Tara Boyle, Shannon Hicks,
Julia Hoffman, Morgan Weis

Drop N' Dine Neighborhood Food Delivery Service

Overall Problem Statement:
A new business Drop N' Dine has opened for business in Blacksburg, VA. The premise of their business strategy is to create a food delivery service that uses customers who already plan on eating out to pick up and bring food home to their neighbors as well. The neighbors having food delivered to their house will pay a fee for that convenient service. 44

Restaurant Recommendations Description:
To start, we assumed that every customer is willing to drive to a restaurant. Drop N' Dine's goal it to inform customers which restaurant to pick up food from to minimize distance driven. The company wants to limit the number of customers going to each restaurant to five.

Model:
I = {1...35} set of houses
J = {1...14} set of restaurants

Variables:
x_{ij} = 1 if house i is assigned to restaurant j, 0 otherwise

Parameters:
d_{ij} = the distance from house i to restaurant j

Minimize
$$\sum_{i=1}^{35} \sum_{j=1}^{14} d_{ij}x_{ij} \text{ (minimize total driving distance from houses to restaurants)}$$

Such that:

$$\sum_{i=1}^{35} x_{ij} \leq 5 \, for \, j \; = \; 1, 2, ..., 14 \text{ (assigns 5 or less houses to each restaurant)}$$

$$\sum_{j=1}^{14} x_{ij} = 1 \, for \;\; i = 1, ..., 35 \text{ (assigns house to a restaurant)}$$

$$x_{ij} \; \varepsilon \; \{0, 1\} \text{ (binary constraint)}$$

Tara Boyle, Shannon Hicks,
Julia Hoffman, Morgan Weis

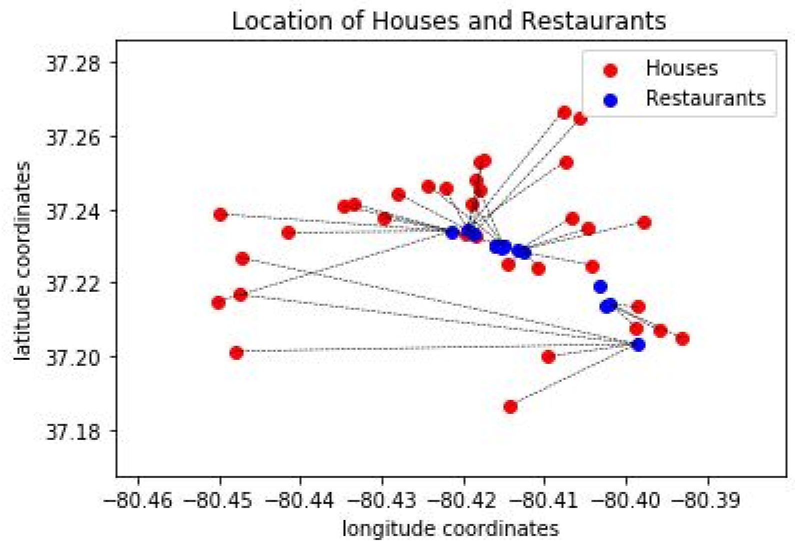
Model Description:
Our model's objective is to minimize distance travelled between houses and restaurants such that each restaurant is not assigned to more than 5 houses, and all houses are assigned to each restaurant.

Restaurant Recommendations Solution:
Objective Value: 9345.7
House1, Buffalo Wild Wings
House2, Bollo's Cafe & Bakery
House3, Boudreaux's Restaurant
House4, Bojangles' Famous Chicken
House5, Buffalo Wild Wings
House6, Boudreaux's Restaurant
House7, Backstreets Restaurant
House8, Blacksburg Taphouse
House9, Bojangles' Famous Chicken
House10, Buffalo Wild Wings
House11, 622 North
House12, 622 North
House13, 622 North
House14, Avellinos Italian & Pizzeria
House15, Bojangles' Famous Chicken
House16, Blacksburg Taphouse
House17, Cabo Fish Taco
House18, Boudreaux's Restaurant
House19, Avellinos Italian & Pizzeria
House20, Boudreaux's Restaurant
House21, Cabo Fish Taco
House22, Blacksburg Taphouse
House23, Avellinos Italian & Pizzeria
House24, Blacksburg Taphouse
House25, Backstreets Restaurant
House26, 622 North
House27, Buffalo Wild Wings
House28, Bojangles' Famous Chicken
House29, Bojangles' Famous Chicken
House30, Blacksburg Taphouse
House31,Cabo Fish Taco

Tara Boyle, Shannon Hicks,
Julia Hoffman, Morgan Weis

House32,622 North
House33, Avellinos Italian & Pizzeria
House34, Buffalo Wild Wings
House35, Backstreets Restaurant

Solution Plot:



Restaurant Recommendations Improvements:
Time to run code: 0.03089110000018991 seconds

We would not focus on shortening the time it takes the model to run in Spyder because it runs in .03 seconds. Although, we would adjust the model as described below.

Currently, one restaurant is not assigned to any houses. To fix this, we would add a constraint that every restaurant must be assigned at least one house. This would add distance to the optimal solution, but is more realistic. If we had more data, we would additionally consider people's restaurant preference, as opposed to solely the minimum distance of restaurants to houses.

Is this Polynomial Time?:
This problem can be checked in polynomial time because this is an integer program, and integer programs lie in NP-complete. However, NP-complete programs are not solvable in polynomial time. Hypothetically, we could use a different method such as matching that solves the problem in polynomial time.

Tara Boyle, Shannon Hicks,
Julia Hoffman, Morgan Weis

Neighbor Pick-Up Service Description:
The following will be discussing the possibility of neighbors picking up food for each other. This scenario will minimize the distance of the network model because only 10 drivers will be making trips to restaurants. Some constraints to consider are that each driver can pick up for at most 2 other houses, suggesting that of the 35 houses, 5 houses will not be serviced. This is taken into account by telling the 5 furthest houses to their preferred restaurant is out of bounds. A possible solution for these 5 excluded houses would be to ask them if they want food from a restaurant that a nearby driver is going to.

Problem Complexity:
This problem lies in NP-Hard because there are 10 traveling salesman problems within this scenario. This means that the problem is not solvable in polynomial time. Because of this, an exact algorithm would take too long to implement. Instead, a heuristic will be used to find a feasible solution. The specific heuristic used was Simulated Annealing because of its ability to find global mins.

Initial Neighbor Pick-Up Service Thoughts:
Objective: Minimize total distance travelled

Constraints:
Each house only delivers to two other houses
10 delivery houses
25 ordering houses

Problem:
10 drivers are only willing to deliver food to at most 2 other houses, serving 20 houses in total. This leaves 5 houses without food, because 25 houses want to order food.

Our thoughts:
1. Exclude top 5 houses that have greatest distance from their home to restaurant preference
2. Categorize customers by restaurant preference
3. Pick closest house to restaurant to be designated driver

Final Neighbor Pick-Up Service Model:
Set covering:
We initially eliminated 5 houses that have the furthest distance to possible drivers and restaurants. Then we determined the set of houses for each driver of the remaining houses using the model below.

Tara Boyle, Shannon Hicks,
Julia Hoffman, Morgan Weis

Model:
I = {1...35} set of houses
J = {1...14} set of restaurants
K = {1...10} set of drivers

Variables:
x_{ik} = 1 if house i is assigned to driver k, 0 otherwise
x_{kj} = 1 if driver k is assigned to restaurant j, 0 otherwise

Parameters:
d_{ik} = the distance from house i to driver k
d_{kj} = the distance from house i to restaurant j

Minimize
$\sum_{k=1}^{10}\sum_{j=1}^{14} d_{kj}x_{kj} + \sum_{i=1}^{35}\sum_{k=1}^{10} d_{ik}x_{ik}$ (minimize total driving distance from drivers to restaurants to houses)

Such that:

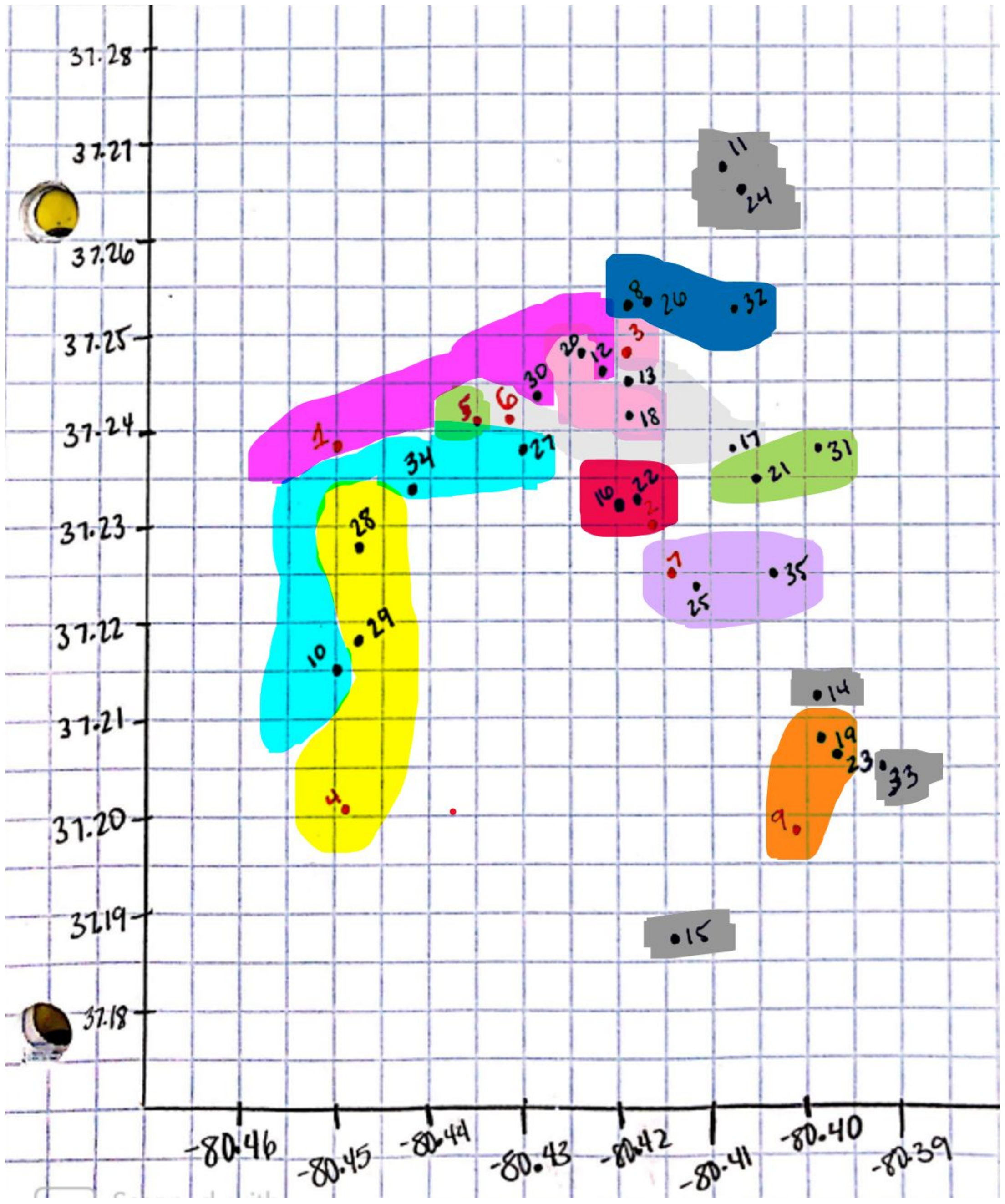
$\sum_{i=1}^{35} x_{ik} \leq 2\, for\ k = 1, 2, ..., 10$ (assigns 2 or less houses to each driver)

$\sum_{j=1}^{14} x_{kj} = 1\, for\ k = 1, ..., 10$ (assigns house to a driver)

$x_{ij}\ \varepsilon\ \{0, 1\}$ (binary constraint)
$x_{kj}\ \varepsilon\ \{0, 1\}$ (binary constraint)

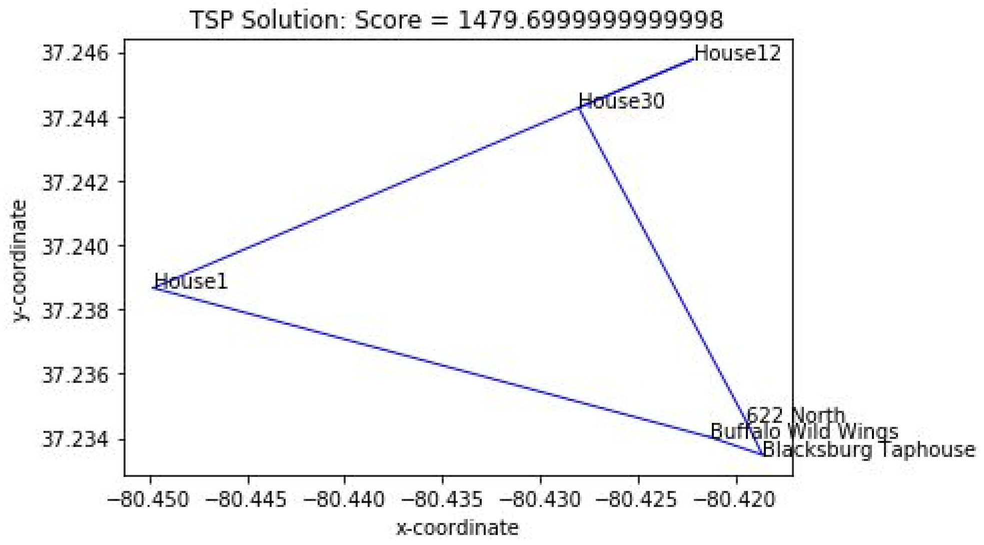
Tara Boyle, Shannon Hicks,
Julia Hoffman, Morgan Weis

Visual of Set of Covering Assignments:
Below is the assignments of houses to drivers. The 5 houses highlighted in dark grey will not be included because their coordinates are out of bounds.



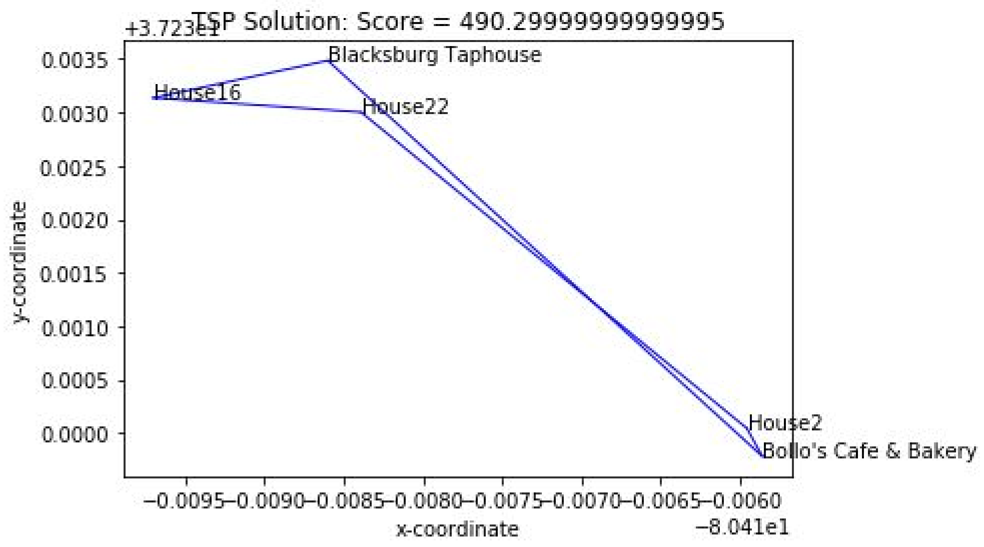
Initial Solution to Individual TSPs:
The next part of the solution figures out the time it takes to complete a delivery route for each driver. This is an initial feasible solution and is not optimal.

Driver 1(Buffalo Wild Wings) is taking 12(622 North) and 30(Blacksburg Taphouse)



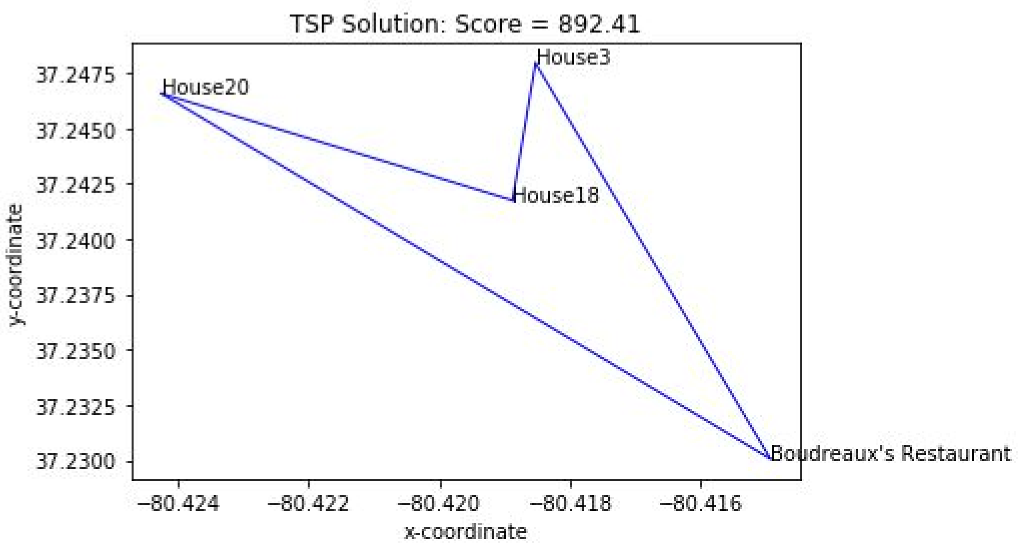
['House1', 'Buffalo Wild Wings', 'Blacksburg Taphouse', '622 North', 'House30', 'House12']

Driver 2 (Bollos) is taking 16(Taphouse) and 22(Taphouse)



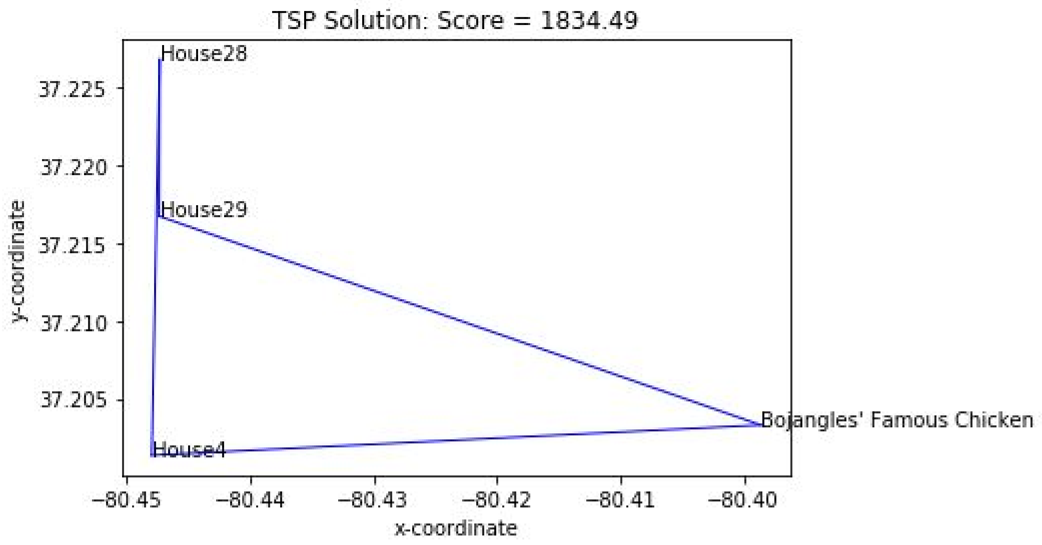
['House2', "Bollo's Cafe & Bakery", 'Blacksburg Taphouse','House16', 'House22']

Driver 3(Boudreaux's Restaurant) is taking 18(Boudreaux's Restaurant) and 20(Boudreaux's Restaurant)



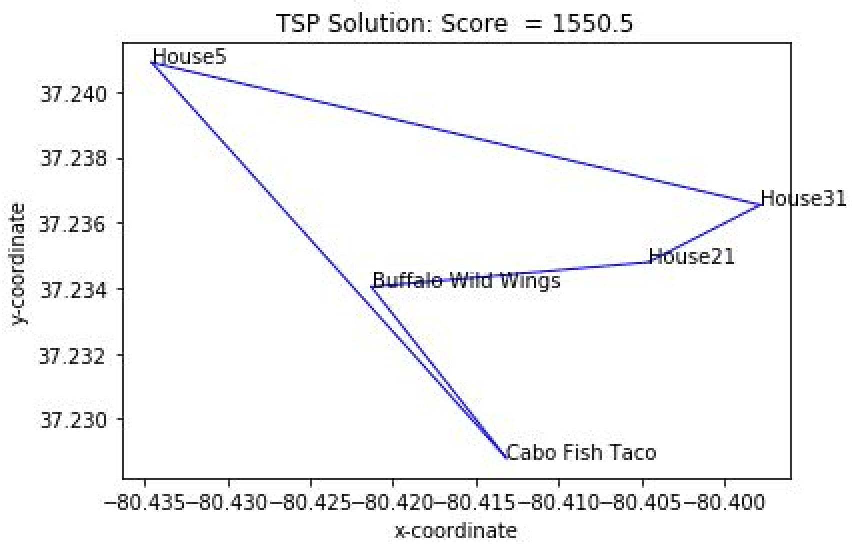
['House3', "Boudreaux's Restaurant", 'House20', 'House18']

Driver 4(Bojangles' Famous Chicken) is taking 28(Bojangles' Famous Chicken) and 29(Bojangles' Famous Chicken)



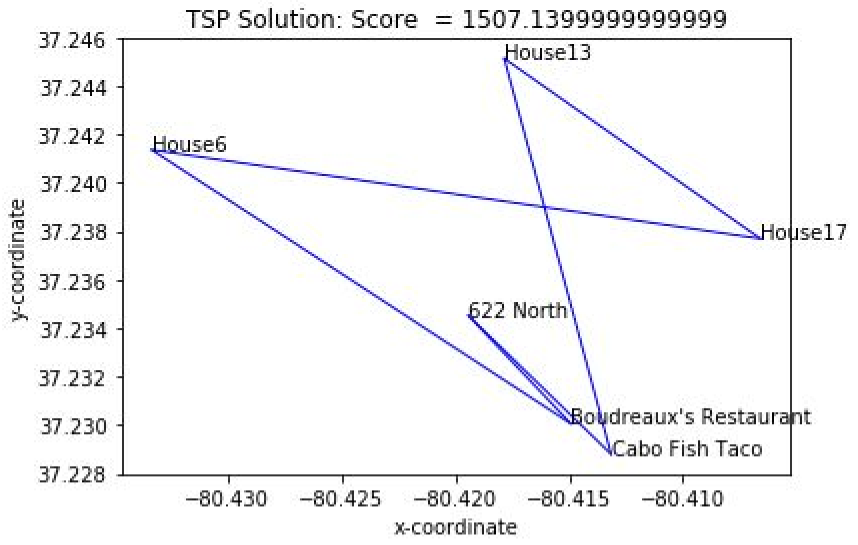
['House4', "Bojangles' Famous Chicken", 'House28', 'House29']

Driver 5(Buffalo Wild Wings) is taking 21(Cabo Fish Taco) and 31(Cabo Fish Taco)



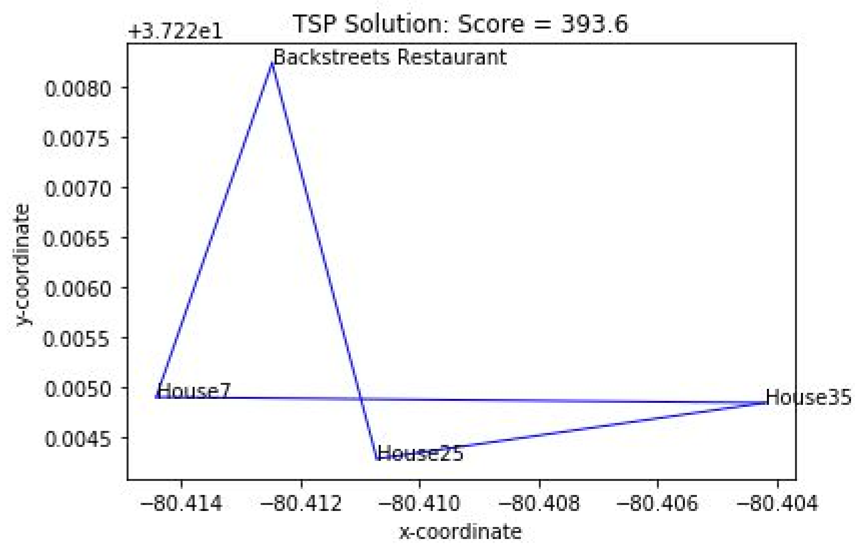
['House5', 'Cabo Fish Taco','Buffalo Wild Wings', 'House21', 'House31']

Driver 6(Bourdeauxs) is taking 13(622) and 17(Cabos)



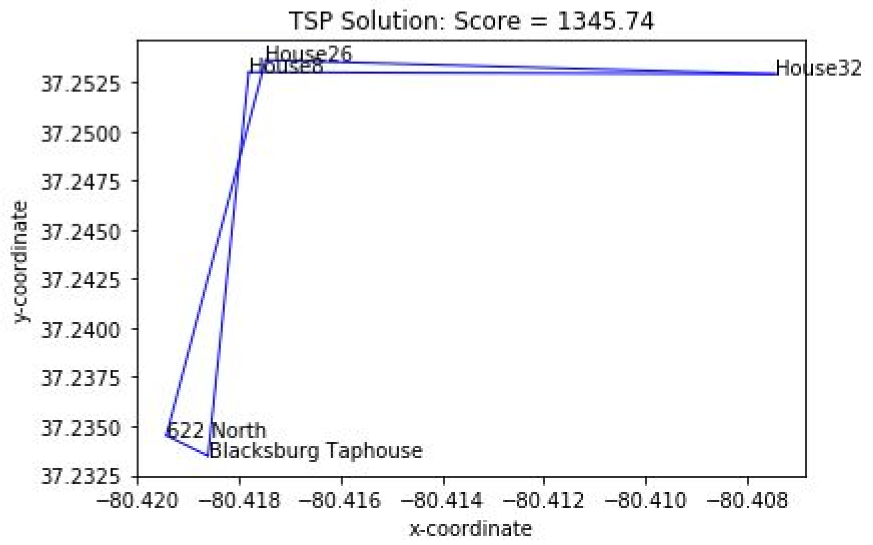
['House6', "Boudreaux's Restaurant",'622 North', 'Cabo Fish Taco','House13', 'House17']

Driver 7(Backstreets Restaurant) is taking 25(Backstreets Restaurant) and 35(Backstreets Restaurant)



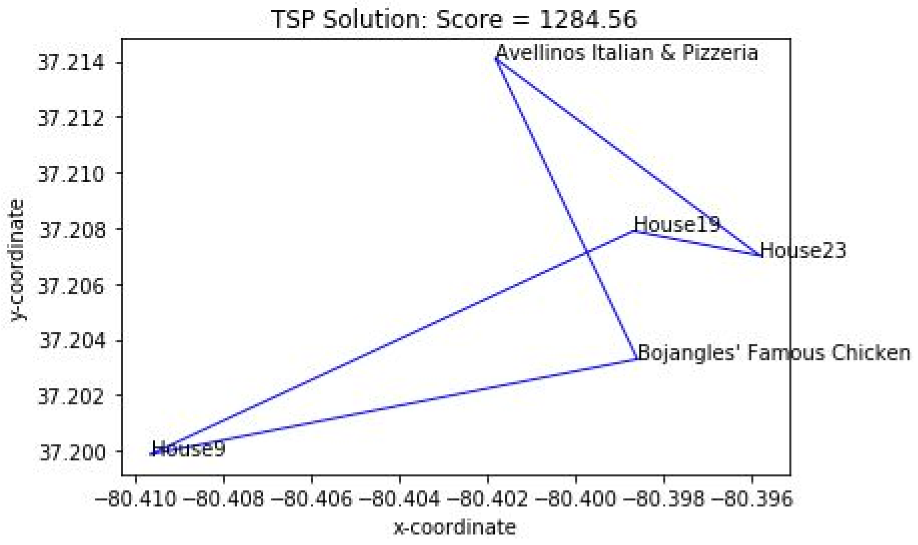
['House7', "Backstreets Restaurant", 'House25', 'House35']

Driver 8(Boudreaux's Restaurant) is taking 26(622 North) and 32(622 North)



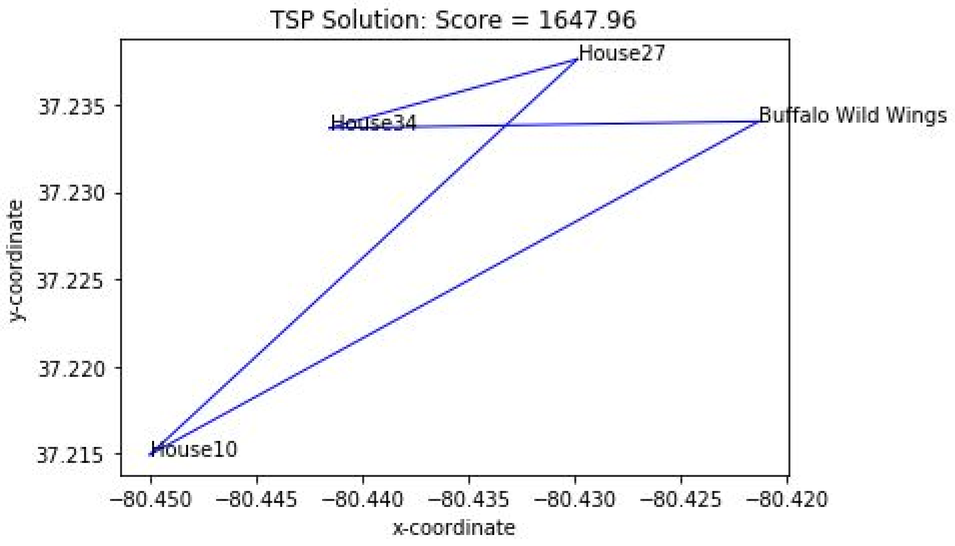
['House8', "Blacksburg Taphouse", "622 North", 'House26', 'House32']

Driver 9 (Bojangles' Famous Chicken) is taking 19 (Avellinos Italian & Pizzeria) and 23 (Avellinos Italian & Pizzeria)



['House9', "Bojangles' Famous Chicken", 'Avellinos Italian & Pizzeria', 'House23', 'House19']

Driver 10 (Buffalo Wild Wings) is taking 34 (Buffalo Wild Wings) and 27 (Buffalo Wild Wings)



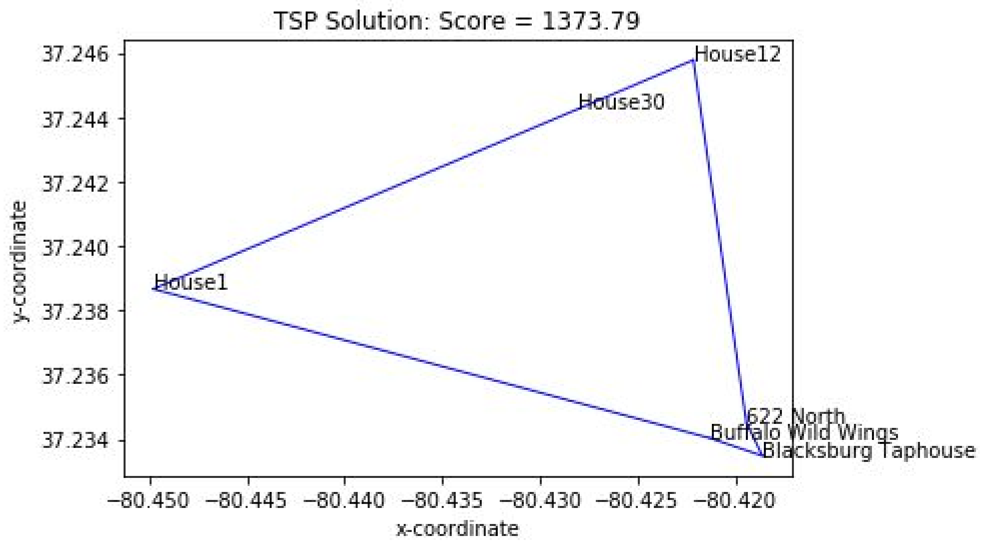
['House10', "Buffalo Wild Wings",'House34','House27']

Tara Boyle, Shannon Hicks,
Julia Hoffman, Morgan Weis

Company Metrics (from initial solution):
Total overall run time: 207.1 minutes (3.45 hours)
Average individual run time: 20.71 minutes (.345 hour)

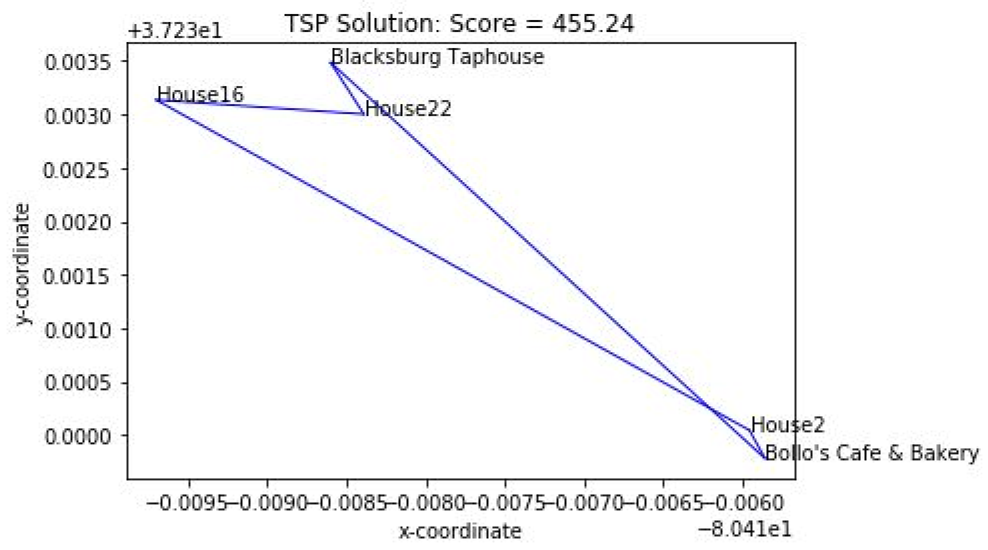
Simulated Annealing Solution:
To find the optimal route that each driver should take to pick up food, simulated annealing was performed. The improved solution is below. Note that there are constraints on the subtours because the restaurant for each house must be visited before the house is visited.

Driver 1 (Buffalo Wild Wings) is taking 30(Blacksburg Taphouse) and 12(622 North)



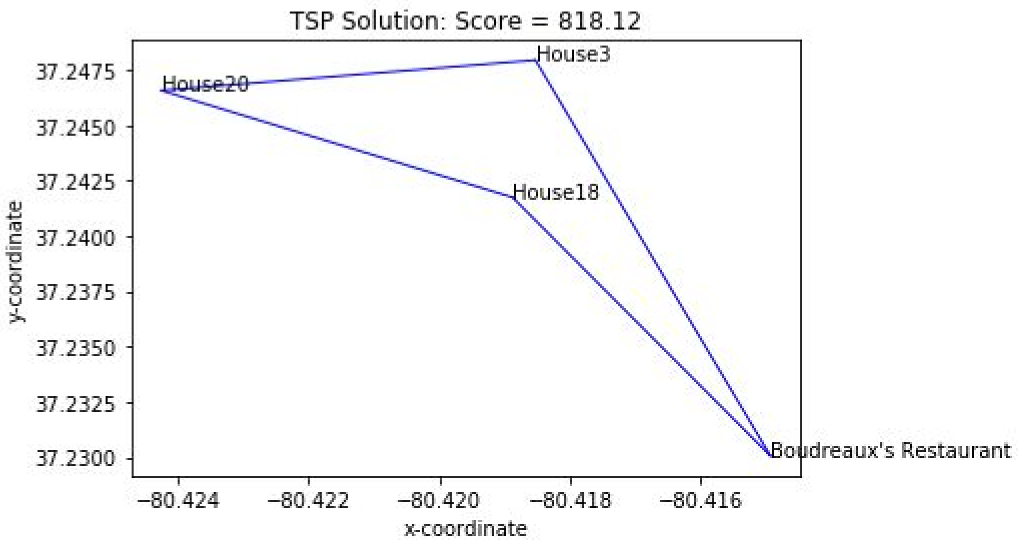
['House1', 'Buffalo Wild Wings', 'Blacksburg Taphouse', '622 North', 'House12', 'House30']

Driver 2 (Bollos) is taking 16(Blacksburg Taphouse) and 22(Blacksburg Taphouse)



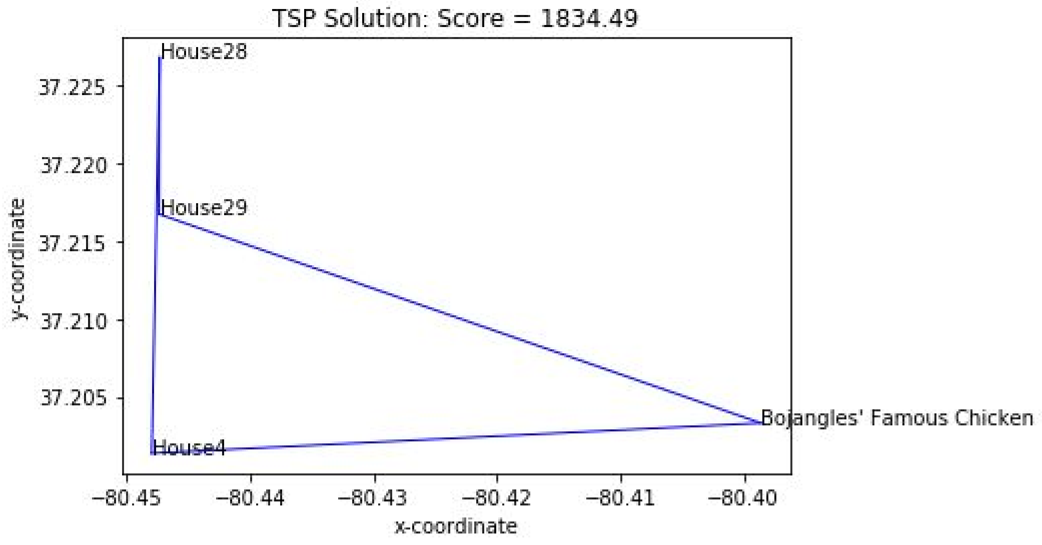
['House2', "Bollo's Cafe & Bakery", 'Blacksburg Taphouse', 'House22', 'House16']

Driver 3(Boudreaux's Restaurant) is taking 18(Boudreaux's Restaurant) and 20(Boudreaux's Restaurant)



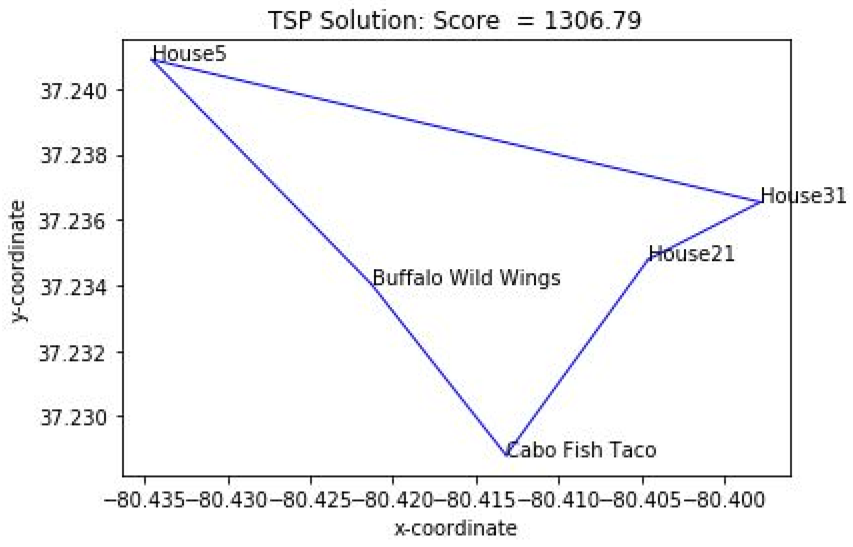
['House3', "Boudreaux's Restaurant", 'House18', 'House20']

Driver 4(Bojangles' Famous Chicken) is taking 28(Bojangles' Famous Chicken) and 29(Bojangles' Famous Chicken)



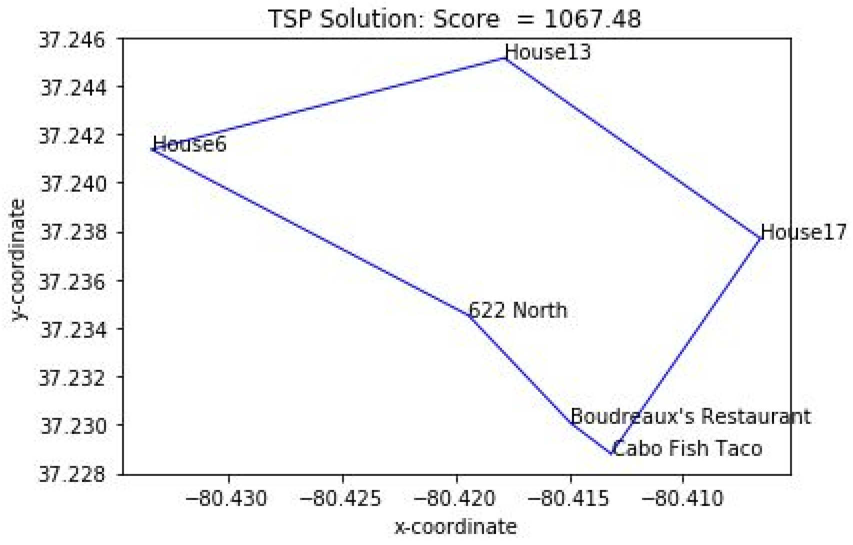
[‘House4’, "Bojangles’ Famous Chicken", 'House29', 'House28']

Driver 5(Buffalo Wild Wings) is taking 21(Cabo Fish Taco) and 31(Cabo Fish Taco)



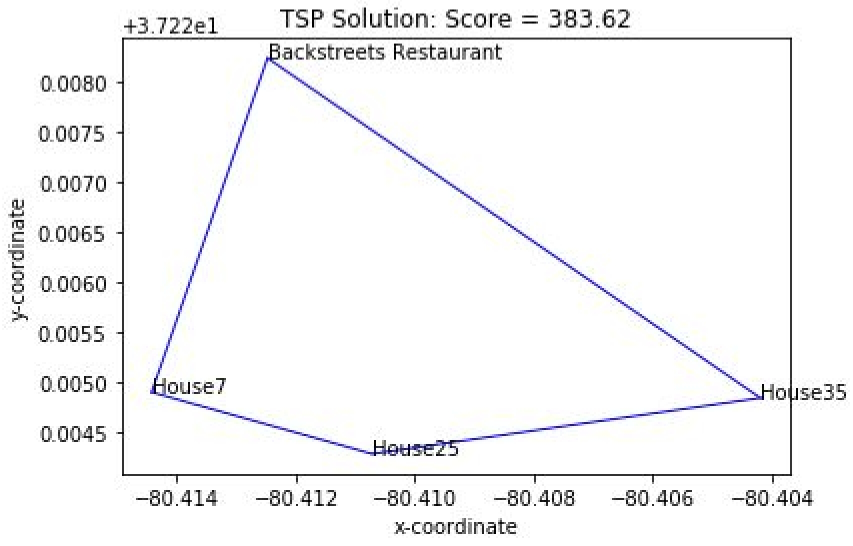
['House5', 'Buffalo Wild Wings', 'Cabo Fish Taco', 'House21', 'House31"]

Driver 6(Bourdeauxs) is taking 13(622) and 17(Cabos)



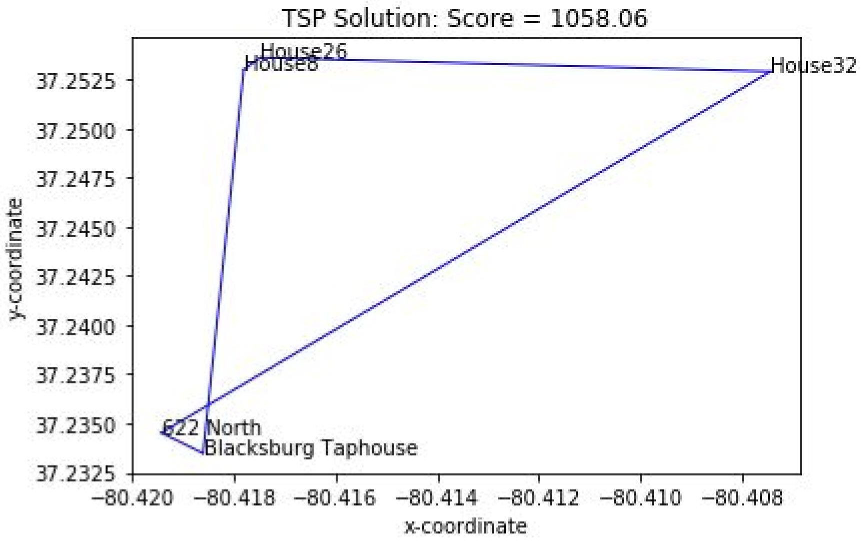
['House6', '622 North',"Boudreaux's Restaurant", 'Cabo Fish Taco','House17', 'House13']

Driver 7(Backstreets Restaurant) is taking 25(Backstreets Restaurant) and 35(Backstreets Restaurant)



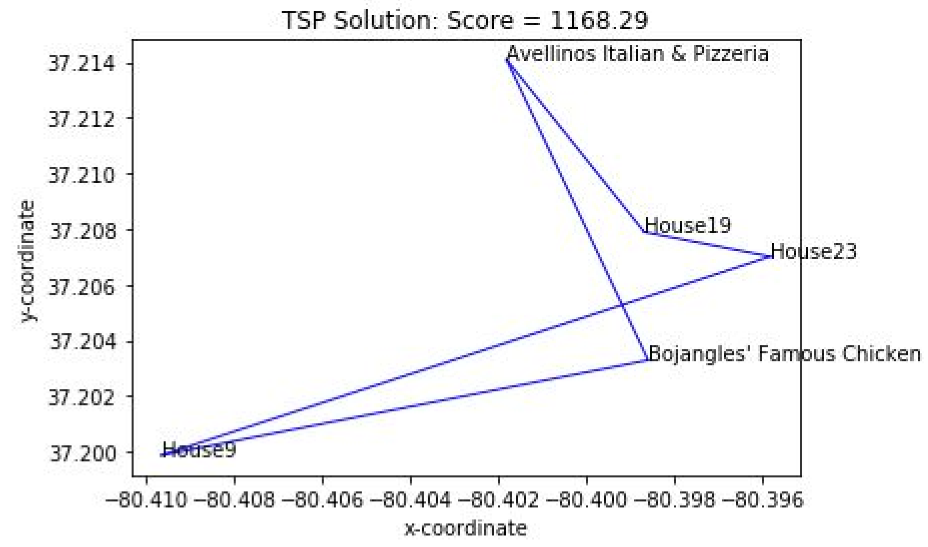
['House7', "Backstreets Restaurant", 'House35', 'House25']

Driver 8(Boudreaux's Restaurant) is taking 26(622 North) and 32(622 North)



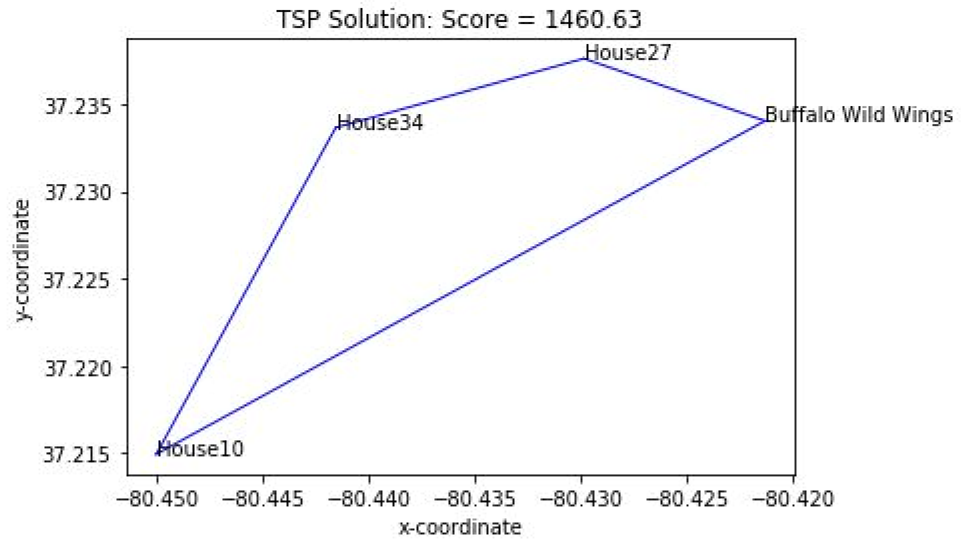
['House8', "Blacksburg Taphouse", "622 North", 'House32', 'House26']

Driver 9 (Bojangles' Famous Chicken) is taking 19 (Avellinos Italian & Pizzeria) and 23 (Avellinos Italian & Pizzeria)



['House9', "Bojangles' Famous Chicken", 'Avellinos Italian & Pizzeria', 'House19', 'House23']

Driver 10 (Buffalo Wild Wings) is taking 34 (Buffalo Wild Wings) and 27 (Buffalo Wild Wings)



['House10', 'Buffalo Wild Wings', 'House27', 'House34']

Company Metrics (from optimal solution):
Total overall run time: 182.1 minutes (3.04 hours)
Average individual run time: 18.21 minutes (.304 hour)

Neighbor Pick-Up Service Solution Conclusions:
After performing simulated annealing for all 10 routes, our solution improves by appromiately 14%, decreasing average run time from 20.7 minutes to 18.2 minutes. This specific TSP algorithm was used because of its ability to find global minimums. Hill-climbing algorithms such as 2-opt swap are not the algorithms of choice for this project because solutions can easily get stuck in local minimums. The optimal average time for a driver to pick up and deliver food to their neighbors is 18.21 minutes. We also calculated an overall time to finish routes for a company standpoint of 3.04 hours. This metric helps the company calculate overall costs, and inform customers about the approximate time to get their order delivered.

Neighbor Pick-Up Service Solution Improvement:
Although we were able to improve our solution, this may not be optimal because of what we assumed in the initial set covering part of the problem. If our sets and assumptions are correct, then we found the optimal routes for each driver. Our solution can be improved by finding a

more optimal way to assign houses to drivers initially. Additionally, this model uses euclidean distances to minimize distance, with no regard to roads. Factoring in routes that can be feasibly taken from houses to restaurants using roads would increase the distance and time needed to deliver food, as roads are usually not the most direct route to your destination. Finally, this model excludes five houses that are farthest from the designated drivers and their restaurant preference. To improve this, we would allow the farthest houses to change their desired preference to make the trip easier on the driver, or increase the price of the driving rate for those drivers willing to deliver to more than two houses.

Coupon Advantage Description:
Drop N' Dine wants to incentivize customers to order the night before they plan to eat dinner using this service with coupons. This is advantageous for the company because then they can optimize the model more easily if they have the information ahead of time. Drop N' Dine will use customers average purchase information and the number of times customers have used this service to determine which customers will receive coupons.

Model:

$$\min \sum w_i^2 + C \sum_{i=1}^{N} \xi_i$$

$$\text{such that } y_i(x \bullet w + b) \geq 1 \ \forall \ i = 1, \ldots$$

$$w \varepsilon \Re$$

Classifier:
Given the past data about average purchase, number of times service used, maximum distance to restaurants, minimum distance to restaurants, and if a coupon was given to, we made a model to predict if a coupon should be given or not. The following values were calculated from past data using the SVM module in python and the model above.

b= -22.992306544791123
w = [0.32237259 0.40916227 0.00656591 0.04280214]
y[0] = 1
learned function of X[0] = 22.66084220509597

Testing Past Data to Check Model:
Before predicting on a new data set, we tested our model to see if it predicts the coupon or no coupon decision correctly. The model predicted values for the past data set and compared this to the actual values. All of the predicted values were the same as the actual values, suggesting that this is an accurate model.

```
T/F y[i] f(X[i]))
True   1   22.66084220509597
Number true: 100
Number false: 0
True   1   6.0019742478584845
Number true: 100
Number false: 0
True   1   4.4650712185070525
Number true: 100
Number false: 0
True   1   16.896046431351483
Number true: 100
Number false: 0
True   1   24.071582374650024
Number true: 100
Number false: 0
True   1   7.02163765565054
Number true: 100
Number false: 0
True   -1   -8.813205669315092
Number true: 100
Number false: 0
True   1   1.8830077837381403
Number true: 100
Number false: 0
True   1   4.861279405169668
Number true: 100
Number false: 0
True   1   0.9996070671157362
Number true: 100
Number false: 0
True   -1   -5.3391632535671185
Number true: 100
Number false: 0
True   1   6.227171617916021
Number true: 100
Number false: 0
True   1   4.499172054502946
```

```
Number true: 100
Number false: 0
True   1   2.301747293364553
Number true: 100
Number false: 0
True   1   7.542549814076008
Number true: 100
Number false: 0
True   1   1.4425446371665451
Number true: 100
Number false: 0
True   1   3.494286972673766
Number true: 100
Number false: 0
True   1   8.998971095868821
Number true: 100
Number false: 0
True   1   12.577719685978348
Number true: 100
Number false: 0
True   1   23.622301385617316
Number true: 100
Number false: 0
True   1   2.2327429869179163
Number true: 100
Number false: 0
True   1   17.684814837526073
Number true: 100
Number false: 0
True   -1   -6.861166907747734
Number true: 100
Number false: 0
True   -1   -4.377380435165108
Number true: 100
Number false: 0
True   1   6.375649708343758
Number true: 100
Number false: 0
True   -1   -2.718504824193065
```

```
Number true: 100
Number false: 0
True   1   6.152301018222985
Number true: 100
Number false: 0
True   1   8.060650575321773
Number true: 100
Number false: 0
True   1   21.788308640518334
Number true: 100
Number false: 0
True   1   13.069228523435807
Number true: 100
Number false: 0
True   1   16.6419168956488
Number true: 100
Number false: 0
True   1   9.551324012745088
Number true: 100
Number false: 0
True   1   9.91820547914823
Number true: 100
Number false: 0
True   1   31.896280212885056
Number true: 100
Number false: 0
True   -1   -2.8579099653979796
Number true: 100
Number false: 0
True   1   11.407884500747503
Number true: 100
Number false: 0
True   1   5.6213229253174255
Number true: 100
Number false: 0
True   -1   -4.215704638308566
Number true: 100
Number false: 0
True   1   8.584873804113844
```

```
Number true: 100
Number false: 0
True   1   16.600359825170624
Number true: 100
Number false: 0
True   1   12.872453204618093
Number true: 100
Number false: 0
True   1   4.335616688226388
Number true: 100
Number false: 0
True   -1   -0.9996473656455969
Number true: 100
Number false: 0
True   1   3.390484672231107
Number true: 100
Number false: 0
True   1   3.717834922381673
Number true: 100
Number false: 0
True   1   20.356071475218894
Number true: 100
Number false: 0
True   -1   -4.213298250638221
Number true: 100
Number false: 0
True   1   10.582741324585545
Number true: 100
Number false: 0
True   1   5.749778145107374
Number true: 100
Number false: 0
True   1   13.470104400118334
Number true: 100
Number false: 0
True   -1   -1.401052699164655
Number true: 100
Number false: 0
True   -1   -1.0002693261491267
```

```
Number true: 100
Number false: 0
True   1   19.592766934782627
Number true: 100
Number false: 0
True   1   7.142231026072086
Number true: 100
Number false: 0
True   1   13.24293659126484
Number true: 100
Number false: 0
True   -1   -1.8180581564355087
Number true: 100
Number false: 0
True   1   9.563771500947507
Number true: 100
Number false: 0
True   1   21.869013878969437
Number true: 100
Number false: 0
True   -1   -4.472369430251678
Number true: 100
Number false: 0
True   -1   -1.0002693261475244
Number true: 100
Number false: 0
True   1   5.044941121425946
Number true: 100
Number false: 0
True   1   9.80692158760878
Number true: 100
Number false: 0
True   1   7.638110730528602
Number true: 100
Number false: 0
True   1   1.4576143797640846
Number true: 100
Number false: 0
True   -1   -2.4217637094895537
```

```
Number true: 100
Number false: 0
True   1   11.95625214930504
Number true: 100
Number false: 0
True   1   4.726439949837609
Number true: 100
Number false: 0
True   1   19.794735625625616
Number true: 100
Number false: 0
True   1   3.138215597650923
Number true: 100
Number false: 0
True   1   31.18452187571025
Number true: 100
Number false: 0
True   1   11.763307859944842
Number true: 100
Number false: 0
True   1   16.200285493520386
Number true: 100
Number false: 0
True   1   6.810620689901221
Number true: 100
Number false: 0
True   1   10.498774823824636
Number true: 100
Number false: 0
True   1   17.391983364311752
Number true: 100
Number false: 0
True   -1   -8.166673245497998
Number true: 100
Number false: 0
True   1   7.528552238259518
Number true: 100
Number false: 0
True   1   4.803325775234171
```

```
Number true: 100
Number false: 0
True   1   23.39197854738877
Number true: 100
Number false: 0
True   1   30.641157410744714
Number true: 100
Number false: 0
True   1   13.181907253960912
Number true: 100
Number false: 0
True   1   11.988644022596116
Number true: 100
Number false: 0
True   1   11.402345108490206
Number true: 100
Number false: 0
True   1   22.10494308046359
Number true: 100
Number false: 0
True   1   11.731197077293071
Number true: 100
Number false: 0
True   1   10.014117184015372
Number true: 100
Number false: 0
True   1   11.632679928891132
Number true: 100
Number false: 0
True   1   8.699881386020355
Number true: 100
Number false: 0
True   1   18.311128281425564
Number true: 100
Number false: 0
True   1   26.669622799112116
Number true: 100
Number false: 0
True   -1   -3.9626295670373715
```

Tara Boyle, Shannon Hicks,
Julia Hoffman, Morgan Weis

```
Number true: 100
Number false: 0
True   1   17.419764566889054
Number true: 100
Number false: 0
True   1   12.943622537652164
Number true: 100
Number false: 0
True   1   10.545586297022634
Number true: 100
Number false: 0
True   1   3.269854810423933
Number true: 100
Number false: 0
True   -1   -0.9994210491789559
Number true: 100
Number false: 0
True   1   11.083650053611674
Number true: 100
Number false: 0
True   1   33.085145371680355
Number true: 100
Number false: 0
True   1   1.2283170891846105
Number true: 100
Number false: 0
True   1   1.2247927079675769
Number true: 100
Number false: 0
[ 1  1  1  1  1  1 -1  1  1  1 -1  1  1  1  1  1  1  1  1  1  1  1 -1 -1
  1 -1  1  1  1  1  1  1  1  1 -1  1  1 -1  1  1  1  1 -1  1  1  1  1 -1  1
  1  1 -1 -1  1  1  1 -1  1  1 -1 -1  1  1  1  1 -1  1  1  1  1  1  1  1
  1  1  1 -1  1  1  1  1  1  1  1  1  1  1  1  1  1  1 -1  1  1  1  1 -1
  1  1  1  1]
[ True  True  True  True  True  True  True  True  True  True  True  True
  True  True  True  True  True  True  True  True  True  True  True  True
  True  True  True  True  True  True  True  True  True  True  True  True
  True  True  True  True  True  True  True  True  True  True  True  True
  True  True  True  True  True  True  True  True  True  True  True  True
```

```
 True  True  True  True  True  True  True  True  True  True  True  True
 True  True  True  True  True  True  True  True  True  True  True  True
 True  True  True  True  True  True  True  True  True  True  True  True
 True  True  True  True]
```

Applying the Classifier to the Houses and Restaurants in Blacksburg Solution:
After testing the accuracy of the model, we used it to predict if a coupon should be given to a new data set of houses, the houses discussed in prior sections. ‘1’ suggests that a coupon should be given to that house while ‘-1’ suggests that no coupon should be given.

House 1: 1 (coupon)
House 2: -1 (no coupon)
House 3: 1 (coupon)
House 4: 1 (coupon)
House 5: 1 (coupon)
House 6: 1 (coupon)
House 7: -1 (no coupon)
House 8: 1 (coupon)
House 9: 1 (coupon)
House 10: 1 (coupon)
House 11: 1 (coupon)
House 12: 1 (coupon)
House 13: -1 (no coupon)
House 14: -1 (no coupon)
House 15: 1 (coupon)
House 16: -1 (no coupon)
House 17: 1 (coupon)
House 18: -1 (no coupon)
House 19: -1 (no coupon)
House 20: 1 (coupon)
House 21: -1 (no coupon)
House 22: -1 (no coupon)
House 23: 1 (coupon)
House 24: 1 (coupon)
House 25: -1 (no coupon)
House 26: 1 (coupon)
House 27: 1 (coupon)
House 28: 1 (coupon)
House 29: 1 (coupon)

House 30: 1 (coupon)
House 31: -1 (no coupon)
House 32: 1 (coupon)
House 33: 1 (coupon)
House 34: 1 (coupon)
House 35: 1 (coupon)

Online Advantage Algorithm Description:
An algorithm to process online orders would be to allow drivers to be “on standby”, who would be able to deliver. The drivers would simply login to await an order while they are home or waiting in some other location. Then, they would be alerted when they are assigned a delivery and the information needed for them to complete it. This would be similar to how Uber works. If multiple drivers are available, the one with the least driving distance could be chosen. Or, if a driver is already picking up food for an earlier order when an alert is sent out, they could also pick up the new order due to the general proximity of the restaurants. There could be incentives for the drivers to be on standby, such as free food orders or payment (this is what would be different than Uber, as having only ten drivers, we are acting on a much smaller scale). For times where there is high demand, there could be a greater incentive for drivers to be on stand-by, similar to Uber’s peak rates. The advantages to this are its flexibility and simplicity, especially for the drivers. They get rewarded even if they do not accept anything. The disadvantages for this are that there are possibilities that orders may not always be fulfilled due to the fact that the driver’s participation is voluntary, and that Drop N’ Dine must reward the drivers in some way for being on stand-by. However, this is perhaps the simplest method of fulfilling the orders, due to the small number of drivers.

Appendix:
<u>Customers Restaurant Preference:</u>
1: '622 North'-5, Houses 11, 12, 13, 26, 32
2: 'Au Bon Pain' - 0
3: 'Avellinos Italian & Pizzeria'-4, Houses 14, 19, 23, 33
4: 'Backstreets Restaurant'- 3, Houses **7**, 25, 35
5: 'Beijing Buffet'- 0
6: "Benny Marzano's"- 0
7: "Big Al's Grille & Sportsbar”- 0
8: 'Blacksburg Taphouse'- 5, Houses 8, 16, 22, 24, 30
9: "Bojangles' Famous Chicken"-5, Houses 4, 9, 15, 28, 29
10: "Bollo's Cafe & Bakery"- 1, House 2
11: "Boudreaux's Restaurant"-4, House 3, 6, 18, 20
12: 'Buffalo Wild Wings'- 5, House 1, 5, 10, 27, 34

Tara Boyle, Shannon Hicks,
Julia Hoffman, Morgan Weis

13: 'Bull & Bones Brewhaus & Grill'- 0
14: 'Cabo Fish Taco'- 3, house 17, 21, 31

Houses and their respective coordinates:

1: Lacy Lane [-80.4497882, 37.2386646] (-80.450, 37.239)
2: College Avenue [-80.41595490000002, 37.230044] (-80.416, 37.230)
3: Seneca Drive [-80.4185263, 37.247930700000005] (-80.419, 37.248)
4: Merrimac Road [-80.4479346, 37.201342100000005] (-80.448, 37.201)
5: University City Blvd [-80.4345629, 37.240897] (-80.435, 37.241)
6: University City Blvd[-80.4333152, 37.2413695] (-80.433, 37.241)
7: Harrell Street [-80.41442579999999, 37.2248956] (-80.414, 37.225)
8: Givens Lane [-80.4177994, 37.2530011] (-80.418, 37.253)
9: Kraft Drive SW [-80.40964220000001, 37.19987020000001] (-80.401, 37.198)
10: Richmond Lane [-80.4500121, 37.2149286] (-80.450, 37.215)
11: Birch Leaf Lane [-80.407616, 37.2667906] (-80.408, 37.267)
12: Progress St NW [-80.4221328, 37.2457728] (-80.422, 37.246)
13: Mary Jane Circle [-80.4178428, 37.2451441] (-80.418, 37.245)
14: Landsdowne St [-80.3984653, 37.213431] (-80.398, 37.213)
15: Davis St [-80.4141099, 37.1866604] (-80.414, 37.187)
16: Turner St NW [-80.4197012, 37.2331366] (-80.420, 37.233)
17: Harding Avenue [-80.4066005, 37.2376964 (-80.407, 37.238)
18: N Main St [-80.4188777, 37.241741299999994] (-80.419, 37.242)
19: S Main St [-80.39868790000001, 37.207890899999995] (-80.399, 37.208)
20: Shenandoah Cir [-80.4242541, 37.2465592] (-80.424, 37.247)
21: Lee St SE [-80.4045744, 37.234787700000005] (-80.405, 37.235)
22: N Main St[-80.4183896, 37.233002899999995] (-80.418, 37.233)
23: Fairfax Road [-80.3958252, 37.2070097] (-80.396, 37.207)
24: Birch Leaf Lane [-80.40552509999999, 37.264704200000004] (-80.406, 37.265)
25: Draper Rd SW [-80.41072120000001, 37.2242783] (-80.411, 37.224)
26: Givens Ln [-80.41747059999999, 37.25361470000001] (-80.417, 37.254)
27: McBryde Dr [-80.42984179999999, 37.2376073] (-80.430, 37.238)
28: Tomko Rd [-80.44730009999999, 37.2268296] (-80.447, 37.227)
29: Heather Dr [-80.4473301, 37.216737200000004] (-80.447, 37.217)
30: Sunridge Dr [-80.42800909999998, 37.2442611] (-80.428, 37.244)
31: Ridgeview Dr [-80.3977963, 37.2365416] (-80.398, 37.237)
32: Hidden Nest Dr [-80.4074484, 37.2529013] (-80.407, 27.253)
33: Essex Ct [-80.39308829999999, 37.20505870000001] (-80.393, 37.205)
34: Quail Dr [-80.441546, 37.2336475] (-80.442, 37.234)
35: Palmer Dr [-80.40419399999999, 37.2248382] (-80.404, 37.225)

Tara Boyle, Shannon Hicks,
Julia Hoffman, Morgan Weis

Restaurant Coordinates

1: 622 North [-80.4194181, 37.234521]
2: Au Bon Pain [-80.41591564, 37.2301771]
3: Avellino's [-80.4018148, 37.2141015]
4: Backstreet's [-80.41247863, 37.22823855]
5: Beijing [-80.4032092, 37.2190136]
6: Benny's [-80.4153014, 37.229418200000005]
7: Big Al's [-80.4148723, 37.2300255]
8: Blacksburg Taphouse [-80.4186, 37.233486]
9: Bojangles [-80.39859990000001, 37.2032862]
10: Bollos [-80.415858, 37.2297829]
11: Boudreaux's [-80.4149254, 37.23006470000001]
12: BWW [-80.42129967, 37.23403715]
13: Bull and Bones [-80.4023644000000355, 37.213730299999995]
14: Cabo's [-80.41315172, 37.22880]