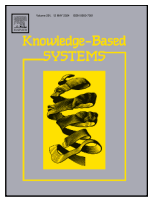
Knowledge-Based Systems 329 (2025) 114430

Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys



Enhanced binary particle swarm optimization for mitigating pandemic spread through passenger air traffic management

Gabriel A. Peña, Antonio Jiménez-Martín *, Alfonso Mateos

Decision Analysis and Statistics Group, Universidad Politécnica de Madrid, Campus de Montegancedo S/N, Boadilla del Monte, 28660, Madrid, Spain

ARTICLE INFO

Keywords:
Pandemic risk mitigation
Passenger air traffic management
Enhanced binary particle swarm optimization
Multi-objective optimization
Decision support system.

ABSTRACT

This study tackles a complex binary multi-objective optimization problem focused on minimizing the risk of pandemic importation through strategic passenger air traffic management. The approach involves determining whether international connections to destination airports within a specified country should be activated or deactivated over a defined time frame, considering epidemiological, economic, and socio-political impacts. We introduce a preliminary decision support system designed to assist decision-makers in the parametrization of the problem and quantify their preferences, thereby facilitating the derivation of a compromise solution via a binary particle swarm optimization (BPSO) metaheuristic. The standard BPSO is prone to particles getting trapped in local optima instead of searching for new solution and does not handle infeasible solutions properly. To overcome these inherent limitations, we propose an enhanced version of the BPSO metaheuristic. This enhanced algorithm incorporates novel mechanisms to promote solution space exploration and a robust strategy for managing infeasible solutions. A rigorous comparative analysis is conducted to evaluate the performance of the enhanced BPSO against both the original BPSO and several established state-of-the-art metaheuristics utilizing three benchmark datasets of a constrained problem. Finally, the effectiveness of the proposed enhanced metaheuristic is demonstrated in the context of the pandemic importation risk reduction problem, where it outperforms the original BPSO.

1. Introduction

Passenger air traffic constitutes a critical vector for the rapid global dissemination of infectious diseases during their initial phases [1,2], underscoring the urgent need to develop analytical frameworks and information and communication technology (ICT) tools for effective management. Such tools encompass early warning systems designed to detect pandemic emergence with minimal latency and immediate response strategies to either prevent or, mitigate disease propagation.

There is a large body of literature on modeling infectious diseases transmission via passenger air traffic. For instance, [3–5] explored early warning systems for pandemic detection. The impact of travel restrictions on the dissemination of COVID-19 was analyzed in [6], while [7] investigated the influence of international travel and border control measures on the spread of SARS-CoV-2. Furthermore, [8] introduced an efficient method for modeling disease transmission across transportation networks. Additionally, [9] studied the evolution of in-flight infection on aircraft, and [10] proposed a method to quantify virus transmission risk at airports using deep learning and computer vision techniques.

To curb pandemic spread, governmental authorities often implement restrictions on passenger air traffic and enforce border closures to mitigate disease transmission [6]. However, while potentially effective in slowing dissemination, an air traffic stoppage would engender substantial global economic and social repercussions. Moreover, pandemic incidence levels vary geographically, and regions with low epidemiological rates could potentially suffer disproportionate consequences from such measures. Consequently, intelligent decision-making at a granular level, specifically concerning the activation or deactivation of air connections between airports, is an imperative. This scenario is framed as a multi-objective optimization problem with conflicting objectives [11], where a balance needs to be struck between the minimization of the epidemiological risk of pandemic importation with the economic and social impacts of such decisions.

In [12], a web-based decision support system (DSS) termed RRPS-PAT was introduced to address the risk of pandemic spread through passenger air traffic management. This system facilitates the selective opening or closing of international connections to destination airports within a specific country over a defined period, considering epidemiological, economic, and socio-political impacts. RRPS-PAT integrates seven objective functions, five of which target economic goals, specifically the minimization of: the percentage of revenue loss in destination airport catchment areas (due to business and tourism declines), the

* Corresponding author.
E-mail address: antonio.jimenez@upm.es (A. Jiménez-Martín).

https://doi.org/10.1016/j.knosys.2025.114430
Received 5 May 2025; Received in revised form 19 August 2025; Accepted 4 September 2025
Available online 9 September 2025

standard deviation of revenue loss across catchment areas (to ensure equitable distribution of losses), the standard deviation of passenger losses per airline, the percentage of airport revenue loss due to aircraft non-landings (tax-related losses), and the standard deviation of airport revenue losses. Additionally, two social objectives are also considered: the minimization of the percentage of passengers missing flights, and the loss of connectivity within the passenger air traffic network. The epidemiological impact is treated as a constraint, where the optimal solution is required to achieve a predetermined percentage reduction in pandemic importation risk at destination airports. Socio-political decisions such as maintaining minimum traffic levels with certain regions (e.g. EU countries, capitals, or other global areas) during the analysis period, further restrict the solution space.

The Susceptible-Infectious-Recovered (SIR) model [13] is employed to characterize the dynamics of in-flight disease transmission within aircraft. This compartmental model relies on a set of epidemiological parameters–such as the transmission rate and the recovery rate–that must be specified for the pathogen under study, while also incorporating the contextual constraints of the environment in which it is applied, namely the aircraft cabin [9]. In contrast, the methodology presented in [10] addresses virus transmission risk in airport settings, leveraging deep learning and computer vision techniques to provide quantitative assessments.

RRPS-PAT leverages a Neo4J graph-oriented database, integrating flight data from Flightradar (sourced from the Spanish ATM Research, Development and Innovation Reference Center, CRIDA), aircraft occupancy data from the International Air Transport Association (IATA), global pandemic incidence data from Johns Hopkins University, and connectivity data from the Airports Council International (ACI). The system enables decision-makers (DMs) to parameterize the problem (e.g., time period, country, targeted risk reduction percentage, etc.); quantify preferences using ordinal information, processed using the sum reciprocal method [14] to derive a surrogate weight vector; solve the resulting optimization problem using the binary particle swarm optimization (BPSO) metaheuristic [15], visualize and export solutions.

Although the BPSO metaheuristic employed in RRPS-PAT outperforms the NSGA-III algorithm used in an earlier approach [11] in terms of execution time and solution quality, it exhibits limitations, notably in handling constrained problems and avoiding premature convergence to local optima. Specifically, the RRPS-PAT in [12] it took approximately five hours to solve a two-week scenario.

Alternative metaheuristics have outperformed BPSO in constrained problems such as the knapsack problem. Examples include global harmony search variants [16,17], inspired by musical improvisation; binary artificial fish swarm [18] by mimicking aquatic swarm behavior; the monkey algorithm [19], based on primate movement patterns; binary elephant herding optimization [20] by simulating inter-elephant interactions. Additionally, various differential evolution variants [21–25] and the novel binary reptile search algorithm (BinRSA) [26], which incorporates a repair/improvement method for infeasible solutions, have shown enhanced efficiency.

In this paper, we propose an enhanced BPSO metaheuristic featuring a robust method for managing infeasible solutions and a mechanism to enhance movement, which balances exploration and exploitation to prevent local optima entrapment. We conduct five comparative analyses of the enhanced BPSO against the original BPSO and also several well-known state-of-the-art metaheuristics using three benchmark datasets, demonstrating that the proposed method matches or improves upon the performance of the competing methods in all cases.

The paper is organized as follows. Section 2 details the binary particle swarm optimization (BPSO) metaheuristic utilized in RRPS-PAT and Section 3 proposes enhancements. Section 4 presents a comparative analysis of the enhanced BPSO against the original BPSO and established metaheuristics. Section 5 illustrates the application of the enhanced BPSO to the pandemic importation risk reduction problem, outperforming over the original BPSO. Finally, some conclusions and future research lines are outlined in Section 6.

2. Particle swarm optimization and binarization techniques

The particle swarm optimization (PSO) metaheuristic was initially proposed by [15]. It draws inspiration from the social behavior of bird flocks, where each individual navigates a spatial domain proximate to a leader, the individual nearest to a food source. This behavior is algorithmically modeled by interpreting each bird as a candidate solution and the leader as the best solution within the population. In PSO, each particle searches for the best solution for a problem in a D-dimensional space while moving at a velocity V at time t. The position update for particle i at dimension d and time $t+1$ is governed by the equation:

$$x_{id}(t+1) = x_{id}(t) + V_{id}(t+1). \tag{1}$$

The velocity V_{id} is computed as:

$$\begin{aligned} V_{id}(t+1) =& V_{id}(t) \cdot w + C_1 \times R_1 \times (Pbest_{id}(t) - x_{id}) \\ &+ C_2 \times R_2 \times (Gbest_d(t) - x_{id}), \end{aligned} \tag{2}$$

where C_1 and C_2 represent the cognitive and social acceleration coefficients, respectively, quantifying the influence of individual and collective decisions on particle movement; w represents the inertia weight, determining the carryover effect of the previous velocity; $Pbest_{id}$ refers to the best position obtained by particle i in dimension d; $Gbest_d$ is the global best position in dimension d; R_1 and R_2 are two random numbers in $[0, 1]$. The velocity in each dimension integrates the distance to both the particle's personal and global best, modulated by the prior velocity.

The parameters C_1 and C_2 dictate distinct behavioral dynamics: a predominance of C_1 over C_2 fosters individualistic exploration around $Pbest$, enhancing diversity, whereas, a predominance of C_2 establishes $Gbest$ as the main attractor of the particles over their $Pbest$, promoting exploitation. As delineated in Algorithm 1, PSO initializes a population, computes the initial $Pbest$ and $Gbest$ values and iterates until a stopping_criterion is met, updating particle position and velocities, and refining $Pbest$ and $Gbest$ iteratively.

Algorithm 1 Particle swarm optimization.

$t \leftarrow 1$
for particle $i \in [1 \ldots N]$ **do**
  Initialize particle $x_i(t)$ and evaluate fitness f($x_i(t)$))
  $Pbest_i(t) \leftarrow x_i(t)$
end for
$Gbest(t) \leftarrow \text{argmin}(f(x_i(t)))$, $i \in [1 \ldots N]$
while *stopping_condition* not met **do**
  for particle $i \in [1 \ldots N]$ **do**
    for dimension $d \in [1 \ldots D]$ **do**
      Compute $v_{id}(t+1)$ with Eq. (2)
      Compute $x_{id}(t+1)$ with Eq. (1)
    end for
    Evaluate fitness of $x_i(t+1)$
    if $f(x_i(t+1)) < f(Pbest_i(t))$ **then**
      $Pbest_i(t+1) \leftarrow x_i(t+1)$
    end if
    if $f(x_i(t+1)) < f(Gbest(t))$ **then**
      $Gbest(t+1) \leftarrow x_i(t+1)$
    end if
  end for
  $t \leftarrow t+1$
end while
return $Gbest$

Originally formulated for continuous search spaces, PSO was adapted for binary domains in the binary particle swarm optimization (BPSO) in [27], which utilizes a sigmoid function to determine the probability of bit flipping between 0 or 1, leveraging Eq. (2). Subsequent

research has proposed various transfer functions to enhance BPSO. The most common type of functions are the S-shaped function [28], including the sigmoid function, but many other types can be found in the literature, such as the V-shaped [28], linear [29,30], U-shaped [31], Z-shaped [32], X-shaped [33] and Taper-shaped [34] functions. Most variants maintain a fixed slope throughout optimization, yielding either extreme or moderate probability outputs, while dynamic approaches adjust the slope based on optimization progress, often varying with iteration count [35,36]. [37] introduces a fuzzy logic-based transfer function that integrates iteration count, population diversity and particle velocity to compute bit-flip probabilities.

Following probability computation, a binarization rule is applied, typically expressed as

$$x_{id}(t+1) = \begin{cases} 1, & \text{if rand} \leq \mathrm{T}(\mathrm{x}_{id}(t)) \\ 0, & \text{if rand} > \mathrm{T}(\mathrm{x}_{id}(t)) \end{cases}, \tag{3}$$

where $T(x_{id}(t))$ is the transfer function output. Despite these advancements, BPSO exhibits some limitations, such as the inverse relationship between velocity and exploration efficiency compared to continuous PSO, as pointed out in [38,39]. Additionally, looking at Eq. (3), we can conclude that the next value of $x_{id}(t+1)$ is computed using the velocity only and is independent of the previous value of the bit.

Premature convergence poses another challenge [40], with high velocities causing the transfer function and binarization rule to potentially flip all bits to 1 and low velocities to 0, which is exacerbated by velocity inflation over time.

Moreover, although transfer functions are generally effective at promoting exploration during the initial iterations, they often lack mechanisms to escape local optima or to mitigate premature convergence in later stages of the search. For example, time-varying functions [35] encourage broad exploration at the beginning through nearly horizontal profiles but soon transition to steep vertical shapes, which accelerate convergence without enabling sufficient local search around *Gbest*. Similarly, alternative schemes such as the fuzzy-based function proposed in [37] produce outputs that adapt to different aspects of the optimization process. However, this method presents a critical drawback: when particles approach positions very close to *Gbest*, the fuzzy rules and the equations governing particle dynamics restrict further movement. As a result, the algorithm's ability to continue exploration is significantly reduced, thereby increasing the likelihood of premature convergence.

Many real-world problems are constrained, necessitating robust constraint-handling strategies to navigate feasible regions. These strategies are categorized into three types [41]: penalty-based, repair-based and separatist-based methods. Penalty-based methods focus on transforming the constrained problem into a penalized unconstrained one by modifying the objective function according to the amount of violated constraint. Repair-based methods focus on converting the infeasible particles into feasible in each iteration. Finally, separatist-based methods handle the objective function and the constraint violation separately, by adding a constraint as a new objective function that needs to be minimized, by separating the population into particles with constraints and particles with potential solutions and combining solutions of both types, or by adding rules that give preference to solutions with the least amount of constraint violation when computing *Gbest* and *Pbest*.

3. Enhanced BPSO

In this section, we propose an enhanced BPSO (EBPSO) variant incorporating a novel transfer function to prevent entrapment in local optima and balance initial exploration with subsequent exploitation.

Firstly, we take the BPSO framework from [38], which addresses the velocity-bit issue with the original BPSO. Secondly, we dynamically adjust parameters w, C_1 and C_2 according to optimization phases in order to balance exploration and exploitation. Lastly, a novel transfer function is added, which maps continuous velocities to binary states while controlling population dispersion to promote movement in particles in case they fall in possible local optima.

The baseline BPSO from [38] employs two probability vectors, $\vec{V}^0_{id}$ and $\vec{V}^1_{id}$, to indicate the likelihood of a bit in dimension d of particle i flipping to 0 or 1, respectively, defined as

$$V^0_{id}(t+1) = w \times V^0_{id}(t) + d^0_{id,1}(t+1) + d^0_{id,2}(t+1) \tag{4}$$

$$V^1_{id}(t+1) = w \times V^1_{id}(t) + d^1_{id,1}(t+1) + d^1_{id,2}(t+1) \tag{5}$$

$$V^c_{id}(t+1) = \begin{cases} V^0_{id}(t+1), & \text{if } x_{id}(t) = 1 \\ V^1_{id}(t+1), & \text{if } x_{id}(t) = 0 \end{cases} \tag{6}$$

where $d^0_{id,1}(t+1)$ and $d^1_{id,1}(t+1)$ represent the cognitive component after calculating the tendency of particle i in dimension d to set its bit to 0 and 1 at time $t+1$, respectively; $d^0_{id,2}(t+1)$ and $d^1_{id,2}(t+1)$ represent the social component after calculating the tendency of particle i in dimension d to set its bit to 0 and 1 at time $t+1$, respectively. These components are computed based on $Pbest_{id}(t)$ and $Gbest_d(t)$, as follows:

$$d^0_{id,1}(t+1) = \begin{cases} -C_1 \times R_1, \text{if } Pbest_{id}(t) = 1 \\ C_1 \times R_1, \text{if } Pbest_{id}(t) = 0 \end{cases} \tag{7}$$

$$d^0_{id,2}(t+1) = \begin{cases} -C_2 \times R_2, \text{if } Gbest_d(t) = 1 \\ C_2 \times R_2, \text{if } Gbest_d(t) = 0 \end{cases} \tag{8}$$

$$d^1_{id,1}(t+1) = \begin{cases} C_1 \times R_1, \text{if } Pbest_{id}(t) = 1 \\ -C_1 \times R_1, \text{if } Pbest_{id}(t) = 0 \end{cases} \tag{9}$$

$$d^1_{id,2}(t+1) = \begin{cases} C_2 \times R_2, \text{if } Gbest_d(t) = 1 \\ -C_2 \times R_2, \text{if } Gbest_d(t) = 0 \end{cases} \tag{10}$$

The binarization rule applies the complement of the bit if $rand > T(x_{id})$:

$$x_{id}(t+1) = \begin{cases} \bar{x}_{id}(t), & \text{if rand} \leq \mathrm{T}(\mathrm{x}_{id}(t)) \\ x_{id}(t), & \text{if rand} > \mathrm{T}(\mathrm{x}_{id}(t)) \end{cases} \tag{11}$$

Unlike alternative versions [39,42], this BPSO allows dynamic adjustment of w, C_1 and C_2 facilitated by time-varying inertia weight [43] and acceleration coefficients [44]. Initially, high w and $C_1 > C_2$ emphasize exploration around *Pbest*, while, low w and $C_2 > C_1$ favor exploitation around *Gbest*. Parameters evolve linearly:

$$w = (w_{max} - w_{min})\frac{T-t}{T} + w_{min}, \tag{12}$$

$$C_1 = (C_{1fin} - C_{1ini})\frac{t}{T} + C_{1ini}, \tag{13}$$

$$C_2 = (C_{2fin} - C_{2ini})\frac{t}{T} + C_{2ini}, \tag{14}$$

where subscripts *max* and *min* denote the maximum and minimum value of the parameter w, and subscripts *ini* and *fin* denote the initial and final value of $C1$ and $C2$.

We propose a novel transfer function:

$$T(x_{id}(t)) = \frac{1}{1 + e^{-\alpha \times V^c_{id}(t)}}, \tag{15}$$

where slope α is defined as:

$$\alpha = \alpha_{min} + (\alpha_{max} - \alpha_{min}) \times (1 - \frac{H}{N}), \tag{16}$$

and H is the number of particles within a Hamming distance less than hd from *Gbest* and $\alpha_{\max} \geq 0$, $\alpha_{min} \geq 0$.

$\alpha > 0$ controls the steepness of the logistic curve. The derivative of T with respect to the velocity is given by

$$\frac{\partial T}{\partial V^c_{id}(t)} = \alpha T(x_{id}(t))(1 - T(x_{id}(t)))$$

This gradient determines the sensitivity of the transfer function to changes in velocity. For small α, the function exhibits a flat slope,

which smooths the mapping from velocity to probability and thereby promotes exploration by allowing a wider distribution of binary transitions. Conversely, for large α, the slope increases around the inflection point ($T = 0.5$), producing sharper transitions that emphasize exploitation by consolidating movements toward the attractor. Unlike conventional time-varying transfer functions, which monotonically reduce particle displacement and risk stagnation near G_{best}, the logistic response in Eq. ((15)) maintains a non-negligible gradient even when particles cluster, thus sustaining dispersive dynamics around local optima.

Note that if H equals N, then the transfer function exhibits its minimal gradient, and if H equals 0, the gradient reaches its maximum magnitude. *hd* is a parameter introduced by the user which specifies how sensitive the transfer function is to the distance of the particles to the *Gbest*. If *hd* equals 1 then the particles need overlap *Gbest* in order to reduce the value of α. As *hd* increases, however, particles can more easily penetrate the area where α is lower. A lower value of α results in a transfer function characterized by a more horizontal configuration. As a result of a more balanced probability distribution between $rand \leq T(x_{id}(t))$ and $rand > T(x_{id}(t))$, particles are more inclined to disperse around the convergence point as they seek an optimal solution. On the contrary, a high value of α makes the particles move towards their main attractor because the transfer function increases the likelihood of $rand \leq T(x_{id}(t))$ occurring over $rand > T(x_{id}(t))$.

The primary attractor of the particles may be either *Pbest* or *Gbest*, it is contingent upon the current phase of the metaheuristic, namely exploration or exploitation. During the exploration phase, particles exhibit greater individualism, preferentially following their personal best or *Pbest*. In the exploitation phase, the particles demonstrate an increased social behavior, aligning more closely with the global best or *Gbest*.

The behavior of the proposed transfer function mitigates the risk of particles becoming trapped in local optima. Specifically, when particles converge near their primary attractor, the function induces a dispersive effect that compels them to spread around this point and continue the search process. This contrasts with alternative transfer functions, such as time-varying schemes, which under similar conditions tend to restrict particle movement to negligible levels, thereby increasing the likelihood of stagnation.

There are four potential scenarios during metaheuristic execution, each managed differently as outlined in Table 1. In the exploration phase, it is imperative that particles are predominantly attracted to their *Pbest* than to *Gbest*, and their trajectories should be faster. Hence, $C_1 > C_2$, and the inertia weight should be set to a higher value. During the exploitation phase, the attraction should shift predominantly towards *Gbest* with reduced velocity in particle movement, thus, $C_2 > C_1$, and w must be set to a lower value.

The slope of the transfer function adapts based on particle spatial distribution, facilitating either their dispersion or convergence towards the primary attractor. The slope changes independently of the metaheuristic's operational phase, enabling precise guidance of particles according to the specific scenario encountered, thereby mitigating any potential biases.

Furthermore, we propose an enhanced generalization of the repair/improvement methodology outlined in [26] to handle feasible and infeasible solutions, shown in Algorithm 2.

The initial step involves calculating the ratio R_d, which is the quotient of the objective function value and the constraint values for each dimensional element (lines 1–3). Hence $f(x)_d$ represents the objective function's value when all bits in each dimension are set to zero, except for dimension d, which is set to one, and $c(x)_d$ denotes the corresponding constraint function value under the same conditions.

Subsequently, for each particle, the difference between the maximum constraint value of the problem ($MaxCons$) and the current constraint value associated with the solution ($c(x_i(t))$) is computed for each particle. A negative difference indicates an infeasible solution, as it exceeds the maximum allowable constraint, triggering the execution of the repair method (lines 6–14). On the other hand, if the solution is feasible, then the improvement method is applied (lines 15–33).

The repair methodology systematically eliminates elements from the solution set until feasibility is achieved. The selection of the element for removal is probabilistically determined by a parameter denoted as $SRate$, which is contingent upon the ratio R_d. Specifically, the element exhibiting the lowest R_d value is removed with a probability of $1 - SRate$, otherwise a randomly selected element is removed. This process is followed until the solution is feasible.

In contrast, the improvement methodology operates in an opposite manner to the repair process. It involves the incremental addition of el-

Table 1
Possible Scenarios and Metaheuristic Responses.

Scenario	w, C_1, C_2, α	Achieved behavior
Particles close together , exploration phase	Low α $C_1 > C_2$ High w	Particles scatter , explore areas around their *Pbest*
Particles far apart , exploration phase	High α $C_1 > C_2$ High w	Particles continue exploring areas around their *Pbest*
Particles close together , exploitation phase	Low α $C_2 > C_1$ Low w	Particles scatter , search near *Gbest*
Particles far apart , exploitation phase	High α $C_2 > C_1$ Low w	Particles continue searching near *Gbest*

Algorithm 2 Modified repair/improvement method.

1: **for** `dimension` $d \in$ `[1 ... D]` **do**
2: $R_d \leftarrow \frac{f(x)_d}{c(x)_d}$
3: **end for**
4: **for** `particle` $i \in$ `[1 ... N]` **do**
5: $diff \leftarrow MaxCons - c(x_i(t))$
6: **if** $diff < 0$ **then**
7: **while** $diff < 0$ **do**
8: **if** $rand \geq SRate$ **then** $a \leftarrow$ `Dimension with lowest` R_d
9: **else** $a \leftarrow$ `Random dimension`
10: **end if**
11: $x_{ia} \leftarrow 0$
12: $diff \leftarrow MaxCons - c(x_i(t))$
13: **end while**
14: **end if**
15: **if** $diff > 0$ **then**
16: $endloop \leftarrow false$
17: **while** $endloop == false$ **do**
18: **for** `dimension` $d \in$ `[1 ... D]` **do**
19: **if** $MaxCons - c(x_i(t)_{d=1}) \geq 0$ **then** $Candidates_d \leftarrow d$
20: **end if**
21: **end for**
22: **if** $length(Candidates) > 0$ **then**
23: **if** $rand \geq SRate$ **then**
24: $b \leftarrow$ `Dimension with largest` R_d `in` $Candidates$
25: **else**
26: $b \leftarrow$ `Random dimension in` $Candidates$
27: **end if**
28: $x_{ib} \leftarrow 1$
29: $diff \leftarrow MaxCons - c(x_i(t))$
30: **else** $endloop = true$
31: **end if**
32: **end while**
33: **end if**
34: `EvaluateFitness`$(x_i(t))$
35: **end for**

ements to the solution from a precomputed list of candidates, initiated at the commencement of a *while* loop, and continues until no further additions are possible without rendering the solution infeasible (lines 18–20), where $c(x_i(t)_{d=1})$ represents the constraint value if the d^{th} dimension of solution $x_i(t)$ were switched to a value of 1. An element from this candidate list is added with a probability of $1 - SRate$, or else a random element is selected for addition.

The criteria for element removal or addition may vary depending on the specific problem context. For instance, in a constrained optimization problem aimed at maximizing the value of selected elements according to a given objective function, the repair method will tend to remove elements with the lowest $f(x)_d/c(x)_d$ ratio. These elements minimally impact the objective function and are typically associated with elevated constraint values. In contrast, the improvement method prioritizes the addition of elements that maximize the positive contribution to the objective function. If the problem is to minimize the value of selected elements, then the repair method would target the element with the highest R_d for removal (line 8), while the improvement method would select the element with the lowest R_d for addition (line 24).

With these modifications, the resulting Enhanced BPSO algorithm is shown in Algorithm 3. After checking whether the stopping condition has been met, the metaheuristic updates C_1, C_2 and w using Eqs. (13), (14), and (12). Then, the position of each particle in every dimension is updated using Eqs. (15) and (16), following the velocity update with Eqs. (4), (5), and (6). The repair/improvement algorithm is applied to each particle once all its positions have been updated. The remainder of the metaheuristic proceeds as in a standard BPSO.

The code for EBPSO is available on GitHub at the following link: https://github.com/GabrielTRK/Visualization_Neo4j_Back_End, to facilitate reproducibility for other researchers.

Algorithm 3 Enhanced binary particle warm optimization.

1: $t \leftarrow 1$
2: **for** `particle` $i \in [1 \ldots N]$ **do**
3: `Initialize particle` $x_i(t)$ `and evaluate fitness` $f(x_i(t))$
4: $Pbest_i(t) \leftarrow x_i(t)$
5: **end for**
6: $Gbest(t) \leftarrow \text{argmin}(f(x_i(t)))$, $i \in [1 \ldots N]$
7: **while** *stopping_condition* not met **do**
8: `Update` C_1, C_2, w `with` Eqs. (13), (14), (12)
9: **for** `particle` $i \in [1 \ldots N]$ **do**
10: **for** `dimension` $d \in [1 \ldots D]$ **do**
11: `Compute` $V_{id}^0(t+1)$, $V_{id}^1(t+1)$, $V_{id}^c(t+1)$ with Eqs. (4), (5), (6)
12: `Compute` $x_{id}(t+1)$ with Eqs. (15), (16)
13: **end for**
14: `Evaluate fitness of` $x_i(t+1)$
15: `Repair/improve particle` $x_i(t+1)$ with Algorithm 2
16: **if** $f(x_i(t+1)) < f(Pbest_i(t))$ **then**
17: $Pbest_i(t+1) \leftarrow x_i(t+1)$
18: **end if**
19: **if** $f(x_i(t+1)) < f(Gbest(t))$ **then**
20: $Gbest(t+1) \leftarrow x_i(t+1)$
21: **end if**
22: **end for**
23: $t \leftarrow t+1$
24: **end while**
25: **return** $Gbest$

4. Comparative performance analysis

The efficacy of the proposed EBPSO variant has been evaluated against state-of-the-art metaheuristics using three benchmark datasets (BDs) representative of the well-known knapsack problem.

BD1 consists of 25 problems sourced from https://pages.mtu.edu/~kreher/cages/Data.html with dimensional ranges from 8 to 24. BD2 includes 20 problems from https://github.-com/whuph/KP_data with dimensions ranging from 100 to 1000. BD3 consists of 18 problems from http://artemisa.unicauca.edu.co/~johnyortega/instances_01_KP/ with dimensions spanning 100 to 5000.

A series of five comparative analyses were performed utilizing these datasets. Performance metrics varied by result source and included the mean and standard deviation of optimal values achieved across execution, the GAP metric defined as

$$GAP = \frac{opt - mean}{opt} \times 100,$$

the best identified solution, the hit value indicating the frequency of achieving the optimal solution, the Friedman rank of the evaluated metaheuristics, as well as a pairwise comparison between EBPSO and the other metaheuristics using the Wilcoxon signed-rank test.

The parameter setting for EBPSO were as follows: 10 particles, initial and final values for C_1 and C_2 set to 1.0, final and initial values for C_1 and C_2 set to 0.0, w_{max} is equal to 0.9 and w_{min} is equal to 0.0, α ranging from 1.0 to 5.0, hd is equal to 1 and $SRate$ probability is defined as follows:

$$SRate = \begin{cases} 0.5, & \text{if Dim} < 100 \\ 200 \times (Dim^{-1.45}), & \text{if Dim} \geq 100 \end{cases}, \quad (17)$$

as proposed in the initial version of the repair/improvement method in [26], where Dim denotes the problem's dimension.

This setting is aligned with the parameters used in the baseline BPSO configuration from [12], with the exception of novel parameters introduced by our enhancements.

The values of C_1, C_2 and w are proposed as the default values in the literature and are the most commonly used in applications [45]. The value of α_{min} is set to 1.0 because the transfer function with a slope of 1.0 produces the standard sigmoid function, which is well-known in the literature for working effectively with binary population-based metaheuristics. α_{max} is set to 5.0 because it produces a function that makes particles highly likely to follow their main attractor if their velocity in any dimension is approximately 1.0–a value reachable by the particles considering the values of C_1, C_2, w.

The parameter hd is set to 1, meaning that to adjust the slope of the transfer function, particles must completely overlap with $Gbest$ to achieve a Hamming distance of 0, which is less than hd. It would not make sense to have a lower hd because it is impossible to obtain a Hamming distance less than 0 with respect to $Gbest$. Conversely, a greater value of hd would hinder the metaheuristic's performance during the final portion of the optimization process, as it would prevent particles from effectively searching the areas closest to $Gbest$.

In the first comparison, utilizing BD1, the metaheuristics evaluated included two BPSO variants [15,30], two binary harmony search variants [16,17], binary artificial fish swarm algorithm (AFSA variant) [18], monkey algorithm (MA) [19], elephant herding optimization (EHO) [20], and the novel binary reptile search (BinRSA) [26]. The parameters were standardized across all metaheuristics, with 50 runs using a number of function evaluations of 1000 for problems *ks_8a* to *ks_12e* and 5000 for the remaining.

Results are detailed in Table 2 and sourced from [19,20,26], except for EBPSO. The quality measures originally employed in the above papers were the mean of the optimal values reached over the 50 runs and the GAP, supplemented by the Friedman rank and a final ranking. Optimal solutions achieved by any metaheuristic are highlighted in bold. To further demonstrate the efficiency of EBPSO, we performed a Wilcoxon signed-rank test between EBPSO and the other metaheuristics. The p-value associated with each test was included in Table 2.

We can conclude that EBPSO performed better, ranking first and consistently achieving optimal values across all problems, including challenging cases like *ks_12a*, *ks_20c*, *ks_20d*, *ks_20e* and *ks_24d*, whose optimum values are not reached by the second-best ranked metaheuristic,

Table 2
Results of comparative analysis 1.

Problem		BPSO	MBPSO	NGHS	DGHS	S-bAFSA	CGMA	BinEHO	BinRSA	EBPSO
ks_8a	Mean	3,921,857.19	**3,924,400.00**	**3,924,400.00**	**3,924,400.00**	**3,924,400.00**	**3,924,400.00**	**3,924,400.00**	**3,924,400.00**	**3,924,400.00**
	GAP%	6.48E-02	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
ks_8b	Mean	3,807,911.86	**3,813,669.00**	**3,813,669.00**	**3,813,669.00**	**3,813,669.00**	**3,813,669.00**	**3,813,669.00**	**3,813,669.00**	**3,813,669.00**
	GAP%	1.51E-01	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
ks_8c	Mean	3,328,608.71	**3,347,452.00**	**3,347,452.00**	**3,347,452.00**	**3,347,452.00**	**3,347,452.00**	**3,347,452.00**	**3,347,452.00**	**3,347,452.00**
	GAP%	5.63E-01	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
ks_8d	Mean	4,186,088.27	**4,187,707.00**	**4,187,707.00**	**4,187,707.00**	**4,187,707.00**	**4,187,707.00**	**4,187,707.00**	**4,187,707.00**	**4,187,707.00**
	GAP%	3.87E-02	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
ks_8e	Mean	4,932,737.28	4,954,571.72	**4,955,555.00**	**4,955,555.00**	**4,955,555.00**	**4,955,555.00**	**4,955,555.00**	**4,955,555.00**	**4,955,555.00**
	GAP%	4.60E-01	1.98E-02	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
ks_12a	Mean	5,683,694.29	5,688,552.41	5,687,724.44	5,687,460.74	5,688,498.02	**5,688,887.00**	5,688,757.34	5,688,757.34	**5,688,887.00**
	GAP%	9.13E-02	5.88E-03	2.04E-02	2.51E-02	6.84E-03	**0.00E+00**	2.28E-03	2.28E-03	**0.00E+00**
ks_12b	Mean	6,478,582.96	6,493,130.57	6,486,450.92	**6,498,597.00**	**6,498,597.00**	**6,498,597.00**	**6,498,597.00**	**6,498,597.00**	**6,498,597.00**
	GAP%	3.08E-01	8.41E-02	1.87E-01	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
ks_12c	Mean	5,166,957.08	5,170,493.30	5,165,079.86	5,165,079.80	**5,170,626.00**	**5,170,626.00**	**5,170,626.00**	**5,170,626.00**	**5,170,626.00**
	GAP%	7.10E-02	2.57E-03	1.07E-01	1.07E-01	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
ks_12d	Mean	6,989,842.73	6,992,144.26	6,991,971.10	**6,992,404.00**	**6,992,404.00**	**6,992,404.00**	**6,992,404.00**	**6,992,404.00**	**6,992,404.00**
	GAP%	3.66E-02	3.71E-03	6.19E-03	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
ks_12e	Mean	5,316,879.59	**5,337,472.00**	5,273,974.06	5,335,555.92	**5,337,472.00**	**5,337,472.00**	**5,337,472.00**	**5,337,472.00**	**5,337,472.00**
	GAP%	3.86E-01	**0.00E+00**	1.19E+00	3.59E-02	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
ks_16a	Mean	7,834,900.26	7,843,073.29	7,795,177.56	**7,850,983.00**	**7,850,983.00**	**7,850,983.00**	7,850,224.60	**7,850,983.00**	**7,850,983.00**
	GAP%	2.05E-01	1.01E-01	7.11E-01	**0.00E+00**	**0.00E+00**	**0.00E+00**	9.66E-03	**0.00E+00**	**0.00E+00**
ks_16b	Mean	9,334,408.62	9,350,353.39	9,253,153.36	**9,352,998.00**	**9,352,998.00**	**9,352,998.00**	**9,352,998.00**	**9,352,998.00**	**9,352,998.00**
	GAP%	1.99E-01	2.83E-02	1.07E+00	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
ks_16c	Mean	9,118,837.47	9,144,118.38	9,055,853.46	**9,151,147.00**	**9,151,147.00**	**9,151,147.00**	**9,151,147.00**	**9,151,147.00**	**9,151,147.00**
	GAP%	3.53E-01	7.68E-02	1.04E+00	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
ks_16d	Mean	9,321,705.87	9,337,915.64	9,310,913.74	9,316,797.66	9,321,195.66	9,345,229.60	9,345,961.48	**9,348,889.00**	**9,348,889.00**
	GAP%	2.91E-01	1.17E-01	4.06E-01	3.43E-01	2.96E-01	3.91E-02	3.13E-02	**0.00E+00**	**0.00E+00**
ks_16e	Mean	7,758,572.21	7,764,131.81	7,738,799.78	7,748,285.68	7,761,387.72	7,767,929.72	7,766,509.36	**7,769,117.00**	**7,769,117.00**
	GAP%	1.36E-01	6.42E-02	3.90E-01	2.68E-01	9.95E-02	1.53E-02	3.36E-02	**0.00E+00**	**0.00E+00**
ks_20a	Mean	10,707,360.91	10,720,314.03	10,592,304.82	**10,727,049.00**	**10,727,049.00**	**10,727,049.00**	**10,727,049.00**	**10,727,049.00**	**10,727,049.00**
	GAP%	1.84E-01	6.28E-02	1.26E+00	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
ks_20b	Mean	9,791,306.65	9,805,480.48	9,728,979.66	**9,818,261.00**	**9,818,261.00**	**9,818,261.00**	**9,818,261.00**	**9,818,261.00**	**9,818,261.00**
	GAP%	2.75E-01	1.30E-01	9.09E-01	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
ks_20c	Mean	10,703,423.34	10,710,947.05	10,585,114.14	10,709,663.64	10,710,831.94	10,712,553.74	10,713,587.70	10,713,993.98	**10,714,023.00**
	GAP%	9.89E-02	2.87E-02	1.20E+00	4.07E-02	2.98E-02	1.37E-02	4.06E-03	2.71E-05	**0.00E+00**
ks_20d	Mean	8,910,152.57	8,923,712.21	8,859,297.78	8,916,496.80	8,917,872.80	8,918,423.20	**8,929,156.00**	8,928,880.80	**8,929,156.00**
	GAP%	2.13E-01	6.10E-02	7.82E-01	1.42E-01	1.26E-01	1.20E-01	**0.00E+00**	3.08E-03	**0.00E+00**
ks_20e	Mean	9,349,546.98	9,355,930.35	9,324,698.84	9,357,518.34	9,357,331.86	9,357,922.38	9,357,751.44	9,357,953.46	**9,357,969.00**
	GAP%	9.00E-02	2.18E-02	3.56E-01	4.82E-03	6.81E-03	4.98E-05	2.32E-03	1.66E-05	**0.00E+00**
ks_24a	Mean	13,510,432.96	13,532,060.07	13,508,995.14	13,527,887.56	13,524,874.30	13,538,097.18	13,546,986.14	**13,549,094.00**	**13,549,094.00**
	GAP%	2.85E-01	1.26E-01	2.96E-01	1.57E-01	1.79E-01	8.12E-02	1.56E-02	**0.00E+00**	**0.00E+00**
ks_24b	Mean	12,205,346.16	12,223,442.61	12,160,953.58	**12,233,713.00**	**12,233,713.00**	**12,233,713.00**	**12,233,713.00**	**12,233,713.00**	**12,233,713.00**
	GAP%	2.32E-01	8.40E-02	5.95E-01	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
ks_24c	Mean	12,427,880.56	12,443,349.03	12,424,467.44	12,448,158.86	12,448,550.28	12,448,618.30	**12,448,780.00**	**12,448,780.00**	**12,448,780.00**
	GAP%	1.68E-01	4.36E-02	1.95E-01	4.99E-03	1.85E-03	1.30E-03	**0.00E+00**	**0.00E+00**	**0.00E+00**
ks_24d	Mean	11,792,064.76	11,803,712.38	11,736,314.98	11,810,261.56	11,810,893.24	11,808,614.90	11,810,682.68	11,814,367.48	**11,815,315.00**
	GAP%	1.97E-01	9.82E-02	6.69E-01	4.28E-02	3.74E-02	5.67E-02	3.92E-02	8.02E-03	**0.00E+00**
ks_24e	Mean	13,922,797.55	13,932,526.16	13,827,901.40	**13,940,099.00**	**13,940,099.00**	**13,940,099.00**	**13,940,099.00**	**13,940,099.00**	**13,940,099.00**
	GAP%	1.24E-01	5.43E-02	8.05E-01	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
Friedman Rank		8.00	6.74	7.16	5.12	4.46	3.76	3.74	3.14	**2.88**
Final Rank		9	7	8	6	5	4	3	2	**1**
Wilcoxon signed-rank	*p-value*	0.000016	0.000121	0.000175	0.002186	0.007650	0.011519	0.0118327	0.043163	-

Table 3
Results of comparative analysis 2.

Problem		BHHA	BSSA	BSCA	BWOA	BMPA	BinRSA	EBPSO
ks_8b	Mean	**3,813,669.00**	**3,813,669.00**	**3,813,669.00**	3,805,921.00	**3,813,669.00**	**3,813,669.00**	**3,813,669.00**
	GAP%	**0.00E+00**	**0.00E+00**	**0.00E+00**	2.03E-01	**0.00E+00**	**0.00E+00**	**0.00E+00**
	Hit	**20**	**20**	**20**	15	**20**	**20**	**20**
ks_12b	Mean	**6,498,597.00**	**6,498,597.00**	**6,498,597.00**	6,494,760.30	**6,498,597.00**	**6,498,597.00**	**6,498,597.00**
	GAP	**0.00E+00**	**0.00E+00**	**0.00E+00**	5.90E-02	**0.00E+00**	**0.00E+00**	**0.00E+00**
	Hit	**20**	**20**	**20**	17	**20**	**20**	**20**
ks_16a	Mean	**7,850,983.00**	7,850,035.00	**7,850,983.00**	7,845,292.00	**7,850,983.00**	**7,850,983.00**	**7,850,983.00**
	GAP	**0.00E+00**	1.21E-02	**0.00E+00**	7.25E-02	**0.00E+00**	**0.00E+00**	**0.00E+00**
	Hit	**20**	19	**20**	14	**20**	**20**	**20**
ks_16d	Mean	**9,348,889.00**	**9,348,889.00**	**9,348,889.00**	9,345,229.60	**9,348,889.00**	**9,348,889.00**	**9,348,889.00**
	GAP	**0.00E+00**	**0.00E+00**	**0.00E+00**	3.91E-02	**0.00E+00**	**0.00E+00**	**0.00E+00**
	Hit	**20**	**20**	**20**	14	**20**	**20**	**20**
ks_16e	Mean	**7,769,117.00**	**7,769,117.00**	**7,769,117.00**	7,768,185.70	**7,769,117.00**	**7,769,117.00**	**7,769,117.00**
	GAP	**0.00E+00**	**0.00E+00**	**0.00E+00**	1.20E-02	**0.00E+00**	**0.00E+00**	**0.00E+00**
	Hit	**20**	**20**	**20**	19	**20**	**20**	**20**
ks_20c	Mean	**10,714,023.00**	10,713,660.25	10,713,370.05	10,712,934.75	**10,714,023.00**	**10,714,023.00**	**10,714,023.00**
	GAP	**0.00E+00**	3.39E-03	6.09E-03	1.02E-02	**0.00E+00**	**0.00E+00**	**0.00E+00**
	Hit	**20**	15	10	5	**20**	**20**	**20**
ks_20d	Mean	**8,929,156.00**	**8,929,156.00**	8,927,780.00	8,922,964.00	**8,929,156.00**	**8,929,156.00**	**8,929,156.00**
	GAP	**0.00E+00**	**0.00E+00**	1.54E-02	6.93E-02	**0.00E+00**	**0.00E+00**	**0.00E+00**
	Hit	**20**	**20**	18	11	**20**	**20**	**20**
ks_20e	Mean	**9,357,969.00**	**9,357,969.00**	**9,357,969.00**	9,357,774.75	**9,357,969.00**	**9,357,969.00**	**9,357,969.00**
	GAP	**0.00E+00**	**0.00E+00**	**0.00E+00**	2.08E-03	**0.00E+00**	**0.00E+00**	**0.00E+00**
	Hit	**20**	**20**	**20**	15	**20**	**20**	**20**
ks_24a	Mean	**13,549,094.00**	13,547,856.30	**13,549,094.00**	**13,549,094.00**	**13,549,094.00**	**13,549,094.00**	**13,549,094.00**
	GAP	**0.00E+00**	9.13E-03	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
	Hit	**20**	19	**20**	**20**	**20**	**20**	**20**
ks_24d	Mean	11,810,840.60	11,810,577.40	11,810,314.60	11,810,577.40	11,811,630.20	**11,815,315.00**	**11,815,315.00**
	GAP	3.79E-02	4.01E-02	4.23E-02	4.01E-02	3.12E-02	**0.00E+00**	**0.00E+00**
	Hit	3	2	1	2	6	**20**	**20**
Friedman Rank		3.35	4.40	4.30	6.50	3.25	**3.10**	**3.10**
Final Rank		3	5	4	6	2	**1**	**1**
Wilcoxon signed-rank	*p-value*	0.317310	0.047188	0.084533	0.005793	0.317310	-	-

BinRSA. The GAP value is 0 for EBPSO for all the problems, indicating consistent attainment of the true optimum. Its Friedman rank, which is the lowest out of the metaheuristics considered with a value of 2.88, confirms its statistical superiority. Finally, a *p*-value lower than 0.05 in the Wilcoxon signed-rank test for each pairwise comparison provides strong evidence against the null hypothesis, indicating that the results obtained by EBPSO differ significantly from those of the other metaheuristics.

BD1 is also used in the second comparison, but the metaheuristics considered were binary harris hawk algorithm (BHHA) [46], binary salp swarm algorithm (BSSA) [47], binary sine cosine algorithm (BSCA) [48], binary whale optimization algorithm (BWOA) [49], binary marine predators optimization algorithm (BMPA) [50] and BinRSA. Instead of using a maximum of 5000 function evaluations, all of the metaheuristics used 100000, except for BinRSA and EBPSO, which used 10,000 and 5000, respectively. All of the methods were run 20 times.

Table 3 shows the results for this second comparison. The problems shown are a subset of BD1 for which not all metaheuristics reached the optimum in every run. All metaheuristics successfully reached the optima of the remaining problems in all runs. Except for EBPSO, the values in this table were obtained from [26,50]. The mean of the optimal values reached, the GAP and the hit ratio were originally considered as quality measures, but Friedman rank has also been incorporated as well as the *p-value* associated to the Wilcoxon signed-rank for each pairwise comparison between EBPSO and the rest of metaheuristics.

EBPSO and BinRSA reached the real optimum in all the runs (GAP = 0) and for all problems under consideration, and both rank top according to Friedman rank. However, EBPSO achieved this feat with a maximum of 5000 function evaluations, whereas BinRSA was allowed 10000. BMPA failed to reach the optimum value for the problem *ks_24d*. The Hit metric shows that EBPSO reached the optimal solution in all the runs (20 out of 20) for every problem, matching BinRSA and outperforming BMPA and BHHA, which have a lower Hit value for *ks_24d*. Since BinRSA achieved the same results as EBPSO, it was not necessary to perform the Wilcoxon signed-rank test between them. BHHA and BMPA obtained results very similar to those of EBPSO; consequently, relatively high *p*-values were obtained, indicating a clear similarity in their performance. In contrast, BSSA and BSCA yielded low *p*-values when compared to EBPSO, allowing us to conclude that there is a notable difference between their results.

In the third comparison, involving BD2, two BPSO versions are considered, as well as 5 differential evolution variants [21–25] and BinRSA. 30 runs were carried out for each problem and the best obtained solution out of those 30 runs was considered. The number of function evaluations was $100 \times Dimensions$, except for BinRSA and EBPSO, which both employed 5000 function evaluations each.

Table 4 shows the results for this third comparison. Except for EBPSO, the values in this table were sourced from [24,26].

EBPSO and NBin-DE reach the real optimum in all the runs and problems under consideration, whereas BinRSA does it for all problems but *kp_uc_300* and DBDE do not reach the real optimum for 5 problems: *kp_uc_1000*, *kp_wc_500*, *wc_1000*, *kp_sc_500* and *kp_sc_1000*. We can conclude that NBin-DE and EBPSO are the metaheuristics with the best performance in this comparison analysis, as they were able to reach the optimum even for the most complex instances.

The metaheuristics included in the fourth comparison on BD3 were simulated annealing (SA) [51], improved genetic algorithm-simulated

Table 4
Results of comparative analysis 3.

Problem	BPSO	MBPSO	BLDE	BinDE	AQDE	DBDE	NBin-DE	BinRSA	EBPSO
kp_uc_100	**1807**	**1807**	**1807**	1732	**1807**	**1807**	**1807**	**1807**	**1807**
kp_uc_200	3402	3378	3401	3146	3390	**3403**	**3403**	**3403**	**3403**
kp_uc_300	5443	5344	5441	4836	5401	**5444**	**5444**	5443	**5444**
kp_uc_500	9492	9145	9484	8058	9381	**9495**	**9495**	**9495**	**9495**
kp_uc_1000	18,829	17,600	18,492	15,380	18,467	18,843	**18,844**	**18,844**	**18,844**
kp_wc_100	**659**	**659**	658	656	**659**	**659**	**659**	**659**	**659**
kp_wc_200	**1332**	1331	1329	1304	1328	**1332**	**1332**	**1332**	**1332**
kp_wc_300	1961	1957	1961	1922	1958	**1963**	**1963**	**1963**	**1963**
kp_wc_500	3246	3230	3247	3163	3236	3247	**3250**	**3250**	**3250**
kp_wc_1000	6478	6401	6458	6252	6446	6463	**6482**	**6482**	**6482**
kp_sc_100	**813**	**813**	812	786	812	**813**	**813**	**813**	**813**
kp_sc_200	1629	1617	1626	1552	1624	**1631**	**1631**	**1631**	**1631**
kp_sc_300	**2433**	2402	2426	2307	2422	**2433**	**2433**	**2433**	**2433**
kp_sc_500	4069	3982	4051	3807	4045	4070	**4078**	**4078**	**4078**
kp_sc_1000	8212	7936	8073	7590	8137	8109	**8228**	**8228**	**8228**
kp_ss_100	**493**	**493**	**493**	**493**	**493**	**493**	**493**	**493**	**493**
kp_ss_200	**1001**	**1001**	**1001**	**1001**	**1001**	**1001**	**1001**	**1001**	**1001**
kp_ss_300	**1523**	**1523**	**1523**	**1523**	**1523**	**1523**	**1523**	**1523**	**1523**
kp_ss_500	**2518**	**2518**	**2518**	**2518**	**2518**	**2518**	**2518**	**2518**	**2518**
kp_ss_1000	**5068**	**5068**	**5068**	**5068**	**5068**	**5068**	**5068**	**5068**	**5068**

annealing hybrid algorithm (IGA-SA) [52], greedy search algorithm (GSA) [53], BWOA, BMPA, BinRSA and EBPSO. SA uses 2000 iterations, GSA keeps running until there are no more items available, and hybrid IGA-SA uses 250 iterations for SA and 50,000 for GA. Furthermore, BWOA and BMPA use 100,000 function evaluations, whereas BinRSA and EBPSO use 5000. Twenty runs were carried out for the BD3 problems.

Table 5 shows the results for this fourth comparison. The values in this table were sourced from [26,52]. The quality measures used were the mean of the optimal values reached and the GAP, but the Friedman rank has been again incorporated, as well as the *p*-value associated to the Wilcoxon signed-rank test.

EBPSO ranked first again. However, its GAP values are not 0 for problems *KP1_2000*, *KP1_5000*, *KP2_2000* and *KP2_5000*, i.e., the optimum value was not reached for all 20 executions for each of those problems. In any case, EBPSO is not outperformed by any other metaheuristic for any of the problems under consideration. Moreover, the second-best metaheuristic, BinRSA, also fails to reach the optimum value in six problems. EBPSO had the lowest Friedman rank out of the metaheuristics evaluated, with a value of 1.67. The low *p*-values in the Wilcoxon signed-rank test for each pairwise comparison suggest a substantial difference between the results of EBPSO and the other metaheuristics. BWOA and BMPA did not achieve acceptable results on some of the benchmark problems; therefore, the test was not performed for these methods, as the performance gap was already evident.

The last comparison, also on BD3, binary archimedes optimization algorithm (BAOA) [54], binary teaching-learning-based optimization (BTLBO) [55], binary farmland fertility algorithm (BFFA) [56], binary tunicate swarm algorithm (BTSA) [57], binary Harris Hawk optimization (BHHO) [46], binary slime mould algorithm (BSMA) [58] and BinRSA are considered, all with 100,000 function evaluations and 30 runs.

Table 6 shows the results for this fifth comparison. The values in this table were sourced from [26,58]. The mean and standard deviation of the optimal values reached, the GAP and Friedman rank were considered as quality measures along the *p*-value of the Wilcoxon signed-rank tests between EBPSO and each metaheuristic.

As in the fourth comparison, EBPSO is best-ranked. However, it does not reach the real optimum value for the problem *KP1_2000*, and the GAP value is not 0 for problems *KP1_5000*, *KP2_2000* and *KP2_5000*. BHHO, BSMA and BAOA reach the real optimum value for problem *KP1_2000*, but they fail to reach the optimum value for 12, 5 and 11 problems, respectively. Moreover, the second-best metaheuristic, BinRSA, also fails to reach the optimum value for five problems. A low standard deviation with respect to *KP1_5000*, *KP2_5000* and *KP3_5000* for EBPSO indicates a high stability even in solving more complex problems. The low *p*-value associated with each metaheuristic after comparison with EBPSO indicates a notable difference between their results and those obtained by EBPSO. Thus, we can conclude that EBPSO is again the metaheuristic with the best performance in this comparison analysis.

4.1. Ablation study of EBPSO enhancements

To assess the individual and combined contributions of the enhancements incorporated into EBPSO, an ablation study was conducted using the BD3 benchmark dataset, which includes the most challenging problem instances in our evaluation. The three primary enhancements introduced to the standard BPSO are: (i) the use of time-varying parameters, (ii) a dynamic transfer function, and (iii) a repair/improvement mechanism designed to improve solution feasibility and quality.

We denote by BPSO1, BPSO2, and BPSO3 the EBPSO variants that implement each of these enhancements in isolation, respectively. In addition, BPSO4, BPSO5, and BPSO6 correspond to variants combining two of the three enhancements as follows:

- BPSO4: time-varying parameters + dynamic transfer function,
- BPSO5: time-varying parameters + repair/improvement mechanism,
- BPSO6: dynamic transfer function + repair/improvement mechanism.

Table 5
Results of comparative analysis 4.

Problem		SA	IGA-SA	GSA	BWOA	BMPA	BinRSA	EBPSO
KP1_100	Mean	**9147.00**	8575.00	2983.00	**9147.00**	**9147.00**	**9147.00**	**9147.00**
	GAP%	**0.00E+00**	6.25E+00	6.74E+01	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
KP1_200	Mean	10,163.00	8576.00	4544.00	**11,238.00**	**11,238.00**	**11,238.00**	**11,238.00**
	GAP%	9.57E+00	2.37E+01	5.96E+01	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
KP1_500	Mean	21,390.00	12,072.00	9865.00	28,845.50	**28,857.00**	**28,857.00**	**28,857.00**
	GAP%	2.59E+01	5.82E+01	6.58E+01	3.99E-02	**0.00E+00**	**0.00E+00**	**0.00E+00**
KP1_1000	Mean	36,719.00	14,563.00	14,927.00	53,382.20	54,408.75	**54,503.00**	**54,503.00**
	GAP%	3.26E+01	7.33E+01	7.26E+01	2.06E+00	1.73E-01	**0.00E+00**	**0.00E+00**
KP1_2000	Mean	65,793.00	27,645.00	25,579.00	93,053.00	96,490.50	110,587.55	**110,590.23**
	GAP%	4.05E+01	7.50E+01	7.69E+01	1.59E+01	1.28E+01	3.39E-02	**3.14E-02**
KP1_5000	Mean	150,731.00	39,677.00	49,306.00	-	-	276,453.60	**276,454.20**
	GAP%	4.55E+01	8.56E+01	8.22E+01	-	-	1.23E-03	**1.01E-03**
KP2_100	Mean	1486.00	1217.00	1041.00	**1514.00**	**1514.00**	**1514.00**	**1514.00**
	GAP%	1.85E+00	1.96E+01	3.12E+01	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
KP2_200	Mean	1537.00	1347.00	1073.00	**1634.00**	**1634.00**	**1634.00**	**1634.00**
	GAP%	5.94E+00	1.76E+01	3.43E+01	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
KP2_500	Mean	3744.00	3038.00	2951.00	4525.50	4557.00	4557.50	**4566.00**
	GAP%	1.80E+01	3.35E+01	3.54E+01	8.87E-01	1.97E-01	1.86E-01	**0.00E+00**
KP2_1000	Mean	6831.00	5435.00	5675.00	8287.00	8919.25	9051.05	**9052.00**
	GAP%	2.45E+01	4.00E+01	3.73E+01	8.45E+00	1.47E+00	1.05E-02	**0.00E+00**
KP2_2000	Mean	12,780.00	10,938.00	11,064.00	14,827.45	17,167.00	18,046.75	**18,046.98**
	GAP%	2.92E+01	3.94E+01	3.87E+01	1.79E+01	4.90E+00	2.35E-02	**2.22E-02**
KP2_5000	Mean	29,220.00	25,448.00	27,387.00	-	-	44,353.15	**44,353.72**
	GAP%	3.41E+01	4.26E+01	3.83E+01	-	-	6.43E-03	**5.14E-03**
KP3_100	Mean	2296.00	1690.00	1095.00	2396.70	**2397.00**	**2397.00**	**2397.00**
	GAP%	4.21E+00	2.95E+01	5.43E+01	1.25E-02	**0.00E+00**	**0.00E+00**	**0.00E+00**
KP3_200	Mean	2594.00	1495.00	1095.00	2695.50	**2697.00**	**2697.00**	**2697.00**
	GAP%	3.82E+00	4.46E+01	5.94E+01	5.56E-02	**0.00E+00**	**0.00E+00**	**0.00E+00**
KP3_500	Mean	6103.00	3412.00	2916.00	-	-	**7117.00**	**7117.00**
	GAP%	1.42E+01	5.21E+01	5.90E+01	-	-	**0.00E+00**	**0.00E+00**
KP3_1000	Mean	11,789.00	6290.00	5589.00	-	-	**14,390.00**	**14,390.00**
	GAP%	1.81E+01	5.63E+01	6.12E+01	-	-	**0.00E+00**	**0.00E+00**
KP3_2000	Mean	22,482.00	12,312.00	10,818.00	-	-	**28,919.00**	**28,919.00**
	GAP%	2.23E+01	5.74E+01	6.26E+01	-	-	**0.00E+00**	**0.00E+00**
KP3_5000	Mean	53,672.00	30,302.00	27,304.00	-	-	**72,505.00**	**72,505.00**
	GAP%	2.60E+01	5.82E+01	6.23E+01	-	-	**0.00E+00**	**0.00E+00**
Friedman Rank		4.22	5.61	6.06	4.53	3.92	2.00	**1.67**
Final Rank		5	6	7	4	3	2	**1**
Wilcoxon signed-rank	*p-value*	0.000213	0.000079	0.000052	-	-	0.014901	-

The parameter configurations for each version are summarized below:

- BPSO1: Implements time-varying parameters with initial values of $C_1 = 1.0$, $C_2 = 0.0$, and final values of $C_1 = 0.0$, $C_2 = 1.0$. The inertia weight w varies from $w_{\max} = 0.9$ to $w_{\min} = 0.0$, and the standard sigmoid function is used as the transfer function.
- BPSO2: Employs a dynamic transfer function with fixed parameters $C_1 = C_2 = 1$, and $w = 0.9$; the shaping parameter $\alpha \in [0.0, 1.0]$ and the slope control parameter $h_d = 1$.
- BPSO3: Incorporates the repair/improvement mechanism, using the same fixed values for C_1, C_2, and w as in BPSO2, and computes the solution rate $SRate$ using Eq. (17). The standard sigmoid function is used for the transfer function.

For the hybrid variants (BPSO4–BPSO6), the same parameter settings were adopted as in their corresponding isolated components. All configurations were tested using a swarm of 100 particles, with 20 independent runs and a maximum of 5000 function evaluations per run.

As the knapsack problem is inherently constrained, BPSO, BPSO1, BPSO2, and BPSO4, which lack an intrinsic mechanism for handling infeasibility, were augmented with a widely accepted constraint-handling approach based on feasibility-preserving rules [40], in order to ensure a fair comparison and consistent search behavior.

The results of this analysis, shown in Table 7, provide insight into the relative effectiveness of each enhancement. BPSO3, which includes the repair/improvement mechanism, consistently outperformed BPSO1 and BPSO2 across the majority of problem instances. This performance can be attributed to its superior ability to handle infeasible solutions and enhance feasible ones.

BPSO1 and BPSO2 showed similar performance levels. While BPSO1 benefits from a more effective balance between exploration and exploitation, it is prone to premature convergence before the shift in acceleration coefficients. Conversely, BPSO2's adaptive transfer function stimulates exploration but is hindered by static velocity parameters and lacks robust infeasibility management.

Among the pairwise combinations, BPSO4 (time-varying parameters + dynamic transfer function) exhibited superior performance,

Table 6
Results of comparative analysis 5.

Problem		BAOA	BTLBO	BFFA	BTSA	BHHO	BSMA	BinRSA	EBPSO
KP1_100	Mean	**9147.00**	**9147.00**	**9147.00**	**9147.00**	**9147.00**	**9147.00**	**9147.00**	**9147.00**
	Std.Dev.	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
	GAP%	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
KP1_200	Mean	**11,238.00**	**11,238.00**	**11,238.00**	**11,238.00**	**11,238.00**	**11,238.00**	**11,238.00**	**11,238.00**
	Std.Dev.	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
	GAP%	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
KP1_500	Mean	28,852.00	28,851.00	28,843.00	28,849.00	28,850.00	**28,857.00**	**28,857.00**	**28,857.00**
	Std.Dev.	6.71E+00	1.08E+01	1.41E+01	1.13E+01	8.07E+00	**0.00E+00**	**0.00E+00**	**0.00E+00**
	GAP%	1.60E-02	2.10E-02	4.70E-02	2.60E-02	2.40E-02	**0.00E+00**	**0.00E+00**	**0.00E+00**
KP1_1000	Mean	53,574.00	53,369.00	53,389.00	53,301.00	53,396.00	**54,503.00**	**54,503.00**	**54,503.00**
	Std.Dev.	9.82E+02	1.11E+03	1.10E+03	1.22E+03	1.14E+03	**0.00E+00**	**0.00E+00**	**0.00E+00**
	GAP%	1.74E+00	2.08E+00	2.04E+00	2.20E+00	2.03E+00	**0.00E+00**	**0.00E+00**	**0.00E+00**
KP1_2000	Mean	**110,625.00**	98,785.00	93,779.00	95,619.00	**110,625.00**	**110,625.00**	110,600.30	110,616.20
	Std.Dev.	**0.00E+00**	2.97E+03	3.16E+03	2.84E+03	**0.00E+00**	**0.00E+00**	1.45E+01	5.27E+00
	GAP%	**0.00E+00**	1.07E+01	1.52E+01	1.36E+01	**0.00E+00**	**0.00E+00**	2.23E-02	7.95E-03
KP1_5000	Mean	274,385.00	273,057.00	274,329.00	272,923.00	273,961.00	274,569.00	276,456.00	**276,456.03**
	Std.Dev.	1.06E+03	1.69E+03	9.94E+02	1.83E+03	1.30E+03	7.74E+02	0.00E+00	**0.198E+00**
	GAP%	7.49E-01	1.23E+00	7.70E-01	1.28E+00	9.03E-01	6.83E-01	3.62E-05	**3.51E-05**
KP2_100	Mean	**1514.00**	**1514.00**	**1514.00**	**1514.00**	**1514.00**	**1514.00**	**1514.00**	**1514.00**
	Std.Dev.	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
	GAP%	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
KP2_200	Mean	**1634.00**	**1634.00**	**1634.00**	**1634.00**	**1634.00**	**1634.00**	**1634.00**	**1634.00**
	Std.Dev.	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
	GAP%	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
KP2_500	Mean	4549.40	4540.80	4544.60	4537.20	4518.70	4565.30	**4566.00**	**4566.00**
	Std.Dev.	2.90E+01	3.67E+01	2.94E+01	4.62E+01	3.73E+01	3.49E+00	**0.00E+00**	**0.00E+00**
	GAP%	3.64E-01	5.52E-01	4.69E-01	6.31E-01	1.04E+00	1.50E-02	**0.00E+00**	**0.00E+00**
KP2_1000	Mean	**9052.00**	8869.70	**9052.00**	8346.80	8835.60	**9052.00**	9051.30	**9052.00**
	Std.Dev.	**0.00E+00**	2.05E+02	**0.00E+00**	1.89E+02	2.27E+02	**0.00E+00**	4.66E-01	**0.00E+00**
	GAP%	**0.00E+00**	2.01E+00	**0.00E+00**	7.79E+00	2.39E+00	**0.00E+00**	7.73E-03	**0.00E+00**
KP2_2000	Mean	15,885.00	15,612.00	15,126.00	14,902.00	15,729.00	17,557.00	18,049.73	**18,050.13**
	Std.Dev.	5.10E+02	5.50E+02	4.89E+02	6.58E+02	5.02E+02	2.90E+02	8.28E-01	**1.01E+00**
	GAP%	1.20E+01	1.35E+01	1.62E+01	1.74E+04	1.29E+01	2.73E+00	7.02E-03	**4.82E-03**
KP2_5000	Mean	44,206.00	43,785.00	44,166.00	42,972.00	43,276.00	44,312.00	44,354.60	**44,355.00**
	Std.Dev.	9.07E+01	5.30E+02	9.64E+01	6.52E+02	9.10E+02	1.13E+01	1.04E+00	**0.00E+00**
	GAP%	3.36E-01	1.29E+00	4.28E-01	3.12E+00	2.43E+00	9.80E-02	3.16E-03	**2.25E-03**
KP3_100	Mean	2396.50	**2397.00**	**2397.00**	2395.20	2396.40	**2397.00**	**2397.00**	**2397.00**
	Std.Dev.	1.06E+00	**0.00E+00**	**0.00E+00**	1.30E+00	1.19E+00	**0.00E+00**	**0.00E+00**	**0.00E+00**
	GAP%	2.10E-02	**0.00E+00**	**0.00E+00**	7.50E-02	2.50E-02	**0.00E+00**	**0.00E+00**	**0.00E+00**
KP3_200	Mean	2695.10	2694.50	2693.80	2692.10	2694.70	**2697.00**	**2697.00**	**2697.00**
	Std.Dev.	1.74E+00	2.21E+00	2.03E+00	3.35E+00	1.65E+00	**0.00E+00**	**0.00E+00**	**0.00E+00**
	GAP%	7.00E-02	9.30E-02	1.19E-01	1.82E-01	8.50E-02	**0.00E+00**	**0.00E+00**	**0.00E+00**
KP3_500	Mean	**7117.00**	7059.20	7064.90	6999.50	**7117.00**	**7117.00**	**7117.00**	**7117.00**
	Std.Dev.	**0.00E+00**	7.60E+01	5.02E+01	1.24E+02	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
	GAP%	**0.00E+00**	8.12E-01	7.32E-01	1.65E+00	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
KP3_1000	Mean	14,279.00	14,325.00	14,308.00	14,191.00	14,225.00	**14,390.00**	**14,390.00**	**14,390.00**
	Std.Dev.	9.85E+01	5.10E+01	4.62E+01	1.32E+02	7.85E+01	**0.00E+00**	**0.00E+00**	**0.00E+00**
	GAP%	7.70E-01	4.46E-01	5.67E-01	1.38E+00	1.15E+00	**0.00E+00**	**0.00E+00**	**0.00E+00**
KP3_2000	Mean	28,597.00	28,702.00	28,793.00	**28,919.00**	28,657.00	**28,919.00**	**28,919.00**	**28,919.00**
	Std.Dev.	2.09E+02	1.68E+02	1.37E+02	**0.00E+00**	1.93E+02	**0.00E+00**	**0.00E+00**	**0.00E+00**
	GAP%	1.11E+00	7.49E-01	4.43E-01	**0.00E+00**	9.04E-01	**0.00E+00**	**0.00E+00**	**0.00E+00**
KP3_5000	Mean	71,579.00	71,708.00	71,451.00	71,018.00	71,826.00	72,075.00	**72,505.00**	**72,505.00**
	Std.Dev.	4.19E+02	5.13E+02	4.01E+02	3.45E+02	4.30E+02	2.26E+02	**0.00E+00**	**0.00E+00**
	GAP%	1.28E+00	1.10E+00	1.45E+00	2.05E+00	9.35E-01	5.93E-01	**0.00E+00**	**0.00E+00**
Friedman Rank		4.42	5.44	5.42	6.75	5.39	3.00	2.97	**2.61**
Final Rank		4	7	6	8	5	3	2	**1**
Wilcoxon signed-rank	*p-value*	0.007956	0.000577	0.000840	0.000084	0.002327	0.088809	0.025892	-

Table 7
Results of ablation analysis on EBPSO.

Problem		BPSO	BPSO1	BPSO2	BPSO3	BPSO4	BPSO5	BPSO6	EBPSO
KP1_100	Mean	3023.07	**9147.00**	**9147.00**	**9147.00**	**9147.00**	**9147.00**	**9147.00**	**9147.00**
	Std.Dev.	1.26E+04	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
	GAP%	66.95E+00	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
KP1_200	Mean	3,748.00	**11,238.00**	**11,238.00**	**11,238.00**	**11,238.00**	**11,238.00**	**11,238.00**	**11,238.00**
	Std.Dev.	1.28E+02	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
	GAP%	66.64E+00	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
KP1_500	Mean	9,054.00	28,851.00	28,853.00	28,855.00	28,856.84	28,855.23	28,855.74	**28,857.00**
	Std.Dev.	1.19E+03	2.91E+01	1.98E+01	1.87E+01	1.44E+02	3.97E+01	2.10E+01	**0.00E+00**
	GAP%	68.62E+00	2.07E-02	1.38E-02	6.93E-02	5.54E-04	6.13E-03	4.36E-03	**0.00E+00**
KP1_1000	Mean	9,929.00	53,893.00	53,961.00	53,979.00	54,129.14	54,051.04	54,003.46	**54,503.00**
	Std.Dev.	2.83E+04	7.10E+02	6.19E+03	3.59E+02	3.20E+01	9.43E+00	2.71E+02	**0.00E+00**
	GAP%	81.78E+00	1.12E+00	0.99E+00	0.96E+00	0.68E+00	0.82E+00	0.91E+00	**0.00E+00**
KP1_2000	Mean	10,139.15	99,510.00	99,943.00	105,382.00	107,682.38	105,968.01	105,936.50	**110,616.20**
	Std.Dev.	5.93E+04	1.17E+01	1.20E+00	2.96E-01	5.78E+00	5.29E+01	5.93E+00	**5.27E+00**
	GAP%	90.83E+00	10.05E+00	9.66E+00	4.74E+00	2.65E+00	4.21E+00	4.24E+00	**7.95E-03**
KP1_5000	Mean	22,492.63	272,603.00	272,381.00	274,183.00	275,005.72	274,541.67	274,709.84	**276,456.03**
	Std.Dev.	0.73E+02	8.41E+02	1.98E+02	3.59E+02	8.93E+02	7.89E+01	8.86E+02	**0.198E+00**
	GAP%	91.86E+00	1.39E+00	1.47E+00	0.82E+00	0.52E+00	0.69E+00	0.63E+00	**3.51E-05**
KP2_100	Mean	893.58	**1514.00**	**1514.00**	**1514.00**	**1514.00**	**1514.00**	**1514.00**	**1514.00**
	Std.Dev.	1.94E+03	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
	GAP%	40.97E+00	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
KP2_200	Mean	946.30	**1634.00**	**1634.00**	**1634.00**	**1634.00**	**1634.00**	**1634.00**	**1634.00**
	Std.Dev.	9.51E+03	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
	GAP%	42.08E+00	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
KP2_500	Mean	958.06	4549.64	4548.59	4550.17	4561.08	4554.00	4555.81	**4566.00**
	Std.Dev.	3.48E+02	1.57E+01	1.20E+01	1.30E+01	9.34E+00	1.45E+02	5.18E+02	**0.00E+00**
	GAP%	79.01E+00	0.36E+00	0.38E+00	0.35E+00	0.11E+00	0.27E+00	0.22E+00	**0.00E+00**
KP2_1000	Mean	932.90	8580.93	8592.61	8683.80	8840.31	8701.53	8696.26	**9052.00**
	Std.Dev.	5.88E+03	6.42E+02	2.98E+02	2.98E+02	9.54E+00	5.52E+01	3.46E+02	**0.00E+00**
	GAP%	89.69E+00	5.20E+00	5.07E+00	4.07E+00	2.34E+00	3.87E+00	3.93E+00	**0.00E+00**
KP2_2000	Mean	2,024.00	15,497.00	15,613.00	16,105.00	16,839.73	16,383.00	16,295.79	**18,050.13**
	Std.Dev.	9.82E+03	1.56E+02	8.12E+02	3.74E+02	6.30E+01	1.70E+00	0.50E+02	**1.01E+00**
	GAP%	88.78E+00	14.15E+00	13.51E+00	10.78E+00	6.71E+00	9.24E+00	9.72E+00	**4.82E-03**
KP2_5000	Mean	5,941.40	42,979.00	43,162.00	43,947.00	44,152.96	43,994.13	44,052.05	**44,355.00**
	Std.Dev.	4.72E+04	8.05E+01	1.30E+01	2.96E+01	6.61E+00	2.55E+01	5.89E+00	**0.00E+00**
	GAP%	86.60E+00	3.10E+00	2.69E+00	0.92E+00	0.46E+00	0.81E+00	0.68E+00	**2.25E-03**
KP3_100	Mean	817.09	**2397.00**	**2397.00**	**2397.00**	**2397.00**	**2397.00**	**2397.00**	**2397.00**
	Std.Dev.	4.50E+02	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
	GAP%	65.91E+00	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
KP3_200	Mean	984.21	2695.21	2694.10	2696.03	**2697.00**	**2697.00**	**2697.00**	**2697.00**
	Std.Dev.	3.41E+03	1.04E+00	1.20E+00	1.74E+00	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
	GAP%	63.50E+00	0.06E+00	0.11E+00	0.03E+00	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
KP3_500	Mean	1985.63	7001.03	6999.17	7005.30	7079.00	7065.81	7059.90	**7117.00**
	Std.Dev.	3.30E+02	6.69E+01	5.95E+01	1.26E+01	1.61E+02	6.35E+01	5.69E+02	**0.00E+00**
	GAP%	72.10E+00	1.63E+00	1.65E+00	1.57E+00	0.53E+00	0.72E+00	0.80E+00	**0.00E+00**
KP3_1000	Mean	3,742.07	14,199.00	14,197.00	14,250.00	14,306.93	14,291.00	14,286.20	**14,390.00**
	Std.Dev.	5.23E+02	3.92E+02	5.67E+02	1.37E+02	6.68E+01	8.61E+00	4.27E+01	**0.00E+00**
	GAP%	73.99E+00	1.32E+00	1.34E+00	0.97E+00	0.58E+00	0.69E+00	0.72E+00	**0.00E+00**
KP3_2000	Mean	5,651.69	28,619.00	28,642.00	28,690.00	28,873.17	28,790.03	28,792.82	**28,919.00**
	Std.Dev.	5.90E+04	3.13E+02	9.95E+02	2.51E+02	3.20E+02	2.91E+02	2.82E+00	**0.00E+00**
	GAP%	80.45E+00	1.04E+00	0.96E+00	0.79E+00	0.15E+00	0.44E+00	0.43E+00	**0.00E+00**
KP3_5000	Mean	9,970.73	71,780.00	71,794.00	72,012.00	72,259.20	72,103.05	72,102.96	**72,505.00**
	Std.Dev.	3.30E+03	2.04E+02	1.98E+02	2.83E+02	1.40E+00	7.31E+02	6.54E+01	**0.00E+00**
	GAP%	86.24E+00	0.99E+00	0.98E+00	0.68E+00	0.34E+00	0.56E+00	0.55E+00	**0.00E+00**

leveraging the synergistic effects of both adaptive mechanisms. BPSO5 and BPSO6 did not yield significant improvements over their individual components due to the aforementioned limitations.

These findings confirm that the full EBPSO algorithm, integrating all three enhancements, offers a substantial performance advantage, particularly for high-dimensional problems (≥ 200 variables). The combination of adaptability, guided search, and solution repair leads to more effective exploration and exploitation of the search space.

Moreover, even the ablated variants of EBPSO–whether isolated or partially combined–outperformed some of the lower-ranked metaheuristics included in Table 6, such as BTSA, as measured by their Friedman ranks. This underscores the robustness and competitiveness of the proposed enhancements, even when applied independently.

Table 8
Best achieved fitness values and execution times for BPSO and EBPSO.

		fitness (BPSO)	fitness (EBPSO)	time (BPSO)	time (EBPSO)
Scenario 1	Mean	0.233	**0.218**	5h 21min	**43min**
	Std. Dev.	6.17E-03	**8.41E-4**	**4.07E-05**	5.14E-5
Scenario 1a	Mean	0.242	**0.225**	5h 22min	**38min**
	Std. Dev.	9.80E-03	**4.75E-4**	8.55E-05	**6.91E-5**
Scenario 2	Mean	0.245	**0.227**	5h 24min	**36min**
	Std. Dev.	6.44E-03	**5.39E-4**	8.64E-05	**7.59E-5**
Scenario 3	Mean	0.256	**0.229**	5h 15min	**40min**
	Std. Dev.	9.70E-03	**2.70E-4**	6.84E-05	**3.14E-5**
Scenario 4	Mean	0.258	**0.230**	5h 21min	**36min**
	Std. Dev.	3.39E-03	**8.41E-4**	6.37E-05	**3.09E-5**
Scenario 5	Mean	0.374	**0.370**	5h 27min	**1h 5min**
	Std. Dev.	9.22E-03	**6.40E-4**	**5.14E-05**	7.05E-5
Scenario 5a	Mean	0.390	**0.382**	5h 24min	**1h 4min**
	Std. Dev.	8.98E-03	**8.80E-4**	**3.95E-05**	6.62E-5
Scenario 6	Mean	0.378	**0.363**	5h 21min	**1h 3min**
	Std. Dev.	9.70E-03	**3.89E-4**	**4.57E-05**	4.89E-5
Scenario 7	Mean	0.369	**0.360**	5h 20min	**1h 5min**
	Std. Dev.	4.46E-03	**1.30E-4**	**2.96E-05**	5.68E-5
Scenario 8	Mean	0.354	**0.350**	5h 16min	**1h 2min**
	Std. Dev.	6.59E-03	**3.43E-4**	5.29E-05	**2.50E-5**
Scenario 9	Mean	0.546	**0.529**	5h 24min	**1h 17min**
	Std. Dev.	8.75E-03	**7.79E-4**	**2.81E-05**	6.32E-5
Scenario 9a	Mean	0.574	**0.562**	5h 24min	**1h 20min**
	Std. Dev.	8.93E-03	**6.60E-4**	**5.77E-05**	6.07E-5
Scenario 10	Mean	0.532	**0.529**	5h 28min	**1h 18min**
	Std. Dev.	7.57E-03	**2.31E-4**	5.43E-05	**4.93E-05**
Scenario 11	Mean	0.530	**0.525**	5h 15min	**1h 23min**
	Std. Dev.	6.14E-03	**3.17E-4**	**2.28E-05**	2.85E-5
Scenario 12	Mean	0.491	**0.482**	5h 23min	**1h 19min**
	Std. Dev.	3.73E-03	**7.86E-4**	**2.97E-05**	3.51E-5

5. Improved pandemic propagation problem solving

The effectiveness of the mitigation the early spread of pandemics through passenger air traffic is demonstrated using the case study presented in [12], which focuses on Spain. This study encompasses 38 national airports and 5000 international connections, analyzing a total of 9678 flights from 237 international airports during the period from September 24, to October 7, 2020.

The Neo4J graph-oriented database was extended to accommodate this case study. In particular, data on Spanish airport taxes and statistical information from the National Statistics Institute (INE) concerning the average expenditure of passengers arriving in Spain disaggregated by country of origin and travel purpose (business, leisure/holiday, or other) were incorporated into the Neo4J database.

A total of 15 scenarios were evaluated, considering three different levels of COVID-19 importation risk reduction at Spanish airports (25 %, 50 %, and 75 %) and varying decision-makers' (DMs) preferences, expressed through rankings from the most to least important objective. Further technical details regarding the extension of the Neo4J database and the characterization of these scenarios are available in [12].

EBPSO employed different parameter values compared to those used in Section 4: C_1 ranges from 1.5 to 0.5. $SRate$ is computed with Eq. (17), which returns a lower value to address the high dimension of the constrained problem by reducing randomness during repair and improvement of solutions.

A higher initial value of C_1 (C_{1ini}) encourages stronger individual exploration by each particle at the beginning of the optimization process, while a higher final value C_{1fin} ensures that some degree of exploration remains toward the final iterations. These modifications to C_1 effectively delay the convergence process.

In general, particles begin to converge when $C_2 > C_1$. If both parameters had the same initial and final values, convergence would typically begin around the midpoint of the execution, when C_2 overtakes C_1. By increasing both C_{1ini} and C_{1fin}, the point at which convergence begins is delayed, shifting it from the middle to a later stage in the optimization process.

The number of particles was set to 100, and the number of function evaluations was limited to 5,000. The remaining parameters were the same as those used in Section 4. BPSO employed the parameters it shares with EBPSO, except for the number of particles, which was set to 200, and C_1, which was fixed at 1.0. All runs were executed on a machine with the same specifications described in [12].

Table 8 reports the mean fitness of both metaheuristics (BPSO and EBPSO) after 20 independent runs, along with the respective mean execution time, also specifying the standard deviation for both fitness and execution time. This is a minimization problem, so the repair/improvement method was adapted accordingly, as described in Section 3.

We found that the newer version of BPSO outperformed the older version in all scenarios, in terms of both fitness and execution time. Moreover, EBPSO is able to obtain higher quality solutions with lower dispersion than BPSO, as evidenced by the difference in their standard deviations.

The difference in execution time between the two is greater for problems with less epidemiological constraint. This is attributed to the fact that particles more frequently explore infeasible regions in more constrained problems, where the repair/improvement method has to be applied more often. For this reason, scenarios 1–4, which are the least constrained, are the ones where EBPSO completed the task fastest. Additionally, since EBPSO has enhanced exploration and exploitation capabilities, it is possible to choose a much lower number of particles, and the performance will still be higher. This is the main cause of the lower execution time of EBPSO compared to the older version. Finally, there is no significant difference in the standard deviations of their execution times.

6. Conclusions

Premature convergence poses a significant obstacle in population-based metaheuristics, such as particle swarm optimization (PSO). This issue is particularly pronounced in large-scale and constrained optimization problems. To overcome this challenge, this study introduces an enhanced version of BPSO (EBPSO), which encourages particles to keep searching for better solutions once they have gathered in a small area of the search space, thus preventing them from getting stuck in possible local optima while also balancing exploration and exploitation with time-varying parameters and a novel transfer function.

The transfer function can make the particles scatter or follow their main attractor, which varies depending on the optimization phase, while the time-varying parameters are in charge of gradually changing each particle's attractor to make them behave in a more individual manner to a more social one, therefore, balancing exploration and exploitation. In addition, a repair/improvement method was added to handle infeasible solutions in a robust manner.

EBPSO has been tested in five comparison analyses against several state-of-the-art binary metaheuristics, using three benchmark datasets for the knapsack problem and different quality measures. The results show that EBPSO outperforms or, at worst, matches the best-performing methods in practically all cases. EBPSO reaches the corresponding real optimum value and with GAP values of 0 in almost all problems. EBPSO is only outperformed in problem *KP1_2000* in the fifth comparison analysis by BHHO, BSMA and BAOA, but they fail to reach the optimum value for problems 12, 5, and 11, respectively.

Although EBPSO was compared with well-known state-of-the-art binary metaheuristics and demonstrated effectiveness in finding optimal solutions through dynamic parameters, transfer functions, and repair/improvement methods, several approaches remain unexplored in the context of the knapsack problem or constrained binary problems in general. These include fuzzy logic-based methods for parameter computation [37] and chaos-based mechanisms [59] to prevent particles from becoming trapped in local optima or to dynamically compute parameters. Applying these methods to such problems and evaluating their effectiveness would constitute an interesting future research line.

It has also been demonstrated that EBPSO clearly outperforms BPSO (used so far) in the complex multi-objective decision-making problem that motivated the research carried out in this paper: the reduction of the risk of pandemic importation through passenger air traffic management. In the 12 scenarios under consideration, EBPSO outperformed BPSO in terms of the quality of the resulting solution obtained as well as the time it takes to reach the solution, which was reduced from over five hours to around one hour.

Parallelization or GPU scaling of EBPSO is a promising option for future implementation, as the population of particles can be distributed across multiple GPUs. By sharing common variables, such as C_1, C_2 and w, it would be possible to update the positions of the particles in parallel, thereby reducing the execution time of the metaheuristic.

Another interesting research line is the implementation of a cloud computing-based system, where users can access an on-demand optimization service through the metaheuristic. While a small-scale version has already been implemented on the RRPS-PAT platform, the metaheuristic currently applies only to the pandemic propagation problem through air traffic. Developing a more general optimization engine capable of handling multiple problem types would be a valuable extension.

The integration of BPSO with methods based on *granular ball computing* can provide a semantic and informative structure to the binary discrete space of BPSO, enhancing search efficiency, robustness to noise, and the ability to escape local optima. Specifically, granular balls allow the grouping of instances with similar characteristics, offering a semantic context that can guide the BPSO toward more relevant regions of the search space. The uncertainty and density within each granular ball can be used as adaptive criteria to adjust the bit-flipping probabilities in the particles (e.g., by modifying the sigmoid function or activation thresholds). In feature selection tasks, the Enhanced Binary Particle Swarm Optimization (EBPSO) algorithm can be applied after reducing the attribute space with the method proposed in [60], thereby mitigating the low efficiency typically associated with rough set–based approaches. Furthermore, rough set approximations can be leveraged to construct three-way decision models, such as the framework described in [61], which introduce an additional layer of interpretability and flexibility in handling uncertainty within the feature selection process.

The paper presents a case study involving 38 Spanish airports with international traffic over a two-week period in 2020, comprising 5000 connections and 9678 flights from 237 international airports. The aim was to illustrate the operation and potential of a decision support system applied to a real-world problem, considering that air traffic management is a national competence; however, during the pandemic, the European Union also intervened, which justifies extending the analysis to a European level. Currently, collaboration is ongoing with CRIDA (Spanish ATM R&D and Innovation Reference Centre) and MUAC (Maastricht Upper Area Control, Eurocontrol) to expand the study within a project proposal for the HORIZON-SESAR-2025 call. This expansion entails significant complexity due to the need for additional data and funding. The model allows for adjustment of the time horizon, with two weeks considered an adequate period to avoid decisions that are either too rapid or too prolonged, which could affect operational stability and data relevance. These issues, along with scalability and real-time operational constraints, will be addressed in the forthcoming European project proposal.

The proposed approach, based on a binary metaheuristic to decide which connections between airports should be closed according to multiple objectives (economic, social, and health-related), has high potential for generalization to other domains characterized by complex networks and optimization decisions under multiple constraints. Below are three relevant possible extensions:

- Freight logistics. In logistics networks, the connections between warehouses, distribution centers, and points of sale can be modeled analogously to air routes. The binary metaheuristic allows for decisions on which logistic routes should be activated or suppressed in order to optimize objectives such as minimizing total cost, reducing delivery times and environmental impact, or improving resilience to disruptions. This approach is particularly useful in network reconfiguration contexts following changes in demand or supply chain issues.
- Emergency response. In scenarios involving the management of natural disasters, pandemics, or large-scale incidents, decisions regarding which routes should be prioritized or deactivated (e.g., roads, bridges, hospitals, or logistics centers) are critical. The model can be adapted to maximize population coverage, minimize response times, and optimize the allocation of limited resources. In this context, binary variables represent the operability of routes or critical nodes, and the objectives include both logistical efficiency and humanitarian impact.
- Land and maritime transportation. Road, rail, or maritime transport networks can also be modeled as graphs in which the activation or temporary closure of routes is optimized. The metaheuristic enables the incorporation of criteria such as congestion, infrastructure maintenance, territorial equity, or environmental sustainability. This allows for the simulation of planning policies or dynamic traffic management strategies based on scenarios involving climate change, mass events, or territorial reorganization.

In these domains and others, using a binary representation for activation/deac-tivation decisions, combined with multi-objective optimization, provides a flexible, interpretable, and adaptable framework for different geographic and political contexts.

CRediT authorship contribution statement

Gabriel A. Peña: Writing – original draft, Software, Formal analysis, Conceptualization; **Antonio Jiménez-Martín:** Writing – review & editing, Validation, Supervision, Methodology, Funding acquisition, Formal analysis; **Alfonso Mateos:** Writing – review & editing, Validation, Supervision, Methodology, Funding acquisition, Formal analysis.

Data availability

The code for EBPSO is available on GitHub at the following link: https://github.com/GabrielTRK/Visualization_Neo4j_Back_End, to facilitate reproducibility for other researchers.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This paper was supported by grants PID2021-122209OB-C31, RED2022-134540-T and PID2024-155179NB-C22 funded by MICIU/AEI/10.13039/ 501100011033.

References

[1] A.J. Tatem, D.J. Rogers, S.I. Hay, Global transport networks and infectious disease spread, Adv. Parasitol. 62 (2006) 293–343.
[2] P. Bajardi, C. Poletto, J.J. Ramasco, M. Tizzoni, V. Colizza, A. Vespignani, Human mobility networks, travel restrictions, and the global spread of 2009 H1N1 pandemic, PLoS One 6 (1) (2011) e16591.
[3] M. Dong, X. Zhang, K. Yang, R. Liu, P. Chen, Forecasting the COVID-19 transmission in Italy based on the minimum spanning tree of dynamic region network, PeerJ 9 (2021) e11603.
[4] M.K.P. So, A.M.Y. Chu, A. Tiwari, J.N.L. Chan, On topological properties of COVID-19: predicting and controling pandemic risk with network statistics, Sci. Rep. 11 (2020) 5112.
[5] A. Fragua, A. Jiménez-Martín, A. Mateos, Complex network analysis techniques for the early detection of the outbreak of pandemics transmitted through air traffic, Sci. Rep. 13 (2023) 18174.

[6] M. Chinazzi, J.T. Davis, M. Ajelli, C. Gioannini, M. Litvinova, S. Merler, A. Vespignani, The effect of travel restrictions on the spread of the 2019 novel coronavirus (COVID-19) outbreak, Science 368 (6489) (2020) 395–400.

[7] C.R. Wells, P. Sah, S.M. Moghadas, A. Pandey, A. Shoukat, Y. Wang, A.P. Galvani, Impact of international travel and border control measures on the global spread of the novel 2019 coronavirus outbreak, in: Proceedings of the National Academy of Sciences, 117, 2020, pp. 7504–7509.

[8] Anypriya, P. Bansal, D.J. Graham, Modelling the propagation of infectious disease via transportation networks, Sci. Rep. 12 (2022) 20572.

[9] J. García, J. Poveda, O. Villasante, P. Sánchez, A. Mateos, E. Vicente, On-line platform for the short-term prediction of risk of expansion of epidemics: proof-of-concept based on COVID-19 evolution, in: 14th USA/Europe Air Traffic Managment Research and Development Seminar, ATM 2021, 2021.

[10] D. Rodrıguez-Escabias, Aplicación de técnicas de visión por computador para medir el riesgo de contagio por virus en aeropuertos, Msc. final project, Universidad Politécnica de Madrid, 2023.

[11] A. Jiménez-Martín, A. Mateos, G.A. Peña, A. Moreno, A multi-objective approach to deal with internationaL air traffic opening/closing in Spain in an early stage pandemic situation, in: Proceedings of 9th International Conference on Control, Decision and Information Technologies, 2023, pp. 1062–1067.

[12] G.A. Peña, A. Mateos, A. Jiménez-Martín, R.G. Sanchis, A decision support system for risk reduction in pandemic spread based on the management of passenger air traffic, Int. Trans. Oper. Res. 32 (2025) 1893–1917.

[13] W.O. Kermack, A.G. McKendrick, A contribution to the mathematical theory of epidemics, in: Proceedings of the Royal Society of London. Series A, Containing Papers of a Mathematical and Physical Character, 115.772, 1927, pp. 700–721.

[14] M. Danielson, L. Ekenberg, Rank ordering methods for multi-criteria decisions, in: P. Zaraté, G.E. Kersten, J.E. Hernández (Eds.), Group Decision and Negotiation. A Process-Oriented View, Springer International Publishing, Cham, 2014, pp. 128–135.

[15] J. Kennedy, R. Eberhart, Particle swarm optimization, in: Proceedings of ICNN'95-International Conference on Neural Networks, 4, IEEE, 1995, pp. 1942–1948.

[16] D. Zou, L. Gao, S. Li, J. Wu, Solving 0–1 knapsack problem by a novel global harmony search algorithm, Appl. Soft Comput. 11 (2) (2011) 1556–1564.

[17] W.-L. Xiang, M.-Q. An, Y.-Z. Li, R.-C. He, J.-F. Zhang, A novel discrete global-best harmony search algorithm for solving 0–1 knapsack problems, Discrete Dyn. Nat. Soc. 2014 (1) (2014) 573731.

[18] M.A.K. Azad, A.M. A.C. Rocha, E.M. Fernandes, A simplified binary artificial fish swarm algorithm for 0–1 quadratic knapsack problems, J. Comput. Appl. Math. 259 (2014) 897–904.

[19] Y. Zhou, X. Chen, G. Zhou, An improved monkey algorithm for a 0–1 knapsack problem, Appl. Soft. Comput. 38 (2016) 817–830.

[20] H. Hakli, BinEHO: a new binary variant based on elephant herding optimization algorithm, Neural Computing and Applications 32 (22) (2020) 16971–16991.

[21] Y. Chen, W. Xie, X. Zou, A binary differential evolution algorithm learning from explored solutions, Neurocomputing 149 (2015) 1038–1047.

[22] T. Gong, A.L. Tuson, Differential evolution for binary encoding, in: Soft Computing in Industrial Applications: Recent Trends, Springer, 2007, pp. 251–262.

[23] A.R. Hota, A. Pat, An adaptive quantum-inspired differential evolution algorithm for 0–1 knapsack problem, in: 2010 Second World Congress on Nature and Biologically Inspired Computing (NaBIC), IEEE, 2010, pp. 703–708.

[24] H. Peng, Z. Wu, P. Shao, C. Deng, Dichotomous binary differential evolution for knapsack problems, Mathematical Problems in Engineering 2016 (1) (2016) 5732489.

[25] I.M. Ali, D. Essam, K. Kasmarik, Novel binary differential evolution algorithm for knapsack problems, Inf. Sci. (NY) 542 (2021) 177–194.

[26] B. Ervural, H. Hakli, A binary reptile search algorithm based on transfer functions with a new stochastic repair method for 0–1 knapsack problems, Comput. Ind. Eng. 178 (2023) 109080.

[27] J. Kennedy, R.C. Eberhart, A discrete binary version of the particle swarm algorithm, in: Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics. Computational Cybernetics and Simulation, 5, IEEE, 1997, pp. 4104–4108.

[28] S. Mirjalili, A. Lewis, S-Shaped versus V-shaped transfer functions for binary particle swarm optimization, Swarm Evol. Comput. 9 (2013) 1–14.

[29] L. Wang, X. Wang, J. Fu, L. Zhen, A novel probability binary particle swarm optimization algorithm and its application, J. Softw. 3 (9) (2008) 28–35.

[30] J.C. Bansal, K. Deep, A modified binary particle swarm optimization for knapsack problems, Appl. Math. Comput. 218 (22) (2012) 11042–11061.

[31] S. Mirjalili, H. Zhang, S. Mirjalili, S. Chalup, N. Noman, A novel U-shaped transfer function for binary particle swarm optimisation, in: Soft Computing for Problem Solving 2019: Proceedings of SocProS 2019, 1, Springer, 2020, pp. 241–259.

[32] S.-S. Guo, Wang, Jie-sheng, M.-W. Guo, Z-shaped transfer functions for binary particle swarm optimization algorithm, Comput. Intell. Neurosci. 2020 (1) (2020) 6502807.

[33] Z. Beheshti, A novel x-shaped binary particle swarm optimization, Soft comput. 25 (4) (2021) 3013–3042.

[34] Y. He, F. Zhang, S. Mirjalili, T. Zhang, Novel binary differential evolution algorithm based on Taper-shaped transfer functions for binary optimization problems, Swarm Evol. Comput. 69 (2022) 101022.

[35] M. Mafarja, I. Aljarah, A.A. Heidari, H. Faris, P. Fournier-Viger, X. Li, S. Mirjalili, Binary dragonfly optimization for feature selection using time-varying transfer functions, Knowl. Based Syst. 161 (2018) 185–204.

[36] Z. Beheshti, UTF: upgrade transfer function for binary meta-heuristic algorithms, Appl. Soft Comput. 106 (2021) 107346.

[37] Z. Beheshti, A fuzzy transfer function based on the behavior of meta-heuristic algorithm and its application for high-dimensional feature selection problems, Knowl. Based Syst. 284 (2024) 111191.

[38] M.A. Khanesar, M. Teshnehlab, M.A. Shoorehdeli, A novel binary particle swarm optimization, in: 2007 Mediterranean Conference on Control & Automation, IEEE, 2007, pp. 1–6.

[39] B.H. Nguyen, B. Xue, P. Andreae, A novel binary particle swarm optimization algorithm and its applications on knapsack and feature selection problems, in: Intelligent and Evolutionary Systems: The 20th Asia Pacific Symposium (IES 2016), Springer, 2017, pp. 319–332.

[40] Y. Zhang, S. Wang, P. Phillips, G. Ji, Binary PSO with mutation operator for feature selection using decision tree applied to spam detection, Knowl. Based Syst. 64 (2014) 22–31.

[41] A.R. Jordehi, A review on constraint handling strategies in particle swarm optimisation, Neural Comput. Appl. 26 (2015) 1265–1275.

[42] X. Li, W. Fang, S. Zhu, An improved binary quantum-behaved particle swarm optimization algorithm for knapsack problems, Inf. Sci. 648 (2023) 119529.

[43] Y. Shi, R.C. Eberhart, Empirical study of particle swarm optimization, in: Proceedings of the 1999 Congress on Evolutionary Computation-CEC99 (Cat. No. 99TH8406), 3, IEEE, 1999, pp. 1945–1950.

[44] A. Ratnaweera, S.K. Halgamuge, H.C. Watson, Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients, IEEE Trans. Evol. Comput. 8 (3) (2004) 240–255.

[45] A.P. Engelbrecht, Computational intelligence: an introduction, Wiley, 2007. https://books.google.es/books?id=IZosIcgJMjUC.

[46] A.A. Heidari, S. Mirjalili, H. Faris, I. Aljarah, M. Mafarja, H. Chen, Harris hawks optimization: algorithm and applications, Future Gener. Comput. Syst. 97 (2019) 849–872.

[47] S. Mirjalili, A.H. Gandomi, S.Z. Mirjalili, S. Saremi, H. Faris, S.M. Mirjalili, Salp swarm algorithm: a bio-inspired optimizer for engineering design problems, Adv. Eng. Softw. 114 (2017) 163–191.

[48] S. Mirjalili, SCA: a sine cosine algorithm for solving optimization problems, Knowl. Based Syst. 96 (2016) 120–133.

[49] S. Mirjalili, A. Lewis, The whale optimization algorithm, Adv. Eng. Softw. 95 (2016) 51–67.

[50] M. Abdel-Basset, R. Mohamed, R.K. Chakrabortty, M. Ryan, S. Mirjalili, New binary marine predators optimization algorithms for 0–1 knapsack problems, Comput. Ind. Eng. 151 (2021) 106949.

[51] S. Kirkpatrick, C.D. Gelatt, Jr, M.P. Vecchi, Optimization by simulated annealing, Science 220 (4598) (1983) 671–680.

[52] A.E. Ezugwu, V. Pillay, D. Hirasen, K. Sivanarain, M. Govender, A comparative study of meta-heuristic optimization algorithms for 0–1 knapsack problem: some initial results, IEEE Access 7 (2019) 43979–44001.

[53] T.A. Feo, M.G.C. Resende, Greedy randomized adaptive search procedures, J. Global Optim. 6 (1995) 109–133.

[54] F.A. Hashim, K. Hussain, E.H. Houssein, M.S. Mabrouk, W. Al-Atabany, Archimedes optimization algorithm: a new metaheuristic algorithm for solving optimization problems, Appl. Intell. 51 (2021) 1531–1551.

[55] R.V. Rao, V.J. Savsani, D.P. Vakharia, Teaching–learning-based optimization: a novel method for constrained mechanical design optimization problems, Comput.-Aided Des. 43 (3) (2011) 303–315.

[56] H. Shayanfar, F.S. Gharehchopogh, Farmland fertility: a new metaheuristic algorithm for solving continuous optimization problems, Appl. Soft. Comput. 71 (2018) 728–746.

[57] S. Kaur, L.K. Awasthi, A.L. Sangal, G. Dhiman, Tunicate swarm algorithm: a new bio-inspired based metaheuristic paradigm for global optimization, Eng. Appl. Artif. Intell. 90 (2020) 103541.

[58] B. Abdollahzadeh, S. Barshandeh, H. Javadi, N. Epicoco, An enhanced binary slime mould algorithm for solving the 0–1 knapsack problem, Eng. Comput. 38 (2021) 3423–3444.

[59] G. Atali, İ. Pehlivan, B. Gürevin, H.İ. Şeker, Chaos in metaheuristic based artificial intelligence algorithms: a short review, Turk. J. Electr. Eng. Comput. Sci. 29 (3) (2021) 1354–1367.

[60] S. Xia, X. Bai, G. Wang, Y. Cheng, D. Meng, X. Gao, Y. Zhai, E. Giem, An efficient and accurate rough set for feature selection, classification, and knowledge representation, IEEE Trans. Knowl. Data Eng. 35 (8) (2022) 7724–7735.

[61] J. Yang, Z. Liu, G. Wang, Q. Zhang, S. Xia, D. Wu, Y. Liu, Constructing three-way decision with fuzzy granular-ball rough sets based on uncertainty invariance, IEEE Trans. Fuzzy Syst. 33 (6) (2025) 1781–1792.