

DATABASE MANAGEMENT SYSTEMS (CS 581)

SEARCH OF SPATIO – TEMPORAL RESOURCES FINAL PROJECT REPORT

GROUP 1

MINU FRANCIS

POOJA DONEKAL

SHVETHA SUVARNA

SURBHI ARORA

Table of Contents

Contents

1. Introduction.....	5
2. Motivation.....	5
3. Goal.....	5
4. Dataset.....	6
4.1 Provided Files	6
4.2 Data Plot.....	6
5. System Architecture.....	7
5.1. Overview.....	7
5.2. User Interface of the Web Application	7
5.3. System Architecture.....	8
5.3.1. User Interface.....	8
5.3.2. Business Logic	8
5.3.3. Data Access Module	8
5.3.4. Database	9
5.3.5. External Interface.....	9
5.3.6. Google Maps API.....	9
6. Methodologies.....	9
6.1. Level of Information - Real Time Parking Information is provided.....	9
6.1.1. Problem Description	9
6.1.2. Holistic Architecture	9
6.1.3. Pre-Analysis of given data	10
6.1.4. Alternate Methods Considered.....	11
6.1.5. Method Chosen - Parking Slot Availability determination based on the Gravitational Pull Algorithm.....	11
6.2. Level of Information – Historical Data Regarding Parking is provided	21
6.2.1. Problem Description	21
6.2.2. Holistic Architecture	22
6.2.3. Pre-Analysis of Given Data	22

6.2.4.	Alternate Methods Considered.....	23
6.2.5.	Method Chosen: Parking Slot Availability estimation based on Weighted Average.....	23
6.3.	Level of Information - Parking with no data about the parking slots availability.....	34
6.3.1.	Problem Description	34
6.3.2.	Holistic Architecture	34
6.3.3.	Alternate Methods Considered.....	35
6.3.4.	Method Chosen - Parking Slot Availability estimation based on nearest neighbor to the source location, in a region having pre-defined boundaries.....	35
7.	Comparison of algorithms.....	45
8.	Technologies Used.....	46
9.	Steps to Run the Project.....	46
10.	Acknowledgment	46
	References.....	48

LIST OF FIGURES

Figure 1 Visualization of the Intersections Provided	6
Figure 2 User Interface of the Web Application	7
Figure 3 System Architecture	8
Figure 4 Holistic Architecture	10
Figure 5 Screenshot of Pre cached Data	11
Figure 6 Simulation of Gravitational Pull Algorithm	12
Figure 7 Flow Chart for GPA	13
Figure 8 Test Results for Gravitational Pull Algorithm with 0% Congestion Level	15
Figure 9 Test Results for Gravitational Pull Algorithm with 10% Congestion Level	16
Figure 10 Test Results for Gravitational Pull Algorithm with 20% Congestion Level	16
Figure 11 Test Results for Gravitational Pull Algorithm with 30% Congestion Level	17
Figure 12 Test Results for Gravitational Pull Algorithm with 40% Congestion Level	17
Figure 13 Test Results for Gravitational Pull Algorithm with 50% Congestion Level	18
Figure 14 Plot between 'Time to find Parking' and 'Congestion Levels'	19
Figure 15 Plot between 'Query Time' and 'Congestion Level' for GPA	20
Figure 16 Graph Comparing GPA and Greedy Approach to find Parking using Real Time Data	21
Figure 17 Holistic Architecture for Algorithm based on Historic Data	22
Figure 18 Pre - Cached Data: Weighted Average based on Day and Time(per hour)	24
Figure 19 Flow Chart for algorithm implemented using Historic Data	25
Figure 20 Screenshot of Test Data used for experiments	26
Figure 21 Test Results for Historic Data Based Algorithm with 0% Congestion	28
Figure 22 Test Results for Historic Data Based Algorithm with 10% Congestion	28
Figure 23 Test Results for Historic Data Based Algorithm with 20% Congestion	29
Figure 24 Test Results for Historic Data Based Algorithm with 30% Congestion	29
Figure 25 Test Results for Historic Data Based Algorithm with 40% Congestion	30
Figure 26 Test Results for Historic Data Based Algorithm with 50% Congestion	30
Figure 27 Graph between 'Time Taken by Algorithm' and 'Congestion Percentage'	31
Figure 28 Graph between 'Total Time Taken by Algorithm' and 'Congestion Percentage'	32
Figure 29 Graph between 'Time taken to Find Parking' and 'Congestion Percentage'	33
Figure 30 Output of the Parking Search on the Web application	34
Figure 31 Holistic Architecture for Uninformed Search	34
Figure 32 Simulation of Uninformed Search Algorithm	36
Figure 33 Flow Chart for the Uninformed Search Algorithm	37
Figure 34 Test Results for Uninformed Search with 0% Congestion	39
Figure 35 Test Results for Uninformed Search with 10% Congestion	39
Figure 36 Test Results for Uninformed Search with 20% Congestion	40
Figure 37 Test Results for Uninformed Search with 30% Congestion	40
Figure 38 Test Results for Uninformed Search with 40% Congestion	41
Figure 39 Test Results for Uninformed Search with 50% Congestion	41
Figure 40 Graph between 'Time to Find Parking' and 'Congestion Levels'	42
Figure 41 Graph between Query Time and Congestion Levels	43
Figure 42 Output of the Uninformed Search on Web Application	44
Figure 43 Graph showing the Comparison between various algorithms implemented	45

LIST OF TABLES

Table 1	Dataset Provided.....	6
Table 2	Alternate Algorithms Considered	11
Table 3	Alternate Methods Considered for implementing Historical Data Based Algorithm	23
Table 4	Pre cached Data for Historic Data based algorithm	23
Table 5	Alternate Methods Considered for Uninformed Search	35

Search of Spatio-Temporal Resources

1. Introduction

In major cities it has been observed that 30% of the traffic comprises of individuals cruising for a parking spot. This shows that the search for traffic is an enormous source of congestion. There are a number of projects whose sole aim is reduce this problem. One instance taken into consideration is the SFPark project. The SFPark is a project initiated by the San Francisco Municipal Transportation Agency, in which wireless sensors were embedded under the pavement in order to provide real time data. This application provides a list of available parking spots and the user would manually have to choose one. This would make it difficult for the user to make a decision while driving. In order to solve this issue we aim to create an algorithm that provides directions to the users at each intersection, from the start location to the destination.

2. Motivation

In order to reduce the contribution of individuals looking for a parking spot to the regular flow of traffic, a method needs to be devised in order to direct the user to a place where they would find a parking slot closer to their destination. The destination should kept in mind in order to take into consideration, the distance the user has to walk from their parking slot to their destination.

3. Goal

The goal of this project is to try to address the issue of parking by evaluating various algorithms for various instances of the problem. This project will consider three instances of Spatio-Temporal data and run various algorithms on these instances. This first instance will consider the given data as real time data and run the Real Time based algorithm on it. The second instance will consider the data as historical data and run the appropriate algorithm whereas the third instance will be completely uninformed. This will be then evaluated against certain metrics than can help us judge the performance of one algorithm against the other.

4. Dataset

4.1 Provided Files

The dataset provided is from the San Francisco's tourist area. There are 40 nodes and 63 edges in this road network. It consists of three separate CSV files containing the following data.

Data	Description
SFPark_nodes_FishermansWharf.csv	Contains node data
SFPark_edges_FishermansWharf2.csv	Contains edge data
dbProjection_4_6_12-5_5_12.csv	Contains the list of available parking slots

Table 1 Dataset Provided

4.2 Data Plot

The diagram is a plot of all the possible intersections in the given data. This diagram was used a reference to visualize the various intersections while testing.

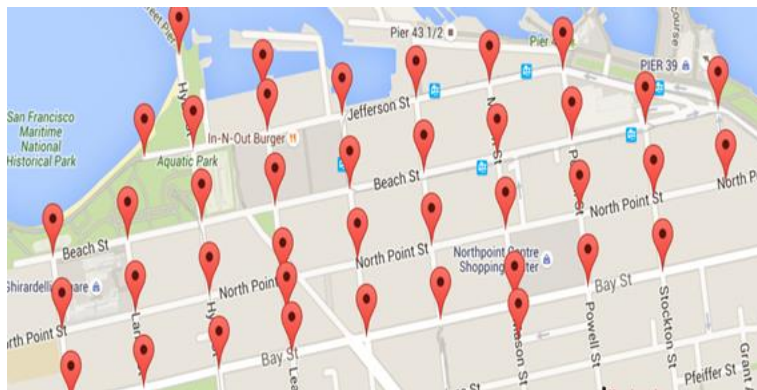


Figure 1 Visualization of the Intersections Provided

5. System Architecture

5.1. Overview

The system is implemented in the form of a web based application which takes parking requests from the user, performs the necessary calculations and directs the users at each intersection until they find a parking slot.

5.2. User Interface of the Web Application

Below is the screen shot of the web application the user will interface with.



Figure 2 User Interface of the Web Application

5.3. System Architecture

Below is the overall architecture for the system.

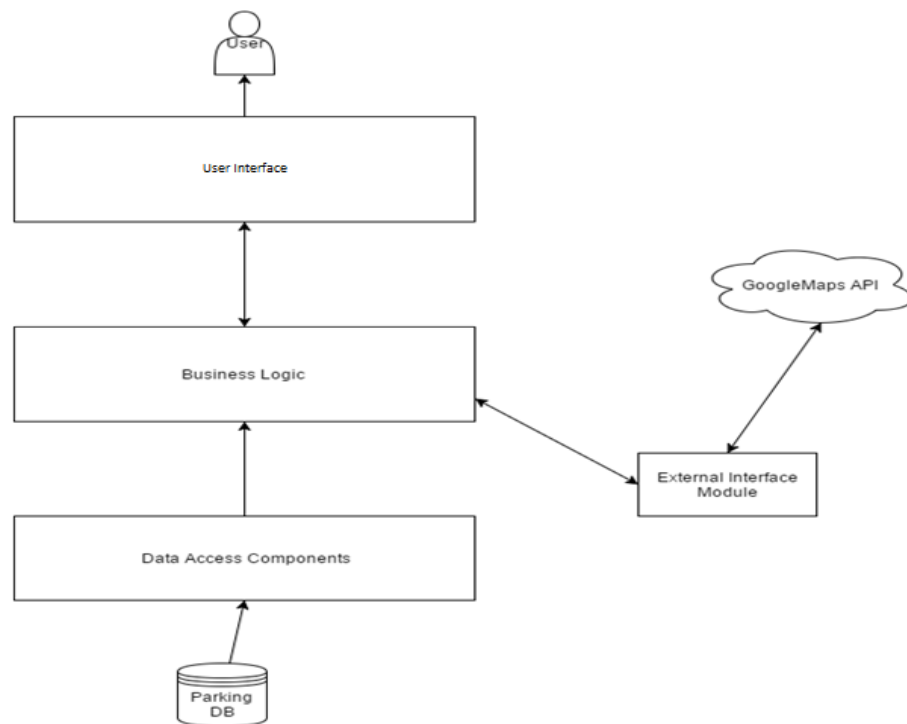


Figure 3 System Architecture

The description of the individual modules are as follows:

5.3.1. User Interface

The User Interface Module provides a web based interface for the user to interact with the system. This interface takes input parameters for location and time. Based on the values passed, it interacts with the business logic to provide the necessary output. This is the only means for the user to communicate with the application.

5.3.2. Business Logic

This component is responsible for making computations based on the algorithm used. This is the heart of the entire system. It interacts with the Data Access component and all other external interfaces to fetch data. The final output from this module is sent to the User Interface module.

5.3.3. Data Access Module

This module directly interfaces with the database. The necessary queries are run in this module are provided as an output to the Business layer.

5.3.4. Database

This layer is populated with the provided and pre-cached data .It stores all the information relevant to the application.

5.3.5. External Interface

This layers acts as an interface for the application to interact with external REST Services. The data obtained from the REST services is fed into the application from this layer.

5.3.6. Google Maps API

This layer is an external interface which provides the necessary output as a response to the application's request. The output from this module is processed by the business layer and is displayed to the user from the User Interface module.

6. Methodologies

Each of algorithms for the individual tasks will be covered under this section.

6.1. Level of Information - Real Time Parking Information is provided

6.1.1. Problem Description

Assuming you lived in San Francisco and had access to the real time data from SFpark. In this task, we are devising an algorithm(s) which guides a user to find a street parking spot. We are using Gravitational Pull of a parking spot which is Analogous to physics. We can define the gravitational force of a parking block b towards a vehicle v as

$$F(v, b) = n_b / cost(v, b)^2$$

n_b : The number of available parking spots at block b.

Cost: The time taken by the vehicle to reach the parking spot plus time taken by the user to walk back[1].

Average Driving Speed Taken: 20 miles/hour

Average Walking Time Per Hour: 3 miles/hour

6.1.2. Holistic Architecture

This architecture represents the modified version of the system architecture in order to suit the needs of the algorithm used in task 1.

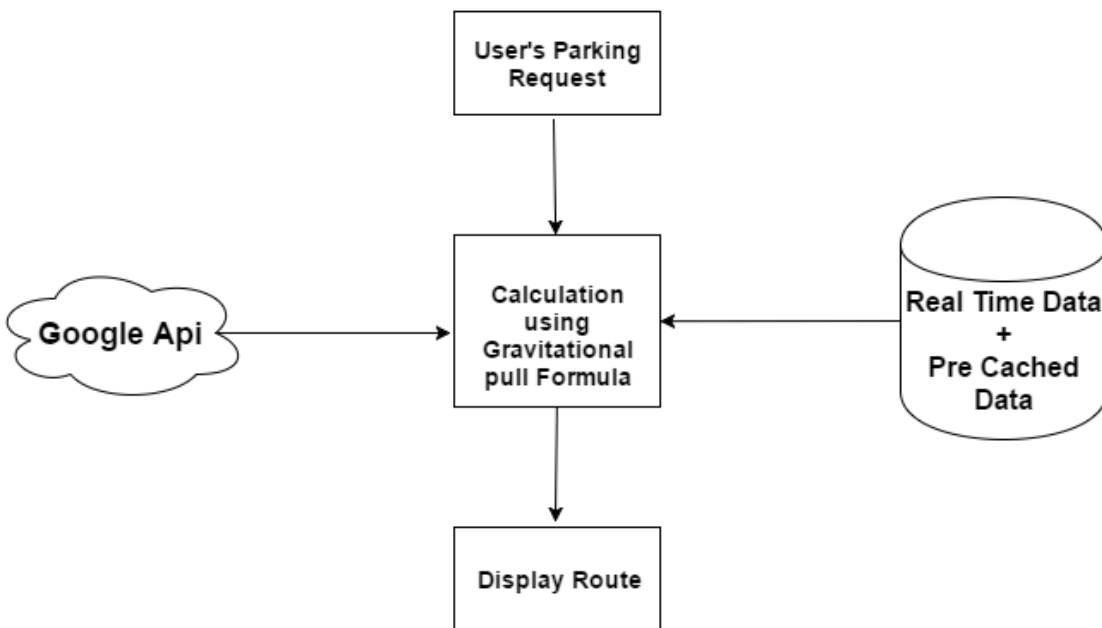


Figure 4 Holistic Architecture

6.1.3. Pre-Analysis of given data

For any input, the algorithm should be able to determine the availability of the parking spot at that location, for the time of the current request based on its current availability. It should also consider the effect of all the parking slots that can be reached by taking a certain turn at an intersection.

6.1.4. Alternate Methods Considered

Some of the methods that were taken into consideration before coming up with the solution were as follows.

No	Method Name	Method Description	Drawback
1	Greedy Approach – Max Slot	The block containing the maximum number of parking slots will be chosen as “Optimal” by the algorithm.	There might be a number of optimal parking slots which are close to the user’s location and this technique fails to take that into consideration.
2	Greedy Approach – Closest Slot	The block closest to the user will be chosen and the user will be redirected to that particular slot	Since the closest slot will be chosen by the algorithm cannot guarantee the availability of the slot once the user reaches the location. This might lead to the algorithm redirecting to different parking spots multiple times.

Table 2 Alternate Algorithms Considered

6.1.5. Method Chosen - Parking Slot Availability determination based on the Gravitational Pull Algorithm

6.1.5.1 Pre-Cached data

The following data is pre-cached in order to enable the algorithm to perform efficiently at runtime. We will pre-cache the distances from and to the nodes and its neighbors in order to reduce the calculation overhead.

The data pre-cached here is the distance from each of the intersections to its neighbors.

node_id	block_f_level	f_level	s_level	t_level
7021	{585051L: 0.06744843423840868, 585052L: 0....	{7016: 0.028866070058155373, 7025: 0.0674...	{7016: {7017: 0.04612540705285345, 7010: 0...	{7016: {7010: {7009: 0.09106012220810651, ...
7022	{363121L: 0.05579318673169384, 363122L: 0....	{7016: 0.02405686330316821, 7026: 0.06872...	{7016: {7017: 0.041316200297866285, 7010: ...	{7016: {7010: {7009: 0.08625091545311935, ...
7023	{681272L: 0.04605141528660256, 847011L: ...	{7024: 0.040340991364960284, 7019: 0.0697...	{7024: {7025: 0.07948240452160374, 7020: 0...	{7024: {7025: {7026: 0.12392711601004677, ...
7024	{681261L: 0.04098086143655241, 681262L: 0....	{7025: 0.03914141315664345, 7020: 0.06713...	{7025: {7026: 0.08358612464508647, 7021: 0...	{7025: {7026: {7022: 0.15230919640666132, ...
7025	{681251L: 0.04444471148844303, 681252L: 0....	{7024: 0.03914141315664345, 7026: 0.04444...	{7024: {7028: 0.1014828697064615, 7020: 0....	{7024: {7020: {7019: 0.13079706243543848, ...
7026	{325052L: 0.06872307176157486, 325042L: 0....	{7025: 0.04444471148844303, 7030: 0.05397...	{7025: {7024: 0.08358612464508647, 7021: 0...	{7025: {7024: {7028: 0.1459275811949045, 7...
7027	{847001L: 0.0626439026898332, 847011L: 0.0...	{7032: 0.0626439026898332, 7028: 0.040949...	{7032: {}, 7028: {7024: 0.1032905453378668...	{7032: {}, 7028: {7024: {7025: 0.1424319584...
7028	{326032L: 0.06234145654981805, 326022L: 0....	{7024: 0.06234145654981805, 7033: 0.06832...	{7024: {7025: 0.1014828697064615, 7020: 0....	{7024: {7025: {7026: 0.14592758119490454, ...
7029	{562241L: 0.04509222487083086, 562242L: 0....	{7025: 0.053314237991754033, 7034: 0.0390...	{7025: {7024: 0.09245565114839749, 7026: 0...	{7025: {7024: {7020: 0.1595899593452143, 7...
7030	{562241L: 0.04509222487083086, 562242L: 0....	{7026: 0.0539714112310959, 7035: 0.067165...	{7026: {7025: 0.09841612271953892, 7022: 0...	{7026: {7025: {7024: 0.13755753587618236, ...
7031	{562231L: 0.007741803133022389}	{7030: 0.007741803133022389}	{7030: {7026: 0.06171321436411829, 7035: 0...	{7030: {7026: {7025: 0.10615792585256131, ...
7032	{847001L: 0.0626439026898332}	{7027: 0.0626439026898332}	{7027: {7028: 0.10359299147788203, 7023: 0...	{7027: {7028: {7024: 0.16593444802770008, ...
7033	{561624L: 0.04439591028636684, 616242L: 0....	{7034: 0.01094883912860559, 7028: 0.06832...	{7034: {7035: 0.03883749266390432, 7029: 0...	{7034: {7035: {7030: 0.10600313778234434}, ...
7034	{561623L: 0.027888653535298732, 585032L: ...	{7033: 0.01094883912860559, 7035: 0.02788...	{7033: {7028: 0.0792758381852184}, 7035: {...	{7033: {7028: {7024: 0.14161729473503645, ...
7035	{325032L: 0.06716564511844002, 616231L: 0....	{7034: 0.027888653535298732, 7030: 0.0671...	{7034: {7033: 0.03883749266390432, 7029: 0...	{7034: {7033: {7028: 0.10716449172051713}, ...
7036	{664232L: 0.03154334752334917}	{7037: 0.03154334752334917}	{7037: {7040: 0.08875901742625214, 7034: 0...	{7037: {7034: {7033: 0.10359041396582178, ...
7037	{664232L: 0.03154334752334917, 585012L: 0....	{7040: 0.057215669902902964, 7034: 0.0610...	{7040: {}, 7034: {7033: 0.0720470664424726...	{7040: {}, 7034: {7033: {7028: 0.1403740654...

Figure 5 Screenshot of Pre cached Data

The distance between neighbors is pre computed and stored in the form of a JSON Object where the key is the intersection ID and the value is the distance between two intersections.

6.1.5.2 Algorithm

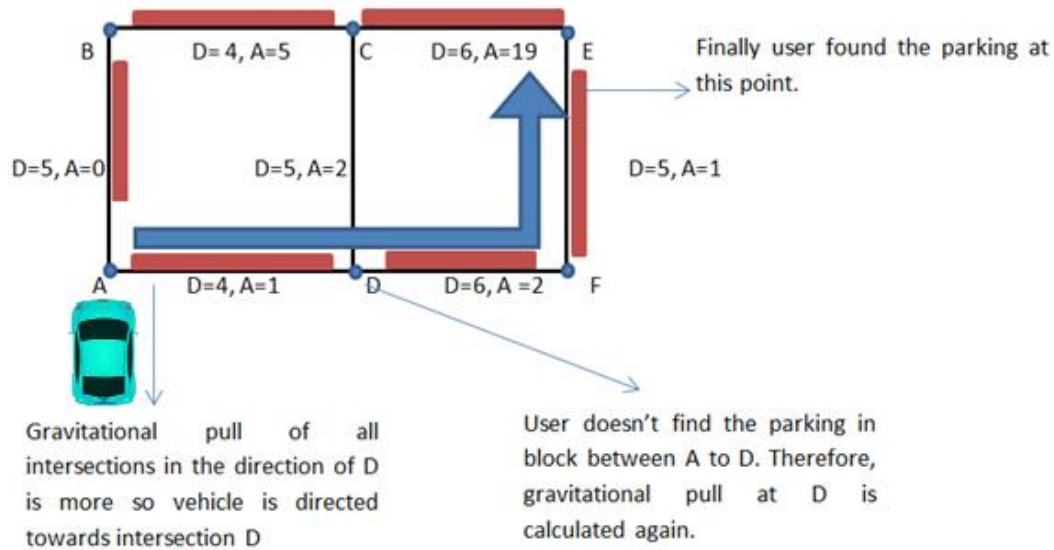


Figure 6 Simulation of Gravitational Pull Algorithm

1. The input taken by the algorithm will be the user's current location and the parking request time.
2. If the user is not at an intersection, he first will be directed to the closest intersection.
3. The algorithm will use the pre-computed data to fetch the neighbors and distance from the user's location.
4. Based on the values fetched the gravitational pull formulae will be applied to all the intersection.

$$F(v, b) = n_b / \text{cost}(v, b)^2$$

5. Based on the value obtained from this formula the user will be sent in the direction with the maximum gravitational pull.
6. If the user does not find a parking slot, the computations will be performed once again using current intersection as the source and the current time stamp as the request time.
7. If the user location is matched to latitude, longitude of the available parking slot at that time, parking is found and the algorithm will terminate.

6.1.5.3 Flowchart

Below is the flow chart representing the steps to implement algorithm to find parking spot using Gravitational Pull formula.

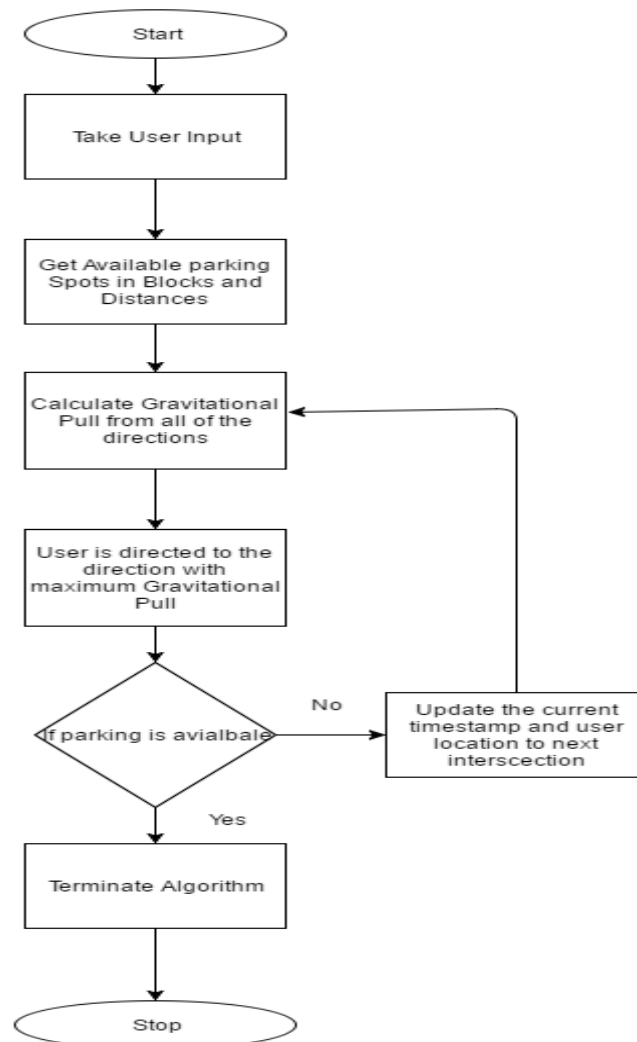


Figure 7 Flow Chart for GPA

6.1.5.4 Individual Experiments

The Experiments were conducted based on the following metrics

- Time taken for the algorithm to find a parking spot under various levels of congestion
- Time taken for the user to find a parking spot under various levels of congestion

6.1.5.5 Experiment Inputs

The test data is stored in a file named “TestData.xls”. The csv file contains the following data corresponding to 225 different source – timestamp combinations.

1. Latitude – latitude of the initial position of user
2. Longitude – longitude of the initial position of user
3. Timestamp – Time at which user is searching for parking

6.1.5.6 Experiment Methodology

In this case, experiments are conducted by adding congestion. Congestion is induced by reducing the number of available parking slots by a certain percentage of the maximum possible value of the parking slot availability as given in the database (i.e. 19). The experiment is conducted for various levels of congestion – 10%, 20%, 30%, 40% and 50%. The test data is stored in a file named “TestData.xls”.

The csv file contains the following data:

1. Latitude – latitude of the initial position of user
2. Longitude – longitude of the initial position of user
3. Timestamp – Time at which user is searching for parking
4. Congestion Percent – The congestion percentage that is taken into consideration.

6.1.5.7 Experiment Steps

In order to find the results of the algorithm, a button named ‘Show Results’ is available. On click of this button, the data in the spreadsheet is taken as input and the algorithm is run by considering the inputs one at a time. The results of the test cases are stored in the database and displayed on the UI.

6.1.5.8 Experiment Results

Results of the experiments conducted for GPA are displayed below.

- UserLocationLatitude/UserLocationLongitude – Initial start location of User
- ParkingSlotLatitude/ParkingSlotLongitude – Location of the allotted parking block
- StartTime – Time at which user starts searching for a parking

- Parking_Time(Minutes) – Time taken by the user to find a parking block.
- WalkingDistance(Miles) – Distance taken by the user to walk to destination from the allotted parking block
- Walking Time(Minutes) – Time taken by the user to walk to destination from the allotted parking block

6.1.5.8.1. Congestion Level – 0%

TEST RESULTS WITHOUT CONGESTION

Id	UserLocationLatitude	UserLocationLongitude	ParkingSlotLatitude	ParkingSlotLongitude	StartTime	Parking_Time	WalkingDistance	WalkingTime
1	37.805530	-122.416246	37.805232	-122.416911	2012-04-10 18:11:03	0	0.04	0.8
2	37.806948	-122.414740	37.806584	-122.413821	2012-04-10 18:11:03	0	0.05	1
3	37.804267	-122.413609	37.805864	-122.411986	2012-04-10 18:11:01	0	0.12	2.4
4	37.805241	-122.416902	37.805232	-122.416911	2012-04-10 18:11:03	0	0	0
5	37.806149	-122.415531	37.806584	-122.413821	2012-04-10 18:11:04	0	0.09	1.8
6	37.807926	-122.417453	37.806584	-122.413821	2012-04-10 18:11:03	0	0.19	3.8
7	37.805180	-122.422308	37.806461	-122.422216	2012-04-10 18:11:02	0	0.08	1.6
8	37.805530	-122.416246	37.805232	-122.416911	2012-04-06 21:29:01	0	0.04	0.8
9	37.806948	-122.414740	37.807508	-122.414014	2012-04-06 21:29:01	0	0.05	1
10	37.804267	-122.413609	37.807508	-122.414014	2012-04-06 21:29:01	0	0.2	4
11	37.805241	-122.416902	37.805232	-122.416911	2012-04-06 21:29:01	0	0	0
12	37.806149	-122.415531	37.806584	-122.413821	2012-04-06 21:29:04	0	0.09	1.8
13	37.807926	-122.417453	37.807508	-122.414014	2012-04-06 21:29:01	0	0.16	3.2
14	37.805180	-122.422308	37.805520	-122.422044	2012-04-06 21:29:01	0	0.02	0.4
15	37.805530	-122.416246	37.805232	-122.416911	2012-04-06 00:09:01	0	0.04	0.8
16	37.806948	-122.414740	37.807508	-122.414014	2012-04-06 00:09:01	0	0.05	1
17	37.804267	-122.413609	37.805864	-122.411986	2012-04-06 00:09:01	0	0.12	2.4
18	37.805241	-122.416902	37.805232	-122.416911	2012-04-06 00:09:01	0	0	0
19	37.806149	-122.415531	37.807508	-122.414014	2012-04-06 00:09:09	0	0.11	2.2
20	37.807926	-122.417453	37.807508	-122.414014	2012-04-06 00:09:01	0	0.16	3.2
21	37.805180	-122.422308	37.805520	-122.422044	2012-04-06 00:09:05	0	0.02	0.4
22	37.806439	-122.415887	37.806584	-122.413821	2012-04-12 16:25:02	0	0.1	2
23	37.805355	-122.415780	37.806584	-122.413821	2012-04-12 16:25:03	0	0.12	2.4
24	37.806162	-122.417074	37.805232	-122.416911	2012-04-12 16:25:01	0	0.06	1.2
25	37.800011	-122.415745	37.805232	-122.416911	2012-04-16 10:06:09	0	0.32	6.4
26	37.805014	-122.412807	37.805652	-122.413617	2012-04-16 10:06:02	0	0.05	1
27	37.807543	-122.413606	37.805444	-122.415275	2012-04-16 10:06:04	0	0.15	3
28	37.807734	-122.414388	37.805444	-122.415275	2012-04-16 10:06:04	0	0.14	2.8
29	37.805530	-122.416246	37.805232	-122.416911	2012-04-10 18:11:03	0	0.04	0.8
30	37.806948	-122.414740	37.806584	-122.413821	2012-04-10 18:11:03	0	0.05	1

Figure 8 Test Results for Gravitational Pull Algorithm with 0% Congestion Level

6.1.5.8.2. Congestion Level – 10%

TEST RESULTS WITH CONGESTION

Id	UserLocationLatitude	UserLocationLongitude	ParkingSlotLatitude	ParkingSlotLongitude	CongestionPercent	StartTime	Parking_Time	WalkingDistance	WalkingTime
1	37.805530	-122.416246	37.805652	-122.413617	10	2012-04-10 18:11:03	0.12	2.4	
2	37.806948	-122.414740	37.807309	-122.415639	10	2012-04-10 18:11:09	0.05	1	
3	37.804267	-122.413609	37.805864	-122.411986	10	2012-04-10 18:11:01	0.12	2.4	
4	37.805241	-122.416902	37.805652	-122.413617	10	2012-04-10 18:11:03	0.16	3.2	
5	37.806149	-122.415531	37.807309	-122.415639	10	2012-04-10 18:11:06	0.07	1.4	
6	37.807926	-122.417453	37.808436	-122.414186	10	2012-04-10 18:11:03	0.16	3.2	
7	37.805180	-122.422308	37.805520	-122.422044	10	2012-04-10 18:11:07	0.02	0.4	
8	37.805530	-122.416246	37.805444	-122.415275	10	2012-04-06 21:29:018	0.05	1	
9	37.806948	-122.414740	37.807101	-122.417292	10	2012-04-06 21:29:02	0.12	2.4	
10	37.804267	-122.413609	37.806190	-122.417115	10	2012-04-06 21:29:020	0.2	4	
11	37.805241	-122.416902	37.805444	-122.415275	10	2012-04-06 21:29:018	0.08	1.6	
12	37.806149	-122.415531	37.807101	-122.417292	10	2012-04-06 21:29:07	0.1	2	
13	37.807926	-122.417453	37.808245	-122.415816	10	2012-04-06 21:29:08	0.08	1.6	
14	37.805180	-122.422308	37.805520	-122.422044	10	2012-04-06 21:29:020	0.02	0.4	
15	37.805530	-122.416246	37.805232	-122.416911	10	2012-04-06 00:09:020	0.04	0.8	
16	37.806948	-122.414740	37.806190	-122.417115	10	2012-04-06 00:09:09	0.12	2.4	
17	37.804267	-122.413609	37.805652	-122.413617	10	2012-04-06 00:09:014	0.08	1.6	
18	37.805241	-122.416902	37.805232	-122.416911	10	2012-04-06 00:09:020	0	0	
19	37.806149	-122.415531	37.807309	-122.415639	10	2012-04-06 00:09:06	0.07	1.4	
20	37.807926	-122.417453	37.808436	-122.414186	10	2012-04-06 00:09:01	0.16	3.2	
21	37.805180	-122.422308	37.805520	-122.422044	10	2012-04-06 00:09:05	0.02	0.4	
22	37.806439	-122.415887	37.807826	-122.419110	10	2012-04-12 16:25:020	0.17	3.4	
23	37.805355	-122.415780	37.805652	-122.413617	10	2012-04-12 16:25:02	0.1	2	
24	37.806162	-122.417074	37.806190	-122.417115	10	2012-04-12 16:25:01	0	0	
25	37.800011	-122.415745	37.805652	-122.413617	10	2012-04-16 10:06:01	0.35	7	
26	37.805014	-122.412807	37.805652	-122.413617	10	2012-04-16 10:06:02	0.05	1	
27	37.807543	-122.413606	37.807309	-122.415639	10	2012-04-16 10:06:04	0.1	2	
28	37.807734	-122.414388	37.807309	-122.415639	10	2012-04-16 10:06:04	0.06	1.2	

Figure 9 Test Results for Gravitational Pull Algorithm with 10% Congestion Level

6.1.5.8.3. Congestion Level – 20%

29	37.805530	-122.416246	37.805232	-122.416911	20	2012-04-10 18:11:020	0.04	0.8	
30	37.806948	-122.414740	37.806893	-122.418944	20	2012-04-10 18:11:020	0.2	4	
31	37.804267	-122.413609	37.805864	-122.411986	20	2012-04-10 18:11:01	0.12	2.4	
32	37.805241	-122.416902	37.805232	-122.416911	20	2012-04-10 18:11:020	0	0	
33	37.806149	-122.415531	37.807309	-122.415639	20	2012-04-10 18:11:06	0.07	1.4	
34	37.807926	-122.417453	37.806893	-122.418944	20	2012-04-10 18:11:020	0.09	1.8	
35	37.805180	-122.422308	37.805520	-122.422044	20	2012-04-10 18:11:07	0.02	0.4	
36	37.805530	-122.416246	37.805444	-122.415275	20	2012-04-06 21:29:018	0.05	1	
37	37.806948	-122.414740	37.806584	-122.413821	20	2012-04-06 21:29:020	0.05	1	
38	37.804267	-122.413609	37.806190	-122.417115	20	2012-04-06 21:29:020	0.2	4	
39	37.805241	-122.416902	37.805444	-122.415275	20	2012-04-06 21:29:018	0.08	1.6	
40	37.806149	-122.415531	37.807826	-122.419110	20	2012-04-06 21:29:020	0.2	4	
41	37.807926	-122.417453	37.808436	-122.414186	20	2012-04-06 21:29:020	0.16	3.2	
42	37.805180	-122.422308	37.805520	-122.422044	20	2012-04-06 21:29:020	0.02	0.4	
43	37.805530	-122.416246	37.805232	-122.416911	20	2012-04-06 00:09:020	0.04	0.8	
44	37.806948	-122.414740	37.806190	-122.417115	20	2012-04-06 00:09:09	0.12	2.4	
45	37.804267	-122.413609	37.805652	-122.413617	20	2012-04-06 00:09:014	0.08	1.6	
46	37.805241	-122.416902	37.805232	-122.416911	20	2012-04-06 00:09:020	0	0	
47	37.806149	-122.415531	37.807309	-122.415639	20	2012-04-06 00:09:06	0.07	1.4	
48	37.807926	-122.417453	37.808436	-122.414186	20	2012-04-06 00:09:01	0.16	3.2	
49	37.805180	-122.422308	37.805520	-122.422044	20	2012-04-06 00:09:05	0.02	0.4	
50	37.806439	-122.415887	37.806381	-122.415468	20	2012-04-12 16:25:03	0.02	0.4	
51	37.805355	-122.415780	37.806190	-122.417115	20	2012-04-12 16:25:020	0.08	1.6	
52	37.806162	-122.417074	37.806893	-122.418944	20	2012-04-12 16:25:020	0.1	2	
53	37.800011	-122.415745	37.805232	-122.416911	20	2012-04-16 10:06:020	0.32	6.4	
54	37.805014	-122.412807	37.805652	-122.413617	20	2012-04-16 10:06:08	0.05	1	
55	37.807543	-122.413606	37.807309	-122.415639	20	2012-04-16 10:06:04	0.1	2	
56	37.807734	-122.414388	37.807309	-122.415639	20	2012-04-16 10:06:04	0.06	1.2	

Figure 10 Test Results for Gravitational Pull Algorithm with 20% Congestion Level

6.1.5.8.4. Congestion Level - 30%

57	37.805530	-122.416246	37.805232	-122.416911	30	2012-04-10 18:11:0	20	0.04	0.8
58	37.806948	-122.414740	37.806893	-122.418944	30	2012-04-10 18:11:0	20	0.2	4
59	37.804267	-122.413609	37.806190	-122.417115	30	2012-04-10 18:11:0	20	0.2	4
60	37.805241	-122.416902	37.805232	-122.416911	30	2012-04-10 18:11:0	20	0	0
61	37.806149	-122.415531	37.807826	-122.419110	30	2012-04-10 18:11:0	20	0.2	4
62	37.807926	-122.417453	37.808436	-122.414186	30	2012-04-10 18:11:0	20	0.16	3.2
63	37.805180	-122.422308	37.805520	-122.422044	30	2012-04-10 18:11:0	9	0.02	0.4
64	37.805530	-122.416246	37.805444	-122.415275	30	2012-04-06 21:29:0	18	0.05	1
65	37.806948	-122.414740	37.806893	-122.418944	30	2012-04-06 21:29:0	20	0.2	4
66	37.804267	-122.413609	37.806190	-122.417115	30	2012-04-06 21:29:0	20	0.2	4
67	37.805241	-122.416902	37.805444	-122.415275	30	2012-04-06 21:29:0	18	0.08	1.6
68	37.806149	-122.415531	37.807826	-122.419110	30	2012-04-06 21:29:0	20	0.2	4
69	37.807926	-122.417453	37.808436	-122.414186	30	2012-04-06 21:29:0	20	0.16	3.2
70	37.805180	-122.422308	37.805520	-122.422044	30	2012-04-06 21:29:0	20	0.02	0.4
71	37.805530	-122.416246	37.805232	-122.416911	30	2012-04-06 00:09:0	20	0.04	0.8
72	37.806948	-122.414740	37.807309	-122.415639	30	2012-04-06 00:09:0	19	0.05	1
73	37.804267	-122.413609	37.806190	-122.417115	30	2012-04-06 00:09:0	20	0.2	4
74	37.805241	-122.416902	37.805232	-122.416911	30	2012-04-06 00:09:0	20	0	0
75	37.806149	-122.415531	37.807309	-122.415639	30	2012-04-06 00:09:0	6	0.07	1.4
76	37.807926	-122.417453	37.808436	-122.414186	30	2012-04-06 00:09:0	1	0.16	3.2
77	37.805180	-122.422308	37.805520	-122.422044	30	2012-04-06 00:09:0	5	0.02	0.4
78	37.806439	-122.415887	37.807826	-122.419110	30	2012-04-12 16:25:0	20	0.17	3.4
79	37.805355	-122.415780	37.805652	-122.413617	30	2012-04-12 16:25:0	8	0.1	2
80	37.806162	-122.417074	37.806893	-122.418944	30	2012-04-12 16:25:0	20	0.1	2
81	37.800011	-122.415745	37.805232	-122.416911	30	2012-04-16 10:06:0	20	0.32	6.4
82	37.805014	-122.412807	37.805652	-122.413617	30	2012-04-16 10:06:0	8	0.05	1
83	37.807543	-122.413606	37.807309	-122.415639	30	2012-04-16 10:06:0	4	0.1	2
84	37.807734	-122.414388	37.807309	-122.415639	30	2012-04-16 10:06:0	4	0.06	1.2

Figure 11 Test Results for Gravitational Pull Algorithm with 30% Congestion Level

6.1.5.8.5. Congestion Level – 40%

85	37.805530	-122.416246	37.805232	-122.416911	40	2012-04-10 18:11:0	20	0.04	0.8
86	37.806948	-122.414740	37.806893	-122.418944	40	2012-04-10 18:11:0	20	0.2	4
87	37.804267	-122.413609	37.806190	-122.417115	40	2012-04-10 18:11:0	20	0.2	4
88	37.805241	-122.416902	37.805232	-122.416911	40	2012-04-10 18:11:0	20	0	0
89	37.806149	-122.415531	37.807826	-122.419110	40	2012-04-10 18:11:0	20	0.2	4
90	37.807926	-122.417453	37.808436	-122.414186	40	2012-04-10 18:11:0	20	0.16	3.2
91	37.805180	-122.422308	37.805520	-122.422044	40	2012-04-10 18:11:0	20	0.02	0.4
92	37.805530	-122.416246	37.805232	-122.416911	40	2012-04-06 21:29:0	20	0.04	0.8
93	37.806948	-122.414740	37.806893	-122.418944	40	2012-04-06 21:29:0	20	0.2	4
94	37.804267	-122.413609	37.806190	-122.417115	40	2012-04-06 21:29:0	20	0.2	4
95	37.805241	-122.416902	37.805232	-122.416911	40	2012-04-06 21:29:0	20	0	0
96	37.806149	-122.415531	37.807826	-122.419110	40	2012-04-06 21:29:0	20	0.2	4
97	37.807926	-122.417453	37.808436	-122.414186	40	2012-04-06 21:29:0	20	0.16	3.2
98	37.805180	-122.422308	37.805520	-122.422044	40	2012-04-06 21:29:0	20	0.02	0.4
99	37.805530	-122.416246	37.805232	-122.416911	40	2012-04-06 00:09:0	20	0.04	0.8
100	37.806948	-122.414740	37.807309	-122.415639	40	2012-04-06 00:09:0	19	0.05	1
101	37.804267	-122.413609	37.806190	-122.417115	40	2012-04-06 00:09:0	20	0.2	4
102	37.805241	-122.416902	37.805232	-122.416911	40	2012-04-06 00:09:0	20	0	0
103	37.806149	-122.415531	37.807309	-122.415639	40	2012-04-06 00:09:0	6	0.07	1.4
104	37.807926	-122.417453	37.808436	-122.414186	40	2012-04-06 00:09:0	20	0.16	3.2
105	37.805180	-122.422308	37.805520	-122.422044	40	2012-04-06 00:09:0	5	0.02	0.4
106	37.806439	-122.415887	37.807826	-122.419110	40	2012-04-12 16:25:0	20	0.17	3.4
107	37.805355	-122.415780	37.805652	-122.413617	40	2012-04-12 16:25:0	8	0.1	2
108	37.806162	-122.417074	37.806893	-122.418944	40	2012-04-12 16:25:0	20	0.1	2
109	37.800011	-122.415745	37.805232	-122.416911	40	2012-04-16 10:06:0	20	0.32	6.4
110	37.805014	-122.412807	37.805652	-122.413617	40	2012-04-16 10:06:0	8	0.05	1
111	37.807543	-122.413606	37.807309	-122.415639	40	2012-04-16 10:06:0	4	0.1	2
112	37.807734	-122.414388	37.807309	-122.415639	40	2012-04-16 10:06:0	4	0.06	1.2

Figure 12 Test Results for Gravitational Pull Algorithm with 40% Congestion Level

6.1.5.8.6. Congestion Level – 50%

113	37.805530	-122.416246	37.805232	-122.416911	50	2012-04-10 18:11:0	20	0.04	0.8
114	37.806948	-122.414740	37.806893	-122.418944	50	2012-04-10 18:11:0	20	0.2	4
115	37.804267	-122.413609	37.806190	-122.417115	50	2012-04-10 18:11:0	20	0.2	4
116	37.805241	-122.416902	37.805232	-122.416911	50	2012-04-10 18:11:0	20	0	0
117	37.806149	-122.415531	37.807826	-122.419110	50	2012-04-10 18:11:0	20	0.2	4
118	37.807926	-122.417453	37.808436	-122.414186	50	2012-04-10 18:11:0	20	0.16	3.2
119	37.805180	-122.422308	37.805520	-122.422044	50	2012-04-10 18:11:0	20	0.02	0.4
120	37.805530	-122.416246	37.805232	-122.416911	50	2012-04-06 21:29:0	20	0.04	0.8
121	37.806948	-122.414740	37.806893	-122.418944	50	2012-04-06 21:29:0	20	0.2	4
122	37.804267	-122.413609	37.806190	-122.417115	50	2012-04-06 21:29:0	20	0.2	4
123	37.805241	-122.416902	37.805232	-122.416911	50	2012-04-06 21:29:0	20	0	0
124	37.806149	-122.415531	37.807826	-122.419110	50	2012-04-06 21:29:0	20	0.2	4
125	37.807926	-122.417453	37.808436	-122.414186	50	2012-04-06 21:29:0	20	0.16	3.2
126	37.805180	-122.422308	37.805520	-122.422044	50	2012-04-06 21:29:0	20	0.02	0.4
127	37.805530	-122.416246	37.805232	-122.416911	50	2012-04-06 00:09:0	20	0.04	0.8
128	37.806948	-122.414740	37.807309	-122.415639	50	2012-04-06 00:09:0	19	0.05	1
129	37.804267	-122.413609	37.806190	-122.417115	50	2012-04-06 00:09:0	20	0.2	4
130	37.805241	-122.416902	37.805232	-122.416911	50	2012-04-06 00:09:0	20	0	0
131	37.806149	-122.415531	37.807309	-122.415639	50	2012-04-06 00:09:0	8	0.07	1.4
132	37.807926	-122.417453	37.808436	-122.414186	50	2012-04-06 00:09:0	20	0.16	3.2
133	37.805180	-122.422308	37.805520	-122.422044	50	2012-04-06 00:09:0	20	0.02	0.4
134	37.806439	-122.415887	37.807826	-122.419110	50	2012-04-12 16:25:0	20	0.17	3.4
135	37.805355	-122.415780	37.805652	-122.413617	50	2012-04-12 16:25:0	8	0.1	2
136	37.806162	-122.417074	37.806893	-122.418944	50	2012-04-12 16:25:0	20	0.1	2
137	37.800011	-122.415745	37.805232	-122.416911	50	2012-04-16 10:06:0	20	0.32	6.4
138	37.805014	-122.412807	37.805652	-122.413617	50	2012-04-16 10:06:0	8	0.05	1
139	37.807543	-122.413606	37.807309	-122.415639	50	2012-04-16 10:06:0	4	0.1	2
140	37.807734	-122.414388	37.807309	-122.415639	50	2012-04-16 10:06:0	4	0.06	1.2

Figure 13 Test Results for Gravitational Pull Algorithm with 50% Congestion Level

6.1.5.9 Experiment Analysis

6.1.5.9.1 Graph1: Time to find Parking Vs Congestion Levels

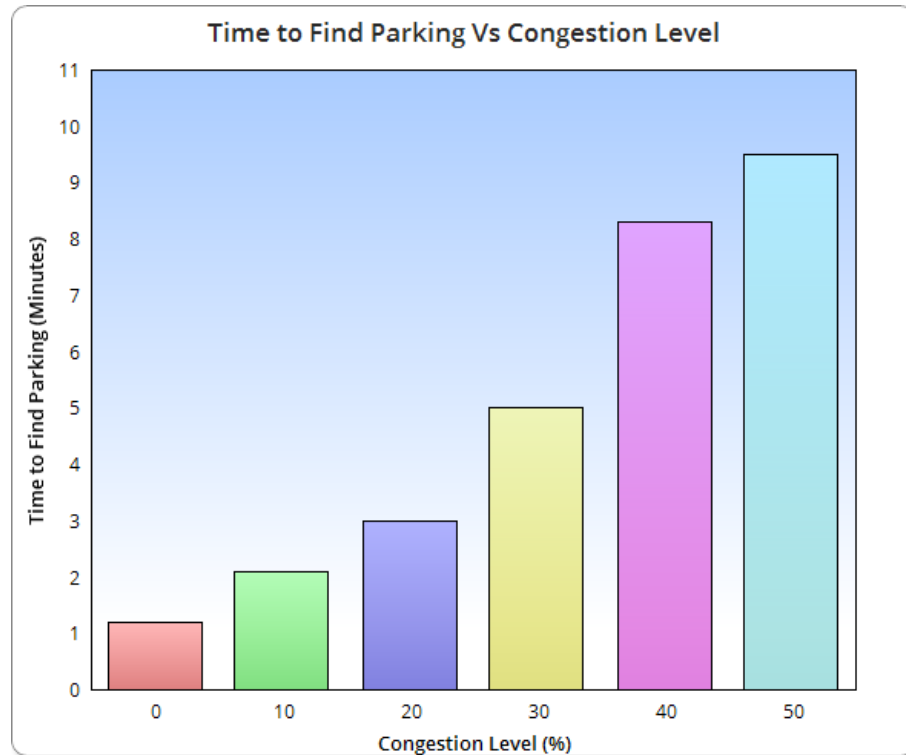


Figure 14 Plot between 'Time to find Parking' and 'Congestion Levels'

6.1.5.9.2 Graph Description

The above graph shows the analysis of 'Time taken to find parking' at different 'Congestion Levels'. We can conclude that the time taken to find a parking slot, on average increases with an increase in the congestion level. This happens since the number of available slots decrease. There is a drastic increase in the total time at 40% congestion. This is because most of the parking slots in the databases is less than 40% congestion level. Thus there is a drastic increase in the total time taken by the user to reach the destination by for the last two levels of congestion.

6.1.5.9.3 Graph2: Query Time Vs Congestion Levels

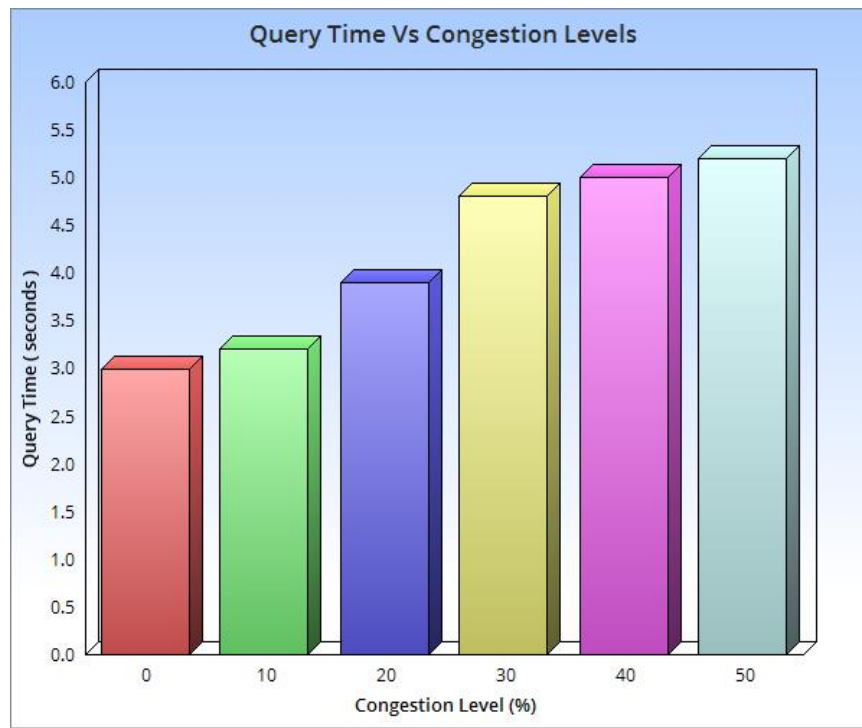


Figure 15 Plot between 'Query Time' and 'Congestion Level' for GPA

6.1.5.9.4 Graph Description:

The above graph displays a plot between 'Query Time' and 'Congestion Level'. Query time refers to the running time of the algorithm and this has been recorded for various congestion levels. For each congestion level, the time plotted on the graph is the average of running time of the algorithm over 45 different inputs.

6.1.5.9.5 Graph3: Gravitational Vs Greedy

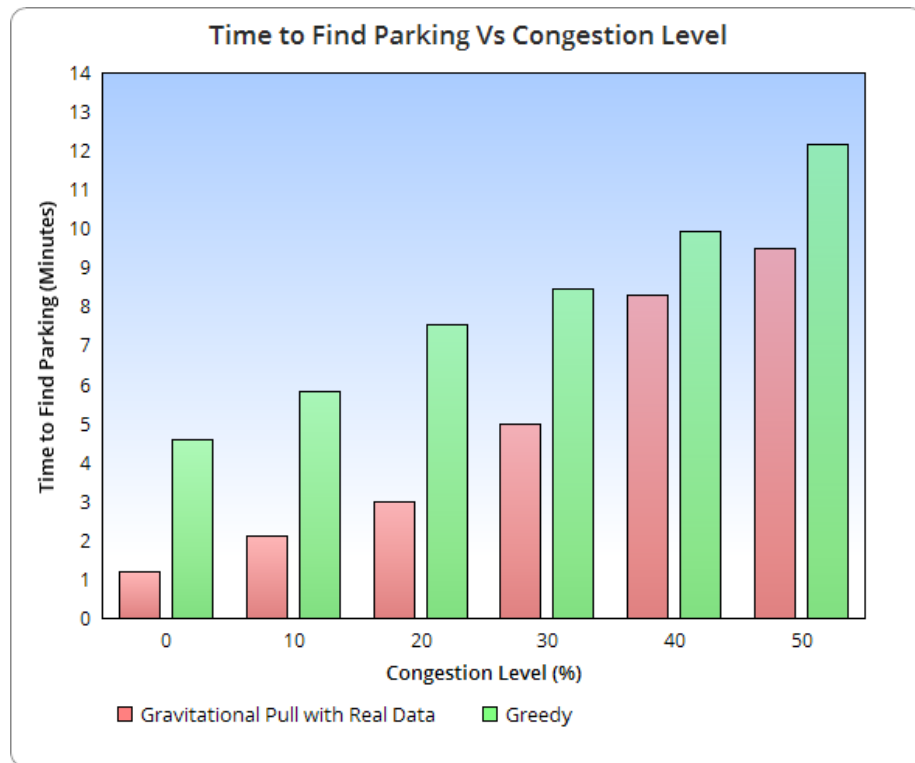


Figure 16 Graph Comparing GPA and Greedy Approach to find Parking using Real Time Data

6.1.5.9.6 Graph Description

As per the graph Greedy takes more time as compared to the Gravitational Pull algorithm since greedy does not consider the factors of distance and weight of the slot together. This results in poor results as compared to the gravitational pull algorithm.

6.2 Level of Information – Historical Data Regarding Parking is provided

6.2.1. Problem Description

Considering the data provided as historical data we need a method that efficiently predicts the availability for the parking slot for the current Time. Metrics to approximate street parking availability information at a block level based on prior data can help reduce the expenditure incurred. This algorithm should be able to make use of the available data from three months and generate a good approximation for the availability for the parking slot for any request time and date.

6.2.2. Holistic Architecture

This diagram describes the modifications to be made to the general system architecture to suit the needs of Task 2.

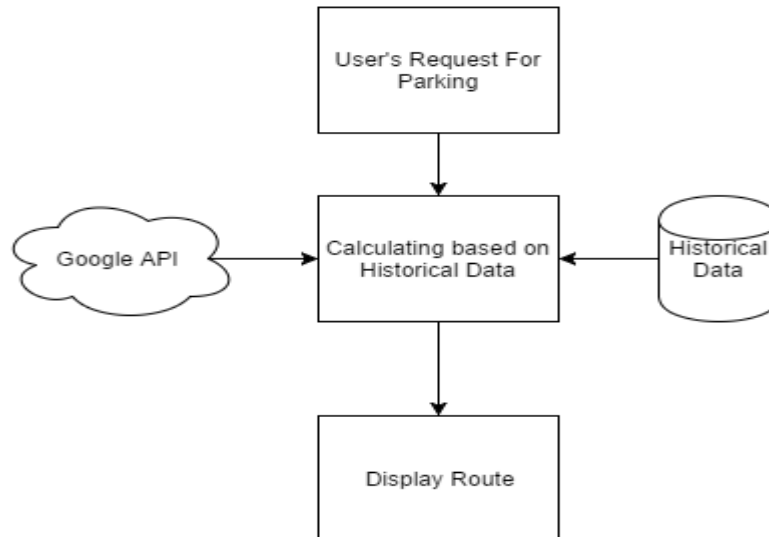


Figure 17 Holistic Architecture for Algorithm based on Historic Data

6.2.3. Pre-Analysis of Given Data

This step was performed to analyze as to what parts of the given data could be utilized for this task. For a random input, the algorithm should be able to determine the availability of the parking spot at that location for that particular time which should be accurate up to a minute. Since we do not have data for every single minute in the day, the time data can be clubbed together in order to determine the availability of the parking slot. The data also considers empty values and unavailable park spots. This should also be taken care of by the algorithm so as to prevent outliers while making decisions for availability.

6.2.4. Alternate Methods Considered

No.	Method Name	Method Description	Drawback
1	Average based Parking Allocation	The block containing parking slots will be assigned individual weights based on the average of the availability of available parking slots for that particular hour.	The outliers cause the algorithm to produce incorrect weights, thus leading to overestimation or underestimation of the availability of the parking slot depending on the type of outlier for that block.
2	Parking Slot Weight Estimation based on Combined Day and Hour Data	This method determines the weight of the parking slot but considering the mode of availability for that particular block	Since the data for all the days and hours will be combined, it results in a drop in the overall accuracy of the algorithm. This is because the availability of some of the parking slots is highly affected by the time and the day.

Table 3 Alternate Methods Considered for implementing Historical Data Based Algorithm

6.2.5. Method Chosen: Parking Slot Availability estimation based on Weighted Average

6.2.5.1. Pre-Cached Data

The weighted averages calculated for each of the parking blocks will be pre-calculated in order to improve efficiency during runtime. Pre-Cached Data is stored in the following format. The distances to the various blocks will be reused from Task 1.

No.	Name	Description
1	Block_Id	The ID used to distinguish blocks
2	Hours (1-24)	The days of the week will be stored in a single level JSON format under each of the hours

Table 4 Pre cached Data for Historic Data based algorithm

block_ID	Hour0	Hour1	Hour2	Hour3	Hour4	Hour5	Hour6	Hour7	Hour8	Hour9	Hour10
325031	5:3:2:3:2:1...	6:1:3:2:2:...	1:1:1:1:1:1...	2:1:1:1:1:1...	2:1:2:2:1:...	2:1:1:2:1:...	2:1:2:2:1:...	2:2:1:3:1:...	3:4:3:3:3:...	3:5:3:6:6:...	6:6:3:7:0
325032	5:4:3:4:4:2...	3:3:4:4:2:...	3:1:2:1:1:1...	1:1:1:2:1:...	1:1:2:1:1:...	1:0:2:2:1:...	1:1:1:0:1:...	1:4:4:1:2:...	2:6:3:7:6:...	3:5:3:6:6:...	3:6:3:0:6
325041	2:1:2:1:1:2...	2:2:1:1:2:...	2:2:1:2:1:...	3:1:2:1:1:...	1:1:0:1:0:...	1:2:0:1:0:...	1:1:0:1:0:...	3:6:0:7:0:...	7:6:5:8:5:...	6:11:4:6:...	5:9:5:5:6
325042	1:2:1:3:1:1...	2:1:1:1:1:...	1:1:2:0:1:...	1:0:1:1:2:...	0:0:1:0:1:...	0:0:1:0:1:...	0:0:0:0:0:...	1:0:2:0:1:...	3:4:2:7:2:...	4:10:2:7:...	4:8:5:9:5
325052	1:1:3:1:1:1...	1:1:1:1:2:...	2:0:1:0:1:...	2:0:1:0:1:...	1:0:2:0:0:...	1:0:0:0:0:...	0:0:0:0:0:...	2:0:0:0:0:...	1:2:1:2:1:...	1:1:1:1:2:...	1:2:1:4:1
326022	4:2:4:3:3:5...	4:4:3:3:1:...	6:2:3:3:2:...	4:1:1:1:1:...	2:0:1:0:0:...	2:0:2:0:1:...	2:0:2:0:1:...	2:0:3:0:3:...	3:4:3:4:4:...	3:6:2:4:5:...	5:4:4:6:6
326032	4:2:4:1:3:2...	6:2:4:2:4:...	4:1:4:3:2:...	2:1:2:2:1:...	2:0:0:0:0:...	0:0:0:0:0:...	1:0:2:0:1:...	3:0:4:0:4:...	3:7:4:6:4:...	4:6:5:6:6:...	3:5:2:5:5
326042	4:2:3:3:2:3...	3:1:3:3:2:...	2:1:2:2:1:...	2:1:1:1:1:...	2:0:0:0:1:...	1:0:1:0:2:...	0:0:1:0:1:...	2:0:3:0:4:...	2:6:4:5:4:...	2:7:3:7:5:...	3:5:3:5:5
326052	4:1:3:2:2:2...	2:1:3:2:3:...	1:1:1:1:2:...	2:0:1:1:1:...	3:0:1:0:2:...	1:0:2:0:1:...	1:0:1:0:1:...	1:0:2:0:1:...	2:1:1:2:2:...	2:1:3:5:3:...	2:5:5:3:5

Figure 18 Pre - Cached Data: Weighted Average based on Day and Time(per hour)

The weighted average will be pre-calculated using the following formulae. This will keep the outliers in check and will give the best approximation of the availability for the parking slot.

$$(\text{Availability of Parking Slot} * \text{Number of Occurrences of the availability of that parking slot}) / \text{Sum of availabilities}$$

6.2.5.2. Pre-Cached Data Generation Code

The code to generate this data is in the CreateCache.aspx and CreateCache.aspx.cs file.

6.2.5.2.1. Algorithm

1. The input taken by the algorithm will be the user's current location and the parking request time.
2. Once the user's current location is obtained the algorithm will fetch the weighted average from the precomputed values for that particular request time.
3. Based on the precomputed values for distance and weighted average the below formulae will be applied to the retrieved data.

$$(\text{Weighted Avg}) / (\text{distance})$$

4. Based on the weights assigned to the parking slots, the algorithm will fetch the path from the user's location to the slot picked by the algorithm using the Google Maps API.
5. The algorithm will obtain a set of points that are intersections which fall on that route.
6. As the user reaches the first intersection on the path the algorithm will re-compute the availabilities for that new location based on the new time.
7. The user will be then redirected to the new location.

8. The algorithm will run recursively till the user finds a parking slot.
9. It will terminate once the user reaches their destination.

6.2.5.3. Flow Chart

Below is the flow chart for the Algorithm implemented based on historical data

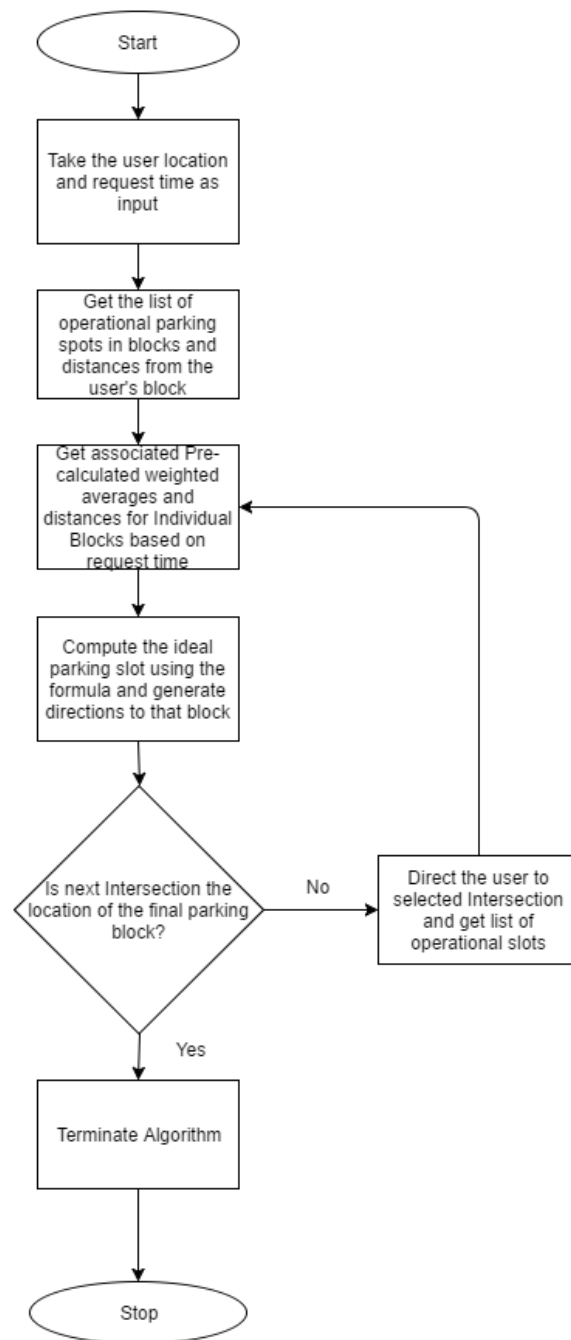


Figure 19 Flow Chart for algorithm implemented using Historic Data

6.2.5.4. Individual Experiments

The Experiments were conducted based on the following metrics

- Time taken for the algorithm to find a parking spot under various levels of congestion.
- Time taken for the user to find a parking spot under various levels of congestion

6.2.5.5. Experiment Inputs

The Algorithm is fed with 225 input values each varying in the following.

- Timestamp
- Start Location – Latitude and Longitude of the location

The inputs consist of 50% randomly generated data and 50% specifically varied inputs. This was done to make sure that most of the possible testing conditions are covered and the measure of accuracy and efficiency would represent the quality of the algorithm.

6.2.5.6. Experiment Methodology

The algorithm will be fed with the following Test Data.

Below is a part of the test data

Latitude	Longitude	TimeStamp
37.803571	-122.419974	4/10/2012 18:11
37.80553	-122.416246	4/10/2012 18:11
37.806948	-122.41474	4/10/2012 18:11
37.804267	-122.413609	4/10/2012 18:11
37.805241	-122.416902	4/10/2012 18:11
37.806149	-122.415531	4/10/2012 18:11
37.807926	-122.417453	4/10/2012 18:11
37.80518	-122.422308	4/10/2012 18:11
37.803571	-122.419974	4/6/2012 21:29
37.80553	-122.416246	4/6/2012 21:29
37.806948	-122.41474	4/6/2012 21:29
37.804267	-122.413609	4/6/2012 21:29
37.805241	-122.416902	4/6/2012 21:29
37.806149	-122.415531	4/6/2012 21:29
37.807926	-122.417453	4/6/2012 21:29
37.80518	-122.422308	4/6/2012 21:29
37.803571	-122.419974	4/6/2012 0:09
37.80553	-122.416246	4/6/2012 0:09
37.806948	-122.41474	4/6/2012 0:09
37.804267	-122.413609	4/6/2012 0:09

Figure 20 Screenshot of Test Data used for experiments

In order to measure the accuracy of the algorithm we will fetch the time taken by the user to reach his destination for various levels of congestion. Congestion is simulated by reducing the availability of the blocks by a particular level. The algorithm is tested for 10%, 20%, 30%, 40% and 50% Congestion levels. The Test Data is in the form of a CSV File (Test Data.csv).

The Test Data is in the following format.

SourceAddress	Latitude	Longitude	TimeStamp	CongestionPercent
---------------	----------	-----------	-----------	-------------------

This will be fed to the function calculating the route for the user.

6.2.5.7. Experiment Steps

1. The Test File should be in the format mentioned above.
2. On the Main Home Page “HistoricMainPage.aspx” choose the location of the Test File.
3. Click the “Show Results” button.
4. The Test Cases will be run for Various Levels of Congestion.

6.2.5.8. Experiment Results

Results of the experiments conducted for Historical Data based Search are displayed below.

- UserLocationLatitude/UserLocationLongitude – Initial start location of User
- ParkingSlotLatitude/ParkingSlotLongitude – Location of the allotted parking block
- StartTime – Time at which user starts searching for a parking
- Parking Time(Minutes) – Time taken by the user to find a parking block.
- WalkingDistance(Miles) – Distance taken by the user to walk to destination from the allotted parking block
- Walking Time(Minutes) – Time taken by the user to walk to destination from the allotted parking block

6.2.5.8.1 Congestion Level – 0%

Id	UserLocationLatitude	UserLocationLongitude	CongestionPercent	ParkingSlot Latitude	ParkingSlot Longitude	StartTime	Parking_Time	Walking_Time
1126	37.803571	-122.419974	0	37.806561	-122.420693	4/10/2012 18:11	4.68	2.86
1127	37.80553	-122.416246	0	37.806472	-122.413902	4/10/2012 18:11	3.66	2.27
1128	37.806948	-122.41474	0	37.805632	-122.417729	4/10/2012 18:11	4.38	2.12
1129	37.804267	-122.413609	0	37.806567	-122.420692	4/10/2012 18:11	3.55	2.27
1130	37.805241	-122.416902	0	37.807308	-122.415832	4/10/2012 18:11	3.19	2.47
1131	37.806149	-122.415531	0	37.806567	-122.420692	4/10/2012 18:11	3.99	2.48
1132	37.807926	-122.417453	0	37.806261	-122.415555	4/10/2012 18:11	4.38	2.28
1133	37.80518	-122.422308	0	37.807239	-122.417226	4/10/2012 18:11	4.63	2.34
1134	37.803571	-122.419974	0	37.805632	-122.417729	4/6/2012 21:29	5.02	2.23
1135	37.80553	-122.416246	0	37.806562	-122.414267	4/6/2012 21:29	3.77	2.89
1136	37.806948	-122.41474	0	37.806567	-122.420692	4/6/2012 21:29	3.38	3.09
1137	37.804267	-122.413609	0	37.805673	-122.41375	4/6/2012 21:29	3.28	2.21
1138	37.805241	-122.416902	0	37.805632	-122.417729	4/6/2012 21:29	3.52	2.61
1139	37.806149	-122.415531	0	37.806472	-122.413902	4/6/2012 21:29	3.46	3.04
1140	37.807926	-122.417453	0	37.807508	-122.414193	4/6/2012 21:29	4.61	2.65
1141	37.80518	-122.422308	0	37.806386	-122.41565	4/6/2012 21:29	3.69	2.66
1142	37.803571	-122.419974	0	37.806261	-122.415555	4/6/2012 0:09	4.79	2.23
1143	37.80553	-122.416246	0	37.806267	-122.41717	4/6/2012 0:09	3.12	2.26
1144	37.806948	-122.41474	0	37.805546	-122.422147	4/6/2012 0:09	4.43	2.94
1145	37.804267	-122.413609	0	37.806261	-122.415555	4/6/2012 0:09	4.7	2.61
1146	37.805241	-122.416902	0	37.805546	-122.422147	4/6/2012 0:09	3.94	2.84
1147	37.806149	-122.415531	0	37.805546	-122.422147	4/6/2012 0:09	3.57	2.99
1148	37.807926	-122.417453	0	37.806809	-122.412292	4/6/2012 0:09	3.84	2.73

Figure 21 Test Results for Historic Data Based Algorithm with 0% Congestion

6.2.5.8.2 Congestion Level – 10%

37.803571	-122.419974	10	37.805546	-122.422147	4/10/2012 18:11	4.28	3.54
37.80553	-122.416246	10	37.805546	-122.422147	4/10/2012 18:11	5.4	3.95
37.806948	-122.41474	10	37.807308	-122.415832	4/10/2012 18:11	4.25	3.36
37.804267	-122.413609	10	37.807508	-122.414193	4/10/2012 18:11	4.56	3.98
37.805241	-122.416902	10	37.806472	-122.413902	4/10/2012 18:11	5.11	3.56
37.806149	-122.415531	10	37.805534	-122.415332	4/10/2012 18:11	5.18	4.07
37.807926	-122.417453	10	37.805534	-122.415332	4/10/2012 18:11	5.24	3.31
37.80518	-122.422308	10	37.806386	-122.41565	4/10/2012 18:11	4.62	3.4
37.803571	-122.419974	10	37.805673	-122.41375	4/6/2012 21:29	5.11	3.73
37.80553	-122.416246	10	37.807508	-122.414193	4/6/2012 21:29	5.53	3.29
37.806948	-122.41474	10	37.805673	-122.41375	4/6/2012 21:29	5.6	3.76
37.804267	-122.413609	10	37.806261	-122.415555	4/6/2012 21:29	4.53	3.88
37.805241	-122.416902	10	37.805546	-122.422147	4/6/2012 21:29	5.02	4.01
37.806149	-122.415531	10	37.805546	-122.422147	4/6/2012 21:29	5.4	3.5
37.807926	-122.417453	10	37.806567	-122.420692	4/6/2012 21:29	4.66	3.32
37.80518	-122.422308	10	37.805546	-122.422147	4/6/2012 21:29	4.99	3.59
37.803571	-122.419974	10	37.806649	-122.412485	4/6/2012 0:09	4.32	3.36
37.80553	-122.416246	10	37.806261	-122.415555	4/6/2012 0:09	5.42	3.68
37.806948	-122.41474	10	37.805534	-122.415332	4/6/2012 0:09	4.52	3.13
37.804267	-122.413609	10	37.807508	-122.414193	4/6/2012 0:09	4.51	3.58
37.805241	-122.416902	10	37.806261	-122.415555	4/6/2012 0:09	4.87	3.57
37.806149	-122.415531	10	37.805546	-122.422147	4/6/2012 0:09	5.6	3.14
37.807926	-122.417453	10	37.807239	-122.417226	4/6/2012 0:09	4.65	3.81
37.80518	-122.422308	10	37.806561	-122.420693	4/6/2012 0:09	4.27	3.9

Figure 22 Test Results for Historic Data Based Algorithm with 10% Congestion

6.2.5.8.3 Congestion Level – 20%

37.803571	-122.419974	20	37.806386	-122.41565	4/10/2012 18:11	6.78	4.43
37.80553	-122.416246	20	37.80526	-122.417511	4/10/2012 18:11	5.44	4.6
37.806948	-122.41474	20	37.806567	-122.420692	4/10/2012 18:11	5.69	4.53
37.804267	-122.413609	20	37.806261	-122.415555	4/10/2012 18:11	5.46	4.12
37.805241	-122.416902	20	37.805546	-122.422147	4/10/2012 18:11	5.41	4.32
37.806149	-122.415531	20	37.806261	-122.415555	4/10/2012 18:11	5.11	4.96
37.807926	-122.417453	20	37.805632	-122.417729	4/10/2012 18:11	5.48	4.89
37.80518	-122.422308	20	37.805632	-122.417729	4/10/2012 18:11	6.82	4.12
37.803571	-122.419974	20	37.805546	-122.422147	4/6/2012 21:29	5.68	5.09
37.80553	-122.416246	20	37.807239	-122.417226	4/6/2012 21:29	6.76	4.49
37.806948	-122.41474	20	37.806261	-122.415555	4/6/2012 21:29	5.33	4.9
37.804267	-122.413609	20	37.805546	-122.422147	4/6/2012 21:29	5.9	4.13
37.805241	-122.416902	20	37.807308	-122.415832	4/6/2012 21:29	5.31	4.17
37.806149	-122.415531	20	37.805632	-122.417729	4/6/2012 21:29	6.52	4.83
37.807926	-122.417453	20	37.805673	-122.41375	4/6/2012 21:29	6.75	4.14
37.80518	-122.422308	20	37.805546	-122.422147	4/6/2012 21:29	5.58	4.14
37.803571	-122.419974	20	37.806359	-122.422317	4/6/2012 0:09	6.05	4.32
37.80553	-122.416246	20	37.805632	-122.417729	4/6/2012 0:09	6.89	5.06
37.806948	-122.41474	20	37.807508	-122.414193	4/6/2012 0:09	6.74	4.28
37.804267	-122.413609	20	37.805546	-122.422147	4/6/2012 0:09	6.23	4.87
37.805241	-122.416902	20	37.805546	-122.422147	4/6/2012 0:09	5.63	4.24
37.806149	-122.415531	20	37.805878	-122.412144	4/6/2012 0:09	5.86	4.96
37.807926	-122.417453	20	37.807308	-122.415832	4/6/2012 0:09	5.9	4.5
37.80518	-122.422308	20	37.805546	-122.422147	4/6/2012 0:09	6.35	4.12

Figure 23 Test Results for Historic Data Based Algorithm with 20% Congestion

6.2.5.8.4 Congestion Level – 30%

37.803571	-122.419974	30	37.805632	-122.417729	4/10/2012 18:11	7.28	5.31
37.80553	-122.416246	30	37.806567	-122.420692	4/10/2012 18:11	7.49	5.24
37.806948	-122.41474	30	37.805673	-122.41375	4/10/2012 18:11	7.12	5.69
37.804267	-122.413609	30	37.805534	-122.415332	4/10/2012 18:11	6.36	5.25
37.805241	-122.416902	30	37.806359	-122.422317	4/10/2012 18:11	7.71	6.09
37.806149	-122.415531	30	37.806261	-122.415555	4/10/2012 18:11	7.04	5.86
37.807926	-122.417453	30	37.806261	-122.415555	4/10/2012 18:11	7.72	5.47
37.80518	-122.422308	30	37.806567	-122.420692	4/10/2012 18:11	7.01	5.83
37.803571	-122.419974	30	37.805007	-122.418568	4/6/2012 21:29	6.26	5.45
37.80553	-122.416246	30	37.807508	-122.414193	4/6/2012 21:29	7.98	5.69
37.806948	-122.41474	30	37.805632	-122.417729	4/6/2012 21:29	7.05	6.03
37.804267	-122.413609	30	37.805546	-122.422147	4/6/2012 21:29	7.26	5.38
37.805241	-122.416902	30	37.805546	-122.422147	4/6/2012 21:29	7.6	5.34
37.806149	-122.415531	30	37.80526	-122.417511	4/6/2012 21:29	7.65	5.15
37.807926	-122.417453	30	37.807308	-122.415832	4/6/2012 21:29	6.84	5.96
37.80518	-122.422308	30	37.806649	-122.412485	4/6/2012 21:29	6.17	5.68
37.803571	-122.419974	30	37.806472	-122.413902	4/6/2012 0:09	7.78	5.27
37.80553	-122.416246	30	37.807239	-122.417226	4/6/2012 0:09	6.36	5.44
37.806948	-122.41474	30	37.805546	-122.422147	4/6/2012 0:09	6.96	5.44
37.804267	-122.413609	30	37.806809	-122.412292	4/6/2012 0:09	7.95	5.16
37.805241	-122.416902	30	37.805546	-122.422147	4/6/2012 0:09	6.39	5.91
37.806149	-122.415531	30	37.807239	-122.417226	4/6/2012 0:09	6.11	5.79
37.807926	-122.417453	30	37.807508	-122.414193	4/6/2012 0:09	7.15	5.19
37.80518	-122.422308	30	37.805546	-122.422147	4/6/2012 0:09	6.42	5.34

Figure 24 Test Results for Historic Data Based Algorithm with 30% Congestion

6.2.5.8.5 Congestion Level – 40%

37.803571	-122.419974	40	37.805632	-122.417729	4/10/2012 18:11	8.5	6.89
37.80553	-122.416246	40	37.805546	-122.422147	4/10/2012 18:11	7.9	6.31
37.806948	-122.41474	40	37.806472	-122.413902	4/10/2012 18:11	9	6.84
37.804267	-122.413609	40	37.805546	-122.422147	4/10/2012 18:11	7.59	6.41
37.805241	-122.416902	40	37.805546	-122.422147	4/10/2012 18:11	7.97	6.75
37.806149	-122.415531	40	37.805632	-122.417729	4/10/2012 18:11	8.54	6.25
37.807926	-122.417453	40	37.806562	-122.414267	4/10/2012 18:11	7.97	6.37
37.80518	-122.422308	40	37.805534	-122.415332	4/10/2012 18:11	7.61	6.53
37.803571	-122.419974	40	37.806472	-122.413902	4/6/2012 21:29	8.29	6.73
37.80553	-122.416246	40	37.807508	-122.414193	4/6/2012 21:29	8.07	7.09
37.806948	-122.41474	40	37.806386	-122.41565	4/6/2012 21:29	9.03	7.09
37.804267	-122.413609	40	37.806261	-122.415555	4/6/2012 21:29	8.04	6.22
37.805241	-122.416902	40	37.805546	-122.422147	4/6/2012 21:29	9.03	6.8
37.806149	-122.415531	40	37.805546	-122.422147	4/6/2012 21:29	8.28	6.4
37.807926	-122.417453	40	37.805546	-122.422147	4/6/2012 21:29	7.77	6.66
37.80518	-122.422308	40	37.806809	-122.412292	4/6/2012 21:29	8.58	6.44
37.803571	-122.419974	40	37.805673	-122.41375	4/6/2012 0:09	8.38	6.63
37.80553	-122.416246	40	37.805632	-122.417729	4/6/2012 0:09	9.01	6.52
37.806948	-122.41474	40	37.806562	-122.414267	4/6/2012 0:09	7.69	6.5
37.804267	-122.413609	40	37.80526	-122.417511	4/6/2012 0:09	7.14	6.54
37.805241	-122.416902	40	37.805546	-122.422147	4/6/2012 0:09	7.72	7.02
37.806149	-122.415531	40	37.805546	-122.422147	4/6/2012 0:09	8.68	6.43
37.807926	-122.417453	40	37.805546	-122.422147	4/6/2012 0:09	7.27	6.97
37.80518	-122.422308	40	37.807508	-122.414193	4/6/2012 0:09	7.39	6.82

Figure 25 Test Results for Historic Data Based Algorithm with 40% Congestion

6.2.5.8.6 Congestion Level – 50%

1	37.803571	-122.419974	50	37.806261	-122.415555	4/10/2012 18:11	8.99	7.78
2	37.80553	-122.416246	50	37.805546	-122.422147	4/10/2012 18:11	9.94	7.96
3	37.806948	-122.41474	50	37.805673	-122.41375	4/10/2012 18:11	8.43	8
4	37.804267	-122.413609	50	37.80526	-122.417511	4/10/2012 18:11	8.49	7.54
5	37.805241	-122.416902	50	37.805546	-122.422147	4/10/2012 18:11	8.27	7.51
6	37.806149	-122.415531	50	37.807239	-122.417226	4/10/2012 18:11	8.47	7.14
7	37.807926	-122.417453	50	37.806649	-122.412485	4/10/2012 18:11	8.21	7.95
8	37.80518	-122.422308	50	37.806261	-122.415555	4/10/2012 18:11	9.8	7.25
9	37.803571	-122.419974	50	37.805632	-122.417729	4/6/2012 21:29	8.86	8.09
10	37.80553	-122.416246	50	37.806267	-122.41717	4/6/2012 21:29	9.29	7.29
11	37.806948	-122.41474	50	37.806567	-122.420692	4/6/2012 21:29	8.76	7.23
12	37.804267	-122.413609	50	37.805632	-122.417729	4/6/2012 21:29	9.4	7.46
13	37.805241	-122.416902	50	37.805878	-122.412144	4/6/2012 21:29	9.31	7.97
14	37.806149	-122.415531	50	37.805632	-122.417729	4/6/2012 21:29	9.41	7.73
15	37.807926	-122.417453	50	37.806562	-122.414267	4/6/2012 21:29	9.86	7.48
16	37.80518	-122.422308	50	37.80526	-122.417511	4/6/2012 21:29	9.17	7.99
17	37.803571	-122.419974	50	37.805546	-122.422147	4/6/2012 0:09	8.11	7.59
18	37.80553	-122.416246	50	37.805546	-122.422147	4/6/2012 0:09	8.48	7.9
19	37.806948	-122.41474	50	37.807239	-122.417226	4/6/2012 0:09	9.91	7.66
20	37.804267	-122.413609	50	37.806562	-122.414267	4/6/2012 0:09	8.86	7.82
21	37.805241	-122.416902	50	37.806359	-122.422317	4/6/2012 0:09	8.48	7.69
22	37.806149	-122.415531	50	37.805546	-122.422147	4/6/2012 0:09	8.93	7.25

Figure 26 Test Results for Historic Data Based Algorithm with 50% Congestion

6.2.5.9 Experiment Analysis

The following Graph is obtained for the above set of test cases

6.2.5.9.1 Graph1 – Time Taken by the Algorithm to Execute at Every Intersection

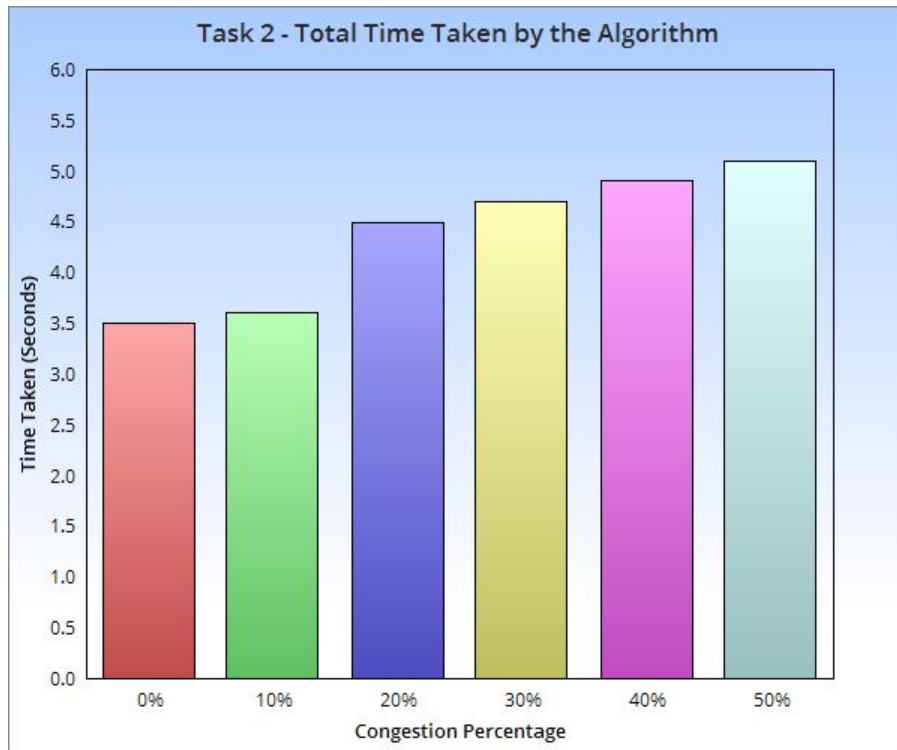


Figure 27 Graph between 'Time Taken by Algorithm' and 'Congestion Percentage'

6.2.5.9.2 Graph Description

The following graph shows the time taken by the algorithm to provide the output to the user (Only Algorithm Time). We have taken the above graph as the sum of the time taken by the algorithm at every intersection. This will give us the execution time of the algorithm under various levels of congestion. For 10% congestion the algorithm takes approximately 2.9 seconds.

Sum of the Time Taken by the algorithm at every intersection
--

This can be considered as the average running time for the algorithm. As the congestion increases the number of available parking slots reduce and thus the time taken by the algorithm to locate the parking slot increases. But as you notice this is not a drastic increase since the algorithm is able to retrieve the next highest weight parking slot as returned by the formula efficiently. Thus the overall time taken by the algorithm here will be determined by the time to retrieve the values from the database.

6.2.5.9.3 Graph2 – Time Taken by the Algorithm to Execute at Every Intersection (Excluding Database Hit Time)

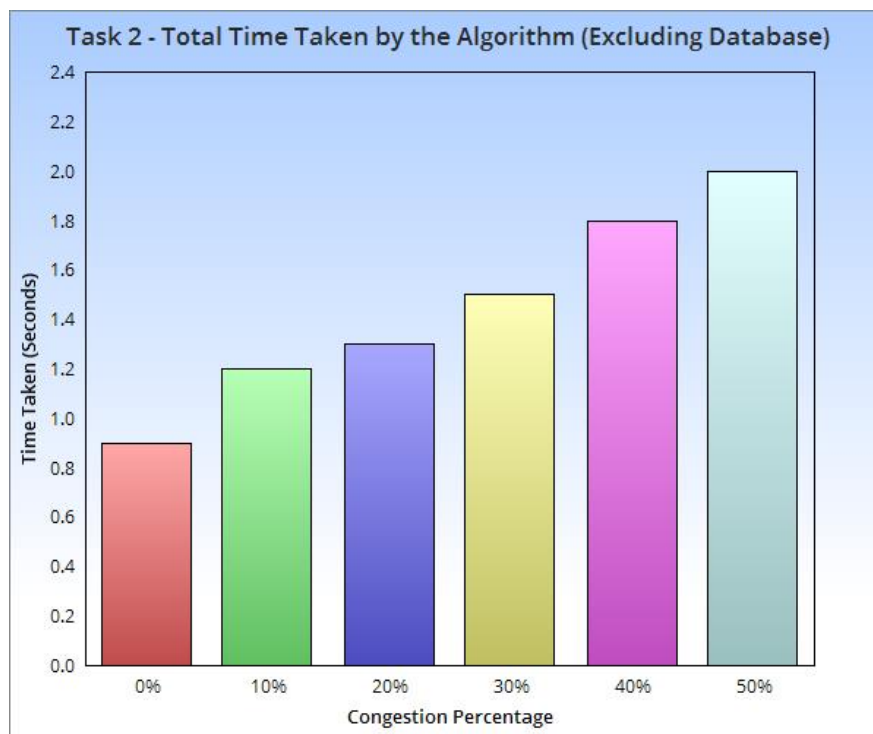


Figure 28 Graph between 'Total Time Taken by Algorithm' and 'Congestion Percentage'

6.2.5.9.4 Graph Description:

In order to make a more accurate description we have also created a graph with the time taken by the algorithm excluding the time taken by the algorithm to hit the database. The above graphs represent the time taken by the algorithm (excluding database hit time).

The time taken by the algorithm is almost constant. This is performed by first measuring the average time taken by the algorithm to retrieve the values from the database. Then the number of times the algorithm hits the database. This is then subtracted from the time taken by the algorithm which then gives us the above graph. This graph is an accurate measurement of the efficiency of the algorithm.

6.2.5.9.5 Graph3 – Total time taken by the user to reach the destination

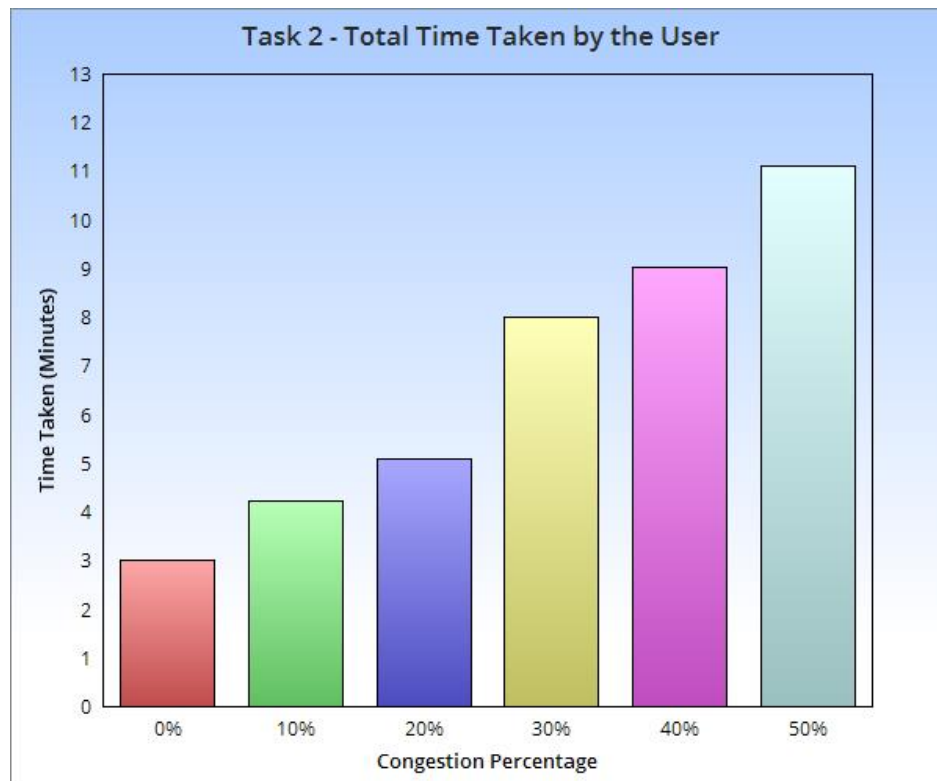


Figure 29 Graph between 'Time taken to Find Parking' and 'Congestion Percentage'

6.2.5.9.6 Graph Description:

In the graph when the congestion percent is zero the average time taken by the algorithm is 3.2 minutes as a whole which directs the user from the source to the destination. Here time increases logarithmically. Since more the congestion the further the parking slot will be from the user's location under normal circumstances.

6.2.5.9.7 Web Output

The following shows the output for the given request time.

The input to this is (37.800113, - 122.420867) and the time it was requested is 05/10/2016 at 02:58 PM

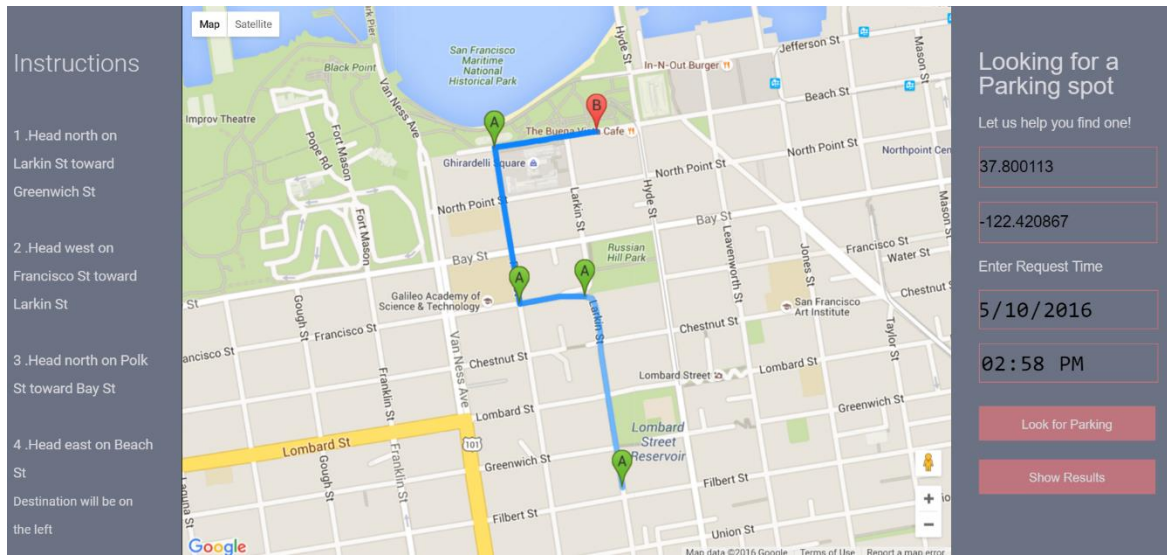


Figure 30 Output of the Parking Search on the Web application

6.3. Level of Information - Parking with no data about the parking slots availability.

6.3.1. Problem Description

When the users do not have any information or previous experience about the neighborhood, they usually take random paths that are closer to the user's intended destination. In this task, the real life scenario of Uninformed Search is being replicated. This algorithm will be the baseline algorithm for other algorithms to compare with.

6.3.2. Holistic Architecture

This diagram shows the modified version of the system architecture to accommodate the requirements of Algorithm 3

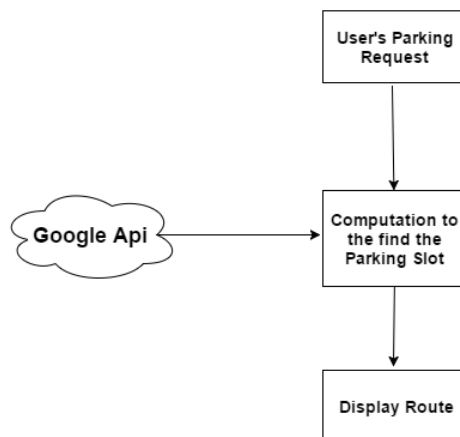


Figure 31 Holistic Architecture for Uninformed Search

6.3.3. Alternate Methods Considered

No.	Method Name	Method Description	Drawback
1	Parking based on nearest intersection, from source, sequentially.	The user is directed to the closest intersection from his original location. Then, if parking is not available in that block, he is directed to the intersection, which is next closest to his original location. This continues till he finds a parking slot, once the blocks are operational.	In this method, the user might be missing on some shortest paths, as he is directed to intersections closest to his original location, every time.
2	Parking based on nearest intersection, computed incrementally.	The user is directed to the closest intersection from his location, every time, till he finds a parking slot, once the blocks are operational.	In this method, the user may be directed to the slot, far away from his location where he intends to find a parking.

Table 5 Alternate Methods Considered for Uninformed Search

6.3.4. Method Chosen - Parking Slot Availability estimation based on nearest neighbor to the source location, in a region having pre-defined boundaries.

6.3.4.1. Pre-Cached Data

Data will not be pre-cached since the simulation is based on a user manually looking for a parking slot with no prior information.

6.3.4.2. Algorithm

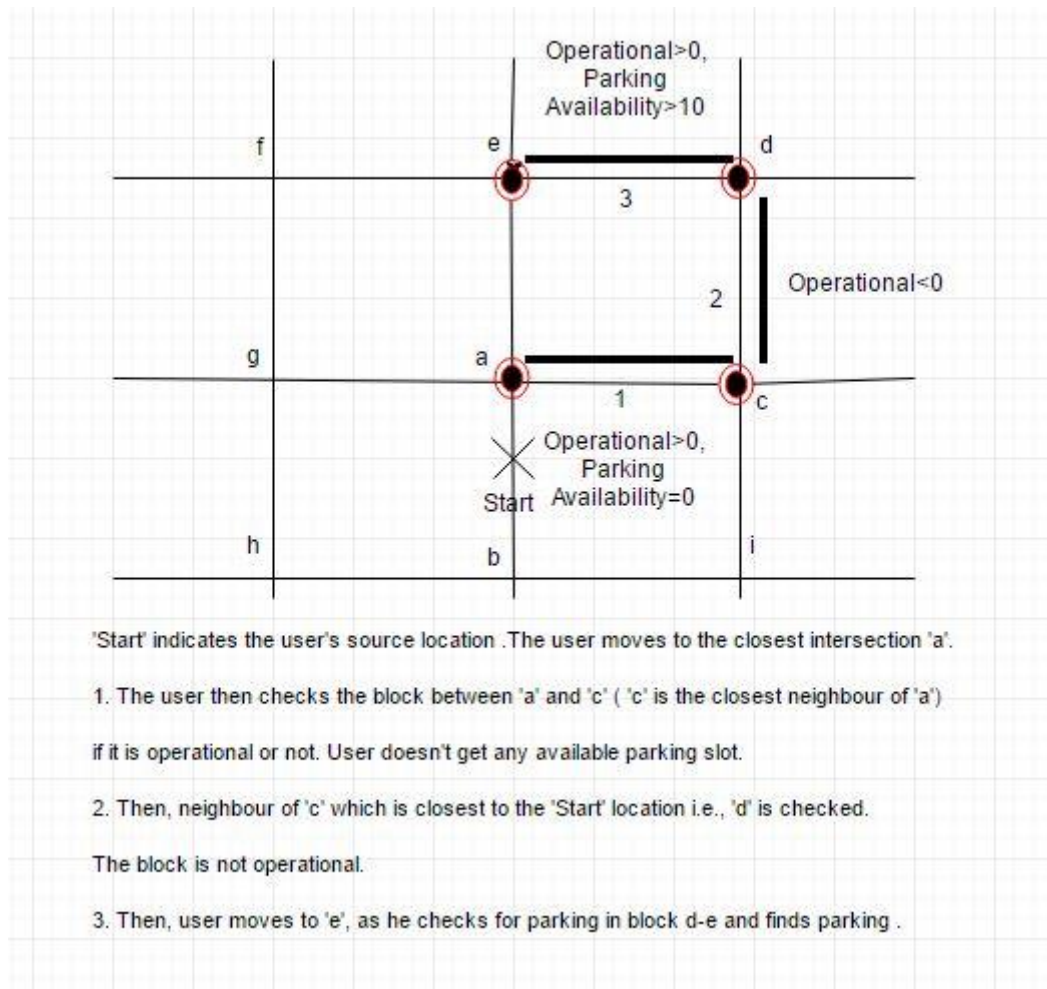


Figure 32 Simulation of Uninformed Search Algorithm

1. The input taken by the algorithm will be the user's current location and the time at which he wants to find the parking.
2. If the user is not at an intersection, he first moves to closest intersection.
3. A list containing all the intersections visited by the user is maintained.
4. Neighbors of this intersection are computed, and the user moves to any random neighboring intersection. A check is done, if this block is operational. If yes, check for parking slots. The algorithm exits, if the user finds a parking slot here. Else, continue to step 5.

5. Neighbors are computed from the user's current location (which is an intersection), and he moves to the intersection, closest to his starting point. This process, repeats until the user finds a parking.
6. List of visited intersections is cleared every time the user visits four intersections. This makes sure that the user is not diverted to the block, farther from his starting point (which is the usual real time scenario).

6.3.4.3. Flowchart

Below is the flow chart representing steps for Uninformed Search.

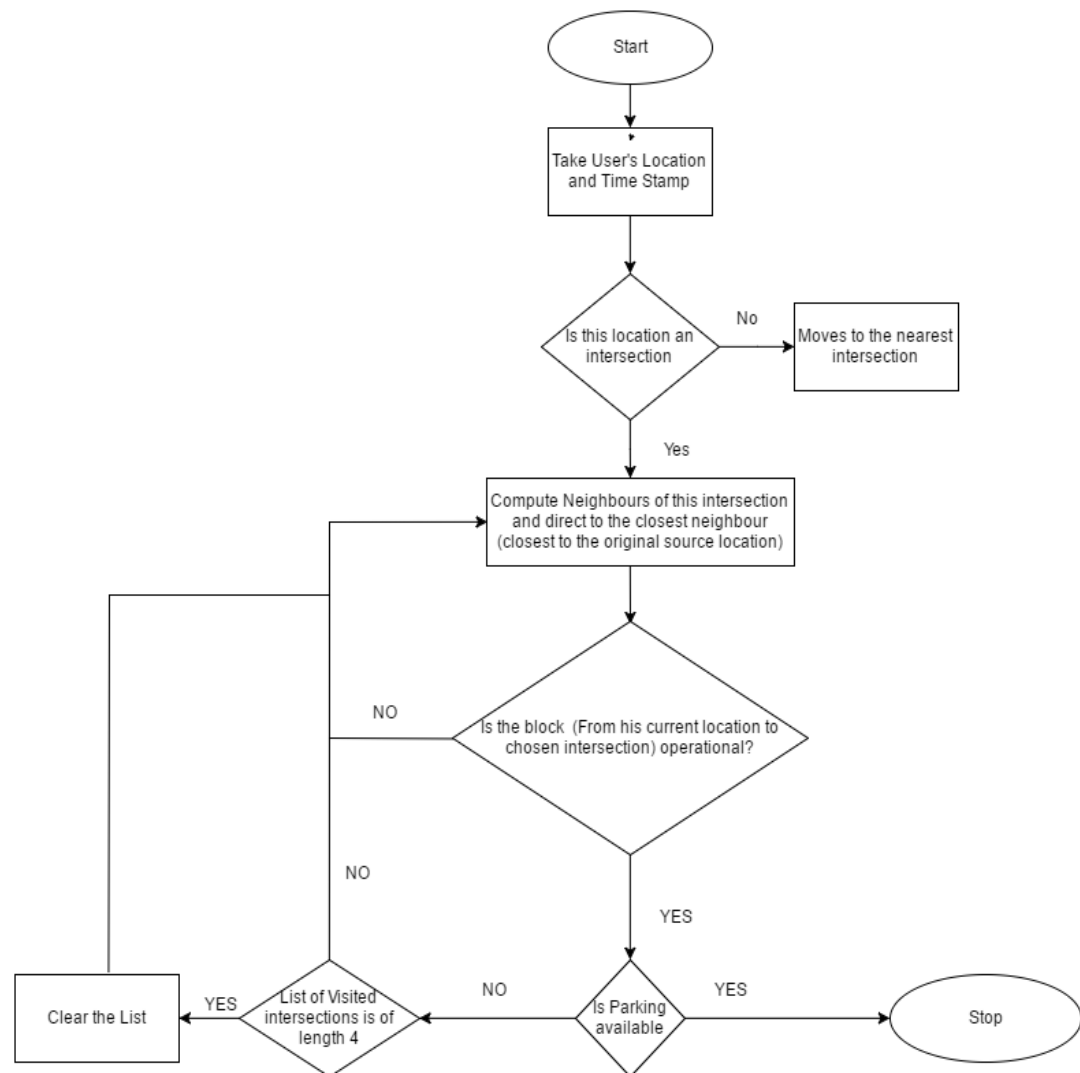


Figure 33 Flow Chart for the Uninformed Search Algorithm

6.3.4.4. Individual Experiments

The Experiments were conducted based on the following metrics

- Time taken for the algorithm to find a parking spot under various levels of congestion
- Time taken for the user to find a parking spot under various levels of congestion

6.3.4.5. Experiment Inputs

The test data is stored in a file named “TestData.xls”. The csv file contains the following data corresponding to 225 different source – timestamp combinations.

1. Latitude – latitude of the initial position of user
2. Longitude – longitude of the initial position of user
3. Timestamp – Time at which user is searching for parking

6.3.4.6. Experiment Methodology

In this case, experiments are conducted by adding congestion. Congestion is induced by reducing the number of available parking slots by a certain percentage of the maximum possible value of the parking slot availability as given in the database (i.e. 19). The experiment is conducted for various levels of congestion – 10%, 20%, 30%, 40% and 50%. The test data is stored in a file named “TestData.xls”.

The csv file contains the following data:

1. Latitude – latitude of the initial position of user
2. Longitude – longitude of the initial position of user
3. Timestamp – Time at which user is searching for parking
4. Congestion Percent – The congestion percentage that is taken into consideration.

6.3.4.7. Experiment Steps

- In order to find the results of the algorithm, a button named ‘Show Results’ is available. On click of this button, the data in the spreadsheet is taken as input and the algorithm is run by considering the inputs one at a time.
- The results of the test cases are stored in the database and displayed on the UI.

6.3.4.8. Experiment Results

- Results of the experiments conducted for Uninformed Search are displayed below.
- UserLocationLatitude/UserLocationLongitude – Initial start location of User
- ParkingSlotLatitude/ParkingSlotLongitude – Location of the allotted parking block
- StartTime – Time at which user starts searching for a parking
- Parking Time(Minutes) – Time taken by the user to find a parking block.

- WalkingDistance(Miles) – Distance taken by the user to walk to destination from the allotted parking block
- Walking Time(Minutes) – Time taken by the user to walk to destination from the allotted parking block

6.3.4.8.1 Congestion Level – 0%

TEST RESULTS WITHOUT CONGESTION

Id	UserLocationLatitude	UserLocationLongitude	ParkingSlotLatitude	ParkingSlotLongitude	StartTime	Parking_Time	WalkingDistance	WalkingTime
1	37.803571	-122.419974	37.806567	-122.420692	4/10/2012 6:11:00 PM	12	0.18	3.6
2	37.805530	-122.416246	37.805632	-122.417729	4/10/2012 6:11:00 PM	8	0.07	1.4
3	37.806948	-122.414740	37.807508	-122.414193	4/10/2012 6:11:00 PM	1	0.04	0.8
4	37.804267	-122.413609	37.806562	-122.414267	4/10/2012 6:11:00 PM	4	0.14	2.8
5	37.805241	-122.416902	37.806261	-122.415555	4/10/2012 6:11:00 PM	2	0.09	1.8
6	37.806149	-122.415531	37.805534	-122.415332	4/10/2012 6:11:00 PM	1	0.04	0.8
7	37.807926	-122.417453	37.807239	-122.417226	4/10/2012 6:11:00 PM	1	0.04	0.8
8	37.805180	-122.422308	37.805546	-122.422147	4/10/2012 6:11:00 PM	4	0.02	0.4
9	37.803571	-122.419974	37.806567	-122.420692	4/6/2012 9:29:00 PM	13	0.18	3.6
10	37.805530	-122.416246	37.805632	-122.417729	4/6/2012 9:29:00 PM	8	0.07	1.4
11	37.806948	-122.414740	37.807508	-122.414193	4/6/2012 9:29:00 PM	1	0.04	0.8
12	37.804267	-122.413609	37.806472	-122.413902	4/6/2012 9:29:00 PM	4	0.13	2.6
13	37.805241	-122.416902	37.806386	-122.415650	4/6/2012 9:29:00 PM	2	0.09	1.8
14	37.806149	-122.415531	37.805534	-122.415332	4/6/2012 9:29:00 PM	1	0.04	0.8
15	37.807926	-122.417453	37.807308	-122.415832	4/6/2012 9:29:00 PM	2	0.09	1.8
16	37.805180	-122.422308	37.805546	-122.422147	4/6/2012 9:29:00 PM	4	0.02	0.4
17	37.803571	-122.419974	37.806567	-122.420692	4/6/2012 12:09:00 AM	13	0.18	3.6
18	37.805530	-122.416246	37.805260	-122.417511	4/6/2012 12:09:00 AM	8	0.06	1.2
19	37.806948	-122.414740	37.807508	-122.414193	4/6/2012 12:09:00 AM	1	0.04	0.8
20	37.804267	-122.413609	37.806809	-122.412292	4/6/2012 12:09:00 AM	5	0.16	3.2
21	37.805241	-122.416902	37.806261	-122.415555	4/6/2012 12:09:00 AM	2	0.09	1.8
22	37.806149	-122.415531	37.806261	-122.415555	4/6/2012 12:09:00 AM	4	0.01	0.2
23	37.807926	-122.417453	37.807239	-122.417226	4/6/2012 12:09:00 AM	1	0.04	0.8
24	37.805180	-122.422308	37.805546	-122.422147	4/6/2012 12:09:00 AM	4	0.02	0.4
25	37.806439	-122.415887	37.806267	-122.417170	4/12/2012 4:25:00 PM	3	0.06	1.2
26	37.803483	-122.419959	37.805007	-122.418568	4/12/2012 4:25:00 PM	16	0.11	2.2
27	37.804335	-122.418101	37.805632	-122.417729	4/12/2012 4:25:00 PM	3	0.08	1.6
28	37.800535	-122.419187	37.805632	-122.417729	4/12/2012 4:25:00 PM	4	0.31	6.2
29	37.801083	-122.419503	37.805546	-122.422147	4/12/2012 4:25:00 PM	4	0.3	6
30	37.805355	-122.415780	37.806261	-122.415555	4/12/2012 4:25:00 PM	3	0.06	1.2
31	37.803050	-122.419078	37.805632	-122.417729	4/12/2012 4:25:00 PM	4	0.17	3.4
32	37.803199	-122.426009	37.806359	-122.422317	4/12/2012 4:25:00 PM	3	0.26	5.2
33	37.797003	-122.418323	37.805673	-122.413750	4/12/2012 4:25:00 PM	4	0.56	11.2

Figure 34 Test Results for Uninformed Search with 0% Congestion

6.3.4.8.2 Congestion Level – 10%

TEST RESULTS WITH CONGESTION

Id	UserLocationLatitude	UserLocationLongitude	ParkingSlotLatitude	ParkingSlotLongitude	CongestionPercent	StartTime	Parking_Time	WalkingDistance	WalkingTime
1	37.803571	-122.419974	37.806567	-122.420692	10	4/10/2012 6:11:00 PM	12	0.18	3.6
2	37.805530	-122.416246	37.805632	-122.417729	10	4/10/2012 6:11:00 PM	8	0.07	1.4
3	37.806948	-122.414740	37.808343	-122.414330	10	4/10/2012 6:11:00 PM	6	0.09	1.8
4	37.804267	-122.413609	37.806562	-122.414267	10	4/10/2012 6:11:00 PM	4	0.14	2.8
5	37.805241	-122.416902	37.806386	-122.415650	10	4/10/2012 6:11:00 PM	2	0.09	1.8
6	37.806149	-122.415531	37.806386	-122.415650	10	4/10/2012 6:11:00 PM	4	0.02	0.4
7	37.807926	-122.417453	37.807239	-122.417226	10	4/10/2012 6:11:00 PM	1	0.04	0.8
8	37.805180	-122.422308	37.805546	-122.422147	10	4/10/2012 6:11:00 PM	4	0.02	0.4
9	37.803571	-122.419974	37.804588	-122.421862	10	4/6/2012 9:29:00 PM	16	0.11	2.2
10	37.805530	-122.416246	37.805260	-122.417511	10	4/6/2012 9:29:00 PM	8	0.06	1.2
11	37.806948	-122.414740	37.806584	-122.413821	10	4/6/2012 9:29:00 PM	16	0.05	1
12	37.804267	-122.413609	37.805177	-122.413531	10	4/6/2012 9:29:00 PM	9	0.05	1
13	37.805241	-122.416902	37.805534	-122.415332	10	4/6/2012 9:29:00 PM	2	0.08	1.6
14	37.806149	-122.415531	37.805534	-122.415332	10	4/6/2012 9:29:00 PM	1	0.04	0.8
15	37.807926	-122.417453	37.807308	-122.415832	10	4/6/2012 9:29:00 PM	2	0.09	1.8
16	37.805180	-122.422308	37.804380	-122.423482	10	4/6/2012 9:29:00 PM	11	0.07	1.4
17	37.803571	-122.419974	37.806567	-122.420692	10	4/6/2012 12:09:00 AM	12	0.18	3.6
18	37.805530	-122.416246	37.805260	-122.417511	10	4/6/2012 12:09:00 AM	8	0.06	1.2
19	37.806948	-122.414740	37.807508	-122.414193	10	4/6/2012 12:09:00 AM	1	0.04	0.8
20	37.804267	-122.413609	37.806809	-122.412292	10	4/6/2012 12:09:00 AM	5	0.16	3.2
21	37.805241	-122.416902	37.806261	-122.415555	10	4/6/2012 12:09:00 AM	2	0.09	1.8
22	37.806149	-122.415531	37.806261	-122.415555	10	4/6/2012 12:09:00 AM	4	0.01	0.2
23	37.807926	-122.417453	37.807308	-122.415832	10	4/6/2012 12:09:00 AM	2	0.09	1.8
24	37.805180	-122.422308	37.805546	-122.422147	10	4/6/2012 12:09:00 AM	4	0.02	0.4
25	37.806439	-122.415887	37.806386	-122.415650	10	4/12/2012 4:25:00 PM	8	0.01	0.2
26	37.803483	-122.419959	37.805007	-122.418568	10	4/12/2012 4:25:00 PM	16	0.11	2.2
27	37.804335	-122.418101	37.805632	-122.417729	10	4/12/2012 4:25:00 PM	3	0.08	1.6
28	37.800535	-122.419187	37.805632	-122.417729	10	4/12/2012 4:25:00 PM	4	0.31	6.2
29	37.801083	-122.419503	37.805546	-122.422147	10	4/12/2012 4:25:00 PM	4	0.3	6
30	37.805355	-122.415780	37.806386	-122.415650	10	4/12/2012 4:25:00 PM	3	0.06	1.2

Figure 35 Test Results for Uninformed Search with 10% Congestion

6.3.4.8.3 Congestion Level – 20%

46	37.803571	-122.419974	37.804588	-122.421862	20	4/10/2012 6:11:00 PM	16	0.11	2.2
47	37.805530	-122.416246	37.805632	-122.417729	20	4/10/2012 6:11:00 PM	8	0.07	1.4
48	37.806948	-122.414740	37.805754	-122.413555	20	4/10/2012 6:11:00 PM	11	0.09	1.8
49	37.804267	-122.413609	37.806472	-122.413902	20	4/10/2012 6:11:00 PM	4	0.13	2.6
50	37.805241	-122.416902	37.806386	-122.415650	20	4/10/2012 6:11:00 PM	2	0.09	1.8
51	37.806149	-122.415531	37.806386	-122.415650	20	4/10/2012 6:11:00 PM	4	0.02	0.4
52	37.807926	-122.417453	37.807308	-122.415832	20	4/10/2012 6:11:00 PM	10	0.09	1.8
53	37.805180	-122.422308	37.805546	-122.422147	20	4/10/2012 6:11:00 PM	4	0.02	0.4
54	37.803571	-122.419974	37.804588	-122.421862	20	4/6/2012 9:29:00 PM	16	0.11	2.2
55	37.805530	-122.416246	37.805466	-122.415455	20	4/6/2012 9:29:00 PM	9	0.04	0.8
56	37.806948	-122.414740	37.806584	-122.413821	20	4/6/2012 9:29:00 PM	16	0.05	1
57	37.804267	-122.413609	37.805177	-122.413531	20	4/6/2012 9:29:00 PM	9	0.05	1
58	37.805241	-122.416902	37.805534	-122.415332	20	4/6/2012 9:29:00 PM	10	0.08	1.6
59	37.806149	-122.415531	37.805534	-122.415332	20	4/6/2012 9:29:00 PM	1	0.04	0.8
60	37.807926	-122.417453	37.807101	-122.417292	20	4/6/2012 9:29:00 PM	15	0.05	1
61	37.805180	-122.422308	37.804380	-122.423482	20	4/6/2012 9:29:00 PM	11	0.07	1.4
62	37.803571	-122.419974	37.806567	-122.420692	20	4/6/2012 12:09:00 AM	12	0.18	3.6
63	37.805530	-122.416246	37.807308	-122.415832	20	4/6/2012 12:09:00 AM	13	0.11	2.2
64	37.806948	-122.414740	37.807508	-122.414193	20	4/6/2012 12:09:00 AM	1	0.04	0.8
65	37.804267	-122.413609	37.806809	-122.412292	20	4/6/2012 12:09:00 AM	5	0.16	3.2
66	37.805241	-122.416902	37.806261	-122.415555	20	4/6/2012 12:09:00 AM	10	0.09	1.8
67	37.806149	-122.415531	37.806267	-122.417170	20	4/6/2012 12:09:00 AM	7	0.08	1.6
68	37.807926	-122.417453	37.807308	-122.415832	20	4/6/2012 12:09:00 AM	2	0.09	1.8
69	37.805180	-122.422308	37.805546	-122.422147	20	4/6/2012 12:09:00 AM	4	0.02	0.4
70	37.806439	-122.415887	37.807101	-122.417292	20	4/12/2012 4:25:00 PM	16	0.08	1.6
71	37.803483	-122.419959	37.805007	-122.418568	20	4/12/2012 4:25:00 PM	16	0.11	2.2
72	37.804335	-122.418101	37.805632	-122.417729	20	4/12/2012 4:25:00 PM	3	0.08	1.6
73	37.805535	-122.419187	37.805632	-122.417729	20	4/12/2012 4:25:00 PM	4	0.31	6.2
74	37.801083	-122.419503	37.805546	-122.422147	20	4/12/2012 4:25:00 PM	4	0.3	6
75	37.805355	-122.415780	37.805528	-122.413732	20	4/12/2012 4:25:00 PM	5	0.1	2
76	37.803050	-122.419078	37.805632	-122.417729	20	4/12/2012 4:25:00 PM	4	0.17	3.4
77	37.803199	-122.426009	37.805520	-122.422044	20	4/12/2012 4:25:00 PM	16	0.23	4.6
78	37.797003	-122.418323	37.805528	-122.413732	20	4/12/2012 4:25:00 PM	4	0.56	11.2
79	37.806162	-122.417074	37.805232	-122.416911	20	4/12/2012 4:25:00 PM	16	0.06	1.2
80	37.804408	-122.422157	37.805546	-122.422147	20	4/16/2012 10:06:00 AM	3	0.07	1.4
81	37.800815	-122.417428	37.805673	-122.413750	20	4/16/2012 10:06:00 AM	3	0.34	6.8

Figure 36 Test Results for Uninformed Search with 20% Congestion

6.3.4.8.4 Congestion Level – 30%

91	37.803571	-122.419974	37.804588	-122.421862	30	4/10/2012 6:11:00 PM	16	0.11	2.2
92	37.805530	-122.416246	37.805260	-122.417511	30	4/10/2012 6:11:00 PM	8	0.06	1.2
93	37.806948	-122.414740	37.806584	-122.413821	30	4/10/2012 6:11:00 PM	16	0.05	1
94	37.804267	-122.413609	37.806472	-122.413902	30	4/10/2012 6:11:00 PM	4	0.13	2.6
95	37.805241	-122.416902	37.806386	-122.415650	30	4/10/2012 6:11:00 PM	2	0.09	1.8
96	37.806149	-122.415531	37.806386	-122.415650	30	4/10/2012 6:11:00 PM	4	0.02	0.4
97	37.807926	-122.417453	37.807101	-122.417292	30	4/10/2012 6:11:00 PM	15	0.05	1
98	37.805180	-122.422308	37.804380	-122.423482	30	4/10/2012 6:11:00 PM	11	0.07	1.4
99	37.803571	-122.419974	37.804588	-122.421862	30	4/6/2012 9:29:00 PM	15	0.11	2.2
100	37.805530	-122.416246	37.805466	-122.415455	30	4/6/2012 9:29:00 PM	9	0.04	0.8
101	37.806948	-122.414740	37.806584	-122.413821	30	4/6/2012 9:29:00 PM	16	0.05	1
102	37.804267	-122.413609	37.805177	-122.413531	30	4/6/2012 9:29:00 PM	9	0.05	1
103	37.805241	-122.416902	37.805444	-122.415275	30	4/6/2012 9:29:00 PM	14	0.08	1.6
104	37.806149	-122.415531	37.805466	-122.415455	30	4/6/2012 9:29:00 PM	15	0.04	0.8
105	37.807926	-122.417453	37.807101	-122.417292	30	4/6/2012 9:29:00 PM	15	0.05	1
106	37.805180	-122.422308	37.804380	-122.423482	30	4/6/2012 9:29:00 PM	11	0.07	1.4
107	37.803571	-122.419974	37.806693	-122.420814	30	4/6/2012 12:09:00 AM	12	0.19	3.8
108	37.805530	-122.416246	37.807308	-122.415832	30	4/6/2012 12:09:00 AM	13	0.11	2.2
109	37.806948	-122.414740	37.805727	-122.413684	30	4/6/2012 12:09:00 AM	11	0.09	1.8
110	37.804267	-122.413609	37.806809	-122.412292	30	4/6/2012 12:09:00 AM	5	0.16	3.2
111	37.805241	-122.416902	37.805528	-122.413732	30	4/6/2012 12:09:00 AM	12	0.15	3
112	37.806149	-122.415531	37.806267	-122.417170	30	4/6/2012 12:09:00 AM	7	0.08	1.6
113	37.807926	-122.417453	37.807308	-122.415832	30	4/6/2012 12:09:00 AM	2	0.09	1.8
114	37.805180	-122.422308	37.805546	-122.422147	30	4/6/2012 12:09:00 AM	4	0.02	0.4
115	37.806439	-122.415887	37.807101	-122.417292	30	4/12/2012 4:25:00 PM	16	0.08	1.6
116	37.803483	-122.419959	37.805007	-122.418568	30	4/12/2012 4:25:00 PM	16	0.11	2.2
117	37.804335	-122.418101	37.805007	-122.418568	30	4/12/2012 4:25:00 PM	16	0.05	1
118	37.805535	-122.419187	37.805546	-122.422147	30	4/12/2012 4:25:00 PM	14	0.33	6.6
119	37.801083	-122.419503	37.805546	-122.422147	30	4/12/2012 4:25:00 PM	10	0.3	6
120	37.805355	-122.415780	37.805528	-122.413732	30	4/12/2012 4:25:00 PM	5	0.1	2
121	37.803050	-122.419078	37.805007	-122.418568	30	4/12/2012 4:25:00 PM	16	0.12	2.4
122	37.803199	-122.426009	37.805520	-122.422044	30	4/12/2012 4:25:00 PM	16	0.23	4.6
123	37.797003	-122.418323	37.805528	-122.413732	30	4/12/2012 4:25:00 PM	4	0.56	11.2
124	37.806162	-122.417074	37.805232	-122.416911	30	4/12/2012 4:25:00 PM	16	0.06	1.2
125	37.804408	-122.422157	37.805546	-122.422147	30	4/16/2012 10:06:00 AM	3	0.07	1.4
126	37.800815	-122.417428	37.805673	-122.413750	30	4/16/2012 10:06:00 AM	3	0.34	6.8

Figure 37 Test Results for Uninformed Search with 30% Congestion

6.3.4.8.5 Congestion Level – 40%

136	37.803571	-122.419974	37.804588	-122.421862	40	4/10/2012 6:11:00 PM	16	0.11	2.2
137	37.805530	-122.416246	37.805232	-122.416911	40	4/10/2012 6:11:00 PM	16	0.04	0.8
138	37.806948	-122.414740	37.806584	-122.413821	40	4/10/2012 6:11:00 PM	16	0.05	1
139	37.804267	-122.413609	37.806472	-122.413902	40	4/10/2012 6:11:00 PM	4	0.13	2.6
140	37.805241	-122.416902	37.806386	-122.415650	40	4/10/2012 6:11:00 PM	2	0.09	1.8
141	37.806149	-122.415531	37.806386	-122.415650	40	4/10/2012 6:11:00 PM	4	0.02	0.4
142	37.807926	-122.417453	37.807101	-122.417292	40	4/10/2012 6:11:00 PM	15	0.05	1
143	37.805180	-122.422308	37.804380	-122.423482	40	4/10/2012 6:11:00 PM	11	0.07	1.4
144	37.803571	-122.419974	37.804588	-122.421862	40	4/6/2012 9:29:00 PM	15	0.11	2.2
145	37.805530	-122.416246	37.805232	-122.416911	40	4/6/2012 9:29:00 PM	16	0.04	0.8
146	37.806948	-122.414740	37.806584	-122.413821	40	4/6/2012 9:29:00 PM	16	0.05	1
147	37.804267	-122.413609	37.805177	-122.413531	40	4/6/2012 9:29:00 PM	9	0.05	1
148	37.805241	-122.416902	37.805444	-122.415275	40	4/6/2012 9:29:00 PM	14	0.08	1.6
149	37.806149	-122.415531	37.806381	-122.415468	40	4/6/2012 9:29:00 PM	16	0.01	0.2
150	37.807926	-122.417453	37.807101	-122.417292	40	4/6/2012 9:29:00 PM	15	0.05	1
151	37.805180	-122.422308	37.804380	-122.423482	40	4/6/2012 9:29:00 PM	11	0.07	1.4
152	37.803571	-122.419974	37.806693	-122.420814	40	4/6/2012 12:09:00 AM	12	0.19	3.8
153	37.805530	-122.416246	37.807308	-122.415832	40	4/6/2012 12:09:00 AM	13	0.11	2.2
154	37.806948	-122.414740	37.806584	-122.413821	40	4/6/2012 12:09:00 AM	16	0.05	1
155	37.804267	-122.413609	37.806809	-122.412292	40	4/6/2012 12:09:00 AM	5	0.16	3.2
156	37.805241	-122.416902	37.805528	-122.413732	40	4/6/2012 12:09:00 AM	12	0.15	3
157	37.806149	-122.415531	37.806381	-122.415468	40	4/6/2012 12:09:00 AM	16	0.01	0.2
158	37.807926	-122.417453	37.807308	-122.415832	40	4/6/2012 12:09:00 AM	2	0.09	1.8
159	37.805180	-122.422308	37.805546	-122.422147	40	4/6/2012 12:09:00 AM	4	0.02	0.4
160	37.806439	-122.415887	37.807101	-122.417292	40	4/12/2012 4:25:00 PM	16	0.08	1.6
161	37.803483	-122.419959	37.805007	-122.418568	40	4/12/2012 4:25:00 PM	16	0.11	2.2
162	37.804335	-122.418101	37.805007	-122.418568	40	4/12/2012 4:25:00 PM	16	0.05	1
163	37.800535	-122.419187	37.804821	-122.420194	40	4/12/2012 4:25:00 PM	16	0.26	5.2
164	37.801083	-122.419503	37.805546	-122.422147	40	4/12/2012 4:25:00 PM	16	0.3	6
165	37.805355	-122.415780	37.805232	-122.416911	40	4/12/2012 4:25:00 PM	15	0.05	1
166	37.803050	-122.419078	37.805007	-122.418568	40	4/12/2012 4:25:00 PM	16	0.12	2.4
167	37.803199	-122.426009	37.805520	-122.422044	40	4/12/2012 4:25:00 PM	16	0.23	4.6
168	37.797003	-122.418323	37.805177	-122.413531	40	4/12/2012 4:25:00 PM	6	0.54	10.8
169	37.806162	-122.417074	37.805232	-122.416911	40	4/12/2012 4:25:00 PM	15	0.06	1.2
170	37.804408	-122.422157	37.805546	-122.422147	40	4/16/2012 10:06:00 AM	3	0.07	1.4

Figure 38 Test Results for Uninformed Search with 40% Congestion

6.3.4.8.6 Congestion Level – 50%

181	37.803571	-122.419974	37.804588	-122.421862	50	4/10/2012 6:11:00 PM	15	0.11	2.2
182	37.805530	-122.416246	37.805232	-122.416911	50	4/10/2012 6:11:00 PM	16	0.04	0.8
183	37.806948	-122.414740	37.806584	-122.413821	50	4/10/2012 6:11:00 PM	16	0.05	1
184	37.804267	-122.413609	37.805177	-122.413531	50	4/10/2012 6:11:00 PM	9	0.05	1
185	37.805241	-122.416902	37.805444	-122.415275	50	4/10/2012 6:11:00 PM	14	0.08	1.6
186	37.806149	-122.415531	37.806381	-122.415468	50	4/10/2012 6:11:00 PM	16	0.01	0.2
187	37.807926	-122.417453	37.807101	-122.417292	50	4/10/2012 6:11:00 PM	15	0.05	1
188	37.805180	-122.422308	37.804380	-122.423482	50	4/10/2012 6:11:00 PM	11	0.07	1.4
189	37.803571	-122.419974	37.804588	-122.421862	50	4/6/2012 9:29:00 PM	15	0.11	2.2
190	37.805530	-122.416246	37.805232	-122.416911	50	4/6/2012 9:29:00 PM	16	0.04	0.8
191	37.806948	-122.414740	37.806584	-122.413821	50	4/6/2012 9:29:00 PM	16	0.05	1
192	37.804267	-122.413609	37.805177	-122.413531	50	4/6/2012 9:29:00 PM	9	0.05	1
193	37.805241	-122.416902	37.805444	-122.415275	50	4/6/2012 9:29:00 PM	14	0.08	1.6
194	37.806149	-122.415531	37.806381	-122.415468	50	4/6/2012 9:29:00 PM	16	0.01	0.2
195	37.807926	-122.417453	37.807101	-122.417292	50	4/6/2012 9:29:00 PM	15	0.05	1
196	37.805180	-122.422308	37.804380	-122.423482	50	4/6/2012 9:29:00 PM	11	0.07	1.4
197	37.803571	-122.419974	37.806693	-122.420814	50	4/6/2012 12:09:00 AM	12	0.19	3.8
198	37.805530	-122.416246	37.807308	-122.415832	50	4/6/2012 12:09:00 AM	13	0.11	2.2
199	37.806948	-122.414740	37.806584	-122.413821	50	4/6/2012 12:09:00 AM	16	0.05	1
200	37.804267	-122.413609	37.805177	-122.413531	50	4/6/2012 12:09:00 AM	9	0.05	1
201	37.805241	-122.416902	37.805444	-122.415275	50	4/6/2012 12:09:00 AM	14	0.08	1.6
202	37.806149	-122.415531	37.806381	-122.415468	50	4/6/2012 12:09:00 AM	16	0.01	0.2
203	37.807926	-122.417453	37.807308	-122.415832	50	4/6/2012 12:09:00 AM	10	0.09	1.8
204	37.805180	-122.422308	37.805546	-122.422147	50	4/6/2012 12:09:00 AM	8	0.02	0.4
205	37.806439	-122.415887	37.807101	-122.417292	50	4/12/2012 4:25:00 PM	16	0.08	1.6
206	37.803483	-122.419959	37.805007	-122.418568	50	4/12/2012 4:25:00 PM	16	0.11	2.2
207	37.804335	-122.418101	37.805007	-122.418568	50	4/12/2012 4:25:00 PM	16	0.05	1
208	37.800535	-122.419187	37.804821	-122.420194	50	4/12/2012 4:25:00 PM	16	0.26	5.2
209	37.801083	-122.419503	37.805520	-122.422044	50	4/12/2012 4:25:00 PM	16	0.29	5.8
210	37.805355	-122.415780	37.805232	-122.416911	50	4/12/2012 4:25:00 PM	15	0.05	1
211	37.803050	-122.419078	37.805007	-122.418568	50	4/12/2012 4:25:00 PM	16	0.12	2.4
212	37.803199	-122.426009	37.805520	-122.422044	50	4/12/2012 4:25:00 PM	16	0.23	4.6
213	37.797003	-122.418323	37.805177	-122.413531	50	4/12/2012 4:25:00 PM	6	0.54	10.8
214	37.806162	-122.417074	37.805232	-122.416911	50	4/12/2012 4:25:00 PM	15	0.06	1.2
215	37.804408	-122.422157	37.805304	-122.423686	50	4/16/2012 10:06:00 AM	16	0.09	1.8

Figure 39 Test Results for Uninformed Search with 50% Congestion

6.3.4.9. Experiment Analysis

6.3.4.9.1. Graph1: Time to find Parking Vs Congestion Levels

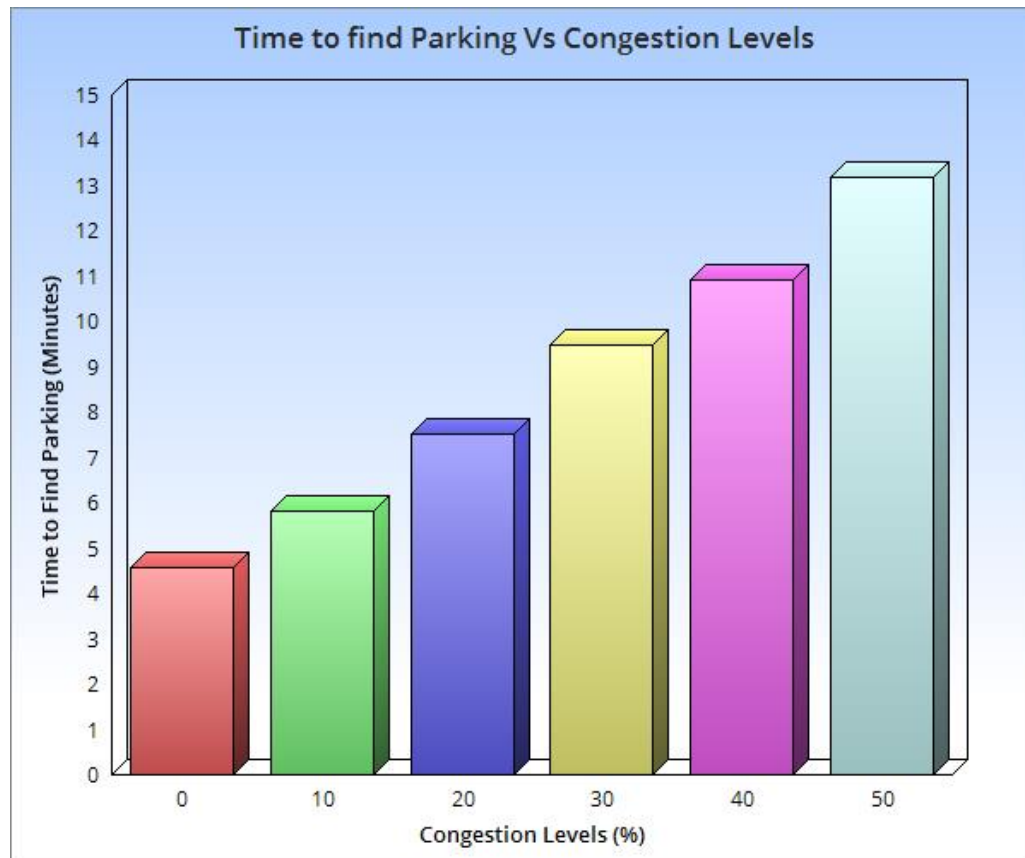


Figure 40 Graph between 'Time to Find Parking' and 'Congestion Levels'

6.3.4.9.2. Graph Description

The above graph shows the analysis of 'Time taken to find parking' at different 'Congestion Levels'. From the above graph, we can conclude that the time taken to find a parking slot, on average increases with increase in congestion as the number of available slots decrease.

6.3.4.9.3. Graph2: Query Time Vs Congestion Levels

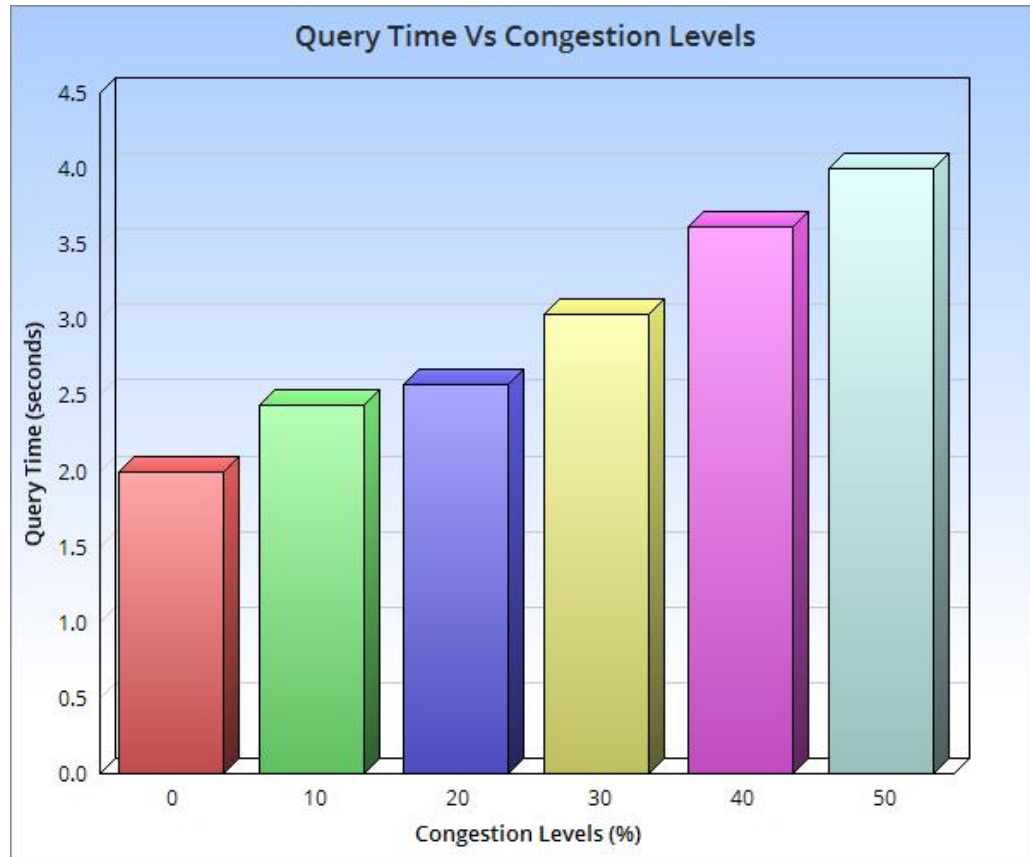


Figure 41 Graph between Query Time and Congestion Levels

6.3.4.9.4. Graph Description:

The above graph displays a plot between 'Query Time' and 'Congestion Level'. Query time refers to the running time of the algorithm and this has been recorded for various congestion levels. For each congestion level, the time plotted on the graph is the average of running time of the algorithm over 45 different inputs.

6.3.4.9.5. Web Output

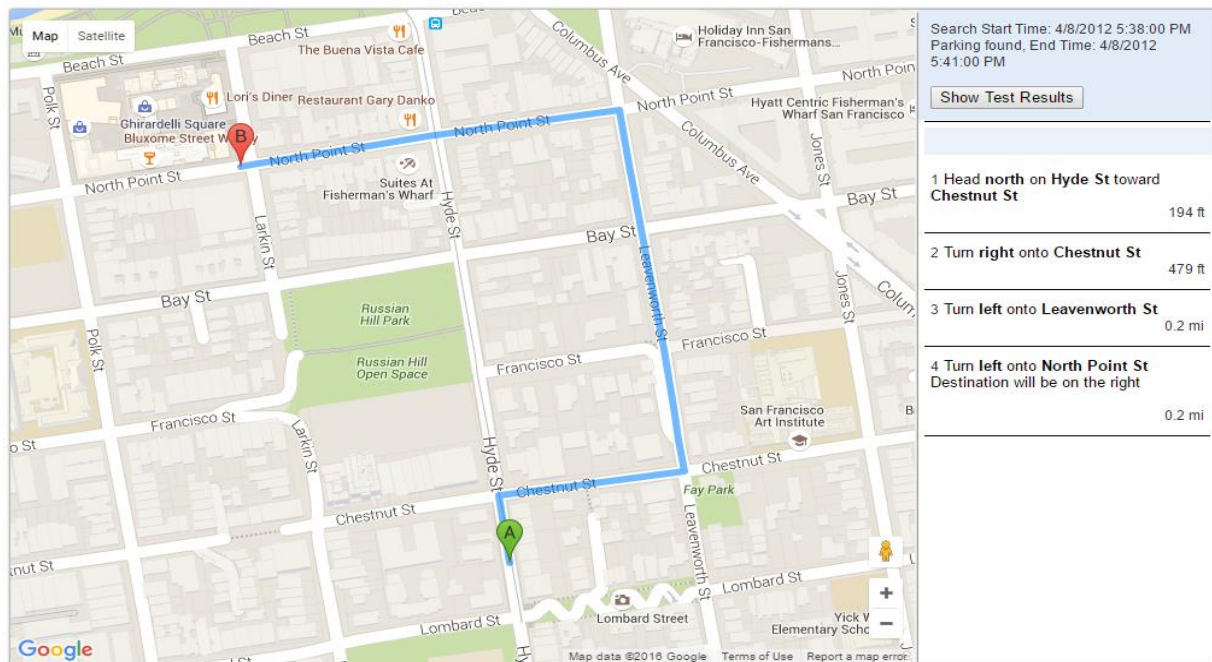


Figure 42 Output of the Uninformed Search on Web Application

7. Comparison of algorithms

The above tasks are compared and contrasted with the same set of inputs. The total time taken by the user is noted down for each of the Algorithms. Each of these inputs are mapped against various congestion levels.

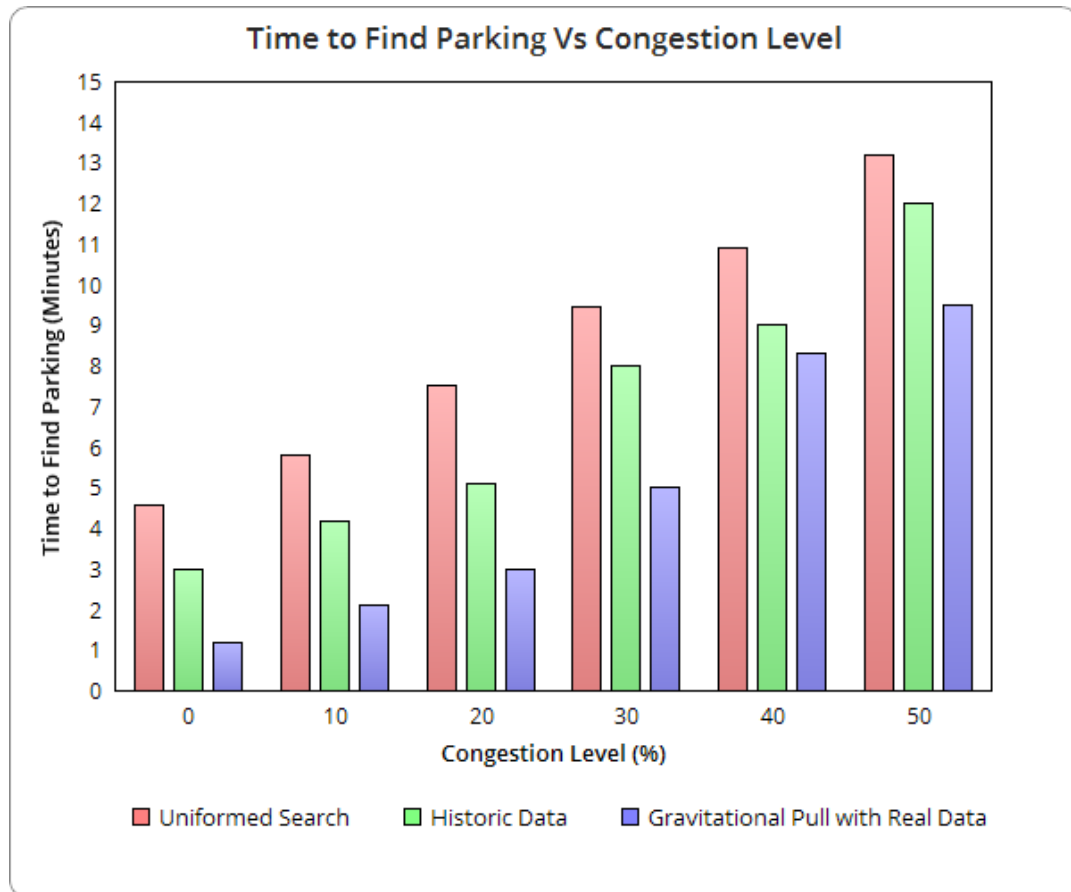


Figure 43 Graph showing the Comparison between various algorithms implemented

As observed the Real Time Gravitational Algorithm has the best performance as compared to the rest. The Historic Based tries to catch up but fails in some areas since the prediction is based of clubbed historic data which might not be very accurate when there is a wide variety of fluctuation for that parking slot. On the other hand the uninformed search performs quite well when a parking lot is readily available. It lags when the congestion level is increased since the user would have to manually travel and check multiple blocks for available parking slots.

8. Technologies Used

These were the technologies used to create the web app

1. ASP.NET
2. C#
3. MYSQL
4. Javascript
5. Python

9. Steps to Run the Project

1. Install Visual Studio 2013 Community Version from here
<https://www.visualstudio.com/en-us/news/vs2013-community-vs.aspx>
2. Download the Zip File from github containing the source code
<https://github.com/francisminu/DBMSFinal.git>
3. Place the 'Projects' folder present in \DBMSFinal folder on github into the below location:
C:\Users\User_name\Documents\Visual Studio 2013\ (User_name : Check the user name of the current machine)
If the folder 'Projects' is already present, then copy only 'DBMSResourceSharingProject'
4. Place the 'Websites' folder present in \DBMSFinal folder on github into the below location:
C:\Users\User_name\Documents\Visual Studio 2013\ (User_name : Check the user name of the current machine)
If the folder 'Websites' is already present, then copy only 'DBMSResourceSharingProject'
5. Open Visual Studio 2013. Open the project by clicking 'Open' and choosing the DBMSResourceSharingProject.sln file present in the location:
C:\Users\User_name\Documents\Visual Studio 2013\Projects\DBMSResourceSharingProject
(User_name - Check the user name of the current machine)
6. Install Newtonsoft JSON package via Nugget from the package console manager
(Tools -> Nugget Package Manager -> Package Manager Console)
7. Build the project (Build -> Build Solution)
8. Set index.html as the start page (Right click index.html page in the solution explorer)
9. Click on 'Run'

10. Acknowledgment

We would like to thank professor Wolfson for taking out his time and helping us at every stage of the project development process.

References

- [1]. https://piazza.com/class_profile/get_resource/ijdvcreqgc36z/ijupxosubfw24k
- [2]. https://en.wikipedia.org/wiki/Spatiotemporal_database
- [3]. <https://www.cs.uic.edu/~dayala/mdm12.pdf>
- [4]. <http://www.esri.com/news/arcnews/winter0809articles/the-geodatabase.html>
- [5]. <http://sfpark.org/how-it-works/open-data-page/>
- [6]. <https://developers.google.com/maps/documentation/javascript/>
- [7]. <https://developers.google.com/maps/web-services/client-library>