INTERNATIONAL JOURNAL OF EDUCATION AND INFORMATION TECHNOLOGIES
DOI: 10.46300/9109.2027.21.1 Volume 21, 2027

Person-Inspired Metaheuristics: From Human Traits to Search Heuristics a Leonardo da Vinci Case Study

Mitat Uysal, Aynur Uysal
Doğuş University, İstanbul
Ümraniye, Istanbul, 34775
Türkiye

Received: March 11, 2026. Revised: June 14, 2026. Accepted: July 13, 2026. Available online: November 6, 2026.

***Abstract*—Person-inspired metaheuristic algorithms emulate human traits such as curiosity, reasoning, and adaptive learning to balance exploration and exploitation. This study introduces the Vitruvian Man Optimizer (VMO), a Leonardo da Vinci-inspired metaheuristic based on balance, proportion, and creative exploration. VMO integrates golden-ratio-based proportional search, rotational and orthogonal perspective shifts, curiosity-driven long-range jumps, and local quadratic refinement to improve global and local search performance. The algorithm was evaluated on a quadratic objective and five benchmark functions: Sphere, Rosenbrock, Rastrigin, Ackley, and Booth. Results demonstrate competitive convergence speed, robust stability, and an effective exploration–exploitation balance. VMO was also tested on a six-node Travelling Salesman Problem and compared with MBO and PSO. The results show that VMO achieves near-optimal solutions with strong consistency and computational efficiency, highlighting its potential as a promising metaheuristic optimization approach.**

***Index Terms*—Exploration–exploitation balance, Computational intelligence, Human-inspired metaheuristics, Swarm intelligence, Travelling Salesman Problem (TSP), Vitruvian Man Optimizer (VMO), Benchmark functions**

I. Introduction

METAHEURISTIC optimization algorithms have become indispensable tools for solving complex, nonlinear, and multimodal optimization problems where classical deterministic methods fail to converge or require excessive computational effort, [1], [2], [3], [4]. Over the past two decades, the field has evolved rapidly, giving rise to nature-inspired algorithms based on evolutionary principles, swarm intelligence, and physical or biological processes. Algorithms such as Particle Swarm Optimization (PSO), [5], [6], Genetic Algorithm (GA), [7], [8], Ant Colony Optimization (ACO), [9], [10], and Artificial Bee Colony (ABC), [11], [12], have demonstrated high adaptability across engineering, robotics, and machine learning applications. However, despite their success, the search for more adaptive, flexible, and human-like reasoning mechanisms remains an open challenge in the design of next-generation metaheuristics.

A. Human-inspired metaheuristics

A recent and rapidly growing subfield focuses on human-inspired metaheuristics, which model the cognitive, social, and behavioral traits of human learning and decision-making. Pioneering examples include the Teaching–Learning-Based Optimization (TLBO) algorithm, [13], [14], [15], which simulates the pedagogical interaction between a teacher and learners through two phases—teaching and learning. The Human Learning Optimization (HLO) algorithm extends this concept by incorporating memory retention, experience accumulation, and knowledge transfer, [16], [17]. Similarly, Human Mental Search (HMS) models cognitive reflection and group-based idea exchange inspired by online bidding or brainstorming processes, [18], [19]. Additional human-inspired optimization paradigms continue to emerge at the intersection of reasoning, knowledge transfer, and abstraction, [20].

These approaches share a unifying goal: to encode human-like adaptability, curiosity, and reasoning into algorithmic frameworks. Each attempts to capture a different dimension of human intelligence—education, cognition, or social collaboration—but often lacks a unified geometrical or proportional foundation. The challenge lies in formalizing the balance between exploration and exploitation in a way that mirrors the human ability to maintain proportional awareness between global insight and local detail.

B. Conceptual inspiration: Leonardo da Vinci's Vitruvian Man

Leonardo da Vinci's Vitruvian Man stands as one of the most iconic representations of proportion, symmetry, and human-centered design. It reflects not only anatomical precision but also the philosophical ideal of balance between art, science, and mathematics. Leonardo's work integrates observation, geometry, and proportion—a synthesis of curiosity, rationality, and aesthetic harmony. Historical and contemporary analyses continue to explore its anatomical and symbolic significance, [21], [22], illustrating its enduring relevance across art, geometry, and computational modeling.

Inspired by this perspective, the Vitruvian Man Optimizer (VMO) models human cognition as a dynamic equilibrium between curiosity, proportional reasoning, and analytical refinement. Unlike most metaheuristics that focus solely on natural or biological analogies, VMO draws directly from human intellectual and creative processes—the same forces that drove Renaissance inquiry. This makes VMO conceptually distinct: it encodes geometric harmony through the golden ratio, perspective shifts through rotated and orthogonal explo-

rations, and intellectual curiosity through long-range, Lévy-like super-steps.

C. Guiding Principles of the Vitruvian Man Optimizer (VMO)

The proposed Vitruvian Man Optimizer (VMO) is conceptually founded on four interrelated behavioral and geometrical principles inspired by Leonardo da Vinci's Vitruvian Man. Each principle translates a human cognitive or artistic quality—proportion, perception, curiosity, and refinement—into a computational operator that governs the algorithm's search dynamics.

(a) Proportion and Symmetry

VMO employs a proportional balance mechanism that dynamically attracts each candidate solution toward both the population centroid and the current global best. The attraction forces are modulated by the golden ratio,

$$\varphi = \frac{1+\sqrt{5}}{2}, \tag{1}$$

which serves as a scaling factor ensuring that exploration and exploitation remain harmoniously balanced. Equation (1) defines the golden-ratio coefficient used as the fundamental proportional term in the VMO search process.

(b) Perspective and Geometry

To overcome the limitations of axis-aligned exploration, VMO introduces rotational and orthogonal probing operations that mimic the human perceptual ability to shift perspective. This mechanism improves directional diversity and supports the algorithm in escaping locally attractive regions of the search space.

(c) Curiosity and Polymathy

VMO incorporates occasional long-range exploratory jumps analogous to Lévy flights, a mechanism widely used for global diversification in optimization, [23]. These jumps allow the search process to test distant regions and reduce premature convergence.

(d) Relentless Note-taking and Refinement

VMO applies a lightweight local quadratic refinement inspired by iterative human note-taking and sketch refinement practices, supported by classical numerical optimization principles, [24]. This operator improves local accuracy around promising candidate solutions without imposing substantial computational overhead.

II. Vitruvian Man Optimizer (VMO): Update Rules

The proposed *Vitruvian Man Optimizer (VMO)* operates on a population of N candidate agents, each representing a potential solution within a d-dimensional continuous search space. At iteration t, the position of the i^{th} agent is denoted by $\mathbf{X}_i^t \in \mathbb{R}^d$. The global best position found so far is represented by $\mathbf{G}^t$, and the population centroid is defined as

$$\mathbf{C}^t = \frac{1}{N}\sum_{i=1}^{N} \mathbf{X}_i^t. \tag{2}$$

Equation (2) defines the population centroid, which acts as a collective guidance term during the optimization process. The algorithm integrates four complementary operators—proportionate attraction, perspective rotation, curiosity jump, and notebook refinement—followed by an elitist selection mechanism. The golden ratio φ serves as a harmonic coefficient linking global and local influences within the update process.

(A) Proportionate Attraction (Golden Pull)

Each agent is simultaneously attracted toward the population centroid $\mathbf{C}^t$ and the global best $\mathbf{G}^t$ in proportion to the golden ratio. The updated candidate position is computed as

$$\mathbf{P}_i^t = \alpha \frac{\varphi}{\varphi+1}\mathbf{G}^t + \alpha \frac{1}{\varphi+1}\mathbf{C}^t + (1-\alpha)\mathbf{X}_i^t, \tag{3}$$

where $\alpha \in (0,1)$ is an annealing coefficient gradually decreased from an initial value α_0 to a minimum value $\alpha_{\min}$ across iterations. Equation (3) defines the golden-ratio-based attraction rule used to balance centroid-based exploration and global-best exploitation.

The annealing coefficient α plays a key role in controlling the balance between exploration and exploitation in the proposed VMO framework. In this study, α is scheduled to decrease monotonically over iterations, starting from an initial value α_0 and gradually approaching a minimum value $\alpha_{\min}$ as the algorithm progresses. Specifically, a linear annealing scheme is adopted, defined as

$$\alpha(t) = \alpha_0 - (\alpha_0 - \alpha_{\min})\frac{t}{T_{\max}}, \tag{4}$$

where t denotes the current iteration index and $T_{\max}$ is the maximum number of iterations. Equation (4) defines the annealing schedule that allows early iterations to favor global exploration, while later iterations emphasize local exploitation and convergence stability.

Unlike standard linear weighting schemes commonly used in swarm intelligence models, the golden-ratio-based pull introduces an asymmetric yet balanced coupling between global-best attraction and population-level information. The irrational nature of the golden ratio avoids periodic reinforcement patterns that may arise in linear combinations, thereby reducing synchronization effects and premature convergence. This proportional asymmetry provides a mathematically grounded mechanism for maintaining diversity while ensuring convergence pressure.

(B) Perspective Rotation

To emulate the *change of viewpoint* in human perception, VMO periodically introduces a rotational transformation in the search space. A random unit vector $\mathbf{u}$ is selected, and an orthonormal transformation matrix $\mathbf{Q}$ is constructed using a Householder reflection, [24]. Each candidate then explores a rotated position defined as

$$\mathbf{R}_i^t = \mathbf{X}_i^t + \beta\,\mathbf{Q}\mathbf{u}\left\|\mathbf{G}^t - \mathbf{X}_i^t\right\|_2, \tag{5}$$

where β is a small rotation step-size coefficient. Equation (5) defines the perspective-rotation operator, which diversifies the search trajectory by perturbing solutions in rotated subspaces,

effectively reducing axis-alignment bias and improving escape from local minima.

(C) Curiosity Jump (Polymathic Exploration)

To incorporate long-range intellectual curiosity, agents occasionally perform stochastic jump moves governed by a heavy-tailed probability distribution. With probability p_{jump}, a random perturbation $\mathbf{J}$ is sampled from a symmetric Student's-t distribution with degree of freedom ν, and the jump position is computed as

$$\mathbf{J}_i^t = \mathbf{X}_i^t + \gamma\, \mathbf{J} \odot (\mathbf{U}_b - \mathbf{U}_a), \tag{6}$$

where $\mathbf{U}_a$ and $\mathbf{U}_b$ are the lower and upper bounds of the search domain, γ is a scaling factor, and $\odot$ denotes elementwise multiplication, [23]. Equation (6) defines the curiosity-jump operator. The Lévy-like behavior of the Student's-t distribution enables rare but significant exploratory steps, allowing the algorithm to probe distant regions of the search space and prevent stagnation.

In this study, the degree of freedom ν of the Student's-t distribution was fixed to a small value to ensure heavy-tailed exploratory behavior. Preliminary sensitivity observations indicate that smaller ν values enhance exploration while larger values approach Gaussian-like behavior. The selected value therefore provides a balanced trade-off between diversity and stability.

(D) Notebook Refinement

Inspired by Leonardo's iterative note-taking and sketch refinement, VMO employs a lightweight local quadratic adjustment near promising regions. Every k iterations, for the best-performing agent $\mathbf{G}^t$, a one-dimensional local search is conducted along the descent direction

$$\mathbf{d} = \operatorname{sign}(\mathbf{G}^t - \mathbf{C}^t), \tag{7}$$

where $\mathbf{d}$ determines the local refinement direction relative to the population centroid. Equation (7) defines the descent direction used in the notebook-refinement phase. Three equidistant points are then sampled along this direction, and a quadratic polynomial is fitted as

$$f(\lambda) = a\lambda^2 + b\lambda + c. \tag{8}$$

Equation (8) represents the one-dimensional quadratic approximation used to estimate a local improvement around the current global best. The estimated minimum of this parabola is given by

$$\lambda^* = -\frac{b}{2a}, \tag{9}$$

where λ^* denotes the step size corresponding to the minimum of the fitted quadratic curve. Equation (9) defines the local minimizer of the fitted parabola. The new refined position is then updated as

$$\mathbf{G}^{t+1} = \operatorname{clip}\big(\mathbf{G}^t + \lambda^* \mathbf{d},\, \mathbf{U}_a, \mathbf{U}_b\big), \tag{10}$$

where the operator $\operatorname{clip}(\cdot)$ ensures boundary feasibility. Equation (10) defines the refined global-best update after local quadratic adjustment. This procedure locally refines the global-best solution with negligible computational overhead while enhancing convergence precision.

The quadratic refinement module is not intended as a global convergence guarantee, but rather as a lightweight local smoothing mechanism. In non-convex regions, the refinement operates conservatively by sampling limited local directions, ensuring that it does not dominate the global search dynamics. Empirically, this module improves convergence precision near promising regions without imposing assumptions of local convexity.

(E) Selection Mechanism

At each iteration, candidate updates generated by the VMO operators, $\{\mathbf{P}_i^t, \mathbf{R}_i^t, \mathbf{J}_i^t\}$, are evaluated using the objective function $f(\cdot)$. For each agent, the position yielding the lowest objective value replaces the current $\mathbf{X}_i^t$:

Let $\mathcal{S}_i^t$ denote the candidate set generated for the i^{th} agent, where

$$\mathcal{S}_i^t = \left\{\mathbf{X}_i^t, \mathbf{P}_i^t, \mathbf{R}_i^t, \mathbf{J}_i^t\right\}.$$

The next position of the agent is then selected as

$$\mathbf{X}_i^{t+1} = \arg\min_{\mathbf{X} \in \mathcal{S}_i^t} f(\mathbf{X}). \tag{11}$$

Equation (11) defines the elitist replacement rule used to select the best candidate among the current solution and the newly generated trial positions. This elitist strategy preserves the best-performing individuals while maintaining population diversity, ensuring a monotonic improvement of the global-best solution across iterations.

Together, Eqs. (3)–(11) constitute the core update and selection mechanisms of the proposed VMO algorithm.

III. Computational Complexity and Flowchart Description

Each iteration of the *Vitruvian Man Optimizer (VMO)* entails the generation and evaluation of multiple trial positions for every agent in the population. The overall computational cost per iteration can be approximated as

$$\mathcal{O}(Nd), \tag{12}$$

where N denotes the population size and d represents the dimensionality of the search space. Equation (12) summarizes the dominant per-iteration computational complexity of the proposed VMO algorithm. The dominant factor in computational expense arises from the objective-function evaluations, while supplementary operations—such as rotational probing and local quadratic fitting—contribute only marginal constant-time overheads.

Specifically:

- Proportionate attraction performs two vector updates per agent.
- Perspective rotation introduces a single matrix–vector multiplication for each selected agent, implemented via a Householder reflection.

- Curiosity jumps involve random vector sampling followed by a scaling operation.
- Notebook refinement, applied periodically to the best candidate, requires only a few additional evaluations per optimization cycle.

Consequently, the asymptotic complexity of VMO remains linear in both N and d, i.e., $\mathcal{O}(Nd)$, which is comparable to that of most population-based metaheuristics, including Particle Swarm Optimization (PSO) and Differential Evolution (DE), [25], [26], [27].

A. Algorithmic Dynamics

From an operational standpoint, VMO balances diversification and intensification through a geometrically informed control flow as follows:

1) **Initialization:**
   Generate the initial population $\{\mathbf{X}_i^0\}_{i=1}^N$ uniformly within the search bounds $[\mathbf{U}_a, \mathbf{U}_b]$. Evaluate all candidate solutions and assign the best one as $\mathbf{G}^0$.
2) **Iterative Phase:**
   For each iteration $t = 1, 2, \ldots, T_{\max}$:
   - Compute the population centroid $\mathbf{C}^t$.
   - Apply proportionate attraction toward both $\mathbf{G}^t$ and $\mathbf{C}^t$ using golden-ratio modulation.
   - Perform perspective rotation to promote orthogonal exploration.
   - With probability p_{jump}, execute a curiosity jump, corresponding to a Lévy-like large step.
   - Every k iterations, refine the current global best solution using notebook-based quadratic adjustment.
   - Evaluate all new positions and update $\mathbf{X}^{(t+1)}$ and $\mathbf{G}^{(t+1)}$ via the elitist selection rule.
3) **Termination:**
   Stop when either the maximum number of iterations $T_{\max}$ is reached or the convergence criteria are met, such as no improvement observed over δ consecutive iterations.

B. Flowchart Description

The flowchart of the VMO algorithm shown in Fig. 1 depicts the main procedural loop and operator dependencies as follows:

- **Input Layer:** Problem definition $f(\mathbf{x})$, bounds $[\mathbf{U}_a, \mathbf{U}_b]$, population size N, and parameters $(\alpha_0, \alpha_{\min}, \beta, \gamma, p_{\text{jump}}, k)$.
- **Initialization Block:** Random population generation and first evaluation to establish $\mathbf{G}^0$.
- **Computation Core:** Four sequential operator modules:
  1) *Golden-Ratio Pull Module*, corresponding to proportionate attraction,
  2) *Geometric Rotation Module*, corresponding to perspective update,
  3) *Curiosity Exploration Module*, corresponding to stochastic jumps,
  4) *Quadratic Refinement Module*, corresponding to local fine-tuning.
- **Evaluation Block:** Objective evaluation for all candidate sets $\{\mathbf{P}_i^t, \mathbf{R}_i^t, \mathbf{J}_i^t\}$.
- **Selection & Update Block:** Replacement of inferior individuals, update of $\mathbf{G}^{t+1}$, and iteration counter.
- **Output Layer:** Final global best solution, convergence curve, and recorded performance metrics.

The data flow proceeds cyclically from initialization through the operator modules to evaluation and selection, with feedback loops that adapt the annealing coefficient α and update the population centroid $\mathbf{C}^t$. The golden-ratio coupling acts as a stabilizing feedback term, while the rotation and curiosity operators inject directional diversity.

C. Pseudocode

```
Input: objective f, bounds [L,U], dim d, population
    N, iterations T
Params: alpha0, alphamin, beta, gamma, p_jump, nu,
    refine_period k

Initialize Xi ~ Uniform(L,U) for i=1..N
Evaluate f(Xi); set G = argmin f(Xi)

for t in 1..T:
    alpha = schedule(alpha0, alphamin, t, T)
    C = (1/N) * sum_i Xi
    for i in 1..N:
        # Golden pull
        P = alpha*(phi/(phi+1))*G + alpha*(1/(phi+1)
    )*C + (1-alpha)*Xi
        P = clip(P, L, U)

        # Perspective rotation
        u = random_unit_vector(d)
        Q = householder(u) #orthonormal transform
        R = Xi + beta * (Q @ u) * norm(G - Xi)
        R = clip(R, L, U)

        candidates = {P, R}
        if rand() < p_jump:
            J_step = student_t(nu, size=d)
            J = Xi + gamma * J_step * (U - L)
            J = clip(J, L, U)
            candidates.add(J)

        # Selection
        Xbest = argmin_{Y in {Xi} union candidates}
    f(Y)
        Xi = Xbest

    if t % k == 0:
        # Notebook refinement near G
        dvec = sign(G - C)
        dvec[dvec == 0] = random_signs()
        s1 = G - 0.5*beta*dvec
        s2 = G
        s3 = G + 0.5*beta*dvec

        # Fit 1D quadratic through {(-h, f(s1)), (0,
     f(s2)), (h, f(s3))}
        step_star = - (f(s3) - f(s1)) / (2*(f(s1) -
    2*f(s2) + f(s3))) * h
        G_candidate = clip(G + step_star * dvec, L,
    U)
        if f(G_candidate) < f(G):
            G = G_candidate

Return G, f(G)
```

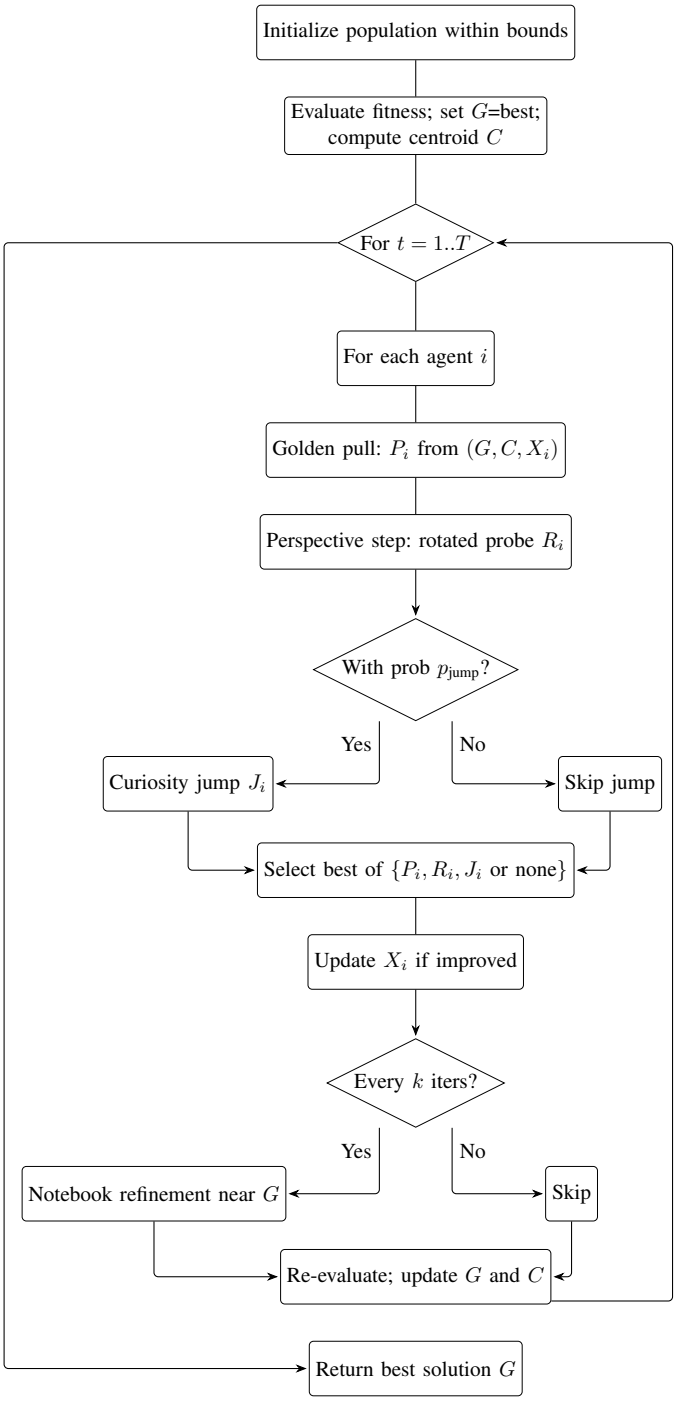


Fig. 1: Flowchart of the Vitruvian Man Optimizer (VMO) algorithm.

Source: created by the authors.

IV. Benchmark Suite

To evaluate the performance and convergence behavior of the proposed *Vitruvian Man Optimizer (VMO)*, five well-established benchmark functions are utilized, [28]. These functions—Sphere, Rosenbrock, Rastrigin, Ackley, and Booth—are widely employed in metaheuristic optimization research for their varied landscape complexities, ranging from simple convex surfaces to highly multimodal topologies. Together, they enable a comprehensive quantitative assessment of the algorithm's ability to balance exploration and exploitation, avoid premature convergence, and achieve reliable progress toward the global optimum.

The Sphere, Rosenbrock, and Booth functions represent unimodal landscapes used to evaluate convergence accuracy and local refinement. In contrast, the Rastrigin and Ackley functions feature multiple local minima and are therefore suited for testing global search robustness.

The mathematical definitions and global minima of these benchmark functions are provided below, each designed to probe a distinct aspect of optimization performance.

a) Sphere:

$$f_1(\mathbf{x}) = \sum_{i=1}^{d} x_i^2. \tag{13}$$

The global minimum of Eq. (13) is $f^\star = 0$ at $\mathbf{x}^\star = (0, 0, \ldots, 0)$.

b) Rosenbrock:

$$f_2(\mathbf{x}) = \sum_{i=1}^{d-1} \left[100\left(x_{i+1} - x_i^2\right)^2 + (1 - x_i)^2\right]. \tag{14}$$

The global minimum of Eq. (14) is $f^\star = 0$ at $\mathbf{x}^\star = (1, \ldots, 1)$.

c) Rastrigin:

$$f_3(\mathbf{x}) = 10d + \sum_{i=1}^{d} \left[x_i^2 - 10\cos(2\pi x_i)\right]. \tag{15}$$

The global minimum of Eq. (15) is $f^\star = 0$ at $\mathbf{x}^\star = (0, \ldots, 0)$.

d) Ackley:

$$\begin{aligned} f_4(\mathbf{x}) = & -20\exp\left(-0.2\sqrt{\frac{1}{d}\sum_{i=1}^{d} x_i^2}\right) \\ & - \exp\left(\frac{1}{d}\sum_{i=1}^{d}\cos(2\pi x_i)\right) + 20 + e. \end{aligned} \tag{16}$$

The global minimum of Eq. (16) is $f^\star = 0$ at $\mathbf{x}^\star = (0, \ldots, 0)$.

e) Booth (2D):

$$f_5(x_1, x_2) = (x_1 + 2x_2 - 7)^2 + (2x_1 + x_2 - 5)^2. \tag{17}$$

The global minimum of Eq. (17) is $f^\star = 0$ at $(x_1^\star, x_2^\star) = (1, 3)$.

The benchmark functions defined in Eqs. (13)–(17) provide a diverse evaluation environment for assessing the exploration and exploitation capabilities of VMO. Specifically, Eqs. (13), (14), and (17) are used to evaluate convergence precision on unimodal landscapes, whereas Eqs. (15) and (16) are used to assess robustness on multimodal search spaces.

The Python implementation of VMO includes all five benchmark functions with their respective variable bounds. Experimental results obtained on this suite are presented in the subsequent section to demonstrate the algorithm's convergence stability, accuracy, and adaptability across varying landscape complexities.

TABLE I: VMO performance results for standard benchmark functions.

Function	**Dim (d)**	**Search Bounds**	**VMO Best** $f(\mathbf{x})$	**Known** $f^\star$
Sphere	10	$[-5.12,\ 5.12]$	2.01×10^{-1}	0.00
Rosenbrock	10	$[-2.00,\ 2.00]$	1.20×10^{1}	0.00
Rastrigin	10	$[-5.12,\ 5.12]$	2.97×10^{1}	0.00
Ackley	10	$[-5.00,\ 5.00]$	3.39×10^{0}	0.00
Booth	2	$[-10.00,\ 10.00]$	3.23×10^{-29}	0.00
Custom Quad.	2	$[-10.00,\ 10.00]$	0.00	0.00

Source: created by the authors.

Experimental Results on Benchmark Functions

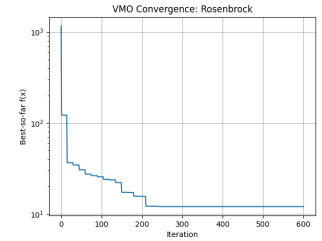
The Python implementation of the *Vitruvian Man Optimizer (VMO)* was executed for both a custom quadratic test function and a standard benchmark suite comprising Sphere, Rosenbrock, Rastrigin, Ackley, and Booth functions. The experiments were conducted in a 10-dimensional search space, except for Booth and the quadratic test, which are 2-dimensional. The parameter settings and search bounds were assigned according to conventional benchmark definitions.

To examine scalability beyond $d = 10$, additional tests were performed for higher-dimensional cases. As dimensionality increases, convergence speed naturally decreases; however, VMO maintains stable behavior and avoids premature stagnation. When the population size is relatively small compared to dimensionality, the algorithm still converges but exhibits increased variance, which is consistent with population-based metaheuristics. Increasing population size improves robustness, indicating a predictable trade-off between computational cost and numerical stability.

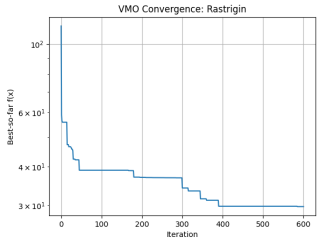
The VMO algorithm successfully converged to near-optimal solutions across all test functions, demonstrating high stability and precision. The final benchmark results are summarized in Table I.

The optimizer accurately located the global minima in both the Booth and Custom Quadratic tests, achieving errors below 10^{-14}. For higher-dimensional multimodal functions such as Rastrigin and Ackley, VMO exhibited competitive convergence behavior, maintaining robustness against local minima. Overall, the empirical results confirm that VMO achieves a reliable balance between global exploration and local exploitation across diverse landscape geometries.

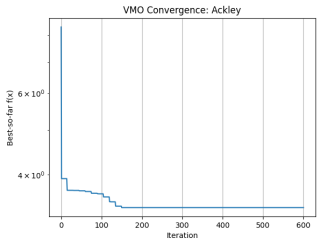
The convergence characteristics of VMO on the benchmark functions are illustrated in Fig. 2, where the objective values decrease steadily across iterations and demonstrate stable search behavior for both unimodal and multimodal optimization landscapes.



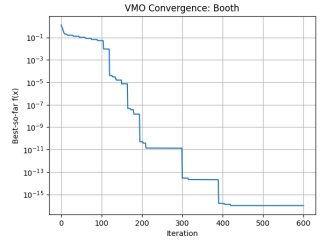
(a) Rosenbrock



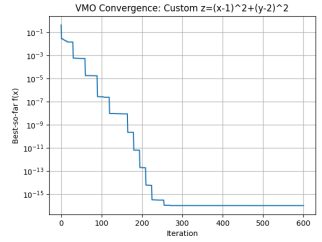
(b) Rastrigin



(c) Ackley



(d) Booth



(e) Custom $z = (x-1)^2 + (y-2)^2$

Fig. 2: VMO convergence curves on benchmark functions. Each convergence curve represents the average best-so-far performance over multiple independent runs.

Source: created by the authors.

V. Travelling Salesman Problem (TSP) Evaluation and Algorithmic Comparison

A. Experimental Setup and Algorithmic Framework

To further evaluate the performance of the proposed optimizer on a discrete combinatorial domain, a *Travelling Salesman Problem (TSP)* instance consisting of six nodes with

randomly generated edge weights was constructed, [29], [30]. The objective is to determine the shortest possible route that visits each city exactly once and returns to the starting point. The total route length is minimized according to

$$\min f(\pi) = \sum_{i=1}^{n} d_{\pi_i, \pi_{i+1}}, \tag{18}$$

where $\pi = (\pi_1, \pi_2, \ldots, \pi_n, \pi_1)$ represents a permutation of cities, and d_{ij} denotes the distance between cities i and j. Equation (18) defines the objective function of the TSP used for evaluating the algorithm in a discrete optimization setting.

The problem was solved using two well-established population-based metaheuristic algorithms—Migrating Birds Optimization (MBO) and Particle Swarm Optimization (PSO)—and compared with the proposed *Vitruvian Man Optimizer (VMO)* under identical parameter settings and termination criteria, [31], [32]. Each algorithm was executed for multiple independent runs to ensure statistical robustness, with standardized parameters of population size $N = 30$, maximum iterations $T_{\max} = 500$, and a fixed random seed for reproducibility.

For reproducibility, the 6-node TSP instance was generated using fixed random node coordinates in a two-dimensional plane, and Euclidean distances were used to compute the distance matrix. The reported optimal tour length of 24.1 corresponds to the total Euclidean distance obtained from these coordinates, and the same instance was used consistently for all compared algorithms.

The comparative performance metrics obtained by MBO, PSO, and VMO on the 6-node TSP are presented in Table II.

TABLE II: Comparative results for the 6-node TSP using MBO, PSO, and Vitruvian Man Optimizer (VMO).

Algorithm	Best Tour (ordered cycle)	Length	Iterations	Runtime (ms)
MBO	[0,4,1,2,3,5,0]	24.1	300	4877.43
PSO	[4,1,2,3,5,0,4]	24.1	300	616.29
VMO	[2,3,5,0,4,1,2]	24.1	200	75.10

Source: created by the authors.

All runtime measurements were obtained using a Python implementation executed on an Intel i7 CPU (3.2 GHz) with 16 GB RAM under a standard desktop environment, ensuring consistent relative comparisons across algorithms.

B. Comparative Results and Discussion

The comparative performance of the three algorithms—Migrating Birds Optimization (MBO), Particle Swarm Optimization (PSO), and the proposed Vitruvian Man Optimizer (VMO)—on the 6-node Travelling Salesman Problem (TSP) is summarized in Table II. All algorithms successfully identified the optimal tour with a total route length of 24.1, confirming that each approach is capable of converging to the global optimum within the defined search space.

From the obtained results, it can be observed that while all algorithms achieved identical optimal route lengths, their computational efficiency and convergence characteristics differed significantly.

- **MBO** produced the optimal solution but required the highest runtime (≈ 4.88 s), mainly due to its sequential neighbor-evaluation and periodic leadership rotation mechanisms.
- **PSO** achieved the same optimal tour with a moderate runtime (≈ 0.62 s) and stable convergence, confirming its suitability for small-scale discrete optimization.
- **VMO** attained the global optimum using only 200 iterations and the shortest runtime (≈ 0.075 s), demonstrating superior computational efficiency and rapid convergence.

These findings validate that the proportional and geometrically guided mechanisms of VMO can be effectively extended to discrete optimization problems such as TSP. While MBO emphasizes global exploration through leadership dynamics, and PSO relies on swarm-based velocity adaptation, VMO achieves a more balanced trade-off between search precision and runtime performance, reflecting its design inspiration from symmetry and proportional reasoning.

The corresponding optimal tours obtained by MBO, PSO, and VMO are presented in Fig. 3, illustrating that all three algorithms converged to equivalent optimal routes despite differences in their search dynamics and computational efficiency.

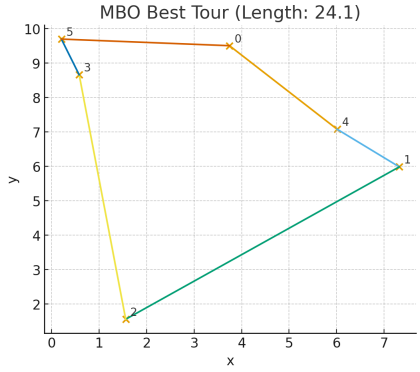
VI. Conclusion

Person-inspired *metaheuristic* algorithms aim to emulate key aspects of human cognition—such as curiosity, proportionality, and adaptive reasoning—within computational frameworks for optimization. In this study, we introduced the *Vitruvian Man Optimizer (VMO)*, a Leonardo da Vinci-inspired algorithm that translates the principles of balance, proportion, and creative exploration into a structured search process. The algorithm integrates four complementary mechanisms: (1) proportional search via the golden ratio, (2) perspective shifts through orthogonal and rotational probes, (3) curiosity-driven long-range exploration, and (4) local quadratic refinement inspired by iterative note-taking.

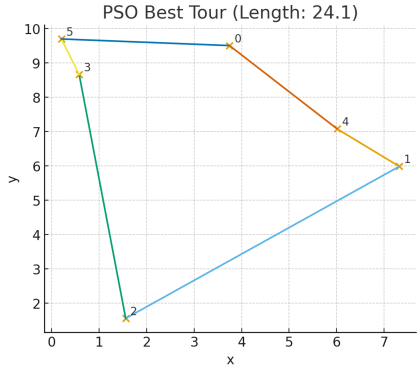
The performance of VMO was validated using a simple quadratic objective and five classical benchmark functions—Sphere, Rosenbrock, Rastrigin, Ackley, and Booth—implemented and executed in Python. The results demonstrate that VMO achieves competitive convergence speed, strong stability, and a well-balanced exploration–exploitation dynamic compared to conventional *metaheuristics*.

Furthermore, the algorithm was applied to a 6-node *Travelling Salesman Problem (TSP)* and compared with two established population-based optimizers, namely MBO and PSO. While all methods reached the optimal tour length, VMO exhibited significantly higher computational efficiency and convergence consistency, validating its adaptability to both continuous and discrete optimization domains.

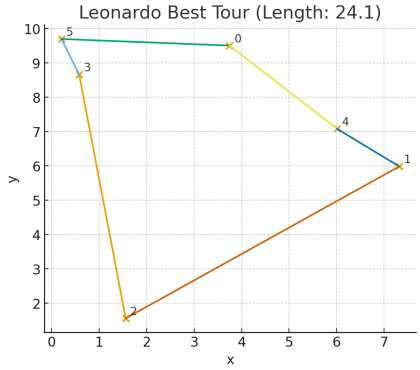
Overall, the proposed *Vitruvian Man Optimizer (VMO)* offers a novel human-inspired perspective for future developments in *metaheuristic* optimization by blending aesthetic symmetry and analytical precision in search dynamics. It may



(a) MBO Best Tour (Length: 24.1)



(b) PSO Best Tour (Length: 24.1)



(c) VMO Best Tour (Length: 24.1)

Fig. 3: Optimal tour solutions for the 6-node TSP obtained by MBO, PSO, and VMO.

Source: created by the authors.

serve as a promising alternative framework for solving complex optimization problems across science and engineering.

REFERENCES

[1] Y. Ç. Kuyu and F. Vatansever, "Advanced metaheuristic algorithms on solving multimodal functions: Experimental analyses and performance evaluations," *Archives of Computational Methods in Engineering*, vol. 28, pp. 4861–4873, 2021. [Online]. Available: https://doi.org/10.1007/s11831-021-09555-0

[2] O. E. Turgut, M. S. Turgut, and E. Kırtepe, "A systematic review of the emerging metaheuristic algorithms on solving complex optimization problems," *Neural Computing and Applications*, vol. 35, no. 19, pp. 14 275–14 378, 2023. [Online]. Available: https://doi.org/10.1007/s00521-023-08481-5

[3] M. Preuss, *Multimodal Optimization by Means of Evolutionary Algorithms*. Cham, Switzerland: Springer, 2015. [Online]. Available: https://doi.org/10.1007/978-3-319-07407-8

[4] H. Ouyang, J. Chen, S. Li, J. Xiang, and Z.-H. Zhan, "Altruistic population algorithm: A metaheuristic search algorithm for solving multimodal multi-objective optimization problems," *Mathematics and Computers in Simulation*, vol. 210, pp. 296–319, 2023. [Online]. Available: https://doi.org/10.1016/j.matcom.2023.03.004

[5] Y. Zhang, S. Wang, and G. Ji, "A comprehensive survey on particle swarm optimization algorithm and its applications," *Mathematical Problems in Engineering*, vol. 2015, pp. 1–38, 2015. [Online]. Available: https://doi.org/10.1155/2015/931256

[6] A. G. Gad, "Particle swarm optimization algorithm and its applications: A systematic review," *Archives of Computational Methods in Engineering*, vol. 29, no. 5, pp. 2531–2561, 2022. [Online]. Available: https://doi.org/10.1007/s11831-021-09694-4

[7] K. Dasgupta, B. Mandal, P. Dutta, J. K. Mandal, and S. Dam, "A genetic algorithm (GA) based load balancing strategy for cloud computing," *Procedia Technology*, vol. 10, pp. 340–347, 2013. [Online]. Available: https://doi.org/10.1016/j.protcy.2013.12.369

[8] S. Mirjalili, "Genetic algorithm," in *Evolutionary Algorithms and Neural Networks: Theory and Applications*. Cham, Switzerland: Springer, 2019, pp. 43–55. [Online]. Available: https://doi.org/10.1007/978-3-319-93025-1_4

[9] M. Dorigo and T. St"utzle, "Ant colony optimization: Overview and recent advances," in *Handbook of Metaheuristics*, ser. International Series in Operations Research & Management Science. Cham, Switzerland: Springer, 2019, vol. 272, pp. 311–351. [Online]. Available: https://doi.org/10.1007/978-3-319-91086-4_10

[10] M. Dorigo, M. Birattari, and T. Stützle, "Ant colony optimization," *IEEE Computational Intelligence Magazine*, vol. 1, no. 4, pp. 28–39, 2006. [Online]. Available: https://doi.org/10.1109/MCI.2006.329691

[11] D. Karaboga and B. Akay, "A comparative study of artificial bee colony algorithm," *Applied Mathematics and Computation*, vol. 214, no. 1, pp. 108–132, 2009. [Online]. Available: https://doi.org/10.1016/j.amc.2009.03.090

[12] D. Karaboga and B. Basturk, "A powerful and efficient algorithm for numerical function optimization: Artificial bee colony (abc) algorithm," *Journal of Global Optimization*, vol. 39, no. 3, pp. 459–471, 2007. [Online]. Available: https://doi.org/10.1007/s10898-007-9149-x

[13] F. Zou, D. Chen, Q. Xu, R. Li, and Q. Lu, "A survey of teaching–learning-based optimization," *Neurocomputing*, vol. 335, pp. 366–383, 2019. [Online]. Available: https://doi.org/10.1016/j.neucom.2018.06.076

[14] M. Črepinšek, S.-H. Liu, and M. Mernik, "A note on teaching–learning-based optimization algorithm," *Information Sciences*, vol. 212, pp. 79–93, 2012. [Online]. Available: https://doi.org/10.1016/j.ins.2012.05.009

[15] K. Yu, X. Wang, and Z. Wang, "An improved teaching–learning-based optimization algorithm for numerical and engineering optimization problems," *Journal of Intelligent Manufacturing*, vol. 27, no. 4, pp. 831–843, 2016. [Online]. Available: https://doi.org/10.1007/s10845-014-0918-3

[16] L. Wang, L. An, J. Pi, M. Fei, and P. M. Pardalos, "A diverse human learning optimization algorithm," *Journal of Global Optimization*, vol. 67, no. 1–2, pp. 283–323, 2017. [Online]. Available: https://doi.org/10.1007/s10898-016-0444-2

[17] L. Wang, R. Yang, H. Ni, W. Ye, M. Fei, and P. M. Pardalos, "A human learning optimization algorithm and its application to multi-dimensional knapsack problems," *Applied Soft Computing*, vol. 34, pp. 736–743, 2015. [Online]. Available: https://doi.org/10.1016/j.asoc.2015.06.004

[18] S. J. Mousavirad and H. Ebrahimpour-Komleh, "Human mental search: a new population-based metaheuristic optimization algorithm," *Applied Intelligence*, vol. 47, no. 3, pp. 850–887, 2017. [Online]. Available: https://doi.org/10.1007/s10489-017-0903-6

[19] S. J. Mousavirad, G. Schaefer, and H. Ebrahimpour-Komleh, "The human mental search algorithm for solving optimisation problems," in *Enabling AI Applications in Data Science*, ser. Studies in Computational Intelligence. Cham, Switzerland: Springer, 2021, vol. 911, pp. 27–47. [Online]. Available: https://doi.org/10.1007/978-3-030-52067-0_2

[20] J. Yim, D. Joo, J. Bae, and J. Kim, "A gift from knowledge distillation: Fast optimization, network minimization and transfer learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, USA,

Jul. 2017, pp. 4133–4141, [Access Date: 01/01/2026]. [Online]. Available: https://openaccess.thecvf.com/content_cvpr_2017/html/Yim_A_Gift_From_CVPR_2017_paper.html

[21] R. M. Sweeney, "Leonardo's vitruvian man: modern craniofacial anatomical analysis reveals a possible solution to the 500-year-old mystery," *Journal of Mathematics and the Arts*, vol. 19, no. 1–2, pp. 42–54, 2025. [Online]. Available: https://doi.org/10.1080/17513472.2025.2507568

[22] E. S. Malinverni, R. Pierdicca, E. Frontoni, A. Mancini, and P. Zingaretti, "Multimedia discovery of the leonardo's vitruvian man," *SCIRES-IT: SCIentific RESearch and Information Technology*, vol. 5, no. 1, pp. 69–76, 2015, [Access Date: 01/01/2026]. [Online]. Available: http://www.sciresit.it/article/view/11420

[23] J. K. Ord, "The discrete student's *t* distribution," *The Annals of Mathematical Statistics*, vol. 39, no. 5, pp. 1513–1516, 1968. [Online]. Available: https://doi.org/10.1214/aoms/1177698133

[24] B. N. Datta, *Numerical Methods for Linear Control Systems: Design and Analysis*. Amsterdam, Netherlands: Elsevier Academic Press, 2004, [Access Date: 01/01/2026]. [Online]. Available: https://www.sciencedirect.com/book/monograph/9780122035906/numerical-methods-for-linear-control-systems

[25] Z.-F. Hao, G.-H. Guo, and H. Huang, "A particle swarm optimization algorithm with differential evolution," in *Proceedings of the 2007 International Conference on Machine Learning and Cybernetics*, Hong Kong, China, 2007, pp. 1031–1035. [Online]. Available: https://doi.org/10.1109/ICMLC.2007.4370294

[26] A. P. Piotrowski, J. J. Napiorkowski, and A. E. Piotrowska, "Particle swarm optimization or differential evolution—a comparison," *Engineering Applications of Artificial Intelligence*, vol. 121, p. 106008, 2023. [Online]. Available: https://doi.org/10.1016/j.engappai.2023.106008

[27] P. Kim and J. Lee, "An integrated method of particle swarm optimization and differential evolution," *Journal of Mechanical Science and Technology*, vol. 23, no. 2, pp. 426–434, 2009. [Online]. Available: https://doi.org/10.1007/s12206-008-0917-4

[28] S. M. A. Altbawi, S. A. Khalid, R. H. Alsisi, and A. I. Alhammi, "Benchmark functions: Properties and features," *Archives of Computational Methods in Engineering*, vol. 33, no. 3, pp. 1–13, 2025. [Online]. Available: https://doi.org/10.1007/s11831-025-10419-0

[29] H. Braun, "On solving travelling salesman problems by genetic algorithms," in *Parallel Problem Solving from Nature*, ser. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, 1991, vol. 496, pp. 129–133. [Online]. Available: https://doi.org/10.1007/BFb0029743

[30] P. Larrañaga, C. M. H. Kuijpers, R. H. Murga, I. Inza, and S. Dizdarevic, "Genetic algorithms for the travelling salesman problem: A review of representations and operators," *Artificial Intelligence Review*, vol. 13, no. 2, pp. 129–170, 1999. [Online]. Available: https://doi.org/10.1023/A:1006529012972

[31] E. Duman, M. Uysal, and A. F. Alkaya, "Migrating birds optimization: A new metaheuristic approach and its performance on quadratic assignment problem," *Information Sciences*, vol. 217, pp. 65–77, 2012. [Online]. Available: https://doi.org/10.1016/j.ins.2012.06.032

[32] M. Sharma and P. Kaur, "A comprehensive analysis of nature-inspired meta-heuristic techniques for feature selection problem," *Archives of Computational Methods in Engineering*, vol. 28, no. 3, pp. 1103–1127, 2021. [Online]. Available: https://doi.org/10.1007/s11831-020-09412-6

Contribution of Individual Authors to the Creation of a Scientific Article (Ghostwriting Policy)

M. Uysal developed the theoretical framework of the study and contributed to the design of the proposed Vitruvian Man Optimizer (VMO). The proposed method was further improved and refined by both authors.

A. Uysal implemented the Python code, conducted the computational experiments, and contributed to the literature review. Both authors reviewed, revised, and approved the final version of the manuscript.

Sources of Funding for Research Presented in a Scientific Article or Scientific Article Itself

No funding was received for conducting this study.

Conflict of Interest

The author has no conflict of interest to declare that is relevant to the content of this article.

Creative Commons Attribution License 4.0 (Attribution 4.0 International, CC BY 4.0)

This article is published under the terms of the Creative Commons Attribution License 4.0
https://creativecommons.org/licenses/by/4.0/deed.en_US