1a. Best-First Search

In [1]:

```
from queue import PriorityQueue


def best_first_search(graph,start,goal,heuristic):
    visited = set()
    pq = PriorityQueue()
    pq.put((heuristic[start],start))

    while not pq.empty():
        h,node = pq.get()

        if node == goal:
            print("Goal Reached :",node)
            return

        if node not in visited:
            for neighbor in graph[node]:
                if neighbor not in visited:
                    pq.put((heuristic[neighbor],neighbor))
            print("Visiting Node : ",node)
            visited.add(node)
    print("Goal Not Found!!")
```

In [2]:

```
graph = {
    'S':['A','B'],
    'A': ['C', 'D'],
    'B': ['E', 'F'],
    'C': [],
    'D': [],
    'E': ['H'],
    'F': ['I', 'G'],
    'H':[],
    'I':[],
    'G':[],
}

start_node = 'S'
goal_node = 'G'

#Heuristic values from curr node -> goal node
heuristic_values = {
    'S': 13,
    'A': 12,
    'B': 4,
    'C': 7,
    'D': 3,
    'E': 8,
    'F': 2,
    'H': 4,
    'I': 9,
    'G': 0,
}

best_first_search(graph, start_node, goal_node, heuristic_values)
```

```
Visiting Node :  S
Visiting Node :  B
Visiting Node :  F
Goal Reached : G
```

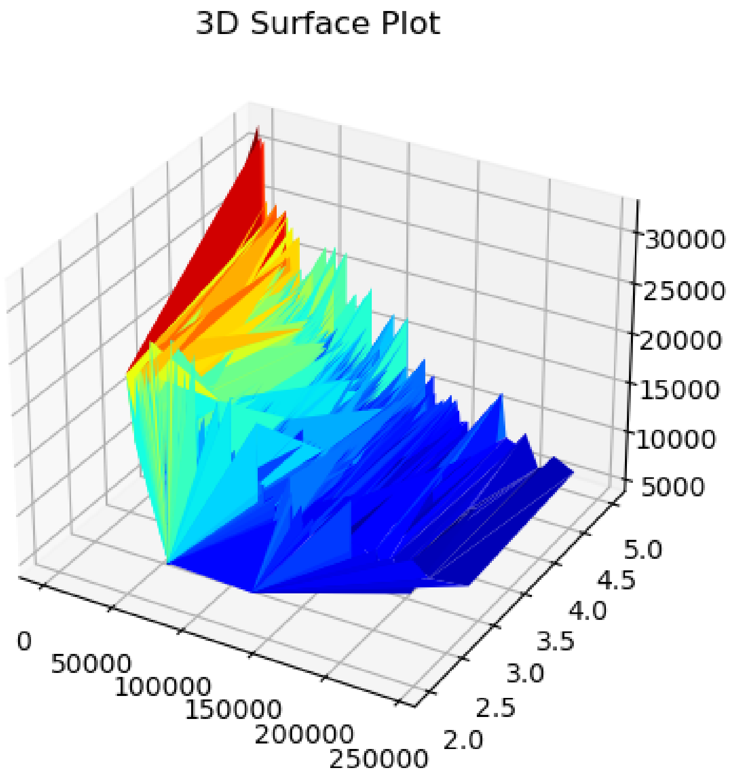
1b. 3D-Plot

In [3]:
```
import pandas as pd
import matplotlib.pyplot as plt
```

In [4]:
```
dataset = pd.read_csv('./corolla.csv')
x = dataset['KM']
y = dataset['Doors']
z = dataset['Price']

ax = plt.axes(projection='3d')
ax.plot_trisurf(x,y,z,cmap="jet")
ax.set_title("3D Surface Plot")

plt.show()
```



3D Surface Plot