# Expanding Select Bicycle Lanes in Boston Using Optimization Model Checking

Enze Yan
enze@bu.edu

May 4, 2016

**Abstract**

The city of Boston has dedicated tons of energy on developing a better safety environment for biker, including adding more bike lanes to more streets. But what are the priority streets to be picked as to install bike lanes in order to achieve a higher connected bike lane network? With the help of automated reasoning tool such as Z3 , I propose a model based on propositional logic and solve the model that satisfies the given constrains, which in return, suggests the specific street to put on bike lane.

## 1 Introduction

Bicycles have become a critical part of Boston transportation system. Each day from dawn till dusk, there are tons of people commuting to office and school on their bikes, not to mentioned on the weekends, families and recreational bikers love to have a fun and enjoyable ride together while exploring this historical city. Because of the increasing usage of bicycle, the city is in need of an organized plan in order to better transit from car-dependent street network to bicycle-friendly roads.

The City of Boston launched its initial *Boston Bicycle Program* in 2007. The city has seen unprecedented impact - "we have gone from being called one of the worst cities for cycling in the country to one of the best," says the *Boston Bike Safety Report* published in 2013. The city has doubled ridership through introducing nearly 60 miles of on-street bicycle lanes, 1000 new bike racks, and the innovating New Balance Hubway bike share system.

However, despite of the booming expansion of bicycle facilities, the city had unfortunately experienced five fatal bicycle incidents during the summer and fall of 2012. In order to prevent similar accidents from happening in the future, the city introduced *Boston Bikes* which dedicates to transform Boston into "safe and inviting conditions for all residents and visitors." The program also includes a 30-year vision and 5-year action plan to support its safety goal, which is to decrease bicycle crashes by 50 percent by the year of 2020.

Since the 5-year action plan considers adding more on-road bike lanes to the city traffic network, this following report will first analyze the impact of bike lanes on cyclist collisions. Then the paper is followed by systematically identifying a comprehensive network of bicycle routes through the city of Boston based on the database of accidents involving bicycle crash. I hope this project will introduce few unique approach to model real life problems based on propositional logic and to solve them using current automated reasoning tools.

# 2 Data

## 2.1 Data Collection

Before analysis, I have collected and compiled a cooperative bike crash dataset from a variety of sources. This dataset includes: Boston Bike Crash Data contains self-reported bicycle accidents provided by Boston Police Department, Boston EMS Crash Data contains responses to bicycle accidents provided by Boston Emergency Medical Services and local emergency departments, and Boston Bike Collision Database provided by Harvard Dataverse. The above dataset is collected between 2009 and 2012, and each entry contains detailed information about each bicycle crash incident, including the geographical location and whether the crash is on the bike lane or off the bike lane, which in turn could be used to prove the impact of existing bike lanes.

Other relative datasets including Existing Bike Network provided by BostonGIS, and Boston OpenStreetMap (OSM) presented by Metro Extracts. Existing Bike Network outlined the information for each existing bike lane in the city of Boston - the length of the bike lane, and the year that it was installed which is important information to use as comparison with the entry in bike crash data. Boston OSM is used to generate all the intersections between each individual road and street, and return the list of pair of 2 roads that share an intersection.

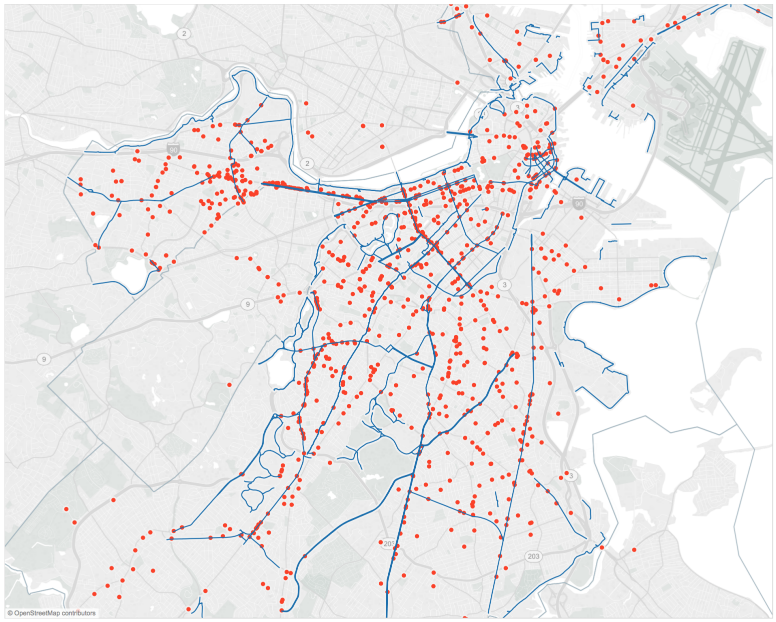

Figure 1: Existing Bike Lanes Map with Bike Collision

Figure 1 maps out the existing bike lanes (blue line) within the city of Boston, and it also includes all of the bike crash site (red dot) that was gathered before on top of the map. This map well indicates the current bike lane network – readers could easily notice the network is not connective, and there are still isolated roads that are not included in the bike network. This gives us incentive to come up with a series of plan for placing bike lanes to achieve better connection.

## 2.2 Data Visualization

Before diving into modeling the problem, I exam the data to make sure that there is actually a positive impact of bike lanes on bike crashes. I particular selected all the bike crash incidents that was happened on roads where a bike lane was presented. There are typically two types of crash incidents: one is where the crashes were happened on the bike lane, the other is where the crashes were happened off the bike lane while there was a bike lane on the road. Comparing the crash incidents frequency histogram chart on Figure 2, one could conclude that there are significantly less collisions occurred within the bike lane, compared to those collisions occurred off the bike lane when there is actually a bike lane presented on the same road. This also suggests the importance of constructing a comprehensive bike lane network.

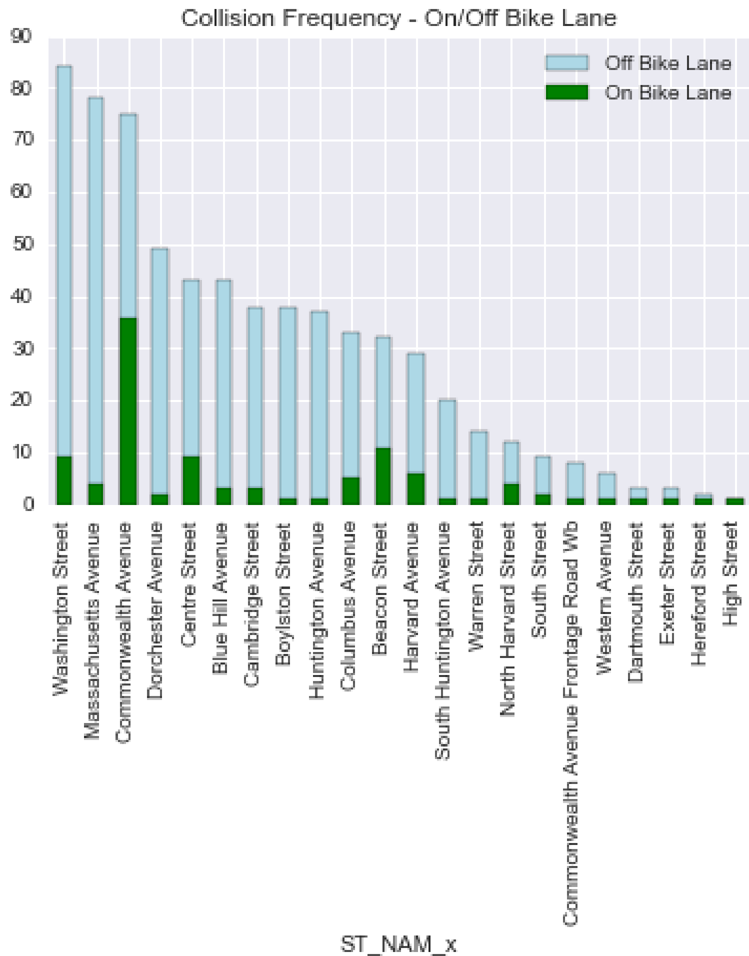


Figure 2: Collision Frequency between On-Lane and Off-Lane

I would also like to understand if there is any correlation between the length of the streets, with or without a bike lane, and the amount of bike crash accidents. I have normalized both the data recording the length of the street and the crash incidents. Below, the correlation plot (Figure 3) does not provide strong evidence on deciding whether the correlation is strong. Meanwhile, calculating the Pearson product-moment correlation coefficient, which is 0.71, does suggest the positive relationship: the amount of incidents increases as the street distance increases.

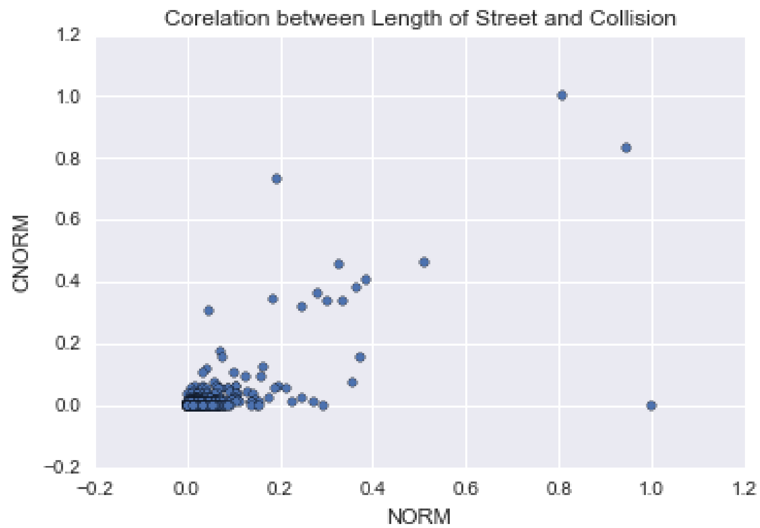


Figure 3:

# 3 Problem Description

Now that we have seen the importance of having a bike lane – it is of the same importance to build a series of bike lanes so that each bike trip could be taken entirely on the bike lanes instead of having missed bike lanes on some other roads. Thus, our goal is to build a better connected bike lane network based on the existing one. To be specific, what are the main streets that need to have a bike lane, if it hasn't have one yet, so that the system could achieve more connective routes?

# 4 Problem Modeling

First, in order to achieve the connective routes, the streets that have the most amount of intersections with other streets should have the priority to add a bike lane, because by deduction, these streets are the hub that link to the other roads. The problem I have described above could be then modeled as follow: given all the roads, divide the roads into pairs $(i, j)$ where both 2 of the streets (street $i$ and street $j$) in each pair share an intersection or cross road. Then, each of the street pairs will have one of the 3 states $(x_i, x_j)$, where $x_{i,j}$ could either be 0 or 1 which indicates if street $i$ or street $j$ currently have a

- State 1 indicates both streets (street $i$ and street $j$) have bike lanes:
  $S_1 := (1, 1)$
- State 2 indicates only 1 street of the pair (either street $i$ or street $j$) has bike lane:
  $S_2 := (0, 1)|(1, 0)$
- State 3 indicates neither street (neither street $i$ nor street $j$) have a bike lane:
  $S_3 := (0, 0)$

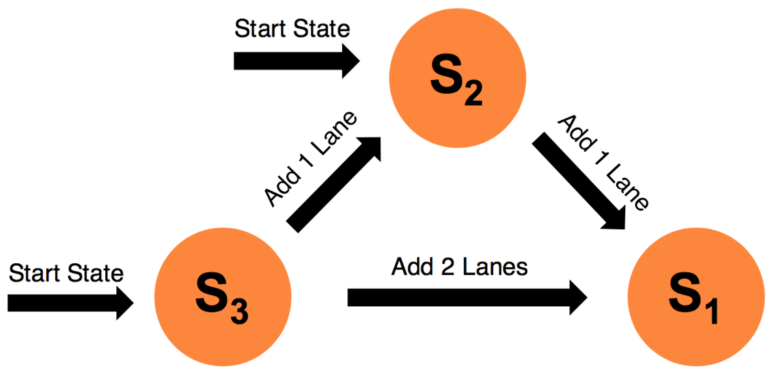


Figure 4: States Transformation

Figure 4 suggests all the possible routes transition between the 3 different states listed above. Each of the street pairs starts at either State $S_3$ or State $S_2$, and then move to the next state by adding 1 or more bike lanes. The goal is then changed to achieve most of the State $S_1$, because the more State $S_1$ we have in our system, the more connected bike lanes we have. In another word, the goal is to have as less State $S_2$ as possible because State $S_2$ indicates the less degree of connections for our bike lane network.

## 4.1 Optimization Model

I propose the following linear model objective function:

$$argmin_{\ i,j \in N} \sum \mid x_i - x_j \mid$$

where $N$ is the entire set of street pairs that share an intersection, $|x_i - x_j|$ expresses as the absolute value of $(x_i - x_j)$. The constrains are:

- $x_{i,j} \in \{0, 1\}$

- $x_1 + x_2 + x_3 + x_4 + ... + x_N \leq k$ where $k$ is a constant that user specifies to indicate the amount of variables that is allowed to change. This constrain limits the amount of bike lanes that needed to be added to system to prevent the solution returns change of all the variables (add all bike lanes to each street).

Since we would like to minimize the amount of State $S_2$, my optimization model would then penalize each occurrence of $S_2$ – every time the model introduces a new State $S_2$, it adds 1 to the objective function which we try to minimize.

# 5 Solving with Z3

The above model is considered more for optimization compared to satisfiability modulo theory (SMT/SAT) problem, but SMT solver Z3 is still considered as one possible solver for this problem. Z3's main functionality is to check the satisfiability of logical formulas over one or more theories. Z3 can produce models for satisfiable formulas. Yet in many cases, as suggested by Microsoft Research, arbitrary models are insufficient and applications are really solving optimization problems: one or more values should be minimal or maximal. The solution to the above model checking is exposed by Z3Opt, that lets users formulate objective functions directly with Z3. Z3Opt is a part of the SMT solver Z3. It allows users to pose and solve optimization problems modulo theories. Z3Opt provides a portfolio of approaches for solving linear optimization problems over SMT formulas.

To give a first idea, we could ask to optimize the above minimization objective function with only 4 variables, $\{x_1, x_2, x_3, x_4\}$, where each variable represents one street that has no bike lane, since we only want to ADD bike lanes to those without one but not change those streets who already have one. $x_1$ is connected with 1 street that has a bike lane, $x_4$ and $x_2$ intersect but neither of them has a bike lane, $x_2$ and $x_3$ intersect but neither of them has a bike lane, and $x_1$ is connected with 1 street that has a bike lane.

```
(declare-const x1 Int)
(declare-const x2 Int)
(declare-const x3 Int)
(declare-const x4 Int)
(define-fun absolute ((x Int)) Int
  (ite (>= x 0) x (- x)))
(assert (or (= x1 0) (= x1 1)))
(assert (or (= x2 0) (= x2 1)))
(assert (or (= x3 0) (= x3 1)))
(assert (or (= x4 0) (= x4 1)))
(assert (<= (+ x1 x2 x3 x4) 3))
(minimize (+ (absolute (- 1 x1) )
  (absolute (- x4 x2) )
  (absolute (- x2 x3) )
  (absolute (- 1 x3) ) ))
(check-sat)
(eval x1)
(eval x2)
(eval x3)
(eval x4)
```

Figure 5: Minimize with Z3Opt

With the last *assert* constrain, the objective function from above is $argmin(|1 - x_1| + |x_4 - x_2| + |x_2 - x_3| + |1 - x_3|)$, which tells the solver to pick 3 variables out of 4 that is allowed to add bike lanes. The output from the Z3 solver:

```
(+ (ite (>= (- 1 x1) 0) (- 1 x1) (- (- 1 x1)))
   (ite (>= (- x4 x2) 0) (- x4 x2) (- (- x4 x2)))
   (ite (>= (- x2 x3) 0) (- x2 x3) (- (- x2 x3)))
   (ite (>= (- 1 x3) 0) (- 1 x3) (- (- 1 x3)))) |-> 1
sat
1
1
1
0
```

Z3 finds a solution and returns the optima. Z3 thinks this model is satisfiable and return a potential possible solution, $\{x_1 = 1, x_2 = 1, x_3 = 1, x_4 = 0\}$, which suggests to add bike lanes to $\{x_1, x_2, x_3\}$ but not $x_4$ in order to achieve the most connective routes. One could then map what each variable represents which street to provide suggestions on where to add bike lanes. Each run of the solver theoretically should return the difference solutions as the input size increases.

## 5.1 Performance

For a complete solution to our original problem, which contains 500+ variables, I wrote a Python script with Z3 API. Detailed execution steps is documented inside the README file. Of course, the runtime is extremely slow since the input data is extremely large. In order to comprehend the performance of Z3 on optimization problem, I chop the file into smaller inputs with increment order. Figure 6 shows the runtime of the solver versus the size of the input (the amount of variables needed to address):

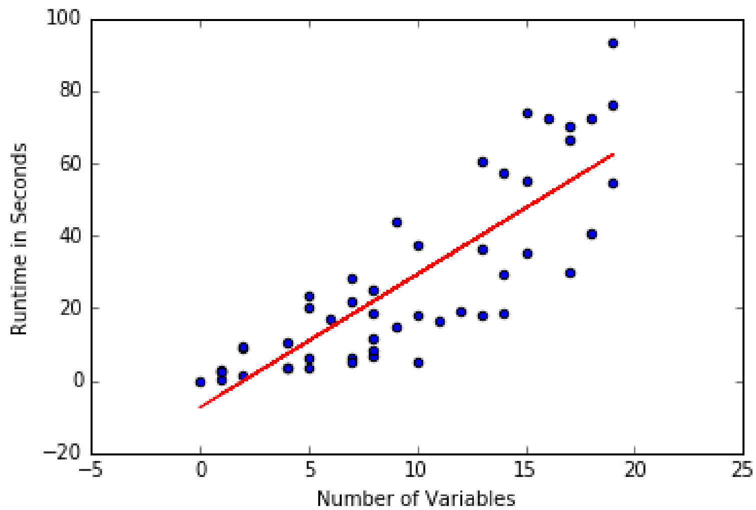


Figure 6: Z3 Performance with Various Number of Variables

By fitting into a linear regression line, one could conclude that the runtime increases linearly with the increase of size of the input variables, given the same objective model.

# 6 Future Work

## 6.1 Solving with dReal

**dReal** is an automated reasoning tool which focuses on solving problems involving nonlinear real functions over the real numbers. We could then transform the above linear model into a non-linear model and solve the objective function using dReal.

### 6.1.1 Alternative Optimization Model

I propose the following non-linear model objective function:

$$argmax_{\ i,j \in N} \sum (-1)^{x_i + x_j}$$

where $N$ is the entire set of street pairs that share an intersection. The constrains are:

- $x_{i,j} \in \{0, 1\}$
- $x_1 + x_2 + x_3 + x_4 + ... + x_N \leq k$ where $k$ is a constant that user specifies to indicate the amount of variables that is allowed to change. This constrain limits the amount of bike lanes that needed to be added to system to prevent the solution returns change of all the variables (add all bike lanes to each street).

Here, instead of penalize each occurrence of $S_2$, I reward State $S_1$, which has the most connective bike lanes. The objective function could be understood as reaching the highest reward score with the most amount of State $S_1$. A sample run of the solver using dReal could be implemented as follow:

```
(declare-const x1 Real)
(declare-const x2 Real)
(declare-const x3 Real)
(declare-const x4 Real)
(assert (or (= x1 0) (= x1 1)))
(assert (or (= x2 0) (= x2 1)))
(assert (or (= x3 0) (= x3 1)))
(assert (or (= x4 0) (= x4 1)))
(assert (<= (+ x1 x2 x3 x4) 3))
(minimize (+ (^ -1 (- 1 x1) )
  (^ -1 (- x4 x2) )
  (^ -1 (- x2 x3) )
  (^ -1 (- 1 x3) ) ))
(check-sat)
(eval x1)
(eval x2)
(eval x3)
(eval x4)
```

Figure 7: Maximize with dReal

Notice that we have declare the variable to be *Real* number between $[0, 1]$ instead of *Integer*. The results are real numbers with decimal points. More constrains need to be added in order to decide what are the variable to keep for adding bike lanes. However, dReal does not have Python API and such has difficulty for user to feed in large inputs with large amount of variables.

The Python binding feature could be developed in the future. After one could compare the performance between Z3 and dReal on solving the same type of modeling question.

### 6.2 Other Dataset

I encourage the future work to include more datasets. Data that provides more information about the conditions of each street and road could potentially add more constrains to the model. For example, mobile application *Strava* provides bikers with accurate information on each street's condition which also indicates whether this street is suitable for biking. We could extend this information by formulating constrains to our model defined above, and in return the results are more beneficial for recognizing what streets are of the priority to add bike lanes.

## 7 Acknowledgement

Special thank to Professor Kfoury and Professor Lapets for assistant on this project.

## 8 References

1. Boston Bikes, http://www.cityofboston.gov/bikes
2. Figure 1, https://public.tableau.com/profile/asross#!/
3. http://rise4fun.com/z3opt/tutorialcontent/guide#h26
4. https://dreal.github.io/
5. https://github.com/Z3Prover/z3