Metaheuristic Optimization

9. Comparing Optimization Algorithms

Thomas Weise · 汤卫思

tweise@hfuu.edu.cn · http://iao.hfuu.edu.cn

Hefei University, South Campus 2
Faculty of Computer Science and Technology
Institute of Applied Optimization
230601 Shushan District, Hefei, Anhui, China
Econ. & Tech. Devel. Zone, Jinxiu Dadao 99

合肥学院 南艳湖校区/南2区
计算机科学与技术系
应用优化研究所
中国 安徽省 合肥市 蜀山区 230601
经济技术开发区 锦绣大道99号

Outline

1. Introduction
2. Performace Indicators
3. Statistical Measures
4. Statistical Comparisons
5. Testing is Not Enough
6. Benchmarking
7. Summary

website

Section Outline

- There are many optimization algorithms

Introduction

- There are many optimization algorithms
- For solving an optimization problem, we want to use the algorithm most suitable for it.

- There are many optimization algorithms
- For solving an optimization problem, we want to use the algorithm most suitable for it.
- What does this mean?

Randomized Algorithms

- Special situation: Randomized Algorithms

Randomized Algorithms

- Special situation: Randomized Algorithms
- Performance values cannot be given absolute!

- Special situation: Randomized Algorithms
- Performance values cannot be given absolute!
- 1 run = 1 application of an optimization algorithm to a problem, runs are indepdentent from all prior runs

- Special situation: Randomized Algorithms
- Performance values cannot be given absolute!
- 1 run = 1 application of an optimization algorithm to a problem, runs are indepdentent from all prior runs
- Results can be different for each run!

- Special situation: Randomized Algorithms
- Performance values cannot be given absolute!
- 1 run = 1 application of an optimization algorithm to a problem, runs are indepdentent from all prior runs
- Results can be different for each run!
- Executing algorithm one time does not give reliable information

- Special situation: Randomized Algorithms
- Performance values cannot be given absolute!
- 1 run = 1 application of an optimization algorithm to a problem, runs are indepdentent from all prior runs
- Results can be different for each run!
- Executing algorithm one time does not give reliable information
- Statistical evaluation over a set of runs necessary

Section Outline

- Key parameters [1–3]

Performance Indicators

- Key parameters [1–3]:
  1. Solution quality reached after a certain runtime

- Two key parameter [1–3]:
  1. Solution quality reached after a certain runtime
  2. Runtime to reach a certain solution quality

- Two key parameter [1–3]:
  1. Solution quality reached after a certain runtime
  2. Runtime to reach a certain solution quality
- Measure data samples A containing the results from multiple runs and estimate key parameters.

- What actually is *runtime*?

Absolute Runtime

Measure the (absolute) consumed runtime of the algorithm in ms

Absolute Runtime

Measure the (absolute) consumed runtime of the algorithm in ms

- Advantages

Absolute Runtime

Measure the (absolute) consumed runtime of the algorithm in ms

- Advantages:
  - Results in many works reported in this format

Absolute Runtime

Measure the (absolute) consumed runtime of the algorithm in ms

- Advantages:
    - Results in many works reported in this format
    - A quantity that makes physical sense

Absolute Runtime

Measure the (absolute) consumed runtime of the algorithm in ms

- Advantages:
  - Results in many works reported in this format
  - A quantity that makes physical sense
  - Includes all "hidden complexities" of algorithm

Absolute Runtime

Measure the (absolute) consumed runtime of the algorithm in ms

- Advantages:
  - Results in many works reported in this format
  - A quantity that makes physical sense
  - Includes all "hidden complexities" of algorithm
- Disadvantages

Measure the (absolute) consumed runtime of the algorithm in ms

- Advantages:
  - Results in many works reported in this format
  - A quantity that makes physical sense
  - Includes all "hidden complexities" of algorithm
- Disadvantages:
  - Strongly machine dependent

Absolute Runtime

Measure the (absolute) consumed runtime of the algorithm in ms

- Advantages:
  - Results in many works reported in this format
  - A quantity that makes physical sense
  - Includes all "hidden complexities" of algorithm
- Disadvantages:
  - Strongly machine dependent
  - Granularity of about 10ms: many things seem to happen at the same time

Absolute Runtime

Measure the (absolute) consumed runtime of the algorithm in ms

- Advantages:
  - Results in many works reported in this format
  - A quantity that makes physical sense
  - Includes all "hidden complexities" of algorithm
- Disadvantages:
  - Strongly machine dependent
  - Granularity of about 10ms: many things seem to happen at the same time
  - Can be biased by "outside effects", e.g., OS, scheduling, other processes, I/O, swapping, ...

Absolute Runtime

Measure the (absolute) consumed runtime of the algorithm in ms

- Advantages:
  - Results in many works reported in this format
  - A quantity that makes physical sense
  - Includes all "hidden complexities" of algorithm
- Disadvantages:
  - Strongly machine dependent
  - Granularity of about 10ms: many things seem to happen at the same time
  - Can be biased by "outside effects", e.g., OS, scheduling, other processes, I/O, swapping, ...
  - Inherently incomparable

Absolute Runtime

Measure the (absolute) consumed runtime of the algorithm in ms

- Advantages:
  - Results in many works reported in this format
  - A quantity that makes physical sense
  - Includes all "hidden complexities" of algorithm
- Disadvantages:
  - Strongly machine dependent
  - Granularity of about 10ms: many things seem to happen at the same time
  - Can be biased by "outside effects", e.g., OS, scheduling, other processes, I/O, swapping, ...
  - Inherently incomparable
- Hardware, software, OS, etc. all have nothing to do with the *optimization algorithm* itself and are relevant only in a specific application...

Function Evaluations: FEs

Measure the number of fully constructed and tested candidate solutions

Function Evaluations: FEs

Measure the number of fully constructed and tested candidate solutions

- Advantages

Function Evaluations: FEs

Measure the number of fully constructed and tested candidate solutions

- Advantages:
  - Results in many works reported in this format (or FEss can be deduced)

Function Evaluations: FEs

Measure the number of fully constructed and tested candidate solutions

- Advantages:
  - Results in many works reported in this format (or FEss can be deduced)
  - Machine-independent measure

Function Evaluations: FEs

Measure the number of fully constructed and tested candidate solutions

- Advantages:
  - Results in many works reported in this format (or FEss can be deduced)
  - Machine-independent measure
  - Cannot be influenced by "outside effects"

Function Evaluations: FEs

Measure the number of fully constructed and tested candidate solutions

- Advantages:
  - Results in many works reported in this format (or FEss can be deduced)
  - Machine-independent measure
  - Cannot be influenced by "outside effects"
  - In many optimization problems, computing the objective value is the most time consuming task

Function Evaluations: FEs

Measure the number of fully constructed and tested candidate solutions

- Advantages:
  - Results in many works reported in this format (or FEss can be deduced)
  - Machine-independent measure
  - Cannot be influenced by "outside effects"
  - In many optimization problems, computing the objective value is the most time consuming task
- Disadvantages

Function Evaluations: FEs

Measure the number of fully constructed and tested candidate solutions

- Advantages:
  - Results in many works reported in this format (or FEss can be deduced)
  - Machine-independent measure
  - Cannot be influenced by "outside effects"
  - In many optimization problems, computing the objective value is the most time consuming task
- Disadvantages:
  - No clear relationship to real runtime
  - Does not contain "hidden complexities" of algorithm

Function Evaluations: FEs

Measure the number of fully constructed and tested candidate solutions

- Advantages:
  - Results in many works reported in this format (or FEss can be deduced)
  - Machine-independent measure
  - Cannot be influenced by "outside effects"
  - In many optimization problems, computing the objective value is the most time consuming task
- Disadvantages:
  - No clear relationship to real runtime
  - Does not contain "hidden complexities" of algorithm
  - 1 FE: very different costs in different situations!
- Relevant for comparing algorithms, but not so much for the practical application

- Rewrite the two key parameters by choosing a time measure [1–3]

- Rewrite the two key parameters by choosing a time measure [1–3]:
  1. Solution quality reached after a certain number of FEs

- Rewrite the two key parameters by choosing a time measure [1–3]:
  1. Solution quality reached after a certain number of FEs
  2. Number FEs needed to reach a certain solution quality

- Common measure of solution quality: Objective function value of best solution discovered.

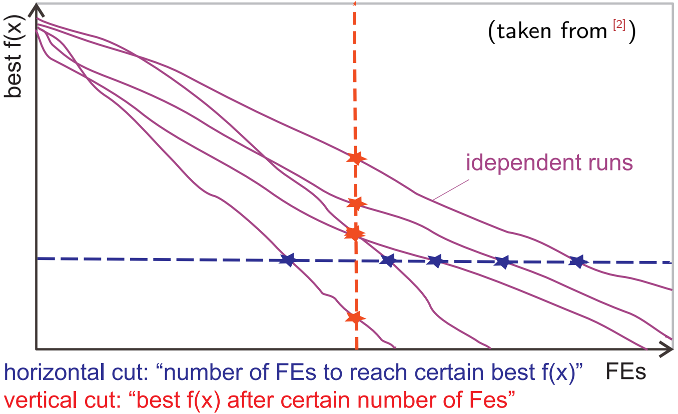
Solution Quality

- Common measure of solution quality: Objective function value of best solution discovered.
- Rewrite the two key parameters [1–3]

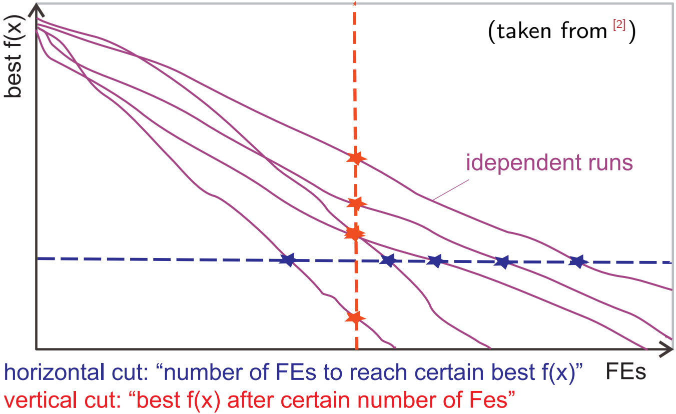
- Common measure of solution quality: Objective function value of best solution discovered.
- Rewrite the two key parameters [1–3]:
    1. Best objective function value reached after a certain number of FEs

- Common measure of solution quality: Objective function value of best solution discovered.
- Rewrite the two key parameters [1–3]:
  1. Best objective function value reached after a certain number of FEs
  2. Number FEs needed to reach a certain objective function value

- Which one is the better performance indicator?
  1. Best objective function value reached after a certain number of FEs



- Which one is the better performance indicator?
  1. Best objective function value reached after a certain number of FEs
  2. Number FEs needed to reach a certain objective function value



Which Indicator is better?

- Number FEs needed to reach a certain objective function value
- Prefered by Hansen et al. [2]

Which Indicator is better?

- Number FEs needed to reach a certain objective function value
- Prefered by Hansen et al. [2]:
  - Measures a time needed to reach a target function value ⇒ "Algorithm A is two/ten/hundred times faster than Algorithm B in solving this problem"

Which Indicator is better?

- Number FEs needed to reach a certain objective function value
- Prefered by Hansen et al. [2]:
  - Measures a time needed to reach a target function value $\Rightarrow$ "Algorithm A is two/ten/hundred times faster than Algorithm B in solving this problem"
  - Benchmark Perspective: No interpretable meaning to the fact that Algorithm A reaches a function value that is two/ten/hundred times smaller than the one reached by Algorithm B

Which Indicator is better?

- Best objective function value reached after a certain number of FEs

- Best objective function value reached after a certain number of FEs
- Prefered by many benchmark suites such as [4].

Which Indicator is better?

- Best objective function value reached after a certain number of FEs
- Prefered by many benchmark suites such as [4].
- Practice Perspective: Best results achievable with given time budget wins.

Which Indicator is better?

- Best objective function value reached after a certain number of FEs
- Prefered by many benchmark suites such as [4].
- Practice Perspective: Best results achievable with given time budget wins.
- This perspective maybe less suitable for benchmarking, but surely true in practice.

Key Parameters

- No official consesus on which view is “better”.

Key Parameters

- No official consesus on which view is “better”.
- This also strongly depends on the situation.

Key Parameters

- No official consesus on which view is “better”.
- This also strongly depends on the situation.
- Best approach: Evaluate algorithm according to both methods.

- How to determine the right maximum FEs or target function values?

Determining Target Values

- How to determine the right maximum FEs or target function values?
  1. From studies in literature regarding similar or the same problem.

Determining Target Values

- How to determine the right maximum FEs or target function values?
  1. From studies in literature regarding similar or the same problem.
  2. From experience.

- How to determine the right maximum FEs or target function values?
  1. From studies in literature regarding similar or the same problem.
  2. From experience.
  3. From prior, small-scale experiments.

Determining Target Values

- How to determine the right maximum FEs or target function values?
  1. From studies in literature regarding similar or the same problem.
  2. From experience.
  3. From prior, small-scale experiments.
  4. Based on known lower bounds

Section Outline

Important Distinction

- Crucial Difference: distribution and sample

Important Distinction

- Crucial Difference: distribution and sample
- A sample is what we *measure*

Important Distinction

- Crucial Difference: distribution and sample
- A sample is what we *measure*
- A distribution is the asymptotic result of the ideal process

Important Distinction

- Crucial Difference: distribution and sample
- A sample is what we *measure*
- A distribution is the asymptotic result of the ideal process

- Statistical parameters of the distribution can be estimated from a sample

- Crucial Difference: distribution and sample
- A sample is what we *measure*
- A distribution is the asymptotic result of the ideal process

- Statistical parameters of the distribution can be estimated from a sample
- Example: Dice Throw

- Crucial Difference: distribution and sample
- A sample is what we *measure* (10 throws, mean result 4)
- A distribution is the asymptotic result of the ideal process

- Statistical parameters of the distribution can be estimated from a sample
- Example: Dice Throw

Important Distinction

- Crucial Difference: distribution and sample
- A sample is what we *measure* (10 throws, mean result 4)
- A distribution is the asymptotic result of the ideal process

- Statistical parameters of the distribution can be estimated from a sample
- Example: Dice Throw
- Never foget: All measured parameters are just estimates.

Important Distinction

- Crucial Difference: distribution and sample
- A sample is what we *measure* (10 throws, mean result 4)
- A distribution is the asymptotic result of the ideal process (Expected value: 3.5)
- Statistical parameters of the distribution can be estimated from a sample
- Example: Dice Throw
- Never foget: All measured parameters are just estimates.

Definition (Arithmetic Mean)

The arithmetic mean $\mathrm{mean}(A)$ is an estimate of the expected value of a data sample $A = (a_1, a_2, \ldots, a_n)$. It is computed as the sum of all n elements a_i in the sample data A divided by the total number of values.

$$\mathrm{mean}(A) = \frac{\sum_{\forall a \in A} a}{n} = \frac{1}{n} \sum_{i=0}^{n-1} a_i$$

Definition (Median)

The median $\mathrm{med}(X)$ is the value right in the middle of a sample or distribution, dividing it into two equal halves.

$$P(X \leq \mathrm{med}(X)) \geq \frac{1}{2} \wedge P(X \geq \mathrm{med}(X)) \geq \frac{1}{2} \tag{1}$$

Median

Definition (Median)

The median $\mathrm{med}(X)$ is the value right in the middle of a sample or distribution, dividing it into two equal halves.

$$P(X \leq \mathrm{med}(X)) \geq \frac{1}{2} \wedge P(X \geq \mathrm{med}(X)) \geq \frac{1}{2} \tag{1}$$

- The probability of drawing an element less than or equal to $\mathrm{med}(X)$ is 50%

Median

Definition (Median)

The median $\mathrm{med}(X)$ is the value right in the middle of a sample or distribution, dividing it into two equal halves.

$$P(X \leq \mathrm{med}(X)) \geq \frac{1}{2} \wedge P(X \geq \mathrm{med}(X)) \geq \frac{1}{2} \tag{1}$$

- The probability of drawing an element less than or equal to $\mathrm{med}(X)$ is 50%
- The probability of drawing an element greater than or equal to $\mathrm{med}(X)$ is 50%

Definition (Median)

The median $\mathrm{med}(X)$ is the value right in the middle of a sample or distribution, dividing it into two equal halves.

$$P(X \leq \mathrm{med}(X)) \geq \frac{1}{2} \wedge P(X \geq \mathrm{med}(X)) \geq \frac{1}{2} \tag{1}$$

- The probability of drawing an element less than or equal to $\mathrm{med}(X)$ is 50%
- The probability of drawing an element greater than or equal to $\mathrm{med}(X)$ is 50%
- For a sorted data sample $A = (a_1, a_2, \ldots, a_n)$ of n elements the median $\mathrm{med}(A)$ can be estimated as:

Median

Definition (Median)

The median $\mathrm{med}(X)$ is the value right in the middle of a sample or distribution, dividing it into two equal halves.

$$P(X \leq \mathrm{med}(X)) \geq \frac{1}{2} \wedge P(X \geq \mathrm{med}(X)) \geq \frac{1}{2} \tag{1}$$

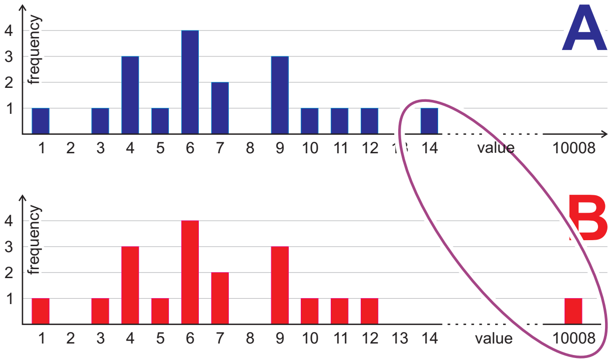
- The probability of drawing an element less than or equal to $\mathrm{med}(X)$ is 50%
- The probability of drawing an element greater than or equal to $\mathrm{med}(X)$ is 50%
- For a sorted data sample $A = (a_1, a_2, \dots, a_n)$ of n elements the median $\mathrm{med}(A)$ can be estimated as:

$$\mathrm{med}(A) = \begin{cases} a_{\frac{n-1}{2}+1} & \text{if } n \text{ is odd} \\ \frac{1}{2}\left(a_{\frac{n}{2}} + a_{\frac{n}{2}+1}\right) & \text{otherwise} \end{cases}$$

Example for Data Samples

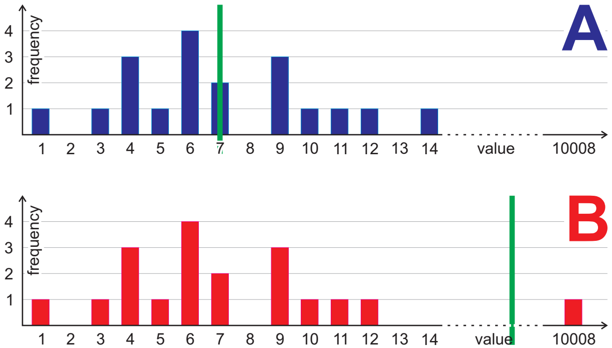
- Two sets of data samples A and B with $n_a = n_b = 19$ values.

$$
\begin{aligned}
A &= (1, 3, 4, 4, 4, 5, 6, 6, 6, 6, 7, 7, 9, 9, 9, 10, 11, 12, 14) \\
B &= (1, 3, 4, 4, 4, 5, 6, 6, 6, 6, 7, 7, 9, 9, 9, 10, 11, 12, \textcolor{red}{10\,008})
\end{aligned}
$$



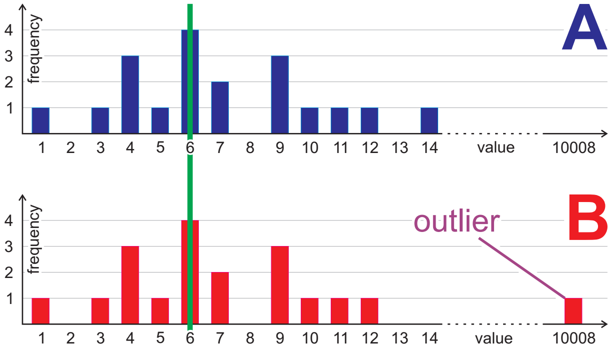
- Two data samples A and B with $n_a = n_b = 19$ values.

$$\mathrm{mean}(A) = \frac{1}{19}\sum_{i=1}^{19} a_i = \frac{133}{19} = 7 \qquad \mathrm{mean}(b) = \frac{1}{19}\sum_{i=1}^{19} b_i = \frac{10\,127}{19} = 533$$



- Two data samples A and B with $n_a = n_b = 19$ values.

$$A = (1,3,4,4,4,5,6,6,6,6,7,7,9,9,9,10,11,12,14) \quad \Rightarrow \quad \mathrm{med}(A) = 6$$
$$B = (1,3,4,4,4,5,6,6,6,6,7,7,9,9,9,10,11,12,10\,008) \quad \Rightarrow \quad \mathrm{med}(B) = 6$$



Mean vs. Median

- When describing a random process, we should always use the median instead of the mean. [5–8]

Mean vs. Median

- When describing a random process, we should always use the median instead of the mean. [5–8], because
  1. the median is more robust towards outliers,

Mean vs. Median

- When describing a random process, we should always use the median instead of the mean. [5–8], because
  1. the median is more robust towards outliers,
  2. the mean is useful (only) for symmetric distributions and badly represents skewed distributions.

Mean vs. Median

- When describing a random process, we should always use the median instead of the mean. [5–8], because
  1. the median is more robust towards outliers,
  2. the mean is useful (only) for symmetric distributions and badly represents skewed distributions.
- The median is the first statistic we should take a look at!

Definition (Standard Deviation)

The statistical estimate $\mathrm{stddev}(A)$ of the standard deviation of a data sample $A = (a_1, a_2, \dots, a_n)$ is the square root of the estimated variance $\mathrm{var}(A)$.

$$\begin{aligned}\mathrm{var}(A) &= \frac{1}{n-1}\sum_{i=0}^{n-1}(a_i - \mathrm{mean}(A))^2 \\ \mathrm{stddev}(A) &= \sqrt{\mathrm{var}(A)}\end{aligned}$$

Definition (Quantile)

The q-quantiles divide a sorted data sample $A = (a_1, a_2, \ldots, a_n)$ into q parts T_i which contain the same amounts of elements

Quantiles

Definition (Quantile)

The q-quantiles divide a sorted data sample $A = (a_1, a_2, \ldots, a_n)$ into q parts T_i which contain the same amounts of elements (i.e., quantiles are a generalized median).

Definition (Quantile)

The q-quantiles divide a sorted data sample $A = (a_1, a_2, \dots, a_n)$ into q parts T_i which contain the same amounts of elements (i.e., quantiles are a generalized median).
The k^{th} q-quantile of A, i.e., $\mathrm{quantile}_q^k(A)$, can be estimated as follows:

$$
\begin{aligned}
t &= \frac{k * n}{q} \\
\mathrm{quantile}_q^k(A) &= \begin{cases} \frac{1}{2}\left(a_t + a_{t+1}\right) & \text{if } t \text{ is integer} \\ a_{\lceil t \rceil} & \text{otherwise} \end{cases}
\end{aligned}
$$

Quantiles

Definition (Quantile)

The q-quantiles divide a sorted data sample $A = (a_1, a_2, \dots, a_n)$ into q parts T_i which contain the same amounts of elements (i.e., quantiles are a generalized median).
The k^{th} q-quantile of A, i.e., $\mathrm{quantile}_q^k(A)$, can be estimated as follows:

$$
\begin{aligned}
t &= \frac{k * n}{q} \\
\mathrm{quantile}_q^k(A) &= \begin{cases} \frac{1}{2}\left(a_t + a_{t+1}\right) & \text{if } t \text{ is integer} \\ a_{\lceil t \rceil} & \text{otherwise} \end{cases}
\end{aligned}
$$

- The $\mathrm{quantile}_2^1(A)$ is the median of A

Quantiles

Definition (Quantile)

The q-quantiles divide a sorted data sample $A = (a_1, a_2, \dots, a_n)$ into q parts T_i which contain the same amounts of elements (i.e., quantiles are a generalized median).
The k^{th} q-quantile of A, i.e., $\mathrm{quantile}_q^k(A)$, can be estimated as follows:

$$
\begin{aligned}
t &= \frac{k * n}{q} \\
\mathrm{quantile}_q^k(A) &= \begin{cases} \frac{1}{2}\left(a_t + a_{t+1}\right) & \text{if } t \text{ is integer} \\ a_{\lceil t \rceil} & \text{otherwise} \end{cases}
\end{aligned}
$$

- The $\mathrm{quantile}_2^1(A)$ is the median of A
- 4-quantiles are called quartiles.

Standard Deviation: Example

- Two data samples A and B with $n_a = n_b = 19$ values.

$$\begin{aligned}
A &= (1, 3, 4, 4, 4, 5, 6, 6, 6, 6, 7, 7, 9, 9, 9, 10, 11, 12, 14) \\
\mathrm{mean}(A) &= 7 \\
B &= (1, 3, 4, 4, 4, 5, 6, 6, 6, 6, 7, 7, 9, 9, 9, 10, 11, 12, 10\,008) \\
\mathrm{mean}(B) &= 533
\end{aligned}$$

Standard Deviation: Example

- Two data samples A and B with $n_a = n_b = 19$ values.

$$
\begin{aligned}
A &= (1, 3, 4, 4, 4, 5, 6, 6, 6, 6, 7, 7, 9, 9, 9, 10, 11, 12, 14) \\
\mathrm{mean}(A) &= 7 \\
B &= (1, 3, 4, 4, 4, 5, 6, 6, 6, 6, 7, 7, 9, 9, 9, 10, 11, 12, 10\,008) \\
\mathrm{mean}(B) &= 533 \\
\mathrm{var}(A) &= \frac{1}{19-1} \sum_{i=1}^{19} (a_i - \mathrm{mean}(a))^2 = \frac{198}{18} = 11 \\
\mathrm{var}(B) &= \frac{1}{19-1} \sum_{i=1}^{19} (b_i - \mathrm{mean}(b))^2 = \frac{94\,763\,306}{18} \approx 5\,264\,628.1
\end{aligned}
$$

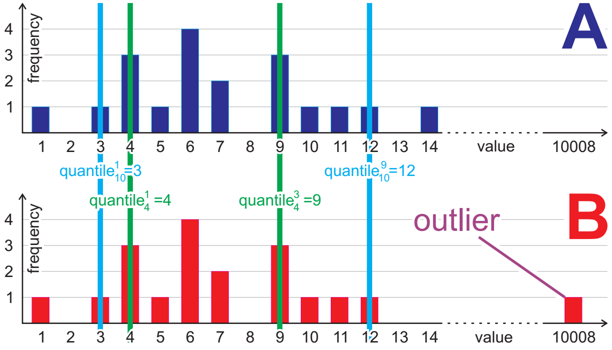
Standard Deviation: Example

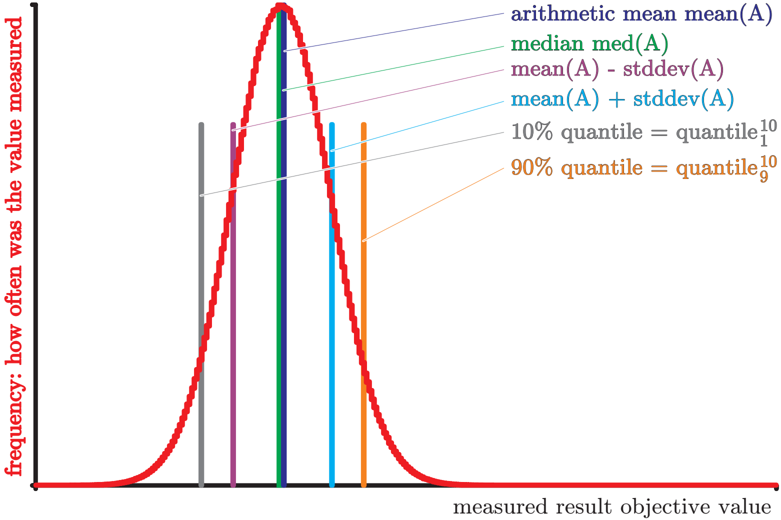
- Two data samples A and B with $n_a = n_b = 19$ values.

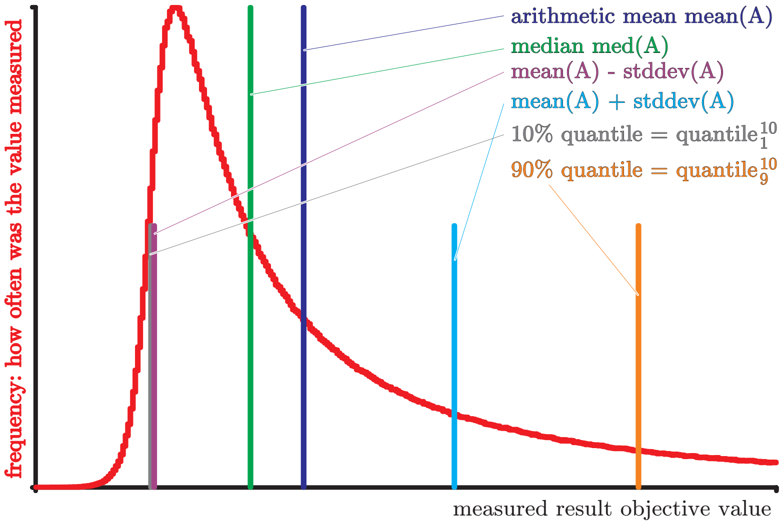
$$
\begin{aligned}
A &= (1, 3, 4, 4, 4, 5, 6, 6, 6, 6, 7, 7, 9, 9, 9, 10, 11, 12, 14) \\
\text{mean}(A) &= 7 \\
B &= (1, 3, 4, 4, 4, 5, 6, 6, 6, 6, 7, 7, 9, 9, 9, 10, 11, 12, 10\,008) \\
\text{mean}(B) &= 533 \\
\text{var}(A) &= \frac{1}{19-1} \sum_{i=1}^{19} (a_i - \text{mean}(a))^2 = \frac{198}{18} = 11 \\
\text{var}(B) &= \frac{1}{19-1} \sum_{i=1}^{19} (b_i - \text{mean}(b))^2 = \frac{94\,763\,306}{18} \approx 5\,264\,628.1 \\
\text{stddev}(A) &= \sqrt{\text{var}(A)} = \sqrt{11} \approx 3.316\,624\,79 \\
\text{stddev}(B) &= \sqrt{\text{var}(B)} = \sqrt{\frac{94\,763\,306}{18}} \approx 2\,294.477\,743
\end{aligned}
$$

- Two data samples A and B with $n_a = n_b = 19$ values.

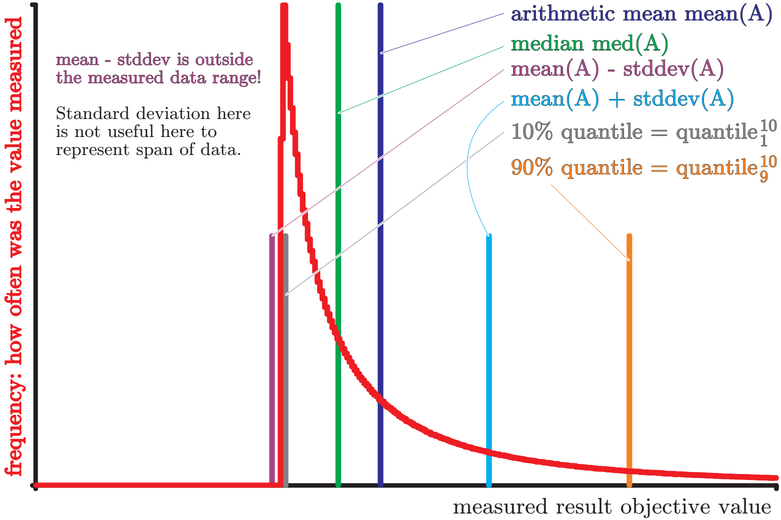
$$A = (1, 3, 4, 4, 4, 5, 6, 6, 6, 6, 7, 7, 9, 9, 9, 10, 11, 12, 14)$$
$$B = (1, 3, 4, 4, 4, 5, 6, 6, 6, 6, 7, 7, 9, 9, 9, 10, 11, 12, 10\,008)$$




arithmetic mean mean(A)
median med(A)
mean(A) - stddev(A)
mean(A) + stddev(A)
10% quantile = quantile_1^{10}
90% quantile = quantile_9^{10}
frequency: how often was the value measured
measured result objective value


arithmetic mean mean(A)
median med(A)
mean(A) - stddev(A)
mean(A) + stddev(A)
10% quantile = quantile_1^{10}
90% quantile = quantile_9^{10}
frequency: how often was the value measured
measured result objective value

Further Example



- Robust statistic measures are:
  1. Median
  2. Quantiles

- Robust statistic measures are:
  1. Median
  2. Quantiles
- *Only if necessary*, compute the estimates of the
  1. Arithmetic Mean
  2. Standard Deviation

Section Outline

1. Introduction
2. Performace Indicators
3. Statistical Measures
4. **Statistical Comparisons**
5. Testing is Not Enough
6. Benchmarking
7. Summary

- We can now e.g., perform 20 runs each with two different optimization algorithms on one problem and compute the median of one of the two performance measures.

- We can now e.g., perform 20 runs each with two different optimization algorithms on one problem and compute the median of one of the two performance measures.
- Likely, they will be different.

- We can now e.g., perform 20 runs each with two different optimization algorithms on one problem and compute the median of one of the two performance measures.
- Likely, they will be different.
- For one of the two algorithms, the results will be better.

- We can now e.g., perform 20 runs each with two different optimization algorithms on one problem and compute the median of one of the two performance measures.
- Likely, they will be different.
- For one of the two algorithms, the results will be better.
- What does this mean?

- We can now e.g., perform 20 runs each with two different optimization algorithms on one problem and compute the median of one of the two performance measures.
- Likely, they will be different.
- For one of the two algorithms, the results will be better.
- What does this mean?
- It means that one of the two algorithms is better

- We can now e.g., perform 20 runs each with two different optimization algorithms on one problem and compute the median of one of the two performance measures.
- Likely, they will be different.
- For one of the two algorithms, the results will be better.
- What does this mean?
- It means that one of the two algorithms is better with a certain probability

- We can now e.g., perform 20 runs each with two different optimization algorithms on one problem and compute the median of one of the two performance measures.
- Likely, they will be different.
- For one of the two algorithms, the results will be better.
- What does this mean?
- It means that one of the two algorithms is better with a certain probability
- If we say "A is better than B", we have a certain chance α to be wrong.

- We can now e.g., perform 20 runs each with two different optimization algorithms on one problem and compute the median of one of the two performance measures.
- Likely, they will be different.
- For one of the two algorithms, the results will be better.
- What does this mean?
- It means that one of the two algorithms is better with a certain probability
- If we say "A is better than B", we have a certain chance α to be wrong.
- The statement "A is better than B" makes only sense if we can give an upper bound α for the error probability!

- Compare two data samples $A = (a_1, a_2, \dots)$ and $B = (b_1, b_2, \dots)$ and

- Compare two data samples $A = (a_1, a_2, \dots)$ and $B = (b_1, b_2, \dots)$ and
- Get a result (e.g., "The median of A is bigger than the median of B") together with an error probability p that the conclusion is wrong.

- Compare two data samples $A = (a_1, a_2, \dots)$ and $B = (b_1, b_2, \dots)$ and
- Get a result (e.g., "The median of A is bigger than the median of B") together with an error probability p that the conclusion is wrong.
- If p is less than a significance level (upper bound) α, we can accept the the conclusion.

- Compare two data samples $A = (a_1, a_2, \dots)$ and $B = (b_1, b_2, \dots)$ and
- Get a result (e.g., "The median of A is bigger than the median of B") together with an error probability p that the conclusion is wrong.
- If p is less than a significance level (upper bound) α, we can accept the the conclusion.
- Otherwise, the observation is not significant.

Underlying General Idea

- We observe some ongoing process P and make some kind of observation O.

Underlying General Idea

- We observe some ongoing process P and make some kind of observation O.
- Question: Can we say: "The observation O is a good approximation of what process P does"?

Underlying General Idea

- We observe some ongoing process P and make some kind of observation O.
- Question: Can we say: "The observation O is a good approximation of what process P does"?
- Question: How likely is this observation O in the case that it is NOT an approximation of P.

Underlying General Idea

- We observe some ongoing process P and make some kind of observation O.
- Question: Can we say: "The observation O is a good approximation of what process P does"?
- Question: How likely is this observation O in the case that it is NOT an approximation of P.
- In other words: What is the probability that O occurs if it does not represent the statistical distribution of the sampled process P?

- Coin flip game: We flip a coin. If it is heads, I give you 1 EUR, if it is tails, you give me 1 EUR.

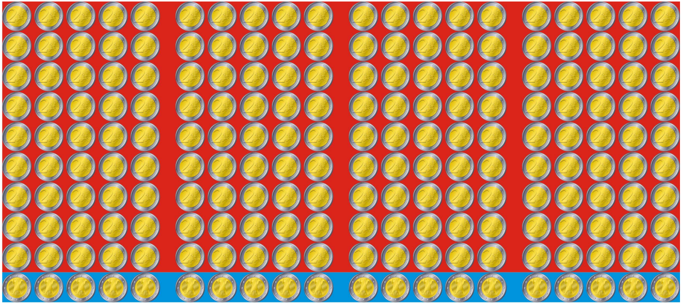
- Coin flip game: We flip a coin. If it is heads, I give you 1 EUR, if it is tails, you give me 1 EUR.

Example for Underlying Idea

- Coin flip game: We flip a coin. If it is heads, I give you 1 EUR, if it is tails, you give me 1 EUR.
- We play 200 times.
- I win 180 times. You win 20 times.

Example for Underlying Idea

- Coin flip game: We flip a coin. If it is heads, I give you 1 EUR, if it is tails, you give me 1 EUR.
- We play 200 times.
- I win 180 times. You win 20 times.



Example for Underlying Idea

- Coin flip game: We flip a coin. If it is heads, I give you 1 EUR, if it is tails, you give me 1 EUR.
- We play 200 times.
- I win 180 times. You win 20 times.
- Did I cheat? Is my coin "fixed"? (i.e., is your chance to win $\neq 50\%$)

Example for Underlying Idea

- Coin flip game: We flip a coin. If it is heads, I give you 1 EUR, if it is tails, you give me 1 EUR.
- We play 200 times.
- I win 180 times. You win 20 times.
- Did I cheat? Is my coin "fixed"? (i.e., is your chance to win $\neq 50\%$)
- Assumption: I cheat. (alternative hypothesis H_1)

Example for Underlying Idea

- Coin flip game: We flip a coin. If it is heads, I give you 1 EUR, if it is tails, you give me 1 EUR.
- We play 200 times.
- I win 180 times. You win 20 times.
- Did I cheat? Is my coin "fixed"? (i.e., is your chance to win $\neq$ 50%)
- Assumption: I cheat. (alternative hypothesis H_1)
- It is impossible to compute my winning probability if I cheated...

Example for Underlying Idea

- Coin flip game: We flip a coin. If it is heads, I give you 1 EUR, if it is tails, you give me 1 EUR.
- We play 200 times.
- I win 180 times. You win 20 times.
- Did I cheat? Is my coin "fixed"? (i.e., is your chance to win $\neq 50\%$)
- Assumption: I cheat. (alternative hypothesis H_1)
- It is impossible to compute my winning probability if I cheated...
- Counter-Assumption: I did not cheat. (null hypothesis H_0)

Example for Underlying Idea

- Coin flip game: We flip a coin. If it is heads, I give you 1 EUR, if it is tails, you give me 1 EUR.
- We play 200 times.
- I win 180 times. You win 20 times.
- Did I cheat? Is my coin "fixed"? (i.e., is your chance to win $\neq$ 50%)
- Assumption: I cheat. (alternative hypothesis H_1)
- It is impossible to compute my winning probability if I cheated...
- Counter-Assumption: I did not cheat. (null hypothesis H_0)
- Question: How likely is it that I win at least 180 times if I did not cheat?

Example for Underlying Idea

- Question: How likely is it that I win at least 180 times if I did not cheat?

Example for Underlying Idea

- Question: How likely is it that I win at least 180 times if I did not cheat?
- In this case, the probabilities for heads and tails are $q = P(\texttt{head}) = P(\texttt{tail}) = 0.5$.

Example for Underlying Idea

- Question: How likely is it that I win at least 180 times if I did not cheat?
- In this case, the probabilities for heads and tails are $q = P(\texttt{head}) = P(\texttt{tail}) = 0.5$.
- Flipping a coin n times is a Bernoulli Process [9–11]

Example for Underlying Idea

- Question: How likely is it that I win at least 180 times if I did not cheat?
- In this case, the probabilities for heads and tails are $q = P(\texttt{head}) = P(\texttt{tail}) = 0.5$.
- Flipping a coin n times is a Bernoulli Process [9–11]
- The probability $P(k|n)$ to flip $k \in 0..n$ times heads (or tails) is thus:

$$P(k|n) = \binom{n}{k} 0.5^k * (1-0.5)^{n-k} = \binom{n}{k} 0.5^k * 0.5^{n-k} = \binom{n}{k} \frac{1}{2^n}$$

- Question: How likely is it that I win at least 180 times if I did not cheat?
- In this case, the probabilities for heads and tails are $q = P(\texttt{head}) = P(\texttt{tail}) = 0.5$.
- Flipping a coin n times is a Bernoulli Process [9–11]
- The probability $P(k|n)$ to flip $k \in 0..n$ times heads (or tails) is thus:

$$P(k|n) = \binom{n}{k} 0.5^k * (1-0.5)^{n-k} = \binom{n}{k} 0.5^k * 0.5^{n-k} = \binom{n}{k} \frac{1}{2^n}$$

- For winning at least $z = 180$ times, we need to compute:[1]

$$P(k \geq z|n) \quad = \quad \sum_{i=z}^{n} P(i|n)$$

[1] For the large n and k computation, I used the websites [12, 13].

Example for Underlying Idea

- Question: How likely is it that I win at least 180 times if I did not cheat?
- In this case, the probabilities for heads and tails are $q = P(\texttt{head}) = P(\texttt{tail}) = 0.5$.
- Flipping a coin n times is a Bernoulli Process [9–11]
- The probability $P(k|n)$ to flip $k \in 0..n$ times heads (or tails) is thus:

$$P(k|n) = \binom{n}{k} 0.5^k * (1-0.5)^{n-k} = \binom{n}{k} 0.5^k * 0.5^{n-k} = \binom{n}{k} \frac{1}{2^n}$$

- For winning at least $z = 180$ times, we need to compute:

$$P(k \geq z|n) \;=\; \sum_{i=z}^{n} P(i|n) = \sum_{i=180}^{200} P(i|200) = \sum_{i=180}^{200} \left[\binom{200}{i} \frac{1}{2^{200}} \right]$$

- Question: How likely is it that I win at least 180 times if I did not cheat?
- In this case, the probabilities for heads and tails are $q = P(\texttt{head}) = P(\texttt{tail}) = 0.5$.
- Flipping a coin n times is a Bernoulli Process [9–11]
- The probability $P(k|n)$ to flip $k \in 0..n$ times heads (or tails) is thus:

$$P(k|n) = \binom{n}{k} 0.5^k * (1-0.5)^{n-k} = \binom{n}{k} 0.5^k * 0.5^{n-k} = \binom{n}{k} \frac{1}{2^n}$$

- For winning at least $z = 180$ times, we need to compute:

$$P(k \geq z|n) = \sum_{i=z}^{n} P(i|n) = \frac{1}{2^{200}} \sum_{i=180}^{200} \binom{200}{i}$$

Example for Underlying Idea

- Question: How likely is it that I win at least 180 times if I did not cheat?
- In this case, the probabilities for heads and tails are $q = P(\texttt{head}) = P(\texttt{tail}) = 0.5$.
- Flipping a coin n times is a Bernoulli Process [9–11]
- The probability $P(k|n)$ to flip $k \in 0..n$ times heads (or tails) is thus:

$$P(k|n) = \binom{n}{k} 0.5^k * (1-0.5)^{n-k} = \binom{n}{k} 0.5^k * 0.5^{n-k} = \binom{n}{k} \frac{1}{2^n}$$

- For winning at least $z = 180$ times, we need to compute:

$$\begin{aligned} P(k \geq z|n) &= \sum_{i=z}^{n} P(i|n) = \frac{1}{2^{200}} \sum_{i=180}^{200} \binom{200}{i} \\ &= \frac{1\,812\,514\,088\,583\,649\,808\,418\,096\,716}{1\,606\,938\,044\,258\,990\,275\,541\,962\,092\,341\,162\,602\,522\,202\,993\,782\,792\,835\,301\,376} \end{aligned}$$

Example for Underlying Idea

- Question: How likely is it that I win at least 180 times if I did not cheat?
- In this case, the probabilities for heads and tails are $q = P(\texttt{head}) = P(\texttt{tail}) = 0.5$.
- Flipping a coin n times is a Bernoulli Process [9–11]
- The probability $P(k|n)$ to flip $k \in 0..n$ times heads (or tails) is thus:

$$P(k|n) = \binom{n}{k} 0.5^k * (1-0.5)^{n-k} = \binom{n}{k} 0.5^k * 0.5^{n-k} = \binom{n}{k} \frac{1}{2^n}$$

- For winning at least $z = 180$ times, we need to compute:

$$P(k \geq z|n) \;\; = \;\; \sum_{i=z}^{n} P(i|n) = \frac{1}{2^{200}} \sum_{i=180}^{200} \binom{200}{i} \approx \frac{1.8125 \cdot 10^{27}}{1.6069 \cdot 10^{63}}$$

Example for Underlying Idea

- Question: How likely is it that I win at least 180 times if I did not cheat?
- In this case, the probabilities for heads and tails are $q = P(\texttt{head}) = P(\texttt{tail}) = 0.5$.
- Flipping a coin n times is a Bernoulli Process [9–11]
- The probability $P(k|n)$ to flip $k \in 0..n$ times heads (or tails) is thus:

$$P(k|n) = \binom{n}{k} 0.5^k * (1-0.5)^{n-k} = \binom{n}{k} 0.5^k * 0.5^{n-k} = \binom{n}{k} \frac{1}{2^n}$$

- For winning at least $z = 180$ times, we need to compute:

$$\begin{aligned} P(k \geq z|n) &= \sum_{i=z}^{n} P(i|n) = \frac{1}{2^{200}} \sum_{i=180}^{200} \binom{200}{i} \approx \frac{1.8125 \cdot 10^{27}}{1.6069 \cdot 10^{63}} \\ &\approx 0.000\,000\,000\,000\,000\,000\,000\,000\,000\,000\,001\,127\,930\,286\,459\,461\,553 \end{aligned}$$

- Question: How likely is it that I win at least 180 times if I did not cheat?
- In this case, the probabilities for heads and tails are $q = P(\texttt{head}) = P(\texttt{tail}) = 0.5$.
- Flipping a coin n times is a Bernoulli Process [9–11]
- The probability $P(k|n)$ to flip $k \in 0..n$ times heads (or tails) is thus:

$$P(k|n) = \binom{n}{k} 0.5^k * (1-0.5)^{n-k} = \binom{n}{k} 0.5^k * 0.5^{n-k} = \binom{n}{k} \frac{1}{2^n}$$

- For winning at least $z = 180$ times, we need to compute:

$$\begin{aligned} P(k \geq z|n) &= \sum_{i=z}^{n} P(i|n) = \frac{1}{2^{200}} \sum_{i=180}^{200} \binom{200}{i} \approx \frac{1.8125 \cdot 10^{27}}{1.6069 \cdot 10^{63}} \\ &\approx 1.279 \cdot 10^{-33} \end{aligned}$$

- Question: How likely is it that I win at least 180 times if I did not cheat?

Example for Underlying Idea

- Question: How likely is it that I win at least 180 times if I did not cheat?
- If the coin was an ideal coin, the chance that I win at least 180 out of 200 times is about $1 \cdot 10^{-33}$.

Example for Underlying Idea

- Question: How likely is it that I win at least 180 times if I did not cheat?
- If the coin was an ideal coin, the chance that I win at least 180 out of 200 times is about $1 \cdot 10^{-33}$.
- If you claim that I cheat, your chance to be wrong is about $1 \cdot 10^{-33}$.

Example for Underlying Idea

- Question: How likely is it that I win at least 180 times if I did not cheat?
- If the coin was an ideal coin, the chance that I win at least 180 out of 200 times is about $1 \cdot 10^{-33}$.
- If you claim that I cheat, your chance to be wrong is about $1 \cdot 10^{-33}$.
- Thus, if we cannot accept a chance p to be wrong higher than a significance level $\alpha = 1\%$, we can still say:

  The observation is significant, I did likely cheat.

- Compare two datasets $A = (a_1, a_2, \dots)$ and $B = (b_1, b_2, \dots)$ ...

Statistical Tests: General Idea

- Compare two datasets $A = (a_1, a_2, \dots)$ and $B = (b_1, b_2, \dots)$
- sampled from two distributions D_A and D_B ...

Statistical Tests: General Idea

- Compare two datasets $A = (a_1, a_2, \dots)$ and $B = (b_1, b_2, \dots)$
- sampled from two distributions D_A and D_B
- according to some statistical measure γ (e.g., mean, median, ...).

- Compare two datasets $A = (a_1, a_2, \dots)$ and $B = (b_1, b_2, \dots)$
- sampled from two distributions D_A and D_B
- according to some statistical measure γ (e.g., mean, median, ...).
- We observe that the (sample-based estimates of the) statistical measures are different: $\gamma(A) \neq \gamma(B)$.

- Compare two datasets $A = (a_1, a_2, \dots)$ and $B = (b_1, b_2, \dots)$
- sampled from two distributions D_A and D_B
- according to some statistical measure γ (e.g., mean, median, . . .).
- We observe that the (sample-based estimates of the) statistical measures are different: $\gamma(A) \neq \gamma(B)$.
- Question: If the observed difference in terms of γ representative for the real difference of D_A and D_B in terms of γ?

Statistical Tests: General Idea

- Compare two datasets $A = (a_1, a_2, \dots)$ and $B = (b_1, b_2, \dots)$
- sampled from two distributions D_A and D_B
- according to some statistical measure γ (e.g., mean, median, ...).
- We observe that the (sample-based estimates of the) statistical measures are different: $\gamma(A) \neq \gamma(B)$.
- Question: If the observed difference in terms of γ representative for the real difference of D_A and D_B in terms of γ?
- In other words: How likely am I to observe an experimental outcome at least as extreme as what I saw if actually $D_A = D_B$ (null hypothesis H_0)?

- Assumption: Observed difference in $\gamma(A)$ and $\gamma(B)$ are significant, i.e., likely to occur (H_1).

Statistical Tests: General Idea

- Assumption: Observed difference in $\gamma(A)$ and $\gamma(B)$ are significant, i.e., likely to occur (H_1).
- Counter-Assumption: Observed differences are result of random fluke (H_0).

- Assumption: Observed difference in $\gamma(A)$ and $\gamma(B)$ are significant, i.e., likely to occur (H_1).
- Counter-Assumption: Observed differences are result of random fluke (H_0). ... $D_A = D_B$ (and hence $\gamma(D_A) = \gamma(D_B)$)

Statistical Tests: General Idea

- Assumption: Observed difference in $\gamma(A)$ and $\gamma(B)$ are significant, i.e., likely to occur (H_1).
- Counter-Assumption: Observed differences are result of random fluke (H_0). … $D_A = D_B$ (and hence $\gamma(D_A) = \gamma(D_B)$)
- Compute the probability p of making an observation at least as extreme as $\gamma(A)$ or $\gamma(B)$ if H_0 is true

Statistical Tests: General Idea

- Assumption: Observed difference in $\gamma(A)$ and $\gamma(B)$ are significant, i.e., likely to occur (H_1).
- Counter-Assumption: Observed differences are result of random fluke (H_0). ... $D_A = D_B$ (and hence $\gamma(D_A) = \gamma(D_B)$)
- Compute the probability p of making an observation at least as extreme as $\gamma(A)$ or $\gamma(B)$ if H_0 is true
- If p is less than a significance level α (usually 1% or 2%), we can reject H_0 and accept H_1.

Statistical Tests: General Idea

- Assumption: Observed difference in $\gamma(A)$ and $\gamma(B)$ are significant, i.e., likely to occur (H_1).
- Counter-Assumption: Observed differences are result of random fluke (H_0). … $D_A = D_B$ (and hence $\gamma(D_A) = \gamma(D_B)$)
- Compute the probability p of making an observation at least as extreme as $\gamma(A)$ or $\gamma(B)$ if H_0 is true
- If p is less than a significance level α (usually 1% or 2%), we can reject H_0 and accept H_1.
- In this case, the optimization algorithm which produced the better set of key parameter measurements is significantly better.

Statistical Tests: General Idea

- Assumption: Observed difference in $\gamma(A)$ and $\gamma(B)$ are significant, i.e., likely to occur (H_1).
- Counter-Assumption: Observed differences are result of random fluke (H_0). … $D_A = D_B$ (and hence $\gamma(D_A) = \gamma(D_B)$)
- Compute the probability p of making an observation at least as extreme as $\gamma(A)$ or $\gamma(B)$ if H_0 is true
- If p is less than a significance level α (usually 1% or 2%), we can reject H_0 and accept H_1.
- In this case, the optimization algorithm which produced the better set of key parameter measurements is significantly better.
- Otherwise, there is no difference

A More Specific Example for Tests

- Let's do a small, simple example:

$$\begin{aligned} A &= (2, 5, 6, 7, 9, 10) \\ B &= (1, 3, 4, 8) \end{aligned}$$

A More Specific Example for Tests

- Let's do a small, simple example:

$$
\begin{aligned}
A &= (2, 5, 6, 7, 9, 10) \\
B &= (1, 3, 4, 8)
\end{aligned}
$$

- As statistical measure γ, we take the Expected Value.

A More Specific Example for Tests

- Let's do a small, simple example:

$$
\begin{aligned}
A &= (2, 5, 6, 7, 9, 10) \\
B &= (1, 3, 4, 8)
\end{aligned}
$$

- As statistical measure γ, we take the Expected Value.
- The expected values are estimated with the arithmetic means:

- Let's do a small, simple example:

$$\begin{aligned} A &= (2, 5, 6, 7, 9, 10) \\ B &= (1, 3, 4, 8) \end{aligned}$$

- As statistical measure γ, we take the Expected Value.
- The expected values are estimated with the arithmetic means:

$$\begin{aligned} \text{mean}(a) &= \frac{39}{6} = 6.5 \\ \text{mean}(b) &= \frac{16}{4} = 4 \end{aligned}$$

$$\begin{aligned} \mathrm{mean}(a) &= \frac{39}{6} = 6.5 \\ \mathrm{mean}(b) &= \frac{16}{4} = 4 \end{aligned}$$

- Question: Is the difference between $\mathrm{mean}(a)$ and $\mathrm{mean}(b)$ significant at $\alpha = 2\%$?

A More Specific Example

$$\begin{aligned}\mathrm{mean}(a) &= \frac{39}{6} = 6.5\\ \mathrm{mean}(b) &= \frac{16}{4} = 4\end{aligned}$$

- Question: Is the difference between $\mathrm{mean}(a)$ and $\mathrm{mean}(b)$ significant at $\alpha = 2\%$?
- Null Hypothesis H_0: A and B come from the same process, the difference is due to the random character of sampling

A More Specific Example

$$\begin{aligned} \mathrm{mean}(a) &= \frac{39}{6} = 6.5 \\ \mathrm{mean}(b) &= \frac{16}{4} = 4 \end{aligned}$$

- Question: Is the difference between $\mathrm{mean}(a)$ and $\mathrm{mean}(b)$ significant at $\alpha = 2\%$?
- Null Hypothesis H_0: A and B come from the same process, the difference is due to the random character of sampling
- Idea: If A and B are from the same distribution

A More Specific Example

$$\begin{aligned} \mathrm{mean}(a) &= \frac{39}{6} = 6.5 \\ \mathrm{mean}(b) &= \frac{16}{4} = 4 \end{aligned}$$

- Question: Is the difference between $\mathrm{mean}(a)$ and $\mathrm{mean}(b)$ significant at $\alpha = 2\%$?
- Null Hypothesis H_0: A and B come from the same process, the difference is due to the random character of sampling
- Idea: If A and B are from the same distribution, then
  1. We actually have one big sample $O = A \cup B$ from the *same* distribution

$$O = A \cup B = (1, 2, 3, 4, 5, 6, 7, 8, 9, 10)$$

A More Specific Example

$$\text{mean}(a) = \frac{39}{6} = 6.5$$
$$\text{mean}(b) = \frac{16}{4} = 4$$

- Question: Is the difference between $\text{mean}(a)$ and $\text{mean}(b)$ significant at $\alpha = 2\%$?
- Null Hypothesis H_0: A and B come from the same process, the difference is due to the random character of sampling
- Idea: If A and B are from the same distribution, then
  1. We actually have one big sample $O = A \cup B$ from the *same* distribution
  2. The observed division into A and B occured by accident or chance!

$$O = A \cup B = (1, 2, 3, 4, 5, 6, 7, 8, 9, 10)$$

A More Specific Example

$$\begin{aligned}\text{mean}(a) &= \frac{39}{6} = 6.5\\ \text{mean}(b) &= \frac{16}{4} = 4\end{aligned}$$

- Question: Is the difference between $\text{mean}(a)$ and $\text{mean}(b)$ significant at $\alpha = 2\%$?
- Null Hypothesis H_0: A and B come from the same process, the difference is due to the random character of sampling
- Idea: If A and B are from the same distribution, then
  1. We actually have one big sample $O = A \cup B$ from the *same* distribution
  2. The observed division into A and B occured by accident or chance!
  3. Any division C into two sets with 4 and 6 elements has the same probability

$$O = A \cup B = (1, 2, 3, 4, 5, 6, 7, 8, 9, 10)$$

A More Specific Example

$$O = A \cup B = (1, 2, 3, 4, 5, 6, 7, 8, 9, 10)$$

- Any division C into two sets with 4 and 6 elements has the same probability

A More Specific Example

$$O = A \cup B = (1, 2, 3, 4, 5, 6, 7, 8, 9, 10)$$

- Any division C into two sets with 4 and 6 elements has the same probability
- $|O| = 10$

A More Specific Example

$$O = A \cup B = (1, 2, 3, 4, 5, 6, 7, 8, 9, 10)$$

- Any division C into two sets with 4 and 6 elements has the same probability
- $|O| = 10$
- There are $\binom{10}{4} = 210$ different ways to draw 4 (or 6) elements from O

A More Specific Example

$$O = A \cup B = (1, 2, 3, 4, 5, 6, 7, 8, 9, 10)$$

- Any division C into two sets with 4 and 6 elements has the same probability
- $|O| = 10$
- There are $\binom{10}{4} = 210$ different ways to draw 4 (or 6) elements from O
- If H_0 holds, all have the same probability

A More Specific Example

$$O = A \cup B = (1, 2, 3, 4, 5, 6, 7, 8, 9, 10)$$

- Any division C into two sets with 4 and 6 elements has the same probability
- $|O| = 10$
- There are $\binom{10}{4} = 210$ different ways to draw 4 (or 6) elements from O
- If H_0 holds, all have the same probability
- Use a program to test the combinations

A More Specific Example

$$O = A \cup B = (1, 2, 3, 4, 5, 6, 7, 8, 9, 10)$$

Listing: Small tester program...

```
public class EnumerateAtLeastAsExtremeScenarios {
  public static void main(String[] args) {
    int meanLowerOrEqualTo4 = 0; //how often did we find a mean <= 4
    int totalCombinations   = 0; //total number of tested combinations

    for (int i = 10; i > 0; i--) {                // as O = numbers from 1 to 10
      for (int j = (i - 1); j > 0; j--) {         // we can conveniently iterate
        for (int k = (j - 1); k > 0; k--) {       // over all 4-element combos
          for (int l = (k - 1); l > 0; l--) {     // with 4 such nested loops
            if (((i + j + k + l) / 4.0) <= 4) {   // check for the extreme cases
              meanLowerOrEqualTo4++; }            // count the extreme case
            totalCombinations++;                  // add up combos, to verify
    } } } }

    System.out.println(meanLowerOrEqualTo4 + " " + totalCombinations);
  }
}
```

A More Specific Example

$$O = A \cup B = (1, 2, 3, 4, 5, 6, 7, 8, 9, 10)$$

- Any division C into two sets with 4 and 6 elements has the same probability
- $|O| = 10$
- There are $\binom{10}{4} = 210$ different ways to draw 4 (or 6) elements from O
- If H_0 holds, all have the same probability
- There are 27 such combinations with a mean of less or equal 4.

A More Specific Example

$$O = A \cup B = (1, 2, 3, 4, 5, 6, 7, 8, 9, 10)$$

- Any division C into two sets with 4 and 6 elements has the same probability
- $|O| = 10$
- There are $\binom{10}{4} = 210$ different ways to draw 4 (or 6) elements from O
- If H_0 holds, all have the same probability
- There are 27 such combinations with a mean of less or equal 4.
- The probability p to observe a constallation at least as extreme as A and B under H_0 is thus:

A More Specific Example

$$O = A \cup B = (1, 2, 3, 4, 5, 6, 7, 8, 9, 10)$$

- Any division C into two sets with 4 and 6 elements has the same probability
- $|O| = 10$
- There are $\binom{10}{4} = 210$ different ways to draw 4 (or 6) elements from O
- If H_0 holds, all have the same probability
- There are 27 such combinations with a mean of less or equal 4.
- The probability p to observe a constallation at least as extreme as A and B under H_0 is thus:

$$p = \frac{\#\text{cases } C : \mathrm{mean}(c) \leq \mathrm{mean}(b)}{\#\text{all cases}} = \frac{27}{210} = \frac{9}{70} \approx 0.1286$$

- Extreme cases into the other direction are the same:

$$O = A \cup B = (1, 2, 3, 4, 5, 6, 7, 8, 9, 10)$$

- Extreme cases into the other direction are the same:

$$
\begin{aligned}
O &= A \cup B = (1, 2, 3, 4, 5, 6, 7, 8, 9, 10) \\
\sum_{\forall o \in O} o &= \sum_{o=1}^{10} o = \frac{10(10+1)}{2} = 55
\end{aligned}
$$

A More Specific Example

- Extreme cases into the other direction are the same:

$$\begin{aligned} O &= A \cup B = (1,2,3,4,5,6,7,8,9,10) \\ \sum_{\forall o \in O} o &= \sum_{o=1}^{10} o = \frac{10(10+1)}{2} = 55 \\ \text{mean}(b) = \left(\frac{1}{4} \sum_{\forall b \in B} b\right) \leq 4 &\implies \left(\sum_{\forall b \in B} b\right) \leq 4 * 4 \leq 16 \end{aligned}$$

A More Specific Example

- Extreme cases into the other direction are the same:

$$\begin{aligned}
O &= A \cup B = (1, 2, 3, 4, 5, 6, 7, 8, 9, 10) \\
\sum_{\forall o \in O} o &= \sum_{o=1}^{10} o = \frac{10(10+1)}{2} = 55 \\
\text{mean}(b) = \left(\frac{1}{4} \sum_{\forall b \in B} b\right) \leq 4 &\implies \left(\sum_{\forall b \in B} b\right) \leq 4 * 4 \leq 16 \\
O = A \cup B &\implies \sum_{\forall a \in A} a = \left(\sum_{\forall o \in O} o\right) - \left(\sum_{\forall b \in B} b\right)
\end{aligned}$$

A More Specific Example

- Extreme cases into the other direction are the same:

$$
\begin{aligned}
O &= A \cup B = (1,2,3,4,5,6,7,8,9,10) \\
\sum_{\forall o \in O} o &= \sum_{o=1}^{10} o = \frac{10(10+1)}{2} = 55 \\
\text{mean}(b) = \left(\frac{1}{4}\sum_{\forall b \in B} b\right) \leq 4 &\implies \left(\sum_{\forall b \in B} b\right) \leq 4 * 4 \leq 16 \\
O = A \cup B &\implies \sum_{\forall a \in A} a = \left(\sum_{\forall o \in O} o\right) - \left(\sum_{\forall b \in B} b\right) \\
\sum_{\forall b \in B} b \leq 16 &\implies \left(\sum_{\forall a \in A} a\right) \geq 55 - 16 \geq 39
\end{aligned}
$$

- Extreme cases into the other direction are the same:

$$
\begin{aligned}
\sum_{\forall o \in O} o &= \sum_{o=1}^{10} o = \frac{10(10+1)}{2} = 55 \\
\text{mean}(b) = \left(\frac{1}{4} \sum_{\forall b \in B} b\right) \leq 4 &\implies \left(\sum_{\forall b \in B} b\right) \leq 4 * 4 \leq 16 \\
O = A \cup B &\implies \sum_{\forall a \in A} a = \left(\sum_{\forall o \in O} o\right) - \left(\sum_{\forall b \in B} b\right) \\
\sum_{\forall b \in B} b \leq 16 &\implies \left(\sum_{\forall a \in A} a\right) \geq 55 - 16 \geq 39 \\
\text{mean}(a) &= \frac{1}{6}\left(\sum_{\forall a \in A} a\right)
\end{aligned}
$$

A More Specific Example

- Extreme cases into the other direction are the same:

$$\begin{aligned}
\text{mean}(b) = \left(\frac{1}{4}\sum_{\forall b \in B} b\right) \leq 4 \quad &\Longrightarrow \quad \left(\sum_{\forall b \in B} b\right) \leq 4 * 4 \leq 16 \\
O = A \cup B \quad &\Longrightarrow \quad \sum_{\forall a \in A} a = \left(\sum_{\forall o \in O} o\right) - \left(\sum_{\forall b \in B} b\right) \\
\sum_{\forall b \in B} b \leq 16 \quad &\Longrightarrow \quad \left(\sum_{\forall a \in A} a\right) \geq 55 - 16 \geq 39 \\
\text{mean}(a) \quad &= \quad \frac{1}{6}\left(\sum_{\forall a \in A} a\right) \\
\text{mean}(b) \leq 4 \quad &\Rightarrow \quad \text{mean}(a) \geq \frac{39}{6} \geq 6.5
\end{aligned}$$

- Extreme cases into the other direction are the same:

$$\text{mean}(b) = \left(\frac{1}{4}\sum_{\forall b \in B} b\right) \leq 4 \quad \Longrightarrow \quad \left(\sum_{\forall b \in B} b\right) \leq 4 * 4 \leq 16$$

$$O = A \cup B \quad \Longrightarrow \quad \sum_{\forall a \in A} a = \left(\sum_{\forall o \in O} o\right) - \left(\sum_{\forall b \in B} b\right)$$

$$\sum_{\forall b \in B} b \leq 16 \quad \Longrightarrow \quad \left(\sum_{\forall a \in A} a\right) \geq 55 - 16 \geq 39$$

$$\text{mean}(a) \quad = \quad \frac{1}{6}\left(\sum_{\forall a \in A} a\right)$$

$$\text{mean}(b) \leq 4 \quad \Rightarrow \quad \text{mean}(a) \geq \frac{39}{6} \geq 6.5$$

- So we could have also done the test the other way around with the same result!

A More Specific Example

- The probability p to observe a constallation at least as extreme as A or B under H_0 is thus:

$$p = \frac{\#\text{cases } C : \mathrm{mean}(c) \leq \mathrm{mean}(b)}{\#\text{all cases}} = \frac{27}{210} = \frac{9}{70} \approx 0.1286$$

A More Specific Example

- The probability p to observe a constallation at least as extreme as A or B under H_0 is thus:

$$p = \frac{\#\text{cases } C : \mathrm{mean}(c) \leq \mathrm{mean}(b)}{\#\text{all cases}} = \frac{27}{210} = \frac{9}{70} \approx 0.1286$$

- If we claim that A and B are from distributions with different means. . .

A More Specific Example

- The probability p to observe a constallation at least as extreme as A or B under H_0 is thus:

$$p = \frac{\#\text{cases } C : \mathrm{mean}(c) \leq \mathrm{mean}(b)}{\#\text{all cases}} = \frac{27}{210} = \frac{9}{70} \approx 0.1286$$

- If we claim that A and B are from distributions with different means...
- ...we are wrong with probability $p \approx 0.13$

A More Specific Example

- The probability p to observe a constallation at least as extreme as A or B under H_0 is thus:

$$p = \frac{\#\text{cases } C : \mathrm{mean}(c) \leq \mathrm{mean}(b)}{\#\text{all cases}} = \frac{27}{210} = \frac{9}{70} \approx 0.1286$$

- If we claim that A and B are from distributions with different means...
- ...we are wrong with probability $p \approx 0.13$
- At a significance level of $\alpha = 2\%$, the means of A and B are not significantly different! $(2\% < 0.13)$

A More Specific Example

- The probability p to observe a constallation at least as extreme as A or B under H_0 is thus:

$$p = \frac{\#\text{cases } C : \mathrm{mean}(c) \leq \mathrm{mean}(b)}{\#\text{all cases}} = \frac{27}{210} = \frac{9}{70} \approx 0.1286$$

- If we claim that A and B are from distributions with different means...
- ...we are wrong with probability $p \approx 0.13$
- At a significance level of $\alpha = 2\%$, the means of A and B are not significantly different! $(2\% < 0.13)$
- This was an example on how statistical tests work APPROXIMATELY

A More Specific Example

- The probability p to observe a constallation at least as extreme as A or B under H_0 is thus:

$$p = \frac{\#\text{cases } C : \text{mean}(c) \leq \text{mean}(b)}{\#\text{all cases}} = \frac{27}{210} = \frac{9}{70} \approx 0.1286$$

- If we claim that A and B are from distributions with different means. . .
- . . . we are wrong with probability $p \approx 0.13$
- At a significance level of $\alpha = 2\%$, the means of A and B are not significantly different! $(2\% < 0.13)$
- This was an example on how statistical tests work APPROXIMATELY
- The method here is only feasible for small sample sets, real tests are more sophisticated

- Two types of tests:

- Two types of tests:
  1. Parametric Tests

- Two types of tests:
  1. Parametric Tests
     - Assume that the data samples follow a certain distribution

- Two types of tests:
  1. Parametric Tests
     - Assume that the data samples follow a certain distribution
     - Examples [14]: t-test (assumes normal distribution)

Statistical Tests: Types

- Two types of tests:
  1. Parametric Tests
     - Assume that the data samples follow a certain distribution
     - Examples [14]: t-test (assumes normal distribution)
     - The distribution of the data we measure is unknown. . .

Statistical Tests: Types

- Two types of tests:
  1. Parametric Tests
     - Assume that the data samples follow a certain distribution
     - Examples [14]: t-test (assumes normal distribution)
     - The distribution of the data we measure is unknown…
     - …and usually not normal, see further example on statistical measures.

- Two types of tests:
  1. Parametric Tests
     - Assume that the data samples follow a certain distribution
     - Examples [14]: t-test (assumes normal distribution)
     - The distribution of the data we measure is unknown...
     - ...and usually not normal, see further example on statistical measures.
     - The condition for using such tests cannot be met (known distribution)

- Two types of tests:
  1. Parametric Tests
     - Assume that the data samples follow a certain distribution
     - Examples [14]: t-test (assumes normal distribution)
     - The distribution of the data we measure is unknown...
     - ...and usually not normal, see further example on statistical measures.
     - The condition for using such tests cannot be met (known distribution)
     - Parametric Tests cannot be used here!

- Two types of tests:
  1. Parametric Tests
  2. Non-Parametric Tests

- Two types of tests:
  1. Parametric Tests
  2. Non-Parametric Tests
     - Make no assumption about the distribution from which the data was sampled.

- Two types of tests:
  1. Parametric Tests
  2. Non-Parametric Tests
     - Make no assumption about the distribution from which the data was sampled.
     - Examples [5]: Mann-Whitney U Test [15–18], Fisher's Exact Test [19], Sign Test [16, 20], Randomization Test [21–24], Wilcoxon's Signed Rank Test [25].

Statistical Tests: Types

- Two types of tests:
  1. Parametric Tests
  2. Non-Parametric Tests
     - Make no assumption about the distribution from which the data was sampled.
     - Examples [5]: Mann-Whitney U Test [15–18], Fisher's Exact Test [19], Sign Test [16, 20], Randomization Test [21–24], Wilcoxon's Signed Rank Test [25].
     - These tests are more robust (less assumptions)

- Two types of tests:
  1. Parametric Tests
  2. Non-Parametric Tests
     - Make no assumption about the distribution from which the data was sampled.
     - Examples [5]: Mann-Whitney U Test [15–18], Fisher's Exact Test [19], Sign Test [16, 20], Randomization Test [21–24], Wilcoxon's Signed Rank Test [25].
     - These tests are more robust (less assumptions)
     - This is the kind of test we want to use!

- Two types of tests:
  1. Parametric Tests
  2. Non-Parametric Tests
     - Make no assumption about the distribution from which the data was sampled.
     - Examples [5]: Mann-Whitney U Test [15–18], Fisher's Exact Test [19], Sign Test [16, 20], Randomization Test [21–24], Wilcoxon's Signed Rank Test [25].
     - These tests are more robust (less assumptions)
     - This is the kind of test we want to use!
     - They work similar to the previous test example, but with larger sample sizes

Compare $N \geq 2$ Algorithms

- For comparing $N \geq 2$ algorithms, we can compare any two algorithms with each other

Compare $N \geq 2$ Algorithms

- For comparing $N \geq 2$ algorithms, we can compare any two algorithms with each other
- N Algorithms $\Rightarrow k = N(N-1)/2$ statistical tests (e.g., Mann-Whitney U)

Compare $N \geq 2$ Algorithms

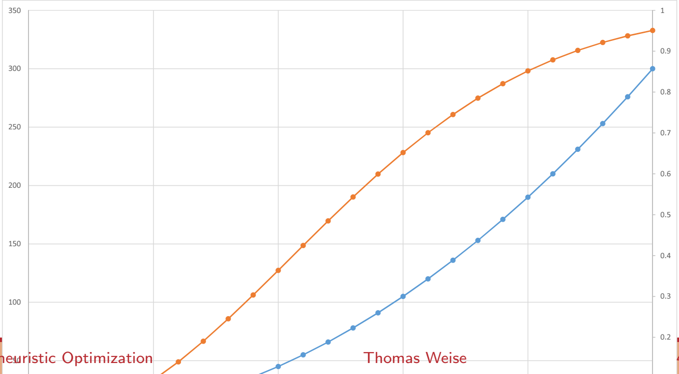
- For comparing $N \geq 2$ algorithms, we can compare any two algorithms with each other
- N Algorithms $\Rightarrow k = N(N-1)/2$ statistical tests (e.g., Mann-Whitney U)
- k tests and each with error proability α

Compare $N \geq 2$ Algorithms

- For comparing $N \geq 2$ algorithms, we can compare any two algorithms with each other
- N Algorithms $\Rightarrow k = N(N-1)/2$ statistical tests (e.g., Mann-Whitney U)
- k tests and each with error proability $\alpha \Longrightarrow$ total probability E to make error $E = 1 - ((1-\alpha)^k)$

Compare $N \geq 2$ Algorithms

- For comparing $N \geq 2$ algorithms, we can compare any two algorithms with each other
- N Algorithms $\Rightarrow k = N(N-1)/2$ statistical tests
- k tests and each with error proability $\alpha \Longrightarrow$ total probability E to make error $E = 1 - ((1-\alpha)^k)$



- For comparing $N \geq 2$ algorithms, we can compare any two algorithms with each other
- N Algorithms $\Rightarrow k = N(N-1)/2$ statistical tests
- k tests and each with error proability $\alpha \Longrightarrow$ total probability E to make error $E = 1 - ((1-\alpha)^k)$
- Correction needed: Bonferroni correction [26] or (better) post-hoc methods [27, 28]

Compare $N \geq 2$ Algorithms

- For comparing $N \geq 2$ algorithms, we can compare any two algorithms with each other
- N Algorithms $\Rightarrow k = N(N-1)/2$ statistical tests
- k tests and each with error proability $\alpha \Longrightarrow$ total probability E to make error $E = 1 - ((1-\alpha)^k)$
- Correction needed: Bonferroni correction [26] or (better) post-hoc methods [27, 28]
- Idea of Bonferroni correction: Use $\alpha' = \alpha/k$ as significance level instead of α, then the overall probability E to make an error will remain $E \leq \alpha$.

- So now we can compare N datasets.

Compare $N \geq 2$ Algorithms

- So now we can compare N datasets.
- Most common representation of results: Table

Compare $N \geq 2$ Algorithms

- So now we can compare N datasets.
- Most common representation of results: Table

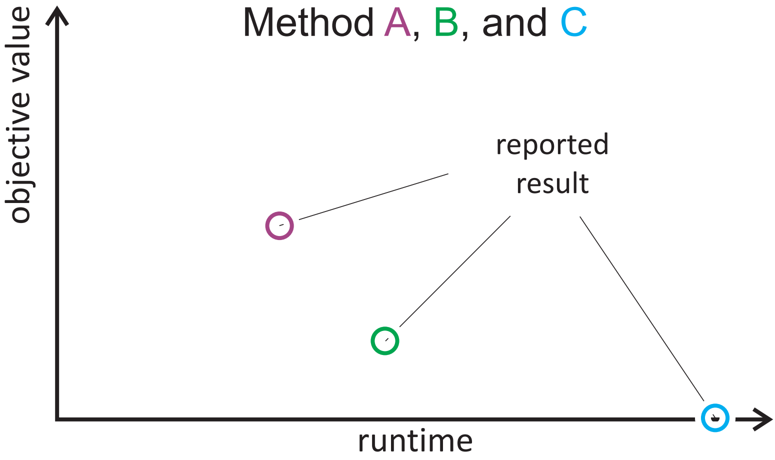
	P_1	P_2	P_3	P_4	P_5	P_6	P_7	P_8
P_1		+	+	+	+	+	0	+
P_2			0	+	0	0	–	+
P_3				+	–	0	–	0
P_4					–	–	–	–
P_5						0	–	0
P_6							–	+
P_7								+

- + in the i^{th} row and j^{th} column means that process P_i has significantly better outputs than process P_j
- – stands for significantly worse outputs
- 0 symbolizes that no significant difference could be detected

- Literature usually reports tuples "(instance, result, runtime)"

- Literature usually reports tuples "(instance, result, runtime)"
  - Example: "8th DIMACS Challenge: The Traveling Salesman Problem" [29–31]

- Literature usually reports tuples "(instance, result, runtime)"
  - Example: "8th DIMACS Challenge: The Traveling Salesman Problem" [29–31]
- Papers often use a different termination criterion


Method A, B, and C
objective value
reported
result
runtime

The question of termination

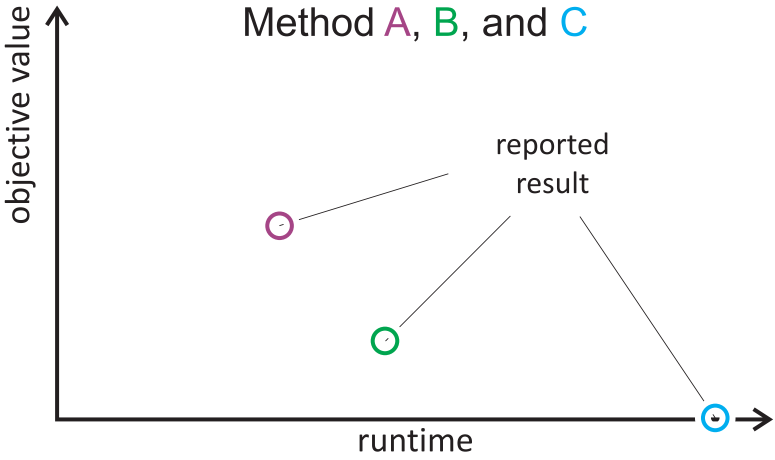
- Literature usually reports tuples "(instance, result, runtime)"
    - Example: "8th DIMACS Challenge: The Traveling Salesman Problem" [29–31]
- Problem: Papers often use a different termination criterion

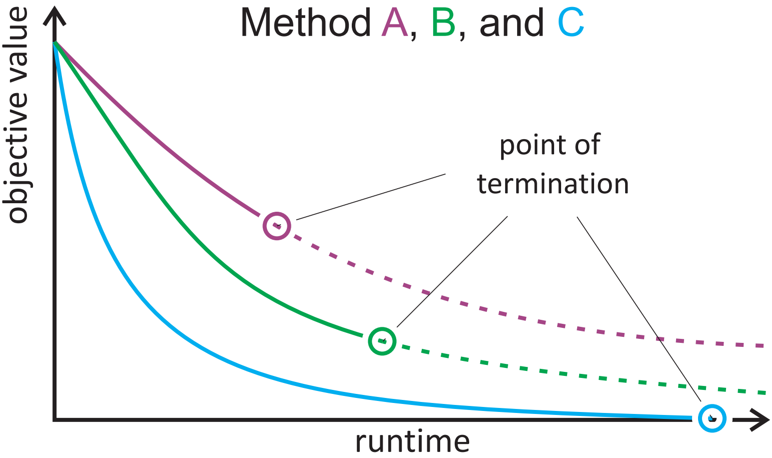
The question of termination

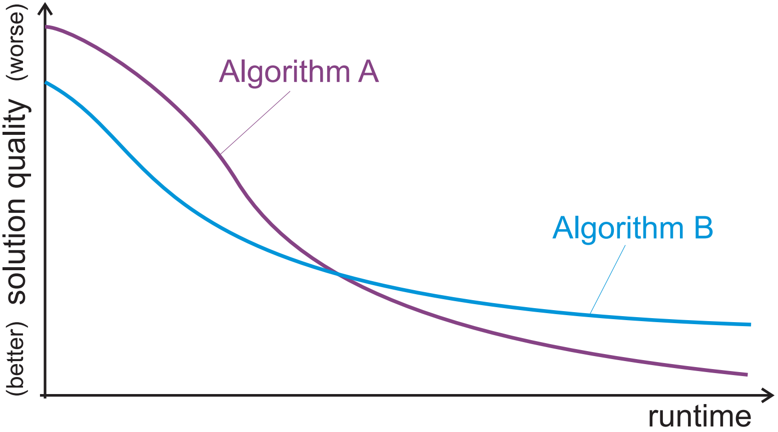
- Literature usually reports tuples "(instance, result, runtime)"
  - Example: "8th DIMACS Challenge: The Traveling Salesman Problem" [29–31]
- Problem: Papers often use a different termination criterion
- Anytime Algorithms [32]

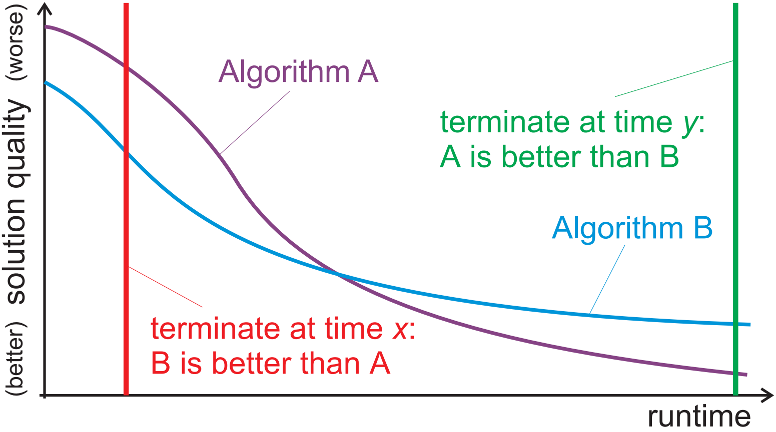
The question of termination

- Literature usually reports tuples "(instance, result, runtime)"
  - Example: "8th DIMACS Challenge: The Traveling Salesman Problem" [29–31]
- Problem: Papers often use a different termination criterion
- Anytime Algorithms [32]: Always have approximate solution, refine it iteratively


Method A, B, and C
objective value
reported
result
runtime


Method A, B, and C
objective value
point of
termination
runtime


Algorithm A
Algorithm B
(better) solution quality (worse)
runtime



solution quality
(worse)
(better)
Algorithm A
terminate at time y:
A is better than B
Algorithm B
terminate at time x:
B is better than A
runtime

The question of termination

- Literature usually reports tuples "(instance, result, runtime)"
  - Example: "8$^{\text{th}}$ DIMACS Challenge: The Traveling Salesman Problem" [29–31]
- Problem: Papers often use a different termination criterion
- Anytime Algorithms [32]: Always have approximate solution, refine it iteratively
- One measure point per run or instance does not tell the whole story!

The question of termination

- Literature usually reports tuples "(instance, result, runtime)"
  - Example: "8$^{\text{th}}$ DIMACS Challenge: The Traveling Salesman Problem" [29–31]
- Problem: Papers often use a different termination criterion
- Anytime Algorithms [32]: Always have approximate solution, refine it iteratively
- One measure point per run or instance does not tell the whole story!
- Using statistical tests cannot solve this issue (still: at one point in time).

The question of termination

- Literature usually reports tuples "(instance, result, runtime)"
  - Example: "8th DIMACS Challenge: The Traveling Salesman Problem" [29–31]
- Problem: Papers often use a different termination criterion
- Anytime Algorithms [32]: Always have approximate solution, refine it iteratively
- One measure point per run or instance does not tell the whole story!
- Using statistical tests cannot solve this issue (still: at one point in time).
- We Should have the "whole curves"!

The question of termination

- Literature usually reports tuples "(instance, result, runtime)"
  - Example: "8th DIMACS Challenge: The Traveling Salesman Problem" [29–31]
- Problem: Papers often use a different termination criterion
- Anytime Algorithms [32]: Always have approximate solution, refine it iteratively
- One measure point per run or instance does not tell the whole story!
- Using statistical tests cannot solve this issue (still: at one point in time).
- We Should have the "whole curves"! ... ideally median curves over several runs!

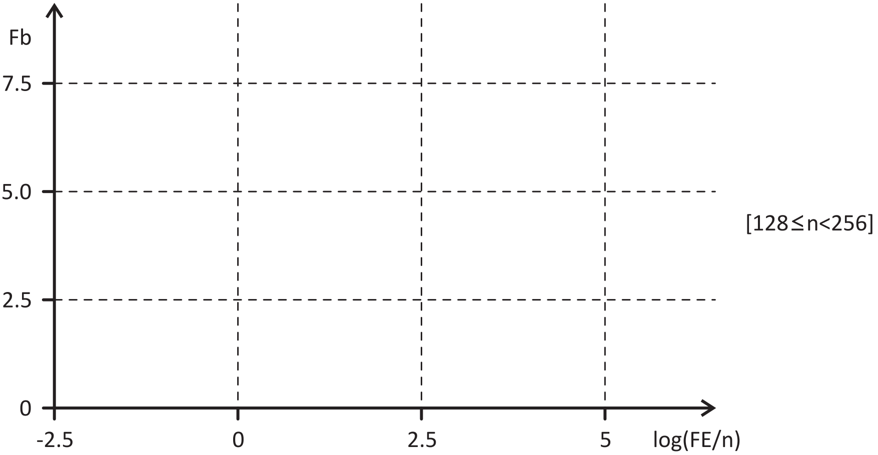
- Plot the best objective value reached over time

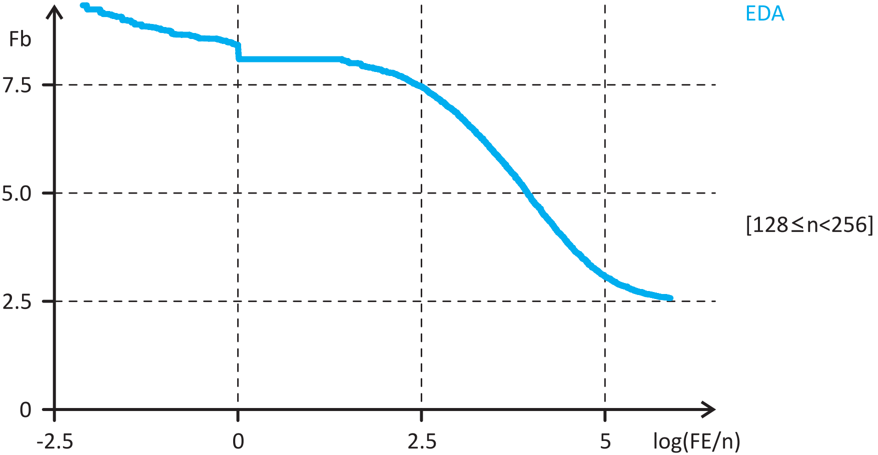
- Plot the median of the best objective value reached over time, over all runs

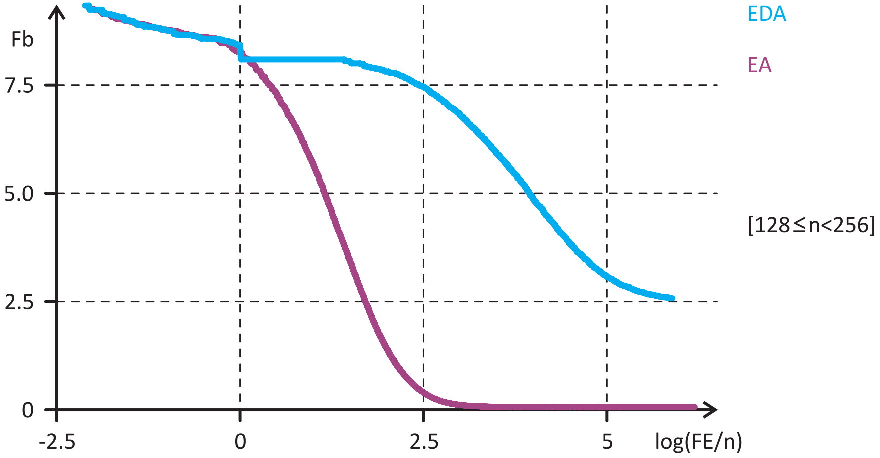
- Plot the median of the best objective value reached over time, over all runs, on a given benchmark instance

Progress Diagrams

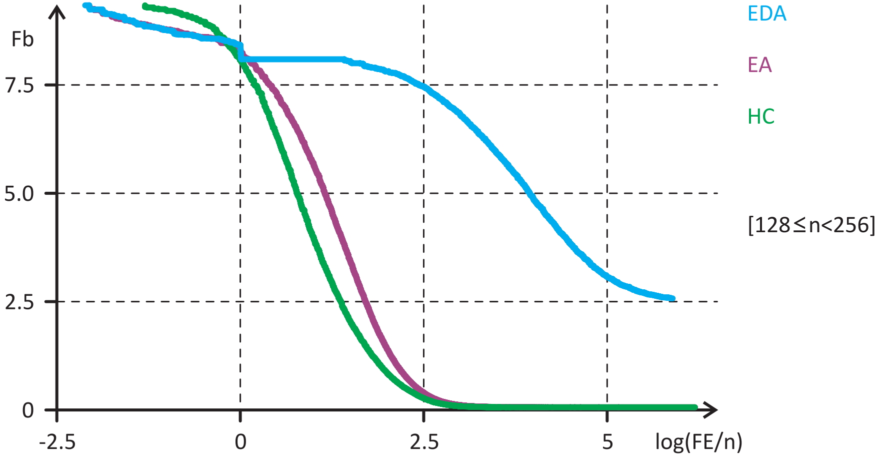
- Plot the median of the best objective value reached over time, over all runs, on a given benchmark instance or aggregated over several instances

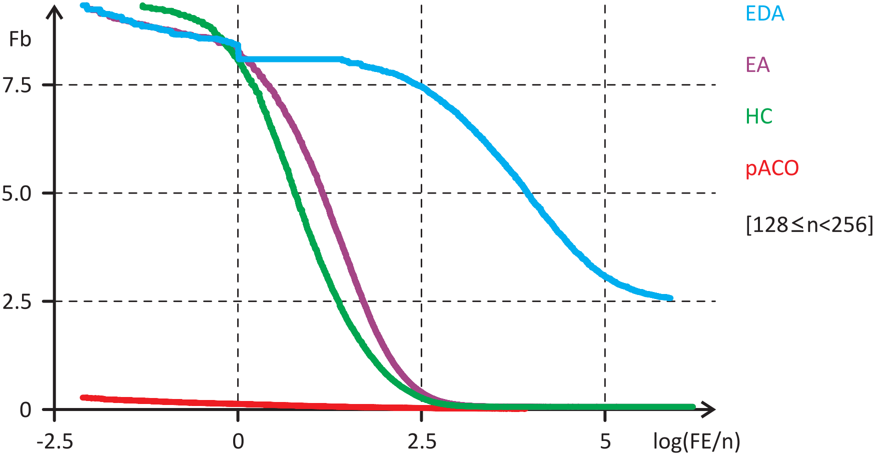

Fb
7.5
5.0
2.5
0
-2.5
0
2.5
5
log(FE/n)
[128≤n<256]


EDA
Fb
7.5
5.0
2.5
0
-2.5
0
2.5
5
log(FE/n)
[128≤n<256]


EDA
EA
Fb
7.5
5.0
2.5
0
-2.5
0
2.5
5
log(FE/n)
[128≤n<256]

Progress Diagrams




Fb
7.5
5.0
2.5
0
-2.5
0
2.5
5
log(FE/n)
EDA
EA
HC
pACO
[128≤n<256]

- Plot the median of the best objective value reached over time, over all runs, on a given benchmark instance or aggregated over several instances
- The smaller the value, the better

Section Outline

- Don’t apply algorithms to just a single problem instance!

- Don't apply algorithms to just a single problem instance!
- Apply algorithm to multiple different instances.
- Apply algorithm to different problems.

- Don't apply algorithms to just a single problem instance!
- Apply algorithm to multiple different instances.
- Apply algorithm to different problems.
- Best: Use existing benchmark suite $\Rightarrow$ results can easily be compared with literature.

- Don't apply algorithms to just a single problem instance!
- Apply algorithm to multiple different instances.
- Apply algorithm to different problems.
- Best: Use existing benchmark suite $\Rightarrow$ results can easily be compared with literature.
- Of course, results cannot simply be "added"

- Don't apply algorithms to just a single problem instance!
- Apply algorithm to multiple different instances.
- Apply algorithm to different problems.
- Best: Use existing benchmark suite $\Rightarrow$ results can easily be compared with literature.
- Of course, results cannot simply be "added"
  1. Evaluation by discussion

- Don't apply algorithms to just a single problem instance!
- Apply algorithm to multiple different instances.
- Apply algorithm to different problems.
- Best: Use existing benchmark suite $\Rightarrow$ results can easily be compared with literature.
- Of course, results cannot simply be "added"
  1. Evaluation by discussion
  2. Evaluation with value-neutral point system, e.g., the point system of Formula 1 car racing

Benchmark Suites

- Combinatorial Problems
- Bit Strings
- Numerical Problems
- Multi-Objective Optimization
- Dynamic Optimization
- Data Mining
- Genetic Programming

- Combinatorial Problems
  - Traveling Salesman Problem [33–36]
  - CARPLib [37] (Capacitated Arc Routing Problems)
  - Bin Packing [34–36, 38]
  - SATLIB [39] (Satisfiability Problems)
  - Vehicle routing Problem [40–42]
  - general combinatorial Operations Research problems [43]
- Bit Strings
- Numerical Problems
- Multi-Objective Optimization
- Dynamic Optimization
- Data Mining
- Genetic Programming

- Combinatorial Problems
- Bit Strings
  - NK-Landscapes [44–49] and similar [50–52]
  - Royal Road [53–62]
  - Tunable Benchmark Model [63]
  - Long Path Problems [64, 65]
  - Spin-Glass Models [66]
  - BinInt Problem [67]
  - OneMax Problem [68–74]
- Numerical Problems
- Multi-Objective Optimization
- Dynamic Optimization
- Data Mining
- Genetic Programming

Benchmark Suites

- Combinatorial Problems
- Bit Strings
- Numerical Problems
  - BBOB [2, 3] (Black-Box Continuous Optimization)
  - CEC SS on Real-Valued Optimization [75, 76]
  - CEC SS on Large-Scale Optimization [4, 77]
- Multi-Objective Optimization
- Dynamic Optimization
- Data Mining
- Genetic Programming

Benchmark Suites

- Combinatorial Problems
- Bit Strings
- Numerical Problems
- Multi-Objective Optimization
  - CEC SS Multi-Objective Optimization [78, 79]
  - CEC SS Constraint Optimization [80]
  - Problems by Deb et al. [81]
- Dynamic Optimization
- Data Mining
- Genetic Programming

Benchmark Suites

- Combinatorial Problems
- Bit Strings
- Numerical Problems
- Multi-Objective Optimization
- Dynamic Optimization
  - Moving Peaks Benchmark [82] (real-valued)
- Data Mining
- Genetic Programming

Benchmark Suites

- Combinatorial Problems
- Bit Strings
- Numerical Problems
- Multi-Objective Optimization
- Dynamic Optimization
- Data Mining
    - UCI Machine Learning Repository [83] contains e.g.,
    - Iris Dataset [84, 85]
    - Wisconsin Breast Cancer Dataset [86]
    - Heart Disease Dataset [87]
- Genetic Programming

- Combinatorial Problems
- Bit Strings
- Numerical Problems
- Multi-Objective Optimization
- Dynamic Optimization
- Data Mining
- Genetic Programming
  - Artificial Ant [88–90],
  - Lawn Mower, Symbolic Regression [90]
  - Greatest Common Divisor Problem [5, 91]
  - Royal Tree Problem [92]
  - . . . and others [93]

Section Outline

Summary

- The optimization algorithms we consider in this lecture are randomized.

Summary

- The optimization algorithms we consider in this lecture are randomized.
- Comparing them must be done in a statistical way using data from multiple runs

Summary

- The optimization algorithms we consider in this lecture are randomized.
- Comparing them must be done in a statistical way using data from multiple runs
- Two key performance indicators

Summary

- The optimization algorithms we consider in this lecture are randomized.
- Comparing them must be done in a statistical way using data from multiple runs
- Two key performance indicators:
  1. best result after fixed number of FEs/runtime

Summary

- The optimization algorithms we consider in this lecture are randomized.
- Comparing them must be done in a statistical way using data from multiple runs
- Two key performance indicators:
  1. best result after fixed number of FEs/runtime
  2. number of FEs/runtime needed to get certain result

Summary

- The optimization algorithms we consider in this lecture are randomized.
- Comparing them must be done in a statistical way using data from multiple runs
- Two key performance indicators:
  1. best result after fixed number of FEs/runtime
  2. number of FEs/runtime needed to get certain result
- For every single algorithm/configuration, compute

Summary

- The optimization algorithms we consider in this lecture are randomized.
- Comparing them must be done in a statistical way using data from multiple runs
- Two key performance indicators:
  1. best result after fixed number of FEs/runtime
  2. number of FEs/runtime needed to get certain result
- For every single algorithm/configuration, compute:
  1. median of key performance indicators

Summary

- The optimization algorithms we consider in this lecture are randomized.
- Comparing them must be done in a statistical way using data from multiple runs
- Two key performance indicators:
  1. best result after fixed number of FEs/runtime
  2. number of FEs/runtime needed to get certain result
- For every single algorithm/configuration, compute:
  1. median of key performance indicators
  2. quartiles or top/bottom 1% quantile

Summary

- The optimization algorithms we consider in this lecture are randomized.
- Comparing them must be done in a statistical way using data from multiple runs
- Two key performance indicators:
  1. best result after fixed number of FEs/runtime
  2. number of FEs/runtime needed to get certain result
- For every single algorithm/configuration, compute:
  1. median of key performance indicators
  2. quartiles or top/bottom 1% quantile
  3. don't trust arithmetic mean or standard deviation

Summary

- The optimization algorithms we consider in this lecture are randomized.
- Comparing them must be done in a statistical way using data from multiple runs
- Two key performance indicators:
  1. best result after fixed number of FEs/runtime
  2. number of FEs/runtime needed to get certain result
- For every single algorithm/configuration, compute:
  1. median of key performance indicators
  2. quartiles or top/bottom 1% quantile
  3. don't trust arithmetic mean or standard deviation
- Do not only collect one data sample per run, try to plot progress curves

Summary

- The optimization algorithms we consider in this lecture are randomized.
- Comparing them must be done in a statistical way using data from multiple runs
- Two key performance indicators:
  1. best result after fixed number of FEs/runtime
  2. number of FEs/runtime needed to get certain result
- For every single algorithm/configuration, compute:
  1. median of key performance indicators
  2. quartiles or top/bottom 1% quantile
  3. don't trust arithmetic mean or standard deviation
- Do not only collect one data sample per run, try to plot progress curves
- For given problem class: Look for well-known benchmarks!

谢谢
Thank you

Thomas Weise [汤卫思]
tweise@hfuu.edu.cn
http://iao.hfuu.edu.cn

Hefei University, South Campus 2
Institute of Applied Optimization
Shushan District, Hefei, Anhui,
China


Caspar David Friedrich, “Der Wanderer über dem Nebelmeer”, 1818
http://en.wikipedia.org/wiki/Wanderer_above_the_Sea_of_Fog

Bibliography

Bibliography I

1. Thomas Weise, Li Niu, and Ke Tang. Aoab – automated optimization algorithm benchmarking. In *Black Box Optimization Benchmarking (BBOB'10), Companion Publication of the Genetic and Evolutionary Computation Conference (GECCO'10 Companion)*, pages 1479–1486, Portland, OR, USA: Portland Marriott Downtown Waterfront Hotel, July 7, 2010. New York, NY, USA: ACM Press. doi: 10.1145/1830761.1830763.
2. Nikolaus Hansen, Anne Auger, Steffen Finck, and Raymond Ros. Real-parameter black-box optimization benchmarking 2010: Experimental setup. Rapports de Recherche 7215, Institut National de Recherche en Informatique et en Automatique (INRIA), March 9, 2010. URL http://hal.inria.fr/docs/00/46/24/81/PDF/RR-7215.pdf.
3. Nikolaus Hansen, Anne Auger, Steffen Finck, and Raymond Ros. Real-parameter black-box optimization benchmarking 2009: Experimental setup. Rapports de Recherche RR-6828, Institut National de Recherche en Informatique et en Automatique (INRIA), October 16, 2009. URL http://hal.archives-ouvertes.fr/inria-00362649/en/. Version 3.
4. Ke Tang, Xiaodong Li, Ponnuthurai Nagaratnam Suganthan, Zhenyu Yang, and Thomas Weise. Benchmark functions for the cec'2010 special session and competition on large-scale global optimization. Technical report, Hefei, Anhui, China: University of Science and Technology of China (USTC), School of Computer Science and Technology, Nature Inspired Computation and Applications Laboratory (NICAL), January 8, 2010.
5. Thomas Weise. *Global Optimization Algorithms – Theory and Application*. Germany: it-weise.de (self-published), 2009. URL http://www.it-weise.de/projects/book.pdf.
6. Illustration of median versus mean, June 17, 2005. URL http://www.bvmarketdata.com/pdf/Median-Mean.pdf.
7. Mark Thoma. Mean vs. median income growth, September 2, 2011. URL http://economistsview.typepad.com/economistsview/2011/09/mean-vs-median-income-growth.html.
8. Why use the median and not the mean?, 2009. URL http://www.cms.murdoch.edu.au/areas/maths/statsnotes/samplestats/medianmore.html.
9. Jakob Bernoulli. *Ars Conjectandi, Opus Posthumum. Accedit Tractatus de Seriebus Infinitis, et Epistola Gallicé Scripta de Ludo Pilae Reticularis*. Basel, Switzerland: Thurneysen Brothers, 1713.
10. Carl W. Helstrom. *Probability and Stochastic Processes for Engineers*. New York, NY, USA: Macmillan Publishers Co., 1984. ISBN 0-02-353560-1 and 9780023535604. URL http://books.google.de/books?id=nYRRAAAAMAAJ.
11. Dimitri P. Bertsekas and John N. Tsitsiklis. *Introduction to Probability*. Nashua, NH, USA: Athena Scientific – A Publisher of Scientific Books, 2nd edition, July 2008. ISBN 978-1-886529-23-6.
12. Monte Ohrt. *Binomial Coefficient Calculator*. Lincoln, NE, USA, December 13, 2007. URL http://www.ohrt.com/odds/binomial.php.

Bibliography II

13. Gary Darby. *Big Floating Point*. Floyd, VA, USA: DelphiForFun.org, October 16, 2009. URL `http://delphiforfun.org/programs/Library/BigFloat.htm`.
14. Shaun Burke. Missing values, outliers, robust statistics & non-parametric methods. *LC.GC Europe Online Supplement*, 1 (2):19–24, January 2, 2001. URL `http://chromatographyonline.findanalytichem.com/lcgceurope/data/articlestandard/lcgceurope/502001/4509/article.pdf`.
15. Henry B. Mann and Donald R. Whitney. On a test of whether one of two random variables is stochastically larger than the other. *The Annals of Mathematical Statistics*, 18(1):50–60, March 1947. doi: 10.1214/aoms/1177730491. URL `http://projecteuclid.org/euclid.aoms/1177730491`.
16. Sidney Siegel and N. John Castellan Jr. *Nonparametric Statistics for The Behavioral Sciences*. Humanities/Social Sciences/Languages. New York, NY, USA: McGraw-Hill, 1956. ISBN 0-07-057357-3, 070573434X, 978-0070573574, and 9780705734349. URL `http://books.google.de/books?id=MO1lGwAACAAJ`.
17. L. C. Dinneen and B. C. Blakesley. Algorithm as 62: A generator for the sampling distribution of the mann-whitney u statistic. *Journal of the Royal Statistical Society: Series C – Applied Statistics*, 22(2):269–273, 1973. URL `http://www.jstor.org/stable/2346934`.
18. N. Neumann. Some procedures for calculating the distributions of elementary non-parametric test statistics. *Statistical Software Newsletter (SSN)*, 14(3), 1988.
19. Sir Ronald Aylmer Fisher. On the interpretation of χ^2 from contingency tables, and the calculation of p. *Journal of the Royal Statistical Society*, 85:87–94, 1922. URL `http://hdl.handle.net/2440/15173`.
20. Lorenz Gygax. Statistik für nutztierethologen – einführung in die statistische denkweise: Was ist, was macht ein statistischer test?, June 2003. URL `http://www.proximate-biology.ch/documents/introEtho.pdf`.
21. Sir Ronald Aylmer Fisher. "the coefficient of racial likeness" and the future of craniometry. *The Journal of the Royal Anthropological Institute of Great Britain and Ireland*, 66:57–63, January–July 1936. URL `http://digital.library.adelaide.edu.au/dspace/bitstream/2440/15228/1/141.pdf`.
22. Jürgen Bortz, Gustav Adolf Lienert, and Klaus Boehnke. *Verteilungsfreie Methoden in der Biostatistik*. Springer-Lehrbuch. Heidelberg, Germany: Springer Medizin Verlag, 3., korrigierte auflage edition, 2008. ISBN 3445110344, 3-540-67590-6, 3-540-74706-0, 9783445110343, 978-3-540-67590-7, 978-3-540-74706-2, and 978-3-540-74707-9. doi: 10.1007/978-3-540-74707-9. URL `http://www.springerlink.com/content/978-3-540-74706-2`.
23. Eugene S. Edgington. *Randomization Tests*. Boca Raton, FL, USA: CRC Press, Inc. and New York, NY, USA: Marcel Dekker Ltd., 3rd revised and expanded edition, July 1995. ISBN 0824796691 and 9780824796693. URL `http://books.google.de/books?id=UxGqdcmL5gMC`.

24. Bernd Streitberg and Joachim Röhmel. Exakte verteilungen für rang- und randomisierungstests im allgemeinen c-stichprobenfall. 18(1):12–19.
25. Frank Wilcoxon. Individual comparisons by ranking methods. *Biometrics Bulletin*, 1(6):80–83, December 1945. URL http://sci2s.ugr.es/keel/pdf/algorithm/articulo/wilcoxon1945.pdf.
26. Olive Jean Dunn. Multiple comparisons among means. *Journal of the American Statistical Association*, 56(293):52–64, March 1961. doi: 10.1080/01621459.1961.10482090. URL http://www.jstor.org/stable/2282330.
27. Janez Demšar. Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research (JMLR)*, 7:1–30, January 2006. URL http://jmlr.csail.mit.edu/papers/volume7/demsar06a/demsar06a.pdf.
28. Salvador García and Francisco Herrera Triguero. An extension on "statistical comparisons of classifiers over multiple data sets" for all pairwise comparisons. *Journal of Machine Learning Research (JMLR)*, 9:2677–2694, December 2008. URL http://jmlr.csail.mit.edu/papers/volume9/garcia08a/garcia08a.pdf.
29. David Stifler Johnson and Lyle A. McGeoch. 8th dimacs implementation challenge: The traveling salesman problem, December 12, 2008. URL http://www2.research.att.com/~dsj/chtsp/.
30. David Stifler Johnson and Lyle A. McGeoch. Experimental analysis of heuristics for the stsp. In Gregory Z. Gutin and Abraham P. Punnen, editors, *The Traveling Salesman Problem and its Variations*, volume 12 of *Combinatorial Optimization*, chapter 9, pages 369–443. Norwell, MA, USA: Kluwer Academic Publishers, 2002. doi: 10.1007/0-306-48213-4_9. URL http://www2.research.att.com/~dsj/papers/stspchap.pdf.
31. David Stifler Johnson, Gregory Z. Gutin, Lyle A. McGeoch, Anders Yeo, Weixiong Zhang, and Alexei Zverovitch. Experimental analysis of heuristics for the atsp. In Gregory Z. Gutin and Abraham P. Punnen, editors, *The Traveling Salesman Problem and its Variations*, volume 12 of *Combinatorial Optimization*, chapter 10, pages 445–487. Norwell, MA, USA: Kluwer Academic Publishers, 2002. doi: 10.1007/0-306-48213-4_10. URL http://www2.research.att.com/~dsj/papers/atspchap.pdf.
32. Nikolaus Hansen, Anne Auger, Steffen Finck, and Raymond Ros. Real-parameter black-box optimization benchmarking: Experimental setup. Technical report, Orsay, France: Université Paris Sud, Institut National de Recherche en Informatique et en Automatique (INRIA) Futurs, Équipe TAO, March 24, 2012. URL http://coco.lri.fr/BBOB-downloads/download11.05/bbobdocexperiment.pdf.
33. William John Cook. Traveling salesman problem, September 18, 2011. URL http://www.tsp.gatech.edu/index.html.
34. Gerhard Reinelt. Tsplib, 1995. URL http://comopt.ifi.uni-heidelberg.de/software/TSPLIB95/.
35. Gerhard Reinelt. Tsplib 95. Technical report, Heidelberg, Germany: Universität Heidelberg, Institut für Mathematik, 1995. URL http://comopt.ifi.uni-heidelberg.de/software/TSPLIB95/DOC.PS.

Bibliography IV

36. Gerhard Reinelt. Tsplib – a traveling salesman problem library. *ORSA Journal on Computing*, 3(4):376–384, Fall 1991. doi: 10.1287/ijoc.3.4.376.
37. José Manuel Belenguer Ribera. *CARPLIB*, November 5, 2005. URL http://www.uv.es/~belengue/carp/READ_ME.
38. Armin Scholl and Robert Klein. Bin packing, September 2, 2003. URL http://www.wiwi.uni-jena.de/Entscheidung/binpp/.
39. Holger H. Hoos and Thomas Stützle. Satlib – the satisfiability library, March 22, 2005. URL http://www.satlib.org/.
40. Giselher Pankratz and Veikko Krypczyk. Benchmark data sets for dynamic vehicle routing problems, June 12, 2007. URL http://www.fernuni-hagen.de/WINF/inhfrm/benchmark_data.htm.
41. Daniele Vigo. Vrplib: A vehicle routing problem library, October 3, 2003. URL http://www.or.deis.unibo.it/research_pages/ORinstances/VRPLIB/VRPLIB.html.
42. Bernabé Dorronsoro Díaz. The vrp web, March 2007. URL http://neo.lcc.uma.es/radi-aeb/WebVRP/index.html.
43. John Edward Beasley. Or-library, June 1990. URL http://people.brunel.ac.uk/~mastjjb/jeb/info.html.
44. Lee Altenberg. Nk fitness landscapes. In Thomas Bäck, David B. Fogel, and Zbigniew Michalewicz, editors, *Handbook of Evolutionary Computation*, Computational Intelligence Library, chapter B2.7.2. New York, NY, USA: Oxford University Press, Inc., Dirac House, Temple Back, Bristol, UK: Institute of Physics Publishing Ltd. (IOP), and Boca Raton, FL, USA: CRC Press, Inc., January 1, 1997. URL http://www.cmi.univ-mrs.fr/~pardoux/LeeNKFL.pdf.
45. Edward D. Weinberger. Local properties of kauffman's nk model, a tuneably rugged energy landscape. *Physical Review A*, 44(10):6399–6413, November 1991. doi: 10.1103/PhysRevA.44.6399.
46. Benjamin Skellett, Benjamin Cairns, Nicholas Geard, Bradley Tonkes, and Janet Wiles. Rugged nk landscapes contain the highest peaks. In Hans-Georg Beyer, Una-May O'Reilly, Dirk V. Arnold, Wolfgang Banzhaf, Christian Blum, Eric W. Bonabeau, Erick Cantú-Paz, Dipankar Dasgupta, Kalyanmoy Deb, James A. Foster, Edwin D. de Jong, Hod Lipson, Xavier Llorà, Spiros Mancoridis, Martin Pelikan, Günther R. Raidl, Terence Soule, Jean-Paul Watson, and Eckart Zitzler, editors, *Proceedings of Genetic and Evolutionary Computation Conference (GECCO'05)*, pages 579–584, Washington, DC, USA: Loews L'Enfant Plaza Hotel, June 25–27, 2005. New York, NY, USA: ACM Press. URL http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.1.5011.

47. Sung-Soon Choi, Kyomin Jung, and Jeong Han Kim. Phase transition in a random nk landscape model. In Hans-Georg Beyer, Una-May O'Reilly, Dirk V. Arnold, Wolfgang Banzhaf, Christian Blum, Eric W. Bonabeau, Erick Cantú-Paz, Dipankar Dasgupta, Kalyanmoy Deb, James A. Foster, Edwin D. de Jong, Hod Lipson, Xavier Llorà, Spiros Mancoridis, Martin Pelikan, Günther R. Raidl, Terence Soule, Jean-Paul Watson, and Eckart Zitzler, editors, *Proceedings of Genetic and Evolutionary Computation Conference (GECCO'05)*, pages 1241–1248, Washington, DC, USA: Loews L'Enfant Plaza Hotel, June 25–27, 2005. New York, NY, USA: ACM Press. doi: 10.1145/1068009.1068212. URL http://web.mit.edu/kmjung/Public/NK%20gecco%20final.pdf. Session: Genetic Algorithms.
48. Yong Gao and Joseph Culberson. An analysis of phase transition in nk landscapes. *Journal of Artificial Intelligence Research (JAIR)*, 17:309–332, October 2002. URL http://www.jair.org/media/1081/live-1081-2104-jair.pdf.
49. Stuart Alan Kauffman and Edward D. Weinberger. The nk model of rugged fitness landscapes and its application to maturation of the immune response. *Journal of Theoretical Biology*, 141(2):211–245, November 21, 1989. doi: 10.1016/S0022-5193(89)80019-0.
50. Lionel Barnett. Ruggedness and neutrality – the nkp family of fitness landscapes. In Christoph Adami, Richard K. Belew, Hiroaki Kitano, and Charles E. Taylor, editors, *Proceedings of the Sixth International Conference on Artificial Life (Artificial Life VI)*, volume 6 of *Bradford Books*, pages 18–27, Los Angeles, CA, USA: University of California (UCLA), June 27–29, 1998. Cambridge, MA, USA: MIT Press. URL http://www.cogs.susx.ac.uk/users/lionelb/downloads/publications/alife6_paper.ps.gz.
51. Nicholas Geard. An exploration of nk landscapes with neutrality. Master's thesis, St Lucia, Q, Australia: University of Queensland, School of Information Technology and Electrical Engineering, October 2001. URL http://eprints.ecs.soton.ac.uk/14213/1/ng_thesis.pdf.
52. Nicholas Geard, Janet Wiles, Jennifer Hallinan, Bradley Tonkes, and Benjamin Skellett. A comparison of neutral landscapes – nk, nkp and nkq. In David B. Fogel, Mohamed A. El-Sharkawi, Xin Yao, Hitoshi Iba, Paul Marrow, and Mark Shackleton, editors, *Proceedings of the IEEE Congress on Evolutionary Computation (CEC'02), 2002 IEEE World Congress on Computation Intelligence (WCCI'02)*, volume 1-2, pages 205–210, Honolulu, HI, USA: Hilton Hawaiian Village Hotel (Beach Resort & Spa), May 12–17, 2002. Piscataway, NJ, USA: IEEE Computer Society, Los Alamitos, CA, USA: IEEE Computer Society Press. URL http://eprints.ecs.soton.ac.uk/14208/1/cec1-comp.pdf.

53. Melanie Mitchell, Stephanie Forrest, and John Henry Holland. The royal road for genetic algorithms: Fitness landscapes and ga performance. In Francisco J. Varela and Paul Bourgine, editors, *Toward a Practice of Autonomous Systems: Proceedings of the First European Conference on Artificial Life (Actes de la Première Conférence Européenne sur la Vie Artificielle) (ECAL'91)*, Bradford Books, pages 245–254, Paris, France, December 11–13, 1991. Cambridge, MA, USA: MIT Press. URL http://web.cecs.pdx.edu/~mm/ecal92.pdf.
54. Terry Jones. A description of holland's royal road function. *Evolutionary Computation*, 2(4):411–417, Winter 1994. doi: 10.1162/evco.1994.2.4.409.
55. Terry Jones. A description of holland's royal road function. Working Papers 94-11-059, Santa Fé, NM, USA: Santa Fe Institute, November 1994. URL http://www.santafe.edu/media/workingpapers/94-11-059.pdf.
56. R. J. Quick, Victor J. Rayward-Smith, and George D. Smith. The royal road functions: Description, intent and experimentation. In Terence Claus Fogarty, editor, *Proceedings of the Workshop on Artificial Intelligence and Simulation of Behaviour, International Workshop on Evolutionary Computing, Selected Papers (AISB'96)*, volume 1143/1996 of *Lecture Notes in Computer Science (LNCS)*, pages 223–235, Brighton, UK, April 1–2, 1996. Berlin, Germany: Springer-Verlag GmbH. doi: 10.1007/BFb0032786.
57. Tobias Storch and Ingo Wegener. Real royal road functions for constant population size. In Erick Cantú-Paz, James A. Foster, Kalyanmoy Deb, Lawrence Davis, Rajkumar Roy, Una-May O'Reilly, Hans-Georg Beyer, Russell K. Standish, Graham Kendall, Stewart W. Wilson, Mark Harman, Joachim Wegener, Dipankar Dasgupta, Mitchell A. Potter, Alan C. Schultz, Kathryn A. Dowsland, Natasa Jonoska, and Julian Francis Miller, editors, *Proceedings of the Genetic and Evolutionary Computation Conference, Part II (GECCO'03)*, volume 2724/2003 of *Lecture Notes in Computer Science (LNCS)*, pages 1406–1417, Chicago, IL, USA: Holiday Inn Chicago, July 12–16, 2003. Berlin, Germany: Springer-Verlag GmbH. doi: 10.1007/3-540-45110-2_14.
58. Tobias Storch and Ingo Wegener. Real royal road functions for constant population size. Reihe Computational Intelligence: Design and Management of Complex Technical Processes and Systems by Means of Computational Intelligence Methods CI-167/04, Dortmund, North Rhine-Westphalia, Germany: Universität Dortmund, Collaborative Research Center (Sonderforschungsbereich) 531, February 2004. URL http://hdl.handle.net/2003/5456.
59. Tobias Storch and Ingo Wegener. Real royal road functions for constant population size. *Theoretical Computer Science*, 320(1):123–134, June 2, 2004. doi: 10.1016/j.tcs.2004.03.047.
60. Erik van Nimwegen, James P. Crutchfield, and Melanie Mitchell. Statistical dynamics of the royal road genetic algorithm. *Theoretical Computer Science*, 229(1–2):41–102, November 6, 1999. doi: 10.1016/S0304-3975(99)00119-X. URL http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.43.3594.

61. Bart Naudts, Dominique Suys, and Alain Verschoren. Generalized royal road functions and their epistasis. *Computers and Artificial Intelligence*, 19(4), 2000.
62. Michaël Defoin Platel, Sébastien Vérel, Manuel Clergue, and Philippe Collard. From royal road to epistatic road for variable length evolution algorithm. In Pierre Liardet, Pierre Collet, Cyril Fonlupt, Evelyne Lutton, and Marc Schoenauer, editors, *Proceedings of the 6th International Conference on Artificial Evolution, Evolution Artificielle (EA'03)*, volume 2936 of *Lecture Notes in Computer Science (LNCS)*, pages 3–14, Marseilles, France, October 27–30, 2003. Berlin, Germany: Springer-Verlag GmbH. doi: 10.1007/b96080. URL http://arxiv.org/abs/0707.0548.
63. Thomas Weise, Stefan Niemczyk, Hendrik Skubch, Roland Reichle, and Kurt Geihs. A tunable model for multi-objective, epistatic, rugged, and neutral fitness landscapes. In Maarten Keijzer, Giuliano Antoniol, Clare Bates Congdon, Kalyanmoy Deb, Benjamin Doerr, Nikolaus Hansen, John H. Holmes, Gregory S. Hornby, Daniel Howard, James Kennedy, Sanjeev P. Kumar, Fernando G. Lobo, Julian Francis Miller, Jason H. Moore, Frank Neumann, Martin Pelikan, Jordan B. Pollack, Kumara Sastry, Kenneth Owen Stanley, Adrian Stoica, El-Ghazali Talbi, and Ingo Wegener, editors, *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO'08)*, pages 795–802, Atlanta, GA, USA: Renaissance Atlanta Hotel Downtown, July 12–16, 2008. New York, NY, USA: ACM Press. doi: 10.1145/1389095.1389252.
64. Jeffrey Horn, David Edward Goldberg, and Kalyanmoy Deb. Long path problems. In Yuval Davidor, Hans-Paul Schwefel, and Reinhard Männer, editors, *Proceedings of the Third Conference on Parallel Problem Solving from Nature; International Conference on Evolutionary Computation (PPSN III)*, volume 866/1994 of *Lecture Notes in Computer Science (LNCS)*, pages 149–158, Jerusalem, Israel, October 9–14, 1994. Berlin, Germany: Springer-Verlag GmbH. doi: 10.1007/3-540-58484-6_259. URL http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.51.9758.
65. Christian Höhn and Colin R. Reeves. Are long path problems hard for genetic algorithms? In Hans-Michael Voigt, Werner Ebeling, Ingo Rechenberg, and Hans-Paul Schwefel, editors, *Proceedings of the 4th International Conference on Parallel Problem Solving from Nature (PPSN IV)*, volume 1141/1996 of *Lecture Notes in Computer Science (LNCS)*, pages 134–143, Berlin, Germany, September 22–24, 1996. Berlin, Germany: Springer-Verlag GmbH. doi: 10.1007/3-540-61723-X_977.
66. Ciro Amitrano, Luca Peliti, and M. Saber. Population dynamics in a spin-glass model of chemical evolution. *Journal of Molecular Evolution*, 29(6):513–525, December 1989. doi: 10.1007/BF02602923.
67. William Michael Rudnick. *Genetic Algorithms and Fitness Variance with an Application to the Automated Design of Artificial Neural Networks*. PhD thesis, Beaverton, OR, USA: Oregon Graduate Institute of Science & Technology, 1992.
68. David H. Ackley. *A Connectionist Machine for Genetic Hillclimbing*. PhD thesis, Pittsburgh, PA, USA: Carnegy Mellon University (CMU), August 31, 1987. URL http://books.google.de/books?id=3ttfAAAAMAAJ.

69. Heinz Mühlenbein and Dirk Schlierkamp-Voosen. Predictive models for the breeder genetic algorithm i: Continuous parameter optimization. *Evolutionary Computation*, 1(1):25–49, Spring 1993. doi: 10.1162/evco.1993.1.1.25. URL http://www.muehlenbein.org/breeder93.pdf.
70. Dirk Thierens and David Edward Goldberg. Convergence models of genetic algorithm selection schemes. In Yuval Davidor, Hans-Paul Schwefel, and Reinhard Männer, editors, *Proceedings of the Third Conference on Parallel Problem Solving from Nature; International Conference on Evolutionary Computation (PPSN III)*, volume 866/1994 of *Lecture Notes in Computer Science (LNCS)*, pages 119–129, Jerusalem, Israel, October 9–14, 1994. Berlin, Germany: Springer-Verlag GmbH. doi: 10.1007/3-540-58484-6_256. URL http://www.cs.uu.nl/groups/DSS/publications/convergence/convMdl.ps.
71. Brad L. Miller and David Edward Goldberg. Genetic algorithms, selection schemes, and the varying effects of noise. *Evolutionary Computation*, 4(2):113–131, Summer 1996. doi: 10.1162/evco.1996.4.2.113. URL http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.31.3449.
72. Thomas Bäck. Generalized convergence models for tournament- and (μ, λ)-selection. In Larry J. Eshelman, editor, *Proceedings of the Sixth International Conference on Genetic Algorithms (ICGA'95)*, pages 2–8, Pittsburgh, PA, USA: University of Pittsburgh, July 15–19, 1995. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc. URL http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.56.5969.
73. Tobias Blickle and Lothar Thiele. A comparison of selection schemes used in evolutionary algorithms. *Evolutionary Computation*, 4(4):361–394, Winter 1996. doi: 10.1162/evco.1996.4.4.361. URL http://www.handshake.de/user/blickle/publications/ECfinal.ps.
74. Dominic Wilson and Devinder Kaur. Using quotient graphs to model neutrality in evolutionary search. In Maarten Keijzer, Giuliano Antoniol, Clare Bates Congdon, Kalyanmoy Deb, Benjamin Doerr, Nikolaus Hansen, John H. Holmes, Gregory S. Hornby, Daniel Howard, James Kennedy, Sanjeev P. Kumar, Fernando G. Lobo, Julian Francis Miller, Jason H. Moore, Frank Neumann, Martin Pelikan, Jordan B. Pollack, Kumara Sastry, Kenneth Owen Stanley, Adrian Stoica, El-Ghazali Talbi, and Ingo Wegener, editors, *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO'08)*, pages 2233–2238, Atlanta, GA, USA: Renaissance Atlanta Hotel Downtown, July 12–16, 2008. New York, NY, USA: ACM Press. doi: 10.1145/1388969.1389051.
75. Ponnuthurai Nagaratnam Suganthan, Nikolaus Hansen, J. J. Liang, Kalyanmoy Deb, Ying-Ping Chen, Anne Auger, and Santosh Tiwari. Problem definitions and evaluation criteria for the cec 2005 special session on real-parameter optimization. KanGAL Report 2005005, Kanpur, Uttar Pradesh, India: Kanpur Genetic Algorithms Laboratory (KanGAL), Department of Mechanical Engineering, Indian Institute of Technology Kanpur (IIT), May 2005. URL http://www.iitk.ac.in/kangal/papers/k2005005.pdf.

76. Ponnuthurai Nagaratnam Suganthan, Nikolaus Hansen, J. J. Liang, Kalyanmoy Deb, Ying-Ping Chen, Anne Auger, and Santosh Tiwari. Problem definitions and evaluation criteria for the cec 2005 special session on real-parameter optimization. Technical Report May-30-05, Singapore: Nanyang Technological University (NTU), May 30, 2005. URL http://www.ntu.edu.sg/home/epnsugan/index_files/CEC-05/Tech-Report-May-30-05.pdf.
77. Ke Tang, Xin Yao, Ponnuthurai Nagaratnam Suganthan, Cara MacNish, Ying-Ping Chen, Chih-Ming Chen, and Zhenyu Yang. Benchmark functions for the cec'2008 special session and competition on large scale global optimization. Technical report, Hefei, Anhui, China: University of Science and Technology of China (USTC), School of Computer Science and Technology, Nature Inspired Computation and Applications Laboratory (NICAL), 2007. URL http://sci2s.ugr.es/programacion/workshop/Tech.Report.CEC2008.LSGO.pdf.
78. Qingfu Zhang, Aimin Zhou, Shizheng Zhao, Ponnuthurai Nagaratnam Suganthan, Wudong Liu, and Santosh Tiwari. Multiobjective optimization test instances for the cec 2009 special session and competition. Technical report, Singapore: Nanyang Technological University (NTU), April 2009. URL http://web.mysites.ntu.edu.sg/epnsugan/PublicSite/Shared%20Documents/CEC2009-MOEA/PDF-Tech-Report.pdf.
79. V. L. Huang, Alex Kai Qin, Kalyanmoy Deb, Eckart Zitzler, Ponnuthurai Nagaratnam Suganthan, J. J. Liang, Mike Preuß, and Simon Huband. Problem definitions for performance assessment of multi-objective optimization algorithms, special session on constrained real-parameter optimization. Technical report, Singapore: Nanyang Technological University (NTU), January 2007. URL http://www3.ntu.edu.sg/home/EPNSugan/index_files/CEC-07/CEC-07-TR-13-Feb.pdf.
80. J. J. Liang, Thomas Philip Runarsson, Efrén Mezura-Montes, Maurice Clerc, Ponnuthurai Nagaratnam Suganthan, Carlos Artemio Coello Coello, and Kalyanmoy Deb. Problem definitions and evaluation criteria for the cec 2006 special session on constrained real-parameter optimization. Technical report, Singapore: Nanyang Technological University (NTU), September 18, 2006. URL http://www.lania.mx/~emezura/util/files/tr_cec06.pdf.
81. Kalyanmoy Deb, Lothar Thiele, Marco Laumanns, and Eckart Zitzler. Scalable multi-objective optimization test problems. In David B. Fogel, Mohamed A. El-Sharkawi, Xin Yao, Hitoshi Iba, Paul Marrow, and Mark Shackleton, editors, *Proceedings of the IEEE Congress on Evolutionary Computation (CEC'02), 2002 IEEE World Congress on Computation Intelligence (WCCI'02)*, volume 1, pages 825–830, Honolulu, HI, USA: Hilton Hawaiian Village Hotel (Beach Resort & Spa), May 12–17, 2002. Piscataway, NJ, USA: IEEE Computer Society, Los Alamitos, CA, USA: IEEE Computer Society Press. doi: 10.1109/CEC.2002.1007032. URL http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.18.7531.
82. Jürgen Branke. The moving peaks benchmark. Technical report, Karlsruhe, Germany: University of Karlsruhe (Friedriciana), Institute for Applied Computer Science and Formal Description Methods (AIFB), December 16, 1999. URL http://www.aifb.uni-karlsruhe.de/~jbr/MovPeaks/.

83. Uc irvine machine learning repository, 2011. URL http://archive.ics.uci.edu/ml/.
84. Edgar Anderson. The irises of the gaspé peninsula. *Bulletin of the American Iris Society*, 59:2–5, 1935.
85. James C. Bezdek, James M. Keller, Raghu Krishnapuram, Ludmila I. Kuncheva, and Nikhil R. Pal. Will the real iris data please stand up? *IEEE Transactions on Fuzzy Systems (TFS)*, 7(3):368–369, June 1999. doi: 10.1109/91.771092.
86. William H. Wolberg and Olvi Mangasarian. *Breast Cancer Wisconsin (Original) Data Set*. Irvine, CA, USA: UCI Machine Learning Repository, Center for Machine Learning and Intelligent Systems, Donald Bren School of Information and Computer Science, University of California, 1989. URL http://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+(Original).
87. David W. Aha, Andras Janosi, William Steinbrunn, Matthias Pfisterer, and Robert Detrano. *Heart Disease Data Set*. Irvine, CA, USA: UCI Machine Learning Repository, Center for Machine Learning and Intelligent Systems, Donald Bren School of Information and Computer Science, University of California, July 22, 1988. URL http://archive.ics.uci.edu/ml/datasets/Heart+Disease.
88. Robert J. Collins and David R. Jefferson. Representations for artificial organisms. In Jean-Arcady Meyer and Stewart W. Wilson, editors, *From Animals to Animats: Proceedings of the First International Conference on Simulation of Adaptive Behavior (SAB'90)*, pages 382–390, Paris, France, September 24–28, 1990. Cambridge, MA, USA: MIT Press.
89. David R. Jefferson, Robert J. Collins, Claus Cooper, Michael G. Dyer, Margot Flowers, Richard E. Korf, Charles Tayler, and Alan Wang. Evolution as a theme in artificial life: The genesys/tracker system. In Christopher Gale Langdon, Charles E. Taylor, Doyne J. Farmer, and Steen Rasmussen, editors, *Proceedings of the Workshop on Artificial Life (Artificial Life II)*, volume X of *Santa Fe Institue Studies in the Sciences of Complexity*, pages 549–578. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc. and Boulder, CO, USA: Westview Press, February 1990.
90. John R. Koza. *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. Bradford Books. Cambridge, MA, USA: MIT Press, December 1992. ISBN 0-262-11170-5 and 978-0-262-11170-6. URL http://books.google.de/books?id=Bhtxo60BV0EC. 1992 first edition, 1993 second edition.
91. Mingxu Wan, Thomas Weise, and Ke Tang. Novel loop structures and the evolution of mathematical algorithms. In Sara Silva, James A. Foster, Miguel Nicolau, Penousal Machado, and Mario Giacobini, editors, *Proceedings of the 14th European Conference on Genetic Programming (EuroGP'11)*, volume 6621/2011 of *Lecture Notes in Computer Science (LNCS)*, pages 49–60, Torino, Italy, April 27–29, 2011. Berlin, Germany: Springer-Verlag GmbH. doi: 10.1007/978-3-642-20407-4_5.

92. William F. Punch, Douglas Zongker, and Erik D. Goodman. The royal tree problem, a benchmark for single and multiple population genetic programming. In Peter John Angeline and Kenneth E. Kinnear, Jr, editors, *Advances in Genetic Programming II*, Bradford Books, pages 299–316. Cambridge, MA, USA: MIT Press, October 26, 1996. URL http://citeseer.ist.psu.edu/147908.html.

93. Emin Erkan Korkmaz and Göktürk Üçoluk. Design and usage of a new benchmark problem for genetic programming. In Adnan Yazici and Cevat Şener, editors, *Proceedings of the Eighteenth International Symposium on Computer and Information Sciences (ISCIS XVIII)*, volume 2869/2003 of *Lecture Notes in Computer Science (LNCS)*, pages 561–567, Antalya, Turkey, November 3–5, 2003. Berlin, Germany: Springer-Verlag GmbH. doi: 10.1007/978-3-540-39737-3_70. URL http://citeseer.ist.psu.edu/650483.html.

Standard Normal Distribution

x	0.00	0.01	0.02	0.03	0.04	0.05	0.06	0.07	0.08	0.09
0.0	0.5000	0.5040	0.5080	0.5120	0.5160	0.5199	0.5239	0.5279	0.5319	0.5359
0.1	0.5398	0.5438	0.5478	0.5517	0.5557	0.5596	0.5636	0.5675	0.5714	0.5753
0.2	0.5793	0.5832	0.5871	0.5910	0.5948	0.5987	0.6026	0.6064	0.6103	0.6141
0.3	0.6179	0.6217	0.6255	0.6293	0.6331	0.6368	0.6406	0.6443	0.6480	0.6517
0.4	0.6554	0.6591	0.6628	0.6664	0.6700	0.6736	0.6772	0.6808	0.6844	0.6879
0.5	0.6915	0.6950	0.6985	0.7019	0.7054	0.7088	0.7123	0.7157	0.7190	0.7224
0.6	0.7257	0.7291	0.7324	0.7357	0.7389	0.7422	0.7454	0.7486	0.7517	0.7549
0.7	0.7580	0.7611	0.7642	0.7673	0.7704	0.7734	0.7764	0.7794	0.7823	0.7852
0.8	0.7881	0.7910	0.7939	0.7967	0.7995	0.8023	0.8051	0.8078	0.8106	0.8133
0.9	0.8159	0.8186	0.8212	0.8238	0.8264	0.8289	0.8315	0.8340	0.8365	0.8389
1.0	0.8413	0.8438	0.8461	0.8485	0.8508	0.8531	0.8554	0.8577	0.8599	0.8621
1.1	0.8643	0.8665	0.8686	0.8708	0.8729	0.8749	0.8770	0.8790	0.8810	0.8830
1.2	0.8849	0.8869	0.8888	0.8907	0.8925	0.8944	0.8962	0.8980	0.8997	0.9015
1.3	0.9032	0.9049	0.9066	0.9082	0.9099	0.9115	0.9131	0.9147	0.9162	0.9177
1.4	0.9192	0.9207	0.9222	0.9236	0.9251	0.9265	0.9279	0.9292	0.9306	0.9319
1.5	0.9332	0.9345	0.9357	0.9370	0.9382	0.9394	0.9406	0.9418	0.9429	0.9441
1.6	0.9452	0.9463	0.9474	0.9484	0.9495	0.9505	0.9515	0.9525	0.9535	0.9545
1.7	0.9554	0.9564	0.9573	0.9582	0.9591	0.9599	0.9608	0.9616	0.9625	0.9633
1.8	0.9641	0.9649	0.9656	0.9664	0.9671	0.9678	0.9686	0.9693	0.9699	0.9706
1.9	0.9713	0.9719	0.9726	0.9732	0.9738	0.9744	0.9750	0.9756	0.9761	0.9767
2.0	0.9772	0.9778	0.9783	0.9788	0.9793	0.9798	0.9803	0.9808	0.9812	0.9817
2.1	0.9821	0.9826	0.9830	0.9834	0.9838	0.9842	0.9846	0.9850	0.9854	0.9857
2.2	0.9861	0.9864	0.9868	0.9871	0.9875	0.9878	0.9881	0.9884	0.9887	0.9890
2.3	0.9893	0.9896	0.9898	0.9901	0.9904	0.9906	0.9909	0.9911	0.9913	0.9916
2.4	0.9918	0.9920	0.9922	0.9925	0.9927	0.9929	0.9931	0.9932	0.9934	0.9936
2.5	0.9938	0.9940	0.9941	0.9943	0.9945	0.9946	0.9948	0.9949	0.9951	0.9952

Appendix Outline

Mann-Whitney U Test

Mann-Whitney U Test

- Mann-Whitney U Test [15–18]:
- Compares two datasets $A = (a_1, a_2, \dots)$ and $B = (b_1, b_2, \dots)$.
- There are $n_a = |A|$ elements in A and $n_b = |B|$ elements in B.
- In total, there are $n = n_a + n_b$ elements.

Example Data

a	b
2	2
3	5
3	5
3	5
4	6
5	6
	7
	7
$\mathrm{med}(a) = 3$	$\mathrm{med}(b) = 5.5$
$n_a = 6$	$n_b = 8$

Mann-Whitney U Test

1. Mixing and sorting.
2. Ranking
3. Compute rank sums R_a, R_b.
4. Compute sample statistics U_a, U_b
5. Set $U = \min\{U_a, U_b\}$
6. Compute critical U_α values.
7. $U < U_\alpha \Rightarrow$ diference between U_a and U_b significant

Mann-Whitney U Test

1. The elements a_i and b_i are mixed together and sorted.
2. Ranking
3. Compute rank sums R_a, R_b.
4. Compute sample statistics U_a, U_b
5. Set $U = \min\{U_a, U_b\}$
6. Compute critical U_α values.
7. $U < U_\alpha \Rightarrow$ diference between U_a and U_b significant

Row	a	b
1	2	
2		2
3	3	
4	3	
5	3	
6	4	
7		5
8	5	
9		5
10		5
11		6
10		6
13		7
14		7
	$\mathrm{med}(a) = 3$	$\mathrm{med}(b) = 5.5$
	$n_a = 6$	$n_b = 8$

1. Mixing and sorting.
2. Each element receives a rank corresponding to its position in the list.
3. Compute rank sums R_a, R_b.
4. Compute sample statistics U_a, U_b
5. Set $U = \min\{U_a, U_b\}$
6. Compute critical U_α values.
7. $U < U_\alpha \Rightarrow$ diference between U_a and U_b significant

Mann-Whitney U Test

1. Mixing and sorting.
2. Each element receives a rank corresponding to its position in the list. Elements which have the same value receive the same rank:

$$r_i = r_{i+1} = \cdots = r_{i+m} = \frac{i + (i+1) + \cdots + (i+m)}{m+1} = \frac{m}{2} + i$$

3. Compute rank sums R_a, R_b.
4. Compute sample statistics U_a, U_b
5. Set $U = \min\{U_a, U_b\}$
6. Compute critical U_α values.
7. $U < U_\alpha \Rightarrow$ diference between U_a and U_b significant

Mann-Whitney U Test

Row	a	b	Ranks r_a	Ranks r_b
1	2		1.5	
2		2		1.5
3	3		4.0	
4	3		4.0	
5	3		4.0	
6	4		6.0	
7		5		8.5
8	5		8.5	
9		5		8.5
10		5		8.5
11		6		11.5
12		6		11.5
13		7		13.5
14		7		13.5
	$\mathrm{med}(a) = 3$	$\mathrm{med}(b) = 5.5$		
$n_a = 6$	$n_b = 8$			

Mann-Whitney U Test

1. Mixing and sorting.
2. Ranking
3. The rank sums R_a and R_b are computed:

$$R_a = \sum_{\forall a_i \in A} r(a_i)$$
$$R_b = \sum_{\forall b_i \in B} r(b_i)$$

4. Compute sample statistics U_a, U_b
5. Set $U = \min\{U_a, U_b\}$
6. Compute critical U_α values.
7. $U < U_\alpha \Rightarrow$ diference between U_a and U_b significant

Mann-Whitney U Test

1. Mixing and sorting.
2. Ranking
3. The rank sums R_a and R_b are computed:

$$R_a = \sum_{\forall a_i \in A} r(a_i) = 28$$

$$R_b = \sum_{\forall b_i \in B} r(b_i) = 77$$

4. Compute sample statistics U_a, U_b
5. Set $U = \min\{U_a, U_b\}$
6. Compute critical U_α values.
7. $U < U_\alpha \Rightarrow$ diference between U_a and U_b significant

Mann-Whitney U Test

Row	a	b	Ranks r_a	Ranks r_b
1	2		1.5	
2		2		1.5
3	3		4.0	
4	3		4.0	
5	3		4.0	
6	4		6.0	
7		5		8.5
8	5		8.5	
9		5		8.5
10		5		8.5
11		6		11.5
12		6		11.5
13		7		13.5
14		7		13.5
	$\mathrm{med}(a) = 3$	$\mathrm{med}(b) = 5.5$	$R_a = 28$	$R_b = 77$
	$n_a = 6$	$n_b = 8$		

Mann-Whitney U Test

1. Mixing and sorting.
2. Ranking
3. The rank sums R_a and R_b are computed:

$$R_a = \sum_{\forall a_i \in A} r(a_i) = 28$$

$$R_b = \sum_{\forall b_i \in B} r(b_i) = 77$$

   For these sums, the following always holds:

$$R_a + R_b = \frac{n(n+1)}{2}$$

4. Compute sample statistics U_a, U_b
5. Set $U = \min\{U_a, U_b\}$
6. Compute critical U_α values.
7. $U < U_\alpha \Rightarrow$ diference between U_a and U_b significant

1. Mixing and sorting.
2. Ranking
3. The rank sums R_a and R_b are computed:

$$\begin{aligned} R_a &= \sum_{\forall a_i \in A} r(a_i) = 28 \\ R_b &= \sum_{\forall b_i \in B} r(b_i) = 77 \end{aligned}$$

For these sums, the following always holds:

$$R_a + R_b = \frac{n(n+1)}{2} \quad \Rightarrow \quad 28 + 77 = \frac{14 * 15}{2} = 105$$

4. Compute sample statistics U_a, U_b
5. Set $U = \min\{U_a, U_b\}$
6. Compute critical U_α values.
7. $U < U_\alpha \Rightarrow$ diference between U_a and U_b significant

Mann-Whitney U Test

1. Mixing and sorting.
2. Ranking
3. Compute rank sums R_a, R_b.
4. The sample statistics are then given as:

$$U_a = R_a - \frac{n_a(n_a+1)}{2}$$
$$U_b = R_b - \frac{n_b(n_b+1)}{2}$$

5. Set $U = \min\{U_a, U_b\}$
6. Compute critical U_α values.
7. $U < U_\alpha \Rightarrow$ diference between U_a and U_b significant

Mann-Whitney U Test

1. Mixing and sorting.
2. Ranking
3. Compute rank sums R_a, R_b.
4. The sample statistics are then given as:

$$\begin{aligned} U_a &= R_a - \frac{n_a(n_a+1)}{2} = 28 - 21 = 7 \\ U_b &= R_b - \frac{n_b(n_b+1)}{2} = 77 - 36 = 41 \end{aligned}$$

5. Set $U = \min\{U_a, U_b\}$
6. Compute critical U_α values.
7. $U < U_\alpha \Rightarrow$ diference between U_a and U_b significant

Mann-Whitney U Test

1. Mixing and sorting.
2. Ranking
3. Compute rank sums R_a, R_b.
4. The sample statistics are then given as:

$$U_a = R_a - \frac{n_a(n_a+1)}{2} = 28 - 21 = 7$$
$$U_b = R_b - \frac{n_b(n_b+1)}{2} = 77 - 36 = 41$$

   where the following always holds

$$U_a + U_b = n_a n_b$$

5. Set $U = \min\{U_a, U_b\}$
6. Compute critical U_α values.
7. $U < U_\alpha \Rightarrow$ diference between U_a and U_b significant

1. Mixing and sorting.
2. Ranking
3. Compute rank sums R_a, R_b.
4. The sample statistics are then given as:

$$U_a = R_a - \frac{n_a(n_a+1)}{2} = 28 - 21 = 7$$
$$U_b = R_b - \frac{n_b(n_b+1)}{2} = 77 - 36 = 41$$

where the following always holds

$$U_a + U_b = n_a n_b \;\Rightarrow\; 7 + 41 = 6 * 8 = 48$$

5. Set $U = \min\{U_a, U_b\}$
6. Compute critical U_α values.
7. $U < U_\alpha \Rightarrow$ diference between U_a and U_b significant

Mann-Whitney U Test

1. Mixing and sorting.
2. Ranking
3. Compute rank sums R_a, R_b.
4. Compute sample statistics U_a, U_b
5. The smaller of the two values is used as statistic U:

$$U = \min\{U_a, U_b\}$$

6. Compute critical U_α values.
7. $U < U_\alpha \Rightarrow$ diference between U_a and U_b significant

Mann-Whitney U Test

1. Mixing and sorting.
2. Ranking
3. Compute rank sums R_a, R_b.
4. Compute sample statistics U_a, U_b
5. The smaller of the two values is used as statistic U:

$$U = \min\{U_a, U_b\} = \min\{7, 41\} = 7$$

6. Compute critical U_α values.
7. $U < U_\alpha \Rightarrow$ diference between U_a and U_b significant

Mann-Whitney U Test

1. Mixing and sorting.
2. Ranking
3. Compute rank sums R_a, R_b.
4. Compute sample statistics U_a, U_b
5. Set $U = \min\{U_a, U_b\}$
6. For the significance level α the critical U_α values can be computed for the two-sided test as

$$U_\alpha = \frac{n_a n_b}{2} - z\left(1 - \frac{\alpha}{2}\right)\sqrt{\frac{n_a n_b\left(n_a + n_b + 1\right)}{12}}$$

7. $U < U_\alpha \Rightarrow$ diference between U_a and U_b significant

Mann-Whitney U Test

1. Mixing and sorting.
2. Ranking
3. Compute rank sums R_a, R_b.
4. Compute sample statistics U_a, U_b
5. Set $U = \min\{U_a, U_b\}$
6. For the significance level α the critical U_α values can be computed for the two-sided test as

$$U_\alpha = \frac{n_a n_b}{2} - z\left(1 - \frac{\alpha}{2}\right)\sqrt{\frac{n_a n_b\left(n_a + n_b + 1\right)}{12}} = 24 - z\left(1 - \frac{\alpha}{2}\right)\sqrt{60}$$

7. $U < U_\alpha \Rightarrow$ diference between U_a and U_b significant

Mann-Whitney U Test

1. Mixing and sorting.
2. Ranking
3. Compute rank sums R_a, R_b.
4. Compute sample statistics U_a, U_b
5. Set $U = \min\{U_a, U_b\}$
6. For the significance level α the critical U_α values can be computed for the two-sided test as

$$U_\alpha = \frac{n_a n_b}{2} - z\left(1 - \frac{\alpha}{2}\right)\sqrt{\frac{n_a n_b\left(n_a + n_b + 1\right)}{12}} = 24 - z\left(1 - \frac{\alpha}{2}\right)\sqrt{60}$$

   where z is the probit function, the inverse cumulative distribution function of the standard normal distribution.

7. $U < U_\alpha \Rightarrow$ diference between U_a and U_b significant

Mann-Whitney U Test

1. Mixing and sorting.
2. Ranking
3. Compute rank sums R_a, R_b.
4. Compute sample statistics U_a, U_b
5. Set $U = \min\{U_a, U_b\}$
6. For the significance level α the critical U_α values can be computed for the two-sided test as

$$U_\alpha = \frac{n_a n_b}{2} - z\left(1 - \frac{\alpha}{2}\right)\sqrt{\frac{n_a n_b\left(n_a + n_b + 1\right)}{12}} = 24 - z\left(1 - \frac{\alpha}{2}\right)\sqrt{60}$$

   where z is the probit function, the inverse cumulative distribution function of the standard normal distribution.
   The values of z can be looked up in the Standard Normal Distribution table in the appendix.
7. $U < U_\alpha \Rightarrow$ diference between U_a and U_b significant

Mann-Whitney U Test

1. Mixing and sorting.
2. Ranking
3. Compute rank sums R_a, R_b.
4. Compute sample statistics U_a, U_b
5. Set $U = \min\{U_a, U_b\}$
6. For the significance level α the critical U_α values can be computed for the two-sided test as

$$U_\alpha = \frac{n_a n_b}{2} - z\left(1-\frac{\alpha}{2}\right)\sqrt{\frac{n_a n_b\left(n_a+n_b+1\right)}{12}} = 24 - z\left(1-\frac{\alpha}{2}\right)\sqrt{60}$$

   where z is the probit function, the inverse cumulative distribution function of the standard normal distribution.
   The values of z can be looked up in the Standard Normal Distribution table in the appendix.
   - For $\alpha = 0.05$ we get $z\left(1-\frac{\alpha}{2}\right) = z(0.975) \approx 1.96$
   - For $\alpha = 0.01$, we find $z\left(1-\frac{\alpha}{2}\right) = z(0.995) \approx 2.575$.
   - Hence, $U_{0.05} \approx 24 - 1.96\sqrt{60} \approx 8.82$ and $U_{0.01} \approx 24 - 2.575\sqrt{60} \approx 4.05$.
7. $U < U_\alpha \Rightarrow$ diference between U_a and U_b significant

Mann-Whitney U Test

1. Mixing and sorting.
2. Ranking
3. Compute rank sums R_a, R_b.
4. Compute sample statistics U_a, U_b
5. Set $U = \min\{U_a, U_b\}$
6. Compute critical U_α values.
7. Compare U with U_α:

1. Mixing and sorting.
2. Ranking
3. Compute rank sums R_a, R_b.
4. Compute sample statistics U_a, U_b
5. Set $U = \min\{U_a, U_b\}$
6. Compute critical U_α values.
7. Compare U with U_α:
   - The difference between U_a and U_b is significant at an error level α only if U is smaller than U_α

Mann-Whitney U Test

1. Mixing and sorting.
2. Ranking
3. Compute rank sums R_a, R_b.
4. Compute sample statistics U_a, U_b
5. Set $U = \min\{U_a, U_b\}$
6. Compute critical U_α values.
7. Compare U with U_α:
    - The difference between U_a and U_b is significant at an error level α only if U is smaller than U_α
    - If $U < U_\alpha$ and $U_a < U_b$: A is from a distribution with a smaller median than B

1. Mixing and sorting.
2. Ranking
3. Compute rank sums R_a, R_b.
4. Compute sample statistics U_a, U_b
5. Set $U = \min\{U_a, U_b\}$
6. Compute critical U_α values.
7. Compare U with U_α:
    - The difference between U_a and U_b is significant at an error level α only if U is smaller than U_α
    - If $U < U_\alpha$ and $U_a < U_b$: A is from a distribution with a smaller median than B (this is wrong with a probability of no more than α)

Mann-Whitney U Test

1. Mixing and sorting.
2. Ranking
3. Compute rank sums R_a, R_b.
4. Compute sample statistics U_a, U_b
5. Set $U = \min\{U_a, U_b\}$
6. Compute critical U_α values.
7. Compare U with U_α:
   - The difference between U_a and U_b is significant at an error level α only if U is smaller than U_α
   - If $U < U_\alpha$ and $U_a < U_b$: A is from a distribution with a smaller median than B (this is wrong with a probability of no more than α)
   - If $U < U_\alpha$ and $U_a > U_b$: A is from a distribution with a larger median than B

Mann-Whitney U Test

1. Mixing and sorting.
2. Ranking
3. Compute rank sums R_a, R_b.
4. Compute sample statistics U_a, U_b
5. Set $U = \min\{U_a, U_b\}$
6. Compute critical U_α values.
7. Compare U with U_α:
   - The difference between U_a and U_b is significant at an error level α only if U is smaller than U_α
   - If $U < U_\alpha$ and $U_a < U_b$: A is from a distribution with a smaller median than B (this is wrong with a probability of no more than α)
   - If $U < U_\alpha$ and $U_a > U_b$: A is from a distribution with a larger median than B (this is wrong with a probability of no more than α)

Mann-Whitney U Test

1. Mixing and sorting.
2. Ranking
3. Compute rank sums R_a, R_b.
4. Compute sample statistics U_a, U_b
5. Set $U = \min\{U_a, U_b\}$
6. Compute critical U_α values.
7. Compare U with U_α:
   - The difference between U_a and U_b is significant at an error level α only if U is smaller than U_α
   - If $U < U_\alpha$ and $U_a < U_b$: A is from a distribution with a smaller median than B (this is wrong with a probability of no more than α)
   - If $U < U_\alpha$ and $U_a > U_b$: A is from a distribution with a larger median than B (this is wrong with a probability of no more than α)
   - If $U \geq U_\alpha$: If we make a statement about the relationship of A and B, the chance to be wrong is greater than α.

Mann-Whitney U Test

1. Mixing and sorting.
2. Ranking
3. Compute rank sums R_a, R_b.
4. Compute sample statistics U_a, U_b
5. Set $U = \min\{U_a, U_b\}$
6. Compute critical U_α values.
7. Compare U with U_α:
   - The difference between U_a and U_b is significant at an error level α only if U is smaller than U_α
   - If $U < U_\alpha$ and $U_a < U_b$: A is from a distribution with a smaller median than B (this is wrong with a probability of no more than α)
   - If $U < U_\alpha$ and $U_a > U_b$: A is from a distribution with a larger median than B (this is wrong with a probability of no more than α)
   - If $U \geq U_\alpha$: If we make a statement about the relationship of A and B, the chance to be wrong is greater than α. There is no significant difference between A and B at level α.

1. Mixing and sorting.
2. Ranking
3. Compute rank sums R_a, R_b.
4. Compute sample statistics U_a, U_b
5. Set $U = \min\{U_a, U_b\}$
6. Compute critical U_α values.
7.
    - $U_a = 7$ and $U_b = 41$, i.e., $U_a < U_b$

Mann-Whitney U Test

1. Mixing and sorting.
2. Ranking
3. Compute rank sums R_a, R_b.
4. Compute sample statistics U_a, U_b
5. Set $U = \min\{U_a, U_b\}$
6. Compute critical U_α values.
7. - $U_a = 7$ and $U_b = 41$, i.e., $U_a < U_b$
   - $U < U_{0.05}$ holds since $7 < 8.82$

1. Mixing and sorting.
2. Ranking
3. Compute rank sums R_a, R_b.
4. Compute sample statistics U_a, U_b
5. Set $U = \min\{U_a, U_b\}$
6. Compute critical U_α values.
7.
   - $U_a = 7$ and $U_b = 41$, i.e., $U_a < U_b$
   - $U < U_{0.05}$ holds since $7 < 8.82$ $\Rightarrow$ We can state that the samples in A tend to be significantly smaller than those in B (with a probability to err of less than 5%).

1. Mixing and sorting.
2. Ranking
3. Compute rank sums R_a, R_b.
4. Compute sample statistics U_a, U_b
5. Set $U = \min\{U_a, U_b\}$
6. Compute critical U_α values.
7.
    - $U_a = 7$ and $U_b = 41$, i.e., $U_a < U_b$
    - $U < U_{0.05}$ holds since $7 < 8.82 \Rightarrow$ We can state that the samples in A tend to be significantly smaller than those in B (with a probability to err of less than 5%).
    - $\neg(U < U_{0.01})$ since $7 > 4.05$

Mann-Whitney U Test

1. Mixing and sorting.
2. Ranking
3. Compute rank sums R_a, R_b.
4. Compute sample statistics U_a, U_b
5. Set $U = \min\{U_a, U_b\}$
6. Compute critical U_α values.
7.
    - $U_a = 7$ and $U_b = 41$, i.e., $U_a < U_b$
    - $U < U_{0.05}$ holds since $7 < 8.82 \Rightarrow$ We can state that the samples in A tend to be significantly smaller than those in B (with a probability to err of less than 5%).
    - $\neg(U < U_{0.01})$ since $7 > 4.05 \Rightarrow$ If we would say that A is different from B, the probability to be wrong is more than 1%, i.e., at $\alpha = 0.01$, the difference between A and B is insignificant

Mann-Whitney U Test

1. Mixing and sorting.
2. Ranking
3. Compute rank sums R_a, R_b.
4. Compute sample statistics U_a, U_b
5. Set $U = \min\{U_a, U_b\}$
6. Compute critical U_α values.
7. $U < U_\alpha \Rightarrow$ diference between U_a and U_b significant

谢谢
Thank you

Thomas Weise [汤卫思]
tweise@hfuu.edu.cn
http://iao.hfuu.edu.cn

Hefei University, South Campus 2
Institute of Applied Optimization
Shushan District, Hefei, Anhui,
China

Caspar David Friedrich, “Der Wanderer über dem Nebelmeer”, 1818
http://en.wikipedia.org/wiki/Wanderer_above_the_Sea_of_Fog