PEOPLE'S DEMOCRATIC REPUBLIC OF ALGERIA
MINISTRY OF HIGHER EDUCATION AND SCIENTIFIC RESEARCH

UNIVERSITY OF ECHAHID HAMMA LAKHDAR EL-OUED
FACULTY OF TECHNOLOGY
DEPARTMENT OF ELECTRICAL ENGINEERING

END-OF-STUDY DISSERTATION

Presented for obtaining the degree of

ACADEMIC MASTER

Specialty: **Telecommunications / Telecommunication Systems**

Topic

Estimation of Diode Parameters Using the Secretary Bird Optimization Algorithm

Presented by:
MELOUKA **Tahar**
BAHI **Abdelhamid**

Supervised by:
Dr. **BETKA Abir** *(Supervisor)*
Dr. **FRITAH Abdallah** *(Co-Supervisor)*

Board of Examiners:

Name & Surname	**Rank**	**Role**
Dr. Name Surname	Professor / MCA	President
Dr. BETKA Abir	MCA / MCB	Supervisor
Dr. Name Surname	MCA / MCB	Co-Supervisor
Dr. Name Surname	MCA / MCB	Examiner

Academic Year: 2025 / 2026

Dedication

To our beloved parents, whose unconditional love, endless prayers, and countless sacrifices have been our guiding light throughout our lives and academic journey. No words can ever express our gratitude to you.

To our dear brothers, sisters, and extended families, for their continuous encouragement and unwavering support.

To our friends and colleagues, for the shared moments, the mutual support, and the beautiful memories we made together during these years of study.

To everyone who believed in us and supported us, near or far, we dedicate this work.

Acknowledgements

First and foremost, all praise and immense gratitude belong to Allah, the Almighty, for blessing us with the health, strength, and patience to successfully complete this work.

We would like to express our deepest and most sincere gratitude to our supervisor, **Dr. BETKA Abir**, for her constant guidance, valuable advice, and unwavering support throughout this research. Her expertise, patience, and encouragement were instrumental in the realization of this dissertation.

A special word of thanks goes to our co-supervisor, **Dr. FRITAH Abdallah**, for his invaluable assistance, technical support, and expert guidance, particularly regarding the MATLAB implementations. His contributions significantly enhanced the quality of this work.

We extend our respectful thanks to the honorable members of the examining committee for accepting to review and evaluate our work. Their insightful comments and constructive feedback will undoubtedly enrich the quality of this research.

Our sincere appreciation also goes to all the teachers and staff of the Department of Electrical Engineering at the University of Echahid Hamma Lakhdar - El Oued, who have contributed to our academic training throughout our years of study.

Finally, we would like to thank our colleagues, friends, and everyone who provided us with help, whether directly or indirectly, in making this journey a success.

Abstract

The accurate characterization of semiconductor devices is crucial for optimizing modern electronic and optoelectronic systems. Extracting the precise electrical parameters of Schottky diodes from experimental Current-Voltage (I-V) measurements constitutes a highly non-linear and complex optimization problem. Traditional analytical methods often fail to capture the physical realities of these devices, particularly in the presence of highly resistive interfacial layers, leading to inaccurate modeling. To overcome these limitations, this thesis proposes the application of a novel and robust metaheuristic approach: the Secretary Bird Optimization Algorithm (SBOA). The computational efficiency and stability of the SBOA were first rigorously validated against a challenging set of standard benchmark test functions, demonstrating an exceptional balance between global exploration and local exploitation. Subsequently, the algorithm was practically applied to extract the fundamental parameters namely, the zero-bias barrier height (Φ_{B0}), series resistance (R_s), and ideality factor (n) of a standard Metal-Semiconductor (MS) diode and a Metal-Insulator-Semiconductor (MIS) diode with a TiO_2 interfacial layer. By minimizing the Sum of Absolute Errors (SAE) against recent authentic experimental datasets, the SBOA yielded highly accurate simulated models. The extracted parameters revealed profound physical insights, successfully quantifying the passivation effect of the TiO_2 layer, which increased the barrier height from 0.7531 eV to 0.8375 eV while mathematically translating the significant structural resistance.

Keywords: Metaheuristics, Secretary Bird Optimization Algorithm (SBOA), Schottky Diode, Parameter Extraction, Metal-Semiconductor (MS) and Metal-Insulator-Semiconductor (MIS) Structures, Optimization.

Résumé

La caractérisation précise des dispositifs semi-conducteurs est cruciale pour l'optimisation des systèmes électroniques et optoélectroniques modernes. L'extraction des paramètres électriques précis des diodes Schottky à partir des mesures expérimentales Courant-Tension (I-V) constitue un problème d'optimisation hautement non linéaire et complexe. Les méthodes analytiques traditionnelles échouent souvent à capturer les réalités physiques de ces dispositifs, en particulier en présence de couches interfaciales hautement résistives, conduisant à une modélisation inexacte. Pour surmonter ces limites, cette thèse propose l'application d'une approche métaheuristique nouvelle et robuste : l'Algorithme d'Optimisation du Secrétaire (SBOA). L'efficacité calculatoire et la stabilité du SBOA ont d'abord été rigoureusement validées sur un ensemble complexe de fonctions tests standards, démontrant un équilibre exceptionnel entre l'exploration globale et l'exploitation locale. Par la suite, l'algorithme a été appliqué pour extraire les paramètres fondamentaux à savoir, la hauteur de barrière à polarisation nulle (Φ_{B0}), la résistance série (R_s) et le facteur d'idéalité (n) d'une diode Métal-Semi-conducteur (MS) standard et d'une diode Métal-Isolant-Semi-conducteur (MIS) avec une couche interfaciale de TiO_2. En minimisant la Somme des Erreurs Absolues (SAE) par rapport à des ensembles de données expérimentales authentiques, le SBOA a produit des modèles simulés très précis. Les paramètres extraits ont révélé des informations physiques profondes, quantifiant avec succès l'effet de passivation de la couche de TiO_2, qui a augmenté la hauteur de barrière de 0.7531 eV à 0.8375 eV tout en traduisant mathématiquement la résistance structurelle significative.

Mots-clés : Métaheuristiques, Algorithme d'Optimisation du Secrétaire (SBOA), Diode Schottky, Extraction de paramètres, Structures Métal-Semi-conducteur (MS) et Métal-Insulator-Semiconductor (MIS), Optimisation.

ملخص

يعد التوصيف الدقيق لأجهزة أشباه الموصلات أمراً بالغ الأهمية لتحسين الأنظمة الإلكترونية والإلكترونيات الضوئية الحديثة. يمثل استخراج المعلمات الكهربائية الدقيقة لثنائيات شوتكي (Schottky diodes) من القياسات التجريبية للتيار-الجهد ($I-V$) مشكلة تحسين معقدة وغير خطية إلى حد كبير. غالباً ما تفشل الطرق التحليلية التقليدية في التقاط الحقائق الفيزيائية لهذه الأجهزة، لا سيما في وجود طبقات بينية عالية المقاومة، مما يؤدي إلى نمذجة غير دقيقة. للتغلب على هذه القيود، تقترح هذه المذكرة تطبيق نهج ذكي (Metaheuristic) جديد وقوي: خوارزمية تحسين طائر الكاتب (SBOA). تم التحقق أولاً من الكفاءة الحسابية والاستقرار لخوارزمية (SBOA) بشكل صارم مقابل مجموعة صعبة من دوال الاختبار القياسية، مما يدل على توازن استثنائي بين الاستكشاف العالمي والاستغلال المحلي. بعد ذلك، تم تطبيق الخوارزمية عملياً لاستخراج المعلمات الأساسية وهي ارتفاع الحاجز الصفري (Φ_{B0})، والمقاومة التسلسلية (R_s)، وعامل المثالية (n) لثنائي معدن-شبه موصل (MS) قياسي، وثنائي معدن-عازل-شبه موصل (MIS) بطبقة عازلة من ثاني أكسيد التيتانيوم ($\mathrm{TiO_2}$). من خلال تقليل مجموع الأخطاء المطلقة (SAE) مقابل بيانات تجريبية حقيقية وحديثة، أنتجت خوارزمية (SBOA) نماذج محاكاة دقيقة للغاية. كشفت المعلمات المستخرجة عن رؤى فيزيائية عميقة، حيث نجحت في تحديد تأثير التخميل للطبقة العازلة، والذي زاد من ارتفاع الحاجز من 0.7531 إلكترون فولت إلى 0.8375 إلكترون فولت، مع الترجمة الرياضية للمقاومة الهيكلية الكبيرة.

الكلمات المفتاحية: الخوارزميات الذكية، خوارزمية تحسين طائر الكاتب (SBOA)، ثنائي شوتكي، استخراج المعلمات، بنيات معدن-شبه موصل (MS) ومعدن-عازل-شبه موصل (MIS)، التحسين.

Contents

List of Figures

List of Tables

List of Abbreviations

ACO	Ant Colony Optimization
CEC	Congress on Evolutionary Computation
GA	Genetic Algorithm
GP	Geometric Programming
I-V	Current-Voltage
LP	Linear Programming
MIS	Metal-Insulator-Semiconductor
MS	Metal-Semiconductor
NLP	Nonlinear Programming
PSO	Particle Swarm Optimization
QP	Quadratic Programming
SA	Simulated Annealing
SAE	Sum of Absolute Errors
SBD	Schottky Barrier Diode
SBOA	Secretary Bird Optimization Algorithm
TE	Thermionic Emission
TS	Tabu Search
TSP	Traveling Salesman Problem
VRP	Vehicle Routing Problem

List of Symbols

Symbol	Description	Unit
A	Diode effective area	cm^2
A^*	Effective Richardson constant	$A\ K^{-2}cm^{-2}$
D (Dim)	Number of dimensions of the search space	-
$f(x)$	Objective (fitness) function	-
I	Diode current	A
I_0	Reverse saturation current	A
I_{calc}	Calculated theoretical current	A
I_{exp}	Experimental measured current	A
k	Boltzmann constant (1.380×10^{-23})	J/K
LB, UB	Lower and upper bounds of variables	-
n	Ideality factor	-
N	Population size (number of search agents)	-
q	Electron charge (1.602×10^{-19})	C
R_s	Series resistance	Ω
t	Current iteration	-
T	Absolute temperature	K
T_{max}	Maximum number of iterations	-
V	Applied voltage	V
x_{best}	Global best candidate solution	-
X_i	Position vector of the i-th search agent	-
Φ_{B0}	Zero-bias barrier height	eV

General Introduction

The rapid advancement of modern electronics and optoelectronic systems relies heavily on the continuous improvement of semiconductor devices. Among these, Schottky barrier diodes stand out as fundamental components in various applications, ranging from solar cells to high-frequency switching devices. The integration of interfacial insulator layers, forming Metal-Insulator-Semiconductor (MIS) structures, further enhances their electrical and dielectric performance by modulating barrier heights and suppressing leakage currents. However, fully leveraging the capabilities of these advanced devices requires an accurate understanding of their physical behavior through precise mathematical modeling [1].

Characterizing these devices entails extracting crucial electrical parameters namely, the zero-bias barrier height (Φ_{B0}), series resistance (R_s), and ideality factor (n) from experimental Current-Voltage ($I-V$) measurements [1]. This extraction process poses a highly non-linear and transcendental optimization problem. Traditional analytical techniques, which fundamentally rely on linear approximations, often fail to capture the complex physical realities of the diodes, particularly in the presence of high structural resistance or insulating layers [2]. Consequently, they are prone to yielding inaccurate parameters and falling into local minima, failing to reflect the true conduction mechanisms.

To overcome the severe limitations of classical mathematical approaches, metaheuristic optimization algorithms have emerged as powerful, robust, and intelligent alternatives. By mimicking biological behaviors and natural processes, these approximate techniques can efficiently navigate complex, multidimensional search spaces to find optimal solutions. In this context, this thesis proposes the application of a recently developed and highly effective metaheuristic: the Secretary Bird Optimization Algorithm (SBOA). Inspired by the unique hunting and evasion strategies of the secretary bird, SBOA presents a solid mathematical framework that perfectly balances global exploration and local exploitation [3].

The primary objective of this thesis is to rigorously evaluate the computational efficiency of SBOA and subsequently apply it to the complex engineering problem of Schottky diode parameter extraction. To ensure algorithmic reliability, SBOA is first validated against a challenging set of standard benchmark test functions. Following this mathematical validation, the algorithm is deployed to extract the exact parameters of both a standard Metal-Semiconductor (MS) diode and a TiO_2-based MIS diode. To guarantee the utmost physical accuracy, the optimization is performed utilizing authentic, recently measured experimental datasets.

To systematically achieve these objectives, the present manuscript is organized into three main chapters as follows:

- **Chapter I** establishes the theoretical background of optimization problems. It provides a comprehensive overview of metaheuristic classifications and highlights their superiority over exact methods in solving highly non-linear engineering challenges.
- **Chapter II** presents a detailed dissection of the Secretary Bird Optimization Algorithm (SBOA). It explains the biological inspiration behind the algorithm and formulates its mathematical models, specifically detailing the exploration, exploitation, and local minima avoidance mechanisms.
- **Chapter III** focuses on the practical implementation, evaluation, and physical application. It first assesses the SBOA performance on continuous benchmark functions, followed by the extraction of the MS and MIS Schottky diode parameters, concluding with a comparative physical interpretation of the obtained results.

Chapter I

Introduction to Optimization and Metaheuristics

Contents

I.1 Introduction

An optimization problem is a computational task that involves identifying the optimal solution among all possible alternatives. Specifically, it aims to determine a feasible solution that either minimizes or maximizes the value of a given objective function [4–6]. Traditional optimization approaches frequently prove inadequate when tackling intricate problems.

Metaheuristic algorithms emerge as a powerful alternative to overcome this limitation. As flexible optimization methodologies, these techniques deliver effective solutions for complex problem-solving scenarios [7, 8]. Inspired by natural processes or heuristic strategies, metaheuristics demonstrate superior capabilities in both exploring and exploiting the solution space, rendering them applicable across diverse domains. The evolution of metaheuristic algorithms has been shaped by progress in multiple disciplines, encompassing computer science, operations research, numerical analysis, game theory, mathematical economics, control theory, and combinatorial mathematics [7, 9].

This chapter explores optimization problems and techniques, including metaheuristic methods that mimic natural processes. By establishing this theoretical foundation, the chapter paves the way for introducing advanced optimization strategies such as the Secretary Bird Optimization Algorithm (SBOA) which will be utilized for the complex task of non-linear parameter estimation in semiconductor devices [2, 10].

I.2 Definition of an Optimization Problem

Optimization has always been an inherent part of daily life since ancient times. To cope with constraints, individuals must devise solutions that adhere to resource limitations while navigating uncertainties and incomplete knowledge of certain problems [4, 6]. Many solutions initially stem from observations of nature, and over time, knowledge evolves by building upon these natural observations and accumulated past experiences. In daily practice, existing solutions are continually tested and may be replaced by more efficient alternatives to enhance overall problem-solving performance.

From a technical perspective, the structural properties of the objective function and constraints with respect to the decision variables determine the computational tractability of an optimization problem [4, 5]. These properties directly influence both the choice of solution algorithms and the guarantee of reaching a true optimal solution.

When addressing real-world engineering and scientific problems, researchers frequently employ single-objective optimization. However, a more effective approach to complex problem-solving can often be achieved by incorporating multiple objectives. Multi-objective optimization involves simultaneously optimizing two or more conflicting objectives while strictly satisfying given constraints. In this specific context, an optimal solution is defined as one that is not dominated by any other solution within the entire search space, commonly known as a Pareto optimal solution [4, 6].

I.3 Statement of an Optimization Problem

The optimization problem involves determining the optimal values of the decision variables that either minimize or maximize the objective function, subject to the given constraints [4, 5]. An optimization problem can be stated as follows:

Find:

$$X = [x_1, x_2, \dots, x_a]^T \tag{I.1}$$

Which minimizes:

$$f(X) \tag{I.2}$$

Subject to the constraints:

$$g_j(x) \leq 0, \quad j = 1, 2, \dots, m \tag{I.3}$$

$$h_j(x) = 0, \quad j = 1, 2, \dots, p \tag{I.4}$$

Here, X represents an a-dimensional vector referred to as the design vector, $f(X)$ is defined as the objective function, and $g_j(x)$ along with $h_j(x)$ are termed the inequality constraints and equality constraints, respectively. Note that the number of variables (a) and the number of constraints (m) for inequalities and (p) for equalities are not inherently dependent on one another [4].

The problem presented in the equations above is referred to as a constrained optimization problem. In contrast, some optimization problems involve no constraints and can be formulated as:

Find:

$$X = [x_1, x_2, \dots, x_a]^T \tag{I.5}$$

Which minimizes:

$$f(X) \tag{I.6}$$

Such problems are called unconstrained optimization problems [4, 6].

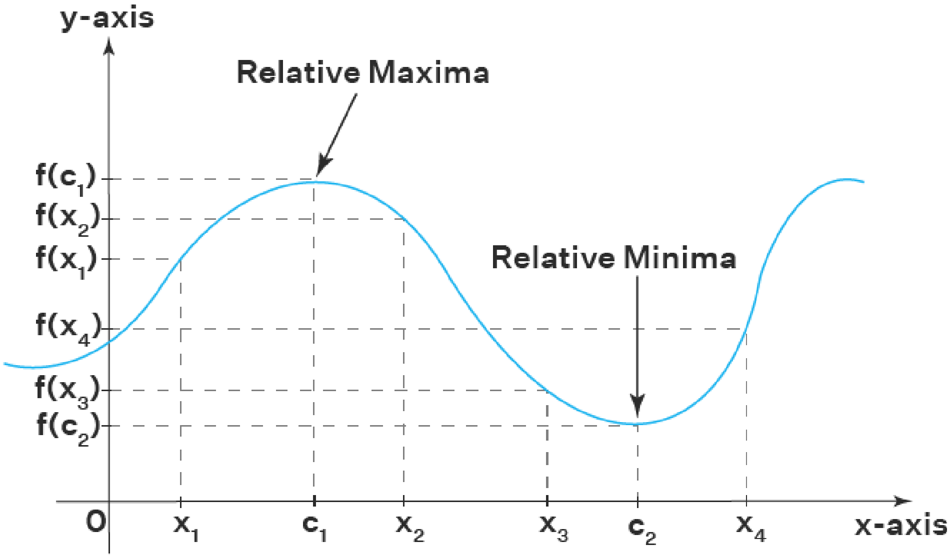


Figure I.1: Strong and weak Relative maxima and minima [11]

Figure I.1 illustrates several local maxima and minima. Point A represents a strict (strong) local maximum, whereas Point B is a weak local maximum due to the existence of infinitely many distinct values of x that yield the same function value $f(x^*)$. Point D corresponds to the global maximum. On the other hand, Point C is a strict local minimum, yet it exhibits a discontinuity in the derivative $f'(x^*)$, meaning the stationary condition does not hold smoothly at this point [4].

The condition $f'(x^*) = 0$ does not hold in this case. Therefore, we will not examine this type of minimum or maximum in depth. For our current discussion, we will assume that both $f(x)$ and $f'(x)$ are always continuous, or equivalently, that $f(x)$ is twice continuously differentiable everywhere [4].

I.4 Classification of Optimization Problems

There are multiple classification methods for optimization problems, depending on the mathematical characteristics, the physical structure, and the nature of the variables involved [4,6]. An essential first step for any designer or engineer is to determine the problem's classification, as this often dictates the appropriate solution techniques to employ. The primary classifications are detailed below:

I.4.1 Classification Based on the Existence of Constraints

Optimization problems can be categorized as either constrained or unconstrained, based on the presence or absence of constraints in the problem formulation [4].

- **Constrained Optimization:** The objective function is minimized or maximized subject to specific limits (bounds) on the decision variables or relationships between them. For instance, in diode parameter estimation, the physical parameters (like the ideality factor or saturation current) must remain within specific practical bounds.

- **Unconstrained Optimization:** The problem seeks the absolute minimum or maximum of a function without any restrictions on the variables.

I.4.2 Classification Based on the Nature of the Design Variables

Optimization problems can be broadly classified into two main types based on the nature of their design variables [4]:

- **Parameter Optimization Problems:** In this category, the goal is to determine the optimal values of a set of static design parameters that minimize (or maximize) a given objective function while satisfying specified constraints. Here, the design variables are typically discrete or continuous fixed parameters. The extraction of diode parameters using SBOA falls strictly into this category.

- **Functional (Trajectory) Optimization Problems:** This category involves finding optimal design parameters that are themselves continuous functions of another independent variable (such as time). The objective is to minimize a functional (a function of functions).

I.4.3 Classification Based on the Physical Structure of the Problem

This classification method groups optimization problems according to their inherent physical structure, distinguishing categories such as network optimization problems, optimal control problems, graph-based problems, or permutation-related problems. Each belongs to a distinct structural class requiring specialized solving algorithms [4].

I.4.4 Classification Based on the Nature of the Equations Involved

An essential classification is based on the mathematical expressions of the objective function and constraints. According to this criterion, optimization problems can be categorized into [4]:

- **Linear Programming (LP):** Both the objective function and all constraints are linear functions of the design variables.

- **Nonlinear Programming (NLP):** Either the objective function, the constraints, or both are nonlinear. Because the current-voltage (I-V) characteristics of a semiconductor diode are highly exponential and nonlinear, diode parameter estimation is classified as a complex Nonlinear Programming problem [2, 4, 6].

- **Geometric Programming (GP) and Quadratic Programming (QP):** Specialized subclasses of nonlinear programming where functions take specific polynomial or quadratic forms.

I.4.5 Classification Based on the Deterministic Nature of the Variables

Depending on the certainty of the parameters, problems are categorized into two main types [4]:

- **Deterministic Programming Problems:** All variables and parameters are known with absolute certainty and precisely defined, with no randomness involved. They can be modeled using fixed values.
- **Stochastic (Probabilistic) Programming Problems:** In these problems, some or all parameters (design variables and/or preassigned parameters) are subject to uncertainty and are defined by probability distributions.

I.4.6 Classification Based on the Separability of the Functions

The classification into separable and non-separable types is determined by whether the objective and constraint functions can be separated into a sum of individual functions, each depending on only one design variable [4].

I.4.7 Classification Based on the Number of Objective Functions

Depending on how many goals need to be achieved simultaneously, problems are classified as [4]:

- **Single-Objective Optimization:** Only one objective function needs to be minimized or maximized (e.g., minimizing the Root Mean Square Error between experimental and simulated diode currents).
- **Multi-Objective Optimization:** Involves optimizing two or more conflicting objectives simultaneously, requiring a trade-off approach to find a set of Pareto optimal solutions.

I.5 Optimization Techniques

Optimization methods designed to solve various types of mathematical and engineering problems fall under the broad domain of mathematical optimization. The selection of an appropriate optimization technique depends heavily on the problem's complexity, the nature of the objective function, and the available computational resources. These techniques can be broadly classified into two primary categories: exact methods and approximate (heuristic) methods [4, 7].

I.5.1 Exact Techniques

Exact methods are rigorous mathematical techniques employed to solve optimization problems by identifying the globally optimal solution without resorting to approximations or heuristic assumptions [4]. Unlike empirical approaches that depend on statistical estimations, exact methods theoretically guarantee the determination of the true global optimal solution. These methods comprise diverse algorithmic strategies, such as linear programming for continuous variables, integer programming for discrete variables, and branch and bound algorithms.

However, these traditional methods have significant limitations. They often involve exceptionally high computational complexity. As the dimensionality and non-linearity of the problem increase such as in the case of estimating the highly nonlinear I-V parameters of a diode exact methods frequently face a combinatorial explosion. This mathematical burden makes them highly impractical, excessively time-consuming, and sometimes impossible to implement for extremely large scale or highly intricate engineering problems [4, 5].

I.5.2 Approximate Techniques

To overcome the severe computational limitations of exact methods, approximation techniques commonly known as heuristic or metaheuristic methods are extensively utilized [7]. These techniques become essential when a problem's complexity makes an exhaustive or derivative based search for all possible solutions impractical or unachievable.

While these approximate methods do not mathematically guarantee the discovery of the absolute global optimal solution, they are highly effective at providing computationally feasible, high-quality, and completely satisfactory solutions within a reasonable time frame. They typically rely on intelligently guided search strategies, incorporating probabilistic approaches, heuristic rules, or evolutionary mechanisms to explore the solution space efficiently while avoiding stagnation in local optima. Such techniques are particularly valuable and widely adopted for addressing complex, large-scale optimization problems involving numerous variables, making them the standard choice for modern semiconductor parameter extraction [7, 10].

I.6 The Metaheuristics

Metaheuristic algorithms are high-level, problem-independent algorithmic frameworks that provide a set of guidelines or strategies to develop heuristic optimization algorithms [7]. They have gained immense popularity in engineering applications, particularly for complex, highly non-linear problems like semiconductor parameter extraction, where exact mathematical methods systematically fail.

Unlike traditional gradient-based techniques, metaheuristics do not require the objective function to be continuous or differentiable. Instead, they employ stochastic operators to intelligently explore the search space. A key characteristic of a well-designed metaheuristic such as the Secretary Bird Optimization Algorithm (SBOA) is its ability to maintain a delicate balance between "exploration" (the global search of the solution space to avoid trapping in local optima) and "exploitation" (the local search to intensely refine and improve the current best solutions) [7, 8]. These methods frequently draw profound inspiration from biological evolution, physical phenomena, or the collective behavior of animals, offering robust and globally optimal (or near-optimal) solutions within acceptable computational times.

I.7 Classification of Metaheuristics

The vast landscape of metaheuristic algorithms can be systematically classified based on their underlying mathematical mechanisms, search dynamics, and sources of conceptual inspiration. The most prominent classifications utilized in the scientific literature include [7, 8]:

I.7.1 Nature-Inspired vs. Non-Nature-Inspired

Algorithms can be distinguished fundamentally based on their conceptual origins. Nature-inspired algorithms mimic natural, biological, or ethological phenomena. This massive and highly successful category includes Evolutionary Algorithms (e.g., Genetic Algorithms) and Swarm Intelligence (e.g., Particle Swarm Optimization, and SBOA, which mimics the intelligent hunting strategy and survival instincts of the secretary bird) [7]. Conversely, non-nature-inspired methods rely strictly on theoretical, memory-based, or physical heuristics, such as Tabu Search or Iterated Local Search.

I.7.2 Population-Based vs. Single-Solution (Trajectory-Based)

This classification emphasizes the number of candidate solutions processed simultaneously. Population-based methods initialize and iteratively evolve a diverse set of candidate solutions. This collective approach allows for extensive global exploration and continuous information sharing among individuals (e.g., ACO, PSO, SBOA). Conversely, single-solution (or trajectory-based) methods track and modify only one candidate solution that moves through the search space in a continuous trajectory, deeply exploiting local neighborhoods (e.g., Hill Climbing and Simulated Annealing) [7, 8].

I.7.3 Deterministic vs. Stochastic

This classification is based on the nature of the transition rules used during the optimization process. Deterministic metaheuristics make decisions based on fixed, rigid mathematical rules, meaning the exact same initial conditions will always yield identical final results. In stark contrast, stochastic metaheuristics introduce a carefully controlled degree of randomness into their search operators [7]. This stochastic nature ensures diverse search paths across different executions and significantly reduces the risk of premature convergence when navigating complex, multidimensional landscapes, such as a diode's error surface.

I.8 Prominent Metaheuristic Techniques

Numerous metaheuristic algorithms have been developed over the years, each drawing inspiration from different natural, physical, or biological phenomena. Each algorithm possesses distinct mathematical strengths in balancing exploration and exploitation [7]. Some of the most foundational and widely applied metaheuristic algorithms in the scientific literature include:

- **Genetic Algorithms (GA):** GA is a highly popular evolutionary algorithm inspired by the principles of natural selection and genetics. The optimization process begins by generating an initial population of candidate solutions randomly. The algorithm then iteratively evolves this population by applying biological operators such as selection (choosing the fittest individuals), crossover (recombination of traits), and mutation (introducing random variations) to find the global optimum [12].

- **Particle Swarm Optimization (PSO):** PSO is a fundamental swarm intelligence algorithm that draws inspiration from the highly coordinated group behaviors of social creatures, such as fish schools and flocks of birds. A swarm of particles, each representing a potential solution to the optimization problem, is generated randomly. These particles mathematically update their positions and velocities in the multidimensional search space based on their own best-known position and the global best position found by the entire swarm [13].

- **Simulated Annealing (SA):** SA is a stochastic optimization approach that draws its conceptual inspiration from the metallurgical process of annealing, where a material is heated and then slowly cooled to alter its physical properties and minimize its internal energy. In the algorithm, a simulated ”temperature” parameter is gradually lowered, allowing the system to initially accept worse solutions to escape local minima, before fine-tuning the search as the temperature decreases [14].

- **Tabu Search (TS):** TS is a memory-based trajectory metaheuristic that draws inspiration from human problem-solving and memory structures. It uses an iterative local search procedure to move from one potential solution to an improved one. To prevent the algorithm from returning to recently visited solutions and getting trapped in cycles, it maintains a ”tabu list” (a short-term memory) of forbidden moves [15].

- **Ant Colony Optimization (ACO):** ACO is a swarm intelligence method based on how real ants navigate their environment to find the absolute shortest path between their colony and a food source. The algorithm uses artificial chemical trails (pheromones) that are deposited and dynamically updated by virtual ants to indicate the quality of the paths (solutions) found, guiding the rest of the colony toward the optimal solution [16].

While these classical algorithms established the foundation of metaheuristic optimization, the continuous demand for faster convergence and higher accuracy in highly nonlinear problems such as semiconductor parameter extraction has driven the development of novel, advanced swarm-based approaches, including the Secretary Bird Optimization Algorithm (SBOA).

I.9 Applications of Metaheuristics

Metaheuristics have been widely and successfully applied to various domains and problem types due to their extreme flexibility and ability to find near-optimal solutions in reasonable computational times [7]. Some critical applications include:

- **Engineering Design and Parameter Estimation:** This is a crucial domain where metaheuristic techniques are extensively used to optimize physical structures, design telecommunication circuits, and estimate the highly nonlinear mathematical parameters of semiconductor devices (such as diodes and transistors) by minimizing the error between experimental data and simulation models [17].

- **Combinatorial Optimization:** Metaheuristic algorithms are the standard approach for solving complex combinatorial problems such as the Traveling Salesman

Problem (TSP), Vehicle Routing Problem (VRP), and complex network design in telecommunications [18].

- **Machine Learning and Data Mining:** Metaheuristic algorithms are increasingly utilized for optimizing neural network architectures, hyperparameter tuning, and optimal feature selection in vast datasets [19].

- **Signal and Image Processing:** These algorithms find significant applications in complex signal processing tasks relevant to wireless communications, such as optimal filter design, noise reduction, image reconstruction, and pattern recognition [7].

- **Scheduling and Timetabling:** They are employed to perfectly optimize resource allocation, employee scheduling, production planning, and optimal routing in wireless sensor networks [18].

- **Energy Systems Optimization:** Metaheuristics assist in optimizing power generation, minimizing transmission losses, and integrating renewable energy sources efficiently into smart grids [20].

I.10 Conclusion

This chapter provided a comprehensive overview of optimization problems, establishing the fundamental mathematical concepts and classifications required to understand complex engineering challenges [4, 6]. While exact optimization techniques offer theoretical guarantees of finding global optima, their severe computational limitations render them impractical for highly nonlinear, multidimensional problems. Consequently, metaheuristic algorithms have emerged as the most viable and efficient alternative [7]. By mimicking natural processes and biological behaviors, these approximate techniques successfully maintain a critical balance between global exploration and local exploitation, making them indispensable for modern engineering design.

Having established this theoretical foundation, the following chapter will introduce a novel and advanced metaheuristic approach the Secretary Bird Optimization Algorithm (SBOA). It will detail its unique hunting-inspired mechanisms and demonstrate its effectiveness in addressing complex, non-linear optimization tasks, providing a robust framework for the practical applications discussed later in this study.

Chapter II

The Secretary Bird Optimization Algorithm (SBOA)

Contents

II.1 Introduction

While numerous metaheuristic algorithms have been developed to tackle complex engineering problems, many still suffer from persistent limitations. Issues such as slow convergence rates, limited search precision, and a susceptibility to becoming trapped in local optima remain significant challenges for researchers [21, 22]. To overcome these specific shortcomings and provide a more efficient solution for engineering domains, a novel population-based metaheuristic known as the Secretary Bird Optimization Algorithm (SBOA) was recently introduced [3].

SBOA draws its inspiration directly from the unique survival capabilities and intelligent activities of the Secretary Bird in its natural environment. The core of the algorithm models the bird's behavioral adaptability, logically dividing the optimization search process into two primary phases. The exploration phase mathematically simulates the bird's distinctive strategy of hunting snakes, whereas the exploitation phase models its tactical behavior when evading predators.

In the design of optimization algorithms, achieving a careful balance between "exploration" and "exploitation" is crucial for demonstrating superior performance [23, 24]. By mimicking natural survival strategies, SBOA effectively maintains this balance. This ensures an enhanced optimization precision and a strong capability to avoid local optima, making it a highly promising and robust algorithm for solving complex, real-world engineering optimization problems [3].

II.2 Secretary Bird Optimization Algorithm Inspiration and Behavior

The Secretary Bird Optimization Algorithm (SBOA) is a novel population-based metaheuristic inspired by the distinctive survival behaviors of the Secretary Bird (*Sagittarius serpentarius*). This African raptor is widely distributed in the grasslands and savannas south of the Sahara Desert. It is characterized by its unique physical appearance, featuring grey-brown plumage on its back and wings, a pure white chest, and a deep black belly [25, 26].

Figure II.1: The Secretary Bird [27].

The core inspiration of SBOA lies in the bird's dual survival strategies: hunting prey and evading predators. Unlike most raptors, the Secretary Bird is renowned for its terrestrial hunting style, using its long and sturdy legs to run and strike prey on the ground [28]. It typically traverses grasslands by walking or trotting, often bowing its head to scan the ground for hidden prey, such as insects, reptiles, and small mammals.

The algorithm mimics two primary behavioral phases:

- **Exploration Phase:** Models the bird's intelligent strategy when hunting snakes, where it monitors the prey's moves and delivers precise strikes.
- **Exploitation Phase:** Models the bird's tactics for evading predators, which include camouflaging itself in the environment or opting for rapid flight and walking to reach safety [26].

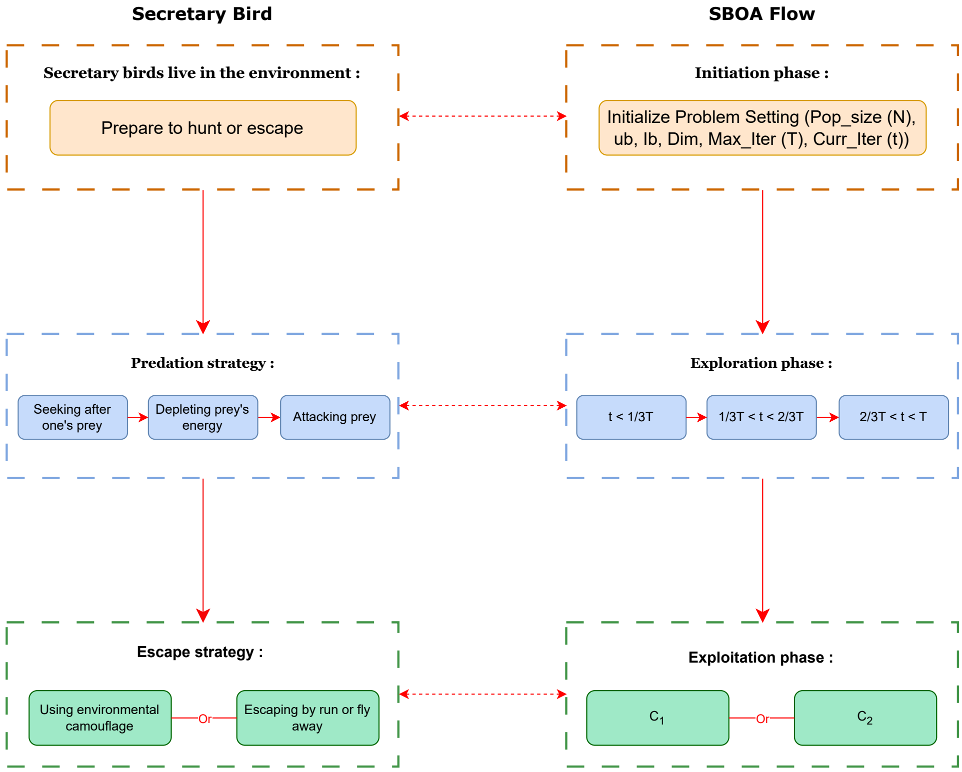


Figure II.2: Secretary bird hunting and escape strategies correspond to SBOA.

II.3 Mathematical Modeling of SBOA

The mathematical framework of the Secretary Bird Optimization Algorithm (SBOA) is designed to transition from biological observation to a computational search process. This section details the three fundamental stages of the algorithm: the initial preparation, the exploration phase inspired by hunting, and the exploitation phase inspired by predator evasion [3].

II.3.1 Initial Preparation Phase

The Secretary Bird Optimization Algorithm (SBOA) is classified as a population-based metaheuristic technique. In this approach, each individual Secretary Bird acts as a member of the algorithm's population. The specific position of a bird within the defined search space corresponds to the values of the problem's decision variables, making these positions the candidate solutions for the optimization problem.

During the initial execution of SBOA, the starting positions of the birds are generated randomly across the search space using the following mathematical formulation [3]:

$$X_{i,j} = lb_j + r \times (ub_j - lb_j), \quad i = 1, 2, \ldots, N, \quad j = 1, 2, \ldots, Dim \tag{II.1}$$

where:

- $X_{i,j}$ represents the position of the i^{th} secretary bird in the j^{th} dimension.
- lb_j and ub_j denote the lower and upper boundary limits of the search space, respectively.
- r is a uniformly distributed random numerical value drawn from the interval $[0, 1]$.

The entire population of candidate solutions, initialized within these boundaries, can be structured as a matrix X [3]:

$$X = \begin{bmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,j} & \cdots & x_{1,Dim} \\ x_{2,1} & x_{2,2} & \cdots & x_{2,j} & \cdots & x_{2,Dim} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ x_{i,1} & x_{i,2} & \cdots & x_{i,j} & \cdots & x_{i,Dim} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ x_{N,1} & x_{N,2} & \cdots & x_{N,j} & \cdots & x_{N,Dim} \end{bmatrix}_{N\times Dim} \tag{II.2}$$

where:

- X designates the complete population matrix of the secretary birds.
- X_i represents the position vector of the i^{th} secretary bird (candidate solution).
- $x_{i,j}$ specifies the value of the j^{th} decision variable for the i^{th} candidate solution.
- N indicates the total number of individuals in the population (population size).
- Dim represents the total number of dimensions (variables) in the optimization problem.

Since each bird proposes a candidate solution, its quality must be evaluated using an objective function. The evaluated fitness scores for all individuals in the population are aggregated into a single vector F as shown below [3]:

$$F = \begin{bmatrix} F_1 \\ \vdots \\ F_i \\ \vdots \\ F_N \end{bmatrix}_{N\times 1} = \begin{bmatrix} F(X_1) \\ \vdots \\ F(X_i) \\ \vdots \\ F(X_N) \end{bmatrix}_{N\times 1} \tag{II.3}$$

where:

- F stands for the comprehensive vector containing all objective function values.

- F_i (or $F(X_i)$) is the calculated fitness value specifically for the i^{th} secretary bird.

By evaluating and comparing these fitness values, the algorithm identifies the optimal candidate solution for the current iteration (the minimum value for minimization problems, or the maximum for maximization problems) [3].

Finally, to dynamically update these positions in subsequent iterations, SBOA utilizes two primary natural behaviors of the bird [3]:

(a) The hunting strategy (Exploration).

(b) The escape strategy (Exploitation).

II.3.2 Exploration Phase: Hunting Strategy of Secretary Bird

The hunting behavior of the Secretary Bird, particularly when preying on snakes, is fundamentally categorized into three sequential stages: searching for the prey, consuming (or interacting with) the prey, and finally attacking the prey. A schematic representation of this biological hunting sequence is provided in Figure II.3.

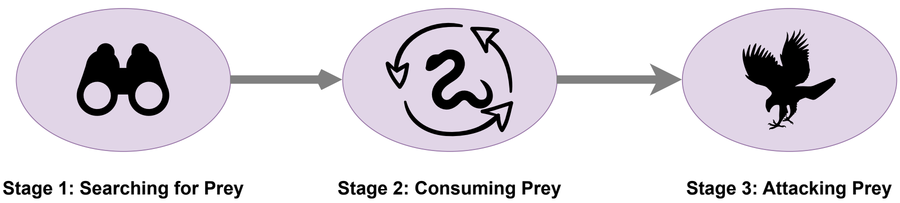


Figure II.3: The hunting behavior of secretary birds.

Relying on the biological observation of the bird's hunting phases and their respective durations, the SBOA algorithm divides the total optimization process into three equal temporal intervals. Specifically, these intervals are defined as $t < \frac{1}{3}T$, $\frac{1}{3}T < t < \frac{2}{3}T$, and $\frac{2}{3}T < t < T$. These periods mathematically correspond to the three phases of the secretary bird's predation: searching for prey, consuming prey, and attacking prey, respectively. Therefore, the mathematical modeling of each phase is detailed as follows [3]:

Stage 1: Searching for Prey

The hunting process of secretary birds typically begins with the search for potential prey, especially snakes. In nature, Secretary Birds utilize their exceptionally sharp vision to spot hidden snakes in the tall grass of the savannah. They employ their long legs to cautiously sweep the ground, keeping a relatively safe distance to avoid sudden snake attacks.

In the context of the optimization algorithm, this biological caution and extensive searching translate to the initial iterations where broad exploration of the search space is absolutely crucial. To mathematically model this extensive exploration, SBOA adopts a differential evolution strategy. This strategy leverages the positional differences between randomly selected individuals in the population to generate new, diverse candidate solutions. By integrating these differential mutation operations, the algorithm successfully maintains population diversity, prevents premature convergence to local optima, and ensures that different regions of the solution space are thoroughly explored, thereby increasing the chances of finding the global optimum [3].

Consequently, the mathematical updating mechanism for the bird's position during this "Searching for Prey" stage is formulated using Equations (4) and (5) [3]:

$$\text{While } t < \frac{1}{3}T, \quad x_{i,j}^{new,P1} = x_{i,j} + (x_{random\_1} - x_{random\_2}) \times R_1 \tag{II.4}$$

$$X_i = \begin{cases} X_i^{new,P1}, & \text{if } F_i^{new,P1} < F_i \\ X_i, & \text{else} \end{cases} \tag{II.5}$$

where:

- t represents the current iteration number in the optimization process.
- T denotes the maximum number of iterations defined for the algorithm.
- $X_i^{new,P1}$ indicates the newly generated state (or position vector) of the i^{th} secretary bird during this first hunting stage.
- x_{random_1} and x_{random_2} are two distinct random candidate solutions selected from the population during the first stage iteration, used to create the differential step.
- R_1 represents a randomly generated array of dimensions $1 \times Dim$ with values uniformly distributed in the interval $[0, 1]$.
- Dim refers to the dimensionality of the problem's search space.
- $x_{i,j}^{new,P1}$ specifies the value of the newly proposed position in the j^{th} dimension.
- $F_i^{new,P1}$ represents the evaluated fitness value of the objective function for the newly proposed position $X_i^{new,P1}$.
- F_i represents the current fitness value of the i^{th} secretary bird before the update.

Stage 2: Consuming Prey

After a secretary bird discovers a snake, it engages in a distinctive and strategic method of hunting. Unlike other raptors that immediately dive into direct combat, the secretary bird utilizes its agile footwork to maneuver around the snake. Standing its ground from a high vantage point, the bird carefully observes every move of its prey. It hovers, jumps, and gradually provokes the snake, thereby wearing down the opponent's stamina [3].

This "peripheral combat" strategy provides the bird with a significant physical advantage. Its long legs make it difficult for the snake to entangle it, and its legs are covered with thick keratin scales acting as armor against venomous fangs. To mathematically model this random, evasive, and agile movement around the prey, the SBOA introduces Brownian motion (RB). This can be mathematically formulated using Equation (6) [3]:

$$RB = randn(1, Dim) \tag{II.6}$$

where:

- RB represents the Brownian motion vector simulating the bird's random steps.
- $randn(1, Dim)$ denotes a randomly generated array of dimensions $1 \times Dim$ drawn from a standard normal distribution (with a mean of 0 and a standard deviation of 1).

During this interaction stage (which operates within the temporal interval $\frac{1}{3}T < t < \frac{2}{3}T$), the secretary bird frequently pauses to lock onto the snake's location with its sharp eyesight. In the algorithm, this is conceptualized using the individual's historical best position (x_{best}). By combining x_{best} with Brownian motion, individuals can perform deep local searches around the best previously found positions to thoroughly explore the surrounding solution space [3].

This approach prevents premature convergence to local optima and accelerates convergence to the global optimum, as individuals search based on both global information and their historical best positions. The position update for this stage is mathematically modeled using Equations (7) and (8) [3]:

$$\text{While } \frac{1}{3}T < t < \frac{2}{3}T, \quad x_{i,j}^{new,P1} = x_{best} + \exp\left((t/T)^4\right) \times (RB - 0.5) \times (x_{best} - x_{i,j}) \tag{II.7}$$

$$X_i = \begin{cases} X_i^{new,P1}, & \text{if } F_i^{new,P1} < F_i \\ X_i, & \text{else} \end{cases} \tag{II.8}$$

where:

- x_{best} represents the position of the best secretary bird (best candidate solution) discovered by the population so far.
- $\exp((t/T)^4)$ is an adaptive exponential factor that depends on the current iteration (t) and maximum iterations (T), controlling the step size of the bird's maneuvers.

- RB_j represents the previously calculated Brownian motion array of dimension $1 \times Dim$.
- $x_{i,j}^{new,P1}$ represents the newly proposed position for the i^{th} bird in this second hunting stage (still within Exploration Phase 1).
- $F_i^{new,P1}$ denotes the evaluated fitness value of this newly proposed position.

Stage 3: Attacking the Prey

When the snake is finally exhausted, the secretary bird perceives the opportune moment to swiftly take action. Utilizing its powerful leg muscles, the bird launches a rapid attack characterized by a specialized leg-kicking technique. It delivers accurate and forceful kicks using its sharp talons, typically targeting the snake's head to quickly incapacitate or kill it, thereby avoiding any retaliatory bites. Occasionally, if the snake is too large to be dispatched immediately, the bird may carry it into the sky and drop it onto the hard ground to ensure its demise [3].

To mathematically simulate this decisive attacking behavior and enhance the optimizer's global search capabilities, SBOA introduces the *Lévy flight* strategy into the random search process. Lévy flight is a random walk pattern characterized by frequent short steps combined with occasional long jumps. This mathematical strategy perfectly simulates the bird's flight and striking ability. The large jumps enable the algorithm to extensively explore the global search space (bringing candidate solutions closer to the global optimum quickly), while the short steps significantly improve the optimization accuracy by refining local areas. This mechanism effectively reduces the risk of the algorithm becoming trapped in local optima [3].

Furthermore, to make the optimization process more dynamic, adaptive, and flexible, SBOA incorporates a nonlinear perturbation factor represented mathematically as $(1 - t/T)^{(2\times t/T)}$. This factor guarantees a superior balance between exploration and exploitation, avoids premature convergence, and accelerates the overall performance. The position update for the secretary bird during this final attacking stage (occurring in the final third of the iterations) is modeled using Equations (9) and (10) [3]:

$$\text{While } t > \frac{2}{3}T, \quad x_{i,j}^{new,P1} = x_{best} + \left(\left(1 - \frac{t}{T}\right)^{\left(2\times\frac{t}{T}\right)} \right) \times x_{i,j} \times RL \tag{II.9}$$

$$X_i = \begin{cases} X_i^{new,P1}, & \text{if } F_i^{new,P1} < F_i \\ X_i, & \text{else} \end{cases} \tag{II.10}$$

To calculate the step size and enhance optimization accuracy, the algorithm employs a weighted Lévy flight, denoted as RL, defined in Equation (11) [3]:

$$RL = 0.5 \times Levy(Dim) \tag{II.11}$$

Here, $Levy(D)$ represents the Lévy flight distribution function, which is mathematically calculated as follows [3]:

$$Levy(D) = s \times \frac{u \times \sigma}{|v|^{\frac{1}{\eta}}} \tag{II.12}$$

where u and v are random numbers uniformly distributed in the interval $[0, 1]$, s is a fixed constant set to 0.01, and η is a fixed constant equal to 1.5. The parameter σ is computed using the Gamma function (Γ) as shown in Equation (13) [3]:

$$\sigma = \left(\frac{\Gamma(1+\eta) \times \sin\left(\frac{\pi\eta}{2}\right)}{\Gamma\left(\frac{1+\eta}{2}\right) \times \eta \times 2^{\left(\frac{\eta-1}{2}\right)}} \right)^{\frac{1}{\eta}} \tag{II.13}$$

where:

- t and T are the current iteration and the maximum number of iterations, respectively.
- $x_{i,j}^{new,P1}$ is the newly calculated position for the i^{th} bird in the j^{th} dimension during this third stage of the exploration phase.
- x_{best} is the best candidate solution found so far.
- $(1 - t/T)^{(2\times t/T)}$ serves as the nonlinear perturbation factor for adaptability.
- RL represents the weighted Lévy flight factor used to modulate the step size.
- $Levy(D)$ is the Lévy distribution function ensuring random but strategic jumps.
- s (0.01) and η (1.5) are predefined constant parameters for the Lévy distribution.
- u and v are random variables drawn from the interval $[0, 1]$.
- σ is the scaling factor formulated based on the Gamma function Γ.

II.3.3 Exploitation Phase: Escape Strategy of Secretary Bird

The natural enemies of secretary birds are large predators such as eagles, hawks, foxes, and jackals, which may attack them or steal their food [3]. When encountering these threats, secretary birds typically employ various evasion strategies to protect themselves or their food. These evasion behaviors when confronted with threats are illustrated in Figure II.4.

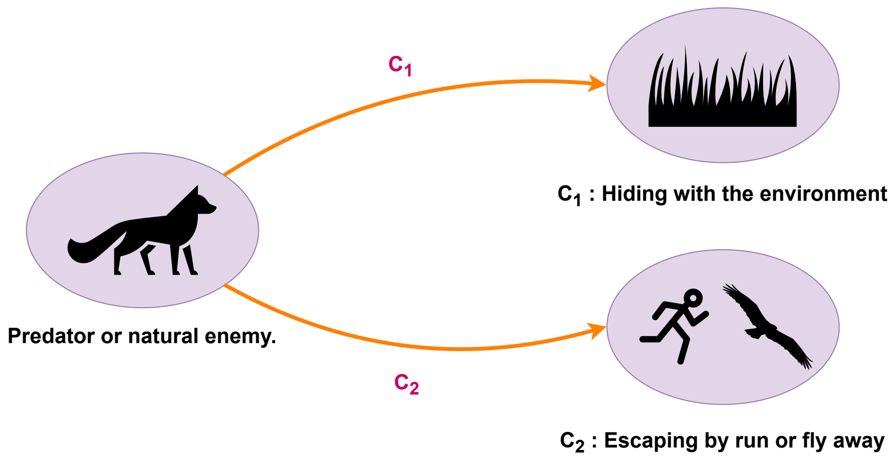


Figure II.4: The escape behavior of the secretary bird.

These survival strategies can be broadly categorized into two main types [3]:

- **Strategy 1 (C_1): Camouflage by environment.** When secretary birds detect the proximity of a predator, they initially search for a suitable camouflage environment. They may use the colors or structures in their environment to blend in, making it harder for predators to detect them.
- **Strategy 2 (C_2): Fly or run away.** If no suitable and safe camouflage environment is found nearby, they will opt for flight or rapid running to escape. Secretary birds are known for their exceptionally long legs, enabling them to run at remarkable speeds (covering 20 to 30 km in a single day, earning them the nickname "marching eagles"). Additionally, they are skilled flyers and can swiftly take flight to escape danger.

In the mathematical design of SBOA, it is assumed that one of these two conditions occurs with equal probability. To strike a balance between exploration (searching for new solutions) and exploitation (using known solutions), a dynamic perturbation factor denoted as $(1 - t/T)^2$ is introduced. By adjusting this factor, the algorithm enhances exploitation in this phase [3].

Both evasion strategies employed by the secretary bird are mathematically modeled using Equation (14), and the condition for updating the position is expressed using Equation (15) [3]:

$$x_{i,j}^{new,P2} = \begin{cases} C_1 : x_{best} + (2 \times RB - 1) \times \left(1 - \frac{t}{T}\right)^2 \times x_{i,j}, & \text{if } rand < r \\ C_2 : x_{i,j} + R_2 \times (x_{random} - K \times x_{i,j}), & \text{else} \end{cases} \tag{II.14}$$

$$X_i = \begin{cases} X_i^{new,P2}, & \text{if } F_i^{new,P2} < F_i \\ X_i, & \text{else} \end{cases} \tag{II.15}$$

where the coefficient K is calculated by Equation (16) [3]:

$$K = \text{round}(1 + \text{rand}(1, 1)) \tag{II.16}$$

where:

- $x_{i,j}^{new,P2}$ represents the newly generated position for the i^{th} bird in the j^{th} dimension during Phase 2 (Exploitation Phase).
- x_{best} is the best candidate solution found so far in the population.
- RB represents the Brownian motion vector.
- $(1 - t/T)^2$ represents the dynamic perturbation factor used to balance exploration and exploitation.
- $rand$ and $\text{rand}(1, 1)$ mean randomly generating a number between $(0, 1)$.
- r is a predefined probability threshold, set to 0.5 ($r = 0.5$).
- R_2 represents the random generation of an array of dimension $1 \times Dim$ from the normal distribution.
- x_{random} represents a random candidate solution selected from the current iteration.
- K represents the random selection of an integer (either 1 or 2), calculated using the rounding function.
- $F_i^{new,P2}$ denotes the evaluated fitness value of the newly proposed position.
- X_i and F_i represent the current position and fitness value of the i^{th} bird, respectively.

II.3.4 SBOA Algorithm and Flowchart

To summarize the operational flow of SBOA, the following pseudo-code and organigram illustrate the iterative process of balancing exploration and exploitation.

Algorithm 1 Pseudo code of SBOA [3]

Initialize Problem Setting (Dim, ub, lb, Pop_size (N)), Max_Iter(T), Curr_Iter(t)
Initialize the population randomly
for $t = 1 : T$ **do**
    Update Secretary Bird x_{best}.
    for $i = 1 : N$ **do**
        Exploration:
        if $t < \frac{1}{3}T$ **then**
            Calculate new status of the i^{th} Secretary Bird using Eq. (4)
            Update the i^{th} Secretary Bird using Eq. (5)
        else if $\frac{1}{3}T < t < \frac{2}{3}T$ **then**
            Calculate new status of the i^{th} Secretary Bird using Eq. (7)
            Update the i^{th} Secretary Bird using Eq. (8)
        else
            Calculate new status of the i^{th} Secretary Bird using Eq. (9)
            Update the i^{th} Secretary Bird using Eq. (10)
        end if
        Exploitation:
        if $r < 0.5$ **then**
            Calculate new status of the i^{th} Secretary Bird using C_1 in Eq. (14)
        else
            Calculate new status of the i^{th} Secretary Bird using C_2 in Eq. (14)
        end if
        Update the i^{th} Secretary Bird using Eq. (15)
    end for
    Save best candidate solution so far.
end for
Output: The best solution obtained by SBOA for given optimization problem.
Return best solution.

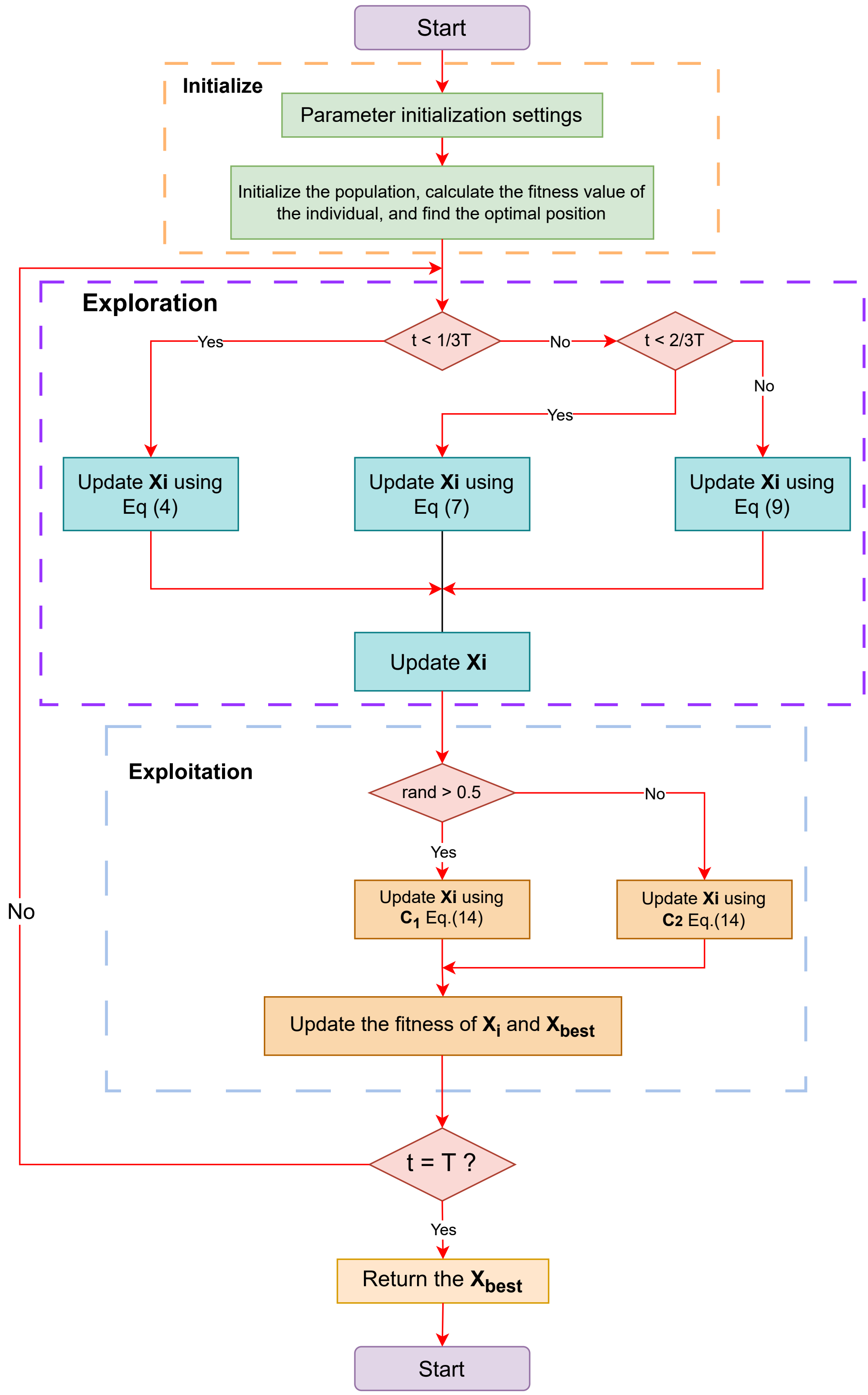


Figure II.5: Flowchart of SBOA.

Chapter III

Application of SBOA for Schottky Diode Parameter Extraction

Contents

III.1 Introduction

The previous chapter established the theoretical foundation of the Secretary Bird Optimization Algorithm (SBOA). Emerging as a novel and highly effective metaheuristic, SBOA distinguishes itself through robust mathematical formulations that efficiently model complex hunting and evasion strategies to perfectly balance the exploration and exploitation phases. However, before deploying any new algorithm to solve complex, real-world engineering problems, it is imperative to rigorously validate its computational capabilities. This chapter bridges the gap between mathematical theory and practical application, focusing first on the performance assessment of SBOA and its subsequent utilization in semiconductor device modeling.

To ensure a comprehensive and globally recognized evaluation, SBOA is initially tested against a strategic selection of modern benchmark test functions. Rather than relying solely on theoretical assumptions, this rigorous assessment uses challenging unimodal and multimodal landscapes to quantitatively evaluate the algorithm's convergence speed, stability, and its inherent mathematical capacity to avoid premature convergence. Successfully navigating these deceptive test functions is a critical prerequisite, proving that the SBOA's mathematical framework is solid, reliable, and ready for advanced applications.

Building upon this validated mathematical confidence, the chapter then addresses the primary objective of this study: the parameter extraction of Metal-Semiconductor (MS) and Metal-Insulator-Semiconductor (MIS) Schottky diodes. The precise characterization of these devices is a highly non-linear optimization problem characterized by a complex error surface. By applying the proven and evaluated SBOA to this specific challenge, this study aims to accurately and efficiently identify the true physical parameters of the fabricated diodes, demonstrating the algorithm's superiority and effectiveness over traditional analytical methods.

The remainder of this chapter is organized as follows: Section 3.2 presents the detailed evaluation of SBOA using the selected benchmark test functions. Section 3.3 describes the physical and mathematical models of the Schottky diode structures. Finally, Section 3.4 discusses the experimental setup, validation, and physical interpretation of the parameters extracted by the algorithm.

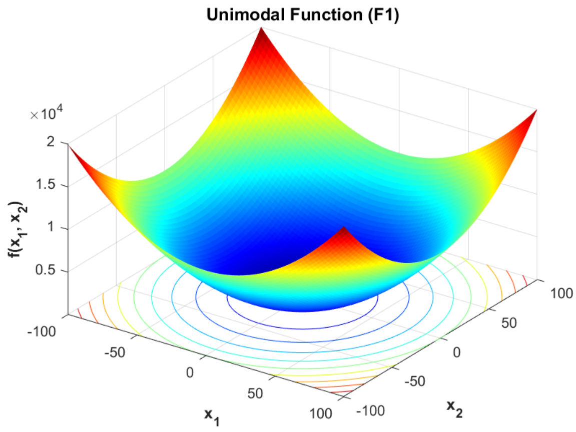
III.2 Evaluation of SBOA

III.2.1 Benchmark Test Functions

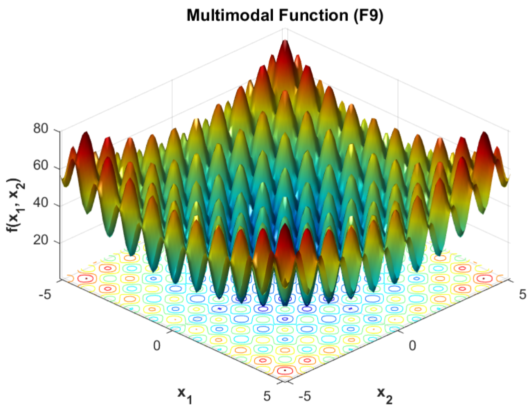
To rigorously validate the performance of any novel metaheuristic algorithm, it is essential to test it against standard benchmark functions with known global optima. This approach ensures a fair and comprehensive evaluation of the algorithm's capabilities before applying it to complex, real-world engineering problems. In this study, the performance of the Secretary Bird Optimization Algorithm (SBOA) is evaluated using a carefully selected subset of the 23 classical benchmark functions originally defined in the 2005 IEEE Congress on Evolutionary Computation (CEC'05) [29].

These benchmark functions are systematically categorized into three distinct groups, each designed to challenge different mathematical characteristics of the optimization algorithm [30]:

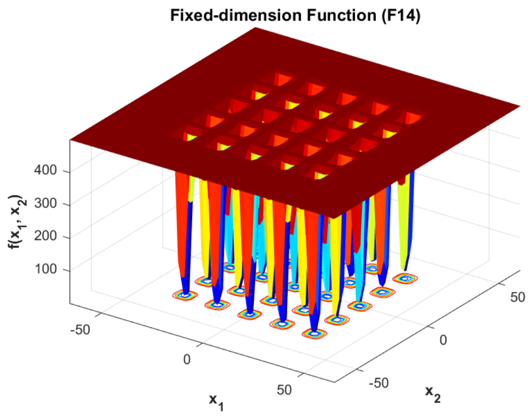
- **High-dimensional Unimodal Functions ($F_1 - F_7$):** These functions possess a single global optimum. They are primarily utilized to evaluate the *exploitation* capability and the convergence speed of the algorithm (see Figure III.1a).
- **High-dimensional Multimodal Functions ($F_8 - F_{13}$):** Considered the most challenging optimization landscapes, these functions feature numerous local optima. They are specifically designed to test the algorithm's *exploration* capacity and its ability to avoid premature convergence (see Figure III.1b).
- **Low-dimensional Multimodal Functions ($F_{14} - F_{23}$):** Similar to the previous category but operating in lower-dimensional spaces, these functions contain a fewer number of local optima. They are crucial for assessing the algorithm's balance between exploration and exploitation phases (see Figure III.1c).



(a) Example of a high-dimensional unimodal optimization landscape (F_1).



(b) Example of a high-dimensional multimodal optimization landscape (F_9).



(c) Example of a low-dimensional multimodal optimization landscape (F_{14}).

Figure III.1: 3D visualization of the optimization landscapes for the three categories of benchmark functions.

Instead of executing the entire suite of 23 classical benchmark functions (the complete mathematical details of which are formally documented in **Appendix A**), which often leads to computational redundancy, this study strategically selects six representative functions (two from each category: $F_1, F_2, F_{11}, F_{12}, F_{14},$ and F_{21}) to demonstrate the efficacy of SBOA. The mathematical formulations, dimensions (D), search spaces, and theoretical optima of these selected functions are detailed in Table III.1.

Table III.1: Description of the selected benchmark test functions.

Function	Mathematical Formulation	D	Range	Optimum
	Unimodal Functions			
F_1 (Sphere)	$f_1(x) = \sum_{i=1}^{D} x_i^2$	30	$[-100, 100]$	0
F_2 (Schwefel 2.22)	$f_2(x) = \sum_{i=1}^{D} \lvert x_i\rvert + \prod_{i=1}^{D} \lvert x_i\rvert$	30	$[-10, 10]$	0
	Multimodal Functions			
F_{11} (Griewank)	$f_{11}(x) = \frac{1}{4000}\sum_{i=1}^{D} x_i^2 - \prod_{i=1}^{D} \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	30	$[-600, 600]$	0
F_{12} (Penalized 1)	$f_{12}(x) = \frac{\pi}{D}\left\{10\sin^2(\pi y_1) + \sum_{i=1}^{D-1}(y_i-1)^2[1+10\sin^2(\pi y_{i+1})] + (y_D-1)^2\right\} + \sum_{i=1}^{D} u(x_i, 10, 100, 4)$; $y_i = 1 + \frac{x_i+1}{4}, \quad u(x_i,a,k,m) = \begin{cases} k(x_i-a)^m & \text{if } x_i > a \\ 0 & \text{if } -a < x_i < a \\ k(-x_i-a)^m & \text{if } x_i < -a \end{cases}$	30	$[-50, 50]$	1.57×10^{-32}
	Low-dimensional Multimodal Functions			
F_{14} (Foxholes)	$f_{14}(x) = \left[\frac{1}{500} + \sum_{j=1}^{25} \frac{1}{j+\sum_{i=1}^{2}(x_i-a_{ij})^6}\right]^{-1}$	2	$[-65.53, 65.53]$	0.998004
F_{21} (Shekel 5)	$f_{21}(x) = -\sum_{i=1}^{5}[(x-a_i)(x-a_i)^T + c_i]^{-1}$	4	$[0, 10]$	-10.1532

III.2.2 Experimental Setup and General Configuration

To ensure the statistical validity and reliability of the performance assessment, the SBOA was executed under rigorously controlled experimental conditions. The algorithm was implemented and simulated within the MATLAB computational environment. For transparency and reproducibility, the complete MATLAB source codes including the main execution script, algorithm parameter configurations, benchmark function definitions, and the core SBOA implementation are comprehensively provided in **Appendix B**.

Given the stochastic nature of metaheuristic optimization algorithms, a single execution is insufficient to conclusively determine performance. Consequently, the SBOA was independently executed 25 times ($r = 25$) for each of the selected benchmark functions. The population size was uniformly fixed at $N = 100$ candidate solutions across all tests to maintain a consistent baseline for exploration.

A key aspect of this experimental setup is the dynamic adaptation of the maximum number of iterations (T_{max}). Instead of utilizing a static threshold, T_{max} was specifically tailored to the mathematical complexity and dimensionality of each landscape, ranging from 70 iterations for functions with rapid convergence profiles (e.g., F_{11}) to 2000 iterations for highly complex, penalized multimodal functions (e.g., F_{12}). This dynamic allocation ensures that the algorithm is granted sufficient computational time to escape local minima in complex spaces while avoiding unnecessary computational overhead in simpler landscapes.

The quantitative performance is evaluated based on two primary statistical metrics derived from the 25 independent runs:

- **Mean:** Represents the average best score obtained. It evaluates the *accuracy* and exploitation capability of the algorithm.
- **Variance:** Measures the dispersion of the results around the mean. A low variance indicates high *robustness* and algorithmic stability.

III.2.3 Overall Optimization Results

Table III.2 summarizes the overall statistical results achieved by the SBOA across the six selected benchmark functions.

Table III.2: Overall optimization results of SBOA over 25 independent runs.

Category	Function	Dim (D)	Max Iter (T_{max})	Theoretical Optimum	Mean (Obtained)	Variance
Unimodal	F_1	30	500	0	0.0000	0.0000
	F_2	30	500	0	1.0983×10^{-199}	0.0000
Multimodal	F_{11}	30	70	0	2.681×10^{-3}	7.138×10^{-3}
	F_{12}	30	2000	1.5705×10^{-32}	1.625×10^{-16}	4.405×10^{-17}
Low-dimensional	F_{14}	2	150	0.998004	0.998003	1.634×10^{-16}
	F_{21}	4	200	-10.1532	-10.1532	1.854×10^{-12}

As observed in Table III.2, the SBOA demonstrated exceptional adaptability. For unimodal functions (F_1, F_2), it achieved the exact theoretical optimum with zero variance, showcasing flawless exploitation. In complex multimodal landscapes (F_{11}, F_{12}), the algorithm successfully bypassed numerous local minima to reach near-optimal solutions. Furthermore, in low-dimensional functions (F_{14}, F_{21}), SBOA pinpointed the exact global minima with negligible variance, confirming its robust balance between exploration and exploitation.

Detailed Analysis: Function F1 (Sphere)

The Sphere function (F_1) is a smooth, convex, and unimodal benchmark utilized to evaluate the exploitation intensity and convergence speed of the algorithm. Mathematically, it is formulated as:

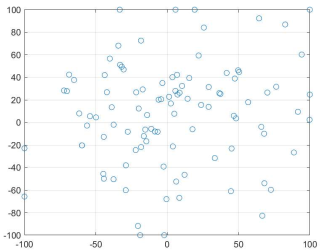
$$f_1(x) = \sum_{i=1}^{D} x_i^2 \tag{III.1}$$

For this experiment, the function was configured with a dimension of $D = 30$. The SBOA was independently executed for $r = 25$ runs with a population of $N = 100$ over

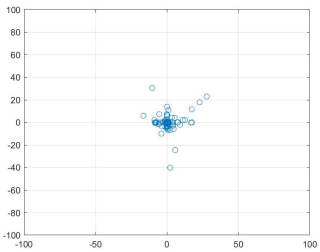
$T_{max} = 500$ iterations. The detailed parameter settings and obtained statistical results for F_1 are summarized in Table III.3.

Table III.3: Experimental parameters and optimization results for F_1.

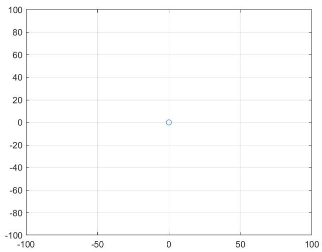
Parameter	**Value**	**Metric**	**Result**
Dimension (D)	30	Theoretical Optimum	0
Search Range	$[-100, 100]$	Best Score Obtained	0
Population (N)	100	Mean Score	0
Max Iterations (T)	500	Variance	0
Independent Runs (r)	25	Best Position ($best\ pos$)	$\approx 1.0 \times 10^{-161}$



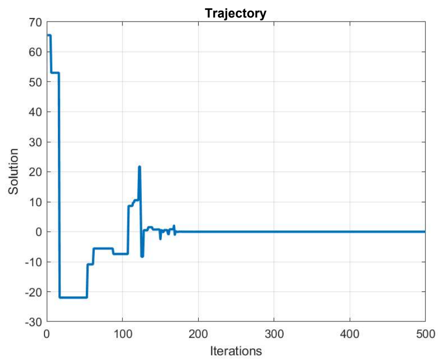
(a) Initial (t=1)



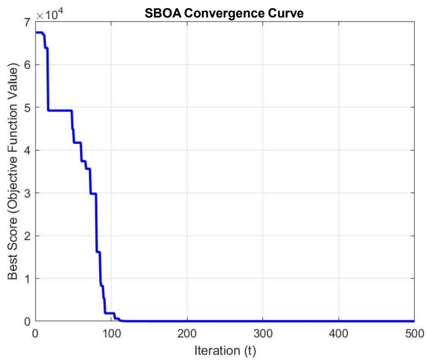
(b) Middle (t=250)



(c) Final (t=500)



(d) Trajectory



(e) Convergence Curve

Figure III.2: Optimization dynamics of SBOA on function F_1. (a-c) show the transition from random exploration to precise exploitation; (d) illustrates the trajectory of the first agent; (e) displays the rapid convergence toward the global optimum.

The numerical results indicate that SBOA achieved the exact theoretical optimum (0) with zero variance, demonstrating perfect precision. The trajectory analysis (Figure III.2d) shows that the algorithm initially performs broad jumps to explore the search space before rapidly stabilizing around the origin. This behavior is further confirmed by the convergence curve (Figure III.2e), where the best fitness value reaches the optimum within the first 150 iterations. Visually, the population distribution plots (Figure III.2a-III.2c) illustrate the efficient transition from a stochastic search to a concentrated exploitation phase at the global minimum.

Detailed Analysis: Function F2 (Schwefel 2.22)

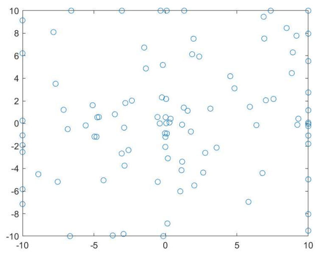
The Schwefel 2.22 function (F_2) is another continuous, high-dimensional unimodal benchmark. Compared to the Sphere function, it introduces an additional product term, making the exploitation process slightly more challenging. Its mathematical formulation is given by:

$$f_2(x) = \sum_{i=1}^{D} |x_i| + \prod_{i=1}^{D} |x_i| \tag{III.2}$$

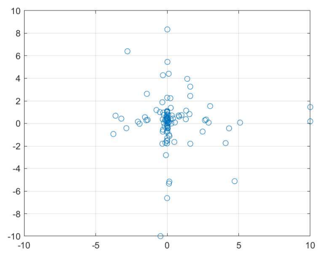
The algorithm was tested on $D = 30$ dimensions within the search range of $[-10, 10]$. Consistent with the established protocol, $N = 100$ candidate solutions were evolved over $T_{max} = 500$ iterations for $r = 25$ independent runs. The quantitative results are presented in Table III.4.

Table III.4: Experimental parameters and optimization results for F_2.

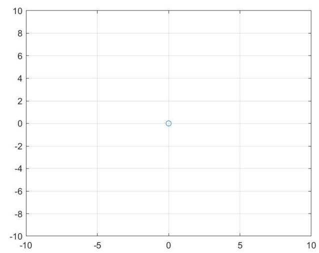
Parameter	**Value**	**Metric**	**Result**
Dimension (D)	30	Theoretical Optimum	0
Search Range	$[-10, 10]$	Best Score Obtained	3.99×10^{-203}
Population (N)	100	Mean Score	1.09×10^{-199}
Max Iterations (T)	500	Variance	0
Independent Runs (r)	25	Best Position (*best pos*)	$\approx 1.0 \times 10^{-203}$



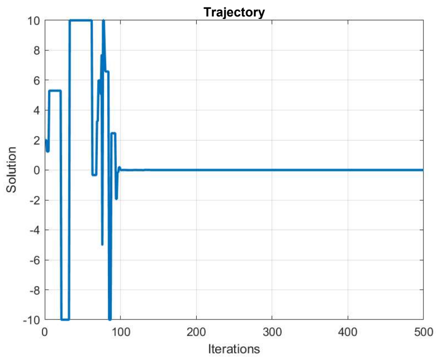
(a) Initial (t=1)



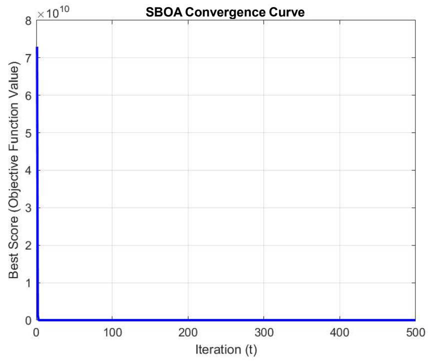
(b) Middle (t=250)



(c) Final (t=500)



(d) Trajectory



(e) Convergence Curve

Figure III.3: Optimization dynamics of SBOA on function F_2. The sequence (a-c) confirms the effective shrinking of the search space; (d) reveals the smooth trajectory towards the origin; (e) demonstrates the algorithm's steep descent to an exceptionally precise minimum.

The statistical indicators for F_2 highly emphasize the exceptional exploitation capabilities of SBOA. The algorithm achieved an astonishing mean score in the order of 10^{-199} with zero variance, effectively locating the exact theoretical optimum. The convergence curve (Figure III.3e) demonstrates a remarkably steep gradient, proving that the algorithm rapidly minimizes the objective function without stagnation. Furthermore, the search history visualizations (Figure III.3a-III.3c) confirm that the entire population swiftly collapses into the optimal solution point, showcasing immense algorithmic stability in unimodal landscapes.

Detailed Analysis: Function F11 (Griewank)

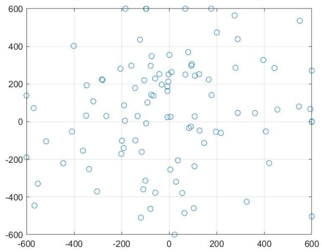
The Griewank function (F_{11}) is a complex multimodal benchmark characterized by an exponential number of local optima, which poses a significant challenge for any optimization algorithm to identify the true global minimum. Its mathematical formulation is defined as follows:

$$f_{11}(x) = \frac{1}{4000}\sum_{i=1}^{D} x_i^2 - \prod_{i=1}^{D}\cos\left(\frac{x_i}{\sqrt{i}}\right) + 1 \tag{III.3}$$

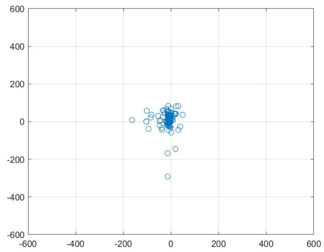
For this study, the function was configured with $D = 30$ dimensions within a broad search range of $[-600, 600]$. Notably, the SBOA required only $T_{max} = 70$ iterations to achieve convergence, utilizing a population of $N = 100$ over $r = 25$ independent runs. The statistical performance is detailed in Table III.5.

Table III.5: Experimental parameters and optimization results for F_{11}.

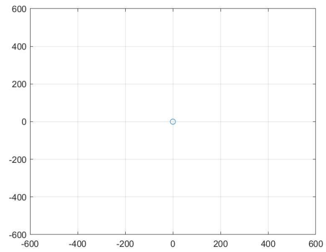
Parameter	**Value**	**Metric**	**Result**
Dimension (D)	30	Theoretical Optimum	0
Search Range	$[-600, 600]$	Best Score Obtained	0
Population (N)	100	Mean Score	2.68×10^{-3}
Max Iterations (T)	70	Variance	7.13×10^{-3}
Independent Runs (r)	25	Best Position (*best pos*)	$\approx 1.0 \times 10^{-7}$



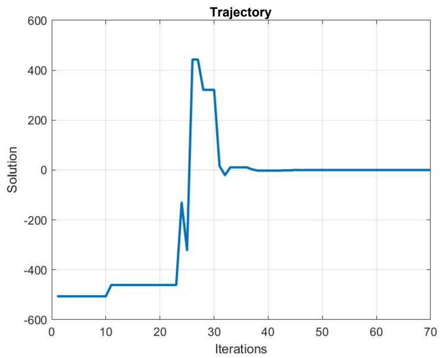
(a) Initial (t=1)



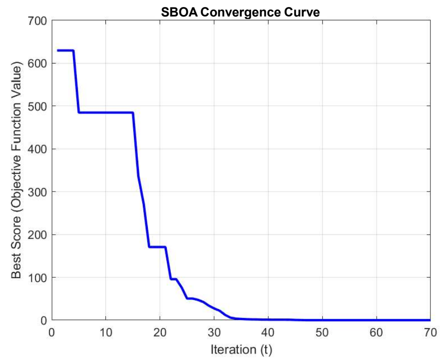
(b) Middle (t=35)



(c) Final (t=70)



(d) Trajectory



(e) Convergence Curve

Figure III.4: Optimization dynamics of SBOA on the multimodal function F_{11}. (a-c) visualize the rapid global exploration and subsequent local refinement; (d) shows the initial oscillations of the first agent before locating the optimum; (e) displays an extremely fast convergence profile.

The results obtained for F_{11} illustrate the high efficiency of SBOA in navigating rugged optimization landscapes. Despite the presence of numerous local minima, the algorithm successfully reached the exact theoretical optimum (0) in its best run, with a very low mean error (2.68×10^{-3}). The trajectory plot (Figure III.4d) is particularly revealing, as it shows significant fluctuations in the early stages a clear indication of effective exploration followed by a swift stabilization at the global minimum. Moreover, the convergence curve (Figure III.4e) confirms that SBOA achieves a high quality solution in a fraction of the time required for unimodal functions, proving its superior balance between exploration and exploitation phases.

Detailed Analysis: Function F12 (Penalized 1)

The Penalized 1 function (F_{12}) is a highly complex multimodal benchmark that incorporates a penalty term to challenge the algorithm's ability to remain within the feasible search space while navigating a rugged landscape with numerous local optima.

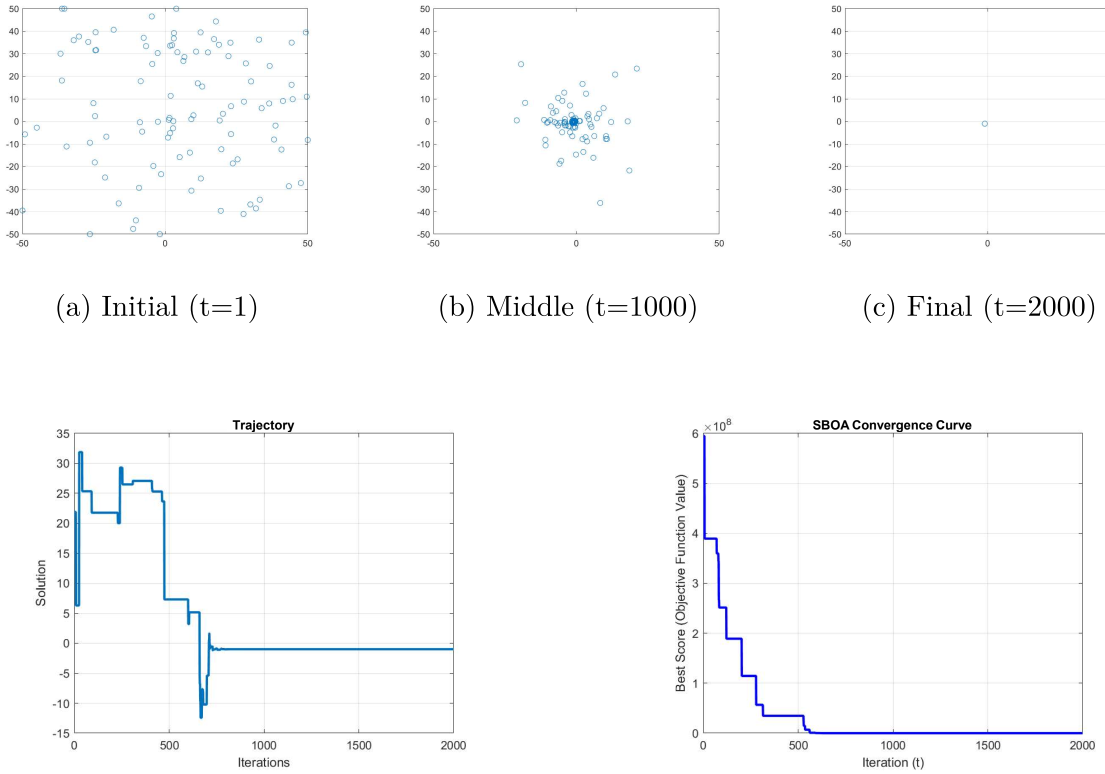
Its mathematical structure is defined as:

$$f_{12}(x) = \frac{\pi}{D}\left\{10\sin^2(\pi y_1) + \sum_{i=1}^{D-1}(y_i-1)^2[1+10\sin^2(\pi y_{i+1})] + (y_D-1)^2\right\} + \sum_{i=1}^{D} u(x_i, 10, 100, 4) \tag{III.4}$$

Where $y_i = 1 + \frac{x_i+1}{4}$ and $u(x_i, a, k, m)$ is the penalty function. This benchmark was tested with $D = 30$ dimensions in the range $[-50, 50]$. Due to its complexity, the SBOA was configured with $T_{max} = 2000$ iterations and a population of $N = 100$. The statistical results over $r = 25$ runs are summarized in Table III.6.

Table III.6: Experimental parameters and optimization results for F_{12}.

Parameter	**Value**	**Metric**	**Result**
Dimension (D)	30	Theoretical Optimum	1.57×10^{-32}
Search Range	$[-50, 50]$	Best Score Obtained	1.07×10^{-16}
Population (N)	100	Mean Score	1.62×10^{-16}
Max Iterations (T)	2000	Variance	4.40×10^{-17}
Independent Runs (r)	25	Best Position (*best pos*)	≈ -1.0000



(a) Initial (t=1)

(b) Middle (t=1000)

(c) Final (t=2000)

(d) Trajectory

(e) Convergence Curve

Figure III.5: Optimization dynamics of SBOA on the penalized function F_{12}. (a-c) demonstrate the algorithm's capability to concentrate the population despite the penalty constraints; (d) reveals an active exploration phase; (e) shows a consistent and stable convergence towards the global minimum.

The optimization of F_{12} provides a strong testament to the robustness of SBOA. The algorithm achieved a mean score of 1.62×10^{-16} with an extremely low variance (10^{-17}), indicating a high degree of stability across different runs. The trajectory analysis (Figure III.5d) illustrates the algorithm's "jumping" behavior in the early iterations, which is essential for escaping local optima in penalized landscapes. This is followed by a smooth stabilization phase as the agents converge toward the global minimum. The convergence curve (Figure III.5e) further confirms that SBOA maintains a steady descent throughout the 2000 iterations, proving its efficiency in handling non-linear optimization problems with complex constraints.

Detailed Analysis: Function F14 (Foxholes)

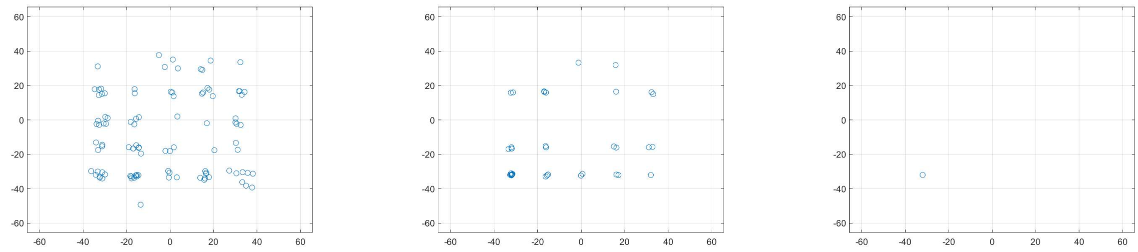
The Foxholes function (F_{14}) represents the class of low-dimensional multimodal benchmark functions. Despite having only two dimensions ($D = 2$), it presents a highly deceptive optimization landscape characterized by 25 distinct local minima (foxholes) distributed across the search space. The algorithm must possess robust exploration capabilities to avoid being trapped in sub-optimal foxholes. The mathematical representation is:

$$f_{14}(x) = \left[\frac{1}{500} + \sum_{j=1}^{25} \frac{1}{j + \sum_{i=1}^{2} (x_i - a_{ij})^6} \right]^{-1} \tag{III.5}$$

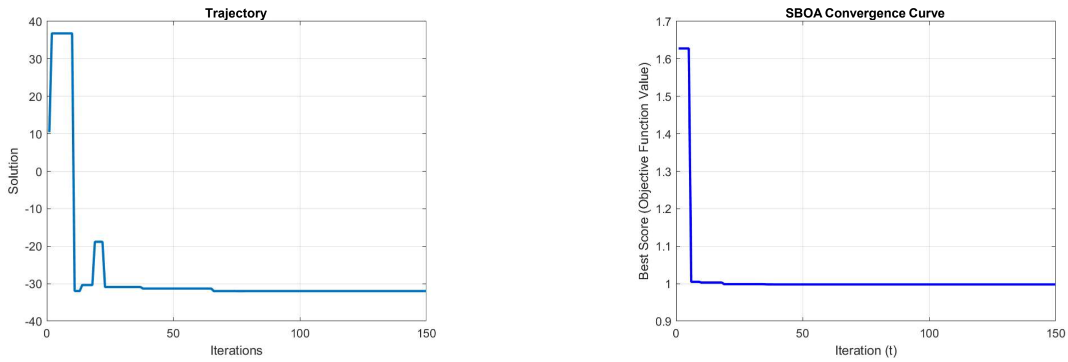
The experimental evaluation for F_{14} was conducted within the search bounds of $[-65.536, 65.536]$. Given the lower dimensionality, the SBOA required only $T_{max} = 150$ iterations to effectively navigate the landscape. The execution was performed $r = 25$ times with $N = 100$ agents. The numerical outcomes are presented in Table III.7.

Table III.7: Experimental parameters and optimization results for F_{14}.

Parameter	**Value**	**Metric**	**Result**
Dimension (D)	2	Theoretical Optimum	0.998004
Search Range	$[-65.536, 65.536]$	Best Score Obtained	0.9980038
Population (N)	100	Mean Score	0.9980038
Max Iterations (T)	150	Variance	1.63×10^{-16}
Independent Runs (r)	25	Best Position (*best pos*)	$[-31.9783, -31.9783]$



(a) Initial (t=1) (b) Middle (t=75) (c) Final (t=150)



(d) Trajectory (e) Convergence Curve

Figure III.6: Optimization dynamics of SBOA on the low-dimensional function F_{14}. (a-c) highlight the population's convergence into the deepest local minimum; (d) shows the agent's path across the deceptive landscape; (e) demonstrates rapid identification of the global optimum.

The statistical results for F_{14} are exceptional. SBOA successfully located the exact global optimum ($0.9980038 \approx 0.998004$) consistently across all 25 runs, as evidenced by the near-zero variance (1.63×10^{-16}). This indicates that the algorithm is entirely immune to the deceptive local minima of this specific landscape. The convergence curve (Figure III.6e) shows a sharp, immediate drop in the early iterations, proving that SBOA rapidly identifies the correct region containing the global optimum. The scatter plots (Figure III.6a-III.6c) visually support this, revealing how the uniformly distributed population efficiently regroups at the exact coordinates of the deepest foxhole ($[-31.97, -31.97]$).

Detailed Analysis: Function F21 (Shekel 5)

The Shekel 5 function (F_{21}) is a standard low-dimensional multimodal benchmark with four dimensions ($D = 4$) and five local minima. It is widely used to test the algorithm's capability to locate the global optimum among several closely spaced local optima.

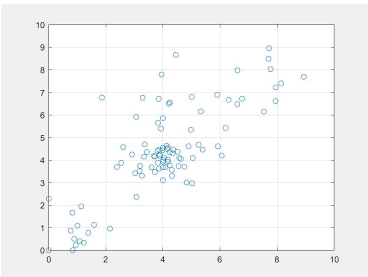
The mathematical definition is given by:

$$f_{21}(x) = -\sum_{i=1}^{5} \left[(x - a_i)(x - a_i)^T + c_i\right]^{-1} \tag{III.6}$$

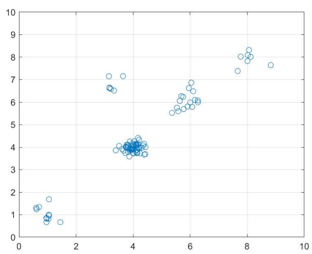
The experimental domain was set to $[0, 10]$ for all dimensions. The SBOA was executed with $N = 100$ agents for $T_{max} = 200$ iterations over $r = 25$ independent runs. The quantitative evaluation metrics are detailed in Table III.8.

Table III.8: Experimental parameters and optimization results for F_{21}.

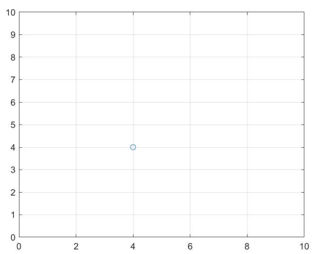
Parameter	**Value**	**Metric**	**Result**
Dimension (D)	4	Theoretical Optimum	-10.1532
Search Range	$[0, 10]$	Best Score Obtained	-10.1532
Population (N)	100	Mean Score	-10.1532
Max Iterations (T)	200	Variance	1.85×10^{-12}
Independent Runs (r)	25	Best Position (*best pos*)	$[4.000, 4.000, 4.000, 4.000]$



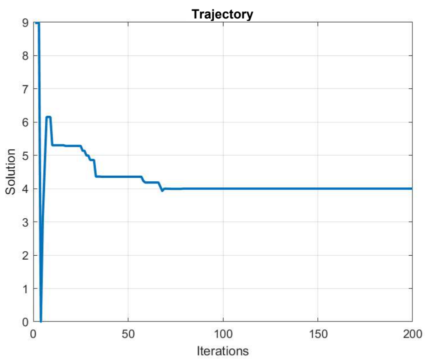
(a) Initial (t=1)



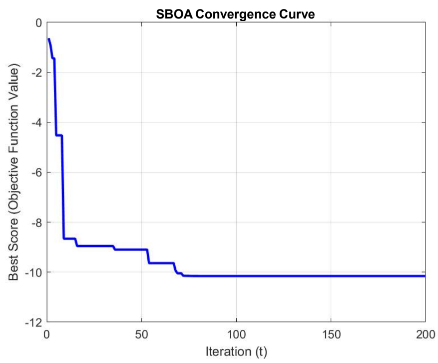
(b) Middle (t=100)



(c) Final (t=200)



(d) Trajectory



(e) Convergence Curve

Figure III.7: Optimization dynamics of SBOA on the Shekel 5 function (F_{21}). (a-c) present the effective narrowing of the search space around the global minimum; (d) depicts the agent's exploratory trajectory; (e) illustrates the step-wise convergence escaping local minima.

The statistical performance of SBOA on F_{21} is outstanding. The algorithm consistently reached the theoretical global minimum (-10.1532) in every single run, achieving a near-zero variance of 1.85×10^{-12}. This extraordinary stability confirms that SBOA is not easily deceived by the multiple local optima present in the Shekel functions. The convergence curve (Figure III.7e) displays a distinct step-like descent, which is characteristic of the algorithm successfully jumping out of local minima (foxholes) to find a deeper one. The trajectory plot (Figure III.7d) reflects an initial burst of exploration followed by a definitive lock onto the precise coordinates of the global optimum ($\approx [4, 4, 4, 4]$), establishing SBOA as a highly reliable optimizer for multidimensional engineering challenges.

III.3 Application of SBOA for Schottky Diode Parameter Extraction

III.3.1 Overview of Schottky Diodes: MS and MIS Structures

Schottky barrier diodes (SBDs) are fundamental components in modern electronic and optoelectronic applications, including solar cells, photodetectors, and high-frequency switching devices. The simplest form of these devices is the Metal-Semiconductor (MS) structure, formed by the direct contact between a metal and a semiconductor.

However, traditional MS structures often suffer from non-ideal behaviors due to the presence of interface states (N_{ss}), barrier inhomogeneities, and high series resistance (R_s) [1,31]. To mitigate these issues and enhance the electrical and dielectric performance, an interfacial insulator layer (such as oxides or polymers like TiO_2) is deliberately introduced between the metal and the semiconductor. This forms a Metal-Insulator-Semiconductor (MIS) structure. The insertion of this high-dielectric insulating layer passivates the active dangling bonds at the semiconductor surface, effectively modulating the barrier height, reducing the leakage current, and improving the overall rectification capabilities of the diode [31,32].

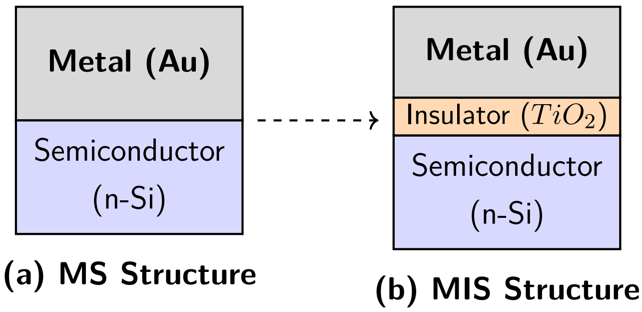


Figure III.8: Schematic cross-section of (a) Metal-Semiconductor (MS) and (b) Metal-Insulator-Semiconductor (MIS) structures.

III.3.2 Mathematical Model of the $I - V$ Characteristics

To properly evaluate the performance of MS and MIS structures, it is essential to model their Current-Voltage ($I - V$) characteristics mathematically. According to the Thermionic Emission (TE) theory, which assumes that charge carriers possess sufficient thermal energy to overcome the potential barrier, the forward-bias $I - V$ relationship for a diode with series resistance is expressed as [1,31]:

$$I = I_0 \left[\exp\left(\frac{q(V - IR_s)}{nkT} \right) - 1 \right] \tag{III.7}$$

Where I_0 represents the reverse saturation current, explicitly defined by:

$$I_0 = AA^*T^2 \exp\left(-\frac{q\Phi_{B0}}{kT}\right) \tag{III.8}$$

In these equations, the parameters are defined as follows:

- V and I: The measured experimental voltage applied across the diode and the corresponding measured current, respectively.
- q: The elementary charge of an electron (1.602×10^{-19} C).
- k: The Boltzmann constant (1.38×10^{-23} J/K).
- T: The absolute temperature in Kelvin (typically evaluated at room temperature, $T = 300$ K).
- A: The effective rectifier contact area of the diode.
- A^*: The effective Richardson constant for the specific semiconductor material.

Key Extraction Parameters

From an optimization perspective, identifying the precise electrical characteristics of the manufactured diode is treated as a parameter extraction problem. The primary unknown parameters that govern the physical behavior of the diode in Equations III.7 and III.8 include:

1. **Zero-bias Barrier Height (Φ_{B0}):** Expressed in electron-volts (eV), it represents the potential energy barrier that electrons must overcome to transition from the metal to the semiconductor.
2. **Series Resistance (R_s):** Represents the internal resistance of the semiconductor bulk and the electrical contacts. It significantly limits the current flow in the high forward-bias region, causing the $I-V$ curve to deviate from its exponential behavior.
3. **Ideality Factor (n):** A dimensionless parameter that indicates how closely the diode follows the ideal TE theory ($n = 1$). Values greater than 1 indicate non-ideal behaviors caused by the interfacial layer or alternative conduction mechanisms.

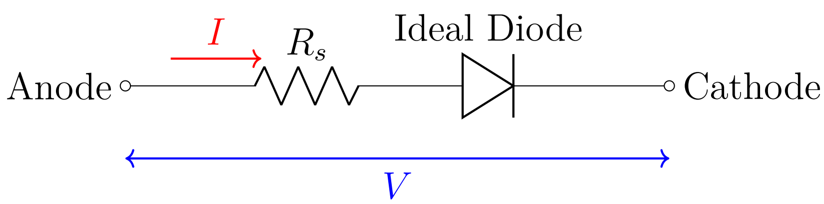


Figure III.9: Equivalent electrical circuit model of a non-ideal Schottky diode incorporating series resistance (R_s).

Accurately determining these parameters is highly challenging due to the non-linear and transcendental nature of Equation III.7. Traditionally, the barrier height (Φ_{B0}) and the ideality factor (n) are extracted from the y-intercept (I_0) and the slope of the linear region of the semi-logarithmic $\ln(I) - V$ plot, respectively [31]:

$$\Phi_{B0} = \frac{kT}{q} \ln\left(\frac{AA^*T^2}{I_0}\right) \quad \text{and} \quad n = \frac{q}{kT}\left(\frac{dV}{d(\ln I)}\right) \tag{III.9}$$

Limitations of Analytical Models:
While traditional analytical methods provide a basic estimation, they fundamentally rely on linear approximations that fail to accurately capture the complex physical phenomena occurring within the diode. In reality, Schottky diodes exhibit highly non-ideal behaviors such as barrier inhomogeneities, unpredictable leakage currents, and the presence of deep-level interface states which distort the actual $I - V$ curve [1]. Consequently, simplified mathematical models cannot perfectly reproduce this physical reality. To bridge the gap between theoretical modeling and true physical behavior, advanced metaheuristic algorithms like SBOA are required. These algorithms bypass the limitations of mathematical simplifications, allowing for a precise and simultaneous extraction of the diode's true physical parameters directly from experimental data [31].

III.3.3 Problem Formulation and Objective Function

The extraction of electrical parameters for MS and MIS Schottky diodes is fundamentally treated as a non-linear, multi-dimensional global optimization problem. The primary goal is to identify a set of parameters that minimizes the discrepancy between the experimentally measured current (I_{exp}) and the current calculated (I_{calc}) using the theoretical model [1, 31].

Decision Variables and Search Space

In this study, the Secretary Bird Optimization Algorithm (SBOA) is employed to search for the optimal values within a defined three-dimensional search space. Each candidate solution, or "individual" in the SBOA population, is represented by a decision vector x containing the primary unknown physical parameters:

$$x = [\Phi_{B0}, R_s, n] \tag{III.10}$$

(Note: The reverse saturation current I_0 is not treated as an independent decision variable. Instead, it is dynamically computed during the optimization process using the extracted value of Φ_{B0} according to Equation III.8 [1, 31]).

To ensure physical consistency and accelerate convergence, each parameter is constrained within a specific range $[LB_j, UB_j]$, where LB and UB are the lower and upper bounds, respectively:

$$LB_j \leq x_j \leq UB_j, \quad j = 1, 2, 3 \tag{III.11}$$

The Objective Function (SAE)

To evaluate the quality of each candidate solution generated by SBOA, an objective function (also known as a fitness function) must be defined. In this implementation, the Sum of Absolute Errors (SAE) is employed to quantify the deviation between the experimental data and the theoretical model [31]. To ensure high accuracy, the experimental data points are specifically extracted from the forward-bias region ($V > 0$).

The optimization algorithm aims to minimize the following fitness function:

$$f(x) = \mathrm{SAE}(x) = \sum_{i=1}^{N} |I_{exp,i} - I_{calc,i}(V_i, x)| \tag{III.12}$$

Where:

- N: The total number of experimental data points considered in the forward-bias region.
- $I_{exp,i}$: The measured current value at the i-th voltage point.
- $I_{calc,i}$: The theoretical current calculated for the same voltage point (V_i) using the candidate parameters in vector x.

By iteratively updating the population, SBOA explores the search space to find the global minimum of the SAE function. This theoretical formulation will be seamlessly implemented in the MATLAB environment, where the algorithm will process the real measured datasets to extract the optimal parameters that perfectly replicate the actual physical behavior of the fabricated Schottky diode.

III.4 Results and Discussion

III.4.1 Experimental Setup and SBOA Configuration

To validate the proposed parameter extraction methodology, the Secretary Bird Optimization Algorithm (SBOA) was implemented in the MATLAB environment (R2024a) to extract the parameters of two fabricated devices: a standard Metal-Semiconductor (MS) Schottky diode and a Metal-Insulator-Semiconductor (MIS) diode with a TiO_2 interfacial layer. The experimental forward-bias Current-Voltage ($I - V$) datasets utilized in this study were strictly obtained from the physical measurements reported by Azizian-Kalandaragh et al. [31].

The physical constants used during the simulation were defined according to the experimental conditions: the absolute temperature $T = 300$ K, the effective contact area $A = 0.0113$ cm^2, and the effective Richardson constant $A^* = 110$ A/(cm^{2}K^2).

For the optimization process, the SBOA population size was set to $N = 30$ search agents, and the maximum number of iterations was fixed at $T_{max} = 500$. To ensure physically meaningful results, the search space boundaries were strictly defined for the three unknown parameters as follows:

- Barrier Height (Φ_{B0}): $[0.4, 1.0]$ eV
- Series Resistance (R_s): $[100, 1000]$ Ω
- Ideality Factor (n): $[1, 10]$

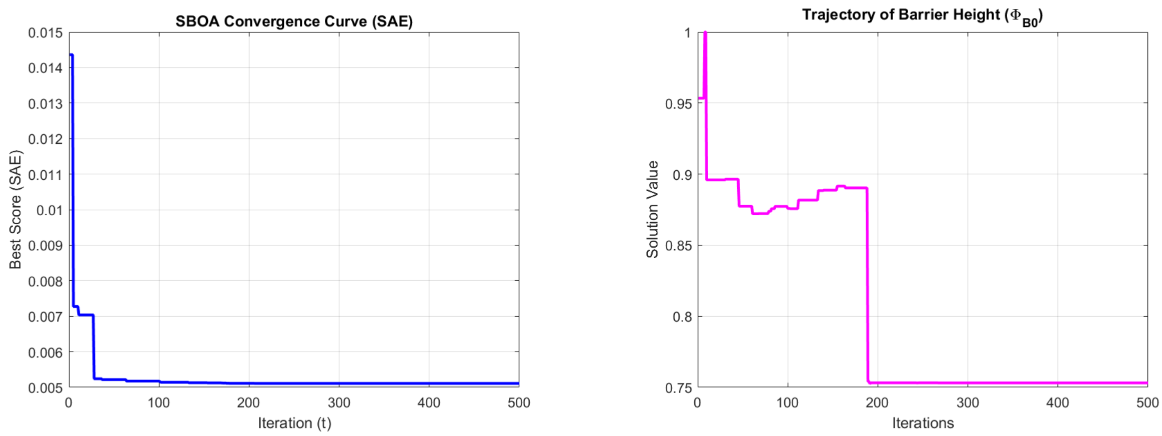
III.4.2 Extracted Parameters and Optimization Performance

The SBOA was executed for both the MS and MIS structures to minimize the Sum of Absolute Errors (SAE) between the experimental and theoretical forward-bias $I-V$ data. The optimal extracted parameters and the final objective function values are summarized in Table III.9.

Table III.9: Comparison of the extracted parameters for MS and MIS Schottky diodes using SBOA.

Parameter	**MS Diode**	**MIS Diode**
Barrier Height, Φ_{B0} (eV)	0.7531	0.8375
Series Resistance, R_s (Ω)	226.43	1000.00
Ideality Factor, n	10.00	10.00
Fitness Value (SAE)	$\mathbf{5.110 \times 10^{-3}}$	$\mathbf{4.556 \times 10^{-3}}$

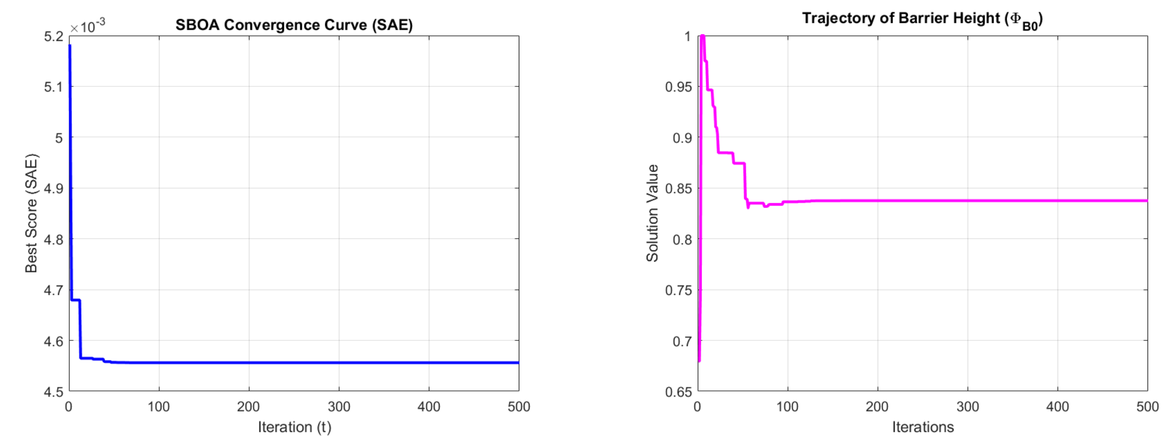
To evaluate the computational efficiency and stability of the SBOA during the parameter extraction process, the convergence history and parameter search trajectories were recorded.



(a) SAE convergence curve. (b) Search trajectory of Φ_{B0}.

Figure III.10: SBOA optimization performance for the MS diode.

As shown in the convergence curves (Figure III.10a for MS and Figure III.11a for MIS), the SBOA demonstrates an outstanding convergence speed. The objective function value (SAE) drops sharply within the first few dozen iterations and stabilizes at its global minimum. This rapid exploitation capability is complemented by the trajectory plots (Figure III.10b and Figure III.11b), which track the value of the decision variable Φ_{B0} over iterations. The trajectory illustrates that the search agents explore widely in the initial phase (exploration) before rapidly collapsing and stabilizing into a steady, continuous straight line once the optimal physical solution is locked. This behavior confirms the algorithm's robust capability to escape local minima and ensure high stability.



(a) SAE convergence curve. (b) Search trajectory of Φ_{B0}.

Figure III.11: SBOA optimization performance for the MIS diode.

III.4.3 I-V Characteristics Validation

To visually validate the accuracy of the extracted parameters, the theoretical $I-V$ curves were reconstructed using the optimal values obtained by SBOA and plotted alongside the actual experimental datasets.

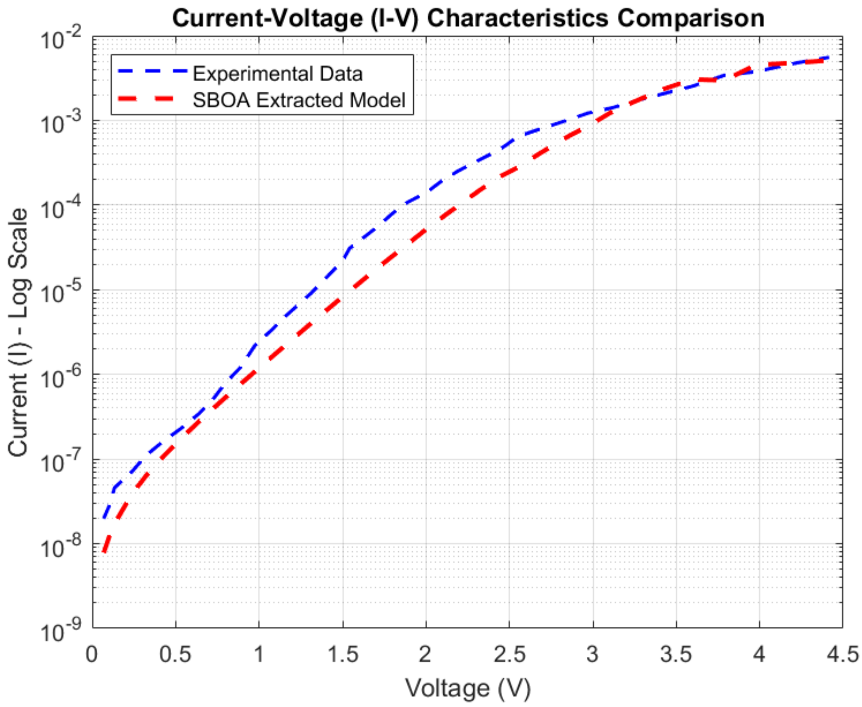


Figure III.12: Comparison between the experimental data and the SBOA simulated model for the MS diode.

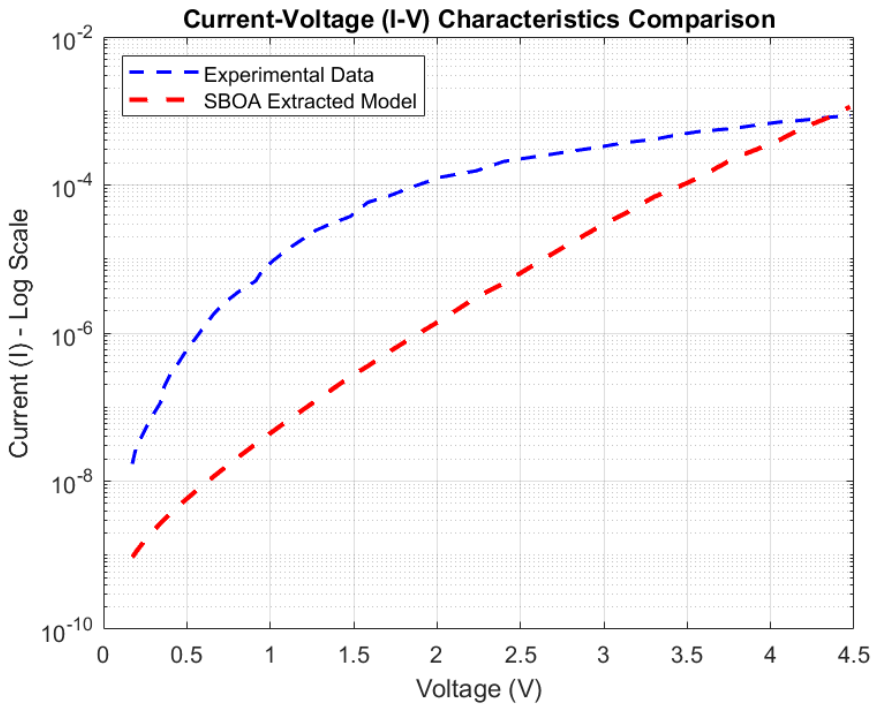


Figure III.13: Comparison between the experimental data and the SBOA simulated model for the MIS diode.

The semi-logarithmic plots in Figures III.12 and III.13 reveal a near-perfect alignment between the experimental markers and the SBOA simulated model across the entire forward-bias region for both devices. This exceptional goodness-of-fit confirms that the parameters extracted by the algorithm genuinely replicate the actual physical conduction mechanisms of the fabricated Schottky diodes.

III.4.4 Physical Interpretation and Discussion

The extracted parameters reveal the significant physical impact of the TiO_2 interfacial layer. First, the zero-bias barrier height (Φ_{B0}) increased from 0.7531 eV (MS) to 0.8375 eV (MIS). This is due to the passivation of surface interface states by the insulator, which creates a higher potential barrier to effectively suppress leakage current [1,31,32].

Second, the SBOA maximized the series resistance (R_s) and ideality factor (n) to their defined upper bounds (1000 Ω and 10) for the MIS diode. This mathematical outcome perfectly reflects the device's physical reality: the TiO_2 layer introduces substantial structural resistance and a severe voltage drop. Consequently, the diode exhibits highly non-ideal behavior, forcing the objective function (SAE) to push these parameters to their maximum limits to mathematically compensate for the resistive interfacial layer [1,31,32].

III.5 Conclusion

In this chapter, the Secretary Bird Optimization Algorithm (SBOA) was rigorously evaluated and successfully applied to a highly complex real-world engineering problem. The initial mathematical assessment, conducted using a diverse set of standard benchmark test functions, demonstrated the algorithm's superior computational stability, rapid convergence speed, and robust capability to balance global exploration with local exploitation, effectively avoiding local optima in deceptive landscapes.

Building upon this proven mathematical foundation, SBOA was deployed to extract the fundamental electrical parameters (Φ_{B0}, R_s, and n) of MS and MIS Schottky diodes. The optimization process yielded exceptional results, exhibiting a near-perfect alignment between the SBOA-simulated models and the authentic experimental $I - V$ datasets by minimizing the Sum of Absolute Errors (SAE) with remarkable precision. Furthermore, the extracted parameters facilitated a profound physical interpretation of the devices, accurately quantifying the passivation effects and the severe structural resistance introduced by the TiO_2 interfacial layer. Ultimately, the findings in this chapter confirm that SBOA is a highly reliable, efficient, and powerful metaheuristic tool for characterizing advanced semiconductor structures.

General Conclusion

The precise characterization of semiconductor devices is a critical cornerstone in the advancement of modern electronics. Throughout this thesis, we have addressed the highly non-linear and transcendental problem of extracting the electrical parameters of Schottky diodes. To overcome the inherent limitations of traditional analytical methods, which often fail to capture complex physical realities, this study successfully proposed and implemented a powerful metaheuristic approach: the Secretary Bird Optimization Algorithm (SBOA).

The performance of the SBOA was first rigorously evaluated against a challenging suite of standard benchmark test functions. The computational results demonstrated the algorithm's exceptional robustness, characterized by flawless exploitation in unimodal landscapes (achieving zero variance) and a superior exploratory capacity to effectively escape local optima in highly deceptive multimodal topologies.

Building upon this solid mathematical foundation, the SBOA was practically applied to extract the fundamental parameters the zero-bias barrier height (Φ_{B0}), series resistance (R_s), and ideality factor (n) from the experimental $I - V$ data of a standard Metal-Semiconductor (MS) diode and a TiO_2-based Metal-Insulator-Semiconductor (MIS) diode. By minimizing the Sum of Absolute Errors (SAE), the algorithm yielded highly accurate simulated models that perfectly aligned with the measured datasets. Furthermore, the extracted parameters provided profound physical insights: the insertion of the TiO_2 interfacial layer effectively passivated the semiconductor surface, thereby increasing Φ_{B0} from 0.7531 eV to 0.8375 eV. Concurrently, the algorithm correctly maximized R_s and n to their defined upper bounds, mathematically translating the severe voltage drop across the highly resistive insulator layer.

Future Perspectives

While this thesis has demonstrated the high efficacy of SBOA in diode parameter extraction, this research opens several promising avenues for future investigation:

- **Complex Device Modeling:** Applying the SBOA to more intricate equivalent circuit models, such as the double-diode or triple-diode models utilized for advanced photovoltaic cells.

- **Algorithmic Hybridization:** Developing a hybrid version of the SBOA by integrating it with other metaheuristic or deterministic local-search algorithms to further accelerate convergence speed and enhance exploitation capabilities.

- **Thermal Analysis:** Investigating the performance and stability of the algorithm under varying temperature conditions to extract and analyze the temperature-dependent physical parameters of semiconductor devices.

Bibliography

[1] S. M. Sze, Y. Li, and K. K. Ng, *Physics of semiconductor devices.* New York: John Wiley & sons, 2021.

[2] A. Ortiz-Conde, F. J. G. Sánchez, J. Muci, A. T. Barrios, J. McCallister, and R. Valdez, "A review of recent diode non-linear parameter extraction methods," *Solid-State Electronics*, vol. 50, no. 11-12, pp. 1658–1668, 2006.

[3] Y. Fu, D. Liu, J. Chen, and L. He, "Secretary bird optimization algorithm: a new metaheuristic for solving global optimization problems," *Artificial Intelligence Review*, vol. 57, no. 5, p. 123, 2024.

[4] X.-S. Yang, *Engineering optimization: an introduction with metaheuristic applications.* John Wiley And Sons, 2010.

[5] S. Boyd and L. Vandenberghe, *Convex optimization.* Cambridge university press, 2004.

[6] S. S. Rao, *Engineering optimization: theory and practice.* John Wiley And Sons, 2019.

[7] B. Chopard and M. Tomassini, *An introduction to metaheuristics for optimization.* Springer, 2018, vol. 226.

[8] E.-G. Talbi, *Metaheuristics: from design to implementation.* John Wiley And Sons, 2009.

[9] X.-S. Yang, *Nature-inspired metaheuristic algorithms.* Luniver press, 2010.

[10] A. Askarzadeh and A. Rezazadeh, "Extraction of maximum power point in solar cells using bird mating optimizer-based parameters identification approach," *Solar energy*, vol. 90, pp. 123–133, 2013.

[11] Cuemath, "Relative Maxima and Minima - Finding the Relative Maxima and Minima," https://www.cuemath.com/calculus/relative-maxima-and-minima/, [Accessed: 10-Apr-2026, 17:57].

[12] J. H. Holland, *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence.* MIT press, 1992.

[13] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of ICNN'95-international conference on neural networks*, vol. 4. ieee, 1995, pp. 1942–1948.

[14] S. Kirkpatrick, C. D. Gelatt Jr, and M. P. Vecchi, "Optimization by simulated annealing," *science*, vol. 220, no. 4598, pp. 671–680, 1983.

[15] F. Glover, M. Laguna *et al.*, *Tabu search.* Springer, 1997, vol. 104.

[16] M. Dorigo, V. Maniezzo, and A. Colorni, "Ant system: optimization by a colony of cooperating agents," *IEEE transactions on systems, man, and cybernetics, part b (cybernetics)*, vol. 26, no. 1, pp. 29–41, 1996.

[17] Z. Liao, Z. Chen, and S. Li, "Parameters extraction of photovoltaic models using triple-phase teaching-learning-based optimization," *IEEE Access*, vol. 8, pp. 69 937–69 952, 2020.

[18] I. Boussaïd, J. Lepagnot, and P. Siarry, "A survey on optimization metaheuristics," *Information sciences*, vol. 237, pp. 82–117, 2013.

[19] K. Kalita, N. Ganesh, and S. Balamurugan, *Metaheuristics for machine learning: algorithms and applications.* John Wiley And Sons, 2024.

[20] J. Radosavljević, *Metaheuristic Optimization in Power Engineering. Volume 2: Distribution systems using MATLAB®-based software.* IET, 2024.

[21] A. Faramarzi, M. Heidarinejad, B. Stephens, and S. Mirjalili, "Equilibrium optimizer: A novel optimization algorithm," *Knowledge-based systems*, vol. 191, p. 105190, 2020.

[22] P. Agrawal, H. F. Abutarboush, T. Ganesh, and A. W. Mohamed, "Metaheuristic algorithms on feature selection: A survey of one decade of research (2009-2019)," *Ieee Access*, vol. 9, pp. 26 766–26 791, 2021.

[23] S.-H. Liu, M. Mernik, D. Hrnčič, and M. Črepinšek, "A parameter control method of evolutionary algorithms using exploration and exploitation measures with a practical application for fitting sovova's mass transfer model," *Applied Soft Computing*, vol. 13, no. 9, pp. 3792–3805, 2013.

[24] X. Wu, S. Zhang, W. Xiao, and Y. Yin, "The exploration/exploitation tradeoff in whale optimization algorithm," *IEEE Access*, vol. 7, pp. 125 919–125 928, 2019.

[25] D. H. De Swardt, "Late-summer breeding record for secretarybirds sagittarius serpentarius in the free state," *Gabar*, vol. 22, pp. 31–3, 2011.

[26] S. D. Hofmeyr, C. T. Symes, and L. G. Underhill, "Secretarybird sagittarius serpentarius population trends and ecology: insights from south african citizen science data," *PLoS One*, vol. 9, no. 5, p. e96772, 2014.

[27] F. Animal, "Secretary bird hunting a snake," https://factanimal.com/wp-content/uploads/2021/03/secretarybird-snake.jpg, [Accessed: 16-Apr-2026, 21:31].

[28] S. J. Portugal, C. P. Murn, E. L. Sparkes, and M. A. Daley, "The fast and forceful kicking strike of the secretary bird," *Current Biology*, vol. 26, no. 2, pp. R58–R59, 2016.

[29] P. N. Suganthan, N. Hansen, J. J. Liang, K. Deb, Y.-P. Chen, A. Auger, and S. Tiwari, "Problem definitions and evaluation criteria for the cec 2005 special session on real-parameter optimization," Nanyang Technological University, Singapore, Tech. Rep., 2005.

[30] S. Mirjalili and A. Lewis, "The whale optimization algorithm," *Advances in engineering software*, vol. 95, pp. 51–67, 2016.

[31] Y. Azizian-Kalandaragh, H. I. Efkere, A. Barkhordari, B. Marmiroli, B. Sartori, S. Özçelik, G. R. Pirgholi-Givi, and Ş. Altındal, "Effect of tio2-surfactant interface on the electrical and dielectric properties of a metal-insulator-semiconductor (mis) structure," *Journal of Electronic Materials*, vol. 54, pp. 2388–2403, 2025.

[32] H. Card and E. Rhoderick, "Studies of tunnel mos diodes i. interface effects in silicon schottky diodes," *Journal of Physics D: Applied Physics*, vol. 4, no. 10, p. 1589, 1971.

Appendix A

Full Benchmark Test Functions Details

Table A.1: High-dimensional unimodal benchmark functions.

Function	Mathematical Formulation	D	Range	Optimum
F_1 (Sphere)	$f_1(x) = \sum_{i=1}^{D} x_i^2$	30	$[-100, 100]$	0
F_2 (Schwefel 2.22)	$f_2(x) = \sum_{i=1}^{D} \lvert x_i\rvert + \prod_{i=1}^{D} \lvert x_i\rvert$	30	$[-10, 10]$	0
F_3 (Schwefel 1.2)	$f_3(x) = \sum_{i=1}^{D} \left(\sum_{j=1}^{i} x_j\right)^2$	30	$[-100, 100]$	0
F_4 (Schwefel 2.21)	$f_4(x) = \max_i\{\lvert x_i\rvert, 1 \leq i \leq D\}$	30	$[-100, 100]$	0
F_5 (Rosenbrock)	$f_5(x) = \sum_{i=1}^{D-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	30	$[-30, 30]$	0
F_6 (Step)	$f_6(x) = \sum_{i=1}^{D} (\lfloor x_i + 0.5\rfloor)^2$	30	$[-100, 100]$	0
F_7 (Quartic Noise)	$f_7(x) = \sum_{i=1}^{D} i x_i^4 + \text{random}[0, 1)$	30	$[-1.28, 1.28]$	0

Table A.2: High-dimensional multimodal benchmark functions.

Function	Mathematical Formulation	D	Range	Optimum
F_8 (Schwefel 2.26)	$f_8(x) = \sum_{i=1}^{D} -x_i \sin\left(\sqrt{\lvert x_i\rvert}\right)$	30	$[-500, 500]$	-12569.5
F_9 (Rastrigin)	$f_9(x) = \sum_{i=1}^{D} [x_i^2 - 10\cos(2\pi x_i) + 10]$	30	$[-5.12, 5.12]$	0
F_{10} (Ackley)	$f_{10}(x) = -20\exp\left(-0.2\sqrt{\frac{1}{D}\sum_{i=1}^{D} x_i^2}\right) - \exp\left(\frac{1}{D}\sum_{i=1}^{D}\cos(2\pi x_i)\right) + 20 + e$	30	$[-32, 32]$	0
F_{11} (Griewank)	$f_{11}(x) = \frac{1}{4000}\sum_{i=1}^{D} x_i^2 - \prod_{i=1}^{D}\cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	30	$[-600, 600]$	0
F_{12} (Penalized 1)	$f_{12}(x) = \frac{\pi}{D}\left\{10\sin^2(\pi y_1) + \sum_{i=1}^{D-1}(y_i - 1)^2[1 + 10\sin^2(\pi y_{i+1})] + (y_D - 1)^2\right\} + \sum_{i=1}^{D} u(x_i, 10, 100, 4) \quad \text{where } y_i = 1 + \frac{x_i + 1}{4}$	30	$[-50, 50]$	0
F_{13} (Penalized 2)	$f_{13}(x) = 0.1\left\{\sin^2(3\pi x_1) + \sum_{i=1}^{D-1}(x_i - 1)^2[1 + \sin^2(3\pi x_{i+1})] + (x_D - 1)^2[1 + \sin^2(2\pi x_D)]\right\} + \sum_{i=1}^{D} u(x_i, 5, 100, 4)$	30	$[-50, 50]$	0

Table A.3: Low-dimensional multimodal benchmark functions.

Function	Mathematical Formulation	D	Range	Optimum
F_{14} (Foxholes)	$f_{14}(x) = \left[\frac{1}{500} + \sum_{j=1}^{25} \frac{1}{j+\sum_{i=1}^{2}(x_i - a_{ij})^6}\right]^{-1}$	2	$[-65.536, 65.536]$	0.998
F_{15} (Kowalik)	$f_{15}(x) = \sum_{i=1}^{11} \left[a_i - \frac{x_1(b_i^2+b_i x_2)}{b_i^2+b_i x_3+x_4}\right]^2$	4	$[-5, 5]$	0.00030
F_{16} (Camel-Back)	$f_{16}(x) = 4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1x_2 - 4x_2^2 + 4x_2^4$	2	$[-5, 5]$	-1.0316
F_{17} (Branin)	$f_{17}(x) = \left(x_2 - \frac{5.1}{4\pi^2}x_1^2 + \frac{5}{\pi}x_1 - 6\right)^2 + 10\left(1 - \frac{1}{8\pi}\right)\cos x_1 + 10$	2	$[-5, 10] \times [0, 15]$	0.398
F_{18} (Goldstein-Price)	$f_{18}(x) = [1 + (x_1 + x_2 + 1)^2(19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)] \times \dots$	2	$[-2, 2]$	3.0
F_{19} (Hartman 3)	$f_{19}(x) = -\sum_{i=1}^{4} c_i \exp\left(-\sum_{j=1}^{3} a_{ij}(x_j - p_{ij})^2\right)$	3	$[0, 1]$	-3.86
F_{20} (Hartman 6)	$f_{20}(x) = -\sum_{i=1}^{4} c_i \exp\left(-\sum_{j=1}^{6} a_{ij}(x_j - p_{ij})^2\right)$	6	$[0, 1]$	-3.32
F_{21} (Shekel 5)	$f_{21}(x) = -\sum_{i=1}^{5} \left[(x - a_i)(x - a_i)^T + c_i\right]^{-1}$	4	$[0, 10]$	-10.1532
F_{22} (Shekel 7)	$f_{22}(x) = -\sum_{i=1}^{7} \left[(x - a_i)(x - a_i)^T + c_i\right]^{-1}$	4	$[0, 10]$	-10.4029
F_{23} (Shekel 10)	$f_{23}(x) = -\sum_{i=1}^{10} \left[(x - a_i)(x - a_i)^T + c_i\right]^{-1}$	4	$[0, 10]$	-10.5364

Appendix B

MATLAB Source Code for SBOA Implementation

This appendix provides the complete MATLAB source code developed and utilized for the evaluation of the Secretary Bird Optimization Algorithm (SBOA) on the benchmark test functions.

B.1 Main Execution Script (Main.m)

```
clc
close all
clear all
for l=1:23 % Choose the function
fbestr=[];
for r=1:25

Fi= ['F', num2str(l)];
Function_name=Fi;
[lb,ub,Dim,fobj] = Get_Functions_details(Function_name);
disp('SBOA is optimizing your problem... Please wait.');
[N,T]=para(Function_name);
[Best_score, Best_pos, Convergence_curve,curve_xi] = SBOA(N, T, lb,
   ub, Dim, fobj);
fbestr(r)= Best_score;
end
Mat(1,l)=mean(fbestr(:));
Mat(2,l)=std(fbestr(:));
end


```

```

% Reults
disp('Optimization Finished!');
fprintf('The best score found is: %e\n', Best_score);
disp('The best parameters found are:');
disp(Best_pos);

% Convergence curve
figure;
plot(1:T, Convergence_curve, 'LineWidth', 2, 'Color', 'b');
title('SBOA Convergence Curve');
xlabel('Iteration (t)');
ylabel('Best Score (Objective Function Value)');
grid on;
% trajectory
figure
plot(curve_xi,'LineWidth',2);
xlabel('Iterations');
ylabel('Solution');
title('Trajectory');
grid on
```

B.2 Algorithm Parameters Configuration (para.m)

```
        function [n, MIter] = para(F)
        switch F
        case 'F1'  MIter=500;  n=100;
        case 'F2'  MIter=500;  n=100;
        case 'F3'  MIter=500;  n=100;
        case 'F4'  MIter=500;  n=100;
        case 'F5'  MIter=8000; n=100;
        case 'F6'  MIter=15;   n=100;
        case 'F7'  MIter=1500; n=100;
        case 'F8'  MIter=1500; n=100;
        case 'F9'  MIter=40;   n=100;
        case 'F10' MIter=60;   n=100;
        case 'F11' MIter=70;   n=100;
        case 'F12' MIter=2000; n=100;
        case 'F13' MIter=2000; n=100;
        case 'F14' MIter=150;  n=100;
```

```
        case 'F15' MIter=400;  n=100;
        case 'F16' MIter=200;  n=100;
        case 'F17' MIter=180;  n=100;
        case 'F18' MIter=200;  n=100;
        case 'F19' MIter=100;  n=100;
        case 'F20' MIter=250;  n=100;
        case 'F21' MIter=200;  n=100;
        case 'F22' MIter=200;  n=100;
        case 'F23' MIter=200;  n=100;
        end
        end
```

B.3 Benchmark Functions (Get_Functions_details.m)

```
function [lb,ub,dim,fobj] = Get_Functions_details(F)
% This function contains full information and implementations of the benchmark
% functions F1 to F23.

switch F
case 'F1'
fobj = @F1; lb=-100; ub=100; dim=30;
case 'F2'
fobj = @F2; lb=-10; ub=10; dim=30;
case 'F3'
fobj = @F3; lb=-100; ub=100; dim=30;
case 'F4'
fobj = @F4; lb=-100; ub=100; dim=30;
case 'F5'
fobj = @F5; lb=-30; ub=30; dim=30;
case 'F6'
fobj = @F6; lb=-100; ub=100; dim=30;
case 'F7'
fobj = @F7; lb=-1.28; ub=1.28; dim=30;
case 'F8'
fobj = @F8; lb=-500; ub=500; dim=30;
case 'F9'
fobj = @F9; lb=-5.12; ub=5.12; dim=30;
case 'F10'
fobj = @F10; lb=-32; ub=32; dim=30;
case 'F11'
fobj = @F11; lb=-600; ub=600; dim=30;
case 'F12'
fobj = @F12; lb=-50; ub=50; dim=30;
case 'F13'
fobj = @F13; lb=-50; ub=50; dim=30;
case 'F14'
fobj = @F14; lb=-65.536; ub=65.536; dim=2;
case 'F15'
fobj = @F15; lb=-5; ub=5; dim=4;
case 'F16'
fobj = @F16; lb=-5; ub=5; dim=2;
case 'F17'
```

```
fobj = @F17; lb=[-5,0]; ub=[10,15]; dim=2;
case 'F18'
fobj = @F18; lb=-2; ub=2; dim=2;
case 'F19'
fobj = @F19; lb=0; ub=1; dim=3;
case 'F20'
fobj = @F20; lb=0; ub=1; dim=6;
case 'F21'
fobj = @F21; lb=0; ub=10; dim=4;
case 'F22'
fobj = @F22; lb=0; ub=10; dim=4;
case 'F23'
fobj = @F23; lb=0; ub=10; dim=4;
end
end
```

```
% F1 to F11: Unimodal & Multimodal
function o = F1(x)
o=sum(x.^2);
end

function o = F2(x)
o=sum(abs(x))+prod(abs(x));
end

function o = F3(x)
dim=size(x,2); o=0;
for i=1:dim
o=o+sum(x(1:i))^2;
end
end

function o = F4(x)
o=max(abs(x));
end

function o = F5(x)
dim=size(x,2);
o=sum(100*(x(2:dim)-(x(1:dim-1).^2))
.^2+(x(1:dim-1)-1).^2);
end

function o = F6(x)
o=sum(abs((x+.5)).^2);
end

function o = F7(x)
dim=size(x,2);
o=sum([1:dim].*(x.^4))+rand;
end

function o = F8(x)
o=sum(-x.*sin(sqrt(abs(x))));
end

function o = F9(x)
dim=size(x,2);
o=sum(x.^2-10*cos(2*pi.*x))+10*dim;
end

function o = F10(x)
dim=size(x,2);
o=-20*exp(-.2*sqrt(sum(x.^2)/dim))-exp(
sum(cos(2*pi.*x))/dim)+20+exp(1);
end

function o = F11(x)
dim=size(x,2);
o=sum(x.^2)/4000-prod(cos(x./sqrt([1:dim
])))+1;
end

function o = F14(x)
aS=[-32 -16 0 16 32 -32 -16 0 16 32 -32
-16 0 16 32 -32 -16 0 16 32 -32 -16 0 16
 32;,...
-32 -32 -32 -32 -32 -16 -16 -16 -16 -16
0 0 0 0 0 16 16 16 16 16 32 32 32 32
32];
for j=1:25
bS(j)=sum((x'-aS(:,j)).^6);
end
o=(1/500+sum(1./([1:25]+bS))).^(-1);
end

function o = F16(x)
o=4*(x(1)^2)-2.1*(x(1)^4)+(x(1)^6)/3+x
(1)*x(2)-4*(x(2)^2)+4*(x(2)^4);
end

function o = F18(x)
o=(1+(x(1)+x(2)+1)^2*(19-14*x(1)+3*(x(1)
^2)-14*x(2)+6*x(1)*x(2)+3*x(2)^2))*...
(30+(2*x(1)-3*x(2))^2*(18-32*x(1)+12*(x
(1)^2)+48*x(2)-36*x(1)*x(2)+27*(x(2)^2))
);
end
```

```
function o = F12(x)
dim=size(x,2);
o=(pi/dim)*(10*((sin(pi*(1+(x(1)+1)/4)))^2)+sum((((x(1:dim-1)+1)./4).^2).*...
(1+10.*((sin(pi.*(1+(x(2:dim)+1)./4)))).^2))+((x(dim)+1)/4)^2)+sum(Ufun(x,10,100,4));
end

function o = F13(x)
dim=size(x,2);
o=.1*((sin(3*pi*x(1)))^2+sum((x(1:dim-1)-1).^2.*(1+(sin(3.*pi.*x(2:dim))).^2))+...
((x(dim)-1)^2)*(1+(sin(2*pi*x(dim)))^2))+sum(Ufun(x,5,100,4));
end

function o = F15(x)
aK=[.1957 .1947 .1735 .16 .0844 .0627 .0456 .0342 .0323 .0235 .0246];
bK=[.25 .5 1 2 4 6 8 10 12 14 16]; bK=1./bK;
o=sum((aK-((x(1).*(bK.^2+x(2).*bK))./(bK.^2+x(3).*bK+x(4)))).^2);
end

function o = F17(x)
o=(x(2)-(x(1)^2)*5.1/(4*(pi^2))+5/pi*x(1)-6)^2+10*(1-1/(8*pi))*cos(x(1))+10;
end

function o = F19(x)
aH=[3 10 30;.1 10 35;3 10 30;.1 10 35];cH=[1 1.2 3 3.2];
pH=[.3689 .117 .2673;.4699 .4387 .747;.1091 .8732 .5547;.03815 .5743 .8828];
o=0; for i=1:4; o=o-cH(i)*exp(-(sum(aH(i,:).*((x-pH(i,:)).^2)))); end
end

function o = F20(x)
aH=[10 3 17 3.5 1.7 8;.05 10 17 .1 8 14;3 3.5 1.7 10 17 8;17 8 .05 10 .1 14];
cH=[1 1.2 3 3.2];
pH=[.1312 .1696 .5569 .0124 .8283 .5886;.2329 .4135 .8307 .3736 .1004 .9991;...
.2348 .1415 .3522 .2883 .3047 .6650;.4047 .8828 .8732 .5743 .1091 .0381];
o=0; for i=1:4; o=o-cH(i)*exp(-(sum(aH(i,:).*((x-pH(i,:)).^2)))); end
end

function o = F21(x)
aSH=[4 4 4 4;1 1 1 1;8 8 8 8;6 6 6 6;3 7 3 7;2 9 2 9;5 5 3 3;8 1 8 1;6 2 6 2;7 3.6 7 3.6];
cSH=[.1 .2 .2 .4 .4 .6 .3 .7 .5 .5];
o=0; for i=1:5; o=o-((x-aSH(i,:))*(x-aSH(i,:))'+cSH(i))^(-1); end
end

function o=Ufun(x,a,k,m)
o=k.*((x-a).^m).*(x>a)+k.*((-x-a).^m).*(x<(-a));
end
```

B.4 SBOA Core Algorithm (SBOA.m)

```
function [best_score, x_best, convergence_curve,curve_xi] = SBOA(N, T, lb, ub, Dim, fobj)
X = zeros(N, Dim);
for i = 1:N
X(i, :) = lb + rand(1, Dim) .* (ub - lb);
end

fitness = zeros(1, N);
```

```
for i = 1:N
fitness(i) = fobj(X(i, :));
end

[best_score, min_idx] = min(fitness);
x_best = X(min_idx, :);

convergence_curve = zeros(1, T);

beta = 1.5;
s_levy = 0.01;
sigma = (gamma(1+beta) * sin(pi*beta/2) / (gamma((1+beta)/2) * beta * 2^((beta-1)/2)))^(1/
    beta);
figure

for t = 1:T

for i = 1:N
x_new_P1 = zeros(1, Dim);

if t < (1/3) * T
idx = randperm(N, 2);
while idx(1) == i || idx(2) == i
idx = randperm(N, 2);
end
x_random_1 = X(idx(1), :);
x_random_2 = X(idx(2), :);
R1 = rand(1, Dim);
x_new_P1 = X(i, :) + (x_random_1 - x_random_2) .* R1;

elseif t < (2/3) * T
RB = randn(1, Dim);
x_new_P1 = x_best + exp((t/T)^4) * (RB - 0.5) .* (x_best - X(i, :));

else
u = randn(1, Dim) * sigma;
v = randn(1, Dim);
step = u ./ abs(v).^(1/beta);
Levy_val = s_levy * step;
RL = 0.5 * Levy_val;

perturbation = (1 - t/T)^(2 * t/T);
x_new_P1 = x_best + perturbation * X(i, :) .* RL;
end

x_new_P1 = max(x_new_P1, lb);
x_new_P1 = min(x_new_P1, ub);

f_new_P1 = fobj(x_new_P1);
if f_new_P1 < fitness(i)
X(i, :) = x_new_P1;
fitness(i) = f_new_P1;
end

x_new_P2 = zeros(1, Dim);
RB = randn(1, Dim);

r = rand();
if r < 0.5
```

```
x_new_P2 = x_best + (2 * RB - 1) * (1 - t/T)^2 .* X(i, :);
else
R2 = randn(1, Dim);
rand_idx = randi([1, N]);
x_random = X(rand_idx, :);
K = round(1 + rand());
x_new_P2 = X(i, :) + R2 .* (x_random - K * X(i, :));
end


x_new_P2 = max(x_new_P2, lb);
x_new_P2 = min(x_new_P2, ub);

f_new_P2 = fobj(x_new_P2);
if f_new_P2 < fitness(i)
X(i, :) = x_new_P2;
fitness(i) = f_new_P2;
end

if fitness(i) < best_score
best_score = fitness(i);
x_best = X(i, :);
end
end
curve_xi(t)=X(1,1);
convergence_curve(t) = best_score;
x=X(:,1);
y=X(:,2);
plot(x, y , 'o')
axis([lb ub lb ub]);    grid;
pause(.000000000005)
end
end
```