Performance of Jellyfish Search Optimiser on Projection Pursuit Problems

Alice Anonymous[a,*], Bob Security[b], Cat Memes[b], Derek Zoolander

[a]*Some Institute of Technology, Department Name, Street Address, City, Postal Code*

[b]*Another University, Department Name, Street Address, City, Postal Code*

Abstract

This is the abstract. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vestibulum augue turpis, dictum non malesuada a, volutpat eget velit. Nam placerat turpis purus, eu tristique ex tincidunt et. Mauris sed augue eget turpis ultrices tincidunt. Sed et mi in leo porta egestas. Aliquam non laoreet velit. Nunc quis ex vitae eros aliquet auctor nec ac libero. Duis laoreet sapien eu mi luctus, in bibendum leo molestie. Sed hendrerit diam diam, ac dapibus nisl volutpat vitae. Aliquam bibendum varius libero, eu efficitur justo rutrum at. Sed at tempus elit.

Keywords: projection pursuit, optimization, jellyfish optimiser

Let's use British English ("American or British usage is accepted, but not a mixture of these")

1. Introduction [Nicolas and Jessica]

The artificial jellyfish search (JS) algorithm [1] is a swarm-based metaheuristic optimisation algorithm inspired by the search behaviour of jellyfish in the ocean. It is one of the newest swarm intelligence algorithms [2], which was shown to have stronger search ability and faster convergence with few algorithmic parameters compared to classic optimization methods [1]-[3].

The rest of the paper is organised as follows: Section 2 introduces the projection pursuit method, including the indexes function and optimisation. Section 3 introduces the jellyfish optimiser and proposes mathematical expressions to measure the . Section 4 applies the jellyfish optimiser through different projection pursuit problems with varying dimensions and index functions. Section 5 concludes the paper.

2. Projection pursuit, index functions and optimisation [Di and Sherry]

3. The jellyfish optimiser and property for good optimisers [Nicolas and Jessica]

The jellyfish optimiser (JSO) mimics the natural movements of jellyfish, which include passive and active motions driven by ocean currents and their swimming patterns, respectively. In the context of optimization, these movements are abstracted to explore the search space in a way that balances exploration (searching new areas) and exploitation (focusing on promising areas). The algorithm aims to find the optimal solution by adapting the jellyfish's behavior to navigate towards the best solution over iterations [1].

Below is the pseudo-code for this visualisation application.

*Corresponding author

Email addresses: `alice@example.com` (Alice Anonymous), `bob@example.com` (Bob Security), `cat@example.com` (Cat Memes), `derek@example.com` (Derek Zoolander)

[Put the code in, add specifics to this visualisation application]

The JSO implementation involves several key parameters that control its search process in optimization problems. These parameters are designed to guide the exploration and exploitation phases of the algorithm. While the specific implementation details can vary depending on the version of the algorithm or its application, we focus on two main parameters that are most relevant to our application: the number of jellyfish and drift.

Laa and Cook [4] has proposed five criteria for assessing projection pursuit indexes (smoothness, squintability, flexibility, rotation invariance, and speed). Since not all the properties affects the execution of the optimisation, here we consider the three relevant properties (smoothness, squintability, and speed), and propose three metrics to evaluate these three properties.

3.1. Smoothness

An intuitive way to measure smoothness would be to find how many continuous derivatives exist. We make use of Sobolev spaces:

[def of sobolev space]

Smoothness would then be the highest p such that the index function belongs to $W^{p,\infty}$. We can make it weaker by considering $W^{p,q}$ Sobolev spaces.

[backup plan when things are not differentiable]

3.2. Squintability

A large squint angle means the function is easy to optimize, because you don't need to be very very close to the perfect view to see the structure. A small squint angle means that the derivative of the index function can still be very large values near the optimal point.

OR that the values taken by the index function can be clustered into two clear groups: one small group of high values near the optimum, and a large group of the remaining small values. OR the rate of change of gradient? Gradient changes very drastically when it's near optimal?

[backup plan when things are not differentiable]

To the best of our knowledge, this is the first attempt to measure the notion of squintability.

3.3. Speed

Computational complexity (in big O notation, with respect to the sample size) of computing the index function.

4. Application [Di and Sherry]

The jellyfish optimiser has been implemented in the tourr package [5] and we will use the diagnostic plots proposed in the ferrn package [6] to visualise the optimisation process.

4.1. Going beyond 10D

The pipe-finding problem is initially used to investigate indexes and optimisers in Laa and Cook [4], and we extend it from a 6D problem to a 12D problem.

Jellyfish optimiser, as a multi-start algorithm, is efficient in [...] for high-dimensional problems

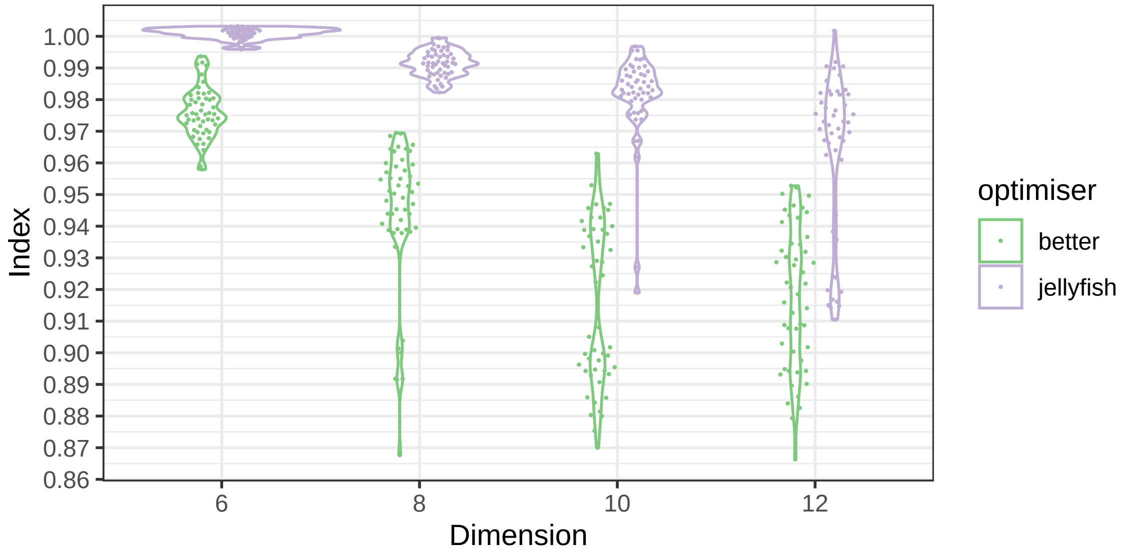

Figure 1: sthis sdfaksdlf

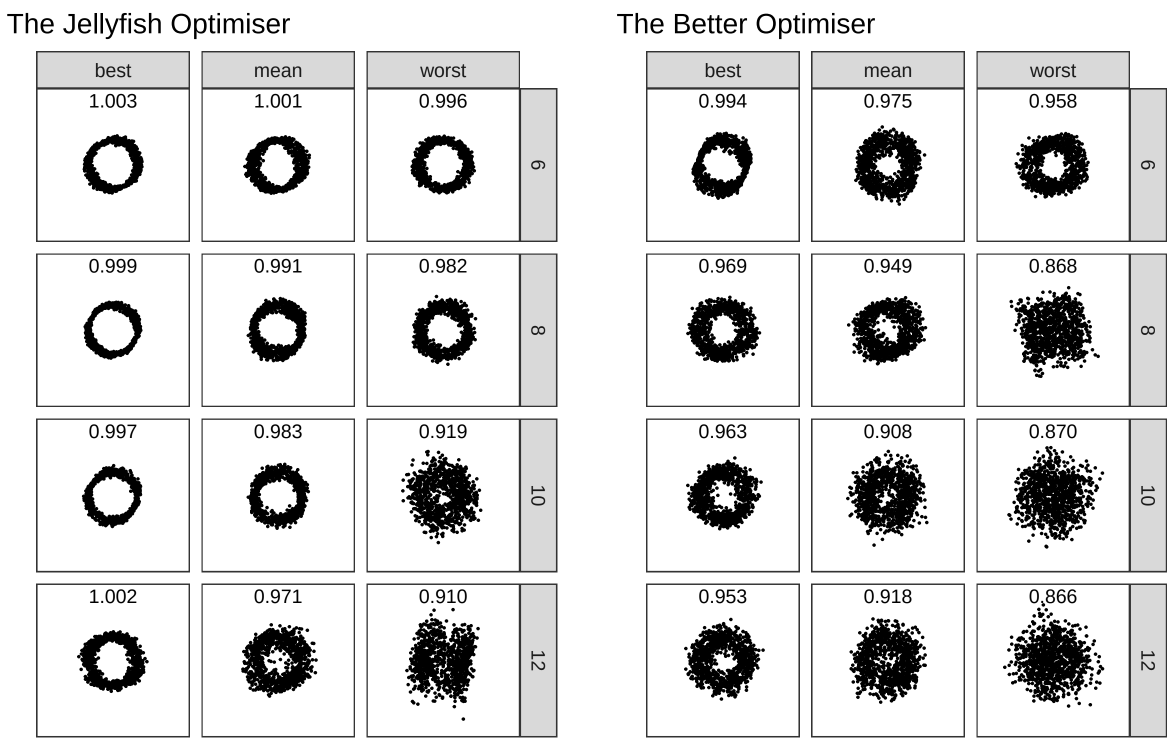

Figure 2: sthis sdfaksdlf

4.2. On skewness and kurtosis index

4.3. Another data example

5. Conclusion [Di and Sherry]

References

[1] J.-S. Chou, D.-N. Truong, A novel metaheuristic optimizer inspired by behavior of jellyfish in ocean, Applied Mathematics and Computation 389 (2021) 125535. doi:10.1016/j.amc.2020.125535.

[2] K. Rajwar, K. Deep, S. Das, An exhaustive review of the metaheuristic algorithms for search and optimization: taxonomy, applications, and open challenges, Artificial Intelligence Review (2023) 1–71. doi:10.1007/s10462-023-10470-y.

[3] J.-S. Chou, A. Molla, Recent advances in use of bio-inspired jellyfish search algorithm for solving optimization problems, Scientific Reports 12 (2022) 19157. doi:10.1038/s41598-022-23121-z.

[4] U. Laa, D. Cook, Using tours to visually investigate properties of new projection pursuit indexes with application to problems in physics, Computational Statistics 35 (2020) 1171–1205. doi:10.1007/s00180-020-00954-8.

[5] H. Wickham, D. Cook, H. Hofmann, A. Buja, tourr: An R Package for Exploring Multivariate Data with Projections, Journal of Statistical Software 40 (2011) 1–18. doi:10.18637/jss.v040.i02.

[6] H. S. Zhang, D. Cook, U. Laa, N. Langrené, P. Menéndez, Visual diagnostics for constrained optimisation with application to guided tours, The R Journal 13 (2021) 624–641. doi:10.32614/RJ-2021-105.