Journal of Computer Science

Original Research Paper

Slime Mould Reproduction: A New Optimization Algorithm for Constrained Engineering Problems

[1]Rajalakshmi Sakthivel and [2]Kanmani Selvadurai

[1]*Department of Computer Science and Engineering, Puducherry Technological University, Puducherry, India*
[2]*Department of Information Technology, Puducherry Technological University, Puducherry, India*

Article history
Received: 28-07-2023
Revised: 11-10-2023
Accepted: 03-11-2023

Corresponding Author:
Rajalakshmi Sakthivel
Department of Computer Science and Engineering, Puducherry Technological University,
Puducherry, India
Email: rajasakthi1996@pec.edu

Abstract: In recent explorations of biologically inspired optimization strategies, the Slime Mould Reproduction (SMR) algorithm emerges as an innovative meta-heuristic optimization technique. This algorithm is deeply rooted in the reproductive dynamics observed in slime molds, particularly the intricate balance these organisms strike between local and global spore dispersal. By replicating this balance, the SMR algorithm deftly navigates between exploration and exploitation phases, aiming to pinpoint optimal solutions across diverse problem domains. For the purpose of evaluation, the SMR algorithm was diligently tested on three engineering problems with inherent constraints: Gear train design, three-bar truss design, and welded beam design. A comprehensive comparative study indicated that the SMR algorithm outperformed esteemed optimization techniques such as Particle Swarm Optimization (PSO), Artificial Bee Colony (ABC), Differential Evolution (DE), Grasshopper Optimization Algorithm (GOA), and Whale Optimization Algorithm (WOA) in these domains. While the exemplary performance of the SMR algorithm is worth noting, it is essential, in line with the No Free Lunch (NFL) theorem, to underscore that the performance of any optimization algorithm invariably depends on the particular problem it addresses. Nevertheless, the SMR algorithm's consistent triumph in benchmark tests underscores its potential as a formidable contender in the vast realm of optimization algorithms. The current exploration not only emphasizes the ever-expanding horizon of bio-inspired algorithms but also positions the SMR algorithm as a pivotal addition to the arsenal of optimization tools. Future implications and the potential scope of the SMR algorithm extend to various domains, from computational biology to intricate industrial designs. Envisioning its broader applicability, upcoming research avenues may delve into refining SMR's core procedures, borrowing insights from a broader range of biological behaviors for algorithmic ideation, and contemplating a binary version of the SMR algorithm, thereby amplifying its versatility in diverse optimization landscapes.

Keywords: Slime Mould Reproduction Algorithm, Bio-Inspired Meta-Heuristics, Optimization Techniques, Constrained Engineering Problems

Introduction

The search for effective solutions to intricate optimization problems has been a persistent endeavor, leading to the development and refinement of meta-heuristic algorithms. Bozorg-Haddad *et al.* (2017) emphasized the role and relevance of these algorithms in offering a comprehensive approach to various optimization challenges. Their adaptability and versatility in addressing both continuous and discrete optimizations make them pivotal in scientific and engineering contexts (Bryden, 2005; Beekman and Latty, 2015). One significant reason for this growing interest stems from the shortcomings of traditional mathematical methodologies, which often struggle with complex, multidimensional tasks.

Meta-heuristic algorithms stand out due to their incorporation of principles from natural phenomena and evolutionary behaviors Beyer and Schwefel (2002). Beheshti and Shamsuddin (2013); Abdel-Basset *et al.* (2018) elucidate, that these nature-inspired algorithms rely on a combination of exploitation and exploration in solution generation. This duality in approach prevents them from getting stuck in local optima, hence, ensuring a more robust convergence to optimal solutions.

Among the myriad of meta-heuristic algorithms, population-based stochastic optimization techniques hold special significance. Their inspiration from nature is evident in the way Genetic Algorithms (GAs) (Chawla and Duhan, 2015; Clerc, 2010; Coello, 2002) utilize the principle of "survival of the fittest" to refine solutions. Similarly, the Particle Swarm Optimization (PSO) technique is modeled after the collective search behavior seen in bird flocks (Coello, 2000) while the Gravitational Search Algorithm (GSA) (Deb, 1991) employs Newtonian physics, specifically the laws of gravity and motion, to guide its search agents. The dynamism of the meta-heuristic research domain can also be attributed to the No Free Lunch (NFL) theorem (Koza, 1994) which underscores the importance of having a diverse toolkit of optimization techniques to cater to various problem types.

In this continually evolving landscape, we introduce a novel algorithm, inspired by the reproductive behavior of slime molds. As a testament to the breadth of nature-inspired algorithms, our proposed Slime Mould Reproduction (SMR) algorithm represents an exciting addition to this category, echoing the findings of Bryden (Mirjalili *et al*., 2018) and Reid and Latty (Mirjalili, 2019) who have extensively studied the behaviors of slime molds.

This study provides an in-depth exploration of the novel SMR algorithm, seamlessly connecting its theoretical underpinnings with real-world applications. Beginning with a review of prevalent stochastic optimization techniques, we delve into the unique reproductive behaviors of slime molds, which underlie our algorithmic framework. Subsequent sections illuminate the intricacies of the SMR algorithm, backed by empirical findings. Furthermore, we emphasize the expansive potential scope of the SMR algorithm, underscoring its relevance in addressing complex engineering challenges. We round off our discussion by spotlighting the broader promise and potential of bio-inspired meta-heuristic algorithms in the optimization landscape.

Related Study

Meta-heuristic algorithms (Beekman and Latty, 2015) have proven to be efficacious strategies in the quest for solutions to complex optimization problems. These high-level processes generate heuristic solutions capable of traversing a wide range of intricate problem domains. A considerable proportion of these algorithms are inspired by nature and evolutionary processes (Bozorg-Haddad *et al*., 2017; Clerc, 2010) transforming observed phenomena into pioneering problem-solving techniques. One such foundational contribution was the Genetic Algorithm (GA), which utilized principles of genetics and natural selection in the optimization process (Chawla and Duhan, 2015). Over the years, the GA has spawned numerous variants that have demonstrated their effectiveness across diverse problem domains (Clerc, 2010; Coello, 2002). Another significant development in the field of optimization has been the notion of swarm intelligence, which has given rise to algorithms that mirror the behaviors of natural swarms. A case in point is the Artificial Bee Colony (ABC) optimization algorithm, which simulates the behaviors of different classes of bees to optimize solutions (Karaboga and Basturk, 2008). Likewise, the Firefly Algorithm (FA) is inspired by the luminescent behavior of fireflies (Łukasik and Żak, 2009) and the Bat Algorithm (BA) adopts the echolocation behavior of bats (Li *et al*., 2020) to guide their optimization strategies.

The incorporation of principles from the field of physics has added an additional layer to algorithmic development. The Gravitational Search Algorithm (GSA), for instance, applies laws of gravity and mass interactions to guide the search process (Deb, 1991) while the Galactic Swarm Optimization (GSO) utilizes principles of galactic motion (Muthiah-Nakarajan and Noel, 2016) and the Inclined Planes System Optimization (IPO) leverages Newton's laws (Mohammadi *et al*., 2022). Meta-heuristic algorithms are typically categorized into two main types: Single-solution-based and population-based. The former initiates with a single random solution and proceeds through iterative improvements (Maniezzo *et al*., 2021) whereas the latter commences with an array of random solutions that undergo refinement across iterations (Chawla and Duhan, 2015; Coello, 2000; Deb, 1991; Karaboga and Basturk, 2008). Each type presents unique advantages; with the key distinction being the strategy they adopt to handle challenges such as local optima, premature convergence, and deception.

Materials and Methods

The Slime Mould Reproduction (SMR) algorithm emerges as a novel contribution in the field of optimization, distinctively addressing the frequent issues of premature convergence and entrapment in local optima commonly faced by traditional algorithms. The unique biological inspiration behind SMR, derived from the reproductive behavior of slime moulds, offers a new perspective on navigating the complex landscapes of optimization problems. This innovative approach suggests a potential for greater adaptability and robustness in finding global optima, setting SMR apart from its predecessors and marking a significant step forward in the development of meta-heuristic optimization methods.

In the subsequent sections, this study provides a detailed description and implementation of the Slime

DOI: 10.3844/jcssp.2024.96.105

Mould Reproduction (SMR) algorithm, particularly focusing on constrained engineering problems. It explores the unique operational mechanics of the SMR algorithm, demonstrating how its bio-inspired approach is translated into an effective optimization algorithm. The application of SMR in solving complex engineering problems highlighting its effectiveness compared to traditional algorithms. Additionally, the broader scope and potential applications of the SMR algorithm in various optimization domains are discussed, underscoring its significance as a versatile and innovative contribution to the field of meta-heuristic optimization.

Slime Mould Reproduction

The peculiar behavioral dynamics of slime mold Meena *et al*. (2019) specifically during their reproductive cycle have been identified as a source of potential applications in the realm of meta-heuristics (Mirjalili *et al*., 2018; Mirjalili, 2019; Price, 2013). This investigation is centered on the reproductive phase, wherein the slime mold forms a mature fruiting body responsible for the production of spores (Price, 2013; Rashedi *et al*., 2009; Reid and Latty, 2016). These spores, encapsulating the genetic material of the organism, disperse into the surrounding environment, laying the foundation for possible proliferation.

Biological Underpinnings and Algorithm Development

The underlying strategies of this intriguing biological process supply a conceptual framework for algorithm development, thereby stimulating innovative problem-solving paradigms. The slime mold's reproductive process is characterized by the local and global dissemination of spores, an essential tactic ensuring species survival and expansion. External influences including, but not limited to, wind, water currents, and biotic agents contribute to the transportation of spores across disparate ecological niches, facilitating the colonization of new territories. The process of spore dispersal follows a stochastic mechanism, which acts to reduce risks related to local environmental changes and simultaneously bolsters the chance of discovering optimal habitats for growth and reproduction.

Meta-Heuristic Principles and Slime Mould Reproduction

The aforementioned process bears an uncanny resemblance to the principles that govern meta-heuristic optimization, wherein solutions are sought at both local and global levels within the search space. A meta-heuristic model inspired by the reproductive behavior of slime molds inherently balances exploration (global search) and exploitation (local search), mirroring the spore distribution strategy of the organism. The impact of external factors on the spore dispersal process resonates with the utilization of randomization techniques in meta-heuristics, infusing an element of unpredictability while retaining a guided approach toward solution discovery. Upon maturity, the fruiting body of the slime mold releases spores, which, with the assistance of external agents, are dispersed over substantial distances before germinating into new organisms (Price, 2013; Rashedi *et al*., 2009; Reid and Latty, 2016).

Mathematical Modeling and Meta-Heuristic Development

The phenomena of asexual reproduction ensure the availability of robust nutritional resources locally while enabling the colonization of new habitats globally. It is this behavior of the slime mold that we emulate in the proposed meta-heuristic. Specifically, the updating of the slime position in the search space is reflective of the spore dissemination strategy. Equation 1 illustrates this principle:

$$X(t+1) = \begin{cases} X(t) + p \times r_1 \times \left(X_{best}(t) - X_A(t)\right) + r_2, \\ \quad \text{if } r_1 \geq NR \\ rand(UB - LB) + LB, otherwise \end{cases} \tag{1}$$

Here, 't' denotes the current iteration, $X(t)$ the current position of a slime at iteration 't', 'p' is a parameter that influences the step size in the position update, ranging from -2-2, 'r_1' is a random number adhering to a uniform distribution in the range [0, 1], 'X_{best}' signifies the best slime position, '$X_A(t)$' denotes a random slime position, 'r_2' is a random number in the range [-1, 1] and '*NR*' is the nutritious rich probability, which linearly decreases from 0-5 over the course of the algorithm's execution.

Slime Mould Reproduction Algorithm Pseudo-Code

Initialize the population of slimes: X_i for i = 1 to n
Initialize the algorithm parameters and maximum iteration: Max_iter
for t = 1 to Max_iter do
    for each slime in the population do
        Calculate the fitness of the slime
    end for
    Define Nutritious Rich Probability (NR)
    for each slime in the population do
        if r > = NR then
            Update the position of the slime:
            $X(t+1) = X(t) + p * r_1 * (X_{best}(t) - X_A(t))$ + 'r_2
        else
            Update the position of the slime:
            $X(t+1) = (UB - LB) * rand + LB$
        end if
    end for
    $t = t + 1$
end for
Return the best fitness
end

DOI: 10.3844/jcssp.2024.96.105

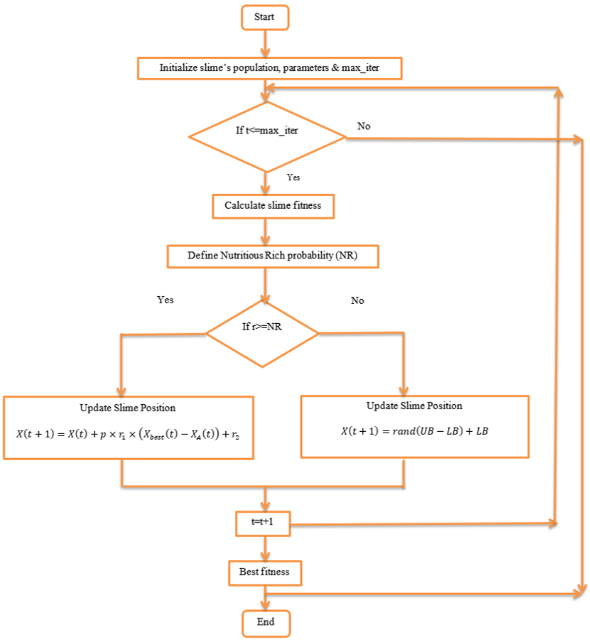

Fig. 1: Flowchart representation of the slime mold reproduction algorithm

In the SMR algorithm, we begin by initializing a population of slimes (Step 1). Each slime represents a potential solution in the search space. We then proceed with an iterative process that cycle through the population of slimes. Each slime's fitness is calculated (Steps 4-6), representing how well the slime (solution) fits the optimization criteria. A Nutritious Rich Probability (NR) is defined (Step 7) and for each slime in the population, it is checked whether a randomly generated number '*r*' is greater than or equal to NR (Step 9). If true, the position of the slime is updated using the best and a random slime's positions (Step 11), mimicking the local distribution of spores. If not, the slime's position is updated randomly within the bounds of the search space (Step 14), simulating the global dispersal of spores. The iteration '*t*' increments (Step 17) until it reaches the maximum iteration number (Max_iter). Finally, the algorithm returns the best fitness score obtained (Step 19), indicating the optimal or near-optimal solution found by the SMR. Following the comprehensive exposition in the pseudo-code, we provide a flowchart, as depicted in Fig. 1, for a streamlined visual comprehension of the slime mold reproduction algorithm.

The computational exploration of this natural phenomenon could give rise to strategies that replicate nature's inherent optimization capabilities, unveiling new opportunities in the arena of meta-heuristic research. Therefore, the reproductive strategies of slime molds, specifically those related to spore production and distribution, yield insightful pathways for the design of innovative meta-heuristics. Their robustness, efficiency, and adaptability suggest promising trajectories for the enhancement of optimization algorithms.

Constrained Engineering Optimization Using SMR

Effectively managing constraints is paramount in the optimization process. Constraints act as a crucial sieve, differentiating feasible solutions from infeasible outcomes birthed by heuristic algorithms. This research delves into the adeptness of the Slime Mould Reproduction (SMR) algorithm in circumnavigating such challenges. Three seminal constrained engineering optimization problems were employed to test the SMR algorithm:

- Gear train design
- Three-bar truss design
- Welded beam design problem (Reid and Latty, 2016; Sandgren, 1990; Soler-Dominguez *et al*., 2017; Vogel *et al*., 2018; Vallverdú *et al*., 2018)

Each problem has been meticulously chosen to shed light on the pragmatic efficacy of the SMR algorithm. Their selection grants a holistic analysis, elucidating the adaptability and utility of the SMR in diverse optimization scenarios:

- Gear train design: This problem, laden with non-linear equations and discrete variables, provides a foundational test for the SMR's prowess in discrete optimization terrains
- Three-bar truss design: Featuring continuous variables and inequality constraints, it gauges the algorithm's proficiency in structural mechanic challenges
- Welded beam design: A rigorous examination of the algorithm, this problem is rife with non-linear objective functions and constraints, testing the SMR's capability in intricate non-linear optimization contexts

To validate the robustness and reliability of the Slime Mould Reproduction (SMR) algorithm, we employed a systematic approach to problem-solving. Each problem was tackled with a consistent set of parameters:

- Population size: A standard population size of 30 entities was maintained for each experiment. This ensures that the search space was explored adequately, offering a balance between exploration and exploitation
- Iterations: A maximum of 1000 iterations was set for every problem. This allowed the algorithm ample opportunity to converge to a solution while keeping computational efforts in check
- Repetitions: To further reinforce the credibility of our findings and rule out the influence of randomness or any anomalies, each experimental run was reiterated 10 times. This repetitive testing guarantees the consistency and reliability of the results achieved by the SMR algorithm

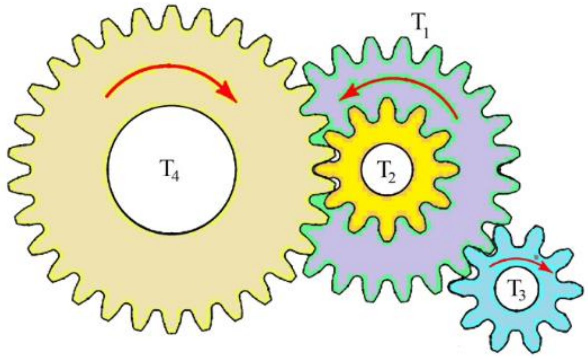

Fig. 2: Gear train design problem

Beyond theoretical assessment, the application of the SMR algorithm holds promising implications in real-world scenarios. The exploration emphasizes the capacity of the SMR algorithm to efficiently solve intricate engineering optimization problems. Our experimental approach across these disparate contexts serves to endorse the potential of the SMR algorithm as a robust tool for handling complex and constrained engineering optimization tasks.

Gear Train Design Problem

In mechanical engineering, the optimization of gear ratios within a four-gear train set is a common challenge. The principal objective within this context is the minimization of the gear ratio for the set under consideration, wherein the number of teeth on each gear forms the decision variables (Sandgren, 1990; Soler-Dominguez *et al*., 2017; Vogel *et al*., 2018; Vallverdú *et al*., 2018). Notably, while there are no explicit constraints within this problem, the range of variables essence, the permissible number of teeth that each gear can host is implicitly operationalized as constraints. These constraints hold a direct influence on the gear ratio and by extension, the overall efficiency of the gear system. The schematic design of the gear system is captured in Fig. 2.

Mathematical Formulation

The mathematical formulation of the optimization problem associated with gear train design, thereby demonstrating the applicability and effectiveness of the Slime Mould Reproduction (SMR) algorithm in managing engineering constraints within practical scenarios:

$$Minimize\ f(T) = \left(\frac{1}{6.931} - \frac{T_2\,T_3}{T_1\,T_4}\right)^2, \quad where\ T = (T_1, T_2, T_3, T_4) \qquad (2)$$

$$Subject\ to: 12 \leq T_i \leq 60, for\ i = 1, 2, 3, 4$$

Results

Table 1 presents the optimal results achieved by the SMR algorithm in comparison to renowned algorithms like Particle Swarm Optimization (PSO) (Coello, 2000) Slime Mould Algorithm (SMA) (Wolpert and Macready, 1997) Artificial Bee Colony (ABC) (Karaboga and Basturk, 2008) Differential Evolution (DE) (Yang, 2010a) Whale Optimization Algorithm (WOA) (Yang, 2010b) and Grasshopper Optimization Algorithm (GOA) (Zäpfel *et al*., 2010).

On examining the pinnacle of objective function values secured by these algorithms, the superior performance of the SMR algorithm emerges distinctly. This not only endorses the algorithm's proficiency in exploring the search space but also its adeptness in managing constraints, resulting in superior optimization outcomes. Such findings punctuate the potential of the SMR algorithm for real-world mechanical engineering challenges, encouraging its broader adoption in the domain of constrained optimization. Additionally, this research lays the groundwork for further exploration into the SMR algorithm's applicability in other complex optimization scenarios.

Furthermore, the success of the SMR algorithm in this problem highlights its versatility and robustness. Such attributes make it a potential candidate for a plethora of real-world engineering challenges that are characterized by multi-dimensional search spaces and stringent constraints. The success of SMR in this problem is an indicative testament to its promise in broader applications, beyond just academic exercises. Additionally, the referenced algorithms like PSO, SMA, and ABC, while competent in their right, seem to find a formidable contender in SMR, which appears to be more adept at navigating and optimizing within the search space of the welded beam design problem.

In the grander scheme of things, while algorithms like SMR serve as pivotal tools for engineers and designers, their real-world implications stretch far beyond. The ability to derive optimal designs can translate to substantial cost savings, enhanced safety standards, and efficient resource utilization in various engineering projects.

Table 1: Performance analysis of gear train design problem

	Optimum values for variables				
Algorithm	T_1	T_2	T_3	T_4	Optimum cost
SMR	40	12	12	16	3.0726E-17
SMA	43	12	19	36	3.0372E-12
PSO	46	20	17	44	2.6008E-011
ABC	43	19	16	43	1.251E-10
DE	44	14	18	48	2.4008E-10
GOA	20	16	43	49	2.75E-10
WOA	34	14	18	48	1.263E-08

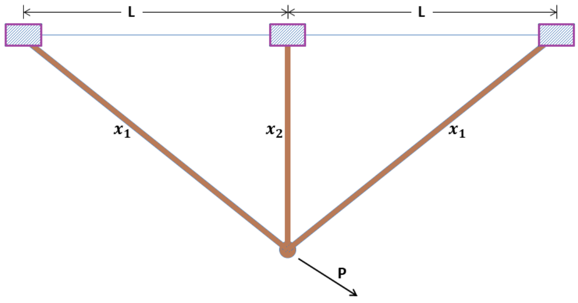

Fig. 3: Three-bar truss design problem

Three Bar Truss Design Problem

The three-bar truss design problem stands as a canonical structural optimization task, ubiquitously encountered within the realm of civil engineering. This problem involves the fine-tuning of two variables with the dual objective of minimizing the truss weight while concurrently adhering to stress, deflection, and buckling constraints. Given the constrained character of this problem's search space, it necessitates an exhaustive exploration and meticulous analysis. A graphical representation of this problem, as detailed in references (Sandgren, 1990; Soler-Dominguez *et al*., 2017; Vogel *et al*., 2018; Vallverdú *et al*., 2018) is depicted in Fig. 3.

Mathematical Formulation

The mathematical encapsulation of the three-bar truss design problem is presented as follows:

$$Minimize\ f(X) = L \times \left(2\sqrt{2}x_1 + x_2\right) \quad (3)$$

Subject to constraints:

$$\frac{\sqrt{2}x_1+x_2}{2x_1x_2+\sqrt{2}x_1^2}P \leq \sigma \quad (4)$$

$$\frac{x_2}{2x_1x_2+\sqrt{2}x_1^2}P \leq \sigma \quad (5)$$

$$\frac{1}{x_1+\sqrt{2}x_2}P \leq \sigma \quad (6)$$

With $0.01 \leq x_i \leq 1\ for\ i = 1, 2$ parameters where, L = 100 cm, P = 2 km/cm^2 and σ = 2 km/cm^2.

Table 2 furnishes the optimal solutions procured using the Slime Mould Reproduction (SMR) algorithm for the three-bar truss design problem. For a more holistic assessment, the SMR algorithm's performance is benchmarked against prominent algorithms such as Particle Swarm Optimization (PSO) (Coello, 2000) Slime Mould Algorithm (SMA) (Wolpert and Macready, 1997) Artificial Bee Colony (ABC) (Karaboga and Basturk, 2008) Differential Evolution (DE) (Yang, 2010a) Whale Optimization Algorithm (WOA) (Yang, 2010b) and Grasshopper Optimization Algorithm (GOA) (Zäpfel *et al*., 2010).

Table 2: Performance analysis of the bar truss design problem

	Decision variables		
Algorithm	X_1	X_2	Optimum cost
SMR	0.78546	0.42680	191.7042
SMA	0.76345	0.46321	215.6541
PSO	0.78997	0.66579	228.6549
DE	0.81815	0.36946	267.8173
ABC	0.74669	0.41526	265.9358
GOA	0.78967	0.41932	264.9815
WOA	0.79603	0.42945	261.8754

Discussion

Inspection of the optimal cost values from various algorithms clearly show cases the superiority of the proposed SMR algorithm. Its exceptional performance suggests that it cannot only parallel but exceed the proficiency of other leading algorithms for complex structural optimization tasks like the three-bar truss design problem. Delving into the practical implications, the SMR algorithm's adaptability and robustness earmark it as a potent tool for real-world engineering applications. The success observed here further beckons exploration of the algorithm's capabilities across a wider array of engineering challenges.

Welded Beam Design Problem

The welded beam design problem manifests as a salient challenge in the realm of structural optimization (Wolpert and Macready, 1997) primarily oriented toward minimizing the manufacturing costs of a welded beam. Figure 4 provides a graphical representation of the welded beam design problem, as delineated in references (Sandgren, 1990; Soler-Dominguez *et al*., 2017; Vogel *et al*., 2018; Vallverdú *et al*., 2018). This problem involves the consideration of several factors, including the weld throat size of the beam (θ), buckling load (Pc), and beam end deflection (δ). The optimization process encompasses four variables: Welded thickness (h), clamping bar length (l), bar height (t), and bar thickness (b).

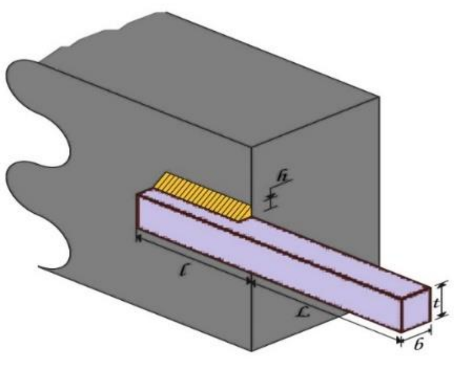

Fig. 4: Welded beam design problem

DOI: 10.3844/jcssp.2024.96.105

Mathematical Formulation

The mathematical formulation of the welded beam design problem is as follows:

$$Consider\ x = (x_1, x_2, x_3, x_4) = [h\ l\ t\ b]$$

$$Minimize\ f(x) = 1.10471x_1^2x_2 + 0.04811x_3x_4(14.0 + x_2) \quad (7)$$

Subject to constraints:

$$g_1(x) = \tau(x) - \tau_{max} \leq 0 \quad (8)$$

$$g_2(x) = \sigma(x) - \sigma_{max} \leq 0 \quad (9)$$

$$g_3(x) = \delta(x) - \delta_{max} \leq 0 \quad (10)$$

$$g_4(x) = x_1 - x_4 \leq 0 \quad (11)$$

$$g_5(x) = P - P_c(x) \leq 0 \quad (12)$$

$$g_6(x) = 0.125 - x_1 \leq 0 \quad (13)$$

$$g_7(x) = 1.10471x_1^2 + 0.04811x_3x_4 (14.0 + x_2) - 5.0 \leq 0 \quad (14)$$

With:

$$0.1 \leq x_1 \leq 2,\ 0.1 \leq x_2 \leq 10,\ 0.1 \leq x_3 \leq 10,\ 0.1 \leq x_4 \leq 2:$$

where:

$$\tau(x) = \sqrt{(\tau')^2 + 2\tau'\tau''\frac{x_2}{2R} + (\tau'')^2} \quad (15)$$

$$\tau' = \frac{P}{\sqrt{2}x_1x_2},\ \tau'' = \frac{MR}{J}, M = P\left(L + \frac{x_2}{2}\right) \quad (16)$$

$$R = \sqrt{\frac{x_2^2}{4} + \left(\frac{x_1+x_3}{2}\right)^2} \quad (17)$$

$$J = 2\left\{\sqrt{2}\ x_1x_2\left[\frac{x_2^2}{4} + \left(\frac{x_1+x_3}{2}\right)^2\right]\right\} \quad (18)$$

$$\sigma(x) = \frac{6PL}{x_4x_3^2} \quad (19)$$

$$\delta(x) = \frac{6PL^3}{Ex_3^2x_4} \quad (20)$$

$$P_c(x) = \frac{4.013E\sqrt{\frac{x_3^2x_4^6}{36}}}{L^2}\left(1 - \frac{x_3}{2L}\sqrt{\frac{E}{4G}}\right) \quad (21)$$

With P = 6000lb, L = 14 in., δ_{max} = 0.25 in., E = 30×1[6] *psi*, $G = 12\times10^6$ *psi*, τ_{max} = 13,600 *psi*, σ_{max} = 30,000 *psi*.

Table 3: Performance analysis of welded beam design problem

Algorithm	Optimum values for variables: h	l	t	b	Optimum cost
SMR	0.2016	3.2324	9.0461	0.2057	1.65704
SMA	0.2046	3.2874	9.0321	0.2086	1.69752
PSO	0.2047	3.3702	9.0462	0.2046	1.76454
DE	0.2035	3.4611	9.0375	0.2081	1.72997
ABC	0.2059	3.4785	9.0158	0.2034	1.72328
GOA	0.2028	3.3603	9.0563	0.2054	1.71492
WOA	0.2043	3.3725	9.0453	0.2145	1.72523

The study employed the Slime Mould Reproduction (SMR) algorithm to address the Welded Beam Design Problem and the results were contrasted with those of renowned algorithms such as Particle Swarm Optimization (PSO) (Coello, 2000) Slime Mould Algorithm (SMA) (Wolpert and Macready, 1997) and others. As illustrated in Table 3, the SMR algorithm showcased its merit by outclassing the other algorithms in formulating a cost-effective design for the welded beam.

Upon thorough scrutiny of the results, it is evident that the SMR algorithm demonstrates considerable prowess in structural optimization tasks, especially when juxtaposed against its contemporaries. This observation holds immense significance as the welded beam design problem is inherently intricate, necessitating precise manipulation of multiple parameters to achieve cost-efficiency without compromising on structural integrity.

Potential Scope of the SMR Algorithm

Beyond the constrained engineering problems explicitly tackled in this study, the underlying mechanics of the SMR algorithm suggest a vast landscape of potential applications:

- Complex system simulations: The SMR's adaptability could be instrumental in modeling and optimizing intricate systems, be it traffic flow in a metropolitan city or the dynamics of a natural ecosystem
- Machine learning and data mining: Given its ability to navigate large search spaces, the SMR algorithm could be applied to feature selection or hyperparameter tuning, ensuring more accurate and efficient machine learning models
- Network design and optimization: Whether optimizing the layout of a computer network or designing efficient supply chains, the SMR's balance of exploration and exploitation might offer unique solutions
- Scheduling problems: In scenarios like airline scheduling, production line optimization, or task allocation, the SMR algorithm could be employed to find solutions that maximize efficiency while adhering to constraints

DOI: 10.3844/jcssp.2024.96.105

Conclusion

In conclusion, this study introduces the Slime Mould Reproduction (SMR) algorithm, a novel optimization approach inspired by the fascinating reproductive behavior of slime molds. Particularly, the SMR algorithm ingeniously simulates the strategies of spore production and dispersal employed by these organisms, balancing an efficient exploration and exploitation of the search space. The algorithm considers external factors and translates them into a stochastic system, enhancing its capability to navigate a diverse range of search domains. The application of the SMR algorithm to three distinct constrained engineering problems underscored its immense potential as an optimization technique. When juxtaposed with conventional optimization algorithms such as Particle Swarm Optimization (PSO), Artificial Bee Colony (ABC), Differential Evolution (DE), Grasshopper Optimization Algorithm (GOA), and Whale Optimization Algorithm (WOA), the SMR algorithm exhibited superior performance in procuring promising solutions for intricate problems.

Nevertheless, it is crucial to bear in mind the principles laid out by the No Free Lunch (NFL) theorem. It advocates that no single optimization algorithm is capable of effectively addressing all optimization problems. As such, while the SMR algorithm has demonstrated significant potential in resolving complex issues across diverse search domains, it should be regarded as one of many optimization algorithms, each with its own merits and limitations. Looking towards future research, several promising avenues can be charted. First, there is scope to further refine the SMR algorithm or amalgamate it with other techniques to optimize its efficiency. Second, as the SMR algorithm derives its inspiration from the reproductive behavior of slime molds, other biological behaviors could potentially be harnessed to design innovative algorithms. Third, the development of a binary version of the SMR algorithm to tackle multi-objective problems could substantially broaden its applicability and effectiveness across a wider range of optimization scenarios. Lastly, the application of the SMR algorithm to problems beyond the engineering domain could offer additional insights into its practicality.

In essence, the SMR algorithm signifies a significant advancement in the field of meta-heuristic optimization. Rooted in the intriguing biological phenomena of slime mold reproduction, its proven efficacy in complex problem-solving scenarios sets a compelling precedent. We anticipate that this innovative approach will continue to inspire new research directions and contribute to the ongoing advancement of optimization techniques.

Acknowledgment

We are profoundly grateful to the faculty, peers, and staff of Puducherry Technological University for their continuous support and insights that significantly enhanced this research. We would also like to extend our gratitude to everyone who provided feedback and guidance at various stages of this project.

Funding Information

This research did not receive any specific grants from external funding agencies in the public, commercial, or not-for-profit sectors. However, we acknowledge the internal resources and support provided by Puducherry Technological University, which greatly facilitated this study.

Author’s Contributions

Rajalakshmi Sakthivel: Contributed to the conceptualization and design of the methodology. Developed and implemented the slime mould reproduction algorithm. Conducted the experimental evaluations and wrote the original draft of the manuscript.

Kanmani Selvadurai: Supervised the optimization techniques and metaheuristic algorithms. Reviewed and improved the design of the methodology. Assisted in the analysis and interpretation of the experimental results. Revised and edited the manuscript for clarity and quality.

Ethics

All aspects of this research were conducted ethically and adhere to the highest standards set forth by Puducherry Technological University. No conflicts of interest or ethical concerns arose during the course of this study.

Conflict of Interest

The authors earnestly declare that they have no known competing financial interests or personal relationships that could have influenced the work reported in this study.

References

Abdel-Basset, M., Abdel-Fatah, L., & Sangaiah, A. K. (2018). Metaheuristic algorithms: A comprehensive review. *Computational Intelligence for Multimedia Big Data on The Cloud with Engineering Applications*, 185-231. https://doi.org/10.1016/B978-0-12-813314-9.00010-4

Bozorg-Haddad, O., Solgi, M., & Loáiciga, H. A. (2017). *Meta-Heuristic and Evolutionary Algorithms for Engineering Optimization*. John Wiley and Sons. https://doi.org/10.1002/9781119387053

Beyer, H. G., & Schwefel, H. P. (2002). Evolution strategies-a comprehensive introduction. *Natural Computing*, *1*, 3-52. https://doi.org/10.1023/A:1015059928466

Bryden, J. (2005). Slime mould and the transition to multicellularity: The role of the macrocyst stage. In *European Conference on Artificial Life* (pp. 551-561). Berlin, Heidelberg: Springer Berlin Heidelberg. https://doi.org/10.1007/11553090_56

Beekman, M., & Latty, T. (2015). Brainless but multi-headed: Decision making by the acellular slime mould Physarum polycephalum. *Journal of Molecular Biology*, *427*(23), 3734-3743. https://doi.org/10.1016/j.jmb.2015.07.007

Beheshti, Z., & Shamsuddin, S. M. H. (2013). A review of population-based meta-heuristic algorithms. *Int. J. Adv. Soft Comput. Appl*, *5*(1), 1-35.

Chawla, M., & Duhan, M. (2015). Bat algorithm: A survey of the state-of-the-art. *Applied Artificial Intelligence*, *29*(6), 617-634. https://doi.org/10.1080/08839514.2015.1038434

Clerc, M. (2010). *Particle Swarm Optimization* (Vol. 93). John Wiley and Sons. https://www.researchgate.net/publication/372412414_Particle_Swarm_Optimization

Coello, C. A. C. (2002). Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: A survey of the state of the art. *Computer Methods in Applied Mechanics and Engineering*, *191*(11-12), 1245-1287. https://doi.org/10.1016/S0045-7825(01)00323-1

Coello, C. A. (2000). Constraint-handling using an evolutionary multiobjective optimization technique. *Civil Engineering Systems*, *17*(4), 319-346. https://doi.org/10.1109/TEVC.2008.2009032

Deb, K. (1991). Optimal design of a welded beam via genetic algorithms. *AIAA Journal*, *29*(11), 2013-2015. https://doi.org/10.2514/3.10834

Koza, J. R. (1994). *Genetic programming II: Automatic Discovery of reusable programs*. MIT press. https://dl.acm.org/doi/abs/10.5555/183460

Karaboga, D., & Basturk, B. (2008). On the performance of artificial bee colony (ABC) algorithm. *Applied Soft Computing*, *8*(1), 687-697. https://doi.org/10.1016/j.cie.2020.107011

Łukasik, S., & Żak, S. (2009). Firefly algorithm for continuous constrained optimization tasks. In *Computational Collective Intelligence. Semantic Web, Social Networks and Multiagent Systems: 1st International Conference, ICCCI 2009, Wrocław, Poland, Proceedings 1* (pp. 97-106). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-04441-0_8

Li, S., Chen, H., Wang, M., Heidari, A. A., & Mirjalili, S. (2020). Slime mould algorithm: A new method for stochastic optimization. *Future Generation Computer Systems*, *111*, 300-323. https://doi.org/10.1016/j.future.2020.03.055

Meena, M., Kumar, R., & Swapnil, P. (2019). Slime molds. In: Vonk, J., Shackelford, T. (Eds.) Encyclopedia of Animal Cognition and Behavior. Springer, Cham. https://doi.org/10.1007/978-3-319-47829-6_1334-1

Muthiah-Nakarajan, V., & Noel, M. M. (2016). Galactic swarm optimization: A new global optimization metaheuristic inspired by galactic motion. *Applied Soft Computing*, *38*, 771-787. https://doi.org/10.1016/j.asoc.2015.10.034

Mohammadi, A., Sheikholeslam, F., & Mirjalili, S. (2022). Inclined planes system optimization: Theory, literature review and state-of-the-art versions for IIR system identification. *Expert Systems with Applications*, *200*, 117127. https://doi.org/10.1016/j.eswa.2022.117127

Maniezzo, V., Boschetti, M. A., & Stützle, T. (2021). Single Solution Metaheuristics. In *Matheuristics: Algorithms and Implementations* (pp. 61-94). Cham: Springer International Publishing. https://doi.org/10.1007/978-3-030-70277-9

Mirjalili, S. Z., Mirjalili, S., Saremi, S., Faris, H., & Aljarah, I. (2018). Grasshopper optimization algorithm for multi-objective optimization problems. *Applied Intelligence*, *48*, 805-820. https://doi.org/10.1007/s10489-017-1019-8

Mirjalili, S. (2019). Evolutionary algorithms and neural networks. In *Studies in Computational Intelligence* (Vol. 780). Berlin/Heidelberg, Germany: Springer. https://doi.org/10.1007/978-3-319-93025-1

Price, K. V. (2013). Differential evolution. In *Handbook of Optimization: From Classical to Modern Approach* (pp. 187-214). Berlin, Heidelberg: Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-30504-7

Rashedi, E., Nezamabadi-Pour, H., & Saryazdi, S. (2009). GSA: A gravitational search algorithm. *Information Sciences*, *179*(13), 2232-2248. https://doi.org/10.1016/j.ins.2009.03.004

Reid, C. R., & Latty, T. (2016). Collective behaviour and swarm intelligence in slime moulds. *FEMS Microbiology Reviews*, *40*(6), 798-806. https://doi.org/10.1093/femsre/fuw033

Sandgren, E. (1990). Nonlinear integer and discrete programming in mechanical design optimization. https://doi.org/10.1115/1.2912596

Soler-Dominguez, A., Juan, A. A., & Kizys, R. (2017). A survey on financial applications of metaheuristics. *ACM Computing Surveys (CSUR)*, *50*(1), 1-23. https://doi.org/10.1145/3054133

Vogel, D., Dussutour, A., & Deneubourg, J. L. (2018). Symmetry breaking and inter-clonal behavioural variability in a slime mould. *Biology Letters*, *14*(12), 20180504. https://doi.org/10.1098/rsbl.2018.0504

Vallverdú, J., Castro, O., Mayne, R., Talanov, M., Levin, M., Baluška, F., ... & Adamatzky, A. (2018). Slime mould: The fundamental mechanisms of biological cognition. *Biosystems*, *165*, 57-70. https://doi.org/10.1016/j.biosystems.2017.12.011

Wolpert, D. H., & Macready, W. G. (1997). No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation*, *1*(1), 67-82. https://doi.org/10.1109/4235.585893

Yang, X. S. (2010a). *Engineering Optimization: An Introduction with Metaheuristic Applications*. John Wiley and Sons. https://doi.org/10.1002/9780470640425

Yang, X. S. (2010b). Nature-inspired metaheuristic algorithms. *Luniver Press*.

Zäpfel, G., Braune, R., & Bögl, M. (2010). Metaheuristic search concepts: A tutorial with applications to production and logistics. https://doi.org/10.1007/978-3-642-11343-7