DM841

Discrete Optimization - Heuristics

Experimental Analysis (cnt'd)

Marco Chiarandini

Department of Mathematics & Computer Science
University of Southern Denmark

Outline

1. Inferential Statistics
    - Statistical Tests
    - Experimental Designs
    - Applications to Our Scenarios

2. Race: Sequential Testing

3. Algorithm Selection

Outline

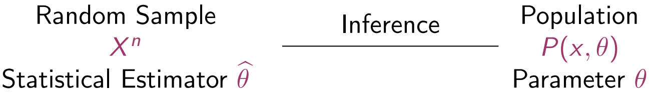
1. Inferential Statistics
   - Statistical Tests
   - Experimental Designs
   - Applications to Our Scenarios

2. Race: Sequential Testing

3. Algorithm Selection

Inferential Statistics

- We work with samples (instances, solution quality)
- But we want sound conclusions: generalization over a given population (all runs, all possible instances)
- Thus we need statistical inference



Random Sample

X^n

Statistical Estimator $\widehat{\theta}$

Inference

Population

$P(x, \theta)$

Parameter θ

Since the analysis is based on finite-sized sampled data, statements like

> *"the cost of solutions returned by algorithm $\mathcal{A}$ is smaller than that of algorithm $\mathcal{B}$"*

must be completed by

> *"at a level of significance of 5%".*

A Motivating Example

- There is a competition and two stochastic algorithms $\mathcal{A}_1$ and $\mathcal{A}_2$ are submitted.
- We run both algorithms once on n instances.
  On each instance either $\mathcal{A}_1$ wins $(+)$ or $\mathcal{A}_2$ wins $(-)$ or they make a tie $(=)$.

Questions:

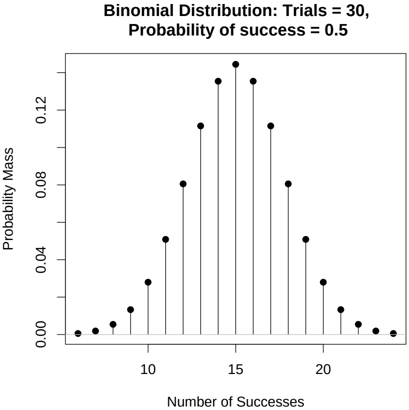
1. If we have only 10 instances and algorithm $\mathcal{A}_1$ wins 7 times how confident are we in claiming that algorithm $\mathcal{A}_1$ is the best?
2. How many instances and how many wins should we observe to gain a confidence of 95% that the algorithm $\mathcal{A}_1$ is the best?

A Motivating Example

- p: probability that $\mathcal{A}_1$ wins on each instance (+)
- n: number of runs without ties
- Y: number of wins of algorithm $\mathcal{A}_1$

If each run is indepenedent and consitent:

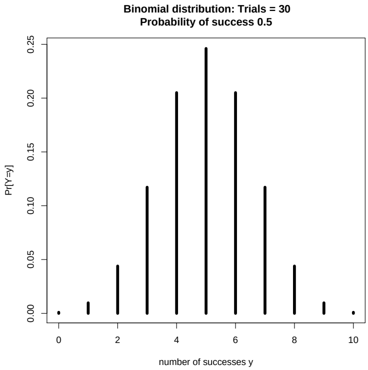
$$Y \sim B(n, p) : \qquad \Pr[Y = y] = \binom{n}{y} p^y (1-p)^{n-y}$$



1. If we have only 10 instances and algorithm $\mathcal{A}_1$ wins 7 times how confident are we in claiming that algorithm $\mathcal{A}_1$ is the best?

Under these conditions, we can check how unlikely the situation is if it was $p(+) \leq p(-)$.

If $p(+) = 0.5$ (ie, $p(+) = p(-)$) then the chance that algorithm $\mathcal{A}_1$ wins 7 or more times out of 10 is 17.2%: quite high!



2 How many instances and how many wins should we observe to gain a confidence of 95% that the algorithm $\mathcal{A}_1$ is the best?

To answer this question, we compute the 95%-quantile, *i.e.*, $y : \Pr[Y \geq y] < 0.05$ with $p = 0.5$ at different values of n:

n	10	11	12	13	14	15	16	17	18	19	20
y	9	9	10	10	11	12	12	13	13	14	15

This is an application example of sign test, a special case of binomial test in which $p = 0.5$

Statistical tests

General procedure:

- Assume that data are consistent with a null hypothesis H_0 (e.g., sample data are drawn from distributions with the same mean value).
- Use a statistical test to compute how likely this is to be true, given the data collected. This "likely" is quantified as the p-value.
- Do not reject H_0 if the p-value is larger than an user defined threshold called level of significance α.
- Alternatively, (p-value $< \alpha$), H_0 is rejected in favor of an alternative hypothesis, H_1, at a level of significance of α.

Inferential Statistics

Two kinds of errors may be committed when testing hypothesis:

$$\alpha = P(\text{type I error}) = P(\text{reject } H_0 \mid H_0 \text{ is true})$$
$$\beta = P(\text{type II error}) = P(\text{fail to reject } H_0 \mid H_0 \text{ is false})$$

General rule:

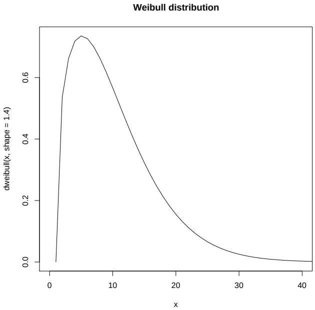
1. specify the type I error or level of significance α
2. seek the test with a suitable large statistical power, i.e., $1 - \beta = P(\text{reject } H_0 \mid H_0 \text{ is false})$

Theorem: Central Limit Theorem

If X^n is a random sample from an **arbitrary** distribution with mean μ and variance σ then the average $\bar{X}^n$ is asymptotically normally distributed, *i.e.*,

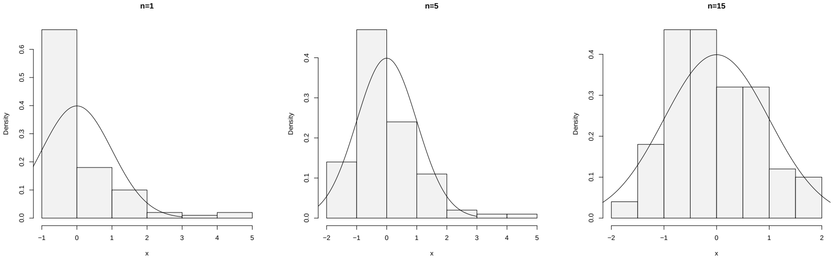
$$\bar{X}^n \approx N\left(\mu, \frac{\sigma^2}{n}\right) \qquad \text{or} \qquad z = \frac{\bar{X}^n - \mu}{\sigma/\sqrt{n}} \approx N(0,1)$$

- Consequences:
    - allows inference from a sample
    - allows to model errors in measurements: $X = \mu + \epsilon$
- Issues:
    - n should be *enough* large
    - μ and σ must be known



$$z = \frac{\bar{X} - \mu}{\sigma / \sqrt{n}}$$

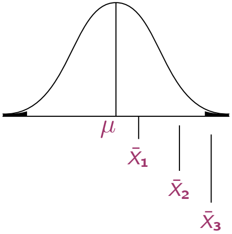
Samples of size 1, 5, 15, 50 repeated 100 times



Hypothesis Testing and Confidence Intervals

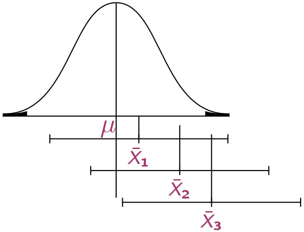
A test of hypothesis determines how likely a sampled estimate $\hat{\theta}$ is to occur under some assumptions on the parameter θ of the population.

$$Pr\left\{\mu - z_1\frac{\sigma}{\sqrt{n}} \leq \bar{X} \leq \mu + z_2\frac{\sigma}{\sqrt{n}}\right\} = 1-\alpha$$



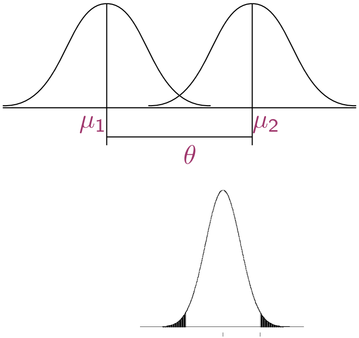
A confidence interval contains all those values that a parameter θ is likely to assume with probability $1-\alpha$: $Pr(\hat{\theta}_1 < \theta < \hat{\theta}_2) = 1-\alpha$

$$Pr\left\{\bar{X} - z_1\frac{\sigma}{\sqrt{n}} \leq \mu \leq \bar{X} + z_2\frac{\sigma}{\sqrt{n}}\right\} = 1-\alpha$$



Statistical Tests

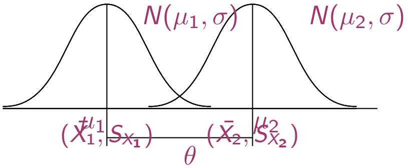
The Procedure of Test of Hypothesis

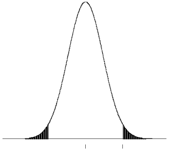


1. Specify the parameter θ and the test hypothesis,
$$\theta = \mu_1 - \mu_2 \qquad \begin{cases} H_0 : \theta = 0 \\ H_1 : \theta \neq 0 \end{cases}$$
2. Obtain $P(\theta \mid \theta = 0)$, the null distribution of θ
3. Compare $\hat{\theta}$ with the $\alpha/2$-quantiles (for two-sided tests) of $P(\theta \mid \theta = 0)$ and reject or not H_0 according to whether $\hat{\theta}$ is larger or smaller than this value.

Statistical Tests

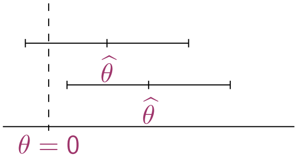
The Confidence Intervals Procedure





$$T = \frac{(\bar{x}_1 - \bar{x}_2) - (\mu_1 - \mu_2)}{\sqrt{\frac{S_{X_1} - S_{X_2}}{r}}}$$

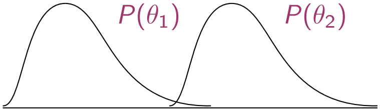
T˜ Student's t Distribution

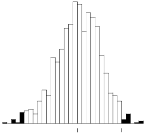


1. Specify the parameter θ and the test hypothesis,
$$\theta = \mu_1 - \mu_2 \qquad \begin{cases} H_0 : \theta = 0 \\ H_1 : \theta \neq 0 \end{cases}$$
2. Obtain $P(\theta, \theta = 0)$, the null distribution of θ in correspondence of the observed estimate $\hat{\theta}$ of the sample X
3. Determine $(\hat{\theta}^-, \hat{\theta}^+)$ such that $Pr\{\hat{\theta}^- \leq \theta \leq \hat{\theta}^+\} = 1 - \alpha$.
4. Do not reject H_0 if $\theta = 0$ falls inside the interval $(\hat{\theta}^-, \hat{\theta}^+)$. Otherwise reject H_0.

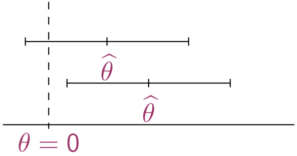
Statistical Tests

The Confidence Intervals Procedure





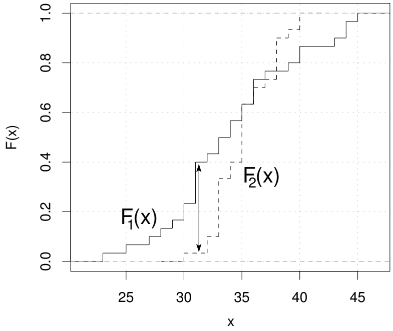
$\theta^* = \bar{X}_1^* - \bar{X}_2^*$



1. Specify the parameter θ and the test hypothesis,
$$\theta = \mu_1 - \mu_2 \qquad \begin{cases} H_0 : \theta = 0 \\ H_1 : \theta \neq 0 \end{cases}$$
2. Obtain $P(\theta, \theta = 0)$, the null distribution of θ in correspondence of the observed estimate $\hat{\theta}$ of the sample X
3. Determine $(\hat{\theta}^-, \hat{\theta}^+)$ such that $Pr\{\hat{\theta}^- \leq \theta \leq \hat{\theta}^+\} = 1 - \alpha$.
4. Do not reject H_0 if $\theta = 0$ falls inside the interval $(\hat{\theta}^-, \hat{\theta}^+)$. Otherwise reject H_0.

Kolmogorov-Smirnov Tests

The test compares empirical cumulative distribution functions.



It uses maximal difference between the two curves, $sup_x|F_1(x) - F_2(x)|$, and assesses how likely this value is under the null hypothesis that the two curves come from the same data

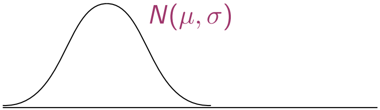
The test can be used as a two-samples or single-sample test (in this case to test against theoretical distributions: goodness of fit)

The test can be done in R with:

Parametric *vs* Nonparametric

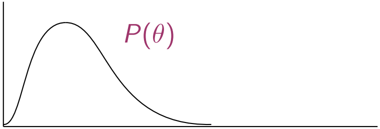
Parametric assumptions:

- independence
- homoschedasticity
- normality



Nonparametric assumptions:

- independence
- homoschedasticity



- Rank based tests
- Permutation tests
  - Exact
  - Conditional Monte Carlo

Preparation of the Experiments

Variance reduction techniques

- Blocking on instances
- Same pseudo random seed

Sample Sizes

- If the sample size is large enough (infinity) any difference in the means of the factors, no matter how small, will be significant
- Real *vs* Statistical significance
  Study factors until the improvement in the response variable is deemed small
- Desired statistical power $+$ practical precision $\Rightarrow$ sample size

Note: If resources available for N runs then the optimal design is one run on N instances [Birattari, 2004]

The Design of Experiments for Algorithms

- Statement of the objectives of the experiment
  - Comparison of different algorithms
  - Impact of algorithm components
  - How instance features affect the algorithms
- Identification of the sources of variance
  - Treatment factors (qualitative and quantitative)
  - Controllable nuisance factors ⇐ blocking
  - Uncontrollable nuisance factors ⇐ measuring
- Definition of factor combinations to test

  Easiest design: Unreplicated or Replicated Full Factorial Design
- Running a pilot experiment and refine the design
  - Bugs and no external biases
  - Ceiling or floor effects
  - Rescaling levels of quantitative factors
  - Detect the number of experiments needed to obtained the desired power.

Experimental Design

Algorithms ⇒ Treatment Factor;   Instances ⇒ Blocking/Random Factor

Design A: One run on various instances (Unreplicated Factorial)

	Algorithm 1	Algorithm 2	...	Algorithm k
Instance 1	X_{11}	X_{12}		X_{1k}
$\vdots$	$\vdots$	$\vdots$		$\vdots$
Instance b	X_{b1}	X_{b2}		X_{bk}

Design B: Several runs on various instances (Replicated Factorial)

	Algorithm 1	Algorithm 2	...	Algorithm k
Instance 1	$X_{111}, \ldots, X_{11r}$	$X_{121}, \ldots, X_{12r}$		$X_{1k1}, \ldots, X_{1kr}$
Instance 2	$X_{211}, \ldots, X_{21r}$	$X_{221}, \ldots, X_{22r}$		$X_{2k1}, \ldots, X_{2kr}$
$\vdots$	$\vdots$	$\vdots$		$\vdots$
Instance b	$X_{b11}, \ldots, X_{b1r}$	$X_{b21}, \ldots, X_{b2r}$		$X_{bk1}, \ldots, X_{bkr}$

Multiple Comparisons

$H_0: \ \mu_1 = \mu_2 = \mu_3 = \ldots$ $\qquad$ H_1 : {at least one differs}

Applying a statistical test to all pairs the error of Type I is not α but higher:

$$\alpha_{EX} = 1 - (1 - \alpha)^c$$

Eg, for $\alpha = 0.05$ and $c = 3 \Rightarrow \alpha_{EX} = 0.14$!

Adjustment methods

- Protected versions: global test + no adjustments
- Bonferroni $\alpha = \alpha_{EX}/c$ (conservative)
- Tukey Honest Significance Method (for parametric analysis)
- Holm (step-wise)
- Other step-wise procedures

Post-hoc analysis: Once the effect of factors has been recognized a finer grained analysis is performed to distinguish where important differences are.

Statistical Tests

Univariate Analysis

Several runs on a single instance

Global tests	Replicated
Parametric	F-test
Non-Parametric Rank based	Kruskall-Wallis Test
Non-Parametric Permutation based	Pooled Permutations
Non-Parametric KS type	Birnbaum-Hall test

Statistical Tests

Univariate Analysis

Several runs on a single instance

Pairwise tests	Replicated
Parametric	t-test Tukey HSD
Non-Parametric Rank based	Kruskall-Wallis Test or Mann-Whitney test ≡ *Wilcoxon Rank Sum Test* or Binomial test
Non-Parametric Permutation based	Pooled Permutations
Non-Parametric KS type	Birnbaum-Hall test

- Matched pairs versions: when, when not
- t-test with different variances

Statistical Tests

Univariate Analysis

On various instances (Designs A and B)

Global tests	Unreplicated (Design A)	Replicated (Design B)
Parametric	F-test	F-test
Non-Parametric Rank based	Friedman Test	Friedman Test
Non-Parametric Permutation based	Simple Permutations	Synchronized Permutations

Statistical Tests

Univariate Analysis

On various instances (Designs A and B)

Pairwise tests	Unreplicated	Replicated
Parametric	t-test Tukey HSD	t-test Tukey HSD
Non-Parametric Rank based	Friedman Test or *Wilcoxon Signed Rank Test*	Friedman Test
Non-Parametric Permutation based	Simple Permutations	Synchronized Permutations

- Matched pairs versions: when, when not
- t-test Welch variant: no assumption of equal variances

An Example

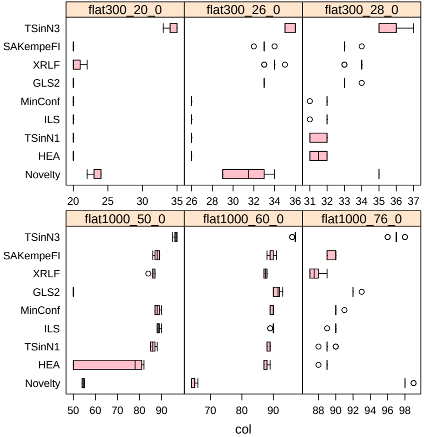
SLS algorithms for Graph Coloring:
Results collected on a set of benchmark instances

Instance	HEA		TS_{N1}		ILS		MinConf		XRLF	
Instance	Succ.	k	Succ.	k	Succ.	k	Succ.	k	Succ.	k
`flat300_20_0`	10	20	10	20	10	20	10	20	6	20
`flat300_26_0`	10	26	10	26	10	26	10	26	1	33
`flat300_28_0`	6	31	4	31	2	31	1	31	1	34
`flat1000_50_0`	4	50	2	85	6	88	4	87	1	84
`flat1000_60_0`	4	87	3	88	1	89	4	89	6	87
`flat1000_76_0`	1	88	1	88	1	89	8	90	6	87

	GLS		SA_{N2}		Novelty		TS_{N3}	
Instance	Succ.	k	Succ.	k	Succ.	k	Succ.	k
`flat300_20_0`	10	20	10	20	1	22	1	33
`flat300_26_0`	10	33	1	32	4	29	6	35
`flat300_28_0`	8	33	8	33	10	35	4	35
`flat1000_50_0`	10	50	1	86	6	54	1	95
`flat1000_60_0`	4	90	1	88	4	64	1	96
`flat1000_76_0`	8	92	4	89	8	98	1	96

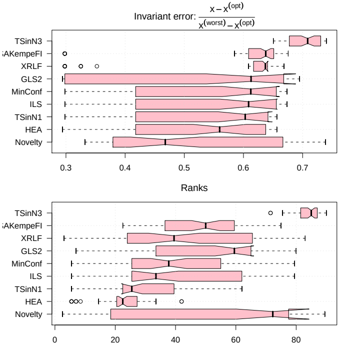
An Example

Raw data on the instances:



```
> load("gcp-all-classes.dataR")
> G <- F[F$class=="Flat",]
> bwplot(alg ~ col | inst,data=G,scales=list(x=list(relation="free")),pch="|")
> boxplot(err3~alg,data=G,horizontal=TRUE,main=expression(paste("Invariant error: ",fr
> boxplot(rank~alg,data=G,horizontal=TRUE,main="Ranks",notch=TRUE,col="pink")
```

An Example

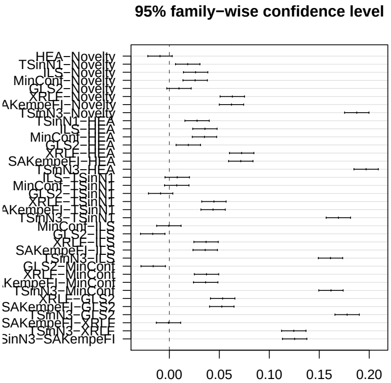


Note: notches are not appropriate for comparative inference

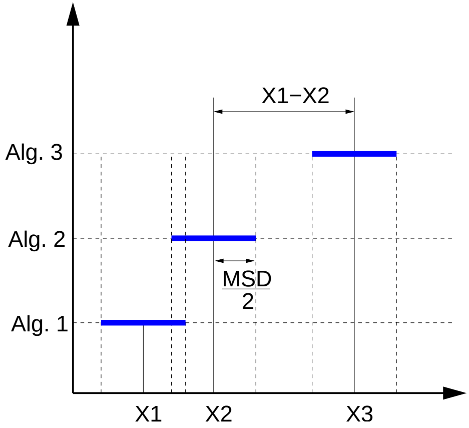
```
> pairwise.wilcox.test(G$err3,G$alg,paired=TRUE)

        Pairwise comparisons using Wilcoxon rank sum test

data:  G$err3 and G$alg
```

```
          Novelty HEA     TSinN1  ILS     MinConf GLS2    XRLF    SAKempeFI
HEA       1.00000 -       -       -       -       -       -       -
TSinN1    1.00000 0.00413 -       -       -       -       -       -
ILS       1.00000 1.3e-05 0.00072 -       -       -       -       -
MinConf   1.00000 9.4e-06 0.00042 1.00000 -       -       -       -
GLS2      1.00000 0.11462 0.94136 1.00000 1.00000 -       -       -
XRLF      0.25509 1.7e-05 0.02624 0.72455 0.47729 1.00000 -       -
SAKempeFI 0.72455 1.4e-07 3.0e-06 0.02708 0.02113 1.00000 1.00000 -
TSinN3    3.7e-08 5.8e-10 5.8e-10 5.8e-10 5.8e-10 5.8e-10 5.8e-10 5.8e-10

P value adjustment method: holm
```

```
> par(las=1,mar=c(3,8,3,1))
> plot(TukeyHSD(aov(err3~alg*inst,data=G),which="alg"),las=1,mar=c(3,7,3,1))
```



An Example

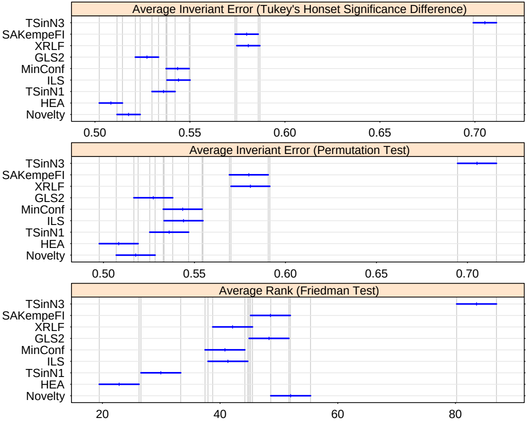


Minimal Significant Difference (MSD)

interval that satisfies simultaneously each comparison

Differences are statistically significant if the confidence intervals do not overlap

An Example


Average Inveriant Error (Tukey's Honset Significance Difference)
TSinN3
SAKempeFI
XRLF
GLS2
MinConf
ILS
TSinN1
HEA
Novelty
0.50
0.55
0.60
0.65
0.70
Average Inveriant Error (Permutation Test)
TSinN3
SAKempeFI
XRLF
GLS2
MinConf
ILS
TSinN1
HEA
Novelty
0.50
0.55
0.60
0.65
0.70
Average Rank (Friedman Test)
TSinN3
SAKempeFI
XRLF
GLS2
MinConf
ILS
TSinN1
HEA
Novelty
20
40
60
80

Outline

1. Inferential Statistics
   - Statistical Tests
   - Experimental Designs
   - Applications to Our Scenarios

2. Race: Sequential Testing

3. Algorithm Selection

Unreplicated Designs

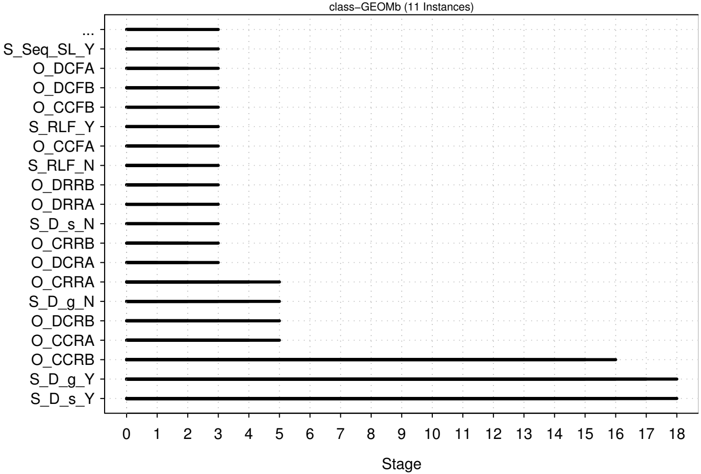
```
Procedure Race [Birattari 2002]:
repeat
    Randomly select an unseen instance and run all candidates on it
    Perform all-pairwise comparison statistical tests
    Drop all candidates that are significantly inferior to the best algorithm
until only one candidate left or no more unseen instances;
```

- F-Race use Friedman test
- Holm adjustment method is typically the most powerful

```
race(wrapper.file, maxExp=0,
             stat.test=c("friedman","t.bonferroni","t.holm","t.none"),
                conf.level=0.95, first.test=5, interactive=TRUE,
                   log.file="", no.slaves=0,...)
```

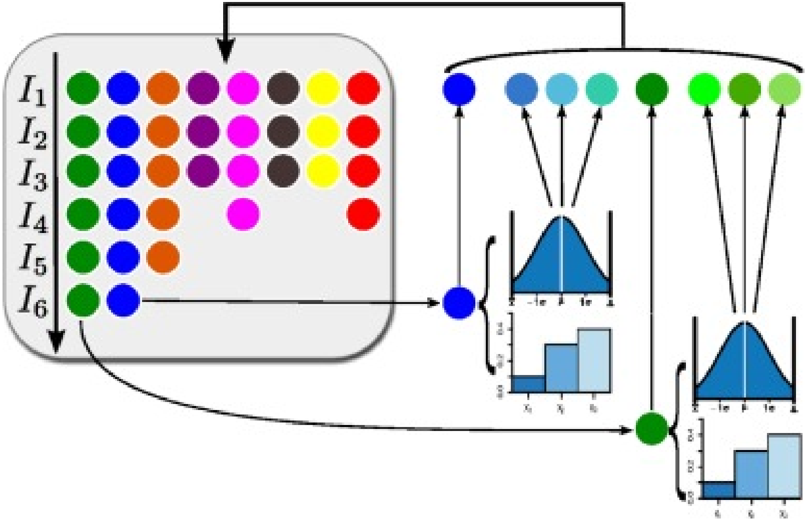
Instances

	θ_1	θ_2	θ_3	θ_4	θ_5	θ_6	θ_7	θ_8	θ_9	θ_{10}	
I_1	●	●	●	●	●	●	●	●	●	●	×
I_2	●	●	●	●	●	●	●	●	●	●	×
I_3	●	●	●	●	●	●	●	●	●	●	×
I_4	●	●	●	●	●	●	●	●	●	●	×
I_5	●	●	●	●	●	●	●	●	●	●	−
I_6	●	●	●	●	●	●	●				−
I_7	●	●	●	●	●						=
I_8	●	●	●	●	●						=
I_9	●	●	●	●	●						−
I_{10}	●	●	●								=
I_{11}	●	●	●								=
I_{12}	●	●	●								=
I_{13}	●	●	●								−
I_{14}	●	●									


class−GEOMb (11 Instances)
...
S_Seq_SL_Y
O_DCFA
O_DCFB
O_CCFB
S_RLF_Y
O_CCFA
S_RLF_N
O_DRRB
O_DRRA
S_D_s_N
O_CRRB
O_DCRA
O_CRRA
S_D_g_N
O_DCRB
O_CCRA
O_CCRB
S_D_g_Y
S_D_s_Y
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
Stage

iRace: iterated racing procedures

1. sample new configurations according to a particular distribution,
2. select the best configurations from the newly sampled ones by means of racing, and
3. update the sampling distribution in order to bias the sampling towards the best configurations


I_1
I_2
I_3
I_4
I_5
I_6
-1σ
μ
1σ
0.4
0.2
0.0

- Each configurable parameter has associated a sampling distribution that is independent of the sampling distributions of the other parameters, apart from constraints and conditions among parameters.
  - numerical parameters: truncated normal distribution
  - categorical parameters: discrete distribution.
- The update of the distributions consists in modifying the mean and the standard deviation in the case of the normal distribution, or the discrete probability values of the discrete distributions.
- The update changes the distributions so as to increase the probability of sampling the parameter values of the best configurations found so far.

Outline

1. Inferential Statistics
    - Statistical Tests
    - Experimental Designs
    - Applications to Our Scenarios

2. Race: Sequential Testing

3. Algorithm Selection

Algorithm Selection

Observation: algorithms' performance depends on the problem instance.

Idea: a set of complementary algorithms can be constructed, then identifying when to use which algorithm, we can improve overall performance

Algorithm Selection Problem (aka per-instance algorithm selection or offline algorithm selection) is a meta-algorithmic technique to choose an algorithm from a portfolio on an instance-by-instance basis.

Problem formulation:
Given a portfolio $\mathcal{P}$ of algorithms $\mathcal{A} \in \mathcal{P}$, a set of instances $i \in \mathcal{I}$ and a cost metric $m : \mathcal{P} \times \mathcal{I} \to \mathbb{R}$, the algorithm selection problem consists in finding a mapping $s : \mathcal{I} \to \mathcal{P}$ from instances $\mathcal{I}$ to algorithms $\mathcal{P}$ such that the cost $\sum_{i \in \mathcal{I}} m(s(i), i)$ across all instances is optimized

Solution Approaches

- The algorithm selection problem is mainly solved with machine learning techniques.
- represent the problem instances by numerical features f,
- then algorithm selection can be seen as a multi-class classification problem by learning a mapping $f_i \mapsto \mathcal{A}$ for a given instance i.
- Instance features are numerical representations of instances. For example, we can count the number of variables, clauses, average clause length for Boolean formulas (static) or the result of running for a short time a stochastic local search solver on a Boolean formula (probing).

Solution Approaches

- Regression Approach
  predict the performance of each algorithm $\hat{m}_{\mathcal{A}} : \mathcal{I} \to \mathbb{R}$ and select the algorithm with the best predicted performance $arg \min_{\mathcal{A} \in \mathcal{P}} \hat{m}_{\mathcal{A}}(i)$ for a new instance i

- Clustering Approach
  Training consists of identifying the homogeneous clusters via an unsupervised clustering approach and associating an algorithm with each cluster. A new instance is assigned to a cluster and the associated algorithm selected.

- Pairwise Cost-Sensitive Classification Approach
  learn pairwise models between every pair of classes (here algorithms) and choose the class that was predicted most often by the pairwise models. We can weight the instances of the pairwise prediction problem by the performance difference between the two algorithms (we care most about getting predictions with large differences correct, but the penalty for an incorrect prediction is small if there is almost no performance difference). Therefore, each instance i for training a classification model $\mathcal{A}_1$ vs $\mathcal{A}_2$ is associated with a cost $|m(\mathcal{A}_1, i) - m(\mathcal{A}_2, i)|$

Variants of Algorithm Selection

- Online Selection
  Online algorithm selection in Hyper-heuristic refers to switching between different algorithms during the solving process. In contrast, (offline) algorithm selection is an one-shot game where we select an algorithm for a given instance only once.

- Computation of Schedules
  we select a time budget for each algorithm on a per-instance base. It improves the performance of selection systems in particular if the instance features are not very informative and a wrong selection of a single solver is likely.

- Selection of Parallel Portfolios
  Given the increasing importance of parallel computation, an extension of algorithm selection for parallel computation is parallel portfolio selection, in which we select a subset of the algorithms to simultaneously run in a parallel portfolio.

References

López-ibáñez M., Branke J., and Paquete L. (2021). **Reproducibility in evolutionary computation**. *ACM Trans. Evol. Learn. Optim.*, 1(4).