Exercise 7: Differential Evolution

Due date: *December 4th, 2025 (push to GitHub repo before the lecture)*
General Requests: *Same as before.*

7.1 Differential Evolution

We had a range of different optimization problem types now: benchmarks (HC), continuous parameter tuning (GA), image "regression" (GA variants), path-finding problems (SA), scheduling problems (ACO), and feature selection (PSO). Another very interesting application of metaheuristics is **shape optimization** [2][3][4]. In the current assignment, we'll thus tackle a geometric / spatial optimization problem using Differential Evolution[1, Chapter 12 and 19].

- we'll take a look at the moving sofa problem, a 2D-simplification of a real-world furniture moving problem
- assume a hallway shaped like an "L", both legs one unit wide. your goal is to find a shape of **largest area** that can still be moved around the corner without 'getting stuck'[1]
- implement a Differential Evolution algorithm to 'evolve' a solution shape to this problem. **important**: use this helper script to do some of the 'heavy lifting' (collision detection, problem setup, animation, etc.) for you; as always, feel free to implement your own solution, however
- the key issue here is **candidate solution representation**. as DE takes **differences** between candidate solutions, you need to repesent your shapes in a consistent way[2]. a simple approach would be to use some kind of **radial representation** that ensures that each dimension of your candidate solution corresponds to a specific angle
- another crux is **candidate diversity**. the problem looks simple, but initializing a diverse population of valid shapes is not trivial and takes serious effort. if you only initialize with radial (star) shapes, you will likely end up with a circle as the best solution. think about how to create more complex shapes (e.g., rectangles, ellipses, combinations of multiple shapes such as barbells, etc.) for your initial population - the more diverse, the better
- for your objective function, you'll find a suggestion in the helper script; you want to maximize the area of your shape ultimately, but you might also want to penalize unwanted behavior (e.g., shapes that cannot be moved around the corner, shapes that are not simply connected, etc.). the shapely.Polygon class has some nice built-in functions that can help you quite a lot with area calculation, finding overlaps, etc., even if it's not the most efficient way to represent shapes
- **important**: a solution to this problem - proposed by Hammersley in 1968 with an area of 2.2074 a.u. - is shown in figure 1[3]. in this little assignment, you will likely **not** end up with a fancy shape even **remotely** similar to figure 1; but that's totally fine. the goal here is to get familiar with DE and shape optimization, to get a feeling for the challenges and pitfalls involved (and there are many), and to maybe compete a little with the other groups on who can find the shape with the largest area. explore, experiment, best case even have some fun with it, but don't stress too much about the final result

7.2 Analysis and Reporting

Add a streamlit page on Simulated Annealing to the documentation you created for the last exercises.
Your final documentation webpage should contain the following sections:

1. Introduction - Overview of the algorithm: basics, strengths / weaknesses, complexity, ...
2. Methods - Describe your implementation: pros/cons, relevant parameters/functions, critical design choices, ...
3. Results - Display the best shape you found after optimization (optimally even animate how the best shape in the population changes across iterations), its area, an animation of the moving process to prove its feasibility, etc.
4. Discussion - Analyze your findings: expected versus unexpected results, solution quality, efficiency, limitations, possible improvements, ...

The documentation doesn't need to be absurdly long, but it should give a comprehensive overview of the topic that you can use later on to study for the exam.

[1] It may touch the walls, but it cannot pass through them.
[2] Consider two identical rectangles perfectly overlaying each other; if you parametrize one of them starting at the right upper corner, the other one at the lower left corner, the difference will nonetheless not be zero.
[3] Gerber later proposed another shape with an area of 2.2195 a.u., though it's still unclear what the actual optimum solution to the problem is.

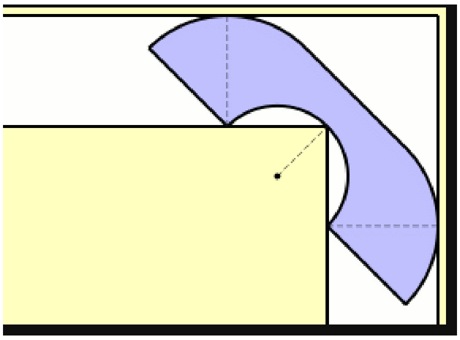

Figure 1: The "Hammersley sofa" attains an area of 2.2074 a.u. — meanwhile, even larger candidate shapes have been found (image from [5]).

References

[1] D. Simon, Evolutionary Optimization Algorithms. Wiley, 2013. [Online]. Available: https://research-1ebsco-1com-1195qzf320241.perm.fh-joanneum.at/c/kofjhs/search/details/4sh2uyq6wn?db=nlebk&db=nlabk

[2] Constrained multi-objective aerodynamic shape optimization via swarm intelligence, GECCO '14, 2014.

[3] M. Alam and T. H. Kwok, ``Multidisciplinary optimization of shoe midsole structures using swarm intelligence,'' Structural and Multidisciplinary Optimization, 2024. [Online]. Available: https://link.springer.com/article/10.1007/s00158-024-03845-4

[4] S. Z. Martínez, A. Arias-Montaño, and C. A. C. Coello, ``Constrained multi-objective aerodynamic shape optimization via swarm intelligence,'' Structural and Multidisciplinary Optimization, 2024. [Online]. Available: https://dl.acm.org/doi/pdf/10.1145/2576768.2598372

[5] ``Moving sofa problem,'' Wikipedia. [Online]. Available: https://en.wikipedia.org/wiki/Moving_sofa_problem