THE PEOPLE'S DEMOCRATIC REPUBLIC OF ALGERIA

MINISTRY OF HIGHER EDUCATION AND SCIENTIFIC RESEARCH

MOHAMED BOUDIAF UNIVERSITY - M'SILA

Faculty of mathematics and informatics **Department of computer science**

Domain: Mathematics and computer Science

Thesis Presented to fulfill the partial requirement
For Master's degree in Computer Science

Specialty: Business intelligence and optimization

Prepared by: BOUDJAMLINE Rayhana

Supervise by:

CHALABI Nour elhouda

ENTITLED

A MULTI-OBJECTIVE OPTIMIZATION ALGORITHM FOR SOLVING ENGINEERING PROBLEMS

JURY MEMBERS:

- BOURAHLA Mustapha : PRESIDENT
- THARAFI Abdellah : EXAMINER

ACADEMIC YEAR 2023/2024

GRATITUDE AND APPRECIATION

At the end of this work, we must first give thanks to God who gave us the strength and courage to pursue our studies and reach this point.

Thank you very much to my supervisor, Mrs. CHALABI, who introduced me to this field, helped me immensely, and enabled me to reach this level, especially during the times we spent together. Her patience and trust in me were invaluable; without her, I would not have been able to complete this humble work.

I would also like to thank the jury members who honored us by reviewing this work.

Finally, I extend my thanks to all those who helped me directly or indirectly in carrying out this work.

Thank you all

TABLE OF CONTENTS

TABLE OF FIGURES

TABLE OF TABLES

GENERAL INTRODUCTION

A combinatorial optimization problem consists of traversing the search space to extract an optimal solution among a finite set of solutions of an often very large size such as its exhaustive enumeration is a tedious task among combinatorial problems. Solving a problem combinatorial optimization requires the use of an algorithmic process allowing the maximization or minimization of one or more “objective” functions while respecting the constraints posed by the problem. The number of “objective” functions of the problem to be treated allows to classify it as a single-objective problem or a multi-objective problem. Actually, a mono problem objective consists of optimizing a single “objective” function, while a multi-objective problem consists of optimizing two or more often-contradictory “objectives”. The resolution of this type of problem (multi-objective problems) consists of finding a compromise between the different imposed objectives. Indeed, most of these problems belong to the class of NP-hard problems and therefore do not currently have a valid efficient algorithmic solution for all data. The fact that we encounter this problem in fields of application as different as economy, industry, engineering ...etc.

Computing gives it great practical interest. The existing exact methods are limited to small instances. Today, very difficult instances are resolved using heuristic approaches, the ability of the latter to provide good quality solutions make them indispensable in the practical field and they also prove very useful for the development of exact methods based on evaluation and separation techniques.

Human beings acquire a high level of intelligence with the ability to understand, reason, recognize, learn, innovate, retain information, make decisions, communicate and solve problems. Thus, integrating human intelligence to develop the optimization technique using human problem-solving ability would definitely take the scenario to the next level, thereby promising an effective solution to real-world optimization problems. These algorithms are invoked in several real problems such as engineering, economics and logistics...etc.

While multi-objective evolutionary algorithms have the ability to find the Pareto optimal set, according to the No Free Lunch theorem [1], there is no single technique that can solve all optimization problems of Equally, then the power of an optimizer to solve a specific problem does not prove its ability to solve another. This theorem allowed researchers to propose and improve algorithms, because there is no standard solution. In view of this theorem in this work, we propose a multi-objective optimization algorithm called Multi-objective Chef Based Optimization Algorithm (MOCBOA) based on the single-objective version Chef Based Optimization Algorithm (CBOA) [2]. The latter is a population-based algorithm

In order to solve difficult multi-objective optimization problems and find a set of better-quality solutions, the main points added in our MOCBOA algorithm are:

- The Pareto dominance method has been integrated into the CBOA algorithm to extract non-dominated solutions from the population.
- An external archive was added to the CBOA algorithm in order to save non-dominated solutions.
- The dominance relationship was integrated in order to update the solutions in the archive with the best solutions while guaranteeing the diversity and convergence of the Pareto optimal solutions.

The organization of this dissertation is as follows:

- **Chapter 01: Discusses the general presentation of combinatorial optimization.**
- **Chapter 02: Optimization problem-solving approaches**
- **Chapter 03: Chef-based optimization algorithm**

- **Chapter 04: Explains the MOCBOA algorithm implementation with an application example.**

CHAPTER 01:

COMBINATORIAL OPTIMIZATION PROBLEM

1. INTRODUCTION

A combinatorial optimization problem consists of finding the best solution in a discrete set of solutions called the set of feasible solutions. In general, this set is finite but of very large cardinality.

In this chapter, we will present some basic concepts of optimization combinatorial relative basic elements, which we will use in our study. we add some examples of these combinatorial optimization problems with their algorithmic complexities and the tools for modelling the problems of combinatorial optimizations then exact and approximate resolution methods. In addition, present the context of multi-objective optimization and its main definitions.

2. DEFINITION OF AN OPTIMIZATION PROBLEM

An optimization problem is defined by:

1. A decision space: configuration solution space made up of different values taken by the decision variables.
2. One or more function(s) called objective(s) to be optimized (minimize or maximize).
3. A set of constraints to respect.

3. COMBINATORIAL OPTIMIZATION PROBLEM DEFINITION

In its most general form, a combinatorial optimization problem (subset with a finite number of solutions of discrete optimization) consists of finding in a discrete set one among the best feasible subsets (or solutions), the notion best solution being defined by an objective function. Formally, given:

- a discreet set N finished
- a set function $f : 2^N \rightarrow R$ called the objective function
- and a set R of subsets of N, the elements of which are called the feasible solutions

A combinatorial optimization problem consists of determining

$$\left\{ \begin{array}{l} \text{Max } \{f(S): S \in R \\ S \subset N \end{array} \right\} \quad (1)$$

We quickly present here four classic combinatorial optimization problems:

The backpack problem, the assignment problem, the traveling salesman problem and the scheduling problem.

4. COMPONENTS OF A COMBINATORIAL OPTIMIZATION PROBLEM

- A search (decision) space: set of solutions or configurations made up of the different values taken by the decision variables.
- Objective function: one or more so-called objective function(s), to be optimized (minimize or maximize).
- Constraints: a set of constraints to respect.

In most problems, the state (decision) space is finite or countable. The problem variables can be of various nature (real, integer, Boolean, etc.) and express quantitative data. The objective function represents the goal to be achieved for the decision maker.

The constraint set defines conditions on the state space that variables must satisfy. These constraints are often constraints of inequality or equality and allow in general to limit the search space (feasible solutions).

5. CHARACTERISTICS OF A COMBINATORIAL OPTIMIZATION PROBLEM

- Number of decision variables:
  - A ⇒ mono variable.
  - Several ⇒ multi variable.
- Type of decision variable:
  - Continuous real number ⇒continuous.
  - Integer⇒ integer or discrete.
  - Permutation on a finite set of numbers⇒ combinatorial.
- Type of objective function:
  - Linear function of decision variables⇒ linear.
  - Quadratic function of decision variables⇒ quadratic.
  - Nonlinear function of decision variables⇒ nonlinear.
- Problem formulation:
  - With constraints ⇒ constrained.
  - Without constraints ⇒ not constrained.

6. SOLVING A COMBINATORIAL OPTIMIZATION PROBLEM

Solving the combinatorial optimization problem requires the study of three point's individuals:

- The definition of all feasible solutions,
- The expression of the objective to be optimized,
- The choice of the optimization method to use,

The first two points relate to the modelling of the problem, the third to its resolution. In order to define all the feasible solutions, it is necessary to express all the constraints of the problem. This can only be done with good knowledge of the problem under study and its field of application.

7. TYPES OF OPTIMIZATION METHODS

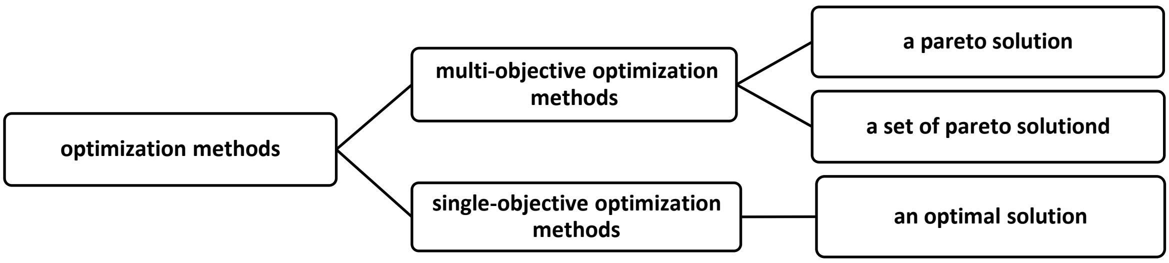

Figure 1.1: Types of optimization methods and their targeted solutions.

8. SINGLE-OBJECTIVE OPTIMIZATION PROBLEMS

A single-objective optimization problem (SOOP) has the objective function ($f(\vec{x})$), which must be minimized or maximized and a number of constraints ($g(\vec{x})$). Equation (2) shows the formula of the SOOP in its general form.

$$\left(\begin{array}{ll} \text{Minimize } f(\vec{x}) & \\ \text{s.t.} & g_j(\vec{x}) \geq 0 \ (j = 1,\ldots, m) \\ \vec{x} \in X \subset R^n & \end{array}\right) \quad (2)$$

Where $\vec{x}$ is a vector of n decision variables, $\vec{x} = (x1, x2,\ldots, x_n)^T$, and X represents a feasible region.

In single-objective optimization (SOO) problem deals with the maximization or minimization of the objective function based upon a single variable given a constraint or an unconstrained problem. The SOO problems have a single variable in the given objective function. The function may vary according to the different values of that variable. The function may have i) Relative or Local Minimum ii) Relative or Local Maximum iii) Absolute or Global Minimum iv) Absolute or Global Maximum. The applications of SOO are related to less complex real time problems. However, at small levels too optimization is needed

9. MULTI OBJECTIVE OPTIMIZATION PROBLEM

Multi-objective optimization is a branch of combinatorial optimization in this type of optimization; we seek to satisfy several objectives at the same time. Thus, the multicriteria optimization problem

Can be put in the following form:

$$\left\{ \begin{array}{c} \text{Max, Min } (f_1(x); f_2(x) \ldots f^p(x))^T,\ p \geq 2 \\ g_i(x) \leq 0,\ j=1;\ 2; \ldots m \end{array} \right\} \quad (3)$$

10. CONCEPT OF DOMINANCE AND DOMINANCE OPTIMALITY

a) Dominance

We say that the decision vector u= $[u_1,\ldots,u_k]^T$ dominates the vector v= $[v_1,\ldots,v_k]^T$ (denoted: u<v), if and only if :

$\forall_i \in \{1,2,\ldots,k\}, f_i(u) f_i(v) \cap \exists_i \in \{1,2,\ldots,N\} f_i(u) < f_i(v)$

b) Pareto set

The Pareto set is the set of optimal solutions in the sense of Pareto, it is also called the set of non-dominated solutions or the set of compromises or the set of efficient solutions.

c) Pareto frontier

The Pareto frontier is the image of the Pareto set in the space of feasible criteria. It is also called the Pareto surface or the Pareto front.

11. COMPLEXITY OF A COMBINATORIAL OPTIMIZATION PROBLEM

Here we address addressing the complexity, definitions and classifications of problems as follows (Figure 1.2).

1. **P Class:**

The class P is the set of languages decidable in polynomial time by a deterministic Turing machine. It is included in NP

2. NP Class:

The class NP is the set of languages decidable in polynomial time by a non-deterministic Turing machine.

3. NP-complete Class:

In complexity theory, an NP-complete problem or NPC problem (i.e., a complete problem for the class NP) is a decision problem satisfying the following properties:

It is possible to verify a solution efficiently (in polynomial time); the class of problems verifying this property is denoted NP.

All the problems of the NP class are reduced to this one via a polynomial reduction; this means that the problem is at least as difficult as all other problems in the NP class.

4. NP-hard Class:

An NP-hard problem is, in complexity theory, a problem belonging to the NP-hard class, which amounts to saying that it is at least as difficult as the most difficult problems of the NP class.

If an NP-hard problem is in NP, then it is an NP-complete problem. So, all NP-complete problems are NP-hard.

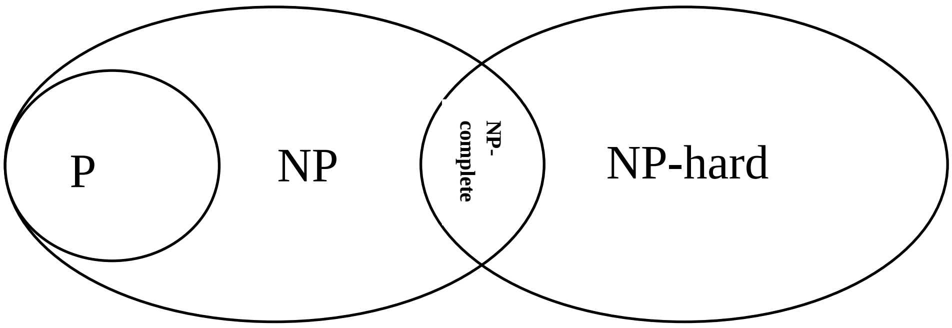

Figure 1.2: Relationship between problem classes.

12.EVOLUTIONARY ALGORITHMS

When solving a multi-objective optimization problem, evolutionary algorithms are a popular method for producing Pareto optimal solutions. The majority of methods for evolutionary multi-objective optimization (EMO) utilize Pareto-based ranking schemes.

Evolutionary algorithms including Strength Pareto Evolutionary Algorithm 2 (SPEA-2) [3], Multiobjective Differential Evolution versions, and the Non-dominated Sorting Genetic Algorithm-II (NSGA-II) [4] and its expanded version, (NSGA-III) [5], have become standard methods. When evolutionary algorithms are utilized to address multi-objective optimization problems, their primary benefit is in their ability to produce sets of solutions that facilitate the computing of an approximate Pareto front.

13.CONCLUSION

In this chapter we introduced the basic concepts of optimization and combinatorial optimization that are essential to understanding our work. We mentioned the types of optimization methods and provided a definition for both single-objective optimization problems and multi-objective optimization problems. In addition to this, we provided an explanation of dominance and dominance optimality. Finally, we added a definition of evolutionary algorithms.

CHAPTER 02:

OPTIMIZATION PROBLEM-SOLVING APPROACHES

1. INTRODUCTION

The necessity to handle numerous problems faced in daily life has prompted researchers to propose and modify solution methods, with the goal of improving performance in terms of calculation time and solution quality. Throughout the years, different ways for addressing issues of varying complexity have been developed, resulting in a wide range of tactics and performance levels. These methods can be broadly classified into two types: exact methods and approximation methods.

Metaheuristic algorithms have numerous applications in solving optimization problems. Integrating human problem-solving abilities into these algorithms may improve the solutions to real-world optimization problems. Human behavior and evolution enable individuals to adapt to their circumstances at speeds that frequently outperform those of natural evolutionary processes.

2. METHODS FOR SOLVING MULTI-OBJECTIVE PROBLEMS

Multi-criteria optimization deals with problems that require the simultaneous optimization of several objectives. These methods can be classified according to a specific framework, as illustrated in (Figure 2.3).

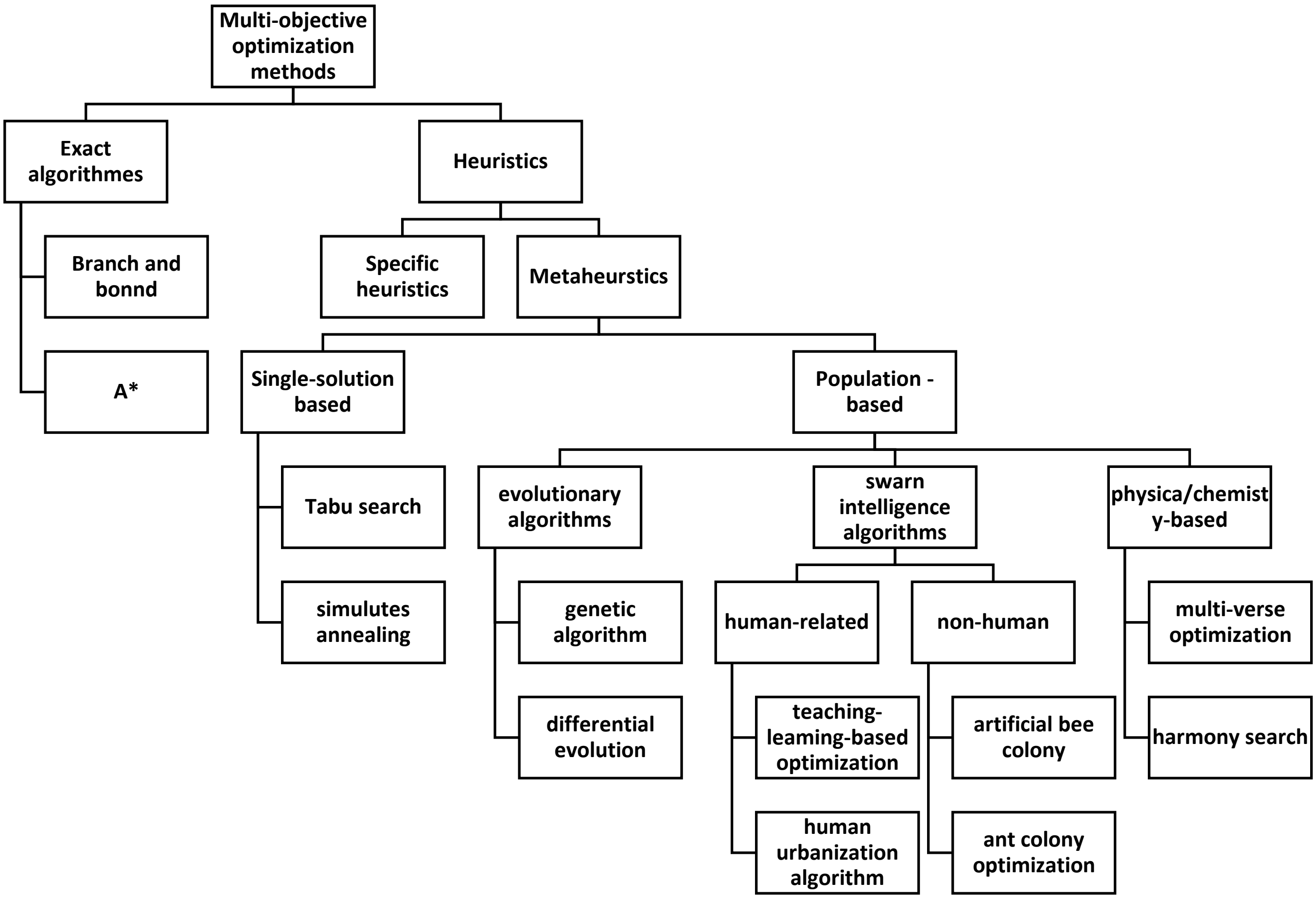

Figure 2.3: classification of multi-objective optimization methods

2.1. EXACT METHODS FOR OPTIMIZATION

The principle of exact methods consists of searching, often implicitly, for a solution, the best solution or all of the solutions to a problem. Optimization exact concerns all the methods making it possible to obtain a result which we know to be optimal for a specific problem. This ranges from simplex methods to Lagrangian methods. Through dynamic programming. The exact methods can be classified into four large classes:

- Dynamic programming,
- Continuous linear or integer programming,
- Nonlinear programming with or without constraints,
- Tree search methods (Branch & Bound).

In order to find accurate solutions to issues that are known to be tough,[6] more recent research has made it possible to build dynamic programming and linear programming in integers. However, these approaches are not always simple to employ, especially when it comes to describing the problems that are likely to be treated. In fact, the size of the issues and the calculation times which might be prohibitive before the algorithm achieves the exact answer limit the exact approaches even though the danger of the calculation time required to discover a solution growing exponentially with the problem's dimensions.

2.2. METAHEURISTIC METHODS

Metaheuristic constitute a class of methods which provide good quality solutions in reasonable time to combinatorial problems known to be difficult for which no more efficient classical method is known. We call Metaheuristic (from the Greek, Meta = which encompasses) methods designed to escape local minima. The term Meta is also explained by the fact that these methods are general structures whose components must be instantiated according to the problem, for example, the neighbourhood, the starting solutions or the stopping criteria. Metaheuristic are generally iterative stochastic algorithms which progress towards a global optimum, that is to say the global extremum of a function by evaluating a certain objective function. They behave like search algorithms, attempting to learn the characteristics of a problem in order to find an approximation of the best solution in a manner close to approximation algorithms. The growing interest in Metaheuristic is entirely justified by the development of machines with enormous computational capacities, which has made it possible to design increasingly complex Metaheuristic which have demonstrated a certain efficiency when solving several NP-hard problems.

There are many Metaheuristic ranging from simple local search to more complex global search algorithms. However, these methods use a high level of abstraction allowing them to be adapted to a wide range of combinatorial optimization problems.

3. HUMAN-INSPIRED ALGORITHMS

Human-inspired algorithms are a subclass of metaheuristics that use human behavior and cognitive processes to address optimization challenges. These algorithms take advantage of human-like processes for social interaction, decision-making, learning, and problem-solving. Integrating these human-like behaviors into optimization algorithms may improve their performance and relevance to real-world situations.

Key Human-Inspired Algorithms:

Human-inspired algorithms have evolved to address complex optimization problems by mimicking various human strategies. Some well-known human-inspired algorithms include:

Table 2.1: Human-Inspired Optimization Algorithms

SI	Name of the HIOA	Year	Author
1	Cultural Algorithm [7]	1994	Reynolds
2	Harmony Search Algorithm [8]	2001	Geem et al.
3	Society and Civilization [9]	2003	Ray et al.
4	Seeker Optimization Algorithm [10]	2006	Dai et al.
5	Imperialist Competitive Algorithm [11]	2007	Gargari and Lucas
6	League Championship Algorithm [12]	2009	Kashan
7	Group Counselling Optimization Algorithm [13]	2010	Eita et al.
8	Anarchic Society Optimization [14]	2011	Ahmadi
9	Cohort Intelligence [15]	2013	Kulkarni et al.
10	Exchange Market Algorithm [16]	2014	Ghorbani and Babaei
11	Election Algorithm [17]	2015	Emami et al.
12	Passing Vehicle Search [18]	2016	Savsani and Savsani
13	Human Mental Search [19]	2017	M.J. Mousavirad

14	Social Engineering Optimizer [20]	2018	Amir Mohammad Fathollahi-Fard
15	Future Search Algorithm [21]	2019	Elsisi
16	Political Optimizer [22]	2020	Qamar Askari et al.
17	Coronavirus Herd Immunity Optimization [23]	2021	Mohammed Azmi Al-Betar
18	Stock Exchange Trading Optimization [24]	2022	Emami

4. ADVANTAGES AND LIMITATIONS OF HUMAN-BASED ALGORITHMS

Human-based metaheuristic algorithms are designed based on modelling communication, interactions, thoughts, decisions, and strategies of humans in their social and individual lives.

4.1. ADVANTAGES

- Human-based algorithms can handle complex decision-making processes that may be difficult for purely automated algorithms.
- Humans are adept at adapting to changing circumstances, making human-based algorithms suitable for dynamic environments where rules or conditions may evolve over time. This adaptability can lead to more robust and flexible systems.
- Human-based algorithms often produce results that are easier to interpret and explain compared to purely automated algorithms. This is particularly important in domains where transparency and understanding of decision-making processes are crucial, such as healthcare and law.
- Humans possess creativity and lateral thinking abilities that are challenging to replicate in purely automated systems. Human-based algorithms can leverage these capabilities to generate innovative solutions to complex problems.

4.2. LIMITATIONS

- Human-based algorithms are susceptible to human biases, prejudices, and subjectivity. The decisions made by humans may vary based on individual perspectives, experiences, and cognitive biases, leading to inconsistent or unfair outcomes.

- Depending on the level of human involvement, human-based algorithms may not scale efficiently to handle large volumes of data or tasks. Human labour can be time-consuming and costly, limiting the scalability of such systems.
- Human-based algorithms may not be feasible or practical for certain tasks or domains where human judgment is difficult to quantify or where automation is preferred for efficiency or consistency reasons.

5. BASIC PRINCIPLES OF HUMAN-BASED ALGORITHMS

Human-based algorithms draw inspiration from the cognitive processes and social behaviors found in humans. These algorithms seek to emulate human problem-solving processes in order to identify optimal or near-optimal solutions to complex optimization issues. This section digs into the underlying ideas that underpin human-based algorithms, focusing on their methods, structures, and common applications.

5.1. Human Cognition & Learning

Human-based algorithms simulate cognitive processes such as perception, memory, reasoning, and problem solving in order to improve optimization efficiency and effectiveness. The key principles include:

Memory Utilization: Keeping track of previously investigated solutions can help guide future searches and avoid repeated assessments.

Learning and Adaptation: Using learning mechanisms to adjust in response to environmental feedback, analogous to how humans learn from their experiences.

5.2. Social Interaction and Collaboration

Human-based algorithms improve performance by modeling social behaviors:

Collaboration and teamwork: Using group decision-making techniques, such as Group Counseling Optimization, to develop strong and innovative solutions.

Balancing rivalry and cooperation among agents, as demonstrated by competitive and cooperative learning processes.

5.3. Strategies for Decision Making and Problem Solving

These algorithms use human decision-making processes to explore complex search areas.

Heuristics and rule of thumb: The Human Mental Search algorithm serves as an example of using simple, efficient decision-making criteria.

Scenario analysis and simulation involve simulating several situations to analyse probable outcomes, similar to human mental simulations.

5.4. Emotional and psychological factors

Some algorithms include human emotional and psychological components:

Motivation and Reward Systems: Use reward systems to incentivize desirable actions, comparable to reinforcement learning.

Stress and weariness: Human weariness and stress are simulated to represent realistic search behaviors, minimizing overexploitation and encouraging exploration.

The optimization process typically includes:

1. **Initialization:** A random generation of candidate solutions.
2. **Evaluation:** involves determining the suitability of each possible solution.
3. **Iteration:** is the process of improving solutions incrementally using predetermined procedures.
4. **Termination:** refers to the end of the search process when stopping requirements are satisfied, such as a maximum number of iterations or a suitable fitness level. (Figure 2.4) shows the optimization process.

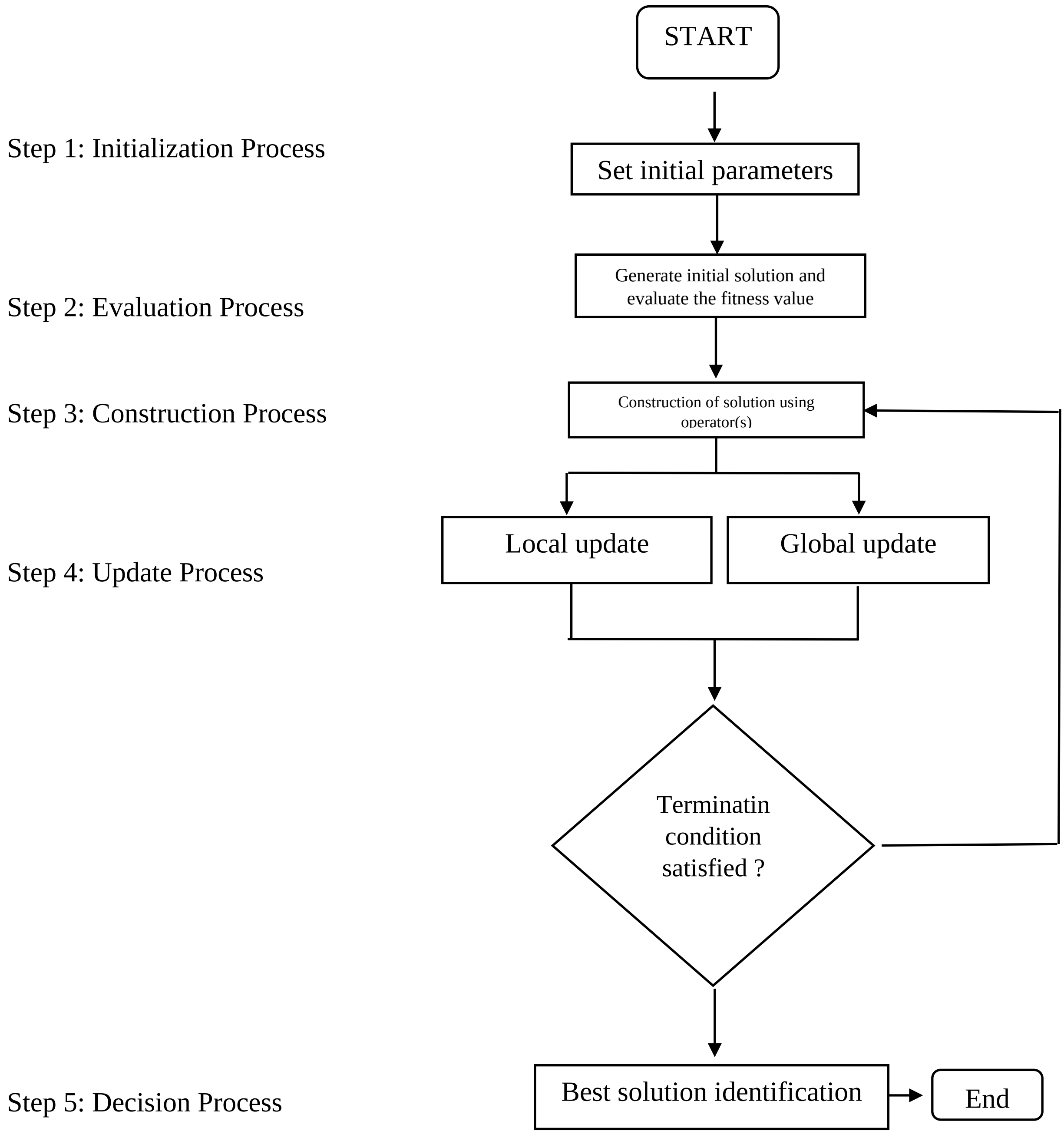

Figure 2.4: Flowchart depicting common structure of HIOAs

6. ENGINEERING APPLICATIONS OF OPTIMIZATION

Optimization is critical in engineering since it allows researchers to develop and enhance systems, processes, and products by identifying the optimal solutions within given limitations. Engineers can improve performance, save costs, increase efficiency, and assure the long-term viability of their designs by utilizing optimization techniques. The significance of optimization in engineering cannot be emphasized enough, since it has a direct impact on the quality and viability of engineering solutions in practical applications.

Here are a few common applications from several engineering disciplines to show the subject's broad reach.

- Design of civil engineering structures such as frames, foundations, bridges, towers, chimneys and dams for minimum cost
- Design of minimum weight structures for earth quake wind and other types of random loading
- Optimal plastic design of frame structures (e.g., to determine the ultimate moment capacity for minimum weight of the frame).
- Design of water resources systems for obtaining maximum benefit.
- Design of optimum pipeline networks for process industry.
- Design of aircraft and aerospace structure for minimum weight
- Finding the optimal trajectories of space vehicles.
- Optimum design of linkages, cams, gears, machine tools, and other mechanical components.
- Selection of machining conditions in metal-cutting processes for minimizing the product cost.
- Design of material handling equipment such as conveyors, trucks and cranes

7. CONCLUSION

In this chapter, we presented some problem-solving methods for combinatorial optimization we found that the exact methods allow to arrive at the optimal solution, but they are too demanding in terms of calculation time and memory space required. However, the approximate methods require research costs reasonable. But they do not guarantee the optimality of the solution. We could note that approximate methods can be divided into two classes: heuristic methods and meta-heuristic methods. A heuristic method is applicable to a given problem. While a Meta heuristic method is more generic and can be applied to a range of optimization problems.

And also, a definition of human-inspired algorithms with some basic concepts to finally discover the importance and advantages of human-based algorithms.

CHAPTER 03:

Chef-based optimization algorithm

1. INTRODUCTION

Metaheuristic algorithms are commonly used for solving optimization problems. This work introduces the Chef-Based Optimization Algorithm (CBOA)[2], a novel metaheuristic based on the process of learning to cook in culinary school. The method mathematically models the cooking training process to improve both local search (exploitation) and global search (exploration). The effectiveness of CBOA is assessed using 52 common objective functions, which show a high efficiency in balancing exploration and exploitation to achieve optimal results.

CBOA's innovation lies in its novel metaheuristic algorithm and systematic approach to culinary training.

Training Process Model: Based on the phases of training new chefs, the study proposes a mathematical model that reflects the dynamics of culinary education.

- **Phase One:** Chefs compete to establish quality ratings, demonstrating the competitive nature of cooking mastery.
- **Phase Two:** Students compete depending on their cooking abilities to determine their own quality scores.

Master Chef Methodologies:

- **Strategy 1:** Chefs pick up new skills from others.
- **Strategy 2:** Chefs do separate experiments to improve previously learnt procedures.

Student tactics:

- **Strategy 1:** Students learn all methods from a single chef.
- **Strategy 2:** Students learn specific skills from several chefs.
- **Strategy 3:** Students conduct self-experiments to improve their skills.

2. CHEF-BASED OPTIMIZATION ALGORITHM

This section introduces the Chef-Based Optimization Algorithm (CBOA), an innovative approach inspired by culinary students & training processes. Culinary schools offer training programs to help students improve their skills and become proficient chefs. This iterative refinement technique is similar to metaheuristic algorithms, in which potential solutions are proposed and revised to determine the optimal one.

CBOA s framework reflects the evolution of student cooks into chefs, with numerous key components:

- **Chef Teachers:** A culinary school has numerous chef teachers, each in charge of guiding a class.

- **Class Selection:** Students can choose which classes to attend, where they will learn various cooking skills and approaches from their chef instructors.
- **Skill Refinement:** Chef teachers continually improve their skills through individual practice and guidance from the top chef instructor.
- **Student Learning:** Cooking students aspire to emulate and perfect the techniques taught by their chef instructors, improving their talents through practice.
- **Graduation:** Culinary students graduate from the program as proficient chefs, equipped with the skills learned and refined during their training.

3. ALGORITHM INITIALIZATION

The population-based algorithm that makes up the suggested CBOA technique is made up of two categories of people: chef instructors and cookery students. Every candidate solution that has details about the issue variables is a member of the CBOA. Each member of the CBOA is a vector from a mathematical perspective, and the set of CBOA members may be represented by a matrix in accordance with Equation (1)

$$X = \begin{bmatrix} X_1 \\ X_i \\ X_N \end{bmatrix}_{N*m} = \begin{pmatrix} x1,1 & x1,j & x1,m \\ xi,1 & xi,j & xi,m \\ xN,1 & xN,j & xN,m \end{pmatrix}_{N*m} \quad (1)$$

Where N is the population size, m is the number of problem variables of the objective function (dimension of the problem), X is the CBOA population matrix, X_i is the I th CBOA member (candidate solution), and $x_{i,j}$ is its j th coordinate (i.e., the value of the j th problem variable for the I th CBOA member).
At the start of the algorithm's execution, the positions of the CBOA members are randomly initialized for I = 1, 2... Nandj = 1, 2..., M using Equation (2)

$$X_{ij} = lb_j + r.(ub_j - lb_j) \quad (2)$$

Where r is a random number in the interval [0, 1], lb_j and ub_j are the lower and the upper bounds of the j th problem variable, respectively.

An objective function value is calculated by substituting each CBOA member's recommended value into the variables. Consequently, N values are computed for the objective function and the objective function is evaluated in N turns (where N is the number of CBOA members). A vector that corresponds to Equation (3) can be used to represent these values.

$$F = \begin{pmatrix} F_1 \\ F_i \\ F_N \end{pmatrix}_{N*1} = \begin{pmatrix} F(X_1) \\ F(X_i) \\ F(X_N) \end{pmatrix}_{N*1} \quad (3)$$

Where Fi is the value of the objective function derived for the I th member of the CBOA, where I = 1,2,…,N and F is the vector of values of the objective function. Important details on the calibre of the potential solutions are provided by the objective function values. The criterion for choosing the optimal candidate solution is the objective function's value. The best member of the population and the best candidate solution among CBOA members is identified as the member with the best value for the objective function. Every time the algorithm runs, the members of the CBOA are updated, and the goal function's associated values are determined. Therefore, it is essential to update the top participant in each iteration based on comparing the values of the objective function

4. MATHEMATICAL MODELING OF CBOA

Once launched, the Chef-Based Optimization Algorithm (CBOA) improves candidate solutions over a number of steps. CBOA is divided into two groups: chef teachers and cooking students, with each following its own set of updating methods.

1. **Group Formation:**
   - Chef Instructors: Selected based on higher objective function values.
   - Cooking Students: Comprise the remaining members with lower objective function values.
2. **Selection Process:**
   - Members are ranked by their objective function values in ascending order.
   - The top NC members form the chef instructors group.
   - The remaining N − NC members become the cooking students group.
3. **Mathematical Representation:**
   - Objective function values and population matrices are sorted and represented in specific equations to facilitate the selection and updating processes.

This systematic strategy assures that the top performers guide others' learning and progress, iteratively optimizing overall solution quality.

$$XS = \begin{bmatrix} XS_1 \\ XS_{NC} \\ XS_{Nc+1} \\ XS_N \end{bmatrix}_{N*m} = \begin{bmatrix} xs_{11} & xs_{1j} & xs_{1m} \\ xs_{Nc,1} & xs_{Ncj} & xs_{Nc,m} \\ xs_{Nc+1,1} & xs_{Nc+1j} & xs_{Nc+1,m} \\ xs_{N1} & xs_{NJ} & xs_{Nm} \end{bmatrix}_{N*m} \quad (4)$$

$$FS = \begin{bmatrix} FS_1 \\ FS_{NC} \\ FS_{NC+1} \\ FS_N \end{bmatrix}_{N*m} \quad (5)$$

Where NC is the number of chef instructors, XS is the sorted population matrix of CBOA, and FS is a vector of ascending objective function values. In the matrix XS, members from XS_1 to XS_{NC} represent the group of chef instructors and members from XS_{NC+1} to XS_N represent the group of cooking students. The vector FS i includes successively the values of the objective functions corresponding to XS_1 to XS_N.

PHASE 1: THE UPDATING PROCESS FOR GROUP OF CHEF INSTRUCTORS (UPDATE OF XS1 TO XSNC).

In a culinary school, many chef professors are supposed to teach students cooking techniques. Chef teachers build their culinary skills through two basic ways. The first tactic entails modeling themselves after the best chef teacher to learn their techniques, demonstrating CBOA's investigation and worldwide search capabilities. This strategy enhances the algorithm by allowing the greatest cooks to develop their abilities before training students, avoiding relying exclusively on the top population member to upgrade student positions. This strategy also keeps the algorithm from becoming stuck in local optima, resulting in a more precise and efficient scan of various sections of the search space. The new locations for each chef teacher are determined by a specific equation for i = 1,2,...,NC and j = 1,2,...,m.

$$Xs^{c/S1}_{ij}=xs_{ij}+r.\ (BC_j - I.\ xs_{ij}) \quad (6)$$

is the new calculated status for the ith sorted member of CBOA (that is XS_i) based on the first strategy (C/S1) of updating the chef instructor, $xs^{C/S1}_{i,j}$ is its jth coordinate, BC is the best chef instructor (denoted as XS_1 in the matrix XS), BC_j is the jth coordinate of the best chef instructor, r is a random number from the interval [0,1], and I is a number that is selected randomly during execution from the set {1,2}. Tis new position is acceptable to the CBOA if it improves the value of the objective function. Tis condition is modelled

$$XS_i = \begin{cases} XS^{C/S1}_i & FS^{C/S1}_i < F_i \\ XS_i & \text{else} \end{cases} \quad (7)$$

Where $FS^{C/S1}_i$ is the value of the objective function of the member $XS^{C/S1}_i$

Using personalized activities and exercises, each chef teacher aims to enhance his cooking abilities in the second strategy. This tactic embodies both the local search and the CBOA's capacity for exploitation. In order to increase the goal function value, a chef teacher will attempt to enhance each problem variable if it is thought of as a culinary talent.

The benefit of updating based on individual workouts and activities is that every member looks for better solutions close to the location where it is located, independent of the position of other members of the population.

There's a chance that local search and exploitation, with small adjustments to population members' locations in the search space, might provide superior results.

$$lb^{local}_j = \frac{lbj}{t} \quad (8)$$

$$ub^{local}_j = \frac{ubj}{t} \quad (9)$$

Where lb^{local}_j and ub^{local}_j are the lower and upper local bound of the jth problem variable, respectively, and the variable t represents the iteration counter

$$Xs^{c/s2}_{ij} = xs_{ij} + lb^{local}_j + r.(ub^{local}_j - lb^{local}_j) ,i=1,2,\dots,Nc,j=1,2,\dots m \quad (10)$$

$$XS_i = \begin{cases} XS^{C/S2}_i , FS^{C/S2}_i < F_i \\ XS_i \quad \text{else} \end{cases} \quad (11)$$

Where $XS^{C/S2}_i$ is the new calculated status for the ith CBOA sorted member (i.e., XS_i) based on the second strategy (C/S2) of chef instructors updating, $xs^{C/S2}_{i,j}$ is its jth coordinate, and $FS^{C/S2}_i$ is its value of the objective function.

PHASE 2: THE UPDATING PROCESS FOR THE GROUP OF COOKING STUDENTS (UPDATE OF XSNC+1 TO XSN).

Culinary school students who want to be chefs employ three basic strategies to learn how to cook, which are incorporated into the Chef-Based Optimization Algorithm. The first strategy involves randomly assigning each student to a class taught by a chef instructor. This strategy assures that students learn different talents from different chefs, increasing diversity in skill development and encouraging study of diverse parts of the search domain. This diversity prevents all pupils from studying solely from the top chef, which would impede an effective worldwide search. In CBOA, this technique is emulated by computing a new position for each culinary student based on the instruction of their assigned chef teacher, where i = NC + 1, NC + 2,..., N and j = 1,2,..., m.

$$Xs^{s/s1}_{ij}=xs_{ij}+r.\ (CI_{kij}-I.\ xs_{ij}) \quad (12)$$

where $XS^{S/S1}_{i}$ is the new calculated status for the ith sorted member of CBOA (i.e., XS_i) based on the frst strategy (S/S1) of the updating of cooking students, $xs^{S/S1}_{i,j}$ is its jth coordinate, and $CI_{ki,j}$ is the selected chef instructor by the ith cooking student, where ki is randomly selected from the set {1,2, ... , NC} (where $CI_{ki,j}$ denotes the value $xs_{ki,j}$).Tis new position replaces the previous position for each CBOA member, if it improves the value of the objective function. Tis concept is modeled for i = NC + 1, NC + 2, ... ,N by Eq. (13).

$$XS_i = \begin{cases} XS^{S/S1}_{i} & FS^{S/S1}_{i} < F_{i} \\ XS_i & \text{else} \end{cases} \quad (13)$$

where $FS^{S/S1}_{i}$ is the value of the objective function of $XS^{S/S1}_{i}$

As every issue variable in the CBOA is presumed to represent a culinary talent, the second technique has each cooking student try to thoroughly understand and mimic one of the chef instructor's abilities (here, "skill" refers to a recipe for a single delicious meal). This tactic improves the CBOA's ability to search and explore globally. This strategy's benefit is that it only modifies one variable a skill, such as a recipe instead of updating all possible solution variables, which would affect all culinary student skills. Updating every member's position coordinate might not be required in order to get superior results.

$$Xs^{s/s2}_{ij} = \begin{cases} CI_{kij} & j=l \\ Xs_{ij} & \text{else} \end{cases} \quad (14)$$

where i = NC + 1, NC + 2,..., N, j = 1,2,..., m, and { is a randomly chosen integer from the set {1,2,...,m}. Ten, if it raises the goal value of the objective function; it is substituted with the prior position according to Eq. (15).

$$XS_i = \begin{cases} XS^{S/S2}_i & FS^{S/S2}_i < F_i \\ XS_i & \text{else} \end{cases} \quad (15)$$

where $XS^{S/S2}_i$ is the new calculated status for the ith sorted member of CBOA (i.e., XS_i) based on the second strategy (S/S2) of updating cooking students, $xs^{S/S2}_{i,j}$ is its jth coordinate, $FS^{S/S2}_i$ is its objective function value.Using their own tasks and exercises, each cooking student seeks to enhance their cooking talents in the third strategy. Actually, this tactic embodies both the CBOA's exploitation potential and the local search. One benefit of upgrading cookery pupils based on individual activity and exercise strategies is that it boosts the effectiveness of local search and algorithmic exploitation to get better feasible solutions close to the found answers. Cooking students use this method, which is comparable to the local search method used by chef teachers, to try to come up with better answers using small, exact procedures. In order to enhance the objective function value, a cooking student will attempt to improve each problem variable if it is thought of as a cooking skill. According to this concept, around each cooking student in the search space, a random position is generated by Eqs. (8), and (9) and a new position is calculated using Eq. (16).

$$Xs^{s/s3}_{ij} = \begin{cases} xs_{ij} + lb^{local}_j + r.(ub^{local}_j - lb^{local}_j), & j=q \\ Xs_{ij} & j\#q \end{cases} \quad (16)$$

where $xs^{C/S3}_{i,j}$ is its jth coordinate, q is a randomly selected number from the set {1,2,..., m}, i = NC + 1, NC + 2,..., N, and j = 1,2,..., m. These values represent the new calculated status for the ith-sorted member of CBOA, or XS_i, based on the third strategy (S/S3) of updating cooking students. This new random position is suitable for updating XS_i, which is described by Eq (17), if It incérasses the value of the goal Function.

$$XS_i = \begin{cases} XS^{S/S3}_i, FS^{S/S3}_i < Fi; \\ XS_i \quad \text{else} \end{cases} \quad (17)$$

Where $FS^{S/S3}_i$ is the value of the objective function of $XS^{S/S3}_i$

5. REPETITION PROCESS, AND FOWCHART OF CBOA.

Updating every person in the population completes a CBOA iteration. These new statuses are carried over to the CBOA for the upcoming iteration, and the groupings of chef instructors and cookery students are reorganized. To the last algorithm iteration, the population members are updated based on the CBOA stages as implemented in accordance with Eqs (4) to (17). When the iteration variable CBOA reaches its maximum value, the problem's solution is shown as the best candidate solution found during the implementation phase. The many stages involved in implementing CBOA are demonstrated in (Figure 3.5).

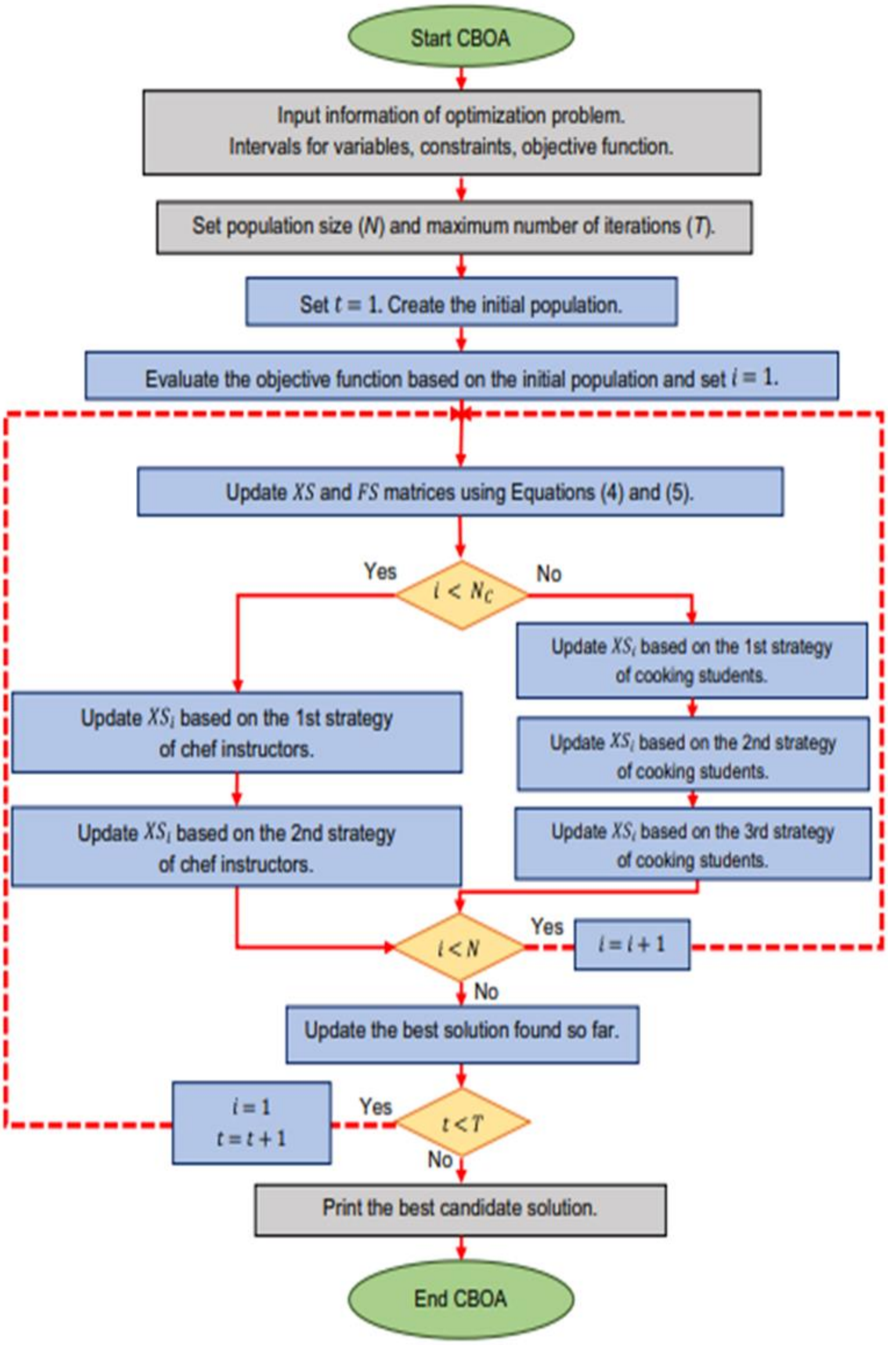

Figure 3.5: A diagram of the implementation stages of the CBOA algorithm

6. CONCLUSION

In this chapter, we presented the single-objective CBOA optimization algorithm based on chef training and explained the stages and steps of the algorithm and how it enables us to find solutions in single-objective optimization problems.

CHAPTER 04:

Design and production.

1. INTRODUCTION

In this chapter, we propose a new Human inspired Multi Objective algorithm named, Multi Objective Optimization Chef-based Optimization Algorithm (MOCBOA), we then validate this algorithm and we compare its performances with that of (multi-objective particle swarm optimization) MOPSO [25], NSGA2[4] and (knee point-driven evolutionary algorithm) KNEA [27] and NSGA3[5].

2. DESCRIPTION OF THE MOCBOA ALGORITHM

One recently proposed single-objective algorithm is the CBOA algorithm proposed by EvaTrojovská Muhammad Dehqani in 2022, this algorithm is inspired by humans in particular chefs and stages culinary training process iteration to develop quality optimization algorithm. As with all single-objective algorithms, the CBOA algorithm cannot solve complex optimization problems with multiple conflicting objectives. In this case we propose a new multi-objective version of the CBOA algorithm. The main idea is to use external CBOA to safeguard the solutions and update them. And help guide the population towards the best solution around the search space. The main motivation of this work is to make the CBOA algorithm multi-objective and capable of finding better quality solutions to engineering problems.

When developing an algorithm, a variety of elements are used, each with a specific purpose to guarantee the algorithm's efficiency and effectiveness. Following the key point of the MOCBOA algorithm:

- ✓ Initialization of the first population (search agent) randomly.
- ✓ Evaluate the population according to the objective function.
- ✓ The Pareto dominance method has been integrated into the CBOA algorithm to extract non-dominated solutions.
- ✓ external archive was added to the CBOA algorithm in order to save non-dominated solutions. The archive was Initialized with the non-dominated solution obtained from applying dominance relation.
- ✓ The update of the population (chefs and students according to the equations introduced in COBA) and using the dominance concepts while comparing the solutions.
- ✓ Update the archive solutions by applying dominance relation and using a combination of the old archive solutions and the current population solutions in order to maintain the diversity and convergence of the Pareto optimal solutions.

Not different from the CBOA algorithm, the MOCBOA algorithm begins with a random

population, where each individual is considered a candidate solution as well as a search agent. The problem variables are their positions in the search space, and an individual can move in more than one dimension. The population is divided according to COBA into Chefs and Students in order to start the optimization process. The population is evaluated using the objective function first.

To save non-dominated solutions, the CBOA algorithm now includes an external archive. The addition of an external archive is a big improvement because it acts as a repository for storing the finest solutions discovered during the optimization process.

This archive guarantees once a non-dominated solution is identified, it is retained, even if it is not currently present in the population. This mechanism is critical for preserving a high-quality collection of solutions throughout the optimization process, limiting the loss of valuable solutions and accelerating convergence towards the true Pareto front. Here the archive is initialized using the non-dominated solution obtained after applying the dominance relation.

The Pareto dominance method or also known as dominance relation has been incorporated into the CBOA algorithm to identify non-dominated solutions in the population. This integration is critical because the Pareto dominance method allows us to assess solutions based on many objectives without grouping them together into a single objective function. By focusing on non-dominated solutions, the algorithm assures that the solutions are optimal in a way that no other solutions in the population outperform them across all objectives. This method is crucial in multi-objective optimization because it assists in the identification of a collection of trade-off solutions, providing decision-makers with a variety of optimal solutions to evaluate.

Updating archive solutions is a critical procedure that involves both previous archive solutions and current population solutions, as determined by the dominance relation. This essential phase ensures that only the best solutions are used, which significantly improves the archive's overall quality and performance. As the optimization process is iterative, the following procedures are repeated: computing the objective function for the newly updated population, including the Pareto dominance approach to generate non-dominated solutions, and updating the archive solutions. This operation continues until the termination condition, or the maximum number of iterations, is met. (Figure 4.6)

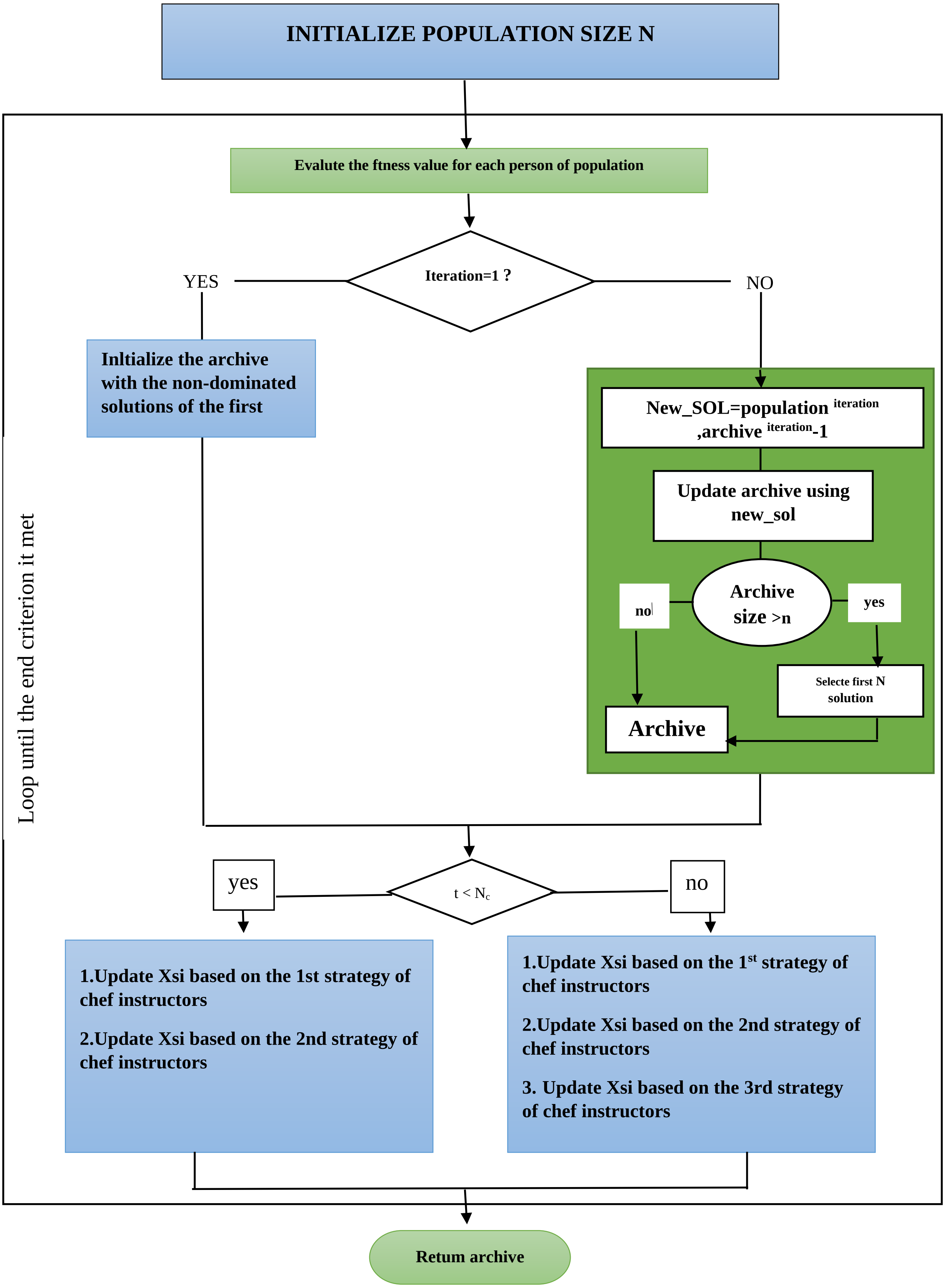
INITIALIZE POPULATION SIZE N
Evalute the ftness value for each person of population
Iteration=1 ?
YES
NO
Inltialize the archive with the non-dominated solutions of the first
New_SOL=population iteration ,archive iteration-1
Update archive using new_sol
Archive size >n
no
yes
Selecte first N solution
Archive
Loop until the end criterion it met
t < Nc
yes
no
1.Update Xsi based on the 1st strategy of chef instructors
2.Update Xsi based on the 2nd strategy of chef instructors
1.Update Xsi based on the 1st strategy of chef instructors
2.Update Xsi based on the 2nd strategy of chef instructors
3. Update Xsi based on the 3rd strategy of chef instructors
Retum archive

Figure 4.6: MOCBOA Flowchart

3. TEST PROBLEMS

Mathematical Description of RWCMOPs problems [29]

Mechanical Design Problems

1. Pressure vessel design [30] (RWMOP01)

This constrained optimization problem contains discrete, integer, and continuous variables. This problem's main objective is to obtain the shape of a helical compression spring having the least volume.

Minimize:

$$f_1 = 1.7781 z_2 x_3^2 + 0.6224 z_1 x_3 x_4 + 3.1661 z_1^2 x_4 + 19.84 z_1^2 x_3$$

$$f_2 = -\pi x_3^2 x_4 - \frac{4}{3}\pi x_3^3$$

Subject to:

$$g_1(\bar{x}) = 0.00954 x_3 \leq z_2,$$

$$g_2(\bar{x}) = 0.0193 x_3 \leq z_1,$$

Where:

$$z_1 = 0.0625 x_1,$$

$$z_2 = 0.0625 x_2.$$

With bounds:

$$10 \leq x_4, x_3 \leq 200$$

$$1 \leq x_2, x_1 \leq 99 \text{ (integer variables).}$$

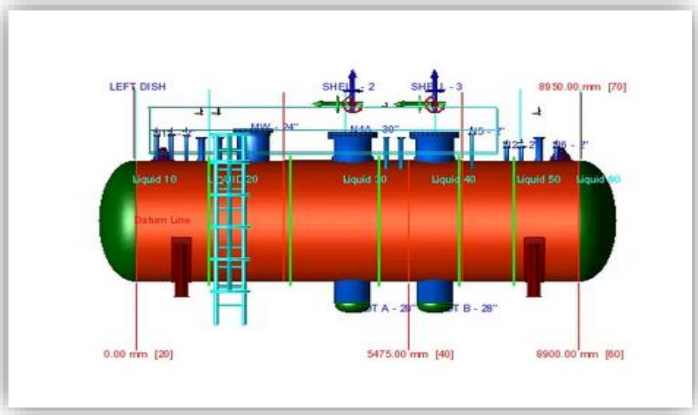

Figure 4.7: Pressure vessel design

2. Vibrating platform design [31] (RWMOP02)

Originally vibrating platform problem is formulated as a single-objective constrained problem with the maximization of the fundamental frequency. This problem is modified to include the cost of operations as the second objective.

Minimize :

$$f_1 = -\frac{\pi}{2L^2}\sqrt{\frac{EI}{\mu}}$$

$$f_2 = 2bL\,(c_1d_1 + c_2\,(d_2 - d_1) + c_3\,(d_3 - d_2))$$

Subject to:

$$g_1 = \mu L - 2800 \leq 0,$$
$$g_2 = d_1 - d_2 \leq 0,$$
$$g_3 = d_2 - d_1 - 0.15 \leq 0,$$
$$g_4 = d_2 - d_3 \leq 0,$$
$$g_5 = d_3 - d_2 - 0.01 \leq 0$$

Where:

$$EI = \frac{2b}{3}(E_1d^3{}_1 + E_2(d^3{}_2 - d^3{}_1) + E_3(d^3{}_3 - d^3{}_2))$$
$$\mu = 2b\,(\rho_1d_1 + \rho_2\,(d_2 - d_1) + \rho_3\,(d_3 - d_2))$$
$$\rho_1 = 100,\ \rho_2 = 2770,\ \rho_3 = 7780,$$
$$E_1 = 1.6,\ E_2 = 70,\ E_3 = 200,$$
$$c_1 = 500,\ c_2 = 1500,\ c_3 = 800$$

With bounds:

$$0.05 \leq d_1 \leq 0.5$$
$$0.2 \leq d_2 \leq 0.5$$
$$0.2 \leq d_3 \leq 0.6$$
$$0.35 \leq b \leq 0.5$$
$$3 \leq L \leq 6$$

Figure 4.8: Vibrating platform design

3. Two bar truss design [32] (RWMOP03)

This problem involves the design of a two-bar truss. Originally, this problem is developed as a single-objective Problem. The problem has been transformed into a bi-objective problem.

Minimize :

$$f_1(x) = x_1\sqrt{16 + x^2{}_3} + x_2\sqrt{1 + x^2{}_3}$$
$$f_2(x) = \frac{20\sqrt{16 + x^2{}_3}}{x_3x_1}$$

Subject to:

$g_1(x) = f_1(x) - 0.1 \leq 0,$

$g_2(x) = f_2(x) - 10^5 \leq 0,$

$g_3(x) = \frac{80\sqrt{1+X^2{}_3}}{X_3X_2} - 10^5 \leq 0$

With bounds:

$10^{-5} \leq x_1 \leq 100,$

$10^{-5} \leq x_2 \leq 100,$

$1 \leq x_3 \leq 3$

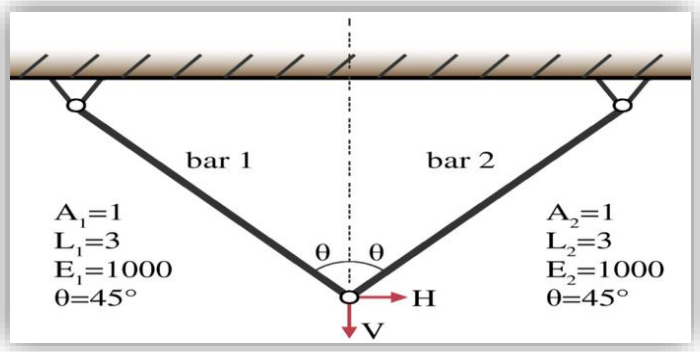

Figure 4.9: Two bar truss design

4. Welded beam design [33] (RWMOP04)

This problem is already well-studied as a single-objective optimization problem, where four design variables need to be optimized for which the beam's cost is minimum. However, the objective functions regarding minimum cost and maximum rigidity are conflicting with each other. Therefore, this problem is redefined as a bi-objective problem.

Minimize :

$f_1(x) = 1.10471x^2{}_1\, x_2 + 0.04811x_3\, x_4(14 + x_2),$

$f_2(x) = \frac{4PL^3}{Ex_4x^3{}_3}$

Subject to:

$g_1(x) = \tau(x) - \tau_{max} \leq 0,$

$g_2(x) = \sigma(x) - \sigma_{max} \leq 0,$

$g_3(x) = x_1 - x_4 \leq 0,$

$g_4(x) = P - P_c(x) \leq 0,$

Where:

$\tau(x) = \sqrt{(\tau')^2 + \frac{2\tau'\tau\prime' x_2}{2R} + (\tau'')^2}$

$\tau' = \frac{P}{\sqrt{2}X_1X_2}$

$M = P(L + \frac{x_2}{2})$

$R=\sqrt{\frac{x^2{}_2}{4}+\left(\frac{x_1+x_3}{2}\right)^2}$

$J=2\left(\sqrt{2}x_1x_2\left(\frac{x^2{}_2}{12}+\left(\frac{x_1+x_3}{2}\right)^2\right)\right)$

$\sigma\ (x)=\frac{6PL}{x_4x^2{}_3}$

$P_c(x)=\frac{4.013E\sqrt{\frac{x^2{}_3x^6{}_4}{36}}}{L^2}\left(1-\frac{x_3}{2L}\sqrt{\frac{E}{4G}}\right)$

$P = 6000$,

$L = 14$,

$E = 30 \times 10^6$

$\tau_{max} = 13600$.

$\sigma_{max} = 30,000$.

With bounds:

$0.125 \leq x_1 \leq 5$,

$0.1 \leq x_2 \leq 10$,

$0.1 \leq x_3 \leq 10$,

$0.125 \leq x_4 \leq 5$.

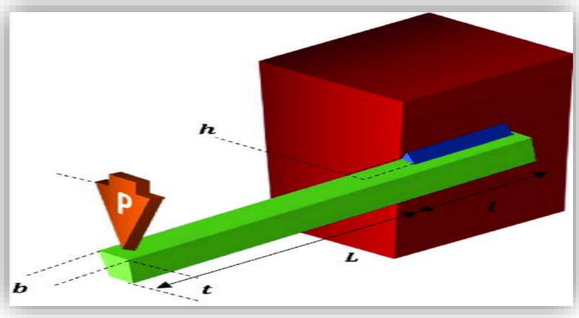

Figure 4.10: Welded beam design

5. Disc brake design [34] (RWMOP05)

This design problem aims to reduce the brake weight and minimize the stopping time. The variables are the radius of internal and external disks, the force of engagement, and the number of friction surfaces. The design constraints involve the maximum break length, friction, temperature, and torque limits.

Minimize:

$f_1(x) = 4.9 \times 10^{-5}\,(x^2{}_2 - x^2{}_1)\,(x_4-1)$

$f_2(x) = 9.82 \times 10^6\left(\frac{x^2{}_2-x^2{}_1}{x_3x_4(x^3{}_2-x^3{}_1)}\right)$

Subject to:

$g_1(x) = 20 - (x_2 - x_1) \leq 0$,

$g_2(x) = \frac{x_3}{3.14(x^2{}_2-x^2{}_1)} - 0.4 \leq 0$

$g_3(x) = \frac{2.22*10-3\ x_3(x^3{}_2-x^3{}_1)}{(x^2{}_2-x^2{}_1)^2} - 1 \leq 0$,

$$g_4(x) = 900 - 2.66 \times 10^{-2} \frac{x_3x_4(x^3{}_2 - x^3{}_1)}{(x^2{}_2 - x^2{}_1)} \le 0$$

With bounds:

$55 \le x_1 \le 80$

$75 \le x_2 \le 110$

$1000 \le x_3 \le 3000$

$11 \le x_4 \le 20$.

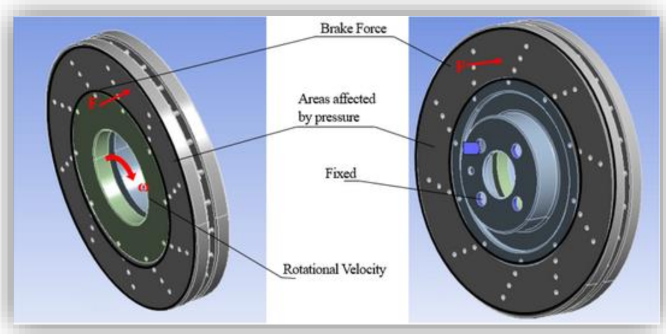

Figure 4.11: Disc brake design

6. Speed reducer design [35] (RWMOP06)

This problem is a bi-objective constrained optimization problem. Here, one of the constraints of the original problem is assumed an extra objective of the problem.

Minimize:

$$f_1(x) = 0.7854x_1 x^2{}_2 \left(\frac{10x^2{}_3}{3} + 14.933x_3 - 43.0934\right) - 1.508x_1(x^2{}_6 + x^2{}_7) + 7.477(x^3{}_6 + x^3{}_7) + 0.7854(x_4 x^2{}_6 + x_5 x^2{}_7)$$

$$f_2(x) = \frac{\sqrt{\left(\frac{745x_4}{x_2x_3}\right)^2 + 1.69*10(7)}}{0.1x^3{}_6}$$

Subject to:

$g_1(x) = \frac{1}{x_1x^2{}_2x_3} - \frac{1}{27} \le 0,$

$g_2(x) = \frac{1}{x_1x^2{}_2x^2{}_3} - \frac{1}{397.5} \le 0,$

$g_3(x) = \frac{x^3{}_4}{x_2x_3x_{46}} - \frac{1}{1.93} \le 0,$

$g_4(x) = \frac{x^3{}_5}{x_2x_3x_{47}} - \frac{1}{1.93} \le 0,$

$g_5(x) = x_2 x_3 - 40 \le 0,$

$g_6(x) = \frac{x_1}{x_2} - 12 \le 0,$

$g_7(x) = -\frac{x_1}{x_2} + 5 \le 0,$

$g_8(x) = 1.9 - x_4 + 1.5x_6 \le 0,$

$g_9(x) = 1.9 - x_5 + 1.1x_7 \le 0,$

$g_{10}(x) = f_2(x) - 1300 \le 0,$

$g_{11}(x) = \frac{\sqrt{(\frac{745x_5}{x_2x_3})^2+1.575*10°8}}{0.1x^3{}_7} - 110 \leq 0.$

With bounds:

$2.6 \leq x_1 \leq 3.6$

$0.7 \leq x_2 \leq 0.8$

$x_3 \in \{17, ..., 28\}$ (integer)

$7.3 \leq x_4 \leq 8.3$

$7.3 \leq x_5 \leq 8.3$

$2.9 \leq x_6 \leq 3.9$

$5 \leq x_7 \leq 5.5$

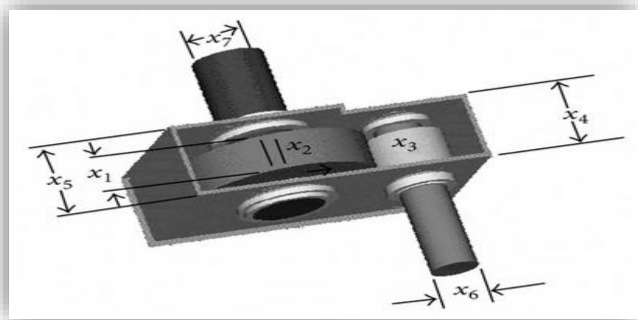

Figure 4.12: Speed reducer design

7. Gear train design [36] (RWMOP07)

This design problem involves minimization of gears' ratio and size (inner and outer radius of gears). The ratio of gear trains can be defined as the ratio of input and output shafts' angular velocities.

Minimize :

$f_1(x) = \left|6.931 - \frac{x_3x_4}{x_1x_2}\right|$

$f_2(x) = \max \{x_1, x_2, x_3, x_4\}$

Subject to:

$g_1(x) = \frac{f_1(x)}{6.931} - 0.5 \leq 0$

With bounds:

$x_1.x_2, x_3, x_4 \in \{12, ..., 60\}$ (interger)

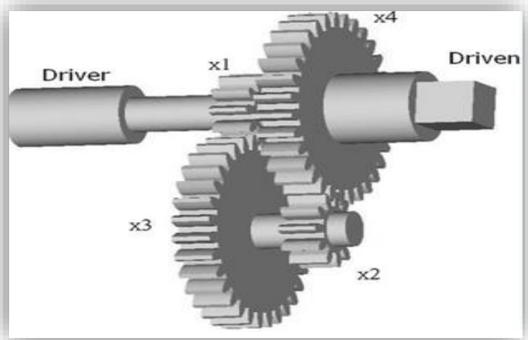

Figure 4.13: Gear train design

8. Car side impact design [37] (RWMOP08)

This design problem is a three-objective constrained optimization problem, where all objectives are minimization type. This problem contains seven variables and 10 inequality constraints.

Minimize:

$f_1(x) = 1.98 + 4.9x_1 6.67x_2 + 6.98x_3 + 4.01x_4 + 1.78x_5 + 10^{-5} x_6 + 2.73x_7,$
$f_2(x) = 4.72 - 0.5x_4 - 0.19x_2 x_3,$
$f_3(x) = 0.5 (V_{MBP}(x) + V_{FD}(x))$

Subject to:

$g_1(x) = -1 + 1.16 - 0.3717x_2 x_4 - 0.0092928x_3 \leq 0,$
$g_2(x) = -0.32 + 0.261 - 0.0159x_1 x_2 - 0.06486x_1 - 0.019x_2 x_7 + 0, 0144x_3 x_5 + 0.0154464x_6 \leq 0,$
$g_3(x) = -0.32 + 0.74 - 0.61x_2 - 0.031296x_3 - 0.031872x_7 + 0.227x_2^2 \leq 0,$
$g_4(x) = -0.32 + 0.214 + 0.00817x_5 - 0.045195x_1 - 0.0135168x_1 + 0.03099x_2 x_6 - 0.018x_2 x_7 + 0.007176x_3 + 0.023232x_3 - 0.00364x_5 x_6 - 0.018x_2^2 \leq 0,$
$g_5(x) = -32 + 33.86 + 2.95x_3 - 5.057x_1 x_2 - 3.795x_2 - 3.4431x_7 + 1.45728 \leq 0,$
$g_6(x) = -32 + 28.98 + 3.818x_3 - 4.2x_1 x_2 + 1.27296x_6 - 2.68065x_7 \leq 0,$
$g_7(x) = -32 + 46.36 - 9.9x_2 - 4.4505x_1 \leq 0,$
$g_8(x) = f_1(x) - 4 \leq 0,$
$g_9(x) = V_{MBP} - 9.9 \leq 0,$
$g_{10}(x) = V_{FD}(x) - 15.7 \leq 0$

Where:

$V_{MBP}(x) = 10.58 - 0.674x_1 x_2 - 0.67275x_2$
$V_{FD}(x) = 16.45 - 0.489x_3 x_7 - 0.843x_5 x_6$

With bounds:

$0.5 \leq x_1 \leq 1.5$
$0.45 \leq x_2 \leq 1.35$
$0.5 \leq x_3 \leq 1.5$
$0.5 \leq x_4 \leq 1.5$

$0.875 \leq x_5 \leq 2.625$
$0.4 \leq x_6 \leq 1.2$
$0.4 \leq x_7 \leq 1.2$

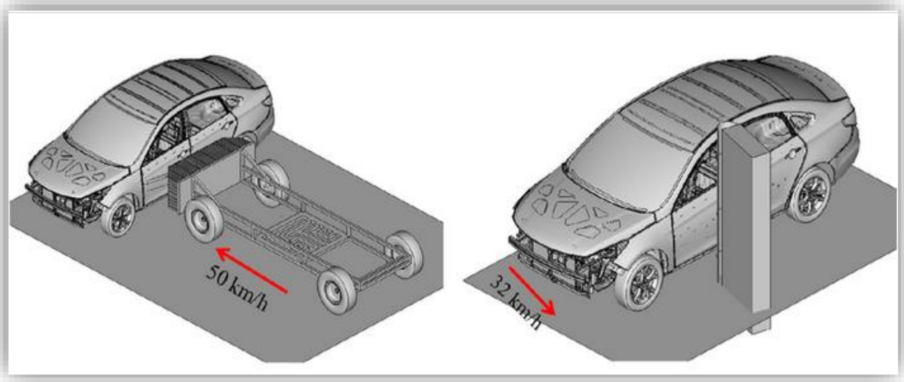

Figure 4.14: Car side impact design

9. Four bar plane truss [38] (RWMOP09)

Four Bar Plane Truss is a bi-objective bound-constrained optimization problem. This problem has four variables.

Minimize:

$$f_1(x) = L\,(2x_1 + \sqrt{(2)}x_2 + \sqrt{(2)}x_3 + x_4)$$
$$f_2(x) = \frac{FL}{E}\left(\frac{2}{x_1} + \frac{2\sqrt{(2)}}{x_2} - \frac{2\sqrt{(2)}}{x_3} + \frac{2}{x_4}\right)$$

With bounds:

$$\frac{F}{\sigma} \leq x_1 \leq 3\frac{F}{\sigma}$$
$$\sqrt{2}\,\frac{F}{\sigma} \leq x_2 \leq 3\frac{F}{\sigma}$$
$$\sqrt{2}\,\frac{F}{\sigma} \leq x_3 \leq 3\frac{F}{\sigma}$$
$$\frac{F}{\sigma} \leq x_4 \leq 3\frac{F}{\sigma}$$

Where:

$F = 10$kN, $E = 2 \times 10^5$ kN/cm², $L = 200$cm, $\sigma = 10$kN/cm²

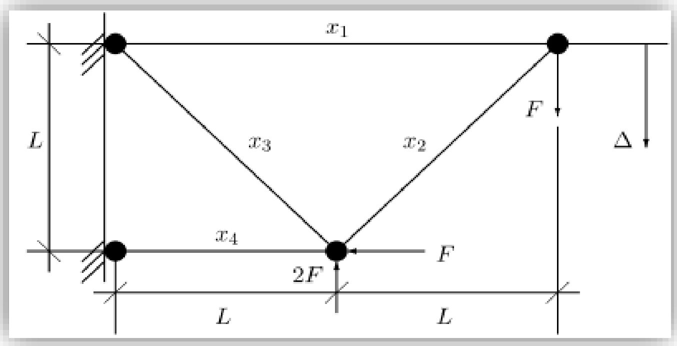

Figure 4.15: Four bar plane truss

10. Two bar plane truss [39] (RWMOP10)

This problem is a bi-objective constrained optimization problem, where all objectives are minimization type. This problem involves two variables and two inequality constraints.

Minimize:

$$f_1(x) = 2\rho h x_2 \sqrt{1} + {x^2}_1$$

$$f_2(x) = \frac{\rho h (1+{X^2}_1)1.5(1+x^1)0.5}{2\sqrt{2E{x^2}_1 x_2}}$$

Subject to:

$$g_1 = \frac{p(1+x_1)(1+{x^2}_1)}{2\sqrt{2x_1x_2}} - \sigma_0 \leq 0,$$

$$g_2 = \frac{p(-x_1+1)(1+{x^2}_1)0.5}{2\sqrt{2x_1}x_2} - \sigma_0 \leq 0$$

With bounds:

$$0.1 \leq x_1 \leq 2,$$

$$0.5 \leq x_2 \leq 2.5$$

Where:

ρ = 0.283lb/in³, h = 100in, P = 104lb, E = 3 × 10^7 lb/in²

σ_0 = 2 × 10 lb/in², A $_{MIN}$= 1in²

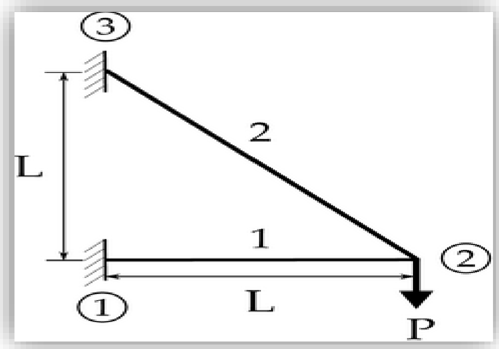

Figure 4.16: Two bar plane truss

4. PERFORMANCE MEASURE

For stochastic optimization methods, no guarantee of quality of results can be made. generally, be given a priori. Indeed, the very notion of experimental performance of these approaches must take into account their stochastic nature, which requires relying on an analysis statistic of independent series of experiments. Many quantitative measures have been proposed to assess the quality of all the optimal compromises. we distinguish between metrics which require knowledge of the exact Pareto surface, called exact, and those which do not require this information, say blind such as hypervolume indicator.

4.1. HYPERVOLUME INDICATOR [40]

The hypervolume indicator (or S-metric, from "size of space covered "for some (approximation) set $A \subset \mathbb{R}^m$ and a reference point $r \in R^m$ that is dominated by all the points in A is defined as:

$$HI(A) = Vol\{b \in \mathbb{R}^m \mid b \leq r \wedge \exists a \in A: a \leq b\} = Vol(\cup_{a \in A} [a, r])$$

Here Vol (.)Denotes the lebesgue measure of a m-dimensional set of points and [a, r] denotes the interval box with lower corner a and upper corner r.

In 2-D this is simply the covered area, and in 3-D the covered volume (see figure 17 for examples). For reviewing some important properties related to this indicator, lets us introduce a comparison operator between sets.

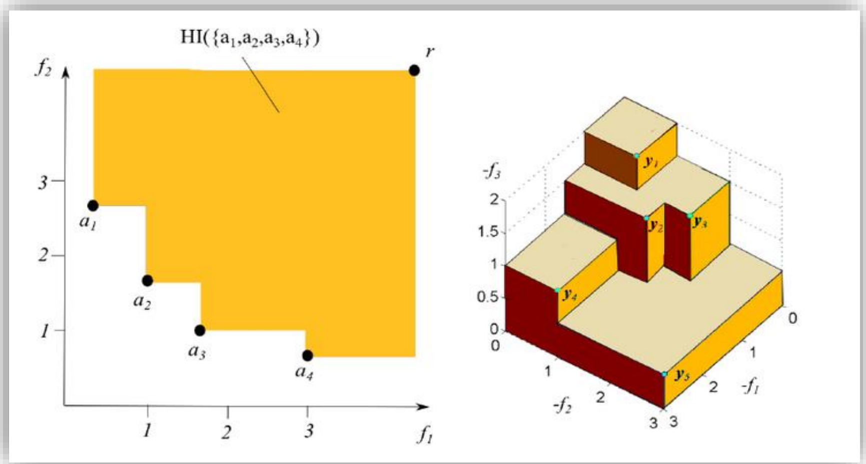

Figure4.17: hypervolume indicator in two dimensions for a set $A=\{a_1,\ldots;a_4\}\subset\mathbb{R}^2$(left) and in three dimensions for a set $Y=\{y_1,\ldots,y_5\}\subset\mathbb{R}^3$(right).

5. STATISTICAL COMPARISON RESULTS

Table 4.2: Comparison condition

Algorithms	Population size	Number of evaluations	Number of executions
MOCBOA	100	1000	30
MOPSO	100	1000	30
NSGA2	100	1000	30

NSGA3	100	1000	30
KNEA	100	1000	30

The MOCBOA algorithm that we proposed is programmed under MATLAB R2021b. This easy tool has been specially designed for different matrix calculations, and run on a PC under WINDOWS 10, core i5, Intel 1.7 GHZ with 4096 MB RAM.

We present in the following table the values of: Best, Worst and Average evaluations taken on 30 independent tests on each of RWMOP01, RWMOP02, RWMOP03, RWMOP04, RWMOP05, RWMOP06, RWMOP07, RWMOP08, RWMOP09, RWMOP10.

Table 4.3: Statistical results for the 10 problems

ALGORITHM PROBLEM	MOCBOA	MOPSO	NSGA2	KNEA	NSGA3
RWMOP01					
BEST	**0.968**	NaN	0.605	0.671	0.606
WORST	**0.968**	NaN	0.603	0.583	0.602
AVERAGE	**0.968**	NaN	0.604	0.593	0.604
RWMOP02					
BEST	**0.827**	NaN	0.392	0.393	0.392
WORST	**0.827**	NaN	0	0	0
AVERAGE	**0.827**	NaN	0.232	0.278	0.257
RWMOP03					
BEST	0	**1**	0.902	0.897	0.897
WORST	0	**0.224**	0.901	0.835	0.892
AVERAGE	0	**NaN**	0.901	0.877	0.894
RWMOP04					
BEST	0.760	**0.999**	0.862	0.850	0.861
WORST	0.760	**0.196**	0.829	0.723	0.797
AVERAGE	0.760	**NaN**	0.855	0.794	0.849
RWMOP05					
BEST	**0.589**	NaN	0.434	0.417	0.434
WORST	**0.589**	NaN	0.430	0.382	0.429

AVERAGE	**0.589**	NaN	0.433	0.402	0.432
RWMOP06					
BEST	**0.284**	NaN	0.277	0.274	0.276
WORST	**0.284**	NaN	0.276	0.213	0.275
AVERAGE	**0.284**	NaN	0.276	0.256	0.276
RWMOP07					
BEST	0.370	**0.999**	0.484	0.483	0.483
WORST	0.370	**0.880**	0.482	0.463	0.479
AVERAGE	0.370	**0.950**	0.483	0.480	0.482
RWMOP08					
BEST	0.018	**0.061**	0.026	0.025	0.025
WORST	0.018	**0**	0.025	0.023	0.024
AVERAGE	0.018	**NaN**	0.025	0.025	0.025
RWMOP09					
BEST	0.336	**0.933**	0.409	0.384	0.409
WORST	0.336	**0.176**	0.408	0.360	0.407
AVERAGE	0.336	**0.609**	0.409	0.373	0.408
RWMOP10					
BEST	0.818	**1**	0.847	0.844	0.839
WORST	0.818	**0.305**	0.846	0.806	0.833
AVERAGE	0.818	**NaN**	0.847	0.823	0.834

6. ANALYSIS AND DISCUSSION

6.1. RWMOP01

The results obtained prove that the performance of MOCBOA was better than other algorithms, as its value was close to 1. As for the algorithms NSGA2 and NSGA3, KNEA, it did not exceed 0.6, in addition to the MOPSO algorithm, which did not find a solution.

6.2. RWMOP02

The results obtained prove that the performance of MOCBOA was better than other algorithms, as its value was 0.8, while for the algorithms NSGA2, NSGA3, and KNEA, it did not exceed 0.3, in addition to the MOPSO algorithm, which did not find a solution.

6.3. RWMOP03

The results obtained prove that the performance of MOPSO was better than other algorithms, as its value was 1. As for the algorithms NSGA2, NSGA3, and KNEA, it did not exceed 0.9, but for the MOCBOA algorithm the result was 0.

6.4. RWMOP04

Convergent results were obtained for the NSGA2, NSGA3, and KNEA algorithms, ranging between 0.8, and the result for both the MOPOS algorithm was 0.999 and the MOCBOA algorithm was 0.760, which indicates that the best performance was for the MOPSO algorithm.

6.5. RWMOP05

The results obtained prove that the performance of MOCBOA was better than other algorithms, as its value was 0.589. As for the algorithms NSGA2, NSGA3, and KNEA, it did not exceed 0.45, in addition to the MOPSO algorithm, which did not find a solution.

6.6. RWMOP06

The results obtained proved that the performance of MOCBOA was better than other algorithms as its maximum value reached 0.284. As for the NSGA2, NSGA3, and KNEA algorithms, they did not exceed 0.277, but their values are very close, and in the end, we conclude that the MOPSO algorithm was unable to solve the problem.

6.7. RWMOP07

The results obtained prove that the performance of MOPSO was better than other algorithms, as its maximum value was 0.999. As for the NSGA2, NSGA3, and KNEA algorithms, it did not exceed 0.484, and as for the MOCBOA algorithm, its performance was greatest 0.370.

6.8. RWMOP08

The results obtained prove that the performance of MOPSO was better than other algorithms, as its maximum value was 0.061. As for the NSGA2, NSGA3, and KNEA algorithms, it did not exceed 0.026, and as for the MOCBOA algorithm, its performance was greatest 0.018, This means that their performance is close.

6.9. RWMOP09

The results obtained prove that the performance of MOPSO was better than other algorithms, as its maximum value was 0.933. As for the NSGA2, NSGA3, and KNEA algorithms, it did not exceed 0.409, and as for the MOCBOA algorithm, its performance was greatest 0.336.

6.10. RWMOP10

The results obtained prove that the performance of MOPSO was better than other algorithms, as its maximum value was 1. As for the algorithms NSGA2 and NSGA3, KNEA, MOCBOA, their performance was close, their maximum value was 0.847, 0.839, 0.844, 0.818.

To summarize, the MOCBOA algorithm performed well, especially on the RWMOP1, RWMOP2, RWMOP5, and RWMOP6 test problems. In comparison, the MOPSO algorithm proved unstable and frequently failed to solve problems. The other algorithms, namely NSGA2, NSGA3, and KNEA, showed intermediate performance, successfully identifying solutions and occasionally approaching MOCBOA's performance, but not consistently outperforming it.

7. CONCLUSION

In this chapter, we present the MOCBOA algorithm as a novel method for tackling optimization problems. This algorithm is an extension of the mono-objective CBOA, which was inspired by the culinary learning process, in which trainees learn from chefs how to improve and grow their abilities so that they can become expert chefs. We presented an approach for addressing RWMOP issues with the MOCBOA algorithm and compared its performance to well-known multi-objective optimization algorithms. Overall, our suggested MOCBOA algorithm has shown significant potential as an effective optimization tool.

GENERAL CONCLUSION

The work performed within the framework of this thesis falls within the field of multi-objective optimization. First, we discuss the basic concepts of combinatorial optimization, as well as the two types of optimization citing single-objective optimization and multi-objective optimization. Then we presented methods for solving combinatorial optimization problems after highlighting human-inspired multi-objective algorithms and then adding some engineering applications of optimization. All of the above information helped us expand our knowledge about the field of multi-objective optimization and direct our approach towards satisfactory solutions.

The approach proposed in this thesis is a multi-objective extension of a new algorithm called CBOA and the recently developed multi-objective chef-based optimization algorithm (MOCBOA). In this proposal we present the following elements:

We added the Pareto dominance method to the algorithm to extract solutions.

An archive has been added to save the solutions and update them in every process we carry out.

We validated our MOCBOA algorithm by comparing it with MOPSO, NSGA2, KnEA and NSGA3 algorithms using test problems RWMOP 01, RWMOP 02, RWMOP 03, RWMOP 04, RMWOP 05, RMWOP 06, RMWOP 07, RMWOP 08, RMWOP 09, RMWOP10. The obtained results show that the MOCBOA algorithm is efficient in terms of convergence towards Pareto integer, and at the same time the MOCBOA algorithm provides more diverse undominated solutions . Finally, we concluded that the MOCBOA algorithm is capable of solving problems and yielding results. This suggests that further development and the integration of new methods, such as [ε-dominance….], can enhance its ability to tackle larger real-world problems and deliver improved performance

REFERENCE

[1] Wolpert DH, Macready WG (1997) No free lunch theorems for optimization. IEEE Trans Evol Comput 1:67–82.

[2] Trojovská, Eva, and Mohammad Dehghani. "A new human-based metahurestic optimization method based on mimicking cooking training." Scientific Reports 12.1 (2022): 14861.

[3] Singh, Jagseer. "A review and comparison of two archive-based Algorithms: SPEA2 and PAES." AIP Conference Proceedings. Vol. 2819. No. 1. AIP Publishing, 2023.

[4] Deb, Kalyanmoy, et al. "A fast and elitist multiobjective genetic algorithm: NSGA-II." IEEE transactions on evolutionary computation 6.2 (2002): 182-197.

[5] Han, Lu, Xianjun Shi, and Yuyao Zhai. "Test optimization selection method based on NSGA-3 and improved Bayesian network model." Xibei Gongye Daxue Xuebao/Journal of Northwestern Polytechnical University 39.2 (2021): 414-422.

[6] Krag, David, et al. "The sentinel node in breast cancer a multicenter validation study." New England Journal of Medicine 339.14 (1998): 941-946.

[7] Reynolds RG (1994) An introduction to cultural algorithms. Proc Third Ann Conf Evol Program 24:131–139

[8] Geem ZW, Kim JH, Loganathan GV (2001) a new heuristic optimization algorithm: harmony search. Simulation 76(2):60–68

[9] Ray T, Liew KM (2003) Society and civilization: an optimization algorithm based on the simulation of social behavior. IEEE Trans Evol Comput 7(4):386–396

[10] Dai C, Zhu Y, Chen W (2006, November) Seeker optimization algorithm. In International conference on computational and information science (pp. 167–176). Springer, Berlin, Heidelberg.

[11] Atashpaz-Gargari E, Lucas C (2007, September). Imperialist competitive algorithm: an algorithm for optimization inspired by imperialistic competition. In 2007 IEEE congress on evolutionary computation (pp. 4661–4667). IEEE.

[12] Kashan AH (2009, December). League championship algorithm: a new algorithm for numerical function optimization. In 2009 international conference of soft computing and pattern recognition (pp. 43–48). IEEE.

[13] Eita MA, Fahmy MM (2010) Group counseling optimization: a novel approach. In Research and development in intelligent systems XXVI (pp. 195–208). Springer: London.

[14] Ahmadi-Javid a (2011, June) Anarchic society optimization: a human-inspired method. In 2011 IEEE congress of evolutionary computation (CEC) (pp. 2586–2592). IEEE.

[15] Kulkarni AJ, Durugkar IP, Kumar M (2013, October) Cohort intelligence: a self-supervised learning behavior. In 2013 IEEE international conference on systems, man, and cybernetics (pp. 1396–1400). IEEE.

[16] Ghorbani N, Babaei E (2014) Exchange market algorithm. Appl Soft Comput 19:177–187

[17] Emami H, Derakhshan F (2015) Election algorithm: A new socio-politically inspired strategy. AI Commun 28(3):591–603

[18] Savsani P, Savsani V (2016) Passing vehicle search (PVS): A novel metaheuristic algorithm. Appl Math Model 40(5–6):3951–3978

[19] Mousavirad SJ, Ebrahimpour-Komleh H (2017) Human men-tal search: a new population-based metaheuristic optimization algorithm. Appl Intell 47(3):850–887

[20] Fathollahi-Fard AM, Hajiaghaei-Keshteli M, Tavakkoli-Moghaddam R (2018) The social engineering optimizer (SEO). Eng Appl Artif Intell 72:267–293

[21] Elsisi M (2019) Future search algorithm for optimization. Evol Intel 12(1):21–31

[22] Askari Q, Younas I, Saeed M (2020) Political optimizer: a novel socio-inspired meta-heuristic for global optimization. Knowl-Based Syst 195:105709

[23] Al-Betar MA, Alyasseri ZAA, Awadallah MA, Doush IA (2021) Coronavirus herd immunity optimizer (CHIO). Neural Comput Appl 33(10):5011–5042

[24] Eita MA, Fahmy MM (2010) Group counseling optimization: a novel approach. In Research and development in intelligent systems XXVI (pp. 195–208). Springer: London.

[25] Lalwani, Soniya, et al. "A comprehensive survey: Applications of multi-objective particle swarm optimization (MOPSO) algorithm." Transactions on combinatorics 2.1 (2013): 39-101.

[27] Dong, Jianhua, et al. "Comparison of four bio-inspired algorithms to optimize KNEA for predicting monthly reference evapotranspiration in different climate zones of China." Computers and Electronics in Agriculture 186 (2021): 106211.

[29] Reynolds RG (1994) An introduction to cultural algorithms. Proc Third Ann Conf Evol Program 24:131–139

[30] B. Kannan, S. N. Kramer, an augmented lagrange multiplier-based method for mixed integer discrete continuous optimization and its applications to mechanical design (1994).

[31] S. Narayanan, S. Azarm, On improving multiobjective genetic algorithms for design optimization, Structural Optimization 18 (1999) 146–155.

[32] G. Chiandussi, M. Codegone, S. Ferrero, F. E. Varesio, Comparison of multi-objective optimization methodologies for engineering applications, Computers & Mathematics with Applications 63 (2012) 912–942.

[33] K. Deb, et al., Evolutionary algorithms for multi-criterion optimization in engineering design, Evolutionary algorithms in engineering and computer science 2 (1999) 135–161.

[34] A. Osyczka, S. Kundu, A genetic algorithm-based multicriteria optimization method, Proc. 1st World Congr. Struct. Multidisc. Optim (1995) 909–914.

[35] S. Azarm, A. Tits, M. Fan, Tradeoff-driven optimization-based design of mechanical systems, in: 4th Symposium on Multidisciplinary Analysis and Optimization, 1999, p. 4758.

[36] T. Ray, K. Liew, A swarm metaphor for multiobjective design optimization, Engineering optimization 34 (2002) 141–153.

[37] H. Jain, K. Deb, an evolutionary many-objective optimization algorithm using reference-point based nondominated sorting approach, partii: handling constraints and extending to an adaptive approach, IEEE Transactions on evolutionary computation 18 (2013) 602–622.

[38] F. Cheng, X. Li, Generalized center method for multiobjective engineering optimization, Engineering Optimization 31 (1999) 641–661.

[39] H.-Z. Huang, Y.-K. Gu, X. Du, An interactive fuzzy multi-objective optimization method for engineering design, Engineering Applications of Artificial Intelligence 19 (2006) 451–460.

[40] Bradstreet, Lucas. The hypervolume indicator for multi-objective optimisation: calculation and use. Perth: University of Western Australia, 2011.

Abstract

"Human-inspired algorithms are part of metaheuristic algorithms, thus providing a suitable way to find solutions to real-world engineering problems. We also see a growing interest in these methods. The main motivation behind the proposal of the multi-objective algorithm MOCBOA is based on single-objective CBOA algorithm, In this algorithm, we combined domination and added solution archive, then tested it on some engineering problems, especially design problems. mechanical The results showed that the MOCBOA algorithm generated solutions for all problems, with significantly better performance than the comparison algorithms on the problem (Pressure vessel design and Vibrating platform design, Disc brake design, Speed reducer design), which confirms. its efficiency and ability to solve problems."

Keywords: Human-inspired algorithms; Metaheuristic algorithms; Engineering problems; Multi-objective algorithm (MOCBOA); Single-objective CBOA; Dominance

Résumé

"Les algorithmes inspirés par l'humain font partie des algorithmes métaheuristiques, constituant ainsi un moyen approprié pour trouver des solutions à des problèmes d'ingénierie réels. Nous constatons également un intérêt croissant pour ces méthodes. La principale motivation derrière la proposition de l'algorithme multi-objectif MOCBOA est basée sur l'algorithme CBOA à objectif unique. Dans cet algorithme, nous avons combiné la domination et ajouté une archive de solutions, puis l'avons testé sur certains problèmes d'ingénierie, en particulier des problèmes de conception mécanique. Les résultats ont montré que l'algorithme MOCBOA a généré des solutions pour tous les problèmes, avec des performances nettement meilleures que les algorithmes de comparaison sur le problème (Conception de récipient sous pression et conception de plate-forme vibrante, conception de frein à disque, conception de réducteur de vitesse), ce qui confirme son efficacité et sa capacité à résoudre des problèmes."

Mots clés : Algorithmes inspirés par l'humain ; Algorithmes métaheuristiques ; Problèmes d'ingénierie ; Algorithme multi-objectif (MOCBOA) ; Algorithme CBOA à objectif unique ; Domination

خلاصة

" الخوارزميات المستوحاة من الإنسان هي جزء من خوارزميات ما فوق الطبيعة، وبالتالي توفر طريقة مناسبة لإيجاد حلول للمشاكل الهندسية في العالم الحقيقي. ونحن نرى أيضًا اهتمامًا متزايدًا بهذه الأساليب. الدافع الرئيسي وراء اقتراح خوارزمية MOCBOA متعددة الأهداف هو بالاعتماد على خوارزمية CBOA أحادية الهدف، قمنا في هذه الخوارزمية

بدمج السيطرة وإضافة أرشيف الحلول، ثم اختبرناها على بعض المشاكل الهندسية، وخاصة المشاكل التصميمية الميكانيكية. وأظهرت النتائج أن خوارزمية MOCBOA ولدت حلولاً لجميع المشاكل، مع أداء أفضل بكثير من خوارزميات المقارنة حول المشكلة (تصميم أوعية الضغط وتصميم منصة الاهتزاز، تصميم قرص الفرامل، تصميم مخفض السرعة)، مما يؤكد كفاءتها وقدرتها على حل المشكلات."

الكلمات المفتاحية

الخوارزميات المستوحاة من الإنسان، خوارزميات ميتاهيرستيك، المشاكل الهندسية، خوارزمية متعددة الأهداف أحادية الهدف، السيطرة، CBOA ، خوارزمية(MOCBOA)

THE LIBRARY

https://fr.wikipedia.org/wiki/Probl%C3%A8me_NP-complet

https://www.nature.com/articles/s41598-022-19313-2