Vince Martinicchio
Evan Schlessinger
Dave Gupta
OR 699 - Master's Project
10/9/2014

Vending Location Trading Platform Progress Report

Algorithm Development

Based on our initial meeting with Professor Hoffman to discuss various approaches, we decided pursue Linear Programming. The group held follow on meetings and designed an initial LP algorithm to consider preferences in location trades. The algorithm considers preferences on a week by week basis, thus, each food truck vendor would have the opportunity of re-entering or updating their preferences each week. This allows us to account for shifts in location preferences that may stem from successful trades conducted in the month.

The algorithm works by using a commercial software called MPL by Maximal Software. The freely available solver utility employed within MPL is LP Solve, which we may later use directly through a PERL script. For all vendors, each entry in the ordered/ranked list of location-day preference is assigned values consisting of 3^15, 3^14, 3^13 etc. according to the preference position in the list. For any location-day combinations below the lowest preferred initial lottery assignment, a large negative value is assigned. Due to the exponential nature of the values for each of the location-day entries, the algorithm places top priority of the highest ranked location-day entries. The initial algorithm, developed is shown below in Figure 1.

```
TITLE
FoodTrucks;

INDEX
 truck := (1..8);
 day := (1..5);
 location := (1..3);

DATA
 space[location] := (2, 2, 2);
 preference[truck, day, location] := (0, -10000, -10000,
     27, 0, -10000,
```

```
     27, 9, 0,
     27, 9, 3,
     0, -10000, -10000,
      0, 27, -10000,
      -10000, 0, -10000,
      9, 27, 0,
      9, 27, 3,
      0, 27, -10000,
     27, 0, -10000,
     27, 9, 0,
     27, 9, 3,
     0, -10000, -10000,
     27, 0, -10000,
      9, 0, 27,
      27, 9, 0,
      3, 9, 27,
      0, 27, 9,
      27, 0, 9,
     -10000, -10000, 0,
     9, 3, 27,
     0, -10000, 27,
     -10000, 0, 27,
     -10000, -10000, 0,
      -10000, 27, 0,
      0, 0, 27,
      0, -10000, 27,
      27, 0, -10000,
      27, -10000, 0,
     3, 27, 9,
     0, -10000, -10000,
     9, 0, 27,
     27, -10000, 0,
     27, 9, 3,
      27, 3, 9,
      0, 27, -10000,
      -10000, 0, -10000,
      9, 27, 0,
      9, 27, 3) ;

BINARY VARIABLES
 assignment[truck, day, location];

MODEL
```

```
 MAX SUM(truck, day, location: preference * assignment);

SUBJECT TO
 Limited_Space[truck, day, location]:
  SUM(truck: assignment) <= space;

All_Assigned[truck, day, location]:
  SUM(location: assignment) = 1;

END
```

Figure 1

To test the script, we created an initial set of preference data for a scaled down set of 8 food truck vendors and 15 location-day pairs. Upon initial review, the output seemed promising since many vendors were seemingly trading locations and getting assignments that were generally higher on their preference rankings. The results from the test run is shown below in Figure 2, where the highlighted entries represent location-day pairs pre-assigned from the lottery (simulated data) and the grey shaded entries represent the final assignments once the algorithm had been run. Prior to running the algorithm, each location-day combination was given a unique corresponding numerical identification number to facilitate in the matrix manipulation and interface.

Ranking Order	Truck #1	Truck #2	Truck #3	Truck #4	Truck #5	Truck #6	Truck #7	Truck #8
1	1	4	13	13	4	10	7	6
2	7	13	7	10	13	9	9	13
3	3	7	10	5	2	13	8	5
4	4	11	3	8	8	1	11	2
5	5	2	5	7	9	8	2	3
6	2	1	9	2	6	7	10	7
7	13	3	8	3	10	11	4	11
8	11	5	6	4	5	4	3	10
9	10	8	11	6	1	5	1	8
10	6	9	14	1	7	15	6	9
11	9	10	2	9	3	3	13	1
12	14	6	4	11	14	6	14	4
13	8	14	12	14	12	14	12	14
14	12	15	1	15	11	12	5	12
15	15	12	15	12	15	2	15	15

Note: (hatched) =Assignment after algorithm
(yellow) =Initial assignment

Figure 2

<u>Algorithm Design Concern</u>

Although the algorithm successfully reassigned vendors to their top ranked preferences, in some cases lower ranked location-day entries that were assigned through the lottery, which can be thought of as being owned by the food trucks, were replaced by other spots lower on the preference rankings. Thus the location “ownership” was not taken into account by the linear programming algorithm since it was not keeping track of specific trades and considering initial given locations, but instead looking to maximize the net increase in value across all location assignments. That is, the model assigned locations in order to maximize each truck’s overall score, which includes assigning lower locations that the trucks initial assignment. Figure 3 below illustrates an example of this for Truck 7. The initial owned location-day pairs, correspond to the identifiers 8, 4, and 12 highlighted. Spot 8 seems to have been reassigned for the higher ranked spot 7, and spot 12 did not change, however, for this truck their initial spot 4 was replaced by a spot lower on their preference rankings, 14. We recognize that the vendors would not be agreeable to moving down on their rankings in any circumstance, thus we have identified this behavior in the algorithm as a concern.

Truck #7
7
9
8
11
2
10
4
3
1
6
13
14
12
5
15

Figure 3

While utilizing available resources to add additional constraints in the algorithm to remove this behavior, we are also concurrently considering alternate approaches of scripting using a step by step algorithm. This way, trucks cannot relinquish their location without receiving a better location in return. The Figure 4 below is a rough draft of an iterative algorithm concept perceived that will allow more control of the algorithm.

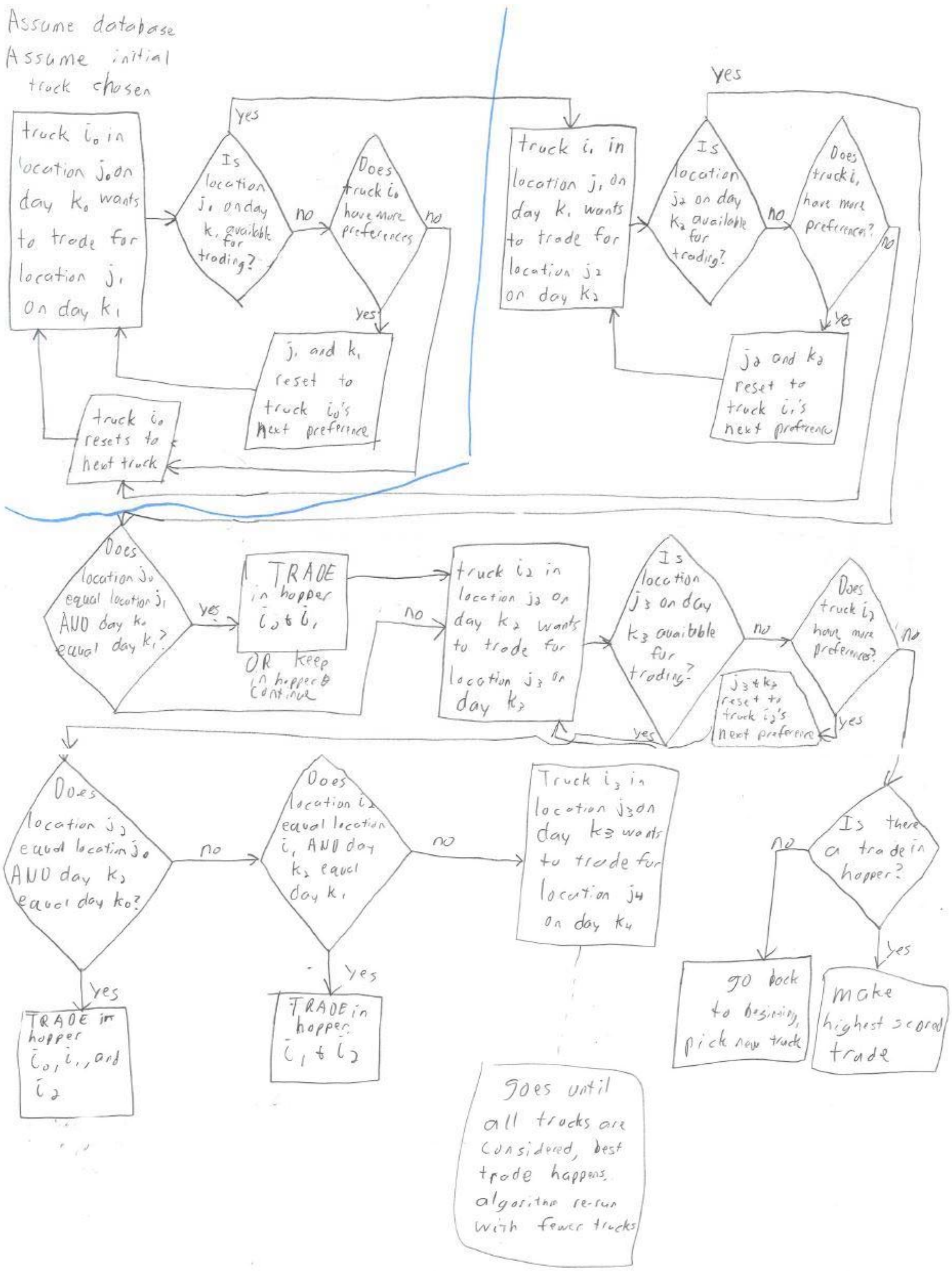
Assume database
Assume initial truck chosen
truck i_0 in location j_0 on day k_0 wants to trade for location j_1 on day k_1
Is location j_1 on day k_1 available for trading?
Does truck i_0 have more preferences
j_1 and k_1 reset to truck i_0's next preference
truck i_0 resets to next truck
truck i_1 in location j_1 on day k_1 wants to trade for location j_2 on day k_2
Is location j_2 on day k_2 available for trading?
Does truck i_1 have more preferences?
j_2 and k_2 reset to truck i_1's next preference
Does location j_0 equal location j_1 AND day k_0 equal day k_1?
TRADE in hopper i_0 & i_1
OR keep in hopper & continue
truck i_2 in location j_2 on day k_2 wants to trade for location j_3 on day k_3
Is location j_3 on day k_3 available for trading?
Does truck i_2 have more preferences?
j_3 & k_3 reset to truck i_2's next preference
Does location j_3 equal location j_0 AND day k_3 equal day k_0?
TRADE in hopper i_0, i_1, and i_2
Does location i_3 equal location i_1 AND day k_3 equal day k_1
TRADE in hopper i_1 & i_2
Truck i_3 in location j_3 on day k_3 wants to trade for location j_4 on day k_4
goes until all trucks are considered, best trade happens, algorithm re-run with fewer trucks
Is there a trade in hopper?
go back to beginning, pick new truck
make highest scored trade
yes
no

Figure 4

Interviews with Food Trucks

We also developed a questionnaire for food truck vendors and started reaching out to them to get some feedback and insight into the problem. From small sample size of 4 trucks, it was confirmed that food truck operators are not satisfied with the current system. Not all food trucks use the current listserv method for identifying and executing trades and some of the trucks affirmed that if they are not satisfied with their assigned location, they simply do not show up. When asked about opinions on reasonable amount of time to complete detailed preference ranking, the trucks were ok with a completion time of approximately 10 minutes. Once we start developing the user interface and preference collection form, we will be reaching out to the trucks who have indicated they would be willing to assist in providing more feedback. Additionally, we are reaching out to more trucks to establish a bigger data set.

Project Schedule and Status

According to the schedule, tasks 1-8 have due dates either today or prior to today, which is why those are the only ones that have a status attached to them.

1. Sign and submit NDAs. (Due September 25) (Dave)

   Status: Incomplete

   Comment: The original NDAs presented by Karen Wrege were found to be unsatisfactory. We are awaiting word from Professor Hoffman about receiving satisfactory NDAs to sign.

2. Get Data from sponsor (Due Monday, September 29) (Evan)

   Status: Incomplete

   Comment: Because the NDAs have been an issue, we have not been able to get data from the sponsor.

3. Agree on Algorithms (Due Thursday, October 2) (Vince and Evan)

   Status: Complete

   Comment: We initially agreed on a linear programming model, but have found a significant issue that we need to correct. We do however want to stay with a linear programming base approach

4. Build Initial Prototypes (Due Thursday, October 2) (Vince and Evan)

   Status: Complete

   Comment: We were able to build an initial prototype using dummy data. It was a linear programming model.

5. Solicit information from Food Trucks (Due Thursday, October 2) (Dave)

   Status: In Progress

   Comment: This task will remain in progress indefinitely, because we will take any information from the food trucks that we can get.

6. Begin writing documentation (Due Thursday, October 9) (Vince)

   Status: Incomplete

   Comment: The focus so far has been the Algorithm and initial documentation has thus far been overlooked.

7. Contact GW Professor H.G. Abeledo (Thursday, October 9 to Thursday October 16) (Evan)

   Status: Complete

   Comment: Professor H.G. Abeledo has been contacted and the team will be meeting with him the week of Monday, October 13.

8. Modify and expand algorithm for whole system/ entire data set (Thursday, October 9 to Thursday, October 16) (Vince & Evan)

   Status: On Hold

   Comment: Need to fix what is currently wrong with the current algorithm

9. Internally review algorithm (group + Dr. Hoffman) (Thursday, October 16 to Thursday, October 23)

10. Integrate algorithm with interfaces (Thursday, October 16 to Thursday, October 30) (Dave)

11. Show program to sponsor and gain feedback (Thursday, October 23 to Thursday, October 30) (Vince)

12. Incorporate comments from sponsor (Thursday, October 30 to Thursday, November 6) (Dave)

13. Update System Documentation (Thursday, October 30 to Thursday, November 6) (Evan)
    a. Requirements Document
    b. Algorithm Commentary
    c. System User Guide

14. Finalize integration of interfaces with program (Thursday, October 30 to Thursday, November 6) (Dave)

15. Show "Final Product" to sponsor (Thursday, November 6 to Thursday, November 13) (Team)

16. Refine "Final Product" from sponsor comments (Thursday, November 13 to Thursday, November 20) (Dave)

17. Finalize documentation (Thursday, November 13 to Thursday, November 20) (Vince)