الجمهورية الجزائرية الديمقراطية الشعبية
République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université Amar Telidji- Laghouat

FACULTE : GENIE CIVIL ET ARCHITECTURE
DEPARTEMENT : GENIE CIVIL

MEMOIRE DE MASTER

Présenté par :

Goual Zineb

Boualoufa fatima Zahra

DOMAINE : Sciences et Techniques
FILIERE : Génie civil
OPTION : Structures

Thème

Structural optimization using recent metaheuristic algorithms

Jury de soutenance :

Nom et Prénom	Grade	Qualité
Khanfer Mohammed	Pr.	Président
Chettih Mohammed	Pr.	Examinateur
Mouattah Kaddour	Pr.	Rapporteur
Tadj Walid	MCA	Co-rapporteur

Promotion : Juin 2023

Dedication

I dedicate this modest work to:

my parents, Mom & Dad for their efforts and duaa;

my brothers: Mohamed Lahsen, Azzedine, Sofiane, Abdeldjebar and Mohsen sayah.

and Karima, Hayat, Assema, for being supporters and always by my side;

my dear grandma

Aicha, Ahmed Aness, Khadidja Malak, Djouairia

To my heart 'Nadjet' and 'fatima Zohra' for staying strong and always by my side;

To my cousin: Ahlem, Sara;

I dedicate this modest work to:

my parents, Mom & Dad for their efforts and duaa;

my sisters: Bahia, Maroua, Kheira, Ikram, and my brother Ilyas for being supporters and always by my side;

my dear grandma and grandpa

my friend's 'Zineb' and 'Ismahane' and 'Widad' for staying strong and always by my side;

my cousin: khaoula, Noura;

Thanks

Above all, we wish to thank God for giving us the strength and the ability to finish this work.

We would like to thank our Supervisors, Mr. Mouattah kadddour and Mr. Tadj walid for their patient guidance and their generous supports and encouragements.

To the jury members.

We thank from the bottom of our hearts and with great love our parents who never stopped believing in us during all our years of study.

Thanks also to our sisters and brothers, and to the whole families who has always encouraged us.

We thank our professors Ms. Belakhel H. Ms. Gotaicha M. Mr. Amara and Mr. Belaidi Akram Mr. souahli Hamza and Mr Goual Idriss

ملخص

إن معظم مشاكل الأمثلية الهيكلية محكومة بمعادلات غير خطية إلى حد كبير، وتتضمن العديد من متغيرات التصميم والقيود المعقدة المختلفة ولا يمكن حلها بكفاءة باستخدام تقنيات الأمثلية التقليدية (طرق التدرج (الحتمية)). وعليه ونظرًا للخواص غير الخطية وغير المحدبة للمسائل الهيكلية، قد تتقارب هذه الطرق فقط مع الحل الأمثل المحلي أو قد تتباعد عندما تكون التقديرات الأولية سيئة التقدير. لذلك فإن استخدام تقنيات الأمثلية العشوائية هو أكثر أهمية وفعالية. لذلك، تقترح الدراسة الحالية استعمال تقنيات الذكاء الاصطناعي عن طريق خوارزميات مطورة مؤخرا لحل هذه المسائل الهيكلية المعقدة. في هذه الحالة، تم اعتماد ثلاث خوارزميات هي خوارزمية الغراب وخوارزمية راو-1 بالإضافة الى خوارزمية تهجين بينهما للوصول الى النتائج المرغوبة. اعطت المنهجية المتبعة خلال هذا العمل نتائج جيدة وأثبتت الخوارزمية الهجينة فعاليتها من حيث الدقة والمتانة وسرعة التقارب نحو الحل الأمثل.

الكلمات الرئيسية: الأمثلية العشوائية، الأمثلية الهيكلية، خوارزمية بحث الغراب، خوارزمية راو-1، الخوارزمية الهجينة

Abstract

Most of the structural optimization problems are highly non-linear, involving many design variables and various complex constraints, which cannot be solved efficiently using traditional optimization techniques (gradient-based methods). Due to the highly non-linear and non-convex aspect of structural problems, these methods may converge only to a local optimum, or may diverge when initial estimates are not well designed. Therefore, the use of stochastic techniques is more interesting and efficient. As such, the present study presents a combined artificial intelligence technique, in this case, the Crow search Algorithm (CSA), Rao's Algorithm (Rao-1) and a new hybrid algorithm (CSARao-1) which have been adopted to solve structural problems in several types of optimizations. The methodology used in this work yielded good results, and the CSARao-1 hybrid algorithm proved to be the best performer in terms of accuracy, robustness and convergence speed.

Keywords: Stochastic optimization, structural optimization, CSA, Rao-1 algorithm, CSARao-1 algorithm.

Résumé

La plupart des problèmes d'optimisation de structures sont hautement non linéaires, impliquant de nombreuses variables de conception et de différentes contraintes complexes et ne peuvent pas être résolus efficacement en utilisant les techniques d'optimisation traditionnelles (méthodes de gradient (déterministes)). En raison de l'aspect hautement non linéaire et non convexe des problèmes de structures, ces méthodes peuvent converger uniquement vers un optimum local ou peuvent diverger lorsque les estimés initiaux ne sont judicieusement choisis. L'utilisation des techniques d'optimisation stochastique est, donc, plus intéressante et plus efficace. De ce fait, la présente étude propose des techniques de l'intelligence artificielle, en l'occurrence, l'algorithme de corbeau (CSA), l'algorithme de Rao (Rao-1) et un nouvel algorithme hybride (CSARao-1) pour résoudre des problèmes structuraux dans plusieurs types d'optimisation. La méthodologie utilisée dans ce travail a donnée de bons résultats et l'algorithme hybride CSARao-1 s'est avéré le plus performant en termes de précision, robustesse et vitesse de convergence.

Mots-clés : Optimisation stochastique, optimisation structurale, CSA, Rao-1 algorithme, CSARao-1 algorithme.

Contents

List of figures

.

List of tables

General Introduction

Civil engineering field offers an important research area. These researches involve the study of structural optimization problems. However, the highly non-linear problems involving many design variables and various complex constraints pose several problems that cannot be solved by simple analytical methods and require appropriate numerical methods.

The process of structural design may be regarded as an optimum design even though optimality is not explicitly pursued. Optimal structural design is driven by the limited material resources, environmental impacts and technological competition which demand lightweight, low-cost and high-performance structures. An optimal design is defined as the best feasible design that satisfies the prescribed performance criteria. With advances in high-speed computers, the objective of structural optimization is to maximize the performance of a structure or structural component. Structural optimization has the potential to become an automated design tool for practicing engineers in civil engineering industries. (Liang et al.,2000). Several optimization algorithms exist, which can be classified into two categories: based on the solutions produced, namely the classical or deterministic algorithms and the non-classical or the stochastic algorithms. The classical methods of optimization are useful in finding the optimum solution of continuous and differentiable functions. The main problem of these classical or determinist algorithm is that they cannot bypass local optimum; so, we have to move to the stochastic or metaheuristics algorithms.

The modern methods are conceptually different from the traditional mathematical programming techniques, most of these methods are based on certain characteristics and behaviour of biological, molecular, swarm of insects, and neurobiological systems. Metaheuristic techniques are capable of using search experience intelligently to explore and exploit the search space in a randomized manner, and the solution methods are inexact and near optimal. Metaheuristics are uncountable, they are classified in four categories: physics-based, swarm intelligence, evolutionary algorithms and bio-inspired.

In this study, we tried to take advantage of the benefits offered by metaheuristic by applying it to some structural optimization problems, such as truss, beam and limit analysis.

This study includes a general introduction, three chapters, a conclusion and perspectives:

The first chapter presents the theoretical notions on structural optimization.

The second chapter is dedicated to optimizations problems and the metaheuristics used in this work.

The third chapter is devoted to the application of the metaheuristics used to solve the selected problems.

Finally, we end this work with a conclusion, which recalls the objectives of this study and the results obtained as well as some perspectives.

Chapter one

Introduction to structural optimization

1.1 Introduction to the Optimization

This chapter introduces basic ideas and terminology of structural optimization that will be important to the remainder of this study.

The existence of optimization methods can be traced to the days of Newton, Lagrange, and Cauchy. The development of differential calculus methods of optimization was possible because of the contributions of Newton and Leibnitz to calculus. The foundations of calculus of variations, which deals with the minimization of functionals, were laid by Bernoulli, Euler, Lagrange, and Weierstrass. The method of optimization for constrained problems, which involves the addition of unknown multipliers, became known by the name of its inventor, Lagrange. Cauchy made the first application of the steepest descent method to solve unconstrained minimization problems. Despite these early contributions, very little progress was made until the middle of the twentieth century, when high-speed digital computers made implementation of the optimization procedures possible and stimulated further research on new methods. Spectacular advances followed, producing a massive literature on optimization techniques. This advancement also resulted in the emergence of several well-defined new areas in optimization theory.

Structural optimization" refers to an optimization which aims to find the best arrangement of structures or structural components to achieve certain objectives under prescribed conditions.

optimization could be described as the process of selecting of the best element(s) from among a series of available alternatives to get the best possible results when solving a particular problem (Galinier et al. 2013).

Optimization is the process of finding the best solution under given constraints on design construction and maintenances, it can be defined as the conditions that give the maximum or minimum value of a function. One of the principal objectives of structural optimization is minimizing the total cost of the structure. In construction projects, a lower cost is always desired on the premise of satisfying the requirements of structural performance.

Since the 20th century, with the advent and development of computational methods for structure design and analysis, optimization methods based on mathematical programming techniques have been proposed and adopted in the field of civil engineering in the past few decades.

In the typical process of structural optimization of finite dimensional structures, the cross-sectional properties, nodal locations, and member locations are chosen as design variables. There are many methods for structural optimization that are classified into:

• Nonlinear programming based on the gradients (sensitivity coefficients or derivatives) of the objective and constraint functions, which is the most popular and straightforward approach.

• Heuristic approaches, including genetic algorithm and simulated annealing, that do not need gradient information.

1.2 Performance-based design

The performance-based design is currently a popular design concept in the field of structural engineering. This concept describes the required and possessed performance of a structure or structural component being designed or evaluated. Structural response parameters such as stresses, strains, displacements and accelerations are employed as performance indices to evaluate the performance of structure.

There are usually multiple performance objectives that must be considered by the structural designer when designing a structure. Main performance objectives can be summarized as follows:

● Functionality

● Serviceability

● Strength

● Economy

The cost performance (economy) of a structure is of great importance to the owner and has a significant effect on the design of the structure. The cost of a structure includes the initial cost and the cost of maintenance, and is significantly influenced by the structural form, materials used and construction methods. As discussed in preceding section, topology optimization of structures can result in more material savings than the pure shape and sizing optimization. Therefore, to improve the cost performance of a structure, it is wiser to select a proper structural form rather than to modify the sizes of structural members.

The performance-based optimal design is to design a structure or structural component that can perform physical functions in a specified manner throughout its design life at minimum cost or weight (Liang et al.,2000). The optimization problem with multiple performance objectives are usually solved by selecting only one performance of the structure as the objective function and others are considered as constraints imposed on the structure. In performance-based optimal design, the weight of a structure is usually selected as the performance objective and structural response parameters such as stresses, displacements, overall stiffness and frequency are treated as performance-based constraints.

1.3 Classification of structural optimization

There are currently three different types of optimization methods which come under the heading of SO, these are: (1) size, (2) shape, and (3) topology optimization.

In **Size optimization**, the engineer or designer knows what the structure will looks like, but does not know the size of the components which make up that structure. For example, if a cantilever beam was going to be used, its length and position may be known, but not its cross-sectional dimensions. Another example would be a truss structure where its overall dimensions may be known but not the cross-sectional areas of each truss element (bar). it has the goal of maximizing the performance of a structure in terms of the weight and overall stiffness or strength while the equilibrium condition and the design constraints are satisfied. The design variable is the cross-sectional area of truss members or the thickness of a plate. In sizing optimization, the design domain is fixed during the optimization process

So basically, any feature of a structure where its size is required, but where all other aspects of the structure are known.

In **Shape optimization**, the unknown is the form or contour of some part of the boundary of a structural domain. The shape or boundary could either be represented by an unknown equation or by a set of points whose locations are unknown (Fig. 1.2). the objective is to find the optimal shape of the design domain, which maximizes its performance. The shape of the design domain is not fixed but rather is a design variable. In shape optimization, only the boundaries of the design domain are changed but not the topology of the domain.

Topology optimization of continuum structures is to determine the optimal number and locations of holes within the continuum design domain. In topology optimization, both topology and shape of a structure are the design variables. The two major distinctive features of topology optimization are that: (1) the elastic property of the material, as a function of its density, can vary over the entire design domain; and (2) material can be permanently removed from the design domain. There are several topology optimization methods which can be grouped into two categories: (1) Optimality Criteria methods and (2) Heuristic or Intuitive methods.

Figure1.1 shows sizing structural optimization problem formulated by optimizing the cross-sectional areas of truss members. (Peter w, et al.,2009)

Figure1.2 shows a shape optimization problem. Find the function $\eta(x)$, describing the shape of the beam-like structure. (Peter w. et al., 2009)

Figure1.3 shows a topology optimization of a truss. Bars are removed by letting cross-sectional areas take the value zero. (Peter w, et al.,2009)

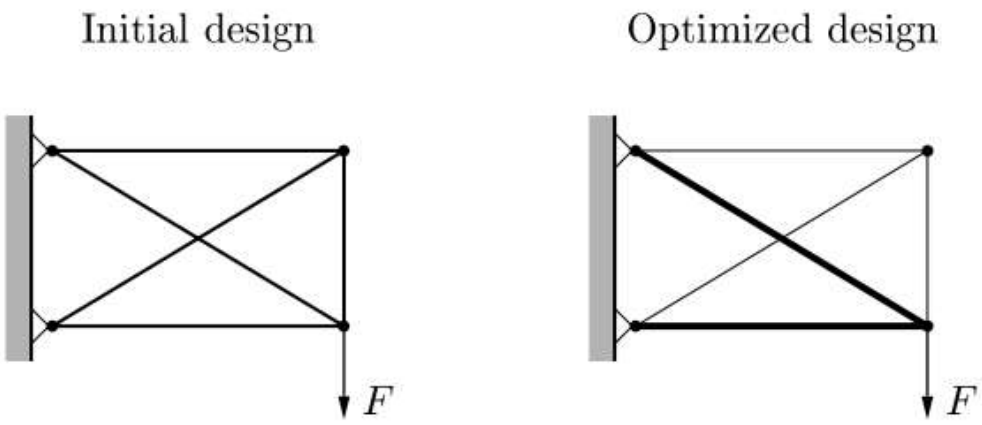

Fig1.1 A sizing structural optimization problem is formulated by optimizing the cross-sectional areas of truss members. (Peter w, et al.,2009)

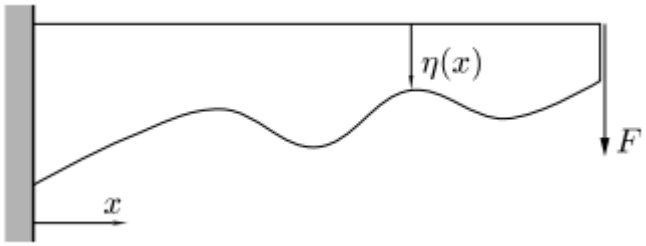

Fig 1.2 A shape optimization problem. Find the function η(x), describing the shape of the beam-like structure. (Peter w. et al., 2009)

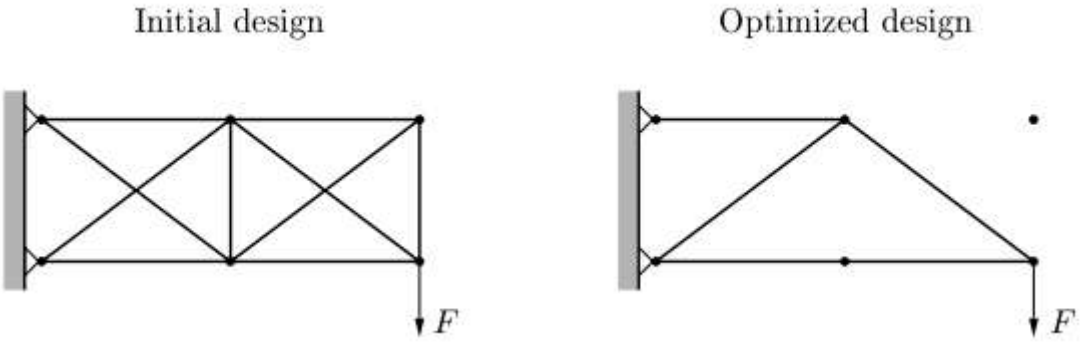

Fig 1.3 Topology optimization of a truss. Bars are removed by letting cross-sectional areas take the value zero. (Peter w, et al.,2009

1.4 Statement of an optimization problem

The following function and variables are always present in a structural optimization problem:

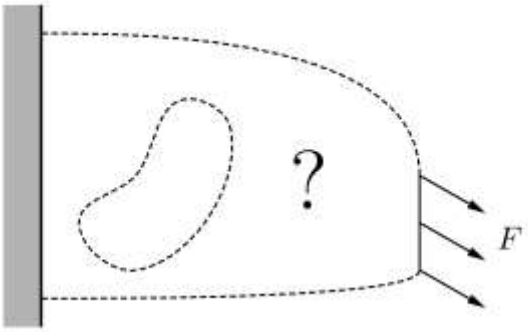

Fig. 1.4 Structural optimization problem. Find the structure which best transmits the load F to the support. Topology optimization of a truss. (Peter w, et al.,2009)

• *Objective function (f):* A function used to classify designs. For every possible design, f returns a number which indicates the goodness of the design. Usually, we choose *f* such that a small value is better than a large one (a minimization problem). Frequently *f* measures weight, displacement in a given direction, effective stress or even cost of production.

• *Design variable (x):* A function or vector that describes the design, and which can be changed during optimization. It may represent geometry or choice of material. When it describes geometry, it may relate to a sophisticated interpolation of shape or it may simply be the area of a bar, or the thickness of a sheet.

A general mathematical programming problem can be stated as follows:

$$\textbf{Find X} = \begin{Bmatrix} x1 \\ x2 \\ . \\ . \\ . \\ . \\ xn \end{Bmatrix} \textbf{ which minimizes f(X)} \qquad 1.1$$

Submitted to the constraints $gj\,(x) \leq 0, j = 1, 2, \ldots, m$

$$lj\,(x) = 0\,, j = 1, 2, \ldots, p$$

- x: n-dimensional vector called the design vector
- $f\,(x)$: termed the objective function
- $gj\,(x)$; $lj\,(x)$: inequality and equality constraints

The problem stated in Eq. (1.1) is called a **constrained optimization problem.**

. If we try to classify optimization problems according to the number of objectives, then there are two categories: mono-objective and multi-objective. Multi-objective optimization is also referred to as multicriteria or even multi-attribute optimization in the literature. In real-world problems, most optimization tasks are multi-objective (Yang.,2014).

The following table summarise the classification of optimization problems based on some other options:

Table 1.1: Classification of optimization problems (Tadj.,2019)

Optimization	objective	• Mono-objective • Multi-objective
	constraints	• Constrained • Unconstrained
	Landscape of $f(x)$	• Unimodal • Multimodal
	Behaviour	• Linear • Non-linear
	variables	• Discretes • Continues
	Techniques	• deterministic • stochastic

1.5 Optimization Techniques

The oldest optimization technique, which is still in use, is the trial-and-error one, it is a manual and subjective technique based on trial and error guided by the experience of the operator, it is very slow and lacks precision, consequently various algorithms optimization was created, and the choice of which one is appropriate for a given problem remains a very active area of research. As the Theorem (No-Free Lunch) indicates, no optimization technique is superior to all others for all problems (Wolpert and Macready,1997); there is no universal optimization algorithm. For different types of optimization problems, we frequently have to

use different optimization techniques, as some algorithms are more suited to certain types of optimizations than others. In an optimization process, the ideal is to find eligible solutions within a short period of time (Tadj ,2019).

The optimization algorithms are divided into two categories: deterministic algorithms and stochastic algorithms.

Figure1.5: shows the Classification of optimization techniques. (Wali, et al .2021).

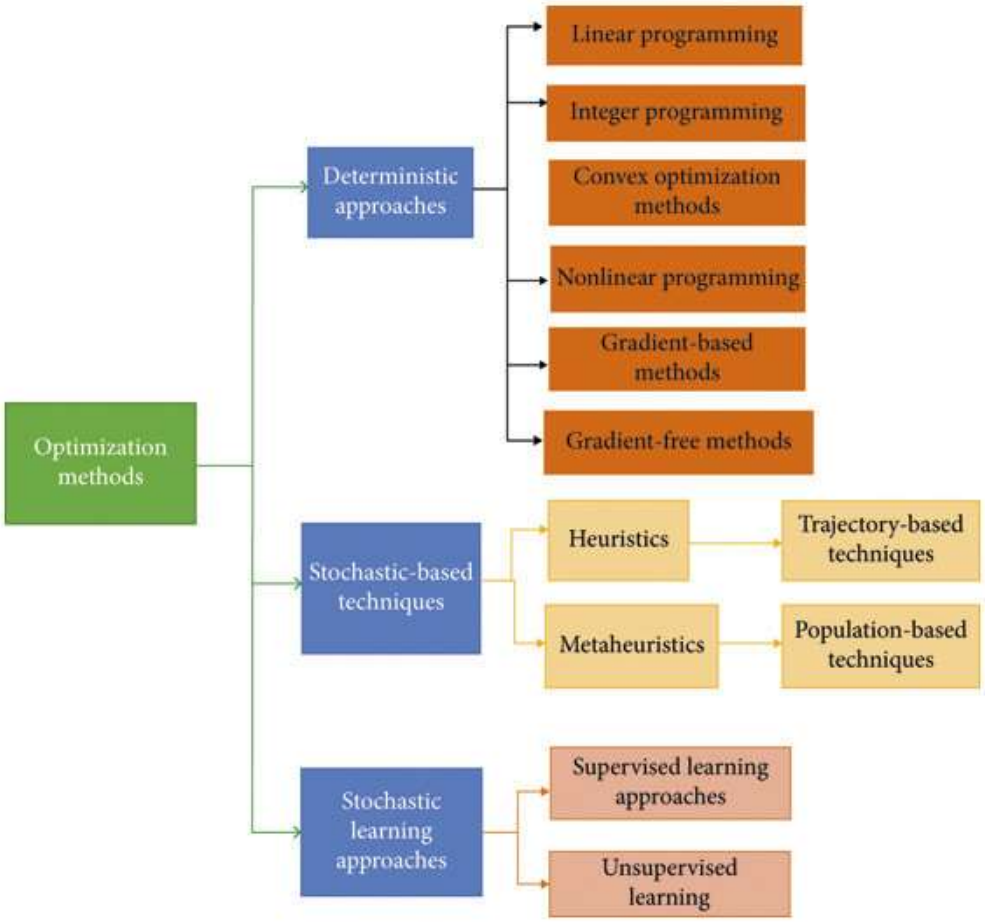

Figure 1.5: Classification of optimization techniques. (Wali, et al .2021)

We can categorize optimization problems into two types based on the solutions produced, namely the classical or deterministic algorithms and the non-classical or non-deterministic also called the stochastic algorithms. The stochastic algorithm can be further classified into the heuristic and metaheuristic techniques (Okwu, andTartibu, 2021).

1.5.1 Deterministic algorithms

Deterministic optimization algorithms (eg., descent gradient, Gauss-Newton, Levenberg. etc.) start their search from an initial guess value. As the name suggests, these techniques follow the same path of research, that means from the same starting values we will always have the same solutions. Take, for example, the downward or ascending gradient algorithm. It is a gradient-based algorithm, also called the steepest slope algorithm. The idea of this algorithm is simple, and can be summed up in the following steps (Kruse et al., 2016):

1. The algorithm starts at an initial point $x^{(0)} = (x1^{(0)}, \cdots x_{Npar}{}^{(0)})$;
2. Calculate the gradient at current point $\nabla f\ (x^{(k)}) = \{ \frac{\partial}{\partial x1} f\ (x^{(k)}), \cdot \cdots, \frac{\partial}{\partial xNpar} f\ (x^{(k)}$);
3. Calculating the next position $x^{(k+1)} = x^{(k)} + \eta \nabla_x f\ (x^{(k)})$, η: discretisation step
4. Repeat steps 2 and 3 to move through the search space in (or against) the direction of the steepest slope of the objective function, until you reach an optimum;
5. Stop condition: The algorithm stops when a maximum number of iterations is reached, or the value of the gradient becomes below a certain threshold.

As shown in Figure 1.6, the effectiveness of this gradient-based deterministic algorithm depends on the parameters of the initial estimate $x^{(0)}$ and the step discretisation value η.

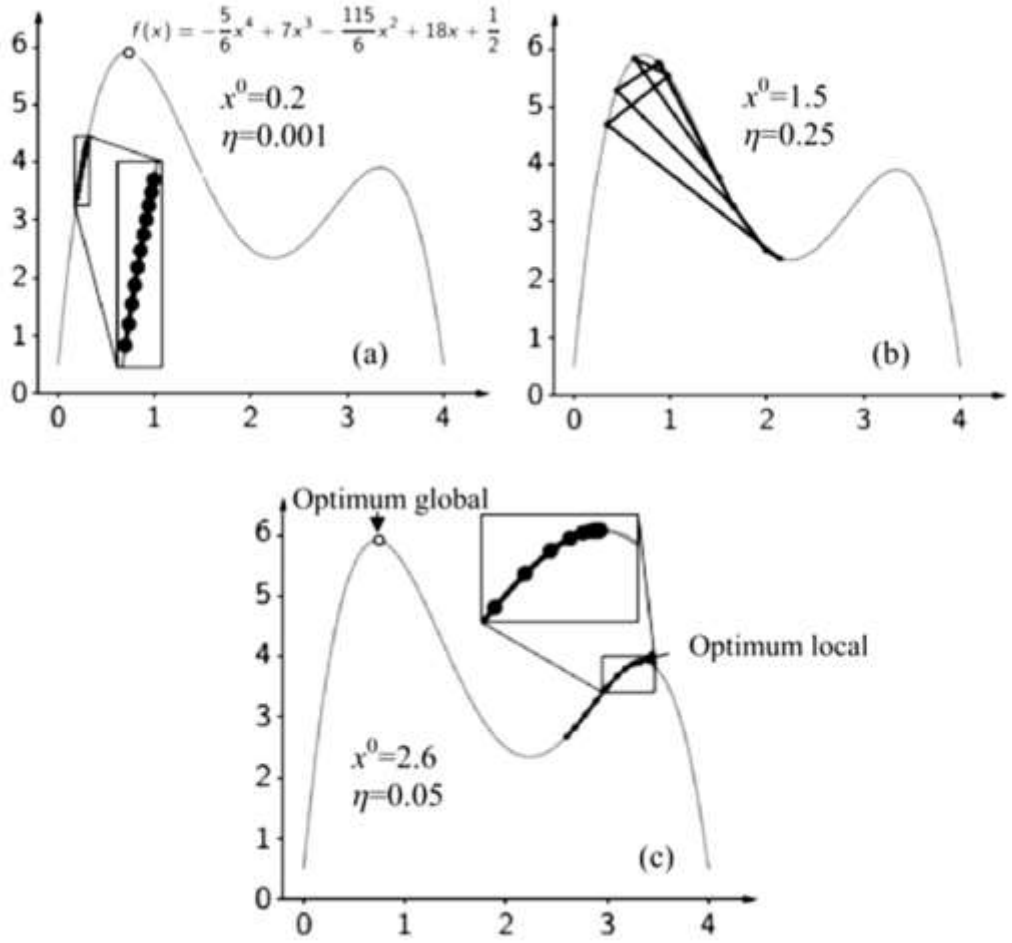

Figure 1.6: Maximizing an f(x) function by the ascending gradient algorithm (Tadj., 2019).

For small value of η (Figure.1.6a), the convergence of the algorithm is very slow requiring high computational time. When the η value is high (Figure 2.4b), oscillations are observed, the solution jumps back and forth.

When the initial estimate $x^{(0)}$ is far from the global optimum (Figure.1.6c), in this case the solution thus corresponds to a local optimum.

Furthermore, deterministic techniques are gradient-based, that means, the evaluation of the derivatives of the objective function in relation to the different parameters is essential, which adds a derivability condition to the objective function. These deterministic techniques are appropriate for convex problems. Unfortunately, this condition is not always verified such as in structural problems. In consequence, solutions obtained using such techniques can be trapped in local optimal. Sometimes the user of this type algorithms is forced to do multiple executions with different starting points when the objective function has many local optimal. (Tadj., 2019)

The main problem of these classical or determinist algorithm is that they cannot bypass local optimum; so, we have to move to the stochastic or metaheuristics algorithms.

1.5.2 Stochastics algorithms

Stochastic optimization algorithms are an ensemble of artificial intelligence methods to solve optimization problems deemed difficult. These are gradient-free methods that require only direct evaluation of the objective function, they have a random character. Stochastic algorithms are called heuristics or metaheuristics (Yang.,2010).

Heuristic algorithms use a trial-and-error approach in generating new solutions, while the metaheuristic algorithms are a higher-level heuristic with the use of memory, solution history and other forms of 'learning' strategy. Nowadays, most metaheuristic algorithms are nature inspired algorithms and most such algorithms are based on swarm intelligence (Yang.,2018).

1.6 Metaheuristics

Metaheuristic algorithms are now used all over, to solve complex optimization problems. Most of these algorithms are based on biological phenomena and behaviour of animals.

Metaheuristic optimization algorithms are growing in popularity in engineering applications for several reasons:

First, they depend on rather simple concepts and are easy to implement;

Secondly, they require no gradient information;

Third, they can bypass local optima;

Finally, they can be utilized in a set of problems in different disciplines (Asef.,2021).

They are generally based on the concept of population, i.e., they solve optimization problems by a population of potential solutions.

In a population-based algorithm, the steps of the design methodology are as follows. A general flowchart is presented in Figure 1.7

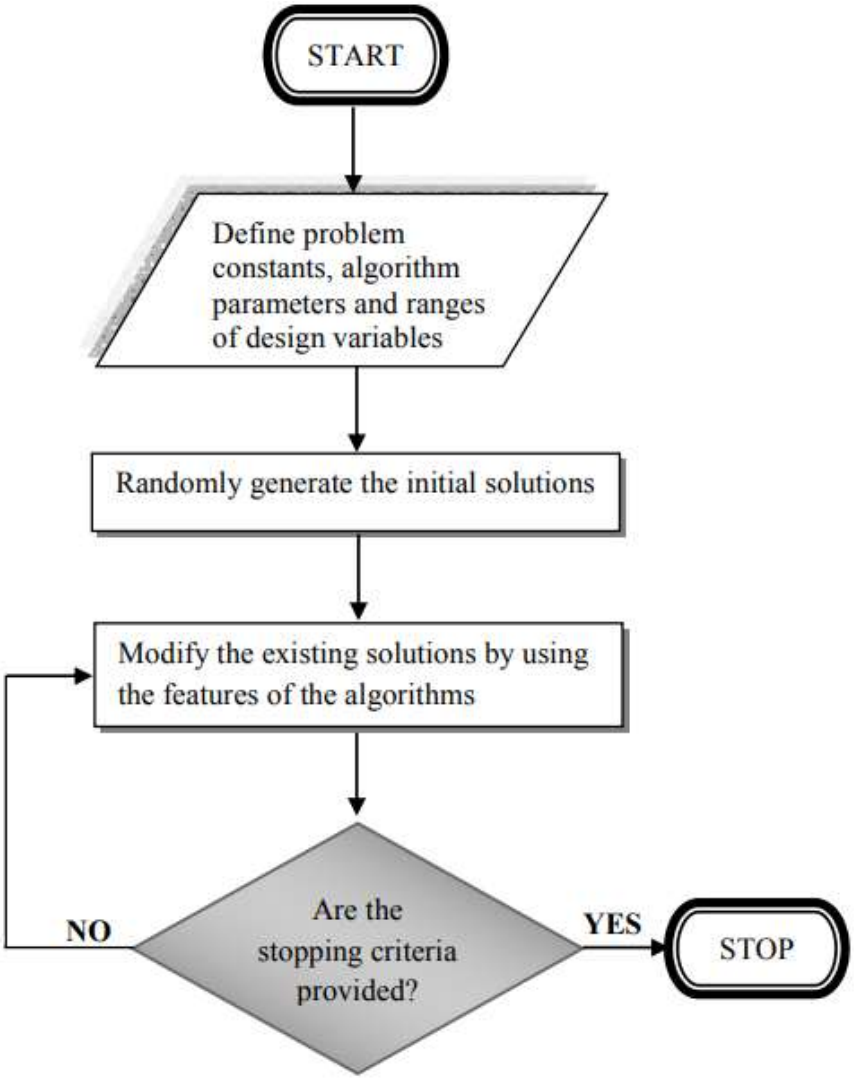

Figure 1.7. General flowchart of metaheuristic algorithms (Yusuf C, et al,2021)

Step I: define the problem constants, algorithm parameters and ranges of design variables.

Step II: randomly generate the initial solutions.

Step III: modify the existing solutions by using the features of the algorithms.

Step IV: continue iterative

Step III: until the stopping criterion (or criteria) is met.

Figure 1.8 shows the population-based metaheuristic algorithms which are classified as four categories called physics-based algorithms, swarm intelligence algorithms, evolutionary algorithms and bio-inspired algorithms (Dhiman, and Kumar., 2018).

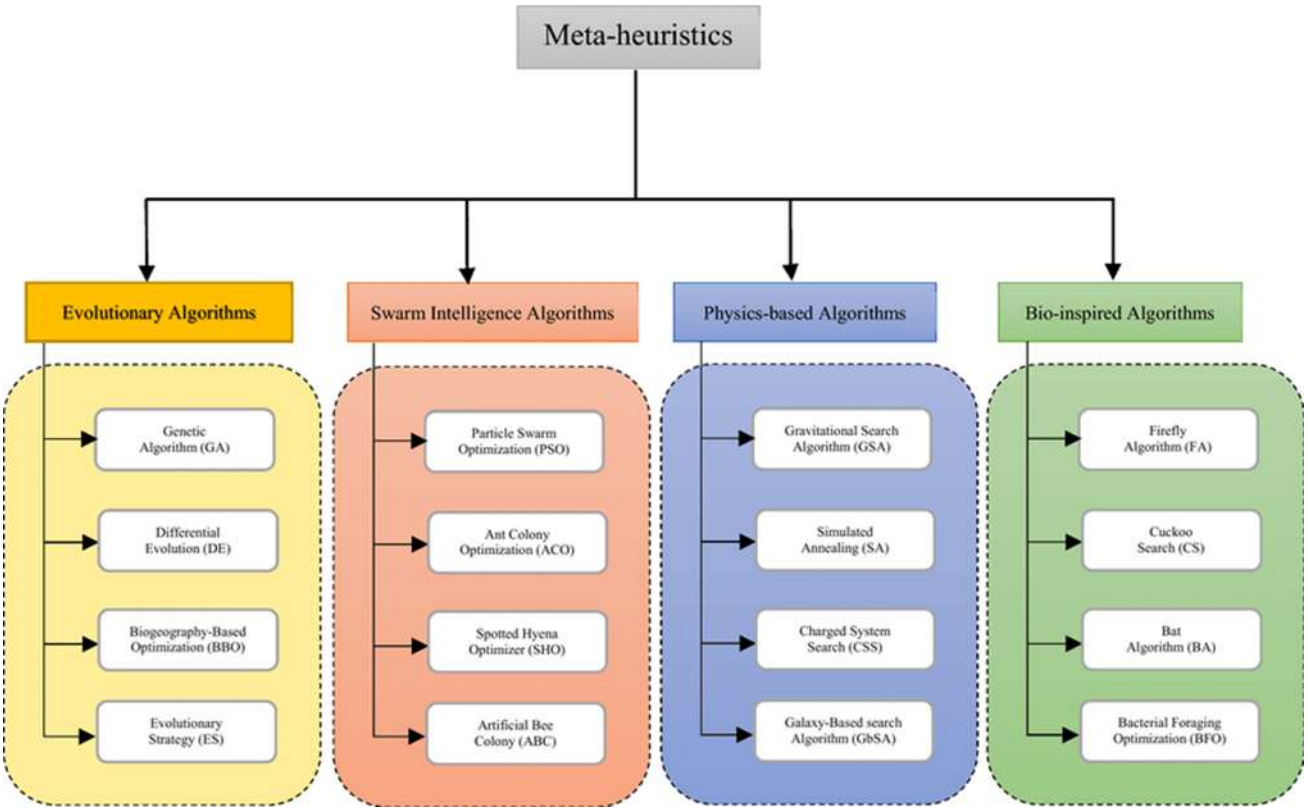

Figure 1.8: The classification of the metaheuristic algorithms (Dhiman, and Kumar.,2018).

In this chapter, several metaheuristic algorithms are briefly summarized; these algorithms are Evolutionary algorithms, swarm intelligence and other metaheuristic algorithms.

1.6.1 Evolutionary Algorithms

The EAs are rapidly expanding in the area of artificial intelligence research. Over the past two decades, there has been increasing attention in algorithms, which are based on the principle of natural evolution (i.e., the survival of the fittest) (Fogel, et al.,966).

The EAs are the population based random search techniques guided with some heuristics (also called as meta-heuristic techniques). The EAs consist of a population of individuals, each representing a search point in the space of feasible solutions and is exposed to a collective learning process which proceeds from generation to generation (Brownlee, 2011).

The population is randomly initialized and then subjected to the process of selection, recombination, and mutation through various generations, in such a way that the newly created generations evolve towards more effective regions of the search space. The progress in the search is achieved by evaluating the fitness of all individuals in the population, choosing the individuals with a better fitness value, and combining them to create new

individuals with an increased likelihood of improved fitness. After some generations, the program converges, and the best individual represents the optimum (or near optimum) solution.

There exist many EAs, but the basic structure of any evolutionary algorithm is presented in Figure 1.9

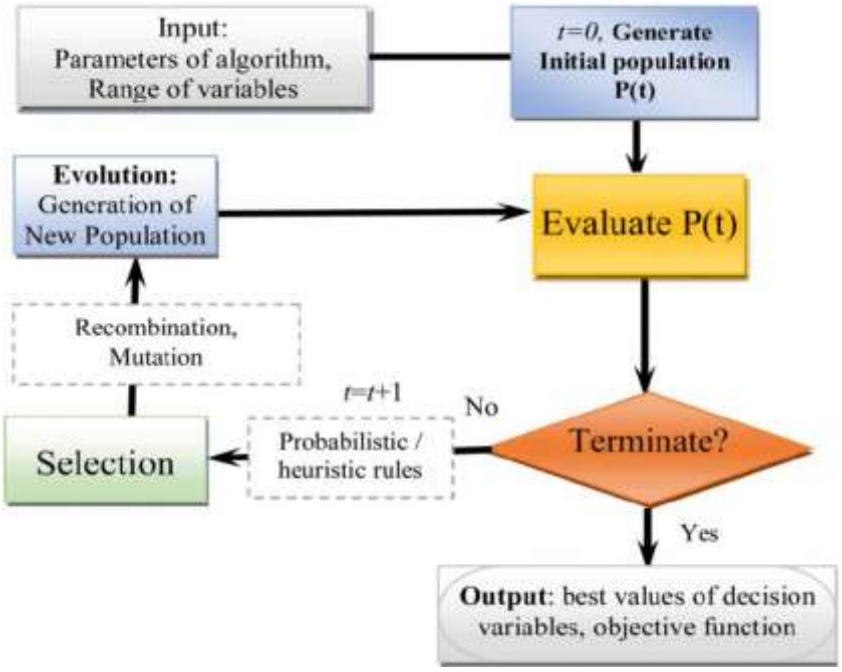

Figure 1.9: The basic structure of an evolutionary algorithm (EA) (Reddy ,and Kumar.,2021)

The key steps involved in EA include (Reddy and Kumar.,2021):

1. Seeding the population using random generation;
2. Rate the fitness of each individual in the population;
3. Repeat the evolution steps until stopping criterion satisfied:
    a) Choose the individuals for reproduction;
    b) Perform genetic operations to generate the offspring;
    c) Evaluate the individual fitness of the offspring;
    d) Exchange the least fit individuals with new best fit individuals.
4. Report the best solution of the fittest individual.

The most popular and oldest algorithms are genetic algorithms. For that reason, the other newly generated evolutionary algorithms are referred to as variants of genetic algorithms. In this section, genetic algorithms are briefly explained.

A. Genetic Algorithms

Genetic algorithms (GA) are one of the oldest metaheuristic algorithms that are still active in engineering problems. John Holland used crossover, mutation and selection in line with the evolution theory of Charles Darwin (Holland.,1975). Goldberg and Samton (1986) first used GA for structural optimization by solving a 10-bar truss structure optimization problem. With the development of the new generation algorithm, the classical form of GA was outperformed; however, new variants of GA and the implementation of the features of GA on other algorithms may still be effective on several engineering problems. (Yusuf C, et al.,2021)

GA was initially developed by John Henry Holland in 1960 with the purpose to understand the phenomenon of natural adaptation, and how this mechanism could be implemented into computers systems to fix complex problems.

In GA, a population of N solutions $xi = [xi,1, xi,2, \ldots, xi, npar]$ is first initialized; each of such solutions (called chromosomes). Every iteration (also called generation) of GA's evolution process, the chromosome population is modified by applying an ensemble of three evolutionary operators, namely: crossover, selection and mutation. For the selection operation, GA randomly selects a pair of chromosomes from within the entire chromosome population, based on their individual selection probabilities. For the crossover operation, the selected chromosomes (now called parents) are recombined to produce two new chromosomes (referred as offspring). Lastly, for the mutation operation, some parameters of the newly generated offspring are perturbed. Mutation can occur over each parameter in the chromosome with a particular probability. This process of selection, crossover, and mutation of individuals takes place until a population of N new chromosomes has been produced, and then, the N best chromosomes among the original and new populations are taken for the next generation, while the remainder individuals are discarded. (Fausto, et al., 2019).

1.6.2 Swarm intelligence

The behaviour of swarms is imitated in algorithms using swarm intelligence. In nature, every living thing has a specific process to find food or prey, reproduce and sustain their species. These processes have a final objective, as we need a precise result in engineering problems. For that reason, the algorithms based on swarm intelligence are suitable and successful in solving structural engineering problems. There are many types of swarm intelligence-based

metaheuristic algorithms. Particle swarm optimization (PSO), developed by Kennedy and Eberhart (1995), is the best-known algorithm. The movement of members in a flock of birds or a school of fish is formulated in PSO. (Yusuf C, et al.,2021).

Swarm intelligence is a natural or artificial self-adaptive, decentralized system comprising of interaction of individuals with one another and their environment. The most common source of inspiration is from nature, especially biological systems. As a part of this domain, we explore further into countless algorithms as it is cited in Table 1.2 (Hussien.,2020).

Table 1.2: Examples of swarm intelligence algorithms**.** (Hussien.,2020).

No.	Algorithm	Abb.	Authors	Year
1	Ant colony optimization [32]	ACO	Dorigo	1992
2	Particle swarm algorithm [23]	PSO	Kennedy and Eberhart	1995
3	Artificial fish swarm algorithm [67]	AFSA	Li et al.	2002
4	Bacterial foraging optimization algorithm [68]	BFOA	Passino	2002
5	Glowworm swarm optimization [69]		Krishnanand and Ghose	2005
6	Cat swarm optimization [26]	CSA	Chu et al.	2006
7	Artificial bee colony [70], [71]	ABC	Karaboga and Basturk	2007
8	Cuckoo search [28]	CS	Yang and Deb	2009
9	Bat algorithm [31]	BA	Yang	2010
10	Firefly algorithm [27]	FA	Yang	2010
11	Krill herd algorithm [34]	KH	Gandomi and Alavi	2012
12	Dolphin echolocation [72]		Kaveh and Farhoudi	2013
13	Chicken swarm optimization [35]	CSO	Meng et al.	2014
14	Grey wolf optimizer [36]	GWO	Mirjalili et al.	2014
15	Ant lion optimizer [38]	ALO	Mirjalili	2015
16	Dragonfly algorithm [43]	DA	Mirjalili	2015
17	Whale optimization algorithm [42]	WOA	Mirjalili and Lewis	2016
18	Grasshopper optimization algorithm [47]	GOA	Saremi et al.	2017
19	Butterfly-inspired algorithm [75]	BOA	Qi et al.	2017
20	Salp swarm algorithm [48]	SSA	Mirjalili et al.	2017
21	Equilibrium optimizer [49]	EO	Faramarzi et al.	2019
22	Bald eagle search [50]	BES	Alsattar et al.	2019
23	Harris hawks optimization [51]	HHO	Heidari et al.	2019
24	Nuclear reaction optimization [52]	NRO	Wei et al.	2019
25	Slime mould algorithm [53]	SMA	Li et al.	2020
26	Border collie optimization [54]	BCO	Dutta et al.	2020

A. Particle swarm optimization

Particle swarm optimization, or PSO, was evolved by Kennedy and Eberhart in 1995 and has become amongst the most widely applied swarm-intelligence-based algorithms by reason of its simplicity and flexibility. instead of using the mutation/crossover or pheromone, it uses real number randomness and global communication among the swarm particles. Therefore, it is also easier to implement due to the fact that there is no encoding or decoding of the parameters into binary strings as with those in genetic algorithms where real-number strings can also be used. (Yang., 2014).

As shown in figure 1.10 Schematic representation of the motion of a particle in PSO moving toward the global best $g*$ and the current best $xi*$ for each particle i (Yang., 2014)

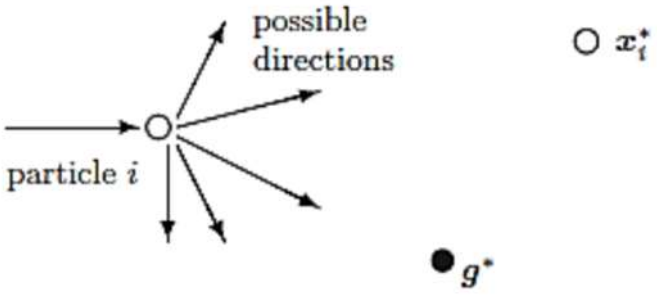

Figure 1.10: Schematic representation of the motion of a particle in PSO moving toward the global best g∗ and the current best xi ∗ for each particle i (Yang., 2014)

a) **Crow search algorithm (CSA**)

Crow Search Algorithm (CSA) is one of the latest algorithms developed by Alireza Askarzadeh in 2016, which simulates the crow actions in storing their food and retrieving it when they need it. Since its appearance, CSA has been widely used and utilised to different optimization issues (Hussien, et al., 2020).

CSA has gained huge attention from all researchers and scientists all over the world. According to Google Scholar 4868 times (accessed in 18rd June 2023): 4580 in journals, 175 in conferences, 98 in book chapters, and 15 in review papers. Table 1.2 shows the top 10 journals with the highest paper numbers dealing with CSA. Also, Figure 1.11 shows Distribution of CSA related papers in many applications, as reported by Scopus

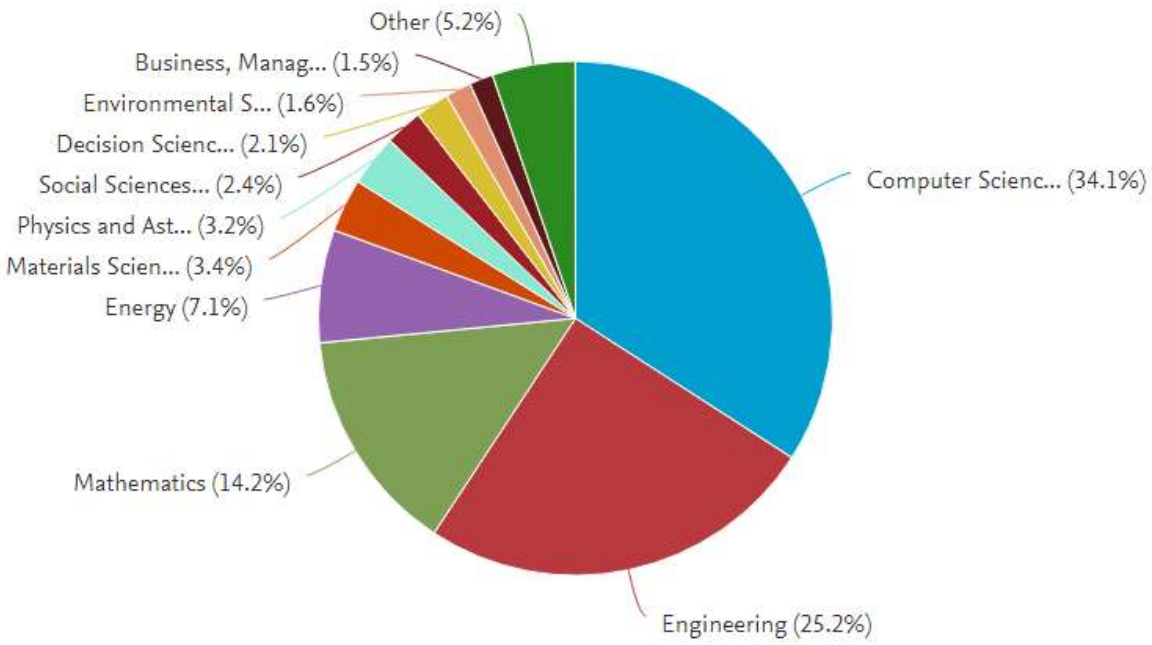

Figure 1.11: Distribution of CSA related papers in many applications, as reported by Scopus**.** (Hussien, et al., 2020).

Table 1.3: The top 10 journals with the largest number of the papers on CSA (Hussien, et al., 2020).

Rank	Journals	Number of papers
1	Applied Soft Computing Journal	24
2	Neural Computing And Applications	15
3	IEEE Access	14
4	Advances In Intelligent Systems And Computing	13
5	Expert Systems With Applications	12
6	Soft Computing	11
6	Studies In Computational Intelligence	11
8	Applied Intelligence	7
8	Swarm And Evolutionary Computation	7
9	Applied Mathematical Modelling	6
9	Engineering Applications Of Artificial Intelligence	6

1.6.3 Rao-1 algorithm:

More recently, Rao (2020) presented three iterative optimization algorithms without needing any natural or artificial metaphors. These metaphor-less algorithms named Rao-1, Rao-2 and Rao-3 are population-based and have no algorithm-dependent parameters. In this work, we focus on the Rao-1 algorithm because of its simple working mode. It is possible to develop potential optimization algorithms without the need of using metaphors related to the behaviour of animals, birds, insects, societies, cultures, planets, musical instruments, chemical

reactions, physical reactions, etc. The proposed three optimization algorithms are not based on any metaphor or algorithm-specific parameters. These require only the tuning of the common controlling parameters of the algorithm for working (e.g., population size and the number of iterations). (Rao,2020)

The research community may take advantage of these algorithms by adapting the same for solving different unconstrained and constrained optimization problems. (Rao,2019).

1.7 Applications of meta heuristics in the field of civil engineering

Civil engineering is defined as a discipline dealing with the design, construction, operation, and maintenance of buildings and infrastructures including a variety of works such as residence, bridges, and roads. Therefore, there has been growing interests in improving the social, economic, and environmental performance of civil engineering projects. Since the 20th century, with the advent and development of computational methods for structure design and analysis, optimization methods based on mathematical programming techniques have been proposed and adopted in the field of civil engineering in the past few decades. (Mei, L.; Wang, Q. 2021)

Several types of objectives used in the optimization of structural problems in civil engineering. These are:

A. Weight

Minimization of the total weight of structural systems is one of the best-known objectives in structural engineering. There are several advantages to constructing a light building. The first advantage is saving on the material. A lighter building may have structural members with smaller cross-sections. Thus, this goal of structural engineering is achieved. Economy factor: The total cost of the structure decreases. Ecology factor: Lower CO2 emissions are produced because of the production and transfer of materials. Architectural and aesthetics factors: The use of a smaller cross-section is effective in the visual and architectural concept of a structure. Comfort factor: smaller cross-sections are easy on production and spacious areas may be provided. Stress factor: By reduction of the weight of structural elements, the self-weight acting as the dead load of the structure is also reduced. In this case, the internal forces become lower and the stresses on the structural elements are reduced. Thus, structural members become safe and the failure of members may be prevented. Seismic factor: The weight of the structure is directly related to the vertical load acting on the structure. According to Newton's

second law the force is the multiple of the acceleration (a) and mass (m) of the rigid body. Since the mass is the weight (W) of the rigid body divided by gravity (g), the increase of the weight affects the force (F).

B. Cost

Building a civil structure with low costs is one of the primary goals of engineers. Generally, the cost is reduced by designing a light structure, because the cost of materials is generally priced with their weights or volumes. If the price index is volume, it is also related to the weight of the structural member, according to the density of the material. Weight cannot be the only parameter in the minimization of the cost of composite materials like reinforced concrete. In reinforced concrete (RC), the components are concrete and steel and they have different unit prices and different behaviours. Therefore, the parameter to be minimized should be combined cost, instead of weight. Steel is much more expensive than concrete, and thus this must be taken into account in the optimization process.

C. Response

In structural control applications, the objective is to reduce structural vibration. The reduction of the amplitude of structural vibration is one of the goals. In this goal, the effectiveness of the optimum control system is measured according to the reduction of a maximum structural response, compared to a structure without any control system. The response can be used as an objective of the structural control applications, but the effectiveness is a better index for the evaluation of the control system.

D. CO2 emissions in construction

CO2 emissions result from the process of production and transfer of materials, such as concrete, reinforcements and formwork. In soil interacting structures, excavation and compacted backfill are also creators of CO2 emissions. (Yusuf C, et al,2021)

1.8 Conclusion

In the first chapter we have touched some essential theoretical foundations for structural optimisation world from the classification to the process of optimization. Then, we mentioned some EA and SI-based algorithms and subsequently presented their main components, characteristics and properties. We focused on the crow search algorithm (CSA) and Rao1 algorithm that will be used in the next chapter. Finally, the applications of some optimization techniques in the field of civil engineering were also presented.

Chapter Tow

Metaheuristic algorithms

Csa; rao-1; the csarao-1 hybrid algorithm

2.1. Introduction

Many optimization problems in engineering and science applications in general are highly complex and challenging and thereby require novel problem-solving approaches (Yang et al. 2017).

The field of optimization has been continuously flooded with new metaheuristic algorithms ever since the introduction of the 'no free lunch theorem' (Wolpert and Macready 1997). This theorem asserts that there is no universally efficient algorithm capable of solving all optimization problems. One of the main reasons for the popularity and success of metaheuristics is that they have been developed by imitating the most successful processes in nature, such as biological systems and physical processes. Metaheuristic algorithms use a compromise between the exploitation (intensification)and global exploration (diversification) of the search space (Tadj, 2019).

Metaheuristics can be divided into two groups, including metaheuristics based on the concept of population (e.g., Genetic Algorithms, Differential Evolution, Particle swarms, Ant colonies, etc.) and those with single solutions, also called trajectory methods (e.g., Simulated Annealing) (Yang, 2010).

This chapter presents three algorithms in detail: the first is the crow search algorithm CSA; This algorithm is based on the concept of population; it manipulates a population of potential solutions through several generations in order to converge to an optimal or near-optimal solution. The second algorithm is Rao; This metaphor-less algorithm based on the best and worst solutions in the population and the random interactions between the candidate solutions. Just like TLBO algorithm (Rao, 2015) and Jaya algorithm (Rao, 2016; Rao, 2019), these algorithms do not require any algorithm-specific parameters and thus the designer’s burden to tune the algorithm-specific parameters to get the best results is eliminated. (Rao, 2020)

The third algorithm is the hybrid CSARao-1; can be seen as an improved version of the CSA has been proposed by (Tadj et al. ,2021).

2.2 Crow search algorithm (CSA)

Crows are widely distributed genus of birds which are now considered to be among the world's most intelligent animals. As a group, crows show remarkable examples of intelligence and often score very highly on intelligence tests. They can memorize faces, use tools, communicate in sophisticated ways and hide and retrieve food across seasons in a crow flock, there is a behaviour which has many similarities with an optimization process. According to this behaviour, crows hide their excess food in certain positions (hiding places) of the environment and retrieve the stored food when it is needed. Crows are greedy birds since they follow each other to obtain better food sources. Finding food source hidden by a crow is not an easy work since if a crow finds another one is following it, the crow tries to fool that crow by going to another position of the environment. From optimization point of view, the crows are searchers, the environment is the search space, each position of the environment is corresponding to a feasible solution, the quality of food source is objective (fitness) function and the best food source of the environment is the global solution of the problem. Based on these similarities, CSA attempts to simulate the intelligent behaviour of the crows to find the solution of optimization problems.

Based on the intelligent behaviour of crows, a novel metaheuristic algorithm, called CSA, is proposed recently. CSA is population-based optimization algorithm which is rather simple with two adjustable parameters (flight length and awareness probability) only, which in turn makes it very attractive for applications in different engineering areas. In CSA, the parameter of awareness probability is directly used to control the diversity of the algorithm.

Crows (crow family or corvids) are considered the most intelligent birds. They contain the largest brain relative to their body size. Based on a brain-to-body ratio, their brain is slightly lower than a human brain. They have demonstrated self-awareness in mirror tests and have tool-making ability. Crows can remember faces and warn each other when an unfriendly one approaches. Moreover, they can communicate in sophisticated ways and recall their food's hiding place up to several months later. Crows have been known to watch other birds, observe where the other birds hide their food, and steal it once the owner leaves. If a crow has committed thievery, it will take extra precautions such as moving hiding places to avoid being a future victim. In fact, they use their own experience of having been a thief to predict the behaviour of a pilferer, and can determine the safest course to protect their caches from being pilfered., based on the above-mentioned intelligent behaviours, a population-based metaheuristic algorithm, CSA, is developed.

The Principles of CSA are listed as follow:

- Crows live in the form of flock.
- Crows memorize the position of their hiding places.
- Crows follow each other to do thievery.

- Crows protect their caches from being pilfered by a probability

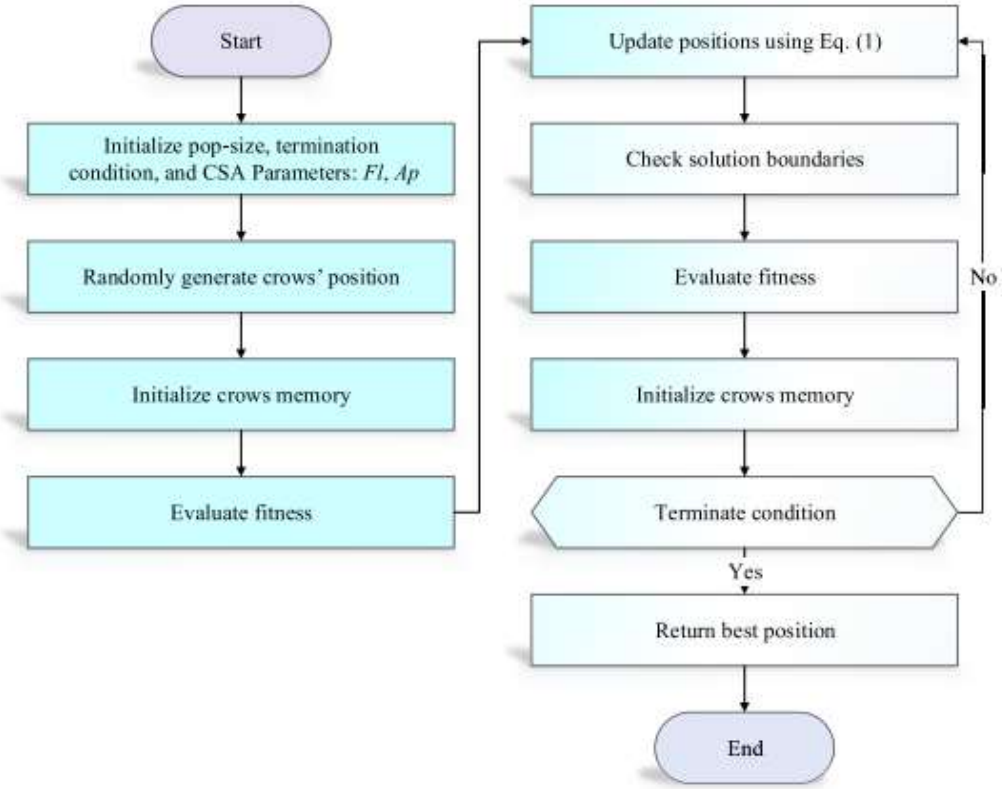

Figure 2.1: represent the flowchart of CSA. (Hussien, et al., 2020).

2.2.1 CSA implementation for optimization

The main phases of CSA can be shown as follows (Askarzadeh.,2016):

Step 1: Initialize problem and adjustable parameters The optimization problem, decision variables and constraints are defined. Then, the adjustable parameters of CSA (flock size (N), maximum number of iterations (itermax), flight length (fl) and awareness probability (AP)) are valued.

Step 2: Initialize position and memory of crows N crows are randomly positioned in a d-dimensional search space as the members of the flock. Each crow denotes a feasible solution of the problem and d is the number of decision variables.

$$Crows = \begin{bmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,Npar} & x_{1,Npar+1} = f(x_{1,j}) \\ x_{2,1} & x_{2,2} & \cdots & x_{2,Npar} & x_{2,Npar+1} = f(x_{2,j}) \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ x_{N,1} & x_{N,2} & \cdots & x_{N,Npar} & x_{N,Npar+1} = f(x_{N,j}) \end{bmatrix} \quad 2.1$$

The memory of each crow is initialized. Since at the initial iteration, the crows have no experiences, it is assumed that they have hidden their foods at their initial positions.

$$Memory = \begin{bmatrix} m_{1,1} & m_{1,2} & \cdots & m_{1,Npar} & m_{1,Npar+1} = f(m_{1,j}) \\ m_{2,1} & m_{2,2} & \cdots & m_{2,Npar} & m_{2,Npar+1} = f(m_{2,j}) \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ m_{N,1} & m_{N,2} & \cdots & m_{N,Npar} & m_{N,Npar+1} = f(m_{N,j}) \end{bmatrix} \quad 2.2$$

Step 3: Evaluate fitness (objective) function for each crow, the quality of its position is computed by inserting the decision variable values into the objective function.

Step 4: Generate new position Crows generate new position in the search space as follows: suppose crow i wants to generate a new position. For this aim, this crow randomly selects one of the flock crows (for example crow j) and follows it to discover the position of the foods hidden by this crow (mj). The new position of crow i is obtained by this process is repeated for all the crows.

$$x_{i,j,iter+1} = \begin{cases} x_{i,j,iter} + rand_i[0,1] \times fl \times (m_{k,j,iter} - x_{i,j,iter}) & \text{si } rand_k[0,1] \geq AP \\ \text{position } aléatoire & \text{si } rand_k[0,1] < AP \end{cases} \quad 2.3$$

with $rand_i[0,1]$ and$rand_k[0,1]$ are random numbers$\in$ [0,1]..

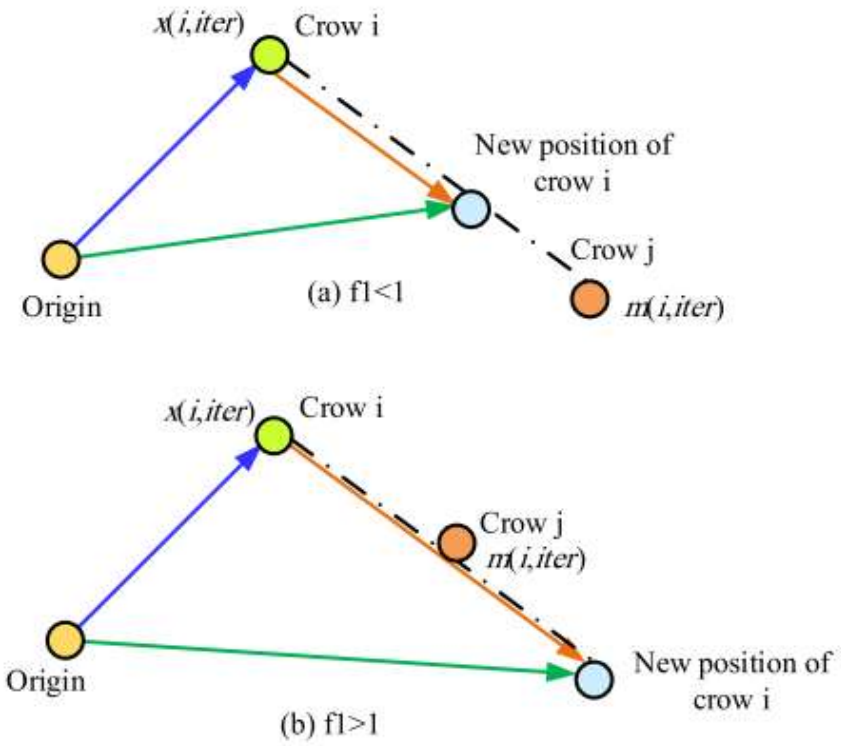

Figure2.2: Exploration and exploitation of CSA. (Hussien et al.,2020).

Step 5: Check the feasibility of new positions, The feasibility of the new position of each crow is checked. If the new position of a crow is feasible, the crow updates its position. Otherwise, the crow stays in the current position and does not move to the generated new position.

Step 6: Evaluate fitness function of new positions The fitness function value for the new position of each crow is computed. Step 7: Update memory the crows update their memory as follows:

$$m_{i,j,iter+1} = \begin{cases} x_{i,j,iter+1} & \text{si } f\left(x_{i,j,iter+1}\right) < f\left(m_{i,j,iter}\right) \\ m_{i,j,iter} & \text{sinon} \end{cases} \qquad 2.4$$

Step 8: Check termination criterion Steps 4–7 are repeated until itermax is reached. When the termination criterion is met, the best position of the memory in terms of the objective function value is reported as the solution of the optimization problem. (Askarzadeh,2016)

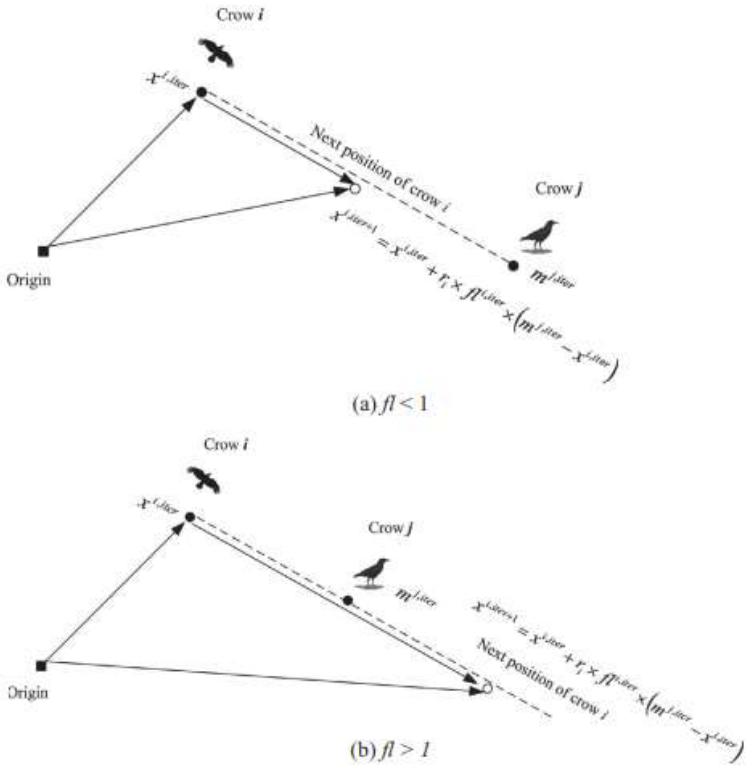

Fig 2.3: Flowchart of state 1 in CSA (a) fl< 1 and (b) fl> 1. Crow i can go to every position on the dash line. (Askarzadeh,2016)

Equation (2.4) shows that the crows only memorise the newly discovered positions if their qualities are better. The optimisation process stops when the stopping criterion is satisfied, in this study, after a maximum number of iterations noted $iter_{max}$. The position in the memory matrix with the best objective function value then corresponds to the optimal solution.

Figure 2.4 show the Pseudo code of CSA. (Askarzadeh,2016)

Crow search algorithm

Randomly initialize the position of a flock of N crows in the search space
Evaluate the position of the crows
Initialize the memory of each crow
while $iter < iter_{max}$
  for $i = 1 : N$ (*all N crows of the flock*)
    Randomly choose one of the crows to follow (*for example j*)
    Define an awareness probability
      if $r_j \geq AP^{j,iter}$
        $x^{i,iter+1} = x^{i,iter} + r_i \times fl^{i,iter} \times \left(m^{j,iter} - x^{i,iter}\right)$
      else
        $x^{i,iter+1} = a\ random\ \ position\ of\ \ search\ space$
      end if
  end for
Check the feasibility of new positions
Evaluate the new position of the crows
Update the memory of crows
end while

Figure 2.4: Pseudo code of CSA. (Askarzadeh,2016)

2.3 Rao-1 Algorithm

In recent years the field of population based meta-heuristic algorithms is flooded with a number of 'new' algorithms based on metaphor of some natural phenomena or behaviour of animals, fishes, insects, societies, cultures, planets, musical instruments, etc. Many new optimization algorithms are coming up every month. Some of these newly proposed algorithms are dying naturally as there are no takers and some have received success to some extent. However, this type of research may be considered as a threat and may not contribute to advance the field of optimization (Sorensen, 2015). It would be better if the researchers focus on developing simple optimization techniques that can provide effective solutions to the complex problems instead of looking for developing metaphor-based algorithms. Keeping this point in view. (Rao.,2020).

More recently, Rao (2020) presented three iterative optimization algorithms without needing any natural or artificial metaphors. These metaphor-less algorithms named Rao-1, Rao-2 and Rao-3 are population-based and have no algorithm-dependent parameters. In this work, we focus on the Rao-1 algorithm because of its simple working mode. The Rao-1 algorithm starts the optimization process from an initial random population. As shown in the previous subsection, we implemented the candidate solutions and their objective function values in the same matrix, so an additional column (N_{par} + 1)) was added to the population matrix. During an iteration iter, the search mechanism uses a random combination between the best and worst candidate solutions found among the population.

Rao-1 algorithm updates candidate solutions as follows:

$$x_{i,j,iter+1} = x_{i,j,iter} + rand_j[0,1]\left(x_{best,j,iter} - x_{worst,j,iter}\right) \qquad 2.4$$

where i=1, N; j=1, $Npar$; iter=1,1iter$_{max}$. $rand_j$ [0.1] is a random number $\in[0,1]$. $x_{best,j,iter}$

$x_{worst,j,iter}$ are the best and worst candidate solutions associated with the lowest and highest objective function values, respectively. Each updated solution must be compared to the previous one according to its objective function value; the one with the worst quality will be eliminated. The optimisation process stops after the $iter_{max}$ iterations.

Figure 2.5 shows the flowchart of the Rao-1 algorithm

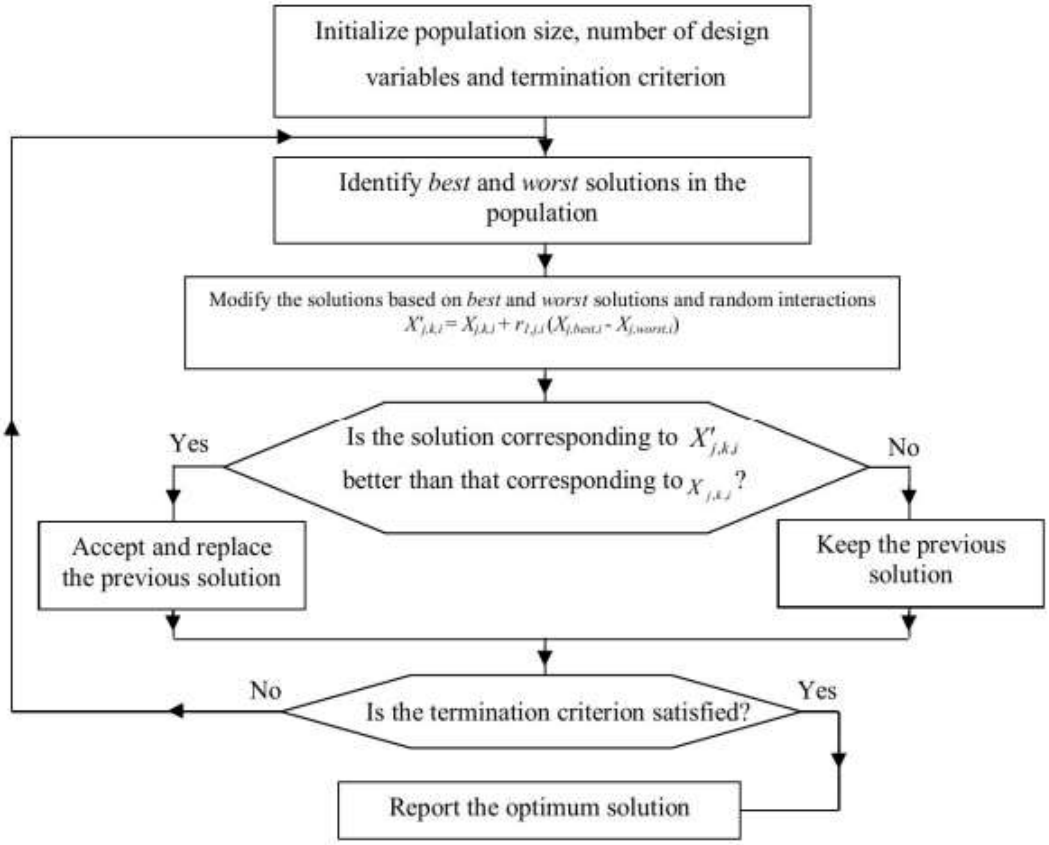

Figure 2.5: Flow chart of the Rao-1 algorithm (Rao,2020).

2.4 The hybrid algorithm CSARao-1

The reason behind the hybridisation of algorithms is to create an even more efficient algorithm in terms of accuracy, convergence speed and robustness. Metaheuristics have two main characteristics: exploration (diversification) and exploitation (intensification).

The new hybrid algorithm CSARao-1 can be regarded as an enhanced version of the original CSA. As mentioned earlier in the CSA subsection, the awareness probability AP dictates to the crows which behaviour to exhibit. When randk [0,1] \AP, then, the crow will move anywhere inside the search space, which may reduce the exploitation ability of the algorithm and delays its convergence. To overcome this drawback, we propose to replace the random explorative component of the CSA by the Rao-1 algorithm's search mechanism which is mainly exploitative than explorative (Suyanto et al. 2021). Here, instead of generating a random solution, the Rao-1 search mechanism exploits the knowledge available in the memory matrix which enhances the new hybrid algorithm's local search. The CSARao-1 hybrid algorithm reads

$$x_{i,j,iter+1} = \begin{cases} x_{i,j,iter} + rand_i[0,1] \times fl \times \left(m_{k,j,iter} - x_{i,j,iter}\right) & si\ rand_k[0,1] \geq AP \\ x_{i,j,iter} + rand_j[0,1] \times \left(m_{best,j,iter} - m_{worst,j,iter}\right) & si\ rand_k[0,1] < AP \end{cases} \quad 2.5$$

This hybridization aims to enhance the exploration and exploitation capabilities of the original CSA. It is important to mention that the Rao-1 algorithm's search mechanism uses quantities that are readily available in the CSA, which means that its running time is the same as that of the replaced randomization component

The CSARao-1 hybrid algorithm also has a philosophical dimension to Sörensen's view (Sörensens, 2015) that considers new nature-inspired algorithms (metaphor-based algorithms such as CSA) as a threat to the field of optimisation, and has invited researchers to focus on developing metaphor-free algorithms such as the Rao-1 algorithm. The new hybrid metaheuristic CSARao-1 reconciled a metaphor-based algorithm (CSA) and a metaphor-free one (Rao-1 algorithm) by making them work synergistically to achieve better performance.

. Figure 2.6 shows the proposed CSARao-1 hybrid algorithm pseudo-code.

Input: N, d, fl, AP, search space's bounds
Generate a random initial population of crows
Evaluate the objective function of each crow
Initialize the memory of each crow
DO $iter = 1, iter_{max}$
  DO $i = 1, N$
    Randomly select a crow $k \in [1, N]$ to follow
    Do j=1, d
      IF ($rand_k[0,1] \geq AP$) THEN
        $x_{i,j,iter+1} = x_{i,j,iter} + rand_i[0,1] \times fl \times (m_{k,j,iter} - x_{i,j,iter})$
      ELSE
        $x_{i,j,iter+1} = x_{i,j,iter} + rand_j[0,1] \times (m_{best,j,iter} - m_{worst,j,iter})$
      END IF
    END DO
  END DO
Verify the feasibility of the new candidate solutions
Evaluate the objective function of each candidate solution
Update the Memory matrix
END DO

Figure 2.6: FORTRAN style pseudocode of the CSARao-1 hybrid algorithm (Tadj, et al. 2021)

2.5 Conclusion

In this chapter, an introduction to metaheuristic algorithms has been presented, as well as three optimisation algorithms, namely: CSA, Rao-1 and CSARao-1and their main components, characteristics and properties and their flowsharts. In the next chapter, the presented algorithms will be adopted as optimization techniques to solve seven structural design problems.

Chapter Three

Application of metaheuristic algorithms to structural problems

1.1 Introduction

In optimization, the best solution for a set of design variables is found. The best solutions are decided according to a function that is related to the goal of the optimization. This function is called an objective function *f*(x), and is generally minimized or maximized for the optimum solutions, according to the problem. engineering problems generally also contain design constraints, which originate from physical rules, design regulations and architectural issues. in structural engineering, the design constraints are highly related to the design variables in optimization problems. For that reason, metaheuristic algorithms are effective in the problems of structural engineering. existence of design constraints is the main reason for using numerical optimization methods in structural engineering. The existence of design variables is the reason for the high nonlinearity of structural engineering problems. Several types of objectives used in the optimization of structural engineering problems are mentioned. These are weight, cost, size and topology. Minimization of the total weight of structural systems is one of the best-known objectives in structural engineering. There are several advantages to constructing a light building. The first advantage is saving on the material. A lighter building may have structural members with smaller cross-sections. Thus, this goal of structural engineering is achieved. Economy factor: The total cost of the structure decreases. Stress factor: By reduction of the weight of structural elements, the self-weight acting as the dead load of the structure is also reduced. In this case, the internal forces become lower and the stresses on the structural elements are reduced. Thus, structural members become safe and the failure of members may be prevented. The convergence of the metaheuristic algorithm is also an index of comparison and reliability. The results must converge to the near optimum results quickly without trapping a local optimum, and the best current results must continue to improve to find a precise optimum solution. (Yusuf Cengiz. et Gebrail B. et al 2021)

In this chapter, three recent metaheuristic algorithms called 'Crow Search Algorithm (CSA); Rao-1 algorithm with a hydride algorithm CSARao-1' is adopted as an optimization techniques and generalities of structural optimization problems, reliability concepts of metaheuristic algorithms and general types of structural optimization are discussed. The numerical applications of several constrained simple structural engineering design optimization problems are presented. These problems include the vertical deflection minimization problem of an I-beam, the cost optimization of a tubular column under compressive load and the weight optimization of cantilever beams and

another structural problem. to test the effectiveness of the proposed approach, several hypothetical cases will be analysed.

1.2 Numerical experiments

Structural optimization problems are complex, sometimes even the optimal solutions of interest do not exist. In order to see how the CSA, Rao-1 and CSARao-1 algorithms perform, seven standard structural engineering test problems are solved. The used algorithms were coded in FORTRAN programming language and executed in Intel(R) Core (TM) i3- 6006U CPU @ 2.0 GHz, 4 GB of RAM. We used the Mersenne Twister (Matsumoto and Nishimura 1998) as a pseudorandom number generator for its very long period and its relevant statistical qualities. The flight length fl and the awareness probability AP were set equal to 2 and 0.1, respectively. The maximum number of iterations varies between 200 and 500 depending on the optimization problem. The population size N was set equal to 50. The objective function values, standard deviation and speed (rate) of convergence were taken as quality metrics evaluate the performances of metaheuristic algorithms when solving a given optimization problem. Because of the stochastic nature of the employed algorithms, each problem was independently executed 30 times. At each run, a given metaheuristic algorithm takes different pathway towards the global optimum depending on the generated sequence of random numbers. An algorithm is said to be robust if it converges towards the global optimum at each run, so the standard deviation (SD) that measures the spread of the obtained thirty solutions should be as small as possible (the smaller the standard deviation, the better the robustness and stability of the algorithm). In other words, robustness is the ability of the algorithm to obtain optimal solutions. Admittedly, the performance of metaheuristic algorithms can be affected by their initial populations and this traces back to the fact that some high-quality candidate solutions may occur by chance in the initial population since they are randomly generated. Therefore, in order to avoid the possibility of an occurring bias in the initial population and to give the algorithms equal chances at the beginning, we controlled the pseudo-random number generator incorporated in each algorithm by the same seed numbers. This will ensure a fair comparison, seeing that the employed metaheuristic algorithms will be supplied with the same sequences of random numbers. The results are presented in terms of mean values, and the best are in **bold**. The convergence curves as common tools to analyze the efficiency of metaheuristic algorithms were also provided. They can be visually inspected to have a clear insight about how fast the best solution converges towards the global optimum. The speed of convergence is directly related to the balance between exploration and exploitation, the good algorithm is the one who manages to find the global

solution in a minimum number of iterations. The analyzed constrained problems were converted into unconstrained ones with the addition of a penalty term to the objective function. The penalized objective function can be expressed in the following general form:

$$F(x) = f(x) + \sum_{i=1}^{n} \lambda_i \cdot g_i(x) \quad 3.1$$

where $F(x)$ is the penalized objective function (also called fitness function), $g_j(x)$ are the inequality constraints. λ_i are positive factors in this study, λ_i were set equal to 10^{15}.

1.2.1 Himmelblau's constrained function

Before solving the structural engineering problems, the algorithms were benchmarked using a well-known problem, namely, Himmelblau's problem. This problem has originally been proposed by Himmelblau's and it has been widely used as a benchmark nonlinear constrained optimization problem. In this problem, there are five design variables [x_1, x_2, x_3, x_4, x_5], six nonlinear inequality constraints, and ten boundary conditions. The problem can be stated as follows:

Consider $x = [x_1, x_2, x_3, x_4, x_5]$

Minimize $f(x) = 5.3578547x_3^2 + 0.8356891x_1x_5 + 37.293239x_1 - 40792.141$

Subject to:

$g_1(x) = 85.334407 + 0.0056858x_2x_5 + 0.0006262x_1x_5 - 0.0022053x_3x_5 - 92.$

$g_2(x) = 80.51249 + 0.0071317\,x_2x_5 + 0.0029955x_1x_5 + 0.0021813x_3^2 - 110$

$g_3(x) = 9.300961 + 0.0047026x_3x_5 + 0.0012547x_1x_3 + 0.0019085x_3 + 0.0019085\,x_3x_4 - 25.$

$g_4(x) = -(85.334407 + 0.0056858\,x_2x_5 + 0.0006262\,x_1x_4 - 0.0022053\,x_3x_5)$

$g_5(x) = -(80.51249 + 0.0071317x_2x_5 + 0.0029955\,x_1x_2 + 0.0021813x_3^2) + 90$

$g_6(x) = -(9.300961 + 0.0047026\,x_3x_5 + 0.0012547x_1x_3 + 0.0019085\,x_1x_3) + 20$

Variable range: $78 \leq x1 \leq 102;\ 33 \leq x2 \leq 45;\ and\ 27 \leq x3;\ x4;\ x5 \leq 45$

Table 3.1 presents the obtained results of the Himmelblau's function using the three employed algorithms. The corresponding constrains values are presented in table 3.2. Figure 3.1 shows the convergence curve of Himmelblau's function.

Table 3.1: Results for the Himmelblau's function

Techniques	x_1	x_2	x_3	x_4	x_5	f(x)	SD
CSA	78.00015	33.00005	29.99543	44.99986	36.77558	-30665.48891	0.0628
Rao-1	78.00000	33.00078	29.99780	45.00000	36.77130	-30665.01529	0.069
CSARao-1	**78.00000**	**33.00000**	**29.99525**	**45.00000**	**36.77583**	**-30665.53860**	**0.0014**

Table 3.2: Himmelblau's function corresponding constraints

Contraints	CSA	Rao-1	CSARao-1
g_1(x)	1.3274E-06	-1.1331E-04	-2.349E-07
g_2(x)	-11.15954	-11.15971	-11.159497
g_3(x)	-4.942823	-4.942885	-4.942753
g_4(x)	-92.0000013	-91.99986	-91.99999
g_5(x)	-8.840461	-8.840283	-8.840502
g_6(x)	6.8549E-05	1.31140E-04	-1.521830

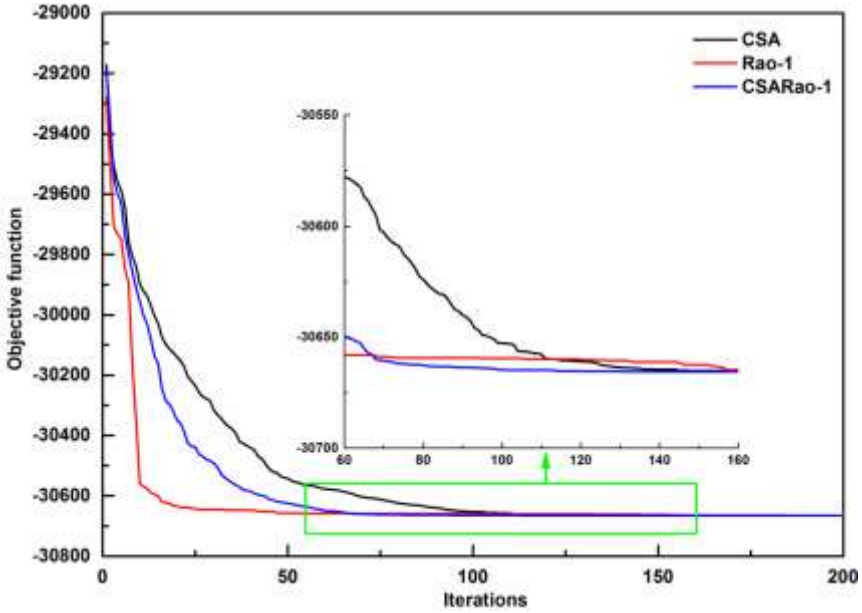

Figure 3.1: Convergence curve of Himmelblau's function

1.2.2 Three-bar truss design problem

The objective of this problem is to minimize the volume of a statistically loaded three-bar truss subject to stress (r) constraints on each of the truss members by adjusting cross sectional areas ($x_1$and x_2). Fig. 1.3 represents the schematic of three-bar truss design problem. This optimization problem has a nonlinear fitness function with three nonlinear inequality constraints and two continuous decision variables as follows:

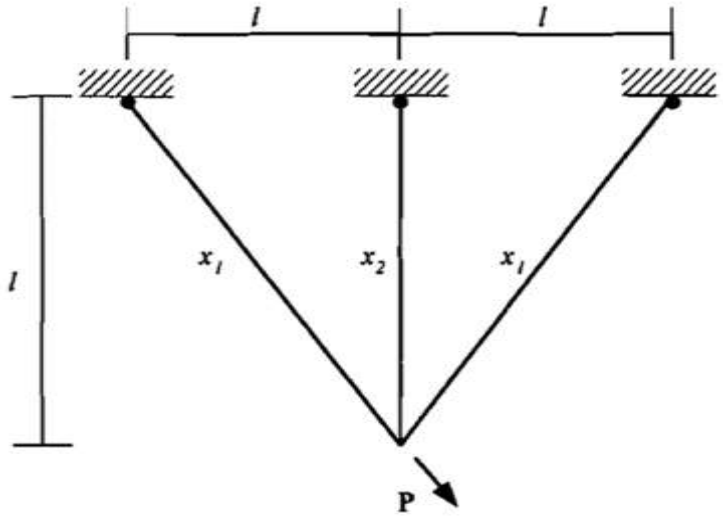

Figure 3.2: Three-bar truss design problem. (Askarzadeh,2016)

The problem can be stated as follows:

Consider$x = [x_1 x_2]\,; l = 100cm, P = 2KN/cm^2, \sigma = 2KN/cm^2$

Minimize $f(x) = \left(2\sqrt{2}x_1 + x_2\right) \times l$

Subject to:

$$g_1(x) = \frac{\sqrt{2}x_1 + x_2}{\sqrt{2}x_1^2 + 2x_1x_2} P - \sigma \leq 0$$

$$g_2(x) = \frac{x_2}{\sqrt{2}x_1^2 + 2x_1x_2} P - \sigma \leq 0$$

$$g_3(x) = \frac{1}{\sqrt{2}x_2 + x_1} P - \sigma \leq 0$$

Variable range: $0.0 \leq x_1, x_2 \leq 1.0$

The performance of each of the algorithms is summarized in Table 3.3 and 3.4

Table 3.3 Results for the 3-barrs stress problem

Techniques	x_1	x_2	f(x)	SD
CSA	0.78868	0.40825	263.895839	3.8862E-09
Rao-1	0.78890	0.40762	263.897080	0.000974
CSARao-1	**0.78868**	**0.40825**	**263.895839**	**5.7815E-14**

Table 3.4: The 3-barrs stress problem corresponding constraints

Contraints	CSA	Rao-1	CSARao-1
g_1(x)	-1.1769792E-05	-5.631608E-06	-1.176979E-05
g_2(x)	-1.464105544	-1.46481886	-1.464105544
g_3(x)	-0.535906204	-0.53518674	-0.535906204

Figure 3.3 shows the convergence curve of Three-bar truss problem

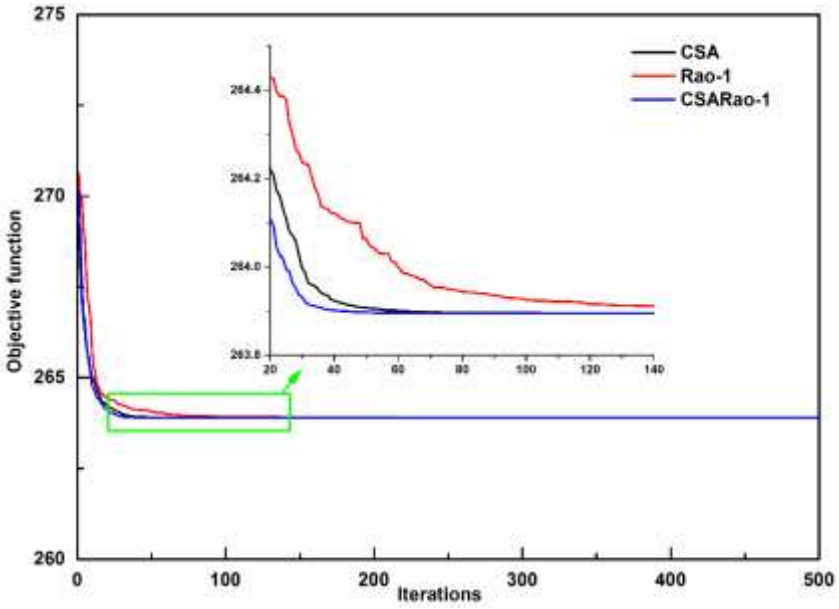

Figure 3.3: the convergence curve of Three-bar truss problem

1.2.3 5bar truss structure optimization problem

This application analyses a theoretical 5-bar truss structure with a fixed base as shown in figure 3.2 (a).

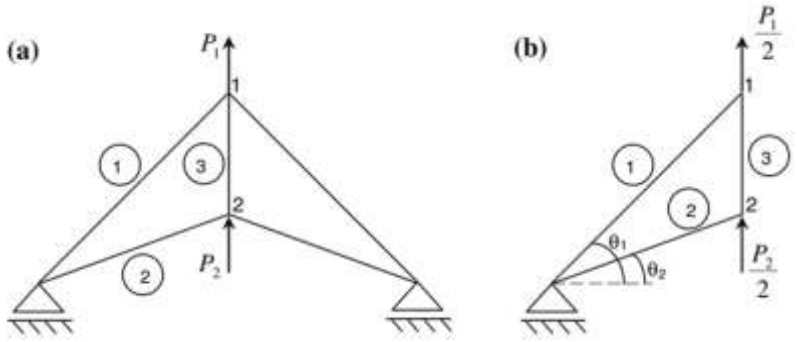

Figure 3.4: 5-bar truss structure (Gandomi et al. 2013)

All members of the truss structure have the same cross-sectional area A equal to 100 mm^2; the vertical forces P_1 and P_2 are, respectively, 100 and 50 kN, and **the length of the base *l* is 1,000** mm. The modulus of elasticity *E* is 200000 MPa. The objective of this optimization problem is to find the structure with the optimum values of angles $\theta_1$and θ_2, which are the angles between the horizontal axis and bars 1 and 2, respectively. Due to symmetry only half the problem needs to be analyzed. Thus, topology optimization is done to find the optimum values of θ_1 and θ_2 angles that minimize the weight of a three members structure as shown in figure x (b). Member *l*3 is modeled as having one-half of its actual cross-sectional area. Nodes 1 and 2 move vertically, and the node displacement vector is$\Delta = (\Delta_1, \Delta_2)^{T.}$ The vertical deflections at nodes 1 and 2 are limited to 5 mm. The lengths of the bars *l*1, *l*2, *l*3 can be calculated as follows:

$$l_1 = \frac{l}{2\cos(\theta_1)} \quad 3.2$$

$$l_2 = \frac{l}{2\cos(\theta_2)} \quad 3.3$$

$$l_3 = \frac{l}{2\cos(\theta_1)\cos(\theta_2)}\sqrt{cos^2(\theta_1) + cos^2(\theta_2) - 2\cos(\theta_1)\cos(\theta_2)\cos(\theta_1 - \theta_2)} \quad 3.4$$

The displacements $\Delta = (\Delta_1, \Delta_2)^T$ are calculated from the following linear system obtained from the finite element formulation:

$$EA\begin{bmatrix} \frac{sin^2(\theta_1)}{l_1} + \frac{1}{2l_3} & -\frac{1}{2l_3} \\ -\frac{1}{2l_3} & \frac{sin^2(\theta_2)}{l_2} + \frac{1}{2l_3} \end{bmatrix}\begin{bmatrix} \Delta_1 \\ \Delta_2 \end{bmatrix} = \begin{bmatrix} \frac{P_1}{2} \\ \frac{P_2}{2} \end{bmatrix} \quad 3.5$$

The optimization problem can be expressed as:

Consider$x = [x_1 x_2] = [\theta_1 \theta_2]$

Minimize $f(x) = \sum_{i=1}^{3} l_i$

Subject to:

$g_1(x) = |\Delta_1(\theta_1, \theta_2)| \leq 5\ mm$

$g_2(x) = |\Delta_2(\theta_1, \theta_2)| \leq 5\ mm$

Variable range: $0.0 \leq x_1, x_2 \leq \pi/3\ rad$

Table 3.5: Results for the 5-bar truss problem

Techniques	x_1	x_2	f(x)	SD
CSA	0.4773843	0.4773843	1125.87290	0.000057
Rao-1	0.4773842	0.477384	1125.87282	0.0000015
CSARao-1	**0.4773842**	**0.477738**	**1125.87282**	**1.6042E-11**

As seen in the optimum results (Table 3.5), θ_1 is nearly equal to θ_2. Since the objective function is the minimization of the total length, the length l_3 of the third member is nearly zero. This result was expected and therefore, the employed algorithms have effectively found the global optimum values. The optimized structure consists of two members as shown in figure 3.5. The constraints are respected since the displacement of node 2 equals to 9.98 mm which is less than twice 5mm.

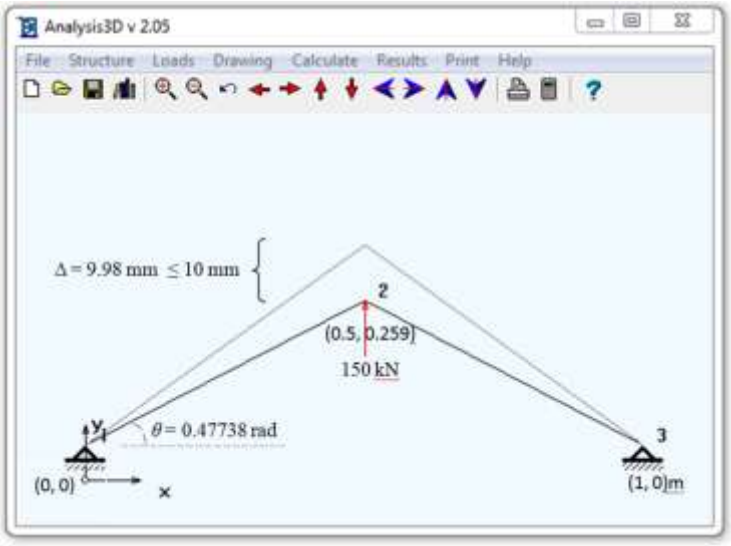

Figure3.5: Optimized 5-bar truss structure

Figure 3.6 shows the convergence curve of five-bar truss problem

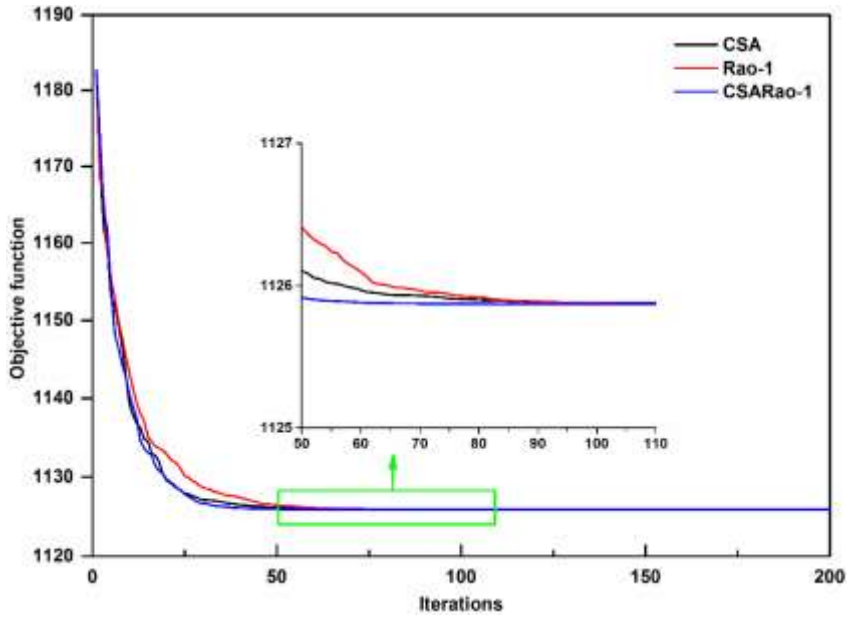

Figure 3.6: shows the convergence curve of five-bar truss problem

1.2.4 Cantilever beam design problem

It is a structural engineering design example that is related to the weight optimization of a cantilever beam with a square cross-section. The beam is rigidly supported at node 1, and there is a given vertical force acting at node 6 (Fig.). The beam consists of five variables: heights (or widths) of the different beam elements and the thickness is held fixed (here t = 2/3).

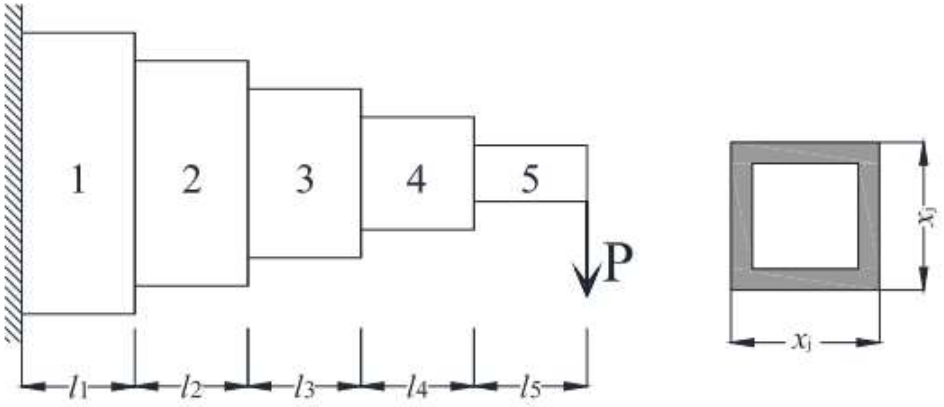

Fig 3.7: Cantilever beam design problem. (Yusuf C.et al,2021)

Consider$x = [x_1 x_2 x_3 x_4 x_5]$

Minimize $f(x) = 0.0624(x_1 + x_2 + x_3 + x_4 + x_5)$

Subject to $g_1(x) = \frac{61}{{x_1}^3} + \frac{37}{{x_1}^3} + \frac{19}{{x_3}^3} + \frac{7}{{x_4}^3} + \frac{1}{{x_5}^3} - 1 \leq 0.$

$0.01 \leq x_i \leq 100; \quad i \in 1, \ldots\ldots, 5.$

Table 3.6: Results for the Cantilever beam design problem

Techniques	x_1	x_2	x_3	x_4	x_5	f(x)	SD
CSA	6.022482	5.33751	4.46796	3.502687	2.16418	1.341423	0.0049
Rao-1	6.02008	5.30661	4.49176	3.508884	2.14095	1.341567	0.00078
CSARao-1	**6.01695**	**5.30993**	**4.49407**	**3.501664**	**2.15272**	**1.33995**	**0.00034**

Table 3.7: The Cantilever beam design problem corresponding constraints

Contraints	CSA	Rao-1	CSARao-1
$g_1(x)$	-2.8519558E-03	-2.4744685E-03	**-2.3719628E-04**

Figure 3.6 shows the convergence curve of the Cantilever beam design problem

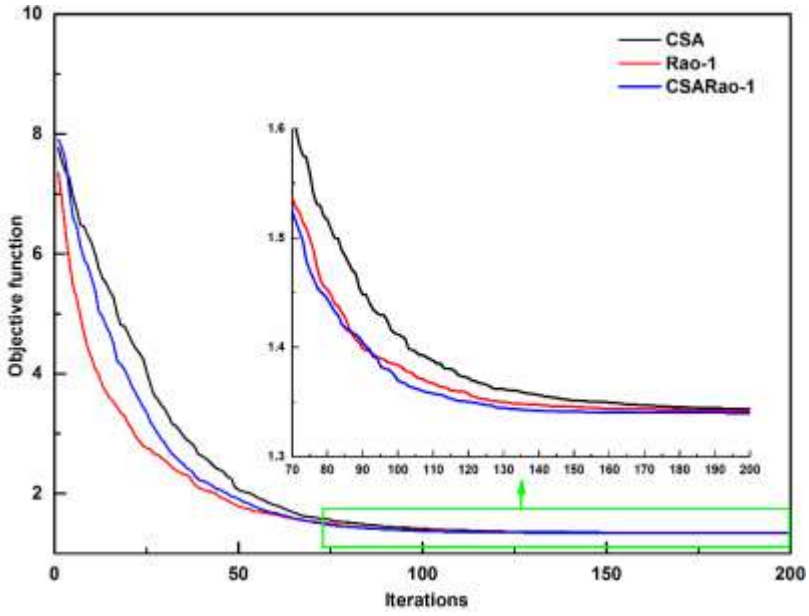

Figure 3.8: the convergence curve of the Cantilever beam design problem

1.2.5 Vertical deflection minimization problem of an I-beam

The minimization of the vertical deflection of an optimum I-beam problem was first presented by Gold and Krishnamurty (1997). The optimization objective is to find the dimensions of the I-beam. The shape and topology are fixed. In this case, it is a sizing optimization problem. Four of the design variables are the flange width (b), the flange thickness (tf), the beam height (h) and the web thickness (tw), as shown in Figure 3.4.

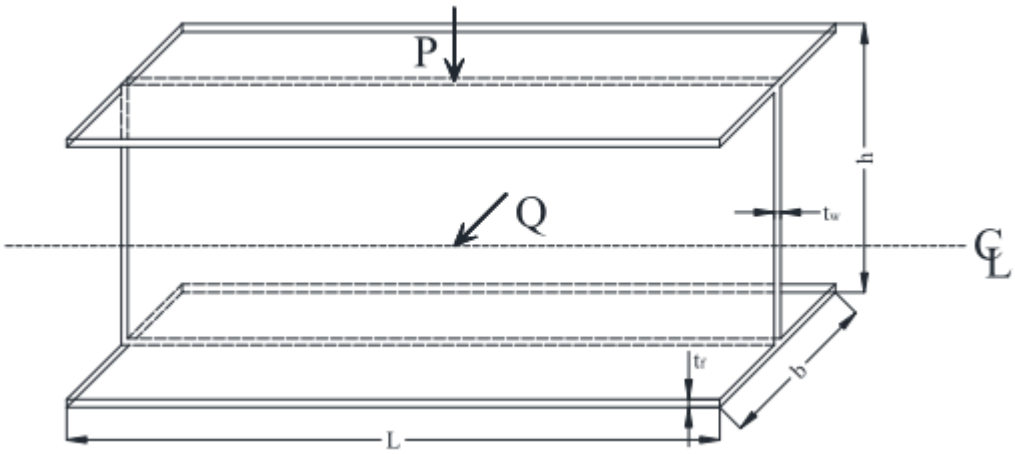

Fig 3.9: I-beam design problem (Yusuf C.et al,2021)

The vertical deflection of the I-beam results from two design loads (P and Q) applied at midspan, vertically and horizontally. These loads are 600 kN and 50 kN for P and Q, respectively. The length of the beam (L) and the modulus of elasticity (E) are the design constants, which are taken as 200 cm and 20,000 kN/cm2, respectively. The vertical deflection of the I-beam subjected to P load can be formulized as follows:

$$f(x) = \frac{PL^3}{48EI} \qquad 3.6$$

The objective function of the vertical deflection minimization problem (f (b, h, tw, tf)) is given in equation. The numerical values of the design constants are applied to the parametric definition of the moment of inertia of the I-beam (I)

The optimum design problem has two inequality constraints, denoted by g1 and g2. The cross-section of the I-beam must be less than 300 cm2, which is the first design constraint. Second, the bending stress of the beam must be less than 6 kN/cm2.

The optimization problem can be expressed as:

Consider $x = [x1x2x3x4] = [b, h, t_w, t_f]$

Minimize $f(x) = \frac{5000}{\frac{x_3(x_2-2x_4)^3}{12} + \frac{x_1 x_4^3}{6} + 2x_1 x_4 (\frac{x_2-x_4}{2})^2}$

Subject to:

$g_1(x) = 2x_1x_4 + x_3(x_2 - 2x_4) - 300$

$g_2(x) = \frac{1800x_2}{x_3(x_2-2x_4)^3 + 2x_1x_3(4x_4^2 + 3x_2(x_2-2x_3))} + \frac{15000x_1}{x_3^3(x_2-2x_4) + 2x_3x_1^3} - 6$

$10 \le x_1 \le 80$

$10 \le x_2 \le 50$

$0.9 \le x_3, x_4 \le$

Table 3.8: Results for the I-beam design problem

Techniques	x_1	x_2	x_3	x_4	f(x)	SD
CSA	80	50	0.89999	2.32179	0.013074	5.097E-12
Rao-1	80	50	0.89999	2.32179	0.013074	6.0107E-12
CSARao-1	**80**	**50**	**0.89999**	**2.32179**	**0.013074**	**1.7643E-18**

Table 3.9: the I-beam design problem corresponding constraints

Contraints	CSA	Rao-1	CSARao-1
$g_1(x)$	-6.82815E-03	-6.82815E-03	-6.82815E-03
$g_2(x)$	**-0.794897**	**-0.794897**	**-0.794897**

Figure 3.10: shows the convergence curve of the I-beam design problem

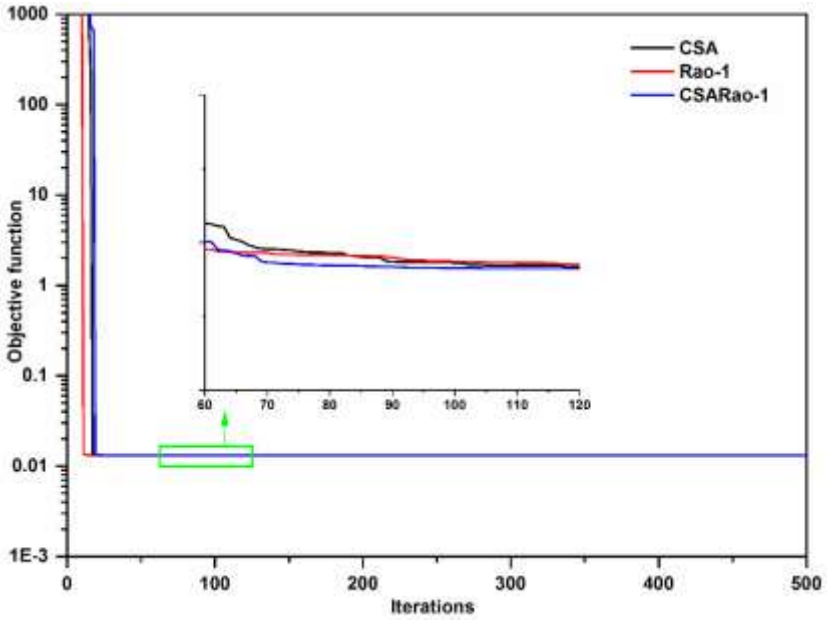

Figure 3.10: The convergence curve of the I-beam design problem

1.2.6 Bending beam-column

This problem consists on sizing the cross-section dimensions of a rectangular steel beam-column required to carry an axial load of 25 lb and a transverse load of 10 lb, as shown in Figure 3.6.

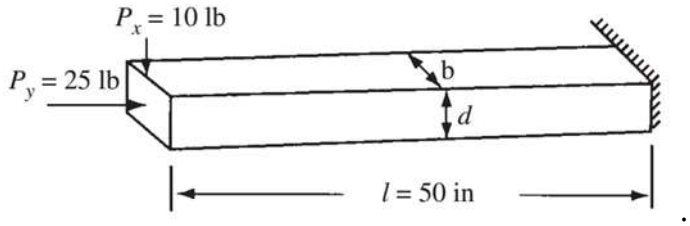

Fig3.11: Beam-column (Rao 2020)

This Beam-column should be designed for minimum weight while avoiding the possibility of yielding and buckling. It must bend only in the vertical (*xy*) plane; moreover, the width *b* of the beam-column is required to be at least 0.5 in, and not greater than twice the depth *d*. The beam-column has the following characteristics:

- Young's modulus: E= 30 × 10^6 psi;
- Specific weight; ρ =0.3 lb/in^3;
- Yield stress. σ_y= 30 10^3psi
- length$l$=50 in

The compressive stress in the beam-column due to $P_y$is $P_y/(bd)$, and that due to $P_x$is:

$$\frac{P_x l d}{2I_{zz}} = \frac{6P_x l}{bd^2} \qquad 3.7$$

The axial buckling load is given by:

$$(P_y)_{\text{cri}} = \frac{\pi^2 E I_{zz}}{4l^2} = \frac{\pi^2 E b d^3}{48l^2} \qquad 3.8$$

The optimization problem can be expressed as:

Consider$x = [x_1 x_2] = [db]$

Minimize $f(x) = \rho l x_1 x_2$

Subject to:

$$g_1(x) = \frac{P_y}{x_2 {x_1}^2} + \frac{6P_x l}{x_2 {x_1}^2} \leq \sigma_y$$

$$g_2(x) = \frac{P_y}{x_2 {x_1}^2} + \frac{6P_x l}{x_2 {x_1}^2} \leq (P_y)_{\text{cri}}$$

$$g_3(x) = -x_2 + 0.5 \leq 0$$

$$g_4(x) = x_2 - 2x_1 \leq 0$$

$$0.0 \leq x_1, x_2 \leq 50.$$

Table 3.10: Results for the Beam-column problem

Techniques	x_1	x_2	f(x)	SD
CSA	0.250006	0.500002	1.875066	5.566E-05
Rao-1	0.250000	0.500000	1.875000	5.937E-08
CSARao-1	**0.250000**	**0.500000**	**1.875000**	**1.021E-09**

Table 3.11: The Beam-column corresponding constraints

Contraints	CSA	Rao-1	CSARao-1
$g_1(x)$	-126194.7184472	-126200.0	**-126200.0**
$g_2(x)$	-96348.9386773	-96354.21	**-96354.21**
$g_3(x)$	-2.682209015E-06	0.00	**0.00**
$g_4(x)$	-9.71555709E-06	0.00	**0.00**

Figure 3.11: shows the convergence curve of the Beam-column design problem

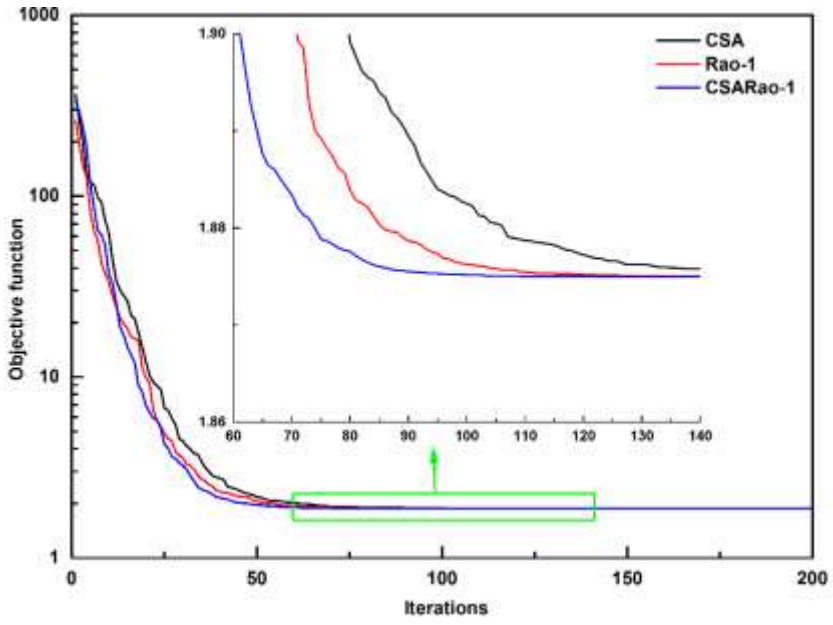

Figure 3.12: the convergence curve of the Beam-column design problem

1.2.7 Tubular column design

Figure 3.5 presents an example for designing a uniform column of tubular section to carry a compressive load P = 2,500 kgf at minimum cost [30]. The column is made of a material with a yield stress (ry) of 500 kgf/cm2, a modulus of elasticity (E) of 0.85 9 106 kgf/cm2, and a density (q) equal to 0.0025 kgf/cm3. The length (L) of the column is 250 cm. The stress included in the column should be less than the buckling stress (constraint g1) and the yield stress (constraint g2). The mean diameter of the column is restricted between 2 and 14 cm (constraint g3 and g4), and columns with thickness outside the range 0.2– 0.8 cm are not commercially available (constraint g5 and g6). The cost of the column includes material and construction costs. It is taken as the objective function. (Hossein.et al,2013) •

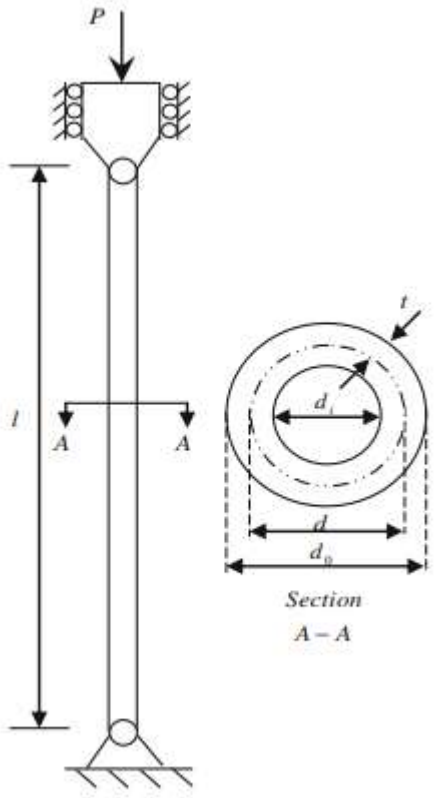

Fig 3.13: Tubular column design. (Hossein.et al,2013)

The optimization model of this problem is given as follows:

Consider$x = [x_1 x_2] = [dt]$

Minimize :$f(x_1 x_2) = 9.8 x_1 x_2 + 2\, x_2$

Subject to:

$g_1(x) = \frac{P}{\pi x_1 x_2 \sigma y} - 1 \leq 0$

$g_2(x) = \frac{8PL2}{\pi 3E x_1 x_2 (x_1\, 2 + x_2 2)} - 1 \leq 0$

$$g_3(x) = \frac{2}{x_1} - 1 \leq 0$$

$$g_4(x) = \frac{x_1}{14} - 1 \leq 0$$

$$g_5(x) = \frac{0.2}{x_2} - 1 \leq 0$$

$$g_6(x) = \frac{x_2}{0.8} - 1 \leq 0$$

Table 3.11: Results for the Tubular column problem

Techniques	x_1	x_2	f(x)	SD
CSA	5.451156	0.29196	26.531329	9.62444E-6
Rao-1	5.451156	0.29196	26.531329	7.16 E-6
CSARao-1	**5.451156**	**0.29196**	**26.531326**	**1.225.E-13**

Table 3.12: Tubular column design corresponding constraints

Constraints	CSA	Rao-1	CSARao-1
$g_1(x)$	-6.90503E-005	-6.17501E-005	-1.80129E-005
$g_2(x)$	-6.75006E-005	-1.47132E-004	-1.61712E-005
$g_3(x)$	-3.45116	-3.45116	-3.45116
$g_4(x)$	-8.54884	-8.54884	-8.54884
$g_5(x)$	-9.19655E-002	-9.19655E-002	-9.19655E-002
$g_6(x)$	-0.50803	-0.508034	-0.50803

Figure 3.13: shows the convergence curve of theTubular column design.

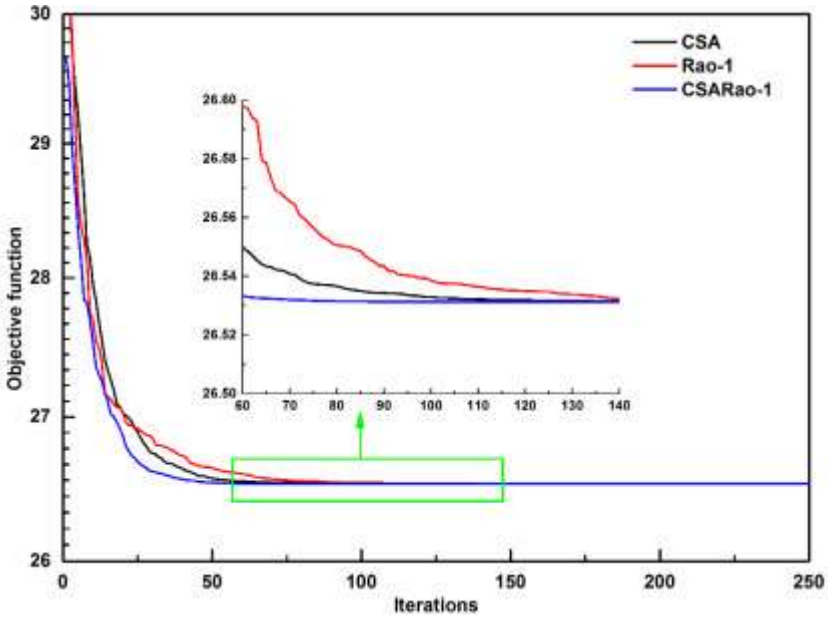

Figure 3.14: The convergence curve of theTubular column design.

1.2.8 Collapse mechanism problem

In the limit design of steel frames, it is assumed that plastic hinges will be developed at points with peak moments. When a sufficient number of hinges develop, the structure becomes an unstable system referred to as a collapse mechanism. Thus, a design will be safe if the energy-absorbing capacity of the frame (U) is greater than the energy imparted by the externally applied loads (E) in each of the deformed shapes as indicated by the various collapse mechanisms. As shown in figure 3.6 (Rao.2020)

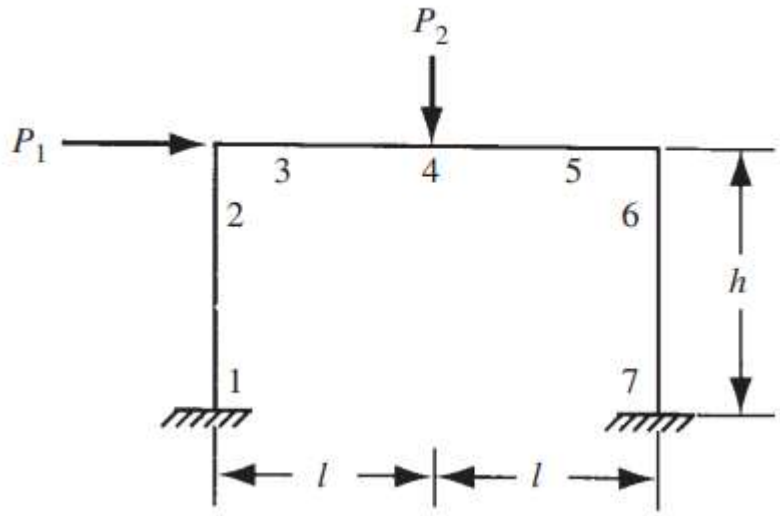

Figure 3.15: Collapse mechanism (Rao.2020)

For the rigid frame shown in Figure 3.15, plastic moments may develop at the points of peak moments (numbered 1 through 7 in Figure 3.15). Assume that the two columns are identical and that: P1 = 3, P2 = 1, h = 8, and l = 10.

The objective function can be expressed as

f (Mb, Mc) = weight of beam + weight of columns

$$f\ (Mb, Mc)\ =\ \alpha\ (2l\ Mb\ +\ 2h\ Mc) \qquad 3.9$$

where α is a constant indicating the weight per unit length of the member with a unit plastic moment capacity. Since a constant multiplication factor does not affect the result, f can be taken as:

$$f\ =\ 2l\ Mb\ +\ 2h\ Mc\ =\ 20Mb\ +\ 16Mc \qquad 3.10$$

The constraints (U ≥E) from the four collapse mechanisms can be expressed as

Mc ≥ 6

Mb ≥ 2.5

$$2Mb\ +\ Mc\ \geq\ 17 \qquad 3.11$$

The optimization model of this problem is given as follows:

Consider$x = [x_1 x_2] = [Mb. Mc]$

Minimize : $f(x_1 x_2) = 20x_1 + 16\, x_2$

Subject to:

g_1(x)= $-x_2$+ 6.

g_2(x)=$-x_1$ + 2.5

g_3(x)= $-2x_1 - x_2$+17

$0.0 \leq x_1, x_2 \leq 100$

Under the principle of virtual work which is a particularly effective way of translating the fundamental law of static equilibrium: for any virtual displacement, the sum of the work of external forces and internal forces is identically zero. three possible collapse mechanisms are shown in Figure 3.7 for this frame. Assuming that the weight is a linear function of the plastic moment capacities, find the values of the ultimate moment capacities Mb and Mc for minimum weight.

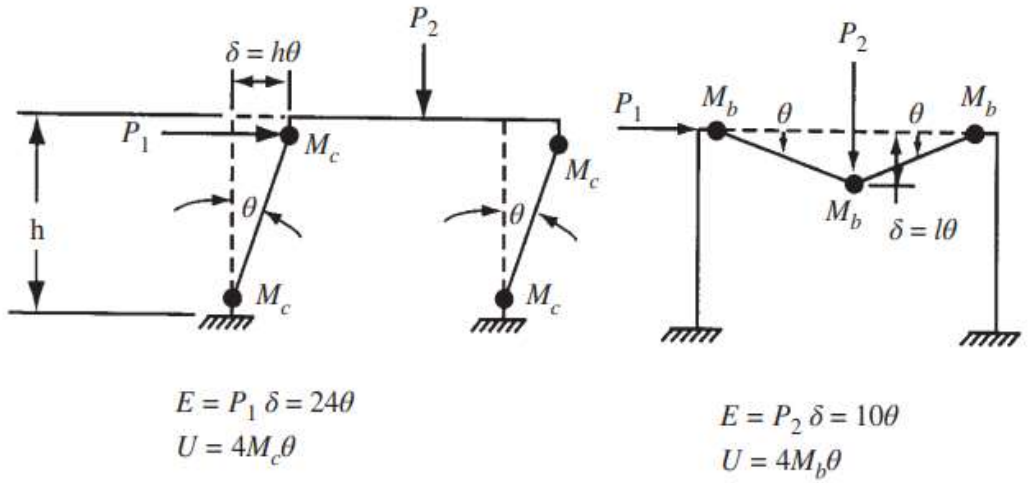

Figure 3.16: The sway and the beam mechanisms (Rao.2020)

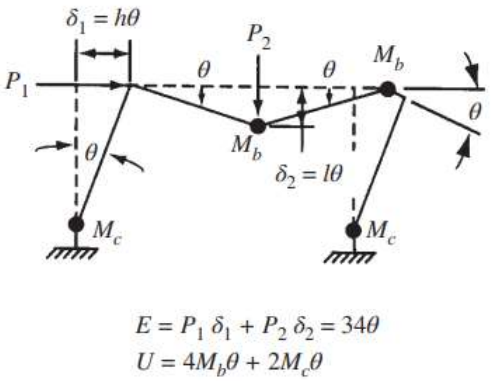

Figure 3.17: The combined mechanism (Rao.2020)

Collapse mechanisms of the frame. Mb, moment carrying capacity of beam; Mc, moment carrying capacity of column.

Table 3.13: Results for the collapse mechanism design problem

Techniques	x_1	x_2	f(x)	SD
CSA	5.49998	6.00005	206.00066	0.00067
Rao-1	5.49983	6.00006	206.000063	0.00063
CSARao-1	**5.5**	**6.0**	**206**	**4.01E-09**

Table 3.14: the collapse problem corresponding constraints

Contraints	CSA	Rao-1	CSARao-1
$g_1(x)$	-5.769729	-6.38962E-05	0.00
$g_2(x)$	-2.999987	-2.9998300	-3.00
$g_3(x)$	-3.194808	2.6417E-04	0.00

Figure 3.17: shows the convergence curve of the Collapse problem.

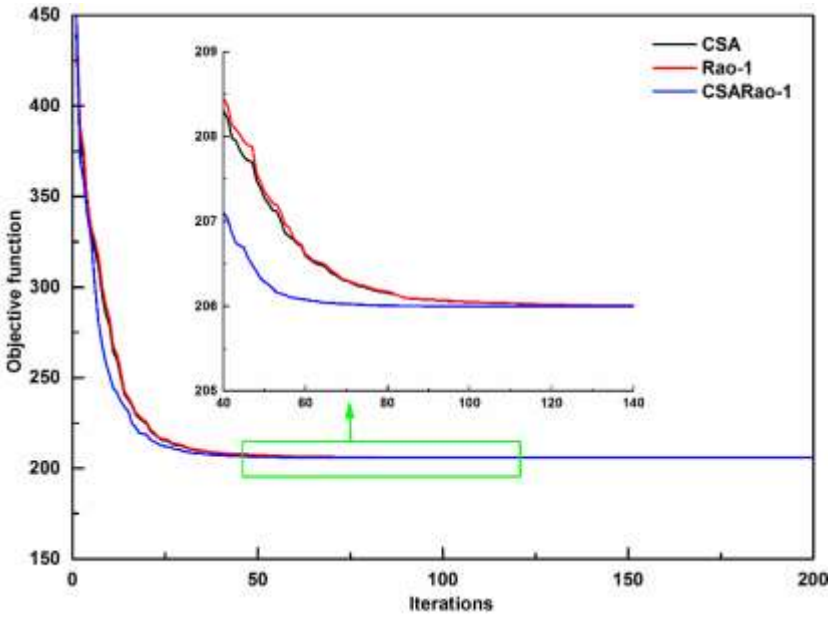

Figure 3.18: The convergence curve of the Collapse problem.

As shown on the convergence curves of each one of the analyzed problems we can summarize that The CSARao-1 hybrid algorithm converges first. Followed by the CSA and Rao-1 algorithm. Regarding the convergence speed, the CSARao-1 algorithm converged faster towards the global optimum than CSA and Rao-1 algorithm. in terms of robustness and the rate of converge and best SD value CSA and CSARao-1 algorithms provided the best optimum solution.

1.3 Conclusion

In this chapter, some structural designs problems were solved using three recent stochastic algorithms to minimize a given objective function under constraints, to achieve the best optimum design. The performance of the employed algorithms was assessed in terms of accuracy, robustness and speed of convergence. Note that the effect of the initial population bias was avoided, and all investigated algorithms started their search from the same conditions. The coupling between structural problems with metaheuristics has shown its effectiveness in this work.

General Conclusion

With the advances observed in the speed and memory capacity of computers, and also in the theoretical and practical knowledge about coding and metaheuristics, it seems like there will be much progress in the application of metaheuristics to structural design and analysis on civil engineering.

The main problem is to make the design of a civil engineering structures the most optimal possible using metaheuristic algorithms.

In this study, we have tried to take advantage of the benefits offered by artificial intelligence techniques, in this case, metaheuristics for the solution of structural problems in civil engineering field.

Three-metaheuristic algorithms were used: the first one called Crow search algorithm based on the cleverness of crows, second one is a metaphor-less algorithm called Rao-1 and the third is a hybrid algorithm CSARao-1 algorithm, this hybrid algorithm was made by replacing the random explorative component of the crow search algorithm with the recent search mechanism of the Rao-1 algorithm. These metaheuristics were then used to solve several cases of structural problems. The performance of the employed algorithms was assessed in terms of accuracy, robustness and speed of convergence. Note that the effect of the initial population bias was avoided, and all investigated algorithms started their search from the same conditions.

The codes that we used in Fortran 95 programming language, allowed us to successfully identify the design variables related to the treated problems.

Finally, the application of the metaheuristics can be extended to the real-world problems in the field of civil engineering.

References

References

Asef, F., Majidnezhad, V., Feizi-Derakhshi, M. R., & Parsa, S. (2021). Heat transfer relation-based optimization algorithm (HTOA). Soft Computing, 1-3

Askarzadeh A (2016) A novel metaheuristic method for solving constrained engineering optimization problems: crow search algorithm. Comput Struct 169:1–12

Brownlee, J. (2011). Clever algorithms: nature-inspired programming recipes. Jason Brownlee.

Bruyneel, Michael_ Craveur, Jean-Charles_ Gourmelen, Pierre - Optimisation des structures mécaniques. -Dunod (2014)

Dhiman, G., & Kumar, V. (2018). Emperor penguin optimizer: a bio-inspired algorithm for engineering problems. Knowledge-Based Systems, 159, 20-50.

Fogel, L. J., Owens, A. J., & Walsh, M. J. (1966). Artificial intelligence through simulated evolution.

Galinier, P., Hamiez, J. P., Hao, J. K., & Porumbel, D. (2013). Handbook of optimization

Goel, L. (2020). An extensive review of computational intelligence-based optimization algorithms: trends and applications. Soft Computing, 24, 16519-16549.

Holland, J.H., 1975. Adaptation in natural and artificial system: an introduction with application to biology, control and artificial intelligence, Ann Arbor, University of Michigan Press

Hossein, A.G., Yang.X. S, Hossein, A.A. (2013). Cuckoo search algorithm: a metaheuristic approach to solve structural optimization problems. Engineering with Computers 29:17–35

Hussien, A. G., Amin, M., Wang, M., Liang, G., Alsanad, A., Gumaei, A., & Chen, H. (2020). Crow search algorithm: theory, recent advances, and applications. IEEE Access, 8, 173548- 173565.

Janga Reddy, M., & Nagesh Kumar, D. (2020). Evolutionary algorithms, swarm intelligence methods, and their applications in water resources engineering: a state-of-the-art review. H2Open Journal, 3(1), 135-188.

Janga Reddy, M., & Nagesh Kumar, D. (2021). Evolutionary algorithms, swarm intelligence methods, and their applications in water resources engineering: a state-of-the-art review. H2Open Journal, 3(1), 135-188

Kruse, R., Borgelt, C., Braune, C., Mostaghim, S., Steinbrecher, M., 2016. Computational Intelligence: A Methodological Introduction, Springer. doi:10.1007/3-540-48774-3

References

Mei, L.; Wang, Q. (2021), Structural Optimization in Civil Engineering: A Literature Review. Buildings 11, 66.

Okwu, M. O., & Tartibu, L. K. (2021). Metaheuristic Optimization: Nature-Inspired Algorithms Swarm and Computational Intelligence, Theory and Applications (Vol. 927). Springer Nature.

QING QUAN LIANG - Performance-Based Optimization of Structures_ Theory and Applications-Spon Press (2004)

Rao RV (2020) Rao algorithms: three metaphor-less simple algorithms for solving optimization problems. Int J and Eng Comput 11 :107–130

Rao RV (1996) S - Engineering optimization_ theory and practice- 3 eme titre. John Wiley & Sons

Tadj, W. (2019). Apports de l'intelligence artificielle au calage automatique des modèles numériques en hydraulique. Ph. D thesis. University of Laghouat.

Walid Tadj, Mohamed Chettih & Kaddour Mouattah (2021). A new hybrid algorithm for estimating confined and leaky aquifers parameters from transient time-drawdown data Soft Computing volume 25, pages15463–15476

Wolpert, D. H., & Macready, W. G. (1997). No free lunch theorems for optimization. IEEE transactions on evolutionary computation, 1(1), 67-82.

Yang, X. S. (2014). Nature-inspired optimization algorithms. Academic Press.

Yang, X. S., Deb, S., Zhao, Y. X., Fong, S., & He, X. (2018). Swarm intelligence: past, present and future. Soft Computing, 22(18), 5923-5933

Yang, X.S., 2010. Engineering Optimization: An Introduction with Metaheuristic Applications. doi:10.1002/9780470640425

Yusuf Cengiz Toklu, Gebrail Bekdas, Sinan Melih Nigdeli - Metaheuristics for Structural Design and Analysis-Wiley-ISTE (2021)

Yusuf Cengiz Toklu, Gebrail Bekdas, Sinan Melih Nigdeli. (2021) - Metaheuristics for Structural Design and Analysis-Wiley-ISTE