Blocking Sidewalk Deadlock

A Short Story by

Jerry Sun (js2746), Peter Wu (plw53), Kevin Zhou (klz23)

for the Cornell Mathematical Contest in Modeling

18 November 2019

Dear Mr. John Licitra,

Our team understands the vital role that sidewalks play in our incredibly walkable City of Ithaca, and we have developed several improvements to the current Sidewalk Improvement Program (SIP) by which to better allocate limited resources.

Knowing that there are limited resources, the SIP team must prioritize which locations to repair first in order to have the most impact on pedestrian safety and sidewalk attractiveness. Thus, our team has developed a new model which utilizes the current model for prioritization but with additional insightful features. Using this model, the SIP team can not only prioritize the most populated yet damaged sidewalks, but also minimize costs by leaning towards prioritizing repairs toward clustered blocks.

Our team understands the importance of minimizing cost, and so we've developed an algorithm that gives the cost-minimizing repair strategy for each sidewalk slab on a block. The algorithm also ensures that after repair, all of the slabs will meet the necessary requirements, including those of the ADA. We hope that this algorithm will help the City of Ithaca save necessary resources and improve on the current high quality and effectiveness of our repair program.

We have also been aware of the imminent budget issues, recently referenced in the memo addressed to the Common Council and Board of Public Works. In an effort to allow for more efficient planning and levying of fees in the foreseeable future, our team has devised a model which, provided an input of a number of years after 2015, results in a predictive budget required for the continued effectiveness of the program in the corresponding year. We believe that, through the use of insights from our models and strategies, the SIP will be effective in allowing all to walk the streets of gorgeous Ithaca for years to come.

Sincerely,
Jerry Sun, Peter Wu, Kevin Zhou

Table of Contents

1 Introduction

Sidewalks are an essential component to any city's prosperity and especially to that of Ithaca where 42% of people walk to work [11]. High quality sidewalks provide health and safety benefits ranging from attractive facilities to walk and exercise to increasing the distance between pedestrians and vehicles, decreasing the number of potential accidents. With Ithaca's successful Sidewalk Improvement Program (SIP), a lengthy amount of sidewalks have been improved with property-owners contributing to the task force's budget, a much needed improvement over the old system where home-owners would have to pay for repairs to their own sidewalk. However, factors ranging from increasing construction bid prices to flat revenues have slowly whittled away at this program's efficacy.

Thus, at the heart of our models, we will focus on the tasks of improving sidewalk quality as well as decreasing the cost-to-revenue ratio. The latter can take the form of either decreasing the operation costs of improving sidewalks or generating additional revenue at marginal cost to property-owners.

2 Global Definitions

City Block: A city block is the smallest area that is surrounded by streets.

Sidewalk: The most common sidewalk is the ribbon sidewalk and in accordance with the most common sidewalk dimensions permitted under New York governance, the dimensions of each slab are 5ft x 5ft and 4 inches deep [4][5]. This is also in accordance with several of the dimensions given by the 2019 SID memo as well as the requirements set by the ADA [7]. Thus, for the sake of consistency, we will utilize this definition across all sidewalks in Ithaca.

Percent Difference in Slope: We define percent differences in slope (used when dealing with running and cross slopes of each slab) as grades. This is what the ADA defines their requirements as, thus we will utilize grades as the measurement rather than percent difference with respect to the road.

3 Models

3.1 Part A: Priority My Disparity

The task at hand is to create our own algorithm to generate a given priority score for each of the blocks within Ithaca that need sidewalk repairs using the various factors that the city already considers as well as some new improvements to this ad hoc model. With our goals in mind, we will consider both criteria that help minimize costs in addition to those that evaluate the impact of the location and quality of the block's sidewalk in question.

3.1.1 Local Assumptions

- The road's slope and each concrete slab's running and cross slope are provided since in order to comply with ADA requirements, these measurements at some point would have been measured in order to obtain the percent difference.

3.1.2 Sidewalk Quality Score

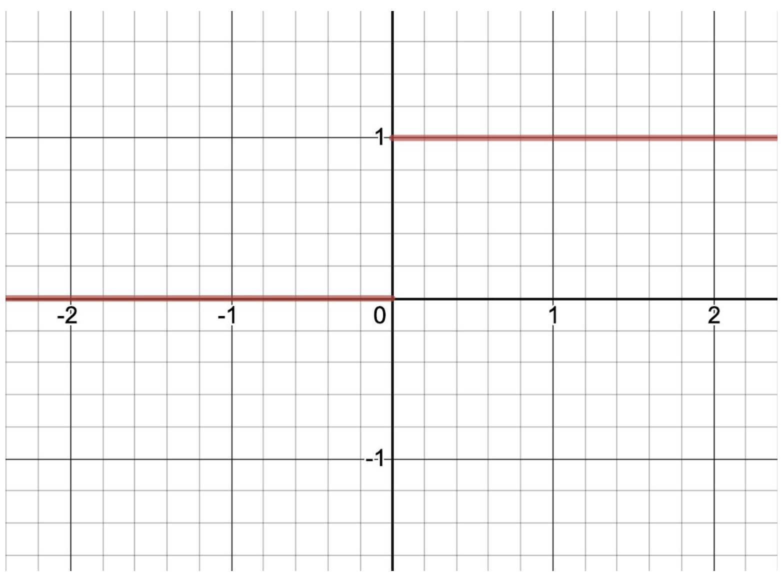
Sidewalk impact can be fractured in two categories as follows: Quality of Sidewalks and Location of Most Impact. We will begin by developing an algorithm to assess the quality of any given block's sidewalks and how necessary repairs are.

Currently, there are 5 requirements given both by the City of Ithaca and the ADA for sidewalk quality given as:

- (1) Unbroken slabs
- (2) $\geq$4ft wide slabs
- (3) Vertical displacement (D_v) $\leq$ 0.5 inch slabs
- (4) Running slope (S_r) $\leq$ $\pm$2% grade difference from road slope
- (5) 1% $\leq$ | Cross slope (S_c) | $\leq$2%.

In our model, we assign penalty (severity) scores from 0-100 depending on how severe these violations are. For criterias (1) and (2), we will assign a Binary Step Activation Function to their values (either 0 or 100) since these criterias have two degrees of noncompliance as shown in Figure 1. This will generate $Score_1$ and $Score_2$.

Figure 1: Binary Step Activation Function



For criterias (3) - (5), we would not want to penalize extreme violations proportionally more after a certain brightline since any point past that brightline is still regarded as a severely damaged slab, thus making any differentiation between the two irrelevant. For example, a very battered and extremely tilted slab is regarded only marginally less worse than a utterly obliterated slab for the fact that both are unwalkable and unsafe to a high extreme either way. Otherwise, the model would run the risk of over-prioritizing the extremely damaged slabs over other considerations. Thus, we will utilize sigmoid functions in order to capture this phenomenon throughout criteria (3) - (5).

The brightline values were calculated for (3) - (5). In the case of (3), the brightline was seen to be $D_v = 4$ inches as each slab's thickness is only 4 inches. This by no means is necessarily the maximum D_v as tree roots and soil erosion can elevate the slab off the ground completely. From this brightline we can utilize Python's SciPy library to fit a sigmoid curve with the features of not penalizing anything less than 0.5 inches and only marginally increasingly penalize values above 4 inches. By running the code seen in Appendix A-C.

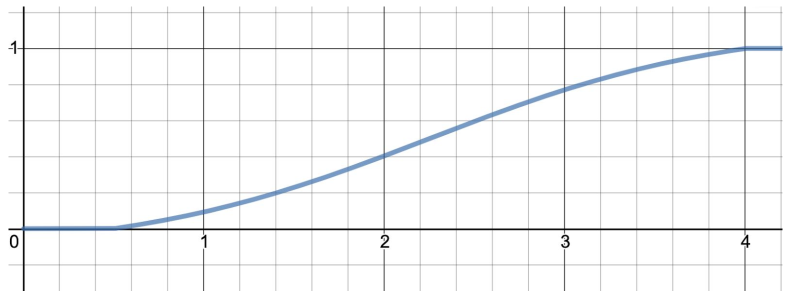
$$D_v \text{ Penalizer } (D_v) = \frac{1.27143152}{1+e^{-1.21397466(D_v - 2.25)}} - 0.13571576 \qquad \text{(Eq 1)}$$

We then create a Vertical Displacement Score (VDS) as a piecewise function of D_v in order to encapsulate the full scope of the domain and create Figure 2:

$$VDS = \begin{cases} 0, 0 \leq D_v \leq 0.5 \\ \frac{1.27143152}{1+e^{-1.21397466(D_v - 2.25)}} - 0.13571576, 0.5 < D_v < 4 \\ 1, D_v \geq 4 \end{cases}$$

(Eq 2)

Figure 2: Piecewise Function for VDS



This will give us a score of 0 - 100. In order to determine the VDS for an entire block we sum the scores of all the slabs and divide by the total number of slabs given as TS:

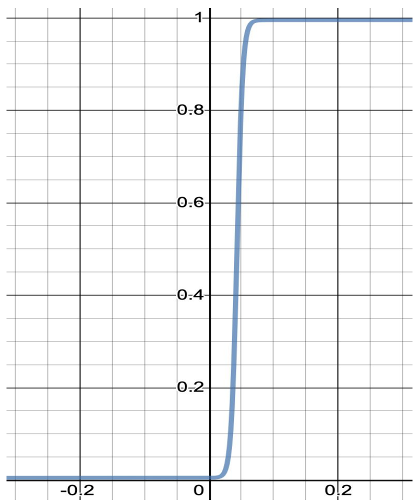
$$\text{Score}_3 = \left(\sum_{i=0}^{Number\ of\ Slabs} VDS\,(\text{D}_{\text{vi}})/\text{TS}\right)*100 \qquad \text{(Eq 3)}$$

In the case of (4) the brightline was found from following a similar logic to (3). The difference in grade between any road and a slab of concrete should be considered fairly severe once it reaches the point where one end of the slab rests entirely on top of its neighboring slab, a height of 4 inches with a length of 60 inches. The difference in the grade between any road and a concrete slab that differs from the slope of the road by 4 inches is 6.68%. Following a similar process as before we can calculate Eq (4) and (5) as well as Figure 3:

$$\text{S}_\text{r}\ \text{Penalizer}\ (\text{S}_\text{r}) = \frac{-0.990112473}{1+e^{225.193726(S_r - 0.043411293)}} + 0.995056236 \qquad \text{(Eq 4)}$$

$$\text{Score}_4 = \left(\sum_{i=0}^{Number\ of\ Slabs} S_\text{r}\ \text{Penalizer}(\text{S}_{\text{ri}}) \div \text{TS}\right)*100 \qquad \text{(Eq 5)}$$

Figure 3: Sigmoidal Function for S_r Penalizer



For criteria (5) this process yet again repeats with finding the brightline. The brightline can be determined from the previous logic that the tilt of grade 6.68% is enough to surpass 4 inches which is fairly severe. Thus, the equation for $S_c > 0.02$ of the piecewise function for (5) will be the same as (Eq 4). For when $S_c < 0.01$, the equation will simply be mirrored producing:

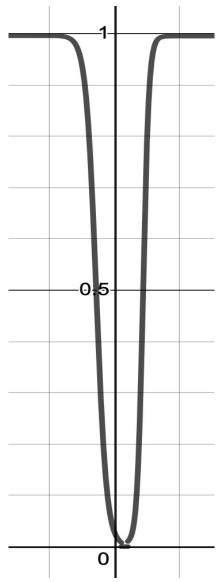
$$S_c \text{ Penalizer } (S_c) = \frac{0.995318329}{1+e^{127.746484(Sc+0.0285198344)}} + 0.000867219272 \quad \text{(Eq 6)}$$

$$CSS = \begin{cases} \frac{0.995318329}{1+e^{-127.746484(Sc-0.0285198344)}} - 0.000867219272, -1 \leq S_c \leq 0.01 \\ 0, 0.01 \leq S_c \leq 0.02 \\ S_r Penalizer(S_c), S_c > 0.02 \end{cases}$$

(Eq 7)

$$\text{Score}_5 = \left(\sum_{i=0}^{Number\ of\ Slabs} S_c \text{ Penalizer}(S_{ci}) \div \text{TS}\right)*100 \quad \text{(Eq 8)}$$

Figure 4: Piecewise Function for S_c



Now each criteria (1) - (5) have a score of 0 - 100 which will form the Numerical Score vector:

$$NS: \langle Score_1, Score_2, Score_3, Score_4, Score_5 \rangle$$

The weight vector of

$$W: \langle w_1, w_2, w_3, w_4, w_5 \rangle$$

have been determined to be a factor of 0.2 each in order to equalize the importance of each of the criteria. The reason for not creating different weights is the fact that many of these criteria are not independent from each other and will have some correlation coefficient above 0, thus without being able to map out the physical values for those coefficients, we can not create weights that represent a proper “weighting” of criteria to determine which one impacts the quality of the sidewalk the most. Once performing the euclidean inner product on these two vectors, a singular score from 0-100 will be outputted resulting in the Sidewalk Quality Score (SQS):

$$SQS = \langle Score_1, Score_2, Score_3, Score_4, Score_5 \rangle \cdot \langle w_1, w_2, w_3, w_4, w_5 \rangle \quad \text{(Eq 9)}$$

3.1.3 Sidewalk Impact Score

Degree of Pedestrian Traffic criteria are as follows:

- (a) Population density
- (b) Proximity to schools, bus stops, and government buildings
- (c) Complaints.

We propose the addition of several other criteria:

- (d) Median Income
- (b) Proximity to restaurants, public parks, and grocery stores.

For lower income individuals, they are less likely to own a car and thus will be walking to work and other public facilities more often, therefore using sidewalks more [1]. Thus, these areas should be prioritized more in contrast to higher income-individuals who are more likely to drive.

The demographic that the original criteria did not factor in are college students who venture from Cornell to dine in Ithaca; as a school that is comprised of over 20,000 students, this population is simply too large with respect to Ithaca's population to ignore in their impact in pedestrian traffic. This impact still exists regardless of the SIP program's exclusion of Cornell's sidewalks. Furthermore, other pedestrian heavy areas that the previous criteria do not factor in are blocks with Restaurants, Parks, and Grocery Stores. These areas are bound to have heavy traffic due to their popular nature and integration in daily life.

In calculating the score for (a), we will take the population density of the particular block being examined and divide it by the largest population density of the entire collection of blocks and multiply by 100. This results in $Score_a$ from 0-100. In calculating the score for (b), we wish to have an additive property for these different areas within the block, thus we will assign a value of 1 for each feature in (b) if present within the block and 0 if not present. We perform a summation and divide by 6 and multiply by 100 in order to produce $Score_b$. To achieve $Score_c$ we simply divide the number of complaints of a given block by the largest number of complaints received by a block and multiply by 100. For (d), there are 9 primary income brackets [2] and thus the scores will be given based on where one lies in the 9 brackets with scores in multiples of $11.\overline{1}$ to $99.\overline{9}$.

These results form another Score vector: $<Score_a, Score_b, Score_c, Score_d>$. In order to find the corresponding weight matrices, there are several observations and justifications to be made. Since (a) and (b) essentially map out pedestrian traffic, we can weigh these the same as they account for residential and non-residential traffic respectively for the most part. This can be assumed through Ithaca's zoning regions which indicate that for the most part, residential blocks do not contain public facilities and other buildings [15]. Thus we will combine scores $Score_a$ and $Score_b$, and take the average to form $Score_{a,b}$. Another observation is that individuals of higher education, and thus usually high income brackets, will have a higher tendency to file complaints in comparison to those with less education, and additionally lower income brackets[8]. Furthermore, low income brackets are generally living in higher population density blocks [9], thus $Score_{a,b}$ also accounts for some of $Score_d$. Complaints as a whole are subject to a variety of biases such as response bias as well as the fact that This leads us to believe that complaints should be weighted less and not ignored altogether, or penalized, for the reason that it's simply a correlation and can still be indicative of pedestrian traffic through the block. This category would also allow blocks without particular concrete slab quality data to still be relevant in the algorithm if there are a substantial number of complaints.

From here the weight vector: $<w_{a,b}, w_c, w_d>$ can be subjectively determined by the City of Ithaca depending on their evaluations of the priorities of these categories. We have decided to model the weights as <0.6, 0.1, 0.3>. The Euclidean Inner Product results in the Sidewalk Impact Score (SIS) with a range from 0 - 100 in the form:

$$SIS = Score_{a,b} \cdot w_{a,b} + Score_c \cdot w_c + Score_d \cdot w_d \qquad \text{(Eq 10)}$$

3.1.4 Cost Score

As given in Part B, transportation costs can easily amount to thousands of dollars, thus as one of our goals, minimizing costs should be able to be taken into account in order to not only save money but to improve more sidewalks with that money. We can measure the Cost Score (CS) via (Eq 9) by summing the product of the SIS and SQS of neighboring blocks and dividing the result by the total number of neighboring blocks (N).

$$CS = \mathtt{min}(SQS, SIS) \cdot \{ \frac{1}{N} \sum_{i=0}^{N} [(SIS_i + SQS_i) \div 2]\} \qquad \text{(Eq 11a)}$$

We multiply the summation by `min`(SQS, SIS) so that either of the two factors, SIS and SQS, will not comprise the majority of the final score. For example, if a block is surrounded by other blocks of high prioritization even though its own priority is low, then it should not be prioritized ahead of those with higher individual scores but lower neighboring scores. Thus by taking `min`(SQS, SIS), if there is a large disparity between the scores, it will lower the Cost Score. We take the average of the SIS and SQS scores for reasons that will be explained in 3.1.4.

3.1.5 Prioritization Score

The final step in calculating the Prioritization Score (PS) is as follows:

$$PS = \langle SQS, SIS, CS\rangle \cdot \langle w_{SQS}, w_{SIS}, w_{CS}\rangle \quad \text{(Eq 12)}$$

Here we have chosen subjective weights from which to prioritize each component: <0.4, 0.4, 0.2>. We have chosen these values since SQS and SIS should be of equal importance as they describe the quality of the sidewalk (how urgent a repair is) and the most impactful locations (where strategic repairs should be made). As alluded to in (Eq 11), the reason for averaging SQS and SIS was due to the relationship between their weights, if the weights changed then the equation would be modified as so:

$$CS = \mathtt{min}(SQS, SIS) \cdot [\ \frac{1}{N} \sum_{i=0}^{N} (SIS_i \cdot \frac{w_{SIS}}{w_{SQS} + w_{SIS}} + SQS_i \cdot \frac{w_{SQS}}{w_{SQS} + w_{SIS}})] \quad \text{(Eq 11b)}$$

Furthermore, CS should not be on equal weighting since doing so might incentivize the algorithm to trivialize isolated blocks of terrible quality sidewalks. We include CS as a measure of reducing cost but not so much as to affect the PS to a large degree.

3.1.6 Strengths and Weaknesses

- This model accounts for a variety of different factors and incorporates them into an algorithm that allows for subjectivity and different considerations. Depending on how the user of the algorithm values different objectives, the user could easily change several constants and repurpose the algorithm. This allows the City of Ithaca to utilize this model freely with their own subjective values while retaining key features of the model.
- The sigmoidal and piecewise functions in 3.1.1 are subject to overfitting since the number of data points from which those curves were fit to were too small.

However, the general shape should provide enough as it stands currently to model the trend intended.

3.2 Part C: Optimal Repair Procedures

Given a block of sidewalk slabs, each slab can either be left alone, replaced, raised, or cut. In order to decide which action to take on a slab, we will have to determine costs of the various procedures through geometric and algebraic analysis, which is what much of this task entails. In the end, the objective is to develop an algorithm that will take a block of sidewalk slabs and determine for each slab which repair procedure should be enacted while also minimizing cost.

3.2.1 Local Assumptions

1. We don't need to care about translational movement since each slab will not reasonably translate so much as to cause an accident if the elevation and slope are proper and wheelchair users if wheeling in the middle should still be able to safely cross.
2. We interpret the given change of the slab via the raising technique as fixing the slope and elevation of a slab rather than physical position, since slabs shouldn't be translating. Even if they were, based on videos online demonstrating the various ways to raise a slab, none involve translation or the movement of position, only slope and elevation.
3. We interpret the "at most 2 inches" condition for cutting a slab to refer to the vertical distance that is being cut.
4. We interpret the elevation to be equivalent to the vertical displacement of the slab.
5. Both the running slope and the cross slope are put in as inputs for a slab.

6. We interpret the area of a slab being raised in the raising repair procedure as the area of the slab that is below the plane defined by the top face of the adjacent slab.
7. We are given whether or not a slab is broken and its width.

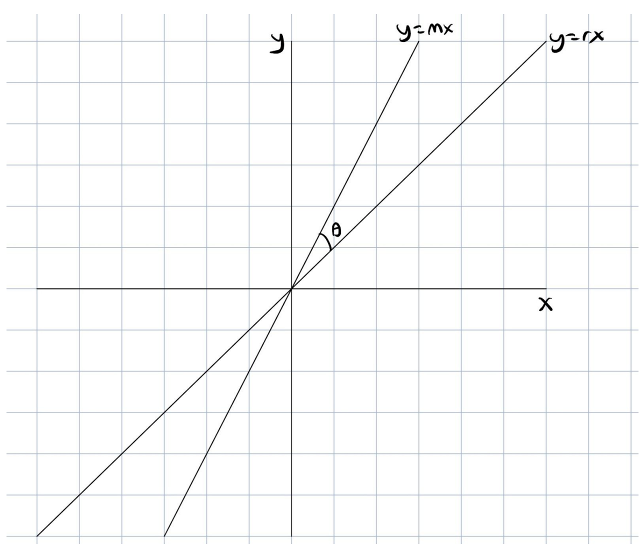
3.2.2 Deriving Values for Key Expressions

In order to develop the algorithm for an optimal repair strategy, we first must derive expressions for values that we will be using in the algorithm given the inputs and known information.

A Slab's Running Slope as an Angle

We wish to express the slab's running slope relative to the road's running slope in the form of an angle. Let the variable Θ represent this angle. We set the slab's running slope as the variable m and the road's running slope as the variable r. If we project the slab and the road to the xy plane, we can visualize the slab and the road as the lines $y = mx$ and $y = rx$ respectively, as seen in figure 5. We set the y-intercept for both of these lines at 0 because we're only evaluating the angle between the lines after the intersection, so we can set the origin as the point of intersection of these lines, thus giving them both a y-intercept of 0.

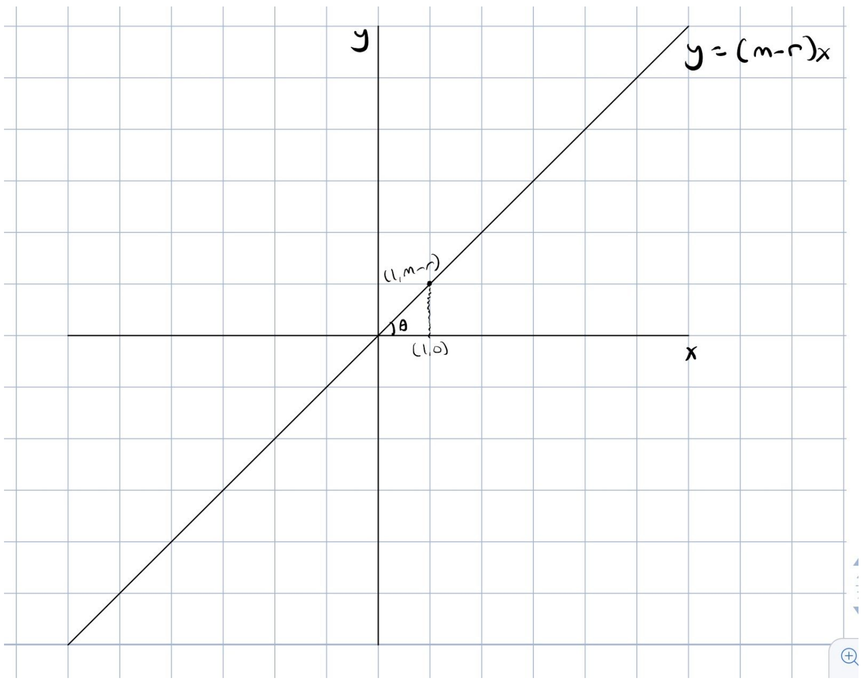
Figure 5: Visualization of the Slab and the Road in the XY Plane



Subtract rx from both lines and we have:

$y = (m - r)x$ and $y = 0$ as our lines. The point on the top line when $x = 1$ is $(0, m - r)$. We can then construct a triangle that goes up from $(1, 0)$ to $(1, m - r)$, as seen in Figure 6.

Figure 6: Visualization of the Slab and Road After Transformation



Solving for Θ in this triangle will give us the Θ between the two lines because of similar triangles; we use the points that have x = 1 to make the derivation simpler.

$$tan(\Theta) \ = \ \frac{m-r}{1}$$

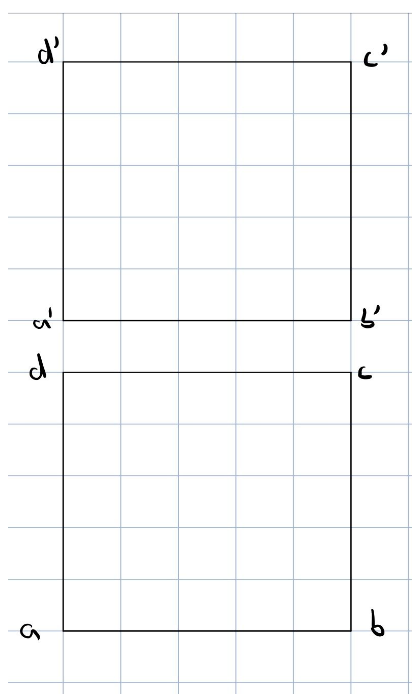
$$\Theta \ = |tan^{-1}(m - r)| \qquad \text{(Eq 13)}$$

We take the absolute value because we are just using Θ to find the side lengths in our geometric analysis later on, so we want to have positive values for our side lengths. In the case where m is less than r, $tan^{-1}(m-r)$ evaluates to a negative, but for the purposes of our geometry, Θ should be positive.

Vertical Displacement

We seek to find the vertical displacement between two slabs. In order to do this, we must utilize the input positions for the slabs. We define the given position for a slab as the z coordinates of its top face. In order to find the vertical displacement, we look to subtract the highest point of the lower slab from the highest point of the higher slab. We label the points of the slabs' top faces as in Figure 7 below.

Figure 7: Bird's-eye View of The Top Faces of Two Adjacent Slabs



We define the slab that's higher to be the one represented by (a, b, c, d), and thus the other slab is represented by (a′, b′, c′, d′).

There are a couple of cases to consider for determining the vertical displacement:

One case is that the higher slab is elevated on the c-d side. In this case, the vertical displacement is the greater of $d - a'$ and $c - b'$.

The other case is that the the higher slab is elevated on the a-b side. In this case, the vertical displacement is the greater of $a - d'$ and $b - c'$.

Linear Cutting Distance

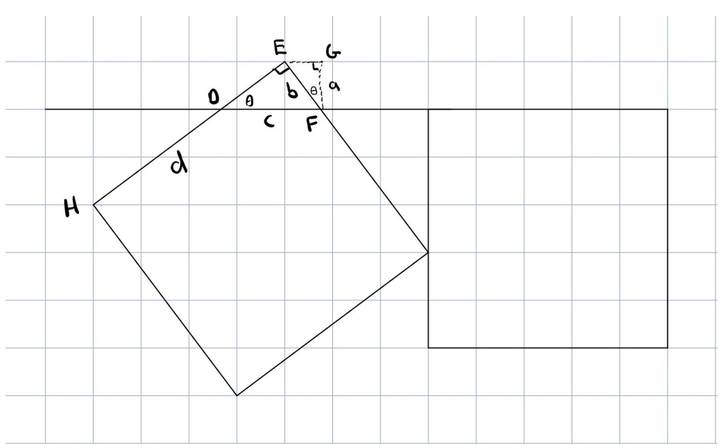
We now have the values needed to derive an expression for the linear cutting distance, which is the value that will be directly used in calculating the cost of a cutting repair procedure.

We visualize the slabs in the xz plane, as the thickness of the slabs shouldn't affect the cutting procedure.

We draw in the right triangle represented by dotted lines that has the vertical displacement as one of its legs, represented by the variable a.

As seen in figure [8] below, our goal is to solve for c.

Figure 8: Visualization of the Adjacent Slabs in the xz Plane



In triangle DEF, angle F is equal to $\frac{\pi}{2} - \Theta$ (in radians) because the angles of a triangle add up to π, and the other two angles in the triangle are Θ and $\frac{\pi}{2}$. In triangle DFE, we know that angle F is equal to $\frac{\pi}{2}$ because it is a right angle. Thus, angle F in triangle EFG is equal to Θ, because $\frac{\pi}{2} - \Theta + (angle\ F\ \ in\ EFG)\ = \frac{\pi}{2}$.

Given this geometric arrangement, we can set up the following system of equations:

$$cos(\Theta)\ =\ \frac{a}{b}\ \ sin(\Theta)\ =\ \frac{b}{c}$$

From the first equation, if we multiply both sides of the equation by b and divide both sides by cos(Θ), we obtain:

$$b\ =\ \frac{a}{cos(\Theta)}$$

Substituting this value of b into the second equation, we obtain:

$$sin(\Theta)\ =\ \frac{\frac{a}{cos(\Theta)}}{c}$$

Rearranging gives us:

$$c\ =\ \frac{a}{cos(\Theta)*sin(\Theta)} \qquad \text{(Eq 14)}$$

Thus, we have solved for the linear cutting distance in terms of the vertical displacement and our running slope angle, which we derived expressions for earlier.

Cost of Cutting

As a cutting repair operation costs $16 per linear foot of slab, the cost of cutting for a slab is equal to $16*c, where c is the linear cutting distance calculated in the previous section.

The Area Being Raised

Now that we have an expression for the cost of cutting, we wish to derive an expression for the cost of raising a slab. In order to do this, we need to obtain an expression for the area of slab that we are raising, as that value is directly used in calculating the cost for a raising operation.

We refer to figure [8] again, which is the same visualization as in the "Linear Cutting Distance," except our goal this time is to solve for the length represented by the line segment HD and the variable d in the diagram. Per our definition of the area being raised, solving for d will give us one of the sides of the rectangular area that is being raised.

As can be seen in figure [8], d is equal to the side length of the overall slab, 5, subtracted by the length of segment ED. By the Pythagorean theorem, segment ED has length $\sqrt{c^2 - b^2}$. Plugging in $b = \frac{a}{cos(\Theta)}$, we get the equation:

$$\overline{ED} = \sqrt{c^2 - \frac{a^2}{cos^2(\Theta)}} \qquad \text{(Eq 15)}$$

The other side of the rectangular area that we are attempting to solve is 5 feet because we are operating with slabs that are 5x5 (in feet) when looking at the top base. Thus, the area in square feet of the area being raised, in feet, is $5 * \sqrt{c^2 - \frac{a^2}{cos^2(\Theta)}}$

Cost of Raising

The cost of raising is given by $\$5.13 * \textit{the square feet of slab raised by the operation}$, which is equal to $\$5.13 * 5 * \sqrt{c^2 - \frac{a^2}{cos^2(\Theta)}}$, which is equal to $\$25.65 * \sqrt{c^2 - \frac{a^2}{cos^2(\Theta)}}$

Cost of Replacing

The cost of replacing a slab is fixed at $22 per square foot, which for a slab in this case is equal to $\$22 * 5 * 5 = \$550$

3.2.3 Developing the Algorithm

We now have the necessary expressions to develop the algorithm for the optimal repair strategy. We first must outline some basic guidelines to ensure that we comply with regulations.

Guidelines to ensure Compliance with Regulations

If the slab is broken, it must be replaced.
If the slab is less than 4 feet wide, it must be replaced.
If the vertical displacement is greater than 2 inches, the slab must either be raised or replaced (it cannot be cut).
A slab that is deemed to be "good" will not have a repair operation done on it. We define a slab to be "good" if it has the following characteristics:

- The slab's road slope complies with the ADA requirement for running slope of being within 2% of the road's running slope
- The slab's cross slope complies with the ADA requirement for cross slope of being between 1% and 2% of the road's cross slope
- The z coordinates of the slab that are inputted for the position are within half an inch of 5 feet, as a new slab that has nothing negative done to it would have z coordinates of 5 feet, since a slab is 5 feet tall.

If a slab is not "good," it is considered "not good," and should have an operation done on it that will make it "good."

Guidelines for Developing the Algorithm

In addition to guidelines that we must follow to ensure that we are within regulations, there are also guidelines that we will use to help us develop the algorithm. These guidelines will provide and justify various courses of action given different situations that will occur in the process of actually using the algorithm to determine the optimal repair strategy for sidewalk slabs on a block.

Given that this algorithm will be used to find the optimal repair strategy for a block, the algorithm will generally run from one end of the block to the other end with the following exception: if the algorithm encounters a string of not good slabs, we keep looking at successive slabs until we find a good slab. We then do the algorithm but for these cases, we move backwards to the previous slab after evaluating a slab. This version of the algorithm runs until all the not good slabs are fixed and operated on. After that, the normal algorithm picks up from the first good slab that was found after the original string of not good slabs.

The general rule of thumb that guides this algorithm is that for a slab that must be fixed, if the slab must be replaced, then replace the slab. Otherwise, choose either cutting or repairing depending on which option is cheaper, and afterwards, set the slab's values of slope and position to match its status after the operation.

In the cases where we are doing an evaluation with two slabs, a, b, c, and d refer to z coordinates for the current slab and a', b', c', and d' refer to coordinates for the other slab

If we are evaluating two slabs and one of them is "good" and the other is fully below the "good" slab but violates one of ADA regulations (3)-(5), the other slab must be either repaired or replaced (it cannot be cut).

If we are evaluating two slabs and one of them is "good" and the other's top face is fully above the "good" slab but violates one of ADA regulations (3)-(5), the other slab must be either cut or replaced (it cannot be raised).

If a slab is not “good,” it is considered “not good” and should have an operation done on it that will make it “good.”

Defining and Expressing Values for the Algorithm

We now develop the procedure that will determine what to do with each slab. There are 4 possible outcomes:

1) The slab should not have an operation done on it
2) The slab should be replaced
3) The slab should be raised
4) The slab should be cut

We will assign each slab a value that corresponds to the above outcome. These values are:

1) Good
2) Replace
3) Raise
4) Cut

This value is stored in a variable called Value.

Given a certain block, the road values are constant throughout the block. We store these road values in variables:

r = the road’s running slope

c = the road’s cross slope

We assign variables contained within each slab to the inputs and information that is known:

Slab.next = the next slab

NextExists = true if there is another slab after the current slab, false otherwise

Broken = true if the slab is broken, false otherwise

w = width of the slab

m = the slab's running slope

n = the slab's cross slope

a = the slab's "first" z coordinate

b = the slab's "second" z coordinate

c = the slab's "third" z coordinate

d = the slab's "fourth" z coordinate

For clarification on the slab's z coordinate position variables, refer to figure [7]

As the algorithm will use several of the key values that we found expressions for in Section 3.2.2, we wish to implement these expressions based on the derivations from that section that will calculate these values for the inputs and information corresponding to each slab. These results can then be referenced in the main algorithm body, which makes the main algorithm more streamline and easier to read. We mark specifications that detail what is being calculated with a **...** line above the specified work.

The vertical displacement

Let the good slab's z coordinates be represented by a′, b′, c′ and d′ (a′, b′, c′, and d′ are in the same relative positions as a, b, c, and d, respectively)

If (c+d is greater than a+b), then the vertical displacement is the greater of d-a′ and c-b′.

Otherwise, the vertical displacement is the greater of a - d′ and b - c′.

The running slope angle for the slab

$$\Theta = tan^{-1}(m - r) \qquad \text{(Eq 13)}$$

The linear distance that is cut

$$\text{Linear distance} = \frac{Vertical\ Displacement}{cos(\Theta) * sin(\Theta)} \qquad \text{(Eq 16)}$$

The cost of a cutting operation

$$\text{Cost of cutting (in \$)} = 16 * Linear\ Distance \qquad \text{(Eq 17)}$$

The area of slab that is raised

$$\text{Area (in square feet)} = 5 * \sqrt{(Linear\ Distance)^2 - \frac{(Vertical\ Displacement)^2}{cos^2(\Theta)}} \quad \text{(Eq 18)}$$

The cost of raising a slab

$$\text{Cost of raising} = \$25.65 * \sqrt{(Linear\ Distance)^2 - \frac{(Vertical\ Displacement)^2}{cos^2(\Theta)}} \quad \text{(Eq 19)}$$

The cost of replacing a slab

Cost of replacing = $550

Whether the slab is good or not

If Broken = true, the slab is not good

If w < 4, the slab is not good

If m >= 1.02*r or m <= .98*r, the slab is not good

If n <= 1.01*c or n >= 1.02*c, the slab is not good

Let a′′, b′′, c′′, and d′′ be the z coordinates of the slab before it, where a′′, b′′, c′′, and d′′ have the same relative position as a, b, c, and d respectively (Not applicable if the current slab is the first slab in the block)

Let a′′′, b′′′, c′′′, and d′′′ be the z coordinates of the slab after it, where a′′′, b′′′, c′′′ and d′′ have the same relative position as a, b, c, and d respectively (Not applicable if the current slab is the last slab in the block)

If $|d'' - a|,\ |c'' - b|,\ |d - a'''|,\ or\ |c - b'''|$ is greater than 1/24, then the slab is not good

If the slab is good, the variable Good = true. If the slab is not good, the variable Good = false.

The Algorithm

```
IF Good = true
    Value = Good
    Move on to the next slab
IF Good = false
    IF Broken = true OR w > 4
        Value = Replace
        IF NextExists = true
            Move on to the next slab
        ELSE
            TERMINATE the Algorithm
    IF Broken = false
        IF Next Slab’s Good = true
            IF Vertical Displacement > 1/6 OR a < a′, b < b′, c < c′, and d < d′
                Value = Raise
                Good = true
                m = slab.next.m
                n = slab.next.n
                a = a′, b = b′, c = c′, d = d′
                IF NextExists = true
                    Move on to the next slab
                ELSE
                    TERMINATE the Algorithm
            IF Cost of cutting < Cost of raising OR a > a′, b > b′, c > c′, & d > d′
                Value = cut
                Good = true
                m = slab.next.m
```

```
            n = slab.next.n
            IF c+d > a+b
                c = c′
                d = d′
            ELSE
                a = a′
                d = d′
            IF NextExists = true
                Move on to the next slab
            ELSE
                TERMINATE the Algorithm
        IF Cost of cutting > Cost of raising
            Value = raise
            Good = true
            m = slab.next.m
            n = slab.next.n
            a = a′, b = b′, c = c′, d = d′
        IF NextExists = true
            Move on to the next slab
        ELSE
            TERMINATE the Algorithm
    IF Slab.next.Good = false
        IF A good slab doesn′t exist for the rest of the block
            CREATE a new slab NewSlab
                NewSlab.Good = true
                NewSlab.m = r
                NewSlab.n = c
```

NewSlab.w = 4

NewSlab.Broken = false

NewSlab.a = 5

NewSlab.b = 5

NewSlab.c = 5

NewSlab.d = 5

IF a good slab still exists

DO the Algorithm with the following specification:

Find the next good slab, and then run the steps of the algorithm up to this point for the slabs, but everytime a slab is evaluated, move on to the **previous** slab and evaluate that slab. In other words, make the keyword “next” refer to the previous slab until all the not good slabs between the last good slab and the first good slab found after successive not good slabs are fixed. After all of those slabs are fixed, reset the keyword “next” to refer to the successive slab and pick up the algorithm from the first good slab that was found after the original string of not good slabs. If a NewSlab object was created, remove that slab from the block.

3.2.4 Strengths and Weaknesses

- A strength is that the expressions for key values were derived largely by using geometric and trigonometric techniques and operations that can be easily followed in the diagrams. For example, many actions in the derivation of expressions were done with techniques such as using similar triangles, Pythagorean’s Theorem, and the definitions of sin and cos, all of which are greatly aided and made easy to follow through diagrams. This means that the reader can more easily follow along in how important values are solved for.

- A weakness is that the development of the algorithm required many calculations and derivations of values that required us to make interpretations and definitions for. As our interpretation of what a value corresponds to could be different from someone else's interpretation, this could lead to a discrepancy in how the algorithm operates for someone else. For example, we interpreted the area of a slab that is being raised as the area of the top face of the slab that is below the plane determined by the top face of the adjacent good slab. This interpretation is clearly by no means universal, and the method that we used in calculating the area is specific to our interpretation. Therefore, it could be hard for someone to smoothly adjust our work to fit their interpretation.

3.3 Part D: Future Expenditures

In this model, we address the problem of predicting the required budget in order for the Sidewalk Improvement Program to continue to be *effective*, as it has for the past five years (2015 - 2019).

3.3.1 Local Assumptions

1. We assume that the rates of change of new blocks of sidewalk being added every year sourced from the ArcGis data for the Sidewalk Improvement Program represent a constant rate of growth [10].
2. We assume that the base lifespan of sidewalk in Ithaca calculation made by Yost and Benjamin of 20 years applies to every sidewalk slab in Ithaca, and accounts for the amount of damage due to natural causes in 2015, i.e. tree root growth, repeated freezing & thawing, soil erosion processes, and excessive weight loads [12].

3. Assume that the average block length is 200 feet, as this represents a center value for the values in the ArcGis data. (Derived using 5200 total linear feet of sidewalk [Memo] / 26 blocks repaired / installed in 2018 [ArcGis])[6][10].
4. Based on the Coupled Global Climate Model simulations for cities in Canada, we assume that Ottawa's (close in proximity to Ithaca of the cities labelled in the table) estimated increase in freeze-thaw cycles (the sum of those temperature-driven and precipitation-driven) from the period of 1961 - 1990 to 2050 apply to Ithaca's change in freeze-thaw cycles as a result of climate change.
    a. This increase can be applied linearly, as the study claims the model represents "a reasonable approximation within sites over the ranges of projected changes in winter temperature and precipitation." [3]
5. We also assume that the Heartland region is approximate to that of Ithaca weather in terms of present freeze-thaw cycles [16].
6. The fast freeze-thaw cycle study performed on ordinary-air-entrained concrete models the effect of freeze-thaw cycles on the strength of the same concrete used to construct all Ithaca sidewalks. Cleavage strength is the selected figure by which to measure the strength of the sidewalk as it most directly represents cracking on the surface due to thermal stresses. [13].
7. Ithaca uses C30 grade concrete as it is a standard for walkways/roadways [14].
8. The inflation rate, represented by the CPI rate of increase from years 2015-2018 can be fit linearly to represent inflation changes indefinitely.

3.3.2 Deriving Key Functions

In this section, we derive functions of time since 2015 which take key factors in the future of the Sidewalk Improvement Program into account.

Considering growth in sidewalk lengths every year

Let length(t) be a function of t (time in years since 2015), given f(0) . It outputs the number of feet of sidewalk needed to be maintained. We are given the blocks repaired or installed in linear feet by the the tables from the Sidewalk Improvement Program ArcGis website.

length(0) = 30005

length(1) = 14094

length(2) = 11453

length(3) = 11734

For years 2019 … 2035:

$$length(t) = \overline{total}_{\text{changed}} \tag{Eq 20}$$

For years 2036 and beyond:

$$length(t) = \overline{total}_{\text{new}} + \overline{total}_{\text{changed}} + counter \tag{Eq 21}$$

where $counter$ is the amount of times that a set of slabs created by the SIP since 2015 have reached its life expectancy.

Since the base lifespan is assumed to be 20 years in 2015, the first time new sidewalk built from the project will need to be fixed is 20 years from 2015.

We apply our assumption that the rate of change for the increase in sidewalk length needed each year is constant, i.e. that the city of Ithaca requires a certain new sidewalk length for installation per year. We reference the data from the years 2015 - 2018 to formulate these mean footing of sidewalks.

Figure 9: Sidewalk Improvements From 2014 - 2018, courtesy of the Sidewalk Improvement Program ArcGis website

	Number of Blocks repaired or Installed	Blocks repaired or Installed in linear feet	Number of cuttings and raisings	Cuttings and raisings in linear feet
2014	11	5,611	31	575
2015	52	30,005	24	699
2016	36	14,094	40	1136
2017	34	11,453	50	684
2018	26	11,734	67	697
TOTAL	159	72,897	212	3,791

where $\overline{total}_{\text{changed}} = \frac{52+36+34+26\ blocks}{4\ blocks\ total} = 37\ blocks = 7400\ feet$, $\overline{total}_{\text{new}} = \frac{6+1+5+3\ \ new\ blocks}{4\ blocks\ total}$ $= 3.75\ blocks = 750\ feet$

Considering the effects of climate change

Increased variability of the climate in Ithaca in recent years has led to the deterioration of the relative strength of its concrete sidewalks, a result of repeated freeze-thaw cycles, an event which occurs as the temperature fluctuates from day to night. This results in the thermal expansion and contraction of water (freezing/melting), which can damage sidewalk concrete through the stress applied. In order to quantify the amount of damage over a certain number of years since 2015 as we did when considering sidewalk growth, we use the Projected Change in Annual Soil Freeze-Thaw Cycles given by a periodical published in Springer Science + Business Media for Ottawa, which we have assumed to have a comparable rate of freeze-thaw cycles as Ithaca.

Table 3 Projected changes in winter temperatures and precipitation from the period 1961–1990 to 2050 based on climate change simulations made with the Coupled Global Climate Model (CGCM2)

Site	Name	Projected winter temperature change (°C)	Projected winter precipitation change (%)	Projected change in annual soil freezing days		Projected change in annual soil freeze–thaw cycles	
				Temperature-driven	Precipitation-driven	Temperature-driven	Precipitation-driven
1	Resolute	5.5	5	−23.9	−0.2	1.4	0.0
2	Clyde	5.5	−5	−4.5	−0.3	1.0	0.0
3	Watson Lake	3.5	15	−8.7	2.4	1.8	−0.7
4	Cree Lake	4.5	5	0.0	0.3	1.8	0.0
5	La Ronge	4.5	15	−5.8	0.2	1.9	−1.4
6	Kapuskasing	3.5	−5	9.0	0.6	1.7	0.5
7	Goose	4.5	−5	−8.3	0.6	1.7	−0.4
8	Peace River	3.5	5	−10.0	1.2	1.3	−0.1
9	Normandin	2.5	−5	−0.6	−0.4	1.6	0.1
10	Wynyard	4.5	5	−10.7	1.3	−1.3	0.2
11	Winnipeg	4.5	−5	0.2	−0.5	2.7	0.3
12	Atikokan	3.5	5	−12.1	−1.4	1.2	−0.3
13	Val-d'Or	3.5	−5	−2.4	1.5	1.2	−0.2
14	Montmorency	2.5	−5	−20.4	−4.7	−1.3	−0.1
15	Estevan	4.5	−5	0.4	0.2	6.1	0.4
16	Beaverlodge	2.5	5	−11.7	0.9	0.0	0.1
17	Lacombe	3.5	5	−3.9	−0.3	5.9	−0.6
18	Swift Current	4.5	5	−10.5	0.5	4.3	−0.5
19	Mirabel	2.5	−5	−4.5	−0.1	−1.6	0.1
20	Lennoxville	2.5	−5	2.1	4.4	0.7	0.2
21	Ottawa	2.5	−5	24.0	5.4	2.9	0.2

[3]

We then sum the projected change in annual soil freeze-thaw cycles for both cases: temperature-driven and precipitation-driven. This yields a total increase of 3.1 soil freeze-thaw cycles from 1990-2050. Since we have also assumed that the rate of increase is linear, we get a result of

$$\frac{3.1\, freeze\text{–}thaw\ cycles}{60\ years} = 0.05167 \text{ increase of freeze-thaw cycles/year}$$

Next, we compute the yearly loss in strength and lifespan based on these increasing freeze-thaw cycles. This is dependent on concrete grade, a metric for measuring the “strength and composition” of concrete based on a measurement 28 days after its initial construction [14]. Since the typical concrete grade for use in sidewalks has been assumed to be C30, we analyze the changes in cleavage strength as a result of freeze-thaw cycle events. This relationship is given in the table below:

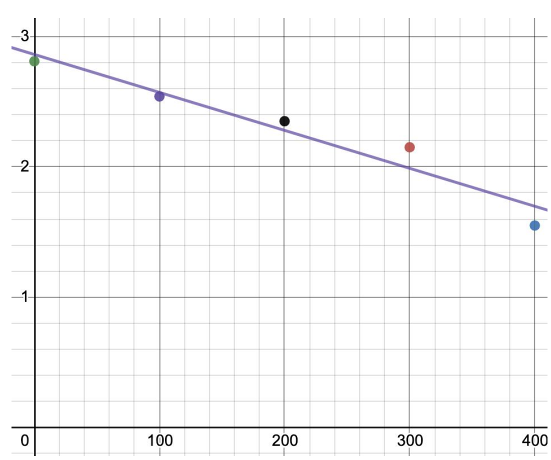
Figure 10: Cleavage strength of C30 O-A-E concrete after fast freeze-thaw cycles (MPa), courtesy of US National Library of Medicine, by Shang et. al

Number of fast freeze-thaw cycles	0	100	200	300	400
Cleavage strength	2.81	2.54	2.35	2.15	1.55

The code ran to obtain the following linear regression of the above table can be found in Appendix D:

$$y = -0.00267417x + 2.81$$

Figure 11: Ridge Regression with R^2: 0.9314



With the knowledge that there are 87 freeze-thaw cycles in Ithaca (assumed from [16]), we can compute the cleavage strength in 2016 of new concrete sidewalks to be 2.7. Assuming that 60 years in lifespan is correlated to 2.81 cleavage strength, the percent change in cleavage strength from 2015 to 2016 was 3.9145%. This correlates to 2.3 years deprecated from the original 60 year lifespan. If this continues for another 20 years (the supposed lifespan of concrete sidewalk in Ithaca), the lifespan would have depleted by 46 years from the original 60 year lifespan, meaning the concrete is now “dead” which it is since 20 years have passed.

$$remaining\ lifespan(t) = (60 - (\frac{-0.00267417 \cdot (87 + 0.05167t) + 2.81}{2.81}) \cdot 60t) - t \qquad \text{(Eq 22)}$$

Considering changes in price per linear foot of sidewalk

In this section, we consider the increase in price per linear foot of sidewalk per year. We will base this function on the prices per foot of 5’ sidewalk for the years 2015 - 2018 in the 2019 SID memo, [6]. We will then perform a regression on the points with the code found in Appendix E.

Figure 12: Years Since 2015 and Price Per Linear Foot [6]

Years Since 2015	Price ($) / Linear Foot of Sidewalk
0	83.18
1	85.75
2	100.27
3	107.52

$$y = 80.00718 + 17.83836\ln(t)$$

3.3.3 Final Model & Budget Projection

Now, given the three functions, we can form the overall budget function of t, time in years since 2015:

$$Budget(t) = length(t) * price(t)\,, \qquad \text{(Eq 23)}$$

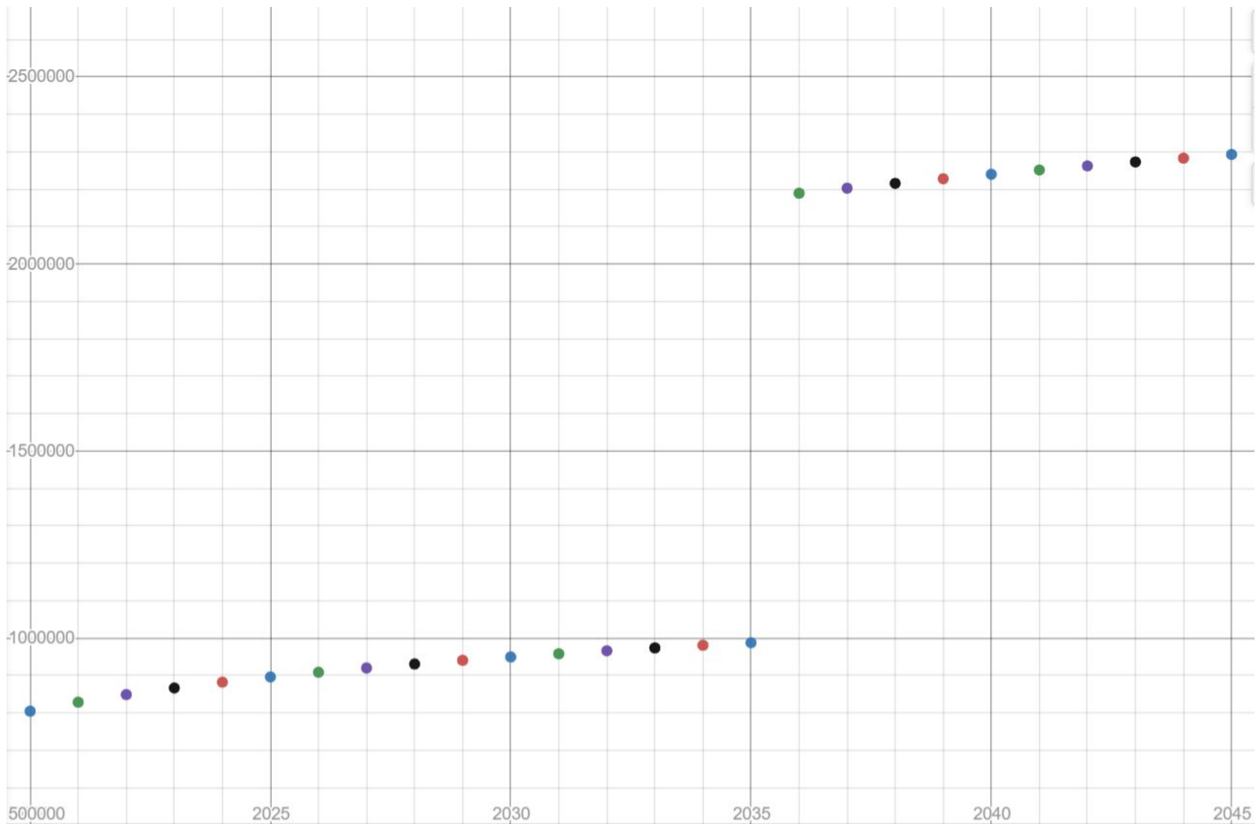
We now calculate the budget increases over the next 25 years, using a Java algorithm (Appendix F) which models the function as above.

A run of the program with input 30 (years after 2015; next 20 years from the present) produced the following data:

```
(2020, $804505.16)
(2021, $828572.31)
(2022, $848920.79)
(2023, $866547.45)
(2024, $882095.27)
(2025, $896003.26)
(2026, $908584.57)
(2027, $920070.41)
(2028, $930636.36)
(2029, $940418.90)
(2030, $949526.22)
(2031, $958045.56)
(2032, $966048.24)
(2033, $973593.37)
(2034, $980730.45)
(2035, $987501.37)
(2036, $2189358.42)
(2037, $2202884.82)
(2038, $2215809.85)
(2039, $2228184.71)
(2040, $2240054.33)
(2041, $2251458.35)
(2042, $2262431.92)
(2043, $2273006.37)
(2044, $2283209.71)
(2045, $2293067.10)
```

The graph of the data is located in the following figure:

Figure 13: Projected Budget Requirements For the r 2015 for the SIP



There is a noticeable "jump" in the data for the graph as a result of the assumption that all the sidewalk concrete breaks at once at the expected lifespan modeled earlier in section 3.3.2. While this is a weakness, this an expected result of the model, the ramifications and causes of which are expanded on in the next section.

3.3.4 Strengths and Weaknesses

- This model's strengths lie in its consideration of freeze-thaw cycles projected using real models, sourced from the study by Henry in 2007, commingling this model with another study concerning the strength of concrete in response to the number of freeze-thaw cycles experienced by C30 concrete. This allows the model to be more predictive in nature.
    - This also introduces a weakness in the model, as the study only predicts freeze-thaw cycle increasing rates up to 2050, and freeze-thaw cycle rates of increase could vary past this date.
- The most glaring weakness of the model is in the final budget projection: initially, the model is underfitting due to the relative shortage of data points as the Sidewalk Improvement Program is still in its infancy. In spite of this, it displays an increasing rate of budget requirements in the graph.
- Another weakness of the model is its jumps in budget after a certain number of years in which the lifespan of sidewalks (which are dependent on the years since 2015, as freeze-thaw cycles change) built that amount of years ago has reached its end. This is a result of the assumption that all the sidewalks built at a certain time break at an end lifespan, while a more accurate and rigorous representation would be a probabilistic function modeling the lifespans of individual blocks of concrete sidewalk slabs.

4 Conclusion

We devised an algorithm that produces a resultant prioritization score based on not only the current criteria that are in use but a variety of new factors as well. Without the proper data, no physical value can be attained; however, the model provides a good and easy to digest template for such calculations. Following the procedures outlined previously, one can obtain numeric scores for assessing the quality of a block's sidewalks as well as their magnitude of impact on the people of Ithaca. The third and final score obtained is the location of the block in relation to neighboring blocks and their sidewalk quality and impact (both of which determine the likelihood of them being chosen to be fixed). This score should increase as the likelihood of its neighbors being prioritized first does and decrease while the likelihood is low. These 3 scores will be weighted and summed to obtain the final priority score on the range of 0 - 100 with 100 being the most prioritized.

We were also able to devise an algorithm that assigns a value to each slab on a block that signifies if it should be left alone, replaced, raised, or cut. The algorithm relies largely on a series of conditional statements denoted by IF and ELSE statements. The statements that determine which repair procedure to use can be viewed as being in one of two groups: determining a procedure based on requirements and determining a procedure based on cost effectiveness. An example of a statement that determines a procedure based on requirements can be seen by the statement IF Vertical Displacement $> 1/6$ OR $a < a'$, $b < b'$, $c < c'$, and $d < d'$. This section of the algorithm ends up saying that if the vertical displacement exceeds 2 inches or the slab is entirely below the good slab, then the raise procedure must be used, as due to regulations and the physical nature of cutting, cutting can't be used. An example of a statement that determines a procedure based on cost effectiveness can be seen by the statement IF Cost of cutting > Cost of raising. This section ends up saying that if it costs less to raise the slab than cut it, we should raise the slab. We could cut it, but raising it would minimize cost, which is one of the goals of the algorithm. The algorithm runs through each slab until all the slabs on a block have been either repaired or determined to be sufficient enough to be left alone.

Lastly, we formed a model of the budget required to sustain the effectiveness of the Sidewalk Improvement Program as a function of time in years since 2015. The function considers different factors leading to a requirement for greater budgets, including the ever-increasing sidewalk lengths of Ithaca due to new properties, faster degradation of sidewalk concrete from variabilities in the climate, resulting in lessened concrete lifespans, and the trend of increasing price per foot of 5’ sidewalk evidenced in recent years (2015-2018). These functions were all combined to form a final predictive model, considering the constant change in concrete lifespans due to continuous climate change. The results of the model display an increasing rate of change in budget required every year, and a spike in budget required after the first sidewalk slabs installed by the program in 2015 reach the end of their lifespan. By using the budgets reported by the projected data, the Common Council can make plans well ahead of time and be prepared for what the future holds in the growth of the sidewalk program.

5 References

[1]http://www.pedbikesafe.org/PEDSAFE/guide_implementation.cfm

[2] http://www.city-data.com/income/income-Ithaca-New-York.html

[3]https://link.springer.com/content/pdf/10.1007%2Fs10584-007-9322-8.pdf

[4]https://en.wikipedia.org/wiki/Sidewalk

[5]http://www.nyc.gov/html/dot/downloads/pdf/nycdot-streetdesignmanual-interior-03-materials.pdf

[6] https://e.math.cornell.edu/sites/mcm/old_problems/SID_memo_2019_update.pdf

[7]https://www.ok.gov/odc/documents/ADAPostConstructCkList.pdf

[8]https://www.theatlantic.com/education/archive/2014/03/highly-educated-countries-have-better-governments/284273/

[9]https://www.federalreserve.gov/econres/feds/files/2017031pap.pdf

[10]https://storymaps.arcgis.com/stories/4df2b76ade6a4171a5ccbed80bf47ceb -

[11]https://ithacavoice.com/2017/04/interactive-map-examining-pedestrian-crash-data-improvements-ithaca/

[12]https://www.ithaca.com/news/broken-state-of-ithaca-s-sidewalk-program-examined/article_04ce114a-6ea8-11e1-a067-001871e3ce6c.html

[13] https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4033560/

[14]https://www.baseconcrete.co.uk/different-types-of-concrete-grades-and-their-uses/

[15]http://www.cityofithaca.org/DocumentCenter/View/1436/City-of-Ithaca-Official-Zoning-Map-PDF?bidId=

[16]https://etd.ohiolink.edu/!etd.send_file?accession=kent1302547210&disposition=attachment

Appendix

A

```python
import scipy
import numpy as np
import pylab
from scipy.optimize import curve_fit


xdata = np.array([0.5, 2.25, 4.0])
ydata = np.array([0, 0.5, 1])

def sigmoid(x, L ,x0, k, b):
    print(x0)
    y = L / (1 + np.exp(-k*(x-x0)))+b
    return (y)

p0 = [max(ydata), np.median(xdata),1,min(ydata)] # this is an mandatory initial guess

popt, pcov = curve_fit(sigmoid, xdata, ydata,p0, method='dogbox')

x = np.linspace(0, 2, 100)
print(popt)
y = sigmoid(x, *popt)


pylab.plot(xdata, ydata, 'o', label='data')
pylab.plot(x,y, label='fit')
pylab.ylim(-1, 1.05)
pylab.xlim(-0.5, 1)
pylab.legend(loc='best')
pylab.show()
```

B

```
import scipy
import numpy as np
import pylab
from scipy.optimize import curve_fit


xdata = np.array([0, 0.02, 0.04341129301, 0.06682258603])
ydata = np.array([0.005, 0.01, 0.5, 0.99])

def sigmoid(x, L ,x0, k, b):
    y = L / (1 + np.exp(-k*(x-x0)))+b
    return (y)

p0 = [max(ydata), np.median(xdata),1,min(ydata)] # this is an mandatory initial guess

popt, pcov = curve_fit(sigmoid, xdata, ydata,p0, method='dogbox')

x = np.linspace(0, 0.15, 100)
print(popt)
y = sigmoid(x, *popt)


pylab.plot(xdata, ydata, 'o', label='data')
pylab.plot(x,y, label='fit')
pylab.ylim(-1, 1.05)
pylab.xlim(-0.1, 0.1)
pylab.legend(loc='best')
pylab.show()
```

C

```python
import scipy
import numpy as np
import pylab
from scipy.optimize import curve_fit


xdata = np.array([0.01, 0.02, -0.06166666666, -0.13333, -0.14])
ydata = np.array([0.01, 0.001, 0.495, 0.98, 0.985])

def sigmoid(x, L ,x0, k, b):
    y = L / (1 + np.exp(-k*(x-x0)))+b
    return (y)

p0 = [max(ydata), np.median(xdata),1,min(ydata)] # this is an mandatory initial guess

popt, pcov = curve_fit(sigmoid, xdata, ydata,p0, method='dogbox')

x = np.linspace(-1, 0.5, 300)
print(popt)
y = sigmoid(x, *popt)
print(y)
pylab.plot(xdata, ydata, 'o', label='data')
pylab.plot(x,y, label='fit')
pylab.ylim(-1, 1.05)
pylab.xlim(-1, 1)
pylab.legend(loc='best')
pylab.show()
```

D

```python
from sklearn.linear_model import Ridge
m = np.arange(0, 100, 1)
lam_vals = len(x)*(1.2**(m))
r_sqrdtest = np.empty(len(lam_vals))
b = '2.81'
score = 0
for i in range(len(lam_vals)):
    ridge_model = Ridge(alpha = lam_vals[i])
    ridge_model.fit(x, y)
    r_sqrdtest[i] = ridge_model.score(x, y)
    if str(ridge_model.intercept_)[:4] == b:
        b = ridge_model.intercept_
        score = ridge_model.score(x, y)
print(b)
print(score)
```

E

```
import scipy as sp
import numpy as np
import pylab
import scipy.optimize
import matplotlib.pyplot as plt


xdata = np.array([1, 2, 3, 4])
ydata = np.array([83.18, 85.75, 100.27, 107.52])
def logFit(x,y):
    # cache some frequently reused terms
    sumy = np.sum(y)
    sumlogx = np.sum(np.log(x))

    b = (x.size*np.sum(y*np.log(x)) - sumy*sumlogx)/(x.size*np.sum(np.log(x)**2) - sumlogx**2)
    a = (sumy - b*sumlogx)/x.size

    return a,b
def logFunc(x, a, b):
    return a + b*np.log(x)

plt.plot(xdata, ydata, ls="none", marker='.')

xfit = np.linspace(1,5,200)
plt.plot(xfit, logFunc(xfit, *logFit(xdata,ydata)))
plt.show()
print(logFit(xdata, ydata))

```

F

```java
import java.util.Scanner;

public class BudgetCalculation {
    public static void main(String[] args) {
        Scanner input= new Scanner(System.in);
        int yrsAfter2015= input.nextInt();
        for (int i= 5; i <= yrsAfter2015; i++ ) {
            System.out.printf("  (" + (2015 + i) + ", " + "%.2f)\n", budget(i));
        }
    }

    static private int counter= 1;
    static private int lastJump= 0;

    public static double budget(int t) {
        if (t <= 20) {
            return 7400 * (80.00718 + 17.83836 * Math.log(t));
        }

        else {
            if (remainingLifespan(t)) {
                lastJump= t;
                counter++ ;
            }
            return counter * 8150 * (80.00718 + 17.83836 * Math.log(t));
        }
    }

    public static boolean remainingLifespan(int t) {
        return 60 - 60 * (t - lastJump) * (1 - (-0.00267417 * (87 + 0.05167 * t) + 2.81) / 2.81) -
            t + lastJump <= 0;
    }
}
```