A Brief Introduction to Nature-inspired Computing, Optimization and Applications

Anupam Biswas[a], Alberto Tonda[b,c], Ripon Patgiri[a]

[a] *Department of Computer Science and Engineering, National Institute of Technology Silchar, Cachar, Silchar, 788010, Assam, Bharat (India)*
[b] *UMR 518 MIA-PS, INRAE, Université Paris-Saclay, 22, place de l'Agronomie, Palaiseau, 91120, France*
[c] *Institut des Systèmes Complexes de Paris Île-de-France (ISC-PIF) - UAR 3611 CNRS, Paris, France, 113, rue Nationale, Paris, 75013, France*

Abstract

This chapter provides preliminary details about nature-inspired computing and optimization techniques. It starts by describing different components of optimization problems and their taxonomy. Nature-inspired computing techniques, created to solve such optimization problems, are then discussed. The chapter briefs the application prospects of nature-inspired optimization techniques in a few representative emerging domains that are covered in the book. Applications presented in this book cover five major domains, which include Controller and Power Systems, Ecological and Economic Systems, Information and Computational Systems, Communication and Networking Systems, and Deep Learning and Neural Networking Systems. The book primarily focuses on the practical challenges that are faced while applying nature-inspired algorithms to different problems in these domains, which include the feasibility of the problem, control parameters and constraints, representation of the solution space, and design of the objective function. Lastly, the chapter concluded with highlighting common challenges, criticisms and future perspectives.

Keywords: Nature-inspired Optimization, Meta-heuristics, Swarm Intelligence, Evolutionary Computation

1. Optimization Problems

Technological advancements have led ever-increasing complexities in the diverse problems associated with different application domains. Optimization problems [1, 2, 3, 4] are one of the widely encountered problems across different domains. Optimization algorithms are computational techniques used to find the best solution to a problem, picked from a set of possible candidate solutions. Researchers and practitioners across the globe are constantly trying to develop models, processes, and application modules that are cost-effective, more reliable and efficient. Despite differences in the nature of the problem depending on applications, the optimization problems have the following common components:

- Decision variables (X): The set of unknown parameters x_i of a problem, which are taken as inputs to the objective functions. Depending on the permissible values of x_i there will be multiple sets of values for the decision variable, which are referred as candidate solutions. All sets of candidate solutions constitutes the search space of the problem, also referred as solution space.

- Objective functions $f(X)$: It determines the quality of a candidate solution i.e. a set of values of decision variables. The objective functions tell whether a set of values of decision variables is good or bad for the problem. Defining appropriate objective function is crucial for an optimization problem.

- Constraints $C(X)$: These are the criterion to define permissible values of decision variables. Normally, constraints are defined as inequalities on decision variables or as a function too.

An optimization problem can be formulated for any application with relevant decision variables, objective functions, and constraints. Objective function can be defined either as a maximization problem [5] (i.e. maximum value of the objective function will imply best solution) or minimization problem [6] (i.e. minimum value of the objective function will imply best solution). A generic minimization problem is given as follows:

$$\text{Minimize: } f_i(x), \text{ for } i = 1, 2, 3..., I \tag{1}$$

$$\text{Subject to: } g_j(x) =\leq a_j, \text{ for } i = 1, 2, 3..., J \tag{2}$$

$$h_k(x) = b_k, \text{ for } i = 1, 2, 3..., K \tag{3}$$

where f_i shows the i^{th} objective function, g_j indicates the j^{th} inequality constraint, h_k is the k^{th} equality constraint, a_j is a constraint for j^{th} in equality constraint, b_k is a constant for k^{th} equality constraint. I is the number of objectives, J is the number of inequality constraints, and K is the number of equality constraints, which are dependent on the application.

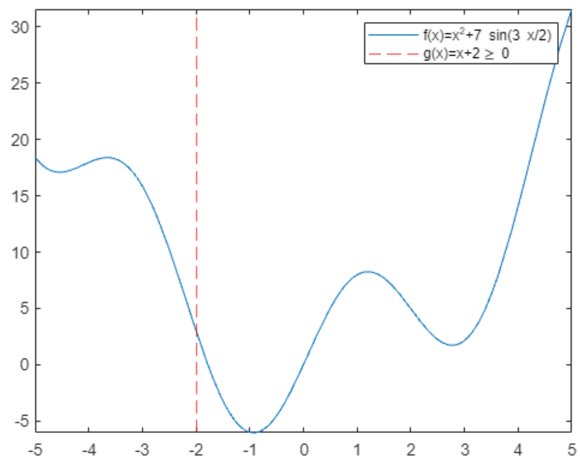

Figure 1: Search space defined by an example objective function, constraint and single decision variable

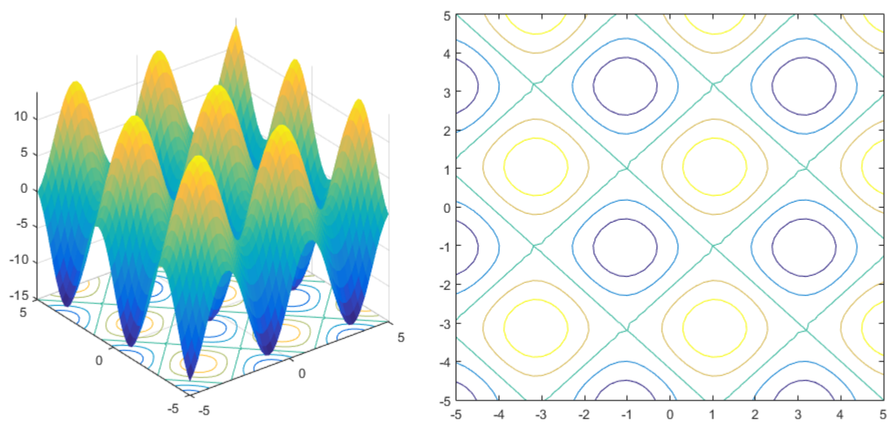

Figure 2: Search space defined by example objective function with two decision variables

Let us now understand the physical interpretation of different components of an optimization problem. Consider the following problem:

$$\text{Minimize: } f(x) = x^2 + 7\,\sin(3x/2) \tag{4}$$

$$\text{Subject to: } g(x) = x + 2 \geq 0 \tag{5}$$

$$\text{Where: } -5 \leq x \leq 5 \tag{6}$$

The above problem is a minimization problem having only one objective function, one decision variable and a constraint. The range of the decision variable is specified as $-5 \leq x \leq 5$. The Fig. 1 shows the shape of the objective function i.e. how the search space defined by it for the range of decision variable x. However, the constraint has restricted the values beyond -2 as invalid indicated with the dotted line. If the same objective function is considered with two decision variables (without constraint) then the function will be as follows:

$$\text{Minimize: } f(x_1, x_2) = x_1^2 + 7\,\sin(3x_1/2) + x_2^2 + 7\,\sin(3x_2/2) \tag{7}$$

$$\text{Where: } -5 \leq x_1, x_2 \leq 5 \tag{8}$$

In the above problem, two decision variables make the problem two dimensional or in other words, number of decision variables determines the dimensionality of the optimization problem. The search space of the problem is presented in 3-D and 2-D as shown in Fig. 2. Important to note that optimization problems can have multiple optima, among which the best one is referred as global optima [7]. There can be multiple global optimas for certain problems. The concepts discussed above in the context of minimization problems are applicable to maximization problems as well.

Despite the similarity among most optimization problems, they can be classified into different categories based on how they are formulated or their characteristics. Firstly, considering the problem formulation, the optimization problem can be classified as shown in Fig. 3. The goal of the optimization problem can be to minimize or maximize or mixed. For minimization problems, the goal is to find the minimum function value the objective functions for all functions. For maximization problems, the goal is to find the maximum function value of the objective functions for all functions. While both goals can be considered, where some of the objective functions' goal will be to minimize and some functions' it will be to maximize.

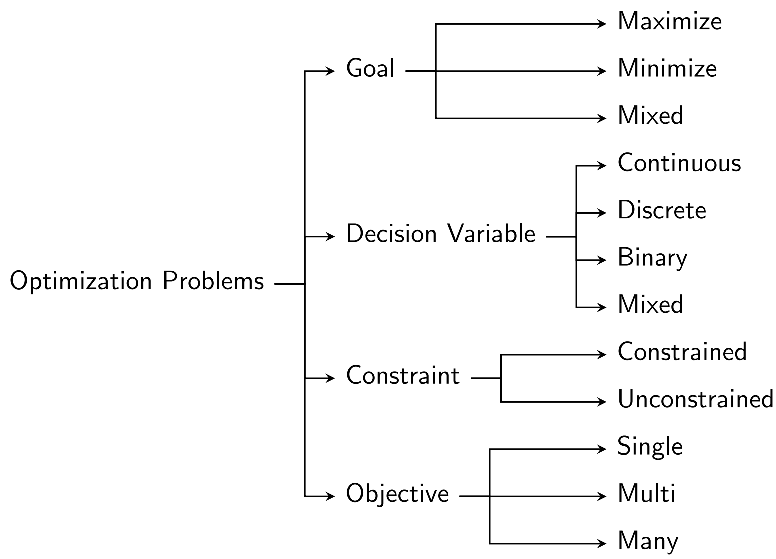

Figure 3: Classification of optimization problems based on problem formulation

Likewise, decision variables can have either continuous or discrete values. When having continuous decision variables, the search space is also continuous and infinite (or specific region defined by the constrains or ranges of decision variable). On the other hand, discrete optimization problems have variables of discrete nature (i.e. certain set of values) that lead to having a finite search space. Mostly, the real-world optimization problems are having mixed decision variables, which can be also divided into two classes unconstrained or constrained.

In terms of the objective functions, optimization problems may have one or more than one objective. A single-objective optimization problem has one objective function, so there is usually only one global optimum. In problems with multiple objectives, however, has more than one objective functions and multiple solutions can be found representing the best trade-offs between the objectives. The category of many-objective refers to problems with many objectives. Though both multi-objective and many-objective optimization have more than one objective functions, they have slight difference that is in case of many-objective problems number of objectives are comparatively larger. The name, many-objective, was coined by researchers to highlight the importance of an ever-increasing number of objectives and the complexity of addressing them all simultaneously.

Optimization problems can also be categorized based on the characteris-

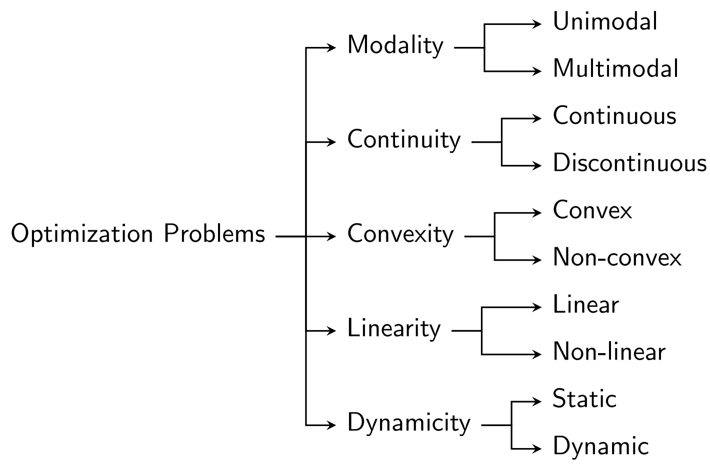

Figure 4: Classification of optimization problems based on problem characteristics

tics of their objective function as shown in Fig. 4. Modality of optimization problem is determines the number of optima present in the search space defined by the objective functions. Optimization problems can have one or more than optima, which are referred as unimodal or multimodal problems respectively. In terms of continuity of the search space defined by the objective, optimization problems are classified into continuous and discontinuous. If highest exponent of the decision variables with which the objective function is defined as well as constraints have highest exponent one then the function will be linear, otherwise it will be nonlinear. Linear optimization problems are comparatively simpler, while nonlinear optimization problems are complex and realistic problems where the objective function and/or constraints are nonlinear.

Again depending on whether the objective function's convexity, the optimization problems can be categorized as convex or non-convex optimization. A function is said to be convex at any interval if, for all pairs of points on the graph, the line segment that connects these two points passes above the curve. An example is shown in Fig. 5. All of the constraints as well as the objectives of a convex function for convex optimization problems. If the problem is minimizing function will be convex, where it will be a concave function if maximizing. Mostly linear optimization problems are convex and non-linear are non-convex only.

Real-life optimization problems in general emerges from different applications, where execution of different phases require different optimal settings

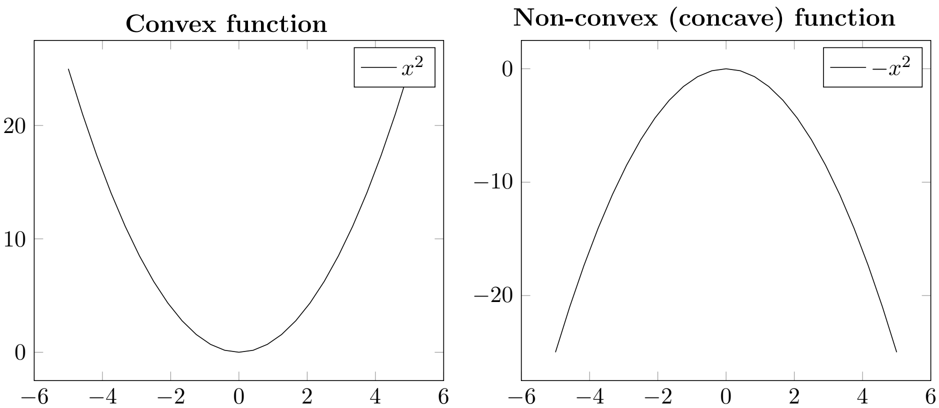

Figure 5: Examples of convex and non-convex functions

of parameters. Thus, the optimal values for the same optimization problem changes over time. The optimization problems which have this kind of characteristics are referred as dynamic optimization problem. In contrast, optimal values of static optimization problems do not change over the time or different phases of the application.

2. Nature-inspired Optimization Techniques

The development of computers and advancements in computational methods have greatly expanded the scope and complexity of optimization problems that can be tackled. Numerous techniques have been developed for solving various types of optimization problems discussed above. Optimization techniques can be classified into two main categories: deterministic and stochastic. Deterministic optimization algorithms aim to find the best solution based on a set of rules or mathematical models, while stochastic algorithms use randomness to explore the solution space and find better solutions [8, 9, 10, 11]. Deterministic algorithms, such as linear programming and dynamic programming, have been widely used in OR, while stochastic algorithms, such as genetic algorithms and particle swarm optimization, have emerged from the field of nature-inspired optimization. Our focus will be mainly on nature-inspired optimization techniques that are developed for solving optimization problems. A generic flow diagram of nature-inspired optimization techniques is presented in Fig. 6.

Nature-inspired optimization techniques mimic natural phenomena, such as natural selection, swarm behavior, and animal movements, to solve com-

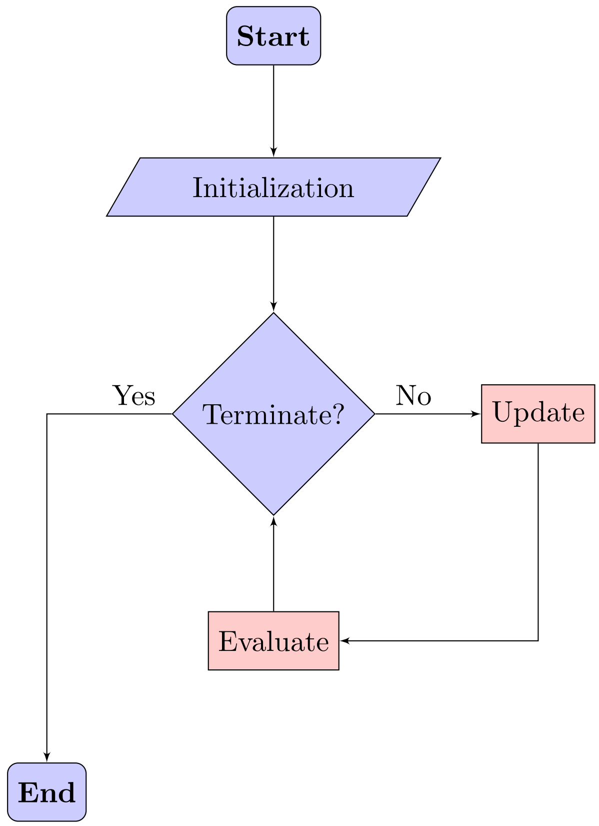

Figure 6: Generic flow diagram of nature-inspired optimization techniques

plex problems. These techniques are inspired by the observation of how biological systems adapt to their environment, and they seek to replicate the way living organisms behave or learn. Irrespective of the natural phenomena it is based on, the nature-inspired optimization techniques comprise four key components: Initialization, Terminate, Update, and Evaluate. The nature-inspired optimization algorithms are initialized with a population comprising a set of solutions and in each iteration the population is updated and evaluated until meets the termination condition.

The development of nature-inspired optimization techniques was motivated by the need for finding effective solutions to complex optimization problems, typically those problems for which traditional or techniques fail, or cannot deliver a solution in a reasonable amount of time. Furthermore, traditional or optimization techniques rely on mathematical models that may not be able to capture the complex relationships between variables and constraints that occur in real-world problems. Nature-inspired optimization techniques, on the other hand, use heuristics that can explore the solution space more effectively, making them well-suited to finding acceptable solutions for harder problems.

While nature-inspired optimization can feature a non-negligible track record of success stories, these techniques still have known drawbacks. Being stochastic in nature, there have no guarantee of finding the best possible solution (global optimum), and two different runs of the same algorithm with the same parameters might lead to different solutions. When compared to techniques that make assumptions on the regularity of the function they aim to optimize, such as gradient-based optimization algorithms, nature-inspired optimization algorithms can be considerably more computationally expensive, as they typically need to evaluate each candidate solution produced during their search in the search space [12, 13]

Despite these limitations, nature-inspired algorithms are currently considered among the state of the art in several domains. Even more classical OR approaches are seeing an increased hybridization with stochastic elements: for example, traditional gradient-based optimization techniques have largely been replaced by stochastic gradient descent and related algorithms, such as adaptive momentum, for the tuning of large neural network models. Nature-inspired optimization techniques have also seen the reverse, incorporating deterministic algorithms to perform local search, as in the case of memetic algorithms.

The "*Applications of Nature-inspired Computing and Optimization Techniques*" aims at portraying a selection of case studies where nature-inspired optimization has been successfully applied to solve problems from a variety of different real-world domains, ranging from summarizing legal documents to modeling air quality. The book primarily focuses on the practical challenges that are faced while applying nature-inspired algorithms to any specific problem, which include the feasibility of the problem, control parameters and constraints, representation of the solution space, and design of the objective function.

3. Application Areas

Nature-inspired optimization techniques have emerged as powerful tools for solving complex problems in a wide range of practical applications. These methods draw inspiration from natural phenomena and biological systems to develop efficient algorithms that can find near-optimal solutions. Such techniques are especially useful in domains where exact optimization is not applicable, due to the large search spaces or non-linearity of the objective function. Applications presented in this book include domains such as Controller and Power Systems, Ecological and Economic Systems, Information and Computational Systems, Communication and Networking Systems, and Deep Learning and Neural Networking Systems. A considerable part of the applications in computer science are focused on deep learning, currently the dominant paradigm in machine learning, where nature-inspired algorithms are used to optimize the architecture of large neural networks. The use of nature-inspired optimization techniques has greatly contributed to these fields and has enabled researchers to tackle complex problems that were previously intractable.

3.1. Controller and Power Systems

Control problems are common in avionics and engineering, involve the design of control systems that can regulate the behavior of complex dynamic systems, such as aircraft and industrial processes, power systems under uncertain and changing conditions, while satisfying multiple constraints, such as safety, efficiency, and stability. Most of these problems comes under non-linear categories. The book includes four chapters on such applications. In chapter 2, *Overview of Non-linear Interval Optimization Problems* starts with various kinds of non-linear optimization problems, with special emphasis on non-linear interval optimization problems. In Chapter 3, *Solving the Aircraft Landing Problem using the Bee Colony Optimization Algorithm*, the authors propose a heuristic approach based on Bee Colony Optimization to solve the Aircraft Landing Problem in a static version, assuming that the information on scheduled aircraft is known in advance. The target objective is to minimize the total deviation of all aircraft from the target landing times while adhering to specific constraints. Chapter 4, *Situation-based Genetic Network Programming to Solve Agent Control Problems* deals with finding agent control strategies in complex environments, using the Tile-World benchmark as a case study. The authors employ an Evolutionary Algorithm, Situation-based Genetic Network Programming, to generate a strategy for each agent based on its situation, rather than finding an optimal strategy for all agents. This approach improves the performance of traditional Genetic Network Programming, and can be added to all versions of the optimization algorithm without additional overhead. Results show that the proposed method can achieve the goal more easily and quickly than finding a strategy that can guide all agents, and generates more flexible solutions. In Chapter 5, *Small Signal Stability Enhancement of Large Interconnected*

Power System using Grasshopper Optimization Algorithm Tuned Power System Stabilizer tackles the optimization of a power system stabilizer, used to damp potentially disruptive low-frequency oscillations in power systems and achieve small signal stability. The authors propose the use of a Grasshopper Optimization Algorithm to optimize an objective function consisting of eigenvalues and damping ratios, searching for optimal control parameters for the stabilizer.

3.2. Ecological and Economic Systems

Biological and Ecological problems are among the most pressing challenges we face today, from climate change to loss of biodiversity and habitat destruction. To address these complex issues, optimization techniques can play a critical role. Optimization approaches can help us model and simulate ecological systems and predict their behavior, identify optimal management strategies for conservation and resource allocation, and improve our understanding of how biological processes work. Furthermore, optimization techniques can help us design more sustainable and efficient systems, such as renewable energy grids and eco-friendly buildings. By using optimization techniques to address ecological and biological problems, we have the potential to greatly benefit the environment, society, and future generations. Chapter 6, *Air Quality Modelling for Smart Cities of India by Nature Inspired AI – A Sustainable Approach*, tackles the issue of air quality modeling to estimate the correlation between pollution levels and their impacts on air quality. The authors propose the use of Particle Swarm Optimization merged as an optimizer for an Artificial Neural Networks tasked to predict air quality in seven Indian smart cities. The proposed approach has outstanding performance while evaluating high-dimensional data, which can help air quality managers forecast the effects of prospective new emissions and policy makers predict ambient air pollution concentrations under various scenarios.

In Chapter 7, *Genetic Algorithm for the Optimization of Infectiological Parameter Values under Different Nutritional Status*, proposes to use evolutionary optimization to find the parameters of mathematical equations describing the severity of infectious diseases. Optimizing these parameters can aid in identifying precautionary measures to mitigate losses in adverse situations. This chapter proposes mathematical equations for the infectiological parameters under normal-nutritional status and malnutrition and integrates nutritional status levels with a genetic algorithm to solve the formulated equations. Experimental results show that malnutrition negatively influences the optimization of infectiological parameters, and susceptibility, infection, and recovery are best optimized under normal-nutritional status, considering heuristic crossover for both sets of values. Chapter 8, *A Novel Influencer Mutation Strategy for Nature-inspired Optimization Algorithms to Solve Electricity Price Forecasting Problem*, proposes a novel approach to the electricity price forecasting problem, an important issue for optimizing the management of energy grids including different types of electricity sources, from renewable, to carbon-based, to nuclear. The approach is based on a new mutation strategy that can be integrated with several nature-inspired optimization algorithms to improve their convergence rate and avoid local stagnation problems. The strategy involves a group of particles, called an influencer group, consisting of top-performing particles that guide the remaining particles towards the optimal point. In Chapter 9, *Recent Trends in Human and Bio Inspired Computing: Use Case Study from Retail Perspective*, the authors apply nature-inspired optimization techniques to the task of extracting semantic descriptors from images and videos in the apparel and fashion industry. Specifically, the authors use Harris Hawk optimization and Progressive Spinalnet algorithms, obtaining satisfying results.

3.3. Information and Computational Systems

Nature-inspired optimization techniques are a popular approach for problems in the field of Computer Science. In particular, the recent developments in the field of information processing systems pose a large number of novel challenges, most of which cannot be tackled resorting to classical optimization techniques or exhaustive search. Chapter 10, *Domain Knowledge Enriched Summarization of Legal Judgment Documents via Grey Wolf Optimization*, proposes a nature-inspired approach to extractive summarization of legal documents, by modeling the task as an optimization problem. The proposed objective function is infused with domain-specific knowledge and pre-trained embeddings to better score candidate summaries. The experimental evaluation is conducted on an annotated Indian Legal Judgement document summarization dataset using ROUGE metrics, with promising results that can have practical utility in summarizing lengthy legal documents.

Chapter 11, *Bio-Intelligent computing and optimization techniques for developing computerized solutions* presents a short survey on the use of nature-inspired in different related domains: design of routing algorithms, sensors and visual systems, auditory systems, brain-controlled systems, and artificial intelligence. Chapter 12, *Optimizing the Feature Selection Methods using a Novel Approach Inspired by the Teaching-Learning-Based Optimization Algorithm for Student Performance Prediction* deals with a pivotal issue in machine learning, feature selection. In feature selection, an algorithm is tasked with selecting the minimal amount of informative input variables for a specific machine learning algorithm, to improve training speed and interpretability of the results. The novel approach to feature selection introduced in the chapter is tested on educational data, where it improves the performance of several different machine learning algorithms.

3.4. Communication and Networking Systems

Optimization techniques play an important role in tackling problems in communication and networking systems. With the increasing complexity and size of communication systems, optimization techniques can be used to improve their performance, reduce their cost, and enhance their reliability. In circuit design, optimization algorithms can help to find the optimal configuration of components and their values, reducing power consumption and improving the overall efficiency. In addition, optimization techniques can be used in manufacturing processes to minimize defects, increase yield, and reduce the time and cost of production. Therefore, optimization techniques can significantly benefit the electronics industry by providing innovative solutions to challenging problems, improving the quality of electronic products related to networking systems, and optimizing their performance. In Chapter 13, *Applying Evolutionary Methods for the Optimization of an Intrusion Detection System to Detect Anomalies in Network Traffic Flow*, the authors propose the use of different evolutionary algorithms, including Particle Swarm Optimization, to optimize the weights and improve the performance of a Multivariate Statistical Network Monitoring, a state-of-the-art system for detect different types of attacks with high performance. The proposed approach performs better than the classical optimization algorithm used for setting the system weights.

Chapter 14, *Modified Grey Wolf Optimization in user scheduling and antenna selection in MU-MIMO Uplink System*, tackles the challenges of user scheduling and antenna selection in multi-user multiple-input multiple-output wireless communication networks. Exhaustive search algorithms can deliver optimal solutions, but their computational cost is prohibitive: techniques such as ant colony optimization, binary particle swarm optimization, and binary grey wolf optimization are employed to reduce the complexity of the scheduling algorithm, achieving high throughput with much less processing load. In Chapter 15, *Spectral Efficiency Optimization by the Application of Metaheuristic Optimization Technique*, the authors employ different optimization techniques, such as binary flower pollination algorithm, binary spider monkey optimization, and binary artificial bee colony optimization, to solve the combined user and receiver antenna scheduling problems in the downlink channel of multi-user multiple-input multiple-outout systems. Such communication systems increase system throughput and spectral efficiency in networks, and their efficient scheduling of users is critical for optimal system capacity. In Chapter 16, *An effective Genetic Algorithm for Solving Traveling Salesman Problem with Group Theory*, a novel variant of Genetic algorithm has been proposed solving traveling salesman problem that explores group theory for initial population generation. In the group tour construction method, each individual tour has distinct start city provided that population size is equal to total number of cities. In the initial population, each individual i.e. tour has a distinct starting city. The distinct starting cites of each tour provide genetic material for exploration for the whole search space. Therefore a heterogeneous starting city of a tour in initial population is generated to have rich diversity. Also introduced crossover based on greedy method of sub-tour connection drives the efficient local search, followed by 2-opt mutation for improvement of tour for optimal solution.

3.5. Deep Learning and Neural Networking Systems

With the recent emergence of the deep learning field in machine learning, nature-inspired techniques found several niches optimization problems with vast search spaces, where classical approaches fail. Most of these challenges lie in the domain of hyperparameter tuning, in other words, finding the best structure, activation function, number of neurons, and so on, for deep neural networks. Chapter 17, *Adaptation of Nature Inspired Optimization Algorithms for Deep Learning*, discussed different architecture design related adaptations required to use nature-inspired optimization techniques in deep learning models. Authors emphasis the need of optimal architectures for improvement of improve and enhance the performance of deep learning models. Various nature-inspired optimization techniques that are being explored in recent years for both optimal architecture designing and weight adaptation for deep learning models. Various approaches that are followed for solution representation, objective function design, and constraint handling in respect to deep learning models are detailed as well as highlighted key challenges encounters during incorporation of nature-inspired optimization techniques in deep learning models.

In Chapter 18, *Long Short-Term Memory Tuning by Enhanced Harris Hawks Optimization Algorithm for Crude Oil Price Forecasting*, the authors propose the use of the Harris Hawk optimization algorithm to tune the hyperparameters of a long-short-term memory network, a Deep Learning architecture used for time series forecasting. The methodology is applied to real-world time series describing the price of crude oil, showing better results than the heuristics commonly used for this task. Chapter 19, *Discovering the Characteristic Set of Metaheuristic Algorithm to Adapt with Adaptive Neuro-Fuzzy Inference System Model*, focuses on the optimization of the parameters of Adaptive Neuro-Fuzzy Inference Systems, a hybrid machine learning architecture using both parts of Artificial Neural Network and Fuzzy Logic, used to create models for classification and regression tasks. Six popular metaheuristic algorithms are assessed, and their results are compared. The study finds that metaheuristic algorithms based on evolutionary computation are more stable than swarm intelligence methods in tuning the parameters of Adaptive Neuro-Fuzzy Inference Systems. Chapter 20, *Artificial Neural Network Optimized with PSO to Estimate the Interfacial Properties between Fiber Reinforced Polymer and Concrete Surface*, tackles the problem of deterioration of concrete structures and the need for sustainable solutions such as strengthening with Fibre Reinforced Polymer composites. The bond between the concrete and the composite is crucial for effective strengthening, but predicting the bond strength is a complex task. A novel approach based on an Artificial Neural Network optimized with Particle Swarm Optimization is proposed, and the methodology is shown to deliver better results than analytical models and classic Artificial Neural Networks.

4. Concluding Remarks

Nature-inspired computation techniques have been used in a wide range of applications to solve complex and large-scale optimization problems in various domains. This book focuses on a few representative application domains that are emerging in recent years, including controller and power systems, ecological and economic systems, information and computational systems, communication and networking systems and lastly, deep learning and neural network models. In addition to other domains, a considerable part of the applications in computer science are nowadays focused on deep learning, currently the dominant paradigm in machine learning, whereas nature-inspired algorithms are often used to optimize the architecture of large neural networks. The use of nature-inspired optimization techniques has greatly contributed to these fields and has enabled researchers to tackle complex problems that were previously intractable. While nature-inspired optimization techniques are being applied to diverse domains, these techniques still have well-known drawbacks. It is also interesting to mention that the proliferation of novel nature-inspired meta-heuristics has drawn severe criticism from researchers, highlighting how some of the newest proposed approaches have been poorly tested and are essentially slightly modified versions of established algorithms. Nevertheless, nature-inspired algorithms are enjoying a growing popularity, and are currently considered among the state of the art in numerous domains.

References

[1] C. Dimopoulos, A. M. Zalzala, Recent developments in evolutionary computation for manufacturing optimization: problems, solutions, and comparisons, IEEE transactions on evolutionary computation 4 (2) (2000) 93–113.

[2] H. Tuy, Monotonic optimization: Problems and solution approaches, SIAM Journal on Optimization 11 (2) (2000) 464–494.

[3] X. Hu, R. Eberhart, et al., Solving constrained nonlinear optimization problems with particle swarm optimization, in: Proceedings of the sixth world multiconference on systemics, cybernetics and informatics, Vol. 5, Citeseer, 2002, pp. 203–206.

[4] A. Seyyedabbasi, F. Kiani, Sand cat swarm optimization: A nature-inspired algorithm to solve global optimization problems, Engineering with Computers 39 (4) (2023) 2627–2651.

[5] R. Zhao, F. Zhu, M. Tang, L. He, Profit maximization in cache-aided intelligent computing networks, Physical Communication 58 (2023) 102065.

[6] S. Bartels, A. Kaltenbach, Explicit and efficient error estimation for convex minimization problems, Mathematics of Computation 92 (343) (2023) 2247–2279.

[7] T. Weise, Global optimization algorithms-theory and application, Self-Published Thomas Weise 361 (2009).

[8] M. Cavazzuti, M. Cavazzuti, Deterministic optimization, Optimization methods: From theory to design scientific and technological aspects in mechanics (2013) 77–102.

[9] S. Uryasev, P. M. Pardalos, Stochastic optimization: algorithms and applications, Vol. 54, Springer Science & Business Media, 2013.

[10] J. C. Spall, Stochastic optimization, Handbook of computational statistics: Concepts and methods (2012) 173–201.

[11] L. Liberti, S. Kucherenko, Comparison of deterministic and stochastic approaches to global optimization, International Transactions in Operational Research 12 (3) (2005) 263–285.

[12] K. Sörensen, Metaheuristics-the metaphor exposed, International Transactions in Operational Research 22 (1) (2013) 3–18. doi:10.1111/itor.12001.
URL `https://doi.org/10.1111/itor.12001`

[13] C. Aranha, C. L. C. Villalón, F. Campelo, M. Dorigo, R. Ruiz, M. Sevaux, K. Sörensen, T. Stützle, Metaphor-based metaheuristics, a call for action: the elephant in the room, Swarm Intelligence 16 (1) (2021) 1–6. doi:10.1007/s11721-021-00202-9.
URL `https://doi.org/10.1007/s11721-021-00202-9`