Facility Location with BDDs: Status update.

A. Bochkarev

1 Status

- I have implemented some of the key machinery, so now I can play with it: weighted BDD transformations, building a simple MIP, creating BDDs (availability, covering, and their intersection), building CPP MIP and a network flow based on the intersection BDD.
- I ran several really simple examples, and stumbled upon a problem: I think my intersection BDD (unsurprisingly) blows up. It is a little more dramatic than I expected: I'd need to find a way to make instances large enough to be difficult for MIP, but still tractable by BDDs (see the last section here).

2 A toy example (updated, again)

This is a simple example updated in the spirit of our recent discussions.

2.1 Problem description

Let us consider a simple problem with two facilities and three customers, as depicted in Figure 1.

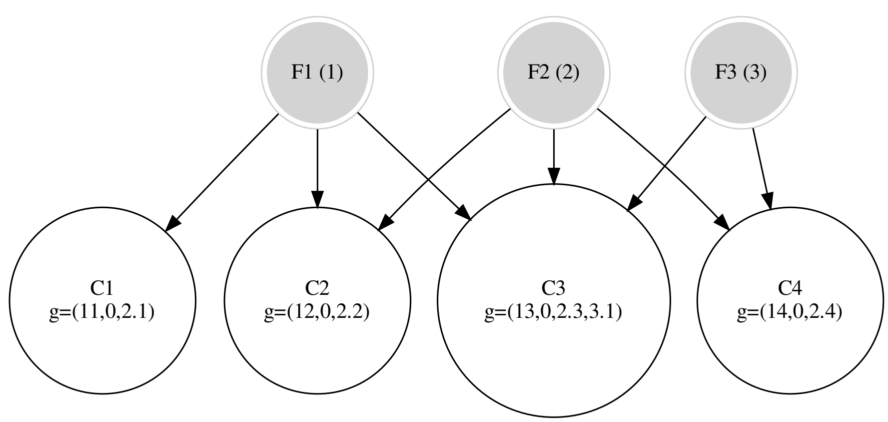


Figure 1: Problem description: facilty location with overlaps.
Facilities: numbers in parentheses indicate locating ("turn-on") costs.
Consumers: overlap penalies are shown, $g = (0, 1, 2)$ would mean that for this consumer zero overlapping coverings imposed no additional cost, covering with one facility brought additional cost 1, with two facilities (i.e., actual overlap) brought cost 2.

I am representing this (or any other) problem in the following three ways.

2.2 Simple MIP

I generate a "naive" MIP right away:

$$\text{Minimize:} 50.0 + x_1 + 2.0x_2 + 3.0x_3 - 11.0v_1^1 - 12.0v_2^1 + 2.2v_2^2 - 13.0v_3^1 + 2.3v_3^2 + 0.8v_3^3 - 14.0v_4^1 + 2.4v_4^2$$

Subject To:

$$\begin{aligned}
&- 1.0x_1 + z_{1\to 1} = 0.0\\
&- 1.0x_1 + z_{1\to 2} = 0.0\\
&- 1.0x_1 + z_{1\to 3} = 0.0\\
&- 1.0x_2 + z_{2\to 2} = 0.0\\
&- 1.0x_2 + z_{2\to 3} = 0.0\\
&- 1.0x_2 + z_{2\to 4} = 0.0\\
&- 1.0x_3 + z_{3\to 3} = 0.0\\
&- 1.0x_3 + z_{3\to 4} = 0.0\\
&- 1.0z_{1\to 1} + b_1 = 0.0\\
&- 1.0z_{1\to 2} - 1.0z_{2\to 2} + b_2 = 0.0\\
&- 1.0z_{1\to 3} - 1.0z_{2\to 3} - 1.0z_{3\to 3} + b_3 = 0.0\\
&- 1.0z_{2\to 4} - 1.0z_{3\to 4} + b_4 = 0.0\\
&b_1 - 1.0v_1^1 = 0.0\\
&- 1.0v_2^1 + v_2^2 \leq 0.0\\
&b_2 - 1.0v_2^1 - 1.0v_2^2 = 0.0\\
&- 1.0v_3^1 + v_3^2 \leq 0.0\\
&- 1.0v_3^2 + v_3^3 \leq 0.0\\
&b_3 - 1.0v_3^1 - 1.0v_3^2 - 1.0v_3^3 = 0.0\\
&- 1.0v_4^1 + v_4^2 \leq 0.0\\
&b_4 - 1.0v_4^1 - 1.0v_4^2 = 0.0,
\end{aligned}$$

where x are "locate" decisions, z are covering decisions (which are kind of dependent on each other, like we discussed), b are numbers of overlaps, and v are used to encode an arbitrary overlap penalty function v_j^k corresponds to customer j being "overlapped" k times.

Binary variables are: $x_1, z_{1\to 1}, z_{1\to 2}, z_{1\to 3}, x_2, z_{2\to 2}, z_{2\to 3}, z_{2\to 4}, x_3, z_{3\to 3}, z_{3\to 4}, v_1^1, v_2^1, v_2^2, v_3^1, v_3^2, v_3^3, v_4^1, v_4^2$ (and I kind of hope that Gurobi's `presolve` takes care of the redundant variables.)

2.3 CPP MIP

Now, I am generating two decision diagrams, as before:

Diagram	**Constraints incorporated**	**Costs incorporated**	**Figure**
Covering	(Costs for) covering each consumer – nothing hard.	$g_j(n_j)$ (overlap)	2a
Availability	"Turn on" and "covering" variables are consistent	f_i (location)	2b

Which allows me to formulate the following MIP: The objective is:

$$\text{Minimize:} \quad v^A_{0\to 1,h} + 2.0v^A_{6\to 8,h} + 3.0v^A_{14\to 16,h} + 11.0v^C_{0\to 1,l} + 2.2v^C_{3\to 4,h} + 12.0v^C_{2\to 4,l} + 3.1v^C_{9\to 10,h} + 2.3v^C_{9\to 10,l} + 2.3v^C_{8\to 10,h} + 13.0v^C_{7\to 10,l} + 2.4v^C_{12\to T,h} + 14.0v^C_{11\to T,l}.$$

Here, for example, variable $v^A_{0\to 1,h}$ corresponds to the flow from node 0 to node 1 of diagram A (availability), along the "hi" ("yes") arc.

Legend.

- From each diagram, two types of constraints are generated:
  - *cont-at-*$\cdot$ are flow continuity constraints at a given node.
  - *bin-link-*<k> are binary linking constraints (needed to link two BDDs – i.e., tangle network flow problems), one per layer, indexed with k.
- A denotes "Availability" diagram, C denotes "Covering" diagram.

All node numbers correspond to the diagrams and have nothing to do with customer and facility indices.

- Arc flow variables, continuous, v (sorry, these v have nothing to do with v^k_j from the previous section).
- Linking variables (binary): $\lambda_{z1-1}, \lambda_{z1-2}, \lambda_{z1-3}, \lambda_{z2-2}, \lambda_{z2-3}, \lambda_{z2-4}, \lambda_{z3-3}, \lambda_{z3-4}$.

Constraints from **covering BDD**:

Type	**Constraint**
cont-at-0	$-1.0v^C_{0\to 1,h} - 1.0v^C_{0\to 1,l} = -1.0$
bin-link-1	$\lambda_{z1-1} - 1.0v^C_{0\to 1,h} = 0.0$
cont-at-1	$v^C_{0\to 1,h} + v^C_{0\to 1,l} - 1.0v^C_{1\to 3,h} - 1.0v^C_{1\to 2,l} = 0.0$
bin-link-2	$\lambda_{z1-2} - 1.0v^C_{1\to 3,h} = 0.0$
cont-at-3	$v^C_{1\to 3,h} - 1.0v^C_{3\to 4,h} - 1.0v^C_{3\to 4,l} = 0.0$
cont-at-2	$v^C_{1\to 2,l} - 1.0v^C_{2\to 4,h} - 1.0v^C_{2\to 4,l} = 0.0$
bin-link-3	$\lambda_{z2-2} - 1.0v^C_{3\to 4,h} - 1.0v^C_{2\to 4,h} = 0.0$
cont-at-4	$v^C_{3\to 4,h} + v^C_{3\to 4,l} + v^C_{2\to 4,h} + v^C_{2\to 4,l} - 1.0v^C_{4\to 6,h} - 1.0v^C_{4\to 5,l} = 0.0$
bin-link-4	$\lambda_{z1-3} - 1.0v^C_{4\to 6,h} = 0.0$
cont-at-5	$v^C_{4\to 5,l} - 1.0v^C_{5\to 8,h} - 1.0v^C_{5\to 7,l} = 0.0$
cont-at-6	$v^C_{4\to 6,h} - 1.0v^C_{6\to 9,h} - 1.0v^C_{6\to 8,l} = 0.0$
bin-link-5	$\lambda_{z2-3} - 1.0v^C_{5\to 8,h} - 1.0v^C_{6\to 9,h} = 0.0$
cont-at-9	$v^C_{6\to 9,h} - 1.0v^C_{9\to 10,h} - 1.0v^C_{9\to 10,l} = 0.0$
cont-at-8	$v^C_{5\to 8,h} + v^C_{6\to 8,l} - 1.0v^C_{8\to 10,h} - 1.0v^C_{8\to 10,l} = 0.0$
cont-at-7	$v^C_{5\to 7,l} - 1.0v^C_{7\to 10,h} - 1.0v^C_{7\to 10,l} = 0.0$
bin-link-6	$\lambda_{z3-3} - 1.0v^C_{9\to 10,h} - 1.0v^C_{8\to 10,h} - 1.0v^C_{7\to 10,h} = 0.0$
cont-at-10	$v^C_{9\to 10,h} + v^C_{9\to 10,l} + v^C_{8\to 10,h} + v^C_{8\to 10,l} + v^C_{7\to 10,h} + v^C_{7\to 10,l} - 1.0v^C_{10\to 12,h} - 1.0v^C_{10\to 11,l} = 0.0$
bin-link-7	$\lambda_{z2-4} - 1.0v^C_{10\to 12,h} = 0.0$
cont-at-12	$v^C_{10\to 12,h} - 1.0v^C_{12\to T,h} - 1.0v^C_{12\to T,l} = 0.0$
cont-at-11	$v^C_{10\to 11,l} - 1.0v^C_{11\to T,h} - 1.0v^C_{11\to T,l} = 0.0$
bin-link-8	$\lambda_{z3-4} - 1.0v^C_{12\to T,h} - 1.0v^C_{11\to T,h} = 0.0$
cont-at-F	$0.0 = 0.0$
cont-at-T	$v^C_{12\to T,h} + v^C_{12\to T,l} + v^C_{11\to T,h} + v^C_{11\to T,l} = 1.0$

Constraints from **availability BBD**

Type	Constraint
cont-at-0	$-1.0v^A_{0\to1,h} - 1.0v^A_{0\to2,l} = -1.0$
bin-link-1	$\lambda_{z1-1} - 1.0v^A_{0\to1,h} = 0.0$
cont-at-1	$v^A_{0\to1,h} - 1.0v^A_{1\to3,h} - 1.0v^A_{1\to5,l} = 0.0$
cont-at-2	$v^A_{0\to2,l} - 1.0v^A_{2\to5,h} - 1.0v^A_{2\to4,l} = 0.0$
bin-link-2	$\lambda_{z1-2} - 1.0v^A_{1\to3,h} - 1.0v^A_{2\to5,h} = 0.0$
cont-at-5	$v^A_{1\to5,l} + v^A_{2\to5,h} - 1.0v^A_{5\to7,h} - 1.0v^A_{5\to7,l} = 0.0$
cont-at-4	$v^A_{2\to4,l} - 1.0v^A_{4\to7,h} - 1.0v^A_{4\to6,l} = 0.0$
cont-at-3	$v^A_{1\to3,h} - 1.0v^A_{3\to6,h} - 1.0v^A_{3\to7,l} = 0.0$
bin-link-3	$\lambda_{z1-3} - 1.0v^A_{5\to7,h} - 1.0v^A_{4\to7,h} - 1.0v^A_{3\to6,h} = 0.0$
cont-at-6	$v^A_{4\to6,l} + v^A_{3\to6,h} - 1.0v^A_{6\to8,h} - 1.0v^A_{6\to9,l} = 0.0$
cont-at-7	$v^A_{5\to7,h} + v^A_{5\to7,l} + v^A_{4\to7,h} + v^A_{3\to7,l} - 1.0v^A_{7\to10,h} - 1.0v^A_{7\to10,l} = 0.0$
bin-link-4	$\lambda_{z2-2} - 1.0v^A_{6\to8,h} - 1.0v^A_{7\to10,h} = 0.0$
cont-at-8	$v^A_{6\to8,h} - 1.0v^A_{8\to11,h} - 1.0v^A_{8\to13,l} = 0.0$
cont-at-10	$v^A_{7\to10,h} + v^A_{7\to10,l} - 1.0v^A_{10\to13,h} - 1.0v^A_{10\to13,l} = 0.0$
cont-at-9	$v^A_{6\to9,l} - 1.0v^A_{9\to13,h} - 1.0v^A_{9\to12,l} = 0.0$
bin-link-5	$\lambda_{z2-3} - 1.0v^A_{8\to11,h} - 1.0v^A_{10\to13,h} - 1.0v^A_{9\to13,h} = 0.0$
cont-at-11	$v^A_{8\to11,h} - 1.0v^A_{11\to14,h} - 1.0v^A_{11\to15,l} = 0.0$
cont-at-13	$v^A_{8\to13,l} + v^A_{10\to13,h} + v^A_{10\to13,l} + v^A_{9\to13,h} - 1.0v^A_{13\to15,h} - 1.0v^A_{13\to15,l} = 0.0$
cont-at-12	$v^A_{9\to12,l} - 1.0v^A_{12\to15,h} - 1.0v^A_{12\to14,l} = 0.0$
bin-link-6	$\lambda_{z2-4} - 1.0v^A_{11\to14,h} - 1.0v^A_{13\to15,h} - 1.0v^A_{12\to15,h} = 0.0$
cont-at-15	$v^A_{11\to15,l} + v^A_{13\to15,h} + v^A_{13\to15,l} + v^A_{12\to15,h} - 1.0v^A_{15\to18,h} - 1.0v^A_{15\to18,l} = 0.0$
cont-at-14	$v^A_{11\to14,h} + v^A_{12\to14,l} - 1.0v^A_{14\to16,h} - 1.0v^A_{14\to17,l} = 0.0$
bin-link-7	$\lambda_{z3-3} - 1.0v^A_{15\to18,h} - 1.0v^A_{14\to16,h} = 0.0$
cont-at-16	$v^A_{14\to16,h} - 1.0v^A_{16\to T,h} - 1.0v^A_{16\to F,l} = 0.0$
cont-at-17	$v^A_{14\to17,l} - 1.0v^A_{17\to F,h} - 1.0v^A_{17\to T,l} = 0.0$
cont-at-18	$v^A_{15\to18,h} + v^A_{15\to18,l} - 1.0v^A_{18\to F,h} - 1.0v^A_{18\to F,l} = 0.0$
bin-link-8	$\lambda_{z3-4} - 1.0v^A_{16\to T,h} - 1.0v^A_{17\to F,h} - 1.0v^A_{18\to F,h} = 0.0$
cont-at-F	$v^A_{16\to F,l} + v^A_{17\to F,h} + v^A_{18\to F,h} + v^A_{18\to F,l} = 0.0$
cont-at-T	$v^A_{16\to T,h} + v^A_{17\to T,l} = 1.0$

2.4 Intersection BDD

This is a little more tricky. First, I make 'availability' and 'covering' diagrams order-associated:

```python
import varseq as vs
from BB_search import BBSearch

print(f"Size *before* alignment: {A.size()} + {C.size()} = {A.size() + C.size()} nodes.")
vs_A = vs.VarSeq(A.vars, [len(L) for L in A.layers[:-1]])
vs_C = vs.VarSeq(C.vars, [len(L) for L in C.layers[:-1]])
b = BBSearch(vs_A, vs_C)

status = b.search()
assert status == "optimal" or status == "timeout"

Ap = A.align_to(b.Ap_cand.layer_var, inplace=False)
```

```
Cp = C.align_to(b.Ap_cand.layer_var, inplace=False)

Ap.show(dir="reports/2021-02-23_Status_BM/", filename="A_aligned.dot", x_prefix='')
Cp.show(dir="reports/2021-02-23_Status_BM/", filename="C_aligned.dot", x_prefix='')
print(f"Size *after* alignment: {Ap.size()} + {Cp.size()} = {Ap.size() + Cp.size()} nodes.")
print(f"The order revised from \n A: {A.vars}, and\n C: {C.vars}...")
print(f"to: {Ap.vars}")
```

```
Size *before* alignment: 19 + 13 = 32 nodes.
Size *after* alignment: 22 + 16 = 38 nodes.
The order revised from
 A: ['z1-1', 'z1-2', 'z1-3', 'z2-2', 'z2-3', 'z2-4', 'z3-3', 'z3-4'], and
 C: ['z1-1', 'z1-2', 'z2-2', 'z1-3', 'z2-3', 'z3-3', 'z2-4', 'z3-4']...
to: ['z1-1', 'z1-2', 'z1-3', 'z2-2', 'z2-3', 'z3-3', 'z2-4', 'z3-4']
```

(This results in the diagrams depicted in Figures 3a and 3b, respectively.)

So, I can generate an intersection BDD (Figure 4) and the corresponding MIP:

The objective is:

$$
\begin{aligned}
\text{Minimize: } & v_{0\to1,h} + 11.0v_{0\to2,l} + 12.0v_{10\to15,l} + 2.0v_{12\to17,h} + 12.0v_{12\to18,l} + 12.0v_{9\to16,l} + \\
& 2.2v_{11\to16,h} + 4.2v_{7\to13,h} + 2.2v_{8\to15,h} + 13.0v_{23\to28,l} + 3.1v_{21\to28,h} + 2.3v_{21\to28,l} + 2.3v_{22\to29,h} + \\
& 2.3v_{24\to26,h} + 2.3v_{20\to28,h} + 13.0v_{25\to30,l} + 3.1v_{19\to26,h} + 2.3v_{19\to27,l} + 3.0v_{26\to31,h} + 3.0v_{29\to35,l} + \\
& 14.0v_{36\to T,l} + 2.4v_{34\to F,h} + 14.0v_{32\to F,l} + 2.4v_{33\to F,h} + 2.4v_{31\to T,h} + 14.0v_{35\to F,l},
\end{aligned}
$$

under the constraints, presented in Table 1. Obviously, here I have continuous variables only.

2.5 A quick cross-check

Of course, I'd like to cross-check somehow. E.g., I can just solve each of the three models and make sure the optimal objective coincide. Indeed:

```
Opt statuses are: 2, 2, 2
('optimal' is encoded by 2)
Optimal objectives are:
Simple MIP: 6.3
CPP MIP:    6.3
NF (linear):6.3
```

Here are, e.g., nonzero variables for the CPP MIP (`-1` encodes **True** terminal node).

```
link_z1-1: 1.0
A_vh0_1: 1.0
link_z1-2: 1.0
A_vh1_3: 1.0
link_z1-3: 1.0
A_vh3_6: 1.0
A_vl6_9: 1.0
A_vl9_12: 1.0
A_vl12_14: 1.0
```

```
link_z3-3: 1.0
A_vh14_16: 1.0
link_z3-4: 1.0
A_vh16_-1: 1.0
C_vh0_1: 1.0
C_vh1_3: 1.0
C_vl3_4: 1.0
C_vh4_6: 1.0
C_vl6_8: 1.0
C_vh8_10: 1.0
C_vl10_11: 1.0
C_vh11_-1: 1.0
```

This implies locating facilities ① and ③, and must carry the cost $1 + 3$ for location plus the overlap of 2.3 for overlapping at customer 3. Seems to work: $1 + 3 + 2.3 = 6.3$ (I am looking at Figure 1).

3 The problem: intersection DD seems to blow up.

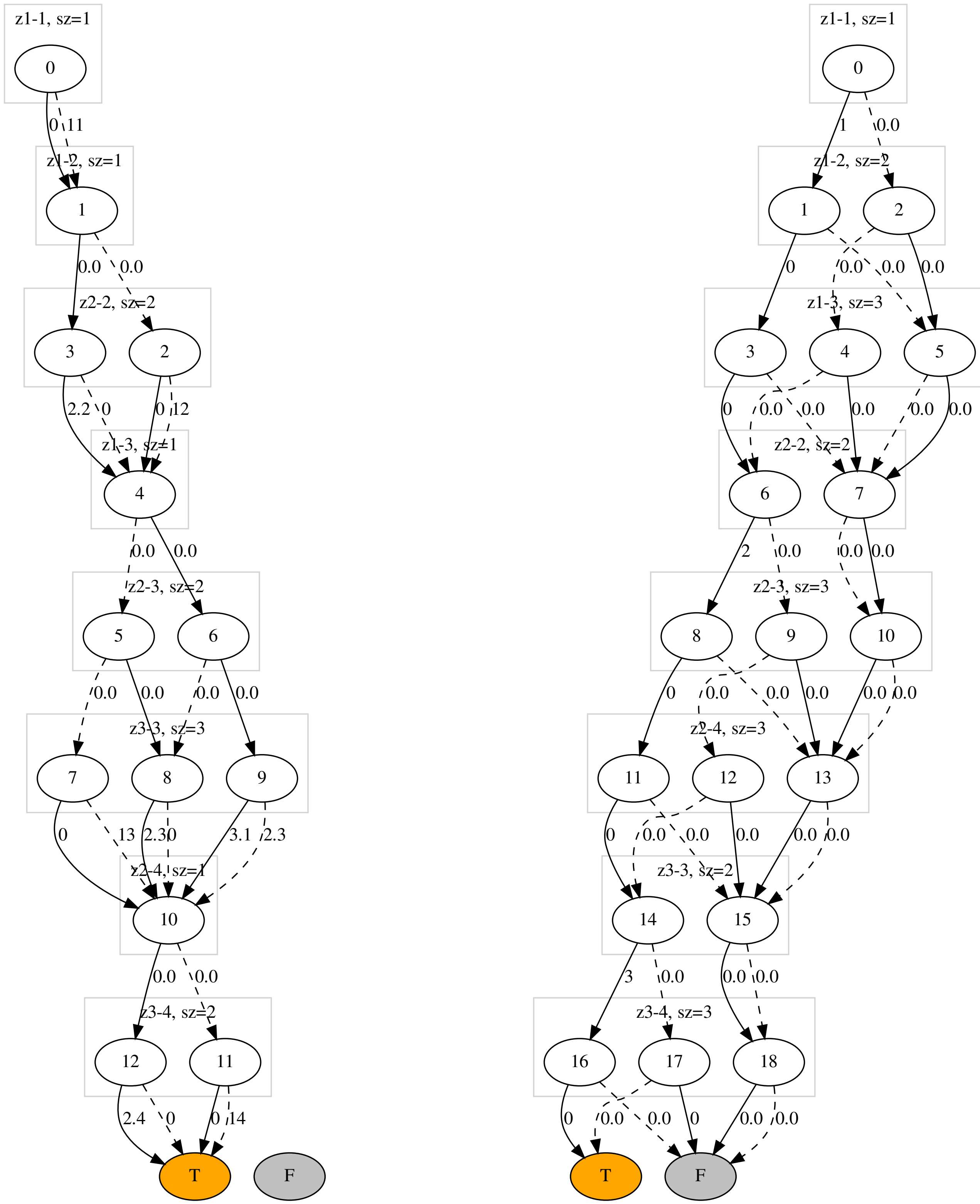
What I have done is a very simple experiment: I generated 15 random instances for each of several problem sizes – say, with number of facilities being $n = 3, 4, 5, 6$, and number of customers $m = 2n$ (in every case). Then, diagram sizes (A for availability and C for covering), along with the number of variables in the plain MIP grow reasonably fast (Figure 5a). However, intersection BDD just blows up (Figure 5b – note I had to draw it in **logarithmic** scale), and so does the runtime of what I am doing (Figure 6). Practically, for the way I am generating instances, it seems I can't get to the point where making a BDD would be beneficial as compared to a plain MIP (it has to have a huge number of variables, right?) – solution just seem to take too long even for moderate problem sizes (I mean, n and m)... That is, maybe even our simplified-problem based approach outperforms, say, some baseline 'sifting' method, but it seems the very idea of constructing a BDD is quite tough to implement, at least in such straightforward way. What I was thinking:

- maybe there is a (performance) bug somewhere, there is always such a possibility. Also, maybe rewriting the whole story in C++ or Julia would help. It is not difficult conceptually, but will take time – and, most importantly, Figure 5b suggests it won't help fundamentally. So, to me, this seems *not* the best starting point.

- in particular, maybe something breaks down when I start to work with *weighted* DDs and that many variables. (we were experimenting with up to 25 variables, now I can have, like, hundreds – and think it is a moderate-sized instance...) So, I might want to look into a more careful testing and optimization of operations with weighted DDs. I kind of checked that weighted-swap/sift work (in a sense of the same terminal nodes and paths), but maybe the resulting DDs are still too big?

- we might want to think about the problem structure. What case would make it especially difficult for a plain MIP formulation? (I remember discussing many overlaps + non-convex structure of g).

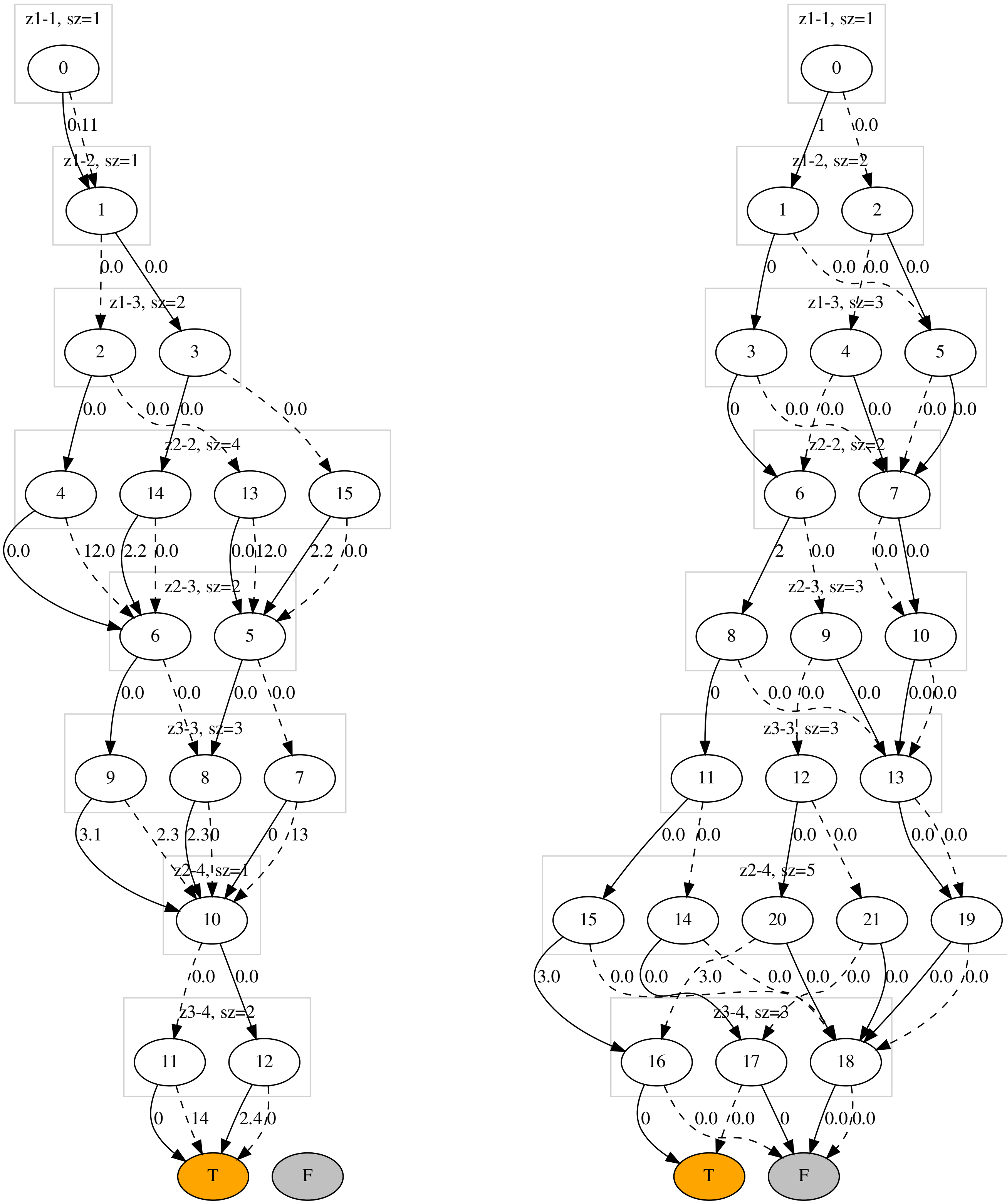
- Like I mentioned, I'd appreciate any comments, but I will keep thinking what we can do.



(a) Covering BDD

(b) Availability BDD

Figure 2: BDDs generated to encode the instance from Figure 1.



(a) Covering BDD

(b) Availability BDD

Figure 3: BDDs generated to encode the instance from Figure 1: after alignment.

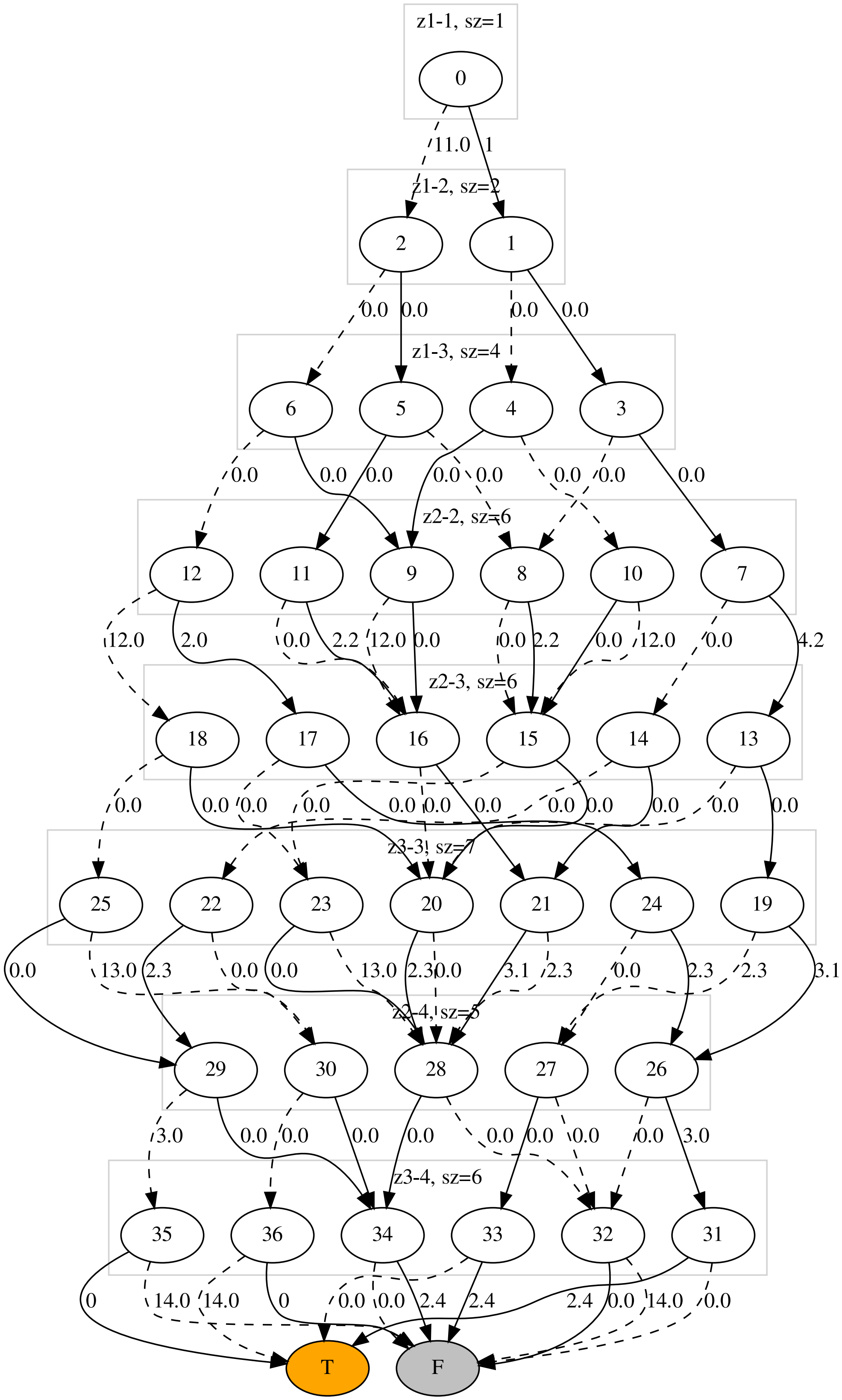
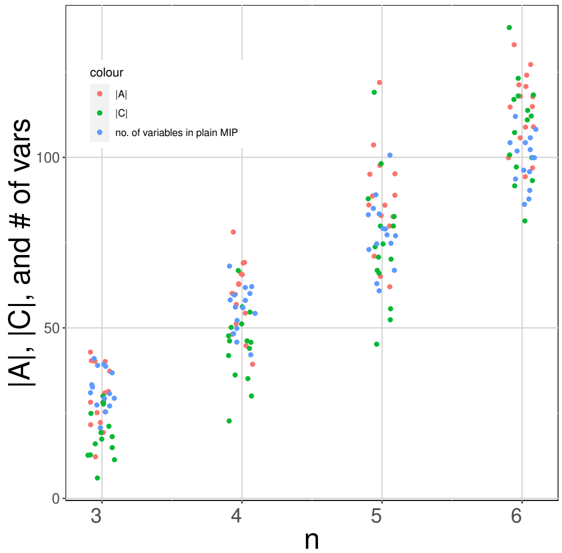


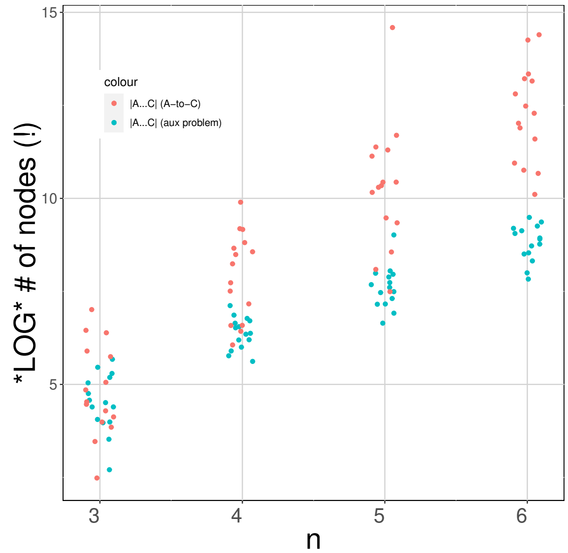
Figure 4: An intersection diagram for 'availability' and 'covering' DDs.

Table 1: Network flow constraints (for the intersection BDD).

Type	**Constraint**
cont-at-0	$-1.0v_{0\to1,h} - 1.0v_{0\to2,l} = -1.0$
cont-at-2	$v_{0\to2,l} - 1.0v_{2\to5,h} - 1.0v_{2\to6,l} = 0.0$
cont-at-1	$v_{0\to1,h} - 1.0v_{1\to3,h} - 1.0v_{1\to4,l} = 0.0$
cont-at-4	$v_{1\to4,l} - 1.0v_{4\to9,h} - 1.0v_{4\to10,l} = 0.0$
cont-at-5	$v_{2\to5,h} - 1.0v_{5\to11,h} - 1.0v_{5\to8,l} = 0.0$
cont-at-3	$v_{1\to3,h} - 1.0v_{3\to7,h} - 1.0v_{3\to8,l} = 0.0$
cont-at-6	$v_{2\to6,l} - 1.0v_{6\to9,h} - 1.0v_{6\to12,l} = 0.0$
cont-at-10	$v_{4\to10,l} - 1.0v_{10\to15,h} - 1.0v_{10\to15,l} = 0.0$
cont-at-12	$v_{6\to12,l} - 1.0v_{12\to17,h} - 1.0v_{12\to18,l} = 0.0$
cont-at-9	$v_{4\to9,h} + v_{6\to9,h} - 1.0v_{9\to16,h} - 1.0v_{9\to16,l} = 0.0$
cont-at-11	$v_{5\to11,h} - 1.0v_{11\to16,h} - 1.0v_{11\to16,l} = 0.0$
cont-at-7	$v_{3\to7,h} - 1.0v_{7\to13,h} - 1.0v_{7\to14,l} = 0.0$
cont-at-8	$v_{5\to8,l} + v_{3\to8,l} - 1.0v_{8\to15,h} - 1.0v_{8\to15,l} = 0.0$
cont-at-18	$v_{12\to18,l} - 1.0v_{18\to20,h} - 1.0v_{18\to25,l} = 0.0$
cont-at-13	$v_{7\to13,h} - 1.0v_{13\to19,h} - 1.0v_{13\to20,l} = 0.0$
cont-at-17	$v_{12\to17,h} - 1.0v_{17\to24,h} - 1.0v_{17\to23,l} = 0.0$
cont-at-16	$v_{9\to16,h} + v_{9\to16,l} + v_{11\to16,h} + v_{11\to16,l} - 1.0v_{16\to21,h} - 1.0v_{16\to20,l} = 0.0$
cont-at-14	$v_{7\to14,l} - 1.0v_{14\to21,h} - 1.0v_{14\to22,l} = 0.0$
cont-at-15	$v_{10\to15,h} + v_{10\to15,l} + v_{8\to15,h} + v_{8\to15,l} - 1.0v_{15\to20,h} - 1.0v_{15\to23,l} = 0.0$
cont-at-23	$v_{17\to23,l} + v_{15\to23,l} - 1.0v_{23\to28,h} - 1.0v_{23\to28,l} = 0.0$
cont-at-21	$v_{16\to21,h} + v_{14\to21,h} - 1.0v_{21\to28,h} - 1.0v_{21\to28,l} = 0.0$
cont-at-22	$v_{14\to22,l} - 1.0v_{22\to29,h} - 1.0v_{22\to30,l} = 0.0$
cont-at-24	$v_{17\to24,h} - 1.0v_{24\to26,h} - 1.0v_{24\to27,l} = 0.0$
cont-at-20	$v_{18\to20,h} + v_{13\to20,l} + v_{16\to20,l} + v_{15\to20,h} - 1.0v_{20\to28,h} - 1.0v_{20\to28,l} = 0.0$
cont-at-25	$v_{18\to25,l} - 1.0v_{25\to29,h} - 1.0v_{25\to30,l} = 0.0$
cont-at-19	$v_{13\to19,h} - 1.0v_{19\to26,h} - 1.0v_{19\to27,l} = 0.0$
cont-at-30	$v_{22\to30,l} + v_{25\to30,l} - 1.0v_{30\to34,h} - 1.0v_{30\to36,l} = 0.0$
cont-at-26	$v_{24\to26,h} + v_{19\to26,h} - 1.0v_{26\to31,h} - 1.0v_{26\to32,l} = 0.0$
cont-at-27	$v_{24\to27,l} + v_{19\to27,l} - 1.0v_{27\to33,h} - 1.0v_{27\to32,l} = 0.0$
cont-at-28	$v_{23\to28,h} + v_{23\to28,l} + v_{21\to28,h} + v_{21\to28,l} + v_{20\to28,h} + v_{20\to28,l} - 1.0v_{28\to34,h} - 1.0v_{28\to32,l} = 0.0$
cont-at-29	$v_{22\to29,h} + v_{25\to29,h} - 1.0v_{29\to34,h} - 1.0v_{29\to35,l} = 0.0$
cont-at-36	$v_{30\to36,l} - 1.0v_{36\to F,h} - 1.0v_{36\to T,l} = 0.0$
cont-at-34	$v_{30\to34,h} + v_{28\to34,h} + v_{29\to34,h} - 1.0v_{34\to F,h} - 1.0v_{34\to F,l} = 0.0$
cont-at-32	$v_{26\to32,l} + v_{27\to32,l} + v_{28\to32,l} - 1.0v_{32\to F,h} - 1.0v_{32\to F,l} = 0.0$
cont-at-33	$v_{27\to33,h} - 1.0v_{33\to F,h} - 1.0v_{33\to T,l} = 0.0$
cont-at-31	$v_{26\to31,h} - 1.0v_{31\to T,h} - 1.0v_{31\to F,l} = 0.0$
cont-at-35	$v_{29\to35,l} - 1.0v_{35\to T,h} - 1.0v_{35\to F,l} = 0.0$
cont-at-T	$v_{36\to T,l} + v_{33\to T,l} + v_{31\to T,h} + v_{35\to T,h} = 1.0$
cont-at-F	$v_{36\to F,h} + v_{34\to F,h} + v_{34\to F,l} + v_{32\to F,h} + v_{32\to F,l} + v_{33\to F,h} + v_{31\to F,l} + v_{35\to F,l} = 0.0$



(a) Diagram growth as number of facilities increases.



(b) Intersection diagram growth as number of facilities increases.

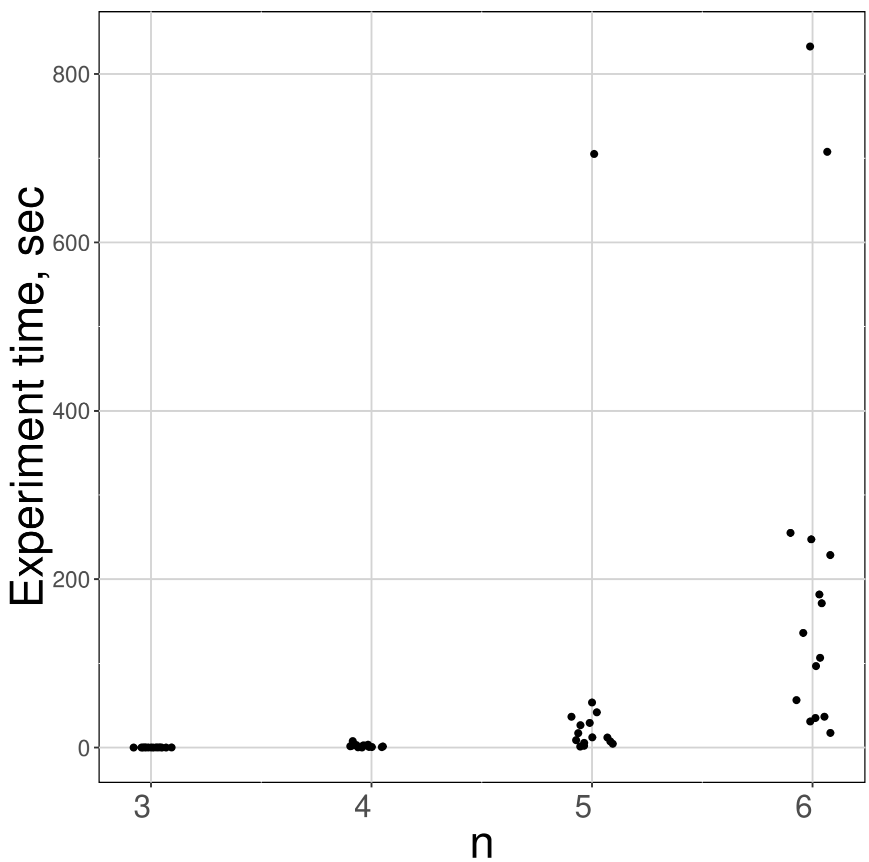


Figure 6: Experiment runtimes, seconds