FIGHTING CONGESTION AND POLLUTION

WITH SELF-DRIVING CARS

Jackson Romero
Fox Chapel Area High School

Null Hypotheses:

1. Increasing the **number of smart systems** in a grid will not significantly **reduce traffic congestion**.
2. Increasing the **% of smart lights** in a grid will not significantly **reduce traffic congestion**.

Alternate Hypotheses:

1. Increasing the **number of smart systems** in a grid will significantly **reduce traffic congestion**.
2. Increasing the **% of smart lights** in a grid will significantly **reduce traffic congestion**.

PROCEDURE

1. Research the fundamentals of grid design
2. Create an effective model
3. Introduce smart vehicles
4. Introduce smart lights
5. Bug Fix / Quality of Life Improvements
6. Analyze results

```
iterate():void {

        this.currentIteration++;

        if(this.gridIteration == this.iterations && this.curre

            this.currentIteration--;
            this.gridResults[this.gridIteration-1] = testImpro
            this.currentIteration++;

            console.log(`ALL FINISHED`);
            console.log(JSON.stringify(GridController.gridResu
            paused = true;
            playPauseButton.elt.innerHTML = 'Play ▯';
        }
        else {
            //if this is the run after the baseline, run agai
            if(this.currentIteration == 2) {
                this.reset();
                console.log(`run adjusted grid ${this.gridIte
                for(let i = 0; i<GridController.Cars.length; i
                    GridController.Cars[i].reRoute = true;
                }

                for(let i = 0; i<GridController.Lights.length
                    GridController.Lights[i].smartLogic = tru
                }
            }

            //otherwise if we have baseline and full logic ru
            else if(this.currentIteration == 3) {
                clearInterval(runInterval);

                this.currentIteration--;
                this.gridResults[this.gridIteration-1] = testI
                this.currentIteration++;
```

THE GRID

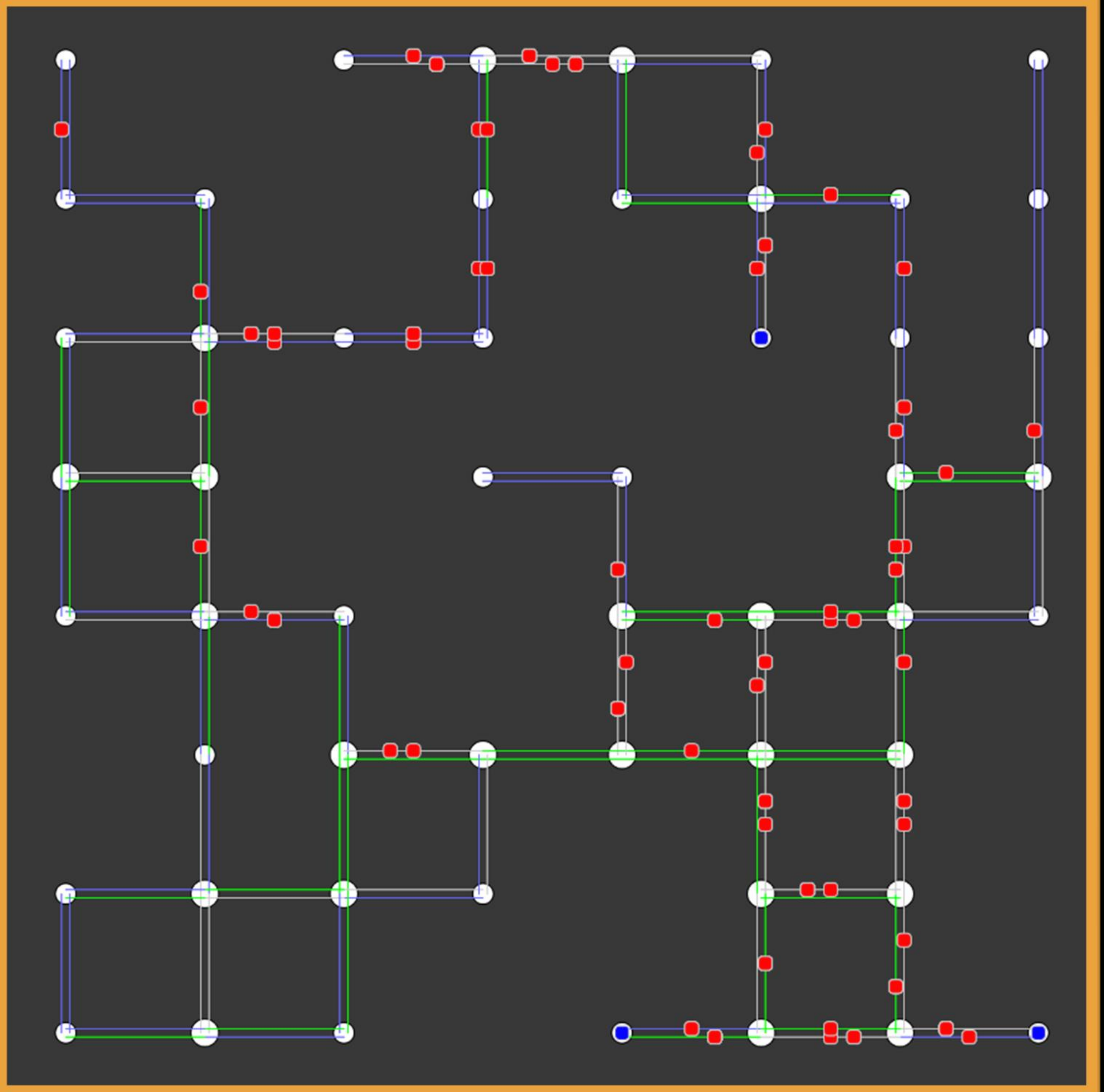
OVERVIEW

1. **Lights**
    1. Laid out in a grid
    2. Optional smart-light functionality
2. **Roads & lanes**
    1. Connect lights
    2. Lanes are one-way, roads are parent of 2 lanes
    3. Lanes keep track of queues and speeds
3. **Cars**
    1. Have entrance and exit
    2. Optional smart-rerouting functionality
    3. Actually control queues

PRIORITY QUEUE

1. Implemented via a **heap**
2. Keeps track of **all events** and **when** they are to execute
3. Allows for **detailed logging** of events and **path tracing**
4. Was crucial in **bug-fixing** the program

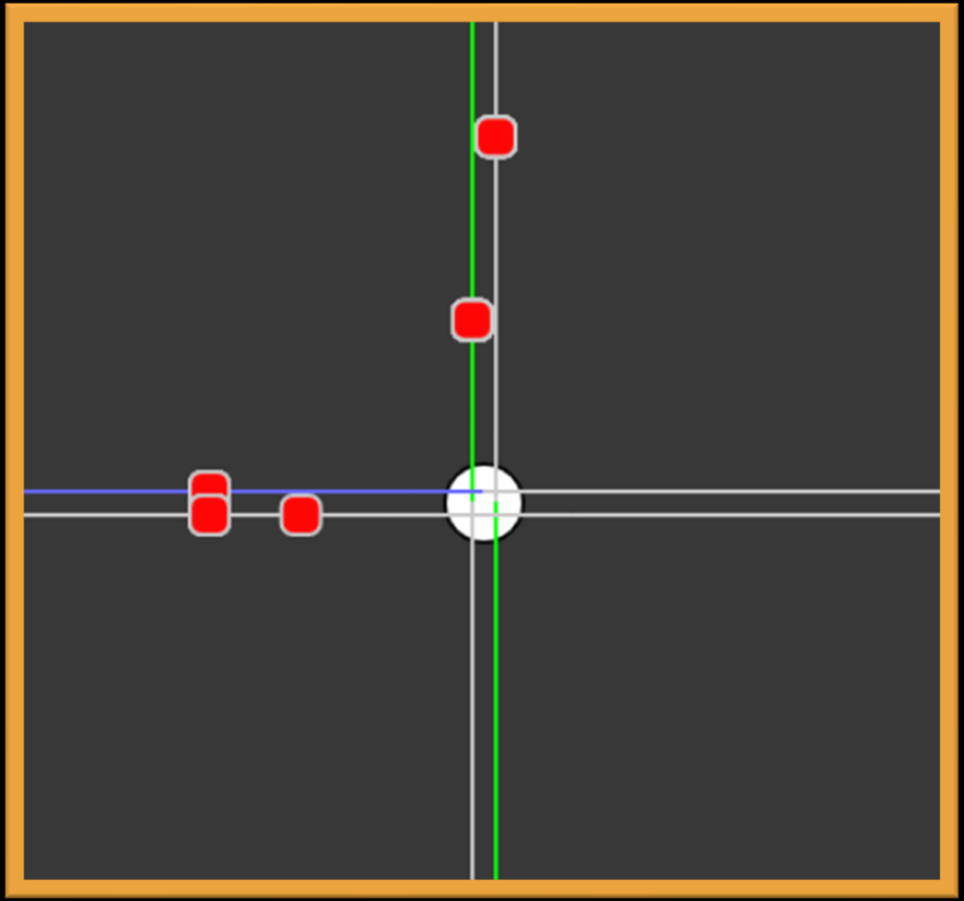
```
41 skipped 3's queue at 253.636363536
moved 8 to lane at 253.909090608
moved 93 to lane at 254
moved 12 to 9's queue at 254.272727072
60 reached destination at 254.545454144000
moved 64 to lane at 254.636363536
5 are waiting at 9
84 left 9's queue at 255/291
start: 222
cycle: 74
Lane 8 to 9
moved 130 to 4's queue at 255.015673880827
69 skipped 0's queue at 255.54545414400002
77 skipped 3's queue at 255.636363536
moved 96 to lane at 255.636363536
40 skipped 3's queue at 256.636363536
moved 113 to lane at 256.90909060800004
moved 2 to lane at 257
moved 102 to 9's queue at 257.272727072
35 reached destination at 257.545454144
moved 0 to lane at 257.636363536
6 are waiting at 9
11 left 9's queue at 258/291
start: 222
cycle: 74
Lane 8 to 9
```



1. **Lights**
   1. Smaller lights indicate 2-way, so no logic
2. **Roads & lanes**
   1. Green active now
   2. White inactive now
   3. Blue always active
3. **Cars**
   1. On lane when visually in middle
   2. In light queue at an end
   3. Blue dot = finished

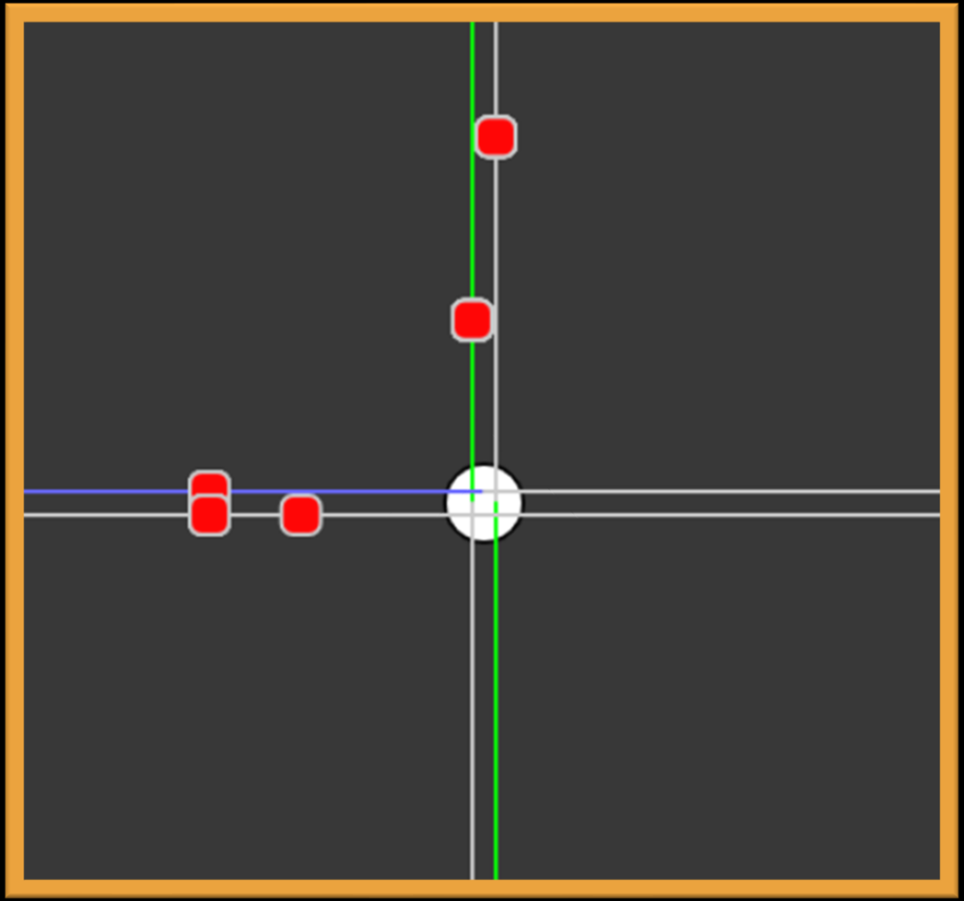
IMPROVEMENTS

LIGHT LOGIC



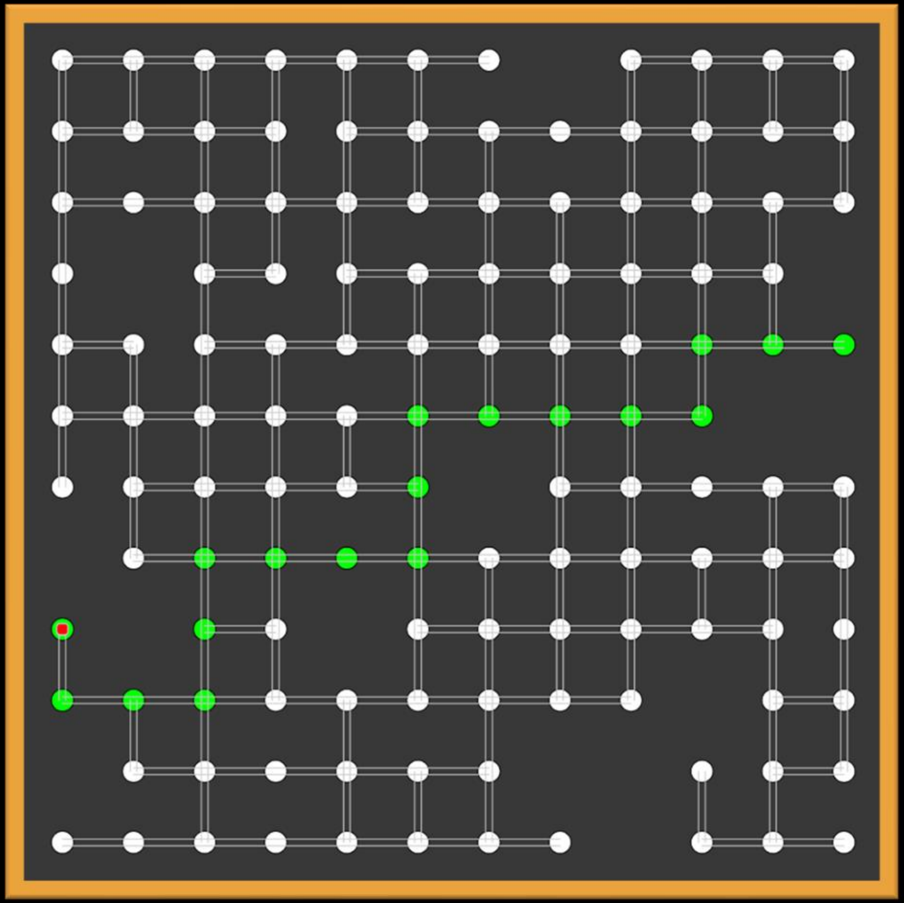
1. Opposite Lanes green at same time
2. Car in north queue will go through almost immediately
3. Car in west queue must wait
4. Eventually light will cycle

SMART-LIGHT LOGIC



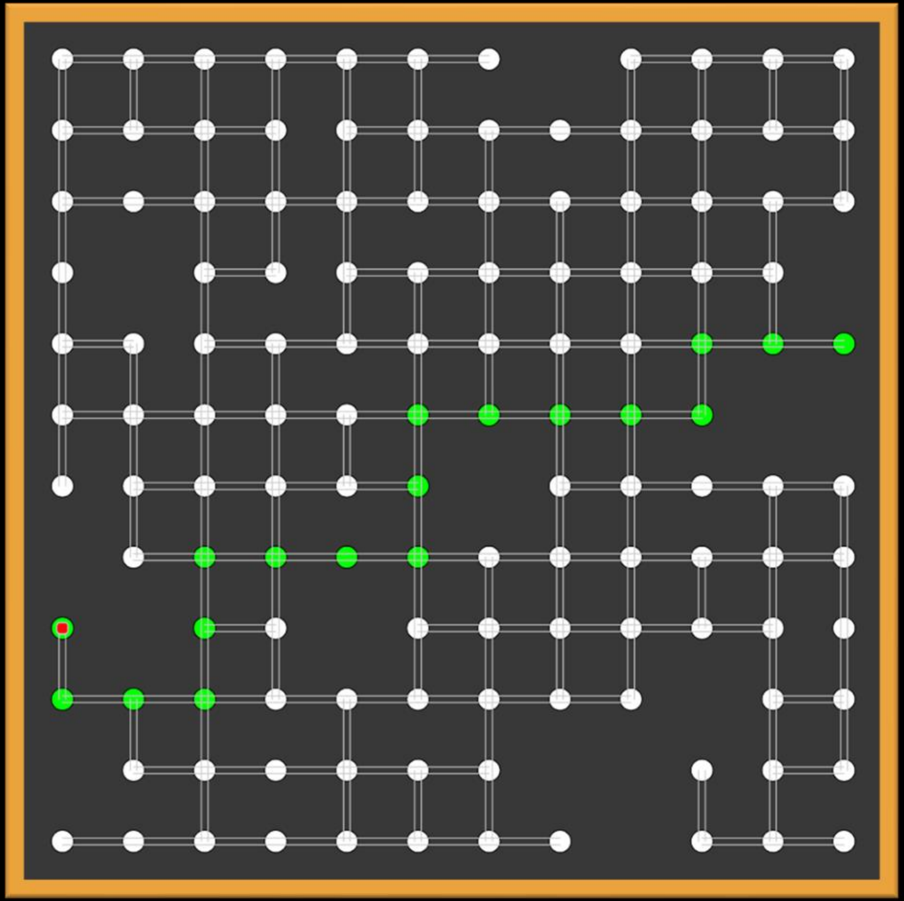
1. If no cars are waiting upon arrival, **immediately begin switching to the lane** of the new car
2. If a car is the last to leave its queue, and there are cars waiting on other lanes, **immediately begin switching away** from current active lanes
3. Imagine as **pressure sensor** or camera equipped lights today

CAR LOGIC



1. Will find shortest path to its exit using the A* algorithm
2. Tries to spawn on a light with only 1 neighbor and end on a light with only 1 neighbor
3. This forces the car to move from low to high congestion during its journey
4. Keeps track of total time and distance

SMART-CAR LOGIC



1. Dynamically **re-runs A*** at every light while **taking into account** that light's **congestion**
2. **A*** **evaluates** a potential path's **cost** based on **how close** it moves towards the destination
    1. To reroute during congestion, **add # of cars** at a light to that cost calculation
3. Weight determined through **experimentation**

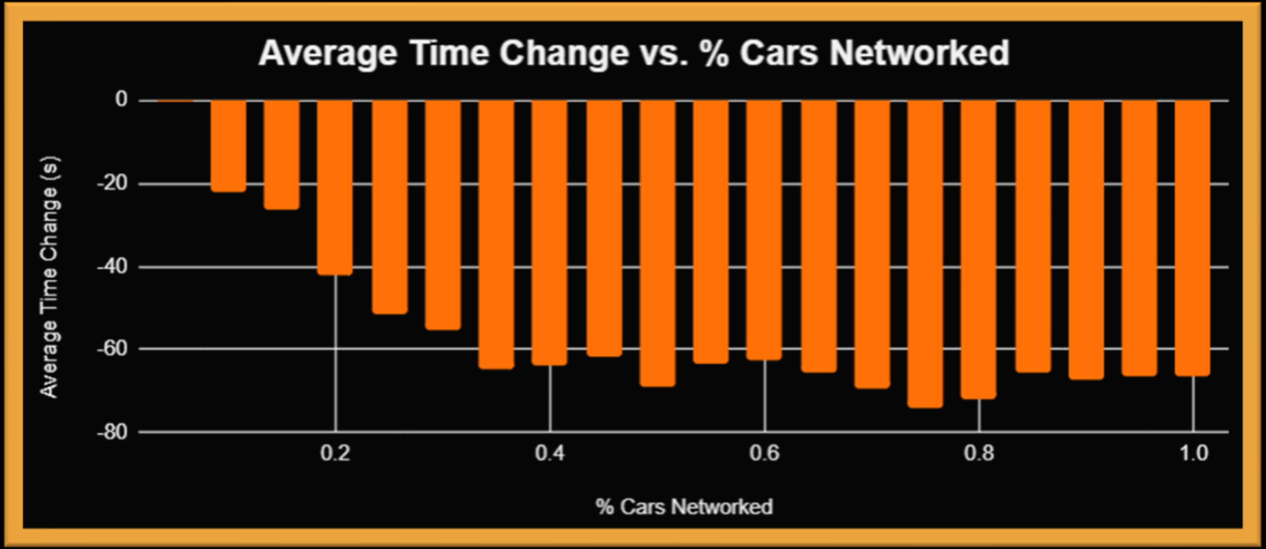
EXPERIMENTATION

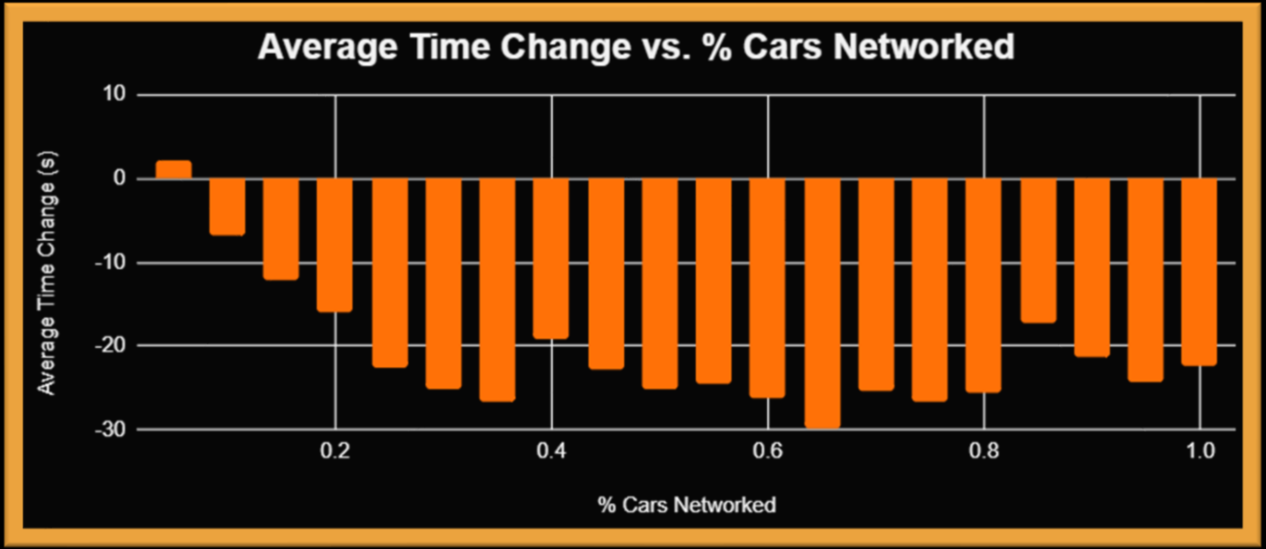
OVERVIEW

1. All tests with the same parameters
    1. Road Length
        1. ~400ft (average city block)
    2. Grid-size
        1. 10
    3. Road-Gen %
        1. 90%
    4. Light-Gen %
        1. 80%
    5. Total Cars
        1. 422
        2. Continuous cars for 10min

Negative numbers in the following slides represent improvement over a baseline run with no smart-light or smart-car functionality

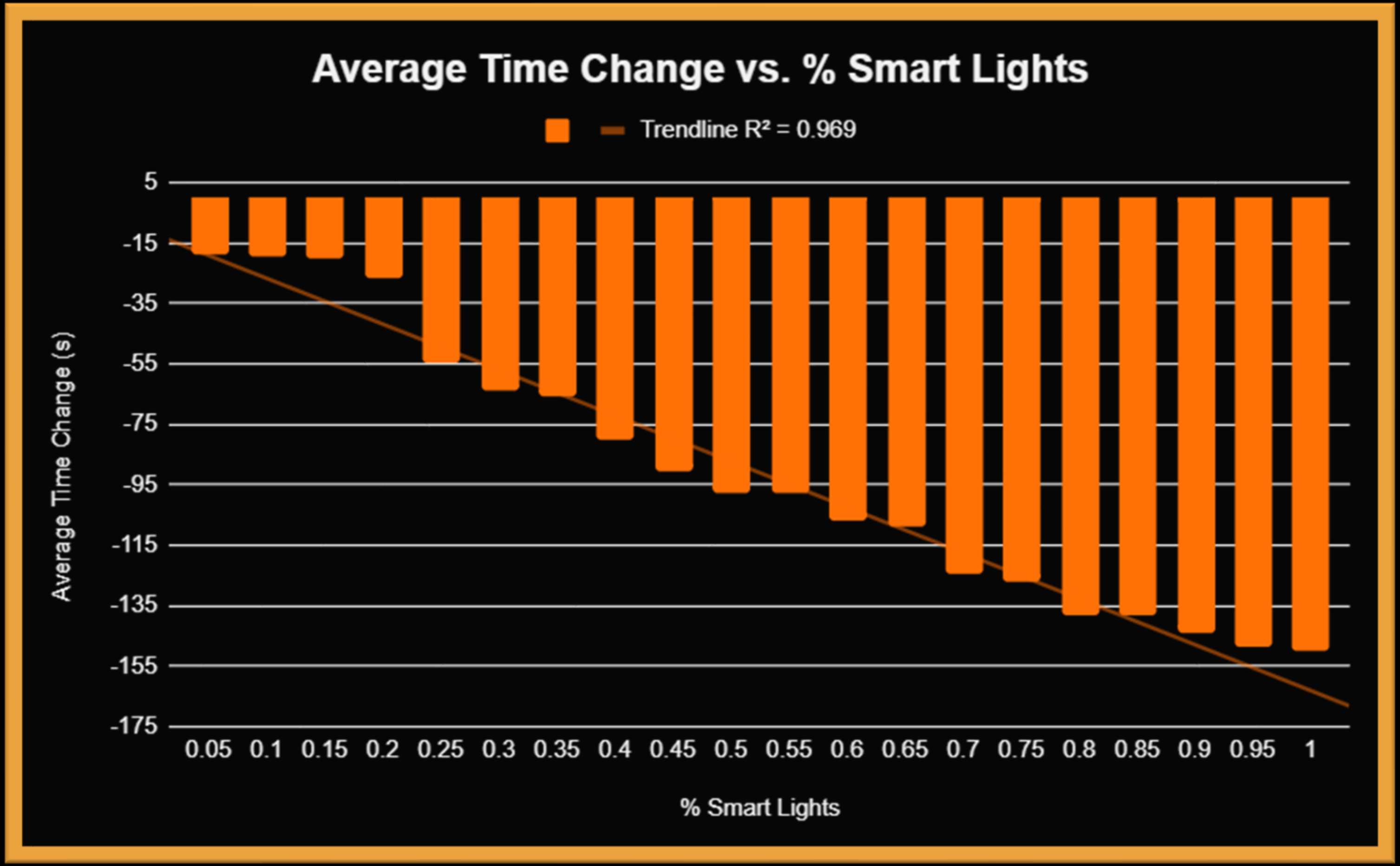
SMART-CAR NETWORKING %





1. **SAME GRID**, % OF CARS NETWORKED
2. NETWORKED CARS CAN **ONLY USE OTHER** NETWORKED CARS IN THEIR RE-ROUTING CALCULATIONS. IMAGINE **GOOGLE MAPS**
3. TWO TRIALS SHOWN, DIFFERENT GRIDS
4. LEVELS OFF AROUND **35%**

SMART-LIGHT %

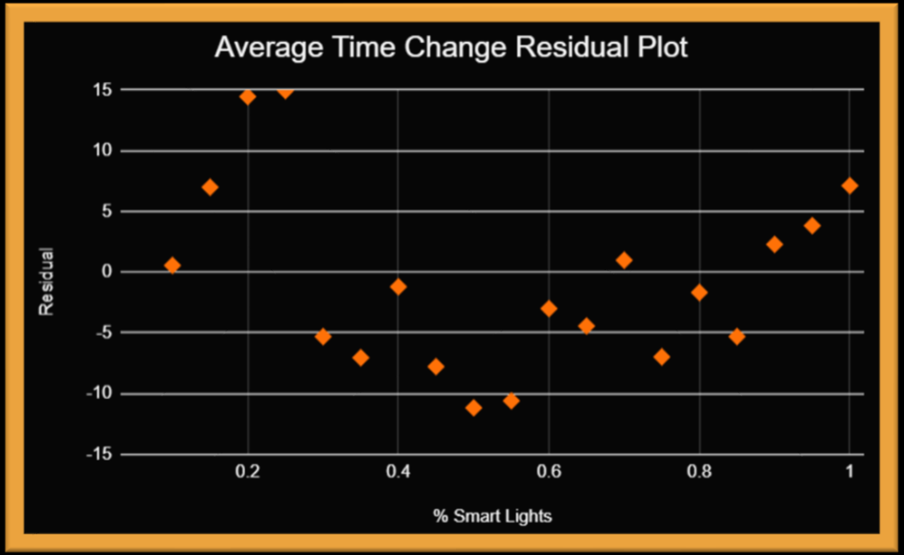


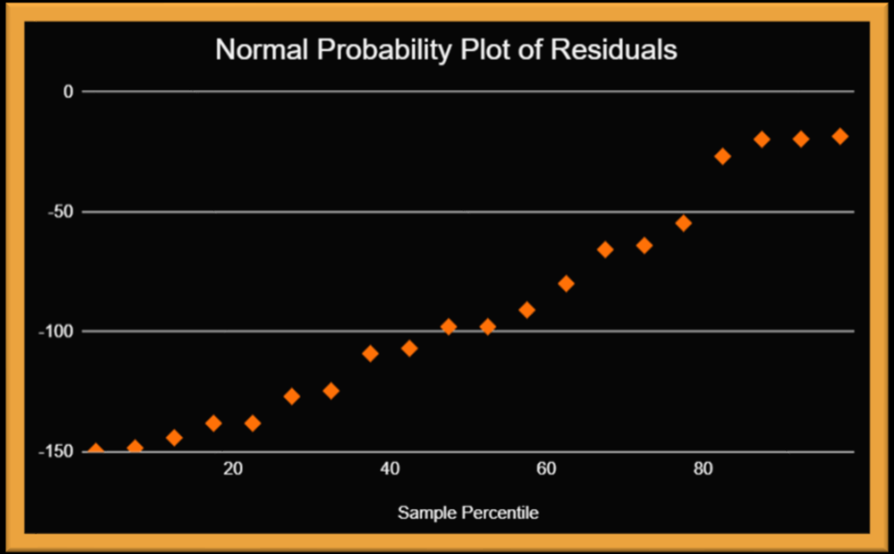
1. Run over **the same grid** while **increasing** the % of **smart lights**
2. Plots **average time difference** of cars between baseline run and smart run

STATISTICAL ANALYSIS

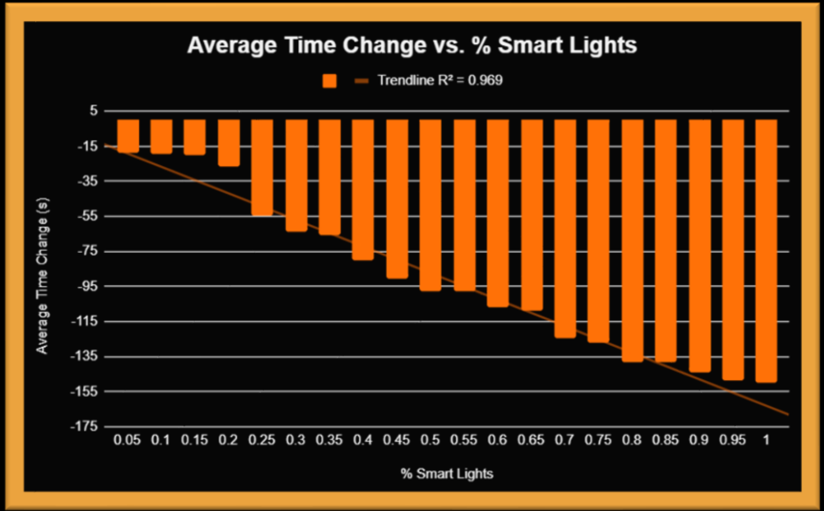
1. Linear Regression T Test Conditions

1. Linear Residual Plot: ✓, no obvious pattern
2. Independent Results: ✓, random simulation
3. Normal: ✓, normal probability plot Linear
4. Equal Variance: ✓, from residual plot
5. Random: ✓, Results are random sample of all results





STATISTICAL ANALYSIS



Results:

Intercept = -11.456s

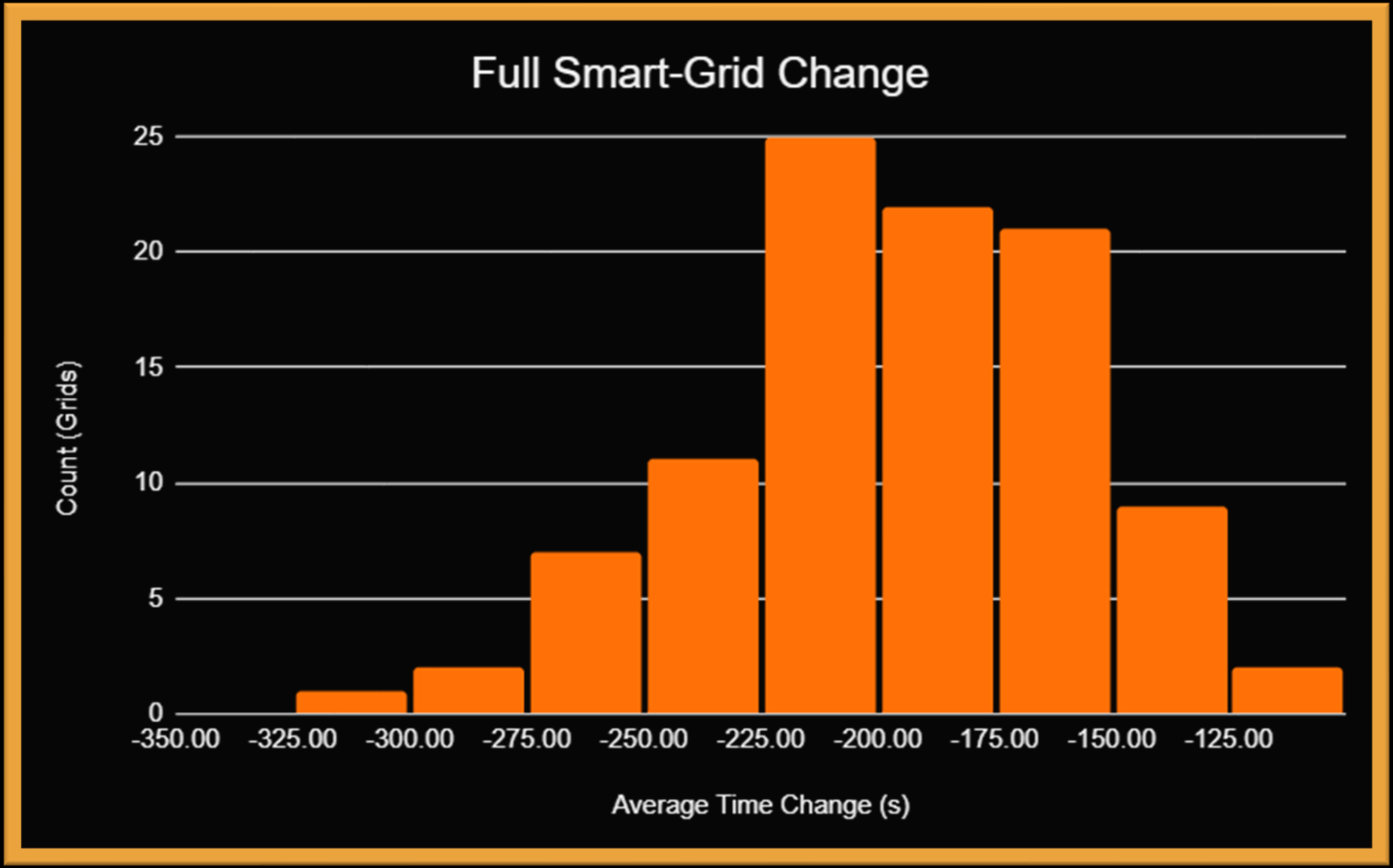
Slope = -157.667

P(Slope) = 4.718×10^{-15}

95% Confidence Interval for Slope: (-165.059, -138.274)

Reject the **second null hypothesis**; smart lights have **statistically significant positive effects** on congestion

ALL OPTIMIZATIONS

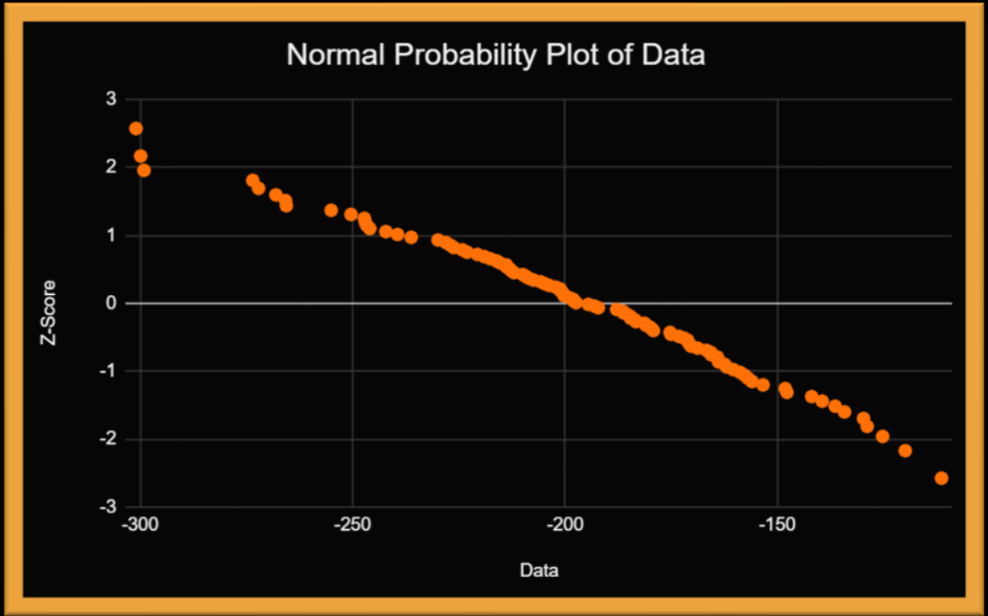


1. Run over **multiple grids** of the **same size**
2. Plot is the **differences between** a baseline run with **no optimizations** and a run with **all optimizations**

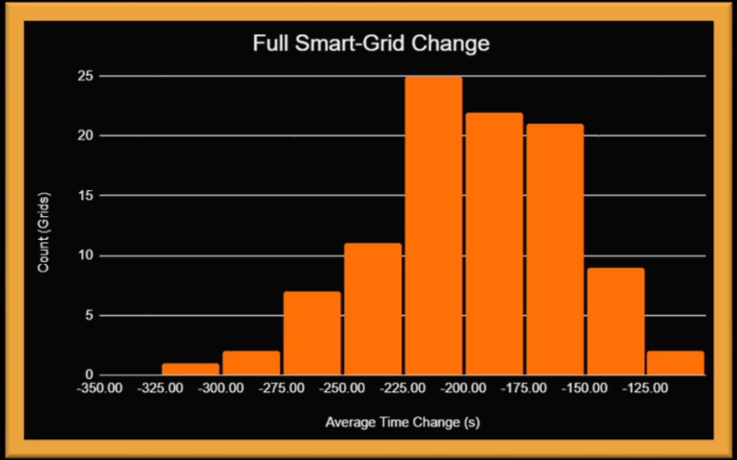
STATISTICAL ANALYSIS

1. One Sample T-Test Conditions:

1. Random: ✓, Results are random sample of all results
2. Normal: ✓, Normal probability plot linear
3. Independent Results: ✓, Random simulation



STATISTICAL ANALYSIS



Results:

Sample Mean = -196.466s

T-Stat = -48.63

P = 0

Reject the **first null hypothesis**; smart cars and smart lights have **statistically significant positive effects** on congestion

Limitations:

1. Lack of **granularity** in the grid
2. Chance of logic errors
3. Very slow with larger grids
4. **Not representative** of the real world

Extensions:

1. Add **more detail** to the program
2. Improve efficiency of code
3. Allow **custom grid** generation (e.g. import Pittsburgh)