GUIDELINES FOR COMPARING METHODS

Éric Taillard

Applied Univ. of Western Switzerland

MIC'05, Wien, 21–26 August 2005

SUMMARY OF THE LECTURE

1. Main dimensions to consider in methods comparisons

Computational resources

Problem sets

Solution's quality

2. Few statistical test

Comparing proportions

Confidence interval

Comparing means of 2 samples

Comparing ranks

3. STAMP software

Statistical tests available

Comparing iterative methods

1. MAIN DIMENSIONS TO CONSIDER

Before making statistics, perform correct measures, clearly state what you want to show and the conditions of the experiments !

1.1 Computational resources

Computational effort

Memory requirement

Number of processors

1.2. Problem sets

Theoretical analysis based on instance size

Library containing only few instances with different structure

Stratification

1.3. Solution's quality (for optimization problems)

Objective value

Multi-objective

Deviation from a reference value

Standardization

1.1 Computational Resources

1.1.1 Computational effort

Relative measure :

Computational time (relative to a given machine)

Dongarra's factors

Absolute measure

Number of characteristic operations

$O(.)$ Notation

Empirical complexity

Relative computational effort

Often the only published measure

Strongly dependent on the machine used but also on :

- The operating system
- The programming language
- The programming style (reusable software)
- The compiler options
- The cache memory size
- Many other factors difficult to analyse

Not very accurate

Dongarra's factors

- Evaluated for linear algebra benchmarks
- Our machine configuration is certainly not the same as those used by Dongarra
- Observation : Factor of about 2 on the times estimated by Dongarra's factors and reality

COMMENTS ON RELATIVE TIME MEASURES

We are not necessarily better than a concurrent that take 3 times more time

Perform a complexity analysis and publish also absolute computational effort

Avoid stopping criteria such as CPU time > 100 s.

For iterative improvement searches : avoid photographic results

ABSOLUTE COMPUTATIONAL EFFORT

Count the number of characteristic operations *a*(*n*) as a function of problem size *n*.

Number of iterations (+ complexity of one iteration)

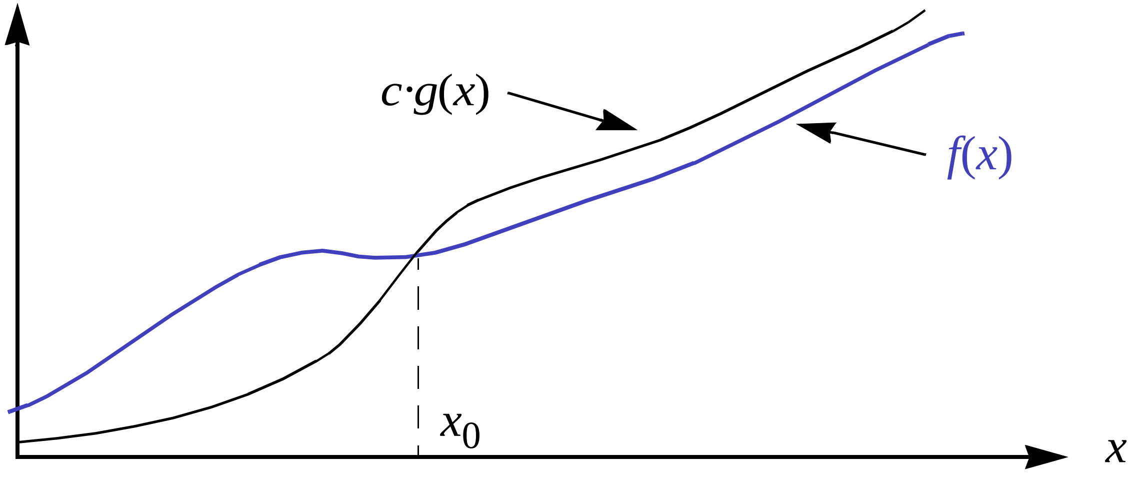
Number of nodes

Number of objective function evaluations

Use the O(.) (or Θ(.) or Ω(.)) notation to have an idea of the relative increase in computational effort when solving larger problem instances

Let f and g : 2 functions of a real variable x.

f is of order lower or equal to g if : $\exists\ x_0 > 0, c > 0$ such that $\forall\ x \geq x_0\ f(x) \leq c \cdot g(x)$



PROBLEM ENCOUNTERED IN PRACTICE WITH THEORETICAL COMPLEXITY

Example : What is the complexity of finding a 2-opt solution to a TSP ?

At least $\Omega(n^2)$, since each neighbour solution of a local optimum has to be checked

At most $O(n!)$ since the total number of solutions is limited by $n!$

The gap between $\Omega(n^2)$ and $O(n!)$ is huge and not so easy to reduce

EMPIRICAL ANALYSIS

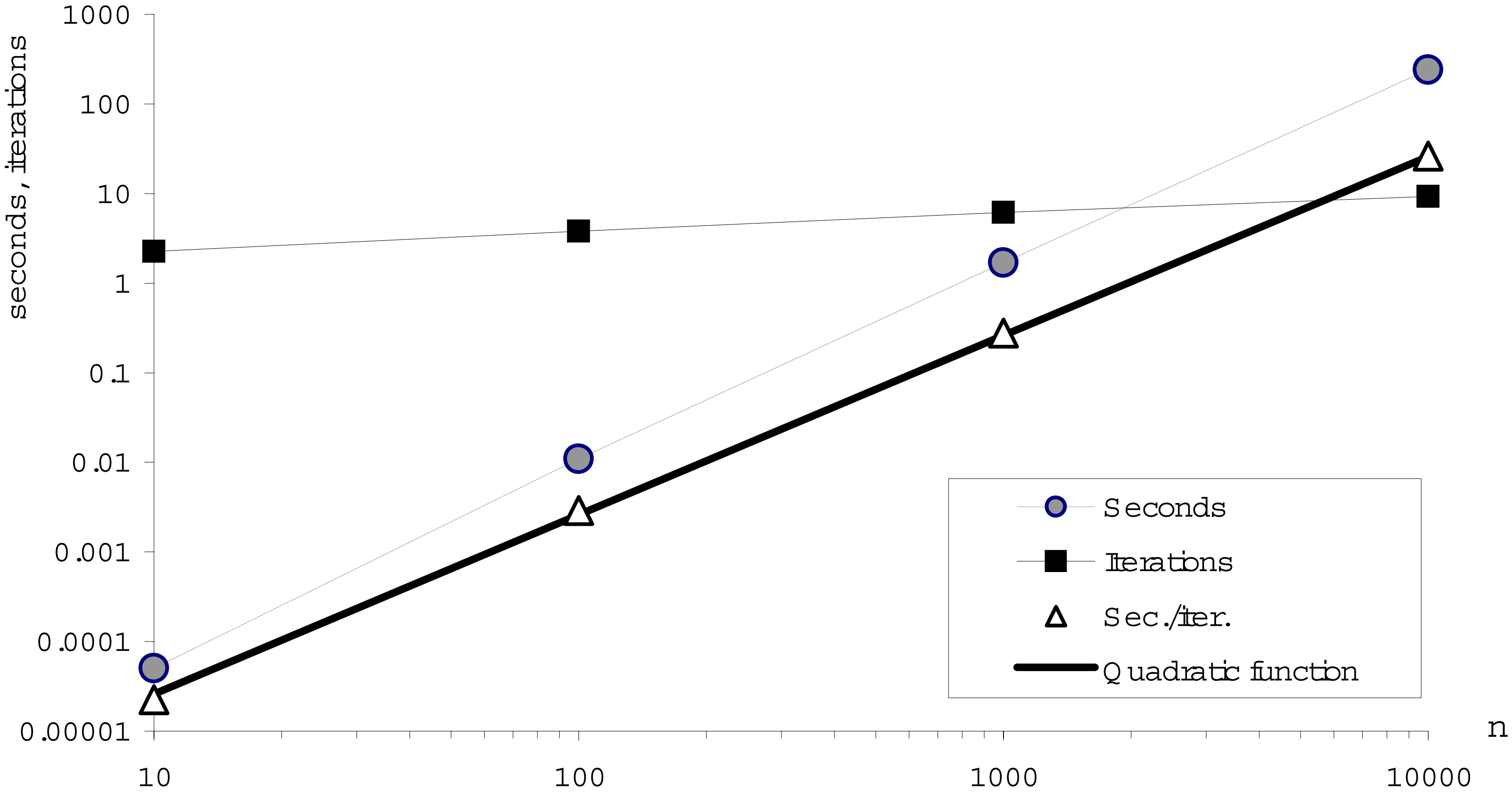
For Euclidean TSP with :

Cities uniformly distributed in the unit square

Randomly generated initial solution

Improving moves immediately performed

etc.



EMPIRICAL ANALYSIS

Usage of logarithmic scales helps for the empirical analysis

The average number of iterations increases polynomially.

The time per iteration increases quadratically, as expected

The total time can be approximated by $c \cdot n^{\alpha}$, with $\alpha \approx 2.22$

Our proposal :

Specify carefully the conditions of the experiment

Use a notation such as $\hat{O}(n^{2.22})$

Similar to statistical usage

Clearly show that this is derived by empirical observations.

1.1.2 Computational resources : Memory requirement

Analysis similar to computational effort

Memory size is not increasing as fast as CPU speed

Practical problem size may increase rapidly

Tomorrow, an application that requires $O(n^2)$ memory size could be intractable.

1.2. Problem instances

Pitfall : Using a limited set of problem instances

Small difference in size. Often, libraries contains instances with a factor of less than 10^2 between the largest and smallest instance. Typically : QAPLIB (toy-size = 15 ; 2 instances of size 150 ; a specific class of size 256)

Few instances with similar characteristics. Statistical analysis impossible

Over-fitted techniques. Methods are efficient for very specific instances

Stratify problem instances :

Several instances with same characteristics

Instances with different sizes

Instances with different structure

Report separate results

Study scalabiliy of the algorithm (POPMUSIC ; parallel implementations)

1.3 QUALITY MEASURE

Multi-objective optimization

Several metrics have been proposed ; well discussed in the literature

Objective function value

Can be used if problem sets are well stratified

Deviation from reference value

Often used : % from optimum, best known or lower/upper bound

Cannot be used if objective value crosses 0

Provide reference value to the reader !

Evolution of reference value

Projection in [0, 1] interval

2. Statistical tests

2.1 Comparing proportions

Proportion of runs that end successfully (exact methods ; decision problems)

2×2 contingency table

2.2 Confidence interval

Provide an interval in which a value of interest lies (mean, median, etc.)

Special case : the standard deviation in case of Gaussian distribution

Percentile Bootstrap technique

2.3 Comparing 2 means

t-Bootstrap technique with pivot

2.4 Comparing ranks

Mann-Whitney

2.1 Comparing proportions

Typical example : counting the number of successes (from Kim[3], JoH 9 (3), June 2003)

Problem instances	Number or runs	TCC	RSC	FSC	SSC	SHC	TMC
Sorting network design, *n* = 7	10	7	5	**8**	**8**	**8**	5
Sorting network design, *n* = 7	10	3	2	3	4	3	2
Sorting network design, *n* = 13	10	0	0	0	0	0	0
2DTTTgame	10	6	8	4	**9**	6	6
Nim(3,4,5,4)	10	3	2	**6**	**6**	4	3
Nim(5,7,11,6)	10	0	0	**1**	**1**	0	0

Question :

Is SSC significantly better than FSC for 2DTTT game ?

i. e. is a 9/10 rate of success significantly better than a 4/10 rate ?

TESTS OF HYPOTHESIS

You want to show that :

Hypothesis H_1 is most probably true (e.g. your method is better than concurrent one)

Technique of “proof” :

Suppose that the reverse hypothesis H_0 (null hypothesis) is true

(e.g. both methods are equally good)

Compute the probability p of obtaining the result observed under H_0

Reject H_0 at significance level α (and accept H_1 at confidence level $1 - \alpha$) if $p < \alpha$

Alternatively : compute a statistic $S_{obs} = S$(observation)

Read a value S_α in a table and reject H_0 at significance level α if $S_{obs} < S_\alpha$

2×2 Contingency table

	Number of successes	Number of failures	Total
Method A	a	$n - a$	n
Method B	b	$m - b$	m

Fisher's test :

Can be seen as a permutation test

Idea : Suppose that difference in proportions is due to chance

Count the number of different tables with same n, m and total number of successes with more extreme proportions

Test proposed by Taillard et al. (2004)

Problem : Compare proportions p_a and p_b of Yes answers in two samples

Samples : (n runs of Method A, a Yes), (m runs of Method B, b Yes), $a/n > b/m$

Null hypothesis : $p_a = p_b = p$

Alternate hypothesis : $p_a > p_b$ (unilateral test)

Method :

Suppose that the Yes answer has the same unknown probability p to appear for both methods A and B

The probability P to observe :

a Yes answers or more for method A

b Yes answers or less for method B

is given by

$$P = \sum_{i=a}^{n} \sum_{j=0}^{b} \binom{n}{i} \cdot p^i \cdot (1-p)^{n-i} \cdot \binom{m}{j} \cdot p^j \cdot (1-p)^{n_b - j}$$

Reject null hypothesis at significance level α if the maximum of P over p is smaller than α

TABLE FOR α = 1%

Pairs (*a*, *b*) for which a rate of success ≥ *a*/*n* is significantly higher than a rate of success ≤ *b*/*m*.

m	*n* = 2	3	4	5	6	7	8	9	10
2	—	—	—	—	—	(7,0)	(8,0)	(9,0)	(10,0)
3	—	—	(4,0)	(5,0)	(6,0)	(7,0)	(7,0)	(8,0)	(9,0)
4	—	(3,0)	(4,0)	(5,0)	(5,0) (6,1)	(6,0) (7,1)	(6,0) (8,1)	(7,0) (9,1)	(8,0) (10,1)
5	—	(3,0)	(4,0)	(4,0) (5,1)	(5,0) (6,1)	(5,0) (7,1)	(6,0) (7,1)	(6,0) (8,1) (9,2)	(7,0) (9,1) (10,2)
6	—	(3,0)	(3,0) (4,1)	(4,0) (5,1)	(4,0) (6,2)	(5,0) (6,1) (7,2)	(5,0) (7,1) (8,2)	(6,0) (8,1) (9,2)	(6,0) (8,1) (10,2)
7	(2,0)	(3,0)	(3,0) (4,1)	(4,0) (5,2)	(4,0) (5,1) (6,2)	(5,0) (6,1) (7,2)	(5,0) (6,1) (8,3)	(5,0) (7,1) (8,2) (9,3)	(6,0) (8,1) (9,2) (10,3)
8	(2,0)	(3,1)	(3,0) (4,2)	(4,1) (5,2)	(4,0) (5,1) (6,3)	(4,0) (6,2) (7,3)	(5,0) (6,1) (7,2) (8,3)	(5,0) (7,1) (8,2) (9,4)	(5,0) (7,1) (8,2) (9,3) (10,4)
9	(2,0)	(3,1)	(3,0) (4,2)	(3,0) (4,1) (5,3)	(4,0) (5,1) (6,3)	(4,0) (5,1) (6,2) (7,4)	(4,0) (6,1) (7,2) (8,4)	(5,0) (6,1) (7,2) (8,3) (9,4)	(5,0) (7,1) (8,2) (9,3) (10,5)
10	(2,0)	(3,1)	(3,0) (4,2)	(3,0) (4,1) (5,3)	(4,0) (5,2) (6,4)	(4,0) (5,1) (6,2) (7,4)	(4,0) (5,1) (6,2) (7,3) (8,5)	(4,0) (6,1) (7,2) (8,3) (9,5)	(5,0) (6,1) (8,2) **(9,4)** (10,5)

e.g. 9/10 rate is significantly higher than 4/10 rate

2.2 Confidence interval

Provide an interval [s_1, s_2] in which a value of interest lies

Typical value of interest : Mean, median

Typical usage

An author only provides average results (without confidence interval)

I want to know if my method provides results that are significantly different

Usual way of doing

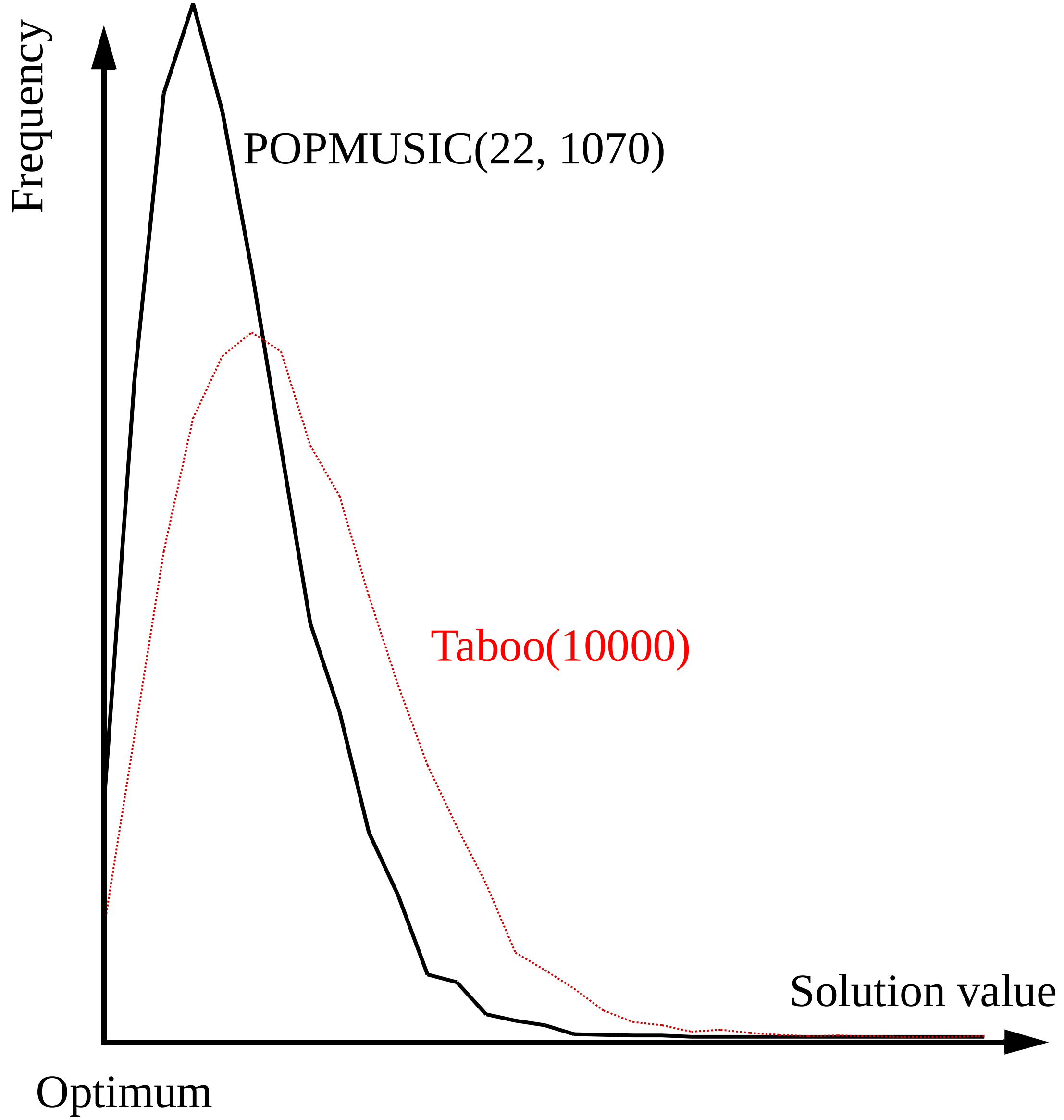
Assume that the distribution is Gaussian (oversimplification !)

Provide observed mean $\hat{\mu}$ (= median) and observed standard deviation $\hat{\sigma}$

At confidence level 95%, μ lies in $[\hat{\mu} - 1.96\hat{\sigma}, \hat{\mu} + 1.96\hat{\sigma}]$

SOLUTION VALUE DISTRIBUTIONS

Not Gaussian, not symmetrical ! Generally not known and cannot be reasonably determined.



BOOTSTRAP TECHNIQUE

Reference books

Efron and Tibshirani (1993)

Davison and Hinkley (2003)

General idea

Simulate data from a limited number of observation (resample from original data)

Making statistical inference easier in case analytical methods are too complicated to apply

Can be applied in almost any situation

Doesn’t need to oversimplify complex problems

PERCENTILE BOOTSTRAP

Observations and statistical function of interest:

$\boldsymbol{x} = (x_1, x_2, \ldots, x_n)$ $\quad s(\boldsymbol{x})$

Resampling :

For b = 1, …, B

Generate vector $\boldsymbol{x}^{\boldsymbol{b}} = (x_1^b, x_2^b, \ldots, x_n^b)$ with x_i^b randomly chosen among $(x_1, x_2, \ldots, x_n)$ with replacement

Compute $s^b = s(\boldsymbol{x}^{\boldsymbol{b}})$

Computation of the interval :

Sort the s^b by increasing values

At confidence level $1 - 2\alpha$, the value of interest lies in $[s_1 = s^{\alpha \cdot B}, s_2 = s^{(1-\alpha) \cdot B}]$

Comments on percentile bootstrap

Easy to implement

Adapted to metaheuristic practitioners familiar with simulation

Typical values :

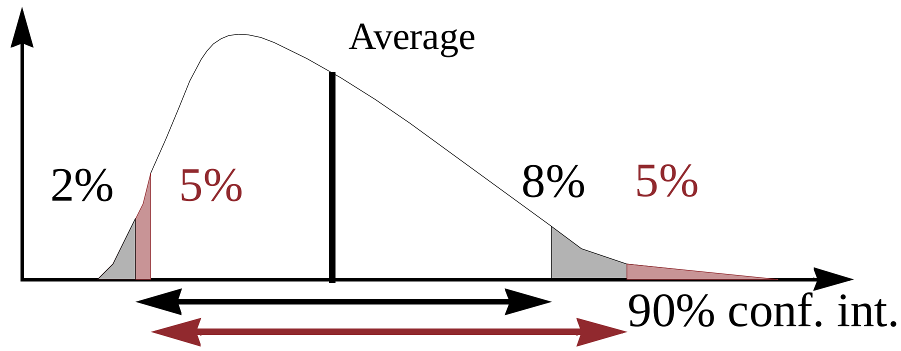
$B = 2000$, $\alpha = 2.5\%$, $s_1 = s^{50}$, $s_2 = s^{1950}$

But :

Does not provide the shortest possible interval

The values obtained can be biased

Requires relatively large resampling



Other technique : *BCa* (bias corrected and accelerated bootstrap)

Slightly more difficult to implement

In case of very asymetric distribution : try to transform data

2.3 Comparing the quality of two methods

Observations :

Method A executed n times, observed solution values $(x_1, x_2, \ldots, x_n)$

Method B executed m times, observed solution values $(y_1, y_2, \ldots, y_m)$

Typical question :

Does Method A provide average values significantly lower than Method B ?

Answers :

1) Oversimplification : Normality and same variance of both samples can be assumed

Use Student t-test

2) Use confidence intervals

3) Distribution functions can be different : use a more specific bootstrap technique

4) If small samples, very asymmetric distribution functions, bad variance estimate : compare ranks

2.4 Pivot-Bootstrap for comparing 2 means

Observations :

$(x_1, x_2, \ldots, x_n)$ $(y_1, y_2, \ldots, y_m)$

Compute :

Respective means and variances $\bar{x}, \bar{y}, v_x, v_y$

Average $\bar{z}$ of all the $n + m$ observations

Value $t_{obs} = t(\boldsymbol{x}, \boldsymbol{y}) = \dfrac{\bar{x} - \bar{y}}{\sqrt{v_x/n + v_y/m}}$

Vectors $\boldsymbol{x'}$ and $\boldsymbol{y'}$ with components $x'_i = x_i - \bar{x} + \bar{z}$ and $y'_i = y_i - \bar{y} + \bar{z}$

Resampling :

For b = 1, …, B

Generate vector $\boldsymbol{x^b} = (x_1^b, x_2^b, \ldots, x_n^b)$, resp. $\boldsymbol{y^b} = (y_1^b, y_2^b, \ldots, y_m^b)$ with

x_i^b, resp. y_i^b randomly chosen among $(x'_1, x'_2, \ldots, x'_n)$, resp. $(y'_1, y'_2, \ldots, y'_m)$

Compute associated values $t^b = t(\boldsymbol{x^b}, \boldsymbol{y^b})$

Significance level (estimated *p*-value) : $\#(t^b \le t_{obs})/B$

Rank-based test : Mann-Whitney

Very asymmetric distribution functions requires relatively large samples (10^2 elements) for having a good accuracy, even with bootstrap techniques.

Safer way of comparing methods : ranking the results and comparing the ranks

But : Information lost (difference between the means)

Something else is compared !

Observations :

Method A : $(x_1, x_2, \ldots, x_n)$ Method B : $(y_1, y_2, \ldots, y_m)$

Null hypothesis :

$\boldsymbol{H_0}$ **:** P(a run of B better than a run of A) < 1/2 (or : P(E(B) < E(A)) < 1/2 if distributions are similar)

Compute :

Mix all $n + m$ observations, rank them by decreasing quality

Decision :

If $\sum$ ranks heuristic $A > T_\alpha(n, m)$, reject $\boldsymbol{H_0}$ (α : significance level ; $T_\alpha(n, m)$ to be read in a table)

Other statistical tests :

Comparing several methods on several problem classes

Analysis of variance : ANOVA, MANOVA

Linear model, invariant variance, gaussian distribution

Friedman’s test

Interest for the practitioner ?

One method behave differently than the others

Which one ?

It cannot be excluded that all method behave similarly

The associated probability is not necessarily near to 1

3. COMPARING ITERATIVE METHODS

Idea :

Provide non-photographic information

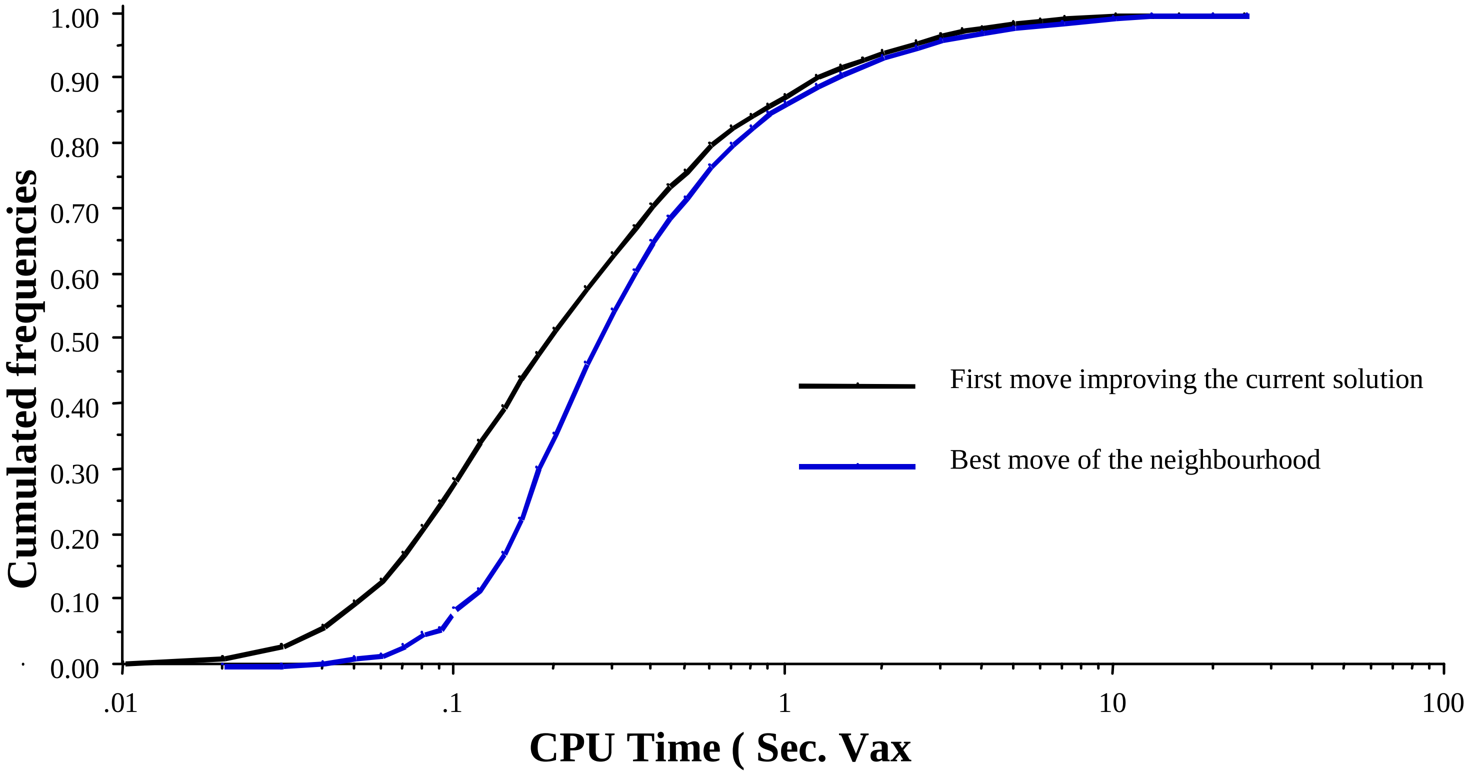
Provide graphical information (in complement to raw results)

Repeat a statistical test for every computational effort for which an improvement is observed

Provide *p*-values as a function of computational effort

COMPARING EVOLUTION OF SUCCESS RATES

From Taillard (1988)



CPU time to find optimal makespan (permutation flow-shop, 9 Jobs, 10 Machines)

Relatively small differences are significant (e.g. 50/100 and 60/100)

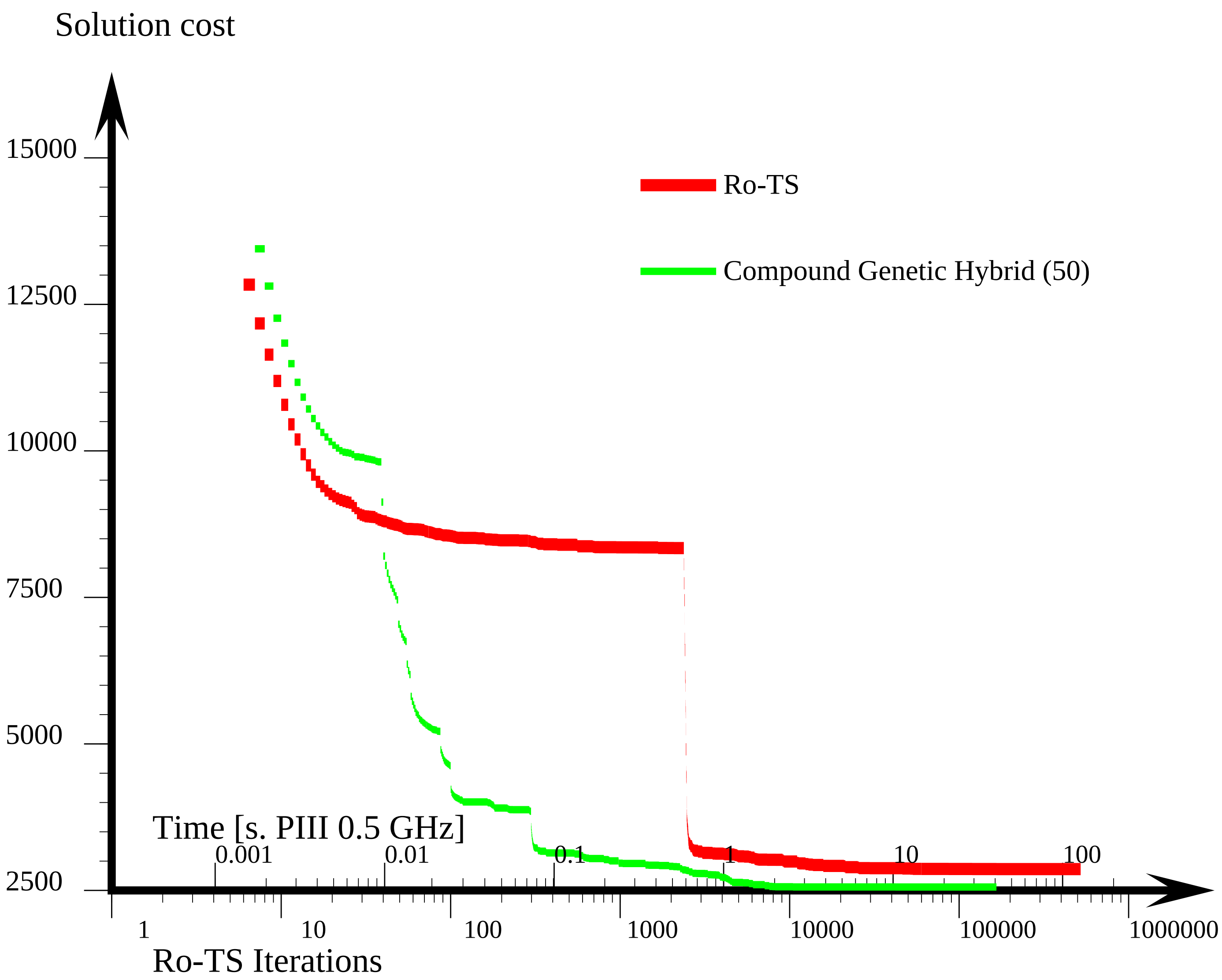
If exponentially distributed : multiple independent runs are equivalent to one long run

Comparing evolution of averages

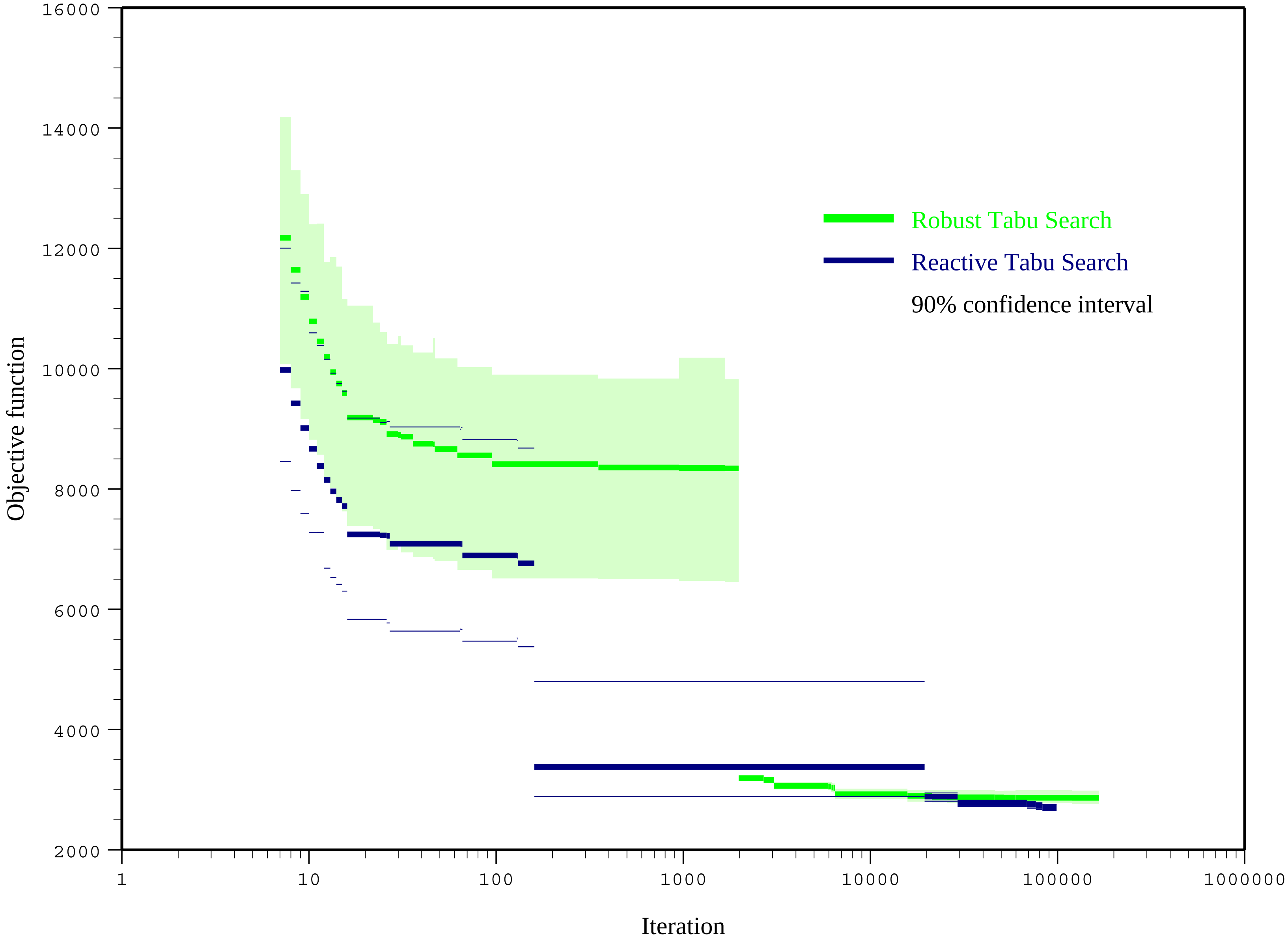
Data format :

Iteration		Quality	Time
r	1		
1		6.91525	0.02
2		5.06791	0.03
24		0.214607	0.11
43		0.167985	0.17
707		0.155952	2.39
1082		0.100172	3.73
2503		0.0844901	8.54
25000		0.0844901	83.0
r	2		
1		5.03441	0.01
…			

STAMP software : showing means or medians



STAMP: SHOWING CONFIDENCE INTERVALS

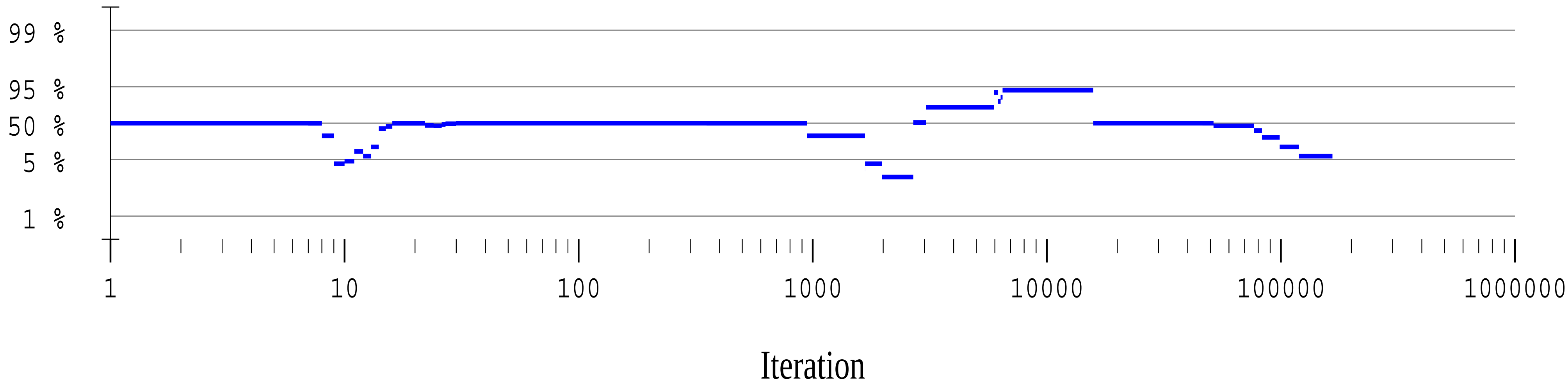


Generated by Ph. Waelti

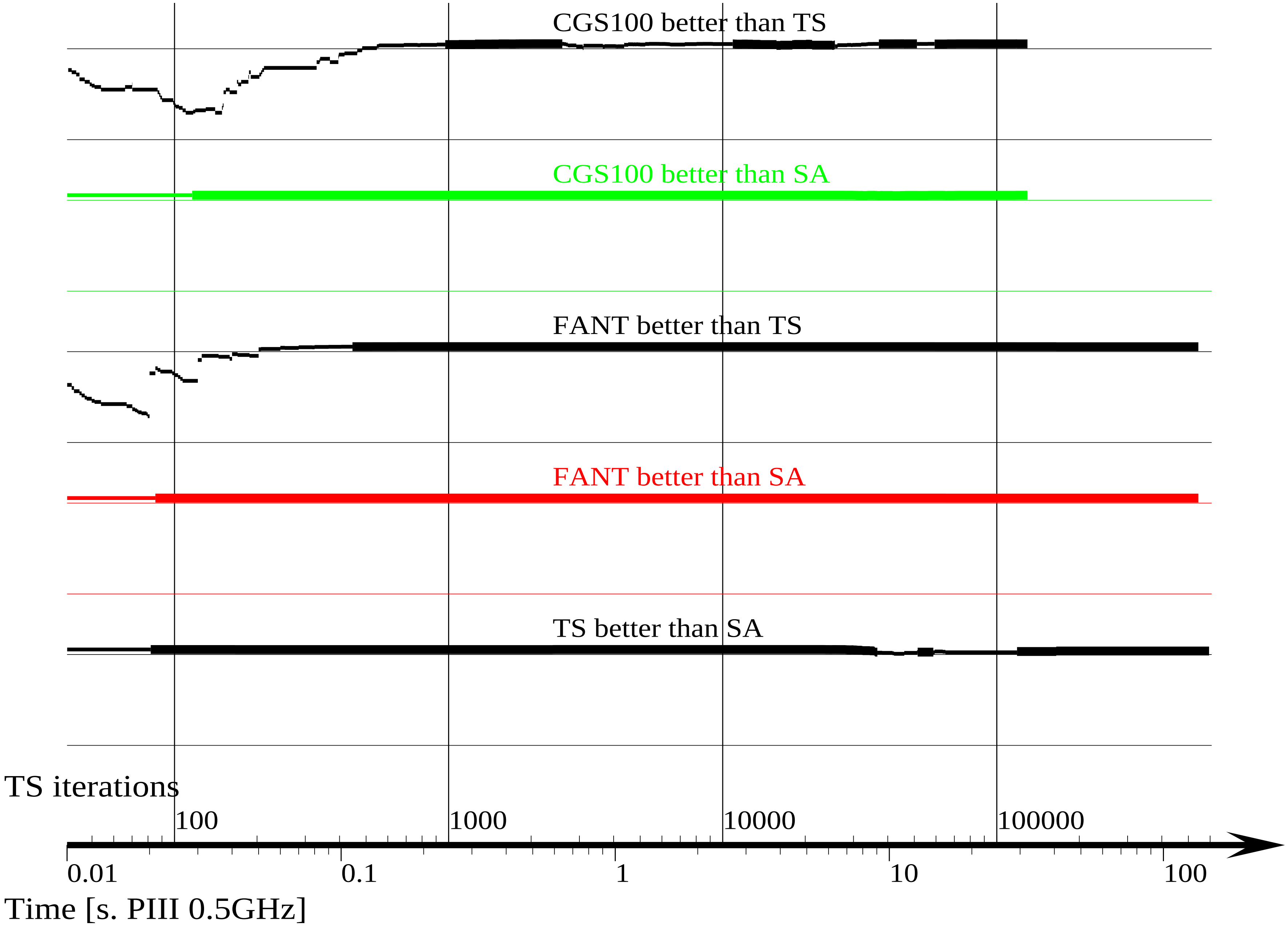
STAMP : SHOWING EVOLUTION OF P-VALUE

Repeated Mann-Whitney test

p-value scale of the form : $\dfrac{(2p-1)^{2k+1}-1}{2}$; $k = 0$: linear ; $k > 0$ expands values near extremities.



STAMP : MULTIPLE COMPARISONS



STAMP CAPABILITIES

Web on-line computation

`http://qualopt.eivd.ch`

Statistical on-line computation :

p-values for proportion comparisons

p-values for Mann-Withney test

Confidence intervals for mean and median, based on *BCa* bootstrap

Comparisons of means, based on studentized pivot Bootstrap

Production of diagrams

Average, median, including confidence intervals

Proportion of successes

Evolution of *p*-values

CONCLUSIONS

Better reflection before performing numerical experiments

Complexity analysis

Benchmarks choice

Better presentation of results

A diagram is 1000 words worth

Keep raw results and codes available on-line

Improvement in results significance

Statistically justified analysis

Better understanding of iterative methods

Help in designing powerful solving methods