Wolf-Bird Optimizer (WBO): A Novel Metaheuristic Algorithm For Building Information Modeling-Based Resource Tradeoff

Journal of Engineering Research xxx (xxxx) xxx

Contents lists available at ScienceDirect

Journal of Engineering Research

journal homepage: www.journals.elsevier.com/journal-of-engineering-research

Wolf-Bird Optimizer (WBO): A novel metaheuristic algorithm for Building Information Modeling-based resource tradeoff

Mahdi Azizi [a,1], Milad Baghalzadeh Shishehgarkhaneh [b,1], Mahla Basiri [a], Robert C. Moehler [c,*], Yihai Fang [b], Melissa Chan [d,e]

[a] Department of Civil Engineering, University of Tabriz, Iran
[b] Department of Civil Engineering, Monash University, Clayton, VIC 3800, Australia
[c] Department of Infrastructure Engineering, The University of Melbourne, Melbourne, Australia
[d] Institute for Sustainable Industries & Liveable Cities (ISILC), Victoria University, Melbourne 3011, Australia
[e] College of Sport, Health and Engineering, Victoria University, Melbourne 3011, Australia

ARTICLE INFO

Keywords:
Wolf-bird optimizer
Bioinspiration
Metaheuristic algorithm
Evolutionary computation, Building information modeling
Resource management

ABSTRACT

In the animal kingdom, a mutually-beneficial ecosystemic coexistence and partnership in predation between wolves and ravens, known as the wolf-bird relationship, is observed in various cultures. The Wolf-Bird Optimizer (WBO), a novel metaheuristic algorithm inspired by this natural zoological relationship, is proposed. This method is developed based on the foraging behaviors of ravens and wolves, wherein the intelligence of ravens in finding prey and sending signals to wolves for assistance in hunting is considered. Furthermore, a framework for resource tradeoffs in project scheduling using metaheuristic algorithms and the Building Information Modeling (BIM) approach is established in this research. For statistical analysis, the algorithms are independently run 30 times with a preset stopping condition, enabling the calculation of descriptive statistical metrics such as mean, standard deviation (SD), and the required number of objective function evaluations. To ensure the statistical significance of the results, several inferential statistical methods, including the Kolmogorov-Smirnov, Wilcoxon, Mann-Whitney, and Kruskal-Wallis tests, are employed. Additionally, the capability of the proposed algorithm in solving resource tradeoff problems in four construction projects is assessed. The performance of the WBO algorithm is also evaluated in two benchmark construction projects, with the results indicating the algorithm's ability to produce competitive and exceptional outcomes regarding tradeoffs.

1. Introduction

Optimization is selecting the optimal solution from a collection of feasible alternatives under specified criteria. It is used to make decisions in various fields, including engineering, management, economics, and finance. Generally, real-world problems are classified as discrete, continuous, constrained, or unconstrained problems. However, it is often challenging to solve specific problems using traditional mathematical programming techniques such as the fast steepest descent, sequential quadratic programming, conjugate gradient, and quasi-Newton methods. In other words, although classical methods guarantee the optimal solution, they remain efficient only for small-scale problems, demanding significant computing work for larger-scale ones. Hence, in recent years, metaheuristic optimization algorithms have gained much popularity in different fields of science and engineering because they can produce optimal solutions to complex real-world problems more efficiently [1–3].

1.1. Category of metaheuristic algorithm

Glover suggested the term metaheuristic for the first time in 1986, which consists of a core word, i.e., heuristic, and a prefix, i.e., meta, both of which have Greek roots. The term 'heuristic' is derived from the Greek word "heuriskein," which means " to find, " and 'meta' implies "beyond the usual or natural limitations of anything." However, the advantages of metaheuristic algorithms are four folds: (i) they can avoid local optimums; (ii) they are generally based on relatively simple concepts and have straightforward implementations; (iii) they do not

* Corresponding author.
E-mail address: robert.moehler@unimelb.edu.au (R.C. Moehler).
[1] These authors contributed equally

https://doi.org/10.1016/j.jer.2023.11.024
Received 26 September 2023; Received in revised form 12 November 2023; Accepted 23 November 2023
Available online 27 November 2023

require gradient information; and (iv) they may be used to solve a wide range of problems across many fields. In addition, because metaheuristics are computer-based, progressive increases in the processing capacity of modern computers have hastened their development [1–4].

Generally speaking, metaheuristics can be categorized into two primary groups: (i) single-solution-based methods (e.g., the Tabu Search [5] and micro-canonical annealing [6]), in which the search process begins with a single solution candidate, which is then improved throughout iterations; and (ii) population-based methods, which carry out the optimization procedure utilizing a set of possible solutions (population), in which the search process begins with a random initial population (many solutions), thereby ameliorating the population over iterations. In other words, single-solution-based algorithms are considered useful in local search, while population-based algorithms are more exploration-oriented [7–9]. The latter category could be divided into four main classes:

- **Evolutionary Algorithms (EAs):** They simultaneously conduct the search process utilizing numerous initial points. They are well-suited for problems with stochastic features, uncertainty, or fitness under noise conditions [10]. The genetic algorithm (GA) based on the Darwinian theory of evolution is one of the renowned population-based metaheuristic algorithms introduced by Holland [11], in which the critical operators of GA are selection, crossover, and mutation [12]. The Differential Evolution (DE) algorithm is another prominent stochastic population-based algorithm for global optimization [13,14]. Furthermore, as a novel sub-area of EAs, genetic programming has gained substantial popularity over the past few decades, proposed for automatic programming and machine learning [15].
- **Swarm Intelligence (SI):** Swarm-based algorithms draw inspiration from the communal behaviors observed in various species, such as ants, bees, birds, fish, and termites. Two notable algorithms in this category are Particle Swarm Optimization (PSO) [16] and Ant Colony Optimization (ACO) [17]. PSO mirrors the flocking behavior of birds, while ACO simulates the foraging patterns of ant colonies [18]. The term "swarm intelligence" was first coined by Beni and Wang [19] in 1993, describing the emergent collective wisdom of simple agent groupings [20]. Other algorithms in this realm include the Border Collie Optimization (BCO) [21], Artificial Bee Colony (ABC) [22,23], Monarch Butterfly Optimization (MBO) [24], Fire Hawk Optimizer (FHO) [25,26], Stochastic Diffusion Search (SDS) [27], Black Widow Optimization Algorithm (BWO) [28], Flower Pollination Algorithm (FPA) [29], Glowworm Swarm Optimization (GSO) [30], and Moth Search Algorithm (MSA) [31].
- **Physics-based algorithms:** Drawing inspiration from the principles of physics, including heat transfer, gravitational force, particle dynamics, and wave propagation, various algorithms have been developed. Among these, the Simulated Annealing (SA) algorithm stands out; it's modeled after the statistical mechanics of annealing in solids and has garnered significant attention in the field [32]. Another prominent physics-based algorithm is the Big-Bang Big--Crunch (BBBC), which is grounded in theories concerning the universe's inception and progression [33]. Furthermore, some of the recently proposed algorithms in this category are the Weighted Vertices Optimizer (WVO) [34], Chaotic Stochastic Paint Optimizer (CSPO) [35], Atomic Orbital Search (AOS) [36,37], Energy valley optimizer [38], Material Generation Algorithm (MGA) [39–41], Cyber-physical Systems (CPS) [42], Archimedes Optimization Algorithm (AOA) [43], Charged System Search (CSS) [44–46], Equilibrium Optimizer (EO) [47], Lichtenberg Algorithm (LA) [48], Thermal Exchange Optimization algorithm (TEOA) [49], Lévy Flight Distribution (LFD) [50], Vibrating Particles System (VPS) [51,52], and some other algorithms [53–57].
- **Human and animal behavior-based algorithms:** They are inspired by some specific behaviors of individuals in society; examples include Human Behavior-Based Optimization (HBBO) [58], Squid Game Optimizer (SGO) [59], Soccer League Competition (SLC) [60], Waterwheel Plant Technique (WWPA) [61], Teaching–Learning-Based Optimization (TLBO) [62], Spider Wasp Optimization (SWO) [63], Mountain Gazelle Optimizer (MGO) [64], Harmony Search (HS) [65], Imperialist Competitive Algorithm (ICA) [66], Colliding Bodies Optimization (CBO) [67], and Interior Search Algorithm (ISA) [68].

However, multi-objective metaheuristics are sophisticated algorithms designed to tackle optimization problems involving multiple, often conflicting, objectives [69,70]. Unlike single-objective optimization, which seeks to find the best solution according to a singular criterion, multi-objective optimization acknowledges the complexity of real-world scenarios where trade-offs must be made between different goals. These algorithms aim to find a set of Pareto optimal solutions, each of which represents a different trade-off among the objectives. In these solutions, no objective can be improved without worsening at least one other, embodying the principle of non-dominance. This approach is crucial in fields like engineering, economics, and environmental planning, where decisions often involve balancing competing interests [71–73].

The development of multi-objective metaheuristics involves integrating mechanisms to simultaneously handle multiple objectives and maintain a diverse set of solutions. These algorithms typically include specialized operators to ensure diversity and convergence towards the Pareto front. Key challenges in multi-objective metaheuristics include maintaining a balance between exploration (searching new areas of the solution space) and exploitation (refining existing solutions), as well as effectively managing the increased computational complexity. Popular examples include algorithms like NSGA-II, MOMGA [74], MOAVOA [75], MOAOS [76], MOTEO [77], MOSPO [78], MOCryStAl [53], and MOEA/D, each employing unique strategies to address the intricacies of multi-objective optimization. The effectiveness of these algorithms is assessed based on their ability to approximate the true Pareto front and maintain diversity among the solutions.

Given an initial random population, metaheuristics can effectively and efficiently explore the optimization search space by avoiding locally optimal solutions while converging to a better solution [79,80]. The search process should be sufficiently intelligent to properly assess different regions of the search space and expand into previously unexplored regions with high-quality solutions. Exploration and exploitation are terms used to describe these processes. Exploration refers to finding new regions in the search space to find new solutions. On the other hand, exploitation concerns the enhancement process of existing solutions to achieve better candidates around the existing ones [81–84]. Population-based metaheuristic algorithms can balance the exploitation and exploration phases of the search space to achieve high efficiency. The exploration phase allows the algorithm to check several possible regions of the search space and generate new solutions to avoid the local optima dilemma [85–87]. Exploration refers to introducing a series of random adjustments through multi-armed bandit schemes before returning to the original recipe, while exploitation refers to following a recipe until it ceases to be successful [88–91]. A proper balance of these two phases may guarantee that the global optimum is achieved.

Even though there are several metaheuristic algorithms, new ones are continually required. According to the No Free Lunch (NFL) hypothesis, no one approach exists for finding an optimal solution to all optimization problems. As a result, developing new metaheuristic optimization algorithms is still a work in progress [92–95]. The scientific community is benefiting from developing new metaheuristics, which may enhance the accuracy or efficiency of the optimization process for a range of problems. Consequently, this statement motivates our efforts to propose a novel metaheuristic algorithm inspired by coexistence and partnership in the predation of wolves and ravens in nature.

Motivated by this practical need, we propose a novel population-based metaheuristic optimization algorithm named the Wolf-bird Optimizer (WBO), inspired by the mutually-beneficial ecosystemic coexistence and partnership in the predation of wolves and ravens in nature. The proposed algorithm finds the optimal solution from a population of candidate solutions modeled in three steps as follows: (1) a group of ravens in nature is attempting to find some prey in their neighborhood; (2) ravens send intelligent signals to wolves to inform them of the location of the prey; and (3) the group of wolves conforms the signals to reach the prey. This inspirational concept is used for the first time in this research to develop a new metaheuristic algorithm. Furthermore, the complexity levels of the test functions utilized in this study are also evaluated for the first time.

1.2. Resource trade-off problems

Highways, housing developments, high-rise structures, tunnel networks, and housing projects are all popular places to see repetitive construction projects. They are defined by repeated activities made by each unit [96]. For the majority of construction operations, such as equipment sizes, construction techniques, the number of employees, overtime, and the kind of materials, there are various construction choices accessible during the planning phase of a project. Consequently, numerous resources may be employed to perform project activities in difficulty with project planning. We are faced a discrete decision-making issue in which the resource allocations determine the choice of projects. The number of resources devoted to the activity will affect the project's total cost and execution time [97,98]. Hence, Time-Cost Trade-off (TCT) problems are one of the prominent and challenging problems among project managers. In other words, construction projects' time and profitability are critical criteria that often determine project success or failure. Project managers and planners should examine and optimize resource usage choices and options to achieve these two crucial goals [99].

Construction managers cannot do an automated analysis of the alternatives using popular project planning tools like Primavera P6 or Microsoft Project. As a result, while planning a schedule, construction managers have just one option for each activity [100]. Eshtehardian, Afshar [101] proposed an approach for TCT problems using GA and fuzzy logic theory; the authors concluded that the mentioned method could accelerate the decision-making process in construction projects. Eshtehardian, Afshar [102] presented a novel approach to TCT problems in an uncertain environment considering fuzzy logic theory. Kalhor, Khanzadi [103] used a non-dominated sorting version of ACO (NAACO) in solving the TCT problems.

However, in recent contracts, performance quality has been considered alongside time and cost. Time cost quality tradeoff (TCQT) problems arise in routine building projects, and they aim to find a schedule that strikes a good balance between the project's competing priorities of effectiveness, cost, and quality [104]. For this purpose, researchers have developed and proposed a plethora of methods to solve the TCT problems, such as neural networks [105], dynamic programming [106,107], linear programming [108], and fuzzy logic [109]. However, in recent years, metaheuristic algorithms have gained much popularity among academics for dealing with tradeoff problems in the construction industry. Since they have fewer control parameters, only two common control parameters (population size and the maximum number of function evaluations) must be modified for metaheuristic algorithms. Wood [110] solved a TCQT problem in a gas and oil project employing a fuzzy memetic optimization algorithm. Kosztyán and Szalkai [111] proposed a matrix-based TCQT model supporting the hybrid project management (HPM) technique. Wang, Abdallah [100] analyzed the TCQT problem in projects considering 20 activities and using Non-Dominated Sorting Genetic Algorithms. The authors claimed that the proposed model could find the shortest execution time, cost, and maximum quality. El-Rayes and Kandil [112] proposed a multi-objective optimization algorithm to solve the TCQT problems in highway construction projects. Adebayo [113] employs the Wavelet Local Multiple Correlation (WLMC) to analyze the interplay between China's economic growth, coal usage, natural resource consumption, and CO_2 emissions from 1970 to 2020, revealing that each of these elements consistently boosts CO_2 emissions, exacerbating environmental harm. Ghasemi, Mousavi [114] presents a new mathematical model for production scheduling in uncertain environments, incorporating decision-making methods, multiple execution modes, and trade-offs between cost, time, and quality, with enhanced activity quality achievable through reworking. Son and Khoi [115] introduced the adaptive selection slime mold algorithm (ASSMA) for managing repetitive projects, merging tournament selection and the slime mold algorithm, and demonstrates its efficacy in a rural water pipeline project, outperforming the previous data envelopment analysis (DEA) method, providing project managers with a superior optimization tool. Furthermore, Yılmaz and Dede [116] integrated the non-dominant sorting method into Rao-1 and Rao-2 algorithms for addressing the time-cost trade-off problem in construction planning, with findings showing the NDS-Rao-2 algorithm outperforms previous models and offers multiple optimal solutions, making it a promising option for such combinatorial problems.

Furthermore, some researchers have considered other factors in TCT problems, such as risk, carbon dioxide emission, energy, etc. Tran and Long [117] proposed adaptive multiple objective DE algorithms to solve time, cost, and risk tradeoff (TCRT) problems. The authors claimed that the proposed algorithm could reduce the risks in projects. Liu, Tao [118] proposed a particle swarm optimization model to help project managers solve the cost and CO_2 emission tradeoff problems in projects. To solve the time–cost–environmental impact tradeoff (TCET) problems, Cheng and Tran [119] suggested the opposition-based multiple-objective differential evolution (OMODE).

1.3. Building Information Modeling (BIM)

How buildings are conceptualized, built, constructed, and operated, has been drastically altered as a result of Building Information Modeling (BIM), a cutting-edge and highly successful technology and technique [120]. BIM's recent growth has opened up new possibilities for enhancing the construction process and using emerging technology across the board, from planning to design to construction to maintenance, in both buildings and infrastructure [121,122]. BIM is described as a systematic approach to managing and distributing information generated throughout a project's design and operational phases [123]. At its core, BIM facilitates the exchange, understanding, and utilization of metadata associated with computer-aided design (CAD) models. This aids various stakeholders involved in both the construction and operational processes [124]. Incorporating BIM during the initial design stages of a project offers a unique advantage in project management [125]. Unlike traditional CAD drawings, BIM serves as a comprehensive digital repository, capturing multidisciplinary construction details and graphical attributes of building models. BIM streamlines the data-sharing process for project teams, minimizing the need to rebuild models and expediting the design process with added iterations [126]. Essentially, BIM elevates design and construction quality, reduces project time and costs, and presents a more efficient and profitable approach to construction management [127].

When formulating a novel metaheuristic algorithm, it's imperative to rigorously evaluate its efficacy by juxtaposing it against a gamut of well-established algorithms under consistent experimental conditions, particularly across an array of homogenous problems. From a statistical vantage point, a sample size of 30 independent optimization runs is deemed appropriate to derive crucial statistical metrics. These metrics encompass the mean, standard deviation, and the indispensable count of objective function evaluations. An established stopping criterion, grounded on 150,000 objective function evaluations coupled with a stringent tolerance threshold of 1×10^{-12} for the globally optimal values of the assessed problems, has been integrated into the evaluation framework. For a more nuanced comparative analysis, various recognized statistical methods, such as the Wilcoxon, Kolmogorov-Smirnov, Kruskal-Wallis, and Mann-Whitney tests, are employed. Notably, the WBO algorithm is distinguished by its rapid convergence dynamics, minimized objective function evaluations, and robust performance across a spectrum of problems. Nonetheless, it is essential to highlight the intrinsic limitations of the WBO. Unlike deterministic algorithms, the WBO, in alignment with its metaheuristic counterparts, is fundamentally an approximation-based method, rendering it incapable of producing exact solutions.

A new, nature-inspired algorithm has been introduced in this research. Three different construction projects, including homes and infrastructure, have been evaluated using it. It has been demonstrated that various project challenges can be effectively addressed by the WBO algorithm, representing another significant contribution of this study. The fundamental contributions of this research work are as follows:

- The coexistence behavior and partnership of wolves and ravens in nature are examined and analyzed to develop an appropriate mathematical model for a novel optimization algorithm.
- Based on this model, a unique nature-inspired algorithm, namely the Wolf-Bird Optimizer (WBO), is developed, in which the process of solution updating depends on the intelligent acts of ravens in nature.
- The performance of WBO is extensively evaluated against resource tradeoff problems in some real construction projects.
- The findings are compared to several established metaheuristic algorithms.

2. Wolf-Bird Optimizer (WBO)

2.1. Inspiration

Regarding the elementary ecology texts, organisms interact in three primary ways: competition, predation, and mutualism [128]. Mutualisms are becoming more widely acknowledged as essential to ecological system patterns and processes. Mutualisms occur in various ecosystems across the globe, and ecologists today recognize that practically every species on the planet is engaged in one or more mutualisms, either directly or indirectly. For example, animal-mediated pollination and seed dispersion are common in tropical forests; the plants profit from the pollen and seeds carried by animals, while the animals are typically drawn to and rewarded by food [129].

Ravens, Corvus corax, belong to the corvid family and are known for their long-term monogamy, with partners staying together all year, which may be found across the Northern Hemisphere. Despite being categorized as a territorial species, mating couples are not socially separated and may be pretty social before becoming territorial [130]. When ravens are hungry or find food they cannot reach, they yell loudly. When their parents return after foraging, juvenile ravens yell; subadult and adult nonbreeders yell while faced with food protected by dominant species, and adult pair-bonded females yell when begging for food from their partners [131]. On the other hand, the raven has adapted to a specific eating niche. Its social connections as a carcass expert often involve predators. In northern climes, where a carcass may linger months rather than days, as in southern climates, the raven may confront predators and other ravens frequently and for extended periods. It is reasonable to assume that to survive with others and get food from them, and one has developed an inherent sense of caution and the ability to predict their responses. Despite the raven's reputation for being clever and intellectual, no studies of its psychological characteristics existed until 1943, when Koehler released a paper claiming that his 10-year-old pet raven Jakob could count to seven [132]. "You know, raven does not seek its food," a Koyukon elder explains. He sluggishly obtains his food just by looking for anything already dead [133]. Food-associated sounds have been observed chiefly in primates and birds and may be used to attract competition, attract possible mates, or indicate status. On the other hand, signals may contain information about the sender's motivational state, behavior, or identity and information about stimuli or events in the environment, such as the location, quality, or amount of food [131].

However, among different species worldwide, ravens and wolves have a remarkable mutualism called "wolf-birds." The most evident beneficiaries of the wolf-raven partnership are ravens. According to studies, ravens are observed at 100% of wolf carcasses, taking approximately two-thirds of the carcass. Predator-scavenger interspecific kleptoparasitism, or a relationship between two distinct species in which the scavenger benefits from the predator by taking a part of their food, is the official title for their interaction. The ravens stay close together, detect a wolf's hunting call, and fly above the hunt. Ravens cannot open a carcass on their own; thus, the eyeballs are the only component of the carcass that a raven can eat without the aid of the wolves. In this case, the raven is the predator the wolf uses [134]. According to anecdotal evidence, ravens track wolf packs by following them directly, tracking their footprints in the snow, or reacting to vocalizations to identify their location [135]. In other words, Ravens are afraid of the carcasses of animals they want to eat when they are alone. Is it a genuine fear, or is the notion that the raven is useless? Ravens also have difficulty getting their hands on meat that has not been prepared. They can only forage on the eyeballs or possibly an exposed tongue in an open mouth as far as they can go. They will shout "glug-glug-glug" in the vicinity of an unopened carcass, naturally attracting wolves, who will investigate and do whatever the raven wants to get into it. It is advantageous to both of them [136]. Raven vocalizations also notify wolves of nearby prey, and wolves respond to raven vocalizations. Wolves have also been observed following ravens as they fly. If a carcass is too difficult for ravens to peck their way through on their own, they will more or less gift it to wolves by discovering and bringing them to it is a fantastic example of give-and-take.

2.2. Mathematical model

This sub-section delves into a comprehensive mathematical depiction of the WBO as an optimization algorithm, building upon the strategies elaborated in earlier sub-sections. Initially, the process begins with the initialization, wherein the search space is viewed as a specific region of the earth. Here, the solution candidates (Xi) are conceptualized as ravens and wolves situated within that earthly region.

Fig. 1. The schematic presentation of nearby random walk by the best raven to find prey.

$$X = \begin{bmatrix} X_1 \\ X_2 \\ \vdots \\ X_i \\ \vdots \\ X_n \end{bmatrix} = \begin{bmatrix} x_1^1 & x_1^2 \cdots x_1^j \cdots x_1^d \\ x_2^1 & x_2^2 \cdots x_2^j \cdots x_2^d \\ \vdots & \vdots \vdots \ddots \vdots \\ x_i^1 & x_i^2 \cdots x_i^j \cdots x_i^d \\ \vdots & \vdots \vdots \ddots \vdots \\ x_n^1 & x_n^2 \cdots x_n^j \cdots x_n^d \end{bmatrix}, \begin{cases} i = 1, 2, \ldots, n \\ j = 1, 2, \ldots, d \end{cases} \tag{1}$$

$$x_i^j = x_{i,\min}^j + rand.\left(x_{i,\max}^j - x_{i,\min}^j\right), \begin{cases} i = 1, 2, \ldots, n \\ j = 1, 2, \ldots, d \end{cases} \tag{2}$$

where n indicates the total number of ravens and wolves (solution candidates) on a place in the earth (search space); d is the considered problem's dimension; x_i^j is the jth decision variable for determining the initial position of the ith candidate; $x_{i,\min}^j$ and $x_{i,\max}^j$ are the lower and upper bounds of the jth variable in the ith candidate; *rand* is a random number with a uniform distribution in the range [0,1].

Then, the function evaluation is conducted for all of the solution candidates in the search space, which demonstrates the Hunger Level (HL) of the ravens and wolves as follows:

$$HL = \begin{bmatrix} HL_1 \\ HL_2 \\ \vdots \\ HL_i \\ \vdots \\ HL_n \end{bmatrix}, \{i = 1, 2, \ldots, n \tag{3}$$

where HL_i is the objective function values of ith solution candidate in the search space. n indicates the total number of ravens and wolves (solution candidates) on a place on the earth (search space).

In the next step, a random integer number (N_r) is generated in the range of [1, n], demonstrating the total number of ravens in the search space. Based on the wolf-bird concept in nature, ravens are the hungriest creatures trying to search for prey. The solution candidates with the lowest objective function values are assumed to be ravens in the search space. In other words, the first N_r solution candidates with the lowest HL values are considered ravens. The position vectors of the ravens are as follows:

$$X^r = \begin{bmatrix} X_1^r \\ X_2^r \\ \vdots \\ X_i^r \\ \vdots \\ X_{N_r}^r \end{bmatrix} = \begin{bmatrix} x_1^1 & x_1^2 \cdots x_1^j \cdots x_1^d \\ x_2^1 & x_2^2 \cdots x_2^j \cdots x_2^d \\ \vdots & \vdots \vdots \ddots \vdots \\ x_i^1 & x_i^2 \cdots x_i^j \cdots x_i^d \\ \vdots & \vdots \vdots \ddots \vdots \\ x_{N_r}^1 & x_{N_r}^2 \cdots x_{N_r}^j \cdots x_{N_r}^d \end{bmatrix}, \begin{cases} i = 1, 2, \ldots, N_r. \\ j = 1, 2, \ldots, d. \end{cases} \tag{4}$$

where N_r indicates the total number of ravens on a place on the earth; d is the considered problem's dimension; x_i^j is the jth decision variable of ith raven.

The Hunger Level of ravens is calculated as follows, which are the first N_r solution candidates with the highest hunger level (lowest objective function values):

$$HL^r = \begin{bmatrix} HL_1^r \\ HL_2^r \\ \vdots \\ HL_i^r \\ \vdots \\ HL_{N_r}^r \end{bmatrix}, \{i = 1, 2, \ldots, \quad N_r \tag{5}$$

where HL_i^r is the Hunger Level of ith raven, and N_r indicates the total number of ravens in the search space.

Regarding the fact that a total number of N_w or n-N_r wolves also exist in the search space; the position vector and hunger level of the wolves are as follows:

$$X^w = \begin{bmatrix} X_1^w \\ X_2^w \\ \vdots \\ X_i^w \\ \vdots \\ X_{N_w}^w \end{bmatrix} = \begin{bmatrix} x_1^1 & x_1^2 \cdots x_1^j \cdots x_1^d \\ x_2^1 & x_2^2 \cdots x_2^j \cdots x_2^d \\ \vdots & \vdots \vdots \ddots \vdots \\ x_i^1 & x_i^2 \cdots x_i^j \cdots x_i^d \\ \vdots & \vdots \vdots \ddots \vdots \\ x_{N_w}^1 & x_{N_w}^2 \cdots x_{N_w}^j \cdots x_{N_w}^d \end{bmatrix}, \begin{cases} i = 1, 2, \ldots, N_w. \\ j = 1, 2, \ldots, d. \end{cases} \tag{6}$$

$$HL^w = \begin{bmatrix} HL_1^w \\ HL_2^w \\ \vdots \\ HL_i^w \\ \vdots \\ HL_{N_r}^w \end{bmatrix}, \{i = 1, 2, \ldots, N_w \tag{7}$$

where N_w indicates the total number of wolves in the search space; d is the considered problem's dimension; x_i^j is the jth decision variable of the ith wolf; HL_i^w is the Hunger Level of the ith wolf.

Exploration.

Each of the ravens in the search space flies across the woods to perform an intelligent search of prey to transmit signals to the wolves in the neighborhood. In the first phase of the algorithm, the best raven, which is the hungriest among other ravens with the lowest objective function value (HL_1^r), tries to fly around itself because the hungriest raven cannot fly around all over the search space due to its higher level of hungriness and tiredness (Fig. 1). For this purpose, an intelligent random walk is implemented in the WBO as follows:

$$X(t)^r = [0, cumsum(2r(t_1) - 1), cumsum(2r(t_2) - 1), \ldots, cumsum(2r(t_n) - 1)] \tag{8}$$

where *cumsum* calculates the cumulative sum, n shows iterations' maximum number, t is the step of random walk (iteration in the current study), and r(t) elucidates a stochastic function represented as follows:

$$r(t) = \begin{cases} 1 if & rand > 0.5 \\ 0 if & rand"0.5 \end{cases} \tag{9}$$

where t is the step of the random walk (iteration in the current paper), and rand shows a random number generated with uniform distribution in the interval of [0,1].

In order to mathematically model this random walk around the best raven, the upper and lower bound of the search space are moved around the considered raven to create random steps in the close neighborhood. Meanwhile, an intelligent ratio is defined as follows, which mimics the tiredness of the ravens during the optimization process regarding the fact that ravens cannot fly to far distances regarding their hungriness. The mathematical presentation of this aspect is as follows:

$$I = 10^w \frac{t}{T} \tag{10}$$

where w is a constant determined depending on the current iteration (w = 2 when t > 0.1 T, w = 3 when t > 0.5 T, w = 4 when t > 0.75 T, w = 5 when t > 0.9 T, and w = 6 when t > 0.95 T), t is the current iteration, T is the maximum number of iterations. The exploitation precision level is adjustable by the constant w.

Meanwhile, a random integer is created in the range of [1, n], indicating the total number of intelligent random walks by the best raven in the search space, which demonstrates the total number of preys in the search space. The position vectors and the hunger level of the preys are as follows:

Fig. 2. The schematic presentation of wolves reaching the prey.

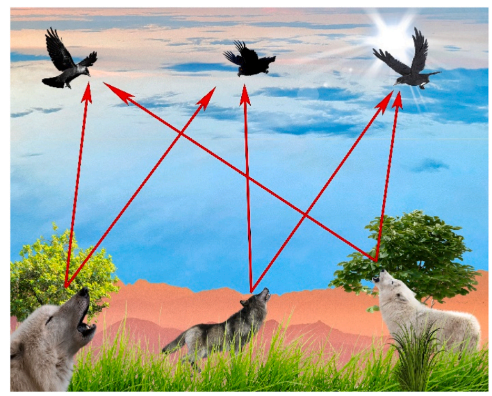

Fig. 3. The schematic presentation of wolves following the ravens.

$$X^p = \begin{bmatrix} X_1^p \\ X_2^p \\ \vdots \\ X_i^p \\ \vdots \\ X_{N_p}^p \end{bmatrix} = \begin{bmatrix} x_1^1 & x_1^2 \cdots x_1^j \cdots x_1^d \\ x_2^1 & x_2^2 \cdots x_2^j \cdots x_2^d \\ \vdots & \vdots \vdots \ddots \vdots \\ x_i^1 & x_i^2 \cdots x_i^j \cdots x_i^d \\ \vdots & \vdots \vdots \ddots \vdots \\ x_{N_p}^1 & x_{N_p}^2 \cdots x_{nN_p}^j \cdots x_{N_p}^d \end{bmatrix}, \begin{cases} i = 1, 2, \ldots, N_p \\ j = 1, 2, \ldots, d \end{cases} \tag{11}$$

$$HL^p = \begin{bmatrix} HL_1^p \\ HL_2^p \\ \vdots \\ HL_i^p \\ \vdots \\ HL_{N_p}^p \end{bmatrix}, \{i = 1, 2, \ldots, N_p \tag{12}$$

where Np indicates the total number of prey found by the best raven in the search space; d is the considered problem's dimension; x_i^j is the jth decision variable of the ith prey; HL_i^p is the Hunger Level of the ith prey.

Due to the intelligent act of ravens in nature, they try to control the movements of the prey around themselves to determine the position of

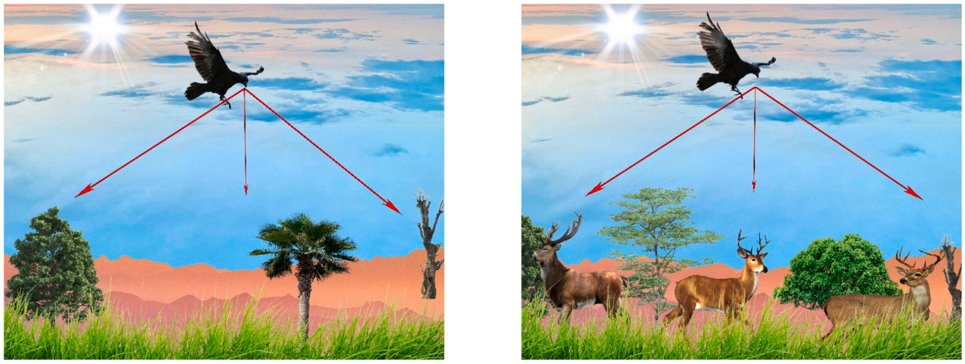

Fig. 4. The schematic presentation of nearby Levy flight by the other ravens to find prey.

prey with the lowest possible movements, which demonstrates that the prey is hungry and is an excellent choice to be captured by wolves. In other words, the preys with the lowest movements are the preys with higher levels of hungriness, so the best raven tries to move toward the prey by conducting the following position-updating process:

$$PCP = \frac{\sum_{i=1}^{N_p} X_i^P}{N_p}, i = 1, 2, \ldots, N_p \tag{13}$$

$$^{New}X_1^r = X_1^r + r_1 \times X_1^p - r_2 \times PCP \tag{14}$$

where *Np* is the number of preys; *PCP* is the preys' center point which mimics the crowd of prey around the best raven; $^{New}X_1^r$ shows the upcoming position vector of the best raven (X_1^r) as the hungriest raven; X_1^p indicates the hungriest prey with the lowest objective function value; r_1 and r_2 are two random numbers in the range of [0,1].

In order to distinguish the behavior of wolves in the search space, the total distance (D) between the wolves and the new position of the hungriest raven ($^{New}X_1^r$) is calculated as follows:

$$D_i = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}, \{i = 1, 2, \ldots, N_w \tag{15}$$

where D_i is the total distance between the lth wolf and the new position of the best raven ($^{New}X_1^r$); N_w is the total number of wolves in the search space; (x_1,y_1) and (x_2,y_2) represent the coordinates of the wolves and the best raven in the search space.

Then, the wolves try to reach the best raven by paying attention to the guggle song of the best raven. For this purpose, the best raven tries to send an intelligent signal to the wolves by measuring its overall hungriness ($^{New}HL_1^r$). Regarding this intelligent act, if the new hunger level of the best raven is higher than its former hunger level or if the new objective function value is lower than the former one by considering a predefined tolerance of α= 1.05 ($^{New}HL_1^r < \alpha HL_1^r$), the raven's signals appear to be stronger. In this situation, the wolves try to reach the best raven and the hungriest prey by conducting the following position-updating process (Fig. 2):

$$^{New}X_i^w = X_i^W + \frac{r_1 \times {}^{New}X_1^r - r_2 \times X_1^p}{D_i} \quad i = 1, 2, \ldots, N_w \tag{16}$$

where $^{New}X_i^w$ shows the upcoming position vector of the *i*th wolf (X_i^W); $^{New}X_1^r$ shows the new position vector of the best raven; d_i is the distance between the *i*th wolf and the best raven; X_1^p indicates the hungriest prey with the lowest objective function value; r_1 and r_2 are two random numbers in the range of [0,1].

If the new hunger level of the best raven is lower than its former hunger level or if the new objective function value is higher than the former one ($^{New}HL_1^r \geq \alpha HL_1^r$), the raven's signals appear to be weaker. In this situation, the wolves try to follow the direction of all ravens in the search space by conducting the following position updating process (Fig. 3):

$$RCP = \frac{\sum_{i=1}^{N_r} X_i^r}{N_r}, i = 1, 2, \ldots, N_r \tag{17}$$

$$^{New}X_i^w = X_i^W + \frac{r_1 \times RCP - r_2 \times X_1^r}{D_i} \quad i = 1, 2, \ldots, N_w \tag{18}$$

where *Nr* is the number of ravens; *RCP* is the ravens' center point which mimics the crowd of ravens; $^{New}X_1^W$ shows the upcoming position vector of the *i*th wolf (X_i^W); D_i is the distance between the *i*th wolf and the best raven; X_1^r shows the position vector of the best raven; r_1 and r_2 are two random numbers in the range of [0,1].

Exploitation

In the second phase of the algorithm, the other ravens, which are not the hungriest but are also hungry enough for food search, in the same manner, are considered for conducting a position updating process. For this purpose, the ravens try to fly all over the search space to find prey and send signals to the wolves in the neighborhood. Regarding the fact that ravens are among intelligent creatures, the mentioned flight is mathematically modeled by using the Lévy flight concept as one of the newly developed random walks in which the step lengths are determined by Lévy distribution as follows:

$$L(s) \sim |s|^{-1-\beta} \tag{19}$$

where $0 < \beta \leq 2$ is an index. The Lévy distribution should be described mathematically in the following Fourier transform.

$$F(k) = \exp\left[-\alpha|k|^\beta\right], 0 < \beta \leq 2 \tag{20}$$

where α is a scale parameter, except in a few specific instances, the inverse of this integral is complex since it lacks analytical form. The case $\beta = 2$ corresponds to a Gaussian distribution, while $\beta = 1$ leads to a Cauchy distribution. For the general case, the inverse integral:

$$L(s) = \frac{1}{\pi}\int_0^\infty \cos(ks)\exp\left[-\alpha|k|^\beta\right]dk \tag{21}$$

It can be estimated only when s is large. We have

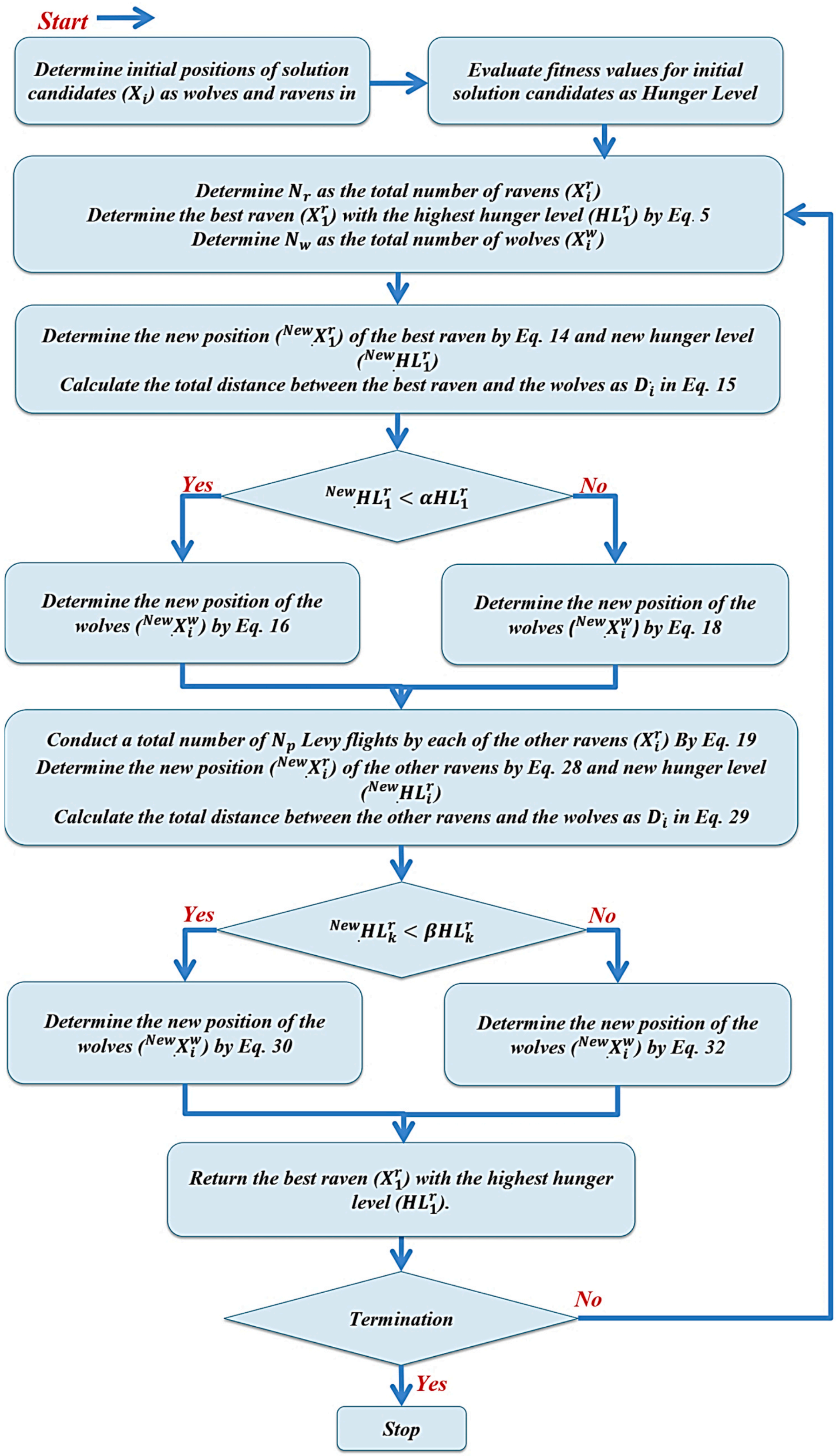

Fig. 5. Flowchart of Wolf-Bird Optimizer.

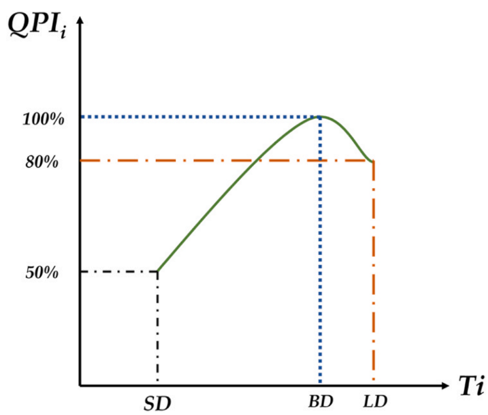

Fig. 6. The quality performance index.

$$L(s) \rightarrow \frac{\alpha\beta\tau(\beta)\sin(\frac{\pi\beta}{2})}{\pi|s|^{1+\beta}}, s \rightarrow \infty. \tag{22}$$

where $\tau(z)$ is the Gamma function.

$$\tau(z) = \int_0^\infty t^{z-1} e^{-t} dt \tag{23}$$

In the case when $z = n$ is an integer, we have $\tau(z) = (n-1)!$..

In exploring unknown, large-scale search space, Lévy flights are more efficient than Brownian random walks. Some factors contribute to this efficiency, including the variation in Lévy flights.

$$\sigma^2(t) \sim t^{3-\beta}, 1 \le \beta \le 2 \tag{24}$$

Brownian random walks have a linear relationship (i.e., $\sigma^2(t) \sim t$) that rises far quicker than the exponential relationship. It is worth noting that a power-law distribution is often associated with scale-free properties, and Lévy flights may therefore exhibit self-similarity and fractal behavior.

For computational purposes, a random integer is created in the range of $[1, n]$, indicating the total number of intelligent Levy flights by each of the other ravens in the search space, demonstrating the total number of preys in the search space (Fig. 4). The position vectors and the hunger level of the preys are as follows:

$$X^p = \begin{bmatrix} X_1^p \\ X_2^p \\ \vdots \\ X_i^p \\ \vdots \\ X_{N_p}^p \end{bmatrix} = \begin{bmatrix} x_1^1 & x_1^2 \cdots x_1^j \cdots x_1^d \\ x_2^1 & x_2^2 \cdots x_2^j \cdots x_2^d \\ \vdots & \vdots\vdots\vdots \ddots \vdots \\ x_i^1 & x_i^2 \cdots x_i^j \cdots x_i^d \\ \vdots & \vdots\vdots\vdots \ddots \vdots \\ x_{N_p}^1 & x_{N_p}^2 \cdots x_{nN_p}^j \cdots x_{N_p}^d \end{bmatrix}, \begin{cases} i = 1, 2, \ldots, N_p. \\ j = 1, 2, \ldots, d. \end{cases} \tag{25}$$

$$HL^p = \begin{bmatrix} HL_1^p \\ HL_2^p \\ \vdots \\ HL_i^p \\ \vdots \\ HL_{N_p}^p \end{bmatrix}, \{i = 1, 2, \ldots, N_p \tag{26}$$

where Np indicates the total number of preys found by each of the other ravens (solution candidates) on a place in the earth (search space); d is the considered problem's dimension; x_i^j is the jth decision variable of the ith prey; HL_i^p is the Hunger Level of the ith prey.

Regarding the fact that the preys with the lowest movements are the preys with higher levels of hungriness, so each of the other ravens tries to move toward the prey by conducting the following position-updating process:

$$PCP = \frac{\sum_{i=1}^{N_p} X_i^P}{N_p}, i = 1, 2, \ldots, N_p \tag{27}$$

$${}^{New}X_k^r = X_k^r + r_1 \times X_1^p - r_2 \times PCP \quad k = 2, \ldots, N_r \tag{28}$$

where Np is the number of preys; PCP is the preys center point which mimics the crowd of prey around the kth raven; ${}^{New}X_k^r$ shows the upcoming position vector of the kth raven (X_k^r) as one of the other hungry ravens; X_1^p indicates the hungriest prey with the lowest objective function value; r_1 and r_2 are two random numbers in the range of [0,1].

Like the previous stage of the algorithm, the distances of wolves (D_i) from the other ravens (${}^{New}X_k^r$) in the search space are determined as follows:

$$D_i = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}, \{i = 1, 2, \ldots, N_w. \tag{29}$$

where D_i is the total distance between the lth wolf and the new position of the kth raven (${}^{New}X_k^r$); N_w is the total number of wolves in the search space, and (x_1, y_1) and (x_2, y_2) represent the coordinates of the wolves and the ith raven in the search space.

Finally, the wolves try to reach the other ravens in the search space by conducting the following position updating process while a tolerance of Beta (1.05) is utilized for this purpose. If ${}^{New}HL_k^r < \beta HL_k^r$, the ravens' signals appear to be stronger, so in this situation, the wolves try to reach the best raven and the hungriest prey by conducting the following position updating process (Fig. 3):

$${}^{New}X_i^w = X_i^W + \frac{r_1 \times {}^{New}X_k^r - r_2 \times X_1^p}{D_i} \quad \begin{cases} i = 1, 2, \ldots, N_w \\ k = 2, \ldots, N_r \end{cases} \tag{30}$$

where ${}^{New}X_i^w$ shows the upcoming position vector of the ith wolf (X_i^W); ${}^{New}X_k^r$ shows the new position vector of the ravens; D_i is the distance between the ith wolf and the raven; X_1^p indicates the hungriest prey with the lowest objective function value; r_1 and r_2 are two random numbers in the range of [0,1].

If the new hunger level of the raven is higher than the former hungriest level of the raven by considering the tolerance of Beta (${}^{New}HL_k^r \ge \beta HL_k^r$), the raven's signals appear to be weaker. In this situation, the wolves try to follow the direction of all ravens in the search space by conducting the following position updating process (Fig. 2):

$$RCP = \frac{\sum_{i=1}^{N_r} X_i^r}{N_r}, i = 1, 2, \ldots, N_r \tag{31}$$

$${}^{New}X_i^w = X_i^W + \frac{r_1 \times RCP - r_2 \times X_k^r}{D_i} \quad \begin{cases} i = 1, 2, \ldots, N_w \\ k = 2, \ldots, N_r \end{cases} \tag{32}$$

where Nr is the number of ravens; RCP is the ravens' center point which mimics the crowd of ravens; ${}^{New}X_1^w$ shows the upcoming position vector of the ith wolf (X_i^W); d_i is the distance between the ith wolf and the best raven; X_k^r shows the position vector of the other hungry ravens; r_1 and r_2 are two random numbers in the range of [0,1].

After the exploration and exploitation phases of the algorithm, the boundary violation control is conducted for all of the new position vectors while the hunger level of the candidates is also calculated, and the best raven is represented as the best solutions candidate of the optimization process. The pseudo-code and flowchart of the WBO are presented in Algorithm 1 and Fig. 5, respectively.

Algorithm 1. The pseudo-code of the WBO.

Procedure Wolf-Bird Optimizer (WBO)

Determine initial positions of solution candidates (X_i) as wolves and ravens in the search space
Evaluate fitness values for initial solution candidates as Hunger Level (HL)
while Iteration (Number of Function Evaluation) < Maximum number of iterations (Function Evaluations)
  Determine N_r as the total number of ravens (X_i^r)
  Determine the best raven (X_1^r) with the highest hunger level (HL_1^r) by Eq. 5
  Determine N_w as the total number of wolves (X_i^w)
  Conduct a total number of N_p random walks by the best raven (X_1^r) by Eq. 8
  Determine the new position (${}^{New}X_1^r$) of the best raven by Eq. 14 and new hunger level (${}^{New}HL_1^r$)
  Calculate the total distance between the best raven and the wolves as D_i in Eq. 15
    for i=1:N_w
      if ${}^{New}HL_1^r < \alpha HL_1^r$
      Determine the new position of the wolves (${}^{New}X_i^w$) by Eq. 16
      else
      Determine the new position of the wolves (${}^{New}X_i^w$) by Eq. 18
      end
    end
  Conduct a total number of N_p Levy flights by each of the other ravens (X_i^r) By Eq. 19
  Determine the new position (${}^{New}X_i^r$) of the other ravens by Eq. 28 and new hunger level (${}^{New}HL_i^r$)
  Calculate the total distance between the other ravens and the wolves as D_i in Eq. 29
   for k=2:N_r
    for i=1:N_w
      if ${}^{New}HL_k^r < \beta HL_k^r$
      Determine the new position of the wolves (${}^{New}X_i^w$) by Eq. 30
      else
      Determine the new position of the wolves (${}^{New}X_i^w$) by Eq. 32
      end
    end
   end
end while
return the best raven (X_1^r) with the highest hunger level (HL_1^r).

end Procedure

2.3. Analysis of key variables and processes in Wolf-Bird Optimization algorithm

Firstly, the initialization process and the determination of each candidate's position within the search space are crucial. The decision variables x_i^j, and their bounds $x_{i,\max}^j$ and $x_{i,\min}^j$, define the range and granularity of the search space. The random number generation in this stage introduces variability and diversity among the solution candidates. This diversity is essential for a comprehensive exploration of the search space, especially in complex optimization problems where the global optimum might be surrounded by multiple local optima. The

Table 1 –
Project Information.

Number	Activity	Logical	Mode 1					Mode 2					Mode 3					Mode 4					Mode 5				
			Time	Cost $	Quality %	Risk	CO_2	Time	Cost $	Quality %	Risk	CO_2	Time	Cost $	Quality %	Risk	CO_2	Time	Cost $	Quality %	Risk	CO_2	Time	Cost $	Quality %	Risk	CO_2
1	Foundation	-	26.00	8100.00	90.65	14.97	225.33	24.00	7850.00	89.20	12.00	198.45	20.00	8120.00	92.10	12.50	187.52	15.00	8400.00	78.90	12.90	98.32	13.00	9408.00	74.96	16.31	108.15
2	Retaining wall	1FS+ 1	15.00	2252.00	94.91	13.22	137.97	13.00	2150.00	94.51	10.50	125.08	11.00	2220.00	95.30	11.30	111.04	9.00	2410.00	87.10	11.54	54.25	8.00	2699.20	82.75	14.41	59.68
3	Columns of ground	2FS	13.00	2015.00	91.16	10.33	116.31	10.00	1980.00	90.21	8.00	101.30	7.00	2042.00	92.10	9.40	98.00	6.00	2100.00	85.45	9.50	36.32	5.00	2352.00	81.18	11.26	39.95
4	Beam and roof of the ground	3FS+ 1	10.00	4325.00	91.98	11.95	188.28	8.00	3652.00	91.40	9.65	169.91	6.00	3920.00	92.56	9.80	152.36	4.00	4150.00	86.41	10.30	111.25	3.00	4648.00	82.09	13.03	122.38
5	Columns of 1st floor	4FS+ 2	13.00	1550.00	93.61	5.58	190.88	10.00	1200.00	92.65	4.20	178.35	7.00	1356.00	94.56	5.40	148.00	6.00	1420.00	89.36	6.00	128.60	5.00	1590.40	84.89	6.08	141.46
6	Beam and roof of 1st floor	5FS+ 1	10.00	3600.00	95.63	12.82	177.77	8.00	3200.00	94.80	10.30	177.88	6.00	3410.00	96.45	10.65	125.36	4.00	3540.00	85.45	11.02	45.25	3.00	3964.80	81.18	13.97	49.78
7	Columns of 2nd floor	6FS+ 2	13.00	1550.00	92.04	8.04	158.51	10.00	1200.00	91.30	6.32	143.65	7.00	1356.00	92.78	7.05	127.63	6.00	1420.00	84.12	7.80	35.98	5.00	1590.40	79.91	8.76	39.58
8	Beam and roof of 2nd floor	7FS+ 1	10.00	3600.00	97.58	9.28	183.86	8.00	3200.00	96.50	7.25	169.25	6.00	3410.00	98.65	8.25	145.25	4.00	3540.00	88.89	8.50	89.54	3.00	3964.80	84.45	10.11	98.49
9	Columns of 3rd floor	8FS+ 2	13.00	1550.00	93.99	6.90	150.19	10.00	1200.00	93.40	5.30	145.25	7.00	1356.00	94.58	6.40	111.25	6.00	1420.00	78.45	6.45	74.63	5.00	1590.40	74.53	7.52	82.09
10	Beam and roof of 3rd floor	9FS+ 1	10.00	3600.00	91.48	3.54	167.47	8.00	3200.00	90.50	2.65	151.72	6.00	3410.00	92.45	3.47	134.89	4.00	3540.00	82.10	3.90	125.25	3.00	3964.80	78.00	3.86	137.78
11	Columns of the 4th floor	10FS+ 2	13.00	1550.00	92.83	6.32	114.52	10.00	1200.00	91.40	4.50	106.58	7.00	1356.00	94.25	6.80	89.25	6.00	1420.00	86.45	7.00	65.32	5.00	1590.40	82.13	6.89	71.85
12	Beam and roof of the 4th floor	11FS+ 1	10.00	3600.00	96.38	15.30	156.73	8.00	3200.00	95.30	11.85	143.56	6.00	3410.00	97.45	13.90	124.58	4.00	3540.00	91.20	14.20	43.56	3.00	3964.80	86.64	16.68	47.92
13	Columns of the 5th floor	12FS+ 2	13.00	1550.00	95.32	11.85	163.65	10.00	1200.00	94.62	9.45	144.32	7.00	1356.00	96.01	10.02	135.98	6.00	1420.00	86.41	11.30	97.20	5.00	1590.40	82.09	12.91	106.92
14	Beam and roof of the 5th floor	13FS+ 1	10.00	3600.00	98.57	4.69	139.11	8.00	3200.00	97.40	3.21	126.98	6.00	3410.00	99.74	5.40	111.04	4.00	3540.00	91.02	5.52	56.98	3.00	3964.80	86.47	5.11	62.68
15	Columns of a ridge roof	14FS+ 1	5.00	420.00	91.82	5.85	124.31	3.00	356.00	91.60	4.25	114.25	2.00	411.00	92.03	6.08	98.40	1.00	580.00	83.25	6.85	75.98	1.00	649.60	79.09	6.38	83.58
16	Beam and roof of ridge floor	15FS+ 1	6.00	1110.00	92.96	3.34	168.63	4.00	980.00	92.45	2.51	156.32	3.00	995.00	93.47	3.25	132.07	2.00	1020.00	87.98	3.65	100.36	2.00	1142.40	83.58	3.64	110.40
17	Brickworks of ground	4FS+ 1	14.00	1620.00	94.04	1.66	166.89	11.00	1480.00	93.00	1.05	157.45	9.00	1620.00	95.07	2.14	127.80	8.00	1740.00	79.99	2.45	98.65	7.00	1948.80	75.99	1.81	108.52
18	Mechanical installations of ground	17FS+ 2	10.00	1300.00	95.36	8.32	109.08	8.00	1220.00	94.50	6.50	101.98	6.00	1352.00	96.21	7.40	84.52	4.00	1480.00	82.14	7.65	24.65	3.00	1657.60	78.03	9.07	27.12
19	Electrical installations of ground	17FS+ 2	15.00	1250.00	95.54	6.08	128.76	13.00	1100.00	95.30	4.90	121.07	9.00	1260.00	95.78	5.01	99.04	6.00	1350.00	89.65	5.63	68.42	5.00	1512.00	85.17	6.63	75.26
20	Brickworks of 1st floor	6FS+ 1	14.00	1800.00	92.21	5.15	125.95	11.00	1620.00	90.70	3.54	114.06	9.00	1870.00	93.72	5.89	101.50	8.00	1942.00	80.45	6.00	45.65	7.00	2175.04	76.43	5.61	50.22
21	Mechanical installations of 1st floor	20FS+ 2	10.00	1600.00	97.53	5.93	130.92	8.00	1520.00	97.00	4.22	125.97	6.00	1710.00	98.05	6.41	97.65	4.00	1780.00	91.45	6.54	82.63	3.00	1993.60	86.88	6.47	90.89
22	Electrical installations of 1st floor	20FS+ 2	9.00	1420.00	97.65	3.79	167.23	7.00	1350.00	96.40	2.87	151.26	5.00	1420.00	98.90	3.61	134.95	4.00	1500.00	87.26	3.75	111.52	3.00	1680.00	82.90	4.13	122.67
23	Brickworks of 2nd floor	8FS+ 1	14.00	1800.00	93.50	5.55	193.39	11.00	1620.00	92.30	4.20	178.32	9.00	1870.00	94.69	5.30	152.47	8.00	1942.00	83.45	5.50	97.52	7.00	2175.04	79.28	6.05	107.27
24	Mechanical installations of 2nd floor	23FS+ 2	10.00	1680.00	94.93	12.07	138.67	8.00	1532.00	94.15	9.34	126.47	6.00	1750.00	95.71	10.98	110.80	4.00	1780.00	88.98	11.36	64.52	3.00	1993.60	84.53	13.15	70.97

(*continued on next page*)

Table 1 – (*continued*)

Number	Activity	Logical	Mode 1					Mode 2					Mode 3					Mode 4					Mode 5				
25	Electrical installations of the 2nd floor	23FS+ 2	9.00	1420.00	92.55	10.74	181.74	7.00	1350.00	90.47	8.45	175.65	5.00	1420.00	94.63	9.41	134.74	4.00	1500.00	78.32	9.50	86.52	3.00	1680.00	74.40	11.71	95.17
26	Brickworks of 3rd floor	10FS+ 1	14.00	1800.00	94.16	2.46	165.55	11.00	1620.00	93.32	1.65	149.08	9.00	1870.00	95.00	2.91	134.29	8.00	1942.00	85.65	3.20	98.42	7.00	2175.04	81.37	2.68	108.26
27	Mechanical installations of 3rd floor	26FS+ 2	10.00	1680.00	91.82	2.87	178.69	8.00	1530.00	91.24	2.04	170.36	6.00	1740.00	92.40	3.09	134.95	4.00	1780.00	86.97	5.20	74.77	3.00	1993.60	82.62	3.12	82.25
28	Electrical installations of the 3rd floor	26FS+ 2	9.00	1420.00	90.44	8.19	159.03	7.00	1350.00	90.00	6.45	156.65		1420.00	90.87	7.14	114.78	4.00	1500.00	82.42	7.65	64.52	3.00	1680.00	78.30	8.92	70.97
29	Brickworks on the 4th floor	12FS+ 1	14.00	1800.00	96.16	12.95	159.09	11.00	1620.00	94.98	10.32	142.36	9.00	1870.00	97.33	11.00	130.02	8.00	1942.00	86.41	11.40	111.78	7.00	2175.04	82.09	14.12	122.96
30	Mechanical installations of the 4th floor	29FS+ 2	10.00	1695.00	93.38	8.26	163.88	8.00	1570.00	92.63	6.40	153.21	6.00	1760.00	94.12	7.50	126.97	4.00	1780.00	86.35	7.70	42.63	3.00	1993.60	82.03	9.00	46.89
31	Electrical installations of the 4th floor	29FS+ 2	9.00	1420.00	94.63	6.65	158.89	7.00	1350.00	94.17	4.98	147.36	5.00	1420.00	95.09	6.50	124.36	4.00	1500.00	87.42	6.52	35.59	3.00	1680.00	83.05	7.25	39.15
32	Brickworks on the 5th floor	14FS+ 1	14.00	1800.00	93.02	4.89	128.85	11.00	1620.00	92.83	3.45	120.32	9.00	1870.00	93.21	5.34	99.99	8.00	1942.00	88.20	5.98	65.42	7.00	2175.04	83.79	5.32	71.96
33	Mechanical installations of the 5th floor	32FS+ 2	10.00	1680.00	94.03	3.14	124.29	8.00	1530.00	93.40	2.09	111.14	6.00	1740.00	94.65	3.77	101.65	4.00	1780.00	85.72	3.89	85.41	3.00	1993.60	81.43	3.42	93.95
34	Electrical installations of the 5th floor	32FS+ 2	9.00	1420.00	95.07	2.35	213.33	7.00	1350.00	94.42	1.52	199.32	5.00	1420.00	95.71	2.95	165.42	4.00	1500.00	90.45	3.02	123.65	3.00	1680.00	85.93	2.56	136.02
35	Rooftop	34FS	15.00	935.00	93.62	8.64	188.61	10.00	870.00	92.41	6.47	178.65	7.00	890.00	94.83	8.45	143.68	5.00	920.00	80.65	9.20	99.98	4.00	1030.40	76.62	9.42	109.98
36	Elevator	34FS+ 2	17.00	2400.00	90.81	7.13	105.35	15.00	2150.00	90.56	5.24	100.36	11.00	2350.00	91.05	7.23	79.65	8.00	2680.00	82.42	7.77	24.63	7.00	3001.60	78.30	7.77	27.09
37	Facade	34FS+ 5	55.00	5320.00	91.58	4.35	194.41	52.00	4580.00	91.15	3.12	189.32	37.00	5120.00	92.00	4.63	142.62	29.00	5980.00	79.00	4.97	75.63	25.00	6697.60	75.05	4.74	83.19
38	Outdoors	35FS+ 1	37.00	2420.00	92.63	11.96	143.95	32.00	2100.00	91.78	9.12	134.65	25.00	2850.00	93.48	11.25	111.45	19.00	3412.00	84.53	11.32	80.25	16.00	3821.44	80.30	13.03	88.28

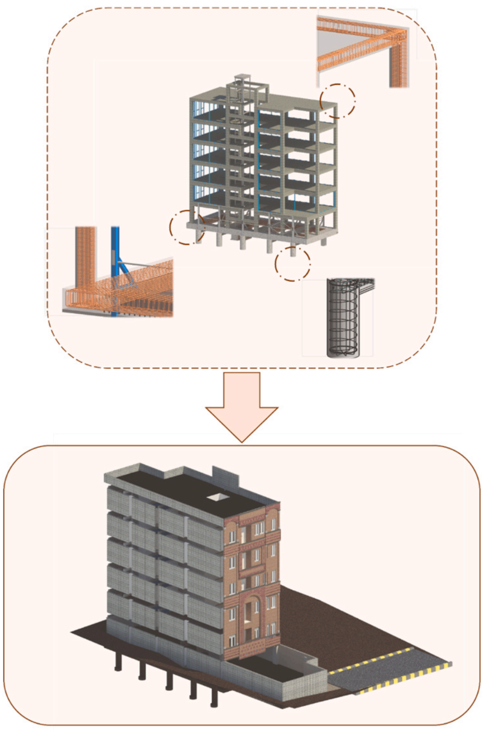

Fig. 7. BIM-based modeling of the case study project.

balance between exploring new areas of the search space (exploration) and intensifying the search around promising regions (exploitation) is largely influenced by how these initial positions are set and randomized.

The classification of solution candidates into ravens and wolves based on their Hunger Levels (*HL*) is another critical aspect. This step implicitly determines the algorithm's focus, as ravens (with lower HL values) are considered better solutions and are given priority in the search process. The number of ravens, determined by the random integer N_r, can significantly influence the algorithm's dynamics. A higher number of ravens might lead to a more intensive exploitation of the best solutions found so far, while a lower number might encourage more exploration by giving wolves (lesser solutions) more prominence. This balance is pivotal in preventing premature convergence to local optima and ensuring a thorough search of the solution space.

Table 2 –
Parameters of the algorithms used in this study.

Algorithm	Parameters
BA	Frequency Range: [0,2], Loudness (A): 0.5, Pulse Rate (*r*): 0.5
BOA	Sensory Modality (c): 1, Power Exponent (*p*): 2
CPA	number of aphids (*nA*): 60, number of colonies (*nC*): 4, Female (*Fr*): [0.1, 0.9]
FPA	Recombination Rate: 0.8, Switch Probability (*p*): 0.1, Gamma: 1.5
HHO	Step Size: 1.5
JA	Parameter-Free
KHA	Beta: [0.002, 0.010] (ms^{-1}), crossover: 0.2

Finally, the exploration and exploitation mechanisms, particularly the use of Lévy flights and the intelligent random walk, are innovative elements of the WBO algorithm. The efficiency of Lévy flights in exploring unknown, large-scale search spaces is crucial, especially in higher-dimensional problems. The step lengths determined by the Lévy distribution enable the algorithm to jump out of local optima, exploring more distant regions of the search space. Simultaneously, the intelligent random walk around the best raven, constrained by variables such as tiredness and hungriness levels, mimics a more focused search. This dual approach allows the algorithm to balance between broad-ranging search patterns and localized intensive searches, adapting to the problem's landscape.

In summary, each variable and process in the WBO algorithm plays a distinct role in shaping its search strategy. Understanding these roles and their interplay is key to appreciating the algorithm's potential and limitations, and is essential for its effective application to various optimization problems.

Problem. definition and formulation

Three main components make up the framework as follows:

- The decision variables and initialization Module;
- The BIM Module;
- The novel metaheuristic optimization algorithm Module.

2.4. Decision variables and initialization module

A building project's activity-on-node (AON) diagram consists of M nodes, and arrows represent the connections among activities. Every task can be completed miscellaneously, and depending on the number of resources, technology, and equipment used, each has its own cost, time, risk, quality, and CO_2 emissions. The TCRQC tradeoff problem optimization technique attempts to improve project quality while reducing project time, cost, risk, and CO_2 emissions by selecting the appropriate course of action for each activity. As a result, the first objective function in Eq. 33 is to shorten the project's duration, which is represented by T_p:

$$\textit{Minimum} \quad T_p = \min(\max(ST_i + D_i)) = \min(\max(FT_i)); i = 1, \dots, M \tag{33}$$

where D_i represents each activity's length; ST_i and FT_i show activities' start and finish times, respectively; M elucidates project scheduling's overall number of nodes[137].

In addition, the overall cost of a project includes indirect costs (IC), direct costs (DC), and delay costs (DC). The cost of the materials, equipment, and resources needed to carry out tasks is included in the direct costs. The costs applied throughout the project, such as management, administration, and insurance, are referred to as indirect costs and are fixed [138]. Other methods exist for assessing the total project's cost differently; however, this research only considers direct, indirect, and delay costs. Eq. 34 illustrates the objective function that reduces the overall cost of the project as follows:

$$\text{Minimum} \quad TC_p = D^j_{C_i} + I^j_{C_i} + DC \tag{34}$$

$$D^j_{C_i} = \sum_{i=1}^{n} C^j_i \tag{35}$$

Table 3 –
The best findings of the WBO and other methods in the case study.

	BA	BOA	CPA	FPA	HHO	JA	KHA	WBO
Time	295.00	325.00	258.00	280.00	258.00	295.00	307.00	258.00
Cost	118,308.00	118,696.00	118,107.80	117,950.80	118,279.00	119,033.04	117,844.00	117,804.00
Quality	94.02	94.23	94.31	94.19	94.62	93.75	94.66	94.66
Risk	6.31	6.01	6.13	6.25	5.79	6.51	6.17	5.79
CO_2	93.03	96.18	77.92	84.16	76.36	89.39	87.90	76.36
All	0.70	0.74	0.69	0.70	0.70	0.73	0.68	0.69

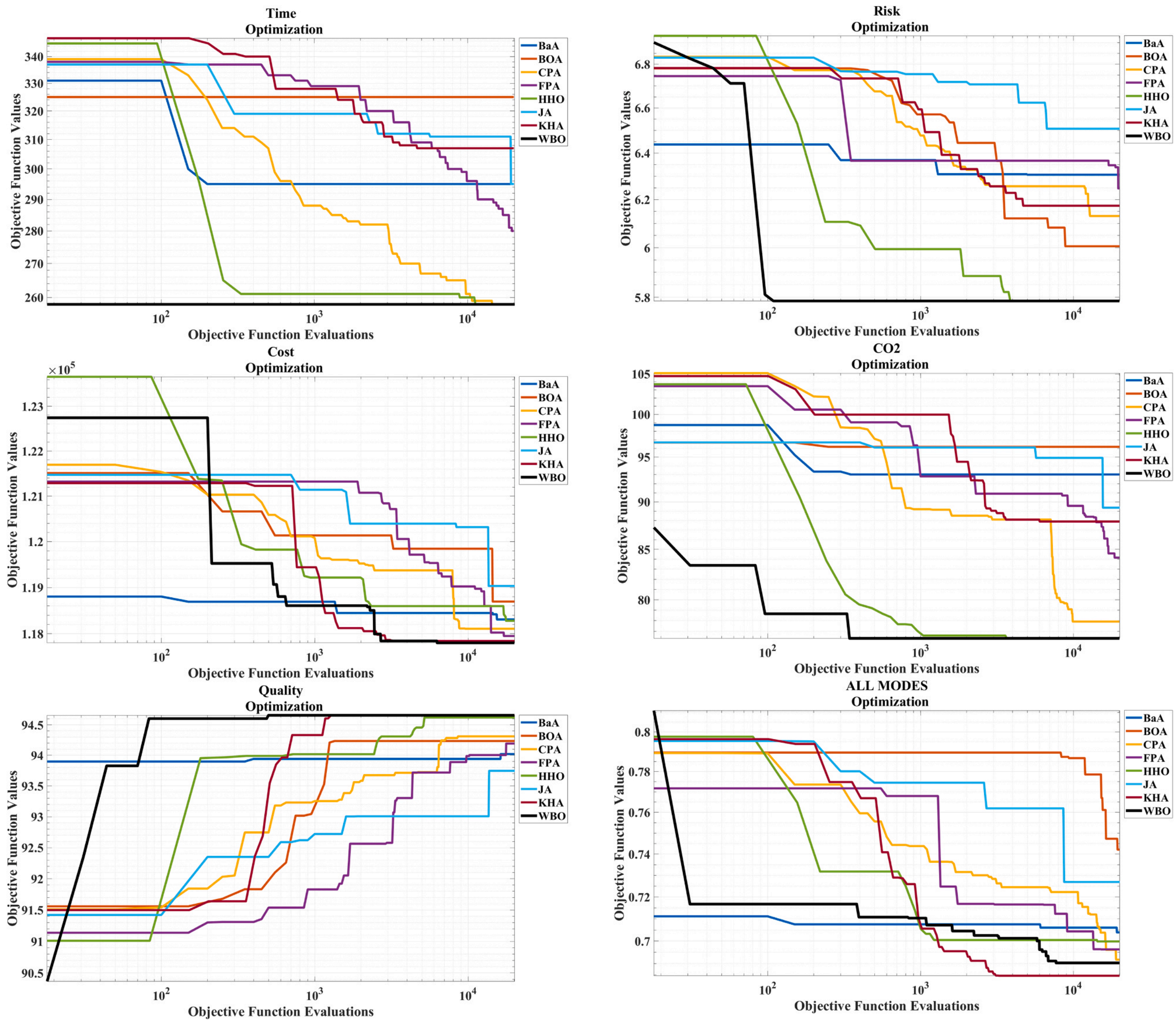

Fig. 8. Convergence history of WBO and other mentioned algorithms.

$$I^j_{C_i} = C_{ic} \times T \tag{36}$$

$$DC = \begin{cases} C_1(T_0 - T) & if \quad T \le T_0 \\ \left(e^{\frac{T-T_0}{T_0}} - 1\right)\left(D^j_{C_i} + I^j_{C_i}\right) & if \quad T > T_0 \end{cases} \tag{37}$$

where TC_p shows the overall project's cost; $D^j_{C_i}$ and $I^j_{C_i}$ represent the direct and indirect costs in conjunction with the j-th mode of the *ith* activity, respectively. DC shows the delay cost; T_0 is the contractual planned duration of the project; C_1 is the prize for early completion of the project, and T is the project's total duration [139,140].

The quality of the whole project is the total quality of all the individual activities since project resources may comprise various materials, equipment, and labor. The quality will increase as the activities are extended, but going beyond a certain point will result in a decline in quality. Consequently, the quality performance index (QPIi), which is determined as follows, serves to indicate the quality of each activity [140]:

$$QPI_i = a_i t_i^2 + b_i t_i + c_i \tag{38}$$

where t_i shows the length of *i*th activity; a_i, b_i, and c_i represent coefficients determined using the quadratic function based on BD (Fig. 6). SD, BD, and LD elucidate the shortest, best, and longest duration. Nonetheless, BD is determined by Eq. 39. Hence, Eq. 8 formulates the objective function for quality as follows:

$$BD = SD + 0.613(LD - SD) \tag{39}$$

$$\max Q = \sum_{i=1}^{M} \frac{QPI_i}{M} \tag{40}$$

Nevertheless, using certain resources might devastate the environment by generating CO_2 during the project's construction phase. There are two ways that CO_2 gas may be released during the on-site construction procedure: directly through energy and fuel burning and indirectly from the manufacture and transportation of building materials. Therefore, Eq. 41 can be used to identify the objective function of reducing the project's whole CO_2 emissions.

$$\min CE = \sum_{i=1}^{M} E_{dij} + \sum_{i=1}^{M} E_{inij} = \left(\sum_{i=1}^{M} Q_{ed} \times F_e + Q_{dd} \times F_d\right) + \left(\sum_{i=1}^{M} Q_k \times F_j + Q_{ek} \times F_e + Q_{dk} \times F_d\right) \tag{41}$$

Table 4 -
The statistical outcomes of the WBO and other algorithms for the main project (Case Study).

Time	BA	BOA	CPA	FPA	HHO	JA	KHA	WBO
Best	295.00	325.00	258.00	280.00	258.00	295.00	307.00	258.00
Mean	317.77	345.87	269.57	291.13	258.30	307.33	321.57	260.83
Worst	341.00	359.00	289.00	300.00	261.00	315.00	331.00	261.00
Std	12.38	8.45	6.88	5.35	0.92	4.82	5.47	0.65
Computational time (s)	1.51	1.37	1.59	1.54	1.40	1.36	2.24	1.81
Cost	BA	BOA	CPA	FPA	HHO	JA	KHA	WBO
Best	118,308.00	118,696.00	118,107.80	117,950.80	118,279.00	119,033.04	117,844.00	117,804.00
Mean	119,180.91	120,041.90	118,986.03	118,432.99	118,912.46	119,937.56	118,121.75	118,236.40
Worst	120,845.84	122,195.30	119,595.44	118,814.40	120,388.04	120,397.60	118,617.00	118,538.00
Std	584.03	899.39	371.34	193.70	557.81	354.56	177.56	190.40
Computational time (s)	2.36	1.36	1.57	1.54	1.40	1.34	2.20	1.98
Quality	BA	BOA	CPA	FPA	HHO	JA	KHA	WBO
Best	94.02	94.23	94.31	94.19	94.62	93.75	94.66	94.66
Mean	93.27	94.12	93.89	93.94	94.10	93.21	94.66	94.64
Worst	94.02	94.23	94.31	94.19	94.62	93.75	94.66	94.66
Std	0.39	0.05	0.20	0.13	0.17	0.20	0.00	0.07
Computational time (s)	1.52	1.42	1.57	1.53	1.41	1.41	2.19	1.97
Risk	BA	BOA	CPA	FPA	HHO	JA	KHA	WBO
Best	6.31	6.01	6.13	6.25	5.79	6.51	6.17	5.79
Mean	6.55	6.14	6.33	6.36	5.89	6.60	6.31	5.79
Worst	6.78	6.27	6.45	6.46	6.29	6.68	6.43	5.79
Std	0.11	0.06	0.08	0.05	0.14	0.04	0.06	0.00
Computational time (s)	1.82	1.37	1.57	1.55	1.40	1.51	2.20	1.97
CO_2	BA	BOA	CPA	FPA	HHO	JA	KHA	WBO
Best	93.03	96.18	77.92	84.16	76.36	89.39	87.90	76.36
Mean	98.83	103.20	82.24	88.36	76.64	92.34	92.98	76.36
Worst	106.11	107.76	87.39	91.99	79.84	94.45	99.14	76.36
Std	3.07	3.15	2.60	2.21	0.88	1.20	2.86	0.00
Computational time (s)	1.74	1.36	1.56	1.54	1.40	1.50	2.19	1.99
All	BA	BOA	CPA	FPA	HHO	JA	KHA	WBO
Best	0.70	0.74	0.69	0.70	0.70	0.73	0.68	0.69
Mean	0.72	0.77	0.72	0.70	0.71	0.75	0.69	0.70
Worst	0.75	0.83	0.74	0.71	0.72	0.76	0.71	0.71
Std	0.01	0.02	0.01	0.00	0.01	0.01	0.00	0.00
Computational time (s)	1.46	1.37	1.58	1.54	1.39	1.39	2.20	1.97

Where CE shows the total CO_2 emission in the project; E_{dij} and E_{inij} are project's direct and indirect CO_2 emissions, respectively; Q_{ed} is activity' electricity consumption; Q_{dd} elucidates the activity's diesel consumption; Q_{ij} shows consumption of material k in activity; Q_{ek} is the amount of electricity used to transport the material k needed for the operation; Q_{dk} A displays the amount of diesel used to move the material k for the activity; F_j, F_e, and F_d represent per-unit production of material k, carbon emission factor per electricity unit, and diesel unit consumption.

The project's conditions, delivery methods, and contract terms significantly determine the real project risk [141–143]. Consequently, Eq. 42 can be used to represent the objective function for risk:

$$\min R = w_1 \times \left(1 - \frac{TF_c + 1}{TF_{\max} + 1}\right) + w_2 \times \left(\frac{\sum_{i=1}^{Pd} (R_t - \overline{R})^2}{P_d (\overline{R})^2}\right) + w_3 \times \left(1 - \frac{\overline{R}}{\max(R_t)}\right) \tag{42}$$

where the project's overall current float and flexible scheduling float are shown by TF_c and $TF_{\max}$, respectively; $\overline{R}$ clarifies the level of uniform resources; R_t is the resource required on day t; and wi demonstrates the weights.

Eq. 43 is utilized to evaluate the capabilities of the WBO to concurrently optimize the time-cost-quality-risk-CO2 (All) tradeoff using the normalizing procedure:

$$F(x) = \frac{T - T_{\min}}{T_{\max} - T_{\min}} + \frac{C - C_{\min}}{C_{\max} - C_{\min}} + \frac{R - R_{\min}}{R_{\max} - R_{\min}} + \frac{CO_2 - CO_{2(\min)}}{CO_{2(\max)} - CO_{2(\min)}} + \frac{Q_{\min} - Q}{Q_{\max} - Q_{\min}} \tag{43}$$

The building information modeling-based resource tradeoff by means of metaheuristic algorithms is an unconstraint optimization problem so there is no need for constraint handling approaches.

2.5. BIM module and design examples

In the current research, three numerical case studies indicate the capability of the proposed metaheuristic algorithm in dealing with resource tradeoff problems in the construction industry.

2.5.1. Main BIM-based project

The case study used to test the model is a five-story residential dwelling with a basement with a total floor area of 930 m^2. The case study also verifies the suggested algorithm concerning the following five factors: cost, time, risk, quality, and CO_2 emissions. Table 1 shows how the BIM procedure, project information, and expert judgments throughout the planning and designing stages elicit information about all 38 activities. All actions follow a Finished to Start (FS) pattern, meaning they end before they begin. Architectural, Structural, and MEP (Mechanical, Electrical, and Piping) modeling were carried out using Autodesk Revit 2022; all components were modeled at LOD 350 according to BIMFourm 2019 standard. After that, a parametric model was made in Revit with the help of Dynamo visual programming. The next research step was using the Navisworks software for soft and hard clash detection. MATLAB is then utilized for programming. Fig. 7 elucidates

Table 5 -
The p-values of different statistical tests.

Main Algorithm	Statistical Test	Alternative Metaheuristic Algorithms						
		BA	BOA	CPA	FPA	HHO	JA	KHA
WBO	**KS Test**	0.999957	0.999957	0.999957	0.999957	0.999957	0.999957	0.999957
	MW Test	0.699134	0.699134	0.731602	0.699134	0.859307	0.699134	0.859307
	W Test	0.031250	0.031250	0.062500	0.031250	0.250000	0.031250	0.125000

the modeling procedure based on BIM.

As seen in Table 1, the BIM process, the project data, and the experts' judgments are used in the planning and designing processes to extract all of the activity information. In other words, the information in this table was compiled using the experiences of a wide variety of highly accomplished individuals and specialists in the relevant sector. The amount of time and cost required for mode 1 is the first suggestion made by the contractor, and a majority of contractors proposed the lowest amount at the initial stage to win the auction; however, given that the majority of contractors do not take into account rework, conflicts, hard or soft clashes, payment delays employers, and harsh weather conditions. Mode 3 is the results produced from the BIM procedure. Modes 2 and 4 were considered based on the recommendations made by specialists and experts working on this project. Mode 5 is the project's actual time and cost, derived from the construction's final state. Then, a random risk percentage and carbon dioxide emissions for each activity are demonstrated regarding the opinions of top academicians and industry authorities.

2.5.2. Benchmark project (1)

The second case study is based on a real-world building project with 24 activities, including two basements, thirteen upper levels, and a foundation of planned poured piles. Its activities' information is extracted from [137]. It should be highlighted that BIM, project data, and expert opinions in the first (case study in this study) and the second project are used to extract all activity data. The phrase "combined activity" is used in the first case study. For instance, installing a steel bar, setting up formwork, pouring concrete, and taking down the formwork are all considered to be part of completing the column assignment for the second level.

2.5.3. Benchmark project (2)

The third example is a moderate-sized construction project, with a total floor space of 12,870 m^2 spread between a basement and six stories. It's an office complex in Vietnam. [144] contains the construction project data sets. The case studies presented here are primarily intended to show how the novel metaheuristic algorithm can be applied to actual cases.

3. Results and discussion

Some rigorous metaheuristic algorithms were used to examine the effectiveness of the WBO algorithm in resolving resource tradeoff problems in building projects, including the Bat Algorithm (BA) [145], Butterfly optimization algorithm (BOA) [146], Cyclical Parthenogenesis Algorithm (CPA) [147], Flower Pollination Algorithm (FPA) [148], Harris Hawks Optimization (HHO) [149], Jaya Algorithm (JA) [150], and Krill Herd Algorithm (KHA) [151]. All algorithms' parameters are shown in Table 2.

3.1. Main BIM-based project results

Table 3 lists potential algorithms for each case and the best results of the WBO. However, 30 separate optimization runs are performed for the main statistical goals to calculate the mean, worst, standard deviation, and computation time. Whereas the stopping condition is considered based on a predefined 20,000 no. of objective function evaluations. Fig. 8 displays the convergence history of WBO and other algorithms in solving the tradeoff as mentioned above problems.

In our analysis, the superior quality scores of the WBO and KHA approaches are primarily attributed to their inherent algorithmic structures, tailored for precise solutions in our study's context. The WBO method achieves this through a weighted balance between exploration and exploitation phases, while KHA utilizes an adaptive search mechanism for enhanced performance. When examining cost and time efficiency, we note marginal variations among the methods, suggesting a similar level of resource utilization efficiency across all, with slight differences likely stemming from the stochastic nature of these algorithms. Additionally, the risk and CO_2 emissions metrics, essential for assessing environmental and operational feasibility, indicate a more favorable environmental impact for HHO and KHA, likely due to more efficient computational processes that reduce energy consumption. Lastly, the overall score—a composite metric—reflects balanced performance across all criteria, with the closeness in overall scores of KHA and HHO to our proposed method highlighting their comparable efficacy in a holistic sense.

Table 4 displays the statistical data obtained from the optimization performed on the case study. The WBO algorithm has the potential to outperform the vast majority of other metaheuristics in the first scenario of time optimization of the case study. This scenario estimates that 258

Table 7 -
The KW test results (mean of the ranks).

Rankings	Algorithms	Mean of Ranks
1	WBO	21.58
2	HHO	23.00
3	CPA	23.33
4	KHA	23.58
5	FPA	24.50
6	BA	26.42
7	BOA	26.50
8	JA	27.08
Chi-sq.	0.8362	
Prob>Chi-sq.	0.9971	

Table 6 -
The maximum difference of metaheuristics.

Main Algorithm	Statistical Test	Alternative Metaheuristic Algorithms						
		BA	BOA	CPA	FPA	HHO	JA	KHA
WBO	**MW Test**	42.00	42.00	41.50	42.00	40.50	42.00	40.50
		36.00	36.00	36.50	36.00	37.50	36.00	37.50
	W Test	21.00	21.00	15.00	21.00	6.00	21.00	14.00
		1.00	1.00	1.00	1.00	1.00	1.00	1.00

Table 8 –
The best findings of the WBO and other methods in the second project.

	BA	BOA	CPA	FPA	HHO	JA	KHA	WBO
Time	101.00	110.00	89.00	91.00	87.00	100.00	102.00	87.00
Cost	11,336.00	11,732.00	10,132.00	10,212.00	10,087.00	10,613.00	10,953.00	10,397.00
Quality	96.30	96.12	96.33	96.50	96.81	95.57	96.81	96.81
CO_2	2400.61	2602.14	1999.19	2207.21	1978.92	2212.27	2556.12	1982.73

days is the best and most optimal time, so the WBO algorithm is likely to be outperformed by the rest of the algorithms. The WBO algorithm produces the result with the slightest standard deviation (Std), followed by the HHO method, which accounts for 0.92. The WBO algorithm is the one with the minimum outcome. In contrast, the BA algorithm generates the highest value of Std, around 12.40. In addition, the JA algorithm was able to complete the time optimization process in the shortest amount of time possible (1.36 s). On the other hand, the KHA algorithm procured the longest computing time, necessitating a significantly longer amount of time to complete the optimization procedure in the mentioned scenario.

In the second scenario of the case study, which focuses on cost optimization, the WBO algorithm performs better than other alternative metaheuristic algorithms. In other words, unlike the JA method, the WBO algorithm can identify the project's least cost, which determines the maximum optimal cost value. Nevertheless, the BA method required the most time spent on calculation in this scenario, followed by the KHA. On the other hand, the BOA approach required the least amount of time to be paid on computation for the project's cost optimization described above. In addition, the KHA algorithm provided the minimum possible value for the standard deviation, which the WBO followed. In the meanwhile, the BOA came out on top with the highest standard deviation of all the algorithms that were investigated for this scenario. As a consequence of this, the WBO algorithm could be an appropriate metaheuristic for the optimization of costs associated with project and construction management.

The statistical results of the quality optimization performed on the case study demonstrate that the WBO and KHA approaches can deliver superior quality. The HHO came in second place behind the CPA algorithm regarding its remarkable quality value, around 94.3. In addition, the KHA has the potential to deliver the least standard deviation, which in this instance is precisely 0.00. In stark contrast, the BA has the highest Std. Compared to the KHA, which needed around 2.19 s of computing time, the HHO and JA methods required much less time to achieve the same level of quality optimization as the latter.

Consequently, the WBO algorithm can provide excellent quality, the more desirable option for project managers to go in this particular scenario. However, WBO determined the lowest risk value within the case study scope, registered at almost 5.79. In addition, the BOA method demanded the least amount of computational time feasible in this case, followed by the HHO algorithm. As a result, the WBO could be a good candidate for risk optimization in project scheduling. During this time, the WBO can determine the least possible Std in the given scenario.

Regarding sustainable construction, the WBO could be feasible to reduce projects' carbon footprint. This is so that environmentally friendly construction could be done, as the WBO was able to determine

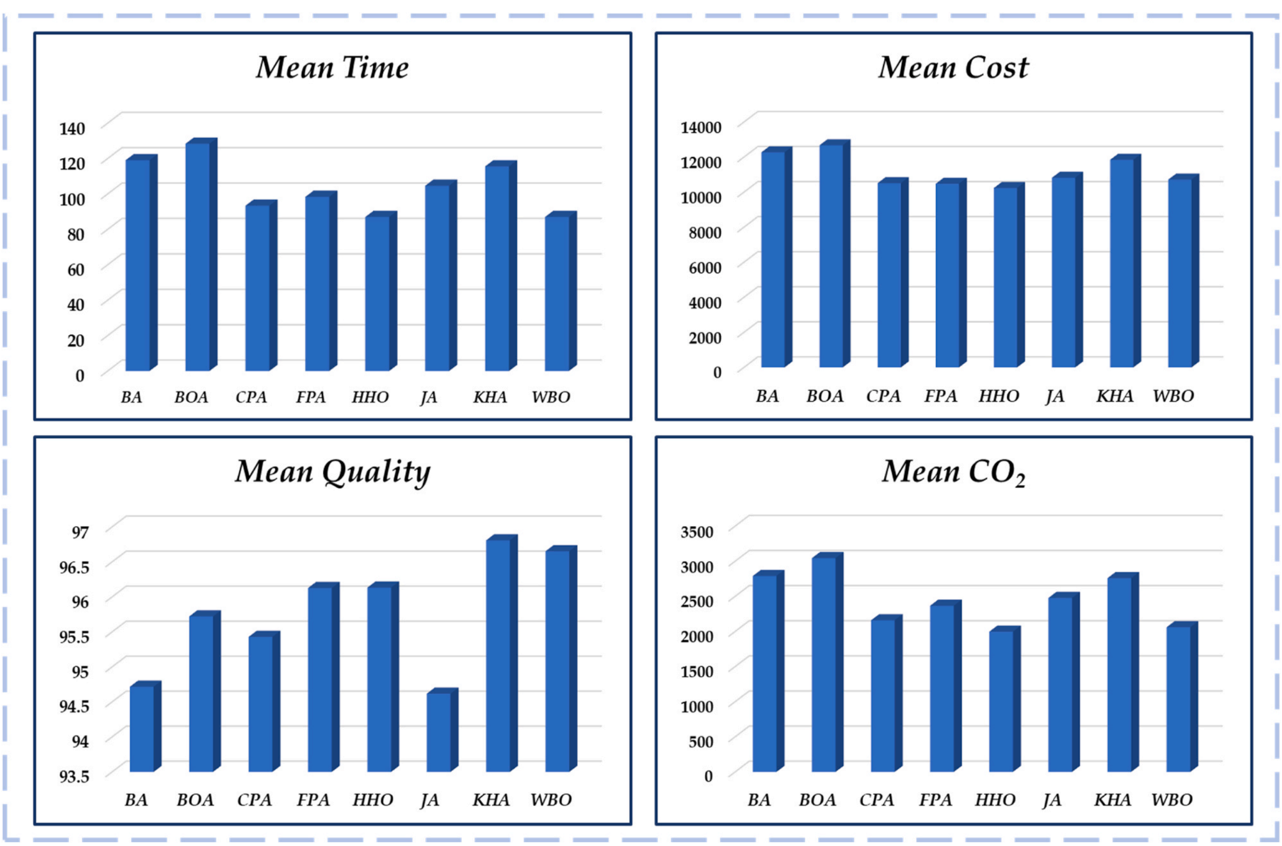

Fig. 9. The mean values of 30 independent optimization run of the WBO and other methods in the second project.

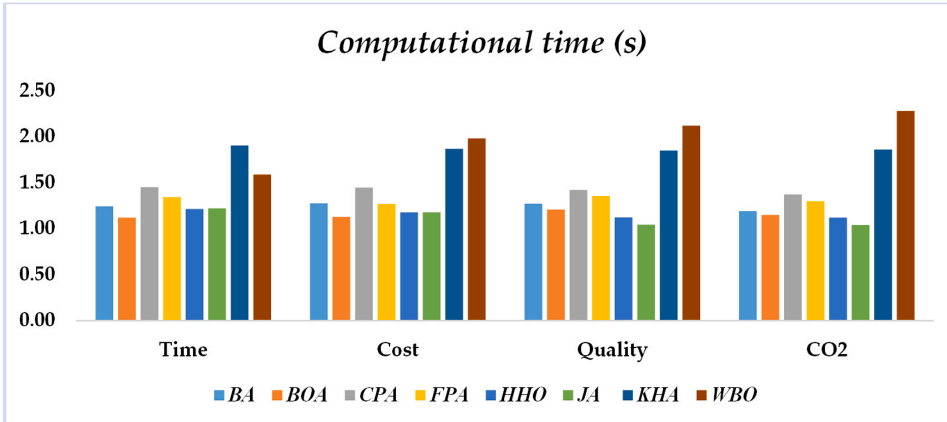

Fig. 10. The computational time of the proposed WBO and alternative methods in the second project.

the least amount of CO_2 in the case study. On the other hand, the BOA gave the highest result for CO_2, which indicates that its performance was unfavorable in attaining the goal of completing the project with a smaller and smaller carbon footprint. On the other hand, the BOA technique resulted in the least amount of time spent computing, which was reported at 1.36 (s), followed by HHO. Consequently, considering the typical amount of time spent computing, the WBO algorithm could be regarded as a suitable option for optimizing the quantity of carbon dioxide in building projects.

Using a dwelling home as a case study, the WBO and KHA algorithms can perform better than other metaheuristic algorithms in tackling the TCQRCT problem. The WBO and KHA algorithms provided the lowest value for the Std value, showing their better performance. The BOA used the least computational time to execute TCQRCT in the case study,

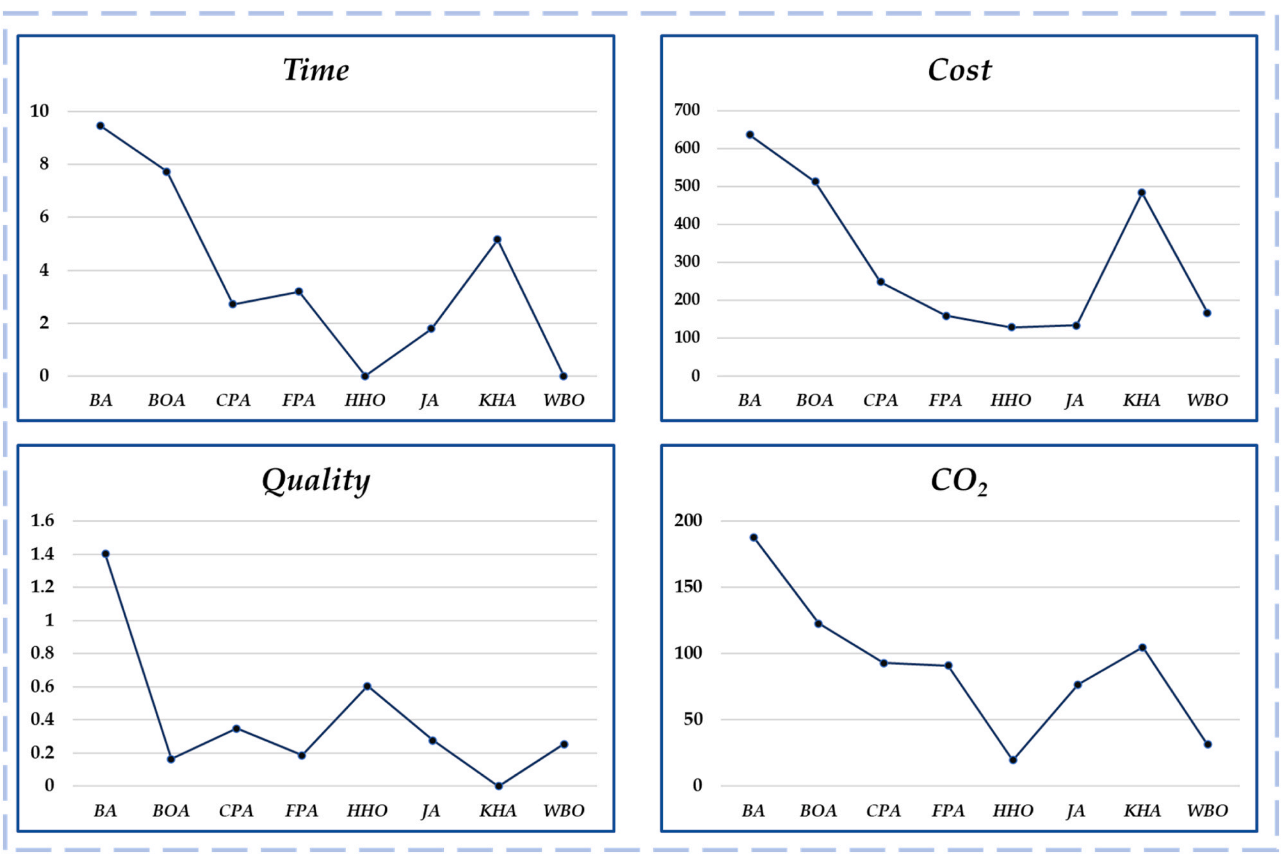

Fig. 11. The standard deviation values of the proposed WBO and alternative methods in the second project.

Table 9 –
The best findings of the WBO and other methods in the third project.

	BA	BOA	CPA	FPA	HHO	JA	KHA	WBO
Time	1882.00	1911.00	1848.00	1870.00	1835.00	1888.00	1892.00	1835.00
Cost	165,138.00	165,590.00	163,016.00	163,835.00	163,179.00	164,746.00	165,417.00	163,181.00

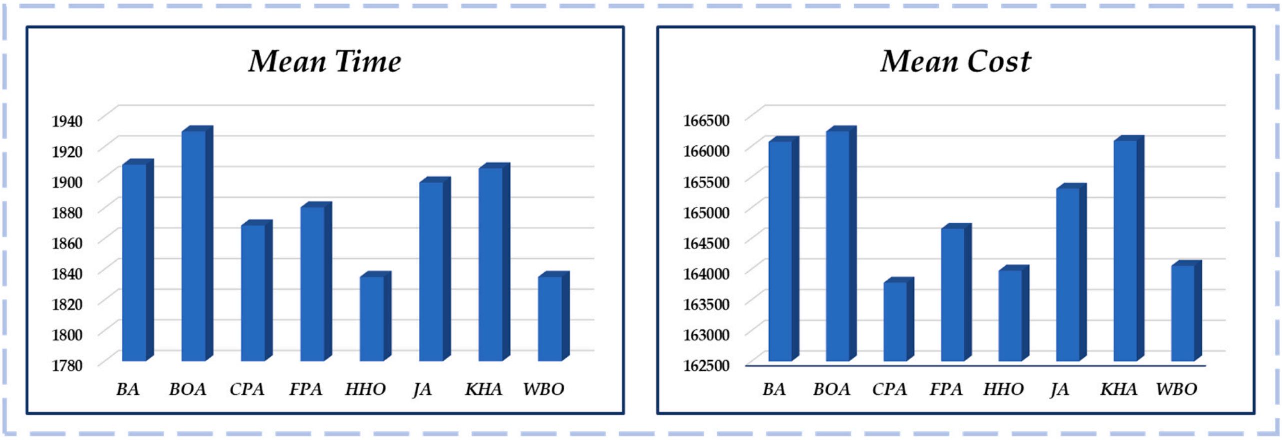

Fig. 12. The mean values of 30 independent optimization run of the WBO and other methods in the third project.

followed by the HHO and JA with about 1.39 (s). The WBO method could be appropriate for solving TCQRCT problems in building projects without considering processing time. The performance of different algorithms in the case study you described can be attributed to their underlying architecture and design, which are tailored to optimize specific aspects of project management. For instance, the HHO method, with its low standard deviation in time optimization, demonstrates its strength in maintaining consistency and accuracy. HHO's design, inspired by the cooperative behavior and hunting strategy of Harris' hawks, enables it to effectively balance exploration and exploitation phases. This balance is crucial in complex optimization problems, allowing HHO to adeptly navigate the optimization landscape and avoid being trapped in local optima.

In contrast, the BA and KHA algorithms show differing levels of performance in various scenarios. The BA algorithm's higher standard deviation in time optimization might be a consequence of its inherent design, which possibly prioritizes exploration over exploitation. This could lead to a wider range of results, impacting its consistency but potentially allowing for the discovery of novel solutions in more diverse problem spaces. On the other hand, the KHA algorithm's minimal standard deviation in cost optimization suggests a design that closely hones in on optimal solutions with high precision. This precision, however, might come at the cost of increased computational time, as seen in the case study. The JA and BOA algorithms also exhibit specific strengths and weaknesses. The JA algorithm's rapid completion of the time optimization process indicates a design optimized for computational efficiency. This efficiency, while advantageous in scenarios demanding quick decision-making, might sacrifice depth of search and thoroughness, possibly leading to less accurate outcomes in certain cases. Meanwhile, the BOA's performance in sustainable construction, particularly in optimizing carbon footprint, suggests a potential trade-off in its design. Its ability to compute quickly, as evidenced by its minimal computational time, contrasts with its less favorable results in reducing CO_2 emissions, highlighting a possible prioritization of speed over environmental optimization.

In summary, the varied performances of these algorithms reflect their design principles, where specific optimization goals - such as accuracy, consistency, computational speed, and environmental impact - are prioritized differently. This underscores the importance of choosing the right algorithm for a given project management task, considering the specific requirements and constraints of the scenario, and the particular strengths and design focus of the algorithm in question.

3.2. Comprehensive statistical analysis of the main project

Four famous statistical tests—Wilcoxon (W), Mann-Whitney (MW), Kolmogorov-Smirnov (KS), and Kruskal-Wallis (KW)—were used to provide a more accurate assessment of how successfully WBO handled the resource tradeoff based on BIM. Henceforth referred to as "KS," the Kolmogorov-Smirnov statistic is a member of the elite group of statistics known as "Empirical Distribution Function" (EDF) statistics, which are founded on the most considerable possible absolute value of the vertical difference between the theoretical and observed distributions [152–154]. The W test compares the rankings' averages by considering

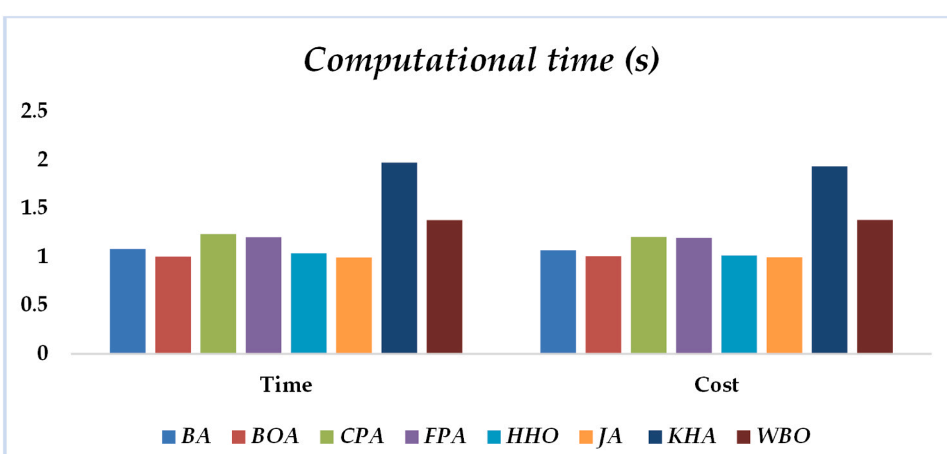

Fig. 13. The computational time of the proposed WBO and alternative methods in the third project.

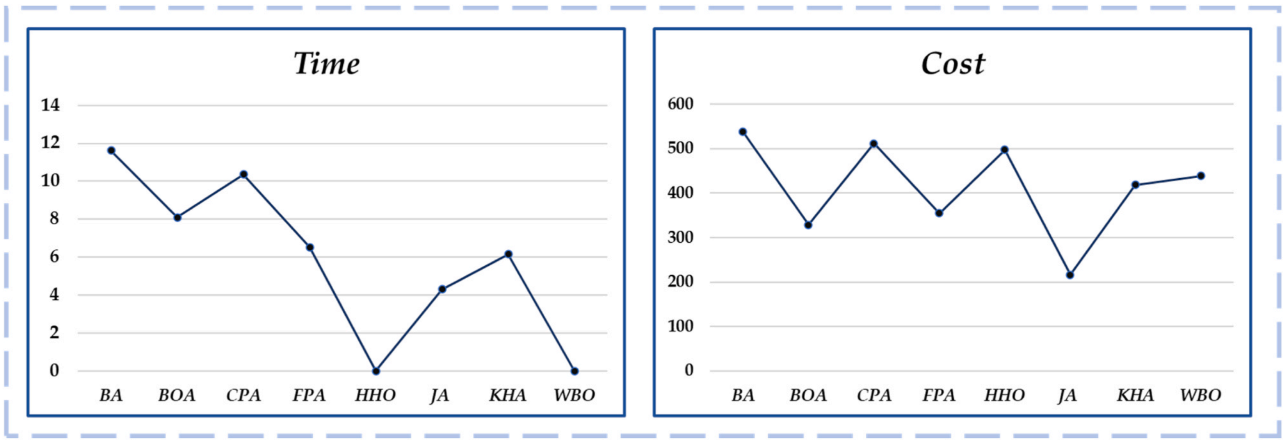

Fig. 14. The standard deviation values of the proposed WBO and alternative methods in the third project.

pairs of ranks. This test's null hypothesis is that there is no difference in the mean rankings of two randomly chosen variables from two datasets. With this metric, the test data set with the lower mean of ranks indicates superior statistical behavior [155]. The null hypothesis in the MW test suggests the difference between two randomly chosen variables from distinct datasets by considering the sum of the variables' rankings. This means that the statistical dataset with a lower rank summation has superior statistical behavior [156]. Furthermore, The KW test is a commonly used statistical test that considers the overall rankings of several variables across various datasets. The KW test compares the mean of rankings across many datasets since the previously performed MW and W tests are applied two-by-two based on the summation and mean of ranks [157,158].

The p-values for the KS, MW, and W statistical tests are included in Table 5 for comparison. To put the capabilities of WBO into perspective, Table 6 shows the highest difference between several techniques, i.e., MW, and W statistical tests. The WBO can manage these problems since the mean (W test), and summation (MW test) rankings in WBO are often lower than in other ways. Table 7 presents the overall rankings of the algorithms for dealing with BIM-based resource tradeoffs, including the mean of ranks. It is also important to note that WBO now has the top rating, which is acceptable.

3.3. Results Overview for First Benchmark Project

Table 8 lists potential algorithms for each case and the best results of the WBO for the second project. However, 30 separate optimization runs are performed for the best and mean results, standard deviation, and computation time. Whereas the maximum number of objective function evaluations is considered as 20,000 for stopping condition. Fig. 9 shows the mean values of 30 separate optimization runs using different algorithms and the proposed WBO. In this project, Time, Cost, Quality, and CO_2 emissions (Kg) are considered. As can be seen, the proposed WBO method could deliver the minimum time in this project, registered at 87 days, followed by CPA. Regarding cost optimization, although the HHO could compute the least cost in this project, the WBO is able to give an acceptable value. In stark contrast, BOA delivered the highest cost in this project, indicating its improper performance.

Furthermore, the highest and best quality has been given by the proposed WBO approach in this project, which FPA follows. However, the HHO method could calculate the least CO_2 emission, indicating its better capability in providing sustainable construction, followed by the novel WBO method. Fig. 10 shows the computational time (seconds) of the different metaheuristic algorithms and the proposed WBO in the second project. As inferred from the information, the proposed WBO method gave the longest time in three scenarios: cost, quality, and CO_2 emissions. In time and cost optimization, BOA took the least time; and in quality and CO_2 emission optimization, JA took the least time.

Fig. 11 shows the proposed WBO and alternative algorithms' standard deviation (Std) values in the second case study. As can be seen, the Std value for the WBO and HHO is zero in the time optimization, which means how close the results obtained from the 30 independent runs are to their mean value. In stark contrast, due to the higher value of Std rather than other algorithms, the BA optimization algorithm could not provide a consistent result in the analysis. Furthermore, the HHO method delivered the most negligible Std value in cost optimization, followed by FPA and WBO algorithms. However, the KHA method gave the least Std value for the quality optimization in this project. Finally, HHO and WBO algorithms could provide the least Std values in CO_2 optimization.

3.4. Results overview for second benchmark project

Table 9 shows potential algorithms for each case and the best results of the WBO for the second project. However, 30 separate optimization runs are performed for the best and mean results, standard deviation, and computation time. Whereas the maximum number of objective function evaluations is considered as 20,000 for stopping condition. In this project, Time and Cost are considered. As can be seen, the proposed WBO method could deliver the minimum time in this project, registered at 1835 days, followed by CPA.

Regarding cost optimization, although the CPA could compute the least cost in this project, the WBO is able to give an acceptable value. In stark contrast, BOA delivered the highest cost in this project, indicating its improper performance. Fig. 12 shows the mean values of 30 separate optimization runs using different algorithms and the proposed WBO. Fig. 13 shows the computational time (seconds) of the different metaheuristic algorithms and the proposed WBO in the second project. As inferred from the information, the proposed WBO method gave the longest time in three scenarios: cost, quality, and CO_2 emissions. In time and cost optimization, BOA took the least time; and in quality and CO_2 emission optimization, JA took the least time.

Fig. 14 presents the standard deviation (Std) values of the proposed WBO alongside other alternative algorithms in the second case study. Notably, the WBO and HHO both register an Std value of zero in time optimization, indicating that the outcomes from the 30 separate runs closely align with their average value. On the other hand, the BA optimization algorithm, with its higher Std value compared to other algorithms, demonstrated inconsistency in its results. Moreover, when it comes to cost optimization, the JA method showcased the lowest Std value.

4. Conclusions and future directions

In this paper, the Wolf-Bird Optimizer (WBO) was presented as a novel metaheuristic algorithm inspired by the coexistence behavior and predatory partnership of wolves and ravens in nature. WBO finds the optimal solution from a population of candidate solutions. To this end, it is assumed that a group of ravens in nature are attempting to find some prey in their neighborhood. Subsequently, ravens send intelligent signals to wolves to inform them of the location of the prey. Finally, the group of wolves conforms to the signals to reach the prey. The primary findings of this study can be summarized as follows:

- The WBO has the capability to reach the optimal global solution using the least number of objective function evaluations, highlighting the superior computational efficiency of this new algorithm.
- Based on the W statistical test findings, WBO can produce superior outputs with lower means of ranks in most cases in comparison with other algorithms.
- The KW statistical test results, including the mean of ranks, demonstrate that WBO outperforms the other algorithms in all data sets studied.
- Regarding the outcomes of the best optimization runs completed by various approaches for time optimization in the main BIM-based project, the WBO algorithm conducted the case study in the least amount of time, 258 days.
- The WBO can give 117,804.00($) for the case study cost, the most fantastic option among the others.
- The WBO can also offer better quality value in the main BIM-based case study.
- The WBO can give the project the best risk and CO_2 optimization outcomes than alternative metaheuristics.
- Compared to other algorithms, the WBO method can deliver 0.69 concerning the solutions to the TCQRCTP, which is quite a competitive value.

Additionally, the introduced WBO algorithm has demonstrated commendable and competitive outcomes in the two benchmark construction projects. The analyses conducted suggest that the WBO algorithm's edge over other metaheuristic algorithms can be attributed to three main factors: (1) Rapid convergence behavior; (2) Minimal requirement for objective function evaluations. Moving forward, it would be beneficial to assess the proposed algorithm on intricate optimization challenges in various domains, such as large-scale design problems in structural engineering, truss structures' shape and size optimization, and the design of tall steel buildings. It would also be pertinent to undertake both experimental and numerical validations of the real-world constrained optimization problems presented in CEC 2020 for addressing future challenges.

Declaration of Competing Interest

The authors declare no conflict of interest regarding the publication of this paper.

References

[1] J.-S. Chou, D.-N. Truong, A novel metaheuristic optimizer inspired by behavior of jellyfish in ocean, Appl. Math. Comput. 389 (2021), 125535.
[2] N. Khodadadi, et al., Optimizing truss structures using composite materials under natural frequency constraints with a new hybrid algorithm based on cuckoo search and stochastic paint optimizer (CSSPO), Buildings 13 (6) (2023) 1551.
[3] N. Khodadadi, S. Talatahari, A.H. Gandomi, ANNA: advanced neural network algorithm for optimisation of structures, Proc. Inst. Civ. Eng. -Struct. Build. (2023) 1–23.
[4] E.-S.M. El-Kenawy, et al., Metaheuristic optimization for improving weed detection in wheat images captured by drones, Mathematics 10 (23) (2022) 4421.
[5] F. Glover, Tabu search—part I, ORSA J. Comput. 1 (3) (1989) 190–206.
[6] M. Creutz, Microcanonical monte carlo simulation, Phys. Rev. Lett. 50 (19) (1983) 1411.
[7] I. Boussaïd, J. Lepagnot, P. Siarry, A survey on optimization metaheuristics, Inf. Sci. 237 (2013) 82–117.
[8] N. Khodadadi, et al., A comparison performance analysis of eight meta-heuristic algorithms for optimal design of truss structures with static constraints, Decis. Anal. J. (2023), 100266.
[9] H.A. Alsayadi, N. Khodadadi, S. Kumar, Improving the regression of communities and crime using ensemble of machine learning models, J. Artif. Intell. Metaheuristics 1 (1) (2022), p. 27-7-34.
[10] B. Galvan, et al., Parallel evolutionary computation for solving complex CFD optimization problems: a review and some nozzle applications, in: K. Matsuno, et al. (Eds.), Parallel Computational Fluid Dynamics 2002, Amsterdam, North-Holland, 2003, pp. 573–604.
[11] J.H. Holland, in: O.G. Selfridge, E.L. Rissland, M.A. Arbib (Eds.), Genetic Algorithms and Adaptation, in Adaptive Control of Ill-Defined Systems, Springer US, Boston, MA, 1984, pp. 317–333.
[12] S. Mirjalili, Genetic algorithm. Evolutionary Algorithms and Neural Networks: Theory and Applications, Springer International Publishing,, Cham, 2019, pp. 43–55.
[13] R. Storn, K. Price, Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces, J. Glob. Optim. 11 (4) (1997) 341–359.
[14] U.K. Chakraborty, Advances in Differential Evolution, Vol. 143, Springer, 2008.
[15] D. Whitley, An overview of evolutionary algorithms: practical issues and common pitfalls, Inf. Softw. Technol. 43 (2001) 817–831.
[16] J. Kennedy, R. Eberhart, Particle Swarm Optimization. Proceedings of ICNN'95-international conference on neural networks, IEEE,, 1995.
[17] M. Dorigo, V. Maniezzo, A. Colorni, Ant system: optimization by a colony of cooperating agents, IEEE Trans. Syst., Man, Cybern., Part B (Cybern.) 26 (1) (1996) 29–41.
[18] L. Xie, et al., Tuna swarm optimization: a novel swarm-based metaheuristic algorithm for global optimization, Comput. Intell. Neurosci. 2021 (2021) 9210050.
[19] G. Beni, J. Wang, Swarm intelligence in cellular robotic systems. in Robots and Biological Systems: Towards A New Bionics?, Springer,, 1993, pp. 703–712.
[20] E. Bonabeau, et al., Swarm Intelligence: From Natural to Artificial Systems, Oxford university press, 1999.
[21] T. Dutta, et al., Border collie optimization, IEEE Access PP (2020), p. 1-1.
[22] D. Karaboga, B. Basturk, Artificial Bee Colony (ABC) Optimization Algorithm for Solving Constrained Optimization Problems. Foundations of Fuzzy Logic and Soft Computing, Springer Berlin Heidelberg, Berlin, Heidelberg, 2007.
[23] D. Karaboga, An idea based on honey bee swarm for numerical optimization. Technical report-tr06, Erciyes university,, 2005.
[24] G.-G. Wang, S. Deb, Z. Cui, Monarch butterfly optimization, Neural Comput. Appl. 31 (7) (2019) 1995–2014.
[25] M. Azizi, S. Talatahari, A.H. Gandomi, Fire Hawk optimizer: a novel metaheuristic algorithm, Artif. Intell. Rev. 77 (1) (2022).
[26] M.B. Shishehgarkhaneh, et al., BIM-based resource tradeoff in project scheduling using fire hawk optimizer (FHO), Buildings 12 (9) (2022) 1472.
[27] S. Nasuto, J. Bishop, Stab. Swarm Intell. Search via Posit. Feedback Resour. Alloc. (2008) 115–123.
[28] V. Hayyolalam, A.A. Pourhaji Kazem, Black widow optimization algorithm: a novel meta-heuristic approach for solving engineering optimization problems, Eng. Appl. Artif. Intell. 87 (2020), 103249.
[29] X.-S. Yang, Flower Pollination Algorithm for Global Optimization, Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.
[30] K.N. Krishnanand, D. Ghose, Detection of multiple source locations using a glowworm metaphor with applications to collective robotics. Proceedings 2005 IEEE Swarm Intelligence Symposium, 2005. SIS 2005, IEEE,, 2005.
[31] G.-G. Wang, Moth search algorithm: a bio-inspired metaheuristic algorithm for global optimization problems, Memetic Comput. 10 (2) (2018) 151–164.
[32] S. Kirkpatrick, C.D. Gelatt Jr., M.P. Vecchi, Optimization by simulated annealing, Science 220 (4598) (1983) 671–680.
[33] O.K. Erol, I. Eksin, A new optimization method: big bang–big crunch, Adv. Eng. Softw. 37 (2) (2006) 106–111.
[34] S. Dolatabadi, Weighted vertices optimizer (WVO): a novel metaheuristic optimization algorithm, Numer. Algebra, Control Optim. 8 (4) (2018) 461.
[35] N. Khodadadi, et al., Chaotic Stochastic Paint Optimizer (CSPO). Proceedings of 7th International Conference on Harmony Search, Soft Computing and Applications: ICHSA 2022, Springer,, 2022.
[36] M. Azizi, Atomic orbital search: a novel metaheuristic algorithm, Appl. Math. Model. 93 (2021) 657–683.
[37] M. Azizi, A.W. Mohamed, M.B. Shishehgarkhaneh, Optimum design of truss structures with atomic orbital search considering discrete design variables. Handbook of Nature-Inspired Optimization Algorithms: The State of the Art, Springer,, 2022, pp. 189–214.
[38] M. Azizi, et al., Energy valley optimizer: a novel metaheuristic algorithm for global and engineering optimization, Sci. Rep. 13 (1) (2023), 226.
[39] S. Talatahari, M. Azizi, A.H. Gandomi, Material generation algorithm: a novel metaheuristic algorithm for optimization of engineering problems, Processes 9 (5) (2021) 859.
[40] M. Azizi, M.B. Shishehgarkhaneh, M. Basiri, Optimum design of truss structures by Material Generation Algorithm with discrete variables, Decis. Anal. J. (2022), 100043.
[41] M. Saraee, et al., Hybrid social network search and material generation algorithm for shape and size optimization of truss structures. Hybrid Metaheuristics in

Structural Engineering: Including Machine Learning Applications, Springer,, 2023, pp. 49–71.
[42] E. Hosseini, et al., Novel metaheuristic based on multiverse theory for optimization problems in emerging systems, Appl. Intell. 51 (6) (2021) 3275–3292.
[43] F.A. Hashim, et al., Archimedes optimization algorithm: a new metaheuristic algorithm for solving optimization problems, Appl. Intell. 51 (3) (2021) 1531–1551.
[44] A. Kaveh, S. Talatahari, A novel heuristic optimization method: charged system search, Acta Mech. 213 (3) (2010) 267–289.
[45] M. Yazdchi, et al., Metaheuristically optimized nano-MgO additive in freeze-thaw resistant concrete: a charged system search-based approach, Eng. Res. Express 3 (3) (2021), 035001.
[46] S. Talatahari, et al., Optimization of large-scale frame structures using fuzzy adaptive quantum inspired charged system search, Int. J. Steel Struct. 22 (3) (2022) 686–707.
[47] A. Faramarzi, et al., Equilibrium optimizer: a novel optimization algorithm, Knowl. -Based Syst. 191 (2020), 105190.
[48] J.L.J. Pereira, et al., Lichtenberg algorithm: a novel hybrid physics-based meta-heuristic for global optimization, Expert Syst. Appl. 170 (2021), 114522.
[49] A. Kaveh, A. Dadras, A novel meta-heuristic optimization algorithm: thermal exchange optimization, Adv. Eng. Softw. 110 (2017) 69–84.
[50] E.H. Houssein, et al., Lévy flight distribution: a new metaheuristic algorithm for solving engineering optimization problems, Eng. Appl. Artif. Intell. 94 (2020), 103731.
[51] A. Kaveh, M.I. Ghazaan, A new meta-heuristic algorithm: vibrating particles system, Sci. Iran. Trans. A, Civ. Eng. 24 (2) (2017) 551.
[52] S. Talatahari, S. Jalili, M. Azizi, Optimum design of steel building structures using migration-based vibrating particles system. in *Structures*, Elsevier,, 2021.
[53] N. Khodadadi, et al., Multi-objective crystal structure algorithm (MOCryStAl): Introduction and performance evaluation, IEEE Access 9 (2021) 117795–117812.
[54] M. Azizi, S. Talatahari, P. Sareh, Design optimization of fuzzy controllers in building structures using the crystal structure algorithm (CryStAl), Adv. Eng. Inform. 52 (2022), 101616.
[55] B. Talatahari, et al., Crystal structure optimization approach to problem solving in mechanical engineering design, Multidiscip. Model. Mater. Struct. (2022).
[56] S. Talatahari, et al., Crystal structure algorithm (CryStAl): a metaheuristic optimization method, IEEE Access 9 (2021) 71244–71261.
[57] M. Azizi, M. Baghalzadeh Shishehgarkhaneh, M. Basiri, Design optimization of truss structures by crystal structure algorithm, AUT J. Civ. Eng. 6 (2) (2022) 205–220.
[58] S.-A. Ahmadi, Human behavior-based optimization: a novel metaheuristic approach to solve complex optimization problems, Neural Comput. Appl. 28 (1) (2017) 233–244.
[59] M. Azizi, et al., Squid Game Optimizer (SGO): a novel metaheuristic algorithm, Sci. Rep. 13 (1) (2023), 5373.
[60] N. Moosavian, B.K. Roodsari, Soccer league competition algorithm, a new method for solving systems of nonlinear equations, Int. J. Intell. Sci. 4 (01) (2013) 7.
[61] A.A. Abdelhamid, et al., Waterwheel plant algorithm: a novel metaheuristic optimization method, Processes 11 (5) (2023) 1502.
[62] R.V. Rao, V.J. Savsani, D. Vakharia, Teaching–learning-based optimization: a novel method for constrained mechanical design optimization problems, Comput. -Aided Des. 43 (3) (2011) 303–315.
[63] M. Abdel-Basset, et al., Spider wasp optimizer: a novel meta-heuristic optimization algorithm, Artif. Intell. Rev. (2023).
[64] B. Abdollahzadeh, et al., Mountain gazelle optimizer: a new nature-inspired metaheuristic algorithm for global optimization problems, Adv. Eng. Softw. 174 (2022), 103282.
[65] Z.W. Geem, J.H. Kim, G.V. Loganathan, A new heuristic optimization algorithm: harmony search, simulation 76 (2) (2001) 60–68.
[66] A. Kaveh, S. Talatahari, Optimum design of skeletal structures using imperialist competitive algorithm, Comput. Struct. 88 (21–22) (2010) 1220–1229.
[67] A. Kaveh, V.R. Mahdavi, Colliding bodies optimization: a novel meta-heuristic method, Comput. Struct. 139 (2014) 18–27.
[68] A.H. Gandomi, Interior search algorithm (ISA): a novel approach for global optimization, ISA Trans. 53 (4) (2014) 1168–1183.
[69] N. Khodadadi, et al., Multi-objective chaos game optimization, Neural Comput. Appl. 35 (20) (2023) 14973–15004.
[70] N. Khodadadi, et al., Multi-objective artificial hummingbird algorithm. Advances in Swarm Intelligence: Variations and Adaptations for Optimization Problems, Springer,, 2022, pp. 407–419.
[71] N. Khodadadi, et al., AMHS: Archive-based multi-objective harmony search algorithm. Proceedings of 7th International Conference on Harmony Search, Soft Computing and Applications: ICHSA 2022, Springer,, 2022.
[72] A. Kaveh, S. Talatahari, N. Khodadadi, Stochastic paint optimizer: theory and application in civil engineering, Eng. Comput. (2020) 1–32.
[73] N. Khodadadi, V. Snasel, S. Mirjalili, Dynamic arithmetic optimization algorithm for truss optimization under natural frequency constraints, IEEE Access 10 (2022) 16188–16208.
[74] B. Nouhi, et al., Multi-objective material generation algorithm (MOMGA) for optimization purposes, IEEE Access 10 (2022) 107095–107115.
[75] N. Khodadadi, F. Soleimanian Gharehchopogh, S. Mirjalili, MOAVOA: A new multi-objective artificial vultures optimization algorithm, Neural Comput. Appl. 34 (23) (2022) 20791–20829.
[76] M. Azizi, et al., Multiobjective atomic orbital search (MOAOS) for global and engineering design optimization, IEEE Access 10 (2022) 67727–67746.
[77] N. Khodadadi, S. Talatahari, A. Dadras Eslamlou, MOTEO: a novel multi-objective thermal exchange optimization algorithm for engineering problems, Soft Comput. 26 (14) (2022) 6659–6684.
[78] N. Khodadadi, L. Abualigah, S. Mirjalili, Multi-objective stochastic paint optimizer (MOSPO), Neural Comput. Appl. 34 (20) (2022) 18035–18058.
[79] Q. Al-Tashi, et al., Moth-flame optimization algorithm for feature selection: a review and future trends, Handb. Moth-Flame Optim. Algorithm (2022) 11–34.
[80] A.A. Abdelhamid, et al., Waterwheel plant algorithm: a novel metaheuristic optimization method, Processes 11 (5) (2023) 1502.
[81] N. Abd-Alsabour, S. Ramakrishnan, Hybrid metaheuristics for classification problems, Pattern Recognit. -Anal. Appl. 10 (2016) 65253.
[82] N. Khodadadi, et al., Space truss structures' optimization using metaheuristic optimization algorithms. in *Comprehensive Metaheuristics*, Elsevier,, 2023, pp. 163–179.
[83] A. Kaveh, N. Khodadadi, S. Talatahari, A comparative study for the optimal design of steel structures using CSS and ACSS algorithms, Iran. Univ. Sci. Technol. 11 (1) (2021) 31–54.
[84] N. Khodadadi, S. Mirjalili, Truss optimization with natural frequency constraints using generalized normal distribution optimization, Appl. Intell. 52 (9) (2022) 10384–10397.
[85] L. Lin, M. Gen, Auto-tuning strategy for evolutionary algorithms: balancing between exploration and exploitation, Soft Comput. 13 (2) (2009) 157–168.
[86] N. Khodadadi, et al., BAOA: binary arithmetic optimization algorithm with K-nearest neighbor classifier for feature selection, IEEE Access (2023).
[87] L. Abualigah, et al., Improved prairie dog optimization algorithm by dwarf mongoose optimization algorithm for optimization problems, Multimed. Tools Appl. (2023) 1–41.
[88] K. Sorensen, M. Sevaux, F. Glover, A history of metaheuristics, arXiv Prepr. arXiv: 1704. 00853 (2017).
[89] N. Khodadadi, S.M. Mirjalili, S. Mirjalili, Optimal design of truss structures with continuous variable using moth-flame optimization. in Handbook of Moth-Flame Optimization Algorithm, CRC Press, 2022, pp. 265–280.
[90] S.M. Mirjalili, et al., Grey wolf optimizer, whale optimization algorithm, and moth flame optimization for optimizing photonics crystals. Advances in Swarm Intelligence: Variations and Adaptations for Optimization Problems, Springer,, 2022, pp. 169–179.
[91] N. Khodadadi, et al., An archive-based multi-objective arithmetic optimization algorithm for solving industrial engineering problems, IEEE Access 10 (2022) 106673–106698.
[92] D.H. Wolpert, W.G. Macready, No free lunch theorems for optimization, IEEE Trans. Evolut. Comput. 1 (1) (1997) 67–82.
[93] A. Kaveh, et al., Optimal design of large-scale frames with an advanced charged system search algorithm using box-shaped sections, Eng. Comput. 37 (2021) 2521–2541.
[94] A. Kaveh, S. Talatahari, N. Khodadadi, Hybrid invasive weed optimization-shuffled frog-leaping algorithm for optimal design of truss structures, Iran. J. Sci. Technol., Trans. Civ. Eng. 44 (2020) 405–420.
[95] E.-S.M. El-kenawy, et al., Al-Biruni Earth Radius (BER) metaheuristic search optimization algorithm, Comput. Syst. Sci. Eng. 45 (2) (2023) 1917–1934.
[96] L. Zhang, X. Zou, Z. Su, GA optimization model for time/cost trade-off problem in repetitive projects considering resource continuity, Appl. Math. Inf. Sci. 7 (2) (2013) 611–617.
[97] N.R. Shankar, et al., Time, cost and quality trade-off analysis in construction of projects, Contemp. Eng. Sci. 4 (6) (2011) 289–299.
[98] M. Baghalzadeh Shishehgarkhaneh, et al., Application of classic and novel metaheuristic algorithms in a bim-based resource tradeoff in dam projects, Smart Cities 5 (4) (2022) 1441–1464.
[99] A. Senouci, K. El-Rayes, Time-profit trade-off analysis for construction projects, J. Constr. Eng. Manag. 135 (8) (2009) 718–725.
[100] T. Wang, et al., Time–cost–quality trade-off analysis for planning construction projects, Eng., Constr. Archit. Manag. (2019).
[101] E. Eshtehardian, A. Afshar, R. Abbasnia, Time–cost optimization: using GA and fuzzy sets theory for uncertainties in cost, Constr. Manag. Econ. 26 (7) (2008) 679–691.
[102] E. Eshtehardian, A. Afshar, R. Abbasnia, Fuzzy-based MOGA approach to stochastic time–cost trade-off problem, Autom. Constr. 18 (5) (2009) 692–701.
[103] E. Kalhor, et al., Stochastic time–cost optimization using non-dominated archiving ant colony approach, Autom. Constr. 20 (8) (2011) 1193–1203.
[104] D.-T. Nguyen, et al., A novel multiple objective whale optimization for time-cost-quality tradeoff in non-unit repetitive projects, Int. J. Constr. Manag. (2021) 1–12.
[105] H. Adeli, A. Karim, Scheduling/cost optimization and neural dynamics model for construction, J. Constr. Eng. Manag. 123 (4) (1997) 450–458.
[106] R.-y Huang, K.-S. Sun, System development for non-unit based repetitive project scheduling, Autom. Constr. 14 (5) (2005) 650–665.
[107] O. Moselhi, A. Hassanein, Optimized scheduling of linear projects, J. Constr. Eng. Manag. 129 (6) (2003) 664–673.
[108] R.M. Reda, RPM: Repetitive project modeling, J. Constr. Eng. Manag. 116 (2) (1990) 316–330.
[109] A. Maravas, J.-P. Pantouvakis, Fuzzy repetitive scheduling method for projects with repeating activities, J. Constr. Eng. Manag. 137 (7) (2011) 561–564.
[110] D.A. Wood, Gas and oil project time-cost-quality tradeoff: integrated stochastic and fuzzy multi-objective optimization applying a memetic, nondominated, sorting algorithm, J. Nat. Gas. Sci. Eng. 45 (2017) 143–164.
[111] Z.T. Kosztyán, I. Szalkai, Hybrid time-quality-cost trade-off problems, Oper. Res. Perspect. 5 (2018) 306–318.

[112] K. El-Rayes, A. Kandil, Time-cost-quality trade-off analysis for highway construction, J. Constr. Eng. Manag. 131 (4) (2005) 477–486.
[113] T.S. Adebayo, Trade-off between environmental sustainability and economic growth through coal consumption and natural resources exploitation in China: New policy insights from wavelet local multiple correlation, Geol. J. 58 (4) (2023) 1384–1400.
[114] M. Ghasemi, et al., A new approach for production project scheduling with time-cost-quality trade-off considering multi-mode resource-constraints under interval uncertainty, Int. J. Prod. Res. 61 (9) (2023) 2963–2985.
[115] P.V.H. Son, L.N.Q. Khoi, Building projects with time–cost–quality–environment trade-off optimization using adaptive selection slime mold algorithm, Asian J. Civ. Eng. 24 (5) (2023) 1333–1350.
[116] M. Yılmaz, T. Dede, Multi-objective time–cost trade-off optimization for the construction scheduling with Rao algorithms, Structures 48 (2023) 798–808.
[117] D.H. Tran, L.D. Long, Project scheduling with time, cost and risk trade-off using adaptive multiple objective differential evolution, Eng., Constr. Archit. Manag. (2018).
[118] S. Liu, R. Tao, C.M. Tam, Optimizing cost and CO_2 emission for construction projects using particle swarm optimization, Habitat Int. 37 (2013) 155–162.
[119] M.-Y. Cheng, D.-H. Tran, Opposition-based multiple-objective differential evolution to solve the time–cost–environment impact trade-off problem in construction projects, J. Comput. Civ. Eng. 29 (5) (2015), 04014074.
[120] A. Salman, M. Khalfan, M. Tayyab, Building information modeling (BIM): now and beyond, Australas. J. Constr. Econ. Build. 12 (2012) 15.
[121] W. Shou, et al., A comparative review of building information modelling implementation in building and infrastructure industries, Arch. Comput. Methods Eng. 22 (2) (2015) 291–308.
[122] M. Baghalzadeh Shishehgarkhaneh, et al., Internet of things (IoT), building information modeling (BIM), and digital twin (dt) in construction industry: a review, bibliometric, and network analysis, Buildings 12 (10) (2022) 1503.
[123] S. Hire, S. Sandbhor, K. Ruikar, Bibliometric survey for adoption of building information modeling (BIM) in construction industry– a safety perspective, Arch. Comput. Methods Eng. (2021).
[124] T. Gerrish, et al., BIM application to building energy performance visualisation and management: Challenges and potential, Energy Build. 144 (2017) 218–228.
[125] M.T.H. Khondoker, Automated reinforcement trim waste optimization in RC frame structures using building information modeling and mixed-integer linear programming, Autom. Constr. 124 (2021), 103599.
[126] A. Farzaneh, D. Monfet, D. Forgues, Review of using building information modeling for building energy modeling during the design process, J. Build. Eng. 23 (2019) 127–135.
[127] R. Charef, et al., Building Information Modelling adoption in the European Union: An overview, J. Build. Eng. 25 (2019), 100777.
[128] D.H. Boucher, S. James, K.H. Keeler, The ecology of mutualism, Annu. Rev. Ecol. Syst. 13 (1) (1982) 315–347.
[129] J.N. Holland, J.L. Bronstein, Mutualism, in: S.E. Jørgensen, B.D. Fath (Eds.), in Encyclopedia of Ecol., Academic Press, Oxford, 2008, pp. 2485–2491.
[130] O.N. Fraser, T. Bugnyar, The quality of social relationships in ravens, Anim. Behav. 79 (4) (2010) 927–933.
[131] T. Bugnyar, M. Kijne, K. Kotrschal, Food calling in ravens: are yells referential signals? Anim. Behav. 61 (5) (2001) 949–958.
[132] B. Heinrich, Conflict, cooperation, and cognition in the common raven. in Advances in the Study of Behavior, Elsevier,, 2011, pp. 189–237.
[133] S. Kondo, Dog and human from Raven's perspective: An interpretation of Raven myths of Alaskan Athabascans. Polar, Science 28 (2021), 100633.
[134] C. Erdas, Wolves and Ravens: Defining a unique relationship, Osmosis Magazine 2020 (2020) 6.
[135] D. Stahler, B. Heinrich, D. Smith, Common ravens, Corvus corax, preferentially associate with grey wolves, Canis lupus, as a foraging strategy in winter, Anim. Behav. 64 (2) (2002) 283–290.
[136] Quammen, D., Mind of the raven-Investigations and adventures with wolf-birds. 1999, NEW YORK TIMES 229 W 43RD ST, NEW YORK, NY 10036–3959 USA.
[137] D.-T. Nguyen, J.-S. Chou, D.-H. Tran, Integrating a novel multiple-objective FBI with BIM to determine tradeoff among resources in project scheduling, Knowl. -Based Syst. 235 (2022), 107640.
[138] L.M. Naeni, A. Salehipour, Optimization for project cost management, in: H. Golpîra (Ed.), in Application of Mathematics and Optimization in Construction Project Management, Springer International Publishing, Cham, 2021, pp. 79–118.
[139] A. Panwar, K.N. Jha, Integrating quality and safety in construction scheduling time-cost trade-off model, J. Constr. Eng. Manag. 147 (2) (2021) 04020160.
[140] L. Zhang, J. Du, S. Zhang, Solution to the time-cost-quality trade-off problem in construction projects based on immune genetic particle swarm optimization, J. Manag. Eng. 30 (2) (2014) 163–172.
[141] K.S. Al-Gahtani, Float allocation using the total risk approach, J. Constr. Eng. Manag. 135 (2) (2009) 88–95.
[142] J.Mdl Garza, A. Prateapusanond, N. Ambani, Preallocation of total float in the application of a critical path method based construction contract, J. Constr. Eng. Manag. 133 (11) (2007) 836–845.
[143] L.D. Long, D.-H. Tran, P.T. Nguyen, Hybrid multiple objective evolutionary algorithms for optimising multi-mode time, cost and risk trade-off problem, Int. J. Comput. Appl. Technol. 60 (3) (2019) 203–214.
[144] V. Toğan, N. Berberoğlu, T. Dede, Optimizing of discrete time-cost in construction projects using new adaptive weight formulations, KSCE J. Civ. Eng. 26 (2) (2022) 511–521.
[145] X.S. Yang, A.H. Gandomi, Bat algorithm: a novel approach for global engineering optimization, Eng. Comput. (2012).
[146] S. Arora, S. Singh, Butterfly optimization algorithm: a novel approach for global optimization, Soft Comput. 23 (3) (2019) 715–734.
[147] A. Kaveh, A. Zolghadr, Cycl. Part. Algorithm: A N. meta-heuristic Algorithm (2017).
[148] X.-S. Yang, Flower pollination algorithm for global optimization. in International Conference on Unconventional Computing And Natural Computation, Springer, 2012.
[149] A.A. Heidari, et al., Harris hawks optimization: algorithm and applications, Future Gener. Comput. Syst. 97 (2019) 849–872.
[150] R.A. Zitar, et al., An Intensive and Comprehensive Overview of JAYA Algorithm, its Versions and Applications, Arch. Comput. Methods Eng. 29 (2) (2022) 763–792.
[151] B. Zolghadr-Asli, O. Bozorg-Haddad, X. Chu, in: O. Bozorg-Haddad (Ed.), Krill Herd Algorithm (KHA), in Advanced Optimization by Nature-Inspired Algorithms, Springer Singapore, Singapore, 2018, pp. 69–79.
[152] N.M. Razali, Y.B. Wah, Power comparisons of shapiro-wilk, kolmogorov-smirnov, lilliefors and anderson-darling tests, J. Stat. Model. Anal. 2 (1) (2011) 21–33.
[153] F.J. Massey Jr, The Kolmogorov-Smirnov test for goodness of fit, J. Am. Stat. Assoc. 46 (253) (1951) 68–78.
[154] V.W. Berger, Y. Zhou, Kolmogorov–smirnov test: overview, Wiley statsref: Stat. Ref. Online (2014).
[155] J. Cuzick, A Wilcoxon-type test for trend. Stat. Med. 4 (1) (1985) 87–90.
[156] S. Yue, C. Wang, The influence of serial correlation on the Mann–Whitney test for detecting a shift in median, Adv. Water Resour. 25 (3) (2002) 325–333.
[157] A.C. Elliott, L.S. Hynan, A SAS® macro implementation of a multiple comparison post hoc test for a Kruskal–Wallis analysis, Comput. Methods Prog. Biomed. 102 (1) (2011) 75–80.
[158] E.F. Acar, L. Sun, A generalized Kruskal–Wallis test incorporating group uncertainty with application to genetic association studies, Biometrics 69 (2) (2013) 427–435.