Academic Journal of Nawroz University (AJNU), Vol.12, No.4, 2023
This is an open access article distributed under the Creative Commons Attribution License
Copyright ©2017. e-ISSN: 2520-789X
https://doi.org/10.25007/ajnu.v12n4a1903

Exploring the Impact of Big Bang-Big Crunch Algorithm Parameters on Welded Beam Design Problem Resolution

Saman M. Almufti

Department of Computer Science, College of Science, Nawroz University, Duhok, KRG - Iraq

ABSTRACT

A Metaheuristic Optimization is a group of algorithms that are widely studied and employed in the scientific literature. Typically, metaheuristics algorithms utilize stochastic operators that make each iteration unique, and they frequently contain controlling parameters that have an impact on the convergence process since their impacts are mostly neglected in most optimization literature, making it difficult to draw conclusions. This paper introduced the Big Bang-Big Crunch (BB-BC) metaheuristic algorithm to evaluate the performance of a metaheuristic algorithm in relation to its control parameter. It also demonstrates the effects of varying the values of BB-BC in solving. The "Welded Beam Design problem" is a well-known engineering optimization problem that is classified as a Single-Objective Constrained Optimization issue. Multiple starting parameter values for the BB-BC are evaluated as part of the experimental findings. This is done in an attempt to find the algorithm's optimal starting settings. The lowest, maximum, and mean values of the penalized objective functions are then computed. Finally, the BB-BC results are compared with various metaheuristics algorithms.

Keywords Optimization Algorithms, Big Bang–Big Crunch, Welded Beam Design problem, Constrained Optimization.

1. Introduction

Optimization algorithms play a critical role in addressing difficult engineering design issues by effectively searching the search space and locating optimal solutions (Kaveh & Bakhshpoori, 2019). The Welded Beam Design problem is a well-known benchmark problem in structural engineering that attempts to calculate the ideal size of a welded beam given various constraints. The goal is to identify the beam size that reduce weight while meeting stress, deflection, and beam stability restrictions (Almufti, 2022a) .

Due to their proficiency in solving challenging optimization issues, nature-inspired optimization algorithms have attracted a lot of interest recently. A good example of one of these algorithms is the Big Bang-Big Crunch (BB-BC) algorithm, which was motivated by the cosmological phenomena of the universe's expansion (Big Bang) and contraction (Big Crunch) (Mbuli & Ngaha, 2022). In a number of optimization applications, such as challenges with engineering design, the BB-BC method has demonstrated promising results(Prayogo et al., 2018).

The objective of this paper is to evaluate the parameters of the BB-BC algorithm in the context of resolving the Welded Beam Design problem. By conducting a systematic analysis of these parameters, this paper aim to identify the optimal parameter settings that lead to improved convergence and solution quality. This investigation can provide valuable insights into the performance of the BB-BC algorithm and its applicability to engineering design optimization.

The remainder of this paper is structured as follows: Section 2, provides a comprehensive review of the literature on the BB-BC algorithm and its applications. Section 3 introduces the Constrained Optimization Problem, Section 4 presents the formulation of the Welded Beam Design problem

Section 5 introduces the fundamental concepts of the BB-BC algorithm, highlighting its key features and stages. And explains how it can be used for solving Welded Beam Design problem. The subsequent section, Section 6, outlines the methodology used to evaluate the BB-BC algorithm's parameters, followed by a discussion of the results. Finally, Section 6 offers concluding remarks, summarizes the key findings.

By investigating the parameter evaluation of the BB-BC algorithm for the Welded Beam Design problem, this research aims to contribute to the broader field of nature-inspired optimization algorithms and their application in solving engineering design problems. Furthermore, the insights gained from this study can aid engineers and practitioners in selecting appropriate parameter settings for optimizing welded beam designs, ultimately leading to more efficient and reliable structural designs.

2. Literature review

Optimization algorithms play a vital role in solving complex engineering design problems by efficiently exploring the search space and identifying optimal solutions. Traditional techniques (Ihsan et al., 2021; Sadeeq et al., 2021a), such as gradient-based methods and evolutionary algorithms like Genetic Algorithms (GA)(M. Almufti et al., 2019) and Particle Swarm Optimization (PSO) (M. Almufti et al., 2019), have been widely employed. However, these algorithms often face challenges when dealing with high-dimensional, nonlinear, and multi-modal optimization problems(Almufti et al., 2018; Asaad & Abdulnabi, 2018).

The Big Bang-Big Crunch (BB-BC) algorithm is a nature-inspired optimization algorithm that draws inspiration from the cosmological phenomena of the expansion and contraction of the universe. Introduced by E. Erol and E. Şehirlioğlu in 2006, the BB-BC algorithm has shown promising results in various optimization applications (Tang et al., 2010). At its core, the algorithm employs a population of candidate solutions that undergo successive stages, including initialization, exploration, and exploitation. The algorithm seeks to iteratively improve the population through a balance of exploration and exploitation, mimicking the expansion and contraction of the universe(Jahwar et al., 2021; Sadeeq et al., 2021b).

Previous applications of the BB-BC algorithm have demonstrated its effectiveness in solving optimization problems in diverse domains, such as engineering, economics, and data mining. For instance, in structural engineering, the BB-BC algorithm has been employed to optimize truss structures, frame designs, and other architectural configurations. These studies have reported favorable results in terms of convergence speed, solution quality, and robustness(Mbuli & Ngaha, 2022).

The Welded Beam Design problem is a classic optimization benchmark in structural engineering, aiming to determine the optimal dimensions of a welded beam that minimizes weight while satisfying constraints related to stress, deflection, and stability. Traditional optimization techniques, including mathematical programming and gradient-based methods, have been applied to this problem. However, they often struggle with the problem's complexity, nonlinearity, and multiple objectives(Almufti, 2022a).

The BB-BC algorithm has shown promise in various engineering design problems beyond the Welded Beam Design problem. Several studies have applied the BB-BC algorithm to optimize truss structures, frame configurations, and composite materials, among others. These applications have demonstrated improved convergence rates, solution quality, and robustness compared to traditional optimization techniques and other metaheuristics (Kaveh & Bakhshpoori, 2019). While the BB-BC algorithm has shown promise in optimization problems, including engineering design, there is a lack of comprehensive studies investigating the parameter sensitivity and performance of the algorithm in specific engineering scenarios. Moreover, its application to the Welded Beam Design problem remains relatively unexplored.

Therefore, this paper aims to bridge these research gaps by conducting a parameter evaluation of the BB-BC algorithm for resolving the Welded Beam Design problem. By systematically analyzing the algorithm's parameters and their impact on convergence and solution quality, the paper seek to provide valuable insights into the performance of the BB-BC algorithm in this context. This investigation can contribute to the understanding of the algorithm's effectiveness, parameter sensitivity, and its potential for enhancing the optimization of welded beam designs in structural engineering.

According to Almufti(M. Almufti, 2019; M. Almufti et al., 2023), there were more than 200 Metaheuristic algorithms have been developed to address a wide range of practical problems.

Indeed, in addition to the Big Bang-Big Crunch (BB-BC) algorithm, several other metaheuristic algorithms have been employed to tackle the Welded Beam Design problem. These algorithms aim to optimize the dimensions of the welded beam while satisfying the given constraints. Here are a few notable metaheuristic algorithms that have been applied to this problem:

1. Genetic Algorithms (GA):

Genetic algorithms are widely recognized and extensively used metaheuristic algorithms for optimization problems. They are inspired by the process of natural selection and evolution. In the context of the Welded Beam Design problem, genetic algorithms generate a population of potential solutions and iteratively evolve them by applying genetic operators such as selection, crossover, and mutation. Fitness-based selection mechanisms drive the convergence towards improved solutions (Yokota et al., 1999).

2. Particle Swarm Optimization (PSO):

Particle Swarm Optimization is a population-based optimization algorithm inspired by the social behavior of bird flocks or fish schools. In PSO, a population of particles moves through the search space, searching for the optimal solution. Each particle adjusts its position based on its own experience and the best-known position of the swarm. PSO has been applied to the Welded Beam Design problem to explore and exploit the search space efficiently (Kamil et al., 2021).

3. Simulated Annealing (SA):

Simulated Annealing is a stochastic optimization algorithm that simulates the annealing process in metallurgy. It starts with an initial solution and iteratively explores the search space by accepting both better and worse solutions based on a probabilistic criterion. Initially, the algorithm allows a higher acceptance rate for worse solutions, enabling it to escape local optima. Over time, the acceptance rate decreases, and the algorithm converges towards the global optimum(Christu Nesam David. D & S. Elizabeth Amudhini Stephen, 2018).

4. Harmony Search (HS):

Harmony Search is a music-inspired metaheuristic algorithm that mimics the improvisation process in music composition. It maintains a population of candidate solutions called "harmonies" and iteratively improves them. In the context of the Welded Beam Design problem, harmony search algorithms explore and refine the search space by adjusting the values of the design variables to optimize the objective function while satisfying the given constraints (O. K. Erol et al., 2011).

These metaheuristic algorithms, along with the BB-BC algorithm, have been applied to the Welded Beam Design problem with varying degrees of success. Each algorithm has its own strengths and weaknesses, and their performance can be influenced by factors such as parameter settings, problem complexity, and the nature of the objective and constraint functions.

Comparative studies that assess the performance of these algorithms on the Welded Beam Design problem have provided valuable insights into their capabilities. Such studies have examined factors like convergence speed, solution quality, robustness, and computational efficiency to evaluate the effectiveness of different metaheuristic algorithms in addressing the problem.

By conducting a parameter evaluation of the BB-BC algorithm in the context of the Welded Beam Design problem, this research aims to contribute to the existing body of knowledge by providing a comprehensive understanding of the algorithm's performance and comparing it with other metaheuristic algorithms.

3. Constrained Optimization Problem

Constrained optimization problems involve finding the best solution to an optimization objective while satisfying a set of constraints. Unlike unconstrained optimization problems where only the objective function needs to be considered, constrained optimization problems require considering the feasibility of solutions within the constraints imposed by the problem(Almufti, 2022b).

The general form of a constrained optimization problem can be expressed as follows, Eq(1):

$$\begin{gathered}\text{Minimize (or maximize) } f(x) \\ \text{Subject to } g(x) \leq 0 \\ h(x) = 0 \\ lb \leq x \leq ub\end{gathered} \tag{1}$$

Where:
- f(x) is the objective function that needs to be minimized or maximized.
- g(x) represents inequality constraints that need to be satisfied, where g(x) ≤ 0.
- h(x) represents equality constraints that need to be satisfied, where h(x) = 0.
- x represents the vector of decision variables.
- lb and ub represent the lower and upper bounds for the decision variables.

In a constrained optimization problem, the objective is to find a solution that minimizes (or maximizes) the objective function while ensuring that all the constraints are satisfied. The constraints can represent various requirements or limitations that the solution must adhere to, such as physical constraints, resource constraints, or design specifications.
Solving a constrained optimization problem involves searching for feasible solutions within the constraints and evaluating their objective function values. Various optimization algorithms, including metaheuristics and mathematical programming techniques, can be used to tackle constrained optimization problems. These algorithms aim to iteratively explore the solution space, refine candidate solutions, and converge towards the best feasible solution(Parsopoulos & Vrahatis, 2005).
The performance evaluation of algorithms in constrained optimization often involves considering factors such as feasibility, optimality, convergence speed, robustness, and computational efficiency. Additionally, techniques like penalty functions, constraint handling mechanisms, or optimization formulations may be employed to handle the constraints effectively during the optimization process.(M. Zhang et al., 2008)
Constrained optimization problems are prevalent in various fields, including engineering design, finance, operations research, and machine learning. Examples of constrained optimization problems include structural design optimization, portfolio optimization with investment constraints, scheduling problems with resource limitations, and parameter estimation with inequality constraints(Almufti, 2022c).

By addressing the challenges posed by constraints, constrained optimization problems provide a framework for finding optimal solutions while considering the real-world limitations and requirements imposed by the problem domain.
Generally, Constrained optimization problems can be broadly categorized into two main categories:

a) Constrained Unimodal Optimization Problems:
Constrained unimodal optimization problems refer to those problems where the objective function has a single optimal solution, and the feasible region defined by the constraints is unimodal. In other words, there is only one global optimum in the search space, and the constraints do not create multiple feasible regions.
Solving constrained unimodal optimization problems involves finding the global optimum within the feasible region. Algorithms used for solving these problems typically aim to converge to the global optimum by balancing exploration and exploitation in the search space(Almufti, 2022b). Examples of constrained unimodal optimization problems include linear programming (LP) and quadratic programming (QP) problems.

b) Constrained Multimodal Optimization Problems:
Constrained multimodal optimization problems are characterized by having multiple local optima within the feasible region defined by the constraints. The objective function may have multiple peaks, valleys, or distinct regions of interest, making it challenging to find the global optimum.
Solving constrained multimodal optimization problems requires exploration of the search space to discover multiple local optima while still ensuring feasibility within the constraints. Algorithms used for

solving these problems need to balance between local search intensification to refine local optima and global search diversification to discover different regions of interest. Metaheuristic algorithms, such as genetic algorithms, particle swarm optimization, or simulated annealing, are commonly applied to address constrained multimodal optimization problems(Almufti, 2022b).

The categorization into constrained unimodal and constrained multimodal optimization problems provides a high-level distinction based on the nature of the search space and the presence of multiple optima. However, within each category, there can be various subtypes and specific problem instances that require tailored solution approaches.

It's important to consider the characteristics of the optimization problem at hand when selecting appropriate algorithms and techniques. Factors such as problem complexity, dimensionality, the number of constraints, the behavior of the objective function, parameters, encoding schemes, and fitness evaluation techniques should be taken into account to choose the most suitable optimization methodology. The most well-known examples of engineering constrained optimization problems that can be effectively solved by metaheuristic algorithms can be classified as:

a) Truss Optimization: Optimal design of truss structures considering constraints such as stress, displacement, and buckling limits. Metaheuristic algorithms like genetic algorithms, particle swarm optimization, and simulated annealing have been successfully applied to solve truss optimization problems (Öztürk & Kahraman, 2023).

b) Shape Optimization: Optimizing the shape of structures or components, such as airfoils, turbine blades, or car bodies, considering constraints on aerodynamic performance, structural integrity, and manufacturing limitations. Metaheuristic algorithms like genetic algorithms and evolutionary strategies have been employed to tackle shape optimization problems.

c) Facility Layout Optimization: Determining the optimal layout of facilities in a manufacturing plant or a warehouse to minimize material handling costs, improve workflow, and ensure operational efficiency. Metaheuristic algorithms such as particle swarm optimization, ant colony optimization, and tabu search have been applied to solve facility layout optimization problems.

d) Process Scheduling: Optimal scheduling of processes in manufacturing systems, chemical plants, or assembly lines to minimize makespan, energy consumption, or resource utilization while satisfying sequencing constraints, resource availability, and production targets. Metaheuristic algorithms like genetic algorithms, particle swarm optimization, and simulated annealing have proven effective in process scheduling problems.

e) Portfolio Optimization: Determining the optimal allocation of investments in a financial portfolio while considering constraints such as risk tolerance, asset allocation limits, and return objectives. Metaheuristic algorithms, including genetic algorithms, particle swarm optimization, and evolutionary algorithms, have been employed to solve portfolio optimization problems.

f) Vehicle Routing: Optimizing the routes of delivery vehicles to minimize travel distance, time, or fuel consumption, while satisfying constraints such as vehicle capacity, time windows, and customer demands. Metaheuristic algorithms like genetic algorithms, ant colony optimization, and tabu search have been widely used to solve vehicle routing problems.

These examples highlight the diverse range of engineering constrained optimization problems that can be effectively tackled using metaheuristic algorithms. Metaheuristic algorithms offer the advantage of handling complex optimization landscapes, nonlinearity, and multiple constraints. Their ability to balance exploration and exploitation allows for the discovery of high-quality solutions in a reasonable amount of time.

4. Welded Beam Design (WBD)

Welded Beam Design that is shown in figure 1, is an engineering Single-Objectives Constrained Optimizations Benchmark Problems. It involves designing a welded beam with the lowest possible cost while taking into account side limitations, shear stress(τ), bending stress, buckling (σ), load on the bar

(Pc), and end deflection (δ). Four variables make up the design: h (x1), l (x2), t (x3), and b (x4). This issue may be expressed quantitatively as follows(Almufti, 2022a; Deb, 1991):

$$\min f(x) = 1.10471{x_1}^2x_2 + 0.04811x_3x_4(14.0 + x_2) \quad (2)$$

$$\text{s.t.}\, g_1(x) = \tau(x) - \tau_{max} \leq 0 \quad (3)$$

$$g_2(x) = \sigma(x) - \sigma_{max} \leq 0 \quad (4)$$

$$g_3(x) = x_1 - x_4 \leq 0 \quad (5)$$

$$g_4(x) = 0.10471{x_1}^2 + 0.04811x_3x_4(14.0 + x_2) - 5.0 \leq 0 \quad (6)$$

$$g_5(x) = 0.125 - x_1 \leq 0 \quad (7)$$

$$g_6(x) = \delta(x) - \delta_{max} \leq 0 \quad (8)$$

$$g_6(x) = P - P_c(x) \leq 0 \quad (9)$$

$$\text{Where } \tau(x) = \sqrt{\left(\tau'\right)^2 + 2\tau'\tau''\frac{x_2}{2R} + (\tau'')^2} \quad (10)$$

$$\tau' = \frac{P}{2^{0.5}\, x_1x_2} \quad (11)$$

$$\tau'' = \frac{MR}{J} \quad (12)$$

$$M = P\left(L + \frac{x_2}{2}\right) \quad (13)$$

$$R = \sqrt{\frac{x_2^2}{4} + \left(\frac{x_1 + x_3}{2}\right)^2} \quad (14)$$

$$J = 2\left\{2^{0.5}x_1x_2\left[\frac{x_2^2}{12} + \left(\frac{x_1 + x_3}{2}\right)\left(\frac{x_1 + x_3}{2}\right)\right]\right\} \quad (15)$$

$$\sigma(x) = \frac{6PL}{x_4x_3^2} \quad (16)$$

$$\delta(x) = \frac{4PL^3}{Ex_3^3x_4} \quad (17)$$

$$P_c(x) = \frac{4.013E\sqrt{\frac{x_3^2x_4^6}{36}}}{L^2}\left(1 - \frac{x_3}{2L}\sqrt{\frac{E}{4G}}\right) \quad (18)$$

Where $P = 6000\text{lb}, L = 14 \text{ in}, E = 30 \text{ x } 106 \text{ psi}, G = 12 \text{ x } 106 \text{ psi}, \tau max = 13{,}600 \text{ psi}, \sigma max = 30{,}000 \text{ psi}, \delta max = 0.25 \text{ in}, 0.1 \leq x1 \leq 2, 0.1 \leq x2 \leq 10, 0.1 \leq x3 \leq 10, 0.1 \leq x4 \leq 2.$

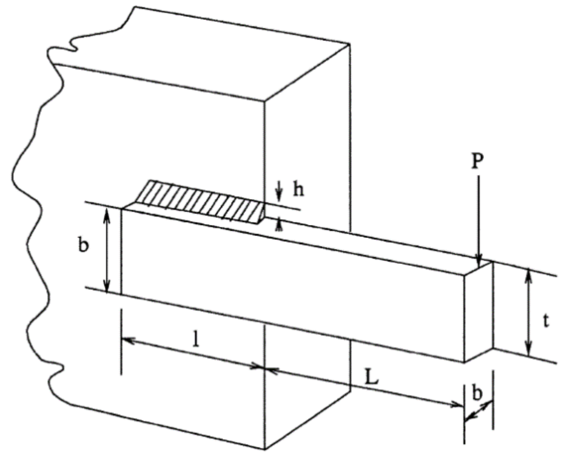

Figure 1:Welded Beam Design

Several approaches and optimization techniques have been employed to address the Welded Beam Design Problem. These include mathematical programming techniques, such as linear programming, nonlinear programming, or mixed-integer programming, as well as metaheuristic algorithms like genetic algorithms, particle swarm optimization, and simulated annealing(Eesa et al., 2023).

Previous approaches have provided valuable insights into the problem and have demonstrated varying degrees of success in finding optimal or near-optimal solutions. However, there is still room for further exploration and improvement, especially by evaluating the effectiveness of the Big Bang-Big Crunch (BB-BC) algorithm for resolving the Welded Beam Design Problem. The BB-BC algorithm's ability to balance exploration and exploitation makes it a promising candidate for optimizing the welded beam design(Dalirinia et al., 2023).

5. Overview of Big Bang-Big Crunch Algorithm

The Big Bang-Big Crunch (BB-BC) algorithm is a nature-inspired optimization algorithm that imitates the cosmological events of the universe's expansion (Big Bang) and contraction (Big Crunch). The BB-BC algorithm was firstly introduced for solving

continuous optimization problems in 2006 by Erol and Eksin (Erol & Eksin, 2006), it aims to discover optimal or near-optimal solutions by balancing exploration and exploitation. It achieves this by iteratively improving a population of candidate solutions.

The BB-BC algorithm begins with an initial population of candidate solutions, and through iterative stages, it explores the search space, refines the solutions, and converges towards optimal or near-optimal solutions(Goel et al., 2023; Mbuli & Ngaha, 2022; Tang et al., 2010).

Step 1. Initialization:

In the initialization stage, the BB-BC algorithm generates an initial population of candidate solutions. This population represents the potential solutions to the Welded Beam Design problem. The initialization process can adopt various strategies such as random sampling or Latin hypercube sampling. The goal is to create a diverse set of candidate solutions that covers the search space effectively.

Step 2. Exploration:

During the exploration stage, the BB-BC algorithm focuses on exploring the search space to discover new and potentially promising solutions. It introduces random perturbations or employs local search techniques to explore the neighborhood of the candidate solutions. This exploration process allows the algorithm to discover regions of the search space that might contain better solutions.

By exploring the search space, the BB-BC algorithm aims to escape local optima and reach areas with more promising solutions. This helps in maintaining diversity within the population and provides a chance to find global or near-global optima.

Step 3. Exploitation:

In the exploitation stage, the BB-BC algorithm intensifies its search around promising solutions to further improve their quality. It applies solution refinement techniques or local optimization methods to exploit the potential of these solutions. By focusing on exploiting the most favorable regions of the search space, the algorithm aims to converge towards optimal or near-optimal solutions.

The exploitation stage allows the BB-BC algorithm to refine the solutions and concentrate on areas that show the most potential for improving the objective function value. This step is crucial for enhancing the overall solution quality and convergence rate.

Step 4. Fitness Evaluation:

The BB-BC algorithm evaluates the fitness of candidate solutions using the objective function and constraints of the Welded Beam Design problem. The fitness evaluation process calculates the performance measure for each candidate solution based on how well it satisfies the objectives and constraints. This measure quantifies the quality or suitability of each solution within the population.

The fitness evaluation guides the search process by providing a comparative measure of the candidate solutions. It helps the algorithm to assess the potential of each solution and make informed decisions during the exploration and exploitation stages.

Step 5. Iterative Improvement:

The BB-BC algorithm performs iterative improvement to gradually enhance the quality of solutions over successive generations. It updates the population of candidate solutions based on their fitness values. Solutions with higher fitness, indicating better performance, have a higher chance of being selected for the next generation.

Through iterative improvement, the BB-BC algorithm converges towards optimal or near-optimal solutions. The algorithm continues the exploration and exploitation stages, evaluating fitness, and updating the population until a termination condition is met, such as reaching a maximum number of iterations or a desired solution quality threshold.

Step 6. Algorithmic Parameters:

The BB-BC algorithm has several parameters that can significantly influence its performance. Some important parameters include population size, maximum number of iterations, mutation rate (perturbation level), local search strategy, and selection mechanisms. These parameters control the balance between exploration and exploitation, the convergence speed, and the algorithm's ability to escape local optima.

Practically, Each particle is a member of the algorithm's population or a potential answer. In the Big Bang phase, a certain number of particles are repeatedly updated in the search space with step sizes based on the convergence operators of the Big Crunch phase in an effort to reach the problem's overall optimal solution(Sharma & Singh, 2022).

Similar to earlier population-based metaheuristics, the Big Bang-Big Crunch (BB-BC) approach starts with a set of randomly generated starting solutions in the search space. A converging operator is formed in the first phase of the algorithm cycle, known as the Big Crunch, and then particles in the search space are updated with step sizes in the region of the converging operator created in the second phase, known as the Big Bang. Consider a population or candidate solutions matrix (P) with a given number of particles (nP), its associated penalized objective function (PFit), and the best observed particle in each iteration (bestP) with the least penalized objective function value. [2].

The weighted average of candidate solution positions, sometimes referred to as the center of mass (CM) or the position of the best candidate solution (bestP), can be used to define the convergence operator based on the Big Crunch phase. CM is stated as follows for minimization problems.(A. Ghasemi & Mirzavand, 2014):

$$CM(i) = \sum_{j=1}^{nP} \left(\frac{P(j,i)}{PFit(j)} \right) / \sum_{j=1}^{nP} \left(\frac{1}{PFit(j)} \right), \qquad i = 1, \ldots, nv \tag{19}$$

At the beginning of the Big Bang phase. Particles are simply updated in the original BB-BC with respect to the previously determined center of mass (CM) or location of the best particle (bestP) by shifting by a random fraction of the permissible step size denoted by the upper (Ub) and lower (Lb) limits of design variables:

$$newP = (CM \text{ or } bestP) + \frac{rand * (Ub - Lb)}{nIT} \tag{20}$$

where rand is a uniformly distributed random number (0, 1). The step size is also divided by the number of algorithm iterations or number of Big Bang phases (NITs) in order to establish the effective search range around the global optimum or center of mass in order to restrict the search as the algorithm progresses. Clearly, the method contains two parameters that are necessary for all metaheuristics: the population size and the maximum number of algorithm iterations as a stopping criterion. Camp provided a novel formulation with two extra parameters for the Big Bang phase and demonstrated its effectiveness. The revised formulation is as follows(Camp, 2007):

$$newP(i) = (\beta * CM + (1 - \beta) * bestP + \frac{rand * \alpha * (Ub - Lb)}{nIT}, \quad i = 1, \ldots, nP \tag{21}$$

This updated formulation is used and encoded inside this section. Notably, the BB-BC algorithm does not need a replacement approach. In other words, particles depart their place regardless of whether their present position is advantageous.

The pseudo code of the method is supplied below, and the BB-BC flowchart is seen in Fig2.

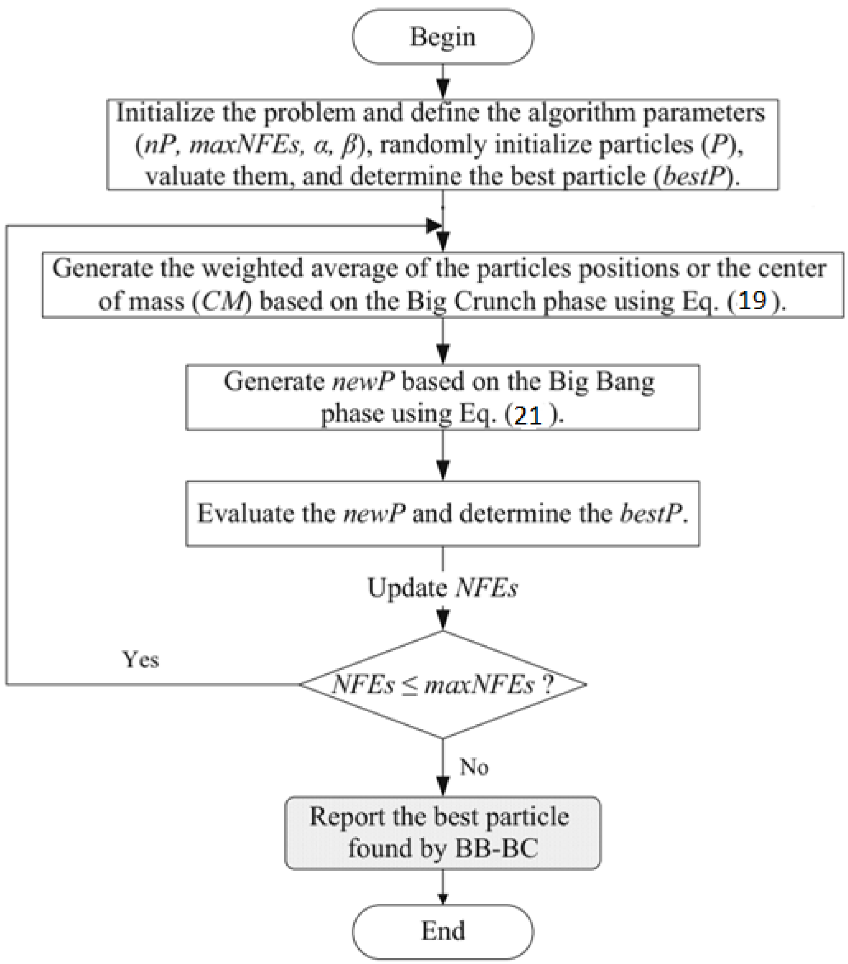

Figure 2:BB-BC Algorithm Flowchart

A. BB-BC applications:

The BB-BC algorithm has shown versatility and effectiveness in a wide range of optimization problems, and these applications highlight its adaptability to various domains. However, research on metaheuristic algorithms is continuously evolving, and there might be more recent studies and applications, Table 1 summarize some applications of BB-BC algorithm.

Table 1: BB-BC Applications

#	Application Name	Author(s)	Summary	Year	References
1.	Structural Optimization	H. Najafi and A. Kaveh	BB-BC applied to optimize the design of steel trusses for minimum weight subject to various constraints.	2007	(Kaveh & Mahdavi, 2016)
2.	Function Optimization	A. Ebrahimzadeh, M. Nezamabadi-pour	BB-BC used to optimize the Rastrigin function, a standard benchmark problem for global optimization.	2011	(Ebrahimzadeh & Nezamabadi-pour, 2011).
3.	Image Segmentation	S. Kamel and A. E. Hassanien	BB-BC applied to the problem of image segmentation to partition an image into distinct regions.	2013	(Kamel & Hassanien, 2013)
4.	Clustering	Abdolreza Hatamlou, Salwani Abdullah, and Masumeh Hatamlou	BB-BC used to cluster data points into groups with similar characteristics, aiding in data analysis.	2011	(Hatamlou et al., 2011)
5.	Portfolio Optimization	V. S. Rathore and N. P. Khandelwal	BB-BC applied to optimize investment portfolios, considering risk and return of different assets.	2020	(Rathore & Khandelwal, 2020)
6.	Robot Path Planning	A. Ghasemi and M. Mirzavand	BB-BC used to plan optimal paths for robots navigating through obstacles and reaching target locations.	2014	(A. Ghasemi & Mirzavand, 2014)
7.	Machine Learning Model Tuning	B. N. Arafa, et al.	BB-BC employed to optimize hyperparameters	2018	(Arafa et al., 2018)

			of machine learning models for improved predictive performance.		
8.	VLSI Circuit Design Optimization	A. Ghasemi and A. R. Alimohammadi	BB-BC applied to optimize Very-Large-Scale Integration (VLSI) circuits for minimized power consumption and area.	2019	(A. , Ghasemi & Alimohammadi, 2019)
9.	Neural Network Training	Sharma Rahul And Singh Amar	BB-BC used for optimizing the weights and biases of neural networks to achieve better training accuracy.	2021	(Sharma & Singh, 2022)
10.	Energy Management in Smart Grids	M. A. Elsoud and M. A. H. Abualrish	BB-BC employed to optimize energy distribution and management in smart grid systems for improved efficiency.	2021	(Elsoud & Abualrish, 2021)

6. Experimental results

The results of utilizing the Big Bang-Big Crunch (BB-BC) method to solve the Welded Beam Design problem (WBD) with various beginning settings for the algorithm's parameters are reported in this section.

Most of the time, BB-BC is determined by a set of constant factors that have a direct influence on how the results converge to the best possible value. Various values for the BB-BC variables are evaluated in this article to illustrate the influence of those factors on BB-BC performance. The study also explores the algorithm's exploration and exploitation capabilities in order to attain the best level of efficacy in resolving the (WBD).

The algorithm executed by considering $P = 6000lb, L = 14\text{ in}, E = 30 \times 106\text{ psi}, G = 12 \times 106\text{ psi}, \tau max = 13{,}600\text{ psi}, \sigma max = 30{,}000\text{ psi}, \delta max = 0.25\text{ in}, 0.1 \le x1 \le 2, 0.1 \le x2 \le 10, 0.1 \le x3 \le 10, 0.1 \le x4 \le 2$.

B. Alpha parameter:

The BB-BC method employs the Alpha parameter to limit the initial search space and provide random first solutions. Table 2 depicts the effect of various Alpha values on the BB-BC algorithm's ability to solve the WBD issue. It displays the convergence history for a single run of the algorithm with the same beginning population and changing Alpha values (0.5, 1, 1.5). In these runs, the value partners fixed to Beta=0.2 and nP=50.

Table 2: BB-BC results with Beta=0.2 nP=50, maxNFEs=20000 and various value of Alpha

Alpha	**X1**	**X2**	**X3**	***X4***	***F(X)***	***time***
0.5	0.205882476184735	7.04876523844401	9.08130386191314	0.205917355010630	***2.2237***	***1.7405***
1	0.203867822684818	7.09055830801460	9.14599275043666	0.205354862778561	***2.2313***	***2.4162***
1.5	0.204955128606107	7.11570416994596	9.07884079625097	0.205563596870117	***2.2261***	***1.6262***

The least fitness of solving WBD with four constraint values (X1, X2, X3, and X4) is shown in Table 2. It demonstrates that when Alpha=0.5, the BB-BC algorithm produces the best results. It shows how differing Alpha values can have a big influence on the results of BB-BC algorithms. It is vital to remember that the size of these affects rises according to both F(X) and the period of time the algorithm has been running.

C. *Beta parameter:*

The beta parameter is used to control the weighted average of particle positions, also known as the center of mass (CM), and the best particle. Table 3 shows how different Beta values affect the BB-BC algorithm's ability to solve the WBD issue. It shows the convergence history for a single run of the algorithm with the same beginning population but different Beta values (0.2, 0.5, and 0.8). The value partners Alpha=0.5 and nP=50 should be specified in these runs.

Table 3: BB-BC results with Alpha=0.5 nP=50, maxNFEs=20000 and various value of Beta

Beta	**X1**	**X2**	**X3**	***X4***	***F(X)***	***time***
0.2	0.205882476184735	7.04876523844401	9.08130386191314	0.205917355010630	2.2237	1.7405
0.5	0.205732934396415	7.09030131981526	9.03885834492323	0.205759121418689	***2.2186***	***1.2659***
0.8	0.331011169651226	4.88479626972069	7.21892100188078	0.332206291109833	***2.7701***	***1.9590***

The least fitness of solving WBD with three constraint values (X1, X2, and X3) is shown in Table 3. It demonstrates that when Beta=0.5, the BB-BC algorithm produces the best results. It shows how differing Beta values can have a big influence on the results of BB-BC algorithms. It is vital to remember that the size of these F(X) affects is best when Beta= 0.5.

D. *BB-BC nP parameter:*

nP denotes the number of Particles involved in the search process. Table 4 shows how different Beta values affect the BB-BC algorithm's ability to solve the WBD issue. It shows the convergence history for a single run of the algorithm with the same beginning population but different nP values (30, 50, and 100). The value of Alpha=0.5 and Beta=0.5 should be specified in these runs.

Table 4: BB-BC results with Alpha=0.5 Beta=0.5, maxNFEs=20000 and various value of nP

nP	X1	X2	X3	X4	*F(X)*	*time*
30	0.205240059100049	7.05601789042014	9.12291346315412	0.205598691776848	*2.2284*	1.4365
50	0.205732934396415	7.09030131981526	9.03885834492323	0.205759121418689	**2.2186**	**1.2659**
100	0.204684299169551	7.10871309598151	9.09290958455787	0.206802452961017	*2.2387*	*1.0309*

The least fitness of solving WBD with three constraint values (X1, X2, and X3) is shown in Table 4. It demonstrates that when nP=100, the BB-BC algorithm produces the best results. It demonstrates how differing nP values may significantly affect the outputs of BB-BC algorithms. It is vital to remember that the size of these F(X) effects is ideal when nP= 50.
Table 5, compares the result of BB-BC algorithm with 20 different metaheuristics algorithms, it shows that the results of BB-BC algorithm comes in 5 position out of 20.

Table 5: comparison of metaheuristics algorithms in solving WBD problem

#	Author	Algorithm	X1	X2	X3	X4	Cost	Ref
1.	Mahdavi et al.	HS	0.2057	3.4705	9.0366	0.2057	1.7248	(Mahdavi et al., 2007)
2.	Fesanghary et al.	HS-SQP	0.2057	3.4706	9.0368	0.2057	1.7248	(Fesanghary et al., 2008)
3.	Xin-She Yang	FA	0.2015	3.562	9.0414	0.2057	1.7312	(Gandomi et al., 2011)
4.	Coello	GA	0.2088	3.4205	8.9975	0.2100	1.7483	(Coello Coello, 2000)
5.	**This study**	**BB-BC**	**0.2057**	**7.0903**	**9.0388**	**0.2057**	***2.2186***	*1.2659*
6.	Almufti	ABC	0.2056	7.0901	9.0419	0.2057	2.2187	(Almufti, 2022a)
7.	Hwang and He	SA-GA	0.2231	1.5815	12.8468	0.2245	2.2500	(Liu, 2005)
8.	Montes and Ocana	BFO	0.2536	7.1410	7.1044	0.2536	2.3398	**(Betania Hernández-Ocaña & Efrén Mezura-Montes, 2009)**
9.	Liu	SA	0.2444	6.2175	8.2915	0.2444	2.3810	**(Hedar & Fukushima, 2006)**
10.	Lee and Geem	HS	0.2442	6.2231	8.2915	0.2443	2.381	**(Lee & Geem, 2005)**
11.	Zhang et al.	DE	0.2444	6.2175	8.2915	0.2444	2.3810	(M. Zhang et al., 2008)
12.	Hedar and Fukushima	SA-DS	0.2444	6.2158	8.2939	0.2444	2.3811	**(Hwang & He, 2006)**
13.	Bernardino et al.	AIS-GA	0.2444	6.2183	8.2912	0.2444	2.3812	(Bernardino et al., 2007)
14.	Lemonge and	GA	0.2443	6.2117	8.3015	0.2443	2.3816	**(Lemonge &**

	Barbosa							**Barbosa, 2004)**
15.	Zhang et al.	EA	0.2443	6.2201	8.2940	0.2444	2.3816	(J. Zhang et al., 2009)
16.	Ray and Liew	SCA	0.2444	6.2380	8.2886	0.2446	2.3854	**(Ray & Liew, 2003)**
17.	Leite and Topping	GA	0.2489	6.1097	8.2484	0.2485	2.4000	**(Leite & Topping, 1998)**
18.	Atiqullah and Rao	SA	0.2471	6.1451	8.2721	0.2495	2.4148	**(ATIQULLAH & RAO, 2000)**
19.	Deb	GA	0.2489	6.1730	8.1789	0.2533	2.4331	(Deb, 1991)
20.	Akhtar et al.	SBM	0.2407	6.4851	8.2399	0.2497	2.4426	(Akhtar et al., 2002)

7. Conclusion

In this article, we have explored the impact of different starting parameters of the Big Bang-Big Crunch Algorithm (BB-BCA) on the resolution of the Welded Beam Design Problem. The BB-BCA, a metaheuristic optimization technique inspired by the concept of the Big Bang and Big Crunch in cosmology, has shown promise in solving complex engineering optimization problems. Our objective was to investigate how the choice of initial parameters affects the algorithm's performance in tackling the Welded Beam Design Problem.

Through a comprehensive series of experiments, we systematically varied the initial parameters, including the Alpha, nP, and Beta with a fixed maximum iteration count. The results obtained from our simulations provide valuable insights into the behavior of the BB-BCA under different configurations.

The analysis revealed that the BB-BCA is sensitive to the initial parameter settings. While certain combinations led to rapid convergence and high-quality solutions, others resulted in slower convergence and suboptimal outcomes. Notably, a careful selection of parameters allowed the algorithm to overcome local optima and explore a more diverse solution space, ultimately leading to improved results.

Based on the results obtained from the experiments conducted on the Big Bang-Big Crunch Algorithm (BB-BCA) with different starting parameters for the resolution of the Welded Beam Design Problem, the following conclusions can be drawn:

1. The Alpha Parameter:

   When maintaining MaxNFEs = 20000, nP = 50, and Beta = 0.2 as constant values, the convergence histories for a single trial run from the same starting population were observed to range between 0 and 1 for different Alpha parameter values (0.5, 1, and 1.5). It was found that the method runs more efficiently when Alpha is set at 0.5. This suggests that lower values of Alpha contribute to faster convergence and better-quality solutions for the given problem.

2. The Beta Parameter:

   With MaxNFEs = 20000, nP = 50, and Alpha = 0.5 held constant, the convergence histories for a single trial run from the same starting population varied between 0.0 and 1.0 for different Beta values (0.2, 0.5, and 0.8). The results indicated that the technique is more efficient when the Beta value is set at 0.5. This implies that moderate values of Beta lead to improved performance and faster convergence.

3. The nP Parameter:

   By setting MaxNFEs = 20000, Alpha = 0.5, and Beta = 0.5 as constant, the convergence histories for a single trial run from the same starting population were observed to vary between 0 and 1 for different nP values (30, 50, and 100). The findings suggest that the approach operates more efficiently when nP is set to 50, indicating that an intermediate number of particles in the population is more effective for the given problem.

By comparing the results of BB-BC algorithm with 20 different metaheuristics algorithm, it can be concluded that BB-BC is well designed for solving optimization problems.

In summary, the investigation into the effect of different starting parameters on the resolution of the Welded Beam Design Problem using the BB-BCA revealed that the algorithm's performance is highly influenced by the values of Alpha, Beta, and nP. Optimal combinations of these parameters lead to more efficient convergence and better-quality solutions.

8. References

Akhtar, S., Tai, K., & Ray, T. (2002). A socio-behavioural simulation model for engineering design optimization. *Engineering Optimization*, *34*(4), 341–354. https://doi.org/10.1080/03052150212723

Almufti, S. M. (2022a). Artificial Bee Colony Algorithm performances in solving Welded Beam Design problem. *Computer Integrated Manufacturing Systems*, *28*(12). https://doi.org/10.24297/j.cims.2022.12.17

Almufti, S. M. (2022b). Vibrating Particles System Algorithm performance in solving Constrained Optimization Problem. *Academic Journal of Nawroz University*, *11*(3), 231–242. https://doi.org/10.25007/ajnu.v11n3a1499

Almufti, S. M. (2022c). Vibrating Particles System Algorithm performance in solving Constrained Optimization Problem. *Academic Journal of Nawroz University*, *11*(3), 231–242. https://doi.org/10.25007/ajnu.v11n3a1499

Almufti, S. M., Asaad, R. R., & Salim, B. W. (2018). Review on Elephant Herding Optimization Algorithm Performance in Solving Optimization Problems. *Article in International Journal of Engineering and Technology*, *7*(4), 6109–6114. https://doi.org/10.14419/ijet.v7i4.23127

Arafa, B. N., El-Henawy, I., Houssein, E. H. , & Rizk-Allah, R. M. (2018). Machine learning model tuning using Big Bang–Big Crunch algorithm. *Computational Intelligence and Informatics* , 31–41.

Asaad, R., & Abdulnabi, N. (2018). Using Local Searches Algorithms with Ant Colony Optimization for the Solution of TSP Problems. *Academic Journal of Nawroz University*, *7*(3), 1–6. https://doi.org/10.25007/ajnu.v7n3a193

ATIQULLAH, M. M., & RAO, S. S. (2000). SIMULATED ANNEALING AND PARALLEL PROCESSING: AN IMPLEMENTATION FOR CONSTRAINED GLOBAL DESIGN OPTIMIZATION. *Engineering Optimization*, *32*(5), 659–685. https://doi.org/10.1080/03052150008941317

Bernardino, H. S., Barbosa, H. J. C., & Lemonge, A. C. C. (2007). A hybrid genetic algorithm for constrained optimization problems in mechanical engineering. *2007 IEEE Congress on Evolutionary Computation*, 646–653. https://doi.org/10.1109/CEC.2007.4424532

Betania Hernández-Ocaña, & Efrén Mezura-Montes. (2009). Modified Bacterial Foraging Optimization for Engineering Design. In *Intelligent Engineering Systems through Artificial Neural Networks* (pp. 357–364). ASME Press. https://doi.org/10.1115/1.802953.paper45

Camp, C. V. (2007). Design of Space Trusses Using Big Bang–Big Crunch Optimization. *Journal of Structural Engineering*, *133*(7), 999–1008. https://doi.org/10.1061/(ASCE)0733-9445(2007)133:7(999)

Christu Nesam David. D, & S. Elizabeth Amudhini Stephen. (2018). COST MINIMIZATION OF WELDED BEAM DESIGN PROBLEM USING NONTRADITIONAL OPTIMIZATION THROUGH MATLAB AND VALIDATION THROUGH ANSYS SIMULATION. *INTERNATIONAL JOURNAL OF MECHANICAL ENGINEERING AND TECHNOLOGY (IJMET)*, *9*(8), 180–192.

Coello Coello, C. A. (2000). Use of a self-adaptive penalty approach for engineering optimization problems. *Computers in Industry*, *41*(2), 113–127. https://doi.org/10.1016/S0166-3615(99)00046-9

Dalirinia, E., Jalali, M., Yaghoobi, M., & Tabatabaee, H. (2023). Lotus effect optimization algorithm (LEA): a lotus nature-inspired algorithm for engineering design optimization. *The Journal of Supercomputing*. https://doi.org/10.1007/s11227-023-05513-8

Deb, K. (1991). Optimal design of a welded beam via genetic algorithms. *AIAA Journal*, *29*(11), 2013–2015. https://doi.org/10.2514/3.10834

Ebrahimzadeh, A., & Nezamabadi-pour, M. (2011). A new global optimization algorithm: Big Bang–Big Crunch. *Advances in Engineering Software*, *42*(10), 967-973.

Eesa, A. S., Hassan, M. M., & Arabo, W. K. (2023). Letter: Application of optimization algorithms to engineering design problems and discrepancies in mathematical formulas. *Applied Soft Computing*, *140*, 110252. https://doi.org/10.1016/j.asoc.2023.110252

Elsoud, M. A., & Abualrish, M. A. H. (2021). Multi-objective big bang-big crunch algorithm for energy management in smart grid. *Electrical Engineering*.

Erol, O. K., & Eksin, I. (2006). A new optimization method: Big Bang–Big Crunch. *Advances in Engineering Software*, *37*(2), 106–111. https://doi.org/10.1016/j.advengsoft.2005.04.005

Fesanghary, M., Mahdavi, M., Minary-Jolandan, M., & Alizadeh, Y. (2008). Hybridizing harmony search algorithm with sequential quadratic programming for engineering optimization problems. *Computer Methods in Applied Mechanics and Engineering*, *197*(33–40), 3080–3091. https://doi.org/10.1016/j.cma.2008.02.006

Gandomi, A. H., Yang, X.-S., & Alavi, A. H. (2011). Mixed variable structural optimization using Firefly Algorithm. *Computers & Structures*, *89*(23–24), 2325–2336. https://doi.org/10.1016/j.compstruc.2011.08.002

Ghasemi, A. , & Alimohammadi, A. R. (2019). High-level synthesis and optimization of VLSI circuits using Big Bang–Big Crunch algorithm. *Integration, the VLSI Journal, , 65*, 234–244.

Ghasemi, A., & Mirzavand, M. (2014). Robot path planning using Big Bang–Big Crunch algorithm. *Robotics and Autonomous Systems*, *62*(3), 390–399.

Goel, L., Kanhar, J., Patel, V. S., & Vardhan, A. (2023). Hybrid Elephant Herding Optimization–Big Bang Big Crunch for pattern recognition from natural images. *Soft Computing*. https://doi.org/10.1007/s00500-023-08667-y

Hatamlou, A., Abdullah, S., & Hatamlou, M. (2011). Data Clustering Using Big Bang–Big Crunch Algorithm. In *Springer-Verlag Berlin Heidelberg* (pp. 383–388). https://doi.org/10.1007/978-3-642-27337-7_36

Hedar, A.-R., & Fukushima, M. (2006). Derivative-Free Filter Simulated Annealing Method for Constrained Continuous Global Optimization. *Journal of Global Optimization*, *35*(4), 521–549. https://doi.org/10.1007/s10898-005-3693-z

Hwang, S.-F., & He, R.-S. (2006). A hybrid real-parameter genetic algorithm for function optimization. *Advanced Engineering Informatics*, *20*(1), 7–21. https://doi.org/10.1016/j.aei.2005.09.001

Ihsan, R. R., Almufti, S. M., Ormani, B. M. S., Asaad, R. R., & Marqas, R. B. (2021). A Survey on Cat Swarm Optimization Algorithm. *Asian Journal of Research in Computer Science*, 22–32. https://doi.org/10.9734/ajrcos/2021/v10i230237

Jahwar, A. F., Mohsin Abdulazeez, A., Zeebaree, D. Q., Asaad Zebari, D., & Ahmed, F. Y. H. (2021). An Integrated Gapso Approach for Solving Problem of an Examination Timetabiking System. *2021 IEEE Symposium on Industrial Electronics & Applications (ISIEA)*, 1–6. https://doi.org/10.1109/ISIEA51897.2021.9509984

Kamel, S., & Hassanien, A. E. (2013). Image segmentation using Big Bang–Big Crunch optimization algorithm. *Applied Soft Computing*, *13*(1), 91–100.

Kamil, A. T., Saleh, H. M., & Abd-Alla, I. H. (2021). A Multi-Swarm Structure for Particle Swarm Optimization: Solving the Welded Beam Design Problem. *Journal of Physics: Conference Series*, *1804*(1), 012012. https://doi.org/10.1088/1742-6596/1804/1/012012

Kaveh, A., & Bakhshpoori, T. (2019). *Metaheuristics: Outlines, MATLAB Codes and Examples*. Springer International Publishing. https://doi.org/10.1007/978-3-030-04067-3

Kaveh, A., & Mahdavi, V. R. (2016). Optimal design of truss structures using a new optimization algorithm based on global sensitivity analysis. *Structural Engineering and Mechanics*, *60*(6), 1093–1117. https://doi.org/10.12989/sem.2016.60.6.1093

Lee, K. S., & Geem, Z. W. (2005). A new meta-heuristic algorithm for continuous engineering optimization: harmony search theory and practice. *Computer Methods in Applied Mechanics and Engineering*, *194*(36–38), 3902–3933. https://doi.org/10.1016/j.cma.2004.09.007

Leite, J. P. B., & Topping, B. H. V. (1998). Improved genetic operators for structural engineering optimization. *Advances in Engineering Software*, *29*(7–9), 529–562. https://doi.org/10.1016/S0965-9978(98)00021-0

Lemonge, A. C. C., & Barbosa, H. J. C. (2004). An adaptive penalty scheme for genetic algorithms in structural optimization. *International Journal for Numerical Methods in Engineering*, *59*(5), 703–736. https://doi.org/10.1002/nme.899

Liu, J. (2005). Novel orthogonal simulated annealing with fractional factorial analysis to solve global optimization problems. *Engineering Optimization*, *37*(5), 499–519. https://doi.org/10.1080/03052150500066646

M. Almufti, S. (2019). Historical survey on metaheuristics algorithms. *International Journal of Scientific World*, *7*(1), 1. https://doi.org/10.14419/ijsw.v7i1.29497

M. Almufti, S., Ahmad Shaban, A., Ismael Ali, R., & A. Dela Fuente, J. (2023). Overview of Metaheuristic Algorithms. *Polaris Global Journal of Scholarly Research and Trends*, *2*(2), 10–32. https://doi.org/10.58429/pgjsrt.v2n2a144

M. Almufti, S., Yahya Zebari, A., & Khalid Omer, H. (2019). A comparative study of particle swarm optimization and genetic algorithm. *Journal of Advanced Computer Science & Technology*, *8*(2), 40. https://doi.org/10.14419/jacst.v8i2.29401

Mahdavi, M., Fesanghary, M., & Damangir, E. (2007). An improved harmony search algorithm for solving optimization problems. *Applied Mathematics and Computation*, *188*(2), 1567–1579. https://doi.org/10.1016/j.amc.2006.11.033

Mbuli, N., & Ngaha, W. S. (2022). A survey of big bang big crunch optimisation in power systems. *Renewable and Sustainable Energy Reviews*, *155*, 111848. https://doi.org/10.1016/j.rser.2021.111848

O. K. Erol, O. Hasançebi, & S. Kazemzadeh Azad. (2011). Evaluating efficiency of big-bang big-crunch algorithm in benchmark engineering optimization problems. *INTERNATIONAL JOURNAL OF OPTIMIZATION IN CIVIL ENGINEERING*, *3*, 495–505.

Öztürk, H. T., & Kahraman, H. T. (2023). Meta-heuristic search algorithms in truss optimization: Research on stability and complexity analyses. *Applied Soft Computing*, *145*, 110573. https://doi.org/10.1016/j.asoc.2023.110573

Parsopoulos, K. E., & Vrahatis, M. N. (2005). *Unified Particle Swarm Optimization for Solving Constrained Engineering Optimization Problems* (pp. 582–591). https://doi.org/10.1007/11539902_71

Prayogo, D., Cheng, M.-Y., Wu, Y.-W., Herdany, A. A., & Prayogo, H. (2018). Differential Big Bang - Big Crunch algorithm for construction-engineering design optimization. *Automation in Construction*, *85*, 290–304. https://doi.org/10.1016/j.autcon.2017.10.019

Rathore, V. S., & Khandelwal, N. P. (2020). Portfolio optimization using Big Bang-Big Crunch algorithm. *2020 International Conference on Emerging Trends in Information Technology and Engineering (Ic-ETITE)*.

Ray, T., & Liew, K. M. (2003). Society and civilization: an optimization algorithm based on the simulation of social behavior. *IEEE Transactions on Evolutionary Computation*, *7*(4), 386–396. https://doi.org/10.1109/TEVC.2003.814902

Sadeeq, H. T., Abdulazeez, A. M., Kako, N. A., Zebari, D. A., & Zeebaree, D. Q. (2021a). A New Hybrid Method for Global Optimization Based on the Bird Mating Optimizer and the Differential Evolution. *2021 7th International Engineering Conference "Research & Innovation amid Global Pandemic" (IEC)*, 54–60. https://doi.org/10.1109/IEC52205.2021.9476147

Sadeeq, H. T., Abdulazeez, A. M., Kako, N. A., Zebari, D. A., & Zeebaree, D. Q. (2021b). A New Hybrid Method for Global Optimization Based on the Bird Mating Optimizer and the Differential Evolution. *2021 7th International Engineering Conference "Research & Innovation amid Global Pandemic" (IEC)*, 54–60. https://doi.org/10.1109/IEC52205.2021.9476147

Sharma, R., & Singh, A. (2022). Big bang–big crunch-CNN: an optimized approach towards rice crop protection and disease detection. *Archives of Phytopathology and Plant Protection*, *55*(2), 143–161. https://doi.org/10.1080/03235408.2021.2003054

Tang, H., Zhou, J., Xue, S., & Xie, L. (2010). Big Bang-Big Crunch optimization for parameter estimation in structural systems. *Mechanical Systems and Signal Processing*, *24*(8), 2888–2897. https://doi.org/10.1016/j.ymssp.2010.03.012

Yokota, T., Taguchi, T., & Gen, M. (1999). A solution method for optimal cost problem of welded beam by using genetic algorithms. *Computers & Industrial Engineering*, *37*(1–2), 379–382. https://doi.org/10.1016/S0360-8352(99)00098-4

Zhang, J., Liang, C., Huang, Y., Wu, J., & Yang, S. (2009). An effective multiagent evolutionary algorithm integrating a novel roulette inversion operator for engineering optimization. *Applied Mathematics and Computation*, *211*(2), 392–416. https://doi.org/10.1016/j.amc.2009.01.048

Zhang, M., Luo, W., & Wang, X. (2008). Differential evolution with dynamic stochastic selection for constrained optimization. *Information Sciences*, *178*(15), 3043–3074. https://doi.org/10.1016/j.ins.2008.02.014