DC Food Truck Secondary Trading Platform

November 20, 2014

Dave Gupta
Evan Schlessinger
Vince Martinicchio

Problem Definition

- Washington D.C. has limited supply of "Prime" locations for Food Trucks
- The current lottery system assigns locations to food trucks randomly
- Formal system for trading location does not exist

This leads to underutilization of Food Truck spaces, which is bad for all parties involved

“As-is” System: Primary Assignments

- Assignments for each day of the week are done once a month by random assignment for 10 locations
- Vendor preference for locations is collected but not incorporated into assignments
- Fee: $25 to enter lottery, $150 to occupy the space

Slide 3

1 are the fees monthly?

Dave Gupta,

“As-is” System: Secondary Trading

- All trades are 1 for 1, and must be approved by DCRA via email
- Trades are conducted informally and through personal connections and email listserv
- No formal system exists to identify and facilitate potential trades

This project aims to improve Secondary Trading of location assignments

System Concept

- With the initial assignments as input, food trucks select:
  - Day/ location combinations that they are assigned and would like to trade
  - Day/ location combinations that they want and do not own
- Using this information, an algorithm re-assigns the day/location pairs amongst the trucks

System Concept (cont’d)

- In entering secondary trading, each food truck either ends up with either:
  - A better location based on their preferences
  - OR their same initial location

- The algorithm reassigns trucks via trades among the trucks, determined by the algorithm based on the preferences

Algorithm Survey and Attempted Approaches

Algorithms Examined

1. **Gale-Shapley**, and variations of its applications:
    - Kidney Exchange Problem
    - Resident - Hospital Matching Problem
    - House Swapping Problem

→ Research did not reveal any applicable problem that could be modified for our use. We discovered this problem is unique because of both the “ownership” property and the assignment of multiple spots.

Algorithm Survey and Attempted Approaches

Algorithms Examined

2. **Linear Programming Approach**
   - Preference-weighted Integer Problem
     - Lead to ambiguous improvements for individual trucks
     - More fundamentally, issues of converting ordinal to cardinal preferences and equating desirability of days and locations

   - Lexicographical/Pre-emptive Goal Programming
     - Same issues as above but also with increased complexity for end-user

Research: Actual Food Trucks Interviews

- Survey conducted to interview actual food truck owners regarding:
    - Current method of trading and frequency
    - Understanding of the assignment process
    - General location preferences
    - Interest in Secondary Trading Platform

Research: Actual Food Trucks Interview Findings

Food Trucks Owners were:

- Generally dissatisfied with complexity of current system
- Interested in any improvements in trading process
- Primarily focused on simplicity of system
- Desire consistency in weekly assignments

Research: DCRA Administrators

- The DCRA desires a system to facilitate food truck assignment trading after the lottery assignments to maximize utilization and revenue

- Requirements
  - Easy to use for Administrators (Automated)
  - Easy to use to Food Trucks (Increased participation)
  - Easy to implement (Automated)
  - Low cost and affordable
  - Maintainable

System Implementation Topics

1. System Boundaries
2. Input/Output
3. System Interfaces
4. Algorithm Description
5. “Week in the Life” timeline
6. Results of Factory Testing

1. System Boundaries

Excluded:

- Day of Enforcement
- Enforcement Method
- “Day of” issues for Food Trucks
- Not concerned with food consumers
- Not concerned with multiple vendors with the same cuisine at a given location

2. Input/Output

Inputs

- MRL Lottery Assignment Schedule
- Truck requested trades and associated preferences

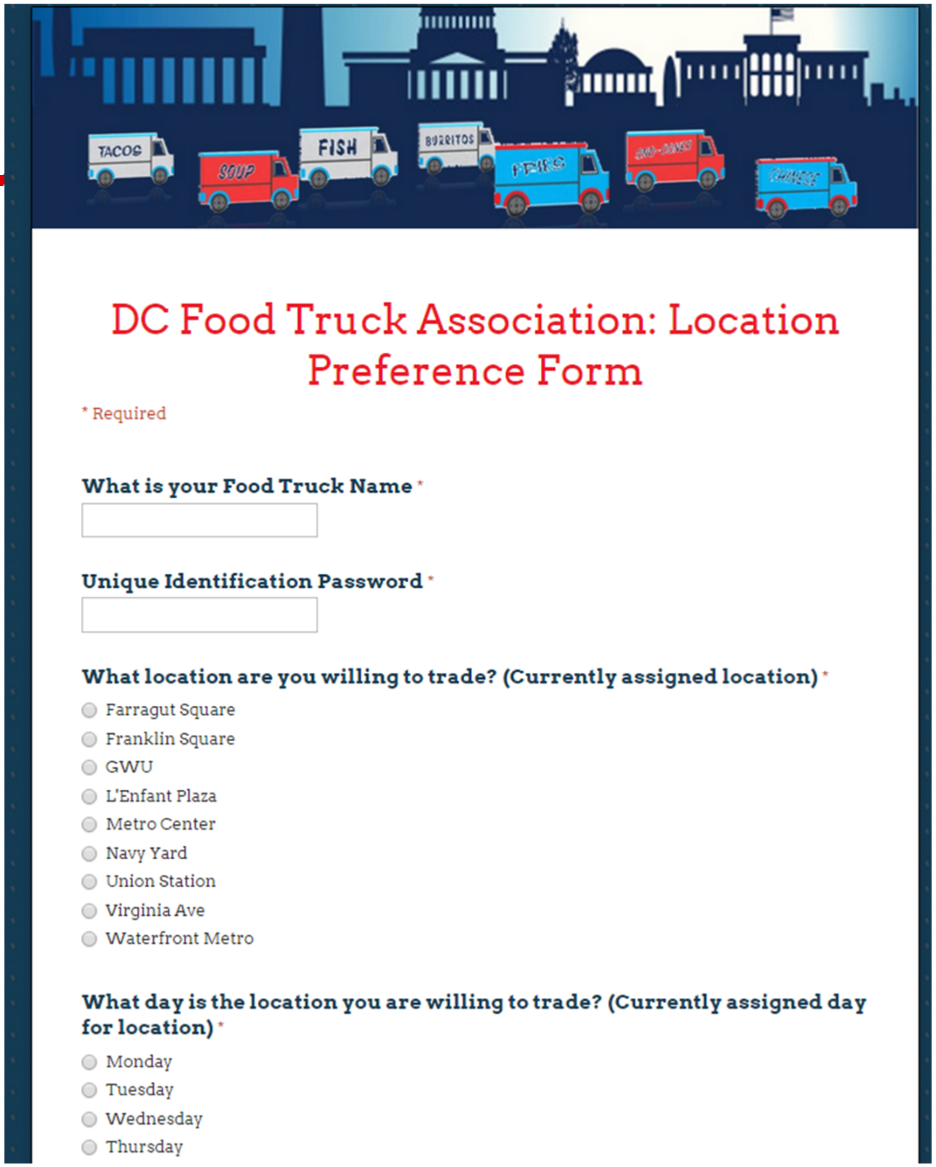
Output

- Reassigned Locations Schedule

3. System Interfaces

- www.foodtrucktrade.com
- Web interface created to allow preference inputs from Food Trucks
- Mobile device ready

- Password protected to authenticate entries
- Current assignment input validation pending



DC Food Truck Association: Location Preference Form

* Required

What is your Food Truck Name *

Unique Identification Password *

What location are you willing to trade? (Currently assigned location) *

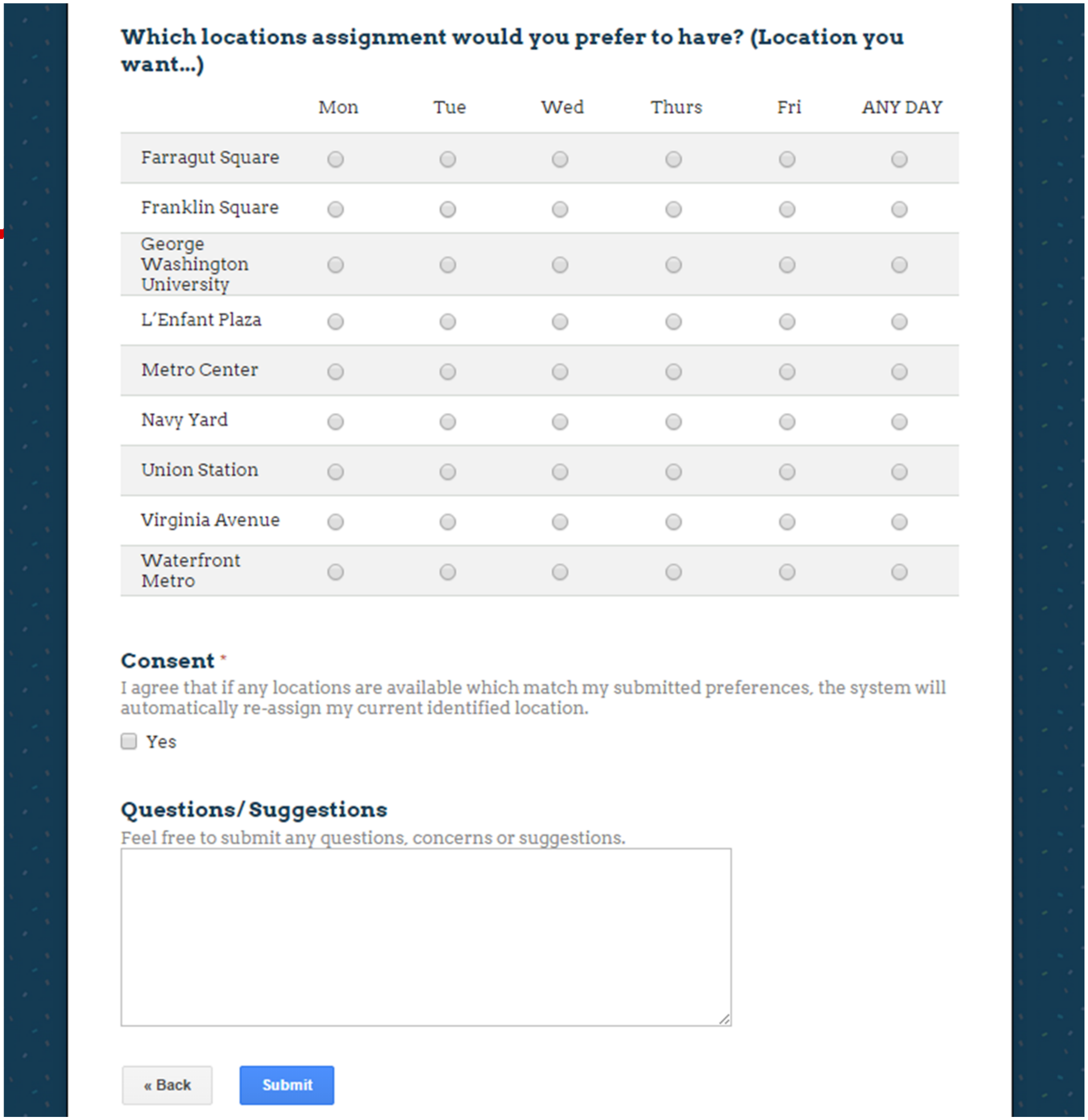
○ Farragut Square
○ Franklin Square
○ GWU
○ L'Enfant Plaza
○ Metro Center
○ Navy Yard
○ Union Station
○ Virginia Ave
○ Waterfront Metro

What day is the location you are willing to trade? (Currently assigned day for location) *

○ Monday
○ Tuesday
○ Wednesday
○ Thursday

3. System Interfaces (cont'd)

- Any potential trades will be automatically executed
- Trucks can select multiple locations on a given day



Which locations assignment would you prefer to have? (Location you want...)

	Mon	Tue	Wed	Thurs	Fri	ANY DAY
Farragut Square	○	○	○	○	○	○
Franklin Square	○	○	○	○	○	○
George Washington University	○	○	○	○	○	○
L'Enfant Plaza	○	○	○	○	○	○
Metro Center	○	○	○	○	○	○
Navy Yard	○	○	○	○	○	○
Union Station	○	○	○	○	○	○
Virginia Avenue	○	○	○	○	○	○
Waterfront Metro	○	○	○	○	○	○

Consent *

I agree that if any locations are available which match my submitted preferences, the system will automatically re-assign my current identified location.

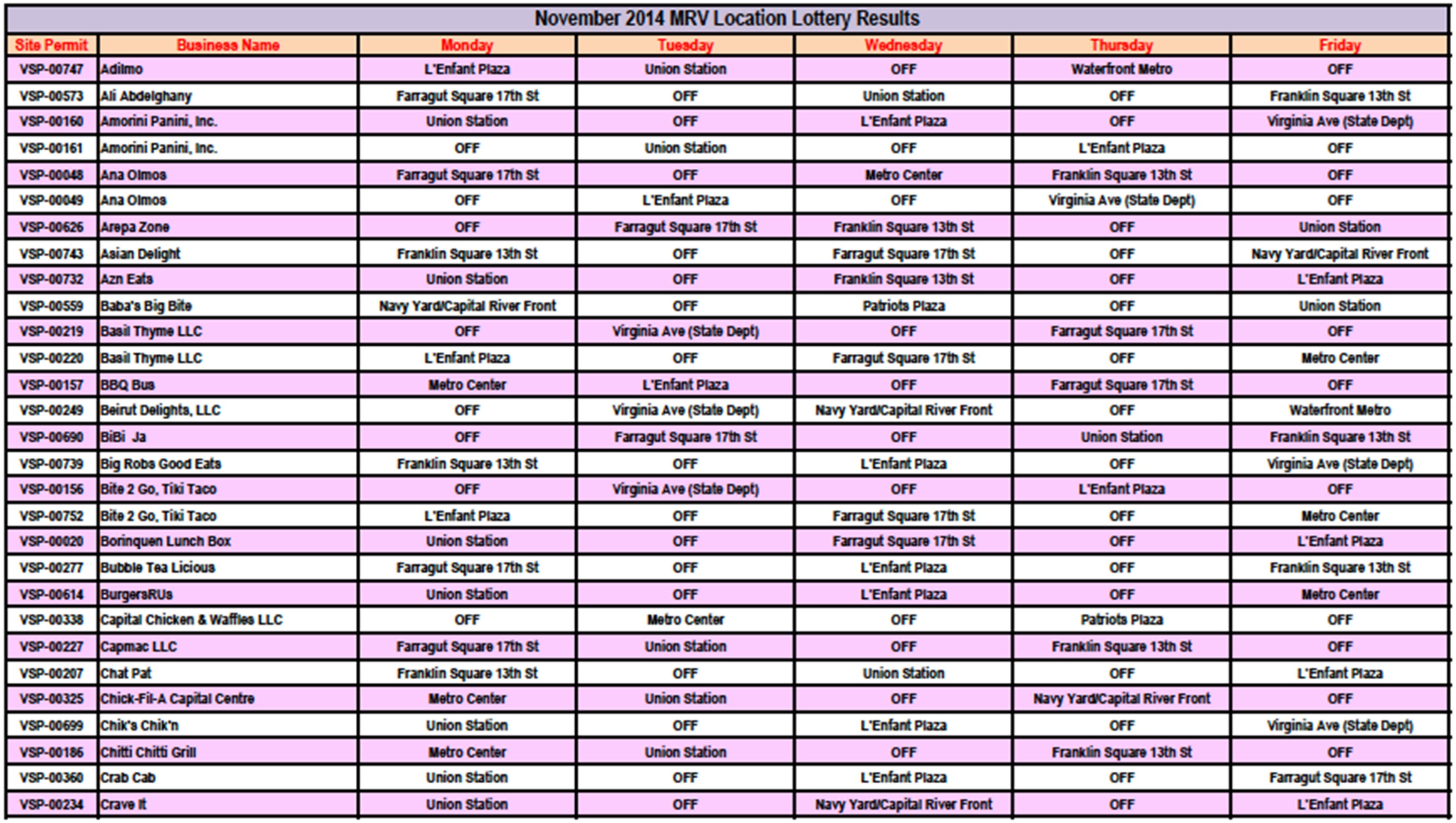
□ Yes

Questions/ Suggestions

Feel free to submit any questions, concerns or suggestions.

4. Algorithm Description

Step 1: Read in MRV Lottery Results



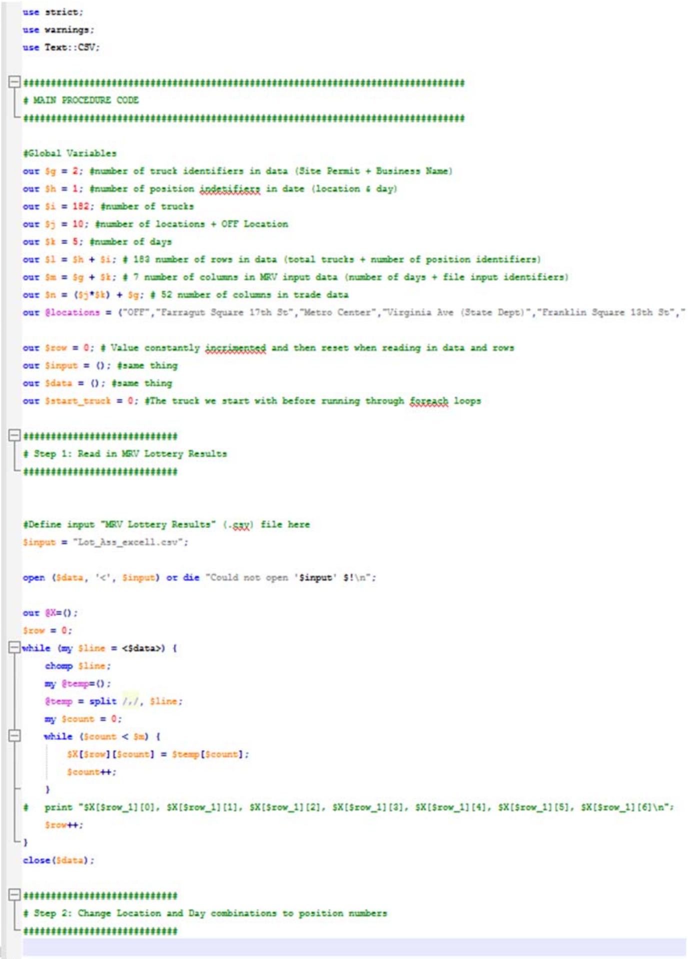
November 2014 MRV Location Lottery Results

Site Permit	Business Name	Monday	Tuesday	Wednesday	Thursday	Friday
VSP-00747	Adilmo	L'Enfant Plaza	Union Station	OFF	Waterfront Metro	OFF
VSP-00573	Ali Abdelghany	Farragut Square 17th St	OFF	Union Station	OFF	Franklin Square 13th St
VSP-00160	Amorini Panini, Inc.	Union Station	OFF	L'Enfant Plaza	OFF	Virginia Ave (State Dept)
VSP-00161	Amorini Panini, Inc.	OFF	Union Station	OFF	L'Enfant Plaza	OFF
VSP-00048	Ana Olmos	Farragut Square 17th St	OFF	Metro Center	Franklin Square 13th St	OFF
VSP-00049	Ana Olmos	OFF	L'Enfant Plaza	OFF	Virginia Ave (State Dept)	OFF
VSP-00626	Arepa Zone	OFF	Farragut Square 17th St	Franklin Square 13th St	OFF	Union Station
VSP-00743	Asian Delight	Franklin Square 13th St	OFF	Farragut Square 17th St	OFF	Navy Yard/Capital River Front
VSP-00732	Azn Eats	Union Station	OFF	Franklin Square 13th St	OFF	L'Enfant Plaza
VSP-00559	Baba's Big Bite	Navy Yard/Capital River Front	OFF	Patriots Plaza	OFF	Union Station
VSP-00219	Basil Thyme LLC	OFF	Virginia Ave (State Dept)	OFF	Farragut Square 17th St	OFF
VSP-00220	Basil Thyme LLC	L'Enfant Plaza	OFF	Farragut Square 17th St	OFF	Metro Center
VSP-00157	BBQ Bus	Metro Center	L'Enfant Plaza	OFF	Farragut Square 17th St	OFF
VSP-00249	Beirut Delights, LLC	OFF	Virginia Ave (State Dept)	Navy Yard/Capital River Front	OFF	Waterfront Metro
VSP-00690	BiBi Ja	OFF	Farragut Square 17th St	OFF	Union Station	Franklin Square 13th St
VSP-00739	Big Robs Good Eats	Franklin Square 13th St	OFF	L'Enfant Plaza	OFF	Virginia Ave (State Dept)
VSP-00156	Bite 2 Go, Tiki Taco	OFF	Virginia Ave (State Dept)	OFF	L'Enfant Plaza	OFF
VSP-00752	Bite 2 Go, Tiki Taco	L'Enfant Plaza	OFF	Farragut Square 17th St	OFF	Metro Center
VSP-00020	Borinquen Lunch Box	Union Station	OFF	Farragut Square 17th St	OFF	L'Enfant Plaza
VSP-00277	Bubble Tea Licious	Farragut Square 17th St	OFF	L'Enfant Plaza	OFF	Franklin Square 13th St
VSP-00614	BurgersRUs	Union Station	OFF	L'Enfant Plaza	OFF	Metro Center
VSP-00338	Capital Chicken & Waffles LLC	OFF	Metro Center	OFF	Patriots Plaza	OFF
VSP-00227	Capmac LLC	Farragut Square 17th St	Union Station	OFF	Franklin Square 13th St	OFF
VSP-00207	Chat Pat	Franklin Square 13th St	OFF	Union Station	OFF	L'Enfant Plaza
VSP-00325	Chick-Fil-A Capital Centre	Metro Center	Union Station	OFF	Navy Yard/Capital River Front	OFF
VSP-00699	Chik's Chik'n	Union Station	OFF	L'Enfant Plaza	OFF	Virginia Ave (State Dept)
VSP-00186	Chitti Chitti Grill	Metro Center	Union Station	OFF	Franklin Square 13th St	OFF
VSP-00360	Crab Cab	Union Station	OFF	L'Enfant Plaza	OFF	Farragut Square 17th St
VSP-00234	Crave It	Union Station	OFF	Navy Yard/Capital River Front	OFF	L'Enfant Plaza

4. Algorithm Description (cont'd)

Step 2: Change Location/Day Combinations to unique position numbers

- 10 locations on 5 days
- Thus,each day and location is numbered 0-49

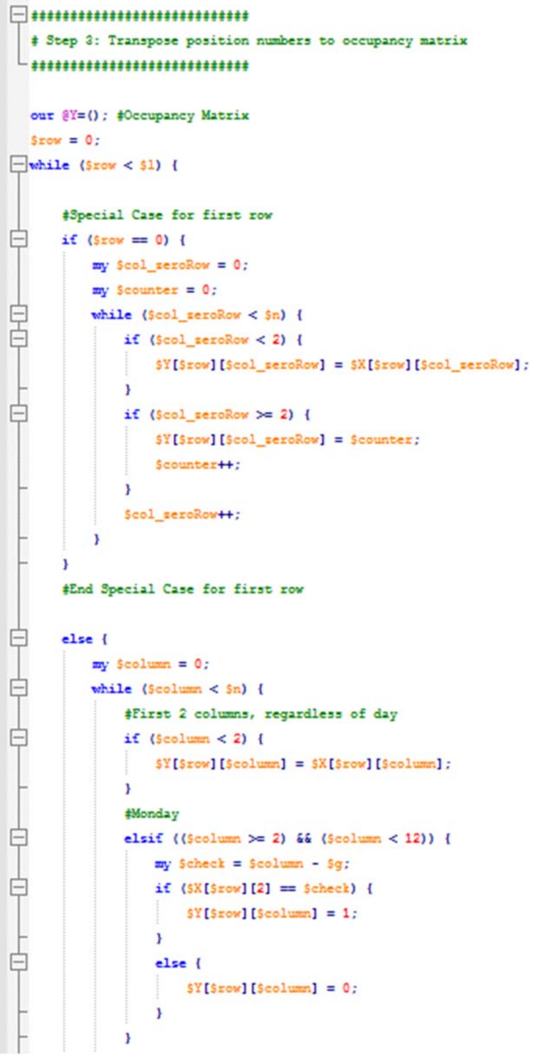


```
use strict;
use warnings;
use Text::CSV;

################################################################################
# MAIN PROCEDURE CODE
################################################################################

#Global Variables
our $g = 2; #number of truck identifiers in data (Site Permit + Business Name)
our $h = 1; #number of position indetifiers in date (location & day)
our $i = 182; #number of trucks
our $j = 10; #number of locations + OFF Location
our $k = 5; #number of days
our $l = $h + $i; # 183 number of rows in data (total trucks + number of position identifiers)
our $m = $g + $k; # 7 number of columns in MRV input data (number of days + file input identifiers)
our $n = ($j*$k) + $g; # 52 number of columns in trade data
our @locations = ("OFF","Farragut Square 17th St","Metro Center","Virginia Ave (State Dept)","Franklin Square 13th St","

our $row = 0; # Value constantly incrimented and then reset when reading in data and rows
our $input = (); #same thing
our $data = (); #same thing
our $start_truck = 0; #The truck we start with before running through foreach loops

############################
# Step 1: Read in MRV Lottery Results
############################


#Define input "MRV Lottery Results" (.csv) file here
$input = "Lot_Ass_excell.csv";

open ($data, '<', $input) or die "Could not open '$input' $!\n";

our @X=();
$row = 0;
while (my $line = <$data>) {
    chomp $line;
    my @temp=();
    @temp = split /,/, $line;
    my $count = 0;
    while ($count < $m) {
        $X[$row][$count] = $temp[$count];
        $count++;
    }
#   print "$X[$row_1][0], $X[$row_1][1], $X[$row_1][2], $X[$row_1][3], $X[$row_1][4], $X[$row_1][5], $X[$row_1][6]\n";
    $row++;
}
close($data);

############################
# Step 2: Change Location and Day combinations to position numbers
############################
```

4. Algorithm Description (cont'd)

Step 3: Transpose Position Numbers to Occupancy Matrix

- Each column represents a position, where the food trucks stay on the rows
- This allows algorithm to easily locate data on multiple matrices using the same inputs

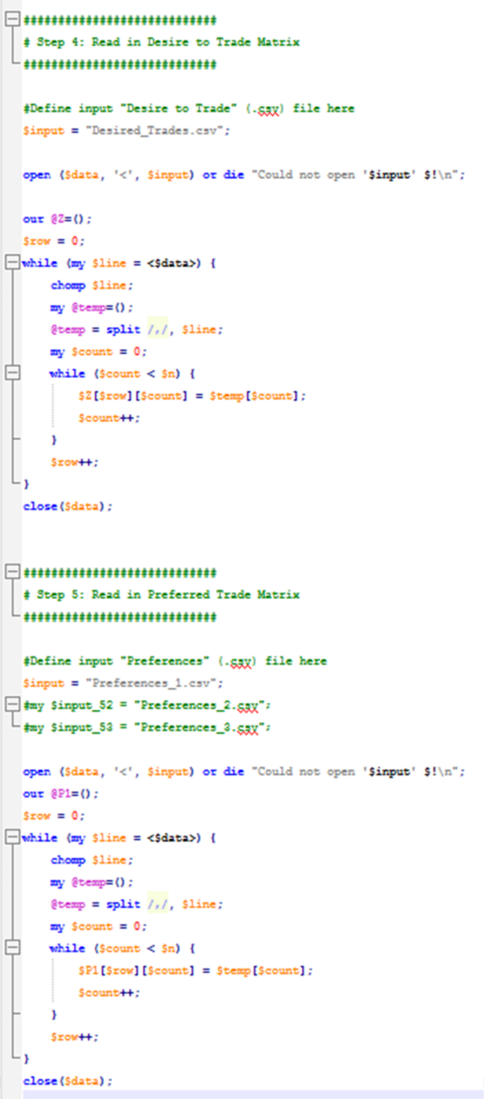


```
##############################
# Step 3: Transpose position numbers to occupancy matrix
##############################

our @Y=(); #Occupancy Matrix
$row = 0;
while ($row < $l) {

    #Special Case for first row
    if ($row == 0) {
        my $col_zeroRow = 0;
        my $counter = 0;
        while ($col_zeroRow < $n) {
            if ($col_zeroRow < 2) {
                $Y[$row][$col_zeroRow] = $X[$row][$col_zeroRow];
            }
            if ($col_zeroRow >= 2) {
                $Y[$row][$col_zeroRow] = $counter;
                $counter++;
            }
            $col_zeroRow++;
        }
    }
    #End Special Case for first row

    else {
        my $column = 0;
        while ($column < $n) {
            #First 2 columns, regardless of day
            if ($column < 2) {
                $Y[$row][$column] = $X[$row][$column];
            }
            #Monday
            elsif (($column >= 2) && ($column < 12)) {
                my $check = $column - $g;
                if ($X[$row][2] == $check) {
                    $Y[$row][$column] = 1;
                }
                else {
                    $Y[$row][$column] = 0;
                }
            }
```

4. Algorithm Description (cont’d)

Steps 4 and 5: Read in “Desire to Trade” and “Preferences” Matrices

- Matrices are in the same format as the Occupancy Matrix



```
############################
# Step 4: Read in Desire to Trade Matrix
############################

#Define input "Desire to Trade" (.csv) file here
$input = "Desired_Trades.csv";

open ($data, '<', $input) or die "Could not open '$input' $!\n";

our @Z=();
$row = 0;
while (my $line = <$data>) {
    chomp $line;
    my @temp=();
    @temp = split /,/, $line;
    my $count = 0;
    while ($count < $n) {
        $Z[$row][$count] = $temp[$count];
        $count++;
    }
    $row++;
}
close($data);


############################
# Step 5: Read in Preferred Trade Matrix
############################

#Define input "Preferences" (.csv) file here
$input = "Preferences_1.csv";
#my $input_52 = "Preferences_2.csv";
#my $input_53 = "Preferences_3.csv";

open ($data, '<', $input) or die "Could not open '$input' $!\n";
our @P1=();
$row = 0;
while (my $line = <$data>) {
    chomp $line;
    my @temp=();
    @temp = split /,/, $line;
    my $count = 0;
    while ($count < $n) {
        $P1[$row][$count] = $temp[$count];
        $count++;
    }
    $row++;
}
close($data);
```

4. Algorithm Description (cont’d)

Steps 6 and 7: Find participating trucks, and eliminate trucks whose preferences are not available

- This allows the algorithm to focus only on trucks that are actually capable of trading
- This also sets up a single matrix of the trucks that are capable of trading

4. Algorithm Description (cont’d)

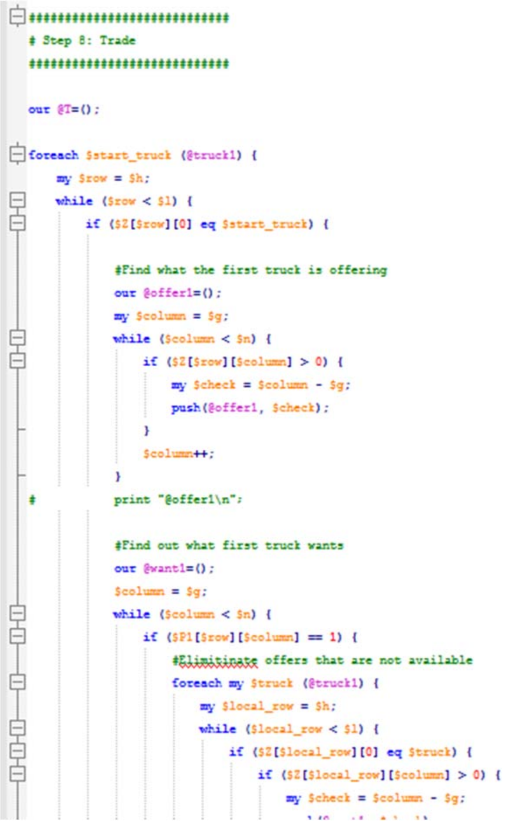
Step 8: “Set up the House”

- This is a single matrix containing all of the positions to be traded
- The dimensions correspond to the single matrix of food trucks generated in the previous step
- This matrix is the foundation of the algorithm (Why it is called “The House”)

4. Algorithm Description (cont’d)

Step 9: Trade

- From a starting position X from “The House”
  - Identify who wants position X
  - What do the trucks who want position X currently own?



```
############################
# Step 8: Trade
############################

our @T=();

foreach $start_truck (@truck1) {
    my $row = $h;
    while ($row < $l) {
        if ($Z[$row][0] eq $start_truck) {

            #Find what the first truck is offering
            our @offer1=();
            my $column = $g;
            while ($column < $n) {
                if ($Z[$row][$column] > 0) {
                    my $check = $column - $g;
                    push(@offer1, $check);
                }
                $column++;
            }
#           print "@offer1\n";

            #Find out what first truck wants
            our @want1=();
            $column = $g;
            while ($column < $n) {
                if ($P1[$row][$column] == 1) {
                    #Elimitinate offers that are not available
                    foreach my $truck (@truck1) {
                        my $local_row = $h;
                        while ($local_row < $l) {
                            if ($Z[$local_row][0] eq $truck) {
                                if ($Z[$local_row][$column] > 0) {
                                    my $check = $column - $g;
```

The Goal is to loop back to the starting position to create a multi-way trade

4. Algorithm Description (cont’d)

Step 9: Trade (Continued)

- Simple Example:

3 way trade between
-Truck A
-Truck B
-Truck D

Truck	A	B	C	D
Position Owned	X	Y	Z	W
Position Desired	W	X	X	Y

4. Algorithm Description (cont'd)

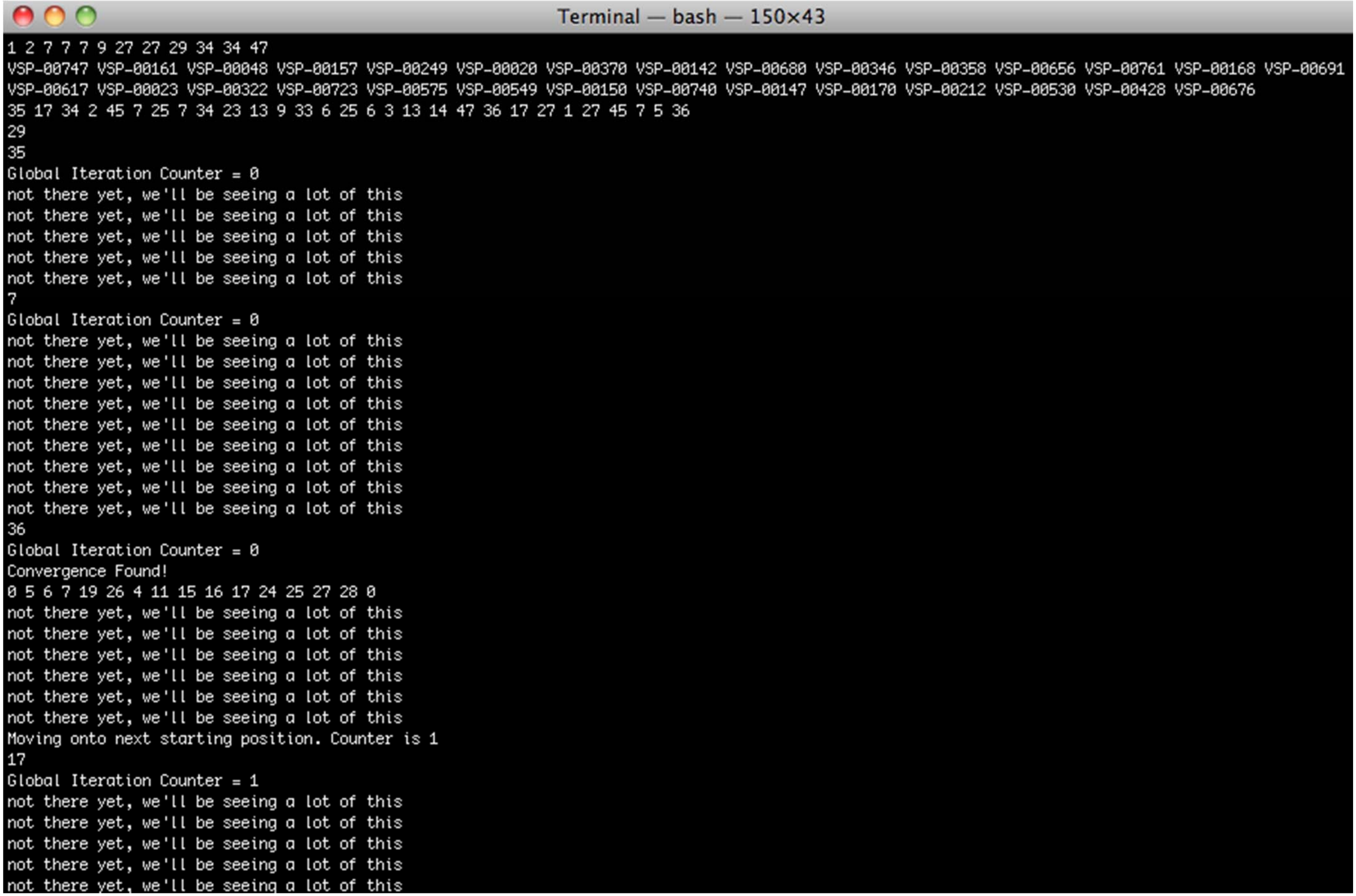
Step 10 and 11: Reformat and Output Results

- Intention is that the output results will look similar in format and style to the input

Algorithm Progress (For Dry Run Presentation Only)

- The idea to use “The House” of positions came this week
- This is a much simpler tracking mechanism while searching for multi-way trades
- The algorithm, with comments, is over 500 lines

Pending minor changes, the algorithm is functional, and will be completed by December



```
1 2 7 7 7 9 27 27 29 34 34 47
VSP-00747 VSP-00161 VSP-00048 VSP-00157 VSP-00249 VSP-00020 VSP-00370 VSP-00142 VSP-00680 VSP-00346 VSP-00358 VSP-00656 VSP-00761 VSP-00168 VSP-00691
VSP-00617 VSP-00023 VSP-00322 VSP-00723 VSP-00575 VSP-00549 VSP-00150 VSP-00740 VSP-00147 VSP-00170 VSP-00212 VSP-00530 VSP-00428 VSP-00676
35 17 34 2 45 7 25 7 34 23 13 9 33 6 25 6 3 13 14 47 36 17 27 1 27 45 7 5 36
29
35
Global Iteration Counter = 0
not there yet, we'll be seeing a lot of this
not there yet, we'll be seeing a lot of this
not there yet, we'll be seeing a lot of this
not there yet, we'll be seeing a lot of this
not there yet, we'll be seeing a lot of this
7
Global Iteration Counter = 0
not there yet, we'll be seeing a lot of this
not there yet, we'll be seeing a lot of this
not there yet, we'll be seeing a lot of this
not there yet, we'll be seeing a lot of this
not there yet, we'll be seeing a lot of this
not there yet, we'll be seeing a lot of this
not there yet, we'll be seeing a lot of this
not there yet, we'll be seeing a lot of this
not there yet, we'll be seeing a lot of this
36
Global Iteration Counter = 0
Convergence Found!
0 5 6 7 19 26 4 11 15 16 17 24 25 27 28 0
not there yet, we'll be seeing a lot of this
not there yet, we'll be seeing a lot of this
not there yet, we'll be seeing a lot of this
not there yet, we'll be seeing a lot of this
not there yet, we'll be seeing a lot of this
not there yet, we'll be seeing a lot of this
Moving onto next starting position. Counter is 1
17
Global Iteration Counter = 1
not there yet, we'll be seeing a lot of this
not there yet, we'll be seeing a lot of this
not there yet, we'll be seeing a lot of this
not there yet, we'll be seeing a lot of this
not there yet, we'll be seeing a lot of this
```

“Month in the Life” Timeline

- Trucks receive initial assignments from Lottery for January
- Trucks enter location and days that they wish to trade and the location and days which they are willing to trade for
- Revised assignments released from secondary trading algorithm

Evaluation of Implementation

- Algorithm only reassigns positions if all trucks involved possess improved positions relative to their previous positions
- Algorithm is written in PERL, an open-source scripting language

Outlook and Future Considerations

- Incorporation of mechanism for “Last Minute” Trades
- Algorithm refinement
- Interface refinement
- Transition to primary assignment