

DIRECT SEARCH METHODS FOR DETERMINING NEW DESIGNS OF AUXETIC COMPOSITE MATERIALS

Iulian Constantin COROPETCHI^{1,2}, Dan Mihai CONSTANTINESCU^{1,3} , Alexandru VASILE^{1,2}, Andrei Ioan INDREȘ^{1,2}, Ștefan SOROHAN¹ , Dragos Alexandru APOSTOL¹

¹ National University for Science and Technology Politehnica Bucharest, Bucharest, Romania

² Military Technical Academy “Ferdinand I”, Bucharest, Romania

³ Technical Sciences Academy of Romania, Bucharest, Romania

*corresponding author, iulian.coropetchi@mta.ro

Composite materials have gained significant attention in various engineering applications due to their ability to exhibit unique properties that can be tailored to meet specific design requirements. Among these, auxetic materials stand out for their counterintuitive behavior of expanding in all directions when stretched, as opposed to traditional materials which contract. This property makes auxetic materials promising candidates for applications such as impact protection, energy absorption, and advanced engineering structures. In this paper we investigate the application of direct search methods in determining new designs of auxetic composite materials.

Keywords: auxetic materials; composite materials; direct search methods.



Articles in JTAM are published under Creative Commons Attribution 4.0 International Unported License <https://creativecommons.org/licenses/by/4.0/deed.en>.
By submitting an article for publication, the authors consent to the grant of the said license.

1. Introduction

Structural optimization is a significant area of focus in mechanical engineering and has been continuously evolving over the years. It is commonly applied to reduce the material usage and total deformation energy of structures while ensuring adequate strength and mechanical stiffness (Li *et al.*, 2020). Traditionally, structural optimization can be categorized into three main types: 1) dimensional optimization, which involves determining the optimal distribution of parameters such as plate thickness or cross-sectional areas of truss bars; 2) shape optimization, aimed at identifying the optimal geometry of a specified domain; and 3) topological optimization, which stands apart from the other two as it allows the material to occupy any position within the defined domain (Bendsøe & Sigmund, 2004).

Structural optimization nowadays encompasses a much broader range of topics, extending beyond the previously mentioned classical approaches. Notable examples include, among others, cellular microstructures (Pan *et al.*, 2020; Ptochos & Labeas, 2012), deformable mechanisms (Zhu *et al.*, 2020), spinodal structures (Kumar *et al.*, 2020), multiphysics problems (Yoon, 2021), and composite materials (Casalotti *et al.*, 2020; Gu *et al.*, 2018a). Alongside the diversity of applications, significant attention is also devoted to the mathematical methods used to solve these problems. These methods can generally be grouped into three major categories: gradient-based approaches (Ghiasi *et al.*, 2009; 2010; Luu & Banh, 2023; Sandhu, 1971; Tavakoli, 2014),

direct search algorithms (Coropețchi *et al.*, 2022; Sait *et al.*, 2023; Vasile *et al.*, 2022), and artificial intelligence techniques (Gu *et al.*, 2018b; Harish *et al.*, 2020; Yu *et al.*, 2019; Zhang *et al.*, 2019).

In mechanical engineering, the development of efficient methodologies for solving optimization problems remains a continually evolving area of research. The intricate and dynamic nature of structures needs advanced optimization techniques capable of exploring the entire solution space and identifying designs that meet specific performance requirements. As computational capabilities and methodologies progress, researchers have access to a variety of optimization techniques, each offering unique advantages and limitations.

This article focuses on reviewing direct search methods for solving structural optimization problems in the field of composite materials. We investigate the application of these methods in determining new designs of auxetic materials. The methods are valued for their simplicity, versatility, and applicability to a wide range of problems. Unlike gradient-based approaches, direct search methods do not depend on explicit knowledge of the gradient of the objective function. Instead, they explore the optimization space through a structured sequence of trial points, making them particularly well suited to problems with non-differentiable, discontinuous, or computationally intensive objective functions.

The purpose of this article is to compare and assess various direct search methods based on their performance in solving structural optimization problems. For each method, its strengths, weaknesses, and practical considerations are analyzed in the context of the problem being addressed. This analysis aims to highlight the specific advantages and trade-offs associated with each method, providing valuable insights for their application in structural optimization tasks.

2. Problem definition

This study aims to evaluate the performance of various direct search methods in solving structural optimization problems. To facilitate this evaluation, we define an optimization problem involving composite materials, which serves as a benchmark for testing the methods. As the primary focus of the paper is to compare the performance of the methods rather than solving a specific structural problem, the selected problem is largely theoretical but has also practical application in the field of auxetic materials.

Composite materials, known for combining the properties of multiple constituents, introduce a new dimension to structural design. The optimization of structures made from these materials highlights the inherent complexity of balancing material proportions and orientations to achieve desired performance criteria. With their unique combination of strength, lightweight properties, and durability, composite materials present a challenging and yet rewarding domain for optimization methodologies.

We consider a 2D periodic structure consisting of two different materials for which the material domains are defined by triangular pixels in a representative volume element (RVE) which is shaped as a hexagon with 4 pixels on each side. Each pixel can be of hard or soft material. Hexagonal cells are obtained by rotation of the base area, as shown in Fig. 1. In the base triangle

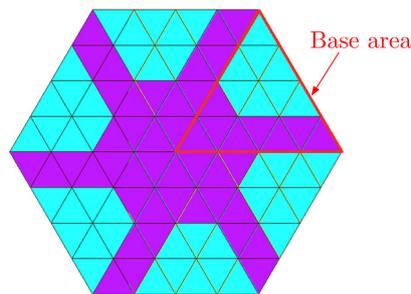


Fig. 1. Base area rotated 6 times with 60° .

area of the hexagon, there are 16 material domains. In our problem, we impose the condition that half of the material domains must be soft and the other half hard.

The RVE can be repeated in a pattern so that it can form the periodic structure presented in Fig. 2. Analyzing the figure, we can identify a rectangular area that contains two RVEs and form the analysis domain that will be used in FEM simulations.

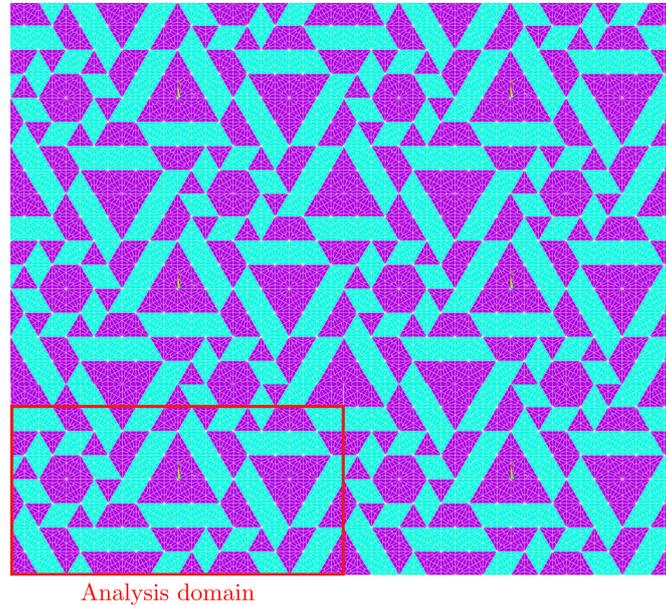


Fig. 2. Periodic structure and analysis domain.

The materials from which the analysis domain is built are represented with magenta and cyan colors. For each material, Young's modulus and Poisson's coefficient are known. The materials properties corresponding to each color are presented in Table 1.

Table 1. Materials description.

Material	Hard	Soft
Color		
Young's modulus	100000 MPa	1000 MPa
Poisson's ratio	0.2	0.4

In order to simulate the analysis domain in a periodic structure, specific boundary conditions must be applied for every load case to determine Poisson's coefficient. According to (Soroohan *et al.*, 2018) for bidimensional analysis, one should use 3 load cases for determining the specific material properties. These load cases are traction alongside the X -axis, traction alongside the Y -axis and shear in the XY plane. The representation of the boundary conditions for the analysis domain in the first load case is shown in Fig. 3, and the boundary conditions for all three load cases are presented in Table 2. In Fig. 3, the nodes that have 0 displacement are represented with blue arrows alongside the direction in which this constraint is imposed, and the nodes that are coupled in terms of displacement are represented with green arrows alongside the direction in which the coupling of the displacement is imposed.

The finite elements utilized can be quadrilaterals with either four or eight nodes. In a linear elastic analysis, the stress distribution within each finite element may vary unless techniques such as reduced integration for linear quadrilateral elements or stress averaging across the element are applied. However, for practical purposes, constant stresses are often assumed for each finite element to simplify the calculations.

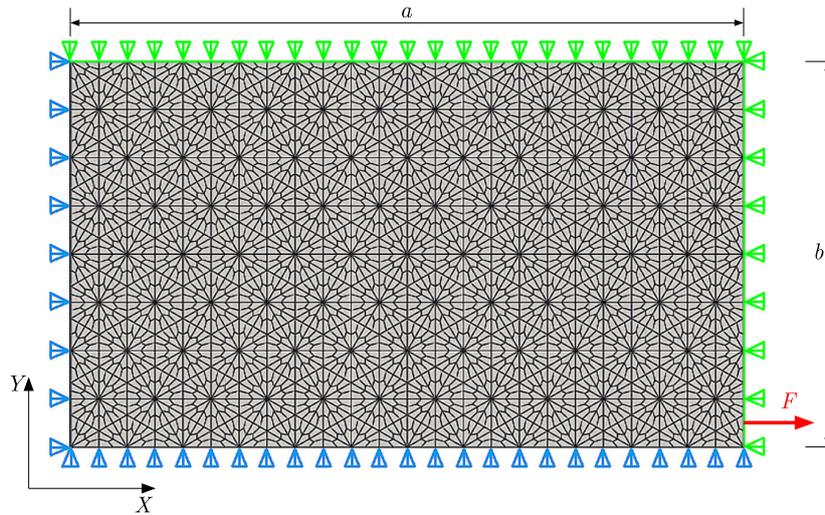


Fig. 3. Representation of load case 1.

Table 2. Periodic boundary conditions for the three load cases.

Load case	Nodes	u_x	u_y	Observation
1 Traction X	1 ($X = 0$)	0	Free	Symmetry X
	2 ($X = a$)	$\bar{\varepsilon}_{01}a$	Free	–
	3 ($Y = 0$)	Free	0	Symmetry Y
	4 ($Y = b$)	Free	Coupled	–
2 Traction Y	1 ($X = 0$)	0	Free	Symmetry X
	2 ($X = a$)	Coupled	Free	–
	3 ($Y = 0$)	Free	0	Symmetry Y
	4 ($Y = b$)	Free	$\bar{\varepsilon}_{02}b$	–
3 Shear XY	1 ($X = 0$)	Free	0	Asymmetry X
	2 ($X = a$)	Free	0	–
	3 ($Y = 0$)	0	Free	Asymmetry Y
	4 ($Y = b$)	$\bar{\gamma}_{03}b$	Free	–

While performing the FEM simulations, we observed that all the solutions are isotropic. This can be explained by the way the periodic model is constructed with the rotation of the base triangle into forming a hexagonal RVE and the repeating of the RVE. In this case, we can use only the first load case in order to determine all the mechanical properties of a configuration.

For the present problem, the parameter of interest is the effective Poisson’s coefficient which can be obtained using the relation:

$$\nu_{12eff} = -\frac{\bar{\varepsilon}_y}{\bar{\varepsilon}_x} = -\frac{\frac{1}{V} \sum_{i=1}^{NE} V_i \varepsilon_{yi}}{\frac{1}{V} \sum_{i=1}^{NE} V_i \varepsilon_{xi}}, \tag{2.1}$$

where ν_{12eff} – effective Poisson’s coefficient, $\bar{\varepsilon}_y$ – average strain in the y -direction, $\bar{\varepsilon}_x$ – average strain in the x -direction, V – total volume of the analysis domain, NE – number of finite elements in the domain, V_i – volume of element i , ε_{yi} – average strain in y -direction in element i , ε_{xi} – average strain in x -direction in element i .

Validating the performance of an optimization algorithm requires knowledge of the optimal solution. To identify the optimal solution, the brute force method is applied by evaluating all

possible configurations. For the described problem with 16 material domains in the base area, there are 12,870 possible solutions. The distribution of Poisson's coefficient values for all solutions is illustrated in Fig. 4 as a histogram. Notably, only 1.2% of the solutions exhibit a negative Poisson's coefficient.

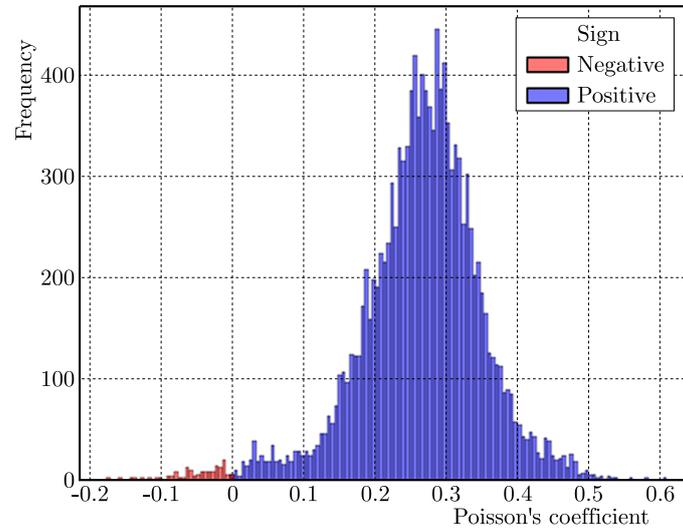


Fig. 4. Histogram of Poisson's coefficient for all solutions.

In evaluating our optimization algorithm, we choose the first three values as optimal solutions, representing almost 0.1% of all solutions. The periodic model of the chosen best solutions and the value for Poisson's coefficient are presented in Fig. 5.

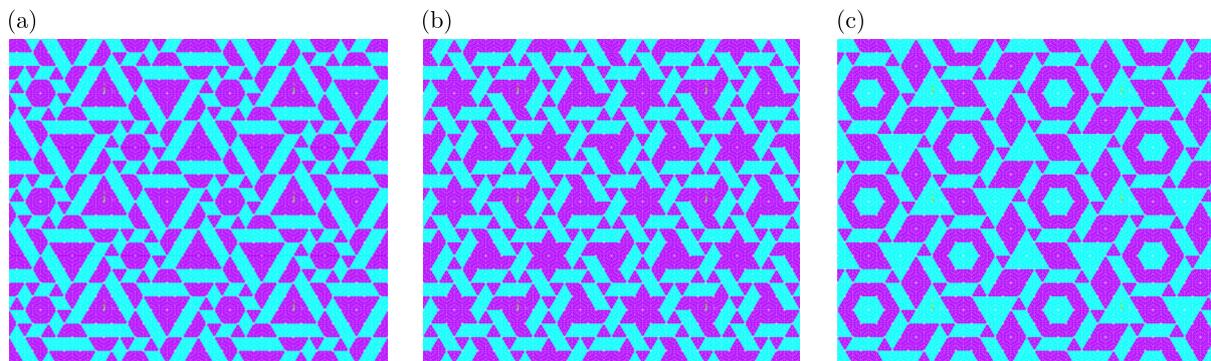


Fig. 5. Optimal solutions for 16 material domains base triangle:
(a) 1st best, $\nu = -0.174072$; (b) 2nd best, $\nu = -0.155901$; (c) 3rd best, $\nu = -0.142588$.

3. Direct search methods

Analytical methods are renowned for their rapid convergence rates. However, direct search methods offer a significant advantage by not requiring gradient information for either the objective function or constraints. This characteristic is particularly beneficial in the field of topological optimization, where computing the gradients of the objective function can be computationally expensive or, in some cases, infeasible. Instead, direct search methods rely solely on the objective function values from previous iterations to identify the optimal solution.

Enumeration, also known as brute force search, is a method that involves exploring and evaluating all possible combinations of the problem's variables to identify the best solution. While this approach guarantees finding the global optimum, it is often impractical or computationally impossible due to the immense number of combinations that must be evaluated. Nevertheless, the brute force method can be valuable for generating data sets in smaller-scale problems. For

instance, in (Gu *et al.*, 2018a), this method is employed to create a labeled dataset used to train a neural network. Once trained, the neural network can efficiently identify optimal structures within a larger search space, where the problem size makes the brute force approach infeasible.

The greedy algorithm iteratively evaluates a set of configurations, making incremental changes toward the best configuration at each step, until no further improvements to the objective function are possible. However, a key limitation of the greedy algorithm is its inability to achieve a better solution if it must pass through a suboptimal solution as an intermediate step. To address this limitation, the algorithm requires a well-defined variable space and an appropriately chosen iteration step. In (Coropetchi *et al.*, 2022), the greedy algorithm was applied to design repetitive composite cellular microstructures aimed at maximizing stiffness in two orthogonal directions. The flowchart of the greedy algorithm is presented in Fig. 6.

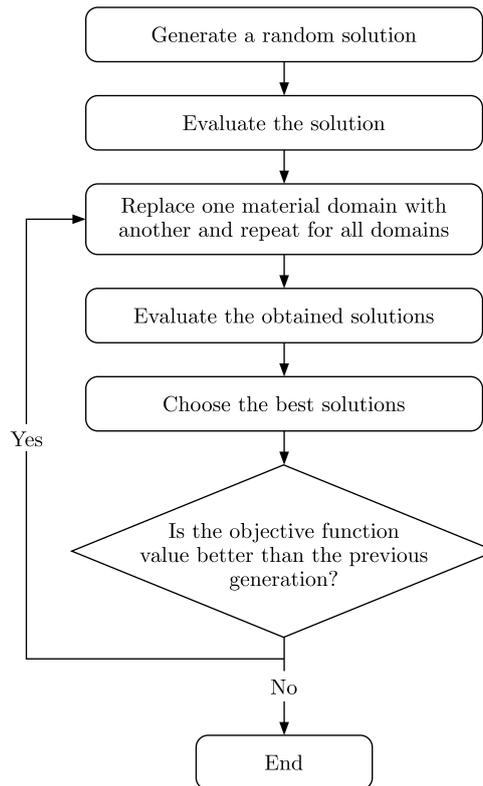


Fig. 6. Flowchart of the greedy algorithm.

As previously discussed, the greedy algorithm evaluates a set of configurations and iteratively makes changes toward the best configuration until no further improvements to the objective function can be achieved. For the described problem, the process begins with a random material distribution. The algorithm evaluates all potential configurations derived from this initial setup by swapping pairs of material domains with different materials. Specifically, in the case of the base triangle, there are 16 material domains – eight of material #1 and eight of material #2. At each iteration, the greedy algorithm examines 64 possible configurations resulting from exchanging pairs of different materials. It selects the configuration with the best objective function and repeats the process in the next step. The algorithm continues this process, selecting the best modification at each iteration, and stops when no further improvements are found, returning the best solution discovered.

The greedy algorithm offers notable advantages, including simplicity of implementation and a relatively small number of objective function evaluations needed to reach an optimal solution. At each iteration, it converges to a local optimum, which, in some cases, may coincide with the global optimum. However, its primary drawback is its inability to guarantee the global

optimum. This limitation arises from the possibility that achieving a better solution may require simultaneous changes to multiple pairs of material domains – something the algorithm does not account for.

According to Clerc (2006), particle swarm optimization (PSO) is a collective, iterative, and decentralized optimization method that emphasizes cooperation among solutions. This partially random algorithm operates without a selection operator. To identify the global optimum within a solution space, the algorithm begins with a swarm of particles (potential solutions), which share information such as their current positions and corresponding objective function values.

Inspired by the social behavior observed in bird flocks (Engelbrecht, 2007), the algorithm models individual solutions, referred to as particles, “flying” through a multidimensional solution space. The movement of each particle is influenced by socio-psychological factors, specifically the tendency of individuals to compete and learn from the success of others. As a result, a particle’s position updates are guided by its own experience as well as the knowledge and performance of its neighboring particles. The flowchart of the PSO algorithm is shown in Fig. 7.

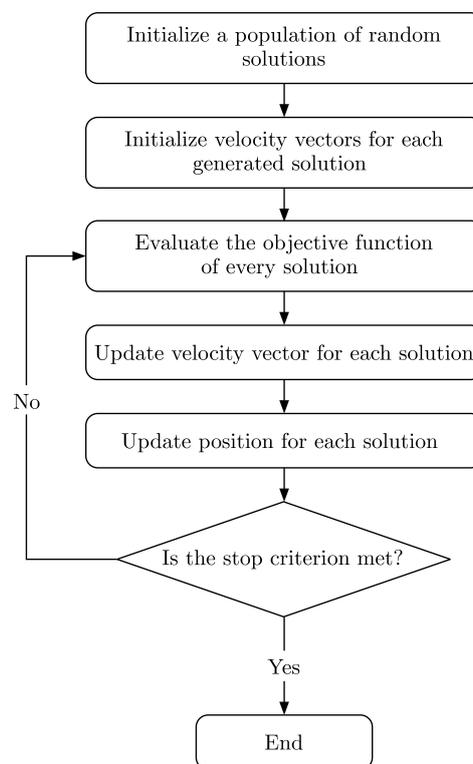


Fig. 7. Flowchart of the PSO algorithm.

This algorithm was proven to be efficient in many optimization problems, among which we can mention Arjomandi *et al.* (2024), Perez and Behdinan (2007).

4. Results and discussions

The optimization algorithms – greedy and particle swarm optimization – were developed using Python, leveraging its simplicity and extensive library support for efficient implementation. The integration of the PyAnsys package facilitated seamless interaction with Ansys for finite element analysis, enabling the algorithms to utilize structural simulations for objective function evaluation. In Table 3 are highlighted key computation times for the brute force method across various configurations. The data presented below was obtained using a computer with the following specifications: Intel® Xeon® W-2104 CPU @ 3.20 GHz, 32 GB DDR5 RAM, and an NVIDIA Quadro P2000 4 GB GDDR5 graphics card.

Table 3. Estimated time necessary for evaluating all solutions in different configurations.

Material domains in base area	Possible combinations	Estimated time
16	12,870	64350 seconds ~17.87 hours
36	9,075,135,300	1,522,325,936 seconds ~2896.35 years

To assess the success rate – defined as the probability of an algorithm achieving the optimal solution or, in this case, one of the top three optimal solutions outlined in [Section 2](#) – each algorithm was executed 100 times for the problem involving 16 material domains in the base area. Additionally, to evaluate the performance of the algorithms, the number of objective function evaluations required to reach the optimum was also recorded. The results are presented in [Fig. 8](#).

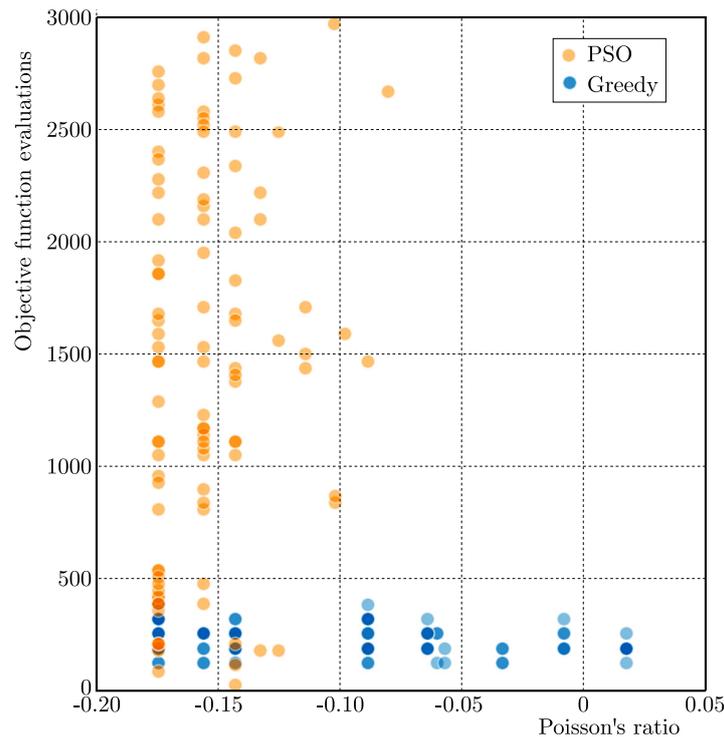


Fig. 8. Final values of Poisson's ratios obtained as depending on number of evaluations – greedy and PSO.

If we analyze this chart, we can see that there is a big difference between how these two optimization algorithms perform. We would like to have an optimization method that would give us a very good objective function value for a small number of evaluations. In this figure, we can see that the greedy algorithm always returns a final solution after a small number of function evaluations but the value is not very good. At the same time, PSO delivers a very good objective function value but with a higher number of evaluations. In this case, the user must make a compromise and should choose which thing is more important: speed or performance. Some summary statistics of these 100 runs are presented in [Table 4](#).

After validating that the two algorithms can find the optimal solutions, we proceeded to the next step in which we performed the optimization process on a problem featuring 36 material domains instead of 16 in the base triangle. For this problem, we showed that it is impossible to apply the brute force method, mainly because there are over 9 billion solutions. We performed 5 runs with each algorithm for this problem and compared the results. To make a direct comparison between these two algorithms, it is necessary to present the evolution of the objective function against the number of evaluations of the function because each algorithm evaluates a different number of solutions at each iteration.

Table 4. Summary statistics of the 100 runs performed on greedy and PSO.

Algorithm	Greedy	PSO
Reached 1st best	27	41
Reached 2nd best	9	26
Reached 3rd best	14	17
Success rate	50 %	84 %
Auxetic	90	100
Mean – objective function value	-0.103819	-0.154334
Mean – number of evaluations	231.04	1470.90
Median – number of evaluations	192	1470
Standard deviation – number of evaluations	67.67	834.05

We can see in Fig. 9 that in the beginning PSO performs better than greedy but then reaches a plateau indicating a blocking in a local minimum. For the greedy algorithm, we can see a steady decrease in the objective function value until it reaches a dead end and for the long run it performs better than PSO in 2 out of 5 runs. The periodic representations of the 3 best obtained configurations are shown in Fig. 10.

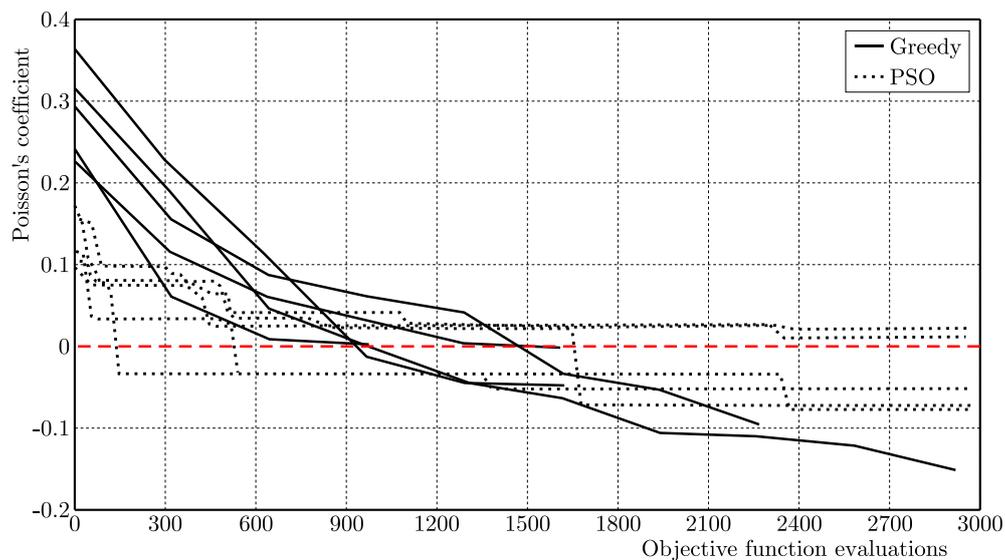


Fig. 9. Objective function evolution vs. the number of evaluations needed for each algorithm – greedy and PSO.

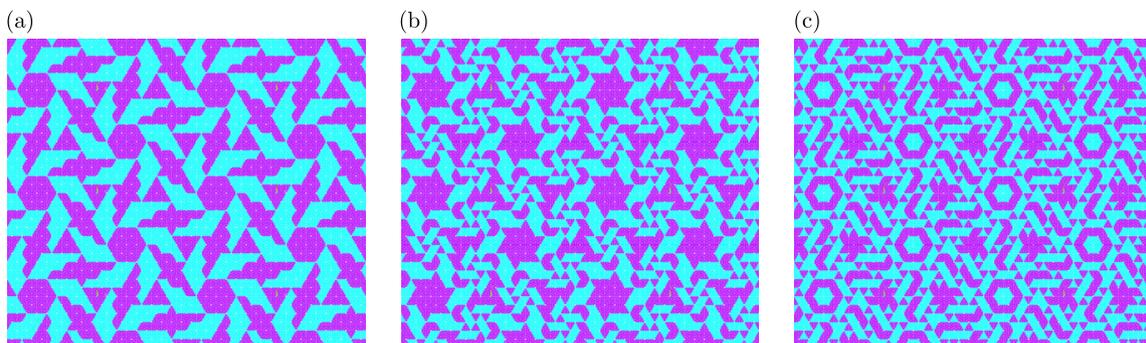


Fig. 10. Best configurations found for the 36 material domains base triangle – periodic model: (a) 1st best – greedy, $\nu = -0.152223$; (b) 2nd best – greedy, $\nu = -0.096786$; (c) 3rd best – PSO, $\nu = -0.074124$.

5. Conclusions

This study evaluates the performance of two optimization algorithms, greedy and PSO, in solving structural optimization problems. The evaluation focuses on metrics such as convergence speed, solution quality, computational efficiency, and robustness. The greedy algorithm demonstrates impressive convergence speed, making it computationally efficient and capable of finding good solutions with minimal cost. However, it has a notable limitation: its tendency to become stuck in suboptimal solutions, as it cannot explore solution spaces requiring intermediate steps through less favorable configurations. In contrast, PSO exhibits a higher success rate, reaching 84 % for the low-dimensional problem with 16 material domains. Both algorithms successfully attain the global maximum for this problem, but the computational efforts required by PSO are notably higher compared to greedy.

When scaling up to the more complex problem of 36 material domains, the differences between the two algorithms become more pronounced. PSO is more likely to encounter deadlocks in local minima, reducing its effectiveness for larger problems, whereas greedy shows greater resilience by consistently reaching a good solution. Although the solutions found by greedy may not always be the global optimum, they are achieved at a lower computational cost, making it a more practical choice for problems with constrained resources. On the other hand, PSO demonstrates its strength by producing higher-quality solutions when computational cost is less of a concern. Overall, the greedy algorithm is an effective choice for cost-efficient optimization, while PSO excels in scenarios where solution quality is prioritized over computational efficiency.

Acknowledgments

Iulian-Constantin Coropetchi acknowledges the grant awarded by the Ministry of Education from Romania for completing his PhD studies under contract no. 06.19/05.10.2020.

References

1. Arjomandi, M.A., Mousavi Asl, S.S., Mosallanezhad, B., & Hajiaghahi-Keshteli, M. (2024). A metaheuristic-based comparative structure for solving discrete space mechanical engineering problem. *Annals of Operations Research*. <https://doi.org/10.1007/s10479-024-06052-y>
2. Bendsøe, M.P., & Sigmund, O. (2004). *Topology optimization. Theory, methods and applications* (2nd ed.). Berlin, Heidelberg: Springer-Verlag. <https://doi.org/10.1007/978-3-662-05086-6>
3. Casalotti, A., D'Annibale, F., & Rosi, G. (2020). Multi-scale design of an architected composite structure with optimized graded properties. *Composite Structures*, 252, Article 112608. <https://doi.org/10.1016/j.compstruct.2020.112608>
4. Clerc, M. (2006). *Particle swarm optimization*. ISTE. <https://doi.org/10.1002/9780470612163>
5. Coropetchi, I.C., Vasile, A., Soroan, Ş., Picu, C.R., & Constantinescu, D.M. (2022). Stiffness optimization through a modified greedy algorithm. *Procedia Structural Integrity*, 37, 755–762. <https://doi.org/10.1016/j.prostr.2022.02.006>
6. Engelbrecht, A.P. (2007). *Computational intelligence: An introduction*. John Wiley & Sons, Ltd. <https://doi.org/10.1002/9780470512517>
7. Ghiasi, H., Fayazbakhsh, K., Pasini, D., & Lessard, L. (2010). Optimum stacking sequence design of composite materials Part II: Variable stiffness design. *Composite Structures*, 93(1), 1–13. <https://doi.org/10.1016/j.compstruct.2010.06.001>
8. Ghiasi, H., Pasini, D., & Lessard, L. (2009). Optimum stacking sequence design of composite materials Part I: Constant stiffness design. *Composite Structures*, 90(1), 1–11. <https://doi.org/10.1016/j.compstruct.2009.01.006>
9. Gu, G.X., Chen, C.T., & Buehler, M.J. (2018a). *De novo* composite design based on machine learning algorithm. *Extreme Mechanics Letters*, 18, 19–28. <https://doi.org/10.1016/j.eml.2017.10.001>

10. Gu, G.X., Chen, C.T., Richmond, D.J., & Buehler, M.J. (2018b). Bioinspired hierarchical composite design using machine learning: Simulation, additive manufacturing, and experiment. *Materials Horizons*, 5, 939–945. <https://doi.org/10.1039/c8mh00653a>
11. Harish, B., Eswara Sai Kumar, K., & Srinivasan, B. (2020). Topology optimization using convolutional neural network. In R.R. Salagame, P. Ramu, I. Narayanaswamy, D.K. Saxena (Eds.), *Advances in Multidisciplinary Analysis and Optimization. Proceedings of the 2nd National Conference on Multidisciplinary Analysis and Optimization* (pp. 301–307). Singapore: Springer. https://doi.org/10.1007/978-981-15-5432-2_26
12. Kumar, S., Tan, S., Zheng, L., & Kochmann, D.M. (2020). Inverse-designed spinodoid metamaterials. *npj Computational Materials*, 6, Article 73. <https://doi.org/10.1038/s41524-020-0341-6>
13. Li, T., Liu, F., & Wang, L. (2020). Enhancing indentation and impact resistance in auxetic composite materials. *Composites Part B: Engineering*, 198, Article 108229. <https://doi.org/10.1016/j.compositesb.2020.108229>
14. Luu, N.G., & Banh, T.T. (2023). A novel preconditioned conjugate gradient multigrid method for multi-material topology optimization. *arXiv*. <https://doi.org/10.48550/arXiv.2301.07457>
15. Pan, C., Han, Y., & Lu, J. (2020). Design and optimization of lattice structures: A review. *Applied Sciences*, 10(18), Article 6374. <https://doi.org/10.3390/app10186374>
16. Perez, R.E., & Behdinan, K. (2007). Particle swarm approach for structural design optimization. *Computers and Structures*, 85(19–20), 1579–1588. <https://doi.org/10.1016/j.compstruc.2006.10.013>
17. Ptochos, E., & Labeas, G. (2012). Elastic modulus and Poisson's ratio determination of micro-lattice cellular structures by analytical, numerical and homogenisation methods. *Journal of Sandwich Structures & Materials*, 14(5), 597–626. <https://doi.org/10.1177/1099636212444285>
18. Sait, S.M., Mehta, P., Gürses, D., & Yildiz, A.R. (2023). Cheetah optimization algorithm for optimum design of heat exchangers. *Materials Testing*, 65(8), 1230–1236. <https://doi.org/doi:10.1515/mt-2023-0015>
19. Sandhu, R. (1971). Parametric study of optimum fiber orientation for filamentary sheet. In *Technical Memorandum FBC-71-1*. Air Force Flight Dynamics Laboratory, Director of Laboratories, Air Force Systems Command, Wright Patterson Air Force Base, Ohio.
20. Sorohan, S., Constantinescu, D.M., Sandu, M., & Sandu, A.G. (2018). On the homogenization of hexagonal honeycombs under axial and shear loading. Part I: Analytical formulation for free skin effect. *Mechanics of Materials*, 119, 74–91. <https://doi.org/10.1016/j.mechmat.2017.09.003>
21. Tavakoli, R. (2014). Multimaterial topology optimization by volume constrained Allen-Cahn system and regularized projected steepest descent method. *Computer Methods in Applied Mechanics and Engineering*, 276, 534–565. <https://doi.org/10.1016/j.cma.2014.04.005>
22. Vasile, A., Coropețchi, I.C., Sorohan, Ș., Picu, C.R., & Constantinescu, D.M. (2022). A simulated annealing algorithm for stiffness optimization. *Procedia Structural Integrity*, 37, 857–864. <https://doi.org/10.1016/j.prostr.2022.02.019>
23. Yoon, G.H. (2021). Multiphysics topology optimization scheme considering the evaporation cooling effect. *Computers & Structures*, 244, Article 106409. <https://doi.org/10.1016/j.compstruc.2020.106409>
24. Yu, Y., Hur, T., Jung, J., & Jang, I.G. (2019). Deep learning for determining a near-optimal topological design without any iteration. *Structural and Multidisciplinary Optimization*, 59(3), 787–799. <https://doi.org/10.1007/s00158-018-2101-5>
25. Zhang, Y., Peng, B., Zhou, X., Xiang, C., & Wang, D. (2019). A deep Convolutional Neural Network for topology optimization with strong generalization ability. *arXiv*. <https://doi.org/10.48550/arXiv.1901.07761>
26. Zhu, B., Zhang, X., Zhang, H., Liang, J., Zang, H., Li, H., & Wang, R. (2020). Design of compliant mechanisms using continuum topology optimization: A review. *Mechanism and Machine Theory*, 143, Article 103622. <https://doi.org/10.1016/j.mechmachtheory.2019.103622>