Discrete Problems in Nature Inspired Algorithms

Anupam Shukla and Ritu Tiwari

CRC Press
Taylor & Francis Group

Discrete Problems in Nature Inspired Algorithms

Taylor & Francis
Taylor & Francis Group
http://taylorandfrancis.com

Discrete Problems in Nature Inspired Algorithms

Anupam Shukla
Ritu Tiwari

CRC Press
Taylor & Francis Group
Boca Raton London New York

CRC Press is an imprint of the
Taylor & Francis Group, an **informa** business

CRC Press
Taylor & Francis Group
6000 Broken Sound Parkway NW, Suite 300
Boca Raton, FL 33487-2742

Printed on acid-free paper

International Standard Book Number-13: 978-1-1381-9606-3 (Hardback)

Visit the Taylor & Francis Web site at
http://www.taylorandfrancis.com

and the CRC Press Web site at
http://www.crcpress.com

Contents

Foreword I

Nature-inspired algorithms are currently gaining increasing popularity for their widespread applications in diverse domains of scientific and engineering optimization problems. There exists a vast literature on nature-inspired algorithms, spread across scientific journals, conference proceedings, and so on. Unfortunately, I did not come across any book covering all aspects of nature-inspired algorithms. This book by Professor Anupam Shukla and Dr. Ritu Tiwari fills this void. It covers almost all the aspects of nature-inspired algorithms in a single, precise, and resourceful volume.

A lot of promising nature-inspired algorithms have emerged over the past two decades. Particle swarm optimization, ant colony optimization, bat algorithm, artificial bee colony algorithm, bacteria foraging optimization algorithm, shuffled frog-leaping algorithm, cuckoo search algorithm, invasive weed optimization algorithm, and flower pollination algorithm are just a few to mention in this regard. At this point, there is a need to educate people about the principles, concepts, applications, issues, and solutions related to the use of nature-inspired algorithms.

I am sure that this book will be of great contribution toward the same and would further carve a deep mark in the nature-inspired algorithms literature for the reference of enthusiasts. It is unique for its content, readability, and above all the presentation style. This book includes both the theoretical foundations along with a variety of applications that further help in understanding the different nature-inspired algorithms. It is always considered to be good to have a practical component attached to the courses whose performance ultimately reflects the overall understanding of the students. Ample coverage of applications, issues, and perspectives, makes this book rich and diverse. This book will be very useful for the students, academicians, and researchers studying/working in the field related to nature-inspired algorithms. It would be equally useful to researchers migrating from mathematics and other disciplines to computer science.

Amit Konar
Jadavpur University, Kolkata

Foreword II

This book provides a systematic introduction to most of the popular nature-inspired algorithms for optimization, including particle swarm optimization, ant colony optimization, bat algorithm, artificial bee colony algorithm, shuffled frog-leaping algorithm, invasive weed optimization algorithm, flower pollination algorithm, and so on.

Nature-inspired computation is one of the most promising areas of research and has a lot of scope in the near future as well. Nature-inspired algorithms have very diverse application areas, which are growing steadily. Nature-inspired algorithms have in fact become the most widely used optimization algorithms. This book covers the theoretical background including extensive literature review of the nature-inspired algorithms and the practical implementations of these algorithms for solving various discrete optimization problems.

This book will definitely be useful as a reference book on nature-inspired algorithms for graduates, postgraduates, doctoral students, researchers, and even faculties in computer science, engineering, and natural sciences.

P.N. Suganthan
Nanyang Technological University

Foreword III

Combinatorial optimization forms an important area under theoretical computer science and applied mathematics with several industrial applications and it aims at identifying an optimal solution from a finite set of solutions. The solutions are normally discrete or can be formed into discrete structures. In most of such problems, the exhaustive search becomes intractable, necessitating various forms of intelligent search and optimization. This book—*Discrete Problems in Nature Inspired Algorithms*, authored by Professor Anupam Shukla and Dr. Ritu Tiwari—comprises a great treatise on solving the usually NP-complete combinatorial optimization problems by using various nature-inspired metaheuristic algorithms. Starting with a lucid introduction to various combinatorial optimization scenarios, the author has aptly guided the reader through a comprehensive journey of the development of the metaheuristics, their qualitative advantages and downsides, and applicability to various related problem domains. The exposure of the material is lucid. Quite complicated concepts are presented in a clear and convincing way, which can be attributed to the expertise of the author.

Finally, I must conclude that this is a very timely volume with a well-compiled exposure of the cutting edge research on and with nature-inspired discrete optimization algorithms. I am quite sure that the researchers and practitioners from the related fields will be immensely benefitted from this book.

Swagatam Das
Indian Statistical Institute, Kolkata

Preface

Nature has been a constant source of inspiration for scientists since centuries. Various laws and theories have been formulated by observing the different phenomena that occur in nature. *Nature-inspired algorithms* is an area inspired by natural processes; in the recent past, many algorithms have been developed taking inspiration from certain biological, geographical, and chemical processes that take place in the natural world. Nature-inspired algorithms have the immense potential for optimizing complex real-world problems and have been applied in various areas. In the past two decades, plethora of new and promising nature-inspired algorithms has been introduced. The highly multidisciplinary nature of the field further has attracted a number of researchers. There is a mammoth of literature available in the field of nature-inspired algorithms. The current literature covers the theoretic concepts, but the practical applications might still be very difficult for students to comprehend and implement. Many students find it difficult to visualize the application of nature-inspired algorithms from theoretical text.

This book not only helps them in understanding these algorithms but also exposes them to the current developments over various related fields. This empowers them to pursue research in these areas and contribute toward research and development at large. The main purpose of this book is to explain the nature-inspired algorithms theoretically and in terms of their practical applications to real-world problems to the readers. This will enable them to get an in-depth, practical exposure to the methodology of the application of the nature-inspired algorithms in various situations.

This book can also be used as a standard text or reference book for courses related to nature-inspired algorithms, evolutionary algorithms, and so on. This book may be referred for the purpose of *elementary* and *advanced* studies as it introduces the basic concepts of nature-inspired computing and then also discusses some state-of-the-art variants of those basic algorithms and then finally explains their practical examples. This book also incorporates some of the recent developments over various areas.

Salient Features

- Introduction to the world of nature-inspired computing
- Foundations of the basic nature-inspired algorithms
- Detailed description of the problem and solution
- Information on recent developments in the various nature-inspired algorithms
- Interdisciplinary applications of the nature-inspired algorithms

Intended Readers

- Students of undergraduate, postgraduate, doctorate, and postdoctorate levels can refer this book as a text or reference book for nature-inspired algorithms.
- Researchers can refer this book to obtain a good insight into the recent developments in the field of nature-inspired algorithms.

Origin of This Book

This book is a result of the continuous research done over the time by the authors. Much of the work presented is done by the authors themselves. Databases to the various problems have either been self-generated by the authors or used from the public database repositories. At some places, the authors have also included a comprehensive discussion of the problems and the solutions present in the literature. This is a result of the efforts put in by various students of the ABV-Indian Institute of Information Technology and Management, Gwalior, India, in forms of projects: B.Tech, M.Tech, and PhD theses.

Acknowledgments

This book is our yet another step into the world of authored books. It began as a dream to use our past research as a base to create a landmark into the world of nature-inspired computing. It is natural that the dream required a lot of encouragement, guidance, and support that were provided by numerous people in various phases of this book. The authors acknowledge all those people who made this possible.

The authors thank Professor S.G. Deshmukh, director, ABV-IIITM, Gwalior, India, for extending all sorts of help and support during the course of writing this book. This book would not have been initiated without his encouragement and support.

The authors thank Professor. Amit Konar, Professor P.N. Suganthan, and Professor Swagatam Das for writing the foreword for this book. Their precious words have added a greater charm to this book. The authors further express their thanks to Dr. A.S. Zadgaonkar, ex-vice-chancellor, Dr. C.V. Raman University, Bilaspur, India, for the guidance and motivation that he has bestowed during the entire course of work with the authors.

This book would have never been possible without the pioneering work carried out by the coresearchers associated with the authors. The authors also thank the student fraternity of the Institute for taking deep interest into the courses titled "Special Topics in Intelligent Systems" and "Soft Computing." Their interest to learn beyond the classroom program has been the key motivation for this book. The authors are highly indebted to all the students who undertook challenging problems as a part of their thesis and came up with excellent solutions and experimental results. The authors thank their PhD students Sanjeev Sharma, Apoorva Mishra, and Saumil Maheshwari for their constant support during each phase of this book. The authors thank their postgraduation students Prashant Shrivastava, Chiranjib Sur, Annupriya, Riya Naval, Alok Sharma, Prashant Pandey, Harshita Lalwani, and others for the quality results they produced during their course of work with the authors. The authors further thank Pritesh Tiwari for helping in editing some parts of the manuscript.

The first author thanks his son Apurv Shukla for his support during the course of writing this book. The second author thanks her mother Parvati Tiwari and her late aunt Dr. Asha Shukla for motivation and blessings for writing this book.

This book required a lot of work by numerous people around the globe. The authors thank the entire staff of Taylor & Francis Group and all the associated units who have worked over different phases of this book. The authors thank all the people who have been associated with and have contributed to this book. Above all, we all thank the Almighty for his constant blessings and love. Without Him, this project would not have been possible.

Authors

Prof. Anupam Shukla is currently a professor in the Department of Information and Communication Technology (ICT) at ABV-Indian Institute of Information Technology and Management (ABV-IIITM), Gwalior, India. He has 29 years of administrative, research, and teaching experience. He is globally renowned for his research on artificial intelligence and robotics, which has won him several academic accolades and resulted in collaborations with academicians across the world.

He has been awarded "Distinguished Professor" by Computer Society of India, Mumbai, 2017; "Dewang Mehta National Education Award" for best Professor, 2016; "Professor Rais Ahmed Memorial Award," Acoustical Society of India in 2015; Engineering Professor of the Month, Higher Education Review, India, 2014; "Best Paper Award," Institute of Engineers, India, 2006; University Gold Medal (Valedictorian) from Jadavpur University, Kolkata, India, 1998, for scoring highest marks at master's level; and "Young Scientist Award," Madhya Pradesh Council of Science and Technology, Bhopal, 1995.

He is the author of three patents, three books titled: (1) *Real Life Applications of Soft Computing*, Taylor & Francis; (2) *Intelligent Planning for Mobile Robotics: Algorithmic Approaches*, IGI Global; and (3) *Towards Hybrid and Adaptive Computing: A Perspective*, Springer–Verlag Publishers, editor of three books published by IGI Global Press, United States, and mentor of 17 doctorate and 106 postgraduate theses. He has 174 peer-reviewed publications and is a reviewer of various international journals, including *ACM Computing Review*, *IEEE Transactions on Information Technology in Biomedicine*, and *Elsevier Neurocomputing journal*. He has successfully completed 13 government-sponsored projects, aimed at developing IT applications for Indian farmers, skill development of the Indian youth and infrastructure development at parent institutes, and so on. He is currently a nominated expert member of the accreditation committee of National Board of Accreditation, India. At NIT Raipur, he was the founding Head of the Department (HOD) of departments of biomedical engineering and biotechnology engineering and the founding coordinator of the Chhattisgarh–IIT Kanpur Knowledge Sharing Program, a first of its kind e-learning program in India.

Dr. Ritu Tiwari is an associate professor (Department of Information and Communications Technology) at ABV-IIITM, Gwalior, India. She has 15 years of teaching and research experience, which includes 10 years of post PhD teaching and research experience. Her field of research includes robotics, artificial intelligence, soft computing, and applications (biometrics, biomedical, and prediction).

She has two patents in her name and has authored three books titled: (1) *Real Life Applications of Soft Computing*, Taylor & Francis; (2) *Intelligent Planning for Mobile Robotics: Algorithmic Approaches*, IGI Global; and (3) *Towards Hybrid and Adaptive Computing: A Perspective*, Springer–Verlag Publishers. She has also edited two books in the area of biomedical engineering from IGI Global. She has supervised 7 PhD students and more than 100 master's students and has published 104 research papers in various national and international journals/conferences.

She has received "Young Scientist Award" from Chhattisgarh Council of Science & Technology, India, in the year 2006. She has also received gold medal in her postgraduation from National Institute of Technology (NIT), Raipur, India. She has completed 10 prestigious research projects sponsored by Department of science and technology (DST) and Department of Information Technology (DIT), Government of India. She is currently involved with the Government of India and is working on three sponsored research projects. She is a reviewer of various international journals, including *ACM Computing Review*, *IEEE Transactions on Information Technology in Biomedicine*, *Elsevier Journal of Biomedical Informatics*, and *Elsevier Neurocomputing journal*.

1

Introduction to Optimization Problems

1.1 Introduction

In the present world, we are obsessed with solving a huge number of problems. With the progress of science and technology, humans have been able to tame some of them [1]. Due to advancements in the field of automation, and knowing the fact that natural resources are on the verge of exhaustion, scientists, today, are paying a lot of attention on the low resource (regarding power, time, and occupancy) factor, which can be achieved through optimization. Optimization can be achieved at many levels.

The discovery of nature-inspired computing and its allied optimization branches has revolutionized computation and automation, the future of which lies in the hands of machine learning and self-adaptive systems [2–4].

One might wonder, why adaptive learning systems? In a real-life environment, every system is associated with some input and output mappings in which the inputs act as the main decisive factor for the outputs. However, unless the system is carefully designed, the range associated with the input and output data variables is highly unpredictable and nonstationary, and there exist a large number of samples. In such a scenario, it becomes a challenge for supervised learning to establish a nonlinear model, which can produce correct judgment through the efficient processing of data and intelligent decision-making. However, there always remain chances of deviation due to irregularities in the datasets. Hence, there is a requirement for the system to adapt to the appearance of a new data and the combination of situations.

This adaption is handled in various ways in various situations and it depends on the mathematics of the algorithms. In this chapter, we will introduce techniques for such kinds of adaption that are involved in nature-inspired metal-heuristics [5] on many problems.

This work can be marked as an introduction to a bunch of bio-inspired computation techniques, which can be readily applied to the graph-based and discrete optimization problems. Before moving further, the term *discrete optimization problems* should be explained and illustrated so that you can have a clear idea of what kind or class of problems we are referring to. Consider an equation of any straight line of the form $y = mx + c$. For any value of $x \in R$, where R is the set of Real numbers, the value of y is valid and acceptable and this is a simple continuous domain problem. There is no minimum or maximum value that can be achieved due to the nonconvergence attitude of this equation. On the other hand, an equation of the form $y = |x_1^2| + |x_2^2|$ has a minimum value at zero and x_1, x_2 both can have any value in R. What if we say that the equation $y = |x_1^2| + |x_2^2|$ also has a minimum value at zero but x_1, x_2 can only take integer values lying within Z, where Z is the set of Integers.

This is what is referred to as the fundamental discrete optimization problem in which the variable parameters take only integer values. However, this is not what we are looking for. We are going for a generalized discrete optimization problem set, which will have the above-mentioned kind of problems as special cases and we would not have to reformulate our equations.

Now, optimization may mean different things for different types of problems. For minimization as optimization of a function f, we have $f(x) \le f(a)$ where $\forall_x \in S$ and $S \subseteq B$ where S denotes the allowed representations for x and B is the constraint for the variable, which can be a domain or range or an integer value depending on the problem. Here, $a \in S$ and $f(a) = \mathrm{MIN}(f(x) : \forall_x \in S \subseteq B$ and is called the optimized solution or the global solution.

1.1.1 Artificial Intelligence

Artificial intelligence is defined as the capability of perfect sensing and intelligent decision-making of a system, which facilitates complex jobs and is worthy of usage for humankind and works under the influence of an external environment (virtual or real). Artificial intelligence can also be defined as the capability of the performance of the system, which cannot be achieved through a fixed sequence of instructions. Systems based on a fixed sequence of instructions would be biased, and an optimal decision would not be possible [6].

In other words, what artificial intelligence does is the same as done by a mechanical machine and that is it *makes life easy*. Similar to machines, it helps in performing a very complex task in the blink of an eye or with the press of a button. The same is done by a *simple computer program*, but can we call it an artificial intelligence system? No, we cannot! And there are several reasons for it. A *simple computer program* is just performing or executing some instructions and, at any point in time, it does not take any decisions, and it is not bothered whether the output generated is desired or not. As we are talking about taking decisions, it is obvious that more than one outcome or combinations of outcomes is involved, but the system will seek out the best or the optimized one (if it can, else it will let one decision go and later find out whether it was the best or if it is the best among all) for the given situation depending on some evaluation criteria. So, an intelligent system must be able to make complex judgments consisting of multiple parameters and situations taking care of the constraints and rules. In modern times, intelligent systems are adaptive and have the capability of learning from experience [6].

Consider the example of any search operation such as playing chess on a computer. The feedback from the users (i.e., the move they make) acts as an influence for the system to reproduce the optimal decision, which will make the game more interesting. Had the system taken any random step, its reaction would not have been considered intelligent, and the result of such a decision would have been a matter of probability and chance. However, sometimes randomness appears to be more useful than more deterministic approaches, especially when the combination of decisions is quite large and consequently the cost (both regarding the computational complexity and time) of estimation of fitness of such a decision is also huge. The purpose of adoption of random behavior in many decision-making functions is to reduce the dimension or the number of possible combinations for all the entities that are involved in the system. Another example is the search engine that had revolutionized the ways of information storage and its retrieval capability when the quantity of such information had grown boundless, and there was a severe requirement for search engines capable of procurement and filtering. What was mainly done was the incorporation of artificial intelligence into the query-making system, which sorted the information of a particular regime according to the importance, and this one kind of ranking system helped in the quick retrieval of relevant information much quicker than expected. However, the mode, criteria, conditions, and implementation of such kind of intelligence-driven engines differ from organization-to-organization, and that is why for particular word search different search engines hit on different sequences of web pages. Another issue that needs to be mentioned is that these search engines are also monitoring systems, which take account of the statistics of usage and frequency of visit at a website and, accordingly, evaluate and maintain the importance of the information of that site. It is worthy of mentioning that with usage, experience, and time the intelligence of the search engines tends to get changed and enhanced.

1.1.2 Soft Computing

Soft computing is another important sister branch of artificial intelligence in which the system starts acknowledging, accepting, and becomes capable of processing values from a continuous range, instead of grouped implicated values. The main conception arose from fuzzy logic in which the system design is flexible, and the data ranges are subdivided into more overlapping classifications and subclassifications. The implications have been proven to generate more accurate decisions and desirable results. In real world, sometimes it is very difficult to define or clearly demarcate ranges for a particular event such as temperature (hot or cold), height (tall or short), and turn (sharp or simple). The values of these events are relative, and numerical ranges are unclear. We do not know the numerical value for *hot temperature*. So, these events need more classifications before making a system intelligent enough to handle such range of values and provide proper decisions. In contrast to the hard rules what *soft computation* tried to develop is flexibility and rule-free classifications. As in the case of height if greater than a threshold is *tall* else *short*, what about the height that is very near to the threshold? Should it be classified as short or tall?

Another problem set that helped in the evolution of soft computing is the calculation of maxima and minima for the discontinuous functions. Differential calculus has been the key to finding the maximum and minimum for functions for years, but a function is only differential when it is continuous at that range. So, for the discontinuous functions, differential calculus is not valid, and this paved the way to the so vast field of nature-inspired computation and soft computing. Apart from fuzzy logic, two other important members of soft computation are neural network and genetic algorithm (GA), which also share their space with nature-inspired computation (as they are derived from the natural phenomenon). They are widely used for numerical optimization of several discontinuous mathematical functions. GA is discussed in detail in Chapter 3.

1.1.3 Intelligent Systems

Intelligent systems are the system, which are provided with an artificial intelligence and are capable of processing data and take decisions intelligently. Unlike expert systems [7], they can handle unknown environments and unknown events that are based on their decision-making capability. The main feature of this kind of system is that they can or may hover over all the possible solutions before making any decision. Playing chess on a computer is an example of an intelligent system. Before going for a turn, it will evaluate all the possible moves and then go for the best one. Intelligent systems are meant for external feedback of some limited range or number of inputs or inputs from a static environment.

1.1.4 Expert Systems

An expert system is one kind of intelligent system, which is more likely an intelligent pattern-matching system capable of making decisions based on the data that are fed as a part of the training. The expertise, intelligence, and decision-making are done on the basis of feed data only. Intelligence lies only for the known environment or the set of known events. A typical example can be a biometric system. An expert system is subjected to an unlimited number of external feedback as inputs, out of which it can only process correctly only the known ones. As for identification of a person, there are several characteristics that are unique. However, for big databases, a collection of such characters can be used as unique identification characteristics. The same is true when there are several features overlapping with the other entity in the same database. Some of the features are not adequate for describing the entity and in these kinds of situations the most overlapping features are regarded as noise factors. The relativity or correlation of noise depends on the number of entities it can represent. So, in an expert system, the learning models extract or rather learn more from the unique features and less from the other subset of features.

Theoretically, imagine a hypothetical situation where a student has to learn lessons. The first lesson is being learned sincerely rather exclusively, but if there are common features between kth and $(k + 1)$th lesson, then it is advisable to learn the extra features of both of them and only learn the common features from only one of them. So, basically, in a mathematical model the standing difference lies only in the unique features of both of them and this difference is then utilized as a feedback error for manipulation of the learning model through self-adjustable techniques such as back-propagation algorithm and so on.

Now the question is "What will be an intrusion detection system?" Is it an intelligent system or an expert system? An intrusion detection system monitors the behavior of a system and takes a decision based on the analysis of the processed data, but at the same time, it tries to linkup with the experience and previously collected knowledge of anomaly detection. It is very difficult to draw a strict classification margin between the systems and its types. So, intelligent systems rely on logic and expert systems rely on training, and there are data associated with them in common. In some sense intelligence of the expert system is *dumb intelligence*. However, the input data for both expert systems and intelligent systems have bounds and limitations, and decision-making for unknown data is characterized mainly by machine-learning intelligence.

1.1.5 Inference Systems

Intelligence or decision-making in inference systems is based on conclusive decisions made out of rational thinking and not on logic. But should we call it nonlogic decision-making? In an inference system, the control statements are cognitive statements, which may or may not be represented mathematically, and thus are very difficult to process for computation. The control statements arise out of rational thinking, and the basic meaning of the statement is not important, but its implications are important for articulating meaningful relationships.

Consider the two mathematical statements. One is $x > 10$ and another $y > x$. Combining the above-mentioned two statements we can conclude that $y > 10$. If the statements would have been like *x is greater than* 10 and *y is greater than x*, then also we could have concluded as *y is greater than* 10 but only through the use of mathematical logic. But what if the statements are not mathematical? For example, *The sun rises from The East* and *Mohan is looking at the rising sun*. The conclusion could have been drawn like *Mohan is looking toward east* (*unless he is using a mirror or watching some video*), but how? There is no mathematical logic behind it, and the conclusion is correct. An inference system plays the role of drawing decision and conclusions from this kind of statements better known as on-logic decision-making. However, there are challenges for inference system for comprehension and handling many nonconclusive sentences.

1.1.6 Machine Learning

Machine learning is another high-level abstraction-based learning phenomenon in which the algorithm gradually explores the environment and its objects and learns from it. Initial moves are random or user defined but later they are updated with the progress of exploration and are also on the basis of monitoring with the learned experience. All the learning process and its storage are updated in the form of state variables and numerical values that indicate some implications. Machine learning is an iterative process, and the system gets better with experience. There are several algorithms for machine learning, which are beyond the scope of this book. There is the strategy of analysis and learning, but there is no specific rule for data representation and learning models in machine learning and each one of them have their own advantages over the other. In some cases one technique proves to be better than the others, whereas in some different situation, the other technique might outperform it. This concept is known as no free lunch theorem [8].

Consider the YouTube search engine; when you search for any video, it not only provides all its versions but also some other video, which is somehow relevant to the searched one and most of the times they are pertinent and useful mainly during the less specific generic word(s) search. How does a search engine know what was I looking for and, more importantly, my taste? There is nothing called magic, and there is hardly any person who is going to spend so much time linking up the similar videos among huge data storage. There are machine-learning algorithms running behind and from the pattern of people watching the videos, the algorithm learns and, accordingly, set up links that are based on priority and frequency of visits, and this statistical monitoring and mathematical framework are acting as a suggestion for other viewers. The machine-learning algorithmic process of refinement of suggestions is getting better day by day. Initially, when there was no learned suggestion model, the suggestion links were random or based on cluster bounded by region, state, or country. Even now the filtering and suggestion based on the country can be manually set in YouTube. Filter bounds are there to make the suggestions and filtering process less complex, and thus the diameter of the graph representing links between a video and its suggestions remains low.

1.1.7 Adaptive Learning

Adaptive learning is yet another process of learning-based models for optimization and control-based systems. During the search process for system optimization, there is a requirement for adaptively learning and responding to the requirement of step variation and identifying the regions where there is more possibility of finding optimal or near optimal solutions occurs. This kind of response is obtained through a feedback generated by a mathematical representation whose response is adaptive and based on the parameters and evaluation generated. This learning model helps us to quick convergence rate and local search and prevents unnecessary swing of space search domain. They are mainly used in the control system and mathematical optimization. In the control system, the strength of control features needs to be adjusted according to the ongoing state of the system and thus, need adaptive mathematical modeling. For example, in a power plant, the inputs of fuel need to be regulated if the temperature or pressure or both are high. However, for numerical optimization of mathematical equations, the step-wise variation must direct toward the place where there always occurs improvement of fitness and when the peak near the step-wise variations is small. However, there is another important requirement for adaptive learning for the neural network during the training phase in which the parameters of the intelligent system (consisting of neural network models) need to be manipulated or tuned so that the error between the original output and the expected output gradually disappears. This kind of system and adaptive control is very helpful in biometric systems, pattern matching-based expert systems, and numerous applications.

1.1.8 Related Work Done

1.1.8.1 Heuristics

Heuristics can be defined as a cognitive search-based algorithm for quick optimization or near-optimized solution, which cannot get solved with the conventional mathematical tools and exhaustive search techniques. In other words, such systems or problems cannot be represented by continuous gradient-based mathematical representation or may have multiple local peaks for minima or maxima. Such cases are similar to nonconvex problems such as multimodal problems, multilocal optima problems, and discontinuous equation-based problems, and are difficult to be solved analytically and sometimes infeasible and impractical. The solutions are measured with the help of an evaluation function or called heuristic function, which is convenient in many cases, but in the majority of the systems and their modeling (such as real life and real time), it is very difficult to achieve and formulate that function. The convenience in the determination of this numerical evaluation is achieved through abstraction and efficient data processing. Examples of heuristic searches are dominated by only random exploration or by the combination of exploration and exploitation. Typical and simple examples of heuristics are educated or wild guess, an intuitive judgment-based guess, sixth sense, and so on.

1.1.8.2 Meta-Heuristics

Meta-heuristics is a semirandomized search algorithm in which the optimal solution of an NP-Hard problem is derived out through the cooperation of agents with the help of deterministic and stochastic operations, which constitutes the algorithm. The joint acknowledgment of deterministic and stochastic operations engages the algorithm in both exploration and exploitation simultaneously. The only deterministic process would not have been able to produce the optimized solutions due to limitations in combination determination and the high cost of the evaluation. The introduction of randomness in parameter generation or direction of movement or acceptance of position or event is the key to success for many meta-heuristic algorithms. The main reason for such success is opportunistic combination formations and escaping from the local minima similar to a situation for the unexplored areas and may be toward global optimization. This random factor has several forms in different aspects and has different operations in different algorithms. A typical example is better known as a mutation factor in evolutionary algorithms such as GA and differential evolution. Every nature-inspired computation can be regarded as a meta-heuristics in some form or the other. If the random features are not present in algorithms, then they would behave similar to a greedy algorithm or deterministic algorithms such as Prim's algorithm and Dijkstra algorithm. Two salient features of meta-heuristic algorithms are intensification and at the same time some diversification. It is another way of saying exploration and exploitation. Diversification means a deviation of mean that is away from the best results or global optimization region. Intensification is another way of reinventing the adjacency of the global best region for better positions. So, in the intensification phase, agents hover around the global best search space and try to detect better solution around, if possible, through variation of parameters. While in the diversification phase, it is ensured that the agents do not leave out of the unexplored search environment. An artificial bee colony (ABC) that consists of scouts and workers is a typical example of the combination of intensification and diversification. ABC is described in Chapter 4. So, briefly what we mean is "The specific objectives of developing modern meta-heuristic algorithms are to solve faster, to solve complex problems, and to obtain more robust methods." Meta-heuristic draw their potential and way of solving problems through the source of inspiration such as any natural phenomenon and in the 2 to 16 chapters we are going to discuss various kinds of such meta-heuristics and their wide range of application.

1.1.8.3 Hyper-Heuristics

Hyper-heuristics is an automated search scheme-based algorithm in which the operations are performed based on the previously learned experience. In other words, it is a cooperative approach of search and machine learning in which the search is driven by an experience that is gathered by learning models. In hyper-heuristics, the search is accompanied by an intelligent decision-making of the choice of parametric variations and operations in the search space. If the best solution is near the peak, in meta-heuristics we will move on exhaustively for new options, but in hyper-heuristics, the agents will only move if the better position is reached as the learned model only searches in the position where it has found better positions. Sometimes the main search process is accompanied by exhaustive local searches, which come into act only when the agents have reached a considerable stage. In particle swarm optimization (PSO), if the particle keeps on moving with the same speed then it is meta-heuristics, but if it lows down with the movement toward the peak, then it is hyper-heuristics. PSO is described in Chapter 2 and GA is described in Chapter 3. In addition, now at the best position, if we implement a GA-based PSO, then it is again hyper-heuristics, instead of just hovering around the search space randomly. The GA-based PSO hybrid has been proven to have a better convergence rate than the PSO itself.

1.1.8.4 Nature-Inspired Computation

Nature-inspired computation is the generalization of a bio-inspired computation and evolutionary computation where in each case the algorithms are derived from the natural phenomenon of the surrounding environment or nature and its habitats. The operations of the algorithms are a replica of what happens in nature and sometimes are mathematically modeled so that it can approximately represent such phenomenon. The nature-inspired algorithms have been very successful in the optimization of many NP-Hard problems,