A MODIFIED FLOWER POLLINATION ALGORITHM AND CARNIVOROUS PLANT ALGORITHM FOR SOLVING ENGINEERING OPTIMIZATION PROBLEM

ONG KOK MENG

UNIVERSITI TUN HUSSEIN ONN MALAYSIA

UNIVERSITI TUN HUSSEIN ONN MALAYSIA

STATUS CONFIRMATION FOR THESIS
DOCTOR OF PHILOSOPHY

A MODIFIED FLOWER POLLINATION ALGORITHM AND CARNIVOROUS PLANT ALGORITHM FOR SOLVING ENGINEERING OPTIMIZATION PROBLEMS

ACADEMIC SESSION: 2020/2021

I, **ONG KOK MENG,** agree to allow Thesis to be kept at the Library under the following terms:

1. This Thesis is the property of the Universiti Tun Hussein Onn Malaysia.
2. The library has the right to make copies for educational purposes only.
3. The library is allowed to make copies of this Thesis for educational exchange between higher educational institutions.
4. The library is allowed to make available full text access of the digital copy via the internet by Universiti Tun Hussein Onn Malaysia in downloadable format provided that the Thesis is not subject to an embargo. Should an embargo be in place, the digital copy will only be made available as set out above once the embargo has expired.
5. ** Please Mark (√)

☐ CONFIDENTIAL (Contains information of high security or of great importance to Malaysia as STIPULATED under the OFFICIAL SECRET ACT 1972) *Title and Abstract only*

☐ RESTRICTED (Contains restricted information as determined by the organization/institution where research was conducted)- *Title, Abstract and Introduction only*

☐ EMBARGO ______ until _______
(date) (date)

☑ FREE ACCESS

Approved by,

(WRITER'S SIGNATURE) (SUPERVISOR'S SIGNATURE)

Permanent Address:

KAMPUNG BEMBAN,
JALAN CHANGLOON,
06000 JITRA, KEDAH

Date : 16.3.2021 Date: 16.3.2021

NOTE: ** If this Thesis is classified as CONFIDENTIAL or RESTRICTED, please attach the letter from the relevant authority/organization stating reasons and duration for such classification.

This thesis has been examined on date …………14 December 2020…………….. and is sufficient in fulfilling the scope and quality for the purpose of awarding the Degree of Doctor of Philosophy.

Chairperson:

ASSOC. PROF. DR. IBRAHIM BIN MASOOD
Faculty of Mechanical and Manufacturing Engineering
Universiti Tun Hussein Onn Malaysia

Examiners

PROF. DR. MUSTAFA BIN MAMAT
Faculty of Informatics and Computing
Universiti Sultan Zainal Abidin

ASSOC. PROF. DR. SH. SALLEH BIN SH. AHMAD
Faculty of Mechanical and Manufacturing Engineering
Universiti Tun Hussein Onn Malaysia

A MODIFIED FLOWER POLLINATION ALGORITHM AND CARNIVOROUS PLANT ALGORITHM FOR SOLVING ENGINEERING OPTIMIZATION PROBLEM

ONG KOK MENG

A thesis submitted in fulfilment of the requirement for the award of the
Doctor of Philosophy in Mechanical Engineering

Faculty of Mechanical and Manufacturing Engineering
Universiti Tun Hussein Onn Malaysia

APRIL 2021

I hereby declare that the work in this project in my own except for quotations and summaries which have been duly acknowledged

Student	:	…………………………………………….
		ONG KOK MENG
Date	:	……16-3-2021…………………………….
Supervisor	:	…………………………………………….
		ASSOC. PROF. DR. ONG PAULINE
Co. Supervisor	:	…………………………………………….
		ASSOC. PROF. DR. SIA CHEE KIONG

To my dear supervisor,

Associate Professor Dr. Ong Pauline

To my dear co-supervisor,

Associate Professor Dr. Sia Chee Kiong

To my beloved parents,

Ong Hock Long and Lee Seow Yean

and to all my family and friends.

ACKNOWLEDGEMENT

First and foremost, I would like to express my deepest gratitude and appreciation to my beloved supervisor, Assoc. Prof. Dr. Ong Pauline, for all of her excellent guidance, responsible supervision, insightful teaching and constructive comment throughout this study. She gave me the golden opportunity to learn, modify and create metaheuristic algorithms to solve real-world applications. She also never gave up hope on me, always inspiring and motivating me to achieve greatness in publication and research work. I was greatly touched and impacted by her selflessness, humility and passion in guiding me throughout my PhD studies. I would like to convey the same gratitude and appreciation to my co-supervisor, Assoc. Prof. Dr. Sia Chee Kiong, for all of his informative teaching and encouragement for this study. He was a fatherly figure to me for always sharing useful mechanical knowledge with me. Both of these individuals had a huge influence on my life as I took them both as references and as role models for my life.

Next, special thanks must go to my father, Ong Hock Long, and my mother, Lee Seow Yean, who have given me unconditional love and support throughout the completion of my PhD studies. Both of them sacrificed a lot just to ensure that I could fully focus on my research work. I am profoundly touched and grateful for their patience and understanding of my condition in completion of this project, which involved many long hours when they had to handle family concerns on their own because their son's face was glued to a computer screen. I do also want to thank my sisters and brother, who are always supportive of me and who believe in me.

Lastly, a thousand thanks go to all my friends who never give up supporting me in every aspect of life. Once again, thank you very much, my friends, I will never forget all of your kindness.

ABSTRACT

Optimization in an essential element in mechanical engineering and has never been an easy task. Hence, using an effective optimiser to solve these problems with high complexity is important. In this study, two metaheuristic algorithms, namely, modified flower pollination algorithm (MFPA) and carnivorous plant algorithm (CPA), were proposed. Flower pollination algorithm (FPA) is a biomimicry optimisation algorithm inspired by natural pollination. Although FPA has shown better convergence than particle swarm optimisation and genetic algorithm in the pioneering study, improving the convergence characteristic of FPA still needs more work. To speed up the convergence, modifications of: (i) employing chaos theory in the initialisation of initial population to enhance the diversity of the initial population in the search space, (ii) replacing FPA's local search strategy with frog leaping algorithm to improve intensification, and (iii) integrating inertia weight into FPA's global search strategy to adjust the searching ability of the global strategy, were presented. CPA, on the other hand, was developed based on the inspiration from how carnivorous plants adapt to survive in harsh environments. Both MFPA and CPA were first evaluated using twenty-five well-known benchmark functions with different characteristics and seven Congress on Evolutionary Computation (CEC) 2017 test functions. Their convergence characteristic and computational efficiency were analysed and compared with eight widely used metaheuristic algorithms, with the superiority validated using the Wilcoxon signed-rank test. The applicability of MFPA and CPA were further examined on eighteen mechanical engineering design problems and two challenging real-world applications of controlling the orientation of a five-degrees-of-freedom robotic arm and moving-object tracking in a complicated environment. For the optimisation of classical benchmark functions, CPA was ranked first. It also obtained the first rank in CEC_{04} and CEC_{07} modern test functions. Both CPA and MFPA showed promising results on the mechanical engineering design problems. CPA improved over the particle swarm optimisation algorithm in terms of the best fitness value by 69.40-

95.99% in the optimisation of the robotic arm. Meanwhile, MFPA demonstrated a better tracking performance in the considered case studies by at least 52.99% better fitness function evaluation and fewer number of function evaluations as compared with the competitors.

ABSTRAK

Pengoptimuman adalah satu elemen penting dalam kejuruteraan mekanikal dan tidak pernah menjadi satu kerja yang mudah. Oleh itu, penggunaan suatu pengoptimum yang efektif untuk menyelesaikan masalah yang berkerumitan tinggi ini adalah mustahak. Dalam kajian ini, dua algoritma metaheuristik, iaitu algoritma pendebungaan bunga yang ditambahbaik (APBT) dan algoritma tumbuhan karnivor (ATK), telah dicadangkan. Algoritma pendebungaan bunga (APB) adalah algoritma pengoptimuman biomimikri yang diilhamkan oleh pendebungaan semula jadi. Walaupun APB telah menunjukkan penumpuan yang lebih baik daripada pengoptimuman kumpulan zarah dan algoritma genetik dalam kajian perintis, peningkatan ciri penumpuan APB masih memerlukan lebih banyak usaha. Untuk mempercepatkan penumpuan, pengubahsuaian (i) Menggunakan teori kekacauan dalam permulaan populasi awal untuk meningkatkan kepelbagaian populasi awal di ruang carian; (ii) Menggantikan strategi pencarian tempatan APB dengan algoritma lompatan katak untuk meningkatkan intensifikasi; dan (iii) mengintegrasikan berat inersia dengan strategi pencarian global APB untuk menyelaraskan kemampuan pencarian strategi global, telah dibentangkan. ATK, di sebaliknya, telah dibangunkan berdasarkan inspirasi daripada bagaimana tumbuhan karnivora menyesuaikan diri untuk bertahan hidup dalam suasana yang sukar. Kedua-dua APBT dan ATK pada mulanya dinilaikan pada dua puluh lima fungsi penanda aras terkenal dengan ciri-ciri yang berbeza dan tujuh fungsi ujian *Congress on Evolutionary Computation* (CEC) 2017. Ciri penumpuan dan kerumitan pengiraan mereka telah dianalisis dan dibandingkan dengan lapan algoritma metaheuristik yang digunakan secara meluas, dengan kelebihan yang disahkan dengan menggunakan ujian *Wilcoxon signed-rank*. Kebolehlaksanaan APBT dan ATK dikaji dengan lebih lanjut dalam lapan belas masalah rekabentuk kejuruteraan mekanikal dan dua aplikasi dunia sebenar yang mencabar, iaitu pengawalan orientasi lengan robot dengan lima darjah kebebasan dan pengesanan objek bergerak di dalam persekitaran yang rumit. Bagi pengoptimuman

fungsi ujian penandaaras klasik, ATK memperolehi kedudukan pertama. Ia juga mendapat kedudukan pertama dalam fungsi ujian moden CEC04 dan CEC07. Kedua-dua ATK dan APBT menunjukkan hasil yang meyakinkan di dalam masalah reka bentuk kejuruteraan mekanikal. Berbanding dengan algoritma pengoptimuman partikel berkelompok, ATK bertambah baik dari segi nilai kecergasan terbaik sebanyak 69.40-95.99% dalam pengoptimuman lengan robot. Sementara itu, APBT menunjukkan prestasi pengesanan yang lebih baik dalam kajian kes yang dipertimbangkan, dengan sekurang-kurangnya 52.99% penilaian fungsi kecergasan yang lebih baik dan jumlah penilaian fungsi yang lebih rendah berbanding dengan pesaing.

CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS AND ABBREVIATIONS

ABC	-	Artificial bee colony
ACO	-	Ant colony optimization
ADSO	-	Adaptive discrete swarm optimization
APSO	-	Accelerated particle swarm optimization
ARSM	-	Adaptive response surface method
ASO	-	Atom search optimization
BAT	-	Bat algorithm
BBBC	-	Big Bang – Big Crunch
BBO	-	Biogeography-based optimization
CEC	-	Congress on evolutionary computation
CoBiDE	-	Covariance matrix learning and bimodal distribution differential evolution
CPA	-	Carnivorous plant algorithm
Cr	-	Crossover probability
CSA	-	Cuckoo search algorithm
CSB	-	Center-seeking bias
C_T	-	Production cost
d	-	Dimension of search space
DC	-	Deflection constraint
DE	-	Differential evolution
DEA	-	Dolphin echolocation algorithm
DeepHGSA	-	Hybrid gravitational search algorithm with a deep convolutional feature
DH	-	Denavit-Hartenberg
DOA	-	Dynastic optimization algorithm
DOF	-	Degree of freedom
DSA	-	Differential search algorithm
FA	-	Firefly algorithm

FK	-	Forward kinematics
FPA	-	Flower pollination algorithm
GA	-	Genetic algorithm
GCA	-	Generalized convex approximation
GOA	-	Grasshopper optimization algorithm
GP	-	Geometric programming
GSA	-	Gravitational search algorithm
GTOA	-	Group teaching optimization algorithm
GWO	-	Grey wolf optimizer
HBA	-	Honey bee algorithm
HGSA	-	Hybrid gravitational search algorithm
HHO	-	Harris hawks optimization
HS	-	Harmony search
HSV	-	Hue saturation value
IAPSO	-	Improved accelerated particle swarm optimization
ICA	-	Imperialist competitive algorithm
IK	-	Inverse kinematics
ILS	-	Iterated local search
IPHS	-	Improving proposed harmony search
IRB	-	Initialization-region bias
it	-	Iterations within each memeplex
IWO	-	Invasive weed optimization
J	-	Number of inequality constraints
K	-	Number of equality constraints
KF	-	Kalman filter
KKT	-	Karush-Kuhn-Tucker
L	-	Lévy flight
Lb	-	Lower boundary
LCA	-	League championship algorithm
LSHADE	-	Success-history based adaptive differential evolution with linear population size reduction
m	-	Number of memeplexes
max_iter	-	Number of maximum iteration
MBA	-	Mine blast algorithm

MBFA	-	Mouth brooding fish algorithm
MCMC	-	Markov chain monte carlo
MGbSA	-	Modified galaxy-based search algorithm
MGWO	-	Modified grey wolf algorithm
MGPEA	-	Multivariable grey prediction evolution algorithm
MOA	-	Moth-flame optimization algorithm
MPA	-	Marine predators algorithm
MS	-	Mean-shift
MVO	-	Multi-verse optimizer
MFPA	-	Modified flower pollination algorithm
MMA	-	Method of moving asymptotes
n	-	Size of population
nCPlant	-	Number of carnivorous plants
NFL	-	No free lunch
NHS	-	New harmony search
NNA	-	Neural network algorithm
NOFE	-	Number of function evaluation
nPrey	-	Number of preys
NSGA-II	-	Non-dominated sorting genetic
p	-	Switching probability
pa	-	Discovery rate
PC	-	Power constraint
PD	-	Fuzzy proportional-derivative
PF	-	Particle filter
PGSA	-	Parallel genetic simulated annealing
PHS	-	Proposed harmony search
PSO	-	Particle swarm optimization
r	-	Pulse rate
SFO	-	Sailfish optimizer
SA	-	Simulated annealing
SC	-	Strength constraint
SCA	-	Sine cosine algorithm
SD	-	Standard deviation
SFLA	-	Shuffled frog-leaping algorithm

SS	-	Squirrel search algorithm
SSA	-	Salp swarm algorithm
TGA	-	Tree growth algorithm
TLBO	-	Teaching learning based optimization
TLCS	-	Teaching leaning based cuckoo search
T_{pr}	-	Total production time
TS	-	Tabu search
TSA	-	Tunicate swarm algorithm
Ub	-	Upper boundary
VBA	-	Virtual bee algorithm
WCA	-	Water cycle algorithm
WOA	-	Whale optimization algorithm
WRP	-	Production rate
WSA	-	Water strider algorithm
γ	-	Absorption coefficient
β_0	-	Attractiveness coefficient
$f\left(\vec{x}\right)$	-	Function of the single-objective optimization problem
$g_j\left(\vec{x}\right)$	-	Inequality constraint
$h_k\left(\vec{x}\right)$	-	Equality constraint
θ	-	Orientation vector
${}^{0}T_{n}$	-	Homogeneous matrix

LIST OF APPENDICES

CHAPTER 1

INTRODUCTION

In this chapter, the definition of optimisation is first introduced, followed by a brief explanation of mathematical optimisation methods and stochastic methods. Subsequently, the problem statement of this study is presented, where the solutions to resolve the limitations of the flower pollination algorithm (FPA) are suggested and a new bio-inspired algorithm, specifically, carnivorous plant algorithm (CPA) is proposed. The objective, scope and contributions of this study are then stated. Lastly, the organisation of the thesis is presented.

1.1. Research Background

Optimization attempts to find the best combination of the design parameters in a given problem under some definite constraints, such that the objective of that particular problem can be met [1]. For single-objective optimisation problems, the mathematical model can be written as:

$$\begin{aligned} &\text{Minimize/Maximize: } f\left(\vec{x}\right),\ \vec{x} = x_1,\ x_2,\ \ldots,\ x_{n-1},\ x_n \\ &\text{Subject to: } g_j\left(\vec{x}\right) \geq 0,\ j = 1,\ 2,\ \ldots,\ J \\ &\qquad h_k\left(\vec{x}\right) = 0,\ k = 1, 2,\ \ldots,\ K \\ &\qquad Lb_i \leq x_i \leq Ub_i,\ i = 1,\ 2,\ \ldots,\ n \end{aligned} \tag{1.1}$$

where $f\left(\vec{x}\right)$ is the function of the single-objective optimization problem, $\vec{x} = x_1,\ x_2,\ \ldots,\ x_{n-1},\ x_n$ is the design variable, n is the number of design variables, $g_j\left(\vec{x}\right)$ is the inequality constraint, J is the number of inequality constraints, $h_k\left(\vec{x}\right)$

is the equality constraint, K is the number of equality constraints, Lb_i is the lower boundary of the i^{th} input and Ub_i is the upper boundary of the i^{th} input.

The importance of optimization has roots in multitudinous areas; particularly in mechanical engineering, optimization is often related to system performance improvement, cost reduction, process streamlining, efficiency and reliability. An example of optimization in mechanical engineering would be to find the optimal combination of depth of cut, feed per tooth and cutting speed such that the total production time for milling a workpiece is minimized. A slight change in one parameter will affect the cost and production time of the milling process as well as the product quality. Conventionally, the parameter setting relies on the operator's experience, which is even a challenging task for an experienced operator. Thus, a successful optimisation of these parameters is critical so that the manufacturer can increase production in limited time to survive in a competitive market.

Optimization, however, has never been an easy task due to the nonlinearity feature of the objective function. The design parameters are discontinuous and some of the design parameters are only considered discrete value [2]. The increase in the number of design variables and constraints makes the optimisation even more complicated and therefore more computationally costly [3]. Before the emergence of the stochastic methods, deterministic approaches, such as hill-climbing, Simplex method, Bundle method and Newton-Raphson, are commonly used to solve optimisation problems [4, 5]. Despite these techniques continuing to receive widespread attention in various domains, challenges still remain of local optima entrapment if given a poorly defined starting point [6]. Furthermore, these methods are usually slow in convergence and incur a high number of function evaluations (NOFE) to search for feasible solutions. In addition, deterministic algorithms, especially the gradient-based approach, are ineffective in solving non-differentiable/discontinuous problems [7] or when the objective function has sharp or multiple peaks. Thus, these methods have limited applicability in solving complex real-world problems.

The rise of the stochastic method as a promising alternative to the deterministic approach indeed lies with its inherent randomness and gradient-free calculation. The use of randomness can be found in different components of a stochastic optimization method, such as the crossover and mutation operators in the popular genetic algorithm (GA) and the hill-climbing method with random restart, allowing the stochastic method to escape from local optima. Moreover, moving the solutions towards the global optima is based on the evaluation of the objective function and a set of rules. This is in contrast with the mathematical optimisation method, which requires the calculation of gradient by the derivative of the objective function. Given this superiority, rapid progress in the research on the stochastic method has spawned the development of a broad range of optimisation solutions, falling into two categories: heuristic and metaheuristic [5].

Bio-inspired and population-based metaheuristic algorithms are gaining steam today and are often implemented in different domains [8]. This is because metaheuristic algorithms are simple to use, only needing the information of fitness function during optimisation. Additionally, metaheuristic algorithms, which use a set of solutions with probabilistic rules in finding the global optima in the search space, also improve the success rate of optimisation. In metaheuristic algorithms, two major elements, namely, exploration and exploitation, play an important role during optimisation. Exploration enables the algorithm to explore the promising areas in the search space and also to escape from the local optima [8, 9]. Meanwhile, exploitation enables the algorithm to obtain a highly accurate solution from the promising areas [7]. In this regard, an algorithm with a good combination of these two elements will prevent itself from premature convergence in the early phase of the optimisation process and quickly converge towards the global optima at the end.

Most metaheuristic algorithms are inspired by nature, such as GA [10] — the most popular metaheuristic algorithm—which imitates the biological evolution of mutation, recombination and selection of biological systems. This has opened a new way of thinking for researchers to link nature to the mathematical computational skill in solving challenging optimisation problems. Since then, a lot of metaheuristic algorithms have been developed and the list is still growing, which can be referred to in [11].

These metaheuristic algorithms can be categorised based on their inspiration source as follows:

a) Evolutionary techniques: GA [10] and differential evolution (DE) [12]. These algorithms are derived from biological evolution, such as mutation, crossover, selection and reproduction.

b) Animal-based techniques: artificial bee colony (ABC) [13], particle swarm optimization (PSO) [14], cuckoo search algorithm (CSA) [15], bat algorithm

(BAT) [16], squirrel search algorithm (SS) [17] and sailfish optimizer (SFO) [9]. Such algorithms are inspired by the behaviour of animals, for instance, bee, bird, bat, squirrel and fish.

c) Plant-based techniques: invasive weed optimization (IWO) [18] and flower pollination algorithm (FPA) [19]. These algorithms imitate plants' behaviour, where IWO mimics the process of weed invasion, while FPA simulates the pollination process of flowers.

d) Human activity-based techniques: harmony search (HS) [20], teaching learning based optimization (TLBO) [21], league championship algorithm (LCA) [22] and imperialist competitive algorithm (ICA) [23]. These kinds of algorithms are derived from human activities, such as guitar tuning, teaching and learning method, the championship process of sports leagues and the colonisation of an empire.

e) Physics-based techniques: gravitational search algorithm (GSA) [24], water cycle algorithm (WCA) [25], Big Bang–Big Crunch (BBBC) [26] and multi-verse optimiser (MVO) [27]. These algorithms mimic the phenomena on earth and also in the universe, for example, the law of gravity, the flow of rivers and streams towards the sea, the Big Bang and Big Crunch theories and the concepts of white holes, black holes and wormholes.

1.2. Problem Statement

Optimisation is an essential and indispensable element in almost every engineering vertical. Minimising the total mass of a speed reducer, maximising the dynamic load-carrying capacity of a rolling bearing and minimising the total cost of a welded beam are examples of optimisation problems in the mechanical engineering field. Hence, liable optimisers are often needed to solve those engineering problems with high complexity. Most real-world engineering problems are constrained optimisation problems. To solve these problems, there are two types of optimisers available: mathematical programming methods and metaheuristic algorithms [28]. Mathematical programming methods, also known as deterministic approaches, are feasible only if the derivatives of the underlying problems are available. These methods, too, are highly affected by the number of local optima and the selection of the initial points. In addition, handling problems with discontinuities still pose a challenge for deterministic approaches. Metaheuristic algorithms, on the other hand, are able to overcome the shortcomings of mathematical programming methods, as shown in the literature [29-32].

Despite the consistently promising performance of metaheuristic algorithms, the increasing complexity of real-world problems has prompted the search for better solutions. For performance enhancement, the studies on metaheuristic algorithms can be divided into three main directions: (i) improving existing metaheuristic algorithms, (ii) hybridising different metaheuristic algorithms, and (iii) proposing new metaheuristic algorithms. The intent of this study focuses on all these three directions.

For the first and second direction, FPA was chosen to be modified due to the preliminary results showing that FPA is far better than GA and PSO [19]. However, scrutinising the literature showed that FPA utilises a typical random technique to initialise the population, contributing to low population diversity, and is prone to poor and premature convergence [33]. The searching process may, therefore, start in an unsuitable search space, which further affects the competence of the algorithm, particularly with regard to the convergence rate. In addition, the local search approach of FPA lacks knowledge sharing among the good solutions and, in turn, may require more function evaluations for convergence. Random walks throughout FPA's exploration phase may also impede the convergence. This is because, at the end of the optimisation system, the ongoing exploration significantly increases the search time. The ideal searching operation in an optimisation system should initially concentrate more on exploration and progressively move towards exploitation at the end. Realising this shortcoming, the formulation of modified flower pollination algorithm (MFPA) is studied in this work, where the following modifications were included: (i) incorporation of the circle map in population initialisation to enhance the diversity of the population, (ii) integration of the frog leaping algorithm to improve information sharing among the good solutions, and (iii) using inertia weight in the search process to balance exploration and exploitation. The performance of the developed MFPA was evaluated using different test problems.

The increasing complexity of real-world problems has prompted the development of more metaheuristic optimisation approaches. One might question the need to have a new metaheuristic algorithm since there are many existing metaheuristic algorithms out there. A positive answer to this question is because of the No Free

Lunch (NFL) Theorem, proposed by Wolpert and Macready [34]. According to the NFL theorem, if algorithm A performs better than algorithm B in the specific problem X, it is not necessary for algorithm A to outperform algorithm B in the specific problem Y. The performances of all algorithms are equally well on average. Concisely, there is no universal optimisation procedure that works perfectly for all optimisation problems and, thus, the continuing flourish of the diversity of optimisation algorithms is encouraged. Hence, a new population-based metaheuristic algorithm, namely, CPA, is proposed, corresponding to the third direction. CPA imitates how carnivorous plants adapt to survive in harsh environments, specifically, hunting insects for its food and pollinating for reproduction. Although MFPA and CPA are in the same class, which is plant-inspired algorithm, the CPA proposed in this work is completely different as compared with former works in terms of biological inspiration, mathematical formulation for solutions updating and real-world applications. To the best of the author's knowledge, an algorithm inspired by the survival skills of carnivorous plants has not yet been studied in the literature. It will be shown that the proposed CPA can successfully address the issues of high-dimensional design variables, the existence of various constraints and the search space with many local optima without having a structural bias in its searching operator.

1.3. Objective

The aim of this study gears towards an modified FPA and a new optimisation algorithm, specifically CPA, for engineering optimisation problems. The objectives of this study are as follows:

1) To formulate an MFPA, incorporating the frog leaping local search, chaos theory and inertia weight.
2) To develop an efficacious novel CPA inspired by the survival skills of carnivorous plants.
3) To assess the beneficial impact of MFPA and CPA on the optimisation of test functions, benchmark mechanical engineering optimisation problems and real-world problems.
4) To compare the optimisation performances of MFPA and CPA with other approaches available in the literature.

1.4. Scope of Study

The scope of this study is as follows:

(i) The optimisation problems were limited to single-objective optimisation problems.

(ii) The performances of MFPA and CPA were evaluated through assessment in thirty-two benchmark test functions and eighteen mechanical engineering optimisation problems.

(iii) Two real-world applications, which were controlling the posture of a 5-degrees-of-freedom (DOF) robotic arm for gripping a target object precisely without colliding with any obstacle and tracking a moving object on visible image sequences in a complicated environment, were selected.

(iv) The results obtained by MFPA and CPA in solving classical benchmark test functions were compared with FPA, Improved PSO, DE, CSA, BAT, firefly algorithm (FA), salp swarm algorithm (SSA) and GA.

(v) The results obtained by MFPA and CPA in solving Congress on Evolutionary Computation (CEC) test functions were compared with FPA, Improved PSO, DE, CSA, BAT, FA, SSA, GA and success-history-based adaptive differential evolution with linear population size reduction (LSHADE).

(vi) The results obtained by MFPA and CPA in the first real-world problem, namely, posture control of a 5-DOF robotic arm, were compared with FPA, DE, PSO and GA.

(vii) The results obtained by MFPA and CPA in the second real-world problem, specifically, moving-object tracking in a complicated environment, were compared with FPA, DE, PSO, GA and particle filter (PF).

(viii) The NOFE required by each algorithm was compared for optimising benchmark test functions.

(ix) The best fitness value obtained by each algorithm was compared in solving mechanical engineering optimisation problems.

(x) The derivation of MFPA and CPA were performed using the MATLAB R2016a software.

1.5. Contributions of the Study

In this study, significant contributions to the field of optimization using bio-inspired metaheuristic algorithms have been made. The main contributions and its novelty are as follows:

1) A new variant of FPA, specifically MFPA, has been developed. A considerable contribution is made in terms of the diversity of the initial population, the enhancement of the local search ability and the balancing of exploration and exploitation. The proposed MFPA demonstrates higher solution accuracy and better convergence characteristic than the classical FPA in solving global optimisation problems.
2) A novel bio-inspired CPA is proposed and presented for the first time in this study. The close mimicking of how carnivorous plants adapt themselves to circumstances that are constantly changing is mathematically formulated. The proposed CPA can effectively deal with optimisation problems with high-dimensional design space that involves 30 design variables, the presence of different constraints and the high complexity landscape of the search space. In addition, the proposed CPA shows no structural bias from the results of optimising seven CEC 2017 test functions.
3) The effectiveness of the proposed MFPA and CPA were tested using the benchmark test functions, benchmark mechanical engineering optimisation problems and real-world applications. Both proposed algorithms are competent algorithms in terms of faster convergence rate and higher solution accuracy.

1.6. Thesis Organization

Chapter 2 begins with the discussion on the preliminaries and the definition of the optimisation problem. The history of optimisation techniques is touched on, followed by the exploration on the literature of metaheuristic algorithms. The procedure of FPA and its applications in the real world is given. The limitations of FPA are reviewed and the need to improve FPA from various aspects is then presented. Lastly, a review of two real-world applications, specifically, the posture control of a 5-DOF robotic arm and dynamic object tracking in a complex environment, is provided.

The formulation of the proposed MFPA and CPA, which is the core of this thesis, is presented in Chapter 3. The proposed MFPA is introduced first in the chapter, followed by the discussion on each component used to improve FPA, which are chaos theory, frog leaping local search and inertia weight. The detailed procedure of MFPA is then explained. Next, the inspiration, mathematical model and flowchart of the proposed CPA are provided. Lastly, the derivation of the mathematical models for the two real-world applications are given.

The performance assessment of MFPA and CPA on solving thirty-two benchmark test functions, eighteen mechanical engineering design problems and two real-world problems are given in Chapter 4. The benchmark test functions are divided into two categories, which are classical benchmark test functions and CEC test functions. The classical benchmark test functions are used to evaluate the exploitation and exploration of the proposed algorithms, while CEC test functions are used to examine whether the proposed algorithms have a structural bias. It will be shown that both proposed algorithms have faster convergence rates and higher solution accuracy as compared with other optimisers.

The conclusions of this research work are provided in Chapter 5. Lastly, the recommendations for further research are given.

CHAPTER 2

LITERATURE REVIEW

2.1. Introduction

The optimisation problem is the primary issue in the optimisation process. Thus, this chapter begins with the preliminaries of the optimisation problem. Then, the classical optimisation methods, which are typically used to solve optimisation problems, are discussed, followed by their disadvantages. The advantages of metaheuristic algorithms are presented and several examples of metaheuristic algorithms are provided, which are designed to resolve the shortcomings of the classical optimisation methods. Subsequently, the details of FPA, which is the selected algorithm for modification in this study, are thoroughly explained, and followed by the discussion of the background of CPA. Lastly, the review of the selected first and second challenging real-world applications - the controlling of robot motion and the tracking of the moving object - are presented.

2.2. Preliminaries of Optimization Problem

The optimisation problem can be categorised as a single-objective optimisation problem and a multi-objective optimisation problem. The former has only one objective, while the latter has more than one objective. Dealing with a multi-objective optimisation problem requires special mechanisms, namely, relational operators such as Pareto optimal dominance, which is the core operator [35]. However, the work studied here focuses on single-objective optimisation problems; interested readers may refer to the work by Zhou *et al.* [36] for more information on multi-objective optimisation problems.

The problem in optimisation is defined by design parameters, which are real numbers (design variables) that have to be determined to obtain the minimum or maximum of the fitness value [37]. These decision variables are known as inputs. Examples of input are the number of teeth in the gear, the face width of the gear, the number of coils in the spring and material of the gear [37]. There are three types of inputs: continuous, discrete and integer. The continuous type of input, which is commonly found in many optimisation problems, is free to assume any value. The discrete type of input, on the other hand, is only assigned to a particular range of value; for instance, the diameter of a screw can only be chosen from a set of standard size. Lastly, the integer type of input considers integer values only, such as the number of threads in a screw and the number of teeth in a gear [37].

Optimisation problems with and without constraint are known as constrained and unconstrained problems, respectively. For constrained optimisation problems, the constraint can be categorised as inequality constraint and equality constraint. The former is represented by $g_j\left(\vec{x}\right)$, while the latter is denoted by $h_k\left(\vec{x}\right)$, as explained in Section 1.1. To differentiate them, an optimisation problem with two inputs and constraints constructed in a graph is illustrated in Figure 2.1. An optimisation problem with constraints increases the difficulty level for an algorithm to solve it, since the solution provided by the algorithm cannot violate the constrained areas. Some algorithms might be good in solving the unconstrained problem but are inefficient in solving the constrained problem. Therefore, the proposed new algorithm should be able to handle both instead of solving the unconstrained problem alone.

In addition, the landscape of the objective function can classify optimisation problems into unimodal and multimodal problems. Since a 3D graph can be constructed from a single-objective problem with two inputs, a single peak shown in the graph of the nonlinear case is known as a unimodal problem. In contrast, when more than one peak appear in the graph, it is considered a multimodal problem. The global optimum of the multimodal problem is hard to find due to many local optima, where the algorithm may get trapped. To visualise them, a 3D graph of the unimodal and multimodal problems with two inputs are generated, as shown in Figure 2.2.

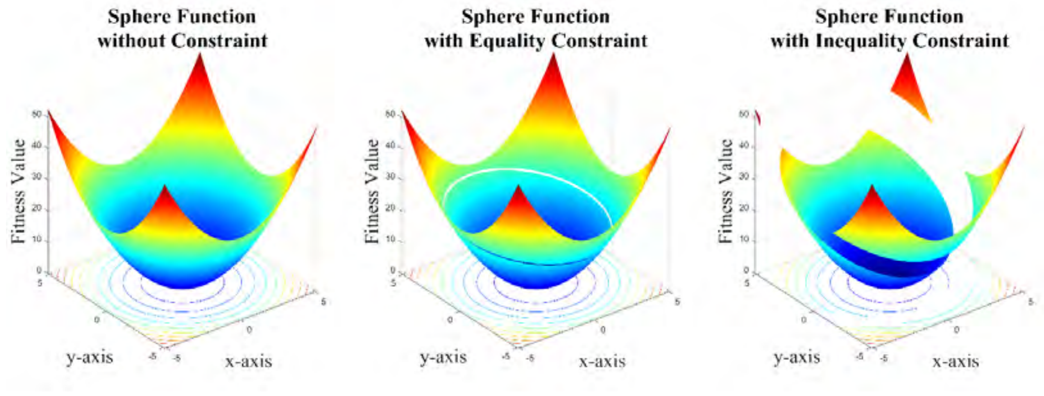

Figure 2.1. Sphere Function with and without Constraint

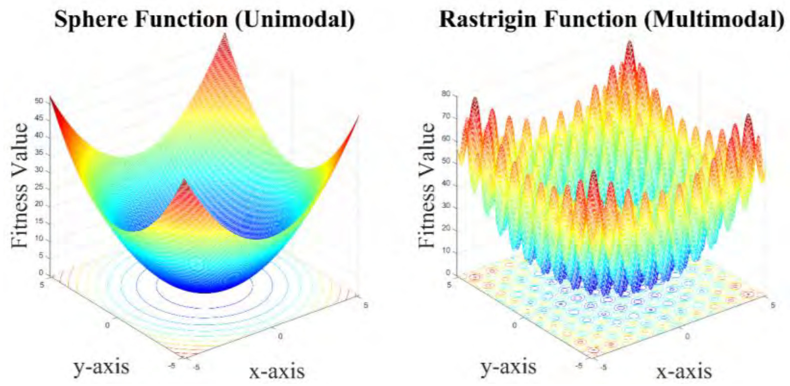

Figure 2.2. Example of unimodal and multimodal functions

A newly proposed or modified metaheuristic algorithm should be able to address the challenges concerning high-dimensional design variables, the existence of various constraints and the search space with many local optima. It shall be able to search for the optimal solution within the shortest period of time despite a large number of design variables to be optimised. In addition, the increasing optimisation difficulty due to the modelling constraints shall not impede the algorithm from reaching the optimal solutions with no violated constraints. Most importantly, a successful search mechanism shall not be prone to stagnation due to the existence of the local optima as in the multimodal function.

2.3. Classical (Deterministic) Optimization Methods

Most classical optimisation methods are deterministic. Since no degree of randomness is involved in the deterministic technique, it always outputs the same optimal solution, if given the same starting point. This technique is considered a gradient-based algorithm if it uses gradient information to solve the optimisation problem. Examples of such technique are Newton's method and the steepest descent method. Meanwhile, a deterministic technique that does not use derivative information is known as a gradient-free algorithm. This technique, such as the Nelder-Mead method, is required when the problem is unsolvable due to the non-differentiable characteristic of the objective function. In addition, linear programming and nonlinear programming are also categorised as deterministic techniques. These techniques, on the other hand, are used to solve constrained optimisation problems. The methods are briefly explained in the following section.

2.3.1. Steepest Descent Method

The steepest descent method is gradient-based optimisation algorithm for searching the optimal solution of a continuous function [38]. This method iteratively searches for the neighbourhood point that has the lowest possible value from the direction of the negative gradient of the current point. The formula for the steepest descent method is shown as follows:

$$x_{n+1} = x_n - \alpha^{(n)} f'(x_n) \tag{2.1}$$

where n is the current iteration, x_{n+1} is the neighbourhood point, x_n is the current point, $\alpha^{(n)}$ is the step size with a single real-value in current iteration and $f'(x_n)$ is the gradient of the current point on a continuous function $f(x)$. When the current point x_n, which is either the initial point or is moved from the previous point through Equation (2.3), is identified, the gradient of the current point $f'(x_n)$ can be determined. The neighbourhood point x_{n+1} with a single unknown variable $\alpha^{(n)}$ is then substituted into the continuous function and becomes $f(x_{n+1}) = f(\alpha^{(n)})$. The step size $\alpha^{(n)}$ is searched such that $f(\alpha^{(n)})$ is at the minimum. Since it has become a new optimisation problem with a single unknown variable $\alpha^{(n)}$, techniques such as Newton's method can be used to identify $\alpha^{(n)}$. Thus, step size $\alpha^{(n)}$ and gradient

$f'(x_n)$ are calculated at every iteration. A good guess on the initial point, which is near to the optimal solution, is useful for accelerating the optimisation process.

2.3.2. Newton's Method

Newton's method is a numerical method used to find the roots of continuous function $g(x)$ [39]. However, such a method can be modified to become an optimization method since optimization in calculus means identifying the root of the first derivative $f'(x)$.

The formula for Newton's method is shown as follows:

$$x_{n+1} = x_n - \frac{g(x_n)}{g'(x_n)} \tag{2.2}$$

where $g(x)$ is a continuous function, $g'(x)$ is the first derivative of the function and x_n is a real variable x on n iteration. The roots of the continuous function can be iteratively obtained using Equation (2.2). To transform Newton's method into an optimisation method, let $g(x) = f'(x)$ and $g'(x) = f''(x)$; thus, Newton's method in solving the optimisation problem by substituting everything together is presented as follows:

$$x_{n+1} = x_n - \frac{f'(x_n)}{f''(x_n)} \tag{2.3}$$

2.3.3. Nelder-Mead Method

In most real-world problems, the objective function is non-differentiable. Thus, the gradient-based algorithm is impracticable to optimise such problems. The Nelder-Mead method, which was developed by John Nelder and Rofer Mead in 1965 [40], was invented to overcome this shortcoming. This technique uses the flexibility of a geometrical shape, also known as a simplex, to search for the optimal solution. The

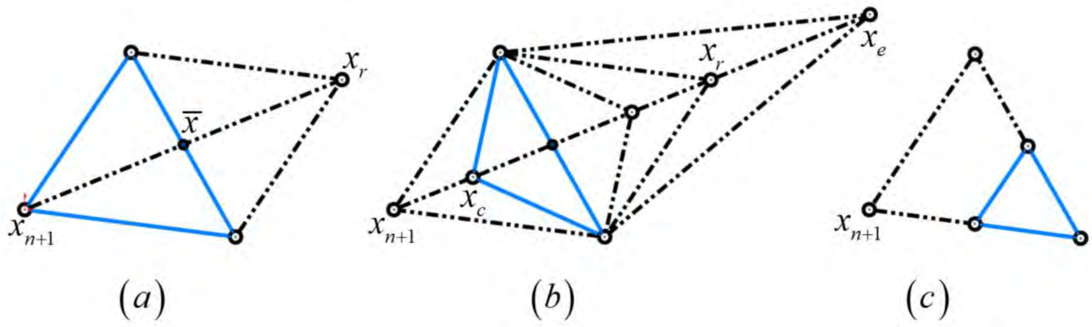

Figure 2.3. The transformation of the constructed simplex: (a) reflection, (b) expansion or contraction along the line of reflection and (c) shrink contraction

constructed simplex is reshaped at each iteration through the process of reflection, expansion, contraction and shrink contraction, as illustrated in Figure 2.3.

The Nelder-Mead method begins with the initialisation of *n+1* solutions on the search domain. The fitness values of the solutions are evaluated and sorted in ascending order (for minimisation), as in Equation (2.4).

$$f(x_1) \le f(x_2) \le \ldots \le f(x_{n+1}) \tag{2.4}$$

The *n+1* points are rearranged corresponding to their fitness value. Then, the centroid $\bar{x}$ of all solutions, excluding x_{n+1} (the worst solution), is calculated using Equation (2.5).

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i \tag{2.5}$$

The reflection of the worst solution x_{n+1} is identified through Equation (2.6).

$$x_r = \bar{x} + \alpha(\bar{x} - x_{n+1}), \qquad \alpha > 0 \tag{2.6}$$

where x_r is a reflected solution and $\alpha = 1$ is usually used. Figure 2.3(a) shows how the reflection looks like in a two-dimensional problem.

To update the worst solution, there are three possibilities when it is being compared with the reflected solution.

1. If $f(x_1) \le f(x_r) < f(x_{n+1})$, the worst solution x_{n+1} is replaced by the reflected solution x_r.

2. If $f(x_r) < f(x_1)$, it means the best solution has been improved. Thus, a bold move is taken by expanding the simplex, as shown in Figure 2.3(b), to determine whether the best solution has further improvement. The expansion formula is shown as follows:

$$x_e = x_r + \beta(x_r - \bar{x}) \tag{2.7}$$

where x_e is an expanded solution and $\beta = 2$ is frequently used. If $f(x_e) < f(x_r)$, the worst solution x_{n+1} is replaced by the expanded solution x_e. Otherwise, the worst solution x_{n+1} is replaced by the reflected solution x_r.

3. If $f(x_r) > f(x_1)$, it means there is no improvement. Thus, the size of the simplex is reduced through contraction, as illustrated in Figure 2.3(b). The contraction formula is represented as follows:

$$x_c = x_{n+1} + \gamma(\bar{x} - x_{n+1}) \tag{2.8}$$

where x_c is a contracted solution and $\gamma = 0.5$ is often used [5]. If $f(x_c) < f(x_{n+1})$, the worst solution x_{n+1} is replaced by the contracted solution x_c. Else, the size of the simplex is shrunk towards the best solution, as demonstrated in Figure 2.3(c). The shrink contraction formula is expressed as follows:

$$x_i = x_1 + \delta(x_i - x_1), \qquad i = 2,3,\ldots,n+1 \tag{2.9}$$

where x_i is a shrunk solution of whole solutions except the best solution and $\delta = 0.5$ is regularly used.

After that, the fitness values of the solutions are rearranged again in ascending order, and the process starts over until the termination condition is fulfilled.

2.3.4. Linear Programming: Graphical Method

In Section 2.3.1 to Section 2.3.3, the discussed techniques are only able to deal with the unconstrained optimisation problem. To handle the constrained optimisation problem, linear programming is developed to optimise a linear objective function, which is subjected to linear equality constraints and linear inequality constraints [5]. In this section, a linear programming using the graphical method for solving a linear problem is discussed.

Consider a linear problem, where the search for the best solution (x_1, x_2) is attempted to maximise its fitness value. The mathematical model of the linear problem is presented as:

$$\begin{aligned} \text{Maximize: } & \mathrm{P}(x_1, x_2) = \alpha x_1 + \beta x_2 \\ \text{Subject to: } & x_1 + x_2 \leq n, \\ & 0 \leq x_1 \leq n_1, \\ & 0 \leq x_2 \leq n_2. \end{aligned} \tag{2.10}$$

where the first, second and third inequality constraints are demonstrated with line BC, line CD and line AB, respectively, as shown in Figure 2.4. With these constraints, the feasible solutions lie within the polygon 0ABCD. Since the objective of this problem is to maximise function $\mathrm{P}(x_1, x_2)$, the optimal solution is located at point B, as it is the maximum region at which the objective line (dashed line) can reach.

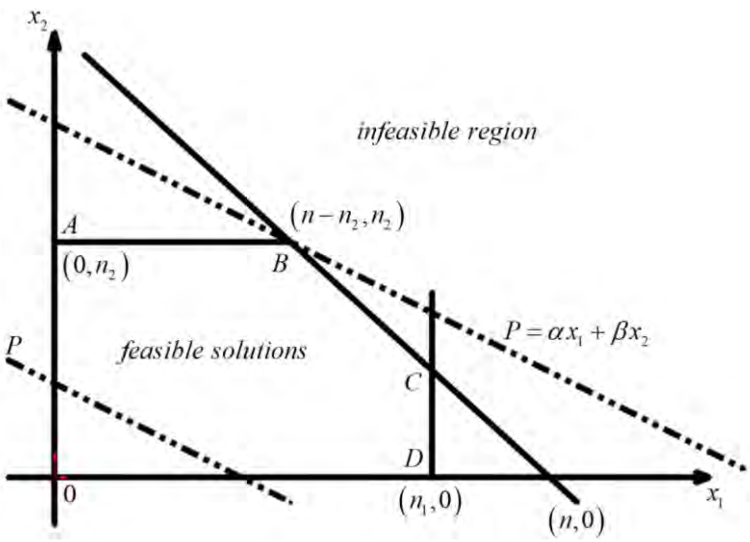

Figure 2.4. Linear Programming by Graphical Method [5]

2.3.5. Nonlinear Programming – Karush-Kuhn-Tucker Conditions

There are a lot of nonlinear programming problems in real-world applications. Most of them involve nonlinear objective function subjected to nonlinear equality and nonlinear inequality constraints. Thus, Karush-Kuhn-Tucker (KKT) conditions are invented as a solution for nonlinear programming optimization problems [41].

Consider Equation (1.1) as a nonlinear optimization problem. The formulas that involve KKT conditions are as follows:

$$\nabla L(\vec{x}) = \nabla f(\vec{x}) + \sum_{i=1}^{K} \lambda_i \nabla h_i(\vec{x}) + \sum_{j=1}^{J} \mu_j \nabla g_j(\vec{x}) = 0 \tag{2.11}$$

and

$$g_j(\vec{x}) \le 0, \qquad \mu_j g_j(\vec{x}) = 0, \quad (j = 1, 2, \ldots, J) \tag{2.12}$$

where

$$\mu_j \ge 0, \quad (j = 1, 2, \ldots, J) \tag{2.13}$$

The constants $\vec{\lambda}$ and $\vec{\mu}$ must satisfy the following condition of

$$\sum_{i=1}^{K} |\lambda_i| + \sum_{j=1}^{J} \mu_j \ge 0 \tag{2.14}$$

The optimal solution can be found by solving Equation (2.11) and Equation (2.12) through algebra.

2.3.6. Limitations of Deterministic Optimization

Although classical optimisation methods can find the true optimum, obtaining the first derivative for the complex functions is a hard and tedious process [42]. Furthermore, these methods are infeasible when the objective function is non-differentiable [43]. Even though there are gradient-free algorithms, such as the Nelder-Mead method, the algorithms might be getting trapped in the local optima due to no randomisation (exploration) in the searching mechanism [7, 44]. Thus, solving a complex real-world problem with many local optima is hard for classical optimisation methods. In addition, these methods are problem-specific because different methods target different types of optimisation problems [44]. Moreover, classical optimisation methods do not guarantee finding the global optimum, as it depends on the initial point [45]. Therefore, the development of metaheuristic optimisation algorithms is to overcome the drawbacks of classical optimisation methods.

Metaheuristic optimisation algorithms have gained popularity over the past decade due to the simplicity, gradient-free mechanism, local optima avoidance and flexibility of these algorithms [46]. A metaheuristic optimiser is simple to implement. It benefits from the simple natural behaviour concept, which can be incorporated easily as different operators in the searching mechanism [47]. In addition, metaheuristic optimisers optimise the problems stochastically without knowing the derivative information of the problems. Thus, such optimisers are good options for solving real-world problems with unknown derivative information [35]. In contrast to classical approaches, the stochastic procedure in metaheuristic optimisers helps the algorithms to escape from the local optima by exploring the search space [7]. Hence, they can effectively deal with the real-world problems with many local optima. Lastly, metaheuristic optimisers can solve a variety of problems without changing their structure [47]. Therefore, they are flexible in dealing with most problems by assuming the problems as black boxes.

2.4. Metaheuristic Optimisation Algorithms

GA, known as the classical metaheuristic algorithm, was proposed in 1960 to overcome the drawbacks of deterministic algorithms. The simplicity and robustness of GA have shown that nature can always serve as a source of inspiration to solve complex optimisation problems. Since then, more and more nature-inspired metaheuristic algorithms have been proposed, as shown in Table 2.1. The metaheuristic algorithms used as the competing algorithms in this study will be discussed further in Section 2.4.1 to Section 2.4.7.

Metaheuristic algorithms can be classified into two groups, namely, single-based algorithms and population-based algorithms. In single-based algorithms, only one single solution is generated during initialisation. The solution is then being

Table 2.1. Timeline of metaheuristic algorithm

Metaheuristic Algorithm	**Inspiration**	**Year**
Genetic Algorithm (GA) [10]	Process of natural selection	1960
Simulated Annealing (SA) [48]	Annealing process in metallurgy	1983
Ant Colony Optimisation (ACO) [49]	Ant colony	1992
Particle Swarm Optimisation (PSO) [14]	Intelligent social behaviour of bird flock	1995
Differential Evolution (DE) [12]	Natural evolution	1997
Harmony Search (HS) [20]	Improvisation of music players	2001
Bacterial Foraging Optimisation [50]	Social foraging of *E. coli* bacterial	2002
Honey Bee Algorithm (HBA) [51]	Social foraging of honey bee colonies	2004
Virtual Bee Algorithm (VBA) [52]	Swarm interactions of social honey bee	2005
Artificial Bee Colony (ABC) [13]	Intelligent behaviour of honey bee swarms	2006
Big Bang–Big Crunch (BBBC) [26]	Evolution of the universe	2006
Invasive Weed Optimisation (IWO) [18]	Colonising weeds	2006
Imperialist Competitive Algorithm (ICA) [23]	Imperialistic competition	2007
Biogeography-Based Optimisation (BBO) [53]	Geographical distribution of biological organisms	2008
Firefly Algorithm (FA) [54]	Social behaviour of fireflies	2009
Gravitational Search Algorithm (GSA) [24]	Law of gravity and mass interactions	2009
Cuckoo Search Algorithm (CSA) [15]	Obligate brood parasitism of some cuckoo species	2009
Bat Algorithm (BAT) [16]	Echolocation behaviour of bats	2010
Teaching-Learning-Based Optimisation (TLBO) [21]	Philosophy of the teaching–learning process	2011
Water Cycle Algorithm (WCA) [25]	Water cycle process	2012
Mine Blast Algorithm (MBA) [55]	Mine bomb explosion	2012
Flower Pollination Algorithm (FPA) [19]	Pollination process of flowering species	2013
Dolphin Echolocation Algorithm (DEA) [56]	Echolocation ability of dolphins	2013
Grey Wolf Optimiser (GWO) [47]	Social hierarchy and hunting behaviour of grey wolves	2014
Moth-flame Optimisation Algorithm (MOA) [4]	Navigation method of moths	2015
Multi-Verse Optimiser [27]	Multi-verse theory	2015
Whale Optimisation Algorithm (WOA) [46]	Social behaviour of humpback whales	2016
Sine Cosine Algorithm (SCA) [57]	Mathematical model based on sine and cosine functions	2016
Grasshopper Optimisation Algorithm (GOA) [7]	Swarming behaviour of grasshoppers	2017
Salp Swarm Algorithm (SSA) [35]	Swarming behaviour of salps during navigating and foraging in oceans	2017
Atom Search Optimisation (ASO) [58]	Interaction and constraint forces of atom	2018
Mouth Brooding Fish Algorithm (MBFA) [59]	Life cycle of mouth brooding fish	2018
Neural Network Algorithm (NNA) [60]	Structure of artificial neural networks and biological nervous systems	2018
Squirrel Search Algorithm (SS) [17]	Dynamic foraging behaviour of southern flying squirrels	2018
Tree Growth Algorithm (TGA) [11]	Tree's growing behaviour	2018
Harris Hawks Optimisation (HHO) [61]	Cooperative behaviours and chasing styles of predatory birds, Harris' hawks	2019
Sailfish Optimiser (SFO) [9]	Group of hunting sailfish	2019

Table 2.1. (continued)

Metaheuristic Algorithm	Inspiration	Year
Multivariable Grey Prediction Evolution Algorithm (MGPEA) [62]	Grey prediction theory	2020
Group Teaching Optimisation Algorithm (GTOA) [43]	Group teaching mechanism	2020
Tunicate Swarm Algorithm (TSA) [42]	Jet propulsion and swarm behaviours of tunicate	2020
Marine Predators Algorithm (MPA) [44]	Foraging strategy of ocean predators	2020
Water Strider Algorithm (WSA) [63]	Life cycle of water strider bugs	2020
Dynastic Optimisation Algorithm (DOA) [64]	Social behaviour in human dynasties	2020

improved after every iteration. Meanwhile, in population-based algorithms, more than one solution are initialised and iteratively enhanced. The simulations of these two types of algorithms are shown in Figure 2.5 and Figure 2.6. The advantage of single-based algorithms is that the required NOFE is low in solving a specific problem. Examples of popular algorithms in this group are Tabu Search (TS) [65, 66], Iterated Local Search (ILS) [67] and Simulated Annealing (SA) [48]. However, this type of algorithm may experience premature convergence, where the single solution gets

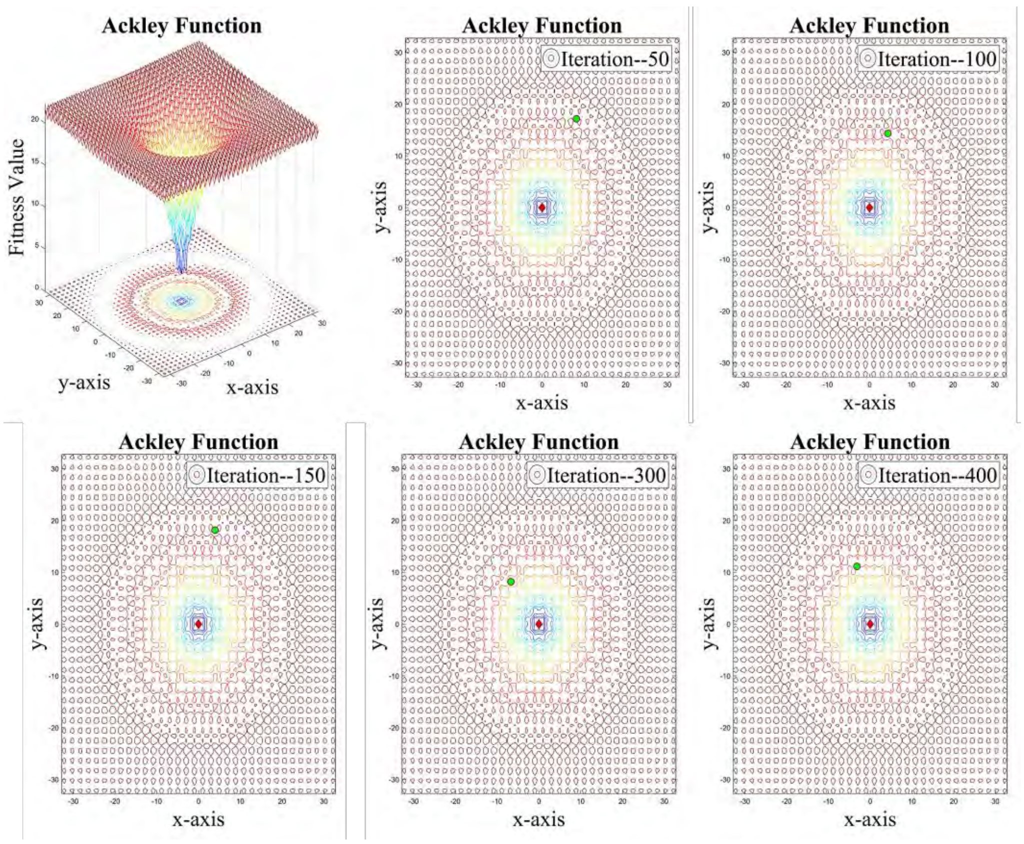

Figure 2.5. Solving Ackley function using single-based algorithm. It cannot converge to the global optima due to getting trapped in different local optima.

trapped in the local optima, as shown in Figure 2.5.

In contrast, population-based algorithms can escape from the local optima due to information sharing with each other. With information exchange, population-based algorithms can explore the search space better than a single-based algorithm and move towards the promising regions of the search space. In Figure 2.6, the green-coloured solution represents the best solution in that iteration, while the red diamond-shaped icon indicates the global optimum. As shown in Figure 2.6, 15 solutions are generated initially, and they converge towards the global optimum without getting trapped in the local optima. Therefore, this has shown that population-based algorithms are better in exploring and exploiting the search space but it requires a large NOFE.

A metaheuristic algorithm always begins with population initialisation. A set of initial solutions then experiences reproduction through the updating mechanism of

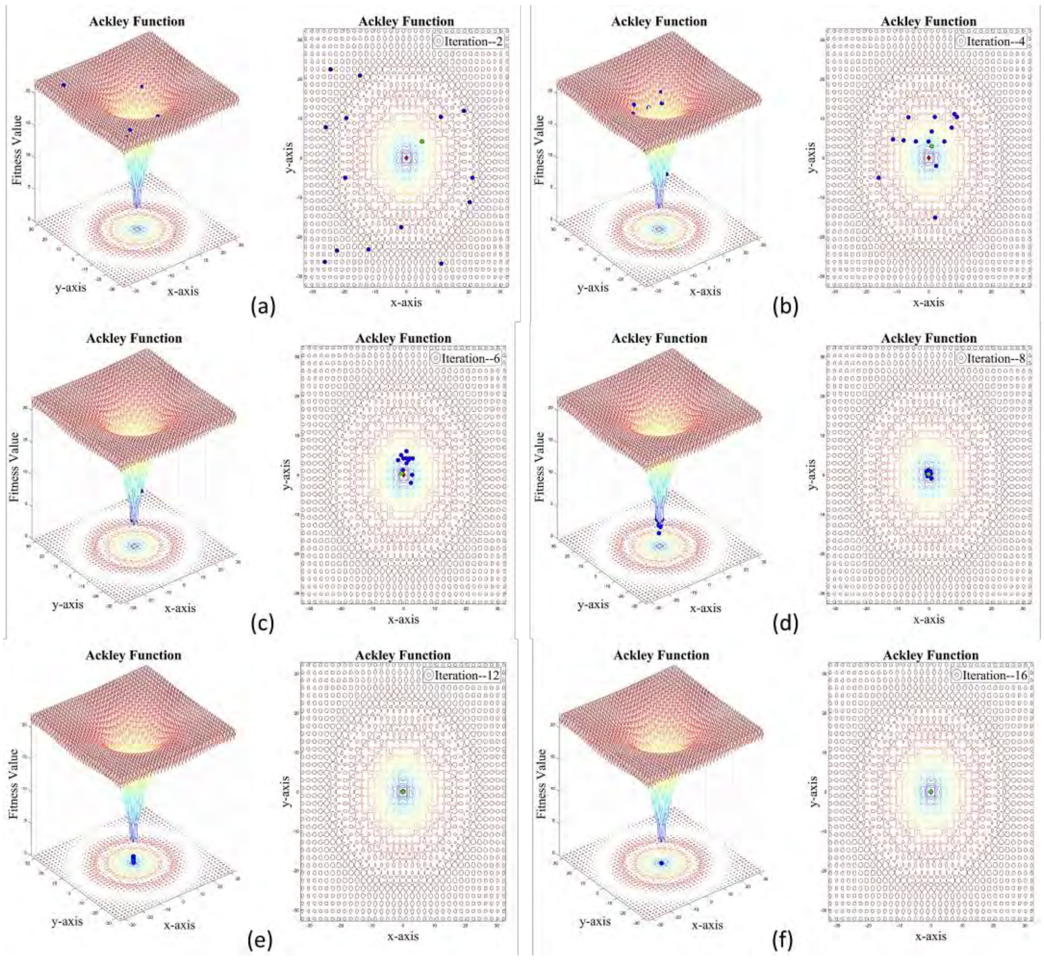

Figure 2.6. Solving Ackley function using population-based algorithm: (a) 2 iterations, (b) 4 iterations, (c) 6 iterations, (d) 8 iterations, (e) 12 iterations and (f) 16 iterations

the algorithm. The updating mechanism is usually composed of the local and global search. The local search is to exploit the solutions near the promising solutions. Meanwhile, the global search is to randomly explore for solutions far away so that to escape from the local minima. The new set of solutions is then compared with their corresponding previous set of solutions. The solution with better fitness value is preserved for the next cycle of evolution. The cycle is repeated until the termination condition is fulfilled, which is either the cycle number has reached its maximum value or the fitness of the global best solution has reached the tolerance value. In general, a metaheuristic algorithm has the components of population initialisation, reproduction and selection.

2.4.1. Genetic Algorithm

GA, which was proposed by John Holland, is the most popular metaheuristic algorithm [10]. The mechanism of GA, which is based on biological evolution from Charles Darwin's theory, has opened a new way of thinking for researchers to combine mathematical computational skills with nature in tackling difficult optimisation problems. Selection, crossover, mutation and reproduction are the main mechanisms in GA, which imitate the evolution of gene in nature. In the optimisation process, GA initialises a set of solutions randomly, which corresponds to chromosomes. Then, the fitness value of each chromosome is evaluated on a specific objective function. Similar to gene evolution, two chromosomes are randomly selected and undergo a crossover process to produce two new chromosomes. After that, these new chromosomes experience a mutation process when the probability is less than a predefined threshold, which is usually a very low value. The production of new chromosomes is then compared with their parents based on their fitness. Lastly, the chromosomes with better fitness value are inherited for the next generation. This process is iterated until the values converge towards global optimum [8].

Since crossover and mutation are two important mechanisms used to improve the solutions in the population, the mechanisms of crossover and mutation are demonstrated in Figure 2.7. For a single-point crossover operation, a crossover point is selected randomly on the parent gene pair. Then, the crossover is achieved by exchanging the data beyond that crossover point in the parent gene pair, as shown in

Figure 2.7(a). Meanwhile, the mutation operator flips the randomly selected bits, as illustrated in Figure 2.7(b). The pseudocode of GA is shown in Figure 2.8.

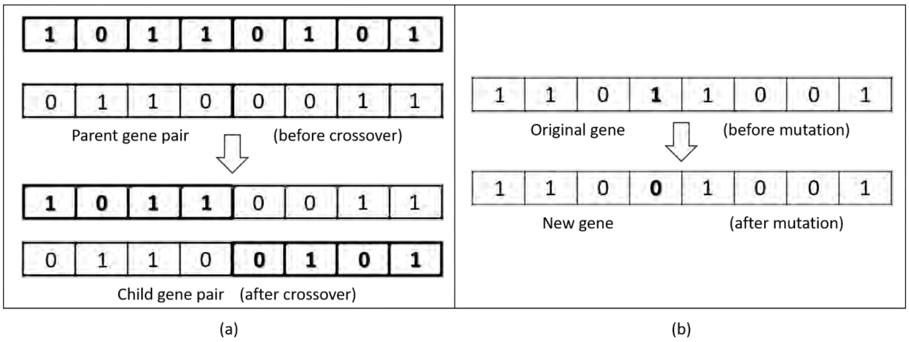

Figure 2.7. Updating mechanism: (a) crossover – the crossover point is randomly generated and (b) mutation – the bit is randomly selected [5]

```
Genetic Algorithm
Determine min or max f(x⃗),  x⃗ = (x_1, x_2, .., x_d)
Define population size, n, number of mutation sites, crossover probability, pc, mutation probability, pm and string length
Define optimal solution, tol
Encode the solution into chromosomes
Generate the initial population
Evaluate the fitness
Identify and record the best solution
while (fmin>tol)
    for i=1:n ( all n solutions in the population)
        if pc>rand,
            Select two solutions randomly
            Generate two new solutions by crossover
            Accept the new solutions if their fitness are better
        end
        if pm>rand,
            Select one solution randomly
            Generate the new solution by mutation
            Accept the new solution if its fitness is better
        end
    end for
Identify current best solution
Record the current best solution
end while
```

Figure 2.8. The pseudocode of GA [10]

2.4.2. Particle Swarm Optimisation

PSO, another famous metaheuristic algorithm, was developed by Kennedy and Eberhart a few years after the creation of GA [14]. Different from GA, it mimics the

REFERENCES

1. V. Kumar and D. Kumar, "An astrophysics-inspired Grey wolf algorithm for numerical optimization and its application to engineering design problems," *Advances in Engineering Software,* vol. 112, pp. 231-254, 2017.
2. A. H. Gandomi and X.-S. Yang, "Benchmark Problems in Structural Optimisation," in *Computational Optimization, Methods and Algorithms*, S. Koziel and X.-S. Yang, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 259-281.
3. S. Mahdavi, M. E. Shiri, and S. Rahnamayan, "Metaheuristics in large-scale global continues optimization: A survey," *Information Sciences,* vol. 295, pp. 407-428, 2015.
4. S. Mirjalili, "Moth-flame optimization algorithm : A novel nature-inspired heuristic paradigm," *Knowledge-Based Systems,* vol. 89, pp. 228-249, 2015.
5. X. S. Yang, *Engineering Optimization: An Introduction with Metaheuristic Applications*. Hoboken, New Jersey: John Wiley & Sons, Inc., 2010.
6. S. Mirjalili, "The Ant Lion Optimizer," *Advances in Engineering Software,* vol. 83, pp. 80-98, 2015.
7. S. Saremi, S. Mirjalili, and A. Lewis, "Grasshopper Optimisation Algorithm: Theory and application," *Advances in Engineering Software,* vol. 105, pp. 30-47, 2017.
8. X.-S. Yang, *Nature-Inspired Metaheuristic Algorithms*. Luniver press, 2014.
9. S. Shadravan, H. R. Naji, and V. K. Bardsiri, "The Sailfish Optimizer : A novel nature-inspired metaheuristic algorithm for solving constrained engineering optimization problems," *Engineering Applications of Artificial Intelligence,* vol. 80, pp. 20-34, 2019.
10. J. Holland, *Adaptation in Natural and Artificial Systems*. Ann Arbor: University of Michigan Press, 1992.
11. A. Cheraghalipour, M. Hajiaghaei-Keshteli, and M. M. Paydar, "Tree Growth Algorithm (TGA): A novel approach for solving optimization problems," *Engineering Applications of Artificial Intelligence,* vol. 72, no. April, pp. 393-414, 2018.
12. R. Storn and K. Price, "Differential Evolution - A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces," *Journal of Global Optimization,* vol. 11, no. 4, pp. 341-359, 1997.
13. D. Karaboga and B. Basturk, "An Artificial Bee Colony (abc) Algorithm for Numeric Function Optimization," *IEEE Swarm Intelligence Symposium 2006,* 2006.
14. J. Kennedy and R. Eberhart, "Particle Swarm Optimization," Piscataway, New Jersey: IEEE Service Center.

15. X. S. Yang and S. Deb, "Cuckoo search via Lévy flights," *2009 World Congress on Nature and Biologically Inspired Computing, NABIC 2009 - Proceedings,* pp. 210-214, 2009.
16. X. S. Yang, "A new metaheuristic Bat-inspired Algorithm," *Studies in Computational Intelligence,* vol. 284, pp. 65-74, 2010.
17. M. Jain, V. Singh, and A. Rani, "A novel nature-inspired algorithm for optimization: Squirrel search algorithm," *Swarm and Evolutionary Computation,* no. November 2017, pp. 1-28, 2018.
18. A. R. Mehrabian and C. Lucas, "A novel numerical optimization algorithm inspired from weed colonization," *Ecological Informatics,* vol. 1, pp. 355-366, 2006.
19. X.-S. Yang, "Flower Pollination Algorithm for Global Optimization," *arXiv,* 2012.
20. Z. Geem, J. Kim, and G. V. Loganathan, "A New Heuristic Optimization Algorithm: Harmony Search," *Simulation,* vol. 76, no. 2, pp. 60-68, 2001.
21. R. V. Rao, V. J. Savsani, and D. P. Vakharia, "Teaching-learning-based optimization : A novel method for constrained mechanical design optimization problems," *Computer-Aided Design,* vol. 43, no. 3, pp. 303-315, 2011.
22. A. H. Kashan, "League Championship Algorithm (LCA): An algorithm for global optimization inspired by sport championships," *Applied Soft Computing Journal,* vol. 16, pp. 171-200, 2014.
23. E. Atashpaz-Gargari and C. Lucas, "Imperialist Competitive Algorithm: An Algorithm for Optimization Inspired by Imperialistic Competition," *2007 IEEE Congress on Evolutionary Computation,* pp. 4661-4667, 2007.
24. E. Rashedi, H. Nezamabadi-pour, and S. Saryazdi, "GSA : A Gravitational Search Algorithm," *Information Sciences,* vol. 179, no. 13, pp. 2232-2248, 2009.
25. H. Eskandar, A. Sadollah, A. Bahreininejad, and M. Hamdi, "Water cycle algorithm - A novel metaheuristic optimization method for solving constrained engineering optimization problems," *Computers and Structures,* vol. 110-111, pp. 151-166, 2012.
26. O. K. Erol and I. Eksin, "A new optimization method : Big Bang – Big Crunch," *Advances in Engineering Software,* vol. 37, pp. 106-111, 2006.
27. S. Mirjalili, S. M. Mirjalili, and A. Hatamlou, "Multi-Verse Optimizer : a nature-inspired algorithm for global optimization," *Neural Computing and Applications,* vol. 27, pp. 495-513, 2016.
28. J. Liu, C. Wu, G. Wu, and X. Wang, "A novel differential search algorithm and applications for structure design," *Applied Mathematics and Computation,* vol. 268, pp. 246-269, 2015.
29. I. Brajevic and M. Tuba, "An upgraded artificial bee colony (ABC) algorithm for constrained optimization problems," *Journal of Intelligent Manufacturing,* vol. 24, no. 4, pp. 729-740, 2012.
30. J. Yi, X. Li, C.-H. Chu, and L. Gao, "Parallel chaotic local search enhanced harmony search algorithm for engineering design optimization," *Journal of Intelligent Manufacturing,* vol. 30, no. 1, pp. 405-428, 2016.
31. I. Brajević and J. Ignjatović, "An upgraded firefly algorithm with feasibility-based rules for constrained engineering optimization problems," *Journal of Intelligent Manufacturing,* vol. 30, no. 6, pp. 2545-2574, 2018.
32. H. Liu, Y. Wang, L. Tu, G. Ding, and Y. Hu, "A modified particle swarm optimization for large-scale numerical optimizations and engineering design

problems," *Journal of Intelligent Manufacturing,* vol. 30, no. 6, pp. 2407-2433, 2018.

33. J.-S. Chou and N.-T. Ngo, "Modified firefly algorithm for multidimensional optimization in structural design problems," *Structural and Multidisciplinary Optimization,* 2016.
34. D. H. Wolpert and W. G. Macready, "No free lunch theorems for search," *Most,* pp. 1-38, 1997.
35. S. Mirjalili, A. H. Gandomi, S. Zahra, and S. Saremi, "Salp Swarm Algorithm : A bio-inspired optimizer for engineering design problems," *Advances in Engineering Software,* vol. 114, pp. 163-191, 2017.
36. A. Zhou, B.-y. Qu, H. Li, S.-z. Zhao, and P. Nagaratnam, "Multiobjective evolutionary algorithms : A survey of the state of the art," *Swarm and Evolutionary Computation,* vol. 1, no. 1, pp. 32-49, 2011.
37. C. Onwubiko, *Introduction To Engineering Design Optimization*. Upper Saddle River, New Jersey: Prentice-Hall Inc., 2000.
38. P. Debye, "Näherungsformeln für die Zylinderfunktionen für große Werte des Arguments und unbeschränkt veränderliche Werte des Index," *Mathematische Annalen,* vol. 67, no. 4, pp. 535-558, 1909/12/01 1909.
39. A. Antoniou and W.-S. Lu, *Practical Optimization: Algorithms and Engineering Applications*, 1 ed. Springer, 2007.
40. J. A. Nelder and R. Mead, "A Simplex Method for Function Minimization," *Comput. J.,* vol. 7, pp. 308-313, 1965.
41. W. Karush, "Minima of Functions of Several Variables with Inequalities as Side Conditions," in *Traces and Emergence of Nonlinear Programming*, G. Giorgi and T. H. Kjeldsen, Eds. Basel: Springer Basel, 2014, pp. 217-245.
42. S. Kaur, L. K. Awasthi, A. L. Sangal, and G. Dhiman, "Tunicate Swarm Algorithm: A new bio-inspired based metaheuristic paradigm for global optimization," *Engineering Applications of Artificial Intelligence,* vol. 90, 2020.
43. Y. Zhang and Z. Jin, "Group teaching optimization algorithm: A novel metaheuristic method for solving global optimization problems," *Expert Systems with Applications,* vol. 148, 2020.
44. A. Faramarzi, M. Heidarinejad, S. Mirjalili, and A. H. Gandomi, "Marine Predators Algorithm: A Nature-inspired Metaheuristic," *Expert Systems with Applications,* 2020.
45. X.-S. Wang, M.-L. Hao, and Y.-H. Cheng, "On the use of differential evolution for forward kinematics of parallel manipulators," *Applied Mathematics and Computation,* vol. 205, no. 2, pp. 760-769, 2008.
46. S. Mirjalili and A. Lewis, "The Whale Optimization Algorithm," *Advances in Engineering Software,* vol. 95, pp. 51-67, 2016.
47. S. Mirjalili, S. Mohammad, and A. Lewis, "Grey Wolf Optimizer," *Advances in Engineering Software,* vol. 69, pp. 46-61, 2014.
48. S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by Simulated Annealing," *Science,* vol. 220, no. 4589, pp. 671-680, 1983.
49. A. Colorni, M. Dorigo, and V. Maniezzo, "Distributed Optimization by Ant Colonies," Cambridge: MIT Press.
50. K. M. Passino, "Biomimicry of bacterial foraging for distributed optimization and control," *IEEE Control Systems,* vol. 22, no. June, pp. 52-67, 2002.
51. S. Nakrani and C. Tovey, "On Honey Bees and Dynamic Allocation in an Internet Server Colony," pp. 223-240, 2004.

52. X.-S. Yang, "Engineering Optimization via Nature-inspired Virtual Bee Algorithms," *International Work-conference on the Interplay between Natural and Artificial Computation (IWINAC) 2005,* pp. 317-323, 2005.
53. D. Simon and S. Member, "Biogeography-Based Optimization," *IEEE Transactions on Evolutionary Computation,* vol. 12, no. 6, pp. 702-713, 2008.
54. X.-s. Yang, "Firefly Algorithms for Multimodal Optimization," *Stochastic Algorithms: Foundations and Applications,* vol. Vol. 5792, pp. 169-178, 2009.
55. A. Sadollah, A. Bahreininejad, H. Eskandar, and M. Hamdi, "Mine blast algorithm: A new population based algorithm for solving constrained engineering optimization problems," *Applied Soft Computing Journal,* vol. 13, no. 5, pp. 2592-2612, 2013.
56. A. Kaveh and N. Farhoudi, "A new optimization method: Dolphin echolocation," *Advances in Engineering Software,* vol. 59, pp. 53-70, 2013.
57. S. Mirjalili, "SCA : A Sine Cosine Algorithm for solving optimization problems," *Knowledge-Based Systems,* vol. 96, pp. 120-133, 2016.
58. W. Zhao, L. Wang, and Z. Zhang, "Atom search optimization and its application to solve a hydrogeologic parameter estimation problem," *Knowledge-Based Systems,* vol. 163, pp. 283-304, 2019.
59. E. Jahani and M. Chizari, "Tackling global optimization problems with a novel algorithm – Mouth Brooding Fish algorithm," *Applied Soft Computing Journal,* vol. 62, pp. 987-1002, 2018.
60. A. Sadollah, H. Sayyaadi, and A. Yadav, "A dynamic metaheuristic optimization model inspired by biological nervous systems : Neural network algorithm," *Applied Soft Computing Journal,* vol. 71, pp. 747-782, 2018.
61. A. A. Heidari, S. Mirjalili, H. Faris, I. Aljarah, M. Mafarja, and H. Chen, "Harris Hawks Optimization : Algorithm and Applications," *Future Generation Computer Systems,* 2019.
62. X. Xu, Z. Hu, Q. Su, Y. Li, and J. Dai, "Multivariable grey prediction evolution algorithm: A new metaheuristic," *Applied Soft Computing,* vol. 89, 2020.
63. A. Kaveh and A. Dadras Eslamlou, "Water strider algorithm: A new metaheuristic and applications," *Structures,* vol. 25, pp. 520-541, 2020.
64. S.-u.-R. Massan, A. I. Wagan, and M. M. Shaikh, "A new metaheuristic optimization algorithm inspired by human dynasties with an application to the wind turbine micrositing problem," *Applied Soft Computing,* vol. 90, 2020.
65. F. Glover, "Tabu Search — Part I," *ORSA Journal on Computing,* vol. 1(3), pp. 190-206, 1989.
66. F. Glover, "Tabu Search — Part II," *ORSA Journal on Computing,* vol. 2(1), pp. 4-32, 1990.
67. H. R. Lourenco, O. C. Martin, and T. Stutzle, "Iterated Local Search," *arXiv: preprint math/0102188,* pp. 1-49, 2001.
68. B. J. Glover, *Understanding Flowers & Flowering: An Integrated Approach*. New York: Oxford University Press, 2007, pp. 1-223.
69. A. Mishra and S. Deb, "Assembly sequence optimization using a flower pollination algorithm-based approach," *Journal of Intelligent Manufacturing,* vol. 30, no. 2, pp. 461-482, 2019.
70. P. Ong, D. Vui, C. S. Ho, and C. H. Ng, "Modeling and optimization of cold extrusion process by using response surface methodology and metaheuristic approaches," *Neural Computing and Applications,* 2016.
71. S. Zhang, Y. Xu, W. Zhang, and D. Yu, "A new fuzzy QoS-aware manufacture service composition method using extended flower pollination algorithm,"

Journal of Intelligent Manufacturing, journal article vol. 30, no. 5, pp. 2069-2083, June 01 2019.

72. E. Emary, H. M. Zawbaa, A. E. Hassanien, and B. Parv, "Multi-objective retinal vessel localization using flower pollination search algorithm with pattern search," *Advances in Data Analysis and Classification,* vol. 11, no. 3, pp. 611-627, 2017.
73. J. Oesterle, L. Amodeo, and F. Yalaoui, "A comparative study of Multi-Objective Algorithms for the Assembly Line Balancing and Equipment Selection Problem under consideration of Product Design Alternatives," *Journal of Intelligent Manufacturing,* vol. 30, no. 3, pp. 1021-1046, 2019.
74. M. Eusuff, K. Lansey, and F. Pasha, "Shuffled frog-leaping algorithm: A memetic meta-heuristic for discrete optimization," *Engineering Optimization,* vol. 38, no. 2, pp. 129-154, 2006.
75. D. Tang, J. Yang, S. Dong, and Z. Liu, "A Lévy flight-based shuffled frog-leaping algorithm and its applications for continuous optimization problems," *Applied Soft Computing Journal,* vol. 49, pp. 641-662, 2016.
76. H.-b. Wang, X.-n. Ren, and X.-y. Tu, "Bee and Frog Co-Evolution Algorithm and its application," *Applied Soft Computing Journal,* vol. 56, pp. 182-198, 2017.
77. R. Salgotra and U. Singh, "Application of mutation operators to flower pollination algorithm," *Expert Systems with Applications,* vol. 79, pp. 112-129, 2017.
78. S. Xu, Y. Wang, and F. Huang, "Optimization of multi-pass turning parameters through an improved flower pollination algorithm," *International Journal of Advanced Manufacturing Technology,* vol. 89, no. 1-4, pp. 503-514, 2017.
79. P. Ong, K. M. Ong, and C. K. Sia, "An Improved Flower Pollination Algorithm with Chaos Theory for Function Optimization," 2016.
80. D. Yousri, A. M. AbdelAty, L. A. Said, A. S. Elwakil, B. Maundy, and A. G. Radwan, "Chaotic Flower Pollination and Grey Wolf Algorithms for parameter extraction of bio-impedance models," *Applied Soft Computing,* vol. 75, pp. 750-774, 2019.
81. M. Abdel-Baset, H. Wu, Y. Zhou, and L. Abdel-Fatah, "Elite opposition-flower pollination algorithm for quadratic assignment problem," *Journal of Intelligent and Fuzzy Systems,* vol. 33, no. 2, pp. 901-911, 2017.
82. S. Xu and Y. Wang, "Parameter estimation of photovoltaic modules using a hybrid flower pollination algorithm," *Energy Conversion and Management,* vol. 144, pp. 53-68, 2017.
83. Y. Zhou, R. Wang, and Q. Luo, "Elite opposition-based flower pollination algorithm," *Neurocomputing,* vol. 188, pp. 294-310, 2016.
84. M. Abdel-Baset and I. M. Hezam, "A hybrid flower pollination algorithm for solving ill-conditioned set of equations," *International Journal of Bio-Inspired Computation,* vol. 8, no. 4, 2016.
85. A. Kaur, S. K. Pal, and A. P. Singh, "Hybridization of Chaos and Flower Pollination Algorithm over K-Means for data clustering," *Applied Soft Computing,* 2019.
86. E. Nabil, "A Modified Flower Pollination Algorithm for Global Optimization," *Expert Systems With Applications,* vol. 57, pp. 192-203, 2016.
87. D. Chakraborty, "DE-FPA : A Hybrid Differential Evolution-Flower Pollination Algorithm for Function Minimization," *International Conference on High Performance Computing and Applications (ICHPCA),* pp. 1-6, 2014.

88. S. Mandal, A. Khan, G. Saha, and R. K. Pal, "Large-Scale Recurrent Neural Network Based Modelling of Gene Regulatory Network Using Cuckoo Search-Flower Pollination Algorithm," *Advances in Bioinformatics,* vol. 2016, p. 5283937, 2016.

89. Y. Zhou and R. Wang, "An Improved Flower Pollination Algorithm for Optimal Unmanned Undersea Vehicle Path Planning Problem," *International Journal of Pattern Recognition and Artificial Intelligence,* vol. 30, no. 04, p. 1659010, 2016.

90. M. Gao, J. Shen, and J. Jiang, "Visual tracking using improved flower pollination algorithm," *Optik,* vol. 156, pp. 522-529, 2018.

91. L. H. Pham, N. H. An, and D. T. Tam, "Modified Flower Pollination Algorithm for Solving Economic Dispatch Problem," in *AETA 2017 - Recent Advances in Electrical Engineering and Related Sciences: Theory and Application*(Lecture Notes in Electrical Engineering, 2018, pp. 934-942.

92. M. Lei, Q. Luo, Y. Zhou, C. Tang, and Y. Gao, *BFPA: Butterfly Strategy Flower Pollination Algorithm* (Intelligent Computing Theories and Application). 2019, pp. 739-748.

93. Y. Chen and D. Pi, "An innovative flower pollination algorithm for continuous optimization problem," *Applied Mathematical Modelling,* vol. 83, pp. 237-265, 2020.

94. J. Andreas, A. Sciligo, T. Witt, A. M. El-sayed, and D. M. Suckling, "Pollinator-prey conflict in carnivorous plants," *Biological Reviews,* vol. 87, no. 3, pp. 602-615, 2011.

95. A. Pavlovic and M. Saganova, "A novel insight into the cost-benefit model for the evolution of botanical carnivory " *Annals of Botany,* vol. 115, no. 7, pp. 1075-1092, 2015.

96. A. Mithöfer, "Carnivorous pitcher plants : Insights in an old topic," *Phytochemistry,* vol. 72, pp. 1678-1682, 2011.

97. S. McPherson, *Carnivorous Plants and their Habitats*. Redfern Natural History Productions, 2010, pp. 1442-1442.

98. Salmon and Bruce, *Carnivorous Plants of New Zealand*. Ecosphere Publications, 2001.

99. Y. Zhang and J. Wang, "Obstacle Avoidance for Kinematically Redundant Manipulators Using a Dual Neural Network," vol. 34, no. 1, pp. 752-759, 2004.

100. M. Sajjad, H. Talpur, and M. H. Shaikh, "Automation of Mobile Pick and Place Robotic System for Small," pp. 522-526, 2012.

101. M. Wang, M. Luo, T. Li, and M. Ceccarelli, "A Unified Dynamic Control Method for a Redundant Dual Arm Robot," *Journal of Bionic Engineering,* vol. 12, no. 3, pp. 361-371, 2015.

102. S. Glumac and Z. Kovacic, "Microimmune Algorithm for Solving Inverse Kinematics of Redundant Robots," presented at the 2013 IEEE International Conference on Systems, Man, and Cybernetics, 2013.

103. T. Rybus, "Obstacle avoidance in space robotics : Review of major challenges and proposed solutions," *Progress in Aerospace Sciences,* vol. 101, no. February, pp. 31-48, 2018.

104. A. Li, P. Stief, J.-y. Dantan, A. Etienne, and A. Siadat, "Kinematics Analysis and Trajectory Planning of collaborative welding robot with multiple manipulators," *Procedia CIRP,* vol. 81, pp. 1034-1039, 2019.

105. C. Lopez-Franco, J. Hernandez-Barragan, A. Y. Alanis, and N. Arana-Daniel, "A soft computing approach for inverse kinematics of robot manipulators," *Engineering Applications of Artificial Intelligence,* vol. 74, pp. 104-120, 2018.
106. "Robot Forward and Inverse Kinematics Research using Matlab," *International Journal of Recent Technology and Engineering,* vol. 8, no. 2S3, pp. 29-35, 2019.
107. H.-C. Huang, C.-P. Chen, and P.-R. Wang, "Particle swarm optimization for solving the inverse kinematics of 7-DOF robotic manipulators," *2012 IEEE International Conference on Systems, Man, and Cybernetics (SMC),* pp. 3105-3110, 2012.
108. Z. M. Li, C. G. Li, and S. J. Lv, "A method for solving inverse kinematics of PUMA560 manipulator based on PSO-RBF network," presented at the 2012 8th International Conference on Natural Computation, Chongqing, 2012.
109. A. El-Sherbiny, M. A. Elhosseini, and A. Y. Haikal, "A comparative study of soft computing methods to solve inverse kinematics problem," *Ain Shams Engineering Journal,* vol. 9, no. 4, pp. 2535-2548, 2018.
110. R. Köker, "A genetic algorithm approach to a neural-network-based inverse kinematics solution of robotic manipulators based on error minimization," *Information Sciences,* vol. 222, pp. 528-543, 2013.
111. Y. Wang, "An Incremental Method for Forward Kinematics of Parallel Manipulators," *2006 IEEE Conference on Robotics, Automation and Mechatronics,* pp. 1-5, 2006.
112. R. Chandra and L. Rolland, "On solving the forward kinematics of 3RPR planar parallel manipulator using hybrid metaheuristics," *Applied Mathematics and Computation,* vol. 217, no. 22, pp. 8997-9008, 2011.
113. S. Li, S. Chen, B. Liu, Y. Li, and Y. Liang, "Decentralized kinematic control of a class of collaborative redundant manipulators via recurrent neural networks," *Neurocomputing,* vol. 91, pp. 1-10, 2012.
114. M. Duguleana, F. Grigore, A. Teirelbar, and G. Mogan, "Obstacle avoidance of redundant manipulators using neural networks based reinforcement learning," *Robotics and Computer Integrated Manufacturing,* vol. 28, no. 2, pp. 132-146, 2012.
115. A. R. J. Almusawi, L. C. Dulger, and S. Kapucu, "A New Artificial Neural Network Approach in Solving Inverse Kinematics of Robotic Arm (Denso VP6242)," *Comput Intell Neurosci,* vol. 2016, p. 5720163, 2016.
116. A. C. Nearchou, "Solving the inverse kinematics problem of redundant robots operating in complex environments via a modified genetic algorithm," *Mechanism and Machine Theory,* vol. 33, no. 3, pp. 273-292, 1998.
117. Y. Yang, G. Peng, Y. Wang, and H. Zhang, "A New Solution for Inverse Kinematics of 7-DOF Manipulator Based on Genetic Algorithm," *2007 IEEE International Conference on Automation and Logistics,* pp. 1947-1951, 2007.
118. G. S. Chyan and S. G. Ponnambalam, "Obstacle avoidance control of redundant robots using variants of particle swarm optimization," *Robotics and Computer Integrated Manufacturing,* vol. 28, no. 2, pp. 147-153, 2012.
119. C.-J. Lin, T.-H. S. Li, P.-H. Kuo, and Y.-H. Wang, "Integrated particle swarm optimization algorithm based obstacle avoidance control design for home service robot," *Computers & Electrical Engineering,* 2015.
120. A. El-Sherbiny, M. A. Elhosseini, and A. Y. Haikal, "A new ABC variant for solving inverse kinematics problem in 5 DOF robot arm," *Applied Soft Computing,* vol. 73, pp. 24-38, 2018.

121. L. Zhao, Z. He, W. Cao, and D. Zhao, "Real-Time Moving Object Segmentation and Classification from HEVC Compressed Surveillance Video," *IEEE Transactions on Circuits and Systems for Video Technology,* vol. 28, no. 6, pp. 1346-1357, 2018.
122. A. Aflakian, A. Safaryazdi, M. Tale Masouleh, and A. Kalhor, "Experimental study on the kinematic control of a cable suspended parallel robot for object tracking purpose," *Mechatronics,* vol. 50, no. November 2017, pp. 160-176, 2018.
123. A. Koubaa and B. Qureshi, "DroneTrack: Cloud-Based Real-Time Object Tracking using Unmanned Aerial Vehicles," *IEEE Access,* vol. 6, pp. 1-1, 2018.
124. W. Barhoumi, "Detection of highly articulated moving objects by using co-segmentation with application to athletic video sequences," *Signal, Image and Video Processing,* vol. 9, no. 7, pp. 1705-1715, 2015.
125. D. Koniar, L. Hargaš, Z. Loncová, A. Simonová, F. Duchoň, and P. Beňo, "Visual system-based object tracking using image segmentation for biomedical applications," *Electrical Engineering,* vol. 99, no. 4, pp. 1349-1366, 2017.
126. A. Banharnsakun and S. Tanathong, "A hierarchical clustering of features approach for vehicle tracking in traffic environments," *International Journal of Intelligent Computing and Cybernetics,* vol. 9, no. 4, pp. 354-368, 2016.
127. A. M. Abdel Tawab, M. B. Abdelhalim, and S. E. D. Habib, "Efficient multi-feature PSO for fast gray level object-tracking," *Applied Soft Computing Journal,* vol. 14, no. PART C, pp. 317-337, 2014.
128. X. Li, W. Hu, C. Shen, Z. Zhang, A. Dick, and A. v. d. Hengel, "A Survey of Appearance Models in Visual Object Tracking," pp. 1-42, 2013.
129. M.-L. Gao, X.-H. He, D.-S. Luo, and Y.-M. Yu, "Object tracking based on harmony search: comparative study," *Journal of Electronic Imaging,* vol. 21, no. 4, pp. 043001-1, 2012.
130. D. Comaniciu, V. Ramesh, and P. Meer, "Kernel-based object tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 25, no. 5, pp. 564-577, 2003.
131. T.-l. Liu, "Real-Time Tracking Using Trust-Region Methods," vol. 26, no. 3, pp. 397-402, 2004.
132. M. Kass, a. Witkin, and D. Terzopoulos, "Snakes: Active contour models," *International Journal of Computer Vision,* vol. 1, no. 4, pp. 321-331, 1988.
133. C. J. Veenman, M. J. T. Reinders, and E. Backer, "Resolving motion correspondence for densely moving points," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 23, no. 1, pp. 54-72, 2001.
134. M.-j. Zhang and B.-s. Kang, "Particle Filter Tracking Method Using Graphical Model," no. 1, pp. 1-11, 2015.
135. C. Bae, K. Kang, G. Liu, and Y. Y. Chung, "A novel real time video tracking framework using adaptive discrete swarm optimization," *Expert Systems with Applications,* vol. 64, pp. 385-399, 2016.
136. L. Martino, V. Elvira, D. Luengo, J. Corander, and F. Louzada, "Orthogonal parallel MCMC methods for sampling and optimization," *Digital Signal Processing,* vol. 58, pp. 64-84, 2016.
137. J.-B. Lacambre, M. Narozny, and J.-M. Louge, "Limitations of the unscented Kalman filter for the attitude determination on an inertial navigation system," *2013 IEEE Digital Signal Processing and Signal Processing Education Meeting (DSP/SPE),* pp. 187-192, 2013.

138. H. S. Boroujeni, N. M. Charkari, M. Behrouzifar, and P. T. Makhsoos, "Tracking Multiple Variable-Sizes Moving Objects in LFR Videos Using a Novel genetic Algorithm Approach," *Communications in Computer and Information Science,* pp. 143-153, 2012.
139. K. Shanmugapriya and R. S. M. Malar, "A multi-balanced hybrid optimization technique to track objects using rough set theory," *Signal, Image and Video Processing,* vol. 11, no. 3, pp. 415-421, 2017.
140. C.-H. Chen, C.-C. Wang, and M.-C. Yan, "Robust tracking of multiple persons in real-time video," *Multimedia Tools and Applications,* vol. 75, pp. 16683-16697, 2016.
141. M.-L. Gao, D.-S. Luo, Q.-Z. Teng, X.-H. He, and J. Jiang, "Object tracking using firefly algorithm," *IET Computer Vision,* vol. 7, no. 4, pp. 227-237, 2013.
142. X. Zhang, W. Hu, W. Qu, and S. Maybank, "Multiple object tracking via species-based particle swarm optimization," *Circuits and Systems for Video Technology, IEEE Transactions on,* vol. 20, no. 11, pp. 1590-1602, 2010.
143. M.-L. Gao *et al.*, "A novel visual tracking method using bat algorithm," *Neurocomputing,* vol. 177, pp. 612-619, 2016.
144. J. Fourie, S. Mills, and R. Green, "Harmony filter: A robust visual tracking system using the improved harmony search algorithm," *Image and Vision Computing,* vol. 28, no. 12, pp. 1702-1716, 2010.
145. M.-L. Gao, L.-J. Yin, G.-F. Zou, H.-T. Li, and W. Liu, "Visual tracking method based on cuckoo search algorithm," *Optical Engineering,* vol. 54, no. 7, pp. 073105-073105, 2015.
146. T. Ljouad, A. Amine, and M. Rziza, "A hybrid mobile object tracker based on the modified Cuckoo Search algorithm and the Kalman Filter," *Pattern Recognition,* vol. 47, no. 11, pp. 3597-3613, 2014.
147. F. Sardari and M. Ebrahimi Moghaddam, "A hybrid occlusion free object tracking method using particle filter and modified galaxy based search meta-heuristic algorithm," *Applied Soft Computing Journal,* vol. 50, pp. 280-299, 2017.
148. H. Nenavath, D. R. Kumar Jatoth, and D. S. Das, "A synergy of the sine-cosine algorithm and particle swarm optimizer for improved global optimization and object tracking," *Swarm and Evolutionary Computation,* no. August 2017, pp. 1-30, 2018.
149. K. Kang, C. Bae, H. W. F. Yeung, and Y. Y. Chung, "A hybrid gravitational search algorithm with swarm intelligence and deep convolutional feature for object tracking optimization," *Applied Soft Computing Journal,* vol. 66, pp. 319-329, 2018.
150. W. Li, J. Cao, J. Wu, C. Huang, and R. Buyya, "A collaborative filtering recommendation method based on discrete quantum-inspired shuffled frog leaping algorithms in social networks," *Future Generation Computer Systems,* vol. 88, pp. 262-270, 2018.
151. B. Alatas, "Chaotic bee colony algorithms for global numerical optimization," *Expert Systems with Applications,* vol. 37, no. 8, pp. 5682-5687, 2010.
152. L. d. S. Coelho and V. C. Mariani, "Use of chaotic sequences in a biologically inspired algorithm for engineering design optimization," *Expert Systems with Applications,* vol. 34, no. 3, pp. 1905-1913, 2008.
153. A. Baykasoğlu and F. B. Ozsoydan, "Adaptive firefly algorithm with chaos for mechanical design optimization problems," *Applied Soft Computing,* vol. 36, pp. 152-164, 2015.

154. G. Heidari-Bateni and C. D. McGillem, "A Chaotic Direct-Sequence Spread-Spectrum Communication System," *IEEE Transactions on Communications,* vol. 42, no. 234, pp. 1524-1527, 1994.
155. A. H. Gandomi, X. S. Yang, S. Talatahari, and A. H. Alavi, "Firefly algorithm with chaos," *Communications in Nonlinear Science and Numerical Simulation,* vol. 18, no. 1, pp. 89-98, 2013.
156. S. Talatahari, B. Farahmand Azar, R. Sheikholeslami, and A. H. Gandomi, "Imperialist competitive algorithm combined with chaos for global optimization," *Communications in Nonlinear Science and Numerical Simulation,* vol. 17, no. 3, pp. 1312-1319, 2012.
157. A. H. Gandomi and X.-s. Yang, "Chaotic bat algorithm," *Journal of Computational Science,* 2013.
158. G. G. Wang, L. Guo, A. H. Gandomi, G. S. Hao, and H. Wang, "Chaotic Krill Herd algorithm," *Information Sciences,* vol. 274, pp. 17-34, 2014.
159. A. A. Ewees and M. A. Elaziz, "Performance analysis of Chaotic Multi-Verse Harris Hawks Optimization: A case study on solving engineering problems," *Engineering Applications of Artificial Intelligence,* vol. 88, 2020.
160. E. Elbeltagi, T. Hegazy, and D. Grierson, "A modified shuffled frog-leaping optimization algorithm: applications to project management," *Structure and Infrastructure Engineering,* vol. 3, no. 1, pp. 53-60, 2007.
161. H. T. Liang and F. H. Kang, "Adaptive mutation particle swarm algorithm with dynamic nonlinear changed inertia weight," *Optik,* vol. 127, no. 19, pp. 8036-8042, 2016.
162. C. Gan, W. Cao, M. Wu, and X. Chen, "A new bat algorithm based on iterative local search and stochastic inertia weight," *Expert Systems with Applications,* vol. 104, pp. 202-212, 2018.
163. M. R. Ramli, Z. A. Abas, M. I. Desa, Z. Z. Abidin, and M. B. Alazzam, "Enhanced convergence of Bat Algorithm based on dimensional and inertia weight factor," *Journal of King Saud University - Computer and Information Sciences,* pp. 0-6, 2018.
164. J. Denavit, "A kinematic notation for lower-pair mechanisms based on matrices," *Trans. of the ASME. Journal of Applied Mechanics,* vol. 22, pp. 215-221, 1955 1955.
165. Y. Wang, L. Jiang, Q. Liu, and M. Yin, "Optimal Appearance Model for Visual Tracking," no. ii, pp. 1-15, 2016.
166. K.-j. Kim, S.-m. Park, and Y.-j. Choi, "Deciding the Number of Color Histogram Bins for Vehicle Color Recognition," *2008 IEEE Asia-Pacific Services Computing Conference,* pp. 134-138, 2008.
167. A. Bhattacharyya, "Indian Statistical Institute," *Sankhyā: The Indian Journal of Statistics,* vol. 7, no. 4, pp. 401-406, 1946.
168. D. Comaniciu, V. Ramesh, and P. Meer, "Real-time tracking of non-rigid objects using mean shift," *IEEE Conference on Computer Vision and Pattern Recognition,* vol. 2, no. 7, pp. 142-149, 2000.
169. M. Jamil and X.-s. Y. Blekinge, "A Literature Survey of Benchmark Functions For Global Optimization Problems," pp. 1-47, 2013.
170. M. Hellwig and H.-G. Beyer, "Benchmarking evolutionary algorithms for single objective real-valued constrained optimization – A critical review," *Swarm and Evolutionary Computation,* vol. 44, pp. 927-944, 2019/02/01/ 2019.
171. N. Hansen, A. Auger, D. Brockhoff, D. Tusar, and T. Tusar, "COCO: Performance Assessment," 2016.

172. J. J. Liang, P. N. Suganthan, and K. Deb, "Novel Composition Test Functions for Numerical Global Optimization," *Proceedings 2005 IEEE Swarm Intelligence Symposium, 2005. SIS 2005.,* pp. 68-75, 2005.
173. F. Caraffini, A. V. Kononova, and D. Corne, "Infeasibility and structural bias in differential evolution," *Information Sciences,* vol. 496, pp. 161-179, 2019.
174. A. V. Kononova, D. W. Corne, P. De Wilde, V. Shneer, and F. Caraffini, "Structural bias in population-based algorithms," *Information Sciences,* vol. 298, pp. 468-490, 2015.
175. G. Wu, R. Mallipeddi, and P. N. Suganthan, "Problem Definitions and Evaluation Criteria for the CEC 2017 Competition and Special Session on Constrained Single Objective Real-Parameter Optimization Problem Definitions and Evaluation Criteria for the CEC 2017 Competition on Constrained Real- Parameter Optimization," no. October 2016, 2017.
176. R. Tanabe and A. Fukunaga, "Improving the Search Performance of SHADE Using Linear Population Size Reduction," in *2014 IEEE Congress on Evolutionary Computation (CEC)*, 2014, pp. 1658 - 1665.
177. J. Derrac, S. García, D. Molina, and F. Herrera, "A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms," *Swarm and Evolutionary Computation,* vol. 1, no. 1, pp. 3 - 18, 2011.
178. Y.-L. Hsu and T.-C. Liu, "Developing a fuzzy proportional-derivative controller optimization engine for engineering design optimization problems," *Engineering Optimization,* vol. 39, no. 6, pp. 679-700, 2007.
179. J. Kennedy and R. Eberhart, "Particle Swarm Optimization," in *Proceedings of the 1995 IEEE International Conference on Neural Networks*, Piscataway, New Jersey: IEEE Service Center.
180. A. H. Gandomi, X. S. Yang, and A. H. Alavi, "Cuckoo search algorithm: A metaheuristic approach to solve structural optimization problems," *Engineering with Computers,* vol. 29, no. 1, pp. 17-35, 2013.
181. S. S. Rao, *Engineering Optimization: Theory and Practice*, 4 ed. Hoboken, New Jersey: John Wiley & Sons, Inc, 2009.
182. S. M. Nigdeli, G. Bekdas, and X.-S. Yang, "Optimum tuning of mass dampers for seismic structures using flower pollination algorithm," vol. 1, pp. 264-268, 2016.
183. N. B. Guedria, "Improved accelerated PSO algorithm for mechanical engineering optimization problems," *Applied Soft Computing Journal,* vol. 40, pp. 455-467, 2016.
184. E. Sandgren, "Nonlinear Integer and Discrete Programming in Mechanical Design Optimization," *Journal of Mechanical Design,* vol. 112, no. 2, pp. 223-229, 1990.
185. C. A. Coello, "Use of a self-adaptive penalty approach for engineering optimization problems," *Computers in Industry,* vol. 41, no. 2, pp. 113-127, 2000.
186. Q. He and L. Wang, "An effective co-evolutionary particle swarm optimization for constrained engineering design problems," *Engineering Applications of Artificial Intelligence,* vol. 20, no. 1, pp. 89-99, 2007.
187. P. Civicioglu, "Transforming geocentric cartesian coordinates to geodetic coordinates by using differential search algorithm," *Computers and Geosciences,* vol. 46, pp. 229-247, 2012.

188. Y. Wang, H.-x. Li, T. Huang, and L. Li, "Differential evolution based on covariance matrix learning and bimodal distribution parameter setting," *Applied Soft Computing Journal,* vol. 18, pp. 232-247, 2014.
189. D. Karaboga and B. Basturk, "Artificial Bee Colony (ABC) Optimization Algorithm for Solving Constrained Optimization Problems," no. January 2007, 2015.
190. K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation,* vol. 6, no. 2, pp. 182-197, 2002.
191. M. Jaberipour and E. Khorram, "Two improved harmony search algorithms for solving engineering optimization problems," *Communications in Nonlinear Science and Numerical Simulation,* vol. 15, no. 11, pp. 3316-3331, 2010.
192. K. S. Lee, "A new meta-heuristic algorithm for continuous engineering optimization : harmony search theory and practice," *Computer Methods in Applied Mechanics and Engineering,* vol. 194, pp. 3902-3933, 2005.
193. H. Chickermane and H. Change Gea, "Structural Optimization Using A New Local Approximation Method," *International Journal for Numerical Methods in Engineering,* 1996.
194. A. Ravindran, K. M. Ragsdell, and G. V. Reklaitis, *Engineering Optimization: Methods and Application*, 2 ed. Hoboken, New Jersey: John Wiley & Sons, Inc, 2006.
195. P. Kim and J. Lee, "An integrated method of particle swarm optimization and differential evolution †," *Mechanical Science and Technology,* vol. 23, pp. 426-434, 2009.
196. G. G. Wang, "Adaptive Response Surface Method Using Inherited Latin Hypercube Design Points," *Journal of Mechanical Design,* vol. 125, no. June, pp. 210-220, 2003.
197. B. D. Youn, K. K. Choi, R. J. Yang, and L. Gu, "Reliability-based design optimization for crashworthiness of vehicle side impact," *Structural and Multidisciplinary Optimization,* vol. 26, no. 3-4, pp. 272-283, 2004.
198. J. Huang, L. Gao, and X. Li, "An effective teaching-learning-based cuckoo search algorithm for parameter optimization problems in structure designing and machining processes," *Applied Soft Computing Journal,* vol. 36, pp. 349-356, 2015.
199. P. J. Pawar and R. V. Rao, "Parameter optimization of machining processes using teaching–learning-based optimization algorithm," *The International Journal of Advanced Manufacturing Technology,* vol. 67, no. 5-8, pp. 995-1006, 2013.
200. M. Hashish, "A Modeling Study of Metal Cutting With Abrasive Waterjets," *Journal of Engineering Materials and Technology,* vol. 106, no. 1, pp. 88-88, 1984.
201. M. Hashish, "A Model for Abrasive-Waterjet (AWJ) Machining," *Journal of Engineering Materials and Technology,* vol. 111, no. 2, pp. 154-154, 1989.
202. N. K. Jain, V. K. Jain, and K. Deb, "Optimization of process parameters of mechanical type advanced machining processes using genetic algorithms," *International Journal of Machine Tools and Manufacture,* vol. 47, no. 6, pp. 900-919, 2007.
203. R. V. Rao, P. J. Pawar, and J. P. Davim, "Optimisation of process parameters of mechanical type advanced machining processes using a simulated annealing

algorithm," *International Journal of Materials and Product Technology,* vol. 37, no. 1/2, pp. 83-83, 2010.

204. X. M. Wen, A. A. O. Tay, and A. Y. C. Nee, "Micro-computer-based optimization of the surface grinding process," *Journal of Materials Processing Technology,* vol. 29, no. 1-3, pp. 75-90, 1992.

205. R. Saravanan, P. Asokan, and M. Sachidanandam, "A multi-objective genetic algorithm (GA) approach for optimization of surface grinding operations," *International Journal of Machine Tools and Manufacture,* vol. 42, no. 12, pp. 1327-1334, 2002.

206. P. J. Pawar, R. V. Rao, and J. P. Davim, "Multiobjective optimization of grinding process parameters using particle swarm optimization algorithm," *Materials and Manufacturing Processes,* vol. 25, no. 6, pp. 424-431, 2010.

207. R. V. Rao and P. J. Pawar, "Grinding process parameter optimization using non-traditional optimization algorithms," *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture,* vol. 224, no. 6, pp. 887-898, 2010.

208. A. I. Sönmez, A. Baykasoğlu, T. Dereli, and I. H. Filiz, "Dynamic optimization of multipass milling operations via geometric programming," *International Journal of Machine Tools and Manufacture,* vol. 39, no. 2, pp. 297-320, 1999.

209. Z. G. Wang, M. Rahman, Y. S. Wong, and J. Sun, "Optimization of multi-pass milling using parallel genetic algorithm and parallel genetic simulated annealing," *International Journal of Machine Tools and Manufacture,* vol. 45, no. 15, pp. 1726-1734, 2005.

210. G. C. Onwubolu, "Performance-based optimization of multi-pass face milling operations using Tribes," *International Journal of Machine Tools and Manufacture,* vol. 46, no. 7-8, pp. 717-727, 2006.

211. R. Venkata Rao and P. J. Pawar, "Parameter optimization of a multi-pass milling process using non-traditional optimization algorithms," *Applied Soft Computing Journal,* vol. 10, no. 2, pp. 445-456, 2010.

212. R. Tanabe and A. Fukunaga, "Success-History Based Parameter Adaptation for Differential Evolution," *2013 IEEE Congress on Evolutionary Computation,* no. 3, pp. 71-78, 2013.

213. S. Rahnamayan, H. R. Tizhoosh, and M. M. A. Salama, "Opposition versus randomness in soft computing techniques," *Applied Soft Computing,* vol. 8, no. 2, pp. 906-918, 2008.

214. L. Wen, J. Cheng, F. Li, J. Zhao, Z. Shi, and H. Zhang, "Global optimization of controlled source audio-frequency magnetotelluric data with an improved artificial bee colony algorithm," *Journal of Applied Geophysics,* vol. 170, 2019.

215. S. Das, A. Abraham, U. K. Chakraborty, and A. Konar, "Differential Evolution Using a Neighborhood-Based Mutation Operator," *IEEE Transactions on Evolutionary Computation,* vol. 13, no. 3, pp. 526-553, 2009.

216. H. Wang, W. Wang, S. Xiao, Z. Cui, M. Xu, and X. Zhou, "Improving artificial Bee colony algorithm using a new neighborhood selection mechanism," *Information Sciences,* vol. 527, pp. 227-240, 2020.

LIST OF PUBLICATIONS

1) K. M. Ong, P. Ong, and C. K. Sia, "A carnivorous plant algorithm for solving global optimization problems," *Applied Soft Computing*, vol. 98, p. 106833, 2021. (ISI Indexed)
2) P. Ong, T. K. Chong, K. M. Ong, and E. S. Low, "Tracking of moving athlete from video sequences using flower pollination algorithm," *The Visual Computer,* 2021. (ISI Indexed)
3) K. M. Ong, P. Ong, C. K. Sia, and E. S. Low, "Effective moving object tracking using modified flower pollination algorithm for visible image sequences under complicated background," *Applied Soft Computing Journal,* vol. 83, pp. 105625-105625, 2019. (ISI Indexed)
4) K. M. Ong, P. Ong, C. K. Sia, E. S. Low, and W. K. Lee, "A Novel Real Time Visual Tracking Method Using Modified Flower Pollination Algorithm," *Journal of Physics: Conference Series,* vol. 1150, 2019. (Scopus Indexed)
5) K. M. Ong, P. Ong, E. S. Low, and C. K. Sia, "Robotic Arm System with Computer Vision for Colour Object Sorting," *International Journal of Engineering & Technology,* vol. 7, pp. 50-56, 2018. (Scopus Indexed)
6) K. M. Ong, P. Ong, and C. K. Sia, "Optimization of the Angle of Twist of Propeller Using Modified Flower Pollination Algorithm," in *Handbook of Research on Predictive Modeling and Optimization Methods in Science and Engineering* (Advances in Computational Intelligence and Robotics), 2018, pp. 299-327. (Book Chapter)