Maximum Dispersion Problem on Neuron Morphological Graph

Jizheng Dong,[1] Tianxiao He [1]

[1] *Computer Science Department, New York University Tandon School of Engineering, New York, 11201*
jd4587@nyu.edu, th3129@nyu.edu

Abstract: In this report, we investigate the neuron morphology simplification problem by selecting the keypoints that have maximum dispersion (distance) with each other. We take a combined approach from optimization and graph theory, and apply it on neuroscience graph dataset Flywire. In order to solve the maximum dispersion problem, we employed various methodologies, including quadratic optimization, two distinct greedy algorithms, and a random algorithm. Theoretical analysis and empirical studies highlight the greedy algorithm's effectiveness as a practical and near-optimal solution for the maximum dispersion problem, offering linear runtime and close approximation to ground truth. This work holds promise for advancing the efficiency and interpretability of large-scale neuron morphology analyses.

1. Introduction

In the field of neuroscience, the analysis of large-scale neuron morphology has gained significant traction. [1] [2] The huge dataset of neuron morphology is benefited from the advent of Electric Microscopy for layer-by-layer brain scanning and Deep Learning techniques, which facilitates the segmentation and combination of distinct components acquired from different layers. [3] [4] However, the sheer scale of neurons, such as the 130,000 neurons in Drosophila(fruit fly), poses a challenge to analysis when each neuron is represented by thousands of points. [1]

To address this challenge, novel methodologies are required to simplify neuron morphology while preserving its fundamental structure. In this study, we propose a method to identify key nodes within a neuron, aiming to reduce the node number and keep the structure. Through preliminary experiments, we determined that selecting nodes with maximum dispersion provides an efficient means to capture the essential features of neuron morphology. [5]

While previous methods for maximum dispersion primarily focus on optimization in operational research, neuron morphological graphs, with their simpler structure, invite exploration of more efficient, greedy approaches. This project aims to explore how effective is greedy algorithm in maximizing the dispersion of neuron morphological graph, compared to randomized baseline and well-studied optimization algorithm. To answer this question, we conduct a theoretical analysis of the greedy algorithm's approximation bounds and an empirical study comparing model performances on the FlyWire graph dataset.

2. Problem Setup

Maximum Dispersion Problem can be formulated as an optimization problem as follows. For an undirected graph $G = (V,E)$, we want to select a subset S of k vertices from V such that average distances between the selected vertices is maximized

$$V_s = \underset{v_i,v_j \in S}{\operatorname{argmax}} \binom{k}{2}^{-1} \sum_{i<j} d(v_i, v_j)$$

where $d(v_i, v_j)$ is the shortest path distance from v_i to v_j. To simplify the setup, we assume that this graph is also complete, then $d(v_i, v_j)$ becomes the weight of edge (v_i, v_j). Then for real world applications, we can convert any graph to complete graph by adding additional edges with weights equal to the shortest path distance.

Now the problem can be reformulated as a combinatorial optimization problem where x is an indicator vector such that $x_i = 1$ means the ith vertex is selected.

$$\begin{aligned} \underset{x}{\text{maximize}} \quad & x^T D x \\ \text{subject to} \quad & \sum_i x_i = k \\ & x_i \in \{0,1\} \end{aligned}$$

Notice that this is a non-convex quadratic programming problem, and it's NP-hard to obtain an exact solution if we allow negative edges [6]. Therefore in the following session, we aim to achieve a good approximation in polynomial time.

3. Optimization method

To avoid the non-linearity due to the product of two variables, we can linearize it by introducing another variable $y_{i,j} = x_i x_j$. Then the optimization problem can be rewritten as

$$\begin{aligned} \underset{x}{\text{maximize}} \quad & \sum_{i<j} d(v_i, v_j) y_{i,j} \\ \text{subject to} \quad & \sum_i x_i = k \\ & y_{i,j} \le x_i && \forall i < j \\ & y_{i,j} \le x_j && \forall i < j \\ & x_i \in \{0,1\} \\ & y_{i,j} \in \{0,1\} \end{aligned}$$

This formulation is equivalent to the original setup because from the definition of y, we know $y_{i,j} \le x_i$, $y_{i,j} \le x_j$ and the rest remains unchanged. To solve this problem, we first apply LP relaxation to allow x_i and y_{ij} to range between 0 and 1. Simplex's Method is utilized to determine an optimal solution within the relaxed constraints, which serves as an upper bound for our objective function. Then we can use branch and bound algorithm to obtain the integer solution.

According to Warren [7], we can use a tighter linearization method to solve this type of zero-one quadratic programming problem.

$$\begin{aligned} \underset{x}{\text{maximize}} \quad & \sum_{i<j} d_{i,j} y_{i,j} \\ \text{subject to} \quad & \sum_i x_i = k \\ & y_{i,j} \le x_i && \forall i < j \\ & y_{i,j} \le x_j && \forall i < j \\ & y_{i,j} \ge x_i + x_j - 1 && \forall i < j \\ & \sum_{i<j} y_{i,j} + \sum_{i>j} y_{i,j} = (k-1) x_j && \forall j \\ & x_i \in \{0,1\} \\ & y_{i,j} \in \{0,1\} \end{aligned}$$

This directly follows from previous formulation because this is equivalent to multiplying the constraint $\sum_i x_i = k$ by x_j and adding it as a new constraint.

$$kx_j = (\sum_i x_i) x_j = \sum_{i<j} x_i x_j + x_j^2 + \sum_{i>j} x_i x_j = \sum_{i<j} y_{ij} + \sum_{i<j} y_{ji} + x_j^2$$

$$\sum_{i<j} y_{ij} + \sum_{i<j} y_{ji} = kx_j - x_j^2 = (k-1) x_j$$

Again we can use a similar approach as before to solve the optimization problem to obtain our final result x, which indicates the set of vertices we would choose to maximize their pairwise distance. Previous researchers have shown that this is a good approximation in polynomial time. [8]

4. Greedy method

We present two greedy algorithms for the maximum dispersion problem. The first algorithm sequentially selects the maximum-weighted edge from the set of edges, aiming to maximize the dispersion of selected nodes. The second algorithm iteratively chooses nodes with the maximum sum of edges with other selected nodes, thereby enhancing the overall dispersion.

4.1. Greedy approximation method 1

In this algorithm, we denote V as the set of vertices, and the E is the set of edges. Then we iteratively select a pair of indices such that their edge has maximum weight and delete from E the edges connected to v_i or v_j .

Algorithm 1 Greedy Algorithm 1

```
function GREEDY-MAX-DISPERSION($D, k$)
    $S := \varnothing$
    $E' := E$
    for $t = 1$ to $\lfloor k/2 \rfloor$ do
        select $i, j$ such that $d(v_i, v_j) = \max\{d(v_i, v_j) | e_{ij} \in E'\}$
        set $S := S \cup \{v_i, v_j\}$
        set $E' := E' / \{e_{li}, e_{lj} | v_l \notin S\}$
    end for
    return S
end function
```

Previous research has shown a bound on the correctness of this greedy approximation, that this algorithm is a $\frac{1}{2}$ approximation for the dispersion of the optimal solution. [6] [9] However, obtaining a tighter bound more than $\frac{1}{2}$ is NP-hard. [10].

4.2. Greedy approximation method 2

In this algorithm, we iteratively select the vertex that have largest distance to previously selected vertices, and add it to our selected set S.

Algorithm 2 Greedy Algorithm 2

```
function GREEDY-MAX-DISPERSION($D, k$)
    select $i, j$ such that $d(v_i, v_j) = \max\{d(v_i, v_j) | v_i \notin S, v_j \notin S\}$
    set $S := \{v_i, v_j\}$
    for $t = 1$ to $k - 2$ do
        select $v_t := \arg\max_{v_t \notin S} \sum_{v_i \in S} \mathrm{dist}(v_i, v_t)$
        set $S := S \cup \{v_t\}$
    end for
    return S
end function
```

In the following, we aim to prove the ratio between optimal dispersion and dispersion obtained by this algorithm is bounded by $\frac{2}{k}$. Here we denote GA as the result of our greedy algorithm, which is the vertex set selected by the greedy algorithm. OPT is the optimal result.

Proof. Let us consider the case when $k = 2$. In this scenario, it is evident that $GA_2 = OPT_2 = \{a, b\}$ as the optimal and greedy solutions will naturally select the two points with the maximum dispersion.

Next, for $k = 3$, the greedy algorithm GA_3 extends GA_2 by adding a third point, say c, chosen for its maximum dispersion with respect to points a and b. Without loss of generality, assume that there exists another point c' in $V/\{a, b\}$ and $c' \in OPT_3$. According to the greedy algorithm, we have the inequality:

$$ac + bc \geq ac' + bc'$$

Consider the following cases:

- **Case 1:** If $a, b \in OPT_3$, then $c = c'$, and so $OPT_3 = GA_3$.

- **Case 2:** If $a, b \notin OPT_3$, then $OPT_3 = \{a', b', c'\}$.

$$\begin{aligned}
3ab &\geq a'b' + b'c' + a'c' && = W(OPT_3) \\
&\geq ab + bc + ac && = W(GA_3) \\
\text{Greedy Definition:} &\geq ab + bc' + ac' \\
\text{Triangle Inequality:} &\geq ab + ab
\end{aligned}$$

  Therefore, we can get:

$$W(GA_3) \geq \frac{2}{3} W(OPT_3)$$

- **Case 3:** If $a \in OPT_3$ and $b \notin OPT_3$, then $OPT_3 = \{a, b', c'\}$.

$$\begin{aligned}
3ab &\geq ab' + b'c' + ac' && = W(OPT_3) \\
&\geq ab + bc + ac && = W(GA_3) \\
\text{Greedy Definition:} &\geq ab + bc' + ac' \\
\text{Triangle Inequality:} &\geq ab + ab
\end{aligned}$$

  Therefore, we can get:

$$W(GA_3) \geq \frac{2}{3} W(OPT_3)$$

For $k > 3$:

Suppose that we have $W(GR_{k-1}) \geq c_{k-1} W(OPT_{k-1})$,
then

$$\begin{aligned}
W(GR_{k-1}) &\geq c_{k-1} W(OPT_{k-1}) \\
&\geq c_{k-1} \frac{\binom{k-1}{2}}{\binom{k}{2}} W(OPT_k) \\
&= c_{k-1} \frac{k-2}{k} W(OPT_k)
\end{aligned}$$

and

$$\begin{aligned}
W(GR_k) &= W(GR_{k-1}) + \sum_{j=1}^{k-1} v_k v_j \\
&\geq W(GR_{k-1}) + \sum_{j=1}^{k-1} v'_k v_j \\
&\geq W(GR_{k-1}) + \frac{1}{2(k-2)} \sum_{j=1}^{k-1} \sum_{l=1}^{k-1} v_j v_l \\
&= W(GR_{k-1}) + \frac{1}{(k-2)} W(GR_{k-1}) \\
&= \frac{k-1}{k-2} W(GR_{k-1}) \\
&\geq \frac{k-1}{k-2} c_{k-1} \frac{k-2}{k} W(OPT_k) \\
&= c_{k-1} \frac{k-1}{k} W(OPT_k)
\end{aligned}$$

So we can get:

$$c_k = \frac{k-1}{k} c_{k-1} = \frac{2}{k}$$

that is

$$W(GR_k) \geq \frac{2}{k} W(OPT_k)$$

□

Although the first greedy algorithm has a tighter bound than the second one, in practice, we found that the second always performs better. Its performance remains competitive even in comparison with the optimization method.

5. Experiment

We conduct experiments using graph dataset from Flywire [3], which is an advanced whole-brain neuronal wiring diagram, or connectome, of a female Drosophila (fruit fly) brain. This dataset consists of a detailed mapping from over 130,000 neurons and more than 30 million synapses, along with annotations of cell types, nerves and neurotransmitter identities, as shown in Figure 1a. In this large graph dataset, each neuron is stored as a tree and each vertex is sampled from reconstructed 3d point cloud of the neuron. This simplified tree like structure is called neuron morphological graph and shown in Figure 1b.

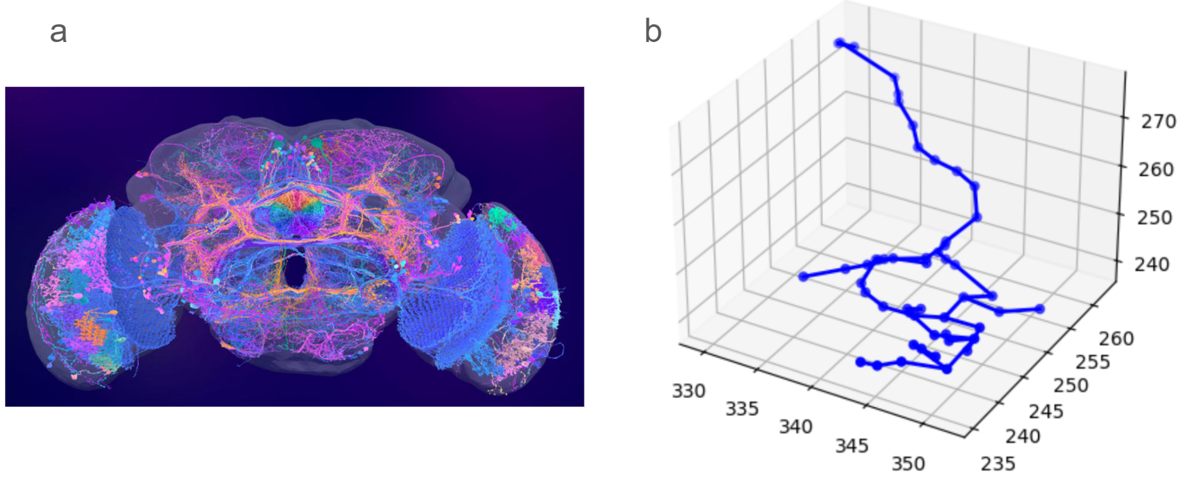

Fig. 1. Flywire dataset and graph representation of one neuron

In this project, our task is to simplify the morphological graph by selecting a subset of vertices such that their mean distance is maximized. We also noticed that this graph is a relatively simple structure, which prompted us to test our greedy algorithm that has less computational cost compared with optimization approach.

6. Results

In this section, we present the main finding of this project from experimenting with various max dispersion algorithms on neuron morphological graph. For each neuron (graph) in the sample dataset, we applied on them both greedy algorithm and optimization algorithm. Additionally, we established a ground truth for comparison by exhaustively exploring all potential selections and identifying the one with the highest dispersion. To evaluate the performance of our model, we calculated the mean dispersion of the indices our model selected and compare against the ground truth dispersion. The results are summarized in Fig 2.

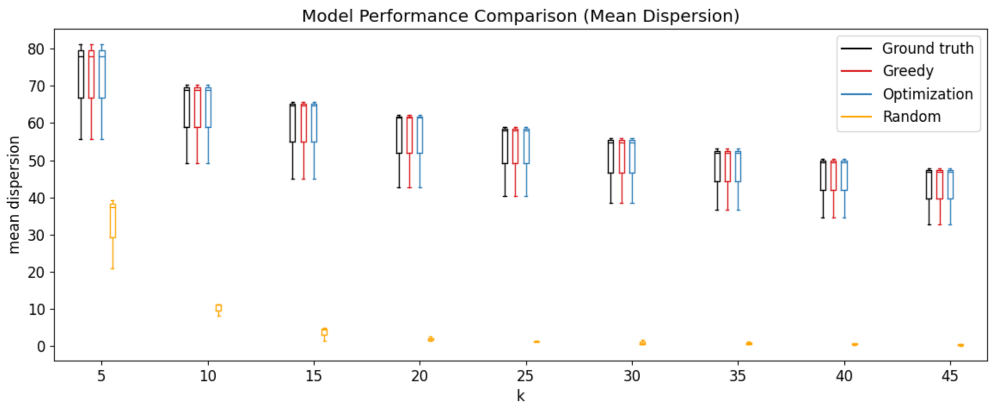

Fig. 2. Comparative analysis of mean dispersion values across various models at different k values

Our result leads to two observations. Firstly, both greedy algorithm and optimization algorithm displayed near optimal performance compared to ground truth, significantly higher than the dispersion score of random selection baseline. Secondly, we noticed a slight decrease in mean dispersion value as we increase k, which makes sense intuitively since selecting more vertices in a finite graph would cause them to be more cluttered, therefore decreasing average distance between them. Similar effect can also be observed in Fig 3, where we compare the total dispersion value without normalization. The greedy algorithm and optimization algorithm both displayed near optimal performance and were significantly higher than the random baseline.

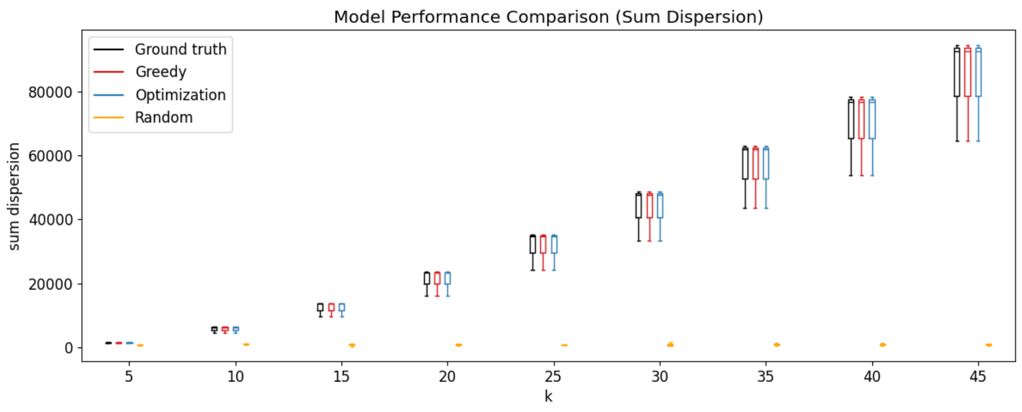

Fig. 3. Comparative analysis of total dispersion values across various models at different k values

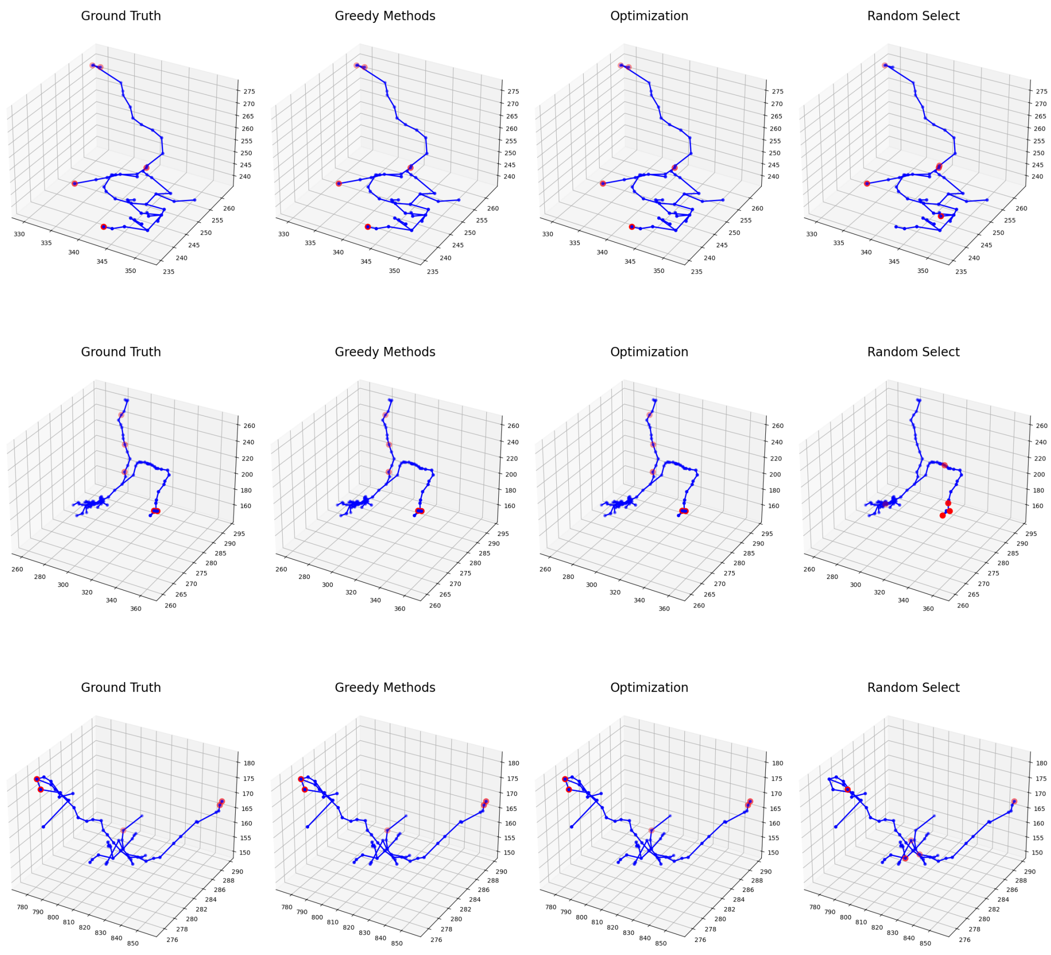

Fig. 4. Visualization of selected indices on morphological graph across various models

To validate the practicality of our dispersion-based selection approach, we visualized the selected vertices for different neurons by different models in Fig 4. These visualization confirms our intuition that this method effectively identify the vertices on the boundary. Furthermore, it also supports our previous observation that greedy solution achieves near optimal selection of points with faster computational time than optimization based approach.

7. Conclusion

In this project, we have studied the max dispersion problem in neuron morphological graph. Due to the special structure of this graph, we aim to explore a more simple and computationally effective greedy approach and test its effectiveness using both theoretical and empirical approach. Empirically, we traverse through all possible combinations of k nodes within the graph and use it as our ground truth. We implemented one optimization approach with linear relaxation and two greedy approaches where we select optimal edges or vertices at each iteration. Comparison of accuracy and time complexity of these methods reveal that the vertex-based greedy algorithm has superior performance. Theoretical analysis also shown that greedy approach is a good approximation to optimal solution, although the bound could be improved for future studies. In conclusion, the efficiency and accuracy of greedy approach make it a standout choice in solving the maximum dispersion problem on neuron morphological graphs.

References

1. A. Lin, R. Yang, S. Dorkenwald, A. Matsliah, A. R. Sterling, P. Schlegel, S.-c. Yu, C. E. McKellar, M. Costa, K. Eichler, *et al.*, "Network statistics of the whole-brain connectome of drosophila," *bioRxiv*, 2023.
2. K. M. Stiefel and T. J. Sejnowski, "Mapping function onto neuronal morphology," *Journal of neurophysiology*, vol. 98, no. 1, pp. 513–526, 2007.
3. S. Dorkenwald, A. Matsliah, A. R. Sterling, P. Schlegel, S.-C. Yu, C. E. McKellar, A. Lin, M. Costa, K. Eichler, Y. Yin, *et al.*, "Neuronal wiring diagram of an adult brain," *bioRxiv*, 2023.
4. L. K. Scheffer, C. S. Xu, M. Januszewski, Z. Lu, S.-y. Takemura, K. J. Hayworth, G. B. Huang, K. Shinomiya, J. Maitlin-Shepard, S. Berg, *et al.*, "A connectome and analysis of the adult drosophila central brain," *Elife*, vol. 9, p. e57443, 2020.
5. R. Martí, A. Martínez-Gavara, S. Pérez-Peló, and J. Sánchez-Oro, "A review on discrete diversity and dispersion maximization from an or perspective," *European Journal of Operational Research*, vol. 299, no. 3, pp. 795–813, 2022.
6. O. A. Prokopyev, N. Kong, and D. L. Martinez-Torres, "The equitable dispersion problem," *European journal of operational research*, vol. 197, no. 1, pp. 59–67, 2009.
7. W. P. Adams and H. D. Sherali, "A tight linearization and an algorithm for zero-one quadratic programming problems," *Management Science*, vol. 32, no. 10, pp. 1274–1290, 1986.
8. J.-C. Picard and M. Queyranne, "A network flow solution to some nonlinear 0-1 programming problems, with applications to graph theory," *Networks*, vol. 12, no. 2, pp. 141–159, 1982.
9. R. Hassin, S. Rubinstein, and A. Tamir, "Approximation algorithms for maximum dispersion," *Operations research letters*, vol. 21, no. 3, pp. 133–137, 1997.
10. S. S. Ravi, D. J. Rosenkrantz, and G. K. Tayi, "Heuristic and special case algorithms for dispersion problems," *Operations research*, vol. 42, no. 2, pp. 299–310, 1994.