

The Status-based Optimization: Algorithm and comprehensive performance analysis

Jian Wang^a, Yi Chen^a, Chenglang Lu^b, Ali Asghar Heidari^c, Zongda Wu^d,
Huiling Chen^{a,*}

^a*Key Laboratory of Intelligent Informatics for Safety & Emergency of Zhejiang Province,
Wenzhou University, Wenzhou 325035, China*

^b*College of modern information technology, Zhejiang Institute of Mechanical and Electrical
Engineering, Hangzhou, Zhejiang 310051, China*

^c*School of Surveying and Geospatial Engineering, College of Engineering, University of
Tehran, Tehran, Iran*

^d*Department of Computer Science and Engineering, Shaoxing University, Shaoxing 312000,
China*

1. Proposed Status-based Optimization

This section focuses on the SBO algorithm’s exposition, detailing its mathematical modeling and computational complexity.

1.1. SBO Inspiration

The SBO algorithm models humanity’s essential drive to climb social ladders—a behavior rooted in our need for self-improvement [43]. This ambition mirrors optimization’s core goal: iterative refinement. Like people gaining advantages by connecting with successful peers [44], SBO agents learn from high-performing solutions to enhance search efficiency. Research in cognitive science and behavioral economics confirms that learning from high-status individuals improves problem-solving in complex scenarios. SBO translates this into computational terms, creating a collective intelligence where:

- Agents share knowledge (like human networks)
- Diverse strategies emerge naturally

*Corresponding Author: Huiling Chen (chenhuiling.jlu@gmail.com)

Email addresses: jona.wzu@gmail.com (Jian Wang), kenyoncy2016@gmail.com (Yi Chen), luchenglang@zime.edu.cn (Chenglang Lu), aliasghar68@gmail.com (Ali Asghar Heidari), zongda1983@163.com (Zongda Wu), chenhuiling.jlu@gmail.com (Huiling Chen)

- The system balances exploration and exploitation

In short, we can say this is how SBO works:

1. **Elite Engagement (Exploration)**

- Agents follow top performers to discover promising regions
- Analogous to seeking mentors in social hierarchies

2. **Resource Phases (Exploitation)**

- **Acquisition:** Gather information from elites
- **Evaluation:** Refine solutions like professionals improving skills

Several optimization algorithms inspired by human status-driven social behaviors and educational interactions have successfully solved complex problems. The Human Behavior-Based Optimization (HBBO) algorithm [45] mimics collective human behaviors such as cooperation, competition, imitation, and social learning. HBBO balances social learning with individual creativity through mechanisms like imitation, innovation, and collaboration, making it suitable for dynamic or multi-objective problems.

Similarly, the Educational Competition Optimizer (ECO) [46] models competitive learning environments where solutions compete and learn from top performers, guided by the best solution, akin to a teacher. This approach promotes rapid convergence and adaptability to constrained optimization scenarios, showcasing its efficiency in applications like academic performance modeling and game theory.

By formalizing status-seeking behaviors, SBO outperforms predecessors in:

- Balancing global/local search
- Reducing manual parameter tuning
- Scaling to high-dimensional problems

1.2. Mathematical Modeling of SBO

Drawing inspiration from human status-seeking behavior, the SBO algorithm frames optimization as both a personal and social development process. It begins by generating two diverse populations of agents—representing individuals from different social backgrounds—who then evolve through a process modeled after seeking mentorship from society’s elite.

Key Phases are as follows:

1. **Elite Pursuit:** Agents identify and move toward high-performing solutions (“mentors”)
2. **Resource Acquisition:** They gain valuable information (social capital)
3. **Strategic Integration:** Agents critically evaluate and adopt only the most beneficial improvements

This mirrors how people:

- **Advance socially** by learning from successful peers
- **Selectively adopt** behaviors that enhance their status
- **Systematically climb** hierarchies through accumulated advantages

The algorithm culminates by consolidating these improvements to deliver an optimal solution—mathematically representing the pinnacle of status achievement. (Full mathematical details follow in later sections.)

1.2.1. Initialization

The Initialization phase lays the foundation of the SBO algorithm by generating two populations, X^1 and X^2 . In this model, each index i corresponds to a unique family, where the same-indexed individuals across X^1 and X^2 represent family members with distinct knowledge levels and social standings. This dual-population design ensures that each family is represented by at least two

individuals, thereby capturing intra-family diversity and enabling dynamic updating of the elite member as the algorithm iterates.

$$X^1, X^2 = \begin{bmatrix} x_1 \\ \vdots \\ x_i \\ \vdots \\ x_N \end{bmatrix}_{N \times D} = \begin{bmatrix} x_{1,1} & \cdots & x_{1,j} & \cdots & x_{1,D} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ x_{i,1} & \cdots & x_{i,j} & \cdots & x_{i,D} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ x_{N,1} & \cdots & x_{N,j} & \cdots & x_{N,D} \end{bmatrix}_{N \times D} \quad (1)$$

Each individual's state is defined by Eq. (2).

$$x_{i,j} = U(lb_j, ub_j) \quad (2)$$

where $x_{i,j}$ is the j^{th} decision variable of the i^{th} individual, D is the number of decision variables, and lb_j and ub_j are the lower and upper bounds, respectively. This uniform initialization across the $N \times D$ matrices for both populations establish the problem's dimensional nature and ensures a diverse starting point.

After initialization, a selection process identifies the elite member for each family to form the elite population X^e . Specifically, for the i^{th} family,

$$x_i^e = \begin{cases} x_i^1 & \text{if } fobj(x_i^1) < fobj(x_i^2) \\ x_i^2 & \text{Otherwise} \end{cases} \quad (3)$$

where $fobj(\cdot)$ is the objective function.

This dual-population approach does more than just find top performers in each group—it mirrors real-world social mobility, where progress depends on both individual merit and strategic connections. The interactions between regular individuals x_i and their elite counterparts x_i^e simulate real-world status-oriented social networks, illustrating how elite figures facilitate progress and resource sharing within and across family units.

1.2.2. Elite Engagement

In the Elite Engagement phase, the SBO algorithm replicates the complex dynamics of human social status structures to enhance the search for optimal

solutions. This phase models individuals seeking guidance from high-status mentors—elite agents in the algorithm—to expedite their progress. Unlike isolated family frameworks, this progression extends beyond self-contained groups by establishing interconnections between different social units, creating a more adaptive and robust search mechanism.

To emulate this behavior, the SBO algorithm uses the Roulette Wheel selection method [47] to choose an individual from a subset of the population. This subset represents the most successful members across different families. This probabilistic selection process ensures that individuals do not solely rely on a single dominant peer but instead consider multiple influential agents, reflecting the unpredictable yet strategic nature of human networking.

The selected individual, denoted as x_r^e , and the best individual in the population, x_b , together define a high-status circle—a metaphorical yet computationally significant region within the solution space that agents aim to integrate into. This dynamic representation of social mobility ensures that individuals systematically transition towards more promising areas of the search space.

To mathematically articulate this behavior, Eq. (4) and Fig. 1 delineate the generation of individuals x_i within the high-status circle—defining the area of promise within the solution space. This high-status circle represents an adaptive region where individuals navigate toward better solutions, balancing both structured progression and exploratory randomness. The movement of an individual is governed by:

$$x'_i = \begin{cases} (1 - w_1 - w_2) \times x_i + w_1 \times x_r^e + w_2 \times x_b & \text{if } rand < w_3 \\ w_4 \times ((1 - w_1 - w_2) \times x_i + w_1 \times x_r^e + w_2 \times x_b) & \text{Otherwise} \end{cases} \quad (4)$$

where x_i represents the i^{th} individual in the population, x'_i denotes the next iteration, x_r^e is an elite individual selected via the Roulette Wheel method from the X^e population, and x_b is the best solution found so far. The movement strategy in Eq. (4) ensures that individuals are influenced by their own position, a high-performing peer, and the best-known solution.

The parameters w_1 and w_2 are generated using randn, providing normally

distributed randomness to weight the contributions of x_i , x_r^e , and x_b . These values introduce stochasticity while ensuring the movement remains within a logical bound, fostering a controlled yet diverse search across the solution space.

In contrast, w_3 and w_4 are designed parameters that dynamically adjust the influence of the high-status circle on exploration and exploitation. w_3 is calculated as:

$$w_3 = \tanh \left(\left(\frac{\sqrt{[MaxFEs - \text{randn} \times FEs]}}{i} \right)^{\frac{FEs}{MaxFEs}} \right) \quad (5)$$

where $MaxFEs$ denotes the maximum number of function evaluations, FEs is the current number of evaluations, and i is the index of the individual. This formulation allows w_3 to adapt as optimization progresses, determining whether the standard update rule or a more randomized search should be applied.

If $\text{rand} \geq w_3$, the second formulation in Eq. (4) is used, where w_4 serves as a scaling factor that generates a uniformly distributed random number between $[-w_3, w_3]$. This mechanism increases exploration diversity by enabling step-size adjustments, particularly when escaping local optima.

$$w_4 = \text{unifrnd}(-w_3, w_3) \quad (6)$$

By integrating these components, the SBO algorithm mirrors real-world decision-making—where individuals pursue successful peers while strategically exploring unconventional paths to optimize outcomes. The initial formulation in Eq. (4) strategically calculates an optimal position within the high-status circle, mirroring how individuals gain access to influential networks for better prospects. Through evolutionary computation, solutions progressively improve, moving toward more promising regions of the search space.

In contrast, the second formulation introduces a randomized scaling factor ranging from $[-w_3, w_3]$, allowing the algorithm to explore beyond the immediate promising area. This feature prevents premature convergence while enabling SBO to discover potentially superior solutions in unexplored areas. This balance reflects human decision-making. People sometimes diverge from estab-

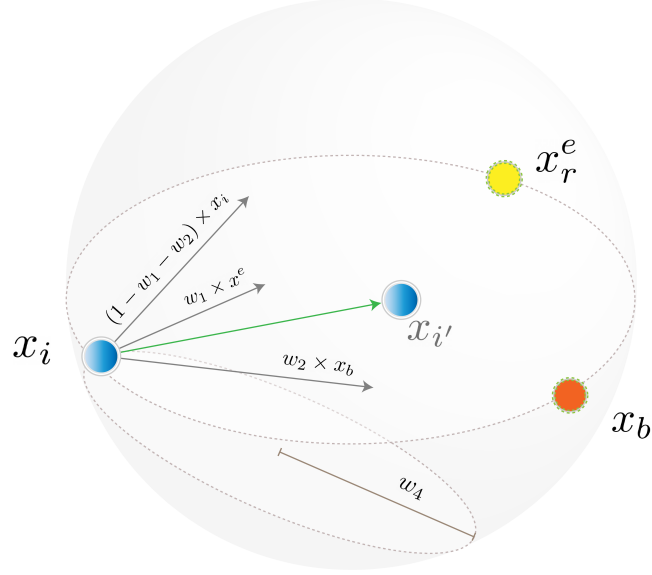


Fig. 1. Elite Engagement phase of SBO.

lished paths, whether through career changes or innovative ventures, to find opportunities missed by conventional approaches.

By combining structured learning and exploratory flexibility, SBO achieves an optimal balance between exploitation and exploration. This allows the algorithm to adapt effectively to complex optimization landscapes. The resulting approach improves solution quality while maintaining robustness across diverse problems, proving SBO's capability for high-performance optimization.

1.2.3. Resource Acquisition

The Resource Acquisition phase is crucial in transitioning from exploration to exploitation by acquiring and utilizing valuable insights—like social capital in human networks. In this phase, a *flag* vector is created for all individuals in the X population, initially set to 1 to indicate tentative status-related success. This flag later updates during the Resource Evaluation phase, serving as a dynamic indicator of each individual's efficacy in status improvement.

The resource acquisition mechanism varies based on status-related success. For socially successful individuals, resources are acquired selectively by averaging inputs from two sources: one from the elite individual within the same family unit and another from the overall best individual in the population. This process, captured by Eq. (7),

$$x_{i,j_1}^S = \frac{x_{i,j_2}^e + x_{b,j_3}}{2} \quad (7)$$

with $j_{idx} = \text{randi}(D)$ for $idx = 1, 2, 3$, reflects the blend of familial and external elite influences.

Conversely, socially unsuccessful individuals rely solely on familial resources. Their resource update follows Eq. (8)

$$x_{i,j}^s = x_{i,j}^e \quad \text{if } m_j = 1 \quad (8)$$

where the row vector m is initially zero and updated prior to social interactions by Eq. (9)

$$m(u(1 : \text{ceil}(\text{rand} \times D))) = 1 \quad (9)$$

where $u = \text{randperm}(D)$ providing a random permutation of decision variable indices.

As shown in Fig. 2, this phase directs the population toward promising regions of the solution space to maximize exploitation.

- Fig. 2(a): Successful individuals refine their positions by using resources from higher-status agents.
- Fig. 2(b): Struggling individuals reposition themselves through familiar resource.

The algorithm replicates status-driven social dynamics, where resource-rich individuals naturally attract more opportunities, to systematically guide the population toward better solutions, thereby significantly boosting exploitation and enhancing overall optimization performance.

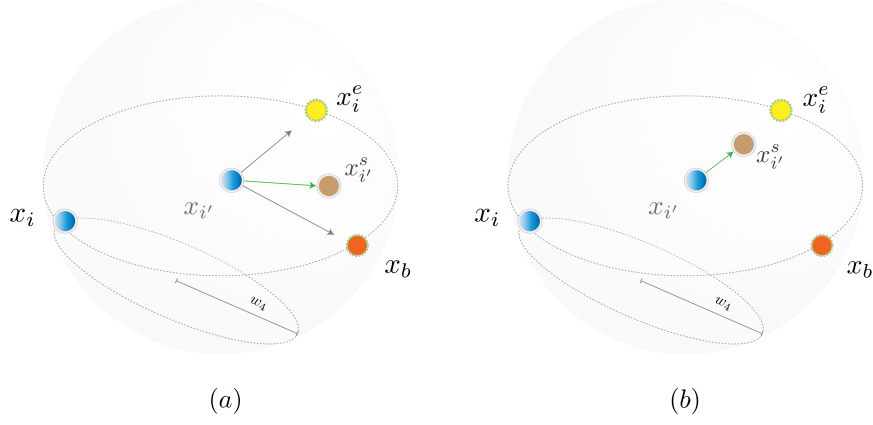


Fig. 2. Resource Acquisition phase of SBO.

1.2.4. Resource Evaluation

During the Resource Evaluation phase, the algorithm assesses whether acquired resources enhance an individual's fitness. Using the flag vector established earlier, it tracks progress:

- 1 = Fitness improvement (success)
- 0 = No improvement (failure)

In practice, if the objective function value of the updated individual x_i^s is better than that of the original x_i , the new state is retained:

$$x_i = x_i^s \quad \text{if } fobj(x_i^s) < fobj(x_i) \quad (10)$$

Simultaneously, the flag vector is refreshed as follows:

$$flag_i = \begin{cases} 1 & \text{if } fobj(x_i^s) < fobj(x_i) \\ 0 & \text{Otherwise} \end{cases} \quad (11)$$

Individuals showing no improvement maintain their current positions, while successful ones relocate to superior locations. This selective process mirrors real-world social advancement, where only valuable resources, those demonstrably

improving an agent’s status, are retained. This refinement progressively steers the search toward optimal solutions.

1.2.5. Consolidation

The Consolidation phase activates when termination criteria are met, either after reaching maximum function evaluations or achieving a sufficiently optimized solution (verified by enhancement metrics). Prior to this, the algorithm repeatedly cycles through its core phases:

- **Elite Engagement**
- **Resource Acquisition**
- **Resource Evaluation**

Each phase simulates status-driven interactions to progressively improve solutions.

Implementation Details:

- Algorithm 1 provides pseudo-code
- Fig. 3 shows the workflow

During Consolidation, the algorithm:

- **Compiles and assesses** results against objectives
- **Produces** a final solution embodying status-based heuristic
- **Ensures** efficient resource use and detailed documentation for analysis/application

1.3. Computational Complexity Analysis of SBO

The computational complexity of the SBO algorithm is primarily determined by the population size (N), the problem dimension (D), and the maximum number of iterations (T), which collectively define its termination criterion. In this

Algorithm 1 Pseudo-code for the Status-based Optimization

```
1: Input:  $N, D, MaxFEs, lb, ub, fobj$ ;  
2: Initialization:  
3:   Initialize  $X, X^e, Fit, Fit^e, flag$ ;  
4:   Calculate  $Fit$  and  $Fit^e$ ;  
5:   Update  $X^e$  and  $x_b$ ;  
6: while  $FEs < MaxFEs$  do  
7:   Select  $x_r^e$  from  $X^e$  by Roulette Wheel;  
8:   Elite Engagement:  
9:     Update  $w_1, w_2, w_3$ , and  $w_4$ ;  
10:    Update  $X$  by Eq. (4);  
11:    Apply boundary control to  $X$ ;  
12:    Initialize  $X^s$  as  $X$ ;  
13:    Resource Acquisition:  
14:      Initialize row vector  $m$  as 0;  
15:      Update  $m$  by Eq. (9);  
16:      For each  $x^s$  in  $X^s$ :  
17:        If  $x^s$  is successful:  
18:          Update  $x^s$  by Eq. (7);  
19:        Else:  
20:          Update  $x^s$  by Eq. (8);  
21:      End For  
22:    Resource Evaluation:  
23:      Update  $X$  by Eq. (10);  
24:      Update  $flag$  by Eq. (11);  
25:    Consolidation:  
26:      Update  $X^e$  and  $x_b$ ;  
27:      Increment  $FEs = FEs + 2N$ ;  
28: end while  
29: Return  $x_b$ .
```

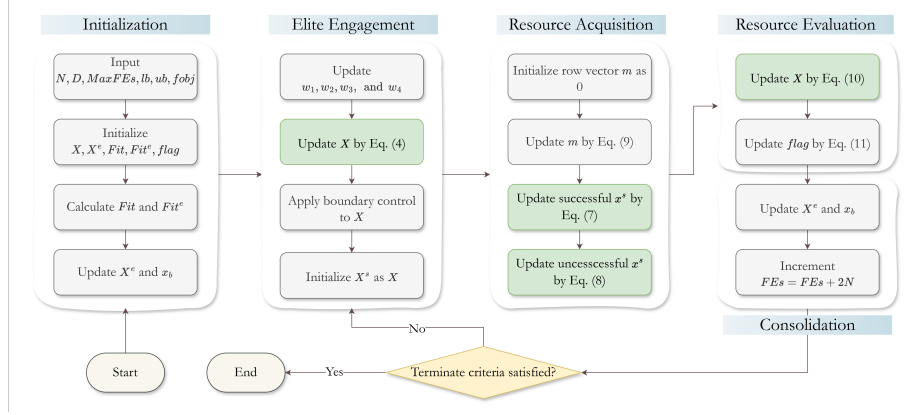


Fig. 3. Flowchart of SBO.

analysis, we focus on the algorithm's most computationally demanding operations while omitting less impactful vector updates. The Initialization phase, involving the generation of two populations of size $N \times D$, requires $O(2ND)$ time. The Elite Engagement phase updates the solution in $O(ND)$ per iteration, culminating in a total complexity of $O(TND)$ over T iterations. Both the Resource Acquisition and Resource Evaluation phases operate in $O(N)$ time per iteration, contributing $O(TN)$ cumulatively, while the Consolidation phase, which entails sorting, adds $O(TN \log N)$ to the overall cost. Summing these contributions, the total computational complexity of SBO is expressed as $O(TND + TN + TN \log N)$, a formulation that encapsulates the sequential and interdependent nature of its core operations. This analysis provides a concise quantitative estimate of the algorithm's efficiency and scalability in addressing a range of optimization challenges.