Heuristic Algorithms

Master's Degree in Computer Science/Mathematics

Roberto Cordone

DI - Università degli Studi di Milano

Schedule:	Thursday 14.30 - 16.30 on MS-Teams
	Friday 14.30 - 16.30 on MS-Teams
Office hours:	on appointment
E-mail:	roberto.cordone@unimi.it
Web page:	https://homes.di.unimi.it/cordone/courses/2020-ae/2020-ae.html
Ariel site:	https://rcordoneha.ariel.ctu.unimi.it

Compact statistical descriptions

The distribution function F_{δ_A} can be replaced or accompanied by more compact characterizations of the effectiveness of an algorithm

This typically involves classical statistical indices of

- position, such as the sample mean

$$\bar{\delta}_A = \frac{\sum_{I \in \bar{\mathcal{I}}} \delta_A(I)}{|\bar{\mathcal{I}}|}$$

- dispersion, such as the sample variance

$$\bar{\sigma}_A^2 = \frac{\sum_{I \in \bar{\mathcal{I}}} \left(\delta_A(I) - \bar{\delta}_A\right)^2}{|\bar{\mathcal{I}}|}$$

These indices “suffer” from the influence of outliers

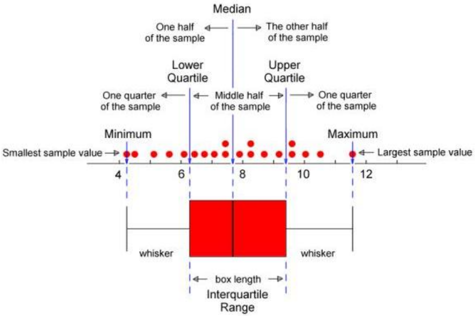
Other statistical indices are “stabler” and more detailed

- the sample median
- suitable sample quantiles

Boxplots

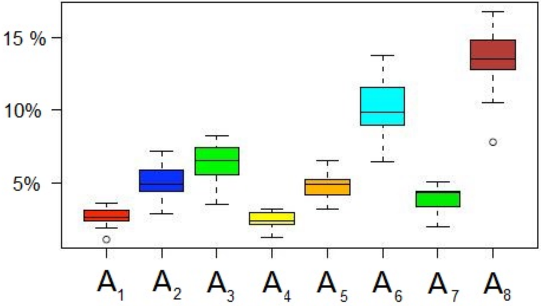
A graphic representation is the *boxplot* (or *box and whiskers diagram*)

- sample median ($q_{0.5}$)
- lower and upper sample quartiles ($q_{0.25}$ and $q_{0.75}$)
- the extreme sample values (often excluding the "outliers")



Comparison between algorithms with *boxplot* diagrams

A more compact comparison can be performed with *boxplot* diagrams



Strict dominance holds if and only if a boxplot is fully below the other (e. g., $A_7 - A_8$)

Probabilistic dominance holds only if each of the five quartiles represented in a boxplot is below the corresponding one of the other (e. g., $A_2 - A_3$)
(*but this is not sufficient*)

Relation between quality and computational time

Many heuristic algorithms find several solutions during their execution, instead of a single one, and consequently can be terminated prematurely

In particular, metaheuristics (random steps or memory mechanisms) have a computational time t fixed by the user and potentially unlimited

Let $\delta_A(t, I)$ be

- the relative difference achieved by A at time t on instance I
- $+\infty$ if A has not yet found a feasible solution at time t

As a function of time t, $\delta_A(t, I)$ is

- stepwise monotone nonincreasing
- constant after the regular termination $(t \geq T(I))$

Randomized algorithms

For randomized algorithms the relative difference $\delta_A(t, I, \omega)$ depends on

1. the execution time t
2. the instance $I \in \mathcal{I}$
3. the outcome $\omega \in \Omega$ of the random experiment guiding the algorithm (*that is the random seed*)

These algorithms therefore can be tested (for a fixed time)

1. on a sample of instances $\bar{\mathcal{I}}$ with constant seed ω
2. on a single instance I with a batch of seeds $\bar{\Omega}$

or both (several instances and several runs on each instance)

The results of tests on ω are usually summarized providing both:

- the minimum relative difference $\delta_A^*(t, I)$ and the total time $|\bar{\Omega}|\, t$
- the average relative difference $\bar{\delta}_A(t, I)$ and the single-run time t

Classification

The relation between solution quality and computational time allows to classify the algorithms into:

- complete: for each instance $I \in \mathcal{I}$, find the optimum in finite time

$$\exists \bar{t}_I \in \mathbb{R}^+ : \delta_A (I, t) = 0 \text{ for each } t \geq \bar{t}_I, I \in \mathcal{I}$$

(*It is another name for exact algorithms*)

- probabilistically approximately complete: for each instance $I \in \mathcal{I}$, find the optimum with probability converging to 1 as $t \to +\infty$

$$\lim_{t \to +\infty} Pr [\delta_A (I, t) = 0] = 1 \text{ for each } I \in \mathcal{I}$$

(*many randomized metaheuristics*)

- essentially incomplete: for some instances $I \in \mathcal{I}$, find the optimum with probability strictly < 1 as $t \to +\infty$

$$\exists I \in \mathcal{I} : \lim_{t \to +\infty} Pr [\delta_A (I, t) = 0] < 1$$

(*most greedy algorithms, local search algorithms, ...*)

A generalization

An obvious generalization replaces the search for the optimum with that of a given level of approximation

$$\delta_A(I,t) = 0 \;\rightarrow\; \delta_A(I,t) \leq \alpha$$

- α-complete algorithms: for each instance $I \in \mathcal{I}$, find an α-approximated solution in finite time (α-*approximated algorithms*)
- probabilistically approximately α-complete algorithms: for each instance $I \in \mathcal{I}$, find an α-approximated solution with probability converging to 1 as $t \rightarrow +\infty$
- essentially α-incomplete algorithms: for some instances $I \in \mathcal{I}$, find an α-approximated solution with probability strictly < 1 as $t \rightarrow +\infty$

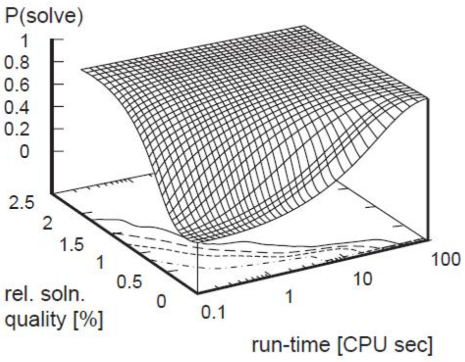
In conclusion, every algorithm provides compromises between

- a quality measure, described by the threshold α
- a time measure, described by the threshold t

The probability of success

Let the success probability $\pi_{A,n}(\alpha, t)$ be the probability that algorithm A find in time $\leq t$ a solution with a gap $\leq \alpha$ on an instance of size n

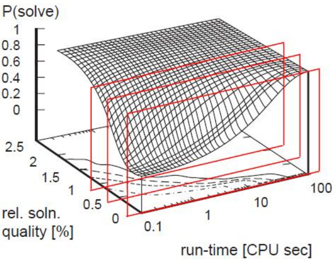
$$\pi_{A,n}(\alpha, t) = Pr\left[\delta_A(I, t) \leq \alpha | I \in \mathcal{I}_n, \omega \in \Omega\right]$$

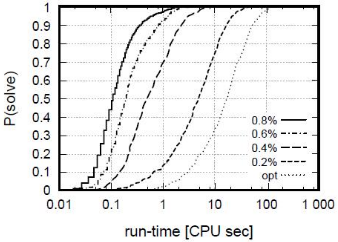


This yields different secondary diagrams

Qualified Run Time Distribution (*QRTD*) diagrams

The *QRTD diagrams* describe the profile of the time required to reach a specified level of quality





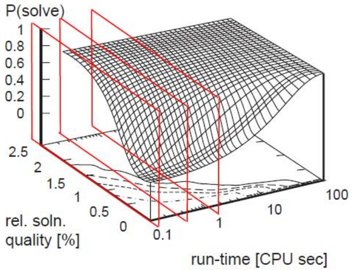
They are useful when the computational time is not a tight resource

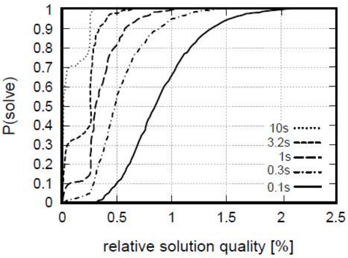
If the algorithm is

- complete, all diagrams reach 1 in finite time
- $\bar{\alpha}$-complete, all diagrams with $\alpha \geq \bar{\alpha}$ reach 1 in finite time
- $\bar{\alpha}$-incomplete, all diagrams with $\alpha \leq \bar{\alpha}$ do not reach 1

Timed Solution Quality Distribution (*TSQD*) diagrams

The *TSQD* diagrams describe the profile of the level of quality reached in a given computational time





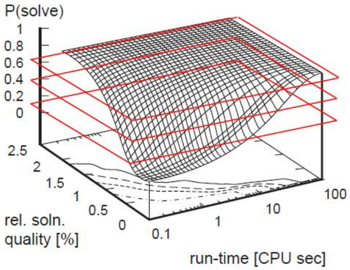
They are useful when the computational time is a tight resource

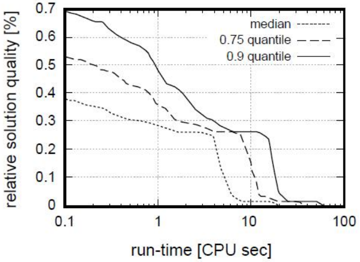
If the algorithm is

- complete, all diagrams with a sufficient t are step functions in $\alpha = 0$
- $\bar{\alpha}$-complete, all diagrams with a sufficient t reach 1 in $\alpha = \bar{\alpha}$
- probab. approx. $\bar{\alpha}$-complete, the diagrams converge to 1 in $\alpha = \bar{\alpha}$
- $\bar{\alpha}$-complete, all diagrams keep < 1 in $\alpha = \bar{\alpha}$

Solution Quality statistics over Time (*SQT*) diagrams

Finally, one can draw the level lines associated to different quantiles





They describe the compromise between quality and computational time

For a robust algorithm the level lines are very close to each other

Statistical tests

Diagrams and boxplots are qualitative: how to evaluate quantitatively if the empirical difference between algorithms A_1 and A_2 is significant?

Wilcoxon's test focuses on effectiveness (neglecting robustness)

- $f_{A_1}(I) - f_{A_2}(I)$ is a random variable defined on the sample space $\mathcal{I}$
- formulate a null hypothesis H_0 according to which
  the theoretical median of $f_{A_1}(I) - f_{A_2}(I)$ is zero
- extract a sample of instances $\bar{\mathcal{I}}$ and run the two algorithms on it, obtaining a sample of pairs of values (f_{A_1}, f_{A_2})
- compute the probability p of obtaining the observed result, or a more "extreme" one, assuming that H_0 is true
- set a significance level $\bar{p}$, i. e. the maximum acceptable probability
  - to reject H_0 assuming that it is true
  - that is, to consider two identical medians as different
  - that is, to consider two equivalent algorithms as differently effective (referring to the median of the gap)

  and reject H_0 when $p < \bar{p}$

Typical values for the significance level are $\bar{p} = 5\%$ or $\bar{p} = 1\%$

Wilcoxon's test (assumptions)

It is a nonparametric test, that is it does not make assumptions on the probability distribution of the tested values

It is useful to evaluate the performance of heuristic algorithms, because the distribution of the result $f_A(I)$ is unknown

It is based on the following assumptions:

- all data are measured at least on an ordinal scale
  (*the specific values do not matter, only their relative size*)
- the two data sets are matched and derive from the same population
  (*we apply A_1 and A_2 to the same instances, extracted from $\mathcal{I}$*)
- each pair of values is extracted independently from the others
  (*the instances are generated indipendently from one another*)

Wilcoxon's test (application)

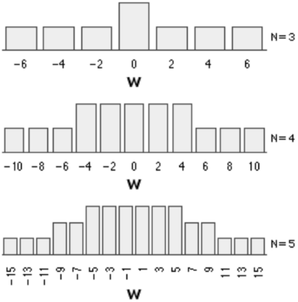
1. compute the absolute differences $|f_{A_1}(I_i) - f_{A_2}(I_i)|$ for all $I_i \in \bar{\mathcal{I}}$
2. sort them by increasing values and assign a rank R_i to each one
3. separately sum the ranks of the pairs with a positive difference and those of the pairs with a negative difference

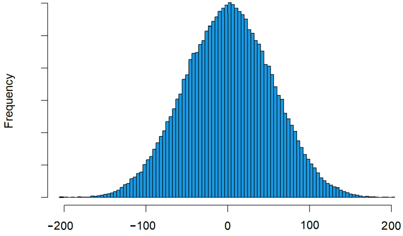
$$\begin{cases} W^+ = \displaystyle\sum_{i:f_{A_1}(I_i) > f_{A_2}(I_i)} R_i \\ W^- = \displaystyle\sum_{i:f_{A_1}(I_i) < f_{A_2}(I_i)} R_i \end{cases}$$

   If the null hypothesis H_0 were true, the two sums should be equal
4. the difference $W^+ - W^-$ allows to compute the value of p:
   each of the $|\bar{\mathcal{I}}|$ differences can be positive or negative: $2^{|\bar{\mathcal{I}}|}$ outcomes;
   p is the fraction with $|W^+ - W^-|$ equal or larger than the observed value
5. if $p < \bar{p}$, the difference is significant and
   - if $W^+ < W^-$, A_1 is better than A_2
   - if $W^+ > W^-$, A_1 is worse than A_2

The value of p is usually

- computed explicitly by enumeration when $|\bar{\mathcal{I}}| < 20$
- approximated with a normal distribution when $|\bar{\mathcal{I}}| \geq 20$





Of course, precomputed tables also exist

Possible conclusions

Wilcoxon's test can suggest

- that one of the two algorithms is significantly better than the other
- that the two algorithms are statistically equivalent

(*but take it as a stochastic response, and keep an eye on* p)

If the sample includes instances of different kinds, two algorithms could be overall equivalent, but nonequivalent on the single classes of instances

Dividing the sample could reveal

- classes of instances for which A_1 is better
- classes of instances for which A_2 is better
- classes of instances for which the two algorithms are equivalent

What about testing $\delta_A(I)$ *instead of* $f_A(I)$*?*