Galaxies and Their Core: Identifying the Galactic Center Using Graph Theory

Guntara Hambali - 13523114[1]
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
[1]guntarahambali3@gmail.com, 13523114@std.stei.itb.ac.id

***Abstract*—Galaxy is an astronomical object that have significant role to human understanding toward our universe. Understanding what is in its core is one of the major work of today astronomers. Because of that, finding the galactic center would be a great opening to such an inquiry. The graph theory offers one of the method to find the center of an object that contains discrete objects (as a galaxy contains stars). The method is called degree centrality. In this paper, we will implement degree centrality to find the center of the galaxy.**

***Keywords*— Degree Centrality, Galactic Core, Graph Theory, Star.**

I. INTRODUCTION

Galaxy is one of the astronomical objects that has a significant role in astronomy observation. It plays role as one of the great milestone in human understanding toward universe. Understanding the structure of the galaxy is one of the major works for astronomers in recent decades. One of the keys to understanding galaxy is to understand its center. The galactic center contains exotic objects such as blackhole, dust, gas, massive stars, etc. Therefore, finding the center of the galaxy would help us a lot in our work to understand our universe.

In discrete mathematics, graph theory offers an alternative solution for this. A simple but powerful method called degree centrality. By representing stars as node and gravitational force as the edge, we could find the node which has most connection to other nodes. By this, we will find a galactic center not far from this star.

The challenge of this implementation relies on the dataset that will be used. One of the biggest datasets of stars in milky way galaxy (our galaxy) is Gaia DR3. Unfortunately, it "only" contains around 1 billion data. Indeed, that dataset is very large, but it still is not enough to represent all of the stars in a galaxy. Therefore, in this paper, we will simulate the data using the laws that govern the structure of the galaxy.

This paper is organized as follows: Section 2 will be the explanation of theories that will be used for implementation. This section also explains the assumptions made when dealing with galaxy and its contents. Then, section 3 will discuss the implementation of degree centrality in determining the center of a galaxy. This section contains three parts. Each explains how the data is simulated, implementation of degree centrality, and evaluation of the implementation using right metric. In section 4, we will see the conclusion of this paper based on evaluation.

II. THEORETICAL BASIS

A. Star and Stellar Mass

In everyday life, star is often defined as an astronomical object that could shine from itself. Although it is not wrong, it is generally accepted among astronomers to define star as an astronomical object that reaches equilibrium between its outward force (nuclear force in form of radiation and thermal pressure) and inward force (gravity). This is called hydrostatic equilibrium. The nuclear force is coming formed when the star starts fusion chain reaction that merges smaller atoms into bigger one. The chain reaction from one atom to another atom define the characteristic of star and will determine the death of the star later.

Mass of a star can also be one of the predictors for its age. It is based on the fact that high-mass stars conduct nuclear fusion in its core faster than low-mass stars do. Therefore, its mass decreases faster than low-mass star. Hence, high-mass stars generally are shorter age than low-mass stars.

Most stars in galaxies have unique distribution of mass. Based on the law that first introduced by astronomer Edwin Salpeter—Salpeter Initial Mass Function (IMF) Law [1], low-mass stars (below solar mass) dominate galaxy population whereas high-mass stars (above 10 times solar mass) are rarely find in a galaxy. Mathematically:

$$\xi(m) \propto m^{\alpha}$$

where:
$\xi(m)$ = the number of stars per unit mass (e.g., stars per $M_{\odot}$)
m = stellar mass in solar masses ($M_{\odot}$).
α = the Salpeter power-law slope for stars in the mass range $0.5M_{\odot} \lesssim m \lesssim 50M_{\odot}$
because $\alpha = -2.5 < 0$ it means that the number of stars per unit mass is decreasing as the mass of stars is increased. This law aligns with the fact that the high-mass star has a shorter age compared to low-mass star. The high-mass star dead first, so its population does not significant inside the galaxy.

B. Galaxy and Its Structures

Galaxy is a massive astronomical object that does not stand alone. It contains various other astronomical objects such as gas, dust, dark matter, blackhole, and mostly stars. Based on their morphology, galaxies can be classified into 4 groups following Hubble classification. They are spiral, elliptical, irregular, and lenticular galaxy. Our galaxy, the Milky Way galaxy is a spiral galaxy which contains bulge part and disk part.

The bulge part of the galaxy is located at the center, and it forms sphere formation. The size of bulge has a characteristic or profile that can be described by Plummer density profile [2]. This law can be used to model distribution of stars in spherical systems such as bulge (as mentioned before has spherical form) or globular cluster. Mathematically, it can be expressed as:

$$F(r) = \frac{r^3}{(r^2 + a^2)^{\frac{3}{2}}}$$

Where:
F = distribution function
r = distance of the stars
a = scale length

to get the radius, we can utilize this equation and make it in this form:

$$r = a\left(u^{-\frac{2}{3}} - 1\right)^{-\frac{1}{2}}$$

where u is random variable that has uniform distribution.

The disk part of the galaxy is located after the center of the galaxy from the galactic center. It forms disk formation (thin and wide). The same as bulge, the disk formation has also its distribution that can be described by exponential disk density profile [3]. This law says that the stars distribution in disk decrease exponentially as the distance from galactic center grows. Mathematically, it can be described as:

$$\Sigma(r) = \Sigma_0 e^{-\frac{r}{R_d}}$$

Where:
Σ = Distribution function
Σ_0 = Distribution of stars at center
r = distance to center
R_d = scale length

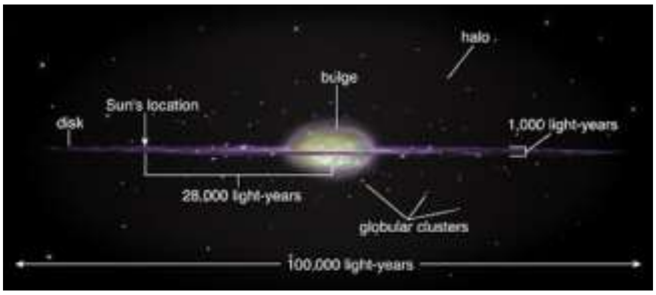

Fig 1. Galaxy and Its Structure (side view)
Source: NMSU Astronomy

C. Newton's Law on Gravity

Astronomical objects tend to be grouped thanks to Newton's Law of gravity. Gravity is a force that pull two masses towards each other. This force governs the entire astronomical objects in this universe from they were born until they death. It governs the move of the stars and defines the large structure of galaxy. Mathematically, it can be expressed as follows:

$$F = G\frac{m1 \times m2}{r^2}$$

Where:
F = force (N)
r = distance of two objects (m)
m1, m2 = mass of object 1 and 2

physically, this law says that the heavier the mass of the object, the more they tend to pull each other. Inversely, the further distance between these two objects, the smaller the force it will exert.

D. Graph

In discrete mathematics, a graph is defined as mathematical object that can be used to represent discrete objects and their relations. Graph G could be defined as G = (V, E) where V is a set of vertices $\{v_1, v_2, \dots, v_n\}$ or nodes which represent discrete objects. E is a set of edges $\{e_1, e_2, \dots, e_n\}$ that connect two vertices. [4]

One of the key usages of graph is the concept of degree centrality. Degree is the measure on how a node is connected to other nodes. Formally, the degree of a node is the number of edges connected to the node. Degree can be utilized to measure the centrality of a node. The higher the degree of a node, the more central it is. This degree centrality could be applied to many fields such as social networks (to find the most influential account), epidemiology (to describe people with most interaction), etc. In this paper, degree centrality will be used to find the center of a galaxy.

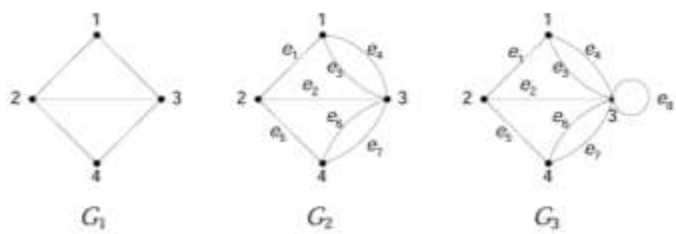

Fig 2. Three kinds of graphs (simple graph, double graph, pseudo graph)
source: https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2024-2025/20-Graf-Bagian1-2024.pdf

III. IMPLEMENTATION AND RESULT ANALYSIS

A. Data Simulation

Before doing graph analysis in searching for the center of the galaxy, we must first have data of stars in a galaxy. Unfortunately, there is no single observation that could go all through the stars in a galaxy because it would require extra work and time. One of the biggest data of stars in milky way galaxy (our galaxy) only contains around 1 billion data. Indeed, that dataset is very large, but it still is not enough to represent all of the stars in a galaxy.

So, there are 2 difficulties here, the dataset is very large (problem 1), and the dataset is not representative enough (problem 2). Fortunately, it is observed that galaxy has such structure that follows some law. These laws are representative enough to solve problem number 2 and can be scaled down to solve problem number 1.

Therefore, in this inquiry, we will simulate the data using the laws that have been described on theoretical basis section. We will implement that as follows:

First, we implement Plummer Radius function to describe radius for bulge part of galaxy

```
def plummer_radius(N, scale_length):
  u = np.random.uniform(0, 1, N)
  return scale_length * (u**(-2/3) -
1)**(-0.5)
```

then, implement the function to generate star position in the bulge part using plummer_radius function made earlier.

```
def spherical_bulge(num_stars,
scale_length=1.0):
  r = plummer_radius(num_stars,
scale_length
  theta = np.random.uniform(0, 2 * np.pi,
num_stars)
  phi = np.arccos(2 * np.random.uniform(0,
1, num_stars) - 1)

  x = r * np.sin(phi) * np.cos(theta)
  y = r * np.sin(phi) * np.sin(theta)
  z = r * np.cos(phi)

  return x, y, z
```

After that, generate star position in the disk part of the galaxy that follows exponential function distribution.

```
def exponential_disk(num_stars,
scale_length=3.5, scale_height=0.1):
  r =
np.random.exponential(scale=scale_length,
size=num_stars)
  theta = np.random.uniform(0, 2 * np.pi,
num_stars)
  z = np.random.normal(0, scale_height,
num_stars)

  x = r * np.cos(theta)
  y = r * np.sin(theta)
  return x, y, z
```

After all-star position have been made, we then generate star mass based on its position using Salpeter IMF law.

```
def stellar_masses(num_stars,
min_mass=0.1, max_mass=50):
    alpha = -2.35
    masses = ((np.random.uniform(0, 1,
num_stars) * (max_mass**(alpha+1) -
min_mass**(alpha+1))) +
min_mass**(alpha+1))**(1/(alpha+1))
    return masses
```

Then, we integrate all stars from the bulge and disk part of the galaxy. We also combine the mass of the stars into one dataframe. The resulting dataframe will contain the position of the stars (x,y,z) with the reference point (0,0,0) at the center/core of the galaxy.

```
def simulate_galaxy(num_stars=10000,
bulge_fraction=0.2):
  bulge_stars = int(num_stars *
bulge_fraction)
      disk_stars = num_stars - bulge_stars

  x_bulge, y_bulge, z_bulge =
spherical_bulge(bulge_stars)
  bulge_masses =
stellar_masses(bulge_stars)

  x_disk, y_disk, z_disk =
exponential_disk(disk_stars)
  disk_masses = stellar_masses(disk_stars)

  x = np.concatenate([x_bulge, x_disk])
  y = np.concatenate([y_bulge, y_disk])
  z = np.concatenate([z_bulge, z_disk])
  masses = np.concatenate([bulge_masses,
disk_masses])

  df = pd.DataFrame({'x': x, 'y': y, 'z':
z, 'mass': masses})

  return df
```

We then call this function to make 10000 stars in a galaxy. This means we will analyze 10000 nodes in a graph. This is representative enough to simulate a galaxy while maintaining less cost in the implementation.

```
simulated_df =
simulate_galaxy(num_stars=10000)
```

We then separate the positions array and masses array because the positions will be the nodes, and the masses will be used for computing the edge. The unit of positions will be kiloparsec (1 kpc = $3.0857 \times 10^{19} m$) and for the mass will be solar mass ($1M_{\odot}$ = $1.989 \times 10^{30}\ kg$).

```
positions = simulated_df[['x', 'y',
'z']].to_numpy()

masses = simulated_df['mass'].to_numpy()
```

The simulated data can be visualized as:

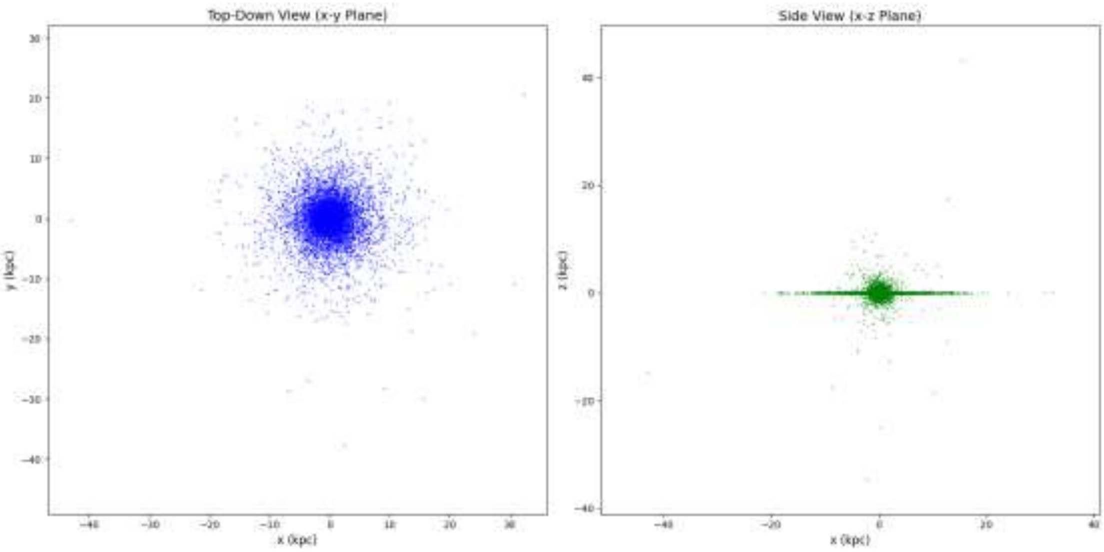

Fig 2. Simulated Galaxy

B. Graph Analysis Implementation

After the data simulation is done, we then ready to jump to the analysis. A has been told before, the graph will contain stars as its nodes. For the edge, we will calculate the force for each node to another node using Newton's Law of Gravity. So, it will be a weighted graph.

For the weight, we only compute the force for two nodes that have distance below threshold. Beyond that, we will consider it as insignificant and therefore there is no edge connection between them. Besides distance threshold, we will also apply force threshold. The force below that threshold will not be considered as an edge. Now, we will begin constructing the graph using networx library.

First, we initiate nodes for the graph. The nodes (as star) have 2 properties, positions and masses.

```
import networkx as nx

G = nx.Graph()

for i, (x, y, z, mass) in
enumerate(zip(simulated_df['x'],
simulated_df['y'], simulated_df['z'],
simulated_df['mass'])):

      G.add_node(i, position=(x, y, z),
mass=mass)
```

Then, we implement a helper function for calculating gravitational force between two stars using Newton's Law of Gravity. Because the unit of positions is in kiloparsec and masses in solar mass, the constant G will slightly be different to make the gravitational force in Newton unit.

```
G_CONST = 4.3 * 1e-6

def gravitational_force(m1, m2, r):
  if r == 0:
     return 0
  return G_CONST * m1 * m2 / r**2
```

After that, we are ready to connect the nodes with edges. Here, we use distance_threshold = 1 kpc. The force threshold is set when the distance is 1 kpc and masses of interacting stars are 1 $M_{\odot}$. Therefore, force_threshold = G_CONSTANT.

```
def compute_edges(i, positions, tree,
masses, distance_threshold, G_CONST):
  edges = []
  neighbors =
tree.query_ball_point(positions[i],
distance_threshold)
  for neighbor in neighbors:
    if neighbor > i:
      distance =
np.linalg.norm(positions[i] -
positions[neighbor])

      force =
gravitational_force(masses[i],
masses[neighbor], distance)

      if force > G_CONST:
```

```
        edges.append((i,neighbor,{'weight'
: force}))
  return edges
```

Furthermore, we then execute the computation of the edge with parallelization and put the results in a list. The edge then be applied to the graph.

```
tree = KDTree(positions)
distance_threshold = 1

edges_list = Parallel(n_jobs=-1)(
      delayed(compute_edges)(i, positions,
tree, masses, distance_threshold, G_CONST)
for i in range(len(positions))
  )

edges = [edge for sublist in edges_list
for edge in sublist]

G.add_edges_from(edges)
```

Finally, we have the whole graph done. Now, to find the galactic center, we will search for the node that has most degree utilizing network library.

```
degrees = dict(G.degree())
galactic_center = max(degrees,
key=degrees.get)
```

```
print("Galactic Center Node:",
galactic_center)
print("Degree of Galactic Center:",
degrees[galactic_center])
```

This yields a result:

```
Galactic Center Node: 6448
Degree of Galactic Center: 2594
```

Then, to get the galactic center position

```
center_position =
G.nodes[galactic_center]['position']

print("Estimated Galactic Center Position
(x, y, z):", center_position)
```

this yields result:

```
Estimated Galactic Center Position (x, y,
z): (0.04211628256088834, -
0.21063561957060217, 0.07010646255481412)
```

C. Evaluation

From the above result, we will evaluate it to know if our method in finding galactic center yields expected result or not. As the ground truth, we will use reference point at (0,0,0) as the center of the galaxy because when we simulate the data, we design it to be there. We will then evaluate the Euclidean distance between the galactic center we get from the implementation and the reference point and call it as absolute error. After that, we will compare the absolute error with the farthest distance of the star in a galaxy from the center. Based on exponential disk density profile, the density of stars decreases exponentially. Beyond 10 kpc, the stars density is very low. Therefore, it can be assumed that the farthest star relies at 10 kpc from the galactic center. Below is the implementation

```
reference_position = (0,0,0)
distances = np.linalg.norm(positions,
axis=1)

farthest_distance = 10

error_absolute =
np.linalg.norm(np.array(center_position) -
np.array(reference_position))

error_normalized = error_absolute /
farthest_distance
```

```
print("Absolute Error (distance to
reference position):", error_absolute)
print(f"Normalized Error (relative to
galaxy scale): {error_normalized*100}%")
```

The above codes yield results as follows:

```
Absolute Error (distance to reference
position): 0.22595588414686082
Normalized Error (relative to galaxy
scale): 2.2595588414686083%
```

IV. CONCLUSION

From the implementation, we get galactic center at position (0.042, -0.211, 0.07). Based on the evaluation, we get error/deviation only 2.26% from the actual galactic center. Therefore, it can be concluded that this inquiry is successful in finding the galactic center using graph analysis. Further development will be recommended in the next section

V. RECOMMENDATION

Based on simulated data, graph analysis could successfully determine the galactic center with minor deviation from the actual value. It will be a development in further research if we use real data from space agencies. That research would be a verification of this paper whether it is still true or not. After all, using real data will require extra cost, especially in connecting node with other nodes (determining the edge). Further research on modeling the edge in another way would also be a recommendation.

VI. CLOSING STATEMENT

The author would like to thank the entire academic community who have organized the Discrete Mathematics lectures for the 2024/2025 Informatics Engineering class of the Bandung Institute of Technology. Especially to the lecturer, Mr. Rila Mandala and also all the assistants of the Innovation and Computational Engineering (IRK) laboratory.

The author realizes that there are still many shortcomings in this study. Therefore, this study is very open to all kinds of constructive criticism and suggestions. The author is also open to any intention to collaborate in developing this research.

REFERENCES

[1] G. Chabrier, "The Initial Mass Function: From Salpeter 1955 to 2005," arXiv preprint astro-ph/0409465, Aug. 2004. [Online]. Available: https://arxiv.org/abs/astro-ph/0409465. [Accessed on: Jan. 8, 2025, 21:37].

[2] H. C. Plummer, "On the problem of distribution in globular star clusters," Monthly Notices of the Royal Astronomical Society, vol. 71, pp. 460-470, 1911.

[3] "Flattened Mass Distributions," Galaxies: Dynamics, Evolution, and Formation. [Online]. Available: https://galaxiesbook.org/chapters/II-01.-Flattened-Mass-Distributions.html. [Accessed on: Jan. 8, 2025, 21:45].

[4] Munir, Rinaldi. (2024), “Graf (Bagian 1)”. [Accessed on: Jan. 8, 2025, 19:52].

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 8 Januari 2024

Guntara Hambali (13523114)