarchical self-organizing networks. Int J Neural Syst. 2014;24(4):1450016.

2. Prigogine I. The end of certainty. New York: The Free Press; 1996.
3. De Castro LN. Fundamentals of natural computing: an overview. Phys Life Rev. 2007;4:1–36.
4. Kari L, Rozenberg G. Many facets of natural computing. Commun ACM. 2008;51(10):72–83.
5. Arango C, Cortés P, Onieva L, Escudero A. Simulation–optimisation models for the dynamic berth allocation problem. Comput Aided Civil Infrastruct Eng. 2013;28(10):769–79.
6. Chow JYJ. Activity-based travel scenario analysis with routing problem reoptimization. Comput Aided Civil Infrastruct Eng. 2014;29(2):91–106.
7. Adeli H, Park HS. Neurocomputing for design automation. Boca Raton: CRC Press; 1998.
8. Chen X, Zhang L, He X, Xiong C, Li Z. Surrogate-based optimization of expensive-to-evaluate objective for optimal highway toll charging in a large-scale transportation network. Comput Aided Civil Infrastruct Eng. 2014;29(5):359–81.
9. Jia L, Wang Y, Fan L. Multiobjective bilevel optimization for production-distribution planning problems using hybrid genetic algorithm. Integr Comput Aided Eng. 2014;21(1):77–90.
10. Faturechi R, Miller-Hooks E. A mathematical framework for quantifying and optimizing protective actions for civil infrastructure systems. Comput Aided Civil Infrastruct Eng. 2014;29(8):572–89.
11. Aldwaik M, Adeli H. Advances in optimization of highrise building structures. Struct Multidiscip Optim. 2014;50(6):899–919.
12. Adeli H, Kamal O. Efficient optimization of space trusses. Comput Struct. 1986;24(3):501–11.
13. Smith R, Ferrebee E, Ouyang Y, Roesler J. Optimal staging area locations and material recycling strategies for sustainable highway reconstruction. Comput Aided Civil Infrastruct Eng. 2014;29(8):559–71.
14. Peng F, Ouyang Y. Optimal clustering of railroad track maintenance jobs. Comput Aided Civil Infrastruct Eng. 2014;29(4):235–47.
15. Luo D, Ibrahim Z, Xu B, Ismail Z. Optimization the geometries of biconical tapered fiber sensors for monitoring the early-age curing temperatures of concrete specimens. Comput Aided Civil Infrastruct Eng. 2013;28(7):531–41.
16. Adeli H. Advances in design optimization. London: Chapman and Hall; 1994.
17. Adeli H, Sarma K. Cost optimization of structures—fuzzy logic, genetic algorithms, and parallel computing. West Sussex: Wiley; 2006.
18. Gao H, Zhang X. A Markov-based road maintenance optimization model considering user costs. Comput Aided Civil Infrastruct Eng. 2013;28(6):451–64.
19. Zhang G, Wang Y. Optimizing coordinated ramp metering—a preemptive hierarchical control approach. Comput Aided Civil Infrastruct Eng. 2013;28(1):22–37.
20. Yang X-S. Engineering optimisation: an introduction with metaheuristic application. New York: Wiley; 2010.
21. Lin M-H, Tsai J-F, Yu C-S. A review of deterministic optimization methods in engineering and management. Math Probl Eng Optim Theory Methods Appl Eng. edt, 2012; vol 2012, article ID 756023.
22. Glover F. Heuristics for integer programming using surrogate constraints. Decis Sci. 1977;8(1):156–66.
23. Glover F, Kochenberger GA. Handbook of metaheuristic. New York: Kluwer; 2003.
24. Fister I Jr, Yang X-S, Fister I, Brest J, Fister D. A brief review of nature-inspired algorithms for optimisation. Elektroteh Vestn. 2013;80(3):1–7.
25. Manjarres D, Landa-Torres I, Gil-Lopez S, Del Ser J, Bilbao MN, Salcedo-Sanz S, Geem ZW. A survey on applications of the harmony search algorithm. Eng Appl Artif Intell. 2013;26(8):1818–31.
26. Kirkpatrick S, Gelatto CD, Vecchi MP. Optimization by simulated annealing. Science. 1983;220:671–80.
27. Glover F. Tabu search—part I. ORSA J Comput. 1989;1(3):190–206.
28. Hejazi F, Toloue I, Noorzaei J, Jaafar MS. Optimization of earthquake energy dissipation system by genetic algorithm. Comput Aided Civil Infrastruct Eng. 2013;28(10):796–810.
29. Kociecki M, Adeli H. Shape optimization of free-form steel space-frame roof structures with complex geometries using evolutionary computing. Eng Appl Artif Intell. 2015;38:168–82.
30. Iacca G, Caraffini F, Neri F. Multi-strategy coevolving aging particle optimization. Int J Neural Syst. 2014;24(1):1450008.
31. Shafahi Y, Bagherian M. A customized particle swarm method to solve highway alignment optimization problem. Comput Aided Civil Infrastruct Eng. 2013;28(1):52–67.
32. Szeto WY, Wang Y, Wong SC. The chemical reaction optimization approach to solving the environmentally sustainable network design problem. Comput Aided Civil Infrastruct Eng. 2014;29(2):140–58.
33. Geem ZW, Kim JH, Loganathan GV. A new heuristic optimization algorithm: harmony search. Simulation. 2001;76(2):60–8.
34. Siddique N, Adeli H. Harmony search algorithm and its variants. Int J Pattern Recognit Artif Intell. 2015;29(8):1539001.
35. Siddique N, Adeli H. Hybrid harmony search algorithms. Int J Artif Intell Tools. 2015;24(6):1–16.
36. Siddique N, Adeli H. Applications of harmony search algorithms in engineering. Int J Artif Intell Tools. 2015;24(6):1–15.
37. Zaránd G, Pázmándi F, Pál KF, Zimányi GT. Hysteretic optimization. Phys Rev Lett. 2002;89(15):1502011–4.
38. Birbil I, Fang SC. An electro-magnetism-like mechanism for global optimization. J Glob Optim. 2003;25:263–82.
39. Spears DF, Spears WM. Analysis of a phase transition in a physics-based multiagent system. Lect Notes Comput Sci. 2003;2699:193–207.
40. Formato RA. Central force optimization: a new metaheuristic with applications in applied electromagnetics. PIER. 2007;77(1):425–91.
41. Siddique N, Adeli H. Central force metaheuristic optimization. Sci Iran Trans A Civil Eng. 2015;22(6):2015.
42. Rashedi E, Nezamabadi-pour H, Saryazdi S. GSA: a gravitational search algorithm. Inf Sci. 2009;179(13):2232–48.
43. Flores J, Lopez R, Barrera J. Gravitational interactions optimization. Learning and intelligent optimization. Berlin: Springer; 2011. p. 226–37.
44. Kaveh A, Mahdavi VR. Colliding bodies optimization: a novel meta-heuristic method. Comput Struct. 2014;139:18–27.
45. Hsiao YT, Chuang CL, Jiang JA, Chien CC. A novel optimization algorithm: space gravitational optimization. In: Proceedings of 2005 IEEE international conference on systems, man and cybernetics, Oct 2005, vol. 3, p. 2323–8.
46. Kenyon IR. General relativity. Oxford: Oxford University Press; 1990.
47. Erol OK, Eksin I. A new optimization method: big bang–big crunch. Adv Eng Softw. 2006;37(2):106–11.
48. Chuang C, Jiang J. Integrated radiation optimization: inspired by the gravitational radiation in the curvature of space–time. IEEE Congr Evolut Comput (CEC). 2007;25–28:3157–64.
49. Hosseini HS. Principal component analysis by galaxy-based search algorithm: a novel meta-heuristic for continuous optimisation. Int J Comput Sci Eng. 2011;6(1–2):132–40.

50. Tamura K, Yasuda K. Spiral dynamics inspired optimisation. J Adv Comput Intell Intell Inform. 2011;15(8):1116–22.
51. Hatamlou A. Black hole: a new heuristic optimization approach for data clustering. Inf Sci. 2013;222:175–84.
52. Kaveh A, Khayatazad M. A new meta-heuristic method: ray optimization. Comput Struct. 2012;112–113:283–94.
53. Shah-Hosseini H. Intelligent water drops algorithm—a new optimisation method for solving the multiple knapsack problem. Int J Intell Comput Cybern. 2008;1(2):193–212.
54. Rabanal P, Rodríguez I, Rubio F. Using river formation dynamics to design heuristic algorithms. In: Unconventional computation, UC'07, LNCS 4618, Springer, 2007, p. 163–77.
55. Eskandar H, Sadollah A, Bahreininejad A, Hamdi M. Water cycle algorithm—a novel metaheuristic optimization method for solving constrained engineering optimization problems. Comput Struct. 2012;110–111:151–66.
56. Fraser AS. Simulation of genetic systems by automatic digital computers, I. Introduction. Aust J Biol Sci. 1957;10:484–91.
57. Box GEP. Evolutionary operation: a method for increasing industrial productivity. Appl Stat. 1957;6(2):81–101.
58. Friedberg RM. A learning machine: part I. IBM J Res Dev. 1958;2(1):2–13.
59. Fogel LJ. Autonomous automata. Ind Res. 1962;4:14–9.
60. Holland J. Outline for a logical theory of adaptive systems. J ACM. 1962;3:297–314.
61. Rechenberg I. Cybernetic solution path of an experimental problem, royal aircraft establishment. Library translation no. 1122, Farnborough, Hants, UK; 1965.
62. Schwefel H-P. Projekt MHD-Strausstrhlrohr: Experimentelle Optimierung einer Zweiphasenduese, Teil I, Technischer Bericht 11.034/68, 35, AEG Forschungsinstitute, Berlin, Germany; 1968.
63. De Jong KA. Evolutionary computation: a unified approach. Cambridge: The MIT Press; 2006.
64. Reyes O, Morell C, Ventura S. Evolutionary feature weighting to improve the performance of multi-label lazy algorithms. Integr Comput Aided Eng. 2014;21(4):339–54.
65. Koza John R. Genetic programming: on the programming of computers by means of natural selection. Cambridge: The MIT Press; 1992.
66. Reynolds RG. An overview of cultural algorithms: advances in evolutionary computation. New York: McGraw Hill Press; 1999.
67. Storn R, Price K. Differential evolution—a simple and efficient heuristic for global optimisation over continuous space. J Glob Optim. 1997;11(4):431–59.
68. Molina-García M, Calle-Sánchez J, González-Merino C, Fernández-Durán A, Alonso JI. Design of in-building wireless networks deployments using evolutionary algorithms. Integr Comput Aided Eng. 2014;21(4):367–85.
69. Lin DY, Ku YH. Using genetic algorithms to optimize stopping patterns for passenger rail transportation. Comput Aided Civil Infrastruct Eng. 2014;29(4):264–78.
70. Adeli H, Kumar S. Distributed computer-aided engineering for analysis, design, and visualization. Boca Raton: CRC Press; 1999.
71. Badawy R, Yassine A, Heßler A, Hirsch B, Albayrak S. A novel multi-agent system utilizing quantum-inspired evolution for demand side management in the future smart grid. Integr Comput Aided Eng. 2013;20(2):127–41.
72. Campomanes-Álvareza BR, Cordón O, Damasa S. Evolutionary multi-objective optimization for mesh simplification of 3D open models. Integr Comput Aided Eng. 2013;20(4):375–90.
73. Joly MM, Verstraete T, Paniagua G. Integrated multifidelity, multidisciplinary evolutionary design optimization of counterrotating compressors. Integr Comput Aided Eng. 2014;21(3):249–61.
74. Kim H, Adeli H. Discrete cost optimization of composite floors using a floating point genetic algorithm. Eng Optim. 2001;33(4):485–501.
75. Kociecki M, Adeli H. Two-phase genetic algorithm for size optimization of free-form steel space-frame roof structures. J Constr Steel Res. 2013;90:283–96.
76. Kociecki M, Adeli H. Two-phase genetic algorithm for topology optimization of free-form steel space-frame roof structures with complex curvatures. Eng Appl Artif Intell. 2014;32:218–27.
77. Siddique N, Adeli H. Computational intelligence: synergies of fuzzy logic, neural networks and evolutionary computing. Chichester: Wiley; 2013.
78. Camazine S, Deneubourg J-L, Franks NR, Sneyd J, Theraulaz G, Bonabeau E. Self-organization in biological systems. New Jersey: Princeton University Press; 2001.
79. Amini F, Khanmohamadi Hazaveh N, Abdolahi Rad A. Wavelet PSO-based LQR algorithm for optimal structural control using active tuned mass dampers. Comput Aided Civil Infrastruct Eng. 2013;28(7):542–57.
80. Kennedy J, Eberhart R. Swarm intelligence. San Francisco: Morgan Kaufmann Publishers Inc; 2001.
81. Wu JW, Tseng JCR, Tsai WN. A hybrid linear text segmentation algorithm using hierarchical agglomerative clustering and discrete particle swarm optimization. Integr Comput Aided Eng. 2014;21(1):35–46.
82. Zeng Z, Xu J, Wu S, Shen M. Antithetic method-based particle swarm optimization for a queuing network problem with fuzzy data in concrete transportation systems. Comput Aided Civil Infrastruct Eng. 2014;29(10):771–800.
83. Bergh FVD, Engelbrecht AP. A study of particle swarm optimization particle trajectories. Inf Sci. 2006;176:937–71.
84. Jiang M, Luo YP, Yang SY. Stochastic convergence analysis and parameter selection of the standard particle swarm optimization algorithm. Inf Process Lett. 2007;102:8–16.
85. Tsai H, Lin Y. Modification of the fish swarm algorithm with particle swarm optimization formulation and communication behavior. Appl Soft Comput. 2011;11:5367–74.
86. Montalvo I, Izquierdo J, Herrera M, Pérez-García R. Water distribution system computer-aided design by agent swarm optimization. Comput Aided Civil Infrastruct Eng. 2014;29(6):433–48.
87. Shaw E. The schooling of fishes. Sci Am. 1962;206:128–38.
88. Shaw E. Fish in schools. Nat History. 1975;84(8):40–6.
89. Reynolds C. Flocks, herds, and schools: a distributed behavioural model. Comput Graph. 1987;21(4):25–34.
90. Momen S, Amavasai BP, Siddique NH. Mixed species flocking for heterogenous robotic swarms. In: The international conference on computer as a tool (EUROCON 2007), Piscataway, NJ. IEEE Press; 2007, p. 2329–36.
91. Turgut AE, Çelikkanat H, Gökçe F, Sahin E. Self-organized flocking in mobile robot swarms. Swarm Intell. 2008;2:97–120.
92. Sun Q, Wu S. A configurable agent-based crowd model with generic behaviour effect representation mechanism. Comput Aided Civil Infrastruct Eng. 2014;29(7):531–45.
93. Pinto T, Praça I, Vale Z, Morais H, Sousa TM. Strategic bidding in electricity markets: an agent-based simulator with game theory for scenario analysis. Integr Comput Aided Eng. 2013;20(4):335–46.
94. Parrish JK, Viscido SV, Grunbaum D. Self-organized fish schools: an examination of emergent properties. Biol Bull. 2002;202:296–305.
95. Mackinson S. Variation in structure and distribution of prespawning Pacific herring shoals in two regions of British Columbia. J Fish Biol. 1999;55:972–89.

96. MacArthur R, Wilson E. Theory of biogeography. Princeton: Princeton University Press; 1967.
97. Simon D. Biogeography-based optimization. IEEE Trans Evolut Comput. 2008;12(6):702–13.
98. Farmer JD, Packard N, Perelson A. The immune system, adaptation and machine learning. Phys D. 1986;2:187–204.
99. Lindenmayer A. Mathematical models for cellular interactions in development, parts I and II. J Theor Biol. 1968;18:280–315.
100. Aono M, Kunii TL. Botanical tree image generation. IEEE Comput Graph Appl. 1984;4(5):10–34.
101. Smith AR. Plants, fractals, and formal languages. In: Proceedings of SIG-GRAPH'84 in computer graphics, ACM SIG-GRAPH, Minneapolis, Minnesota, July 22–27, 1984, p. 1–10.
102. Chen J, Wu T. A Computational intelligence optimization algorithm: cloud drops algorithm. Integr Comput Aided Eng. 2014;21(2):177–88.
103. Dorigo M, Birattari M, Stutzle T. Ant colony optimization. IEEE Comput Intell Mag. 2006;1(4):28–39.
104. Forcael E, González V, Orozco F, Vargas S, Moscoso P, Pantoja A. Ant colony optimization model for tsunamis evacuation routes. Comput Aided Civil Infrastruct Eng. 2014;29(10): 723–37.
105. Nakrani S, Tovey C. On honey bees and dynamic server allocation in internet hosting centers. Adapt Behav. 2004;12: 223–40.
106. Pham DT, Ghanbarzadeh A, Koc E, Otri S, Rahim S, Zaidi M. The bees algorithm, technical note. Manufacturing Engineering Centre, Cardiff University, UK; 2005.
107. Karaboga D. An idea based on honey bee swarm for numerical optimisation, technical report TR06. Erciyes University, Turkey; 2005.
108. Yang XS. Engineering optimisation via nature-inspired virtual bee algorithms, IWINAC 2005. Lect Notes Comput Sci. 2005;3562:317–23.
109. Yang X-S. A new metaheuristic bat-inspired algorithm. In: Cruz C, Gonzalez J, Krasnogor N, Terraza G, editors. Nature inspired cooperative strategies for optimization (NISCO 2010), studies in computational intelligence, vol. 284. Berlin: Springer; 2010. p. 65–74.
110. Yang XS, Deb S. Engineering optimisation by cuckoo search. Int J Math Modell Numer Optim. 2010;1(4):330–43.
111. Yang X-S. Firefly algorithms for multimodal optimization. In: Stochastic algorithms: foundations and applications, SAGA 2009. Lect Notes Comput Sci. 2009;5792:169–78.
112. Passino KM. Biomimicry of bacterial foraging for distributed optimization and control. IEEE Control Syst Mag. 2002;22(3):52–67.
113. Adeli H, Park HS. A neural dynamics model for structural optimization—theory. Comput Struct. 1995;57(3):383–90.
114. Adeli H, Park HS. Optimization of space structures by neural dynamics. Neural Netw. 1995;8(5):769–81.
115. Adeli H, Karim A. Scheduling/cost optimization and neural dynamics model for construction. J Constr Manag Eng ASCE. 1997;123(4):450–8.
116. Adeli H, Kim H. Cost optimization of composite floors using the neural dynamics model. Commun Numer Methods Eng. 2001;17:771–87.
117. Huo J, Gao Y, Yang W, Yin H. Multi-instance dictionary learning for detecting abnormal event detection in surveillance videos. Int J Neural Syst. 2014;24(3):1430010.
118. Park HS, Adeli H. Distributed neural dynamics algorithms for optimization of large steel structures. J Struct Eng ASCE. 1997;123:880–8.
119. Wang Z, Guo L, Adjouadi M. A generalized leaky integrate-and-fire neuron model with fast implementation method. Int J Neural Syst. 2014;24(5):1440004.
120. Yang YB, Li YN, Gao Y, Yin HJ, Tang Y. Structurally enhanced incremental neural learning for image classification with subgraph extraction. Int J Neural Syst. 2014;24(7): 1450024.
121. Menendez H, Barrero DF, Camacho D. A genetic graph-based approach to the partitional clustering. Int J Neural Syst. 2014;24(3):1430008.
122. Ahmadlou M, Adeli H. Fuzzy synchronization likelihood with application to attention-deficit/hyperactivity disorder. Clin EEG Neurosci. 2011;42(1):6–13.
123. Kodogiannis VS, Amina M, Petrounias I. A clustering-based fuzzy-wavelet neural network model for short-term load forecasting. Int J Neural Syst. 2013;23(5):1350024.
124. Boutalis Y, Christodoulou M, Theodoridis D. Indirect adaptive control of nonlinear systems based on bilinear neuro-fuzzy approximation. Int J Neural Syst. 2013;23(5):1350022.
125. Forero Mendoza L, Vellasco M, Figueiredo K. Intelligent multiagent coordination based on reinforcement hierarchical neuro-fuzzy models. Int J Neural Syst. 2014;24(8):1450031.
126. Adeli H, Hung SL. Machine learning—neural networks, genetic algorithms, and fuzzy sets. New York: Wiley; 1995.
127. Alexandridis A. Evolving RBF neural networks for adaptive soft-sensor design. Int J Neural Syst. 2013;23(6):1350029.
128. Cabessa J, Siegelmann HT. The super-turing computational power of evolving recurrent neural networks. Int J Neural Syst. 2014;24(8):1450029.