**The Status-based Optimization:**

**Algorithm and comprehensive performance analysis**

Jian Wanga, Yi Chena, Chenglang Lub, Ali Asghar Heidaric, Zongda Wud, Huiling Chena\*

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

*jona.wzu@gmail.com, kenyoncy2016@gmail.com*:*,* [*chenhuiling.jlu@gmail.com*](mailto:chenhuiling.jlu@gmail.com)

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

*luchenglang@zime.edu.cn*

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

[*aliasghar68@gmail.com*](mailto:aliasghar68@gmail.com)

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

[zongda1983@163.com](mailto:zongda1983@163.com)

***\*Corresponding Author:*** *Huiling Chen (*[chenhuiling.jlu@gmail.com)](mailto:chenhuiling.jlu@gmail.com))

# Abstracts

In recent years, real-world optimization problems have grown increasingly complex, challenging the effectiveness of traditional deterministic approaches. This paper introduces the Status-based Optimization (SBO), an efficient algorithm inspired by human desire for status advancement. By emulating the process through which individuals seek proximity to and subsequently learn from or acquire resources from high-status figures, SBO translates these social patterns into a robust computational method for addressing challenging optimization tasks. Its strength is validated through comprehensive comparative analyses against various meta-heuristic algorithms—including those based on human behaviors, classical methods, and state-of-the-art techniques—across problem dimensions of 10, 30, 50, and 100 as defined by the IEEE CEC 2017 test suite. Moreover, SBO demonstrates superior performance in feature selection by identifying fewer but more relevant features compared to 8 established binary optimizers in experiments involving 9 high-dimensional datasets. The algorithm's adaptability is further demonstrated in a multi-threshold image segmentation application at 3 threshold levels. SBO was benchmarked against 7 advanced meta-heuristic methods to segment 9 breast cancer histology images, successfully identifying optimal threshold values to enhance image clarity and segmentation accuracy. These promising results underscore the potential of SBO for future research and practical applications in a wide array of optimization scenarios. The SBO source code is available at <https://aliasgharheidari.com/SBO.html> for interested researchers.

**Keywords:** Meta-heuristic algorithm; Status-based Optimization; Global optimization; Feature selection; Multi-threshold image segmentation

# Proposed Status-based Optimization

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

### 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
* 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

### 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.)

#### 3.2.1 Initialization

The Initialization phase lays the foundation of the SBO algorithm by generating two populations, and . In this model, each index corresponds to a unique family, where the same-indexed individuals across and 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.

|  |  |  |
| --- | --- | --- |
|  |  |  |

Each individual's state is defined by Eq. :

|  |  |  |
| --- | --- | --- |
|  |  |  |

where is the decision variable of the individual, is the number of decision variables, and and are the lower and upper bounds, respectively. This uniform initialization across the 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 . Specifically, for the family,

|  |  |  |
| --- | --- | --- |
|  |  |  |

where 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 and their elite counterparts simulate real-world status-oriented social networks, illustrating how elite figures facilitate progress and resource sharing within and across family units.

#### 3.2.2 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 reflects the process of individuals seeking guidance from high-status mentors—represented in the algorithm as elite agents—to accelerate their growth. 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 selects an individual from the population—a subset representing the most successful members across different families—using the Roulette Wheel selection method [47]. 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 , and the best individual in the population, , 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. and **Fig. 2** delineate the generation of individuals 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:

|  |  |  |
| --- | --- | --- |
|  |  |  |

where represents the individual in the population, denotes the next iteration, is an elite individual selected via the Roulette Wheel method from the population, and is the best solution found so far. The movement strategy in Eq. ensures that individuals are influenced by their own position, a high-performing peer, and the best-known solution.

The parameters and are generated using , providing normally distributed randomness to weight the contributions of , , and . 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, and are designed parameters that dynamically adjust the influence of the high-status circle on exploration and exploitation. is calculated as:

|  |  |  |
| --- | --- | --- |
|  |  |  |

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

If , the second formulation in Eq. is used, where serves as a scaling factor that generates a uniformly distributed random number between . This mechanism increases exploration diversity by enabling step-size adjustments, particularly when escaping local optima.

|  |  |  |
| --- | --- | --- |
|  |  |  |

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. strategically calculates an optimal position within the high-status circle, paralleling 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 , 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, where people sometimes diverge from established 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.

#### 3.2.3 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 vector is created for all individuals in the 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

|  |  |  |
| --- | --- | --- |
|  |  |  |

with for , reflects the blend of familial and external elite influences.

Conversely, socially unsuccessful individuals rely solely on familial resources. Their resource update follows

|  |  |  |
| --- | --- | --- |
|  |  |  |

where the row vector is initially zero and updated prior to social interactions by

|  |  |  |
| --- | --- | --- |
|  |  |  |

where providing a random permutation of decision variable indices.

As shown in **Fig. 3**, this phase directs the population toward promising regions of the solution space to maximize exploitation.

* **Fig. 3**(a)**:** Successful individuals refine their positions by using resources from higher-status agents
* **Fig. 3**(b)**:** Struggling individuals reposition themselves through resource acquisition

The algorithm replicates status-driven social dynamics—where resource-rich individuals naturally attract more opportunities—to systematically guide the population toward better solutions. This significantly boosts exploitation while improving overall optimization performance.

#### 3.2.4 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 is better than that of the original , the new state is retained:

|  |  |  |
| --- | --- | --- |
|  |  |  |

Simultaneously, the flag vector is refreshed as follows:

|  |  |  |
| --- | --- | --- |
|  |  |  |

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.

#### 3.2.5 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. 4** shows the workflow

During Consolidation, the algorithm:

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

|  |
| --- |
| **Algorithm 1** Pseudo-code for the Status-based Optimization |
| Input ; |
| Initialization: |
| Initialize *X*; |
| Calculate and ; |
| Update and ; |
| **While** |
| Select from by Rolette Wheel; |
| Elite Engagement: |
| Update ; |
| Update by Eq. ; |
| Apply Boundary control to ; |
| Initialize as ; |
| Resource Acquisition: |
| Initialize row vector as 0; |
| Update by Eq. ;  **For** each in :  **If** is successful: |
| Update by Eq. ; |
| **Else**: |
| Update by Eq. ;  **End For** |
| Resource Evaluation: |
| Update by Eq. ; |
| Update by Eq. ; |
| Consolidation: |
| Update and ; |
| Increment; |
| **End While** |
| **Return** . |

### Computational Complexity Analysis of SBO

The computational complexity of the SBO algorithm is primarily determined by the population size (), the problem dimension (), and the maximum number of iterations (), which collectively define its termination criterion. In this 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 , requires time. The Elite Engagement phase updates the solution in per iteration, culminating in a total complexity of over iterations. Both the Resource Acquisition and Resource Evaluation phases operate in time per iteration, contributing cumulatively, while the Consolidation phase, which entails sorting, adds to the overall cost. Summing these contributions, the total computational complexity of SBO is expressed as , 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.