Paul Seward
142012
MSC 8660
Final Project Report (Mad data scientist level) April 14, 2024

Insights into Metaheuristics: Visualizing Swarm Behavior for Optimization

The visualization tool can be accessed at: https://github.com/PaulASeward/metaheuristic-visualization
(Instructions and example usage can be found in the README.md)

Original Research Hypothesis:

The original hypothesis of this project suggested that using dimension reduction techniques to visualize the spatial dynamics of a Particle Swarm Optimization (PSO) algorithm, combined with Reinforcement Learning, will reveal clusters of swarm behavior, indicating possible stalemates and inefficient strategies. Additionally, I proposed potential relationships between observed states and chosen actions could become detectable from analysis with this visual tool. These patterns could ultimately guide better action selection, thus enhancing the PSO algorithm's performance and decision-making ability.

Adjusted Methodology:

In this project, I adopted a two-dimensional representation for the swarm's search space, a decision driven by two primary considerations:

1. **Lack of Inter-Dimensional Correlation**:

Preliminary analysis indicated a weak or non-existent relationship between many of the dimensions in our limited dimensional search space. Such a lack of correlation undermines the effectiveness of dimensionality reduction techniques like PCA and t-SNE, which rely on capturing significant variance across dimensions to project a meaningful lower-dimensional space. Given this, maintaining a high-dimensional model would add complexity without corresponding interpretative benefits.

2. **Distortion in Spatial Representation**:

Utilizing a high-dimensional search space in a visual model introduces significant challenges in accurately portraying the search behavior. As the dimensions increase, so does the complexity of the search landscape, leading to potential distortions when these dimensions are forced into a two- or three-dimensional display. Important details about the landscape, such as nuanced search basins or unexplored areas, can become obscured or misrepresented in such reductions.

Data and Tools

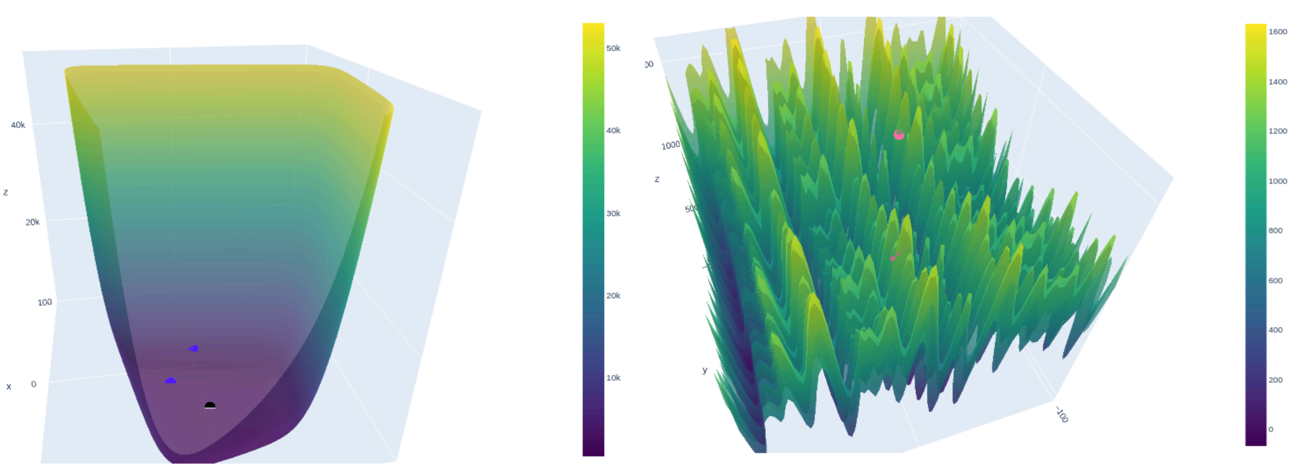
Data consisted of observations from a 10-particle swarm navigating a 2-dimensional space across one of 28 available benchmark functions from the 2013 CEC Particle Optimization set, featuring well-known functions such as Rastrigin's and Schwefel's. The function evaluation (FE) budget was set at 20,000 (10,000 per dimension), derived from 20 episodes, each comprising 100 evaluation steps involving 10 particles. The visualization tool, detailed on its GitHub repository, enabled dynamic and interactive exploration of swarm behavior across these multiple benchmark functions.

Algorithm 3.14 Template of the particle swarm optimization algorithm.

Random initialization of the whole swarm ;
Repeat
Evaluate $f(x_i)$;
For all particles i
Update velocities:
$v_i(t) = v_i(t-1) + \rho_1 \times (p_i - x_i(t-1)) + \rho_2 \times (p_g - x_i(t-1))$;
Move to the new position: $x_i(t) = x_i(t-1) + v_i(t)$;
If $f(x_i) < f(pbest_i)$ **Then** $pbest_i = x_i$;
If $f(x_i) < f(gbest)$ **Then** $gbest = x_i$;
Update(x_i, v_i) ;
EndFor
Until Stopping criteria

(Talbi, 2009)

Visualized datasets, as shown in the below figures, were either generated on-the-fly using a simplified PSO algorithm for the user-selected function or were preloaded from completed experiments of a Reinforcement Learning (RL) implementation of PSO, ie. an int. As outlined in the figure above, this generated data comes with adjustable hyperparameters such as the social learning component (Swarm's Best Gravity), cognitive learning component (Particle's Best Gravity), inertia weight, particle best replacement threshold, and maximum velocity.



Functions 19 (left): Rotated Expanded Griewank's Function and Function 14 (right): Schwefel's Function

In the case of preloaded data, the current values of these hyperparameters are displayed, along with the previous action decided by the swarm. The available actions, which adjust these parameters, were selected by an RL agent based on the best policy (set of actions) derived from an observation space that includes three observations per particle:

1. The magnitude of velocity at the episode's final step,
2. The number of times a particle established a new best location (replacement counts),
3. The particle's relative fitness compared to the current best of the swarm.

Results

The visual analysis highlighted the following key findings:

- **Swarm Dynamics**: The tool effectively displayed how particles explore and exploit the search space, revealing tendencies towards certain regions or behaviors, as individuals and collectively as a swarm in the reduced three-dimensional view. This was crucial in identifying trends in phases of rapid exploration versus periods of exploitation.

- **Observational Data Limitations**: The tool highlighted limitations in the observational data available to the learning agent. Specifically, the observations collected do not sufficiently capture the nuances of the different functions. Since these varying function topologies affect the performance of a swarm's action policy, it is critical for the swarm to better recognize its environment with respect to the topology of the function.

- **Missed Learning Opportunities:** Although initial plans included using clustering techniques to analyze observed states and action mappings, the visualization tool revealed that the current observations were severely limited. Specifically, it illuminated the lack of a connecting path for an agent to learn the performance of a search strategy.

Discussion

The decision not to employ clustering techniques stemmed from a pivotal realization brought about by the visualization tool. The data gathered from the swarm's behavior did not sufficiently capture the complexity required for a detailed behavioral analysis. Specifically, the tool demonstrated that the observational data currently collected—velocity magnitude, frequency of finding a new optimal position (replacement), and relative fitness—are overly simplistic. These metrics do not provide the nuanced understanding necessary to effectively train the reinforcement learning model to recognize and adapt to the complex dynamics of different search functions.

The visualization also shed light on the need for careful control of particle velocities and exploration strategies. It revealed how varying velocities could dramatically affect exploration efficiency, showing that too rapid or too slow movements could either skip over potential optimal areas or get stuck in suboptimal ones. This insight led to a deeper understanding of the need for a balanced exploration strategy that adapts to the function's topography being optimized.

These findings also emphasized the need for a non-linear scaling approach within the visualization tool. This feature proved essential in dynamically adjusting the visualization to match the swarm's exploration phase, thereby providing a clearer picture of the emergent behaviors and patterns that could go unnoticed in a linear scaled model. The sensitivity of the tool to these nuances mirrors the desired sensitivity of the swarm's exploration strategies, ensuring that both the tool and the swarm can adapt to the complexities of the search space in real-time.

Thesis Implications

The insights derived from using the visualization tool are pivotal for advancing my thesis on intelligent swarms in metaheuristic algorithms. The primary insight is that the current data collected by the algorithm is insufficient for fostering an understanding that leads to effective decision-making. This underfitting of the model to the complexity of the search space is akin to learning to navigate with incomplete sensory input, such as a driver seeing only fragments of the road or a language model working with partial sentences.

To enhance this, I aim to explore the implementation of more sophisticated data collection and analysis techniques. Approaches such as convolutional layers or attention mechanisms could potentially provide a richer feature vector from which the neural network can learn. These would enhance the swarm's ability to recognize and adapt to the topological characteristics of the search space, particularly in complex functions like the Rotated Expanded Griewank's and Schwefel's Function, which exhibit unique challenges due to their smooth and rugged topological variations respectively.

Inefficiencies in learning how different exploration strategies perform were also observed. The agent can test out different combinations of actions to learn the value of different exploration and exploitation strategies from the final reward of how closely the swarm approached a function's true optimum. If I could include performance metrics of search space exploration as additional observations, more frequent connections could be made by the agent on how well a strategy performs and improve the learnability of exploration strategies.

Examples of these metrics include the average number of steps to convergence, defined by the proximity of certain points to each other or the rate at which particles slow their velocities. These measures would provide insights into the efficiency of the swarm's exploration strategy, helping to determine if the swarm quickly finds a poor search basin or takes too long to find an optimal one. Recognizing when to rapidly explore the search space or proceed more methodically could drastically improve the optimization processes.

I present two new facets to explore in my graduate research:

1. **Enhanced Data Capture**: By integrating more comprehensive metrics and employing techniques such as Convolutional Layers or Attention Mechanisms, the swarm could potentially learn from a richer dataset that includes detailed positional data and function evaluations. This could transform how the swarm perceives its environment, leading to more informed and effective exploration strategies.
2. **Exploration Strategy Performance Measures**: Introducing metrics to quantify the effectiveness of exploration strategies in real-time could significantly enhance the swarm's ability to adapt its strategies based on the topology of the search space and the timing within its lifecycle.

Conclusion

This research confirmed the original hypothesis to a degree, demonstrating that visualization and dimension reduction could expose critical behaviors and inefficiencies within PSO operations. However, it also highlighted the limitations of current data strategies and the need for more robust methodologies to fully leverage the potential of visualization in understanding and enhancing metaheuristic algorithms. Future work will focus on expanding observational data and refining the integration of reinforcement learning to better guide the swarm's exploratory and exploitative behaviors.

This study sets the groundwork for more informed and effective use of visualization tools in optimizing metaheuristic algorithms, proposing a path forward that involves both deeper analytical techniques and more intuitive, interactive exploration tools.

References:

Talbi, El-Ghazali (2009). Metaheuristics: From Design to Implementation. John Wiley & Sons,.