**COMP 8700 Fall 2020 Final Project**

**Darshil M. Bhakhar**,

Master of Applied Computing,

University of Windsor,

bhakhar@uwindsor.ca

**Gurneet Singh**,

Master of Applied Computing,

University of Windsor,

gurneet@uwindsor.ca

**Dharmikkumar V. Patel**,

Master of Applied Computing,

University of Windsor,

patel8m2@uwindsor.ca

**Dharmikkumar V. Patel**,

Master of Applied Computing,

University of Windsor,

patel8m2@uwindsor.ca

**Wen Dong**,

Master of Applied Computing,

University of Windsor,

dong23@uwindsor.ca

***Abstract –*** this is the report of our team final project of Artificial Intelligence Introduction, in this project, we have experimented and explored a set of search algorithms including informed and uninformed ones with the famous Eight Puzzles Game, and found that the performance of different algorithms varies markedly in terms of optimality, completeness, time complexity and space complexity; we also found that heuristic function plays a significant role in the search for informed algorithms, and impact the outcome tremendously.

1. **Introduction**

The project is the final project of our AI Introduction course, which is to put the comprehensive knowledge and theory we have learned from class into reality, from abstract intuition to tangible experiment. It is beneficial for all the team members to consolidate the learning and build hands-on experience on Artificial Intelligence.

In the project, we have chosen and explored a set of informed algorithms:

AStarSearch – A graph traversal and path search algorithm, an import and widely used search algorithm in computer science due to its completeness, optimality and optimal efficiency. It uses function f(n) = g(n) + h(n) as estimate function, based on which to expand cheapest node first. The sole drawback is its space complexity therefore not practical for large-scale problems.

RecursiveBestFirstSearch – It is a variant version of A Star for large-scale problems where A Star is unable to tackle due to space complexity.

GreedyBestFirstSearch – it is designed to find goal quickly by expanding the node that is closest to the goal, on the grounds that this is likely to lead to a solution quickly. By f(n) = h(n). So, nodes not on the solution path are not expanded.

Cost low but not optimal as not complete because optimal solution might be in other unexplored paths.

and a couple of uninformed as well:

BreadFirstSearch

UniformCostSearch

References:

[https://en.wikipedia.org/wiki/A\*\_search\_algorithm](https://en.wikipedia.org/wiki/A*_search_algorithm)