

## # Correctness and Performance Charts

This version of the document is dated 2022-11-07.

The following charts show the correctness of many of the algorithms in "**Bernoulli Factory Algorithms**" and show their performance in terms of the number of bits they use on average. For each algorithm, and for each of 100  $\lambda$  values evenly spaced from 0.0001 to 0.9999:

- 500 runs of the algorithm were done. Then...
- The number of bits used by the runs were averaged, as were the return values of the runs (since the return value is either 0 or 1, the mean return value will be in the interval  $[0, 1]$ ). The number of bits used included the number of bits used to produce each coin flip, assuming the coin flip procedure for  $\lambda$  was generated using the `Bernoulli#coin()` method in *bernoulli.py*, which produces that probability in an optimal or near-optimal way.

For each algorithm, if a single run was detected to use more than 5000 bits for a given  $\lambda$ , the entire data point for that  $\lambda$  was suppressed in the charts below.

In addition, for each algorithm, a chart appears showing the minimum number of input coin flips that any fast Bernoulli factory algorithm will need on average to simulate the given function, based on work by Mendo (2019)[<sup>1</sup>]. Note that some functions require a growing number of coin flips as  $\lambda$  approaches 0 or 1. Note that for the 2014, 2016, and 2019 algorithms—

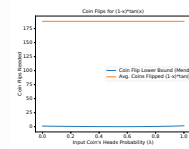
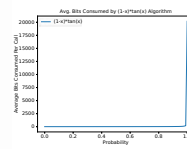
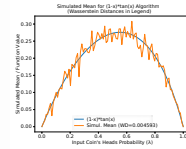
- an  $\epsilon$  of  $1 - (x + c) * 1.001$  was used (or 0.0001 if  $\epsilon$  would be greater than 1), and
- an  $\epsilon$  of  $(x - c) * 0.9995$  for the subtraction variants.

Points with invalid  $\epsilon$  values were suppressed. For the low-mean algorithm, an  $m$  of  $\max(0.49999, x*c*1.02)$  was used unless noted otherwise.

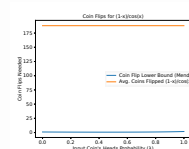
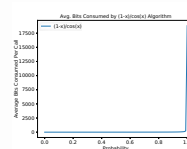
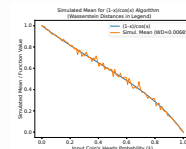
## 0.1 The Charts

Algorithm	Simulated Mean	Average Bits Consumed	Coin Flips
-----------	----------------	-----------------------	------------

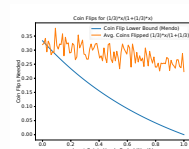
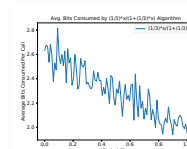
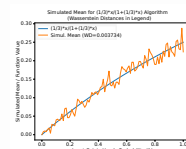
$$(1-x)*\tan(x)$$



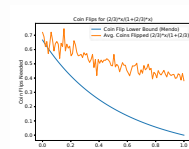
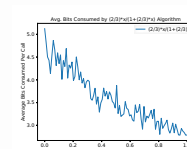
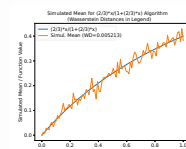
$$(1-x)/\cos(x)$$



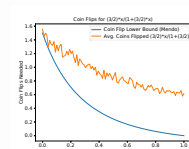
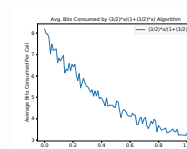
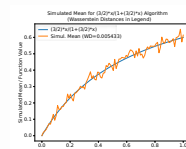
$$(1/3)*x/(1+(1/3)*x)$$



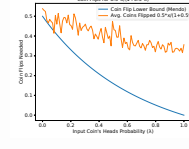
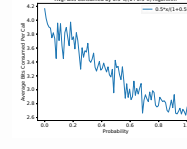
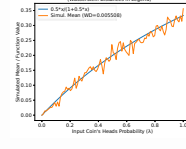
$$(2/3)*x/(1+(2/3)*x)$$



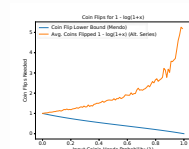
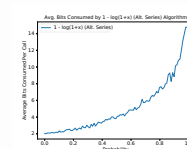
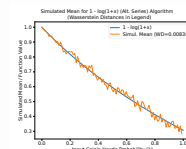
$$(3/2)*x/(1+(3/2)*x)$$



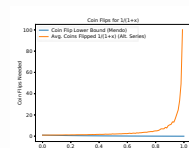
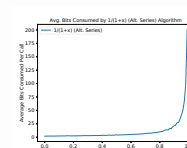
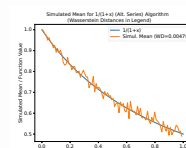
$$0.5*x/(1+0.5*x)$$



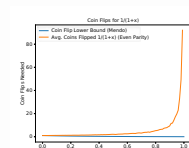
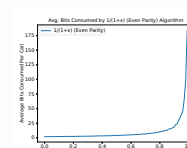
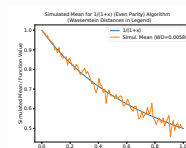
$$1 - \ln(1+x) \text{ (Alt. Series)}$$



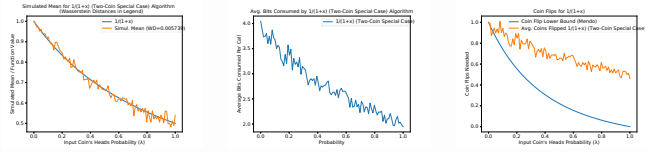
$$1/(1+x) \text{ (Alt. Series)}$$



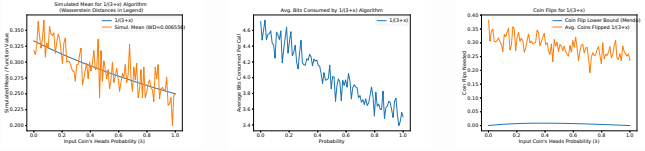
$$1/(1+x) \text{ (Even Parity)}$$



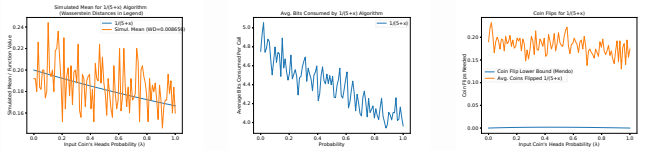
1/(1+x) (Two-Coin Special Case)



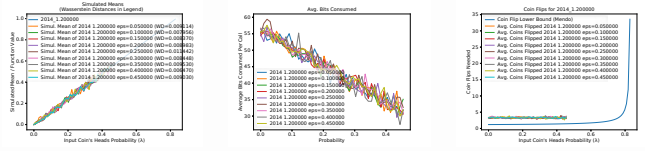
1/(3+x)



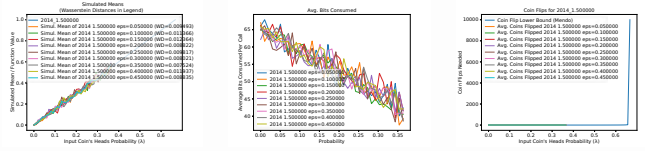
1/(5+x)



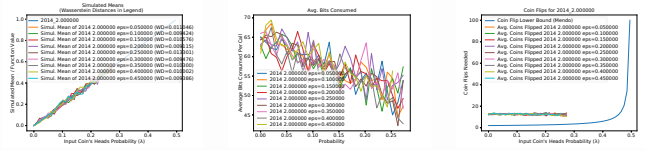
2014 1.200000  
eps=0.050000



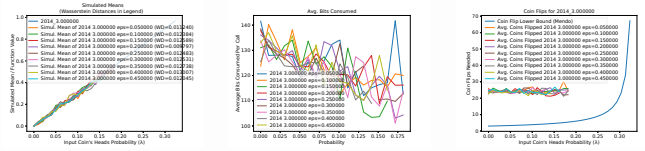
2014 1.500000  
eps=0.050000



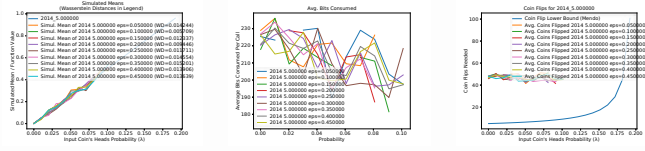
2014 2.000000  
eps=0.050000



2014 3.000000  
eps=0.050000



2014 5.000000  
eps=0.050000



2014 Add. x+0.1

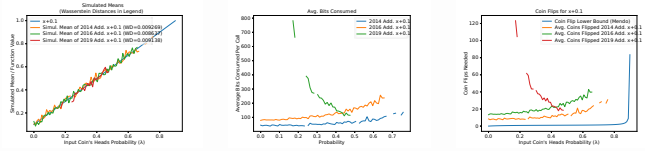


Figure 1 consists of three subplots. The left plot, titled 'Dispersed Values (Observations vs. Predictions)', shows True Values (mm) on the y-axis (0.0 to 1.0) versus Predicted Values (mm) on the x-axis (0.0 to 1.0). It includes data for 2014, 2015, and 2016, with a legend for 'v=2' and 'v=3'. The middle plot, titled 'Avg. RMSE (mm) vs. Normalized Error', shows Average RMSE (mm) on the y-axis (0.0 to 0.5) versus Normalized Error on the x-axis (0.0 to 0.6). It includes data for 2014, 2015, and 2016, with a legend for 'v=2' and 'v=3'. The right plot, titled 'Error Histogram for v=2', shows the number of errors on the y-axis (0 to 80) versus Error (mm) on the x-axis (0.0 to 0.7). It includes data for 2014, 2015, and 2016, with a legend for 'v=2' and 'v=3'.

[illegible][illegible]

Figure 1 consists of three subplots. Subplot (a) shows ROC curves for six datasets: Causal, Email, Enron, Higgs, Sift, and TREC. The x-axis is 'Input Data's Feature Probability (x)' and the y-axis is 'Model's ROC Curve (y)'. Subplot (b) shows 'Average ROC Curves for v1.5' for the same datasets, with the y-axis being 'Average ROC Curve for v1.5'. Subplot (c) shows the 'Cost-Peak Ratio' for v1.5, with the y-axis being 'Cost-Peak Ratio' and the x-axis being 'Input Data's Feature Probability (x)'. The legend for all plots is: Causal (blue), Email (orange), Enron (green), Higgs (red), Sift (purple), and TREC (brown).

Figure 1 consists of three subplots labeled (a), (b), and (c).

Subplot (a) is a scatter plot titled "Scatter Plot of Predicted vs. Actual Mean (Covariance)". The x-axis is "Input C<sub>1</sub> Mean Probability (%)" ranging from 0.0 to 0.5. The y-axis is "Predicted Mean (Covariance)" ranging from 0.0 to 1.0. Data points are colored circles representing different input mean probabilities: 0.0, 0.1, 0.2, 0.3, 0.4, and 0.5. A red diagonal line represents the ideal prediction. The points closely follow this line.

Subplot (b) is a line plot titled "Avg. Mean Squared Error (MSE) vs. Probability". The x-axis is "Probability" ranging from 0.0 to 0.5. The y-axis is "Average Mean Squared Error (MSE)" ranging from 0.0 to 0.05. Multiple lines represent different input mean probabilities: 0.0 (blue), 0.1 (orange), 0.2 (green), 0.3 (red), 0.4 (purple), and 0.5 (brown). The MSE generally decreases as the input mean probability increases.

Subplot (c) is a line plot titled "Cost Ratio vs. Probability". The x-axis is "Input C<sub>1</sub> Mean Probability (%)" ranging from 0.0 to 0.5. The y-axis is "Cost Ratio" ranging from 0 to 100. Multiple lines represent different input mean probabilities: 0.0 (blue), 0.1 (orange), 0.2 (green), 0.3 (red), 0.4 (purple), and 0.5 (brown). The cost ratio is high for low input mean probabilities and decreases as the input mean probability increases, with a sharp drop at 0.5.

Figure 1 consists of three subplots. Subplot (a) is a Receiver Operating Characteristic (ROC) curve showing the True Positive Rate (Y-axis) versus the False Positive Rate (X-axis) for various models. The models compared are: Proposed (blue line), Logistic (red line), SVM (green line), Naive Bayes (orange line), Decision Tree (purple line), Random Forest (brown line), and AdaBoost (pink line). The Proposed model shows the highest performance, with a True Positive Rate of approximately 0.95 at a False Positive Rate of 0.05. Subplot (b) is an Average ROC Curve showing the True Positive Rate (Y-axis) versus the False Positive Rate (X-axis) for different input data sets. The data sets are: Input Data Set 1 (blue line), Input Data Set 2 (red line), Input Data Set 3 (green line), Input Data Set 4 (orange line), Input Data Set 5 (purple line), Input Data Set 6 (brown line), and Input Data Set 7 (pink line). The Proposed model shows the highest performance, with a True Positive Rate of approximately 0.95 at a False Positive Rate of 0.05. Subplot (c) is a Confusion Matrix for the proposed model. The matrix shows the relationship between the predicted and actual classes. The diagonal elements represent correct classifications, while the off-diagonal elements represent misclassifications. The matrix is as follows:

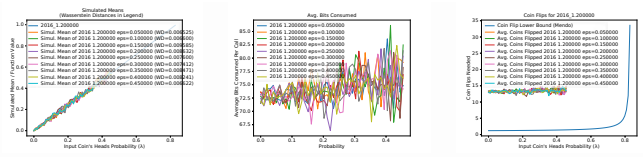
	Actual Class 1	Actual Class 2	Actual Class 3	Actual Class 4	Actual Class 5	Actual Class 6	Actual Class 7
Predicted Class 1	1000	10	5	2	1	0	0
Predicted Class 2	10	1000	5	2	1	0	0
Predicted Class 3	5	5	1000	2	1	0	0
Predicted Class 4	2	2	2	1000	1	0	0
Predicted Class 5	1	1	1	1	1000	0	0
Predicted Class 6	0	0	0	0	0	1000	0
Predicted Class 7	0	0	0	0	0	0	1000

The figure consists of three vertically stacked plots sharing common x-axis labels.

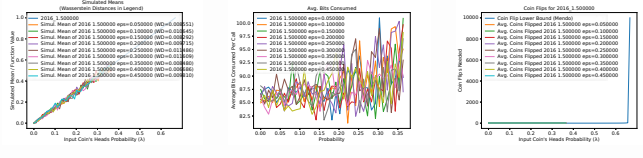
- Top Plot:** Y-axis is "Displaced Net Effective Mass". X-axis is "Input Core's mean Probability (%)" ranging from 0.00 to 0.15. Legend includes: Cofin Ltd. Reactor Based (blue), Proposed Method (red), Reg. Core Pignatelli 2010 v#6 (green), Reg. Core Pignatelli 2011 v#7 (purple), Reg. Core Pignatelli 2012 v#8 (orange), Low Heat Flux v#6 (brown), and Low Heat Flux + W-9 (pink).
- Middle Plot:** Y-axis is "Average Displacement Power (%)". X-axis is "Probability" ranging from 0.00 to 0.15. Legend is the same as the top plot.
- Bottom Plot:** Y-axis is "Core Mean Temperature (K)". X-axis is "Input Core's mean Probability (%)" ranging from 0.00 to 0.15. Legend is the same as the top plot.

[illegible]

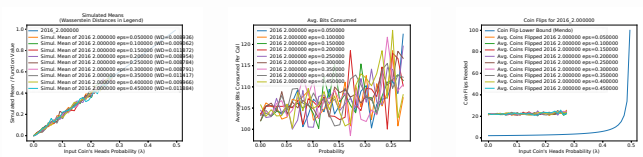
```
2016 1.200000
eps=0.050000
```



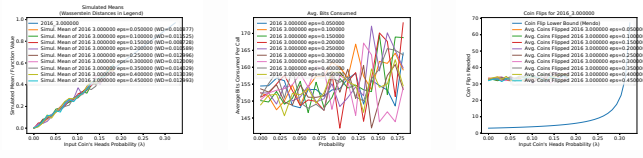
```
2016 1.500000
eps=0.050000
```



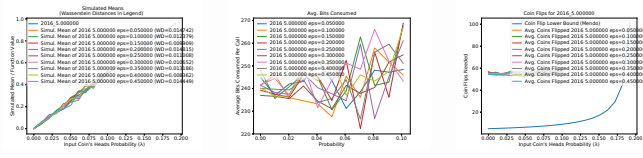
```
2016 2.000000
eps=0.050000
```



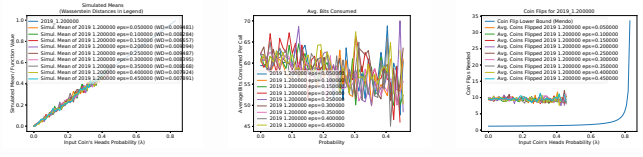
```
2016 3.000000
eps=0.050000
```



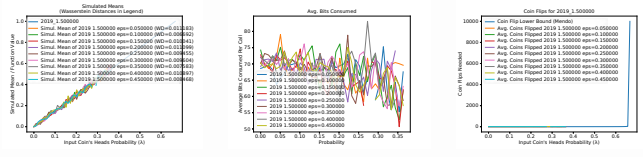
```
2016 5.000000
eps=0.050000
```



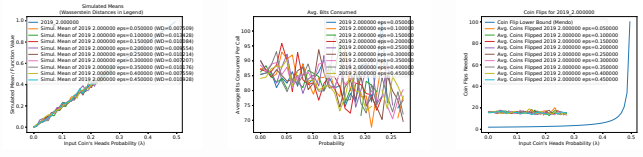
```
2019 1.200000
eps=0.050000
```



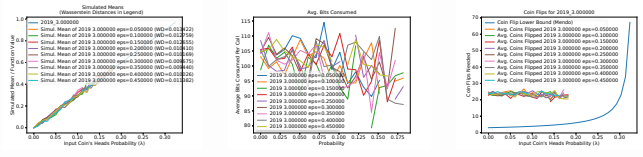
```
2019 1.500000
eps=0.050000
```



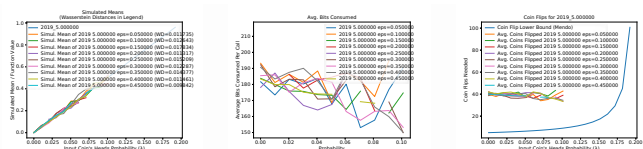
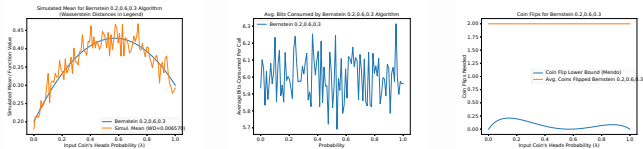
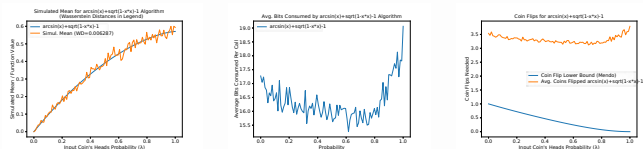
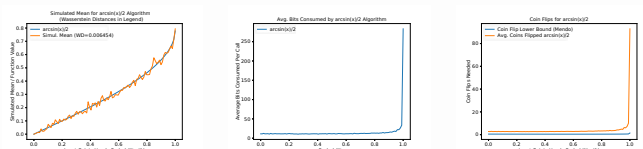
```
2019 2.000000
eps=0.050000
```



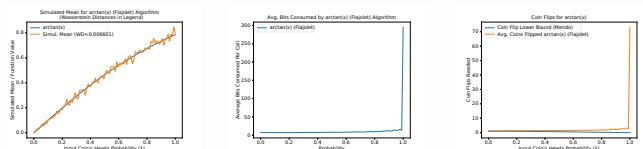
2019 3.000000  
eps=0.050000



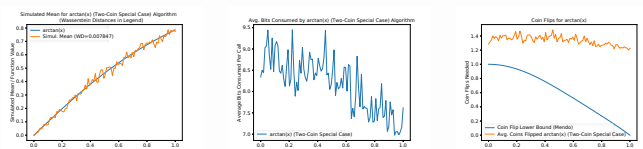
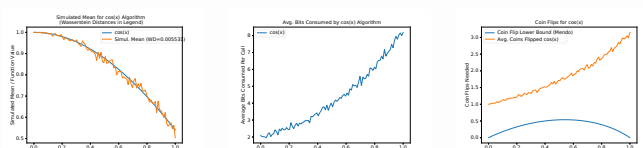
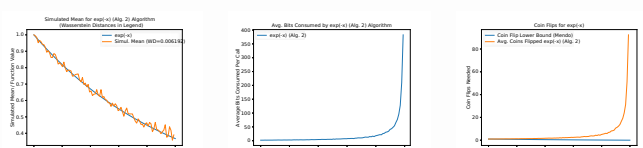
```
2019 5.000000
eps=0.050000
```

Bernstein  
0.2,0.6,0.3
$$\arcsin(x) + \sqrt{1-x^2} - 1$$
 $\arcsin(x)/2$ 

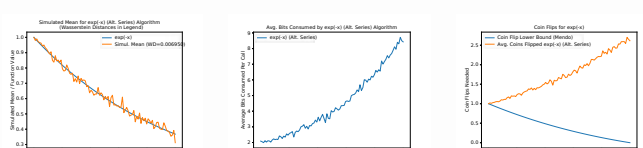
arctan(x)  
(Flajolet)



arctan(x) (Two-Coin Special Case)

 $\cos(x)$  $\exp(-x)$  (Alg. 2)

exp(-x) (Alt.  
Series)



Estimated Mean for each of (Proposed) Algorithm

Standard Error (times) Risk

Logarithmic Coefficient ( $\beta$ )

Exact Mean (0.001-0.005)

Prop. Mean (0.001-0.005)

Average Coefficient ( $\beta$ )

Logarithmic Coefficient ( $\beta$ )

CPU Time (sec)

Logarithmic Coefficient ( $\beta$ )

Exact Mean (0.001-0.005)

Prop. Mean (0.001-0.005)

Figure 10 consists of three subplots:

- Left Plot:** Scalability of the proposed algorithm. The x-axis is 'Input Data Size (Number of Nodes)' ranging from 0.0 to 1.0. The y-axis is 'Scalability of the Proposed Algorithm' ranging from 0.0 to 1.0. The plot shows two curves: 'Proposed' (blue line with circles) and 'Baseline' (orange line with circles). Both curves start at 1.0 and decrease as the input data size increases, with the proposed algorithm maintaining higher scalability than the baseline.
- Middle Plot:** Avg. Bits Consumed by the proposed algorithm. The x-axis is 'Input Data Size' ranging from 0.0 to 1.0. The y-axis is 'Average Bits Consumed' ranging from 0.0 to 1.0. The plot shows two curves: 'Proposed' (blue line with circles) and 'Baseline' (orange line with circles). Both curves start at 0.0 and increase as the input data size increases, with the proposed algorithm consuming fewer bits than the baseline.
- Right Plot:** Gap Error for input data size. The x-axis is 'Input Data Size (Number of Nodes)' ranging from 0.0 to 1.0. The y-axis is 'Gap Error' ranging from 0.0 to 1.0. The plot shows two curves: 'Proposed' (blue line with circles) and 'Baseline' (orange line with circles). Both curves start at 0.0 and increase as the input data size increases, with the proposed algorithm showing a lower gap error than the baseline.

Serialized Mean for  $\log_2(x+1)$  (Pigeon) Algorithm  
Serialized Mean for  $\log_2(x+1)$  (Proposed) Algorithm

Serialized Mean for  $\log_2(x+1)$  (Pigeon) Algorithm  
Serialized Mean for  $\log_2(x+1)$  (Proposed) Algorithm

Cost Ratio for  $\log_2(x+1)$

Cost-Flip Greedy  
Avg. Cost-Flip Greedy

Figure 1 consists of three subplots labeled (a), (b), and (c).  
 Subplot (a) is titled 'Scalability: Disabled flow capacity (kA) vs. Number of nodes'. The x-axis is 'Number of nodes' from 0.0 to 1.0. The y-axis is 'Disabled flow capacity (kA)' from 0.0 to 1.0. It shows two curves: 'points 1-10' (blue line with dots) and 'points 100-1000' (orange line with dots). Both curves start at (0,0) and increase monotonically, with the orange curve being slightly higher than the blue curve.  
 Subplot (b) is titled 'Convergence: Average flow capacity (kA) vs. Iteration number'. The x-axis is 'Iteration number' from 0.0 to 1.0. The y-axis is 'Average flow capacity (kA)' from 0.0 to 1000.0. It shows two curves: 'points 1-10' (blue line with dots) and 'points 100-1000' (orange line with dots). Both curves start at approximately 1000 kA and drop sharply to near zero by iteration 0.2, remaining there until iteration 1.0.  
 Subplot (c) is titled 'Error: Error (kA) vs. Iteration number'. The x-axis is 'Iteration number' from 0.0 to 1.0. The y-axis is 'Error (kA)' from 0 to 400. It shows two curves: 'points 1-10' (blue line with dots) and 'points 100-1000' (orange line with dots). Both curves start at approximately 400 kA and drop sharply to near zero by iteration 0.2, remaining there until iteration 1.0.

Figure 10 consists of three subplots comparing the performance of the proposed algorithm (grey line) with the state-of-the-art algorithm (orange line) for the  $p=213$  case. The x-axis for all plots is the Signal-to-noise ratio (SNR) ranging from 0.0 to 1.0.

- Left Plot:** Dryad Error ( $1/2 ||x - \hat{x}||_2^2$ ) vs. SNR. The y-axis ranges from 0.0 to 1.8. The legend indicates 'Proposed (213)' (grey line) and 'Oracle (Mean (0.000000))' (orange line). Both lines show an increasing trend, with the proposed algorithm's error being slightly lower than the oracle's error for SNR > 0.5.
- Middle Plot:** Average RMSE ( $||x - \hat{x}||_2$ ) vs. SNR. The y-axis ranges from 0.0 to 0.6. The legend indicates 'Prop. (213)' (grey line). The line shows a noisy increasing trend, starting around 0.25 at SNR=0.0 and reaching about 0.55 at SNR=1.0.
- Right Plot:** Case File Loss ( $||x - \hat{x}||_2$ ) vs. SNR. The y-axis ranges from 0.00 to 2.00. The legend indicates 'Case File Loss (Oracle (0.000000))' (orange line) and 'Prop. (213)' (grey line). The orange line is a straight line from (0,0) to (1,2). The grey line starts at approximately 1.4 at SNR=0.0 and increases to meet the orange line at SNR=1.0.

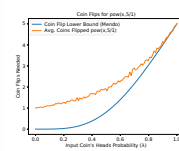
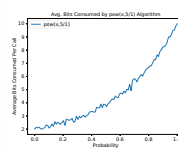
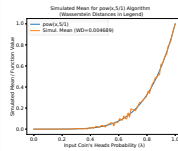
Figure 1 consists of three subplots showing the performance of the proposed algorithm on the 2014 dataset. The x-axis for all plots is 'SE' (Standard Error), ranging from 0.0 to 1.0.

- Left Subplot:** Titled 'Simulated SE for points 243-244 Algorithm'. The y-axis is 'Standard Error (SE) vs. SE'. It shows two curves: a blue line for 'points 243' and an orange line for 'points 244'. Both curves start at (0,0) and increase monotonically, reaching 1.0 at SE=1.0.
- Middle Subplot:** Titled 'Avg. SE Computed by points 243-244 Algorithm'. The y-axis is 'Average SE Computed by points 243-244 Algorithm'. It shows two curves: a blue line for 'points 243' and an orange line for 'points 244'. Both curves start at (0,1.0) and decrease monotonically, reaching 0.0 at SE=1.0.
- Right Subplot:** Titled 'CDF Plots for points 243-244'. The y-axis is 'CDF vs. SE'. It shows two curves: a blue line for 'points 243' and an orange line for 'points 244'. Both curves start at (0,1.0) and decrease monotonically, reaching 0.0 at SE=1.0.

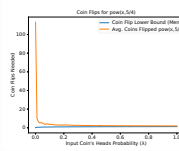
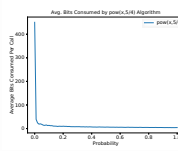
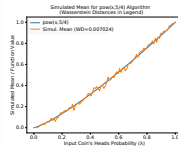
Figure 1 consists of three subplots. Subplot (a) is an ROC curve titled 'Stratified Mean for point-1, 2 Algorithm'. The x-axis is 'Specificity (True Negative Rate)' and the y-axis is 'Sensitivity (True Positive Rate)', both ranging from 0.0 to 1.0. Three curves are shown: 'point-1' (blue), 'point-2' (orange), and 'Avg. Mean' (red). All curves are very close to the top-left corner, indicating high performance. Subplot (b) is titled 'Avg. Bits Consumed by point-1, 2 Algorithm'. The x-axis is 'Probability' (0.0 to 1.0) and the y-axis is 'Average Bits per Symbol' (0.0 to 1.0). A single blue curve for 'point-1' starts at (0, 1) and decreases sharply to near zero by probability 0.2. Subplot (c) is titled 'Cost Flips for point-1, 2'. The x-axis is 'Reduced Count's Feature Dimension' (0.0 to 1.0) and the y-axis is 'Classification Error' (0.0 to 1.4). Three curves are shown: 'Avg. Cuts' (blue), 'Avg. Based' (orange), and 'Avg. Cuts Based point-1, 2' (red). The red curve is significantly lower than the others, indicating better performance.

Figure 1 consists of three subplots. Subplot (a) is titled 'Simulated Power by power-KS Algorithm' and shows 'Simulated Power of Power-KS' on the y-axis (0.0 to 1.0) versus 'Power-KS Algorithm' on the x-axis (0.0 to 1.0). It includes a legend for 'Power-KS Algorithm' with 'Power-KS (0.000000)' in blue and 'Power-KS (0.000000)' in orange. The orange line is slightly above the blue line. Subplot (b) is titled 'Avg. Size Computed by power-KS Algorithms' and shows 'Average Size of Power-KS' on the y-axis (0.0 to 4.0) versus 'Power-KS Algorithm' on the x-axis (0.0 to 1.0). It includes a legend for 'Power-KS' with 'Power-KS' in blue. The blue line starts at approximately 4.0 and decreases to approximately 1.0. Subplot (c) is titled 'Con. Flaps for power-KS' and shows 'Con. Flaps for Power-KS' on the y-axis (0.0 to 4.0) versus 'Power-KS Algorithm' on the x-axis (0.0 to 1.0). It includes a legend for 'Power-KS' with 'Power-KS (0.000000)' in blue and 'Power-KS (0.000000)' in orange. The orange line is slightly above the blue line.

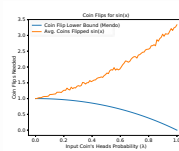
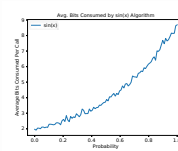
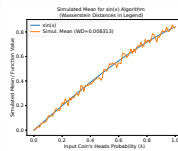
pow(x,5/1)



pow(x,5/4)



sin(x)



sqrt(x)

