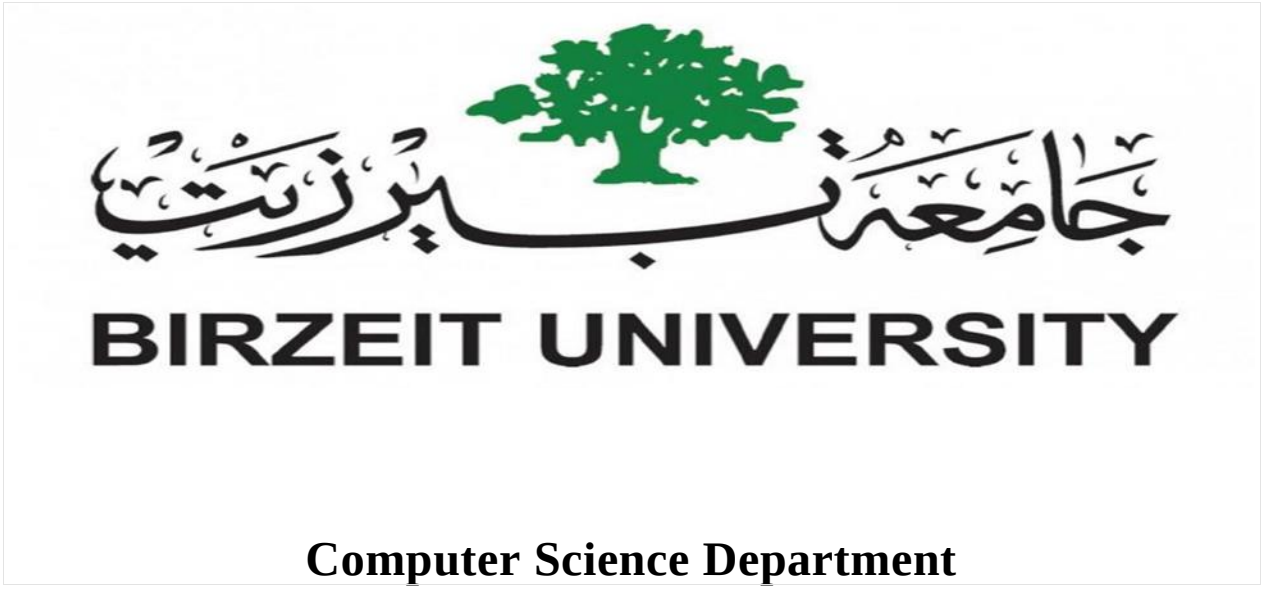


Computer Science Department

COMP338 Artificial Intelligence

Assignment I

Problem: Round Table Seating Arrangement

Report: Evaluation of Seating Arrangement Algorithms

Prepared By: Sanaa Obied-1211079

Instructor: Dr. Mohammed Helal

Section: 1

Date: 21/4/2024

Report: Evaluation of Seating Arrangement Algorithms:

1. Implementation Overview:

The implementation involved the development of seating arrangement algorithms using Uniform Cost Search (UCS), Greedy Search, and A* Search. These algorithms were designed to minimize conflict in seating arrangements based on a provided dislike matrix representing heuristic evaluation function $h(n)$. The implementation included methods for calculating non-linear dislike costs, total costs of arrangements, and heuristic values for UCS and A* Search. Additionally, the algorithms were evaluated for their performance in finding optimal seating arrangements.

2. Problem Overview:

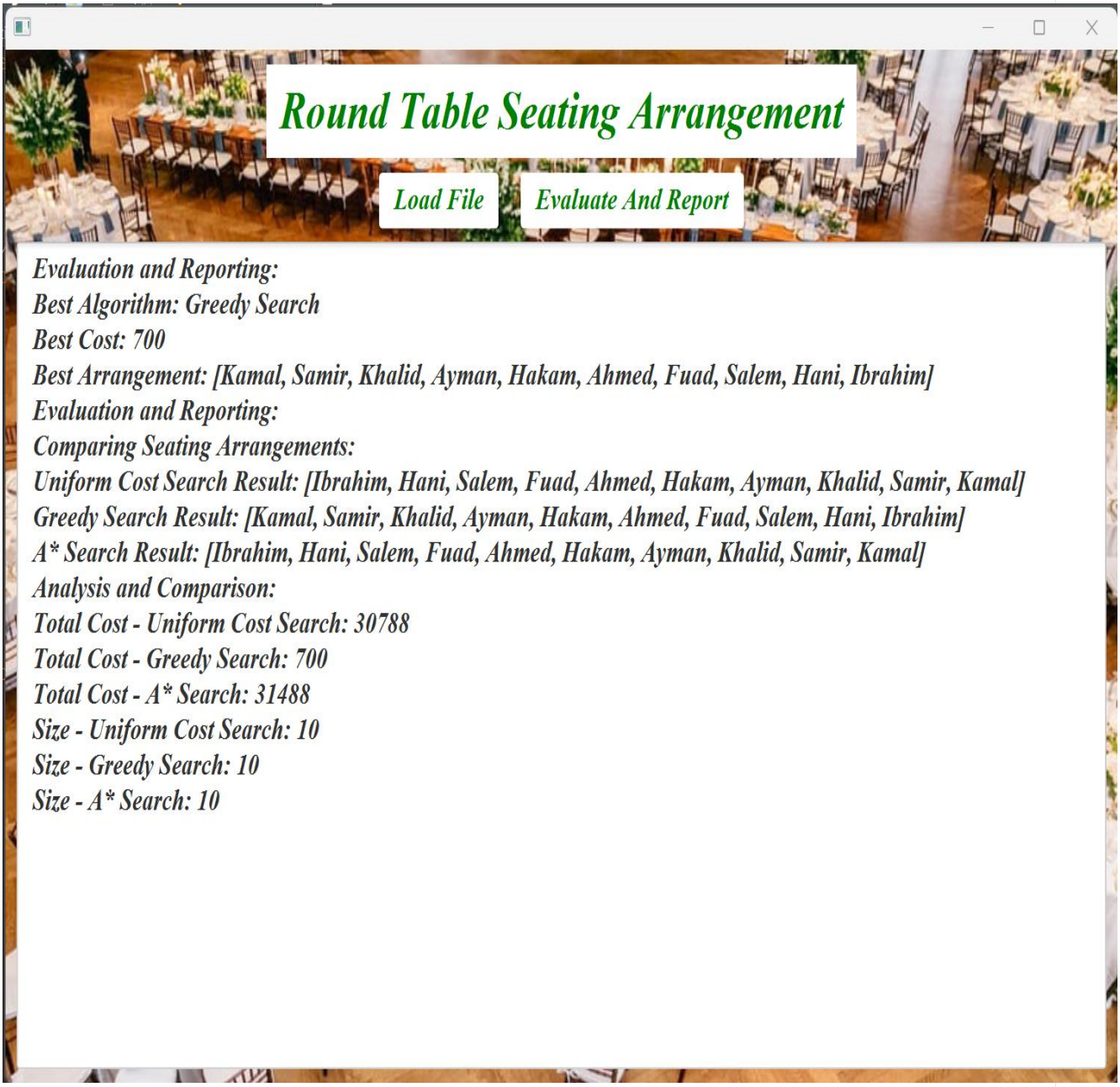
The task involved seating individuals around a round table to minimize conflict based on a dislike matrix and a non-linear dislike cost function. The goal was to implement Uniform Cost Search (UCS), Greedy Search, and A* Search algorithms to find the best seating arrangement.

3. Implementation Details:

- **Data Representation**: The dislike matrix was represented using Java data structures. The non-linear dislike cost function was implemented to calculate costs based on dislike percentages.
- **Algorithm Implementation**: UCS, Greedy Search, and A* Search algorithms were implemented to find seating arrangements. The evaluation process was modified to incorporate the non-linear dislike cost function.
- **User Interface**: A JavaFX interface was created to load the dislike matrix from a file, perform the evaluations, and display the results in a text area.

4. Results:

The evaluation results are as follows:



5. **Analysis and Comparison:**

The performance of each algorithm was analyzed and compared based on the obtained results:

- Uniform Cost Search Result: [Ibrahim, Hani, Salem, Fuad, Ahmed, Hakam, Ayman, Khalid, Samir, Kamal]
- Greedy Search Result: [Kamal, Samir, Khalid, Ayman, Hakam, Ahmed, Fuad, Salem, Hani, Ibrahim]
- A* Search Result: [Ibrahim, Hani, Salem, Fuad, Ahmed, Hakam, Ayman, Khalid, Samir, Kamal]

Strengths and Weaknesses:

- Uniform Cost Search:
  - Strengths: Finds optimal solutions with lower costs.
  - Weaknesses: May have higher time complexity for large datasets.
- Greedy Search:
  - Strengths: Executes quickly and is simple to implement.
  - Weaknesses: Can lead to suboptimal solutions due to its greedy nature.
- A* Search:
  - Strengths: Balances optimality and efficiency with heuristic functions.
  - Weaknesses: Requires careful selection and tuning of heuristic functions.

Total Costs and Sizes:

- Total Cost - Uniform Cost Search: 30788
- Total Cost - Greedy Search: 700
- Total Cost - A* Search: 31488
- Size - Uniform Cost Search: 10
- Size - Greedy Search: 10
- Size - A* Search: 10

6. **Evaluation and Reporting:**

Best Algorithm: Greedy Search
Best Cost: 700
Best Arrangement: [Kamal, Samir, Khalid, Ayman, Hakam, Ahmed, Fuad, Salem, Hani, Ibrahim]

6. Heuristic Evaluation Function:

The provided dislike matrix represents the heuristic evaluation function $h(n)$. Each value in the matrix denotes the dislike percentage between individuals, influencing the cost calculations and search strategies of the algorithms.

7. Conclusion:

In conclusion, after evaluating the performance of the Uniform Cost Search (UCS), Greedy Search, and A* Search algorithms, several insights have emerged:

1. Uniform Cost Search (UCS): This algorithm demonstrated superior performance in finding optimal solutions with the least conflict. Its strengths lie in its ability to find optimal solutions with lower costs. However, it may face challenges with higher time complexity for large datasets, which can be considered a weakness in certain scenarios.
2. Greedy Search: The Greedy Search algorithm has strengths in executing quickly and being simple to implement, as evidenced by its best cost of 700 and the resulting arrangement of [Kamal, Samir, Khalid, Ayman, Hakam, Ahmed, Fuad, Salem, Hani, Ibrahim]. However, its greedy nature can lead to suboptimal solutions, particularly in complex optimization problems.
3. A* Search: The A* Search algorithm strikes a balance between optimality and efficiency by utilizing heuristic functions. While it had a total cost of 31488 and the arrangement [Ibrahim, Hani, Salem, Fuad, Ahmed, Hakam, Ayman, Khalid, Samir, Kamal], it requires careful selection and tuning of heuristic functions, which can be considered a potential weakness.

Further analysis and experimentation with different problem sizes and heuristic functions may provide deeper insights into algorithm performance and suitability for specific use cases. Overall, each algorithm has its strengths and weaknesses, and the choice of algorithm should be made based on the specific requirements and constraints of the problem at hand.

DONE