Optimizing Pedestrian Crossing Methods with Individual Agent Model

Geoff Stevens, Ben Zhou, Matt Wu

November 9, 2015

Abstract

Pedestrian crossing patterns in a gridded city, such as New York City, can take on a variety of forms. In the most simple case, a pedestrian may choose to walk a predetermined path in order to reach their destination. In other cases, the road that an individual decides to cross is dramatically influenced by which road has a cross signal, as well as the general willingness for an individual to wait for a do-not-cross signal to change. Through examining the variety of potential crossing patterns via an Individual Agent Model, we are able to quantify the optimal pedestrian crossing strategies in a gridded system.

1 Introduction and Background

Transportation is a fundamental aspect of any city, providing residents with an efficient method of navigating the latticework of streets that carve out neighborhoods, boroughs, and blocks. Oftentimes, residents are tasked with the mission of traversing these grids to minimize the time they spend getting from their current location to their destination. City-goers can utilize public transportation, including subways and trolleys, or they may prefer private methods of travel, such as by car and bike.

For the purposes of the following simulation however, we will look specifically at ideal pedestrian walking behavior. When confronted with a city grid, the question remains: what is the best method of getting to your destination, given the multitude of street lights that could potentially impede your progress? Using an individual agent model and running various trials, it is possible to discern the positive or negative impact of choosing specific strategies of crossing streets in a variety of contexts.

As urbanization continues to grow across the world, the importance of understanding pedestrian behavior and how they may seek to optimize their routes becomes more and more crucial to understand. Minimizing the time spent at do-not-cross signals can help urban planners diffuse congestion and assist city residents reach their destinations efficiently.

2 Individual Agent Model

To answer the questions about optimizing pedestrian behavior, we constructed an individual agent model that can simulate the pathway of a single pedestrian attempting to cross a grid.

First, one of the parameters to the model involves the size of the grid and the amount of time that is required to traverse one horizontal and one vertical segment. Each grid was assumed to be perfect (i.e. it consists of identical squares or rectangles). We will identify a single segment in the horizontal direction as a width and a single segment in the vertical direction as a length.

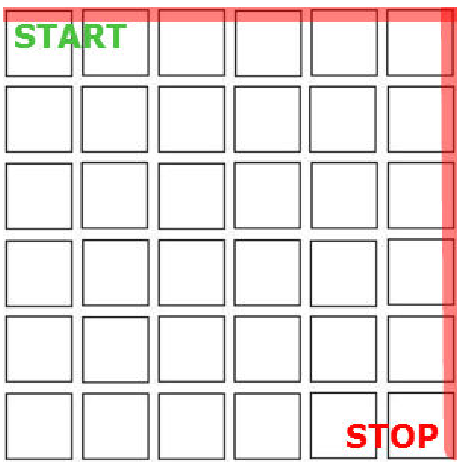
A second parameter to the model is the duration of a do-not-cross signal in a given simulation. Although the units are fairly arbitrary (i.e. the results will be accurate so long as units are consistent throughout the analysis), for the purposes of this model we will be referring to the duration of streetlights in terms of seconds.

Lastly, a uniform distribution is implemented to simulate a crosswalk signal such that the probability of a pedestrian encountering a particular signal follows a uniform distribution. Essentially, a pedestrian approaching an intersection has 50% probability of seeing a do-not-cross signal and a 50% probability of seeing a cross signal and the wait time for the do-not-cross direction that he or she faces will be a uniform distribution between zero and the max wait time. Of course, like in reality, if at an intersection, one crossing is showing a do-not-cross signal, the perpendicular crossing is free to cross at.

Several different individual agent models were created to model different behaviors, as outlined below.

2.1 Basic Model

The Basic Model represents the most simple path a pedestrian could take to traverse a grid. A pedestrian will walk every single width first and then every length, or vice versa. This can be viewed as someone who travels along the "edges" of the grid without regard for street lights.



As we can see above, a person following this method will encounter a number of lights equal to the number of lengths and widths, all along the edges. At each of these lights, they have a 50% probability of encountering a cross signal, allowing them to cross, and a 50% probability of encountering a do-not-cross signal. Given that the pedestrian has reached a do-not-cross signal, the amount of time remaining that they must wait to cross is selected uniformly from zero to the maximum possible length of a signal. These probabilities and distributions are described below:

P(immediate cross signal) = 0.5
P(waiting) = 0.5
X is time left to wait so,
X = U(0, maximum light length)
Expected wait time at a random light = 0.5 * $\frac{maximumlightlength}{2}$

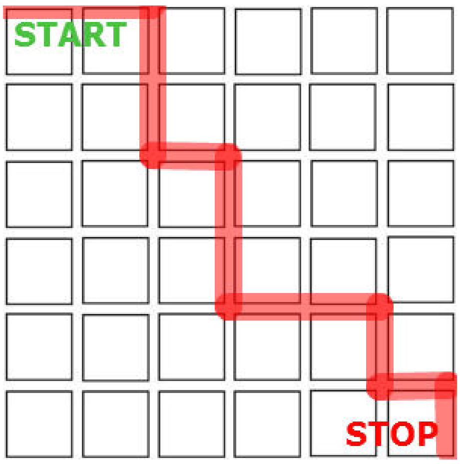
While this method may be the easiest option in terms of navigation, there are some pitfalls that become evident as we look at the individual agent model. With no care for the current signal the pedestrian has come across, we can envision that this pedestrian will spend a significant amount of time waiting at signals.

Since all direct paths have the same distance and guiding software has an incentive to suggest as few turns as possible (Jiang 1), this would be the most likely suggestion from direction generating software (GPS, Google Maps, etc.)

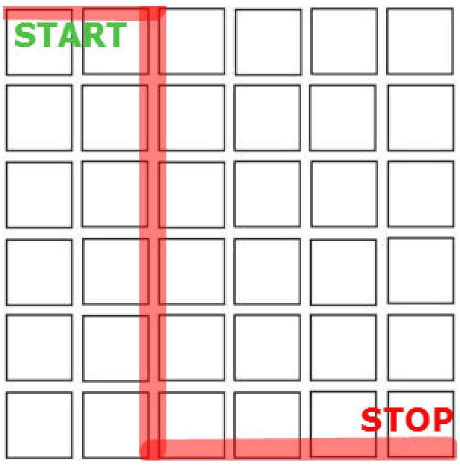
2.2 Naive Model

The Naive Model takes a far different approach from the Basic Model that attempts to optimize the pedestrian's behavior at a very rudimentary level. The Naive Model operates on the premise that should a pedestrian encounter a cross signal, he should cross in that particular direction, and should a pedestrian encounter a do-not-cross signal, he should cross in the perpendicular direction if possible (assuming he has not become blocked at an edge). This method attempts to optimize the path by crossing at every opportunity available.

Below is a sample path that a naive walker may take:



While this method may succeed in minimizing time in the beginning of a journey, there are possible situations where we can envision this method failing and resorting to the conditions prescribed in the Basic Model. Should a pedestrian in the Naive Model reach an edge (the right and bottom edges in the diagram below), he must wait at any do-not-cross signals that he encounters on the rest of his journey.

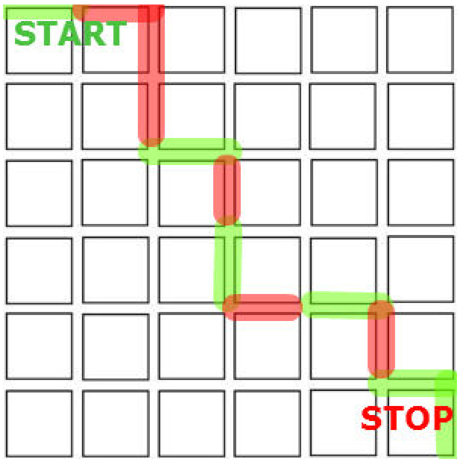


So, while the Naive Model seems like an improvement over the Basic Model, there are certain cases where the two behave very similarly.

2.3 Biased Model

The Biased Model attempts to optimize pedestrian behavior even further by biasing a pedestrian to avoid the edges. By maintaining a path that stays towards the middle, or diagonal, of the grid, the pedestrian can hopefully avoid the pitfalls in the Basic Model and Naive Model where he gets trapped along the edges and is forced to wait at all remaining do-not-cross signals. The Biased Model takes a new parameter, bias, that represents how many seconds a pedestrian is willing to wait at a do-not-cross signal for it to turn into a cross signal (an underlying assumption here is that the pedestrian knows how much time is left on the signal, and can thus view that information and make an instantaneous decision).

Below is a sample path that a pedestrian under the Biased Model could use. Here, the green represents a segment that is not affected by the bias and is made with no bias, while a red segment is a decision that is influenced by the bias as it tries to push the pedestrian away from the edges.



The probabilities utilized by this model are described below:

P(cross lengthwise | lengths = widths) = 0.5
P(cross widthwise | lengths = widths) = 0.5

P(cross widthwise | lengths remaining < widths remaining) = 0.5
P(wait and cross widthwise | lengths remaining < widths remaining) = 0.5 (bias / maximum light time)

P(cross lengthwise | widths remaining < lengths remaining) = 0.5
P(wait then cross lengthwise | widths remaining < lengths remaining) = 0.5 (bias / maximum light time)

P(wait-to-cross | stuck on edge) = 0.5

The Biased Model is based on a reasonable intuition; however, the bias is consistent throughout the entire time a pedestrian is navigating a grid. Under the Biased Model, a small deviation from the middle path is equivalent to being one step away from a catastrophic edge case.

2.4 Skewed Model

The Skewed Model builds on the Biased Model by changing the bias value dynamically throughout the simulation. Similar to the Biased Model, the Skewed Model attempts to stay away from the edges and remain in the middle, or diagonal, of the grid where the pedestrian's traversal options are maximized. The Skewed Model adjusts the bias value, however, by increasing it as the pedestrian approaches the edges and decreasing it as the pedestrian remains in the middle of the grid. The Skewed Model will take an additional parameter that will adjust the weight of the newly calculated location-dependent bias.

The probabilities utilized by the Skewed Model are outlined below:

P(cross lengthwise | lengths remaining = widths remaining) = 0.5
P(cross widthwise | lengths remaining = widths remaining) = 0.5

P(cross lengthwise | widths remaining < lengths remaining) = 0.5
P(wait then cross lengthwise | widths remaining < lengths remaining) = 0.5 * ((lengths / (lengths + widths) -0.5) / maximum wait) * skew
P(cross widthwise | lengths remaining < widths remaining) = 0.5

P(wait and cross widthwise | lengths remaining < widths remaining) = 0.5 * ((widths / (lengths + widths) -0.5) / maximum wait) * skew

P(wait-to-cross | stuck on edge) = 0.5

The Basic, Naive, Biased, and Skewed Models are all seemingly reasonable methods of modeling a pedestrian's behavior in a gridded system. Now, to truly test their abilities to optimize the pedestrian's time, we manipulate the parameters of the individual agent model to garner useful insights.

3 Analysis of Models

3.1 Conditions

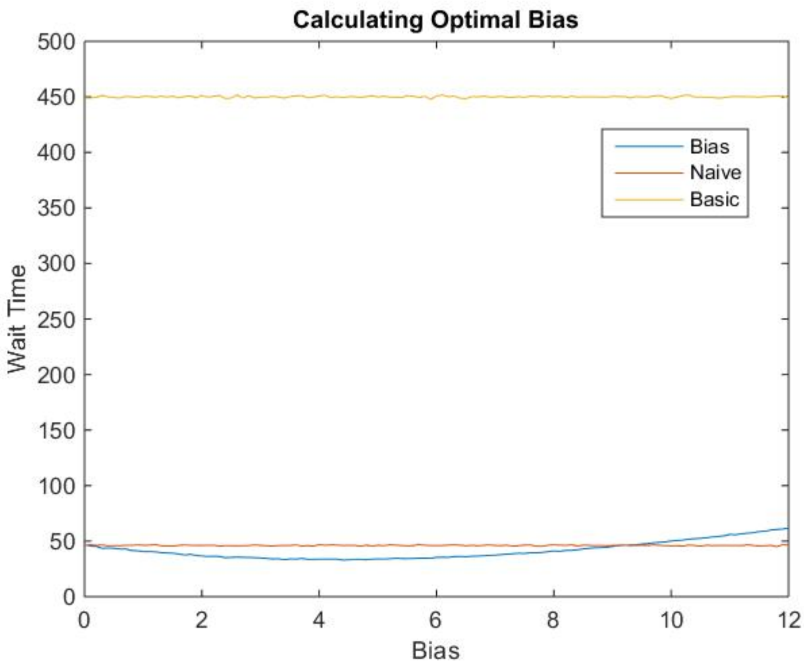
We ran the simulation on a 30 by 30 grid with 30 second lights at each intersection. This is a fairly reasonable construction given real life conditions.

3.2 Optimal Bias

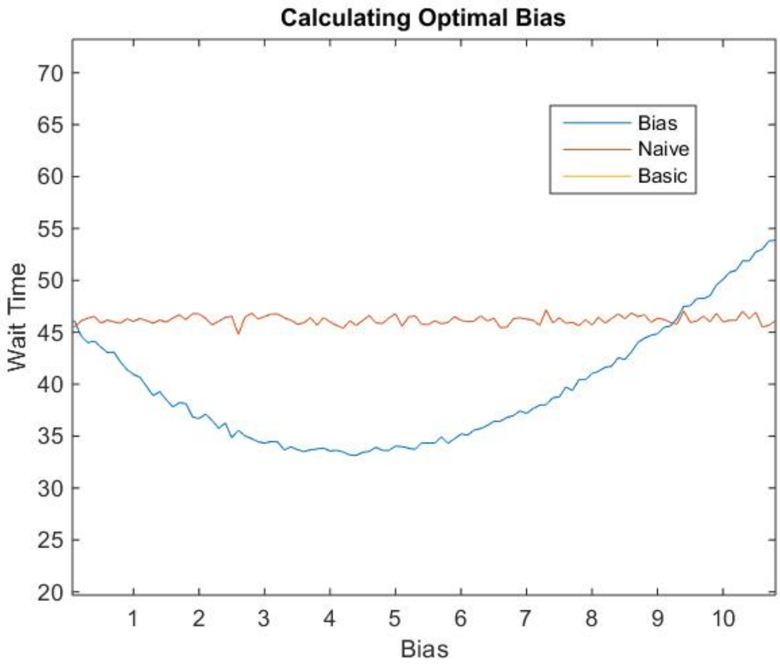
To understand how to use the bias to optimize travel time, we try to find the optimal bias by plotting the additional wait necessitated by the Basic Model, Naive Model, and Biased Model over a range of biases.

Note: the wait time is calculated as the total amount of time spent waiting at do-not-cross signals.

Iterating over possible biases generates the following figure:



Zooming in on the Biased and Naive functions reveals the following plot:



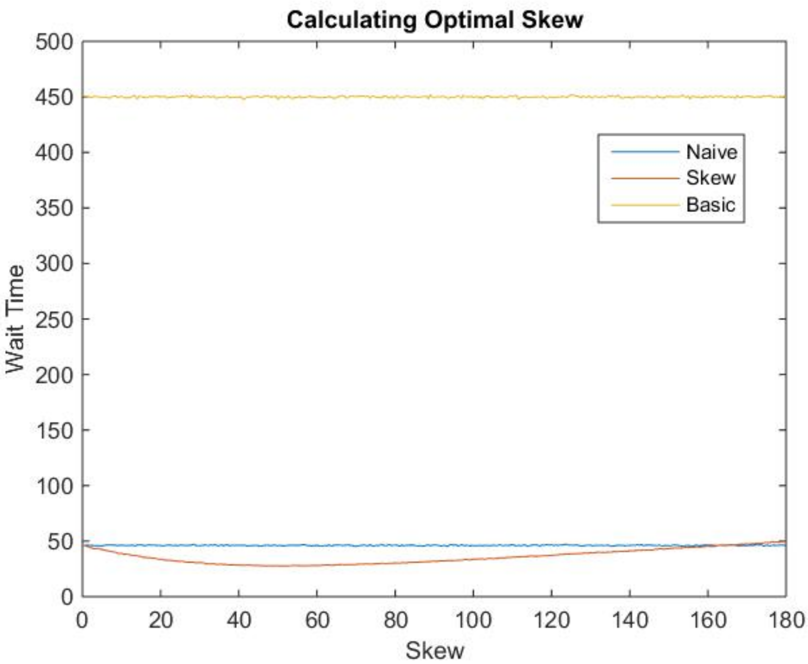
The graph yields the following information:

Naive average value: 46.182
Basic average value: 449.694
Bias minimum: 33.1405 where bias is 4.4

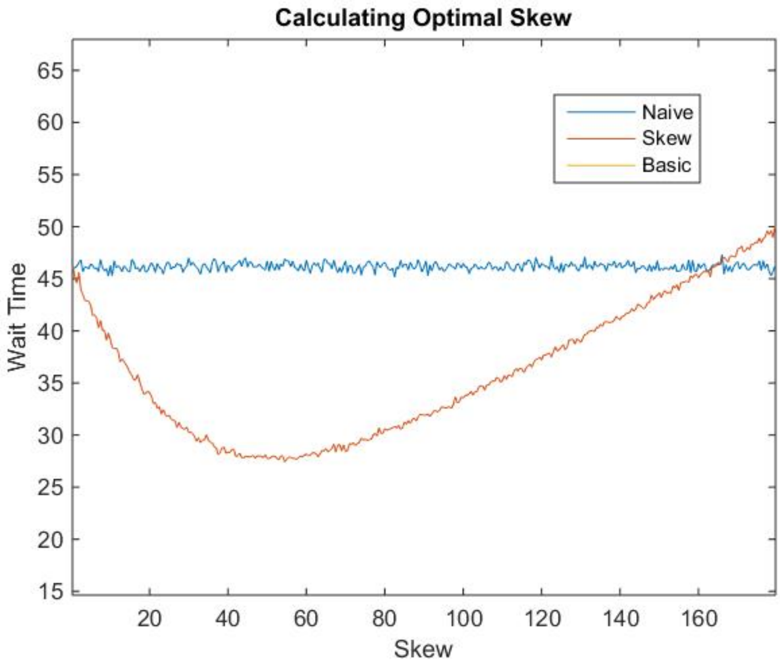
What we see here is that a bias value of around 4.4 generates the best walking strategy and actually makes the biased walker better on average than the naive walker. Once the bias value goes above the realistic limitations of the bias value (the bias value cannot be greater than the maximum wait time at a do-not-cross signal), we see a biased walker that is waiting extra time at a red light.

3.3 Optimal Skew

Next, find the optimal skew for the same grid conditions by performing a similar iteration over possible skews values as was done for bias values. This creates the following results:



Zooming in on the Biased and Naive functions reveals the following plot:



The graph yields the following information:

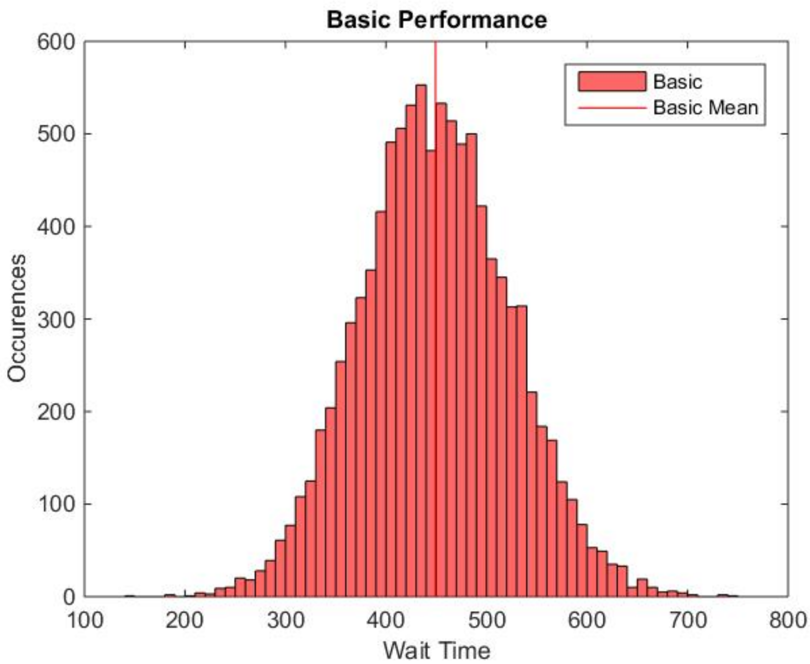
Naive average value: 46.1667
Basic average value: 450.03025
Skew minimum: 27.4122 where skew is 54.5

Once again, we see that there is an optimal skew value of 54.5 that results in a faster path than in the naive case.

3.4 Basic Model Performance

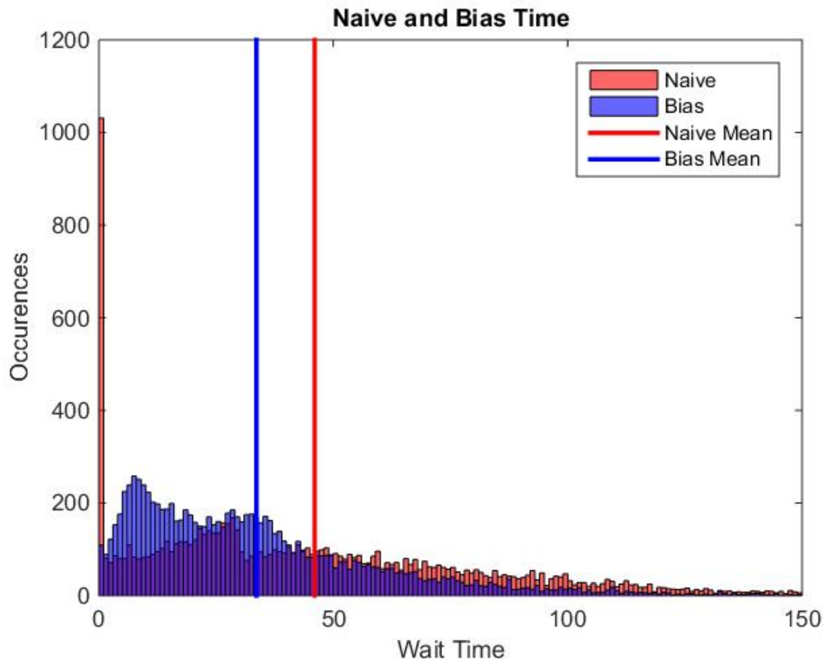
Now, using these best-performing bias and skew values, we can see a distribution of the wait times generated by these parameters by running numerous trials with the optimal bias and skew values implemented.

First, the Basic Model produced the figure below, which describes the number of occurrences for wait times after 10,000 trials:



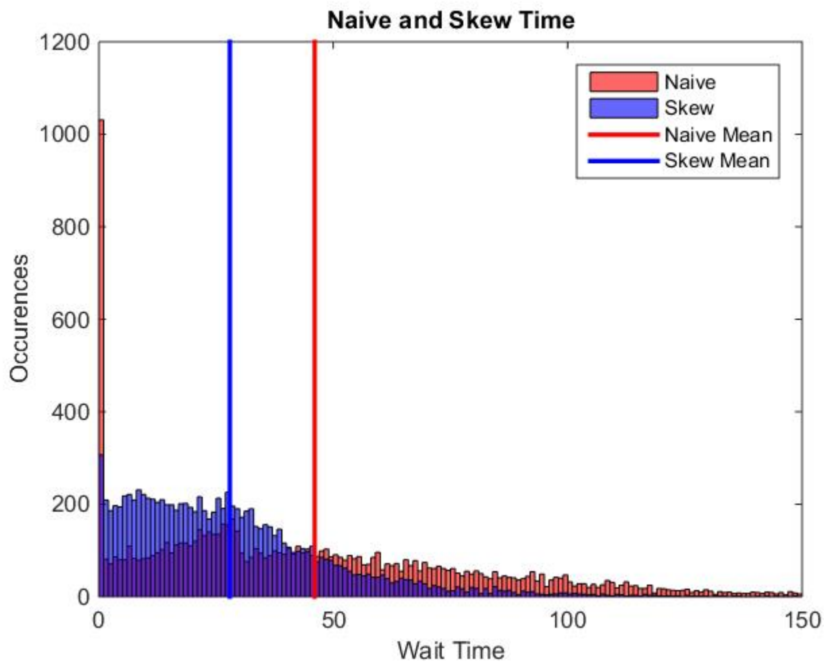
3.5 Biased Model and Naive Model Performance

Next, the Biased Model using the optimal bias and the Naive Model produces the following distributions of wait times:



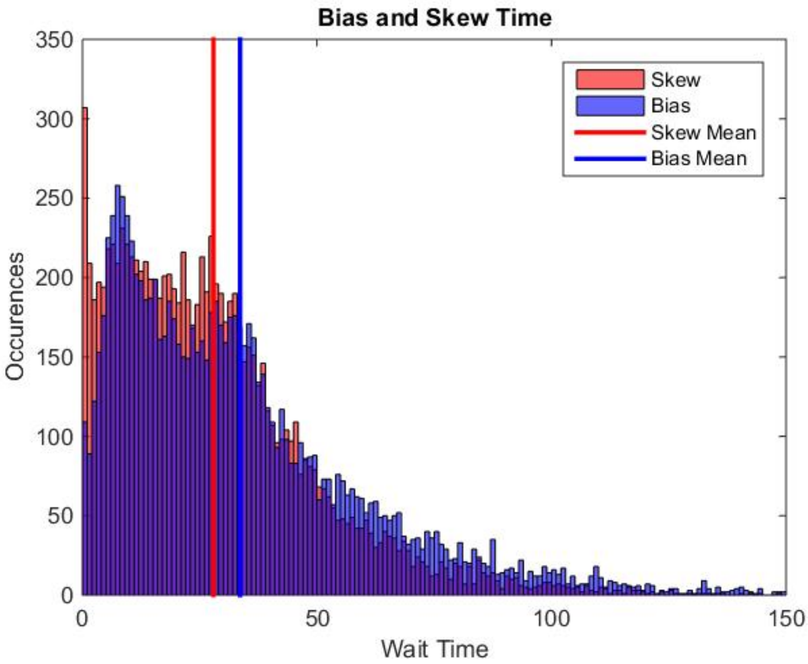
3.6 Skewed Model and Naive Model Performance

Lastly, the Skewed Model using the optimal skew and the Naive Model generates the following distributions of wait times:



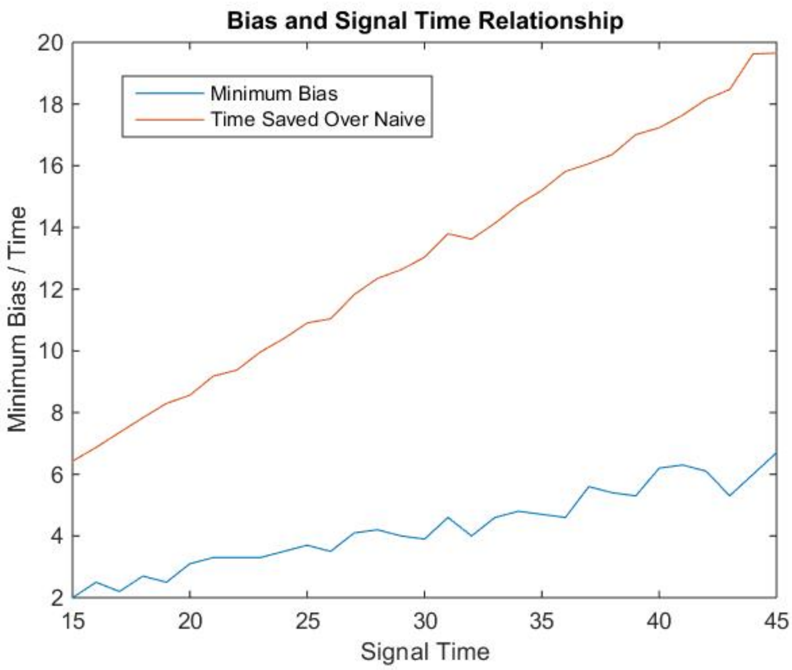
3.7 Biased Model and Skewed Model Comparison

Visualizing the Biased Method using the optimal bias method and the Skew Method using the optimal skew value relative to each other produces the following results:



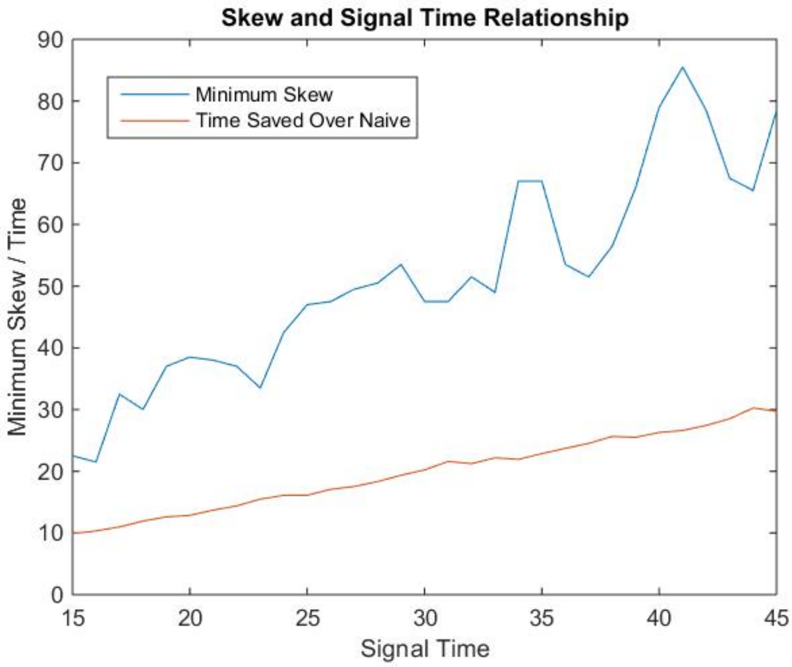
3.8 Bias and Signal Time

How are bias and the length of time for the cross signal related? To generate the plot below, using the same 30 by 30 grid, we iterated over signal time lengths from 15 to 45, incrementing by 1, and found the minimum bias. This was plotted alongside the amount of time that the Biased Model saved over the Naive Model.



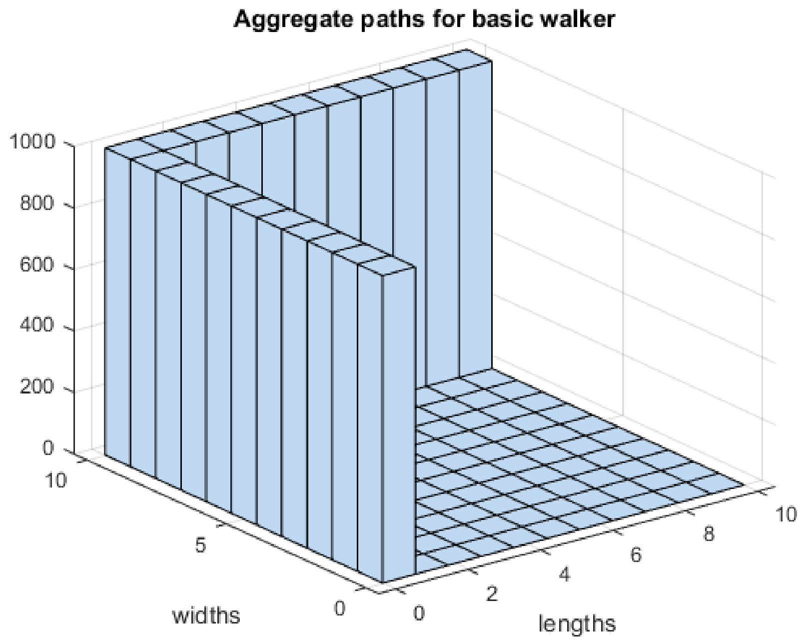
3.9 Skew and Signal Time

The same can also be done with the skew and the signal time.

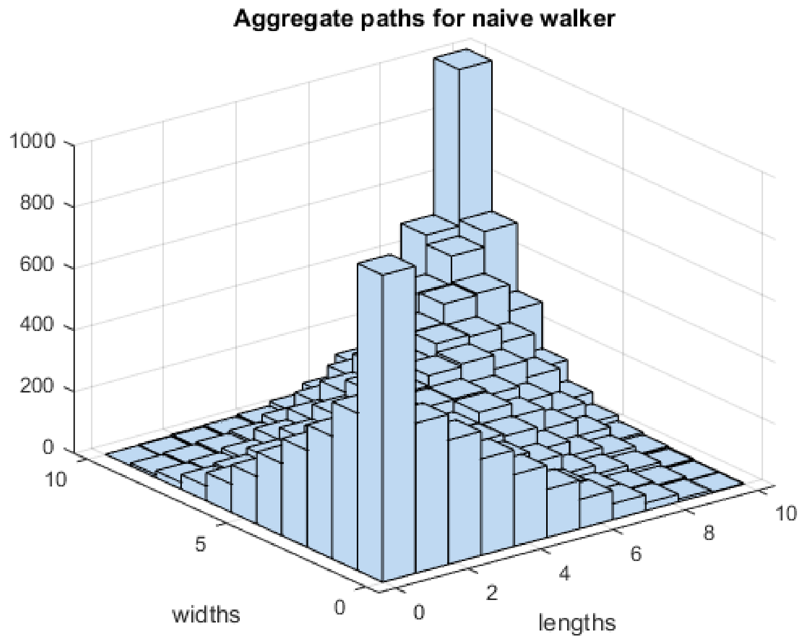


3.10 Aggregate Path - Basic Walker

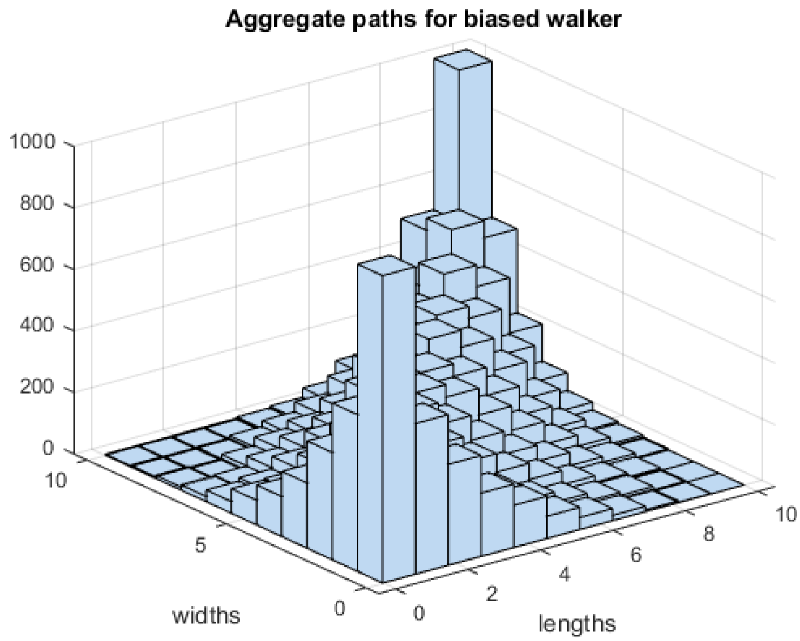
Functionally, what do these models look like? We visualized the differences in paths by creating 3D histograms that show the aggregate frequency of (width, length) pairs for each of the strategies, combining the results of 1,000 trials along a 10 by 10 grid, producing the following graphs:



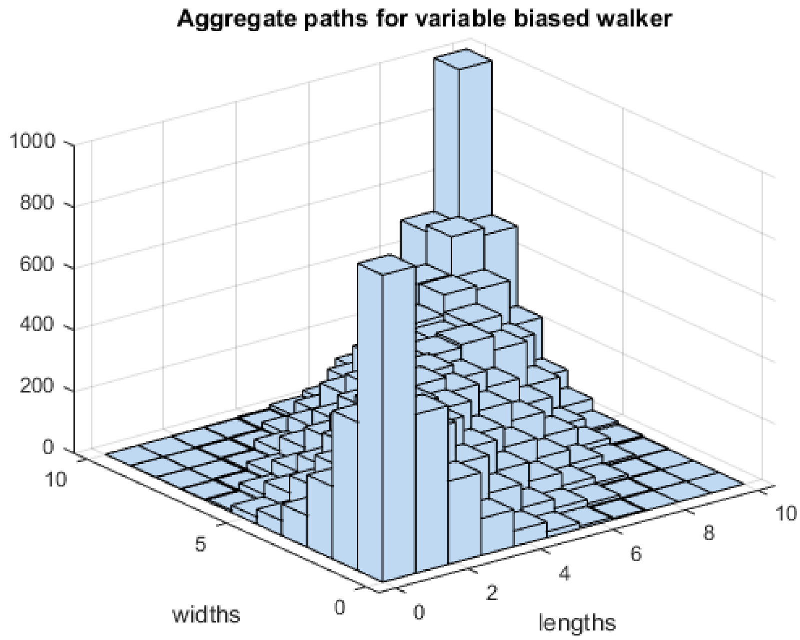
3.11 Aggregate Path - Naive Walker



3.12 Aggregate Path - Biased Walker



3.13 Aggregate Path - Skewed Walker



4 Discussion of Results

4.1 Figure 1 - Optimal Bias

The Biased Model had a lower mean than the Naive Model, and at its optimal bias of 4.4, saved the pedestrian approximately 13 seconds relative to the naive walker. Relative to the Basic Model, the Biased Model saved 416.55 seconds, or 6.94 minutes, and the Naive Model saved 403.512 seconds, or 6.73 minutes. Thus, the Biased Model can be a more optimal method than the Naive Model and Basic Model.

4.2 Figure 2 - Optimal Skew

The Skewed Model had a lower mean than the Naive Model, and at its optimal skew of 54.5, saved the pedestrian approximately 18.75 seconds relative to the naive walker. Relative to the Basic Model, the Skewed Model saved 422.62 seconds, or 7.04 minutes. Like the Biased Model, the Skewed Model outperforms the Naive Model and Basic Model.

4.3 Figure 3 - Basic Performance

The Basic Model, when implemented, appears to follow a normal distribution centered around a mean that is 25% of the total possible wait time if a pedestrian waited at every single full do-not-cross signal. This result is consistent mathematically, if we recognize that a pedestrian has a 50% chance of stopping and the wait time for a stop is selected from a uniform distribution that has an expected value of half the maximum wait time. So, the expected total wait time should be half of half of the maximum total wait time (which in the case of the 30 by 30 grid with 30 second lights is 1800 seconds). Thus, we can see the mean wait time is around 450 seconds.

4.4 Figure 4 - Biased and Naive Comparison

The Biased Model evidently generally performs better than the Naive Model. The Biased Model had a lower mean than the Naive Model when all the trials are taken into account, implying that the Biased Model will probably take less time than the Naive Model on any given traversal. The Naive Model had a significant number of very short traversal times, as shown by the large bar near 0, but had a much longer tail, implying that while the Naive Model can certainly be extremely successful, it can also fail dramatically in certain cases. The Biased Model has a shorter right tail and does not have a dramatic peek, meaning it is a more consistent method with fewer catastrophic failures.

4.5 Figure 5 - Skewed and Naive Comparison

The Skewed Model generally performs better than the Naive Model. The Skewed Model has a lower mean than the Naive Model when all the trials are taken into account, implying that the Skewed Model will generally take less time than the Naive Model. The Skewed Model has a much shorter right tail than the Naive Model and most of its trials are clustered in the region less than 50. Similar to the Biased Model, the Skewed Model avoids the catastrophic failures of the Naive Model.

4.6 Figure 6 - Biased and Skewed Comparison

After having compared the Biased Model and Skewed Model to the Naive Model, we know that the Biased Model and Skewed Model perform better in general. When we compare these two methods, however, we can see that the Skewed Model is slightly better than the Biased Model. A possible reason for this behavior is that the Skewed Model only makes the pedestrian wait significant amounts of time if he is very close to the right or bottom edge. Whereas the Biased Model will possibly make a pedestrian wait if he has deviated at all from the diagonal, the Skewed Model will allow a pedestrian to wander slightly around this diagonal but will harshly penalize the pedestrian for getting to close to a catastrophic edge.

4.7 Figure 7 - Bias and Signal Relationship

The relationship between bias and the performance of the Biased Model seem to be a linear relationship. The graph implies that as the maximum signal time doubles, the minimizing bias also doubles, and the improvement over the Naive Model doubles as well. This implication carries significant meaning: the bias should always be the same percentage of the maximum signal time if you wish to minimize the traversal time in the square grid.

4.8 Figure 8 - Skew and Signal Relationship

The relationship between the skew and the performance of the Skew is very similar to that of the Biased Model. Doubling the light times necessitates a doubling of the skew in order to minimize traversal time. Like bias, skew should always be the same percentage of the maximum signal time if you wish to minimize the traversal time in the square grid.

4.9 Aggregate Paths

From the aggregate paths we can see that a pedestrian under the Basic Model will follow the same path regardless of the signals he encounters - this is expected and was shown above to be a poorly performing navigation strategy. Interestingly, the Naive Model, Biased Model, and Skewed Model all have very similar location distributions. What we observe is the confirmation that our simulated path follow our expectations that biased walkers will more likely follow a central axis from origin to destination. If done with a reasonable amount of bias, we should have the improvements demonstrated above. Essentially, we notice that a naive walker is more likely to reach an undesirable corner of the grid. The Biased Model makes small improvements in forcing pedestrians away from these corners, and the Skewed Model does an even better job of making sure pedestrians refrain from reach these corners. From these models we can see that the best way to optimize a pedestrian's path is to maintain a path along the central axis to a reasonable degree.

4.10 Assumptions and Limitations

First of all, we assume that all signals are random in this environment. Specifically, this means that when we approach a crossing, there is a perfect uniform distribution for possible wait times. However, one possibility we neglect is the intentional timing of stoplights. This includes the ability of green lights to be spaced at even walking intervals (i.e. once you cross in one direction, the next light should turn green in the same direction once you arrive) or the ability of your arrival to affect the stoplight (some stoplights have buttons you can press that affect the timing of the lights) and being able to see future stoplights (you can potentially adjust your trajectory if you can see what future stoplights are presently set to be). all of these possibilities were not included in our simulation.

Second of all, we assume that the walker has a fixed speed. If this does not hold, then we cannot assume that it would take him the same time to traverse a length or a width. At best, we assume that the numbers used reflect average speed.

We further assume a perfect grid, assuming that all length and width crossings are the same. This will not be the case in most environments, where block lengths may vary from block to block. We chose not to simulate this situation because it would have required a much more intensive array and much more room for error. Furthermore, this assumes perfectly rectangular blocks. The benefit of this structure is that as long as the walker goes in the right direction, the distance traveled will be the same. If a similar model was implemented in Boston, we would need to account for the total travel time in addition to the crossing times, which would be more computationally intensive and follow more closely algorithms such as Djikstra's algorithm (with expected waits rather than simulated waits). In this scenario, the results would differ depending on the relationship between walking speed and stoplight times. For example, optimal paths for faster walkers might involve more distance but less stoplight waits than slower walkers.

4.11 Future Exploration

As many of the assumptions and limitations above imply, there is much room for improvement in the individual agent model constructed in this report. In future models, the grid shape and size could vary greatly to see the impact of the grid dynamics on the pedestrian behavior. Additionally, more realistic pedestrian behavior could be accounted for, such as potential jaywalking and aggressive individuals. Specifically, traffic plays a key role in pedestrian behavior and is probably a good indicator of when pedestrians will cross a street in addition to the cross signal. In a system as complex as a city, there a multitude of interacting factors that can improve and change the individual agent model.

5 Summary

We see now that pedestrian street crossing patterns have varying degrees of efficacy, but that at the very least, the presence of a flexible strategy saves time. This is most evident when we examine the large disparity between the basic walker and the naive walker. Walking in two straight lines down the width and length of the grid you are traversing and walking in the simplest path possible results in a significant wait time. Even when we adopt the simplistic strategy of crossing immediately when possible, we find that we dramatically cut down the wait time. Furthermore, there does not appear to be significant improvement of this wait time when we further optimize our crossing strategy. There is certainly improvement, but the reason the improvement is not as dramatic is that there is simply not as much room to improve. The naive walker's strategy is already sufficient in cutting down the wait time.

What does this mean for optimizing crossing strategies? It means that adopting a crossing strategy of any kind of the most important thing. It is less important to find an optimal crossing strategy - as long as our walker has some sensible rationale for how he is crossing the road. Furthermore, the shorter the time it takes the physically traverse the actual distance from your start point to your end point, the more important the optimal crossing strategy becomes. For example, someone who is running from one place to another might find that the optimal crossing strategy which minimizes wait time will actually make a significant difference. On the other hand, the pedestrian who is slowly walking does not really see the same kind of benefit as the requisite walking time is already incredibly long.

6 Appendix of Matlab Code

6.1 basicwalker.m

Implements the Basic Model

6.2 naivewalker.m

Implements the Naive Model

6.3 biasedwalker.m

Implements the Biased Model

6.4 varbiasedwalker.m

Implements the Skewed Model

6.5 trials.m

Simulates many trials of the models and plots a distribution of the wait times

6.6 optimal_bias.m

Iterates over possible bias values to find the optimal one

6.7 optimal_skew.m

Iterates over possible skew values to find the optimal one

6.8 bias_light_relationship.m

Plots the relationship between bias and the length of a signal

6.9 skew_light_relationship.m

Plots the relationship between skew and the length of a signal

6.10 aggregatepaths.m

Plots the aggregate paths taken utilizing different strategies (requires modified code for walkers)

6.11 trials_for_aggregate.m

Creates aggregate (x,y) locations for each strategy

6.12 basicwalker_mod.m, naive_walker_mod.m, biasedwalker_mod.m, varbiasedwalker_mod.m

Implements same models, with outputs for aggregatepaths.m

7 References

Azad, Kalid. "Navigate a Grid Using Combinations And Permutations."BetterExplained. N.p., n.d. Web. 08 Nov. 2015.

Jain, Akash, Ankit Gupta, and Rajat Rastogi. "Pedestrian Crossing Behaviour Analysis At Intersections." Ijtte International Journal For Traffic And Transport Engineering 4.1 (2014): 103-16. Web.

Jiang, Bin, and Xintao Liu. "Computing the Fewest-turn Map Directions Based on the Connectivity of Natural Roads." International Journal of Geographical Information Science: 1069-082. Print.

Marisamynathan, Vedagiri Perumal, Study on pedestrian crossing behavior at signalized intersections, Journal of Traffic and Transportation Engineering (English Edition), Volume 1, Issue 2, April 2014, Pages 103-110, ISSN 2095-7564, http://dx.doi.org/10.1016/S2095-7564(15)30094-5.

Therakomen, Preechaya. The Experiments for Exploring Dynamic Behaviors in Urban Places. Thesis. 2001. N.p.: n.p., n.d. Print.

Weisstein, Eric W. "Staircase Walk." From MathWorld–A Wolfram Web Resource. http://mathworld.wolfram.com/Stairc

8 Attribution

All group members played a role in developing the individual agent model, writing the Matlab code, and writing up the results.