**For Meeting 10/19**

Runs on 100x5 with menu size 2

10x10—tests 10,000 nodes then returns NaN, 10x6 runs sometimes

For all sizes, some iterations are Infeasible in solver, thought this appears to be using branching so some branches would be infeasible

10x10 runs for binary and ternary matrix, original system. Doesn’t work for 4 different levels of utility

Runs when all u\_ij=1 for 30x30, but y is no longer integer so it doesn’t yield a valid assignment, and we can’t have binary variables in the lower level or complementarity constraints (only LPs and convex QPs)

Made a reformulate to fix this. Runs on 10x10 I think, but gets a bit slow for any bigger. Running when U\_ij can be 0, 1, or 2, 10x13 crashes matlab for original formulation (works for 6x7), is pretty slow for new version and didn’t converge in time.

Bilevel seems to take complementarity constraints but bugs out even for 3x3, says matrix is ‘singular to working precision’

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**FactorInput\_generator\_and\_DOE2.m**

Setting a bunch of parameters I don’t get, calls Hung\_noChoice

Hung\_noChoice

Does some sort of assignment using repmap munkres

**For Meeting 10/26**

**Original formulation,** menu size 1 : 30x30 took 5400s to max out the 10,000 nodes, 30x15 took 963s to max out. I swear 10x10 ran at once in just a few seconds but now it maxes out after 397s.

**When y is non-integer: Menu size =1**

Ran both reformulation and original with U=1, 5x5, 10x10, 20x20, 20x10, all were integer. Sometimes the reformulation freezes for awhile on the first iteration (10x20 crashed the computer)

For 5x10, y was non-binary (some were, others were 0.5), only for reformulation. Was integer for mxn when m≥n+1, and has non-integer for m≤ nk. The solution y will have m-1 rows with y=1 for one j, and two rows with n-m+1 entries being 0.5—one of these rows will be the no-choice row. Example for a 4x7:

0 0 0 0 0 0 1

0 1 0 0 0 0 0

0 0 0 0 1 0 0

0.5000 0 0.5000 0.5000 0 0.5000 0

0.5000 0 0.5000 0.5000 0 0.5000 0

When I made U a matrix of zeros and ones, the solution was similar (for same dimensions) except the number of 0.5 entries per row was always four (except 2x4 was 3)

When U is zeros, ones, twos: same but now mxm+2 gives an integer result and the number of 0.5 entries is three.

**When y is non-integer: Menu size =2**

When U is zeros, ones and twos, mxm+2 gives an integer result. When it’s non-integer, there may be one or more columns with one 1.0 entry only, each other column has two 0.5 entries from the remaining rows. Same for when U is zeros and ones, tho 2x4 was integer sometimes.

Ran both reformulation and original with U=1, 5x5, 10x10, 20x20, 20x10, all were integer.

**Simplifying the v constraint: Menu size=2**

For 5x5 with U=zeros, ones, everything is the same (including obj) except some of the menus.

With 4x4 rand U was same for everything but some menus sometimes.

For 4x6 rand U, first time initial\_bilevel capped out and simpe\_v ran in 3 seconds, but the second trial had simple\_v cap out and initial\_bilevel ran in 144s.

Still won’t run 10x10

**For Meeting 11/1**

**When y is non-integer: Menu size =2**

When U is zeros and ones: even when D and r are random and w>=0 there are still decimal y values ☹

Ranking U column entries from 1 to m: still doesn’t run for more than 5 drivers, behavior is same as when U=rand.

Letting U=zeros and ones with adding perturbation 0.01\*rand: appears to be same as when U=rand

**For the single-level semi-interdicting formulation:** All params random, menu=2.

3x8: y has non-integer values (two 0.5’s per column )

8x8: w=z=0, every job is offered twice (3 trials)

8x8, menu=3: non-integer y (two 0.5’s per column for some columns)

8x3: seemingly no problems for menu=2 or 3

**Good god I finally can run 10x10:** New vectorized solver took about 18s (w/ all params random, menu size 2). Tok 17s with menu size 1. For 20x20 menu size 1, took 84s, for menu size 2, 91s. For menu size 1, 30x15, it took 143s and 134 for menu 5, compared to the average of 291s and 305s in Shahab’s paper

Originally just multiplying the matrices by the variables returned NaN, so I tried slicing them up into blocks for each set of constraints. This seems to work. After getting 10x10 to work I tried the original simpler matrix multiplication again and it doesn’t return NaN, but returns all zeros which shouldn’t be feasible.

There doesn’t appear to be the issue of non-integer y when U has only a couple different entries

**For Meeting 11/30**

**Bilevel Interdiction:**

Bumping up the rejection matrix values, the solution was no longer integer even for 3x3

Even changing all of the utility values to 1 didn’t help the algorithm process 4x4

**Single level interdiction:**

If you don’t include any supplier utility in the objective, the problem becomes a simple assignment problem matching pairs of drivers to tasks. So if C, D, and r are all ones, the objective value is

(m-(floor[n/2])).

For the same C, D, and r and U~unif(-0.5,0.5): if n is even, the objective will usually be the same as when utility isn’t included. But depending on the utility weight (and if some tasks are extra undesirable to drivers) the interdiction problem can give more interesting results like three or more drivers accepting the same task.

One of the setbacks for this formulation (that could be adjusted after solving) is that because the utility is deterministic, the system knows what the suppliers will pick so the ‘leftover’ menu option(s) are often the default slot of request 1.

**For Meeting 12/7**

**Gurobi Seminar Note:**

* Benders can be good though not super easy for large stochastic two-stage problems
* To pick number of samples: use as many as possible, use the boundary techniques to get estimate of quality
* Column generation can be good for huge number of scenarios

**Summary of Xing Wang’s Thesis:**

She assumes menu size 2 without the no-choice option. The main difference from what we’ve talked about is that instead of looking at an overall rank for each driver, she assumes for any menu of two items offered to any driver, the probability eta they pick the one the platform prefers is a fixed number. However, I believe setting eta=0.5 is equivalent to our probability assumption that all scenarios are equally likely and menu size = 2. Because of their assumption the problem can be captured as a single-level MIP. In addition, they only consider platform utility, not driver rejection costs and request rejection costs.

They do both exhaustive scenarios and SAA. For exhaustive scenarios, they do mxm for m=3,…,7. The platform utilities are randomly generated and they do 50 trials. The average performance is compared with the best case scenario. Something I found surprising was that different values for the scenario probability significantly changed the runtime.

For SAA, she ran the program with 5, 10, or 15 samples. She would do this 20 times and pick the best outcome. Choosing the ‘best’ outcome appears to be based on the exhaustive case expected value. It was run for 6x6 and the ratio to the best case scenario seems about the same as the exhaustive scenario method. Taking more samples per trial seemed to only slightly improve performance.

They also explore a partially pre-determined menu generation, where one of the options for each driver is the request they would accept in the best case scenario plus a second option with lower utility (since they assume the probability of picking the platform’s preference is >.5) If there are no options with a lower utility, they provide two strategies for picking the next best option, but the wording is exactly the same for the two strategies…Whatever they ended up doing it seemed to perform well and can be constructed very quickly.

**How should I decide what U to use when solving?**

6x6 x for basic:

x =

0 1 0 0 1 1

1 0 0 1 1 1

0 0 1 1 0 0

0 0 1 0 0 0

1 0 0 0 0 0

0 1 0 0 0 0

6x6 for single level interdiction:

0 0 0 0 0 0

0 1 1 0 0 0

0 0 0 0 1 1

0 0 0 1 1 1

1 0 1 0 0 0

1 1 0 1 0 0

For random x:

0 0 0 0 0 0

1 0 0 0 0 1

0 1 1 1 0 1

0 0 0 1 1 0

0 1 0 0 1 0

1 0 1 0 0 0

**For Meeting 12/14**

**Numerical Results:**

4x4 menu 2, 3 trials, 3 sampls:

x =

1 1 0 0

1 0 1 1

0 1 0 1

0 0 1 0

x =

1 0 0 1

0 1 0 0

0 1 1 1

1 0 1 0

x =

0 1 1 1

1 0 1 0

1 0 0 1

0 1 0 0

objave = -0.1605

objsd = 1.6514

dupave = 0.4979

dupsd = 0.6007

rejave = 1.8313

rejsd = 0.7435

x =

0 0 0 1 0 0

1 0 0 0 1 0

1 0 0 0 0 1

0 1 1 0 0 1

0 1 1 0 0 0

0 0 0 1 1 0

x =

0 1 0 0 1 0

1 0 1 0 0 0

0 0 0 1 1 0

1 0 0 0 0 1

0 0 0 1 0 0

0 1 1 0 0 1

x =

0 1 0 0 1 -0

0 1 1 0 0 0

1 0 1 1 0 1

0 0 0 1 1 0

1 0 0 0 0 0

0 0 0 0 0 1

x =

0 1 1 0 0 0

1 0 1 0 0 -0

0 0 0 1 1 1

0 1 0 0 1 0

0 0 0 0 0 1

1 0 0 1 0 0

x =

1 1 0 0 0 0

0 1 0 0 1 0

0 0 1 1 0 0

0 0 1 1 0 0

0 0 0 0 1 1

1 0 0 0 0 1

x =

0 0 0 0 -0 1

1 1 -0 0 0 -0

0 0 0 1 0 0

0 0 1 0 1 -0

0 0 1 1 1 -0

1 1 -0 0 -0 1

x =

0 1 0 0 0 1

0 1 1 0 0 0

1 0 0 0 0 1

0 0 1 0 1 0

0 0 0 1 1 0

1 0 0 1 0 0

x =

0 1 0 1 0 0

0 0 0 0 1 1

0 0 0 0 1 1

1 0 1 0 0 0

1 0 0 1 0 0

0 1 1 0 0 0

Linear matrix variable 6x60 (full, real, 360 variables)

Linear scalar (real, 120 variables)

Linear scalar (real, 240 variables)

Linear scalar (real, 360 variables)

Linear scalar (real, 480 variables)

Linear scalar (real, 600 variables)

Linear scalar (real, 720 variables)

x =

0 0 0 0 0 1

1 1 0 0 0 1

1 0 1 0 0 0

0 0 0 1 1 0

0 0 1 1 0 0

0 1 0 0 1 0

Linear matrix variable 6x60 (full, real, 360 variables)

Linear scalar (real, 120 variables)

Linear scalar (real, 240 variables)

Linear scalar (real, 360 variables)

Linear scalar (real, 480 variables)

Linear scalar (real, 600 variables)

Linear scalar (real, 720 variables)

x =

1 -0 0 0 0 0

1 0 1 1 0 0

0 0 0 1 1 0

-0 1 0 0 0 1

0 0 1 -0 0 1

0 1 0 0 1 0

>> main

Linear matrix variable 3x50 (full, real, 150 variables)

Linear scalar (real, 100 variables)

Linear scalar (real, 200 variables)

Linear scalar (real, 300 variables)

x =

1 1 1 1 1

1 0 0 1 1

0 1 1 0 0

Linear matrix variable 3x50 (full, real, 150 variables)

Linear scalar (real, 100 variables)

Linear scalar (real, 200 variables)

Linear scalar (real, 300 variables)

x =

1 0 1 1 1

0 1 0 0 1

1 1 1 1 0

Linear matrix variable 3x50 (full, real, 150 variables)

Linear scalar (real, 100 variables)

Linear scalar (real, 200 variables)

Linear scalar (real, 300 variables)

x =

1 1 1 0 1

0 1 1 1 0

1 0 0 1 1

Linear matrix variable 3x50 (full, real, 150 variables)

Linear scalar (real, 100 variables)

Linear scalar (real, 200 variables)

Linear scalar (real, 300 variables)

x =

1 1 1 1 0

1 0 1 1 1

0 1 0 0 1

Linear matrix variable 3x50 (full, real, 150 variables)

Linear scalar (real, 100 variables)

Linear scalar (real, 200 variables)

Linear scalar (real, 300 variables)

x =

1 0 1 0 0

1 1 1 1 1

0 1 0 1 1

Linear matrix variable 3x25 (full, real, 75 variables)

Linear scalar (real, 50 variables)

Linear scalar (real, 100 variables)

Linear scalar (real, 150 variables)

x =

0 0 0 1 1

1 1 1 1 0

1 1 1 0 1

Linear matrix variable 3x25 (full, real, 75 variables)

Linear scalar (real, 50 variables)

Linear scalar (real, 100 variables)

Linear scalar (real, 150 variables)

x =

1 1 1 1 0

1 0 0 1 1

0 1 1 0 1

Linear matrix variable 3x25 (full, real, 75 variables)

Linear scalar (real, 50 variables)

Linear scalar (real, 100 variables)

Linear scalar (real, 150 variables)

x =

1 0 1 1 1

0 1 1 1 0

1 1 0 0 1

Linear matrix variable 3x25 (full, real, 75 variables)

Linear scalar (real, 50 variables)

Linear scalar (real, 100 variables)

Linear scalar (real, 150 variables)

x =

0 1 0 0 1

1 1 1 1 0

1 0 1 1 1

Linear matrix variable 3x25 (full, real, 75 variables)

Linear scalar (real, 50 variables)

Linear scalar (real, 100 variables)

Linear scalar (real, 150 variables)

x =

1 1 0 1 0

1 0 1 0 1

0 1 1 1 1

Linear matrix variable 5x30 (full, real, 150 variables)

Linear scalar (real, 60 variables)

Linear scalar (real, 120 variables)

Linear scalar (real, 180 variables)

Linear scalar (real, 240 variables)

Linear scalar (real, 300 variables)

x =

1 0 1

1 0 0

0 1 0

0 0 1

0 1 0

Linear matrix variable 5x30 (full, real, 150 variables)

Linear scalar (real, 60 variables)

Linear scalar (real, 120 variables)

Linear scalar (real, 180 variables)

Linear scalar (real, 240 variables)

Linear scalar (real, 300 variables)

x =

1 1 0

1 0 0

0 0 1

0 1 0

0 0 1

Linear matrix variable 5x30 (full, real, 150 variables)

Linear scalar (real, 60 variables)

Linear scalar (real, 120 variables)

Linear scalar (real, 180 variables)

Linear scalar (real, 240 variables)

Linear scalar (real, 300 variables)

x =

0 1 0

1 0 1

0 0 1

0 1 0

1 0 0

Linear matrix variable 5x30 (full, real, 150 variables)

Linear scalar (real, 60 variables)

Linear scalar (real, 120 variables)

Linear scalar (real, 180 variables)

Linear scalar (real, 240 variables)

Linear scalar (real, 300 variables)

x =

0 1 0

0 1 0

1 0 1

0 0 1

1 0 0

Linear matrix variable 5x30 (full, real, 150 variables)

Linear scalar (real, 60 variables)

Linear scalar (real, 120 variables)

Linear scalar (real, 180 variables)

Linear scalar (real, 240 variables)

Linear scalar (real, 300 variables)

x =

0 0 1

1 1 0

0 0 1

1 0 0

0 1 0

objave =

-1.3778

objsd =

1.6942

dupave =

0.1259

dupsd =

0.3370

rejave =

3.1259

rejsd =

0.8182

Linear matrix variable 5x15 (full, real, 75 variables)

Linear scalar (real, 30 variables)

Linear scalar (real, 60 variables)

Linear scalar (real, 90 variables)

Linear scalar (real, 120 variables)

Linear scalar (real, 150 variables)

x =

1 0 0

0 1 0

1 0 1

0 0 1

0 1 0

Linear matrix variable 5x15 (full, real, 75 variables)

Linear scalar (real, 30 variables)

Linear scalar (real, 60 variables)

Linear scalar (real, 90 variables)

Linear scalar (real, 120 variables)

Linear scalar (real, 150 variables)

x =

1 0 1

0 1 0

1 0 0

0 0 1

0 1 0

Linear matrix variable 5x15 (full, real, 75 variables)

Linear scalar (real, 30 variables)

Linear scalar (real, 60 variables)

Linear scalar (real, 90 variables)

Linear scalar (real, 120 variables)

Linear scalar (real, 150 variables)

x =

1 0 0

0 1 1

0 0 0

0 0 1

1 1 0

Linear matrix variable 5x15 (full, real, 75 variables)

Linear scalar (real, 30 variables)

Linear scalar (real, 60 variables)

Linear scalar (real, 90 variables)

Linear scalar (real, 120 variables)

Linear scalar (real, 150 variables)

x =

0 0 1

1 0 1

0 1 0

1 1 0

0 0 0

Linear matrix variable 5x15 (full, real, 75 variables)

Linear scalar (real, 30 variables)

Linear scalar (real, 60 variables)

Linear scalar (real, 90 variables)

Linear scalar (real, 120 variables)

Linear scalar (real, 150 variables)

x =

0 1 0

1 0 1

0 1 0

1 0 1

0 0 0

For theta =2 6x6 30 samples

x =

0 0 0 0 1 0

0 0 1 1 0 0

0 1 0 1 0 0

1 0 0 0 1 1

0 1 0 0 0 1

1 0 1 0 0 0

x =

0 1 0 1 0 0

1 0 1 0 1 0

0 0 1 1 0 0

0 1 0 0 0 1

1 0 0 0 1 0

0 0 0 0 0 1

x =

0 1 0 1 0 0

0 1 1 -0 0 0

0 0 1 0 1 0

0 0 0 1 1 0

1 0 0 0 0 1

1 0 0 0 0 1

x =

-0 0 1 -0 0 1

0 0 1 1 0 0

1 0 0 0 -0 1

-0 1 0 1 1 0

1 1 0 -0 0 0

0 0 -0 0 1 -0

x =

1 1 0 1 0 0

0 0 1 0 1 0

0 0 1 0 0 1

0 0 0 1 0 0

0 0 0 0 1 1

1 1 0 0 0 0

x =

0 0 1 0 1 0

1 0 1 0 -0 0

0 1 0 1 0 0

0 0 0 0 1 1

1 0 0 1 0 -0

0 1 -0 0 0 1

AFTER CORRECTIONS

x =

0 1.0000 0 0 1.0000 0

1.0000 0 1.0000 0 0 0.0000

0 1.0000 0 1.0000 1.0000 1.0000

0 0 0 1.0000 0.0000 0

0 0 0 0 0 1.0000

1.0000 0 1.0000 0 0 0

Linear matrix variable 6x30 (full, real, 180 variables)

Linear scalar (real, 60 variables)

Linear scalar (real, 120 variables)

Linear scalar (real, 180 variables)

Linear scalar (real, 240 variables)

Linear scalar (real, 300 variables)

Linear scalar (real, 360 variables)

x =

0 0 1.0000 0 0 0

0 1.0000 0 0 0 0

1.0000 0 0 0 0 1.0000

1.0000 0 0 0.0000 0 1.0000

0 0 0 1.0000 1.0000 0

0 1.0000 1.0000 1.0000 1.0000 0

Linear matrix variable 6x30 (full, real, 180 variables)

Linear scalar (real, 60 variables)

Linear scalar (real, 120 variables)

Linear scalar (real, 180 variables)

Linear scalar (real, 240 variables)

Linear scalar (real, 300 variables)

Linear scalar (real, 360 variables)

x =

0 1.0000 1.0000 0 0 0

-0.0000 1.0000 0 1.0000 0 0

0 0 0 0 1.0000 0

-0.0000 0 0 0 1.0000 1.0000

1.0000 0 1.0000 0 0 1.0000

1.0000 0 0 1.0000 0 0

Linear matrix variable 6x30 (full, real, 180 variables)

Linear scalar (real, 60 variables)

Linear scalar (real, 120 variables)

Linear scalar (real, 180 variables)

Linear scalar (real, 240 variables)

Linear scalar (real, 300 variables)

Linear scalar (real, 360 variables)

x =

0 0 1.0000 0 0 1.0000

0 0 0 1.0000 1.0000 0

0 0 0 1.0000 0 0

1.0000 1.0000 0 0 1.0000 0

0 0 1.0000 0 0 1.0000

1.0000 1.0000 0 0 0 0

Linear matrix variable 6x30 (full, real, 180 variables)

Linear scalar (real, 60 variables)

Linear scalar (real, 120 variables)

Linear scalar (real, 180 variables)

Linear scalar (real, 240 variables)

Linear scalar (real, 300 variables)

Linear scalar (real, 360 variables)

x =

0 1.0000 1.0000 0 1.0000 0

0 1.0000 0 0 0 0

0 0 1.0000 1.0000 0 0

1.0000 0 0 0 0 1.0000

1.0000 0 0 1.0000 0 0

0 0 0 0 1.0000 1.0000

Linear matrix variable 6x30 (full, real, 180 variables)

Linear scalar (real, 60 variables)

Linear scalar (real, 120 variables)

Linear scalar (real, 180 variables)

Linear scalar (real, 240 variables)

Linear scalar (real, 300 variables)

Linear scalar (real, 360 variables)

x =

0 0 0 0 1 1

0 1 1 0 0 0

1 0 0 1 0 0

0 1 0 0 0 1

1 0 1 1 0 0

0 0 0 0 1 0

Linear matrix variable 6x30 (full, real, 180 variables)

Linear scalar (real, 60 variables)

Linear scalar (real, 120 variables)

Linear scalar (real, 180 variables)

Linear scalar (real, 240 variables)

Linear scalar (real, 300 variables)

Linear scalar (real, 360 variables)

x =

0 0 0 1 0 0

0 0 1 0 0 1

0 0 0 0 1 1

1 0 1 0 1 0

1 1 0 0 0 0

0 1 0 1 0 0

Linear matrix variable 6x30 (full, real, 180 variables)

Linear scalar (real, 60 variables)

Linear scalar (real, 120 variables)

Linear scalar (real, 180 variables)

Linear scalar (real, 240 variables)

Linear scalar (real, 300 variables)

Linear scalar (real, 360 variables)

x =

0 0 1.0000 0 0 0

0 0 0 1.0000 0 1.0000

1.0000 1.0000 1.0000 0 0 1.0000

0 1.0000 0 0 1.0000 0

0 0 0 1.0000 0 0

1.0000 0 0 0 1.0000 0

Linear matrix variable 6x30 (full, real, 180 variables)

Linear scalar (real, 60 variables)

Linear scalar (real, 120 variables)

Linear scalar (real, 180 variables)

Linear scalar (real, 240 variables)

Linear scalar (real, 300 variables)

Linear scalar (real, 360 variables)

x =

1 1 0 0 0 0

0 0 1 0 1 1

0 1 0 0 0 0

0 0 0 1 0 1

1 0 1 1 0 0

0 0 0 0 1 0

Linear matrix variable 6x30 (full, real, 180 variables)

Linear scalar (real, 60 variables)

Linear scalar (real, 120 variables)

Linear scalar (real, 180 variables)

Linear scalar (real, 240 variables)

Linear scalar (real, 300 variables)

Linear scalar (real, 360 variables)

x =

1 1 0 1 0 0

0 0 0 0 1 1

1 0 0 1 1 0

0 0 1 0 0 0

0 1 1 0 0 0

0 0 0 0 0 1

Linear matrix variable 6x60 (full, real, 360 variables)

Linear scalar (real, 120 variables)

Linear scalar (real, 240 variables)

Linear scalar (real, 360 variables)

Linear scalar (real, 480 variables)

Linear scalar (real, 600 variables)

Linear scalar (real, 720 variables)

x =

0 0 0 0 1 1

1 0 1 1 0 0

0 1 0 0 0 0

0 1 0 1 0 0

0 0 0 0 1 1

1 0 1 0 0 0

Linear matrix variable 6x60 (full, real, 360 variables)

Linear scalar (real, 120 variables)

Linear scalar (real, 240 variables)

Linear scalar (real, 360 variables)

Linear scalar (real, 480 variables)

Linear scalar (real, 600 variables)

Linear scalar (real, 720 variables)

x =

1.0000 -0.0000 1.0000 0 1.0000 0

0 1.0000 0 0 0 1.0000

0 0 1.0000 0 1.0000 0

0 1.0000 0 1.0000 0 0

0 0.0000 0 1.0000 0 1.0000

1.0000 0 0 0 0 0

Linear matrix variable 6x60 (full, real, 360 variables)

Linear scalar (real, 120 variables)

Linear scalar (real, 240 variables)

Linear scalar (real, 360 variables)

Linear scalar (real, 480 variables)

Linear scalar (real, 600 variables)

Linear scalar (real, 720 variables)

x =

0 0 1 0 0 0

0 1 0 0 1 0

1 0 0 0 0 0

0 0 0 1 0 1

1 1 1 0 0 0

0 0 0 1 1 1

Linear matrix variable 6x60 (full, real, 360 variables)

Linear scalar (real, 120 variables)

Linear scalar (real, 240 variables)

Linear scalar (real, 360 variables)

Linear scalar (real, 480 variables)

Linear scalar (real, 600 variables)

Linear scalar (real, 720 variables)

x =

1.0000 1.0000 0 0 0 1.0000

0 1.0000 -0.0000 1.0000 0 0

1.0000 0 -0.0000 0 0 1.0000

-0.0000 -0.0000 0 1.0000 1.0000 0

0 0 1.0000 0 1.0000 0

0 -0.0000 1.0000 0 0 -0.0000

Linear matrix variable 6x60 (full, real, 360 variables)

Linear scalar (real, 120 variables)

Linear scalar (real, 240 variables)

Linear scalar (real, 360 variables)

Linear scalar (real, 480 variables)

Linear scalar (real, 600 variables)

Linear scalar (real, 720 variables)

x =

0 0 0 0 0 1.0000

0 0 0 1.0000 1.0000 0

1.0000 0.0000 0 0 -0.0000 0

0.0000 1.0000 0 1.0000 0 0.0000

0 1.0000 1.0000 0 0 1.0000

1.0000 0 1.0000 0 1.0000 0.0000

Linear matrix variable 6x60 (full, real, 360 variables)

Linear scalar (real, 120 variables)

Linear scalar (real, 240 variables)

Linear scalar (real, 360 variables)

Linear scalar (real, 480 variables)

Linear scalar (real, 600 variables)

Linear scalar (real, 720 variables)

x =

0 1 1 0 0 0

0 0 0 1 0 1

0 0 0 1 1 0

1 0 1 0 0 0

0 1 0 0 1 0

1 0 0 0 0 1

Linear matrix variable 6x60 (full, real, 360 variables)

Linear scalar (real, 120 variables)

Linear scalar (real, 240 variables)

Linear scalar (real, 360 variables)

Linear scalar (real, 480 variables)

Linear scalar (real, 600 variables)

Linear scalar (real, 720 variables)

x =

1 0 1 0 0 0

0 1 0 1 0 0

0 1 0 0 0 1

0 0 0 0 1 0

1 0 0 0 1 0

0 0 1 1 0 1

Linear matrix variable 6x60 (full, real, 360 variables)

Linear scalar (real, 120 variables)

Linear scalar (real, 240 variables)

Linear scalar (real, 360 variables)

Linear scalar (real, 480 variables)

Linear scalar (real, 600 variables)

Linear scalar (real, 720 variables)

x =

0 1 1 1 0 0

0 0 0 0 1 1

0 0 0 1 0 0

0 0 0 0 1 1

1 0 1 0 0 0

1 1 0 0 0 0

Linear matrix variable 6x60 (full, real, 360 variables)

Linear scalar (real, 120 variables)

Linear scalar (real, 240 variables)

Linear scalar (real, 360 variables)

Linear scalar (real, 480 variables)

Linear scalar (real, 600 variables)

Linear scalar (real, 720 variables)

x =

0 0 0 1.0000 1.0000 0

0 1.0000 0 0 0 0

0 0 1.0000 0 0 0

1.0000 0 1.0000 1.0000 0 0

0 1.0000 0 0 1.0000 1.0000

1.0000 0 0.0000 0 0 1.0000

Linear matrix variable 6x60 (full, real, 360 variables)

Linear scalar (real, 120 variables)

Linear scalar (real, 240 variables)

Linear scalar (real, 360 variables)

Linear scalar (real, 480 variables)

Linear scalar (real, 600 variables)

Linear scalar (real, 720 variables)

x =

0 1 0 0 0 1

0 1 0 0 1 0

0 0 0 1 0 1

0 0 1 0 1 0

1 0 0 1 0 0

1 0 1 0 0 0

objave =

-0.2366

objsd =

2.0852

dupave =

0.7455

dupsd =

0.7296

rejave =

2.7455

rejsd =

0.9485

Linear matrix variable 6x120 (full, real, 720 variables)

Linear scalar (real, 240 variables)

Linear scalar (real, 480 variables)

Linear scalar (real, 720 variables)

Linear scalar (real, 960 variables)

Linear scalar (real, 1200 variables)

Linear scalar (real, 1440 variables)

x =

0 0 0 1 1 0

1 1 0 0 0 1

0 0 1 1 0 0

0 1 0 0 0 1

0 0 0 0 1 0

1 0 1 0 0 0

Linear matrix variable 6x120 (full, real, 720 variables)

Linear scalar (real, 240 variables)

Linear scalar (real, 480 variables)

Linear scalar (real, 720 variables)

Linear scalar (real, 960 variables)

Linear scalar (real, 1200 variables)

Linear scalar (real, 1440 variables)

x =

1 0 0 1 0 0

0 0 0 0 1 1

0 0 0 0 1 1

1 0 0 1 0 0

0 1 1 0 0 0

0 1 1 0 0 0

Linear matrix variable 6x120 (full, real, 720 variables)

Linear scalar (real, 240 variables)

Linear scalar (real, 480 variables)

Linear scalar (real, 720 variables)

Linear scalar (real, 960 variables)

Linear scalar (real, 1200 variables)

Linear scalar (real, 1440 variables)

x =

1 0 1 0 0 0

0 1 0 1 0 0

0 0 1 0 0 1

1 0 0 0 1 0

0 1 0 0 1 0

0 0 0 1 0 1

Linear matrix variable 6x120 (full, real, 720 variables)

Linear scalar (real, 240 variables)

Linear scalar (real, 480 variables)

Linear scalar (real, 720 variables)

Linear scalar (real, 960 variables)

Linear scalar (real, 1200 variables)

Linear scalar (real, 1440 variables)

x =

0 0 1 0 1 0

1 1 0 0 0 0

0 0 1 1 0 0

1 0 0 0 0 1

0 0 0 0 1 1

0 1 0 1 0 0

Linear matrix variable 6x120 (full, real, 720 variables)

Linear scalar (real, 240 variables)

Linear scalar (real, 480 variables)

Linear scalar (real, 720 variables)

Linear scalar (real, 960 variables)

Linear scalar (real, 1200 variables)

Linear scalar (real, 1440 variables)

x =

0 0 0 0 1 1

0 1 1 0 0 0

0 0 0 1 0 1

0 1 1 0 0 0

1 0 0 0 1 0

1 0 0 1 0 0

Linear matrix variable 6x120 (full, real, 720 variables)

Linear scalar (real, 240 variables)

Linear scalar (real, 480 variables)

Linear scalar (real, 720 variables)

Linear scalar (real, 960 variables)

Linear scalar (real, 1200 variables)

Linear scalar (real, 1440 variables)

x =

0 0 0 0 0 1.0000

0 0 0 1.0000 1.0000 -0.0000

0 1.0000 1.0000 0 0 0

1.0000 0 0 1.0000 1.0000 0

1.0000 0 0 -0.0000 0 1.0000

0 1.0000 1.0000 0 0 0

Linear matrix variable 6x120 (full, real, 720 variables)

Linear scalar (real, 240 variables)

Linear scalar (real, 480 variables)

Linear scalar (real, 720 variables)

Linear scalar (real, 960 variables)

Linear scalar (real, 1200 variables)

Linear scalar (real, 1440 variables)

x =

1.0000 1.0000 0 0 0 0

1.0000 1.0000 1.0000 0 0 0

0 -0.0000 0 0 1.0000 0

0 0.0000 0 1.0000 0 1.0000

0 0 0 1.0000 0 1.0000

0.0000 0 1.0000 0 1.0000 0

Linear matrix variable 6x120 (full, real, 720 variables)

Linear scalar (real, 240 variables)

Linear scalar (real, 480 variables)

Linear scalar (real, 720 variables)

Linear scalar (real, 960 variables)

Linear scalar (real, 1200 variables)

Linear scalar (real, 1440 variables)

x =

0 1 0 0 0 0

0 0 0 1 0 0

0 0 1 0 0 1

0 0 1 0 1 0

1 0 0 0 1 1

1 1 0 1 0 0

Linear matrix variable 6x120 (full, real, 720 variables)

Linear scalar (real, 240 variables)

Linear scalar (real, 480 variables)

Linear scalar (real, 720 variables)

Linear scalar (real, 960 variables)

Linear scalar (real, 1200 variables)

Linear scalar (real, 1440 variables)

x =

1.0000 0 0 0 0 1.0000

0 1.0000 0 0 0 0

0 1.0000 0 0 1.0000 0

1.0000 0 0 0 1.0000 1.0000

0 0 1.0000 1.0000 0 0.0000

0 0 1.0000 1.0000 0 0

Linear matrix variable 6x120 (full, real, 720 variables)

Linear scalar (real, 240 variables)

Linear scalar (real, 480 variables)

Linear scalar (real, 720 variables)

Linear scalar (real, 960 variables)

Linear scalar (real, 1200 variables)

Linear scalar (real, 1440 variables)

x =

0 0 0 1 0 1

0 0 0 1 0 1

1 0 1 0 0 0

0 1 0 0 1 0

1 0 1 0 0 0

0 1 0 0 1 0

objave =

-0.1852

objsd =

2.1097

dupave =

0.7284

dupsd =

0.7181

rejave =

2.7284

rejsd =

0.9917

**For Meeting 3.27.2019**

**Independent y probabilities, C random, theta=3, two experiments:**

* In each experiment, the y probabilities and C values are the same so performance between trials and number of samples can be directly compared
* For all trials with 1000 samples (and most of 100 samples, some of 50 samples), converges to the same menu. For both experiments, this menu has menu size 1 for all drivers. However, both experiments had one or more trials where the menu found was the converged menu plus a couple more menu options and it nontrivially improved performance—**why do the trials converge to a non-optimal menu?**
* Even the 1 and 5 sample menus often had a few common menu options with the converged menu, and beyond the converged menu options there were still a lot of overlapping options between menus.
* As sample size increases the simulated and exhaustive performance evens out between trials, but there is always a significant gap between the simulated objective and the exhaustive objective
* Increasing sample size improves exhaustive performance but only to an extent, since the higher sample trials converged to a menu that didn’t perform the best out of the trials

**For Meeting 4.9.2019**

**Same experiments as last entry**

It was noted that both trials converged to a menu with theta=1 all drivers. It turns out this menu is the menu that **maximizes the expected number of fulfilled assignments** when each driver is only offered one request.

* This could potentially change if the c\_ij are different enough that some task(s) become much more/less important than others

With different popularities (overall yprobs were also reduced):

* with reduced yprobs and popular/unpopular items. The base yprob value was 0.1+0.5\*rand. The first two requests are ‘popular’ so yprobs are multiplied by 1.5. Last two requests are ‘unpopular’ so multiplied by 0.75
* As expected, the popular items are offered only once, regular items are offered a few times, and unpopular items are offered to almost everyone
* This menu did not converge at 1000 samples, and the menus have more than one option for most suppliers. It doesn’t even converge at 5000 samples
* However for some reason the performance analysis is taking forever for these menus