OKU Fen Bilimleri Enstitüsü Dergisi
9(3): 1245-1280, 2026

OKU Journal of The Institute of Science and Technology, 9(3): 1245-1280, 2026

Osmaniye Korkut Ata Üniversitesi Fen Bilimleri Enstitüsü Dergisi

Osmaniye Korkut Ata University Journal of The Institute of Science and Technology

0–1 Sırt Çantası Problemleri için Transfer Fonksiyonlarına Dayalı İkili Zebra Optimizasyon Algoritması

Emine BAŞ[1*], Avni Avnullah KAŞIKÇI[2]

[1]Konya Teknik Üniversitesi, Bilgisayar ve Bilişim Bilimleri Fakültesi, Yazılım Mühendisliği Bölümü, Konya
[2]Konya Teknik Üniversitesi, Lisansüstü Eğitim Enstitüsü, Yazılım Mühendisliği Programı, Konya

[1]https://orcid.org/0000-0003-4322-6010
[2]https://orcid.org/0009-0006-4698-8634
*Sorumlu yazar: ebas@ktun.edu.tr

Araştırma Makalesi

Makale Tarihçesi:
Geliş tarihi: 01.07.2025
Kabul tarihi: 10.11.2025
Online Yayınlanma:15.06.2026

Anahtar Kelimeler:
Zebra
Sırt çantası problemi
İkili optimizasyon
S-Şekilli ve V-Şekilli
Transfer fonksiyonları

ÖZ

Bu çalışmada, sürekli optimizasyon problemlerini çözmeye yönelik olarak önerilmiş yenilikçi bir yaklaşım olan Zebra Optimizasyon Algoritmasını (ZOA) incelemiştir. ZOA, original makalesinde yazarlar tarafından sürekli optimizasyon problemlerini çözmedeki başarısı kanıtlanmıştır. Fakat gerçek dünya problemleri her zaman sürekli değişkenler içermemektir. Birçok gerçek dünya problemleri ayrık değişkenlerde içerebilmektedir. Bu yüzden sürekli optimizason için önerilmiş meta sezgisel algoritmaların ayrık problemler üzerinde de başarıları test edilmelidir. Bu çalışmada, ZOA algoritması ayrık bir problem türü olan ikili optimizasyon problemlerini çözebilecek şekilde tekrar yapılandırmıştır. ZOA'nın sürekli arama uzayı S- ve V- şekilli aktarım fonksiyonları ile ikili arama uzayına haritalanmıştır. Böylece İkili ZOA (BinZOA) algoritması önerilmiştir. BinZOA algoritmasının başarı testleri literatürde sıklıkla ikili optimizasyon problemleri için kullanılan 0-1 sırt çantası problemi üzerinde gerçekleştirilmiştir. 0-1 Sırt çantası problemi, belirli bir kapasite kısıtı altında en değerli öğelerin seçimini içermekte olup, çeşitli alanlarda geniş uygulama alanına sahiptir. ZOA, uyarlanabilir yapısı ve zebraların doğal davranışlarından esinlenen yenilikçi arama stratejileriyle, geleneksel yöntemlere alternatif bir çözüm sunmaktadır. Bu çalışmada, ZOA'nın ikili optimizasyona uyarlanması ele alınmış ve sekiz farklı aktarım fonksiyonu kullanılarak İkili ZOA (BinZOA1, BinZOA2, …, BinZOA8) türetilmiştir. Sırt çantası problemi üzerine yapılan karşılaştırmalı analizler, BinZOA4 varyantının en etkili BinZOA türü olduğunu ortaya koymuştur. Aktarım fonksiyonlarının tek başına sınırlı etki gösterdiği görülürken, XOR tabanlı BinZOAX varyantı tüm veri kümelerinde sıfır optimum değerden ortalama sapma (DMO) (optimal) çözümler elde ederek üstün performans sergilemiştir. BEOV3, WOS2, PSOS2, FPS2 ve BAS2 gibi mevcut yöntemlerle karşılaştırıldığında, hem BinZOA4 hem de BinZOAX doğruluk ve verimlilik açısından daha iyi sonuçlar vermiştir. Bu bulgular, ZOA çerçevesinin karmaşık ve NP-zor problemleri çözmedeki esnekliğini ve etkinliğini doğrulamaktadır. Ayrıca, bu çalışma ZOA'nın ikili optimizasyondaki potansiyelini ortaya koymakta ve algoritmanın daha geniş uygulama alanlarına yönelik kullanılabilirliğine dair ipuçları sunmaktadır. Bununla birlikte, yalnızca aktarım fonksiyonlarına dayanmanın sınırlamaları da kabul edilerek, gelecekteki çalışmalar için çeşitli metodolojilerle geliştirme yapılması önerilmektedir. ZOA'nın ikili optimizasyon problemlerine yönelik ilk uygulamasını temsil eden bu çalışma, alana yeni bakış açıları ve analitik derinlik kazandırarak, daha ileri düzey araştırmalara zemin hazırlamaktadır.

Binary Zebra Optimization Algorithm Based on Transfer Functions for 0–1 Knapsack Problems

Research Article

Article History:
Received: 01.07.2025
Accepted: 10.11.2025
Published online:15.06.2026

Keywords:
Zebra
Knapsack problem
Binary optimizastion
S-Shaped and V-Shaped
Transfer functions

ABSTRACT

This study examines the Zebra Optimization Algorithm (ZOA), an innovative approach proposed for solving continuous optimization problems. In the original paper, the authors demonstrated the success of ZOA in solving continuous optimization problems. However, real-world problems do not always involve continuous variables. Many real-world problems also involve discrete variables. Therefore, meta-heuristic algorithms proposed for continuous optimization should also be tested on discrete problems. In this study, the ZOA algorithm is restructured to solve binary optimization problems, a discrete type of problem. The continuous search space of ZOA is mapped to the binary search space using S- and V-shaped transfer functions. Thus, the Binary ZOA (BinZOA) algorithm is proposed. Success tests of the BinZOA algorithm are conducted on the 0-1 knapsack problem, a frequently used method for binary optimization problems in the literature. The 0-1 knapsack problem, which involves selecting the most valuable items within a given capacity constraint, has a wide range of applications across various domains. ZOA offers an alternative to traditional methods, distinguished by its adaptive structure and innovative search strategies rooted in the natural behaviors of zebras. In advancing ZOA, this paper presents its adaptation to binary optimization, incorporating eight different transfer functions to create Binary ZOA (BinZOA1, BinZOA2, …, BinZOA8). A comparative study on the knapsack problem revealed BinZOA4 as the most effective BinZOA variant. While transfer functions alone showed limited impact, the XOR-based BinZOAX variant achieved optimal, zero- the deviation of the mean from the optimum value (DMO) solutions across all datasets. When compared with established methods (e.g., BEOV3, WOS2, PSOS2, FPS2, BAS2), both BinZOA4 and BinZOAX demonstrated superior performance in accuracy and efficiency. These results confirm the ZOA framework's adaptability and strength in solving complex NP-hard problems. In addition to validating the binary optimization potential of ZOA, this investigation provides a lead for the application of the algorithm in broader fields. However, the study also recognizes the limitations of relying solely on transfer functions, suggesting avenues for future improvements through diversified methodologies. Marking the first application of ZOA to binary optimization problems, this work contributes new perspectives and analytical depth to the field, paving the way for further exploration and optimization.

To Cite: Baş E., Kaşıkçı AA. Binary Zebra Optimization Algorithm Based On Transfer Functions for 0–1 Knapsack Problems. Osmaniye Korkut Ata Üniversitesi Fen Bilimleri Enstitüsü Dergisi 2026; 9(3): 1245-1280.

1. Introduction

Optimization presents a challenge laden with diverse solutions, and the pursuit of optimization involves navigating through the array of potential answers to find the most optimal one for a given problem. Decision variables, constraints, and an objective function delineate the contours of each optimization problem. A spectrum of analytical methods has been introduced to tackle optimization challenges, ranging from gradient-based approaches to numerical computations. However, gradient-based methods, exemplified by techniques like gradient descent, grapple with the limitation of handling only straightforward derivative functions. These methods become impractical when applied to functions that are neither continuous nor differentiable. Moreover, the effectiveness of numerical computation techniques largely depends on the careful selection of the initial solution. Although they can be highly effective in addressing optimization problems, their performance is vulnerable to suboptimal outcomes

if the starting point is not appropriately chosen (Trojovská et al., 2022; Bas and Ihsan, 2023; Ihsan and Doğan, 2023; Sag and Ihsan, 2023; Ihsan and Doğan, 2024). The development of metaheuristic algorithms, predicated upon the quintessential principles of exploration and exploitation, endows these algorithms with the proficiency to discern apt resolutions for optimization quandaries. Exploration denotes the algorithm's aptitude for a comprehensive and precise global scrutiny of the search domain, thereby pinpointing the optimal sector. Conversely, exploitation encapsulates the algorithm's finesse in conducting a localized scrutiny of the search domain to ameliorate solutions. It is imperative for optimization algorithms to calibrate an equitable equilibrium between exploration and exploitation to secure a satisfactory resolution (Liu et al., 2013). The development of metaheuristic algorithms often involves the abstraction of processes observed in nature, including biological evolution, physical dynamics, and behavioral patterns of living organisms. These algorithms are generally classified into three main groups: swarm intelligence-based, evolutionary-based, and physics-inspired techniques (Trojovská et al., 2022). Particularly, physics-based methods have gained attention by mathematically modeling fundamental natural laws to guide the search process. Prominent examples include Simulated Annealing (SA) (Yao, 1995), which mimics the metallurgical annealing process, the Gravitational Search Algorithm (GSA) (Rashedi et al., 2009), based on Newtonian gravitational theory, and the Big Bang–Big Crunch (BB-BC) algorithm (Erol and Eksin, 2006), inspired by cosmological cycles of expansion and contraction.

Beyond these, several other algorithms rooted in physical concepts have been introduced, such as the Water Cycle Algorithm (WCA) (Sadollah et al., 2016), Nuclear Reaction Optimization (NRO) (Wei et al., 2019), Ray Optimization (RO) (Kaveh et al., 2013), Central Force Optimization (CFO) (Formato, 2007), and Galaxy-Based Search Algorithm (GbSA) (Shah-Hosseini, 2011). These techniques rely on analogies from fluid dynamics, nuclear physics, optics, force fields, and astrophysics, respectively.

Among the well-known benchmark problems used to test such algorithms is the Knapsack Problem, a fundamental combinatorial optimization problem. First formally addressed by Smith in 1997, it requires selecting a subset of items, each with a defined weight and value, to maximize total value without exceeding a given capacity constraint (Smith et al., 1997). The problem’s computational complexity increases rapidly with problem size, rendering it NP-hard for large instances (Singh, 2011). Typically, it is expressed as choosing the optimal set of items to fit within a fixed capacity, balancing between total weight and value (Kulkarni and Shabir, 2016).

The Knapsack Problem is commonly explored in two variants: classical and multiple. While the classical version focuses on a single container, the Multiple Knapsack Problem (MKP) extends this to several containers, increasing the complexity. Efficient algorithms and strong upper bounds for MKP are crucial for solving advanced resource allocation problems across diverse fields, making it a vital research area in combinatorial optimization (Smith et al., 1997; Abdel-Basset et al., 2021).

A novel upper boundary proposition is introduced for the Multiple Knapsack Problem (MKP), achieved through the transformation of MKP into the Bounded Sequential Multiple Knapsack Problem (BSMKP)

via a method of relaxation. This methodological relaxation hinges on a variant of the multiple knapsack dilemma wherein item dimensions are divisible. Empirical analyses underscore the utility of this newly proposed upper limit, especially in instances characterized by a minimal ratio of items to knapsacks. Detti's research delves into the pragmatic ramifications of this innovative strategy for MKP, critically evaluating the caliber of upper boundaries across disparate problem magnitudes (Detti, 2021). The essence of Detti's proposition is to unveil an efficacious upper boundary through the lens of "Bounded Sequential Multiple Knapsack Problem" (BSMKP), aiming to validate the efficiency of this relaxation technique in addressing the Multiple Knapsack Problem (MKP). Notably, this upper boundary shines in situations marked by diminutive ratios of items to knapsacks (Detti, 2021). In academic literature, numerous strategies have been extensively investigated to address the complexities of the Knapsack Problem. Among these, the Genetic Algorithm (GA) has emerged as a robust tool for solving combinatorial optimization tasks across diverse fields. Rezoug et al. introduced a hybrid heuristic known as the Guided Genetic Algorithm (GGA), specifically designed for the Multidimensional Knapsack Problem (MKP) (Rezoug et al., 2018). This memetic framework integrates a two-phase structure that combines preliminary data analysis with an improved GA mechanism.

The initial phase employs an efficiency-based approach to extract critical information from the dataset, which subsequently informs both the initialization of the population and the fitness evaluation process. A comprehensive experimental evaluation is conducted using standard MKP benchmark datasets, including the tuning of key GGA parameters and comparison with other established algorithms. The results demonstrate that GGA significantly outperforms the conventional GA and exhibits strong competitiveness against other heuristic methods (Rezoug et al., 2018).

Additionally, the literature presents an enhanced hybrid Genetic Algorithm, referred to as the Guided Genetic Algorithm (GGA), as an effective approach for solving the Multiple Constrained 0-1 Knapsack Problem (MKP). GGA represents a novel and strategic advancement in optimization methodologies, specifically tailored to handle the complexities inherent in MKP instances. Its design is characterized by the incorporation of a pre-optimization phase that leverages the linear programming (LP) relaxation of the MKP, enabling more informed guidance during the evolutionary search process. This approach is further strengthened by the implementation of novel repair and local search operators aimed at refining newly generated solutions. The algorithm is carefully designed to mitigate common issues such as loss of genetic diversity within the population and difficulties in converging to optimal or near-optimal solutions, thereby establishing the Guided Genetic Algorithm (GGA) as an effective method for tackling the Multiple Knapsack Problem (MKP) (Raidl, 1998).

In the context of optimization for the 0-1 Knapsack Problem, Berberler et al. introduced a genetic algorithm that represents solutions as binary strings, where each gene corresponds to the inclusion (1) or exclusion (0) of an item. Fitness evaluation involves summing the values of selected items, while parent selection is conducted via a roulette wheel mechanism, favoring individuals with higher fitness to ensure the transmission of superior traits. The crossover operator aligns genes by index to enable

dominant parents to transfer advantageous features, and mutation is applied to 25% of genes randomly, recursively over the population. This algorithm is extensively tested against randomly generated instances and benchmarked with dynamic programming, consistently achieving optimal solutions (Berberler and Güler, 2016).

Further foundational contributions include approximate algorithms proposed by Sahni et al. in 1975 and a fully polynomial-time approximation scheme developed by Ibarra and Kim in 1975 (Ibarra and Kim, 1975; Sahni et al., 1975). Martello and Toth offered a highly efficient algorithm for large-scale knapsack instances (Martello and Toth, 1990), while Horowitz et al. presented a simpler yet effective solution method (Horowitz et al., 1997). Collectively, these works underscore the rich variety of algorithmic strategies developed to address the 0-1 Knapsack Problem.

Additionally, Baş and Guner enhanced the Crayfish Optimization Algorithm by incorporating bitwise and repair operators to solve the 0-1 knapsack problem effectively (Baş and Guner, 2025).

The Zebra Optimization Algorithm (ZOA), introduced by Trojovská et al. in 2022, represents a state-of-the-art metaheuristic inspired by zebra herd behaviors (Trojovská et al., 2022). It has demonstrated strong performance across various benchmark functions, including unimodal and multimodal test suites, as well as the CEC2015 and CEC2017 reference sets. These comprehensive evaluations highlight ZOA's adaptability and robustness (Trojovská et al., 2022).

ZOA's practical applications extend notably into renewable energy optimization, such as hybrid Maximum Power Point Tracking (MPPT) for solar and wind energy systems. Its integration with Adaptive Neuro-Fuzzy Inference Systems (ANFIS) has led to significant reductions in computational time 26.17% for solar panels and 35.5% for wind turbines thereby improving operational efficiency (Elymany et al., 2024). Furthermore, ZOA has been applied to the tuning of PID controllers within Automatic Voltage Regulator (AVR) systems, demonstrating enhanced voltage regulation when used alongside the Osprey Optimization Algorithm (OOA) (Pazhanimuthu et al., 2023). These applications affirm ZOA's potential for advancing control system performance and energy management.

Overall, these studies validate the effectiveness of ZOA as proposed by Trojovská et al., illustrating its broad applicability and potential to foster further innovation in diverse optimization problems, particularly those related to renewable energy and control systems (Trojovská et al., 2022). This research proposes the novel use of ZOA for solving the knapsack problem, addressing a gap in the literature concerning the application of bio-inspired algorithms based on zebra social and defensive behaviors. The paper presents the algorithm's mathematical formulation and procedural steps, showcasing how the natural dynamics of zebra herds can be translated into an efficient metaheuristic for binary knapsack optimization (Trojovská et al., 2022). Additionally, Baş and Baş have utilized ZOA for estimating weight parameters in multilayer perceptron neural networks (Baş and Baş, 2024a), while ZOA has also been successfully applied to feature selection and uncapacitated facility location problems, further demonstrating its versatility (Baş, 2024; Baş and Baş, 2024b).

This study introduces a novel metaheuristic algorithm inspired by the behavioral patterns of zebras, aimed at effectively solving binary knapsack problems through mathematical modeling of zebra behaviors. Experimental results on various optimization benchmarks demonstrate the Zebra Optimization Algorithm's (ZOA) strong performance. Originally designed for continuous optimization tasks, the classic ZOA algorithm has been adapted in this work to address binary optimization challenges such as the well-known binary knapsack problem. To convert the continuous search space of ZOA into a binary domain, eight well-established transfer functions are utilized comprising four S-shaped and four V-shaped variants. These transfer functions enable the transformation of continuous values into binary representations, thereby improving the balance between exploration and exploitation in the binary search process. Accordingly, eight Binary ZOA (BinZOA) variants named BinZOA1 through BinZOA8 are developed based on these transfer functions. The performance of these variants is rigorously evaluated over 25 knapsack datasets, spanning low, medium, and high dimensionalities. Results indicated that relying solely on transfer functions do not sufficiently enhance BinZOA's effectiveness. Building on this, the most successful BinZOA variant, BinZOA4, is further improved by integrating an XOR logic gate operation, resulting in a new variant termed BinZOAX. BinZOAX demonstrated remarkable success by achieving zero optimality gaps across all tested knapsack datasets. Both BinZOA4 and BinZOAX are benchmarked against six other heuristic algorithms from the literature namely BEOV3, WOS2, PSOS2, FPS2, and BAS2. While BinZOA4 showed competitive but slightly lower performance, BinZOAX consistently ranked first by attaining zero-DMO results in every instance. These findings confirm the effectiveness of the logic gate-enhanced BinZOAX variant for binary optimization problems and suggest promising potential for broader applications in this domain. This study marks the first introduction of a transfer function-based binary adaptation of ZOA, enriching the metaheuristic optimization literature with valuable insights and tools.

1.1. Main Contributions of The Study

In this study, the recently proposed ZOA algorithm is examined. A review of the literature on the ZOA algorithm reveals its success in continuous optimization problems. However, real-world problems do not always involve continuous variables. Most engineering problems involve discrete variables. For optimization algorithms to solve such problems, continuous search space structures must be modified into discrete search space structures. Binary optimization problems are a type of discrete optimization problem. In discrete optimization problems, the search space consists of real integer values, while in binary optimization problems, the search space consists of binary values. This study examines in detail the success of the binary version of the ZOA algorithm on the discrete problem of the 0-1 knapsack problem. The main contributions of the study to the literature are as follows:

- Although the binary version of the ZOA algorithm, Binary ZOA, has been presented in various papers by Baş and colleagues on feature selection and UFL problems, it has not been adequately compared with literature studies. This study addresses this gap. The binary version of ZOA,

BinZOA, is examined based on transfer functions and its performance on the 0-1 knapsack problem is analyzed.

- While the primary goal of this study was to analyze BinZOA's performance on the 0-1 knapsack problem, it was found that even the most successful BinZOA variant, BinZOA4, was insufficient to solve binary optimization problems. Therefore, BinZOA's binary search space was developed using xor gates. This newly derived version was named BinZOAX. This paper introduced the BinZOAX version to the literature for the first time.

1.2. Purpose Of The Study

The aim of this study is to demonstrate the success of the newly proposed ZOA algorithm for continuous optimization problems in binary optimization problems in recent years. Real-world problems do not always consist of continuous variables. In such cases, the proposed metaheuristic algorithms require some updates and improvements to their structures to provide solutions to discrete problems as well. The aim of this study is to configure ZOA to provide solutions to binary problems and to improve its search space exploration and exploitation capabilities. Binary problems are a subset of discrete problems. Instead of real integer values in discrete problems, binary problems contain binary values. These binary values are used in the problem-solving phase. In this study, we utilized S- and V-shaped transfer functions, frequently used in the literature, to transform the continuous search space of the ZOA algorithm into a binary search space. Because there is more than one transfer function, the success of these functions on ZOA is analyzed in this study. Furthermore, the population size and the maximum iteration number, which are fixed parameters in optimization algorithms, are also analyzed in this study. Detailed parameter analyses are important for metaheuristic algorithms. In this study, the success of the proposed Binary ZOA algorithm was tested on the 0-1 knapsack problem, a challenging binary problem. The 0-1 knapsack problem was chosen because its optimal values are generally known in the literature and because open access datasets are abundant. Analyses revealed that BinZOA versions were not sufficiently successful. Binary search space exploration and exploitation capabilities were enhanced using xor logic gates, and the BinZOAX algorithm was proposed. BinZOAX achieved outstanding success by finding optimal values on 25 knapsack datasets. This demonstrates that BinZOAX can also successfully solve various binary optimization problems.

Section 2 of the study details the working structures of the ZOA, BinZOA, and BinZOAX algorithms. Section 3 analyzes the success of the BinZOA and BinZOAX algorithms on 0-1 knapsack problems using experimental results. BinZOAX's success is demonstrated by comparing the results of literature algorithms (BEOV3, WOS2, PSOS2, FPS2, and BAS2) with those of BinZOAX. Section 4 interprets the results, discusses BinZOAX's limitations, and discusses planned applications for future work.

2. Material And Methods

2.1. Zebra Optimization Algorithm (ZOA)

This section introduces the proposed nature-inspired Zebra Optimization Algorithm (ZOA) and presents its corresponding mathematical formulation.

Inspiration:

Zebras, belonging to the equine family, are indigenous to the regions of Eastern and Southern Africa. They are most recognizable by their distinctive black and white striped coats, with the stripes usually running vertically along the neck and body. These unique markings provide various benefits such as camouflage against predators and protection from biting insects. Zebras typically have a body length ranging from 210 to 300 cm, tails measuring between 38 and 75 cm, shoulder heights from 110 to 160 cm, and weigh between 175 and 450 kilograms. Known for their endurance, zebras possess long, slender legs that enable them to reach high speeds when necessary. Similar to other wild equids, each of their feet has a single toe, and their long necks and heads are adapted for efficient grazing on grasses (Trojovská et al., 2022).

In the wild, zebras exhibit two primary social behaviors: guiding the group during foraging and employing defensive tactics against predators. When searching for food, a leading zebra directs the herd toward areas with plentiful grazing, with the rest following closely. To escape threats, zebras often utilize a zigzag running pattern but may also band together to confuse or deter predators. These complex social interactions have been mathematically modeled and serve as the core inspiration behind the Zebra Optimization Algorithm (ZOA) (Trojovská et al., 2022).

Mathematical Modelling:

This subsection presents mathematical simulations of natural zebra behaviors to formulate the modeling of the Zebra Optimization Algorithm (ZOA).

Initialization:

The Zebra Optimization Algorithm (ZOA) is conceptualized as a population-based metaheuristic optimization framework, wherein the metaphorical representation of the population units as zebras stands central. Mathematically, each zebra is delineated as a potential solution vector within the optimization landscape. The habitat or plain inhabited by these zebras is analogous to a specific locus within the multidimensional search space of the problem at hand. The position of each zebra within this search space inherently dictates the value assignments to the decision variables under consideration. Thus, within the ZOA paradigm, each zebra is effectively a vectorial representation of a candidate solution, encapsulating the variable values as its components. The collective of zebras, or the population, is mathematically structured into a matrix form, facilitating the representation and manipulation of multiple candidate solutions simultaneously. The inception of the zebra positions within this matrix is

initiated through stochastic processes, thereby determining the preliminary configuration of the ZOA population matrix (Trojovská et al., 2022).

Further, the decision-making parameters for each zebra, or optimization agent, are intrinsically tied to its spatial coordinates within the search domain. Hence, every zebra within the ZOA framework is not merely a metaphorical entity but serves as a vectorial embodiment of solution hypotheses, with each vector component mirroring a decision variable pertinent to the optimization challenge. This vector is, in essence, a conduit through which the ZOA seeks to navigate the solution space. The architectural underpinnings of the zebra collective, or population matrix, are thus established through a matrix representation, offering a structured approach to simulate and evaluate the efficacy of multiple solution vectors in parallel. The initial deployment of zebras within the search terrain is governed by aleatory mechanisms, setting the stage for the exploratory dynamics of the ZOA. Equation (1) delineates the formal parameters and structure defining the initial ZOA population matrix, marking the commencement of the optimization journey (Ghadi et al., 2023).

$$X = \begin{bmatrix} X_1 \\ \vdots \\ X_i \\ \vdots \\ X_N \end{bmatrix}_{N\times m} = \begin{bmatrix} x_{1,1} & \cdots & x_{1,j} & \cdots & x_{1,m} \\ \vdots & \ddots & \vdots & \iddots & \vdots \\ x_{i,1} & \cdots & x_{i,j} & \cdots & x_{i,m} \\ \vdots & \iddots & \vdots & \ddots & \vdots \\ x_{N,1} & \cdots & x_{N,j} & \cdots & x_{N,m} \end{bmatrix}_{N\times m}, \tag{1}$$

In this context, X denotes the entire zebra population, while X_i refers to the i^{th} individual zebra. The element $x_{i,j}$ represents the value assigned by the i^{th} zebra to the j^{th} decision variable. The population size, or the total number of zebras, is given by N, and m indicates the number of decision variables involved. Each zebra corresponds to a potential solution within the optimization problem. Consequently, the objective function can be calculated based on the variable values suggested by each zebra. The resulting objective function values are collectively represented as a vector, as shown in Equation (2) (Trojovská et al., 2022).

$$F = \begin{bmatrix} F_1 \\ \vdots \\ F_i \\ \vdots \\ F_N \end{bmatrix}_{N\times 1} = \begin{bmatrix} F(X_1) \\ \vdots \\ F(X_i) \\ \vdots \\ F(X_N) \end{bmatrix}_{N\times 1}, \tag{2}$$

The vector F contains the objective function values corresponding to each zebra, with F_i representing the objective function value for the i^{th} zebra. By comparing these values, the quality of each candidate solution can be assessed, enabling the determination of the optimal solution. For minimization tasks, the zebra with the lowest objective function value is considered the best solution, while for maximization problems, the zebra with the highest value holds that distinction. Since the zebras' positions-and thus

their objective values-are updated at every iteration, it is essential to re-evaluate and identify the best solution in each cycle (Trojovská et al., 2022).

The algorithm updates the population members by mimicking two key natural behaviors of wild zebras: grazing and defense against predators. Accordingly, each iteration in ZOA involves two main stages:

(i) Grazing,

(ii) Defense strategies against predators.

Phase 1: Foraging Behavior

In the initial phase, individuals update their positions by simulating the grazing behavior of zebras. Zebras primarily feed on grasses and reeds but can adapt their diet to include buds, fruits, bark, roots, and leaves when preferred food is scarce. They spend about 60% to 80% of their time grazing, depending on the quality and availability of vegetation. Plain zebras consume taller, less nutritious grasses, which helps create better conditions for other species that feed on shorter, more nutrient-rich grasses below. In the Zebra Optimization Algorithm (ZOA), the best individual is called the lead zebra and guides others toward its position in the search space during the grazing phase. The position updates during this phase are mathematically expressed by specific equations (Trojovská et al., 2022).

$$x_{i,j}^{\text{new},P1} = x_{i,j} + r \cdot \left(PZ_j - I \cdot x_{i,j}\right) \tag{3}$$

$$X_i = \begin{cases} X_i^{\text{new},P1}, & F_i^{\text{new},P1} < F_i \\ X_i, & \text{else} \end{cases}, \tag{4}$$

where, $x_i^{\text{new},P1}$ represents the new state of the i^{th} zebra in the first phase, $x_{i,j}^{\text{new},P1}$ is the j^{th} dimension value, $F_i^{\text{new},P1}$ is the objective function value, PZ is the lead zebra, PZ_j is its j^{th} dimension, r is a random number in the [0, 1] range, and $I = round\ (1 + rand)$, where $rand$ is a random number in the [0, 1] range. Thus, $I \in 1,2$, and if the parameter I is 2, there will be much more variation in the population movement during this phase (Trojovská et al., 2022).

Phase 2: Defense Strategies Against Predators

In the second phase of the Zebra Optimization Algorithm (ZOA), the position updates of individuals are inspired by zebras' defensive behaviors against predators. Lions are the main threats to zebras, but other predators such as cheetahs, leopards, wild dogs, and brown hyenas also pose significant risks. Zebras adjust their defense tactics depending on the predator's nature and behavior. Against lions, zebras typically use evasive actions like zigzag running and sudden, unpredictable turns to avoid capture. On the other hand, when facing smaller predators like hyenas or wild dogs, zebras often respond more aggressively by grouping together and coordinating their defense to confuse and discourage the

attackers. In ZOA, these behaviors are abstracted into two distinct scenarios, each selected with equal probability for updating positions during this defense phase (Trojovská et al., 2022).

(i) A zebra succumbs to a lion attack, prompting the zebra to adopt an evasive strategy;

(ii) Other predators assail a zebra, eliciting the zebra to opt for an aggressive strategy. In the former scenario, when zebras are subjected to lion attacks, they expeditiously retreat from the locus of the lion's assault within their present states. Mathematically articulated, this strategy can be modeled employing Equation (5). In the latter scenario, when other predators initiate an assault on a zebra, fellow zebras within the herd converge towards the targeted zebra, assume a defensive posture, and endeavor to confound the assailant. The mathematical formulation of this strategy is similarly expressed using Equation (5). During the process of updating the positions of zebras, a novel position is accepted for a zebra if it yields a superior value for the objective function in the new position. This updating condition is formalized by Equation (6) (Trojovská et al., 2022).

$$x_{i,j}^{\text{new},P2} = \begin{cases} S_1: x_{i,j} + R \cdot (2r-1) \\ \quad \cdot \left(1 - \frac{t}{T}\right) \cdot x_{i,j}, & P_s \leq 0.5 \\ S_2: x_{i,j} + r \cdot \left(AZ_j - I \cdot x_{i,j}\right), & \text{else} \end{cases} \tag{5}$$

$$X_i = \begin{cases} X_i^{\text{new},P2}, & F_i^{\text{new},P2} < F_i \\ X_i, & \text{else}, \end{cases} \tag{6}$$

where, $x_{i,j}^{\text{new},P2}$ represents the new state of the i^{th} zebra in the second phase, $x_{i,j}^{\text{new},P2}$ denotes its j^{th} dimensional value, F_i^{new} (P2) represents the updated objective function value, where t denotes the current iteration number and T is the total number of iterations allowed. The constant R is set to 0.01, and P_s indicates the probability of randomly choosing between two strategies, ranging from 0 to 1. AZ refers to the state of the attacked zebra, with AZ_j being its value in the j^{th} dimension. Figure 1 illustrates the overall flowchart of the Zebra Optimization Algorithm (ZOA), while Figure 2 provides the corresponding pseudo-code for the algorithm's implementation (Trojovská et al., 2022).

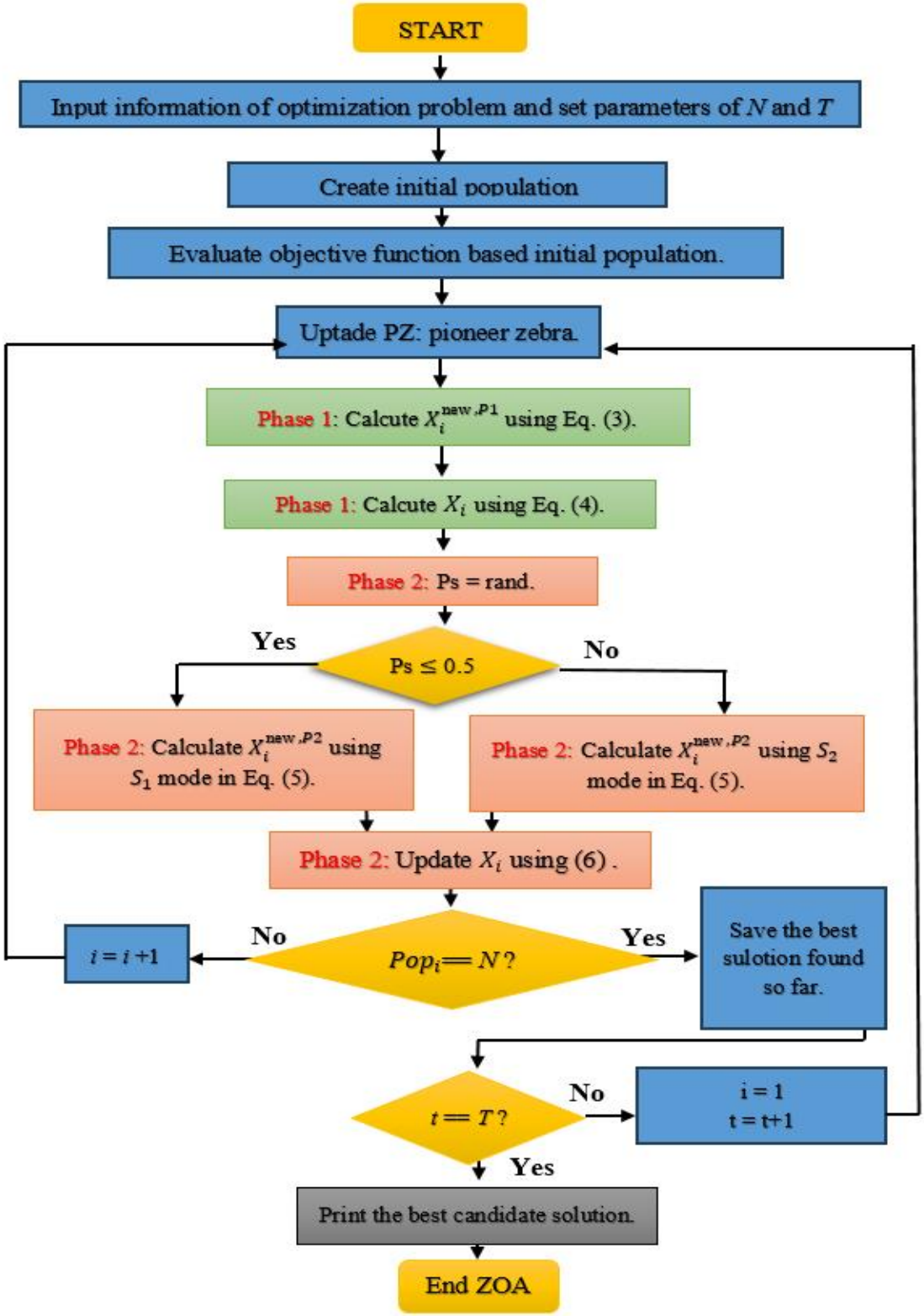

Figure 1. The flowchart of ZOA (Trojovská et al., 2022)

Start ZOA:
Input: The optimization problem information.
Set the number of iterations (T) and the number of zebras' population (N).
Initialize the position of zebras and evaluate their fitness function.
For t = 1: *T*
  Update zebra leader (ZL).
  For i = 1: *N*
    Stage I: Foraging activity
    Calculate new status of the i^{th} zebra using Eq. (3).
    Update the i^{th} zebra using Eq. (4).
    Stage II: Anti-predators' defensive techniques
    Ps = rand
    If Ps <0.5 Then
      Exploitation (Defensive technique against lion)
      Calculate the new status of the i^{th} zebra using Eq. (5).
    Else
    Exploration (Defensive techniques against other predators)
      Calculate the new status of the i^{th} zebra using Eq. (5).
    End If
    Update the i^{th} zebra using Eq. (6).
  End For
Save best candidate solution so far.
End For
Output: The best solution obtained by ZOA for given optimization problem.
End ZOA.

Figure 2. The pseudo-code of ZOA (Trojovská et al., 2022)

2.1.1. The computational complexity of ZOA

The computational complexity of ZOA in the initialization phase is obtained by multiplying the number of zebras (N) and the problem size variables (m) ($O(N \times m)$). In ZOA, in each iteration, each population member is updated in two stages and the objective function is evaluated. This is obtained by computation in $O(2 \times N \times m \times T)$, where T represents the maximum number of iterations. The total computational complexity for ZOA is calculated as $O(N \times m \times (1 + 2 \times T))$ (Trojovská et al., 2022).

2.2. Binary Zebra Optimization Algorithm (BinZOA)

This section introduces a binary version of the Zebra Optimization Algorithm (BinZOA) tailored for solving 0-1 knapsack problems. Although the original ZOA excels in continuous optimization, it cannot directly handle discrete problems like the knapsack. To overcome this limitation, the algorithm is adapted to work within a binary search space by applying transfer functions that convert continuous outputs into binary decisions where 0 means excluding an item and 1 means including it in the knapsack. By mapping ZOA's continuous search space to a binary domain, BinZOA becomes suitable for discrete optimization tasks. Literature offers various transfer functions to perform this conversion, such as S-shaped, V-shaped, taper-shaped, U-shaped, Z-shaped, and X-shaped functions. Among these, S-shaped and V-shaped functions are the most widely used. This work employs four different S-shaped and four

different V-shaped transfer functions, summarized in Table 1 along with their mathematical expressions and key features. Consequently, eight BinZOA variants are created: BinZOA1 to BinZOA4 use transfer functions S1 to S4, while BinZOA5 to BinZOA8 correspond to V1 through V4. The transformation of continuous values into binary form follows the mathematical expression given in Equation (7).

$$X_{bin(i,:)} = transfer_{func}(X(i,:), i, m, transfer_{number}) \tag{7}$$

According to the equation above, $x_{bin(i,:)}$ represents the binary version of the i^{th} row of the X matrix. The function call to $transfer_{func}(X(i,:), i, m, transfer_{number})$ refers to a process that converts the value at the i^{th} index of the vector, utilizing the dimension (m) and the specified transfer function ($transfer_{func}$). The $X_{bin(i,:)}$ matrix can be expressed as follows in the subsequent equation.

$$X_{bin(i,:)} = \begin{bmatrix} X\_Bin_1 \\ \vdots \\ X\_Bin_i \\ \vdots \\ X\_Bin_N \end{bmatrix}_{N \times m} \tag{8}$$

In the study, all elements of the population are converted into discrete expressions as depicted in Equation (7), and this conversion is applied to the entire population. A total of eight transfer functions are used as the transfer function. The $transfer_{number}$ variable in Equation (7) indicates which of these eight functions will be utilized. The $transfer_{func}$ is designed to transform a population using a specific type of transfer function (S-Shape or V-Shape) and a particular set of parameters. Figure 3 shows the pseudo-code of BinZOA.

Table 1. S-Shaped and V-Shaped transfer functions (Ervural et al., 2023; Abdel-Basset et al., 2021)

S-Shaped		V-Shaped	
S1(x)	$\frac{1}{1+e^{-x}}$	**V1*(x)***	$\left\lvert \frac{2}{\pi} arc\mathrm{Tan}\left(\frac{2}{\pi}x\right) \right\rvert$
S2*(x)*	$\frac{1}{1+e^{-2x}}$	**V2*(x)***	$\lvert \tan h(x) \rvert$
S3*(x)*	$\frac{1}{1+e^{-\frac{x}{2}}}$	**V3*(x)***	$\left\lvert \frac{x}{\sqrt{1+x^2}} \right\rvert$
S4*(x)*	$\frac{1}{1+e^{-\frac{x}{3}}}$	**V4*(x)***	$\left\lvert \mathrm{erf}(\frac{\sqrt{\pi}}{2}x) \right\rvert$

Start BinZOA:
Input: The optimization problem information.
Set the number of iterations (T) and the number of zebras' population (N).
Initialize the position of zebras and evaluate their fitness function.
Convert position of zebras to binary form with transfer function using Eq. (7).
For t = 1: *T*
  Update zebra leader (ZL).
  Convert ZL to binary form with transfer function using Eq. (7).
  For *i* = 1: *N*
    Stage I: Foraging activity
    Calculate new status of the i^{th} zebra using (3).
    Convert i^{th} zebra to binary form with transfer function using Eq. (7).
    Update the i^{th} zebra using (4).
    Stage II: Anti-predators' defensive techniques
    Ps= rand
    If *Ps* <0.5 Then
      Exploitation (Defensive technique against lion)
      Calculate the new status of the i^{th} zebra using (5).
    Else
    Exploration (Defensive techniques against other predators)
      Calculate the new status of the i^{th} zebra using (5).
      Convert i^{th} zebra to binary form with transfer function using Eq. (7).
    End If
    Update the i^{th} zebra using (6).
  End For
  Save best candidate solution so far.
End For
Output: The best solution obtained by BinZOA for given optimization problem.
End BinZOA.

Figure 3. The pseudo-code of BinZOA (Trojovská et al., 2022)

2.2.1. The computational complexity of BinZOA

Since the BinZOA algorithm code does not contain any loop structures that would affect the calculation of complexity differently than the ZOA algorithm, the computational complexities of BinZOA and ZOA algorithms are the same. In other words, the computational complexity of BinZOA is $O(N \times m \times (1 + 2 \times T))$.

2.2.2. XOR logic gate method (for BinZOAX)

In recent years, logic gates have been frequently used in the development of binary optimization algorithms (Baş et al., 2024). It is very suitable for binary optimization problems because the logic gate inputs are binary values ({0, 1}) and the output values are binary values ({0, 1}). In particular, the xor logic gate (exclusive OR) produces four different states for two different variables (x and y), and 50% of the outputs are 1 value and 50% are zero values. The xor truth table of xor logic gate is shown in Table 2. In this study, logic gates were used in the development of BinZOA. It has been shown in the experimental part of the study that transfer functions alone are not sufficient to explore binary space. Therefore, after the most successful transfer function was determined as S4, binary candidate solutions for BinZOAS4 are produced using the xor logic gate method. The xor logic gate method is shown in Equation (9). The candidate zebra population produced with this xor gate is effective in increasing the

performance of BinZOA. The proposed new BinZOA variation is named BinZOAX.

$$X_{bin\_new(i)} = (X_{bin(i)} \text{ xor } (X_{bin(rand)} \text{ xor } PZ_{bin})) \quad (9)$$

where $X_{bin\_new(i)}$ represents i^{th} new binary the zebra, $X_{bin(i)}$ represents i^{th} binary the zebra, $X_{bin(rand)}$ represents a random binary the zebra, and PZ_{bin} represents the best binary zebra in binary zebra population.

Table 2. Truth table of xor logic gate

Input variables		Output variable
A	***B***	***C***
0	0	0
0	1	1
1	0	1
1	1	0

The computational complexity of BinZOAX:

Since the BinZOAX algorithm code does not contain any loop structures that would affect the calculation of complexity differently than the ZOA and BinZOA algorithms, the computational complexities of BinZOAX, BinZOA, and ZOA algorithms are the same. In other words, the computational complexity of BinZOAX is *O(N × m × (1 + 2 × T)).*

2.3. Knapsack Problem Definition

The 0-1 knapsack problem is a classic optimization challenge that involves selecting a subset of items to include in a knapsack without exceeding its capacity. This problem belongs to the NP-hard category, meaning that the time required to find an exact solution grows rapidly as the problem size increases. The mathematical formulation of the 0-1 knapsack problem is given in Equation 10 (Abdel-Basset, et al., 2021; Baş, 2023).

$$Max \quad f(x) = \sum_{s=1}^{m} X_{bin(s)} \times Profit_s, \quad (10)$$

$$Subject\ to\ = \sum_{s=1}^{m} Weight_s \times X_{bin(s)} \leq Capacity_{bag}$$

$X_{bin(s)} \epsilon\ 0\ or\ 1$, *s=0, 1, 2, …, m*

where $Profit_s$ represents the profit (or value) of each item, while $Weight_s$ denotes the weight of each item. The maximum carrying capacity of the knapsack is defined as Capacitybag. The variable m indicates the dimension, and $X_{bin(s)}$ is a binary decision variable, where $X_{bin(s)} = 1$ means the item is

included in the knapsack, and $X_{bin(s)} = 0$ means it is excluded.

3. Experimental Result And Discussion

This study presents a tailored adaptation of the Zebra Optimization Algorithm (ZOA) for solving the knapsack problem, referred to as Binary ZOA (BinZOA). Eight different BinZOA variants have been developed based on distinct S-shaped and V-shaped transfer functions. The original ZOA code is obtained from the Mathworks repository and then modified accordingly.

3.1. Experimental Setup

The all implementations have been coded in the Matlab programming environment, and performance evaluations have been conducted using a computer equipped with an Apple Silicon M2 CPU and 16.0 GB of RAM. All applications performed in this study are performed on a computer with the hardware and software specifications specified in Table 3.

Table 3. Computer specifications

Name	Settings
Hardware	16C
CPU	Apple M2 Pro
Frequency	3.49 Mhz
RAM	16.0 GB
Operation System	macOS Sonoma
Software Environment	Matlab R2018b

3.2. Details Of The Dataset Used In The Study

BinZOA's performance is tested on small, medium, and large-scale knapsack datasets, using a collection of 25 diverse problem instances. Table 4 details the definitions of these datasets, which include problem sizes ranging from 8 to 24 items and are sourced from https://pages.mtu.edu/~kreher/cages/Data.html.

Table 4. Details of the dataset used in the study

Dataset ID	Max. bag weight	Dimension	Opt. value	Dataset ID	Max. bag weight	Dimension	Opt. value
KP8a	1863633	8	3924400	***KP16d***	4550938	16	9348889
KP8b	1822718	8	3813669	***KP16e***	3760429	16	7769117
KP8c	1609419	8	3347452	***KP20a***	5169647	20	10727049
KP8d	2112292	8	4187707	***KP20b***	4681373	20	9818261
KP8e	2493250	8	4955555	***KP20c***	5063791	20	10714023
KP12a	2805213	12	5688887	***KP20d***	4286641	20	8929156
KP12b	3259036	12	6498597	***KP20e***	4476000	20	9357969
KP12c	2489815	12	5170626	***KP24a***	6404180	24	13549094
KP12d	3453702	12	6992404	***KP24b***	5971071	24	12233713
KP12e	2520392	12	5337472	***KP24c***	5870470	24	12448780
KP16a	3780355	16	7850983	***KP24d***	5762284	24	11815315
KP16b	4426945	16	9352998	***KP24e***	6654569	24	13940099
KP16c	4323280	16	9151147				

3.3. Parameter Analysis On BinZOA1 For KPs

This subsection analyzes two variable parameter values (population size (*N*) and maximum number of cycles (*T*)) that directly affect the success of ZOA. The ZOA algorithm does not have a fixed parameter setting. Therefore, this subsection focuses on population size and maximum operating cycle analysis. To analyze the impact of population size (*N*) and maximum number of cycles (*T*) on algorithm performance, BinZOA1 is evaluated across ten different population sizes and six different maximum number of cycles (population size (*N*) = {10, 20, ..., 100}; maximum number of cycles (*T*)={100, 500, 1000, 2000, 5000, and 10000}) on five knapsack instances (KP16a to KP16e), with results shown in Table 5 and Table 6. The maximum iteration count is fixed at 500 for population size (*N*) analysis. The population size (*N*) is fixed at 20 for maximum number of cycles (*T*) analysis. Findings indicate that increasing the population size improves BinZOA1's success rate but also leads to longer computational times. The decrease in running time in Table 5 depends on the population size. As the population size increases, the study time also increases. Table 5 shows an inverse relationship between the population value and running time. This situation indicates the need to choose a more reasonable population size rather than a high one. However, for fairness in literature comparisons, this value is kept equal across all algorithms.

A similar situation to the population size analyses is observed in the maximum cycle analyses. Increasing the maximum number of cycles increased the success rate of BinZOA1 but prolonged the runtimes. Analyses of BinZOA variants determined a population size as 20 and a maximum number of cycles as 5000. These values are chosen because these parameter values are frequently used in literature comparisons and to ensure fair comparisons.

Table 5. The average fitness and time results of population size (*N*) analysis on BinZOA1 for KPs (*T*=500; *N*={10, 20, 30, 40, 50, 60, 70, 80, 90, and 100})

Dataset ID	BinZOA1									
	Average									
	10	**20**	**30**	**40**	**50**	**60**	**70**	**80**	**90**	**100**
KP16a	7801929	7830468	7830981	7838511	7834770	7838479	7842529	7847191	**7849087**	7847191
KP16b	9298973	9289819	9327551	9336867	9342377	9339211	**9352472**	9349075	9346436	9351419
KP16c	9084718	9112616	9126483	9134266	9143582	9134670	9130890	9142617	**9146223**	**9146223**
KP16d	9319854	9329207	9335665	9335665	9339707	9346449	9334822	9347669	9346449	**9347669**
KP16e	7746434	7748888	7760707	7759720	7763531	7761014	7758350	7761697	7767254	**7767633**
	Time									
KP16a	**0.3716**	0.4015	0.4300	0.4641	0.5062	0.4965	0.5645	0.5857	0.6189	0.6330
KP16b	**0.3106**	0.3327	0.3613	0.3984	0.4260	0.4347	0.4871	0.5179	0.5636	0.5703
KP16c	**0.2973**	0.3306	0.3628	0.4076	0.4382	0.4411	0.4910	0.5317	0.5390	0.5562
KP16d	**0.2993**	0.3376	0.3839	0.4011	0.4945	0.4344	0.5509	0.4815	0.5348	0.5365
KP16e	**0.3038**	0.3858	0.4150	0.3816	0.4316	0.4928	0.5226	0.4845	0.5209	0.5146

Table 6. The average fitness and time results of maximum number of cycles (*T*) analysis on BinZOA1 for KPs (*N*=20; *T*={100, 500, 1000, 2000, 5000, and 10000})

Dataset ID	BinZOA1					
	Average					
	100	**500**	**1000**	**2000**	**5000**	**10000**
KP16a	7779060.75	7830468	7837553.1	7841580.5	7847616.55	**7850983**
KP16b	9230345.5	9289819	9343808.55	9349338.5	9349338.5	**9352998**
KP16c	9061085.85	9112616	9133412.7	9139304.95	9146971.45	**9151147**
KP16d	9274849.05	9329207	9336329.9	9341536.5	9345517.7	**9348889**
KP16e	7709982.4	7748888	7761035.8	7766890.85	7767632.9	**7769117**
	Time					
KP16a	**0.2814**	0.4015	0.4894	0.5944	0.9321	1.5407
KP16b	**0.2840**	0.3327	0.3892	0.5290	0.8774	1.5642
KP16c	**0.2807**	0.3306	0.4098	0.5139	0.8860	1.6575
KP16d	**0.2847**	0.3376	0.3900	0.4991	0.8649	1.7010
KP16e	**0.2849**	0.3858	0.3943	0.6835	0.8800	1.5409

3.4. Detailed Analysis Of BinZOA Variations For KPs

For the comparative study of BinZOA variants, the parameters are set as summarized in Table 7: population size was fixed at 20, maximum iterations at 5000, and problem dimensions varied with dataset sizes of 8, 12, 16, 20, and 24. Each variant is independently run 20 times to ensure statistical robustness. The analysis computed key performance metrics including best, worst, average, standard deviation (std), and the deviation of the mean from the optimum value (DMO) metrics (defined in Equation (11) (Baş, 2023)). Results are presented both in tables and column charts to support clear comparison and interpretation.

$$DMO(\%) = \frac{(Opt.\ fitness\ value - average\ fitness\ value)}{Opt.\ fitness\ value} \times 100 \tag{11}$$

When analyzing the variations of the eight transfer functions (S-shaped and V-shaped) in BinZOA, the best results as indicated in Table 8 reveal that the optimal values change as the number of dimensions increases. The best results are found for dimensions 8, 12, 16, 20, and 24. These best results are distinctly highlighted and emphasized in bold. The worst outcomes for the BinZOA variations across the 8 transfer functions are presented in Table 9. The worst results do not show any variation across the 8 different transfer functions in 8 dimensions, indicating consistency in performance across these functions. Average results are provided in Table 10 and a comparison of the sums of these results across the 8 transfer functions is shown in Figure 4. Standard deviation values are depicted in Table 11 and a comparison of the sums of these results across the 8 transfer functions is shown in Figure 5. Table 12 presents the DMO values, while Figure 6 compares the summed results of all eight transfer function variants. Based on the overall average results, BinZOA3 emerged as the best-performing variation. When considering the total standard deviation, BinZOA4 shows the most consistent performance. The DMO values align closely with the standard deviation findings, reinforcing that BinZOA3 and BinZOA4 are the most effective BinZOA variants overall. Conversely, BinZOA2 and BinZOA8 were identified

as the least successful.

Figures 7 through 9 display convergence graphs for the BinZOA variants on three randomly selected knapsack problem instances. These graphs visually demonstrate the algorithm's performance, effectiveness, and stability across different problem settings. They provide insight into how quickly and reliably each BinZOA variant converges toward optimal or near-optimal solutions, illustrating the dynamics of the search process in various scenarios.

Tables 13 and 14 present the results of the Wilcoxon signed-rank test conducted between BinZOA4 the most successful BinZOA variant and the other BinZOA variants. The Wilcoxon test is a non-parametric statistical method used to assess whether two related samples differ significantly. A *p* value below 0.05 indicates a statistically significant difference between the results, corresponding to an *h* value of 0. Conversely, a *p* value above 0.05 suggests no significant difference, with an *h* value of 1 (Baş and Yildizdan, 2024). Table 12 reports the *p* values, while Table 14 shows the associated *h* values.

The analysis reveals that BinZOA4 produces significantly different results from BinZOA1, BinZOA2, and BinZOA8 when applied to high-dimensional knapsack problems. However, for lower-dimensional instances, the results across all BinZOA variants tend to be similar. Furthermore, there is no significant difference detected between the outcomes of BinZOA4 and BinZOA3. Similarly, BinZOA4 and BinZOA5 yield comparable results overall, except in the cases of KP16d, KP16e, and KP20b problem instances, where some differences are observed.

Table 7. Basic settings used in the study for BinZOAs

Parametric variables	Value
Number of zebras in the population (*N*)	20
Maximum number of cycles (*T*)	5000
Knapsack Problem Dimension (*m*)	8, 12, 16, 20, 24

Table 8. The best fitness values of the BinZOAs on binary conversion functions (S- and V- shaped transfer functions)

Dataset ID	BinZOAs							
	BinZOA1	**BinZOA2**	**BinZOA3**	**BinZOA4**	**BinZOA5**	**BinZOA6**	**BinZOA7**	**BinZOA8**
KP8a	3924400	3924400	3924400	3924400	3924400	3924400	3924400	3924400
KP8b	3813669	3813669	3813669	3813669	3813669	3813669	3813669	3813669
KP8c	3347452	3347452	3347452	3347452	3347452	3347452	3347452	3347452
KP8d	4187707	4187707	4187707	4187707	4187707	4187707	4187707	4187707
KP8e	4955555	4955555	4955555	4955555	4955555	4955555	4955555	4955555
KP12a	5688887	5688887	5688887	5688887	5688887	5688887	5688887	5688887
KP12b	6498597	6498597	6498597	6498597	6498597	6498597	6498597	6498597
KP12c	5170626	5170626	5170626	5170626	5170626	5170626	5170626	5170626
KP12d	6992404	6992404	6992404	6992404	6992404	6992404	6992404	6992404
KP12e	5337472	5337472	5337472	5337472	5337472	5337472	5337472	5337472
KP16a	7850983	7850983	7850983	7850983	7850983	7850983	7850983	7850983
KP16b	9352998	9352998	9352998	9352998	9352998	9352998	9352998	9352998
KP16c	9151147	9151147	9151147	9151147	9151147	9151147	9151147	9151147
KP16d	9348889	9348889	9348889	9348889	9348889	9348889	9348889	9348889
KP16e	7769117	7769117	7769117	7769117	7769117	7769117	7769117	7769117
KP20a	10727049	10727049	10727049	10727049	10727049	10705222	10727049	10717569
KP20b	9818261	9774200	9818261	9818261	9818261	9818261	9818261	9790311
KP20c	10714023	10712572	10714023	10714023	10714023	10714023	10714023	10712572
KP20d	8915396	8929156	8929156	8929156	8929156	8929156	8929156	8929156

KP20e	9357969	9357969	9357969	9357969	9357969	9357969	9357969	9357969
KP24a	13521334	13517399	13521334	13521334	13514028	13514028	13517963	13517963
KP24b	12176736	12162024	12213704	12233713	12220754	12193732	12190267	12170258
KP24c	12445379	12437046	12445379	12408833	12448780	12440695	12448780	12448780
KP24d	11810051	11761633	11815315	11801123	11793333	11780518	11783685	11780518
KP24e	13897782	13885329	13930566	13889768	13940099	13925043	13909292	13858935

Table 9. The worst fitness values of the BinZOAs on binary conversion functions (S- and V- shaped transfer functions)

	BinZOAs							
Dataset ID	**BinZOA1**	**BinZOA2**	**BinZOA3**	**BinZOA4**	**BinZOA5**	**BinZOA6**	**BinZOA7**	**BinZOA8**
KP8a	3924400	3924400	3924400	3924400	3924400	3924400	3924400	3924400
KP8b	3813669	3813669	3813669	3813669	3813669	3813669	3813669	3813669
KP8c	3347452	3347452	3347452	3347452	3347452	3347452	3347452	3347452
KP8d	4187707	4187707	4187707	4187707	4187707	4187707	4187707	4187707
KP8e	4955555	4955555	4955555	4955555	4955555	4955555	4955555	4955555
KP12a	5688887	5682404	5688887	5688887	5688887	5688887	5688887	5688887
KP12b	6498597	6460377	6498597	6498597	6473019	6498597	6498597	6498597
KP12c	5170626	5164941	5170626	5170626	5164941	5169676	5169676	5169676
KP12d	6992404	6992404	6992404	6992404	6992404	6992404	6992404	6988075
KP12e	5337472	5337472	5337472	5337472	5337472	5337472	5337472	5337472
KP16a	7821574	7796706	7823318	7823318	7823318	7821255	7789517	7805404
KP16b	9324296	9286887	9347736	9347736	9324296	9292636	9324296	9292636
KP16c	9116878	9072812	9126526	9151147	9151147	9105949	9116878	9079887
KP16d	9305859	9270229	9348889	9336691	9288862	9296536	9302090	9296272
KP16e	7754276	7690592	7769117	7754276	7747451	7733329	7726504	7726504
KP20a	10663195	10594306	10665994	10659097	10604758	10606867	10645474	10599355
KP20b	9728193	9660838	9753810	9744578	9717541	9689498	9666409	9679371
KP20c	10646849	10559278	10642722	10655671	10565530	10555608	10584640	10564217
KP20d	8822946	8766076	8865083	8865508	8821793	8839299	8822209	8783240
KP20e	9290479	9309340	9330924	9330924	9320597	9319658	9330924	9253107
KP24a	13385042	13281317	13401096	13415824	13339723	13366858	13355225	13353196
KP24b	12074951	12016797	12102100	12108937	12095767	12060548	12042082	12004186
KP24c	12304523	12284395	12329608	12334644	12309551	12297196	12293290	12261498
KP24d	11677031	11608256	11708673	11684111	11673377	11661004	11649678	11578840
KP24e	13792494	13685095	13817213	13801423	13769243	13721891	13765976	13717756

Table 10. The average fitness values of the BinZOAs on binary conversion functions (S- and V- shaped transfer functions)

Dataset ID	BinZOAs							
	BinZOA1	**BinZOA2**	**BinZOA3**	**BinZOA4**	**BinZOA5**	**BinZOA6**	**BinZOA7**	**BinZOA8**
KP8a	3924400	3924400	3924400	3924400	3924400	3924400	3924400	3924400
KP8b	3813669	3813669	3813669	3813669	3813669	3813669	3813669	3813669
KP8c	3347452	3347452	3347452	3347452	3347452	3347452	3347452	3347452
KP8d	4187707	4187707	4187707	4187707	4187707	4187707	4187707	4187707
KP8e	4955555	4955555	4955555	4955555	4955555	4955555	4955555	4955555
KP12a	5688887	5688562.85	5688887	5688887	5688887	5688887	5688887	5688887
KP12b	6498597	6494128.2	6498597	6498597	6496039.2	6498597	6498597	6498597
KP12c	5170626	5169962.5	5170626	5170626	5170341.75	5170531	5170531	5170341
KP12d	6992404	6992404	6992404	6992404	6992404	6992404	6992404	6992187.55
KP12e	5337472	5337472	5337472	5337472	5337472	5337472	5337472	5337472
KP16a	7847616.55	7831863.2	7849599.75	7848651.75	7842528.5	7842425.35	7837959.1	7838327.9
KP16b	9349338.5	9333960.55	9352734.9	9352734.9	9349075.4	9348927.5	9351299.8	9330321
KP16c	9146971.45	9131671.7	9149915.95	9151147	9151147	9142934.05	9140333.8	9128188.2
KP16d	9345517.7	9322908.6	9348889	9348279.1	9330459.85	9343219.05	9331810.35	9327349.65
KP16e	7767632.9	7742826.35	7769117	7768374.95	7760366.15	7759092.3	7762360.2	7748957.5
KP20a	10686806.1	10664753.6	10701196.8	10701630.8	10686675.9	10674475.4	10686793	10673749.2
KP20b	9766318.9	9727949.65	9785342.05	9784199.55	9767862.15	9766195.75	9759078.8	9741572.75
KP20c	10688738.9	10660474	10698762.8	10700678.4	10696073.4	10678195.6	10673260.2	10642738.8
KP20d	8866705.85	8854330.6	8905453.7	8898056.8	8892496.95	8897157.95	8886586.45	8869505.1
KP20e	9330475.15	9341014.25	9347862.2	9345035.95	9344210.55	9344013.45	9349280.2	9334471.65
KP24a	13446959.4	13421129.1	13454595.5	13465425.7	13440097.7	13449822.5	13450648.6	13420694
KP24b	12124082	12087269.4	12154805.4	12167023.7	12147839.8	12137243.8	12131228.2	12083826.5
KP24c	12364231.8	12346202.7	12377252	12366592.9	12387132.9	12371089.9	12367166.5	12350934
KP24d	11740335	11689040.8	11754520.2	11742266.2	11735554.4	11712917.4	11723511.5	11688010.1
KP24e	13850265.6	13809228.9	13867935.1	13855697.6	13859200.6	13845808.4	13840395.4	13798770.5
Sum	**206238765**	**205875936**	**206434751**	**206412564**	**206304648**	**206230192**	**206208386**	**205893684**

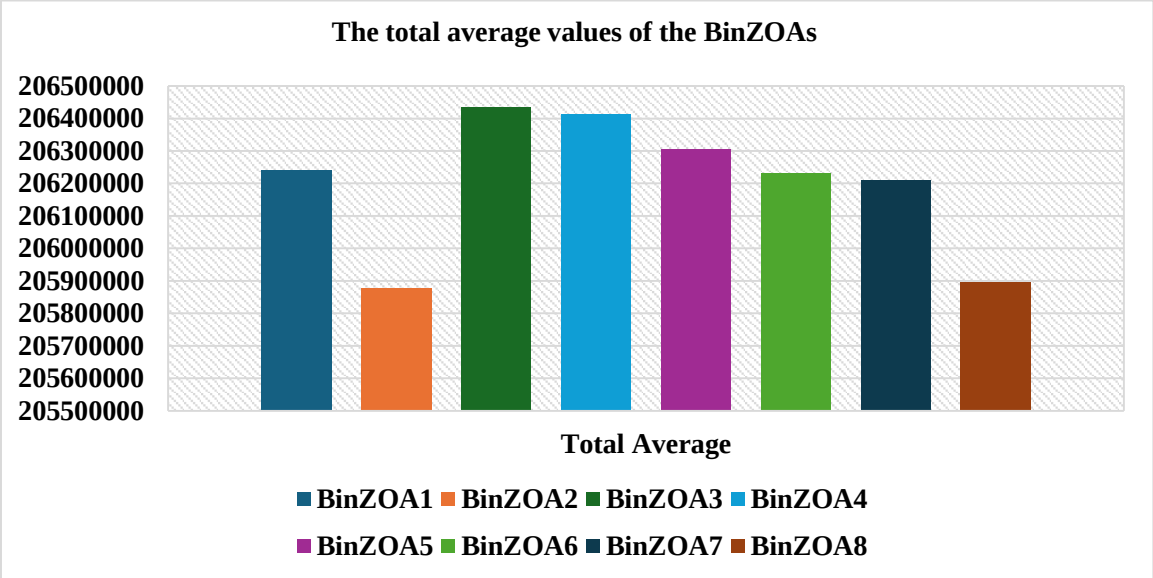

Figure 4. The total average values of the BinZOAs

Table 11. The std (standard deviation) values of the BinZOAs

Dataset ID	BinZOAs							
	BinZOA1	**BinZOA2**	**BinZOA3**	**BinZOA4**	**BinZOA5**	**BinZOA6**	**BinZOA7**	**BinZOA8**
KP8a	0	0	0	0	0	0	0	0
KP8b	0	0	0	0	0	0	0	0
KP8c	0	0	0	0	0	0	0	0
KP8d	0	0	0	0	0	0	0	0
KP8e	0	0	0	0	0	0	0	0
KP12a	0	1412.93709	0	0	0	0	0	0
KP12b	0	10885.3872	0	0	7673.4	0	0	0
KP12c	0	1697.62783	0	0	1239.01702	285	285	435.344691
KP12d	0	0	0	0	0	0	0	943.483676
KP12e	0	0	0	0	0	0	0	0
KP16a	8237.69619	13431.3654	6029.44696	7127.90044	10625.4944	10819.7235	16931.9286	13877.3787
KP16b	8552.30009	25910.5519	1146.82631	1146.82631	8516.53794	13082.3282	6300.06263	26528.8548
KP16c	10094.6701	25242.0973	5366.02254	0	0	15091.7997	13504.4154	22052.2938
KP16d	9802.67829	25855.4326	0	2658.49247	21931.6045	15219.7048	19890.1982	21940.8193
KP16e	4452.3	21031.2588	0	3234.52096	8973.72224	10778.6648	12808.924	11301.8041
KP20a	17778.278	36595.2108	19432.2886	18499.4068	29071.0963	26796.4538	24092.2228	28023.8157
KP20b	24177.2187	34983.0117	16194.1487	21805.4028	31269.1737	35564.9263	32761.2848	35025.6691
KP20c	20132.7521	47724.8536	18009.4277	15307.0962	34937.5227	47569.4449	37015.5593	41969.4675
KP20d	23161.9636	42937.2173	22410.674	25193.6663	31158.0592	26330.5672	26173.0942	36462.4885
KP20e	21703.7272	15219.8506	9071.11583	9181.85878	13629.1449	11213.235	9379.37643	22514.0158
KP24a	36811.1337	57268.2182	33442.5815	27009.9648	40294.0149	33741.854	43703.1956	45241.6676
KP24b	28023.4118	43952.7485	28616.836	32774.0732	32895.1119	37305.8602	36139.2825	45659.5061
KP24c	40843.9305	37589.922	24547.3036	18347.1511	39133.323	42686.6862	41352.1158	48535.8301
KP24d	33248.9368	41675.4436	26697.7303	27020.178	30474.5346	31220.4739	38734.3505	53740.6381
KP24e	31478.4299	44486.5164	28865.5576	24711.8633	41788.8606	51325.658	35810.0737	48833.9281
Total	**318499.427**	**527899.651**	**239829.96**	**234018.402**	**383610.618**	**409032.38**	**394881.084**	**503087.006**

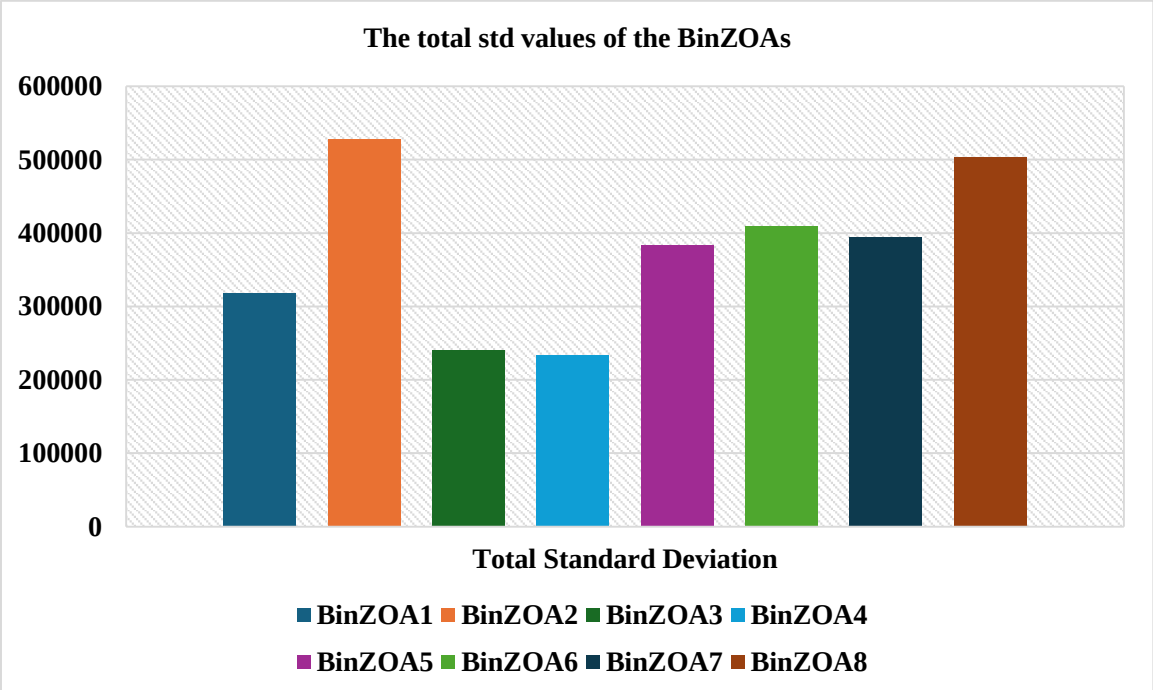

Figure 4. The comparasion of the total std values of the BinZOAs

Table 12. The *DMO* values of the BinZOAs

Dataset ID	BinZOAs							
	BinZOA1	**BinZOA2**	**BinZOA3**	**BinZOA4**	**BinZOA5**	**BinZOA6**	**BinZOA7**	**BinZOA8**
KP8a	0	0	0	0	0	0	0	0
KP8b	0	0	0	0	0	0	0	0
KP8c	0	0	0	0	0	0	0	0
KP8d	0	0	0	0	0	0	0	0
KP8e	0	0	0	0	0	0	0	0
KP12a	0	0.00569795	0	0	0	0	0	0
KP12b	0	0.06876561	0	0	0.03935926	0	0	0
KP12c	0	0.0128321	0	0	0.0054974	0.0018373	0.0018373	0.00551191
KP12d	0	0	0	0	0	0	0	0.0030955
KP12e	0	0	0	0	0	0	0	0
KP16a	0.0428	0.24353384	0.01761881	0.02969373	0.10768715	0.109001	0.16588878	0.16119128
KP16b	0.039126	0.20354383	0.002813	0.002813	0.04193949	0.0435208	0.01815675	0.24245702
KP16c	0.04563	0.21281813	0.01345241	0	0	0.08974777	0.11816224	0.2508844
KP16d	0.03606	0.27789826	0	0.00652377	0.19712663	0.06064838	0.18268107	0.23039476
KP16e	0.01910256	0.33839946	0	0.00955128	0.11263635	0.12903268	0.08696999	0.25948251
KP20a	0.37515397	0.58073194	0.24100011	0.23695426	0.37636726	0.28721123	0.37527562	0.40885951
KP20b	0.52903	0.47318809	0.3352829	0.34691938	0.51331748	0.53028994	0.60277681	0.49782126
KP20c	0.23599072	0.48632579	0.14243203	0.12455312	0.16753417	0.33439727	0.38046213	0.65188127
KP20d	0.55286	0.83798962	0.26544838	0.34828824	0.41055448	0.3583547	0.47674775	0.66804634
KP20e	0.29380146	0.18117981	0.10800207	0.1382036	0.14702389	0.14913012	0.09284921	0.25109455
KP24a	0.55005408	0.71219286	0.49357963	0.41348213	0.54706376	0.47510261	0.49796297	0.71955368
KP24b	0.43241473	0.61465592	0.48223373	0.54512763	0.59664281	0.46325604	0.48431097	0.71018667
KP24c	0.65202715	0.73042546	0.5474084	0.34040389	0.49520596	0.55949527	0.65559436	0.78598867
KP24d	0.59031117	0.61719533	0.51454278	0.4987394	0.48992596	0.57383385	0.51065138	0.78526173
KP24e	0.34189916	0.54806155	0.44959372	0.24529171	0.58032909	0.5690083	0.49532787	0.43412102
Total	**4.736261**	**7.14543553**	**3.61340796**	**3.28654515**	**4.82821116**	**4.73386725**	**5.14565518**	**7.06583207**

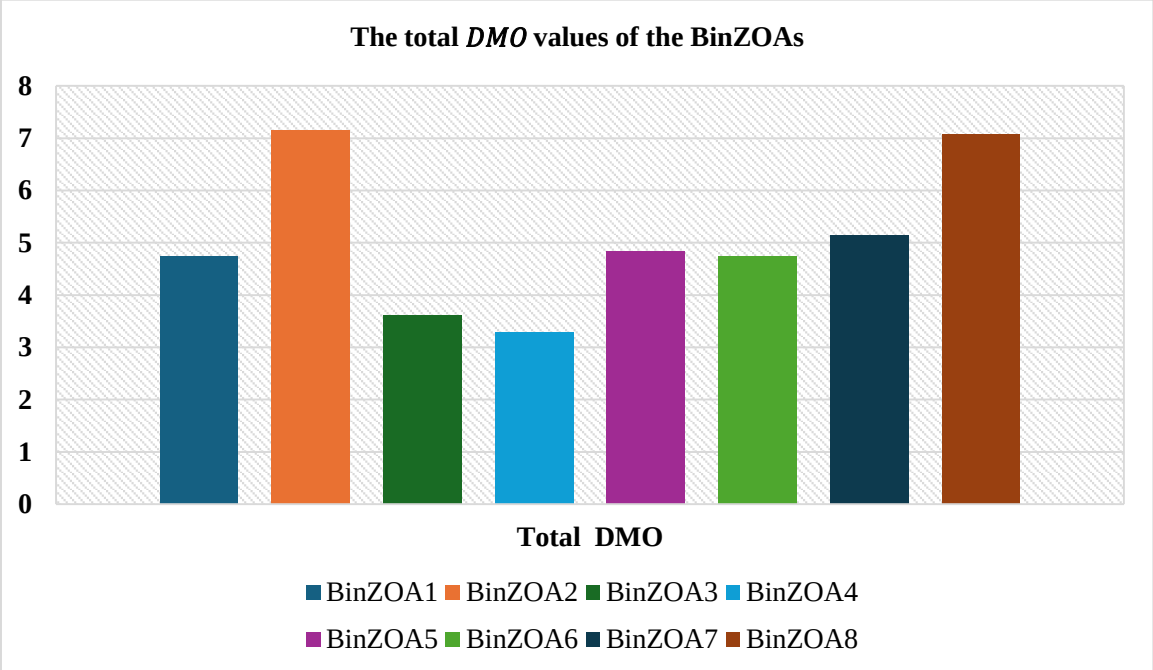

Figure 6. Comparasion of the total *DMO* values of the BinZOAs

Table 13. The wilcoxon sign test results on BinZOA4 with other BinZOAs

Dataset ID	BinZOA4 - BinZOA1	BinZOA4 - BinZOA2	BinZOA4 - BinZOA3	BinZOA4 - BinZOA5	BinZOA4 - BinZOA6	BinZOA4 - BinZOA7	BinZOA4 - BinZOA8
				***p* values**			
KP8a	1.00	1.00	1.00	1.00	1.00	1.00	1.00
KP8b	1.00	1.00	1.00	1.00	1.00	1.00	1.00
KP8c	1.00	1.00	1.00	1.00	1.00	1.00	1.00
KP8d	1.00	1.00	1.00	1.00	1.00	1.00	1.00
KP8e	1.00	1.00	1.00	1.00	1.00	1.00	1.00
KP12a	1.00	1.00	1.00	1.00	1.00	1.00	1.00
KP12b	1.00	0.25	1.00	0.5	1.00	1.00	1.00
KP12c	1.00	0.125	1.00	1.00	0.50	0.5	0.0313
KP12d	1.00	1.00	1.00	1.00	1.00	1.00	1.00
KP12e	1.00	1.00	1.00	1.00	1.00	1.00	1.00
KP16a	0.75	0.001	1.00	0.0723	0.015625	0.007813	0.0117
KP16b	0.15625	0.004883	1.00	0.09375	0.1875	0.75	0.00293
KP16c	0.5	0.01171875	1.00	1.00	0.0625	0.0156	0.000488
KP16d	0.25	0.0010	1.00	0.002	0.25	0.00195	0.000488
KP16e	0.5	0.0003	1.00	0.002	0.002	0.0625	0.00028
KP20a	0.040	0.0017	0.9199	0.1418	0.006	0.091	0.0043
KP20b	0.007	0.0002	0.4723	0.0086	0.04996	0.0052	0.000253
KP20c	0.100	0.0033	0.3486	0.836	0.2598	0.02352	0.00025
KP20d	0.004	0.0096	0.0936	0.856	0.5869	0.601	0.1625
KP20e	0.030	0.251	0.50148	0.6138	0.9405	0.234	0.049996
KP24a	0.279	0.0080	0.47813	0.156	0.41146	0.526	0.0206
KP24b	0.040	0.0032	0.50159	0.7652	0.4330	0.279	0.00034
KP24c	0.247	0.0176	0.91082	0.7089	0.3507	0.37	0.0442
KP24d	0.550	0.00089	0.19133	0.156	0.00405	0.09	0.00034
KP24e	0.0419	0.000163	0.21217	0.17896	0.0276	0.004	0.0001

Table 14. The wilcoxon sign test values on BinZOA4 with other BinZOAs

Dataset ID	BinZOA4 - BinZOA1	BinZOA4 - BinZOA2	BinZOA4 - BinZOA3	BinZOA4 - BinZOA5	BinZOA4 - BinZOA6	BinZOA4 - BinZOA7	BinZOA4 - BinZOA8
				***h* values**			
KP8a	0.00	0.00	0.00	0.00	0.00	0.00	0.00
KP8b	0.00	0.00	0.00	0.00	0.00	0.00	0.00
KP8c	0.00	0.00	0.00	0.00	0.00	0.00	0.00
KP8d	0.00	0.00	0.00	0.00	0.00	0.00	0.00
KP8e	0.00	0.00	0.00	0.00	0.00	0.00	0.00
KP12a	0.00	0.00	0.00	0.00	0.00	0.00	0.00
KP12b	0.00	0.00	0.00	0.00	0.00	0.00	0.00
KP12c	0.00	0.00	0.00	0.00	0.00	0.00	**1.00**
KP12d	0.00	0.00	0.00	0.00	0.00	0.00	0.00
KP12e	0.00	0.00	0.00	0.00	0.00	0.00	0.00
KP16a	0.00	**1.00**	0.00	0.00	**1.00**	**1.00**	**1.00**
KP16b	0.00	**1.00**	0.00	0.00	0.00	0.00	**1.00**
KP16c	0.00	**1.00**	0.00	0.00	0.00	**1.00**	**1.00**
KP16d	0.00	**1.00**	0.00	**1.00**	0.00	**1.00**	**1.00**
KP16e	0.00	**1.00**	0.00	**1.00**	**1.00**	0.00	**1.00**
KP20a	**1.00**	**1.00**	0.00	0.00	**1.00**	0.00	**1.00**
KP20b	**1.00**	**1.00**	0.00	**1.00**	**1.00**	**1.00**	**1.00**
KP20c	0.00	**1.00**	0.00	0.00	0.00	**1.00**	**1.00**
KP20d	**1.00**	**1.00**	0.00	0.00	0.00	0.00	0.00
KP20e	**1.00**	0.00	0.00	0.00	0.00	0.00	**1.00**
KP24a	0.00	**1.00**	0.00	0.00	0.00	0.00	**1.00**
KP24b	**1.00**	**1.00**	0.00	0.00	0.00	0.00	**1.00**
KP24c	0.00	**1.00**	0.00	0.00	0.00	0.00	**1.00**
KP24d	0.00	**1.00**	0.00	0.00	**1.00**	0.00	**1.00**
KP24e	**1.00**	**1.00**	0.00	0.00	**1.00**	**1.00**	**1.00**

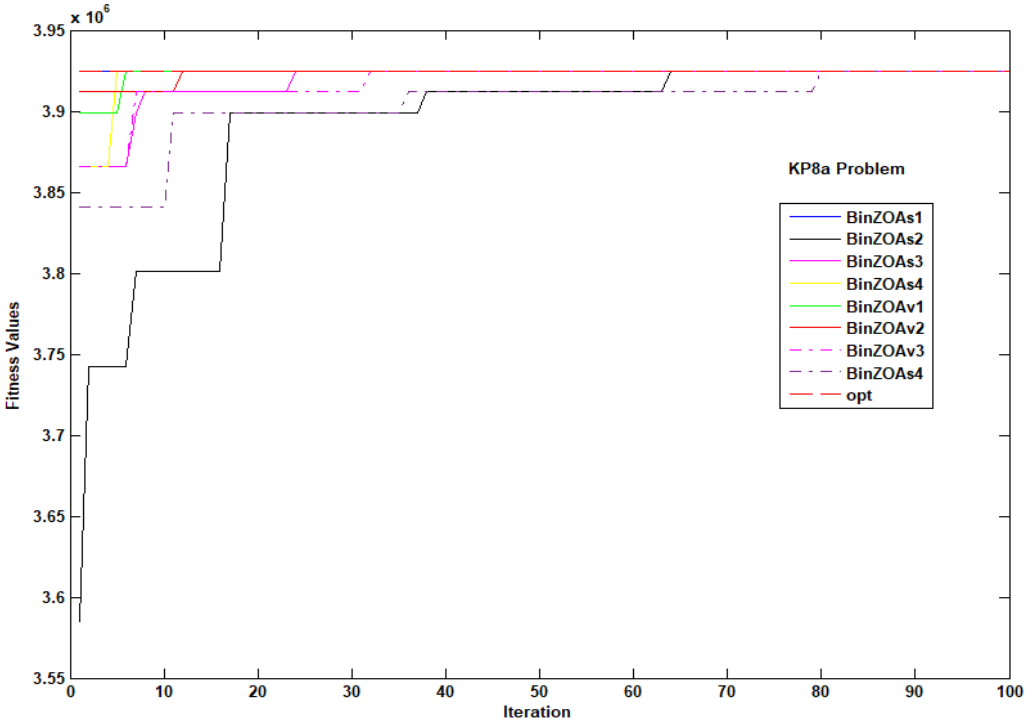

Figure 7. The convergence graph of BinZOAs on KP8a KP dataset

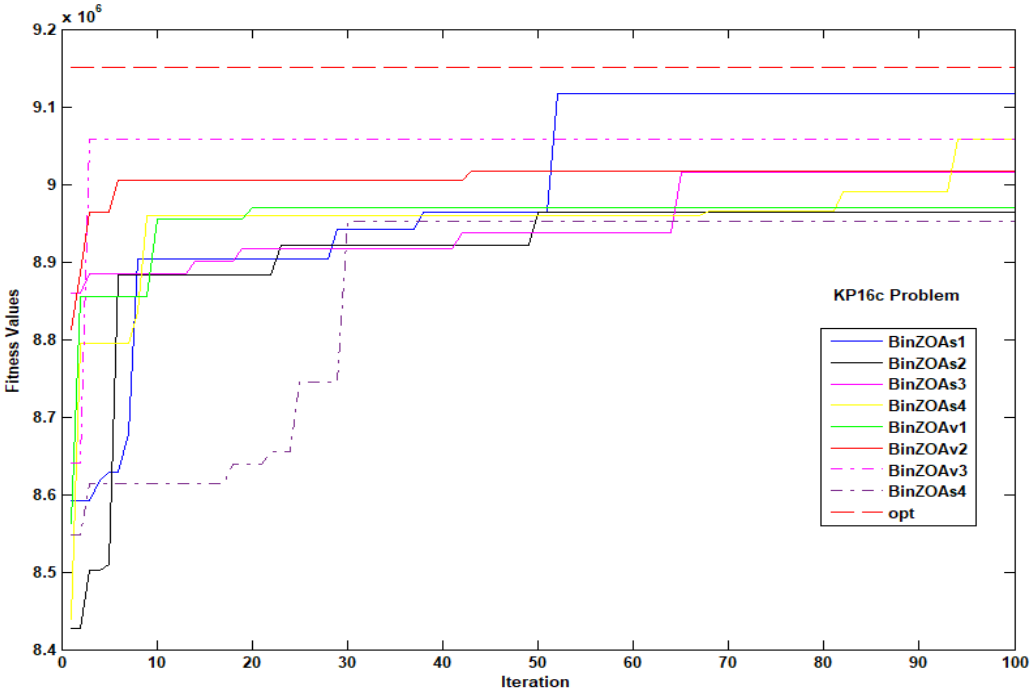

Figure 8. The convergence graph of BinZOAs on KP16c KP dataset

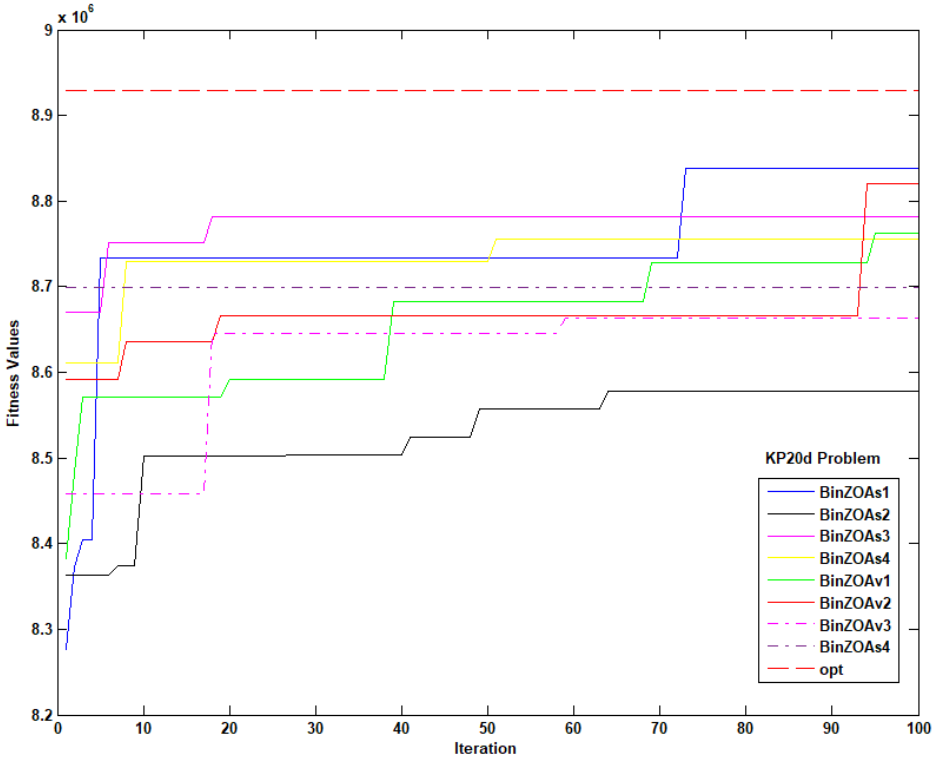

Figure 9. The convergence graph of BinZOAs on KP20d KP dataset

3.5. Literature Comparisons Of BinZOA

In this subsection, the most successful BinZOA variant based on DMO metrics (BinZOA4) is compared against several binary heuristic algorithms from the literature. Additionally, a new version called BinZOAX is developed by integrating the XOR logic gate method into BinZOA4. Both BinZOA4 and BinZOAX are benchmarked against FPS2 (Yang et al., 2014), WOS2 (Mirjalili and Lewis, 2016), BAS2 (Yang and He, 2013), PSOS2 (Yang et al., 2014), and BEOV3 (Abdel-Basset et al., 2021).

To ensure a fair comparison, all algorithms are run with a population size of 20 and a maximum of 5000 iterations, consistent with the settings used in (Abdel-Basset et al., 2021), where the benchmark results for the comparative algorithms are reported. Maintaining identical parameter settings is crucial for an unbiased evaluation. Specifically, the BEOV3 study employed these same parameters, and the FPS2, WOS2, BAS2, and PSOS2 algorithms are similarly executed under comparable configurations.

Performance is assessed using key statistical measures: best, worst, average, standard deviation (std), and DMO values. The detailed outcomes are summarized in Tables 15–19. The algorithms compared include:

Flower Pollination Algorithm with S2 transfer function (FPS2) (Yang et al., 2014).

Whale Optimization Algorithm with S2 transfer function (WOS2) (Mirjalili and Lewis, 2016).

Bat Algorithm with S2 transfer function (BAS2) (Yang and He, 2013).

Particle Swarm Optimization with S2 transfer function (PSOS2) (Kennedy and Eberhart, 1995; Yang et al., 2014).

Binary Equilibrium Optimization Algorithm with V3 transfer function (BEOV3) (Abdel-Basset et al., 2021).

Figure 10 illustrates the total DMO results for these seven algorithms in a column chart. According to the figure, BinZOAX demonstrated the highest overall success, while BinZOA4 was the least successful among the group. The BEOV3 algorithm ranked as the second-best performer after BinZOAX. This highlights that relying solely on transfer functions may not be sufficient for achieving optimal performance in binary optimization algorithms, emphasizing the need for enhancements through additional methods.

In this study, the incorporation of the XOR logic gate in BinZOAX significantly improved the algorithm's ability to explore and exploit the binary search space, leading to superior performance compared to the other heuristics. Thus, BinZOAX emerges as a highly effective approach for solving binary optimization problems.

Table 15. Comparisons on 8-dimensional KP datasets

Dataset ID	Opt.	Metaheuristics Alg.	Best	Worst	Average	Std	DMO	Rank
KP8a	3924400	*BinZOA4*	3924400	3924400	3924400	0.00	0.0000	1
		BinZOAX	3924400	3924400	3924400	0.00	0.0000	1
		BEOV3	3924400	3924400	3924400	0.00	0.0000	1
		WOS2	3924400	3924400	3924400	0.00	0.0000	1
		PSOS2	3924400	3924400	3924400	0.00	0.0000	1
		FPS2	3924400	3924400	3924400	0.00	0.0000	1
		BAS2	3924400	3924400	3924400	0.00	0.0000	1
KP8b	3813669	*BinZOA4*	3813669	3813669	3813669	0.00	0.0000	1
		BinZOAX	3813669	3813669	3813669	0.00	0.0000	1
		BEOV3	3813669	3813669	3813669	0.00	0.0000	1
		WOS2	3813669	3813669	3813669	13419	0.20	2
		PSOS2	3813669	3813669	3813669	0.00	0.0000	1
		FPS2	3813669	3813669	3813669	0.00	0.0000	1
		BAS2	3.813669	3.813669	3.813669	0.00	0.0000	1
KP8c	3347452	*BinZOA4*	3347452	3347452	3347452	0.00	0.0000	1
		BinZOAX	3347452	3347452	3347452	0.00	0.0000	1
		BEOV3	3347452	3347452	3347452	0.00	0.0000	1
		WOS2	3347452	3347452	3347452	0.00	0.0000	1
		PSOS2	3347452	3347452	3347452	0.00	0.0000	1
		FPS2	3347452	3347452	3347452	0.00	0.0000	1
		BAS2	3347452	3347452	3347452	0.00	0.0000	1
KP8d	4187707	*BinZOA4*	4187707	4187707	4187707	0.00	0.0000	1
		BinZOAX	4187707	4187707	4187707	0.00	0.0000	1
		BEOV3	4187707	4187707	4187707	0.00	0.0000	1
		WOS2	4187707	4187707	4187707	0.00	0.0000	1
		PSOS2	4187707	4187707	4187707	0.00	0.0000	1
		FPS2	4187707	4187707	4187707	0.00	0.0000	1
		BAS2	4187707	4187707	4187707	0.00	0.0000	1
KP8e	4955555	*BinZOA4*	4955555	4955555	4955555	0.00	0.0000	1
		BinZOAX	4955555	4955555	4955555	0.00	0.0000	1
		BEOV3	4955555	4955555	4955555	0.00	0.0000	1
		WOS2	4955555	4955555	4955555	0.00	0.0000	1
		PSOS2	4955555	4955555	4955555	0.00	0.0000	1
		FPS2	4955555	4955555	4955555	0.00	0.0000	1
		BAS2	4955555	4955555	4955555	0.00	0.0000	1

Table 16. Comparisons on 12-dimensional KP datasets

Dataset ID	Opt.	Metaheuristics Alg.	Best	Worst	Average	Std	DMO	Rank
KP12a	5688887	*BinZOA4*	5688887	5688887	5688887	0.00	0.000	1
		BinZOAX	5688887	5688887	5688887	0.00	0.0000	1
		BEOV3	5.688.887	3924400	3924400	0.00	0.0000	1
		WOS2	5688887	5683266	5665980	5062.05	0.01	2
		PSOS2	5688887	3924400	3924400	0.00	0.0000	1
		FPS2	5688887	3924400	3924400	0.00	0.0000	1
		BAS2	5688887	3924.400	3924400	0.00	0.0000	1
KP12b	6473019	*BinZOA4*	6498597	6498597	6498597	0.00	0.0000	1
		BinZOAX	6498597	6498597	6498597	0.00	0.0000	1
		BEOV3	6498597	3813669	3813669	0.00	0.0000	1
		WOS2	**6498597**	6494760	6473019	**9133.17**	0.06	2
		PSOS2	6498597	3813669	3813669	0.00	0.0000	1
		FPS2	6498597	3813669	3813669	0,00	0,0000	1
		BAS2	**6498597**	6488365	6473019	12530.6	0.15	3
KP12c	5170626	*BinZOA4*	5170626	5170626	5170626	0.00	0.0000	1
		BinZOAX	5170626	5170626	5170626	0.00	0.0000	1
		BEOV3	5170626	5170626	5170626	0.00	0.0000	1
		WOS2	5170626	5170626	5170626	0.00	0.0000	1
		PSOS2	5170626	5170626	5170626	0.00	0.0000	1
		FPS2	5170626	5170626	5170626	0.00	0.0000	1
		BAS2	5170626	5170626	5170626	0.00	0.0000	1
KP12d	6941564	*BinZOA4*	6992404	6992404	6992404	0.00	0.0000	1
		BinZOAX	6992404	6992404	6992404	0.00	0.0000	1
		BEOV3	6941564	6941564	6941564	0.00	0.0000	1
		WOS2	6941564	6941564	6941564	0.00	0.0000	1
		PSOS2	6941564	6941564	6941564	0.00	0.0000	1
		FPS2	6941564	6941564	6941564	0.00	0.0000	1
		BAS2	6992404	6992404	6992404	**23341.7**	0.29	2
KP12e	5337472	*BinZOA4*	5337472	5337472	5337472	0.00	0.0000	1
		BinZOAX	5337472	5337472	5337472	0.00	0.0000	1
		BEOV3	5337472	5337472	5337472	0.00	0.0000	1
		WOS2	5337472	5337472	5337472	0.00	0.0000	1
		PSOS2	5337472	5337472	5337472	0.00	0.0000	1
		FPS2	5337472	5337472	5337472	0.00	0.0000	1
		BAS2	5337472	5337472	5337472	0.00	0.0000	1

Table 17. Comparisons on 16-dimensional KP datasets

Dataset ID	Opt.	Metaheuristics Alg.	Best	Worst	Average	Std	DMO	Rank
KP16a	7850983	*BinZOA4*	7850983	7823318	7848651.75	7127.90044	0.02969373	2
		BinZOAX	7850983	7850983	7850983	0.00	0.0000	1
		BEOV3	7850983	7850983	7850983	0.00	0.0000	1
		WOS2	**7850983**	7832971	7832023	**4132.236**	0.22	3
		PSOS2	7850983	7850983	7850983	0.00	0.0000	1
		FPS2	7850983	7850983	7850983	0.00	0.0000	1
		BAS2	7850983	7850983	7850983	0.00	0.0000	1
KP16b	9352998	*BinZOA4*	9352998	9352998	9352734.9	1146.82631	0.002813	2
		BinZOAX	9352998	9352998	9352998	0.00	0.0000	1
		BEOV3	9.352.998	9352998	9352.998	0.00	0.0000	1
		WOS2	**9352998**	9334569.85	9213927	0.00	0.20	3
		PSOS2	9352998	9352998	9352998	42733.237	0.0000	1
		FPS2	9352998	9352998	9352998	0.00	0.0000	1
		BAS2	9352998	9352998	9352998	0.00	0.0000	1
KP16c	9151147	*BinZOA4*	9151147	9151147	9151147	0.00	0.0000	1
		BinZOAX	9151147	9151147	9151147	0.00	0.0000	1
		BEOV3	9151147	9151147	9151147	0.00	0.0000	1
		WOS2	9151147	9151147	9151147	0.00	0.0000	1
		PSOS2	9151147	9151147	9151147	0.00	0.0000	1

		FPS2	9151147	9151147	9151147	0.00	0.0000	1
		BAS2	9151147	9151147	9151147	0.00	0.0000	1
KP16d	9348889	*BinZOA4*	9348889	9336691	9348279.1	2658.49247	0.00652377	2
		BinZOAX	**9348889**	**9348889**	**9348889**	0.00	0.0000	1
		BEOV3	**9348889**	**9348889**	**9348889**	0.00	0.0000	1
		WOS2	**9348889**	**9348889**	**9348889**	**0.00**	0.12	4
		PSOS2	**9348889**	**9348889**	**9348889**	0.00	0.0000	1
		FPS2	**9348889**	**9348889**	**9348889**	0.00	0.0000	1
		BAS2	**9348889**	9347059.3	9.336.691	**4879.200**	0.019	3
KP16e	**7769117**	*BinZOA4*	7769117	7754276	7768374.95	3234.52096	0.00955128	2
		BinZOAX	7769117	7769117	7769117	0.00	0.0000	1
		BEOV3	7769117	7769117	7769117	0.00	0.0000	1
		WOS2	**7769117**	7761876.6	7691882	16054.133	0.022	3
		PSOS2	7769117	7769117	7769117	0.00	0.0000	1
		FPS2	7769117	7769117	7769117	0.00	0.0000	1
		BAS2	7769117	7769117	7769117	0.00	0.0000	1

Table 18. Comparisons on 20-dimensional KP datasets

Dataset ID	Opt.	Metaheuristics Alg.	Best	Worst	Average	Std	DMO	Rank
KP20a	**10727049**	*BinZOA4*	**10727049**	10659.097	10701630.8	18499.4068	0.23695426	3
		BinZOAX	**10727049**	**10727049**	**10727049**	0.00	0.0000	1
		BEOV3	**10727049**	**10727049**	**10727049**	0.00	0.0000	1
		WOS2	**10727049**	1071775.5	10674.314	15820.500	0.05	2
		PSOS2	**10727049**	**10.727049**	**10727049**	0.00	0.0000	1
		FPS2	**10727049**	**10.727049**	**10727049**	0.00	0.0000	1
		BAS2	**10727049**	**10.727049**	**10727049**	0.00	0.0000	1
KP20b	9818261	*BinZOA4*	9818261	9744578	9784199.55	21805.4028	0.34691938	2
		BinZOAX	9818261	9818261	9818261	0.00	0.0000	1
		BEOV3	9818261	9818261	9818261	0.00	0.0000	1
		WOS2	9818261	9818261	9818261	0.00	0.0000	1
		PSOS2	9818261	9818261	9818261	0.00	0.0000	1
		FPS2	9818261	9818261	9818261	0.00	0.0000	1
		BAS2	9818261	9818261	9818261	0.00	0.0000	1
KP20c	10714023	*BinZOA4*	10714023	10655671	10700678.4	15307.0962	0.12455312	3
		BinZOAX	10714023	10714023	10714023	0.00	0.0000	1
		BEOV3	10714023	10714023	10714023	0.00	0.0000	1
		WOS2	10712572	10707307.9	10687036	**7182.320**	0.62	4
		PSOS2	10714023	10714023	10714023	0.00	0.0000	1
		FPS2	10714023	10714023	10714023	0.00	0.0000	1
		BAS2	10.714023	10713732.8	10712572	**580.400**	0.0027	2
KP20d	8929156	*BinZOA4*	8929156	8865508	8898056.8	25193.6663	0.34828824	4
		BinZOAX	8929156	8929156	8929156	0.00	0.0000	1
		BEOV3	8929156	8929156	8929156	0.00	0.0000	1
		WOS2	8929156	8917460	8915396	**4913.303**	0.13	3
		PSOS2	8929156	8925028.0	8915396	**6305.624**	0.040	2
		FPS2	8929156	8929156	8929156	0.00	0.0000	1
		BAS2	8929156	8929156	8929156	0.00	0.0000	1
KP20e	9357969	*BinZOA4*	9357969	9330924	9345035.95	9181.85878	0.1382036	3
		BinZOAX	9357969	9357969	9357969	0.00	0.0000	1
		BEOV3	9357969	9357969	9357969	0.00	0.0000	1
		WOS2	9357969	9352636.4	9332165	8573.682	0.06	2
		PSOS2	9357969	9357969	9357969	0.00	0.0000	1
		FPS2	9357969	9357969	9357969	0.00	0.0000	1
		BAS2	9357969	9357969	9357969	0.00	0.0000	1

Table 19. Comparisons on 24-dimensional KP datasets

Dataset ID	Opt.	Metaheuristics Alg.	Best	Worst	Average	Std	DMO	Rank
KP24a	13549094	*BinZOA4*	13521334	13415.824	13465425.7	27009.9648	0.41348213	3
		BinZOAX	13549094	13549.094	13549094	0.00	0.0000	1
		BEOV3	13549094	13549.094	13549094	0.00	0.0000	1
		WOS2	**13549094**	9332165	13513046.15	8573.682	0.27	2
		PSOS2	13549094	13549094	13549094	0.00	0.0000	1
		FPS2	13549094	13549094	13549094	0.00	0.0000	1
		BAS2	13549094	13549094	13549094	0.00	0.0000	1
KP24b	12233713	*BinZOA4*	12233713	12108937	12167023.7	32774.0732	0.54512763	2
		BinZOAX	12233713	12233713	12233713	0.00	0.0000	1
		BEOV3	12233713	12233713	12233713	0.00	0.0000	1
		WOS2	12233713	12233713	12233713	0.00	0.0000	1
		PSOS2	12233713	12233713	12233713	0.00	0.0000	1
		FPS2	12233713	12233713	12233713	0.00	0.0000	1
		BAS2	12233713	12233713	12233713	0.00	0.0000	1
KP24c	12448780	*BinZOA4*	12408833	12334644	12366592.9	18347.1511	0.34040389	2
		BinZOAX	12448780	12448780	12448780	0.00	0.0000	1
		BEOV3	12448780	12448780	12448780	0.00	0.0000	1
		WOS2	12448780	12448780	12448780	**0.00**	0.0000	1
		PSOS2	12448780	12448780	12448780	0.00	0.0000	1
		FPS2	12448780	12448780	12448780	0.00	0.0000	1
		BAS2	12448780	12448780	12448780	0.00	0.0000	1
KP24d	11815315	*BinZOA4*	11801123	11684111	11742266.2	27020.178	0.4987394	6
		BinZOAX	**11815315**	**11815315**	**11815315**	0.00	0.0000	1
		BEOV3	**11815315**	11810051	11814367	2795.200	0.00361	2
		WOS2	11810051	11810051	11810051	0.00	0.044	5
		PSOS2	**11815315**	11810840.6	11810051	1879.624	0.037	4
		FPS2	**11815315**	11810840.6	11810051	1879.624	0.044	5
		BAS2	**11815315**	11811103.8	11810051	2105.600	0.035	3
KP24e	**13940099**	*BinZOA4*	13889768	13801423	13855697.6	24711.8633	0.24529171	3
		BinZOAX	**13940099**	**13940099**	**13940099**	0.00	0.0000	1
		BEOV3	**13940099**	**13940099**	**13940099**	0.00	0.0000	1
		WOS2	**13940099**	13931016.65	13894074	14162.599	0.07	2
		PSOS2	**13940099**	**13940099**	**13940099**	0.00	0.0000	1
		FPS2	**13940099**	**13940099**	**13940099**	0.00	0.0000	1
		BAS2	**13940099**	**13940099**	**13940099**	0.00	0.0000	1

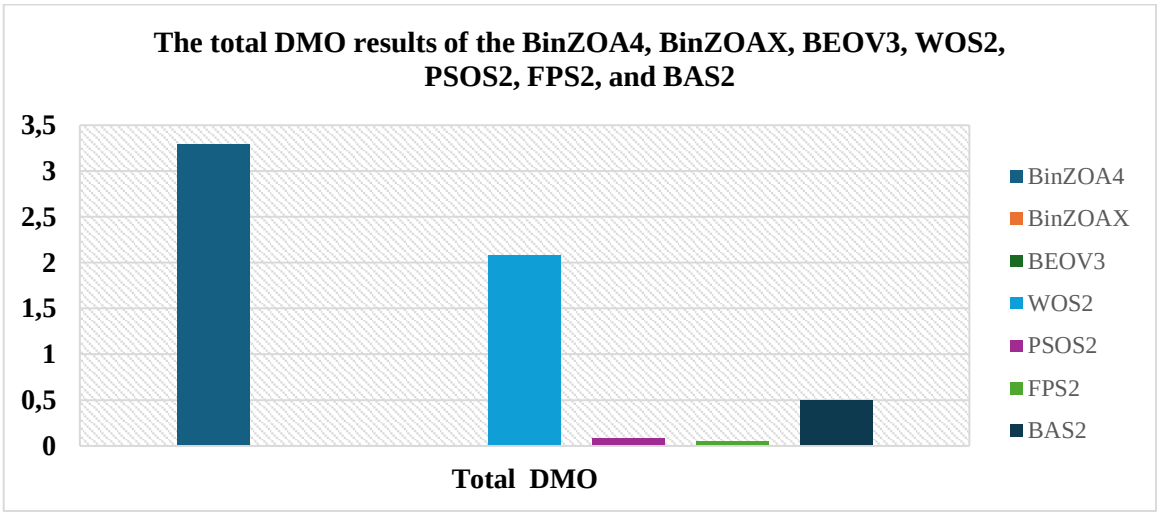

Figure 10. Comparison of seven metaheuristic algorithms with respect to total $DMOs$

4. Conclusions

This study focused on the Zebra Optimization Algorithm (ZOA), a heuristic method inspired by the natural behaviors of zebras. Originally designed for continuous optimization problems, ZOA is a relatively new algorithm with limited exploration and evaluation in existing literature. Literature reviews have shown that the success of ZOA on discrete problems has not been adequately analyzed. To fill this gap, ZOA is extended to binary optimization by incorporating eight different transfer functions comprising four S-shaped and four V-shaped functions variants resulting in eight distinct binary versions called BinZOA (BinZOA1 through BinZOA8). These variants are tested on 25 benchmark instances of the 0–1 knapsack problem, covering small to large scales. The 0–1 knapsack problem, a classic NP-hard problem commonly used for benchmarking binary optimization algorithms, involves selecting items to maximize total profit without exceeding a capacity constraint. Performance evaluation is based on multiple statistical indicators, including best, worst, mean, standard deviation (std), and DMO values. Among the BinZOA variants, BinZOA4 emerged as the most effective according to DMO metrics. To further assess its competitiveness, BinZOA4 is compared with several established binary heuristic algorithms from the literature namely BEOV3, WOS2, PSOS2, FPS2, and BAS2. The results indicated that relying solely on transfer functions is insufficient for achieving high performance, signaling the need for enhanced strategies. Consequently, the BinZOA4 variant is improved by integrating an XOR logic gate, leading to a new version named BinZOAX (using the S4 transfer function). When benchmarked against the comparison algorithms, BinZOAX consistently outperformed all others, achieving optimal solutions across all knapsack datasets. This demonstrates the effectiveness of XOR logic gates in strengthening binary optimization algorithms.

Overall, this study represents one of the first comprehensive applications of ZOA to binary optimization problems, accompanied by a detailed analysis of transfer functions and logic gate integration. The promising experimental results contribute valuable insights to the literature and establish BinZOAX as a competitive heuristic for binary optimization challenges.

4.1. Limitations Of The BinZOA Algorithm And Future Works

Although the ZOA algorithm has solved many different engineering design problems with its continuous version, there are still many limitations associated with the ZOA algorithm. For example, most continuous problems solved with ZOA are small in size. There is a need to solve larger problems with ZOA. Furthermore, in this study, the ZOA algorithm is configured to solve binary optimization problems, and a transfer function-based analysis is conducted. In this study, the success of S- and V-shaped transfer functions, commonly used in the literature, is tested on Binary ZOA. However, there are also newly introduced or more recently proven transfer functions in the literature, such as Taper-shaped, U-shaped, X-shaped, and so on. It is necessary to test the Binary ZOA algorithm with these transfer functions. This paper demonstrates that Binary ZOA cannot achieve sufficient success with transfer functions alone. XOR gates allow the Binary ZOA algorithm to better explore and exploit the binary

search space. The BinZOAX algorithm also achieved optimal performance on the 25 medium-sized KP datasets presented in this study. BinZOAX's success on just the 25 KP dataset is not sufficient. BinZOAX's success needs to be proven on KP datasets of different sizes. Furthermore, BinZOAX must also demonstrate its success on binary optimization problems other than the KP problem.

In future work, we plan to test the performance of BinZOA and BinZOAX on large-scale datasets for the KP problem, filling the gaps in this work. Furthermore, we will run the classical test functions BinZOA and BinZOAX on low, medium, and high-scale datasets to demonstrate the success of BinZOAX. In addition to S- and V-shaped transfer functions, we will investigate other transfer functions proposed in the literature, and examine BinZOA's performance on these functions.

5. Author Contributions Statement

In the conducted study, Yazar 1 and Yazar 2 contributed equally to the development of the idea, study design, literature review, data collection and analysis, evaluation of the obtained results, the writing process, and the content review of the manuscript.

6. Ethics Committee Approval and Conflict of Interest Statement

The study presented in this article does not involve any experimental research on humans or animals, clinical research on humans or animals, or any qualitative or quantitative research methods such as surveys, interviews, focus groups, observations, experiments, or interviews that require data collection from participants. Therefore, ethics committee approval was not required for the preparation of this manuscript. There is no conflict of interest between the authors and any person or institution regarding the preparation and publication of this manuscript.

References

Abdel-Basset M., Mohamed R., Mirjalili S. A binary equilibrium optimization algorithm for 0–1 knapsack problems. Computers & Industrial Engineering 2021; 151: 106946.

Bas E., Ihsan A. Gray Wolf and Krill Herd optimizations: Performance analysis and comparison. Pamukkale Universitesi Muhendislik Bilimleri Dergisi 2023; 29(7): 711–736.

Bas E., Guner LB. The binary crayfish optimization algorithm with bitwise operator and repair method for 0–1 knapsack problems: an improved model. Neural Computing Applications 2025; 37: 4733–4767.

Baş E. Binary Aquila optimizer for 0–1 knapsack problems. Engineering Applications of Artificial Intelligence 2023; 118: 105592.

Baş E. Feature selection problem via a novel binary chaotic zebra optimization. Presented at: Africa 5th International Conference on New Horizons in Science 2024; Apr 25–28, Cairo, Egypt.

Baş E., Baş Ş. An example of classification using a neural network trained by the zebra optimization algorithm. Sinop Universitesi Fen Bilimleri Dergisi 2024; 9(2): 388–420.

Baş E., Baş Ş. Uncapacitated facility location problem via binary zebra optimization algorithm based on S shaped transfer functions. Presented at: Africa 5th International Conference on New Horizons in Science, 2024 Apr 25–28, Cairo, Egypt.

Baş E., Yildizdan G. A new binary arithmetic optimization algorithm for uncapacitated facility location problem. Neural Computing Applications 2024; 36: 4151–4177.

Berberler ME., Güler A., Nuriyev U. A new genetic algorithm for the 0-1 knapsack problem. Academic Platform-Journal of Engineering and Science 2016; 4(3).

Detti P. A new upper bound for the multiple knapsack problem. Computers & Operations Research 2021; 129: 105210.

Elymany MM., Enany MA., Elsonbaty NA. Hybrid optimized-ANFIS based MPPT for hybrid microgrid using zebra optimization algorithm and artificial gorilla troops optimizer. Energy Conversion and Management 2024; 299: 117809.

Erol OK., Eksin I. A new optimization method: big bang–big crunch. Advances in Engineering Software 2006; 37(2): 106–111.

Ervural B., Hakli H. A binary reptile search algorithm based on transfer functions with a new stochastic repair method for 0–1 knapsack problems. Computers & Industrial Engineering 2023; 178: 109080.

Formato R. Central force optimization: a new metaheuristic with applications in applied electromagnetics. Progress in Electromagnetics Research 2007; 77: 425–491.

Ghadi YY., Neamah NM., Hossam-Eldin AA., Alqarni M., AboRas KM. State-of-the-art frequency control strategy based on an optimal fuzzy PI-FOPDF λ for SMES and UPFC integrated smart grids using zebra optimization algorithm. IEEE Access 2023; 11: 122893–122910.

Horowitz E., Sahni S., Rajasekaran S. Computer algorithms C++: C++ and pseudocode versions. New York: Computer Science Press; 1997.

Ibarra OH., Kim CE. Fast approximation algorithms for the knapsack and sum of subset problems. Journal of the ACM (JACM) 1975; 22(4): 463–468.

Ihsan A., Doğan N. Improved affine encryption algorithm for color images using LFSR and XOR encryption. Multimedya Tools Applications 2023; 82: 7621–7637.

Ihsan A., Doğan N. An innovative image encryption algorithm enhanced with the Pan-Tompkins algorithm for optimal security. Multimedya Tools Applications 2024; 83: 82589–82619.

Kaveh A., Ghazaan MI., Bakhshpoori T. An improved ray optimization algorithm for design of truss structures. Period Polytech Civil Engineering 2013; 57(2): 97–112.

Kennedy J., Eberhart R. Particle swarm optimization. In: Proc ICNN'95 - International Conference Neural Networks, 1995 Nov 27–Dec 1, Perth, Australia.

Kreher DL. Available from: [https://pages.mtu.edu/\~kreher/cages/Data.html](https://pages.mtu.edu/~kreher/cages/Data.html). (Accessed: 2025 Apr 18).

Kulkarni AJ., Shabir H. Solving 0–1 knapsack problem using cohort intelligence algorithm. International Journal Machine Learning and Cybernetics 2016; 7: 427–441.

Liu SH., Mernik M., Hrnčič D., Črepinšek M. A parameter control method of evolutionary algorithms using exploration and exploitation measures with a practical application for fitting Sovova's mass transfer model. Applied Soft Computing 2013; 13(9): 3792–3805.

Martello S., Toth P. Knapsack problems: Algorithms and Computer Implementations. New York: John Wiley & Sons; 1990.

Mirjalili S., Lewis A. The whale optimization algorithm. Advances in Engineering Software 2016; 95: 51–67.

Pazhanimuthu C., Saravanan G., Suresh KP. Performance analysis of voltage profile improvement in AVR system using zebra optimization algorithms based on PID controller. e-Prime-Advances in Electrical Engineering, Electronics and Energy 2023: 100380.

Raidl GR. An improved genetic algorithm for the multiconstrained 0-1 knapsack problem. In: Proc 1998 IEEE International Conference Evol Computation, 1998 May 4–9, Anchorage, AK, USA.

Rashedi E., Nezamabadi-Pour H., Saryazdi S. GSA: A gravitational search algorithm. Information Sciences 2009; 179(13): 2232–2248.

Rezoug A., Bader-El-Den M., Boughaci D. Guided genetic algorithm for the multidimensional knapsack problem. Memetic Computing 2018; 10: 29–42.

Sag T., Ihsan A. Particle swarm optimization with a new intensification strategy based on K-means. Pamukkale Universitesi Mühendislik Bilimleri Dergisi 2023; 29(3): 264–273.

Sahni S. Approximate algorithms for the 0/1 knapsack problem. Journal of the ACM (JACM) 1975; 22(1): 115–124.

Sadollah A., Eskandar H., Lee HM., Kim JH. Water cycle algorithm: A detailed standard code. SoftwareX 2016; 5: 37–43.

Shah-Hosseini H. Principal components analysis by the galaxy-based search algorithm: a novel metaheuristic for continuous optimization. International Journal of Computational Science and Engineering 2011; 6(1-2): 132–140.

Singh RP. Solving 0–1 knapsack problem using genetic algorithms. In 2011 IEEE 3rd international conference on communication software and networks 2011 May 27–29, Xi'an, China.

Smith GD., Steele NC., Albrecht RF., Cotta C., Troya JM. A hybrid genetic algorithm for the 0–1 multiple knapsack problem. In Artificial Neural Nets and Genetic Algorithms: Proceedings of the International Conference in Norwich, UK, 1997, Springer, Vienna.

Trojovská E., Dehghani M., Trojovský P. Zebra optimization algorithm: A new bio-inspired optimization algorithm for solving optimization algorithm. IEEE Access 2022; 10: 49445–49473.

Wei Z., Huang C., Wang X., Han T., Li Y. Nuclear reaction optimization: a novel and powerful physics-based algorithm for global optimization. IEEE Access 2019; 7: 66084–66109.

Yang XS., He X. Bat algorithm: literature review and applications. International Journal of Bio-inspired Computation 2013; 5(3): 141–149.

Yang XS., Karamanoglu M., He X. Flower pollination algorithm: a novel approach for multiobjective optimization. Engineering Optimization 2014; 46(9): 1222–1237.

Yao X. A new simulated annealing algorithm. International Journal of Computer Mathematics 1995; 56(3–4): 161–168.