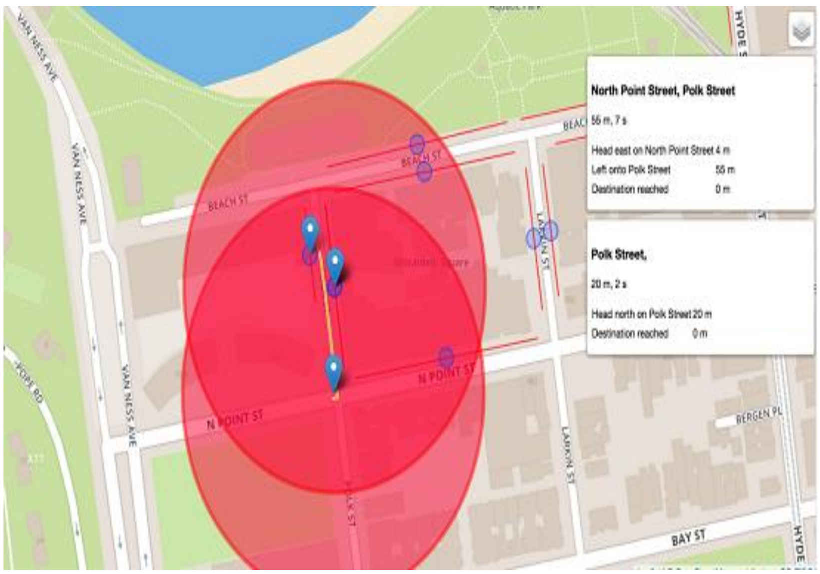


SEARCH OF SPATIO-TEMPORAL RESOURCES

Project Final Report

ABSTRACT

This project evaluates various algorithms on a given Spatio-temporal dataset. The algorithms try to deal with the issues faced by drivers to park in big cities like San Francisco

Group 4

Anuj Mehta, Fnu Ruchi, Kunal Khona, Sharad Tanvar

Acknowledgement

We would like to thank **Prof. Ouri Wolfson** for his professional input, feedback, and support, as well explaining the need required to make our product a successful one. His research experience in database management system domain proved fruitful and extensive when it came to understanding the project requirements and guiding us through the development lifecycle. We look at parking spot allocation is a whole new light after this project and appreciate the thought and execution that goes in it.

We would also like to thank the teaching assistants, **Sandeep Sasidharan** and **Qing Guo**, for their continued guidance and excellent feedback throughout the course of the project and requirements and the deadlines.

Submitted By:

Anuj Mehta

Fnu Ruchi

Kunal Khona

Sharad Tanvar

Table of Contents

Introduction

Parking availability is one of the most debated about and discussed topics in the 21st century due to the number of cars we have in our major cities nowadays. Parking Spots are limited in the cities and the information on their availability is not readily available. Multiple Cities are facing the problem with parking spot information. The drivers who need a parking spot also have to wait for a longer time to find a spot, which can cause traffic congestion and wastage of fuel causing a losses to the state and the individual.

Cities today are trying to come up with novel solutions to address the parking issues. The city of San Francisco implemented a sensor-based approach for this problem. The city installed sensors in the ground to track the availability of the parking slot at that specific time. But the installation and the maintenance costs of these sensors costs a fortune and not every city has the resources to install and maintain.

The major problem, which the project is trying to address, is how to have a similar parking slot finding implementation without incurring the cost of the installing and maintaining the sensors. The project aims at implementing various algorithms working at different levels of information to address the issue better.

The following approaches were discussed and implemented in the project:

1. **Task One - Deterministic Approach:**

   This approach with measuring metrics associated with the SFPark Project being implemented in San Francisco. In this approach, we have real time information from the sensors installed in the bay area in San Francisco. Real Time Information consists of the location of the parking slots and the number of available parking slot at any specific instance of time.

2. **Task Two – Probabilistic Approach:**

   This approach is more open ended than the first approach as we have historical data of the parking slot availability. We use that information to get probability of getting a slot at the specific time.

3. Task three - Baseline Approach

In this approach we don’t have any sort of information on the availability of the parking slot. We have to use this task as our baseline approach if we do not have any level of information on the availability of slots. We only have to consider the static geographic information such as the number of operational slots and their geographic locations.

Goal of the Project

The goal of the project is compare and analyze various approaches to find parking slots. We compare each of the algorithms with each other to interpret how availability of various levels of information controls the performance. The higher level of information comes at a high cost as sensors are needed and they are expensive.

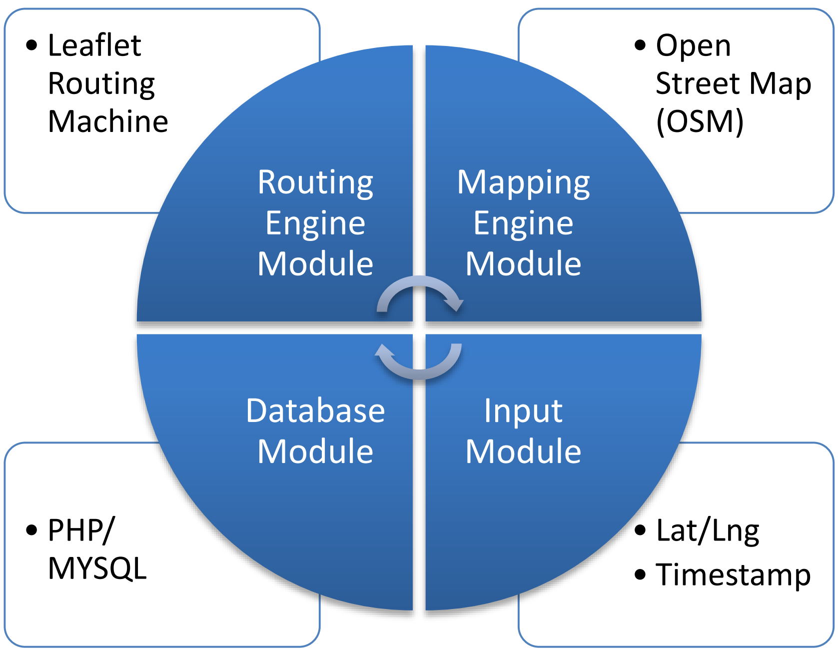
We have various tasks at hand, based on the level of information used. The approach of this project is to identify various algorithms if the real time information of the availability of slots remains unavailable.

Upon completion of the project and the experiments involved, we can comprehensively compare the different approaches with each other using different metrics and also assert the pros and cons of each approach.

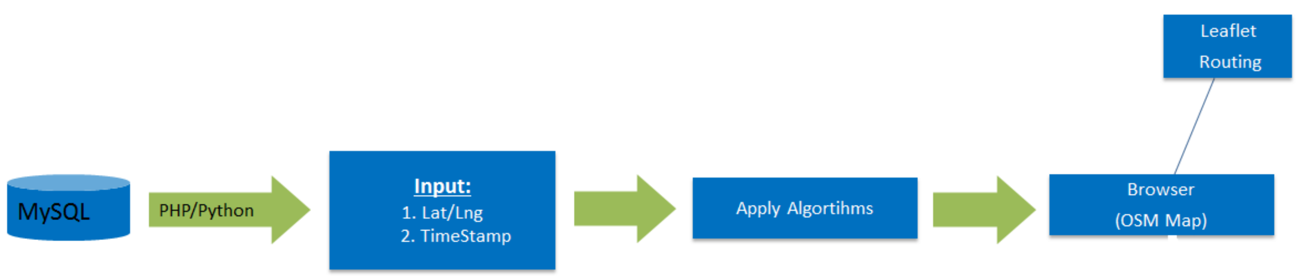
Design Overview

The project deals with applying various algorithms to find a parking spot and measuring the metrics associated with it. For the purpose of simulation and analysis of these algorithms, we require an underlying architecture to generate results.

High Level System Design



Low Level System Design



Modules of the program:

1. MySQL database.
2. OSM Map.
3. Leaflet Routing Machine.
4. HTML input page.

Assumptions:

1. The user starts the search for the parking resources from the location where he intends to go i.e., the final destination. The algorithms start the search for the parking blocks from this point. This means that the user’s starting location and the ending location for our purpose is the same.
2. Travel Speed:
    i) Driving Speed: 22mph / 10 m/s.
    ii) Walking Speed: 3mph/ 1.38 m/s.

Problem and Solution Approach

As discussed earlier, there can be multiple solutions to the problem in varying conditions. The aim/goal of the project is evaluating these solutions under different conditions.

The solution has been divided into three tasks/approaches, each of which deals with the same data but under different conditions which try to simulate a real life situation. The approaches and the level of information available in these approaches are as follows:

1. **Task One - Deterministic Approach**
   Level of Information Available: Maximum
   Maintenance Cost: Most Expensive
   Accuracy: High
2. **Task Two - Probabilistic Approach**
   Level of Information Available: Historical Information Available.
   Maintenance Cost: Medium Cost
   Accuracy: Medium
3. **Task Three - Baseline Approach**
   Level of Information Available: Minimum/ None
   Maintenance Cost: Least Expensive
   Accuracy: Low

Metrics Used to measure efficiency of each Algorithm are discussed as follows:

1. Time to Park the vehicle.
2. Time to walk back to the destination.
3. Total time = (Time to park + Time to walk back to the destination).

These metrics give a comprehensive way to compare tasks and simulate different environment and compare each other.

Implementation Details

Baseline Algorithm (No information):

1. The algorithm is based on greedy approach having no real time information about current available slot.
2. It has information about the operational parking blocks available near to the destination
3. The algorithm is implemented as follows:
    a. Fetch all the blocks which are operational
    b. The blocks are sorted in order of increasing distance from the destination
    c. User is directed to the block which is nearest to the destination
    d. If no parking spot is available at the directed block, user is navigated to the second nearest block from the initial search location.
    e. Continue navigation until an available parking spot is found.
    f. Availability is checked using the data given.

Probabilistic Approach:

Historical Analysis 1 (Greedy):

1. The algorithm takes into account the historical data of SFpark
2. The algorithm ranks the parking blocks based on their weighted means of the historic availability in the block which gives higher probability of finding a parking slot. Slots are filtered further on the basis of standard deviation.
3. The algorithm is implemented as follows:
    a. Find set of blocks in 100 meter radius using input location and timestamp.
    b. Availability information of block is considered for the timestamp of +/- hour for the same day of the week.
    c. Using above information, weighted average of the availability is calculated.
    d. Mean, variance and standard deviation of each block is calculated.
    e. Based upon the values so obtained user is directed to the block.
    f. Blocks are considered based on the following criteria:
        i. Higher mean value.

ii. If the higher mean varies by 20% or less, blocks with lesser deviation are recommended.

g. If no parking slot is available in the recommended block, the radius is doubled and the algorithm is executed once again.

h. Total distance is equal to the distance covered by user till allocated with a slot.

Historical Analysis 2(Gravitational):

1. It implements Gravitational approach using historic data of SFpark.
2. The algorithm is implemented as follows:
   a. Availability information of block is considered for the timestamp of +/- hour for the same day of the weeks.
   b. Calculate the weighted average and means.
   c. Based upon the values so obtained user is directed to the block with maximum force vector.
      i. Calculate force vectors of each slot from vehicle at the given user location and the timestamp with formula:

         F = (historic mean of available slots)/ (distance^2).

   d. If no parking is found, repeat the process.
   e. Total distance is equal to the distance covered by user till allocated with a slot.

Deterministic - Gravitational Algorithm (Real time information is available):

1. The algorithm is based on the concept that the more the available parking spots a block has and the closer it is to the vehicle, the higher gravitational force it has to pull the vehicle towards it.
2. The algorithm implemented as follows:
   a. Find S set of slots available at any step (search all blocks which has availability greater than zero).
   b. Calculate force vectors of each slot from vehicle given by the formula:
      i. $F(v,b) = n_b / \text{distance}(v, b)^2$.

ii. n_b is the number of available parking spots at block b.

iii. distance(v,b) is the routing distance given by routing machine considering the road network.

c. Total gravitational force in each direction is calculated from the user's location node.

d. Then the user is directed to the direction which has maximum force.

Deterministic - Greedy Algorithm (With real time data)

1. The algorithm design is based on basic greedy approach.
2. The algorithm is implemented as follows:
   a. Find S set of blocks at any step which has availability greater than zero.
   b. All the blocks are sorted by distance from the destination (here distance is given by the routing machine.
   c. The user is directed towards block which has shortest distance.

Nash Equilibrium

1. No user can reduce its travel-time unilaterally, if all others are rational (if they attempt to minimize theirs.
2. The inputs are given one after the other with same timestamp and varying or same locations.
3. Each user tries to find the best possible parking spot (strategy).
4. Once the strategy is chosen, the vehicle and the slot are removed for future considerations.
5. Therefore, it is not possible for a single vehicle to decrease its travel time unilaterally.
6. On each iteration:
   a. Choose minimum cost block for the user
   b. Assign the vehicle to that slot
   c. Remove that vehicle-slot pair from future consideration

Experiment and Simulations

For each result, explain: what is the goal of the experiment, what you did, then comes the plot, then interpret the plot. Try and have some comparative result, with prior work.

We have analyzed the results for 10 different locations for 5 different time stamps. The data has been logged into a spread which is attached along named "Data_analysis.xslx". The below image is a representation of the data analyzed for the baseline algorithm for different locations and different timestamps. Each algorithm has been run for various congestion levels – 0%, 20%, 40%, 60%. Each congestion level has 50 simulation results, which results in 200 simulations for each algorithm. Thus resulting in 1200 simulations overall for 6 algorithms.

BaseLine Algo:							
	Congestion Level 0% :						
	Places /Time		4-9-2012 15:00:00	4-19-2012 19:35:15	4-13-2012 14:15:23	4-24-2012 17:03:37	5-2-2012 18:30:51
		Time to park	348	97	97	97	66
		Time to walk back	394	255	255	255	197
	W. end of Jefferson	Total Time	742	352	352	352	263
		Time to park	742	57	256	140	7
		Time to walk back	524	133	258	253	43
	North Point and Larkin	Total Time	1266	190	514	393	50
		Time to park	273	70	115	53	273
		Time to walk back	443	288	361	275	443
	Beach and Stockton	Total Time	716	358	476	328	716
		Time to park	28	28	12	12	28
		Time to walk back	103	103	91	91	103
	Midway 2700 block of Leavenworth	Total Time	131	131	103	103	131
		Time to park	177	36	36	6	36
		Time to walk back	133	55	55	34	34
	North Point and Jones	Total Time	310	91	91	40	70
		Time to park	595	330	90	21	311
		Time to walk back	389	208	133	45	192
	Jefferson and Taylor	Total Time	984	538	223	66	503
		Time to park	16	16	16	15	16
		Time to walk back	35	35	35	31	35
	Bay and Mason	Total Time	51	51	51	46	51

Simulation Procedure:

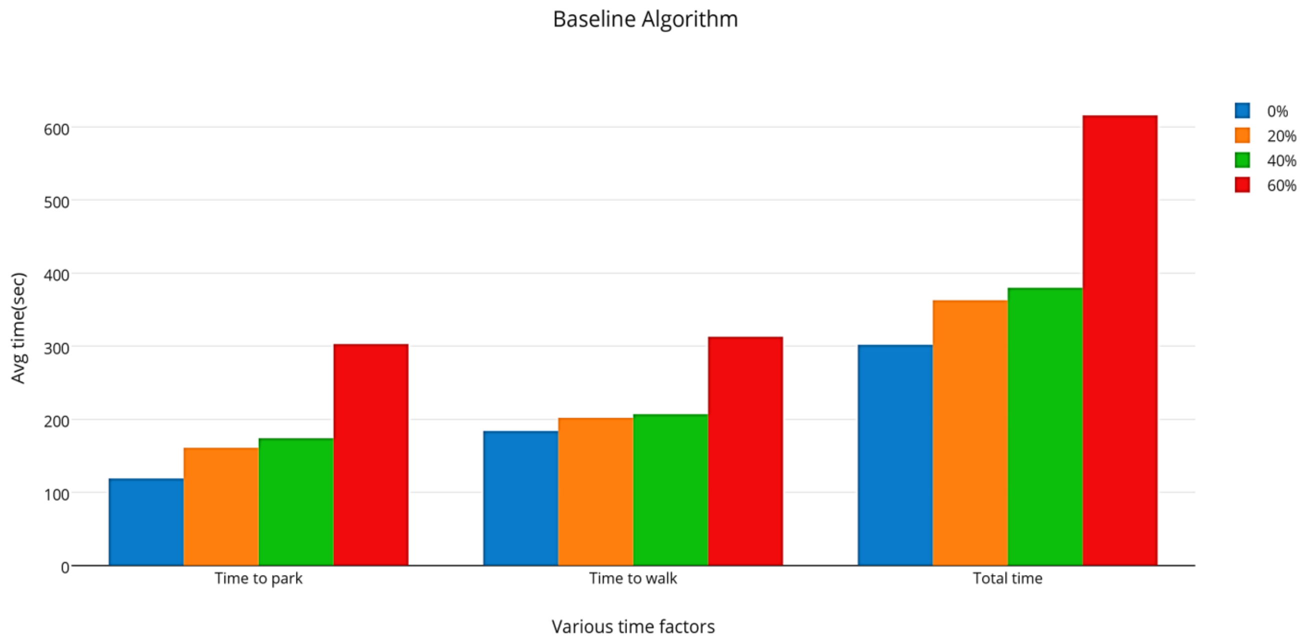
1. Input the destination location and the timestamp.
2. Run the algorithm and identify the parking slots.
3. Direct the user to the spot. If the spot is not available recalculate another spot and redirect until an available slot is found.

4. Compute the average of the individual outputs.
   (Cost = average time to park + average time to walk back).
5. Congestion Levels tested: 0%, 20%, 40% and 60%.

The data from various algorithms need to be compared and analyzed. The comparison has been essentially been done on the basis of same levels of data analyzed and the different levels of data analyzed by each of the algorithm as depicted in the table below. The rows with two algorithms are the ones that analyze the same level of information whereas, a column interprets the analysis of the performance for various levels of information.

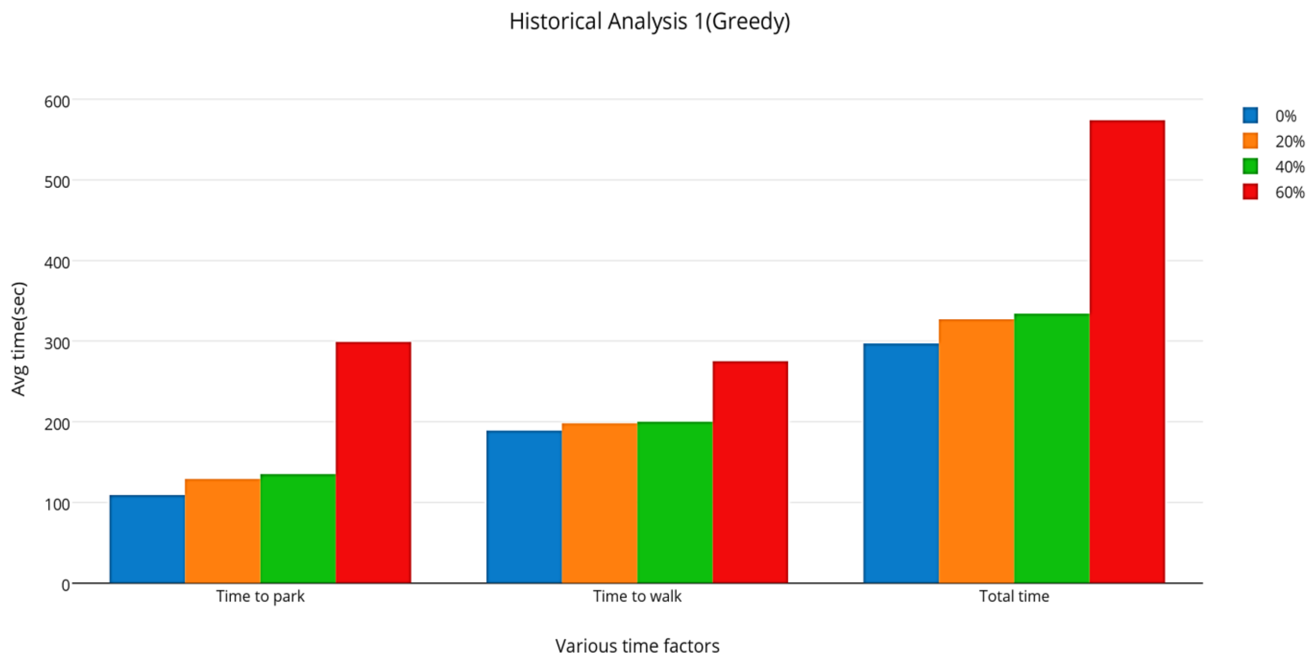
Level of information\Algorithms	Type 1	Type 2
Extra	Nash Equilibrium	-----
Deterministic	Gravitational Approach	Greedy with Real Time Information
Probabilistic	Historical Analysis 1	Historical Analysis 2
Baseline	Uninformed Greedy Approach	-----

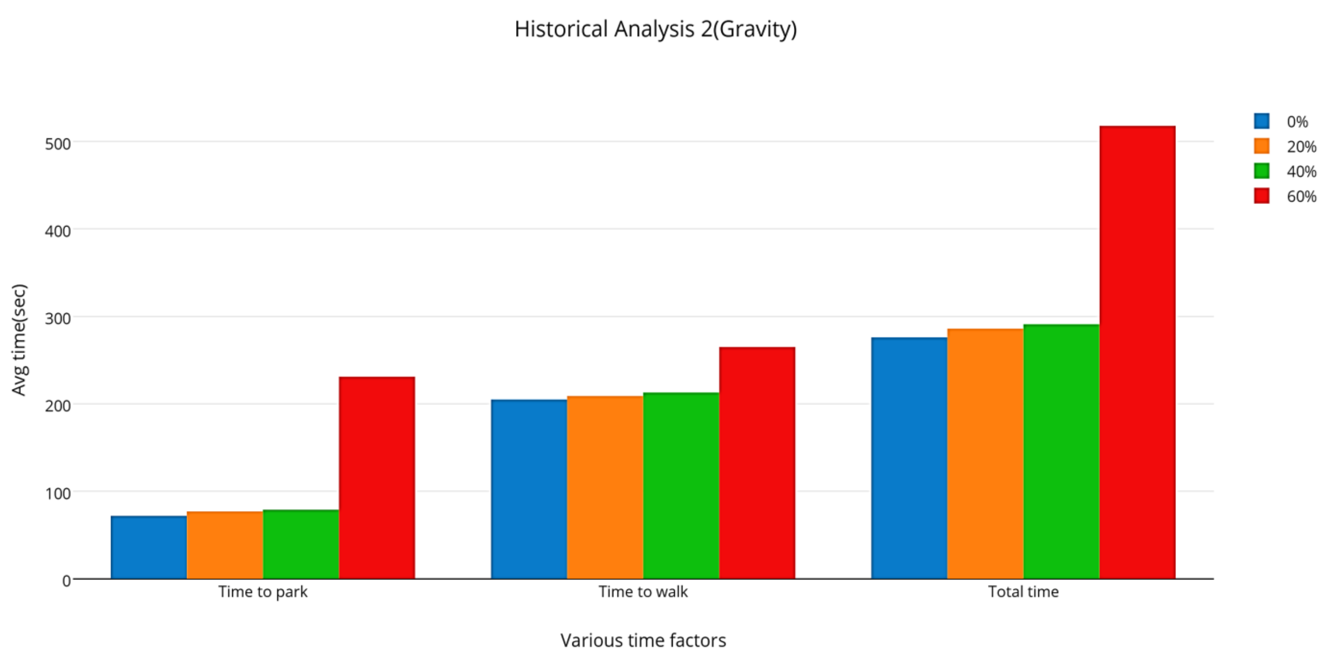
Data Analysis

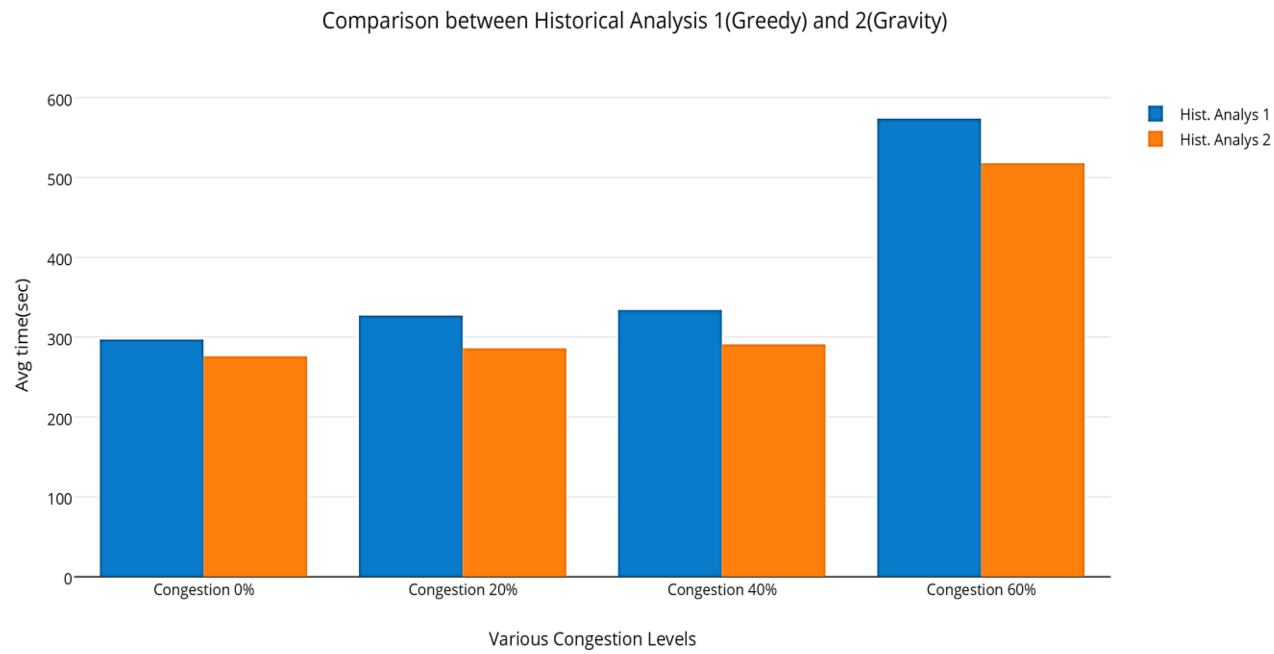


Data Analysis of the Baseline Algorithm:

1. The algorithm follows greedy approach where no real time information is available.
2. The algorithm directs user towards block which is nearest to the destination as can be referenced from the time to park.
3. As the congestion level increases, the availability of slots decreases. Hence, the time to park increases.
4. With 60% congestion, the availability of parking slots decreases by a huge amount, hence the sudden increase in time to park.


Historical Analysis 1(Greedy)
Avg time(sec)
600
500
400
300
200
100
0
Time to park
Time to walk
Total time
Various time factors
0%
20%
40%
60%


Historical Analysis 2(Gravity)
Avg time(sec)
500
400
300
200
100
0
Time to park
Time to walk
Total time
Various time factors
0%
20%
40%
60%



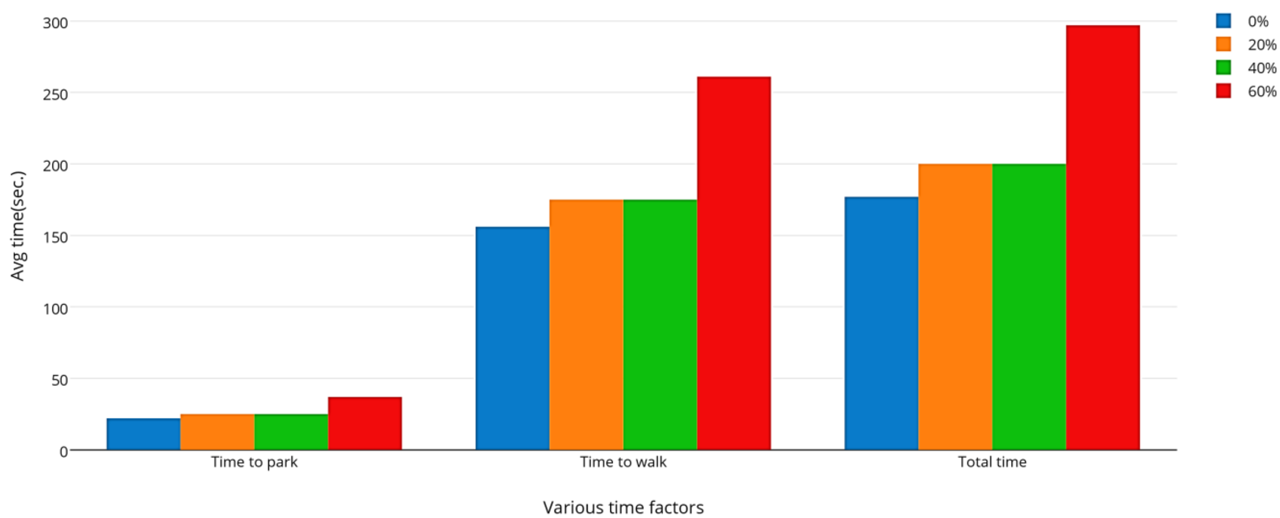
Data Analysis of Probabilistic Algorithms:

1. For congestion levels up to 40%, the time to park increases by a small margin for both the algorithms.

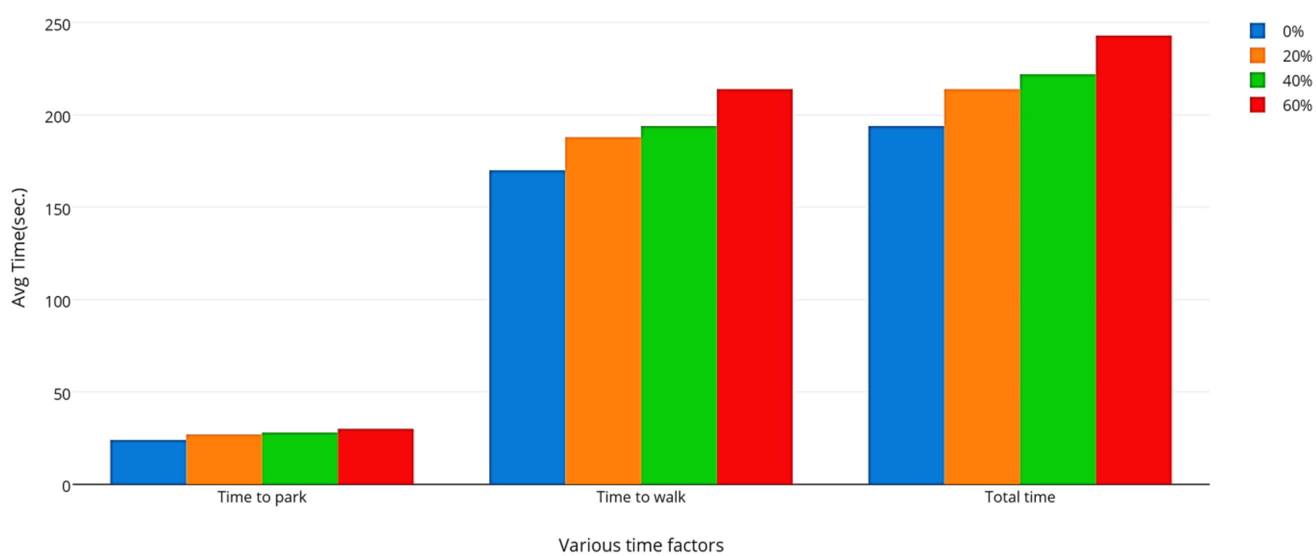
   It can be noted that the historic gravitational approach is already outperforming the historic greedy approach from the start.
2. However, with 60% congestion, the availability of parking slots decreases by a huge amount, hence the sudden increase in time to park.
3. The comparison between the two probabilistic approaches can be referenced from the graph shown above as:

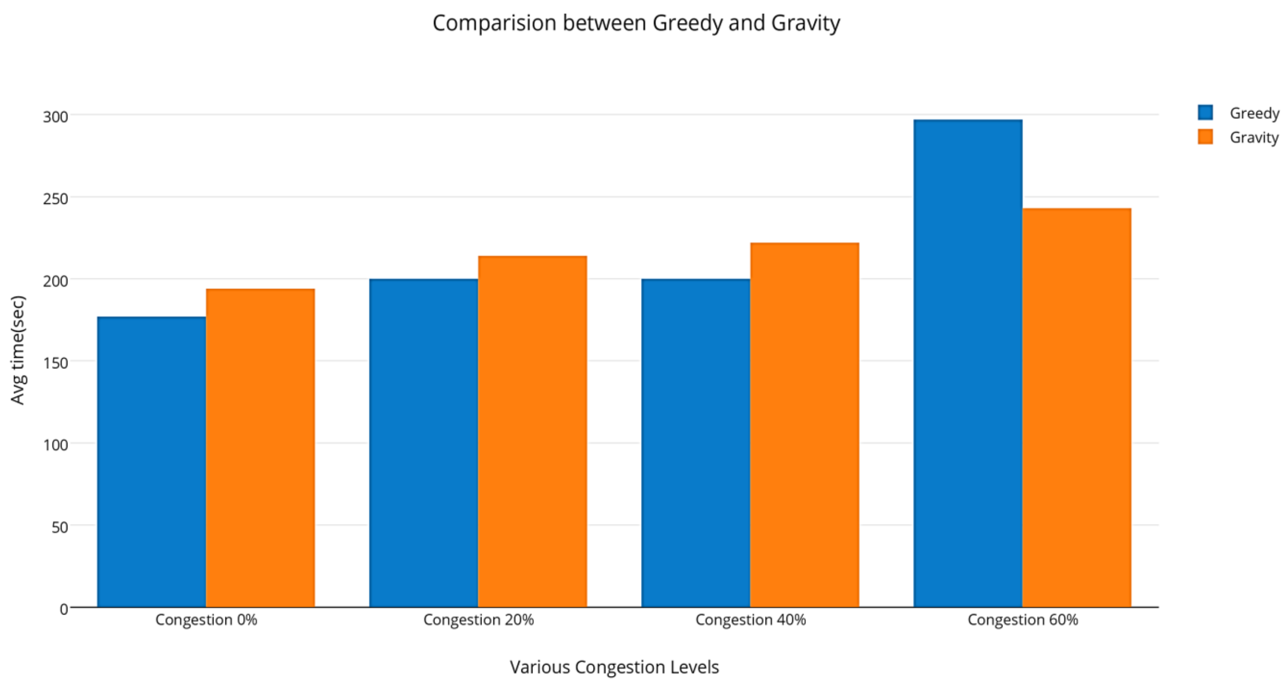
   Total time (Time to park + Time to walk) is higher in historic greedy than historic gravitational because greedy considers only the mean availability ignoring the factor of the distance whereas, gravitational considers ratio of availability and routing distance which yields better results.

Greedy Algorithm with real time information



Gravitational Algorithm

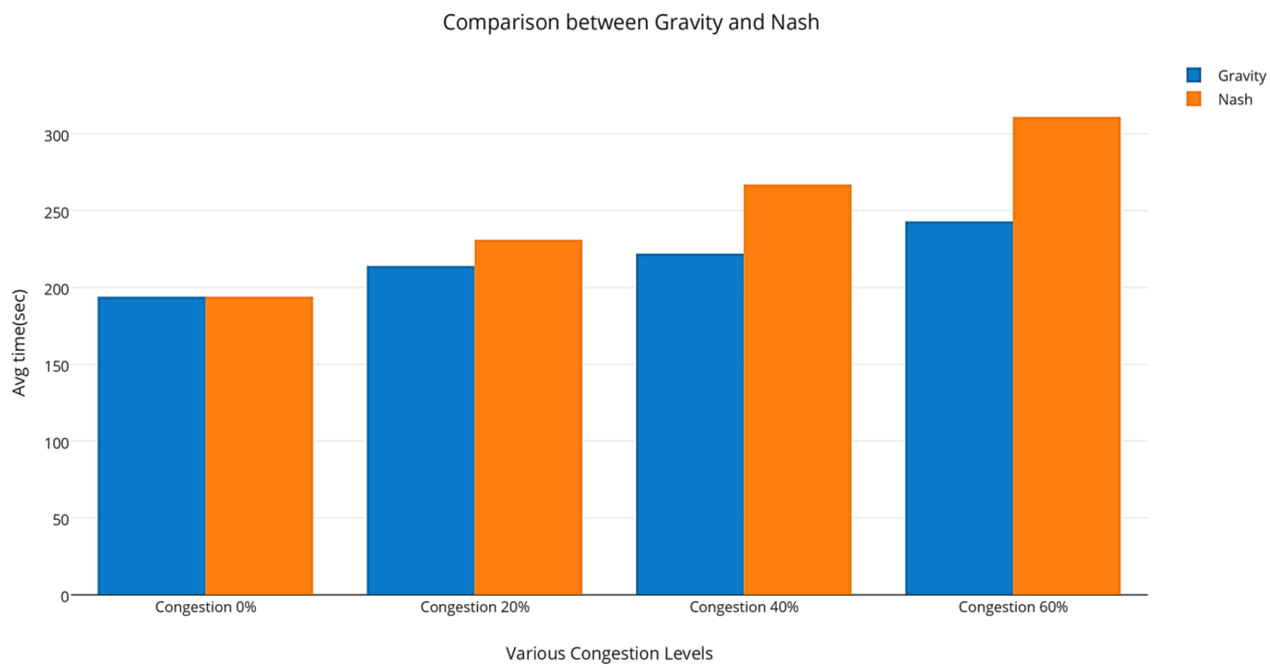




Data Analysis of Deterministic Algorithms:

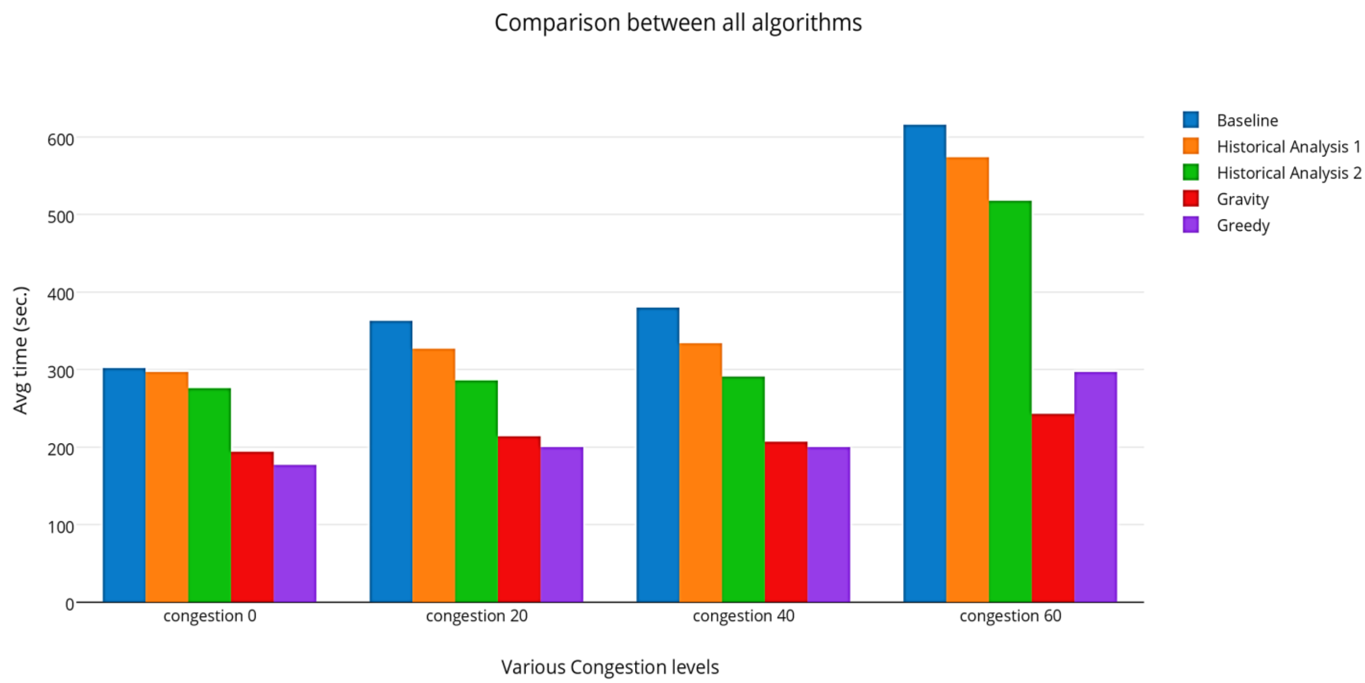
The comparison between the Greedy and Gravitational algorithm can be referenced from the graph shown above as:

1. Total time in greedy is low when the congestion level is low because greedy gives the nearest available slot to the user and gravity follows the optimized way of taking ratio of availability and distance square.
2. When the congestion increases from 0% to 20%, greedy is still outperforming gravitational approach. The same can be said for congestion level of 40%.
3. However, when the congestion level hits 60%, the number of parking slots available are significantly less. Hence, the gravitational approach starts behaving better than the greedy as the approach tries to find a parking block which has better probability of having availability.



Data Analysis of Nash with Gravitational Approach:

1. The comparision between gravity and nash equilibrium is observed from the graph above as follows:
   a) Clearly, at 0% congestion level, the performances of both the algorithms is nearly the same.
   b) With increase in congestion level from 0% to 20% gravity starts to outperform Nash.
   c) At even higher congestions the performance of nash starts is found to be even worse in comparision to gravitational approach.
   d) This behaviour can be concluded from the fact that gravity depends on the number of available slots and gives more preference to the block which has higher number of available slots. When congestion level is increased, gravitational tries to find the blocks which have more chances of finding a parking slot yeilding results having better performance than Nash Equillibrium.



The above graph depicts the performance in terms to the total time to park for various congestion levels of different algorithms. At every congestion level, the baseline algorithm (uninformed approach) Performs the worst. Probabilistic algorithms always perform better than the baseline algorithm. Finally, as expected, the deterministic algorithms perform the best under all circumstances.

It is worth noting that the performances of the probabilistic algorithms and the baseline algorithm decreases very rapidly in comparison to the deterministic approaches with the increase in congestion levels. Additionally, it observed that at lower congestion levels, the performance of greedy algorithm with real time availability information performs the best. However, at higher congestion levels, the gravitational approach performs the best.

Future Research

1. In our models, we have an assumption that every vehicle moves at a uniform speed, which is strictly not true in the real world scenario.
2. We could incorporate the real time traffic information as well to better analyze the performance of the algorithms.
3. The historical data we have is only a month's data. If we manage to get access to data for a longer duration of time, we can make better analysis of the historical data which can possibly yield better performances.

Conclusion

In this project, our main goal was to analyze the vehicular parking under different circumstances. We presented various models that can be used to study the parking problem. For the deterministic models, we implemented a greedy approach which tries to find the closest available parking slot. We also implemented the gravitational approach using the same amount of information as the greedy approach. For the historical interpretation, we implemented two different models, one which ignores the topography considering only the availability, whereas the other keeps in mind the distance factor. The two models were also compared to estimate the best approach of the two. Finally, we have a blind uninformed search which uses no more data that the geographic location of the blocks along with the number of operational slots of these blocks.

From the data analysis, we can conclude that having deterministic information is better for faster parking but at the expense of installation of expensive sensors whereas, the probabilistic approach could be good model for finding the availability. The uninformed can be used as the last resort to direct the user to the blocks for manual inspection of the availability.

References

1. Parking in competitive settings: A Gravitational Approach: Daniel Ayala, Ouri Wolfson, Bo Xu, Bhaskar Das Gupta, Jie Lin: IEEE 13th international Conference on Mobile Data Management, 2012.
2. Predicting the probability of parking vehicle based on dynamic simulation, Sutanto Soehodho Journal of Eastern Asia Society for Transportation Studies, vol. 3 , No. 3, September, 1999.
3. PhonePark: Street parking using mobile phones, Leon Stenneth, Ouri Wolfson, Bo Xu, Philip S, Research for NSF.
4. W. Park, B. Kim, D. Seo, D. Kim, and K. Lee, “Parking space detection using ultrasonic sensor in parking assistance system,” in IEEE Intelligent Vehicles Symposium, 2008.
5. S. Mathur, T. Jin, N. Kasturirangan, J. Chandrashekharan, W. Xue, M. Gruteser, and W. Trappe, “Parknet: Drive-by sensing of road-side parking statistics,” in MobiSys, San Francisco, CA, June 2010.
6. D. Ayala, O. Wolfson, B. Xu, B. Dasgupta, and J. Lin, “Parking slot assignment games,” in Proc. of the 19th Intl. Conf. on Advances in Geographic Information Systems (ACM SIGSPATIAL GIS 2011),Chicago, IL, November 2011.
7. Pricing the Curb: How San Francisco, Chicago and Washington D.C. are reducing traffic with innovative curb-side parking policy, Transportation Alternatives (www.transalt.org), New York, NY, July 2008.
8. San Francisco Park website: http://sfpark.org/