Python Simulation

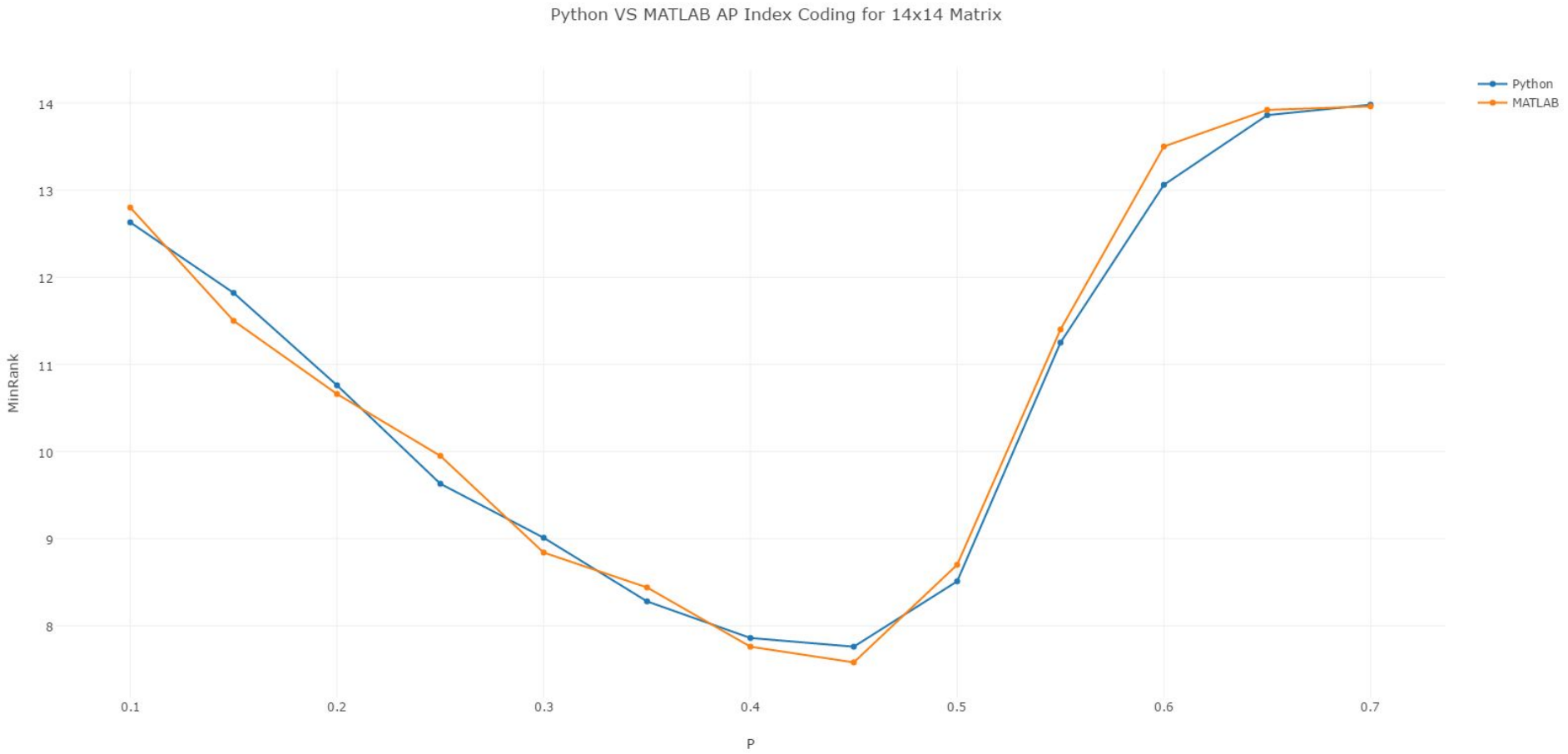
Barak Lidsky

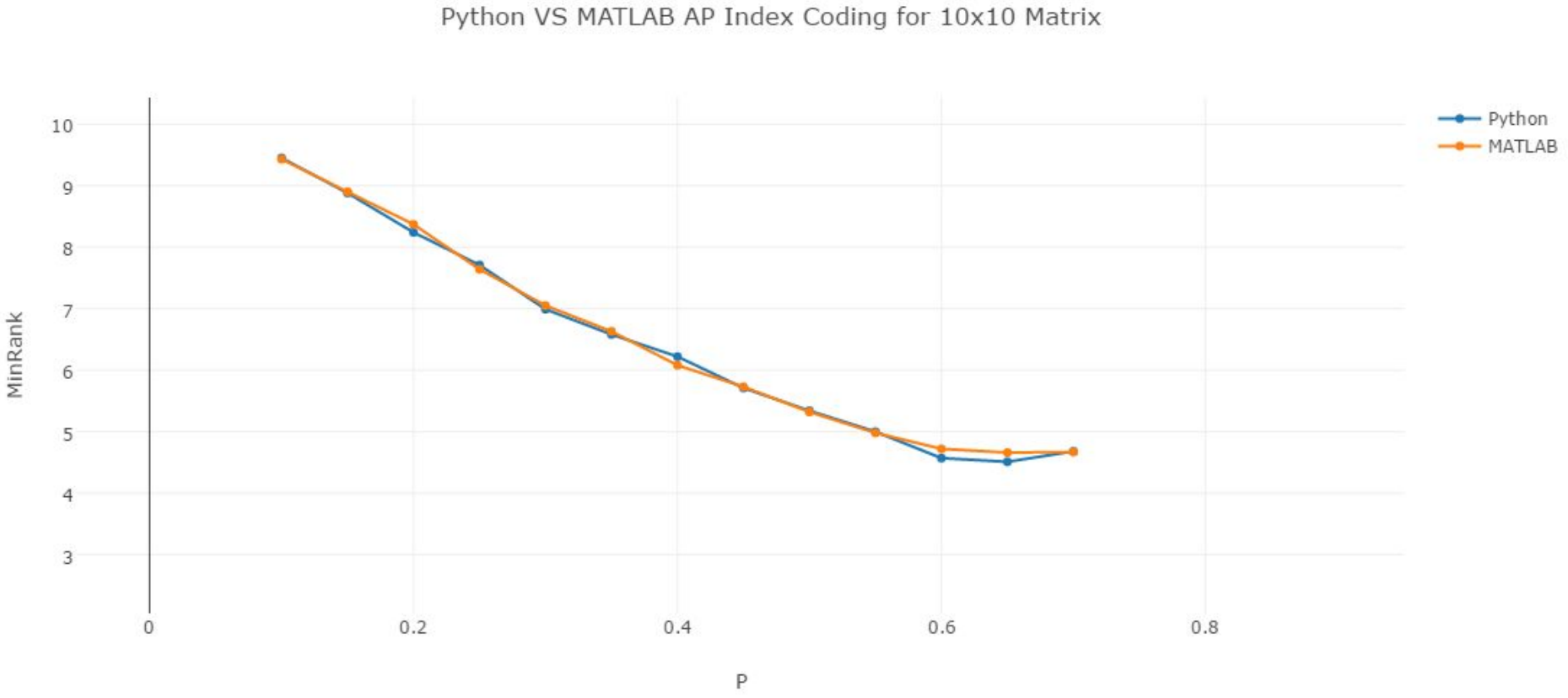
What Was Implemented?

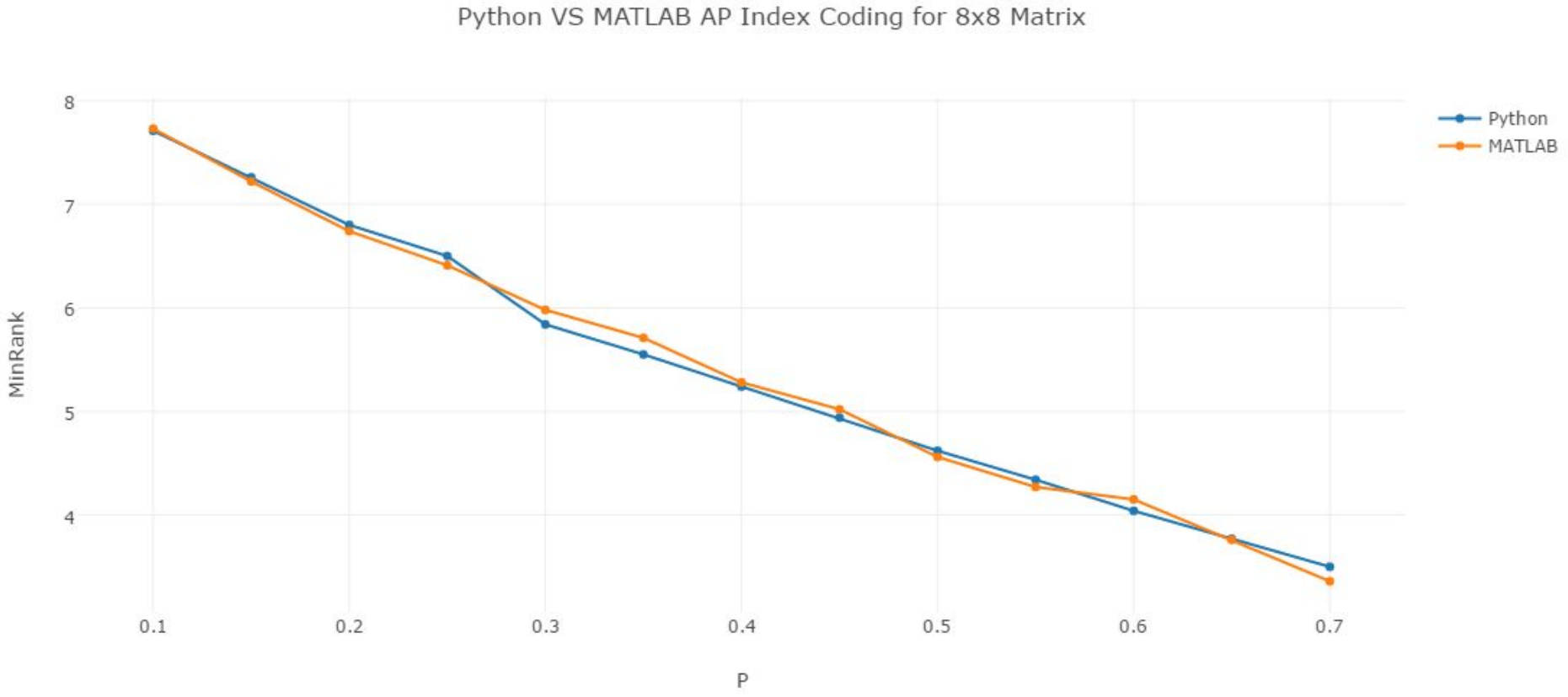
- Directional SVD AP Index Coding
- SVD based encoding and decoding scheme
- Round Robin comparison sending

Directional SVD APIC

- Translated MATLAB version to Python
- Results almost exactly the same as MATLAB algorithm


Python VS MATLAB AP Index Coding for 14x14 Matrix
Python
MATLAB
MinRank
P
14
13
12
11
10
9
8
0.1
0.2
0.3
0.4
0.5
0.6
0.7


Python VS MATLAB AP Index Coding for 10x10 Matrix
MinRank
10
9
8
7
6
5
4
3
0
0.2
0.4
0.6
0.8
P
Python
MATLAB

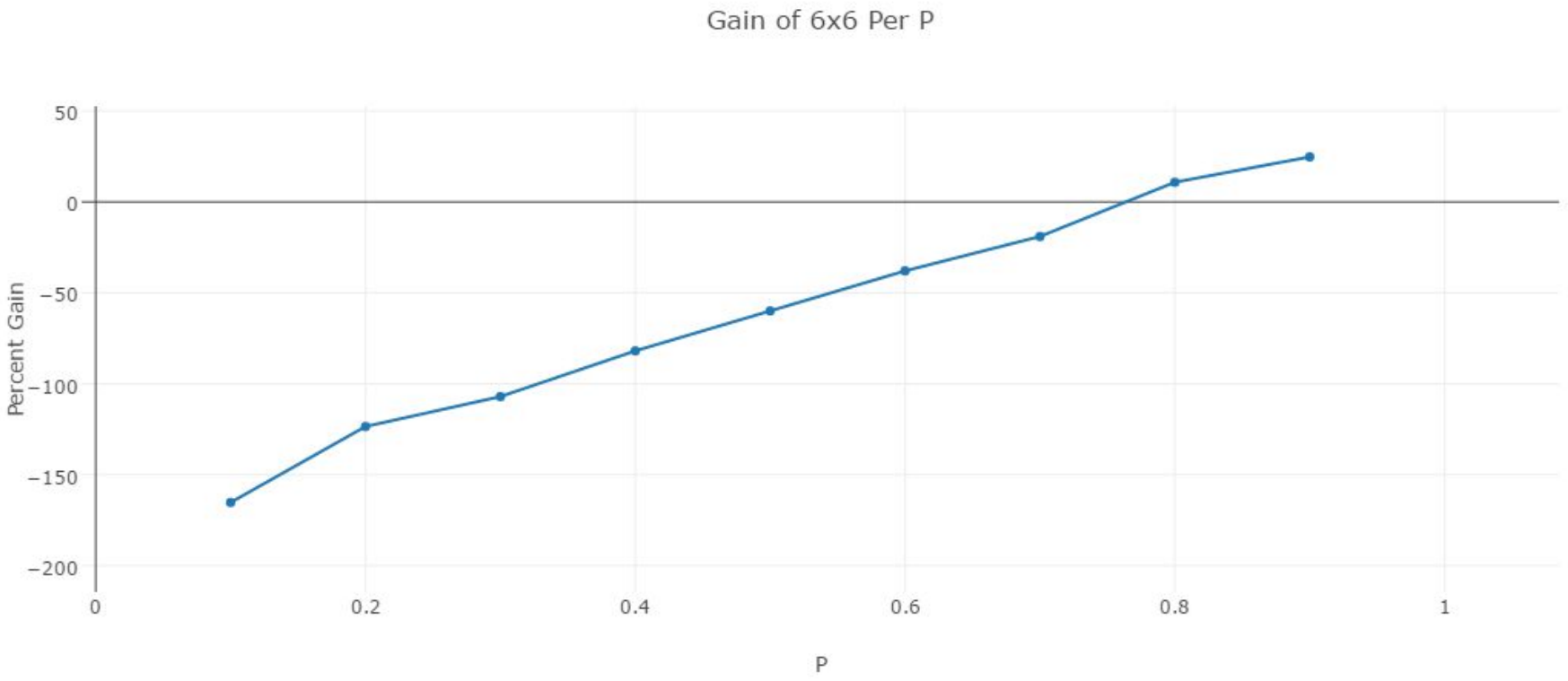

Python VS MATLAB AP Index Coding for 8x8 Matrix
MinRank
P
Python
MATLAB
8
7
6
5
4
0.1
0.2
0.3
0.4
0.5
0.6
0.7

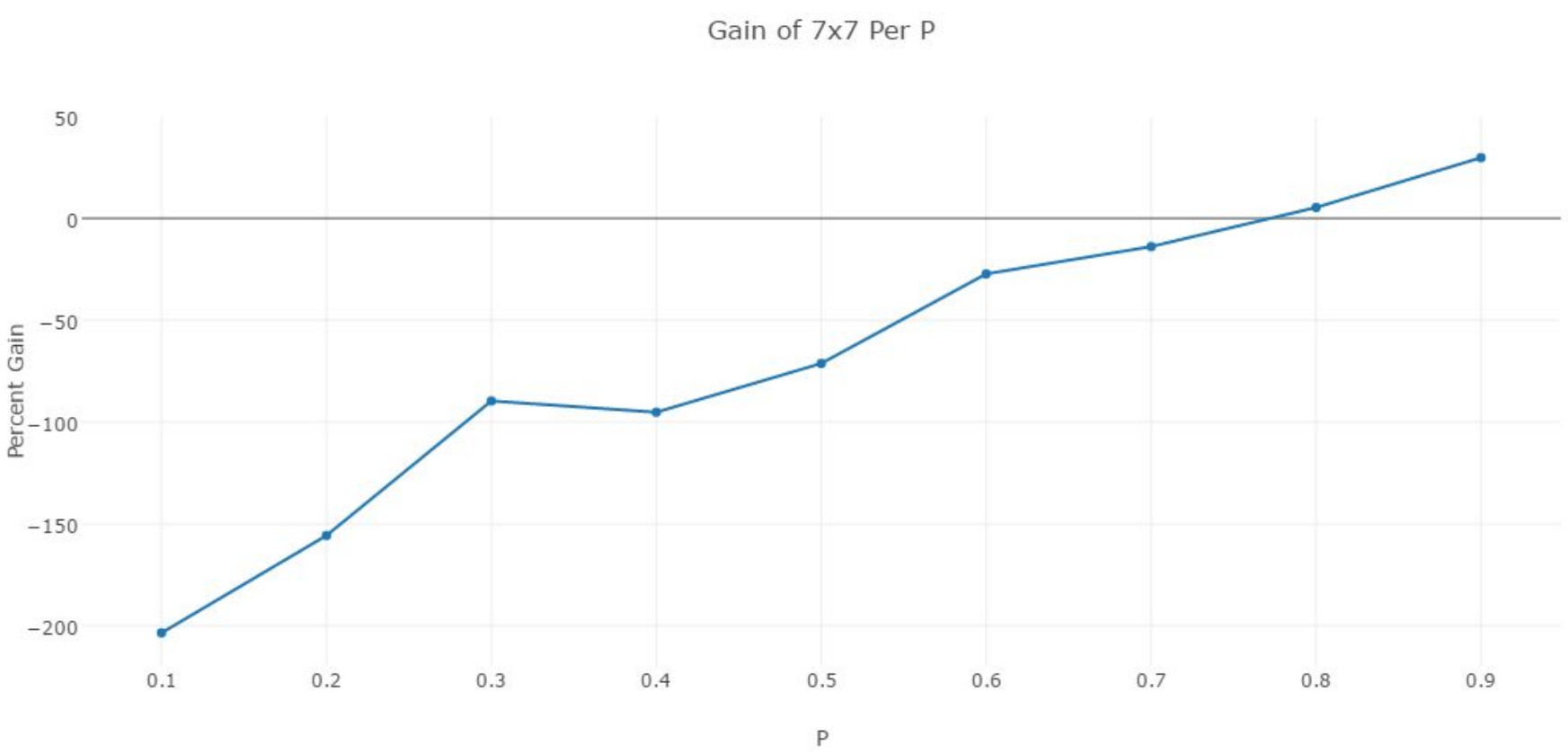
SVD based Decoding

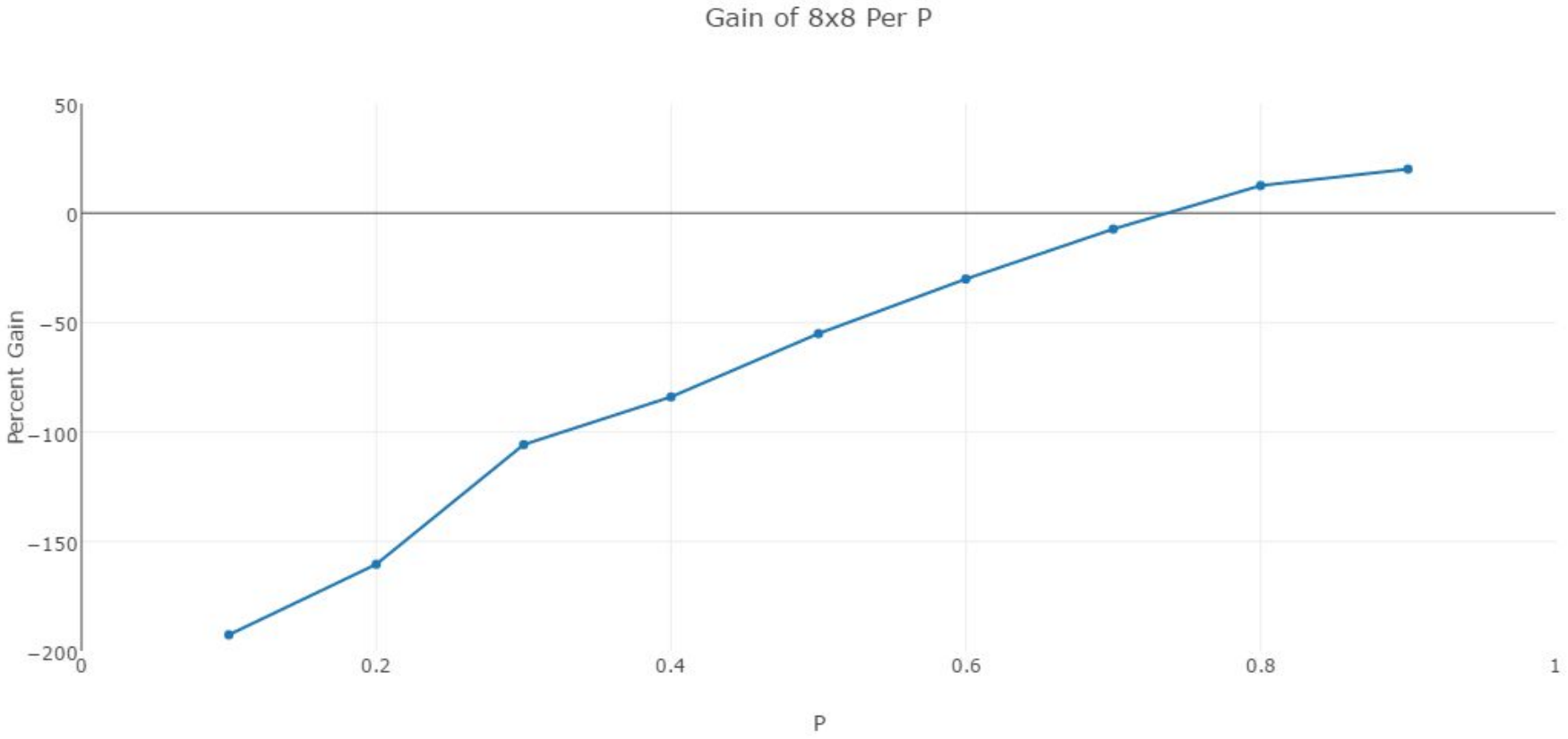
- Encoded X vector of messages using SVD
- Decoded X vector of messages using M using SVD
- Based off of Muryong’s implementation

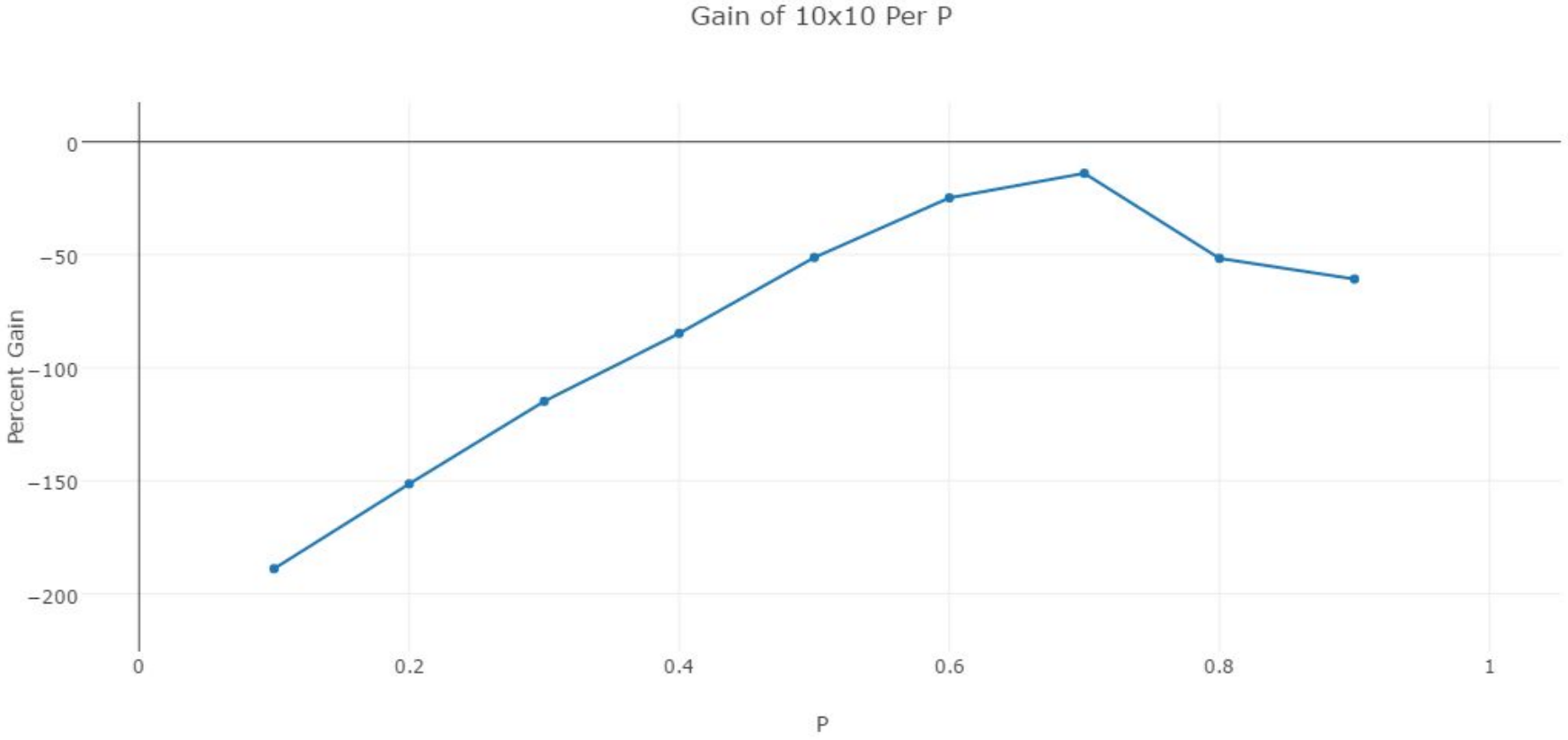
Testing Simulation

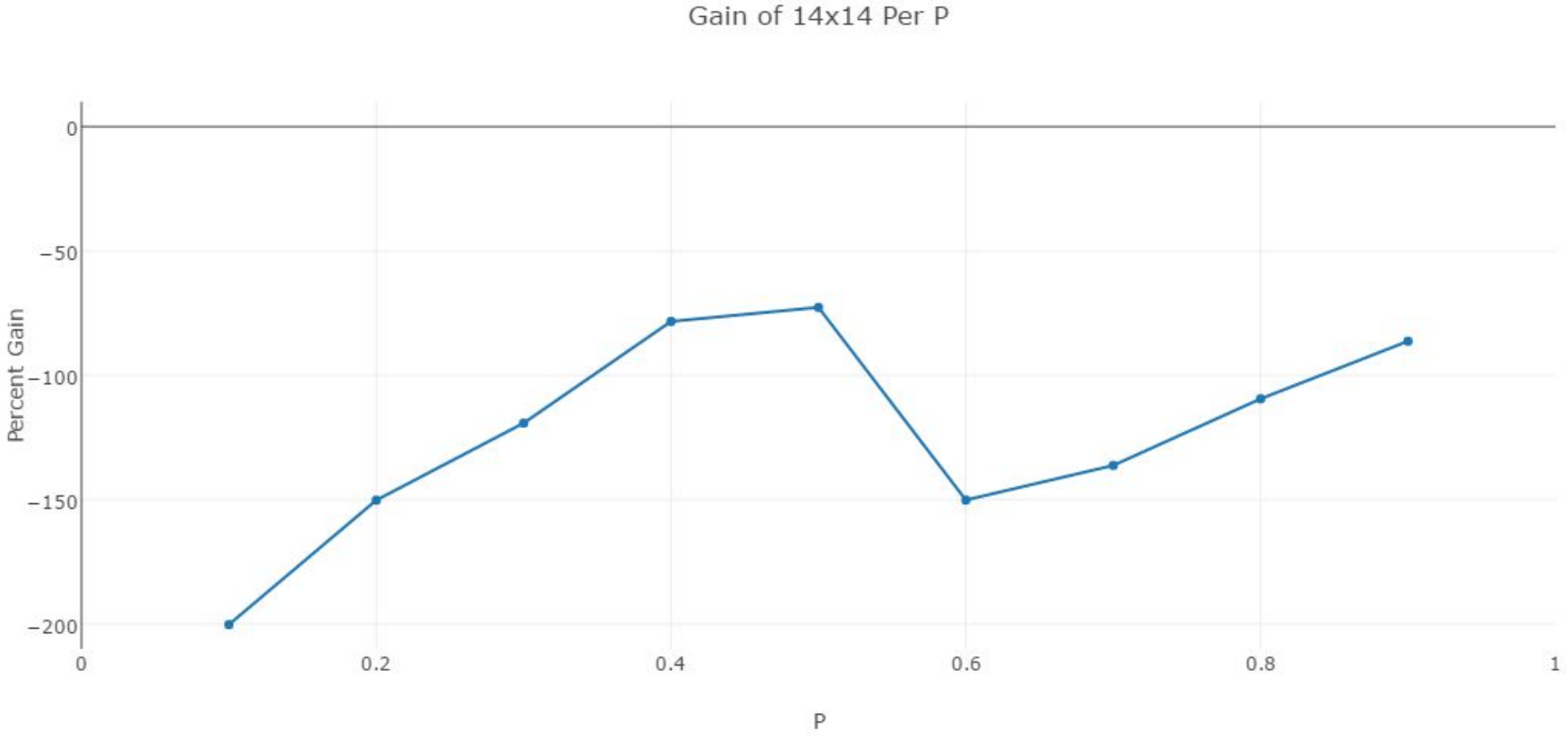
- Initially tested Round Robin(RR) VS AP Index Coding(APIC) with incremented Probabilities(P) for receiving messages
- RR performed much better than APIC with this method
- APIC required sending encoded X vector and M to all receivers which meant more messages had to be sent for everyone to have all of X and M


Gain of 6x6 Per P
Percent Gain
50
0
−50
−100
−150
−200
0
0.2
0.4
0.6
0.8
1
P


Gain of 7x7 Per P
50
0
−50
−100
−150
−200
Percent Gain
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
P

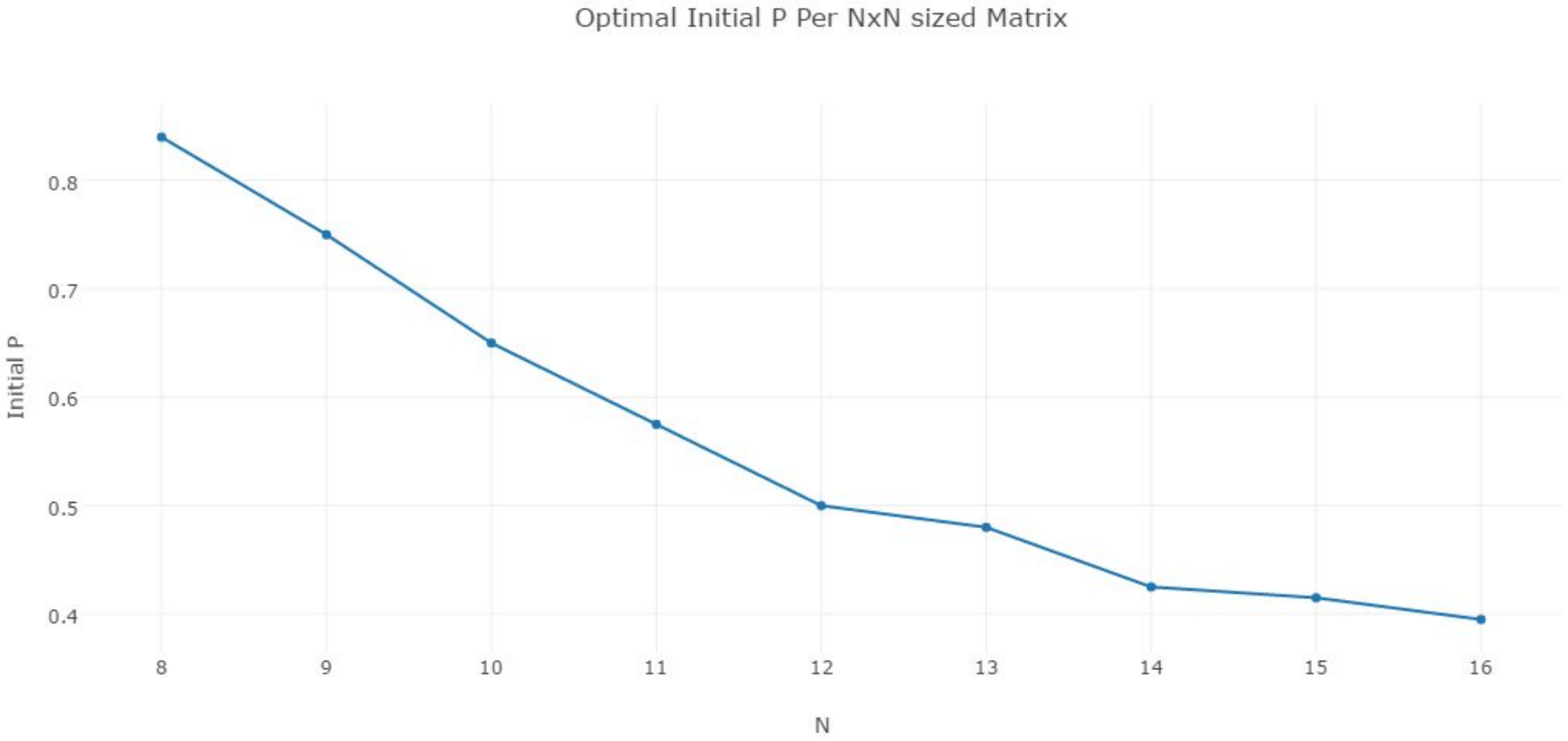

Gain of 8x8 Per P
Percent Gain
50
0
−50
−100
−150
−200
0
0.2
0.4
0.6
0.8
1
P

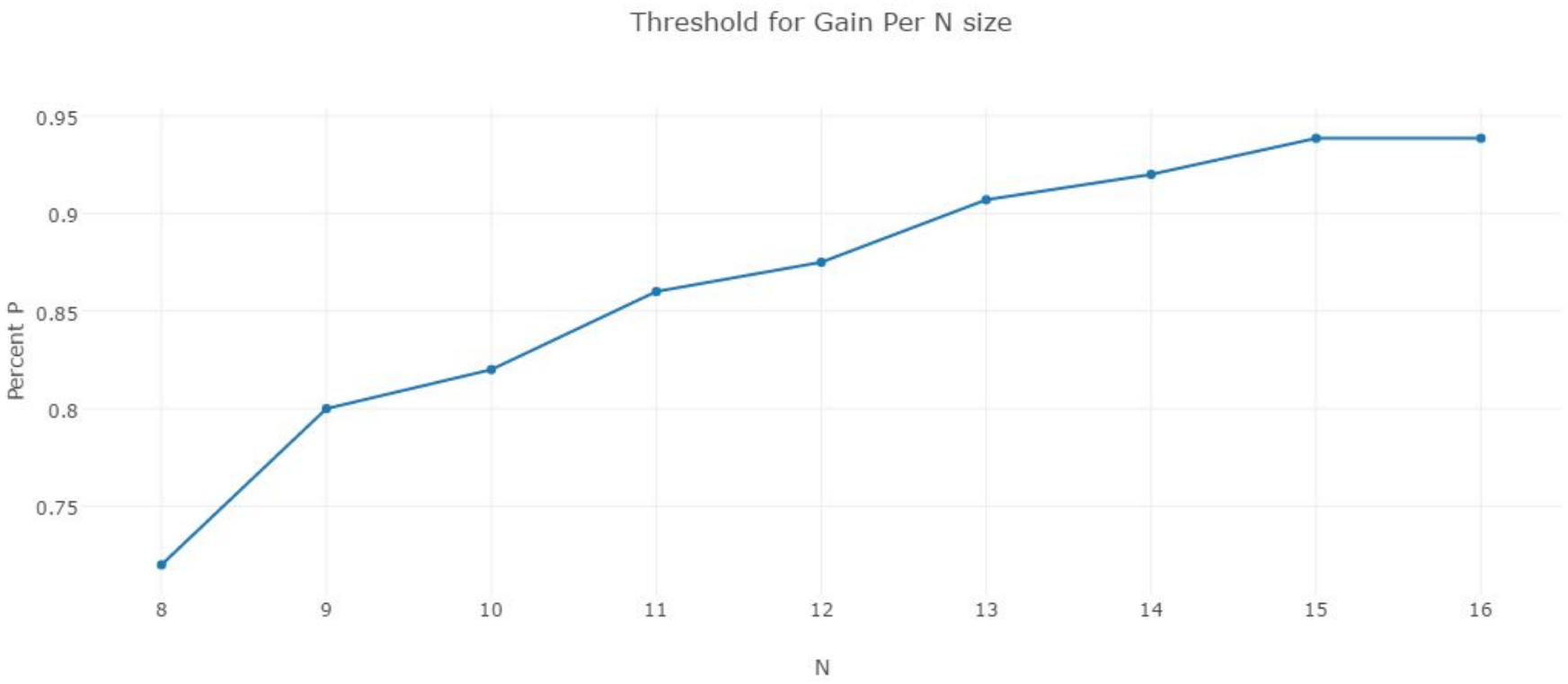

Gain of 10x10 Per P
Percent Gain
0
−50
−100
−150
−200
0
0.2
0.4
0.6
0.8
1
P

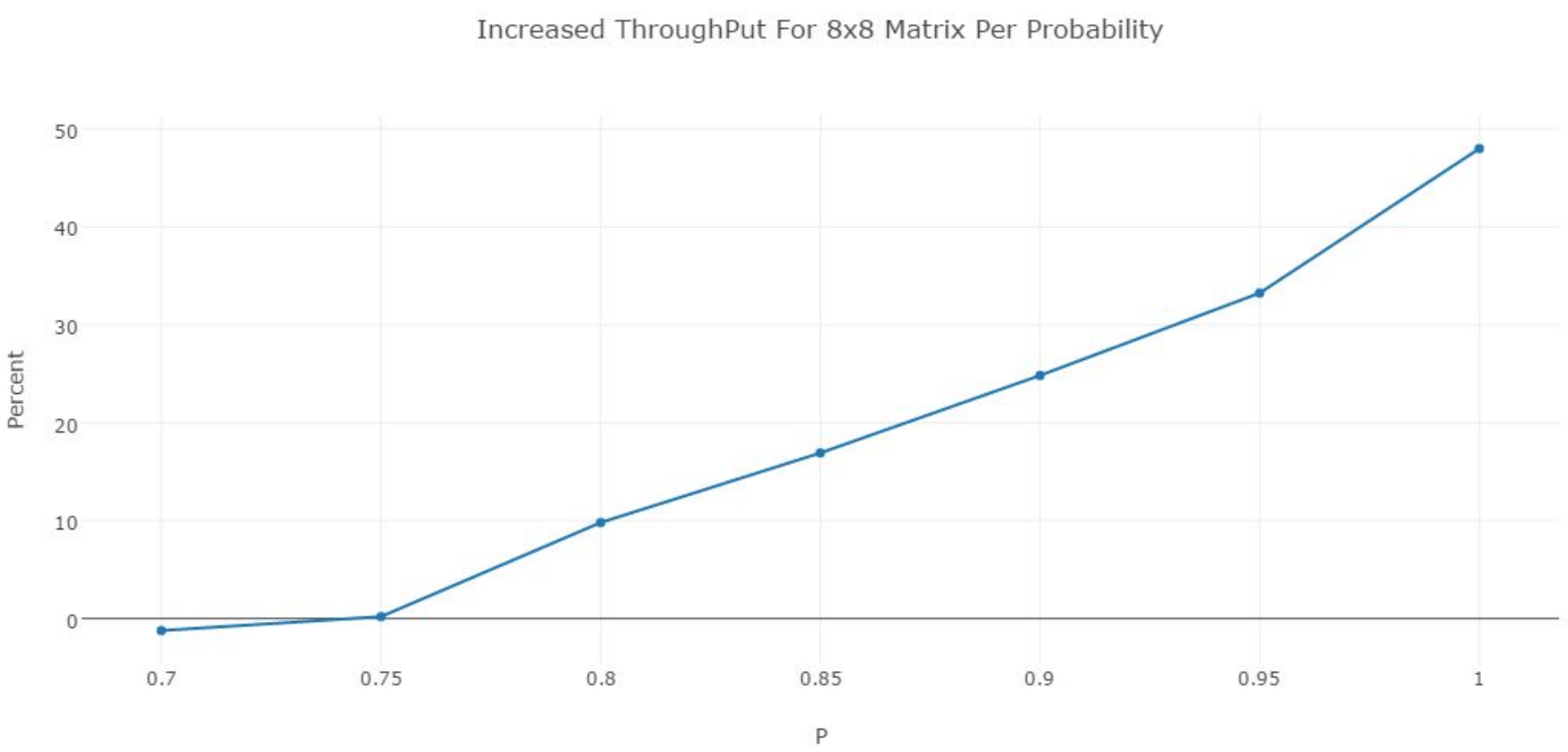

Gain of 14x14 Per P
Percent Gain
0
−50
−100
−150
−200
0
0.2
0.4
0.6
0.8
1
P

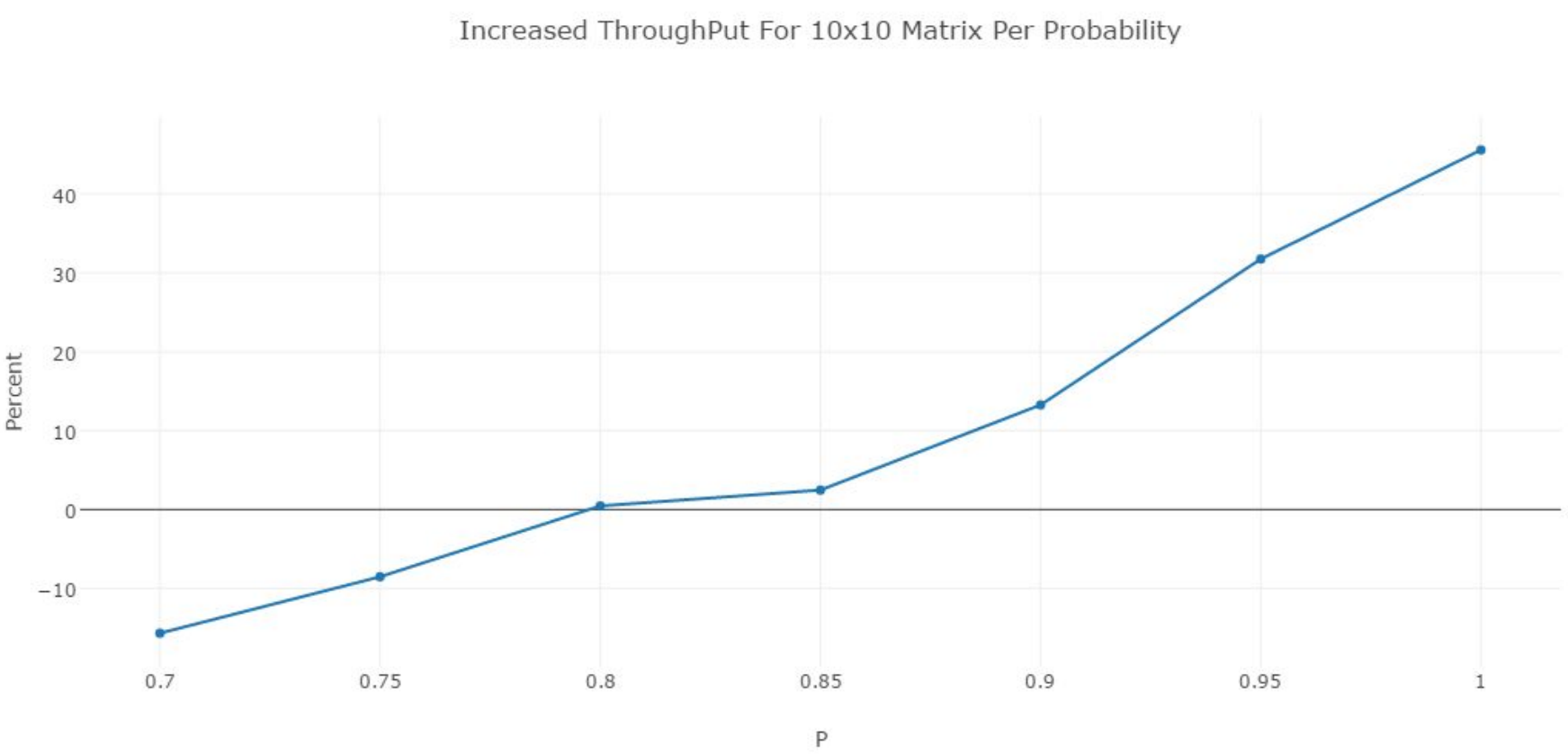
Testing Simulation Continued

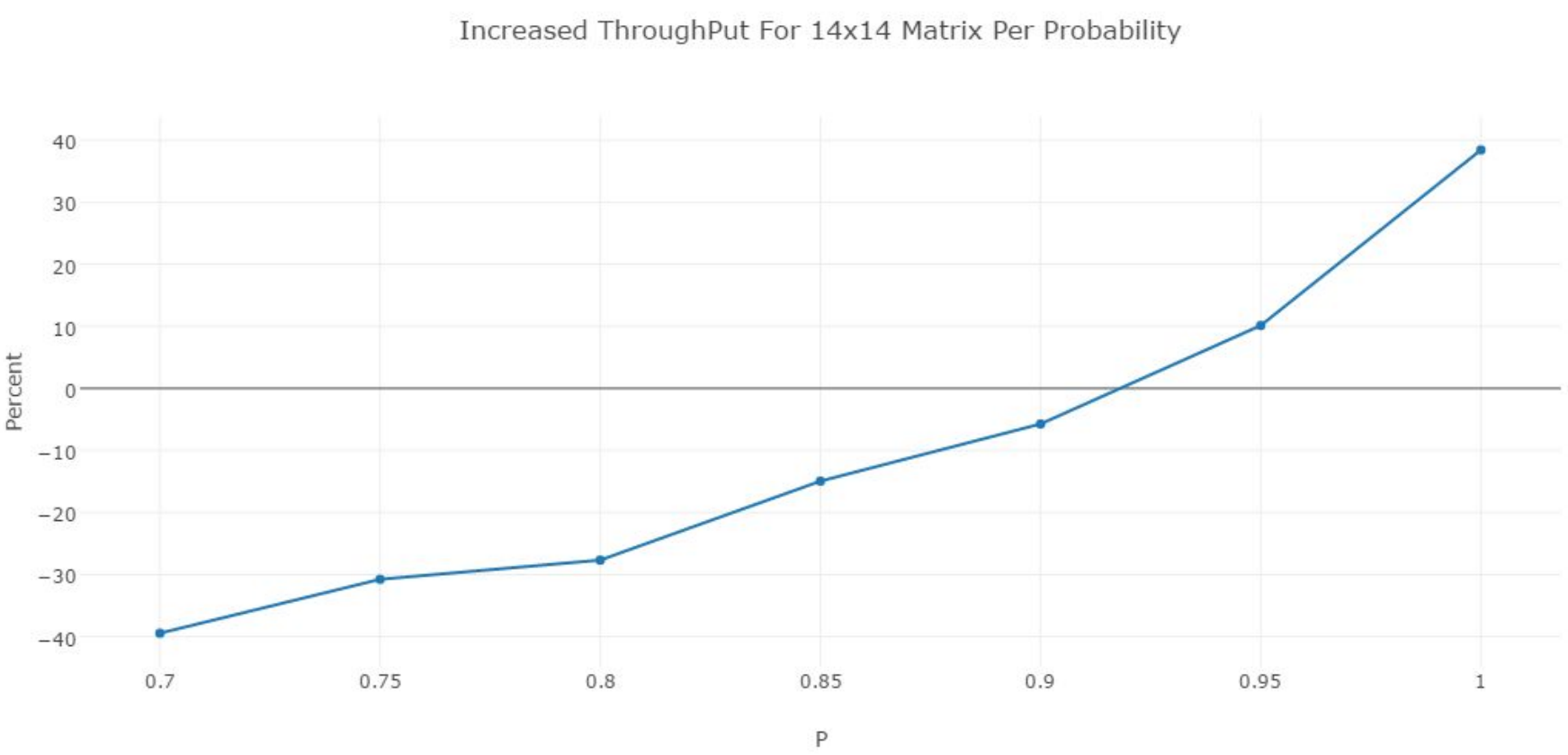
- Changed simulation to help find threshold of when APIC beat RR
- Found optimal probability of side information to yield lowest min rank for APIC
- Found second optimal probability of transmission chance to outperform RR
- For APIC to beat RR, P had to be different for initial message broadcast and X/M message broadcast

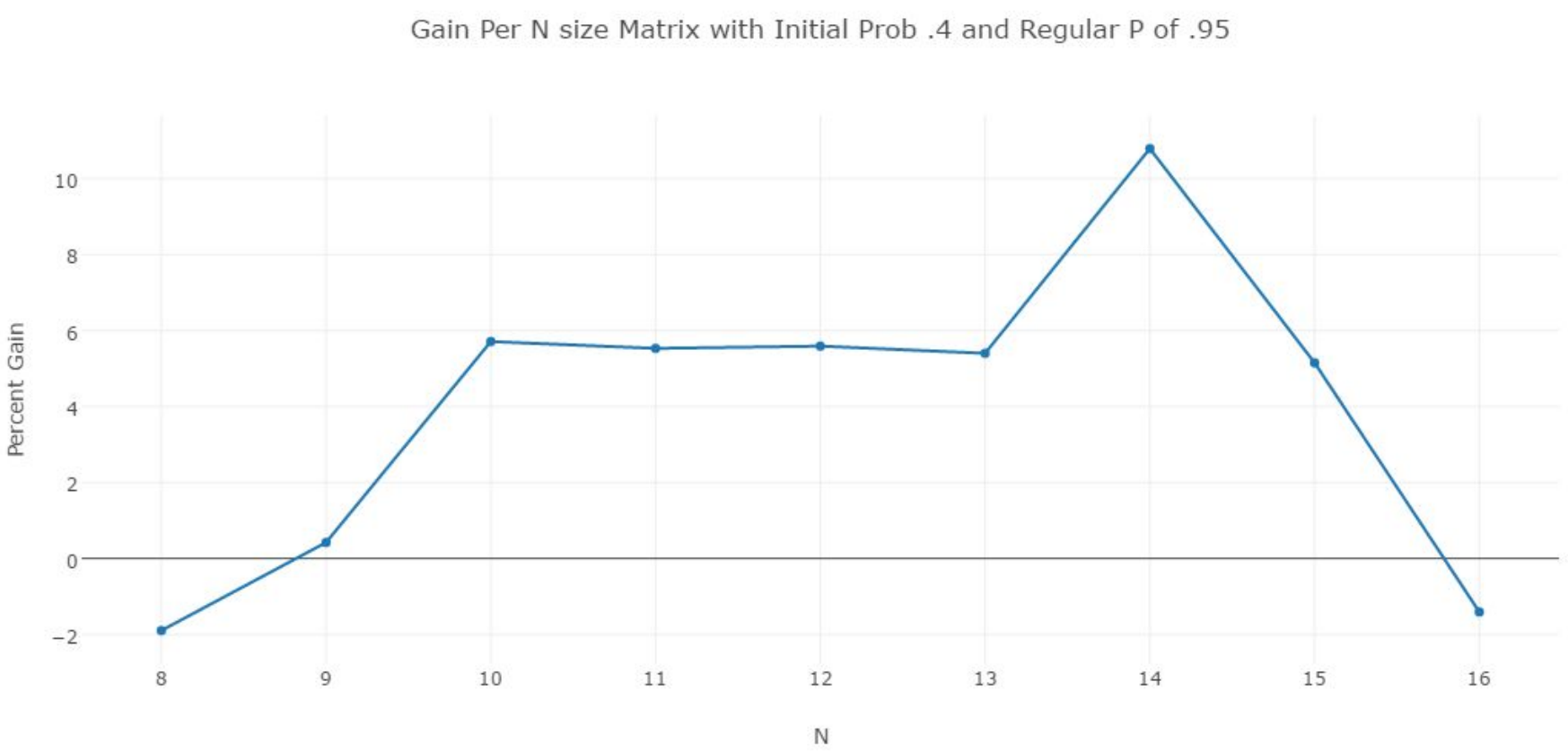

Optimal Initial P Per NxN sized Matrix
Initial P
0.8
0.7
0.6
0.5
0.4
8
9
10
11
12
13
14
15
16
N


Threshold for Gain Per N size
0.95
0.9
0.85
0.8
0.75
Percent P
8
9
10
11
12
13
14
15
16
N


Increased ThroughPut For 8x8 Matrix Per Probability
50
40
30
20
10
0
Percent
0.7
0.75
0.8
0.85
0.9
0.95
1
P


Increased ThroughPut For 10x10 Matrix Per Probability
Percent
40
30
20
10
0
−10
0.7
0.75
0.8
0.85
0.9
0.95
1
P


Increased ThroughPut For 14x14 Matrix Per Probability
Percent
40
30
20
10
0
−10
−20
−30
−40
0.7
0.75
0.8
0.85
0.9
0.95
1
P


Gain Per N size Matrix with Initial Prob .4 and Regular P of .95
Percent Gain
10
8
6
4
2
0
−2
8
9
10
11
12
13
14
15
16
N

Conclusion

- APIC can outperform RR by as much as 50% with certain conditions
- The Probability of initial side information has to be optimal for matrix size
- Probability of second wave of sending needs to be higher than .7
  - Higher for bigger matrices
- APIC with dirSVD does not work well for matrices larger than 16 with too much side information