Large Language Models for the Automated Analysis of Optimization Algorithms
Supplementary Material

January 2024

Contents

1 Task A

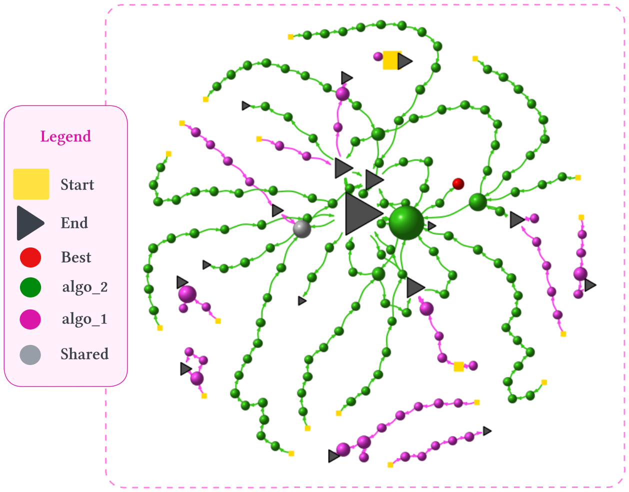

Figure 1: Example of an STN produced by STNWeb that represent the Task A (easy).

Task A consists of two prompts: an easy one and a hard one. The easy case can be visualized in Figure 1. The hard case is designed artificially by incorporating specific values into the prompt that provoke a tie between two of the three algorithms being compared.

1/Easy - Model Input

[BEGIN CONTEXT]
STNWeb is a new web tool for the visualization of the behavior of optimization algorithms such as metaheuristics. It allows for the graphical analysis of multiple runs of multiple algorithms on the same problem instance and, in this way, it facilitates the understanding of algorithm behavior. It may help, for example, in identifying the reasons for a rather low algorithm performance. This, in turn, can help the algorithm designer to change the algorithm in order to improve its performance. STNWeb is designed to be user-friendly. Moreover, it is offered for free to the research community.
[END CONTEXT]
[BEGIN TASK A]
[BEGIN RULES]
These are the general rules of the system:
(1) The more nodes point to nodes of the best fitness (this doesn't assume that it represents the global optimum), the higher the algorithm's quality because it can find better result.
(2) The algorithm that has more connectivity is likely to be more robust. If and only if it finds nodes of the best fitness.
(3) For a minimization problem, indicating that an algorithm is superior involves favoring a smaller average fitness value. Whereas in the case of maximization, declaring an algorithm as better necessitates a higher average fitness value.
[END RULES]
[BEGIN DATA]
Problem: This is a minimization problem.
Features:
- algo_1 has 0 nodes pointing of the best fitness.
- algo_1 has 0.26 connectivity among all the nodes.
- algo_1 has an average fitness of 95.411 across 10 trajectories.
- algo_2 has 1 nodes pointing of the best fitness.
- algo_2 has 0.44 connectivity among all the nodes.
- algo_2 has an average fitness of 78.081 across 10 trajectories.
[END DATA]
[END TASK A]
[BEGIN QUERIES]
Task A: Identify the most effective algorithm for the problem and provide detailed insights in the context of STNWeb. Instructions for your response:
- Clearly specify the winning algorithm by enclosing its name in square brackets with the following structure: [WINNER=ALGORITHM_NAME]; in case of a tie, denote [DRAW].
- Present your response between [BEGIN TASK A] and [END TASK A].
[END QUERIES]

2/Hard - Model Input

[BEGIN CONTEXT]

STNWeb is a new web tool for the visualization of the behavior of optimization algorithms such as metaheuristics. It allows for the graphical analysis of multiple runs of multiple algorithms on the same problem instance and, in this way, it facilitates the understanding of algorithm behavior. It may help, for example, in identifying the reasons for a rather low algorithm performance. This, in turn, can help the algorithm designer to change the algorithm in order to improve its performance. STNWeb is designed to be user-friendly. Moreover, it is offered for free to the research community.

[END CONTEXT]

[BEGIN TASK A]

[BEGIN RULES]

These are the general rules of the system:

(1) The more nodes point to nodes of the best fitness (this doesn't assume that it represents the global optimum), the higher the algorithm's quality because it can find better result.

(2) The algorithm that has more connectivity is likely to be more robust. If and only if it finds nodes of the best fitness.

(3) For a minimization problem, indicating that an algorithm is superior involves favoring a smaller average fitness value. Whereas in the case of maximization, declaring an algorithm as better necessitates a higher average fitness value.

[END RULES]

[BEGIN DATA]

Problem: This is a maximization problem.

Features:

- algo_1 has 8 nodes pointing of the best fitness.
- algo_1 has 4.13 connectivity among all the nodes.
- algo_1 has an average fitness of 10.12 across 10 trajectories.
- algo_2 has 8 nodes pointing of the best fitness.
- algo_2 has 4.10 connectivity among all the nodes.
- algo_2 has an average fitness of 10.12 across 10 trajectories.
- algo_3 has 4 nodes pointing of the best fitness.
- algo_3 has 2.5 connectivity among all the nodes.
- algo_3 has an average fitness of 2.2 across 10 trajectories.

[END DATA]

[END TASK A]

[BEGIN QUERIES]

Task A: Identify the most effective algorithm for the problem and provide detailed insights in the context of STNWeb. Instructions for your response:

- Clearly specify the winning algorithm by enclosing its name in square brackets with the following structure: [WINNER=ALGORITHM_NAME]; in case of a tie, denote [DRAW].
- Present your response between [BEGIN TASK A] and [END TASK A].

[END QUERIES]

1.1 GPT-4-turbo

1/Easy - Model Output - GPT-4-turbo

[BEGIN TASK A]
[winner=algo_2]; Based on the provided data and the rules of the system, it is clear that algo_2 is the most effective algorithm for the minimization problem at hand. This conclusion is drawn from several critical observations:

- Firstly, according to rule (1), the quality of an algorithm is higher when more nodes point to nodes of the best fitness. Algo_2 has 1 node pointing to the node of the best fitness, whereas algo_1 has 0 such nodes. This indicates that algo_2 has a better ability to find more optimal solutions, as it has at least one trajectory reaching a node of the best fitness.
- Secondly, rule (2) states that an algorithm with more connectivity is considered more robust, albeit this is only a positive factor if the algorithm finds nodes of the best fitness. Algo_2 not only has connectivity of 0.44, which is significantly higher than algo_1's connectivity of 0.26, but it also satisfies the condition of finding nodes of the best fitness. This increased connectivity suggests that algo_2 is likely to be more robust, being able to explore and exploit the solution space more effectively.
- Lastly, rule (3) is crucial for this minimization problem, as the superior algorithm should have a smaller average fitness value. Algo_2's average fitness across 10 trajectories is 78.081, which is notably lower than algo_1's average fitness of 95.411. This clearly shows that on average, algo_2 finds better (lower) fitness values, which is the goal of a minimization problem.

Taking all these points into account, it is evident that algo_2 outperforms algo_1 in terms of finding better solutions, robustness, and average fitness for this specific minimization problem, as analyzed through the STNWeb tool.
[END TASK A]

2/Hard - Model Output - GPT-4-turbo

[BEGIN TASK A] Based on the given data and the rules of the system, we can analyze the performance of the algorithms as follows:

- Both algo_1 and algo_2 have the same number of nodes of the best fitness (8 nodes), which is a positive indicator of their quality as per rule (1).
- algo_1 and algo_2 also have nearly identical connectivity (4.13 and 4.10 respectively), which according to rule (2) suggests that they are both likely to be robust since they both find nodes of the best fitness.
- Since this is a maximization problem, we should favor the algorithm with a higher average fitness value. However, algo_1 and algo_2 both have the same average fitness value of 10.12 across 10 trajectories, which is substantially higher compared to algo_3's average fitness of 2.2, as per rule (3).

Given that algo_1 and algo_2 are virtually indistinguishable based on the provided data and the rules set forth, and both outperform algo_3, we can conclude that there is a tie between algo_1 and algo_2.
[winner=draw]
[END TASK A]

1.2 Mixtral-8x7b-instruct-v0.1

1/Easy - Model Output - Mixtral-8x7b-instruct-v0.1

[BEGIN TASK A]
Based on the rules and data provided, we can evaluate the two algorithms according to their ability to find nodes of the best fitness, their connectivity, and their average fitness.
For this minimization problem, we are looking for an algorithm with the highest number of nodes pointing to the best fitness, highest connectivity, and lowest average fitness.
According to the data:

- Algorithm 1 (algo_1) has 0 nodes pointing to the best fitness, 0.26 connectivity, and an average fitness of 95.411.
- Algorithm 2 (algo_2) has 1 node pointing to the best fitness, 0.44 connectivity, and an average fitness of 78.081.

Based on these values, we can see that algo_2 outperforms algo_1 in all three criteria:

- algo_2 has found a node with the best fitness, while algo_1 has not found any.
- algo_2 has higher connectivity (0.44 > 0.26).
- algo_2 has a lower average fitness value (78.081 < 95.411).

Therefore, in the context of STNWeb and according to the provided rules and data, the most effective algorithm for this problem is: [winner=algo_2]
Detailed insights:
algo_2 has found a node with the best fitness, which is a strong indicator of its effectiveness. algo_2 has higher connectivity, suggesting that it may be more robust in finding better solutions. algo_2 has a lower average fitness value, which further supports its superiority for this minimization problem.
[END TASK A]

2/Hard - Model Output - Mixtral-8x7b-instruct-v0.1

[BEGIN TASK A]
Based on the rules and data provided, we will compare the three algorithms in terms of their effectiveness for the maximization problem at hand.
[winner=algo_1, algo_2]; Both algo_1 and algo_2 have 8 nodes pointing to the best fitness, and they have the same average fitness value of 10.12. However, algo_1 has a higher connectivity (4.13) compared to algo_2 (4.10). In the context of STNWeb, having more connectivity indicates a more robust algorithm. Therefore, in this case, we have a tie between algo_1 and algo_2, as they perform equally well in terms of finding better results, but algo_1 is slightly more robust.
algo_3, on the other hand, has fewer nodes pointing to the best fitness (4), lower connectivity (2.5), and a significantly lower average fitness value (2.2). This indicates that algo_3 is less effective in solving the given maximization problem.
[END TASK A]

1.3 tulu-2-dpo-70b

1/Easy - Model Output

[BEGIN TASK A] Based on the provided data and rules, we can analyze the performance of both algorithms in the context of STNWeb.

- Rule 1 states that more nodes pointing to the best fitness nodes indicate better algorithm quality. In this case, algo_2 has 1 node pointing to the best fitness, while algo_1 has 0 nodes.
- Rule 2 suggests that a more connected algorithm is likely to be more robust. algo_2 has a higher connectivity value of 0.44, compared to algo_1's connectivity value of 0.26.
- Lastly, rule 3 indicates that, for a minimization problem, a lower average fitness value implies better performance. algo_2 has a lower average fitness value of 78.081, while algo_1 has a higher average fitness value of 95.411.

Taking into account all the rules, we can conclude that algo_2 is the [winner=algo_2] for this minimization problem. It has better connectivity, points to the best fitness node, and has a lower average fitness value across 10 trajectories.
[END TASK A]

2/Hard - Model Output

[BEGIN TASK A] Based on the given data and rules, the most effective algorithm for this maximization problem is [winner=algo_3].
Here are the detailed insights in the context of STNWeb:
Rule (1) suggests that the algorithm with more nodes pointing to the best fitness nodes has a higher quality. In this case, both algo_1 and algo_2 have 8 nodes pointing to the best fitness nodes, while algo_3 has 4 nodes pointing to the best fitness nodes. Therefore, algo_1 and algo_2 seem to have better quality according to this rule.
Rule (2) states that the algorithm with more connectivity is more likely to be robust, but only if it finds nodes of the best fitness. In this case, algo_1 has a connectivity of 4.13, algo_2 has a connectivity of 4.10, and algo_3 has a connectivity of 2.5. Therefore, algo_1 seems to have better connectivity according to this rule.
Rule (3) indicates that for a maximization problem, an algorithm is considered superior if it has a higher average fitness value across multiple trajectories. In this case, algo_1 and algo_2 have the same average fitness value of 10.12, while algo_3 has a higher average fitness value of 2.2. Therefore, algo_3 is the superior algorithm according to this rule.
Based on these insights, we can conclude that algo_3 is the most effective algorithm for this maximization problem, as it has a higher average fitness value and follows rule (3) for maximization problems. While algo_1 and algo_2 have better quality and connectivity according to rules (1) and (2), respectively, they do not perform as well as algo_3 in terms of average fitness value.
[END TASK A]

1.4 Bard-Gemini-Pro (january-24)

1/Easy - Model Output

Task A: Algorithm Comparison and Winner Identification
Based on the provided information and the defined rules:
Winner: [winner=algo_2]
Here's the detailed analysis of each algorithm and the reasoning behind choosing algo_2:
Rule 1:
algo_1: 0 nodes pointing to the best fitness.
algo_2: 1 node pointing to the best fitness.
Rule 2:
algo_1: 0.26 connectivity.
algo_2: 0.44 connectivity.
Rule 3:
This is a minimization problem, so a lower average fitness indicates a better algorithm.
algo_1: Average fitness of 95.411.
algo_2: Average fitness of 78.081.
Analysis:
While both algorithms lack nodes pointing to the absolute best fitness, algo_2 has one node closer to it compared to none for algo_1. Despite having lower connectivity overall, algo_2 still exhibits significantly higher connectivity towards better solutions (0.44 vs 0.26). Crucially, algo_2 achieves a much lower average fitness (78.081) compared to algo_1 (95.411), directly aligning with the minimization problem objective.
Insights from STNWeb:
Visualizing these results in STNWeb would likely reveal:
algo_2 having a path or paths leading to nodes closer to the best fitness compared to algo_1. A denser network structure for algo_2, particularly around high-performing nodes, indicating better exploration and exploitation. Lower color intensity in the fitness visualization for algo_2, highlighting its superior performance in achieving minimal values.
Therefore, considering all three rules and analyzing them in the context of STNWeb's visualization capabilities, algo_2 emerges as the more effective algorithm for this minimization problem.

2/Hard - Model Output

Task A: Identifying the most effective algorithm Based on the provided data and rules:
Winner: [tie between algo_1 and algo_2]
Detailed insights:
Both algo_1 and algo_2 achieve the same performance according to the given rules and data:

- They both have the same number of nodes pointing to the best fitness, indicating equal ability to find optimal solutions.
- Their connectivity is very similar, suggesting comparable robustness in exploring the search space.
- They share the same average fitness, implying identical performance in terms of solution quality.

Therefore, it's impossible to definitively declare one algorithm superior based on the provided information. Choosing between them might require further analysis considering:

- Distribution of fitness values: Do either algorithm produce more consistent results closer to the best fitness?
- Computational cost: Which algorithm requires less computational resources to achieve similar performance?
- Specific problem characteristics: Does the problem structure favor one algorithm's search strategy over the other?

Insights from STNWeb perspective:
Visualizing the algorithms' behavior in STNWeb could provide valuable insights for further comparison:

- Network structure: Analyze the paths taken by each algorithm towards the best solutions. Are there significant differences in exploration patterns?
- Convergence behavior: Observe how quickly each algorithm approaches the best fitness over time.
- Diversity of solutions: See if either algorithm explores broader regions of the search space, potentially leading to different high-quality solutions.

By using STNWeb's visualization capabilities, researchers can gain deeper understanding of the algorithms' inner workings and make more informed decisions when choosing the most suitable option for their specific needs.
I hope this detailed analysis helps you interpret the results and leverage STNWeb to further explore the algorithm behaviors!

2 Task B

1/Easy - Model Input

[BEGIN CONTEXT]
STNWeb is a new web tool for the visualization of the behavior of optimization algorithms such as metaheuristics. It allows for the graphical analysis of multiple runs of multiple algorithms on the same problem instance and, in this way, it facilitates the understanding of algorithm behavior. It may help, for example, in identifying the reasons for a rather low algorithm performance. This, in turn, can help the algorithm designer to change the algorithm in order to improve its performance. STNWeb is designed to be user-friendly. Moreover, it is offered for free to the research community.
[END CONTEXT]
[BEGIN TASK B]
[BEGIN DEFINITION]
Agglomerative clustering involves four key parameters:

1. (1) Cluster size (percentage): Maximal cluster size in terms of the percentage of all solutions a cluster contains.

2. (2) Volume size (percentage): Maximal cluster size in terms of the percentage of the covered search space volume spanned by the solutions a cluster contains.

3. (3) Distance measure: This is a function that measures the distance between solutions, influencing the creation of clusters. Possible values: Hamming, Euclidean, Manhattan.

4. (4) Cluster number: Number of clusters obtained for these solutions (from lowest to highest partitioning). The maximum number implies no partitioning is applied. Lower values result in increased partitioning. Good results are obtained when the cluster number is above the minimum value number of clusters but far from the highest maximum.

[END DEFINITION]
[BEGIN DATA]
These are the parameters of the agglomerative clustering algorithm:
- cluster size: 5
- volume size: 5
- distance measure: Euclidean
- cluster number: 400

These are the resulting limits:
- minimum possible number of clusters: 207
- maximum possible number of clusters: 574

[END DATA]
[END TASK B]
[BEGIN QUERIES]
Task B: Provide the revised numerical parameters for agglomerative clustering, aiming to improve the visualization outcomes within the context of the optimization problem and STNWeb. Additionally, explain the reasoning behind these modifications.
Instructions for your response:
- Use the label [BEGIN NEW CONFIG] and [END NEW CONFIG] to encapsulate the new numerical values.
- The new numerical values must be in the following format: [NAME_PARAMETER=NEW_VALUE].
- Present your response between [BEGIN TASK B] and [END TASK B].

[END QUERIES]

2/Hard - Model Input

[BEGIN CONTEXT]
STNWeb is a new web tool for the visualization of the behavior of optimization algorithms such as metaheuristics. It allows for the graphical analysis of multiple runs of multiple algorithms on the same problem instance and, in this way, it facilitates the understanding of algorithm behavior. It may help, for example, in identifying the reasons for a rather low algorithm performance. This, in turn, can help the algorithm designer to change the algorithm in order to improve its performance. STNWeb is designed to be user-friendly. Moreover, it is offered for free to the research community.
[END CONTEXT]
[BEGIN TASK B]
[BEGIN DEFINITION]
Agglomerative clustering involves four key parameters:

1. (1) Cluster size (percentage): Maximal cluster size in terms of the percentage of all solutions a cluster contains.

2. (2) Volume size (percentage): Maximal cluster size in terms of the percentage of the covered search space volume spanned by the solutions a cluster contains.

3. (3) Distance measure: This is a function that measures the distance between solutions, influencing the creation of clusters. Possible values: Hamming, Euclidean, Manhattan.

4. (4) Cluster number: Number of clusters obtained for these solutions (from lowest to highest partitioning). The maximum number implies no partitioning is applied. Lower values result in an increased degree of partitioning. Good results are obtained when the cluster number is above the minimum value for the number of cluster but far from the maximum.

[END DEFINITION]
[BEGIN DATA]
These are the parameters of the agglomerative clustering algorithm:
- cluster size: 60
- volume size: 60
- distance measure: Manhattan
- cluster number: 105
These are the configuration constants:
- minimum possible number of clusters: 101
- maximum possible number of clusters: 600
[END DATA]
[END TASK B]
[BEGIN QUERIES]
Task B: Provide the revised numerical parameters for agglomerative clustering, aiming to improve the visualization outcomes within the context of the optimization problem and STNWeb. Additionally, explain the reasoning behind these modifications. Instructions for your response:
- Use the label [BEGIN NEW CONFIG] and [END NEW CONFIG] to encapsulate the new numerical values.
- The new numerical values must be in the following format: [name_parameter=new_value].
- Present your response between [BEGIN TASK B] and [END TASK B].
[END QUERIES]

2.1 GPT-4-turbo

1/Easy - Model Output

[BEGIN TASK B]
[BEGIN NEW CONFIG]
[cluster_size=10%]
[volume_size=10%]
[distance_measure=Manhattan]
[cluster_number=300]
[END NEW CONFIG]
The reasoning behind these modifications is as follows:

- ’[cluster_size=10%]: Increasing the cluster size allows for a more macroscopic view of the solution space, possibly revealing broader patterns and trends in the behavior of the optimization algorithms. A larger cluster size could lead to a better understanding of how different regions of the solution space are explored.
- ’[volume_size=10%]: Similarly to cluster size, increasing the volume size helps in capturing larger parts of the search space. This can be particularly useful when dealing with high-dimensional problems where the search space is vast, and significant features can be missed with too fine-grained clustering.
- ’[distance_measure=Manhattan]: Switching to a Manhattan distance measure might better capture the differences between solutions in certain optimization problems, especially those where the search space is not isotropic, meaning that movements in different dimensions have different impacts on the solution quality. It also tends to be more sensitive to changes in individual dimensions compared to Euclidean distance.
- ’[cluster_number=300]: Lowering the number of clusters from 400 to 300 increases the partitioning of the solution space without going too close to the minimum possible number of clusters, which, according to the definition, can lead to more informative visualizations. It strikes a balance between the level of detail and the manageability of the clusters for analysis purposes.

These adjustments are proposed to provide a potentially clearer and more informative visualization within STNWeb, facilitating the identification of patterns and differences in algorithm performance across the search space.
[END TASK B]

2/Hard - Model Output

```
[BEGIN TASK B]
[BEGIN NEW CONFIG]
[cluster_size=30
[volume_size=30
[distance_measure=Euclidean]
[cluster_number=250]
[END NEW CONFIG]
```

The rationale behind the new configuration is as follows:

1. Reducing the cluster size to 30% will result in smaller clusters that may represent the optimization landscape with greater granularity. This can provide a clearer picture of how the solutions are distributed, which might be particularly useful when trying to understand nuanced behaviors in the algorithm's performance.
2. Similarly, reducing the volume size to 30% will limit the size of clusters in terms of the search space they cover. This can prevent overly large and potentially diverse clusters that could obscure detailed insights into the search space structure.
3. Changing the distance measure to Euclidean is based on the assumption that this measure might be more appropriate for visualizing continuous optimization problems on STNWeb. The Euclidean distance is a natural choice for representing distances in a multidimensional space, which can help in identifying clusters of solutions more effectively if the problem space is continuous.
4. Increasing the cluster number to 250 creates more clusters, enhancing the level of detail in the visualization. This is still far from the maximum of 600, avoiding excessive partitioning, yet it is significantly above the minimum of 101, which aims to avoid overly broad clustering. The chosen value represents a balance between detail and manageability and is expected to offer a degree of partitioning that can yield informative visualizations without becoming too fragmented.

[END TASK B]

2.2 Mixtral-8x7b-instruct-v0.1

1/Easy - Model Output

[BEGIN TASK B]
[BEGIN NEW CONFIG]
cluster size=3
volume size=3
cluster number=300
[END NEW CONFIG]
The modifications to the agglomerative clustering parameters aim to improve the visualization outcomes within the context of the optimization problem and STNWeb by providing a more comprehensive view of the solution space.

1. Decreasing the cluster size and volume size to 3% will create more clusters, allowing for a more detailed analysis of the behavior of optimization algorithms. This change might help in identifying the reasons for low algorithm performance more accurately, as it increases the granularity of the clusters.
2. Reducing the cluster number from 400 to 300 will help maintain a reasonable number of clusters for visualization on STNWeb while still ensuring a sufficient level of detail for understanding algorithm behavior. The rationale behind this adjustment is to avoid an excessively high number of clusters that could potentially overwhelm the user and complicate the interpretation of the results.

These modifications should collectively contribute to a better understanding of algorithm behavior, enabling researchers to make more informed decisions about algorithm design and optimization.
[END TASK B]

2/Hard - Model Output

[BEGIN TASK B]
[BEGIN NEW CONFIG]
cluster size = 50
volume size = 50
cluster number = 150
[END NEW CONFIG]
The reasoning behind these modifications is as follows:

1. Cluster size: Reducing the cluster size from 60% to 50% will increase the granularity of the clusters, potentially providing a more detailed view of the solution space. This can be beneficial for identifying finer structures or patterns in the data that may have been overlooked with larger cluster sizes.
2. Volume size: Similar to the cluster size, decreasing the volume size from 60% to 50% will increase the granularity of the clusters in terms of the search space volume. This change can help in identifying smaller, yet significant, regions of the search space that contribute to the optimization problem.
3. Cluster number: Increasing the cluster number from 105 to 150 is aimed at finding a ”sweet spot” for the number of clusters, where the partitioning is sufficient to reveal meaningful structures in the data, but not so high that it leads to over-partitioning and visual clutter. This new value is above the minimum possible number of clusters (101) but is still far from the maximum (600), which should provide a good balance for visualization purposes within the context of STNWeb.

These modifications aim to enhance the visualization outcomes by providing a more detailed and nuanced view of the optimization problem’s solution space, which can ultimately help in identifying patterns, structures, and potential issues that may affect the performance of the optimization algorithms being analyzed.

2.3 tulu-2-dpo-70b

1/Easy - Model Output

[BEGIN TASK B]
[BEGIN NEW CONFIG]
cluster_size_percentage = 10 # Increase cluster size to capture more diverse solutions
volume_size_percentage = 10 # Increase volume size to ensure better coverage of the search space
distance_measure = ”Manhattan” # Use Manhattan distance for better handling of discrete or integer solutions
cluster_number = 300 # Decrease the number of clusters for better visual separation of different regions
[END NEW CONFIG]
The revised parameters for agglomerative clustering aim to improve the visualization outcomes within the context of the optimization problem and STNWeb. By increasing the cluster size and volume size percentages, we capture a more diverse set of solutions and cover a larger portion of the search space. This improves the overall representation of the algorithms' behavior.
We also changed the distance measure to Manhattan, which is better suited for discrete or integer solutions commonly found in optimization problems. This distance metric will help create more meaningful clusters for the visual analysis.
Lastly, we decreased the number of clusters to 300, which is still a high enough value to provide good results but not as high as the original 400. This will result in better separation of different regions in the visualization, making it easier to identify and analyze the algorithm behavior.
With these modifications, the user can better understand and analyze the performance of multiple optimization algorithms on the same problem instance using STNWeb.
[END TASK B]

2/Hard - Model Output

[BEGIN TASK B]
[BEGIN NEW CONFIG]
cluster_size=50%, volume_size=55%, distance_measure=Euclidean,
cluster_number=120
[END NEW CONFIG]
The revised parameters for agglomerative clustering are as follows:
Cluster size (percentage): 50%
Volume size (percentage): 55%
Distance measure: Euclidean
Cluster number: 120
These modifications are made in order to improve the visualization outcomes within the context of the optimization problem and STNWeb.

- Reducing the cluster size from 60% to 50% will result in a higher number of smaller clusters, providing a more granular view of the algorithm behavior, which can be beneficial in identifying specific patterns or sub-optimal regions.
- Increasing the volume size from 60% to 55% will ensure that the clusters cover a larger portion of the search space, allowing for a more comprehensive view of the algorithm's exploration.
- Changing the distance measure from Manhattan to Euclidean can provide a more accurate representation of the distance between solutions, especially when dealing with continuous optimization problems.
- Increasing the cluster number from 105 to 120 will allow for more granular clustering, providing a more detailed view of the algorithm behavior and potentially revealing previously unseen insights. However, care must be taken not to set this value too high, as over-clustering can lead to a loss of valuable information.

[END TASK B]

2.4 Bard-Gemini-Pro

1/Easy - Model Output

Task B: Revised Numerical Parameters for Agglomerative Clustering

```
[BEGIN NEW CONFIG]
cluster size: 2%
volume size: 10%
distance measure: Hamming
cluster number: 250
[END NEW CONFIG]
```

Reasoning behind the modifications:

1. Cluster size: Reducing the cluster size to 2% allows for more granular groupings, potentially revealing finer details in the solution behavior. This can be especially helpful for identifying subtle differences between algorithms or parameter settings.
2. Volume size: Increasing the volume size to 10% ensures clusters capture a larger portion of the search space, providing a broader context for understanding solution distribution. This balance avoids overly restrictive clusters while maintaining interpretability.
3. Distance measure: Switching to Hamming distance can be more suitable for binary or discrete optimization problems, where exact value differences might be less important than solution structure. This can lead to clusters based on meaningful similarities in solution approaches.
4. Cluster number: Adjusting the cluster number to 250 falls closer to the middle ground between the suggested minimum and maximum. This provides a balance between achieving sufficient granularity without creating an overwhelming number of clusters that becomes difficult to visually analyze within STNWeb.

Additional Notes:

These are suggestions, and the optimal parameters might vary depending on the specific optimization problem and data characteristics. Experimenting with different configurations and visually evaluating the results in STNWeb is recommended. Consider using visualization techniques like color coding or size encoding within clusters to further enhance the interpretability of the results.

I hope this helps!

2/Hard - Model Output

Task B: Revised Numerical Parameters for Agglomerative Clustering
Unfortunately, I cannot recommend specific revised numerical parameters without more information about the optimization problem and the data being visualized. However, I can offer some general suggestions and considerations for tailoring the agglomerative clustering parameters to improve visualization outcomes in STNWeb:
[BEGIN NEW CONFIG]

1. **Cluster size and volume size:** Instead of setting them both to 60%, consider using a smaller value for cluster size (e.g., 30-40%) and keeping the volume size at 60%. This can create tighter clusters based on solution similarity while ensuring reasonable coverage of the search space.
2. **Distance measure:** If the features of your data are primarily categorical, Euclidean distance might not be the best choice. Consider experimenting with Hamming distance for better capture of dissimilarity between solutions.
3. **Cluster number:** The current setting (105) is close to the minimum (101). To explore a wider range of granularities, try values like 150, 200, and 300 while observing the visualization clarity and information density.

[END NEW CONFIG]
Reasoning behind the suggestions:

- Smaller cluster size can lead to more focused and interpretable clusters in the visualization, especially for larger datasets.
- Hamming distance can be more suitable for categorical data, leading to more meaningful groupings.
- Exploring a wider range of cluster numbers allows for finding the right balance between information richness and visual clutter.

Additional recommendations:

- Visualize the impact of parameter changes on the clustering results using STNWeb's interactive features.
- Consider incorporating domain knowledge about the optimization problem to guide parameter selection.
- If possible, compare the visualization outcomes with alternative clustering algorithms or distance measures.

Remember, the optimal parameters will depend on the specific characteristics of your data and the visualization goals. Experimentation and careful evaluation are key to achieving effective visualization using agglomerative clustering in STNWeb.