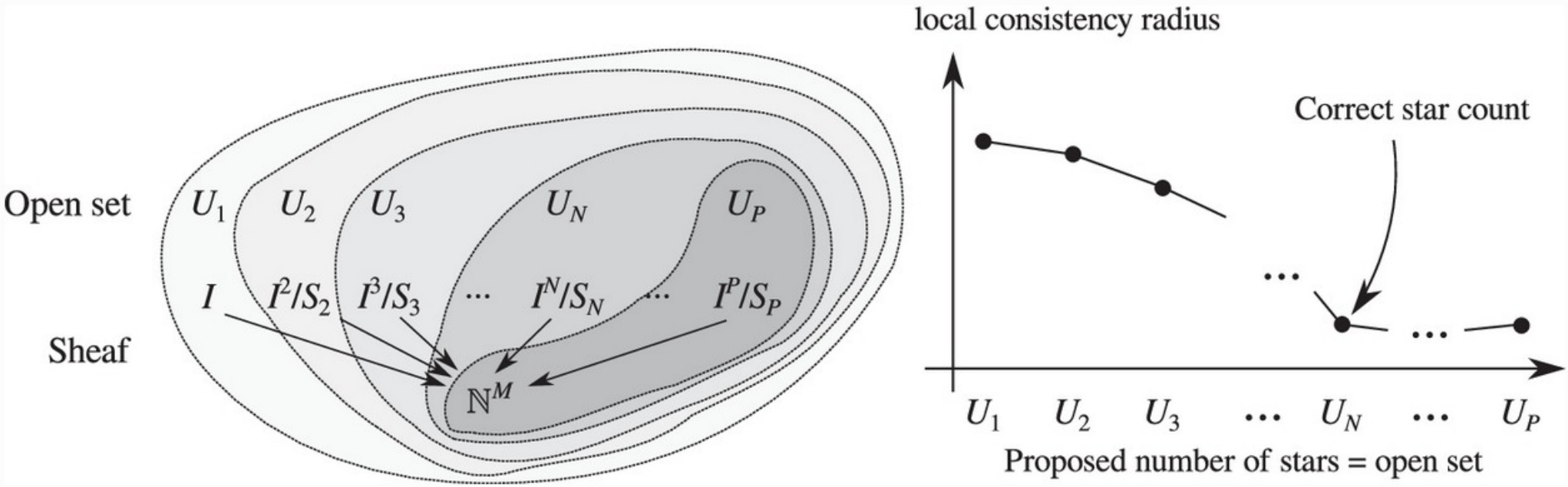
Examples, Examples, Examples

- In this set of slides
  - 9: Polynomials
  - 9B: Simultaneous Polynomials
  - 10: USA East Coast Travel Problem

Polynomials

- This a simplistic exercise designed to demonstrate how the CR goes down and stays down when you get to an appropriate model.
- The inspiration was from MR's "star" counting paper: *Super-resolving star clusters with sheaves*
- https://asp-eurasipjournals.springeropen.com/articles/10.1186/s13634-022-00861-8



Determining the correct number of stars by iterating over open sets in the topology

This also happens with polynomials

- The underlying model is “star-shaped” and works just like our LSR example from earlier.
- Start with points on a cubic.

```
#Here are the data points to fit to a polynomial
P=[(2,9),(4,41),(7,134),(8,177),(10,281),(12,409)]
#Points are from y = 3x^2 -2x +1

print("User can select one of 5 models. Enter number 1 through 5.")
print("1 is constant, 2 is line etc.")
```

Transition is obvious!

```
User can select one of 5 models. Enter number 1 through 5.
1 is constant, 2 is line etc.
> 2
1002.0
Value at P0 is 9
Value at P1 is 41
Value at P2 is 134
Value at P3 is 177
Value at P4 is 281
Value at P5 is 409
Value at M is [-119.000008     40.00000066]
Consistency Radius: 147.00000735758476
Do you wish to continue?
>y
User can select one of 5 models. Enter number 1 through 5.
1 is constant, 2 is line etc.
> 3
625.0
Value at P0 is 9
Value at P1 is 41
Value at P2 is 134
Value at P3 is 177
Value at P4 is 281
Value at P5 is 409
Value at M is [ 1.00011052 -2.00007214  3.00000509]
Consistency Radius: 0.00052164458222137
Do you wish to continue?
```

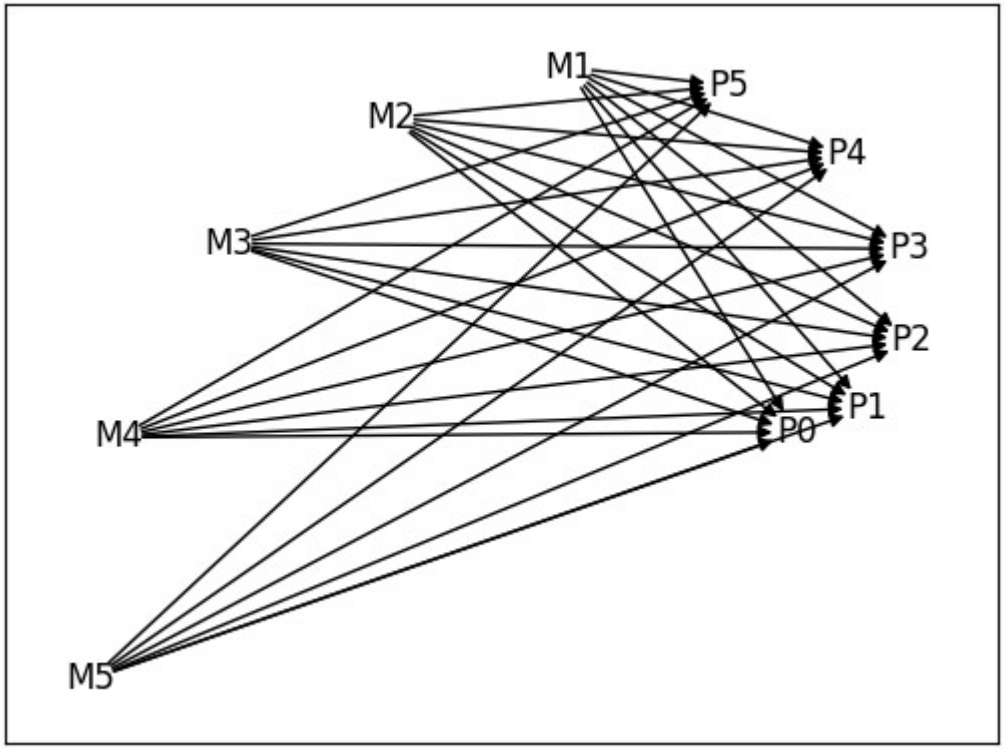
CR stays low, model doesn't change!

```
User can select one of 5 models. Enter number 1 through 5.
1 is constant, 2 is line etc.
> 3
625.0
Value at P0 is 9
Value at P1 is 41
Value at P2 is 134
Value at P3 is 177
Value at P4 is 281
Value at P5 is 409
Value at M is [ 1.00011052 -2.00007214  3.00000509]
Consistency Radius: 0.00052164458222137
Do you wish to continue?
>y
4
User can select one of 5 models. Enter number 1 through 5.
1 is constant, 2 is line etc.
> 3030.0
Value at P0 is 9
Value at P1 is 41
Value at P2 is 134
Value at P3 is 177
Value at P4 is 281
Value at P5 is 409
Value at M is [ 9.99818278e-01 -1.99990404e+00  2.99998367e+00  8.50516173e-07]
Consistency Radius: 0.00018817848760477318
```

Can we do them all at once?

- Also inspired by same MR paper!
- Can we make a sheaf with all this info simultaneously?
- Eventually, we'd want the object itself to select the appropriate model.

All models connected to all data



Results

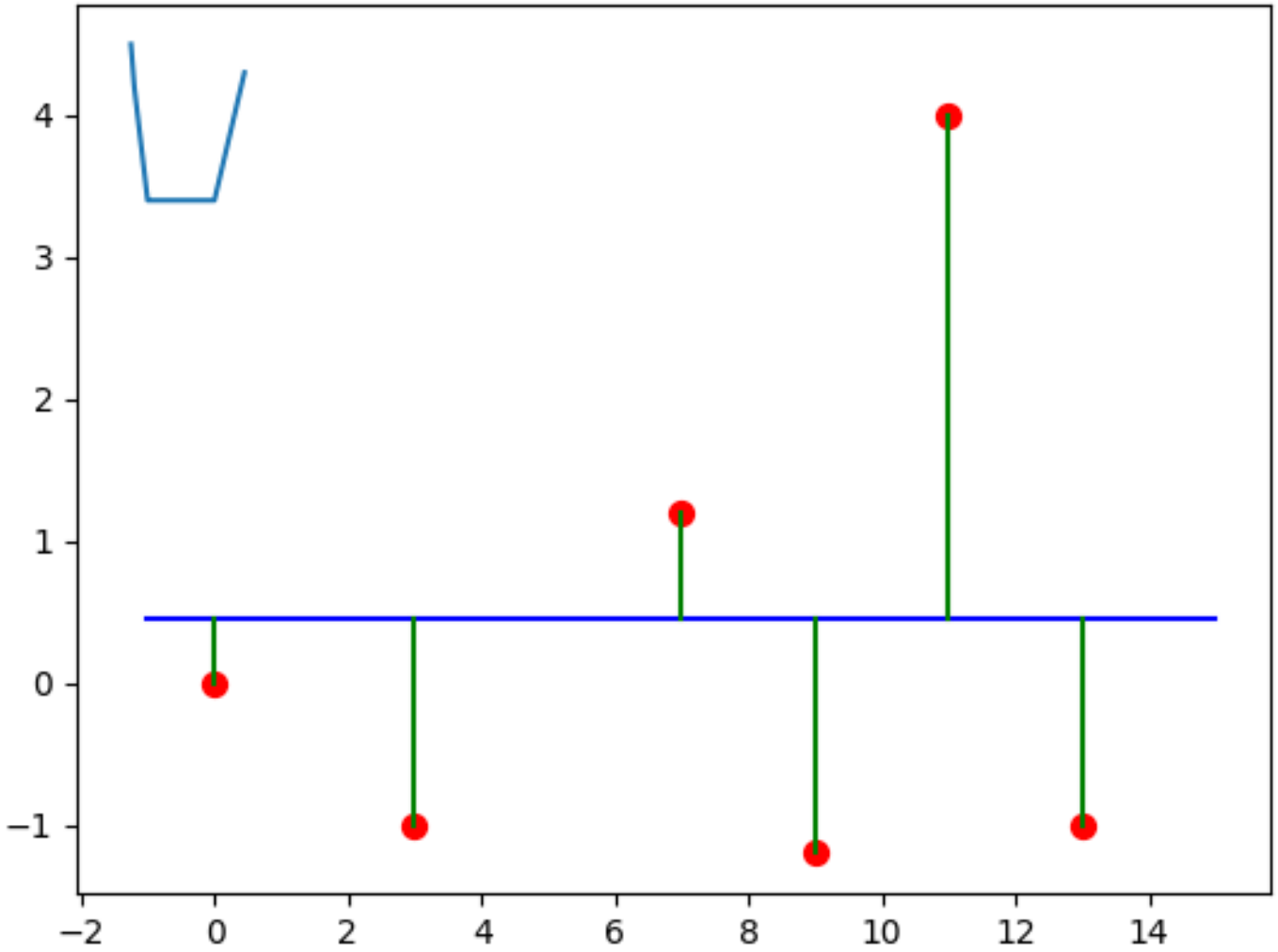
```
M1 model CR:
683.0

M2 model CR:
147.0006133049497

M3 model CR:
0.00029994958598855703

M4 model CR:
0.0007707026398424688

M5 model CR:
0.030445779199926193

KEY: Points are actually all from y = 3x^2 =2x +1 or parabolic model.
```

Results

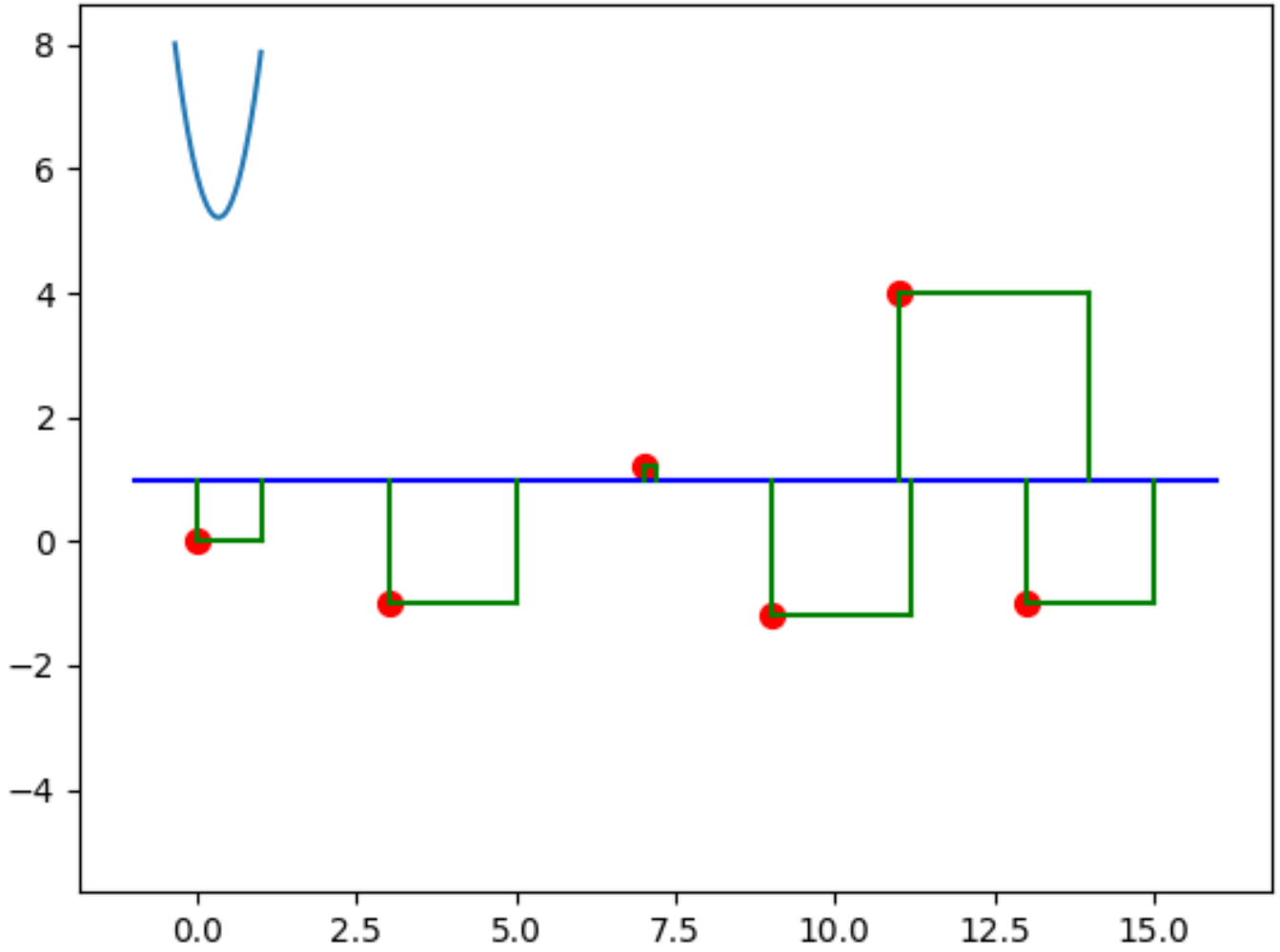
```
M1 model CR:
683.0

M2 model CR:
147.0006133049497

M3 model CR:
0.0002999495859885570 3

M4 model CR:
0.00077070263984246 88

M5 model CR:
0.030445779199926193

KEY: Points are actually all from y = 3x^2 =2x +1 or parabolic model.
```

SLIGHT OF HAND… Let's dig deeper!!

NormType=1



NormType=2



Polynomial models: one-by-one

NormType = 1

n	CR	Coefficients
1	683	157
2	147	-119, 40
3	0.0005	1, -2, 3
4	0.0002	1, -2, 3, 0
5	0.066	1, -2, 3, 0, 0

157 & 176 both solutions! in NormType=1

USED ALL DOF

NormType = 2

n	CR	Coefficients
1	336	175
2	74.6	-108, 39.5
3	0.0000002	1, -2, 3
4	0.003	1, -2, 3, 0
5	0.014	1, -2, 3, 0, 0

USED ALL DOF

A Sheaf of polynomial models:

NormType = 1

n	CR	Coefficients
1	683	176
2	147	-118, 40
3	0.0003	1, -2, 3
4	0.0007	1, -2, 3, 0
5	0.03	1, -2, 3, 0, 0

USED ALL DOF

NormType = 2

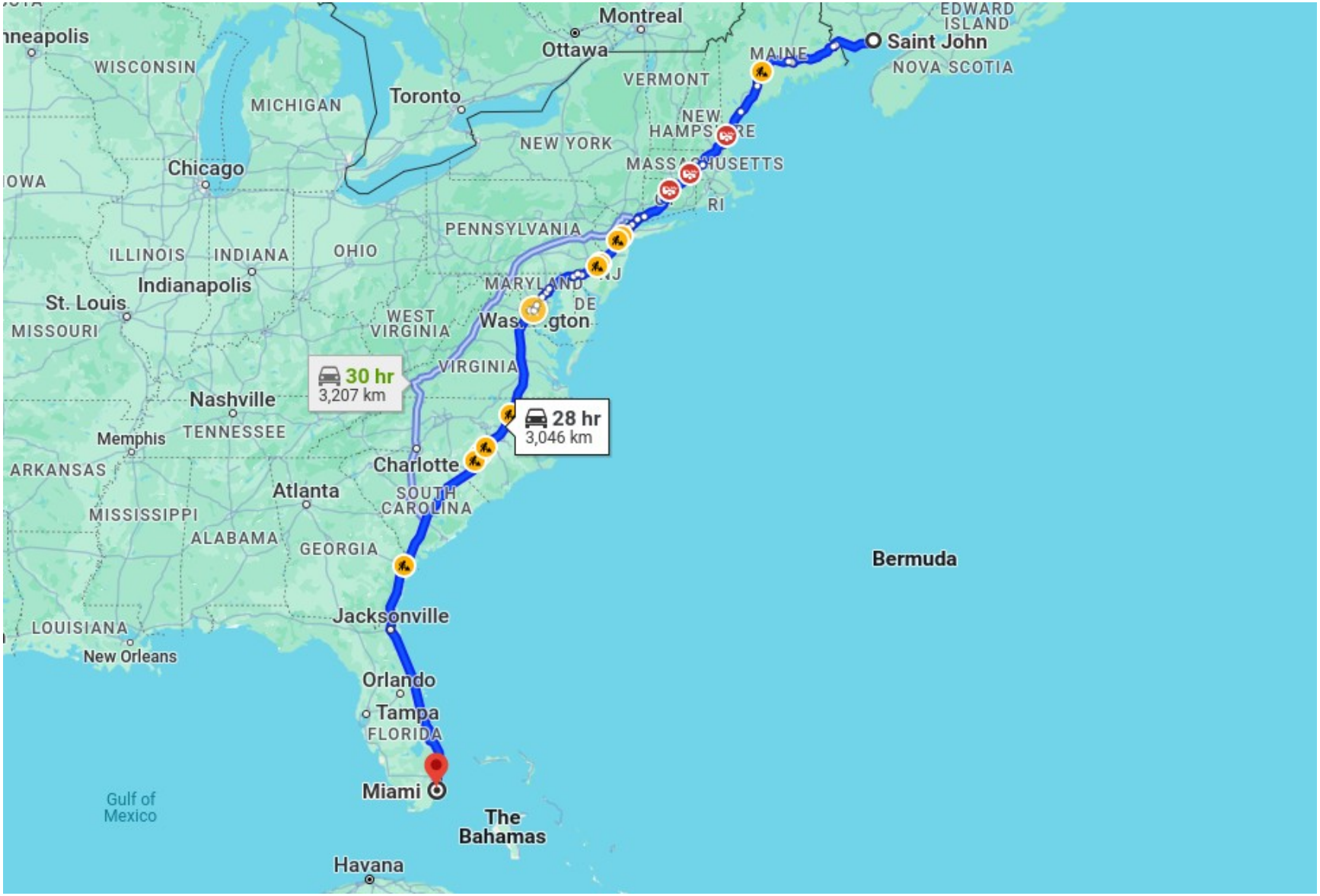
n	CR	Coefficients
1	336	175
2	74.6	-108, 39.5
3	0.3	0.46, -1.8, 3
4	0.34	-0.3, -1.2, 2.8, 0
5	0.22	-0.6, -0.6, 2.6, 0.04, -0.01

NormType=2 can get stuck in local minima!!! Esp w/SLSQP

Key Takeaways

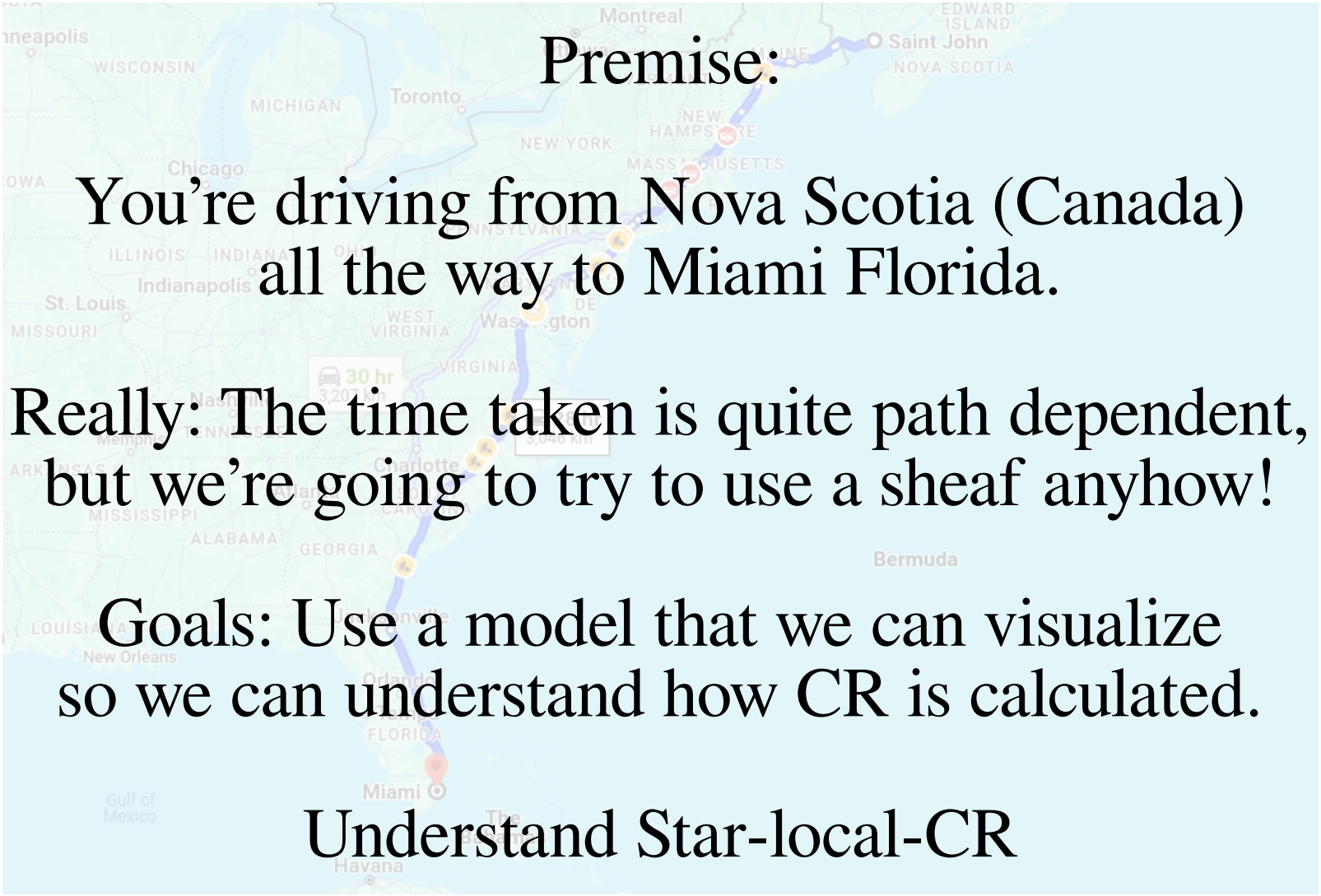
- NormType = 1 is known to encourage “sparsity”
  - It tends to zero out unnecessary DOF
- Used heavily in Signal Processing
- Optimizers exist to work with NormType 1 for this reason!
- Ask MR more!

USA East Coast Travel



Donna Dietz

USA East Coast Travel



Premise:

You're driving from Nova Scotia (Canada) all the way to Miami Florida.

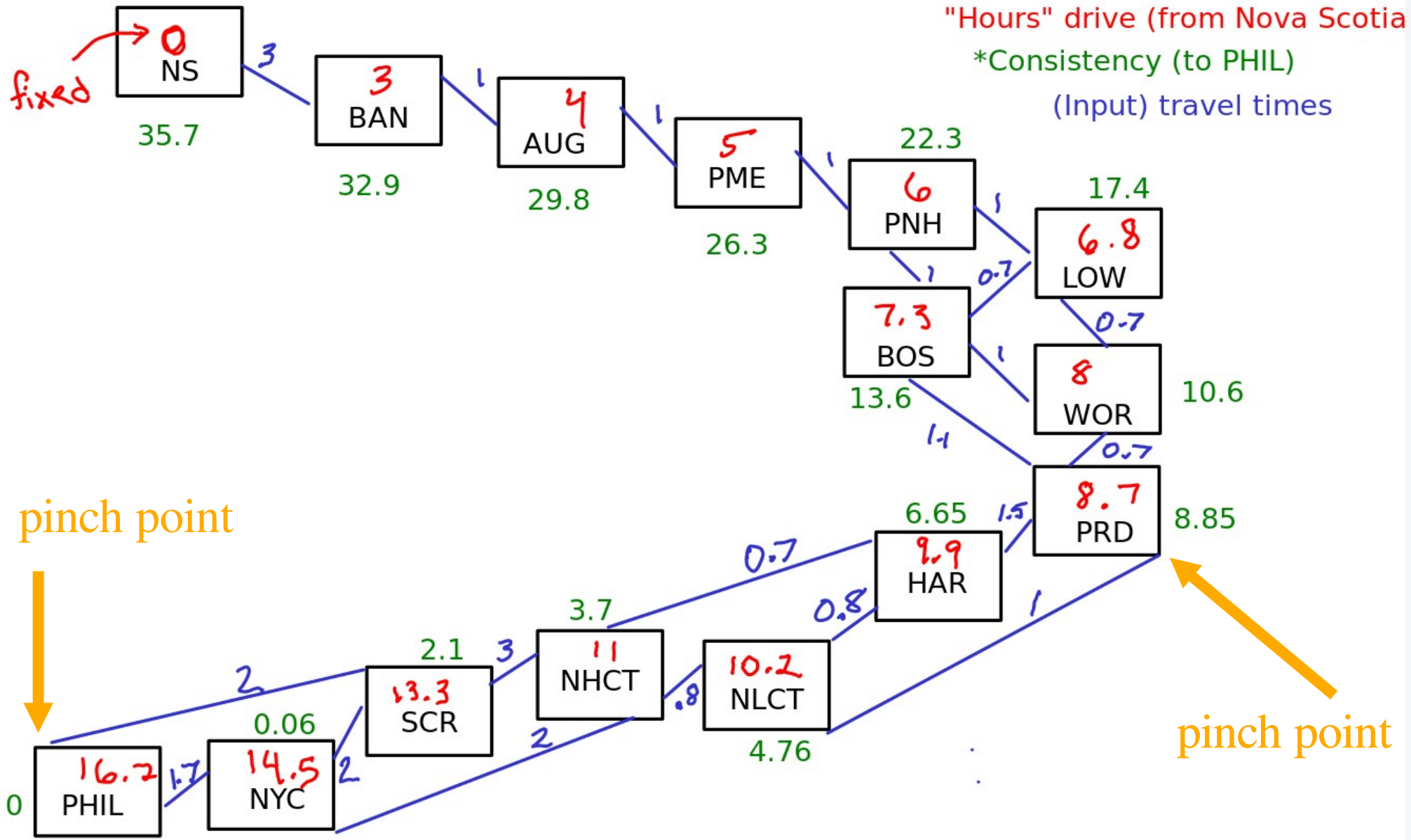
Really: The time taken is quite path dependent, but we're going to try to use a sheaf anyhow!

Goals: Use a model that we can visualize so we can understand how CR is calculated.

Understand Star-local-CR

First lesson

- When I tried linking together all the cities I wanted to link together, this exploded computationally.
- I had to break the problem into smaller sections at the “pinch points” - cities that every path is forced to go through.
- Once you understand how CR’s are calculated, it will immediately make sense why it’s necessary to do this!

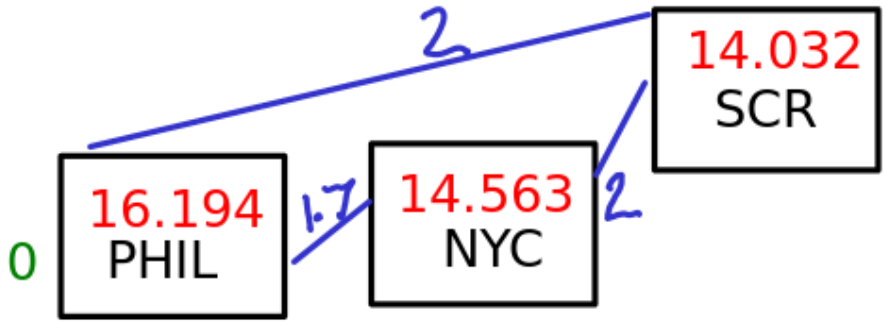

"Hours" drive (from Nova Scotia
*Consistency (to PHIL)
(Input) travel times
fixed
0
NS
35.7
3
BAN
32.9
4
AUG
29.8
5
PME
26.3
22.3
6
PNH
17.4
6.8
LOW
7.3
BOS
13.6
8
WOR
10.6
8.7
PRD
8.85
6.65
9.9
HAR
3.7
11
NHCT
2.1
13.3
SCR
0.06
14.5
NYC
16.7
PHIL
0
10.2
NLCT
4.76
pinch point
pinch point

First two portions of the path

```
Resulting data on the nodes:
Value at NYC is [14.56266579]
Value at PME is [5.00130861]
Value at SCR is [14.03181844]
Value at WOR is [8.04635546]
Value at LOW is [6.82445287]
Value at BAN is [3.00130891]
Value at PRD is [8.68857607]
Value at NHCT is [11.05361325]
Value at HAR is [10.08889351]
Value at NLCT is [10.33361458]
Value at BOS is [7.29052235]
Value at PNH is [6.00130891]
Value at AUG is [4.00130891]
Value at NS is [0]
Value at PHIL is [16.19423775]
Consistency Radius: 35.741699250599986
```

```
Star Local Consistency Radius per node:
NYC 0.06842804173966144
PME 26.30035584103167
SCR 2.1339227475958094
WOR 10.615906406173231
LOW 17.35516626994872
BAN 32.897046045709324
PRD 8.857751711381306
NHCT 3.6852532279077748
HAR 6.65701432372185
NLCT 4.759248727464579
BOS 13.621520548521513
PNH 22.281275924907913
AUG 29.781910246812647
NS 35.74169925059999
PHIL 0.0
```

Notes: This is impossible to read this way!
In the video, we used a different metric, but the idea is the same.



Explaining why: SCR*CR=2.133

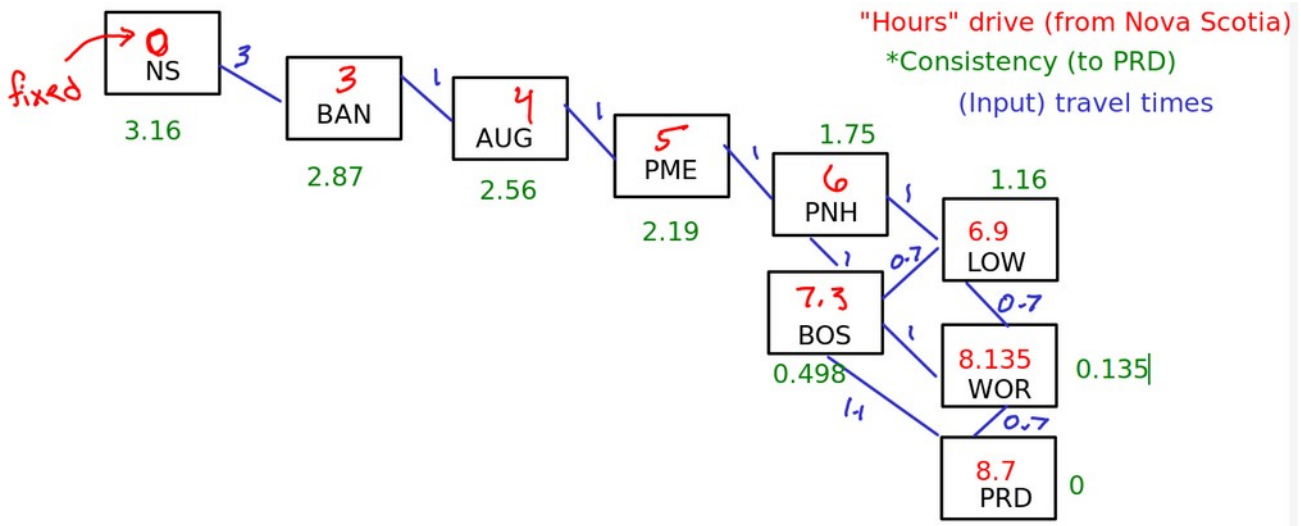
Trace all paths downstream having SCR as a "grandparent":

SCR -> PHIL	14.032+2-16.194 = -0.162 square to get 0.0262
SCR -> NYC -> PHIL	14.032+2+1.7-16.194 = 1.538 square to get 2.3654
SCR -> NYC	14.032+2-14.563 = 1.469 square to get 2.158
NYC -> PHIL	14.563+1.7-16.194 = 0.069 square to get 0.00476

Add up the squared errors to get: 4.55444
Square root of this is 2.134

First portion of path only

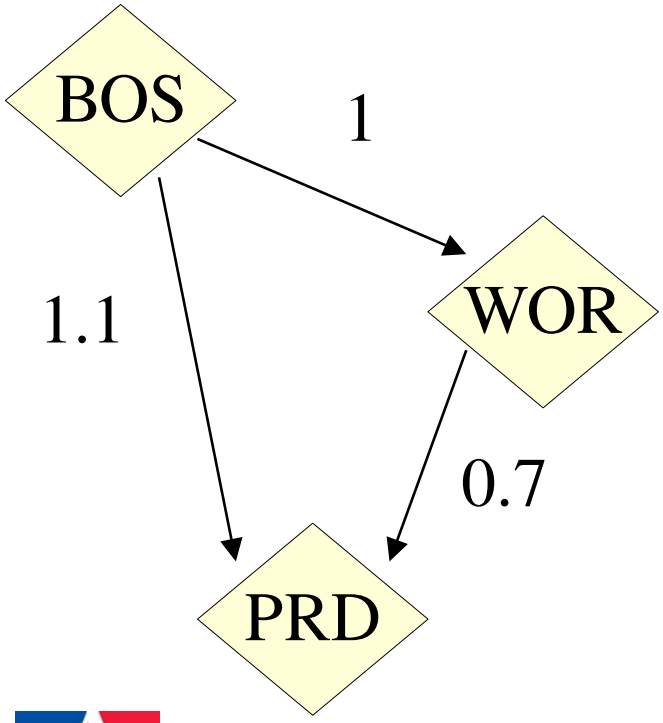
```
Resulting data on the nodes:
Value at PNH is [6.00000203]
Value at PRD is [8.69862077]
Value at WOR is [8.13498967]
Value at AUG is [4.00000183]
Value at PME is [5.00000183]
Value at NS is [0]
Value at BOS is [7.34588738]
Value at LOW is [6.91015696]
Value at BAN is [3.00000209]
Consistency Radius: 3.160885771180035

Star Local Consistency Radius per node:
PNH 1.7530379947533246
PRD 0.0
WOR 0.13636890103009236
AUG 2.5558111304919846
PME 2.191496377568929
NS 3.160885771180035
BOS 0.4975365166382873
LOW 1.1591497095686725
BAN 2.874314648156052
```



Let's do BOS -> PRD

- Resulting data:
- Value at WOR is [8.13498971]
- Value at BOS is [7.34588741]
- Value at PRD is [8.6986208]

- Star Local Consistency Radius:
- WOR 0.1363689123665672 3
- BOS 0.497536514444104 4
- PRD 0.0



WOR->PRD: 8.13498971+0.7-8.6986208= 0.13636891
BOS->WOR: 7.34588741+1-8.13498971= 0.2108977
BOS->PRD: 7.34588741+1.1-8.6986208= -0.25273339
BOS->WOR->PRD: 7.34588741+1.7-8.6986208= 0.34726661

Sum Squares to get: 0.24754258432
Square root gives: 0.49753651556

Pinch-point splits

- Naturally, you need to split on the pinch points to save your sanity and your computing power!
- If there are *n* paths from A to B (ending on B) and *m* paths from B to C (starting on B), there are more than *nm* paths from A to C, because there are still partial paths too!
- I pre-ran the code before I hard-coded the “times” from the top to each pinch-point, so they code is self-consistent if you run the pieces separately.
- Play with this and have fun!

Main concepts in these slides:

- As you adjust your model, and once you land on the “first” of several appropriate models, your CR should drop dramatically!
- You can collect several models into a sheaf and get that larger sheaf to tell you which model was the first good one.
- NormType matters, as does which optimizer you are using!
- NormType = 1 can have “flat spots” which are not necessarily a bad deal, because you’re less likely to get stuck in local mins.
- If there is a nice way to split your problem, you probably should. Computational complexity can blow up quickly!
- You should now understand how CR’s and *CR’s are calculated.