An IP for a Summer Day

Vinayak Sharma
Arizona State University
vinayak1998th@asu.edu

Kavan Nitin Vasani
Arizona State University
kvasani@asu.edu

Yukta Sarode
Arizona State University
ysarode@asu.edu

Abstract

Various navigation services provide their users the ability to follow the shortest path from their source to destination. But in a state like Arizona with a very high skin cancer rate, avoiding the sun is a major priority for pedestrians. We propose a system that presents a weight factor that balances between distance and exposure to the sun. We presents a multi-parameter variant of the Shortest Path called the 'Compound Shortest-Path' that accounts for sunlight intensity. We define all the components of this new method along with different variants and their respective trade-offs. The team introduces the LP for this problem and solve it using Dijkstra's Algorithm, a Primal-Dual algorithm. This proves the feasibility of the LP and gives us the optimal solution in the same time as Shortest-Path. Finally, we discuss the limitations of our approach and possible future work.

1 Introduction

The risk of cancer due to sun exposure has motivated the research to find the shortest path from point A to B. Even though the complexity and efficiency of network analysis problems have increased, and more systems have come up with ways to get the shortest path in a very small amount of time but current network analysis technologies fail to incorporate other factors in their algorithm. Therefore, we put forward a new approach which takes into account not only the distance to define an efficient path, but also takes into consideration other factors to determine the optimally of a path.One parameter that we have considered is the exposure to the sun. Additionally, we consider an instance of walking from the Education Lecture Hall (EDC) to the Hayden Library at ASU as an example.

2 Problem Statement

The mathematical definition of the 'Compound Shortest Path Problem' is as follows:

Definition (COMPOUND SHORTEST-PATH). *Given a graph G = (V,E), edge lengths* $w : E \to \mathbb{R}_{\geq 0}$*, a light intensity function* $l : E, T \to \mathbb{R}_{\geq 0}$*,source and destination nodes* $s, t \in V$ *and a desirability parameter* $\alpha \in \{-1, 1\}$*, find a path P from s to t such that the total length of the path is minimized and the total light intensity is optimized based on* α*.*

3 Light Intensity Function:

One of the main novel Introduction in our project is the light intensity function. We have defined the light intensity function to be based on the time of day and the path. While we consider many approaches to define the light intensity function, the main idea was to break into into a linear combination of 2 factors following factors. We multiply the function with the weight of the edge so the longer paths with sun exposure are further penalized. Here λ_t and λ_p are scaling factors.

$$l(e,T) = \lambda_t . I_{time}(t) \times \lambda_p . I_{edge}(e) \times w(e) \text{ ; } where\ \lambda_t, \lambda_p \in [0,1] \tag{1}$$

The factors are detailed below:

3.1 Time of Day Intensity Factor

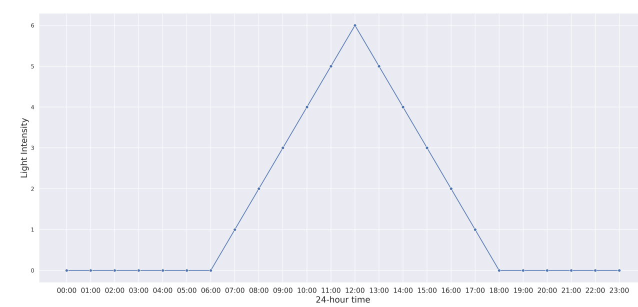

Figure 1: Graph for the Time of Day Intensity Factor

While looking at the possible approaches on how to incorporate time of day into the calculation of light intensity, we defined a simple yet effective function that takes the current time in *hh:mm* 24 hour format and returns a value between 0 and 6. We divide the whole term by 6 so as to obtain a value between 0 and 1 to express the intensity of sunlight as a percentage The function is defined as follows:

$$I_{time}(t) = \begin{cases} \frac{6-|t-12|}{6}, & \text{if } |t-12| \leq 6 \\ 0, & \text{otherwise} \end{cases}$$

It is interesting to note that due to this function definition, at nighttime (between sunset and sunrise) our function returns 0 and makes the Compound Shortest Path Problem equivalent to the Shortest Path Problem.

3.2 Path Specific Intensity Factor

While the Time of Day Intensity Factor can be mathematically tweaked and defined with relative ease, the path specific factor is the major hurdle in scaling this approach as it requires

manual data collection. However, it far easier to do today as compared to the past owing to the recent advances in drones and also 3D simulations. We can see an example with what NVIDIA has done with their AI City Challenge [1].
After extensively looking into the matter we classified the various approaches to this problem into 2 main categories and the approaches under them. While this list is by no means exhaustive, we believe it covers a good set of approaches that can be used be expanded upon in future work.The approaches are as follows:

Based on Output Type:

Using Binary Labels:

The labelling can be simplified into binary labels using a threshold metric, for example we have used any path that exposed us to sunlight for more that 5 seconds as a positive label and the rest as negative.

Using Luminescence Readings:

The exposure of sun to a path can be measured by a luminescence reader and that can give accurate readings for the sun exposure for a path. We can use the average reading(or any other statistically significant property) as the path label in such a case.

Based on Data Collection Method:

Manual Sampling:

The collection of sunlight conditions via manual sampling is a very tedious process and is not scalable. However, it is the most accurate method of data collection.

Simulated or Inferred Sampling:

While this a far more scalable approach that relies on new groundbreaking research in field of 3D simulation and/or Artificial Intelligence, it is not as accurate as the manual sampling method. However, it is extremely useful and promising in the case of large scale data collection.

4 Desirability Parameter:

We introduce a desirability parameter $\alpha \in -1, 1$ that serves the dual purpose of assigning either a penalty or a reward to sunlight. This was done in order to account for things like winter days where the warmth of the sun is desired. The desirability parameter we defined as follows:

$$\alpha = \begin{cases} +1, & \text{sunlight penalty} \\ -1, & \text{sunlight reward} \end{cases}$$

We considered using a Real value for α but found that using the Light Intensity Function as defined in Equation (1) to control the exact weight age was a better approach and reduced the complexity of the problem significantly.

5 Linear Program (LP):

5.1 Cost Function:

$$min \sum_{i \in E} (w(e) + \alpha . l(T,e)) . x_e \tag{2}$$

Where,"T" defines the time of day in a 24 hour format and is otherwise treated as a constant when starting the LP.
Notice that in this state the cost function is similar if not identical to the cost function of the Shortest Path Problem with modified weights. That is to say that this cost function is equivalent to the cost function of the Shortest Path Problem with the following modified weights:

$$w(e) \to w(e) + \alpha . l(T,e)$$

5.2 Variables:

The Variable definition is identical to Shortest Path and consists of a set of variables $x_e \in \{0,1\}$ for each edge $e \in E$. They define whether an edge is part of the path or not, i.e:

$$x_e = \begin{cases} 1 & ; e \in \text{ path} \\ 0 & ; e \notin \text{ path} \end{cases}$$

5.3 Constraints:

Common constraints for both the Shortest Path Problem and the Compound Shortest Path Problem (Sunlight Path Problem) are:

$$\sum_{(u,v) \in E} x_{(u,v)} - \sum_{(v,u_o) \in E} x_{(v,u_o)} = 0 \; ; \forall (u,v) u, v, u_o \in V / \{s,t\}$$

$$\sum_{(u,t) \in E} x_{(u,t)} = 1 \; ; \forall u \in V / \{s,t\}$$

We have also postulated <u>3 variants</u> of the problem that we believe are of practical use and have additional constraints that are specific to the variant. These variants are as follows:

Unconstrained:

1. This is the most basic form and places no constraints in relation to the sunlight present in the path,
2. For any instance where shortest path is feasible, our problem is also feasible because we have not changed the constraints. Just the cost function is modified and the feasibility does not depend on the cost function but only the constraints.

Partially Constrained:

1. In this case where we define a threshold for the total permisible sunlight exposure in the path,
2. In general the additional constraint would lead to a tighter bound on the LP and hence have a smaller feasible region.
3. *Constraint:*

$$\sum_{e \in E} l(T,e) . x_e \leq \text{threshold};$$

Extremely Constraint:

1. In this case we constrain the LP to only allow paths with no sunlight exposure at all,
2. In general the additional constraint would lead to a tighter bound on the LP and hence have a smaller feasible region.
3. *Constraint:*

$$\sum_{e \in E} L(T,e).x_e = \lim_{T \to 0} T$$

Each of these cases define a different level of sensitivity to the optimization problem and can be used based on user preference at the cost of complexity, feasibility and computational time.

6 Primal Dual Formulation

To demonstrate what effect our LP has on computational complexity by adding the additional constraint, we explore the changes to existing Primal-Dual algorithms such as Dijkstra. We cover the unconstrained and partially constraint cases.

Primal Formulation:

$$\min\left(\sum_{e \in E} (w(e) + \alpha.l(T,e)).x_e\right)$$
$$\sum_{(u,v) \in E} x_{(u,v)} - \sum_{(v,u_o) \in E} x_{(v,u_o)} = 0 \quad ; \forall u,v,u_o \in V/\{s,t\}$$
$$\sum_{(u,t) \in E} x_{(u,t)} = 1 \quad ; \forall u \in V/\{s,t\}$$
$$x_e \geq 0 \quad ; \forall e \in E$$
$$\text{(Only for Partially Constrained Case)}$$
$$-\sum_{e \in E} l(T,e).x_e \geq -C_{threshold}$$

Dual Formulation - Unconstrained:
This is created by ignoring the threshold constraint.

$$\max(\pi_t)$$
$$\pi_v - \pi_u \leq w(u,v) + \alpha.l(T,(u,v)) \quad \forall (u,v) \in E$$
$$\pi_v \leq w(s,v) + \alpha.l(T,(s,v))$$
$$\pi_v \in R \quad ; \forall v \in V \setminus \{s\}$$

The dual is almost equivalent to the dual of shortest path with the exception that the weights appear to be increased linearly by the light intensity factor. This implies that there computationally no difference between this case and a shortest-path with larger edge weights. Therefore this is a practically viable solution for every case where Dijkstra is practical.

Dual Formulation - Partially Constrained Case:

$$\max(\pi_t - C_{threshold}.\pi_l)$$
$$\pi_v - \pi_u - l(T,(u,v)).\pi_l \leq w(u,v) + \alpha.l(T,(u,v))$$
$$\forall (u,v) \in E$$
$$\pi_v - l(T,(s,v)).\pi_l \leq w(s,v) + \alpha.l(T,(s,v))$$
$$\pi_l \leq 0$$
$$s\pi_v \in R \quad ; \forall v \in V \setminus \{s\}$$

This case is extremely interesting and has profound implications when looking at the primal and dual with Complementary Slackness. We know that as the variable x_e is either a 1 or a 0, for every edge in the path, the dual constraint must be tight (As seen in class). Also the tightness of the set of constraints creating π_v means we do not get any useful information for this variable. However, the thresh hold constraint is not tight and yields the following:

$$\pi_l.\left(\sum_{e \in E} (l(T,(u,v)).x_e) - C_{threshold}\right) = 0$$
$$\implies \pi_l = 0 \text{ or } \sum_{e \in E} (l(T,(u,v)).x_e) = C_{threshold}$$

This creates 2 distinct possibilities for optimal solutions. The other point to note is that when our light intensity summation is equal to the threshold value, π_t can take any positive value. However, our cost function is inclined to minimize π_t and as all values are valid it will take the minimum possible value of 0. This means that for all optimal solutions, the value of π_t must be equal to 0.This assertion is mathematically verifiable by comparing the optimal solution of the dual and primal.

Primal Optimal Solution:

$$\sum_{e \in E} (w(e) + \alpha.l(T,e)).x_e)$$
$$\implies \sum_{e \in P} (w(e) + \alpha.l(T,e)) \quad ;\text{where P is the optimal path}$$

Dual Optimal Solution:

$$\pi_t - C_{threshold}.\pi_l$$
$$\implies \pi_t \quad ;(\text{as } \pi_l = 0$$

By induction we can show that π_t is equal to

$$\sum_{e \in P} (w(e) + \alpha.l(T,e)) \quad ;\text{where P is the optimal path}$$

Therefore we can see that our Primal and Dual optimal solution are equal. The reverse is also a trivial proof and show optimality to only be possible at $\pi_l = 0$. This means that after applying this substitution we get the following dual:

$$\max(\pi_t)$$
$$\pi_v - \pi_u \leq w(u,v) + \alpha.l(T,(u,v)) \quad \forall (u,v) \in E$$
$$\pi_v \leq w(s,v) + \alpha.l(T,(s,v))$$
$$\pi_v \in R \quad ; \forall v \in V \setminus \{s\}$$

Which is identical to the dual of the unconstrained case, with a smaller feasible area due to the additional threshold constraint. This implies that the optimization steps remain the same as Shortest-Path even in this more constraint scenario albeit with fewer possible feasible paths.

7 Example Instance:

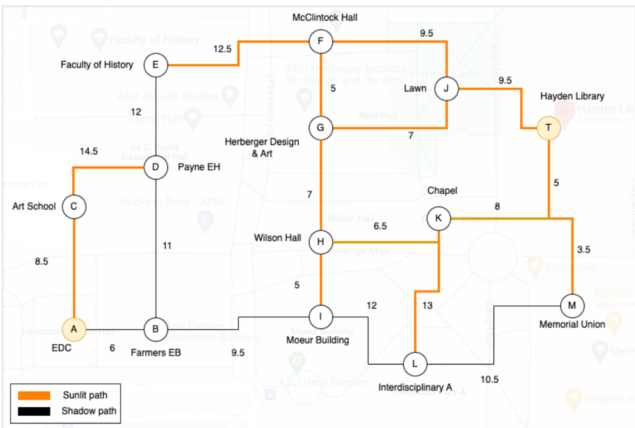

Figure 2: Example Instance Graph

For clarity and proof of applicability we have created an *'Example Instance'* to test our solution out. The graph is shown in Figure (2) and the edges are labelled with their respective lengths and sunlight values.

The configuration used for this instance utilizes the simplest form of the problem, i.e. the *Unconstrained* variant with Binary labels for the I_{edge}(3.1).

We select such a set of parameters because it is the simplest and most intuitive form of the problem and avoids the risk of going into infeasible regions.

7.1 Instance LP

Function Definitions:

$$T = 16:00$$

$$I_{time}(t) = \frac{6-|16-12|}{6} = \frac{1}{3}$$

$$I_e(p) = \begin{cases} 1, & \text{if sunlight exposure on path} \geq 5sec \\ 0, & \text{otherwise} \end{cases}$$

$$L(T,e).x_e = I_{time}(t) * I_{edge}(e) * w(e) = \frac{1}{3} * I_{edge}(e) * w(e)$$

$$\alpha = +1$$

LP:

$$\min\Big(\sum_{e \in E}(w(e) + \alpha.l(T,e)).x_e\Big)$$

$$\sum_{(u,v)\in E} x_{(u,v)} - \sum_{(v,u_o)\in E} x_{(v,u_o)} = 0 \quad ; \forall u, v, u_o \in V/\{s,t\}$$

$$\sum_{(u,t)\in E} x_{(u,t)} = 1 \quad ; \forall u \in V/\{t\}$$

$$x_e \geq 0 \quad ; \forall e \in E$$

7.2 Instance Solution

Due to the nature of the shortest path LP, it is infeasible to use Simplex manually as the constraint matrix 'A' would be of the dimension $|V| \times |V| = 13 \times 13 = 169$ which is not practical to solve by hand.

Hence, we propose an alternative method of solving the instance while still proving LP feasibility - Dijkstra's Algorithm. As Dijkstra's Algorithm is a Primal-Dual algorithm, if a s-t path can be found using Dijkstra's it would imply the existence of a feasible and optimal solution to the LP of the underlying graph. Dijkstra also works here due to the use of the 'weight transformation' outline in section (5.1) $w(e) \to w(e) + \alpha.l(T,e)$ with the standard Shortest-Path optimization.

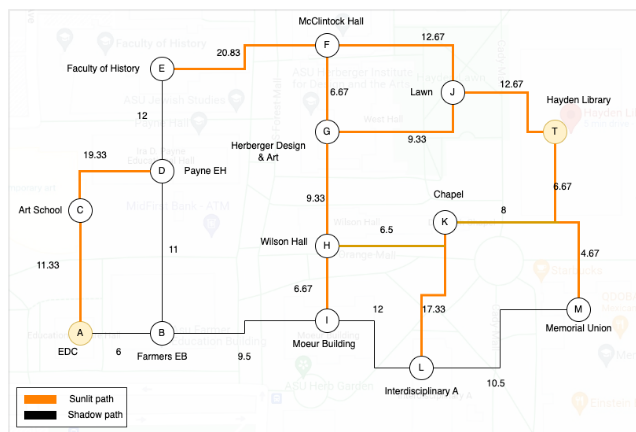

Figure 3: Graph with adjusted weights accounting for sunlight

In figure 3, the weight of the edges with exposure to sunlight have been adjusted as per the definition of the time function by taking the instance of intensity of sunlight at 16:00. Running Dijkstra's on the graph considering the three variations of the constraints, when we consider the unconstrained version, we see that the shortest path is A->B->I->H->K->T with a cost of 40. Now for the partially constrained instance, we add a threshold of 50 and we see that the moment sunlight is accounted for, the shortest path changes to A->B->I->L->M->T with a cost of 49.43 while the shortest path for the unconstrained variant has a cost of 57. For the instance where we have an extreme constrained where no path with sun exposure is not allowed, we see that no path satisfies this constraint.

8 Conclusion

The main contribution of this work is the proof that we can create multi-objective shortest path formulations without any effect on the optimization steps or the computational time. The only effect we saw was the reduction of the feasible region and hence fewer solutions, this fact held true even when adding new constraints to the primal.

We then used Dijkstra's Algorithm which is derived from the primal of shortest path to prove the validity of our solution and demonstrated that indeed a new solution was obtained post addition of the new factor (sunlight).

The floor remains open to finding additional optimization factors such as traffic, handicap friendliness, etc that follow the same properties as our sunlight example and address very real needs. The main limiting factor that we find in all of these cases seems to stem from data collection rather than computational complexity.

References

[1] Hang Chu, Daiqing Li, David Acuna, Amlan Kar, Maria Shugrina, Xinkai Wei, Ming-Yu Liu, Antonio Torralba, and Sanja Fidler. Neural turtle graphics for modeling city road layouts, 2019. URL: https://arxiv.org/abs/1910.02055, doi:10.48550/ARXIV.1910.02055.