OPEN ACCESS

The Journal of Engineering and Exact Sciences – jCEC, Vol. 09 N. 12 (2023)
journal homepage: https://periodicos.ufv.br/jcec
eISSN: 2527-1075
ISSN: 2446-9416

A comparative approach for the optimal design of steel structures using biogeography-based optimization (BBO) algorithm and genetic algorithm (GA)

Un enfoque comparativo para el diseño óptimo de estructuras de acero utilizando el algoritmo de optimización basado en la biogeografía (BBO) y el algoritmo genético (GA)

Article Info:
Article history: Received 2023-11-02 / Accepted 2023-12-03 / Available online 2023-12-24
doi: 10.18540/jcecvl9iss12pp17792

Laid Amamra
ORCID: https://orcid.org/0000-0001-8932-3575
LMPC Laboratory, Department of Civil Engineering, Faculty of Science and Technology, University of Mostaganem, P.O Box 227, Mostaganem 27000 (Algeria)
E-mail: laid.amamra@univ-mosta.dz
Mohamed Bensoula
ORCID: https://orcid.org/0000-0002-4785-3591
LCTPE Laboratory, Department of Civil Engineering, Faculty of Science and Technology, University of Mostaganem, P.O Box 227, Mostaganem 27000 (Algeria),
E-mail: mohamed.bensoula@univ-mosta.dz
Sadek Bahar
ORCID: https://orcid.org/0009-0003-1355-474X
LMPC Laboratory, Department of Civil Engineering, Faculty of Science and Technology, University of Mostaganem, P.O Box 227, Mostaganem 27000 (Algeria)
E-mail: sadek.bahar@univ-mosta.dz

Resumo
La optimización estructural es una de las principales preocupaciones de los diseñadores de ingeniería civil, pero desde un punto de vista matemático, este problema de optimización es muy complejo y complicado debido a un gran número de restricciones de diseño no lineales y al procedimiento iterativo del análisis estructural. La introducción de algoritmos de optimización como la optimización basada en la biogeografía (BBO) y los algoritmos genéticos (GA) en las aplicaciones puede ayudar al usuario a optimizar el coste de la estructura a adoptar más rápidamente y con menos errores en la fase preliminar del estudio de diseño. El objetivo de esta investigación es realizar una aproximación comparativa a la minimización del peso de la estructura utilizando el algoritmo de Optimización Basada en Biogeografía (BBO) y algoritmos genéticos (GA), examinando la influencia del número de poblaciones y el número de iteraciones en los resultados finales. En este estudio, ambos algoritmos arrojaron resultados fiables, pero la comparación de los resultados obtenidos por los dos métodos revela que el algoritmo de Optimización Basada en la Biogeografía (BBO) puede utilizarse con éxito para la optimización de estructuras de acero, garantizando al mismo tiempo la verificación de los criterios de resistencia, capacidad de servicio y estabilidad definidos por el Eurocódigo 3 (Unión, 2006), ya que presenta ciertas ventajas en la detección del mínimo global con respecto a los algoritmos genéticos (AG). Es capaz de encontrar soluciones más ligeras, más rígidas y con menor deflexión que los diseños originales.
Palavras-chave: Estructuras de acero. Optimización. Algoritmo BBO. Algoritmo GA. Sistemas expertos. Optimización multicriterio. Eurocódigo 3.

Abstract
Structural optimization is one of the key concerns of civil engineering designers, but from a mathematical point of view, this optimization problem is highly complex and complicated due to a large number of non-linear design constraints and the iterative procedure of structural analysis. Introducing optimization algorithms such as biogeography-Based Optimization (BBO) and genetic algorithms (GA) into applications can help the user to optimize the cost of the structure to be adopted more quickly and with fewer errors in the preliminary phase of the design study. The aim of this research is to carry out a comparative approach to structure weight minimization using biogeography-Based Optimization (BBO) algorithm and genetic algorithms (GA), examining the influence of the number of populations and the number of iterations in the final results. In this study, both algorithms gave reliable results, but a comparison of the results obtained by the two methods reveals that the biogeography-Based Optimization algorithm (BBO) can be successfully used for the optimization of steel structures while ensuring verification of the strength, serviceability and stability criteria defined by Eurocode 3 (Union, 2006), as it has certain advantages in detecting the global minimum over genetic algorithms (GA). It is capable of finding solutions that are lighter, stiffer and have lower deflection than the original designs.
Keywords: Steel structures. Optimization. BBO algorithm. GA algorithm. Expert systems. Multicriteria optimization. Eurocode 3.

1. Introduction

The problem of the overall design of steel structures can be posed as an optimization problem which consists in minimizing the overall cost of the structure while respecting the constraints of time, resources and dimensioning, thus forcing designers to limit the number of configurations to be considered in the early design phase. Most structural optimization methods use mathematical algorithms that require a large amount of data, where the choice of initial values is important for the convergence of the algorithm toward an optimal solution by Schmit (1960).

Optimizing a structure is complex, complicated and time-consuming due to the large number of non-linear design constraints and the iterative procedure of structural analysis. It must guarantee minimum cost with maximum safety, based on the search for the most admissible dimensioning, while ensuring verification of the resistance, serviceability and stability criteria defined by Eurocode 3 (Union, 2006). Optimization algorithms can be used in several ways to improve the design of steel structures according to Eurocode 3 (Union, 2006), such as topology, form, dimensions, loading and multi-objective optimization. Methods such as genetic algorithms (GA), biogeography-based optimization (BBO), simulated annealing, particle swarm optimization and ant colony optimization are particularly useful for problems with large and complex design space, multiple objectives and non-linear relationships. They can also address problems for which traditional optimization methods fail to find an optimal solution.

In general, the evaluation of the minimum price takes into account several factors such as the weight of the structure, the manufacture of the constituent parts and assembly etc., and given current steel prices, the weight of the frame is a very important factor among those mentioned and therefore its impact on the price of the structure is considerable. In this article, the optimization approach for steel structures formed of beams and columns is essentially based on minimizing the weight of the structure, and consequently, the optimal structure is the one whose weight is minimal while ensuring the structure's stable and economical configuration whatever its complexity.

This optimization approach is based on the application of Biogeography-Based Optimization (BBO) and Genetic Algorithms (GA), which consists in minimizing an objective function representing the weight of the structure and whose design variables are the beam and column cross-sections selected from a database. The design constraints include the mechanical strength of sections and elements, the exploitation of the elasto-plastic behavior of steel (discovery of the plastic optimum of elements), the limitation of displacements at nodes, and the limitation of the structure's first natural frequency.

The use of structural optimization algorithms in Eurocode 3 (Union, 2006) is likely to have a significant impact on the design of steel structures, as it considerably helps engineers to explore quickly and efficiently a wide range of design options and identify the optimum solution that meets performance criteria by automating many calculation and analysis steps.

2. State of art

A review of the literature shows that there are many techniques used to optimize structure design. They fall into two categories: classical mathematical programming techniques and stochastic search with heuristic algorithms.

Early research used mathematical programming techniques to obtain the optimal solution (Arora, 1997; Belegundu et al., 2011; Doan et al., 2012; Perez et al., 2007; Schmit, 1960). They found that, mathematical programming techniques such as linear and nonlinear programming are not easy to apply to real structures and can only be used for steel structures with limited design variables.

To overcome these limitations, other new computational techniques have been developed using stochastic search and algorithms based on the reproduction of the hypotheses found in nature, called metaheuristic optimization. They are inspired by analogies with physics, biology or ethology (Camp et al., 2004; Kaveh et al., 2007; Kaveh et al., 2010) and these optimization methods based on stochastic search have proved their effectiveness in obtaining the solution to problems in such a short time because they use probability rules instead of deterministic rules. Engineering fields have benefited the most from these methods based on metaheuristic techniques, which have become popular topics these days.

The best-known and most widely used heuristic method for solving structural optimization is the genetic algorithm GA (Amamra et al., 2021; Camp et al., 1998; Erbatur et al., 2000; Kaveh et al., 2006; Pezeshk et al., 2000; Rahami et al., 2008; Rajeev et al., 1992; Saka et al., 1998; Shook et al., 2008;). These researchers used and developed a method based on a genetic algorithm to solve complex calculation problems in the design and find the optimal structure to minimize the weight, size, shape or all of these parameters of the steel or reinforced concrete frame. Some researchers have proposed a modified GA algorithm to improve this performance, and others have compared the GA algorithm with other approaches to find the optimal design and the best algorithm to apply.

Rajasekaran et al. (2004 a) used this technique for the optimal design of large spatial steel structures. Ebenau et al. (2005) combined the ES method with an adaptive penalty function by applying an algorithm to find the optimal design of geometrically nonlinear three-dimensional steel structures. Hasançebi (2007 a) used different reformulations of evolution strategies to solve a discrete optimal design for steel frames, on the other hand, Hasançebi (2007 b) used the evolution strategy (ES) method and applied it to truss bridges to obtain an optimal structural design.

Particle Swarm Optimization (PSO) was developed by mimicking the social behavior of entities such as flocks of birds, swarms of insects, or schools of fish. The algorithm summarizes the clustering behavior of social forces that depend on experience sharing from both the local memory of individuals as well as the knowledge acquired by the swarm (Doan et al., 2012; Fourie et al., 2002; Kennedy et al., 1995; Li et al., 2007; Perez et al., 2007) applied the PSO algorithm on different civil construction structures to find optimal size and shape variables.

One other recent addition to metaheuristic algorithms is the Ant Colony Optimization (ACO) algorithm. Dorigo et al. (2006) developed this method in the early 1990s. This method summarizes the behavior of ant colonies in search of food. Articles have been written based on the ant colony optimization technique to optimize space farms and steel frames like the work of (Camp et al., 2004; Kaveh et al., 2007; Kaveh et al., 2010) and many others. (Hasançebi et al., 2010; Lee et al., 2004; Saka, 2009; Saka et al., 2009 ; Saka et al., 2011) have used the harmony search method to address continuous engineering optimization problems.

Biogeography-based optimization (BBO) is metaheuristic algorithm. The main idea is similar to a biology-based approach such as Genetic Algorithms (Ma et al., 2015; Reeves et al., 2002) created by Simon (2008). BBO algorithm is an evolutionary optimization algorithm inspired by the geographical distribution of biological organisms (Simon, 2008). Which are based on the behavior of biological species and their distribution between habitats. Therefore, the BBO algorithm is used to solve many practical optimization problems cited by Malik et al. (2016). As any other evolutionary algorithms, the BBO algorithm deals with a population of candidate solutions, each solution is characterized by a number of features as in the work of Ma et al. (2015). In this paper, BBO and GA algorithms were used to optimize the weights of metallic structure.

3. Methodology and operating principles of the BBO and GA algorithms

3.1 BBO algorithm operating principle

The BBO algorithm is created based on the study of the distribution of species over space and time (biogeography) and is a subdivision of biology. This algorithm is based on the principle of the geographical distribution of biological species on islands (Kaveh et al., 2004). BBO algorithm uses migration and mutation operations to get a global minimum solution. Migration is the movement of species between different habitats. Migration is a probabilistic operator that adjusts each solution by sharing features between solutions. On an island or habitat, if the living conditions are appropriate for the species, then that habitat has a high suitability index (HSI), because this habitat has better features than other habitats. The variables that characterize the habitability of an island are called suitability index variables (SIVs) (Kaveh et al., 2004) for that reason, the species in the high HSI habitats can emigrate to other habitats and have low species immigration rates as they are already full with species. Likewise, the low HSI habitats have a high immigration rate of species because of their low population.

The working principle of the BBO algorithm is motivated by the natural events of biogeography. The basic concepts of the BBO algorithm are analogous to the traits of biogeography. A general framework of the BBO algorithm is shown in Figure 1. Further detailed explanation about the BBO algorithm can be found in (Kaveh et al. 2004; Rajasekaran et al. 2004 b).

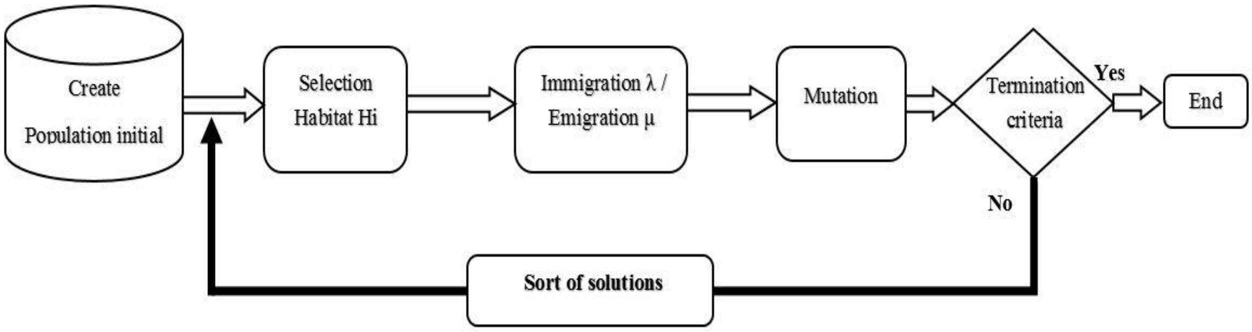

Figure 1 - A general framework of the BBO algorithm.

This section demonstrates how the BBO algorithm can be used to develop an efficient approach for structural optimization. As it has been mentioned before, A BBO algorithm operates to get the or a global minimum solution in a search space defined by an objective function. To get a good result from our approach, we use these steps:

The initialization phase: in this phase, the elements of the population are generated randomly and coded in the search space. A data structure must be linked to each point of the search space. The choice of the initial population and the coding of the data are conditions for the success of the BBO algorithms for finding a good result. The population should be non-homogeneous individuals which makes the convergence of the approach more rapid.

Fitness evaluation: for each individual in the population, the fitness function is evaluated and the rate is assigned based on their fitness value. This is used to select and reproduce the best individuals in the population.

Calculation of immigration, emigration and mutation rates: migration operators, emigration operators and mutation operators make it possible to diversify the population over generations and explore the search space.

Stopping criteria: The immigration and emigration operations are repeated until the stop criterion is reached and the optimal solution is obtained. Optimization based on the biogeography BBO algorithm is given by the following algorithm:

1. Initialize the BBO algorithm parameters
2. Set iteration=1
3. Randomly initialize populations
4. Evaluate the HSI of every habitat with minimized tardiness objective function.
5. Calculate λi and μi
6. Modify the non-elite members of the population probabilistically with the migration operator
7. Mutate the non-elite members of the population with the mutation operator
8. Evaluate new habitats.
9. Replace the old habitats with new ones.
10. Iteration = iteration + 1
11. Go to the step (4) until reaching the maximum iteration.

3.2 GA algorithm operating principle

One of the most known metaheuristic algorithms is the Genetic algorithm (GA), proposed by Holland (1992). Holland inspired his algorithm from the Darwinian theory of biological evolution. The survival process and the fittest in nature are the main ideas of the GA algorithm. Operations of the GA algorithm are selection, mutation, and crossover inspired directly by nature. One element of the population is a candidate as a point in the solution space. This element represents the chromosome in nature. Each element of the population is handled iteratively using genetic operators and replacing it in the population after assigning a value by the fitness function. The procedure of the GA algorithm presented in Figure 2 and detailed as follows:

First the selection operation, based on fitness value, chromosomes are selected for further treatment. In the cross operator, two selected elements are chosen at random and modify the subsequences between them to create descendants. In mutating, some parts of the element will be reversed randomly based on the probability given.

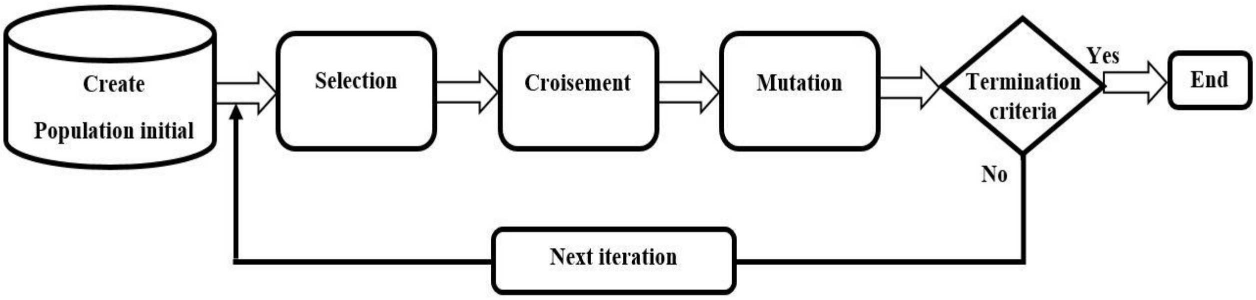

Figure 2 - A general framework of the GA algorithm

Optimization based on the GA algorithm is given by the following algorithm:

1) Initialize the GA algorithm parameters
2) Randomly initialize populations
3) Set iteration = 1
4) Evaluate the fitness of the population

Until (convergence or iteration = max) repeat:

a) Select parents from the population
b) Crossover and generate a new population
c) Perform mutation on new population

d) Calculate fitness for the new population

3.3 Methodology

To show the efficiency of the developed optimization methods to optimize the design of a three-dimension metallic structure with a beam, the weight of the structure is the variable to optimize, and to minimize the elements of the structure that guaranteed accepted construction costs. In order to optimize the structure, we create our own calculation engine to perform the structural modeling, analysis and design.

The BBO algorithm and GA algorithm are implemented and developed with NetBeans IDE 8.0.2 powered by JAVA and are used to optimize the design of a three-dimension metallic structure with a beam. The structure is made of steel and is subjected to a uniformly distributed load. The objective function of the optimization problem is to minimize the weight of the structure while satisfying all of the constraints. The design variables are the cross-sectional area of beams, the thickness of plates, and the shape of the structure.

The BBO algorithm and GA algorithm are initialized with a population of 50 solutions. The two algorithms then start working iteratively to update the population of solutions by migrating solutions from one island to another in the BBO algorithm and by using selection and cross in the GA algorithm. The operations used on solutions are based on the fitness of the solutions. Solutions with higher fitness are more likely to be migrated or to be crossed.

The two algorithms continue to iterate until a termination criterion is met. The termination criterion is set to be 100 iterations.

4. Results and analysis

In this section, the performance of the BBO and GA algorithms for optimizing steel structures is evaluated, taking into account two criteria: the number of populations and the number of iterations. The portal frame shown in Figure 3 is the object of our optimization.

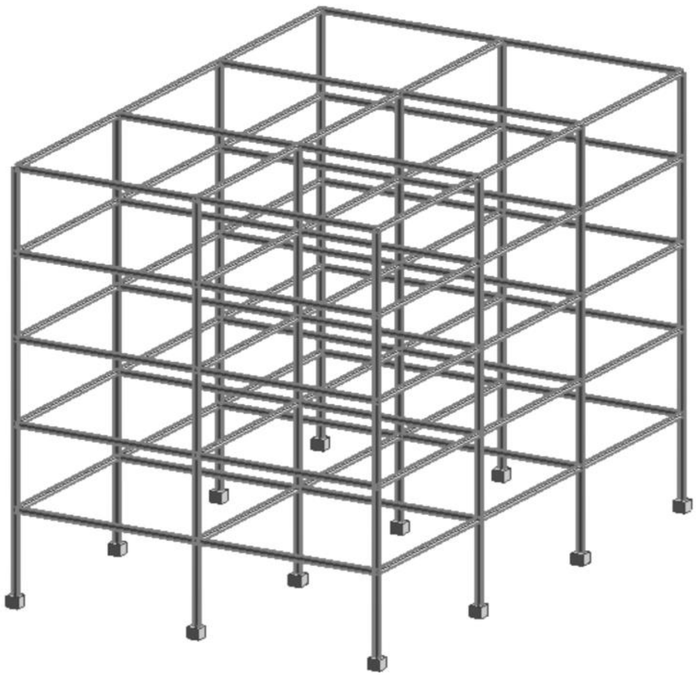

Figure 3 - Typical gantry for optimization

The user can select his profile from a database containing all metal profiles in accordance with Eurocode 3 (Union, 2006), then enter the other parameters of the portal frame design according to the user's preferences listed in Table 1.

Table 1 - Parameters used in gantry optimization

	Num	Parameters
Structure Parameters	1	Length
	2	Height
	3	Loadings
	4	Maximum horizontal displacement
	5	Permissible wane
	6	Steel yield strength f_y
	7	Ratio of moments γ
	8	Young's modulus E
Algorithm Parameters	9	Populations number
	10	Iterations number
	11	Probability of migration
	12	Probability of immigration
	13	Probability of mutation
	14	Probability of selection
	15	Probability of crossover

On the other hand, the user must enter all the parameters of the algorithms listed in table 1 used to run the programs. Once these parameters have been entered, the user can vary them according to the criteria of his choice.

Once the structure has been created, the program runs a series of checks in accordance with Eurocode 3 (Union, 2006), Eurocode 3 such as compressive strength, tensile strength, shear strength, flexural strength and the various structural instabilities of the structure's columns and beams.

Once the structure has been checked and validated, the chosen algorithm starts the calculation part, minimizing the weight of the structure, which has a direct influence on the economic cost. At the end of the run, the program produces a set of optimal structures for the user to explore, and choose which structure to adopt.

To determine the optimal structure, the optimization problem can be formulated as follows:

Minimizer : $w(x) = \sum_{i=1}^{n} p_i A_i \sum_{j=1}^{m} L_j$

Subject to : gi (X) ≤ 0, i= 1, 2, . . . , n

X = {x1 x2 . . . xi . . . xn }T

Where:

w: The weight of the structure;

x: An integer to express the sequence of steel cross-sections for the ith group;

A_i, p_i : The area and weight of the ith section of the element, respectively;

Lj: Element length;

gi: Loading stress.

The set of optimal structures generated by the two algorithms allows the user to make a comparison between the two algorithms using several criteria listed previously in Table 1.

A case study is proposed to illustrate how well the two proposed algorithms work. The chosen structure is a single-story embedded metal gantry, the characteristics of which are shown in Figure

4. A comparison between these two programs will be carried out using only two criteria: the number of populations and the number of iterations.

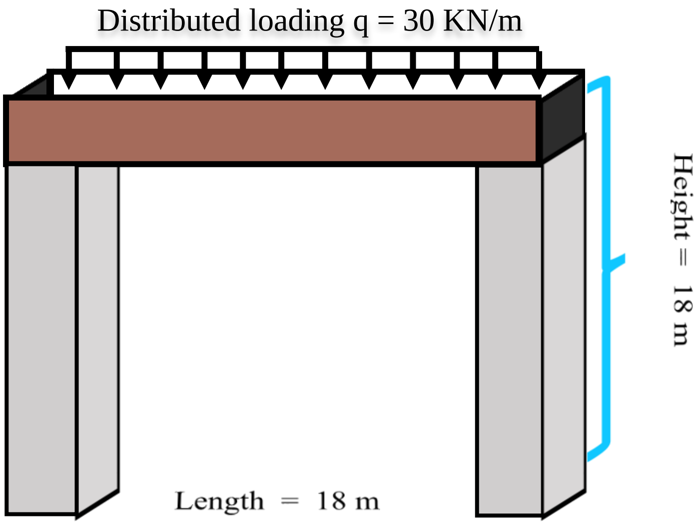

Figure 4 - Typical gantry for optimization

4.1 Criterion No. 01: Population size

To test the influence of this criterion on the result given by the program, the number of iterations is fixed with the variation of the population number, which varies between 20 and 200. During the execution of the algorithms, the results obtained by setting the population number to 20 are summarized in Table 2 for the BBO algorithm and Table 3 for the GA algorithm.

Table 2 - Results of program execution for the BBO algorithm with population number = 20

Num	001	002	003	Global weight (t)	F calculated	Arrow max (L/2)	MA=MD	MB=MC	Sigma Column	Sigma Beam
1	HEB 260	HEB 260	IPE 180	$3.69 \cdot 10^{+3}$	$2.50 \cdot 10^{-7}$	$-2.21 \cdot 10^{-4}$	$3.23 \cdot 10^{+4}$	$-6.46 \cdot 10^{+4}$	$1.93 \cdot 10^{-3}$	$-2.21 \cdot 10^{-4}$
2	HEB 200	HEB 200	IPE 360	$3.23 \cdot 10^{+3}$	$8.13 \cdot 10^{-8}$	$-1.54 \cdot 10^{-5}$	$1.39 \cdot 10^{+4}$	$-2.78 \cdot 10^{+4}$	$2.91 \cdot 10^{-3}$	$-1.54 \cdot 10^{-5}$
3	HEB 200	HEB 200	IPE 360	$3.23 \cdot 10^{+3}$	$8.13 \cdot 10^{-8}$	$-1.54 \cdot 10^{-5}$	$1.39 \cdot 10^{+4}$	$-2.78 \cdot 10^{+4}$	$2.91 \cdot 10^{-3}$	$-1.54 \cdot 10^{-5}$
4	HEA 260	HEA 260	IPE 240	$3.01 \cdot 10^{+3}$	$1.37 \cdot 10^{-7}$	$-8.77 \cdot 10^{-5}$	$2.85 \cdot 10^{+4}$	$-5.69 \cdot 10^{+4}$	$2.63 \cdot 10^{-3}$	$-8.77 \cdot 10^{-5}$
5	HEA 220	HEA 220	IPE 240	$2.37 \cdot 10^{+3}$	$1.59 \cdot 10^{-7}$	$-7.65 \cdot 10^{-5}$	$2.48 \cdot 10^{+4}$	$-4.96 \cdot 10^{+4}$	$3.55 \cdot 10^{-3}$	$-7.65 \cdot 10^{-5}$
6	HEA 220	HEA 220	IPE 200	$2.22 \cdot 10^{+3}$	$2.27 \cdot 10^{-7}$	$-1.47 \cdot 10^{-4}$	$2.86 \cdot 10^{+4}$	$-5.72 \cdot 10^{+4}$	$3.55 \cdot 10^{-3}$	$-1.47 \cdot 10^{-4}$
7	HEA 200	HEA 200	IPE 240	$2.08 \cdot 10^{+3}$	$1.76 \cdot 10^{-7}$	$-6.81 \cdot 10^{-5}$	$2.21 \cdot 10^{+4}$	$-4.42 \cdot 10^{+4}$	$4.24 \cdot 10^{-3}$	$-6.81 \cdot 10^{-5}$
8	HEA 200	HEA 200	IPE 240	$2.08 \cdot 10^{+3}$	$1.76 \cdot 10^{-7}$	$-6.81 \cdot 10^{-5}$	$2.21 \cdot 10^{+4}$	$-4.42 \cdot 10^{+4}$	$4.24 \cdot 10^{-3}$	$-6.81 \cdot 10^{-5}$
9	HEA 160	HEA 260	IPE 80	$1.88 \cdot 10^{+3}$	$1.76 \cdot 10^{-6}$	-0.0016459	$3.30 \cdot 10^{+4}$	$-6.59 \cdot 10^{+4}$	$5.95 \cdot 10^{-3}$	$-1.65 \cdot 10^{-3}$
10	HEA 120	HEA 120	IPE 360	$1.74 \cdot 10^{+3}$	$1.07 \cdot 10^{-7}$	$-2.59 \cdot 10^{-6}$	$2.34 \cdot 10^{+4}$	$-4.68 \cdot 10^{+3}$	$8.92 \cdot 10^{-3}$	$-2.59 \cdot 10^{-6}$
11	HEA 120	HEA 120	IPE 360	$1.74 \cdot 10^{+2}$	$1.07 \cdot 10^{-7}$	$-2.59 \cdot 10^{-6}$	$2.34 \cdot 10^{+3}$	$-4.68 \cdot 10^{+3}$	$8.92 \cdot 10^{-3}$	$-2.59 \cdot 10^{-6}$
12	HEB 140	HEB 140	IPE 180	$1.55 \cdot 10^{+3}$	$3.71 \cdot 10^{-7}$	$-1.61 \cdot 10^{-4}$	$2.35 \cdot 10^{+4}$	$-4.70 \cdot 10^{+4}$	$5.34 \cdot 10^{-3}$	$-1.61 \cdot 10^{-4}$
13	HEB 140	HEB 140	IPE 180	$1.55 \cdot 10^{+3}$	$3.71 \cdot 10^{-7}$	$-1.61 \cdot 10^{-4}$	$2.35 \cdot 10^{+4}$	$-4.70 \cdot 10^{+4}$	5.34 10-3	$-1.61 \cdot 10^{-4}$

14	HEB 140	HEB 140	IPE 180	$1.55 \cdot 10^{+3}$	$3.71 \cdot 10^{-7}$	$-1.61 \cdot 10^{-4}$	$2.35 \cdot 10^{+4}$	$-4.70 \cdot 10^{+4}$	$5.34 \cdot 10^{-3}$	$-1.61 \cdot 10^{-4}$
15	HEA 180	HEA 180	IPE 80	$1.39 \cdot 10^{+3}$	$1.74 \cdot 10^{-6}$	$-1.66 \cdot 10^{-3}$	$3.32 \cdot 10^{+4}$	$-6.64 \cdot 10^{+4}$	$5.08 \cdot 10^{-3}$	$-1.66 \cdot 10^{-3}$
16	HEB 100	HEB 100	IPE 220	$1.21 \cdot 10^{+3}$	$3.36 \cdot 10^{-7}$	$-3.28 \cdot 10^{-5}$	$8.27 \cdot 10^{+3}$	$-1.65 \cdot 10^{+4}$	$8.75 \cdot 10^{-3}$	$-3.28 \cdot 10^{-5}$
17	HEA 100	HEA 100	IPE 240	$1.15 \cdot 10^{+3}$	$2.81 \cdot 10^{-7}$	$-1.58 \cdot 10^{-5}$	$5.13 \cdot 10^{+3}$	$-1.03 \cdot 10^{+4}$	$1.07 \cdot 10^{-2}$	$-1.58 \cdot 10^{-5}$
18	HEA 100	HEA 100	IPE 240	$1.15 \cdot 10^{+3}$	$2.81 \cdot 10^{-7}$	$-1.58 \cdot 10^{-5}$	$5.13 \cdot 10^{+3}$	$-1.03 \cdot 10^{+4}$	$1.07 \cdot 10^{-2}$	$-1.58 \cdot 10^{-5}$
19	HEB 100	HEB 100	IPE 200	$1.14 \cdot 10^{+3}$	$4.11 \cdot 10^{-7}$	$-5.50 \cdot 10^{-5}$	$1.07 \cdot 10^{+4}$	$-2.14 \cdot 10^{+4}$	$8.77 \cdot 10^{-3}$	$-5.50 \cdot 10^{-5}$
20	HEB 100	HEB 100	IPE 200	$1.14 \cdot 10^{+3}$	$4.11 \cdot 10^{-7}$	$-5.50 \cdot 10^{-5}$	$1.07 \cdot 10^{+4}$	$-2.14 \cdot 10^{+4}$	$8.77 \cdot 10^{-3}$	$-5.50 \cdot 10^{-5}$

Table 3 - Results of program execution for the GA algorithm with population number = 20

Num	**001**	**002**	**003**	**Global weight (t)**	**F calculated**	**Arrow max (L/2)**	**MA=MD**	**MB=MC**	**Sigma Column**	**Sigma Beam**
1	HEB 260	HEB 260	IPE 180	$3.69 \cdot 10^{+3}$	$2.50 \cdot 10^{-7}$	$-2.21 \cdot 10^{-4}$	$3.23 \cdot 10^{+4}$	$-6.46 \cdot 10^{+4}$	$1.93 \cdot 10^{-3}$	$-2.21 \cdot 10^{-4}$
2	HEA 260	HEB 260	IPE 200	$3.30 \cdot 10^{+3}$	$2.03 \cdot 10^{-7}$	$-1.59 \cdot 10^{-4}$	$3.09 \cdot 10^{+4}$	$-6.18 \cdot 10^{+4}$	$2.63 \cdot 10^{-3}$	$-1.59 \cdot 10^{-4}$
3	HEB 200	HEB 200	IPE 360	$3.23 \cdot 10^{+3}$	$8.13 \cdot 10^{-8}$	$-1.54 \cdot 10^{-5}$	$1.39 \cdot 10^{+4}$	$-2.78 \cdot 10^{+4}$	$2.91 \cdot 10^{-3}$	$-1.54 \cdot 10^{-5}$
4	HEA 160	HEA 450	IPE 80	$3.18 \cdot 10^{+3}$	$1.76 \cdot 10^{-6}$	$-1.65 \cdot 10^{-3}$	$3.30 \cdot 10^{+4}$	$-6.59 \cdot 10^{+4}$	$5.95 \cdot 10^{-3}$	$-1.65 \cdot 10^{-3}$
5	HEB 220	HEA 240	IPE 240	$2.93 \cdot 10^{+3}$	$1.44 \cdot 10^{-7}$	$-8.39 \cdot 10^{-5}$	$2.72 \cdot 10^{+4}$	$-5.44 \cdot 10^{+4}$	$2.51 \cdot 10^{-3}$	$-8.39 \cdot 10^{-5}$
6	HEB 220	HEA 240	IPE 220	$2.84 \cdot 10^{+3}$	$1.73 \cdot 10^{-7}$	$-1.14 \cdot 10^{-4}$	$2.88 \cdot 10^{+4}$	$-5.76 \cdot 10^{+4}$	$2.51 \cdot 10^{-3}$	$-1.14 \cdot 10^{-4}$
7	HEB 200	HEB 200	IPE 240	$2.76 \cdot 10^{+3}$	$1.57 \cdot 10^{-7}$	$-7.76 \cdot 10^{-5}$	$2.52 \cdot 10^{+4}$	$-5.03 \cdot 10^{+4}$	$2.93 \cdot 10^{-3}$	$-7.76 \cdot 10^{-5}$
8	HEB 200	HEB 200	IPE 240	$2.76 \cdot 10^{+3}$	$1.57 \cdot 10^{-7}$	$-7.76 \cdot 10^{-5}$	$2.52 \cdot 10^{+4}$	$-5.03 \cdot 10^{+4}$	$2.93 \cdot 10^{-3}$	$-7.76 \cdot 10^{-5}$
9	HEA 240	HEA 240	IPE 200	$2.57 \cdot 10^{+3}$	$2.12 \cdot 10^{-7}$	$-1.54 \cdot 10^{-4}$	$3.00 \cdot 10^{+4}$	$-6.00 \cdot 10^{+4}$	$2.97 \cdot 10^{-3}$	$-1.54 \cdot 10^{-4}$
10	HEA 240	HEA 240	IPE 200	$2.57 \cdot 10^{+3}$	$2.12 \cdot 10^{-7}$	$-1.54 \cdot 10^{-4}$	$3.00 \cdot 10^{+4}$	$-6.00 \cdot 10^{+4}$	$2.97 \cdot 10^{-3}$	$-1.54 \cdot 10^{-4}$
11	HEB 200	HEB 200	IPE 180	$2.55 \cdot 10^{+3}$	$2.78 \cdot 10^{-7}$	$-2.07 \cdot 10^{-4}$	$3.03 \cdot 10^{+4}$	$-6.05 \cdot 10^{+4}$	$2.93 \cdot 10^{-3}$	$-2.07 \cdot 10^{-4}$
12	HEB 220	HEB 220	IPE 80	$2.68 \cdot 10^{+3}$	$1.70 \cdot 10^{-6}$	$-1.68 \cdot 10^{-3}$	$3.36 \cdot 10^{+4}$	$-6.72 \cdot 10^{+4}$	$2.52 \cdot 10^{-3}$	$-1.68 \cdot 10^{-3}$
13	HEB 200	HEB 200	IPE 180	$2.55 \cdot 10^{+3}$	$2.78 \cdot 10^{-7}$	$-2.07 \cdot 10^{-4}$	$3.03 \cdot 10^{+4}$	$-6.05 \cdot 10^{+4}$	$2.93 \cdot 10^{-3}$	$-2.07 \cdot 10^{-4}$
14	HEA 200	HEA 200	IPE 330	$2.41 \cdot 10^{+3}$	$1.05 \cdot 10^{-7}$	$-1.82 \cdot 10^{-5}$	$1.30 \cdot 10^{+4}$	$-2.60 \cdot 10^{+4}$	$4.22 \cdot 10^{-3}$	$-1.82 \cdot 10^{-5}$
15	HEA 200	HEA 200	IPE 330	$2.41 \cdot 10^{+3}$	$1.05 \cdot 10^{-7}$	$-1.82 \cdot 10^{-5}$	$1.30 \cdot 10^{+4}$	$-2.60 \cdot 10^{+4}$	$4.22 \cdot 10^{-3}$	$-1.82 \cdot 10^{-5}$
16	HEA 220	HEA 220	IPE 240	$2.37 \cdot 10^{+3}$	$1.59 \cdot 10^{-7}$	$-7.65 \cdot 10^{-5}$	$2.48 \cdot 10^{+4}$	$-4.96 \cdot 10^{+4}$	$3.55 \cdot 10^{-3}$	$-7.65 \cdot 10^{-5}$
17	HEA 220	HEA 220	IPE 200	$2.22 \cdot 10^{+3}$	$2.27 \cdot 10^{-7}$	$-1.47 \cdot 10^{-4}$	$2.86 \cdot 10^{+4}$	$-5.72 \cdot 10^{+4}$	$3.55 \cdot 10^{-3}$	$-1.47 \cdot 10^{-4}$
18	HEB 100	HEA 220	IPE 240	$1.83 \cdot 10^{+3}$	$2.73 \cdot 10^{-7}$	$-1.95 \cdot 10^{-5}$	$6.34 \cdot 10^{+3}$	$-1.27 \cdot 10^{+4}$	$8.72 \cdot 10^{-3}$	$-1.95 \cdot 10^{-5}$
19	HEB 100	HEA 220	IPE 240	$1.83 \cdot 10^{+3}$	$2.73 \cdot 10^{-7}$	$-1.95 \cdot 10^{-5}$	$6.34 \cdot 10^{+3}$	$-1.27 \cdot 10^{+4}$	$8.72 \cdot 10^{-3}$	$-1.95 \cdot 10^{-5}$
20	HEB 140	HEB 140	IPE 180	$1.55 \cdot 10^{+3}$	$3.71 \cdot 10^{-7}$	$-1.61 \cdot 10^{-4}$	$2.35 \cdot 10^{+4}$	$-4.70 \cdot 10^{+4}$	$5.34 \cdot 10^{-3}$	$-1.61 \cdot 10^{-4}$

Population-dependent results are obtained for both algorithms, and Figure 5 illustrates the structure weight optimization for a population of 20. Clearly, the BBO algorithm gave a better optimal structure than the GA algorithm.

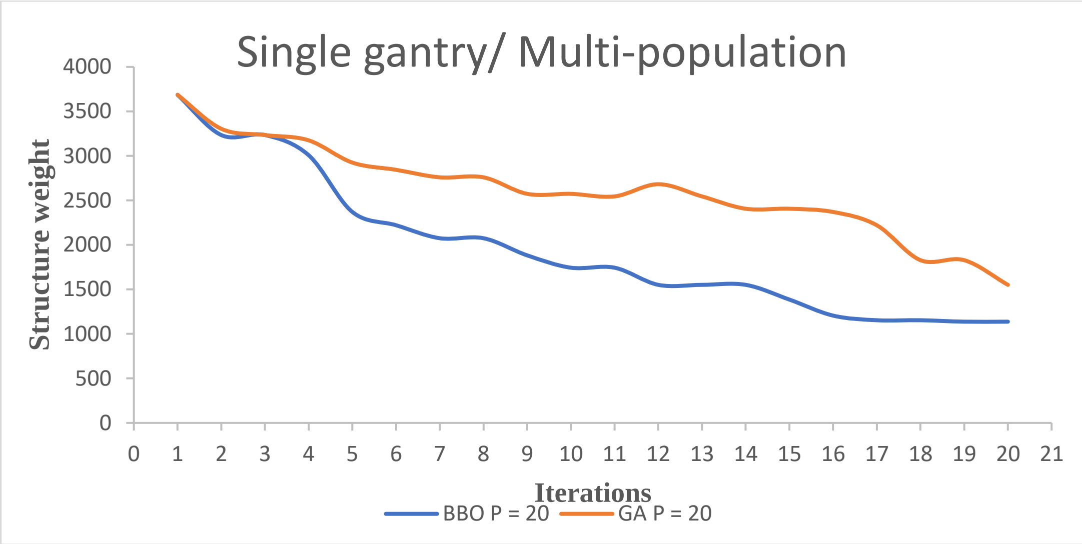

Figure 5 - Comparison of structure weight optimization between BBO and GA algorithms for a population P = 20

To properly validate this comparison between the two algorithms, it was necessary to take multiple population values ranging from 20 to 200. At the end of the algorithms' execution, Figure 6 illustrates this comparison between the BBO and GA algorithms for structure weight optimization, taking population values of 20, 50, 100, 150 and 200. Despite the change in population number, the BBO algorithm still gives better results than the GA algorithm, and the figure clearly shows that it improves these results as the iterations progress.

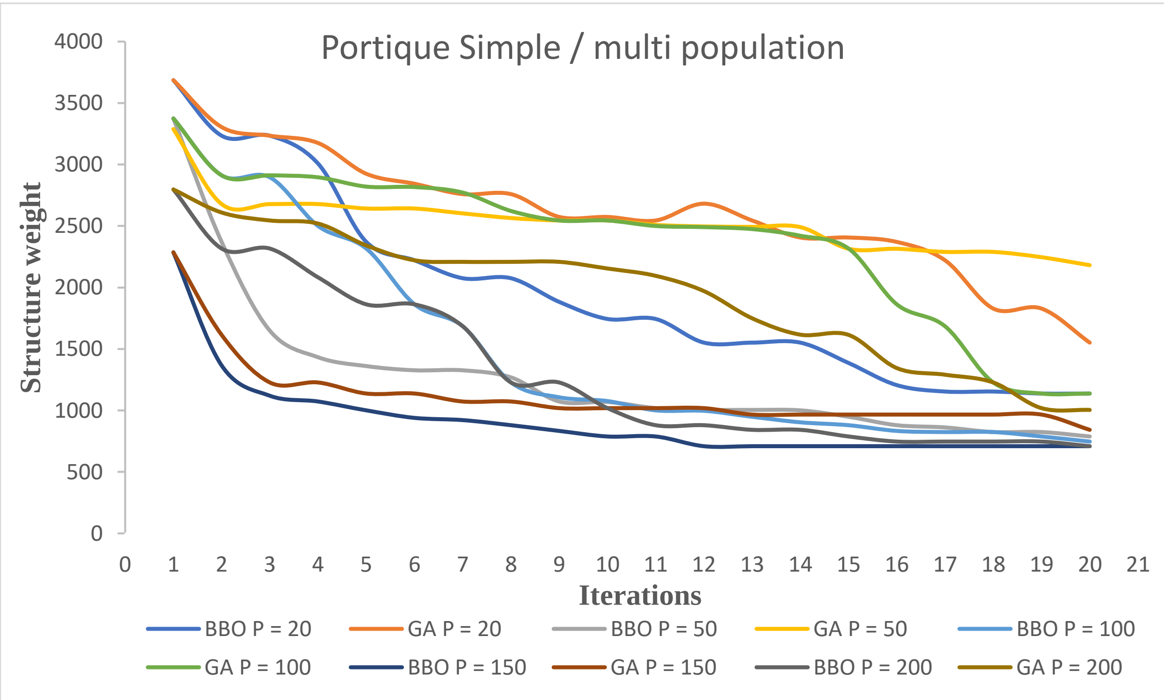

Figure 6. Comparison of structure weight optimization between BBO and GA algorithms for populations ranging from P = 20 to P = 200

4.2 Criterion No. 02: Iteration number

By fixing the number of the initial population and varying the number of iterations between 20 and 200. During the execution of the algorithms, the results obtained by setting the iteration number to 20 are summarized in Table 4 for the BBO algorithm and Table 5 for the GA algorithm.

Table 4 - Results of program execution for the BBO algorithm with iteration number = 20

Num	**001**	**002**	**003**	**Global weight (t)**	**F calculated**	**Arrow max (L/2)**	**MA=MD**	**MB=MC**	**Sigma Column**	**Sigma Beam**
1	HEA 360	HEA 360	IPE 180	$4.37\cdot10^{+3}$	$2.40\cdot10^{-7}$	$-2.26\cdot10^{-4}$	$3.31\cdot10^{+4}$	$-6.62\cdot10^{+4}$	$1.59\cdot10^{-3}$	$-2.26\cdot10^{-4}$
2	HEA 340	HEA 340	IPE 180	$4.12\cdot10^{+3}$	$2.41\cdot10^{-7}$	$-2.25\cdot10^{-4}$	$3.30\cdot10^{+4}$	$-6.59\cdot10^{+4}$	$1.71\cdot10^{-3}$	$-2.25\cdot10^{-4}$
3	HEB 260	HEB 260	IPE 180	$3.69\cdot10^{+3}$	$2.50\cdot10^{-7}$	$-2.21\cdot10^{-4}$	$3.23\cdot10^{+4}$	$-6.46\cdot10^{+4}$	$1.93\cdot10^{-3}$	$-2.21\cdot10^{-4}$
4	HEA 260	HEB 260	IPE 200	$3.30\cdot10^{+3}$	$2.03\cdot10^{-7}$	$-1.59\cdot10^{-4}$	$3.09\cdot10^{+4}$	$-6.18\cdot10^{+4}$	$2.63\cdot10^{-3}$	$-1.59\cdot10^{-4}$
5	HEB 200	HEB 200	IPE 360	$3.23\cdot10^{+3}$	$8.13\cdot10^{-8}$	$-1.54\cdot10^{-5}$	$1.39\cdot10^{+4}$	$-2.78\cdot10^{+4}$	$2.91\cdot10^{-3}$	$-1.54\cdot10^{-5}$
6	HEA 260	HEA 260	IPE 240	$3.01\cdot10^{+3}$	$1.37\cdot10^{-7}$	$-8.77\cdot10^{-5}$	$2.85\cdot10^{+4}$	$-5.69\cdot10^{+4}$	$2.63\cdot10^{-3}$	$-8.77\cdot10^{-5}$
7	HEB 220	HEB 220	IPE 180	$2.91\cdot10^{+3}$	$2.65\cdot10^{-7}$	$-2.13\cdot10^{-4}$	$3.12\cdot10^{+4}$	$-6.24\cdot10^{+4}$	$2.51\cdot10^{-3}$	$-2.13\cdot10^{-4}$
8	HEA 260	HEA 220	IPE 300	$2.90\cdot10^{+3}$	$9.52\cdot10^{-8}$	$-4.33\cdot10^{-5}$	$2.41\cdot10^{+4}$	$-4.82\cdot10^{+4}$	$2.62\cdot10^{-3}$	$-4.33\cdot10^{-5}$
9	HEA 240	HEA 240	IPE 270	$2.82\cdot10^{+3}$	$1.21\cdot10^{-7}$	$-5.73\cdot10^{-5}$	$2.46\cdot10^{+4}$	$-4.92\cdot10^{+4}$	$2.97\cdot10^{-3}$	$-5.73\cdot10^{-5}$
10	HEB 200	HEA 280	IPE 180	$2.82\cdot10^{+3}$	$2.78\cdot10^{-7}$	$-2.07\cdot10^{-4}$	$3.03\cdot10^{+4}$	$-6.05\cdot10^{+4}$	$2.93\cdot10^{-3}$	$-2.07\cdot10^{-4}$
11	HEA 220	HEB 200	IPE 300	$2.77\cdot10^{+3}$	$1.13\cdot10^{-7}$	$-3.42\cdot10^{-5}$	$1.90\cdot10^{+4}$	$-3.81\cdot10^{+4}$	$3.54\cdot10^{-3}$	$-3.42\cdot10^{-5}$
12	HEA 220	HEA 280	IPE 180	$2.62\cdot10^{+3}$	$2.81\cdot10^{-7}$	$-2.06\cdot10^{-4}$	$3.01\cdot10^{+4}$	$-6.02\cdot10^{+4}$	$3.56\cdot10^{-3}$	$-2.06\cdot10^{-4}$
13	HEB 200	HEB 200	IPE 180	$2.55\cdot10^{+3}$	$2.78\cdot10^{-7}$	$-2.07\cdot10^{-4}$	$3.03\cdot10^{+4}$	$-6.05\cdot10^{+4}$	$2.93\cdot10^{-3}$	$-2.07\cdot10^{-4}$
14	HEA 220	HEA 220	IPE 240	$2.37\cdot10^{+3}$	$1.59\cdot10^{-7}$	$-7.65\cdot10^{-5}$	$2.48\cdot10^{+4}$	$-4.96\cdot10^{+4}$	$3.55\cdot10^{-3}$	$-7.65\cdot10^{-5}$
15	HEA 220	HEA 220	IPE 200	$2.22\cdot10^{+3}$	$2.27\cdot10^{-7}$	$-1.47\cdot10^{-4}$	$2.86\cdot10^{+4}$	$-5.72\cdot10^{+4}$	$3.55\cdot10^{-3}$	$-1.47\cdot10^{-4}$
16	HEB 160	HEB 200	IPE 160	$2.15\cdot10^{+3}$	$4.03\cdot10^{-7}$	$-2.65\cdot10^{-4}$	$2.87\cdot10^{+4}$	$-5.75\cdot10^{+4}$	$4.24\cdot10^{-3}$	$-2.65\cdot10^{-4}$
17	HEA 140	HEA 140	IPE 400	$2.08\cdot10^{+3}$	$8.28\cdot10^{-8}$	$-2.39\cdot10^{-6}$	$2.77\cdot10^{+3}$	$-5.53\cdot10^{+3}$	$7.18\cdot10^{-3}$	$-2.39\cdot10^{-6}$
18	HEA 200	HEA 200	IPE 240	$2.08\cdot10^{+3}$	$1.76\cdot10^{-7}$	$-6.81\cdot10^{-5}$	$2.21\cdot10^{+4}$	$-4.42\cdot10^{+4}$	$4.24\cdot10^{-3}$	$-6.81\cdot10^{-5}$
19	HEA 140	HEB 160	IPE 300	$1.97\cdot10^{+3}$	$1.58\cdot10^{-7}$	$-1.20\cdot10^{-5}$	$6.69\cdot10^{+3}$	$-1.34\cdot10^{+4}$	$7.21\cdot10^{-3}$	$-1.20\cdot10^{-5}$
20	HEA 120	HEA 120	IPE 360	$1.74\cdot10^{+3}$	$1.07\cdot10^{-7}$	$-2.59\cdot10^{-6}$	$2.34\cdot10^{+3}$	$-4.68\cdot10^{+3}$	$8.92\cdot10^{-3}$	$-2.59\cdot10^{-6}$

Table 5 - Results of program execution for the GA algorithm with iteration number = 20

Num	**001**	**002**	**003**	**Global weight (t)**	**F calculated**	**Arrow max (L/2)**	**MA=MD**	**MB=MC**	**Sigma Column**	**Sigma Beam**
1	HEA 360	HEA 360	IPE 180	$4.37 \cdot 10^{+3}$	$2.40 \cdot 10^{-7}$	$-2.26 \cdot 10^{-4}$	$3.31 \cdot 10^{+4}$	$-6.62 \cdot 10^{+4}$	$1.59 \cdot 10^{-3}$	$-2.26.10^{-4}$
2	HEA 340	HEA 340	IPE 180	$4.12 \cdot 10^{+3}$	$2.41 \cdot 10^{-7}$	$-2.25 \cdot 10^{-4}$	$3.30 \cdot 10^{+4}$	$-6.59 \cdot 10^{+4}$	$1.71 \cdot 10^{-3}$	$-2.25.10^{-4}$
3	HEB 260	HEB 260	IPE 200	$3.75 \cdot 10^{+3}$	$1.95 \cdot 10^{-7}$	$-1.63 \cdot 10^{-4}$	$3.17 \cdot 10^{+4}$	$-6.34 \cdot 10^{+4}$	$1.93 \cdot 10^{-3}$	$-1.63.10^{-4}$
4	HEB 260	HEB 260	IPE 180	$3.69 \cdot 10^{+3}$	$2.50 \cdot 10^{-7}$	$-2.21 \cdot 10^{-4}$	$3.23 \cdot 10^{+4}$	$-6.46 \cdot 10^{+4}$	$1.93 \cdot 10^{-3}$	$-2.21 \cdot 10^{-4}$
5	HEA 260	HEB 260	IPE 200	$3.30 \cdot 10^{+3}$	$2.03 \cdot 10^{-7}$	$-1.59 \cdot 10^{-4}$	$3.09 \cdot 10^{+4}$	$-6.18 \cdot 10^{+4}$	$2.63 \cdot 10^{-3}$	$-1.59 \cdot 10^{-4}$
6	HEB 200	HEB 200	IPE 360	$3.23 \cdot 10^{+3}$	$8.13 \cdot 10^{-8}$	$-1.54 \cdot 10^{-5}$	$1.39 \cdot 10^{+4}$	$-2.78 \cdot 10^{+4}$	$2.91 \cdot 10^{-3}$	$-1.54 \cdot 10^{-5}$
7	HEB 200	HEB 200	IPE 360	$3.23 \cdot 10^{+3}$	$8.13 \cdot 10^{-8}$	$-1.54 \cdot 10^{-5}$	$1.39 \cdot 10^{+4}$	$-2.78 \cdot 10^{+4}$	$2.91 \cdot 10^{-3}$	$-1.54 \cdot 10^{-5}$
8	HEA 160	HEA 450	IPE 80	$3.17 \cdot 10^{+3}$	$1.76 \cdot 10^{-6}$	$-1.65 \cdot 10^{-3}$	$3.30 \cdot 10^{+4}$	$-6.59 \cdot 10^{+4}$	$5.95 \cdot 10^{-3}$	$-1.65 \cdot 10^{-3}$
9	HEB 220	HEA 240	IPE 240	$2.92 \cdot 10^{+3}$	$1.44 \cdot 10^{-7}$	$-8.39 \cdot 10^{-5}$	$2.72 \cdot 10^{+4}$	$-5.44 \cdot 10^{+4}$	$2.51 \cdot 10^{-3}$	$-8.39 \cdot 10^{-5}$
10	HEB 220	HEA 240	IPE 240	$2.92 \cdot 10^{+3}$	$1.44 \cdot 10^{-7}$	$-8.39 \cdot 10^{-5}$	$2.72 \cdot 10^{+4}$	$-5.44 \cdot 10^{+4}$	$2.51 \cdot 10^{-3}$	$-8.39 \cdot 10^{-5}$
11	HEB 220	HEA 240	IPE 220	$2.84 \cdot 10^{+3}$	$1.73 \cdot 10^{-7}$	$-1.14 \cdot 10^{-4}$	$2.88 \cdot 10^{+4}$	$-5.76 \cdot 10^{+4}$	$2.51 \cdot 10^{-3}$	$-1.14 \cdot 10^{-4}$
12	HEB 200	HEB 200	IPE 240	$2.76 \cdot 10^{+3}$	$1.57 \cdot 10^{-7}$	$-7.76 \cdot 10^{-5}$	$2.52 \cdot 10^{+4}$	$-5.03 \cdot 10^{+4}$	$2.93 \cdot 10^{-3}$	$-7.76 \cdot 10^{-5}$
13	HEB 220	HEB 220	IPE 80	$2.68 \cdot 10^{+3}$	$1.70 \cdot 10^{-6}$	$-1.68 \cdot 10^{-3}$	$3.36 \cdot 10^{+4}$	$-6.72 \cdot 10^{+4}$	$2.52 \cdot 10^{-3}$	$-1.68 \cdot 10^{-3}$
14	HEB 220	HEB 220	IPE 80	$2.68 \cdot 10^{+3}$	$1.70 \cdot 10^{-6}$	$-1.68 \cdot 10^{-3}$	$3.36 \cdot 10^{+4}$	$-6.72 \cdot 10^{+4}$	$2.52 \cdot 10^{-3}$	$-1.68 \cdot 10^{-3}$
15	HEB 200	HEB 200	IPE 180	$2.55 \cdot 10^{+3}$	$2.78 \cdot 10^{-7}$	$-2.07 \cdot 10^{-4}$	$3.03 \cdot 10^{+4}$	$-6.05 \cdot 10^{+4}$	$2.93 \cdot 10^{-3}$	$-2.07 \cdot 10^{-4}$
16	HEA 200	HEA 200	IPE 330	$2.41 \cdot 10^{+3}$	$1.05 \cdot 10^{-7}$	$-1.82 \cdot 10^{-5}$	$1.30 \cdot 10^{+4}$	$-2.60 \cdot 10^{+4}$	$4.22 \cdot 10^{-3}$	$-1.82 \cdot 10^{-5}$
17	HEA 220	HEA 220	IPE 240	$2.37 \cdot 10^{+3}$	$1.59 \cdot 10^{-7}$	$-7.65 \cdot 10^{-5}$	$2.48 \cdot 10^{+4}$	$-4.96 \cdot 10^{+4}$	$3.55 \cdot 10^{-3}$	$-7.65 \cdot 10^{-5}$
18	HEA 240	HEA 240	IPE 100	$2.32 \cdot 10^{+3}$	$1.01 \cdot 10^{-6}$	$-9.76 \cdot 10^{-4}$	$3.34 \cdot 10^{+4}$	$-6.68 \cdot 10^{+4}$	$2.98 \cdot 10^{-3}$	$-9.76 \cdot 10^{-4}$
19	HEA 220	HEA 220	IPE 200	$2.22 \cdot 10^{+3}$	$2.27 \cdot 10^{-7}$	$-1.47 \cdot 10^{-4}$	$2.86 \cdot 10^{+4}$	$-5.72 \cdot 10^{+4}$	$3.55 \cdot 10^{-3}$	$-1.47 \cdot 10^{-4}$
20	HEB 180	HEB 180	IPE 180	$2.18 \cdot 10^{+3}$	$2.98 \cdot 10^{-7}$	$-1.97 \cdot 10^{-4}$	$2.88 \cdot 10^{+4}$	$-5.76 \cdot 10^{+4}$	$3.51 \cdot 10^{-3}$	$-1.97 \cdot 10^{-4}$

The results depend on the number of iterations for both algorithms, and Figure 7 illustrates the structure weight optimization for a population of 20 iterations. Clearly, the BBO algorithm produced a better optimal structure than the GA algorithm.

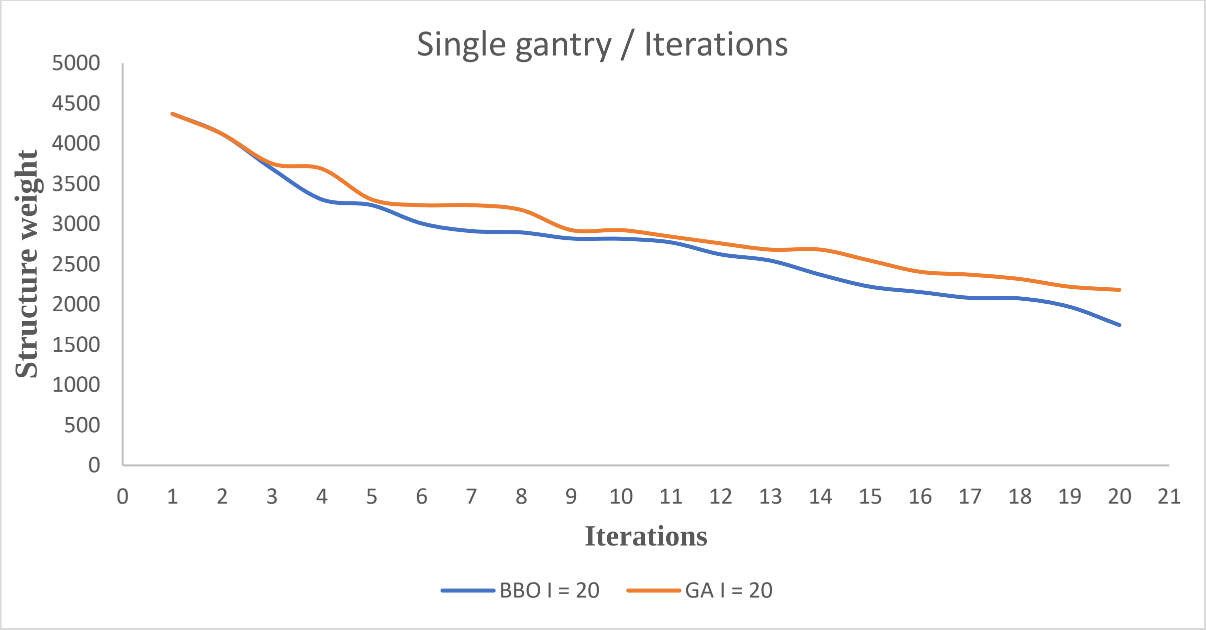

Figure 7 - Comparison of structure weight optimization between BBO and GA algorithms for iteration I = 20

As with the first criterion, to validate this comparison between the two algorithms correctly, it was necessary to take several iteration values ranging from 20 to 200. At the end of the algorithms' execution, Figure 8 illustrates this comparison between the BBO and GA algorithms for structure weight optimization, taking iteration values of 20, 50, 100, 150 and 200. Despite the change in the number of iterations and similar to the first criterion, the BBO algorithm still gives better results than the GA algorithm, and the figure clearly shows that it improves these results with each iteration.

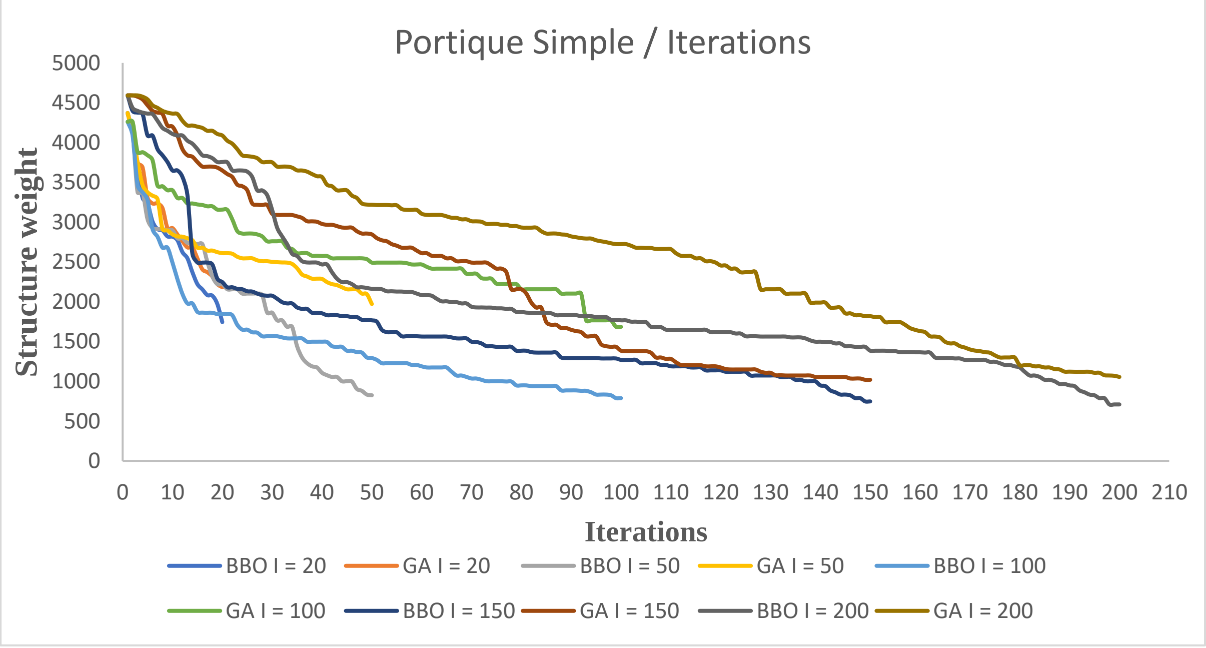

Figure 8 - Comparison of structure weight optimization between BBO and GA algorithms for iteration ranging from I = 20 to I = 200

The GA algorithm is the fastest, but the BBO algorithm gives better optimal structures, and the problem with the GA algorithm is often to loop in the local minimum, unlike the BBO algorithm, which seeks the optimal structure without looping in the local minimum. Clearly, the initial population has a key impact on determining the optimal structure, but increasing the number of iterations has no influence on the final result, especially when the number of populations is high.

5. Conclusions

It is clear that the applications proposed in this study offer optimal solutions for the pre-dimensioning of steel structures while complying with Eurocode 3 standards (Union, 2006), making them a valuable decision-making tool for future users. The idea of introducing optimization algorithms such as BBO and GA algorithms into applications designed to solve civil engineering problems is essential to help the user optimize the cost of the structure to be adopted, and consequently reduce time and errors in the preliminary phase of the design study.

This study investigated the influence of the number of populations and the number of iterations on the final results. The number of iterations parameter gave a more refined result than the population number parameter, because changing the initial population causes the program to retrograde to zero each time it starts execution, whereas changing the number of iterations allows continuity in execution, using the population found in the previous iteration. In this study, both algorithms gave reliable results, but the BBO algorithm converges faster than the GA algorithm, because its execution procedure is continuous and the search for the global minimum by the BBO algorithm is executed without blocking, unlike the GA algorithm, where blocking is sometimes possible. The results of this study show that the BBO algorithm is an effective tool for the optimization of civil engineering structures. The BBO algorithm is able to find solutions that are lighter, stiffer, and have a lower deflection than the original designs.

The BBO algorithm is a promising new tool for the optimization of civil engineering structures. The BBO algorithm is able to find solutions that are more efficient and effective than the traditional trial-and-error approach.

References

Amamra, L., Benanane, S., Benanane, A., Bourdim, S. M., Titoum, M., & Chaibedra, B. (2020). The concepts of genetics applied to a reinforced concrete cantilever beam optimization. *Advances in Science, Technology & Innovation*, 309–315. https://doi.org/10.1007/978-3-030-35533-3_37

Arora, J. S. (1997). Guide to structural optimization. *Manuals and Reports on Engineering Practice, American Society of Civil Engineers* (Issue 90). https://doi.org/10.1061/9780784402207

Belegundu, A. D., & Chandrupatla, T. R. (2011). Optimization Concepts and Applications in Engineering, Second Edition. *Optimization Concepts and Applications in Engineering, Second* Edition.

Camp, C., Pezeshk, S., & Cao, G. (1998). Optimized Design of Two-Dimensional Structures Using a Genetic Algorithm. *Journal of Structural Engineering*, *124*(5). https://doi.org/10.1061/(asce)0733-9445(1998)124:5(551)

Camp, C. V., & Bichon, B. J. (2004). Design of Space Trusses Using Ant Colony Optimization. *Journal of Structural Engineering*, *130*(5). https://doi.org/10.1061/(asce)0733-9445(2004)130:5(741)

Doan, E., & Saka, M. P. (2012). Optimum design of unbraced steel frames to LRFD-AISC using particle swarm optimization. *Advances in Engineering Software*, *46*(1). https://doi.org/10.1016/j.advengsoft.2011.05.008

Dorigo, M., Birattari, M., & Stützle, T. (2006). Ant colony optimization artificial ants as a computational intelligence technique. *IEEE Computational Intelligence Magazine*, *1*(4). https://doi.org/10.1109/CI-M.2006.248054

Ebenau, G., Rottschäfer, J., & Thierauf, G. (2005). An advanced evolutionary strategy with an adaptive penalty function for mixed-discrete structural optimisation. *Advances in Engineering Software*, *36*(1). https://doi.org/10.1016/j.advengsoft.2003.10.008

Erbatur, F., Hasançebi, O., Tütüncü, İ., & Kılıç, H. (2000). Optimal design of planar and space structures with genetic algorithms. *Computers and Structures*, *75*(2). https://doi.org/10.1016/S0045-7949(99)00084-X

Fourie, P. C., & Groenwold, A. A. (2002). The particle swarm optimization algorithm in size and shape optimization. *Structural and Multidisciplinary Optimization*, *23*(4). https://doi.org/10.1007/s00158-002-0188-0

Hasançebi, O. (2007a). Discrete approaches in evolution strategies based optimum design of steel frames. *Structural Engineering and Mechanics*, *26*(2). https://doi.org/10.12989/sem.2007.26.2.191

Hasançebi, O. (2007b). Optimization of truss bridges within a specified design domain using evolution strategies. *Engineering Optimization*, *39*(6). https://doi.org/10.1080/03052150701335071

Hasançebi, O., Erdal, F., & Saka, M. P. (2010). Adaptive Harmony Search Method for Structural Optimization. *Journal of Structural Engineering*, *136*(4). https://doi.org/10.1061/(asce)st.1943-541x.0000128

Holland, J. H. (1992). Genetic Algorithms. *Scientific American*, *267*(1), 66–72. https://doi.org/10.1038/scientificamerican0792-66

Kaveh, A., & Abdietehrani, A. (2004). Design of frames using genetic algorithm, force method and graph theory. *International Journal for Numerical Methods in Engineering*, *61*(14). https://doi.org/10.1002/nme.1170

Kaveh, A., & Rahami, H. (2006). Analysis, design and optimization of structures using force method and genetic algorithm. *International Journal for Numerical Methods in Engineering*, *65*(10). https://doi.org/10.1002/nme.1506

Kaveh, A., & Shojaee, S. (2007). Optimal design of skeletal structures using ant colony optimization. *International Journal for Numerical Methods in Engineering*, *70*(5). https://doi.org/10.1002/nme.1898

Kaveh, A., & Talatahari, S. (2010). An improved ant colony optimization for the design of planar steel frames. *Engineering Structures*, *32*(3). https://doi.org/10.1016/j.engstruct.2009.12.012

Kennedy, J., & Eberhart, R. (1995). Particle swarm optimization. *Proceedings of ICNN'95 - International Conference on Neural Networks*, 1942–1948. https://doi.org/10.1109/ICNN.1995.488968

Lee, K. S., & Geem, Z. W. (2004). A new structural optimization method based on the harmony search algorithm. *Computers and Structures*, *82*(9–10). https://doi.org/10.1016/j.compstruc.2004.01.002

Li, L. J., Huang, Z. B., Liu, F., & Wu, Q. H. (2007). A heuristic particle swarm optimizer for optimization of pin connected structures. *Computers & Structures*, *85*(7–8), 340–349. https://doi.org/10.1016/j.compstruc.2006.11.020

Ma, H., Fei, M., Simon, D., & Chen, Z. (2015). Biogeography-based optimization in noisy environments. *Transactions of the Institute of Measurement and Control*, *37*(2). https://doi.org/10.1177/0142331214537015

Malik, M. R. S., Mohideen, E. R., & Ali, L. (2016). Weighted distance Grey Wolf optimizer for global optimization problems. *2015 IEEE International Conference on Computational* Intelligence *and Computing Research, ICCIC 2015*. https://doi.org/10.1109/ICCIC.2015.7435714

Perez, R. E., & Behdinan, K. (2007). Particle swarm approach for structural design optimization. *Computers and Structures*, *85*(19–20). https://doi.org/10.1016/j.compstruc.2006.10.013

Pezeshk, S., Camp, C. V., & Chen, D. (2000). Design of Nonlinear Framed Structures Using Genetic Optimization. *Journal of Structural Engineering*, *126*(3), 382–388. https://doi.org/10.1061/(ASCE)0733-9445(2000)126:3(382)

Rahami, H., Kaveh, A., & Gholipour, Y. (2008). Sizing, geometry and topology optimization of trusses via force method and genetic algorithm. *Engineering Structures*, *30*(9). https://doi.org/10.1016/j.engstruct.2008.01.012

Rajasekaran, S., Mohan, V. S., & Khamis, O. (2004a). The optimisation of space structures using evolution strategies with functional networks. *Engineering with Computers*, *20*(1), 75–87. https://doi.org/10.1007/s00366-004-0268-4

Rajasekaran, S., Mohan, V. S., & Khamis, O. (2004b). The optimisation of space structures using evolution strategies with functional networks. *Engineering with Computers*, *20*(1), 75–87. https://doi.org/10.1007/s00366-004-0268-4

Rajeev, S., & Krishnamoorthy, C. S. (1992). Discrete Optimization of Structures Using Genetic Algorithms. *Journal of Structural Engineering*, *118*(5). https://doi.org/10.1061/(asce)0733-9445(1992)118:5(1233)

Reeves, C. R., & Rowe, J. E. (2002). Genetic algorithms: Principles and perspectives: A Guide to GA theory. *Operations Research/ Computer Science Interfaces Series*, *20*.

Saka, M. P. (2009). Optimum design of steel sway frames to BS5950 using harmony search algorithm. *Journal of Constructional Steel Research*, *65*(1). https://doi.org/10.1016/j.jcsr.2008.02.005

Saka, M. P., Aydogdu, I., Hasancebi, O., & Geem, Z. W. (2011). Harmony search algorithms in structural engineering. *Studies in Computational Intelligence*, *359*. https://doi.org/10.1007/978-3-642-20986-4_6

Saka, M. P., & Hasançebi, O. (2009). Adaptive harmony search algorithm for design code optimization of steel structures. *Studies in Computational Intelligence*, *239*. https://doi.org/10.1007/978-3-642-03450-3_3

Saka, M. P., & Kameshki, E. S. (1998). Optimum design of multi-storey sway steel frames to BS5950 using a genetic algorithm. *Advances in Engineering Computational Technology*, 135–141.

Schmit, L. A. (1960). Structural Design by Systematic Synthesis. *Proceedings, 2nd Conference on Electronic Computation, ASCE*.

Shook, D. A., Roschke, P. N., Lin, P. Y., & Loh, C. H. (2008). GA-optimized fuzzy logic control of a large-scale building for seismic loads. *Engineering Structures*, *30*(2). https://doi.org/10.1016/j.engstruct.2007.04.008

Simon, D. (2008). Biogeography-Based Optimization. *IEEE Transactions on Evolutionary Computation*, *12*(6), 702–713. https://doi.org/10.1109/TEVC.2008.919004

Union, T. E. (2006). EN 1993-1-1 (2005) (English): Eurocode 3: Design of steel structures - Part 1-1: General rules and rules for buildings. *Concrete (London)*, *40*(1).