Path Optimization of UCLA for the Physically Disabled

Jason Liu, Ryan Liu, Hamza Khan, Shaash Sivakumar, Marc Walden

June 2023
UCLA
Math Department

Contents

1 Abstract

Accessibility for the physically disabled is a prevalent issue at UCLA where stairs and steep slopes make navigating campus arduous. In this project, we aim to model integral parts of UCLA campus and simulate the paths a wheelchair user and a control non-disabled individual would take. Using travel time as the primary metric, we develop two algorithms, Dijkstra's and least-resistance, to find optimal paths from selected nodes. We demonstrate how wheelchair users are dramatically inconvenienced by the lack of accommodations and inefficient ramps which increase their travel times to as much as 6 times the control. We seek to expand the model's scope and improve its accuracy in addition to proposing engineering solutions to make campus more accessible for the physically disabled.

2 Problem Description

Westwood is notoriously hilly and the University of California Los Angeles was subsequently designed with many stairs and steep pathways to optimize the area. This often makes the campus inaccessible to physically disabled students who attempt to navigate primarily through elevators or moderate inclines. Some integral halls and centers are entirely out of reach for wheelchair users and many more require lengthy detours. Physically disabled students comprise 2.1% of the 40,000 students at UCLA. Moreover, the influx of students using wheeled transport, including electric scooters and skateboards, has made campus accessibility an increasingly salient issue. UCLA is one of the largest public universities, hosting countless clubs, sports events, international speakers, performing arts, etc., and accommodating these populations is necessary for UCLA to encourage attendance and visitation regardless of physical ability.

This project aims to identify and access areas of inaccessibility on UCLA's campus, focusing on the most populated routes. For routes that don't support wheelchair users, we propose efficient ramps or similar solutions that would not unreasonably inconvenience disabled students. To highlight inefficient detours, we map UCLA's most populated routes, using travel time as a metric of comparison between a simulated wheelchair individual and a non-disabled individual. This enables us to depict where UCLA's campus struggles with wheel accessibility by analyzing large disparities in travel time between the two test subjects. With this data, we can propose efficient alternatives that wheelchair users could leverage in lieu of stairs or steep inclines. For the current state of campus, we use path optimization to calculate the fastest and most efficient routes for disabled individuals for any given destination. The results of the project will promote inclusivity and equal opportunity for education regardless of physical ability.

3 Simplifications

In order to construct our model, we simplify particular elements and abstract other variables or hold them constant while still trying to preserve the model's accuracy. UCLA campus size is roughly 400 acres, and while it would be beneficial to map the entirety of campus, we simplify our area of assessment to the most integral part of campus that sees the most traffic. This makes data collection for our model feasible while preserving its viability since the region is where physically disabled students are most likely to encounter issues. Once the model is appropriately calibrated, it can also be expanded to represent the entirety of campus. The simplified region is a rectangle with Bunche Hall, Pritzker Hall, Ackerman Turnaround, and Anderson School of Management denoting each of the corners.

We select nodes in the rectangular region which mark major intersections that connect key paths. In everyday scenarios, people navigate from any arbitrary point of campus to another but the simplification of using predetermined nodes allows us to employ graph theory and streamline data collection. The nodes will preserve the model's efficacy because any outdoor point in the rectangle can be reach through the nodes.

In the path optimization model, we will simulate a physically disabled and a non-disabled individual. To simplify the physically disabled simulation, we use a hypothetical non-motorized wheelchair user as the standard when considering factors like speed and necessary accommodations. Though there are a vast array of transport options for the physically disabled, each with their own characteristics, the wheelchair user would encompass the majority of them with the intended use of the model. Wheelchair accessibility would also be reflective of the preferences of scooters, skateboards, and other wheeled transport.

In order to consistently measure time for our wheelchair and non-disabled individuals, we make assumptions about their pace and endurance. For wheelchair users, we assume an average speed of 2 miles per hour and 3 miles per hour for the non-disabled individual. The average speeds may be reductive as individuals have varying speeds, especially for the wheelchairs when paired with motors. However, the aim of this project is to compare the ratios of times between different routes for disabled versus able-bodied, making raw times less relevant. We assume no fatigue, making pace consistent irrespective of the length and rigor of the route. Though this may introduce inaccuracies as many students prefer to avoid stairs and leverage elevators regardless of physical ability, it would be difficult to factor this into the model and provides little benefit for its intended purpose. We also assume a natural gradient of pace decrease when walking up slopes. To represent whether a given destination is accessible via wheelchair, we employ a binary system where 0 represents inaccessibility. Moreover, we assume any route which requires an incline above 30 degrees is inaccessible for the wheelchair user.

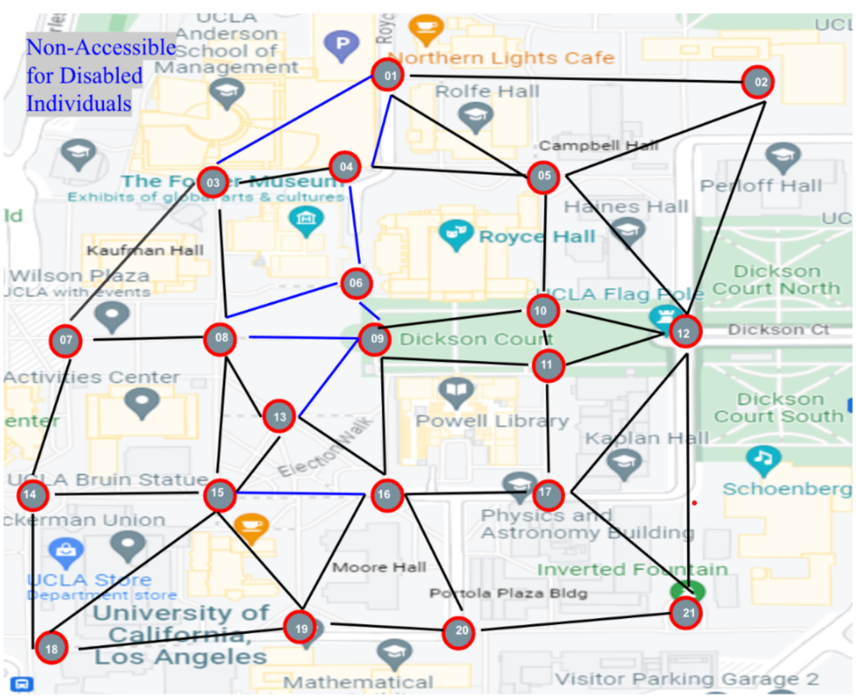
4 Building the Mathematical Model

Data Collection

We first construct a map comprising 21 nodes which represent points of interest that branch into 3 or more paths. We record each node's elevation and coordinates using Google Maps and Elevation Finder. By connecting the nodes we achieve 43 potential paths. We determine which paths are not wheelchair accessible by determining if they require stairs or steep slopes, defined by inclines of 30 degrees or greater. This leaves 32 paths for comparison between a wheelchair user and non-disabled individual.

TABLE	LOCATION DESCRIPTION	COORDINATES	ELEVATION (ft)
1	Northern Lights	34.074291, -118.442521	448.8
2	Bunche Hall	34.074257, -118.440335	459.3
3	Anderson School	34.073473, -118.443653	416.2
4	Parking Structure	34.073471, -118.442710	432.2
5	Rolfe Hall	34.073507, -118.441700	449.1
6	Jan Steps, Bottom	34.072626, -118.442746	435.7
7	Wilson Plaza	34.072209, -118.444543	400.2
8	Court of Philanthropy	34.072222, -118.443639	409.4
9	Jan Steps,	34.072186, -118.442686	443.7
10	Royce-Haines	34.072428, -118.441654	449.4
11	Powell-Kaplan	34.072016, -118.441640	449.1

12	Flag Pole	34.072212, -118.440937	444.2
13	Election Walk	34.071603, -118.443200	421
14	Bruin Statue	34.070987, -118.444762	393.7
15	Bottom Kerckhoff Steps	34.070967, -118.443631	403.2
16	Moore's Hall	34.070974, -118.442612	442
17	Physics Building	34.070984, -118.441646	444.7
18	Computer Science Department	34.069817, -118.444542	397
19	Math-Sciences	34.069825, -118.442942	434.2
20	Franz Hall	34.069877, -118.442096	437.4
21	Inverted Fountain	34.069979, -118.440808	426



To collect data on the travel times, we walk along each possible path at the base 3 mph pace and 2 mph pace for each simulation and record the seconds elapsed. 5 individuals total initially calibrate their walking speeds and traverse each route, slowing down proportionally when using stairs or slopes. The final data is calculated by using the average of the 5 test subjects data. On routes that include slopes or stairs, we record travel times in each direction to reflect changes in elevation. In the data below, note that the y-axis is the initial point and the x-axis is the destination.

Able-bodied simulation data:

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
1		97.28	205.66	51.28																	
2	85.34				106.75							169.41									
3	216.58			107.85			69.19	109.4													
4	59.84		92.34		110.39	105.37															
5		114.99		110.39						63.85		106.19									
6				97.41				93.97	130.63												
7			69.19					65.7						92.56							
8			109.4			93.97	65.7		101.49				64.58		113.4						
9						107.52		72.93		56.71	56.71		60.31			126.62					
10					63.85				56.71		30.34	43.33									
11									56.71	30.34		42.21					63.85				
12		171.38			106.19					43.33	42.21						110.97				205.89
13								54.25	63.55						68.35	101.93					
14							92.56								69.17			79.06			
15								115.71					91.41	59.17		98.7		194.01	133.45		
16									128.96				79.9		66.57		48.85		105.85	79.06	
17											63.85	111.37				45.85					88.11
18														79.85	179.74				86.7		
19															133.23	110.85		86.7		54.82	
20																79.06			43.93		145.41
21												212.11					88.11			145.41	

Wheelchair user simulation data:

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
1		117.03			163.37				117.03			163.37				117.03			163.37		
2	108.13				200.43							283.96									
3				168.32			112.24	122.13													
4			167.36		86.58																
5	163.37	227.27		85.27						128.57		184.12									
6																					
7			127.08					71.44						107.12							
8			131.13				65.34								116.04						
9										161.04	152.49					98.94					
10					135.27				163.29		37.9	74.43									
11									159.16	37.9		74.43					199.99				
12		293.94			179.18					74.43	74.43						214.42				248.45
13															57.87	215.95					
14							108.89								110.04			87.29			
15								113.24					58.3	105.72					252.12		
16									98.94				191.72				50.35		173.62	95.17	
17											133.29	207.71				50.35					217.93
18														90.46							
19															239.1	196.67				108.65	
20																88.02			90.41		168.63
21												243.45					222.3			140.79	

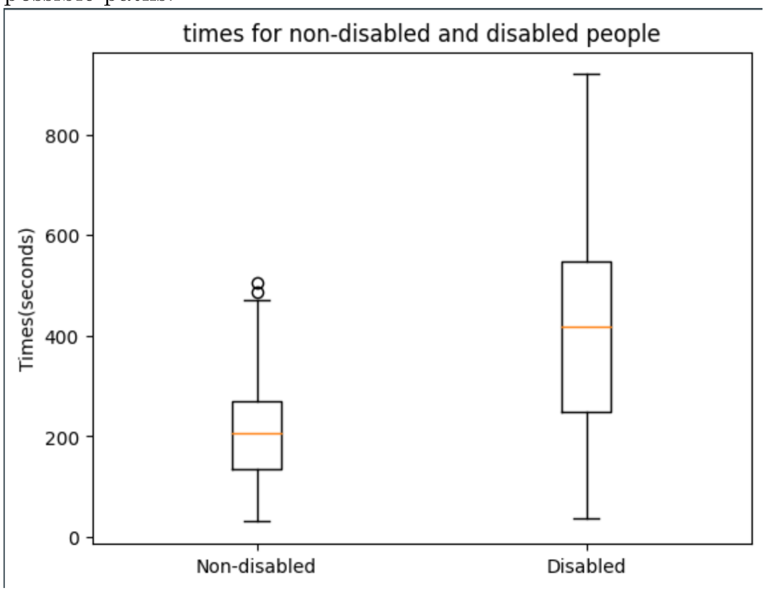
Path Optimization Algorithms

We take two approaches in building the path optimization algorithm. The first approach is using Dijkstra's algorithm in Python to find the shortest time to any destination from a given starting location known as the source node. We use travel time to measure edges, the connections between nodes, and map the shortest time to each node. The algorithm stores a variable for the current known shortest time to a given node by adding the distances along an arbitrary route. The variable is updated each time a shorter route is found to the given node. Once all possible routes are traversed, the node is marked as "visited" and the algorithm proceeds to the next node. Once all nodes are visited, the algorithm returns the shortest route to any destination from the source node.

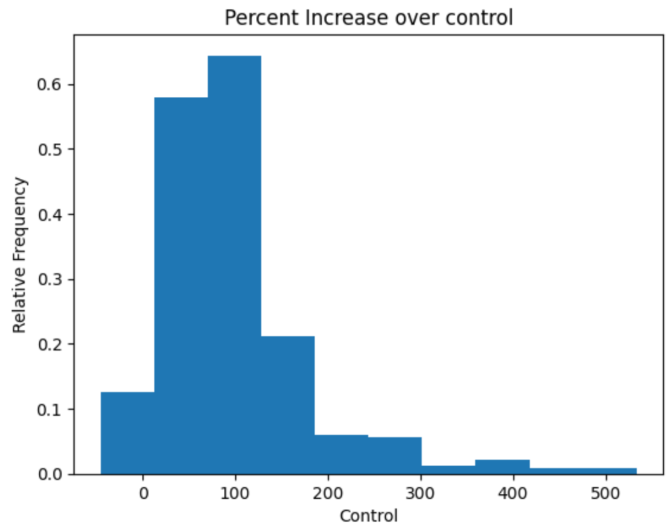
The second method is a least resistance algorithm comparing electrical flows and voltage to walking on a path. The algorithm simulates the flow of energy, which we term as voltage, and it gradually increases as the algorithm explores different paths. We use inputs of start and end to indicate which nodes the user wants to travel between. We include a boolean value to reflect the physical ability of the user, which indicates which data to analyze: disabled or non-disabled. The Python code outputs the total travel time and an array which includes the sequential node path used. The algorithm is centered around two sets of arrays that contain all adjacent nodes to the start and end nodes, respectively, showing all paths that can be completed within the current voltage (besides the initial values). From both the starting and ending nodes, the algorithm will continuously explore all potential paths, considering ones that have enough remaining voltage to continue. Every time a path is traversed, the energy required to travel said path is subtracted from the voltage value, determining how much energy is left to use. And so, as the algorithm runs, it builds a collection of possible paths from the starting and ending nodes to different points on the graph. Eventually, the two paths may converge, either at an intermediate node or in between nodes. When the paths meet in between nodes, the algorithm will check to see if the available voltage is enough to connect the two nodes. Then, the algorithm will then construct the final route by combining the start and reversed end paths. Finally, the time it takes to traverse the route is calculated by finding the sum of the times between all the nodes in the route.
Each algorithm has its own shortcomings, but using them in conjunction allows us to leverage each of their strengths. For example, our first method using Dijkstra's is apt in accurately calculating the fastest times to get from two nodes, but it is devoid of a memory mechanism to return the corresponding path between the nodes. Conversely, the second method is able to provide a path, but it is not as accurate as the first method. To capitalize on both the algorithms, we use the first method's results to verify the accuracy of the second method. This enables us to return the optimal path and ensure the travel times correspond to the most efficient option.

5 Results

The least-resistance algorithm achieved 98% accuracy when tested against Dijkstra's algorithm. This means it returns the most efficient sequence of paths from a starting point to a destination 98% of the time. We compare the mean travel times of the wheelchair user and the control non-disabled individual along possible paths.



The box plot demonstrates that the mean travel time of the wheelchair user is double that of the non-disabled. This can be attributed to paths that are entirely inaccessible, as depicted in the node map, and also the inefficiencies of ramps that wind and have sharp turns. The histogram below reflects the distribution of routes for a wheelchair user by the percentage increase in time over the control.



Some paths see as much as a 500% increase in travel time for wheelchair users. Node 15 to node 19, connecting Kerkhoff to Mathematical Sciences, has one of the largest percent increases in travel time for the wheelchair user because it uses many winding ramps that aren't in a productive direction. Similarly, the path from node 8 to 9, from the bottom of Jans steps to the top, requires the use of winding ramps, stretching travel time. Other cases like node 3 to 1 require U-turns to avoid stairs. To combat the former two, we propose a ramp connecting node 8 to 6 to 9 which would make the paths much more efficient. It would benefit students and visitors for signs to be posted near these junctions informing of nearby elevators, including the one in Ackerman Student Union. Introducing signage for ways around stairs and ramps is crucial for the efficiency of travel time. In cases like node 3 to 1, simple ramps with shallow slopes would be feasible construction projects that would greatly accelerate travel time.

We attempted to implement these changes and in turn make adjustments to our data for these paths. Based on our previous data, we were able to draw conclusions on the difference in time if these changes were implemented into the UCLA campus. Before these changes were made, the mean commute time between a path for a disabled person would be 390 seconds. For the path between 15-19, in building an elevator, the time would be much quicker, approximately 90 seconds, and in turn it would lower the mean commute time to 374 seconds. In building a ramp next to Jans steps, corresponding to path 8-9, the mean commute time would decrease, but not as significantly, to 383. And finally, between 4-1, creating a route through the Anderson parking structure would decrease the mean commute time to 385. Overall, with all 3 solutions implemented, we estimate that the mean commute time would decrease significantly, by 30 seconds, to 362 seconds. It is important to note that an elevator added to the path of 15-19 significantly lowers the mean traversal time because it is a pivotal node in the South side of the map that has many connections to other nodes. So, if the mean time of this path is decreased significantly, then many paths on this side of campus will also see a decrease in time, causing the overall mean traversal time to decrease.

6 Improvements

The first natural improvement of the model involves expanding the borders of the map to include more of UCLA campus. This would highlight more areas of inaccessibility and enable disabled students to find optimal paths to more nodes. The majority of south campus, including the medical school and biomedical library, as well as dormitories on the "the Hill" remain unaccounted in the model. These regions have especially complex layouts with varying elevations and would benefit from being fitted to the model. Another possible means of expansion is by including more indoor paths through buildings that could potentially makes edges shorter.

We can improve the data collection process for new regions by experimenting with automated methods including Google Maps API calls. The current process of documenting data by physically walking the routes is not feasible when expanding the borders of the included regions. Developing automated processes to calculate travel times between nodes would enable us to continuously capture more of UCLA's campus and the surrounding Westwood areas.

Moreover we'd like to refine the model by improving the accuracy of the least-resistance algorithm. We achieved 98% accuracy when using Dijkstra's model as a reference and we will work to continuing raising that figure. We'd also like to look into other modes of verification for testing the accuracy of the model outside of Dijkstra's. Finally, we're interested in including stochastic factors like traffic signals and congestion. These factors would likely be contingent on the time of day, and would optimize routes to circumvent congested pockets.

7 Conclusions

UCLA struggles to make its campus accessible to the physically disabled because of the many steep inclines and stairs used to optimize the hilly terrain. Physically disabled students and visitors are often lost when trying to navigate to their destinations, especially when there aren't accommodations in places like ramps or elevators. To combat this, we map a region of UCLA that sees high traffic, collecting data about the traits and distances between selected nodes. This allows us to depict which paths are inaccessible to disabled students. Using the data, we employ algorithms and techniques, including Dijkstra's and least-resistance, to calculate the most efficient path to a particular destination, based on an individual's physical ability. We were able to achieve 98% accuracy with our mathematical model, and depicted how wheelchair users face as much as 600% percent increases in travel times due to a lack of accommodations or inefficient ramps. Using the data we collected, we propose viable engineering solutions that would optimize campus area and wheelchair accessible paths. We seek to expand the scope of our model by including more of UCLA campus and adjacent areas of Westwood that garner high traffic. Moreover, we aim to increase the accuracy of the model and introduce means of automated data collection to streamline the path optimization process.

8 References

Kapoor, Anmol. "What Is Dijkstra's Algorithm? Here's How to Implement It with Example?" Simplilearn.Com, 21 Feb. 2023, www.simplilearn.com/tutorials/cyber-security-tutorial/what-is-dijkstras-algorithm.

Maverick, John. "How Fast Can a Manual Wheelchair Go: Helpful Guide 2023." Wheelchairsvilly.Com, 24 Jan. 2023, wheelchairsvilly.com/how-fast-can-a-manual-wheelchair-go/.

Metterhausen, Fred. "Elevation Calculator: Find Your Current Elevation, an Address, or a Point on the Map." Elevation Calculator: Find My Elevation on a Map., www.mapdevelopers.com/elevation_calculator.php. Accessed 13 June 2023. Systems, UCLA Facilities Management Information. Map.Ucla.Edu, map.ucla.edu/. Accessed 13 June 2023.

Wilson, bin J. Introduction to Graph Theory. Prentice Hall, 2015.

9 Links

Link to Script

Link to Data

Link to Presentation