

Correctness and Performance Charts

Peter Occil

1 Correctness and Performance Charts

This version of the document is dated 2025-01-27.

The following charts show the correctness of many of the algorithms in “**Bernoulli Factory Algorithms**¹” and show their performance in terms of the number of bits they use on average. For each algorithm, and for each of 100 λ 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 λ was generated using the `Bernoulli#coin()` method in *bernoulli.py*, which produces that probability in an optimal or near-optimal way.

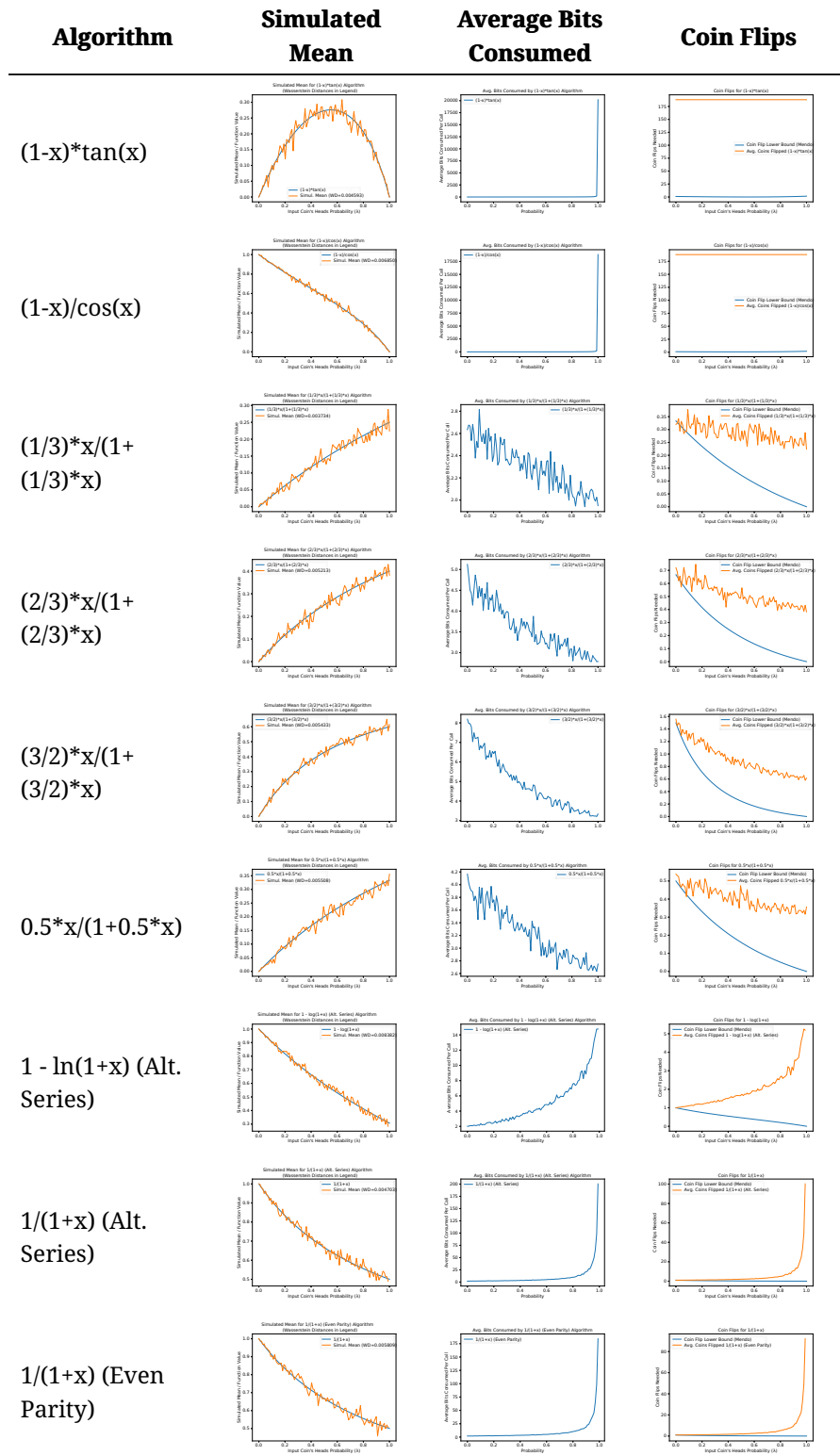
For each algorithm, if a single run was detected to use more than 5000 bits for a given λ , the entire data point for that λ was suppressed in the charts below.

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 specified function, based on work by Mendo (2019)^[^1]. Note that some functions require a growing number of coin flips as λ approaches 0 or 1. Note that for the 2014, 2016, and 2019 algorithms—

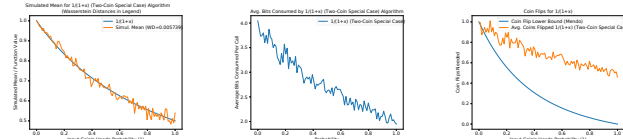
- an ϵ of $1 - (x + c) * 1.001$ was used (or 0.0001 if ϵ would be greater than 1), and
- an ϵ of $(x - c) * 0.9995$ for the subtraction variants.

Points with invalid ϵ values were suppressed. For the low-mean algorithm, an m of $\max(0.49999, xc1.02)$ was used unless noted otherwise.

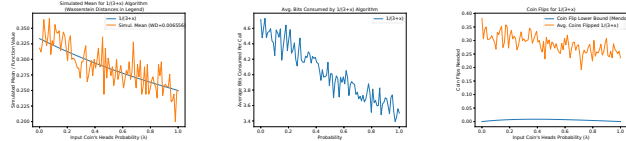
1.1 The Charts



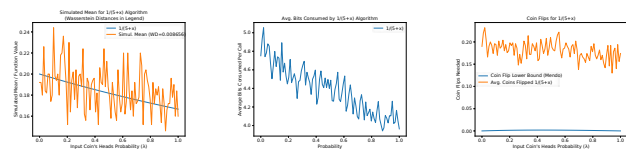
1/(1+x) (Two-Coin Special Case)



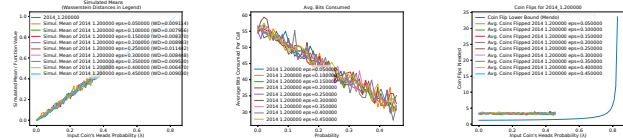
1/(3+x)



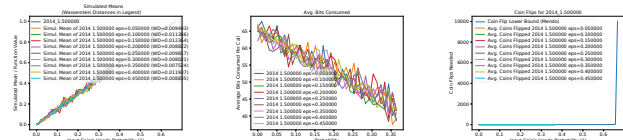
1/(5+x)



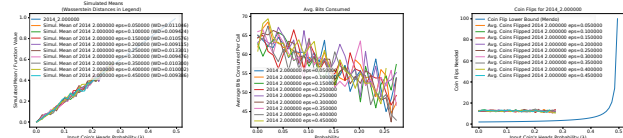
2014 1.200000
eps=0.050000



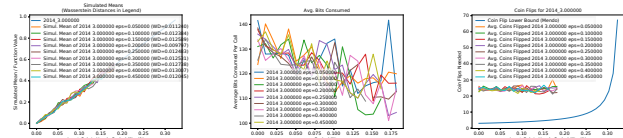
2014 1.500000
eps=0.050000



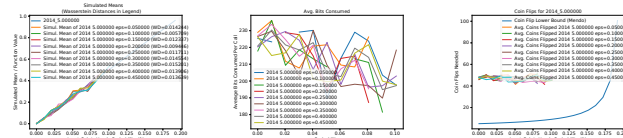
2014 2.000000
eps=0.050000



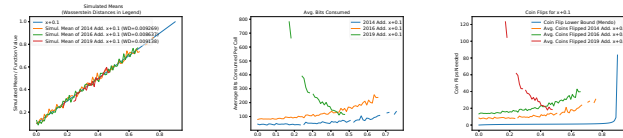
2014 3.000000
eps=0.050000



2014 5.000000
eps=0.050000



2014 Add. x+0.1



2014 Add. x+0.2

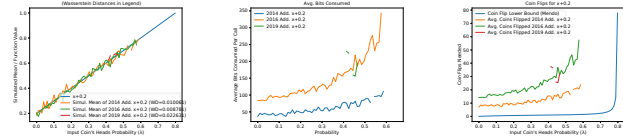


Figure 1 consists of three subplots. The left subplot, titled 'Simplest Model', shows the 'Simplest Model (Exponential = 0.000000)' and the 'Proposed Method (Exponential = 0.000000)'. The y-axis is 'Simplest Model / Proposed Method' and the x-axis is 'alpha'. The right subplot, titled 'Avg. Bias Computed', shows the 'Avg. Bias Computed' for the 'Proposed Method' (blue) and the 'State-of-the-Art' (orange) across different values of alpha. The y-axis is 'Avg. Bias Computed' and the x-axis is 'alpha'. The left subplot, titled 'Cost Ratio for alpha=0.3', shows the 'Cost Ratio for alpha=0.3' for the 'Proposed Method' (blue) and the 'State-of-the-Art' (orange) across different values of alpha. The y-axis is 'Cost Ratio for alpha=0.3' and the x-axis is 'alpha'.

Figure 1 consists of three subplots comparing the proposed method (red line) with other methods (various colored lines) across different metrics and frequency ranges.

Left Subplot: Data Normalized by Variance vs. Input Core-Channel Frequency (Hz)

This plot shows the normalized data for various methods. The x-axis ranges from 0.0 to 0.3 Hz, and the y-axis ranges from 0.0 to 1.0. The methods compared are:

- Prop. w/ σ^2
- Linear fit of 2014-1 w/ σ^2 (1980-2020)
- Linear fit of 2014-1 w/ σ^2 (2010-2020)
- Linear fit of 2014-1 w/ σ^2 (2010-2015)
- Linear fit of 2014-1 w/ σ^2 (2010-2012)
- Linear fit of 2014-1 w/ σ^2 (2010-2011)
- Linear fit of 2014-1 w/ σ^2 (2010-2010)
- Linear fit of 2014-1 w/ σ^2 (2010-2009)
- Linear fit of 2014-1 w/ σ^2 (2010-2008)
- Linear fit of 2014-1 w/ σ^2 (2010-2007)
- Linear fit of 2014-1 w/ σ^2 (2010-2006)
- Linear fit of 2014-1 w/ σ^2 (2010-2005)
- Linear fit of 2014-1 w/ σ^2 (2010-2004)
- Linear fit of 2014-1 w/ σ^2 (2010-2003)
- Linear fit of 2014-1 w/ σ^2 (2010-2002)
- Linear fit of 2014-1 w/ σ^2 (2010-2001)
- Linear fit of 2014-1 w/ σ^2 (2010-2000)
- Linear fit of 2014-1 w/ σ^2 (2010-1999)
- Linear fit of 2014-1 w/ σ^2 (2010-1998)
- Linear fit of 2014-1 w/ σ^2 (2010-1997)
- Linear fit of 2014-1 w/ σ^2 (2010-1996)
- Linear fit of 2014-1 w/ σ^2 (2010-1995)
- Linear fit of 2014-1 w/ σ^2 (2010-1994)
- Linear fit of 2014-1 w/ σ^2 (2010-1993)
- Linear fit of 2014-1 w/ σ^2 (2010-1992)
- Linear fit of 2014-1 w/ σ^2 (2010-1991)
- Linear fit of 2014-1 w/ σ^2 (2010-1990)
- Linear fit of 2014-1 w/ σ^2 (2010-1989)
- Linear fit of 2014-1 w/ σ^2 (2010-1988)
- Linear fit of 2014-1 w/ σ^2 (2010-1987)
- Linear fit of 2014-1 w/ σ^2 (2010-1986)
- Linear fit of 2014-1 w/ σ^2 (2010-1985)
- Linear fit of 2014-1 w/ σ^2 (2010-1984)
- Linear fit of 2014-1 w/ σ^2 (2010-1983)
- Linear fit of 2014-1 w/ σ^2 (2010-1982)
- Linear fit of 2014-1 w/ σ^2 (2010-1981)
- Linear fit of 2014-1 w/ σ^2 (2010-1980)
- Linear fit of 2014-1 w/ σ^2 (2010-1979)
- Linear fit of 2014-1 w/ σ^2 (2010-1978)
- Linear fit of 2014-1 w/ σ^2 (2010-1977)
- Linear fit of 2014-1 w/ σ^2 (2010-1976)
- Linear fit of 2014-1 w/ σ^2 (2010-1975)
- Linear fit of 2014-1 w/ σ^2 (2010-1974)
- Linear fit of 2014-1 w/ σ^2 (2010-1973)
- Linear fit of 2014-1 w/ σ^2 (2010-1972)
- Linear fit of 2014-1 w/ σ^2 (2010-1971)
- Linear fit of 2014-1 w/ σ^2 (2010-1970)
- Linear fit of 2014-1 w/ σ^2 (2010-1969)
- Linear fit of 2014-1 w/ σ^2 (2010-1968)
- Linear fit of 2014-1 w/ σ^2 (2010-1967)
- Linear fit of 2014-1 w/ σ^2 (2010-1966)
- Linear fit of 2014-1 w/ σ^2 (2010-1965)
- Linear fit of 2014-1 w/ σ^2 (2010-1964)
- Linear fit of 2014-1 w/ σ^2 (2010-1963)
- Linear fit of 2014-1 w/ σ^2 (2010-1962)
- Linear fit of 2014-1 w/ σ^2 (2010-1961)
- Linear fit of 2014-1 w/ σ^2 (2010-1960)
- Linear fit of 2014-1 w/ σ^2 (2010-1959)
- Linear fit of 2014-1 w/ σ^2 (2010-1958)
- Linear fit of 2014-1 w/ σ^2 (2010-1957)
- Linear fit of 2014-1 w/ σ^2 (2010-1956)
- Linear fit of 2014-1 w/ σ^2 (2010-1955)
- Linear fit of 2014-1 w/ σ^2 (2010-1954)
- Linear fit of 2014-1 w/ σ^2 (2010-1953)
- Linear fit of 2014-1 w/ σ^2 (2010-1952)
- Linear fit of 2014-1 w/ σ^2 (2010-1951)
- Linear fit of 2014-1 w/ σ^2 (2010-1950)
- Linear fit of 2014-1 w/ σ^2 (2010-1949)
- Linear fit of 2014-1 w/ σ^2 (2010-1948)
- Linear fit of 2014-1 w/ σ^2 (2010-1947)
- Linear fit of 2014-1 w/ σ^2 (2010-1946)
- Linear fit of 2014-1 w/ σ^2 (2010-1945)
- Linear fit of 2014-1 w/ σ^2 (2010-1944)
- Linear fit of 2014-1 w/ σ^2 (2010-1943)
- Linear fit of 2014-1 w/ σ^2 (2010-1942)
- Linear fit of 2014-1 w/ σ^2 (2010-1941)
- Linear fit of 2014-1 w/ σ^2 (2010-1940)
- Linear fit of 2014-1 w/ σ^2 (2010-1939)
- Linear fit of 2014-1 w/ σ^2 (2010-1938)
- Linear fit of 2014-1 w/ σ^2 (2010-1937)
- Linear fit of 2014-1 w/ σ^2 (2010-1936)
- Linear fit of 2014-1 w/ σ^2 (2010-1935)
- Linear fit of 2014-1 w/ σ^2 (2010-1934)
- Linear fit of 2014-1 w/ σ^2 (2010-1933)
- Linear fit of 2014-1 w/ σ^2 (2010-1932)
- Linear fit of 2014-1 w/ σ^2 (2010-1931)
- Linear fit of 2014-1 w/ σ^2 (2010-1930)
- Linear fit of 2014-1 w/ σ^2 (2010-1929)
- Linear fit of 2014-1 w/ σ^2 (2010-1928)
- Linear fit of 2014-1 w/ σ^2 (2010-1927)
- Linear fit of 2014-1 w/ σ^2 (2010-1926)
- Linear fit of 2014-1 w/ σ^2 (2010-1925)
- Linear fit of 2014-1 w/ σ^2 (2010-1924)
- Linear fit of 2014-1 w/ σ^2 (2010-1923)
- Linear fit of 2014-1 w/ σ^2 (2010-1922)
- Linear fit of 2014-1 w/ σ^2 (2010-1921)
- Linear fit of 2014-1 w/ σ^2 (2010-1920)
- Linear fit of 2014-1 w/ σ^2 (2010-1919)
- Linear fit of 2014-1 w/ σ^2 (2010-1918)
- Linear fit of 2014-1 w/ σ^2 (2010-1917)
- Linear fit of 2014-1 w/ σ^2 (2010-1916)
- Linear fit of 2014-1 w/ σ^2 (2010-1915)
- Linear fit of 2014-1 w/ σ^2 (2010-1914)
- Linear fit of 2014-1 w/ σ^2 (2010-1913)

Figure 1 consists of four subplots (a, b, c, d) showing the performance of the proposed algorithm. Subplot (a) is a Receiver Operating Characteristic (ROC) curve for the proposed algorithm, plotting True Positive Rate (TPR) on the y-axis (0.0 to 1.0) against False Positive Rate (FPR) on the x-axis (0.0 to 1.0). The curve is a solid blue line, showing a strong performance. Subplot (b) is an Average ROC curve for the proposed algorithm, plotting Average True Positive Rate (ATPR) on the y-axis (0.0 to 1.0) against Average False Positive Rate (AFPR) on the x-axis (0.0 to 1.0). The curve is a solid blue line, showing a strong performance. Subplot (c) is an Average ROC curve for the proposed algorithm, plotting Average True Positive Rate (ATPR) on the y-axis (0.0 to 1.0) against Average False Positive Rate (AFPR) on the x-axis (0.0 to 1.0). The curve is a solid blue line, showing a strong performance. Subplot (d) is an Average ROC curve for the proposed algorithm, plotting Average True Positive Rate (ATPR) on the y-axis (0.0 to 1.0) against Average False Positive Rate (AFPR) on the x-axis (0.0 to 1.0). The curve is a solid blue line, showing a strong performance.

[illegible]

Figure 1 consists of three subplots, (a), (b), and (c), each showing the Comparison of the proposed model with other models. The x-axis for all plots is 'Input CAGR's means Probability (x)' ranging from 0.00 to 1.00. The y-axis for all plots is 'CAGR Prob'.

Subplot (a) Dry Weather (CAGR vs. Prob). The y-axis ranges from 0.00 to 0.20. The legend includes: Prop., 2014-16, 2014-18, 2014-20, Low Water, High Water, Low Water, High Water, and Low Water, High Water. The proposed model (Prop.) is a solid black line that closely follows the other models, which are represented by various colored lines.

Subplot (b) Wet Weather (CAGR vs. Prob). The y-axis ranges from 0.00 to 0.40. The legend includes: Prop., 2014-16, 2014-18, 2014-20, Low Water, High Water, Low Water, High Water, and Low Water, High Water. The proposed model (Prop.) is a solid black line that shows a significant increase in CAGR Prob as the probability increases, reaching approximately 0.35 at a probability of 1.00.

Subplot (c) CAGR Prob (CAGR vs. Prob). The y-axis ranges from 0.00 to 1.00. The legend includes: Prop., 2014-16, 2014-18, 2014-20, Low Water, High Water, Low Water, High Water, and Low Water, High Water. The proposed model (Prop.) is a solid black line that shows a significant increase in CAGR Prob as the probability increases, reaching approximately 0.85 at a probability of 1.00.

Figure 10 consists of three plots showing the effect of input CNN's mean probability (ξ) on the output of the proposed model. The x-axis for all plots is 'Input CNN's mean probability (ξ)' ranging from 0.00 to 0.60. The y-axis for all plots is 'Output of the proposed model' on a log scale.

- Left Plot:** Shows the output of the proposed model for various models. The y-axis ranges from 10^0 to 10^2 . The models are:
 - Prop. E
 - Mean of 200 Trials (w/o μ) (0.000000-0.000002)
 - Mean of 200 Trials (w/ μ) (0.000000-0.000002)
 - Mean of 200 Trials (w/o μ) (0.000000-0.000002)
 - Mean of 200 Trials (w/ μ) (0.000000-0.000002)
 - Mean of 200 Trials (w/o μ) (0.000000-0.000002)
 - Mean of 200 Trials (w/ μ) (0.000000-0.000002)
 - Mean of 200 Trials (w/o μ) (0.000000-0.000002)
 - Mean of 200 Trials (w/ μ) (0.000000-0.000002)
- Middle Plot:** Shows the output of the proposed model for various models. The y-axis ranges from 10^0 to 10^2 . The models are:
 - Prop. E
 - Mean of 200 Trials (w/o μ) (0.000000-0.000002)
 - Mean of 200 Trials (w/ μ) (0.000000-0.000002)
 - Mean of 200 Trials (w/o μ) (0.000000-0.000002)
 - Mean of 200 Trials (w/ μ) (0.000000-0.000002)
 - Mean of 200 Trials (w/o μ) (0.000000-0.000002)
 - Mean of 200 Trials (w/ μ) (0.000000-0.000002)
 - Mean of 200 Trials (w/o μ) (0.000000-0.000002)
 - Mean of 200 Trials (w/ μ) (0.000000-0.000002)
- Right Plot:** Shows the output of the proposed model for various models. The y-axis ranges from 10^0 to 10^2 . The models are:
 - Prop. E
 - Mean of 200 Trials (w/o μ) (0.000000-0.000002)
 - Mean of 200 Trials (w/ μ) (0.000000-0.000002)
 - Mean of 200 Trials (w/o μ) (0.000000-0.000002)
 - Mean of 200 Trials (w/ μ) (0.000000-0.000002)
 - Mean of 200 Trials (w/o μ) (0.000000-0.000002)
 - Mean of 200 Trials (w/ μ) (0.000000-0.000002)
 - Mean of 200 Trials (w/o μ) (0.000000-0.000002)
 - Mean of 200 Trials (w/ μ) (0.000000-0.000002)

[illegible][illegible]

Figure 10 displays the performance of the proposed model for the 2014 and 2015 datasets, comparing the proposed model (Proposed) against the baseline (Baseline) across various metrics.

The figure is organized into three rows, each corresponding to a different dataset: 2014, 2015, and 2016. Each row contains three subplots:

- Top Row (2014):**
 - Spilloverd Means (Histograms: Correlation vs. Argument):** A scatter plot showing the relationship between Input Cor'n Weak Probability (a) and Spilloverd Means. The y-axis ranges from 0.0 to 1.0, and the x-axis ranges from 0.0 to 0.5. The plot shows a strong positive correlation.
 - Avg. Bits Consumed:** A line plot showing the average bits consumed (y-axis, 0 to 100) versus Probability (x-axis, 0.0 to 0.5). The plot shows a general upward trend in bits consumed as probability increases.
 - Cor'n Bits for 2014, 100000:** A line plot showing the correlation bits (y-axis, 0 to 1000) versus Input Cor'n Weak Probability (a) (x-axis, 0.0 to 0.5). The plot shows a sharp increase in correlation bits as probability increases.
- Middle Row (2015):**
 - Spilloverd Means (Histograms: Correlation vs. Argument):** A scatter plot showing the relationship between Input Cor'n Weak Probability (a) and Spilloverd Means. The y-axis ranges from 0.0 to 1.0, and the x-axis ranges from 0.0 to 0.5. The plot shows a strong positive correlation.
 - Avg. Bits Consumed:** A line plot showing the average bits consumed (y-axis, 0 to 120) versus Probability (x-axis, 0.0 to 0.5). The plot shows a general upward trend in bits consumed as probability increases.
 - Cor'n Bits for 2015, 100000:** A line plot showing the correlation bits (y-axis, 0 to 100) versus Input Cor'n Weak Probability (a) (x-axis, 0.0 to 0.5). The plot shows a sharp increase in correlation bits as probability increases.
- Bottom Row (2016):**
 - Spilloverd Means (Histograms: Correlation vs. Argument):** A scatter plot showing the relationship between Input Cor'n Weak Probability (a) and Spilloverd Means. The y-axis ranges from 0.0 to 1.0, and the x-axis ranges from 0.0 to 0.5. The plot shows a strong positive correlation.
 - Avg. Bits Consumed:** A line plot showing the average bits consumed (y-axis, 0 to 120) versus Probability (x-axis, 0.0 to 0.5). The plot shows a general upward trend in bits consumed as probability increases.
 - Cor'n Bits for 2016, 100000:** A line plot showing the correlation bits (y-axis, 0 to 100) versus Input Cor'n Weak Probability (a) (x-axis, 0.0 to 0.5). The plot shows a sharp increase in correlation bits as probability increases.

The legend for all plots indicates the following series:

- Proposed (Blue line)
- Baseline (Red line)
- Proposed (Green line)
- Baseline (Yellow line)
- Proposed (Cyan line)
- Baseline (Magenta line)
- Proposed (Dark Blue line)
- Baseline (Light Blue line)
- Proposed (Black line)
- Baseline (Grey line)

Figure 10 consists of four subplots arranged in a 2x2 grid, showing the performance of the proposed model for the Hepatitis B dataset. The top row shows results for the 'Hepatitis B (Hepatitis B is negative)' class, and the bottom row shows results for the 'Hepatitis B (Hepatitis B is positive)' class. The left column contains scatter plots of Actual vs. Predicted values, and the right column contains line plots of Average F1 Score and Cost vs. Probability.

- Top-left (Scatter Plot):** Shows Actual vs. Predicted values for 2014-2020. The x-axis is 'Input C49's mean's Probability (%)' and the y-axis is 'Actual C49's mean's Probability (%)'. The plot shows a strong positive correlation between the predicted and actual values.
- Top-right (Line Plot):** Shows Average F1 Score vs. Probability for 2014-2020. The x-axis is 'Probability' and the y-axis is 'Average F1 Score (C49's mean's Probability)'. The plot shows the F1 score fluctuating between approximately 160 and 190 across the probability range.
- Bottom-left (Scatter Plot):** Shows Actual vs. Predicted values for 2014-2020. The x-axis is 'Input C49's mean's Probability (%)' and the y-axis is 'Actual C49's mean's Probability (%)'. The plot shows a strong positive correlation between the predicted and actual values.
- Bottom-right (Line Plot):** Shows Cost vs. Probability for 2014-2020. The x-axis is 'Input C49's mean's Probability (%)' and the y-axis is 'Cost (C49's mean's Probability)'. The plot shows a sharp increase in cost as the probability increases, reaching a maximum of approximately 1000 at a probability of 0.95.

Figure 1 displays nine plots arranged in a 3x3 grid, showing the performance of the proposed model across different CARs (0.00000, 0.00000, 0.00000). The plots are organized into three rows and three columns.

- Top Row:** Calibration Error vs. Input CAR's Net Profit Probability. The y-axis ranges from 0.00 to 1.00. The x-axis ranges from 0.000 to 0.200. The legend indicates 'Simplified Net Profit' (blue line) and 'Calibration Error' (orange line).
- Middle Row:** Calibration Error vs. Probability. The y-axis ranges from 0.00 to 1.00. The x-axis ranges from 0.00 to 1.00. The legend indicates 'Simplified Net Profit' (blue line) and 'Calibration Error' (orange line).
- Bottom Row:** Calibration Error vs. Input CAR's Net Profit Probability. The y-axis ranges from 0.00 to 1.00. The x-axis ranges from 0.000 to 0.200. The legend indicates 'Simplified Net Profit' (blue line) and 'Calibration Error' (orange line).

The plots show that the proposed model's performance is consistent across different CARs, with the calibration error remaining low (near 0.00) across the range of input probabilities.

Figure 1 displays six plots related to the 2019 COVID-19 dataset, organized into three rows and two columns. The top row shows 'GenStatistical Measures' (Histograms, Scatter Plot, and Correlation Matrix). The middle row shows 'Avg. All Compartment' (Histograms, Scatter Plot, and Correlation Matrix). The bottom row shows 'Avg. All Compartment' (Histograms, Scatter Plot, and Correlation Matrix). Each plot includes a legend for the 2019 COVID-19 dataset.

Top Row: GenStatistical Measures

- Histograms (Difference in Legend):** Shows the distribution of 'Input Case Incidence Probability (%)' for the 2019 COVID-19 dataset. The x-axis ranges from 0.00 to 0.35, and the y-axis ranges from 0.0 to 1.0.
- Scatter Plot:** Shows the relationship between 'Input Case Incidence Probability (%)' (x-axis, 0.00 to 0.35) and 'Output Case Incidence Probability (%)' (y-axis, 0.0 to 1.0). The data points are clustered along the diagonal line.
- Correlation Matrix:** Shows the correlation between 'Input Case Incidence Probability (%)' and 'Output Case Incidence Probability (%)'. The correlation is 1.000000.

Middle Row: Avg. All Compartment

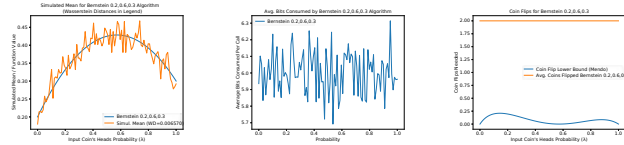
- Histograms (Difference in Legend):** Shows the distribution of 'Input Case Incidence Probability (%)' for the 2019 COVID-19 dataset. The x-axis ranges from 0.00 to 0.35, and the y-axis ranges from 0.0 to 1.0.
- Scatter Plot:** Shows the relationship between 'Input Case Incidence Probability (%)' (x-axis, 0.00 to 0.35) and 'Output Case Incidence Probability (%)' (y-axis, 0.0 to 1.0). The data points are clustered along the diagonal line.
- Correlation Matrix:** Shows the correlation between 'Input Case Incidence Probability (%)' and 'Output Case Incidence Probability (%)'. The correlation is 1.000000.

Bottom Row: Avg. All Compartment

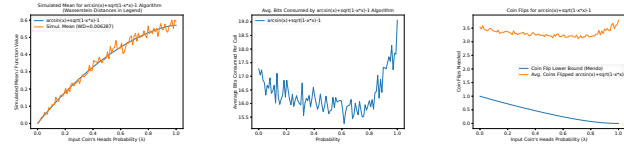
- Histograms (Difference in Legend):** Shows the distribution of 'Input Case Incidence Probability (%)' for the 2019 COVID-19 dataset. The x-axis ranges from 0.00 to 0.35, and the y-axis ranges from 0.0 to 1.0.
- Scatter Plot:** Shows the relationship between 'Input Case Incidence Probability (%)' (x-axis, 0.00 to 0.35) and 'Output Case Incidence Probability (%)' (y-axis, 0.0 to 1.0). The data points are clustered along the diagonal line.
- Correlation Matrix:** Shows the correlation between 'Input Case Incidence Probability (%)' and 'Output Case Incidence Probability (%)'. The correlation is 1.000000.

Figure 1 consists of three subplots. The top-left plot shows the output GCM's network probability (Y-axis, 0.0 to 0.2) versus the input GCM's network probability (X-axis, 0.00 to 0.90). The top-right plot shows the average path length (Y-axis, 100 to 150) versus the probability (X-axis, 0.00 to 0.90). The bottom plot shows the average clustering coefficient (Y-axis, 0.0 to 0.2) versus the input GCM's network probability (X-axis, 0.00 to 0.90). All plots show multiple colored lines representing different input GCMs.

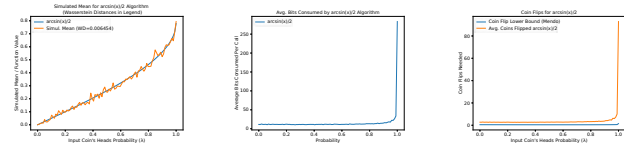
Bernstein
0.2,0.6,0.3



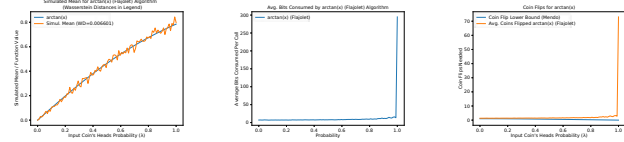
$\arcsin(x) + \sqrt{1-x^2} - 1$



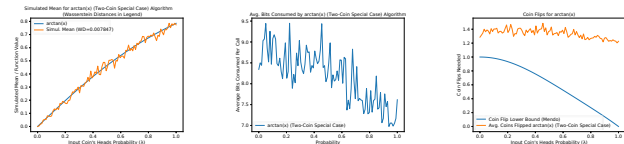
$\arcsin(x)/2$



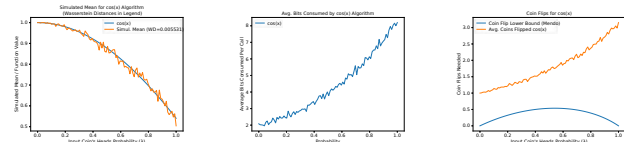
$\arctan(x)$
(Flajolet)



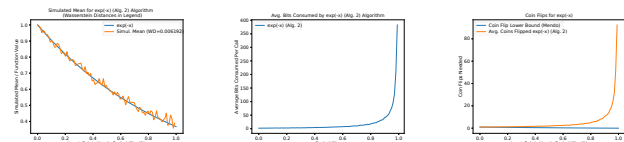
$\arctan(x)$ (Two-Coin Special Case)



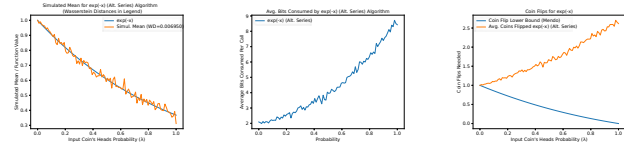
$\cos(x)$



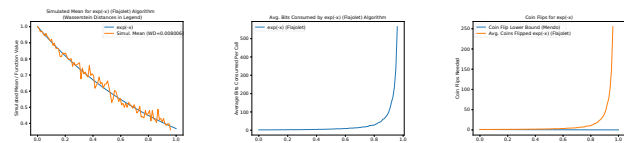
$\exp(-x)$ (Alg. 2)



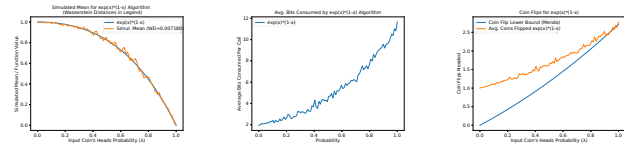
$\exp(-x)$ (Alt. Series)



$\exp(-x)$ (Flajolet)



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



[illegible]

Figure 1 consists of three subplots labeled (a), (b), and (c), each showing performance metrics versus Problem Size N (ranging from 0.0 to 1.0).

- (a) O-iterations (Lower Bound):** The y-axis ranges from 0.0 to 0.6. It compares 'height=400' (blue line) and 'height=1000' (orange line). Both lines show a linear increase from approximately 0.1 at $N=0.0$ to 0.5 at $N=1.0$. The 'height=1000' line is slightly higher than the 'height=400' line.
- (b) Peak Error (Lower Bound):** The y-axis ranges from 0.0 to 0.2. It compares 'height=400' (blue line) and 'height=1000' (orange line). Both lines show a decreasing trend from approximately 0.15 at $N=0.0$ to 0.05 at $N=1.0$. The 'height=1000' line is slightly higher than the 'height=400' line.
- (c) Convergence Rate:** The y-axis ranges from 0.0 to 1.0. It compares 'height=400' (blue line) and 'height=1000' (orange line). The 'height=400' line shows a linear decrease from 1.0 at $N=0.0$ to 0.0 at $N=1.0$. The 'height=1000' line shows a noisy, slightly decreasing trend from approximately 0.9 at $N=0.0$ to 0.7 at $N=1.0$.

Figure 1 consists of three subplots labeled (a), (b), and (c), each showing performance metrics over 100 iterations.

- Subplot (a): Estimated Mean for groups 1,2,3 algorithm.** The y-axis is 'Estimated Mean (1000000)' ranging from 0.2 to 1.0. The x-axis is 'Number of Iterations' ranging from 0 to 100. Two lines are plotted: 'group 1,2,3' (blue) and 'True Mean: 1000000000000000' (orange). Both lines start at approximately 0.25 and increase to 1.0 by iteration 100.
- Subplot (b): RMSE Computed by groups 1,2,3 algorithm.** The y-axis is 'RMSE (1000000000000000)' ranging from 0 to 1600. The x-axis is 'Number of Iterations' ranging from 0 to 100. Two lines are plotted: 'group 1,2,3' (blue) and 'True RMSE: 1000000000000000' (orange). The blue line starts at approximately 1500 and drops to near 0 by iteration 10. The orange line starts at approximately 1500 and drops to near 0 by iteration 20.
- Subplot (c): Conf. Pairs for groups 1,2,3.** The y-axis is 'Conf. Pairs' ranging from 0 to 8000. The x-axis is 'Number of Iterations' ranging from 0 to 100. Two lines are plotted: 'Conf. Pairs (value Based) (orange)' and 'Avg. Conf. Pairs (group 1,2,3) (blue)'. Both lines start at approximately 8000 and drop to near 0 by iteration 10.

Figure 1 consists of three subplots showing the performance of the proposed algorithm for gene1 (10). The x-axis for all plots is 'Costs' ranging from 0.0 to 1.0.

- Left Subplot:** 'Estimated Mean for gene1 (10) algorithm'. The y-axis is 'Estimated Mean (1000000) x 10⁻⁴' ranging from 0.0 to 1.0. It shows a blue line for 'gene1 (10)' and an orange line for 'Error Mean: 0.00100363'. Both lines show an increasing trend as costs increase.
- Middle Subplot:** 'Avg. Mem. Consumed by gene1 (10) algorithm'. The y-axis is 'Avg. Mem. Consumed (1000000) x 10⁻⁴' ranging from 0.0 to 4.0. It shows a blue line for 'gene1 (10)' which fluctuates and generally increases with costs.
- Right Subplot:** 'Cost Fails for gene1 (10)'. The y-axis is 'Costs Failed' ranging from 0.00 to 2.00. It shows three lines: 'Cost Fails: Lower Bound (Heuristic)' (blue), 'Cost Fails: Upper Bound (Heuristic)' (orange), and 'Cost Fails: Proposed' (green). All lines show an increasing trend, with the proposed method (green) being the highest.

Figure 1 consists of three subplots. The left subplot, titled 'Estimated Mean for points 2-6 algorithm', shows the estimated mean for points 2-6 algorithm on the y-axis (ranging from 0.0 to 1.0) versus the estimated mean for points 2-6 algorithm on the x-axis (ranging from 0.0 to 1.0). The proposed algorithm (orange line with markers) and the 2-6 algorithm (blue line) are shown. The middle subplot, titled 'Avg. Time Consumed by points 2-6 algorithm', shows the average time consumed by points 2-6 algorithm on the y-axis (ranging from 0 to 250) versus the estimated mean for points 2-6 algorithm on the x-axis (ranging from 0.0 to 1.0). The proposed algorithm (orange line) and the 2-6 algorithm (blue line) are shown. The right subplot, titled 'Coin Flip for points 2-6', shows the coin flip for points 2-6 on the y-axis (ranging from 0.0 to 1.0) versus the estimated mean for points 2-6 algorithm on the x-axis (ranging from 0.0 to 1.0). The proposed algorithm (orange line) and the 2-6 algorithm (blue line) are shown.

Figure 1 consists of three subplots showing the performance of the proposed algorithm compared to dense, 1-bit algorithms. The x-axis for all plots is 'Normalized Mean for dense, 1-bit algorithm' ranging from 0.6 to 1.0. The y-axis for all plots is 'Normalized Mean for sparse, 1-bit algorithm' ranging from 0.6 to 1.0.

- Left Subplot:** Compares the proposed 1-bit algorithm (red line) with a dense 1-bit algorithm (blue line). Both lines show a sharp increase in performance as the normalized mean increases, with the proposed algorithm performing slightly better at lower means.
- Middle Subplot:** Compares the proposed 1-bit algorithm (blue line) with a dense 1-bit algorithm (blue line). The proposed algorithm shows a sharp increase in performance as the normalized mean increases, while the dense algorithm remains relatively flat.
- Right Subplot:** Compares the proposed 1-bit algorithm (red line) with a dense 1-bit algorithm (blue line). The proposed algorithm shows a sharp increase in performance as the normalized mean increases, while the dense algorithm remains relatively flat.

Figure 1 consists of three subplots labeled (a), (b), and (c), each showing performance metrics for the proposed algorithm compared to two baseline algorithms: power KCS algorithm (approximate) and power KCS algorithm (approximate).

(a) Time Complexity: The y-axis is Time Complexity (in 10^3 of sec) ranging from 0 to 1.5. The x-axis is Power KCS algorithm (approximate) ranging from 0.0 to 1.0. The legend indicates: power KCS algorithm (approximate) (blue line with circles), power KCS algorithm (approximate) (orange line with circles), and power KCS algorithm (approximate) (red line with circles). All three lines are nearly identical, showing a linear increase from 0 to 1.5.

(b) Average CPU Time: The y-axis is Average CPU Time (in sec) ranging from 0 to 14. The x-axis is Power KCS algorithm (approximate) ranging from 0.0 to 1.0. The legend indicates: power KCS algorithm (approximate) (blue line with circles), power KCS algorithm (approximate) (orange line with circles), and power KCS algorithm (approximate) (red line with circles). The blue line starts at approximately 14 and decreases to about 2. The orange and red lines are nearly identical, starting at approximately 14 and decreasing to about 2.

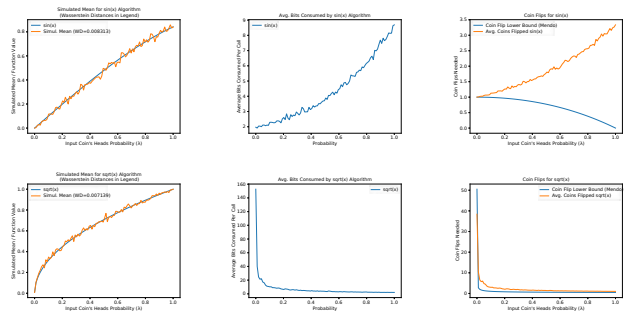
(c) Coin Flips: The y-axis is Coin Flips ranging from 0.0 to 4.0. The x-axis is Power KCS algorithm (approximate) ranging from 0.0 to 1.0. The legend indicates: power KCS algorithm (approximate) (blue line with circles), power KCS algorithm (approximate) (orange line with circles), and power KCS algorithm (approximate) (red line with circles). The blue line starts at approximately 4.0 and decreases to about 0.5. The orange and red lines are nearly identical, starting at approximately 4.0 and decreasing to about 0.5.

[illegible]

Figure 1 consists of three subplots labeled (a), (b), and (c), all showing performance metrics for the proposed algorithm across different values of α (0.0 to 1.0).

- (a) Estimated Mean for power, S-IR algorithm:** The y-axis is 'Estimated Mean for power, S-IR algorithm' (0.0 to 1.0) and the x-axis is 'Input Cost-File (Estimated Mean for power, S-IR algorithm)' (0.0 to 1.0). The plot shows a diagonal line from (0,0) to (1,1) with a shaded confidence interval. The legend indicates 'Estimated Mean for power, S-IR algorithm' (black line) and 'Confidence Interval for power, S-IR algorithm' (shaded area).
- (b) Avg. Bits Consumed by power, S-IR algorithm:** The y-axis is 'Avg. Bits Consumed by power, S-IR algorithm' (0 to 600) and the x-axis is ' α ' (0.0 to 1.0). The plot shows a sharp drop from approximately 550 bits at $\alpha=0$ to near zero for $\alpha > 0.1$. The legend indicates 'power, S-IR' (blue line).
- (c) Cost-File for power, S-IR:** The y-axis is 'Cost-File for power, S-IR' (0 to 1000) and the x-axis is 'Input Cost-File (Estimated Mean for power, S-IR algorithm)' (0.0 to 1.0). The plot shows a sharp drop from approximately 950 at $\alpha=0$ to near zero for $\alpha > 0.1$. The legend indicates 'Cost-File for power, S-IR' (blue line).

$\sin(x)$



\sqrt{x}

1. <https://peteroupc.github.io/bernoulli.md>