IE552 Lecture Notes 7

Performance Evaluation of Heuristics

Introduction

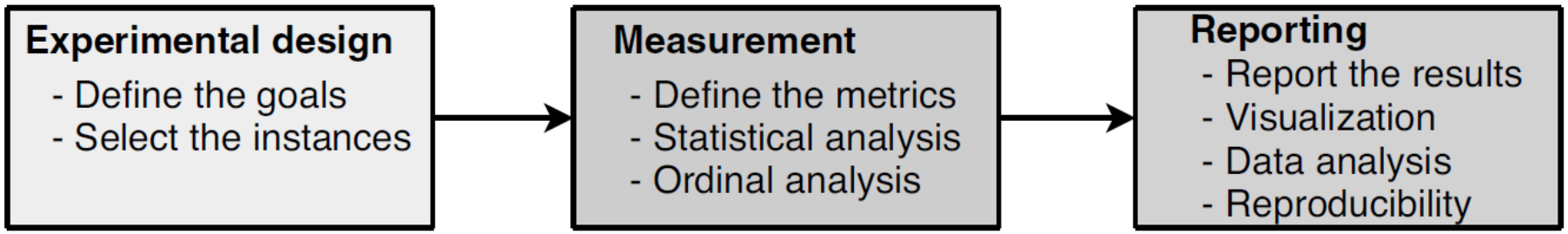
- Performance analysis of metaheuristics is a necessary task to perform and must be done on a fair basis.
- A theoretical approach is generally not sufficient to evaluate a metaheuristic.
- We will addresses some guidelines of evaluating experimentally a metaheuristic and/or comparing metaheuristics in a rigorous way.

Three Steps

To evaluate the performance of a metaheuristic in a rigorous manner, the following three steps must be considered:

- **Experimental design:** In the first step, the goals of the experiments, the selected instances, and factors have to be defined.
- **Measurement:** In the second step, the measures to compute are selected. After executing the different experiments, statistical analysis is applied to the obtained results. The performance analysis must be done with state-of-the-art optimization algorithms dedicated to the problem.
- **Reporting:** Finally, the results are presented in a comprehensive way, and an analysis is carried out following the defined goals. Another main issue here is to ensure the reproducibility of the computational experiments.

Three Steps


Experimental design
- Define the goals
- Select the instances
Measurement
- Define the metrics
- Statistical analysis
- Ordinal analysis
Reporting
- Report the results
- Visualization
- Data analysis
- Reproducibility

Experimental Design

- In the computational experiment of a metaheuristic, the goals must be clearly defined.
- All the experiments, reported measures, and statistical analysis will depend on the purpose of designing the metaheuristic.
- Indeed, a contribution may be obtained for different criteria such as
  - search time,
  - quality of solutions,
  - robustness in terms of the instances,
  - solving large-scale problems,
  - parallel scalability in terms of the number of processors,
  - easiness of implementation,
  - easiness to combine with other algorithms,
  - flexibility to solve other problems or optimization models,
  - innovation using new nature-inspired paradigms,
  - automatic tuning of parameters,
  - providing a tight approximation to the problem, and so on.

Experimental Design

- Once the goals and factors are defined, methods from DOE can be suggested to conduct computational tests to ensure a rigorous statistical analysis.
- It consists in selecting a set of combinations of values of factors to experiment.
- Then, the effect of a parameter (factor) p will be the change in the results obtained by the modification of the values of the parameter.
- Here we need to talk about parameter tuning.

Parameter Tuning

- Many parameters have to be tuned for any metaheuristic.
- Parameter tuning may allow a larger flexibility and robustness, but requires a careful initialization.
- Those parameters may have a great influence on the efficiency and effectiveness of the search.
- It is not obvious to define *a priori* which parameter setting should be used.
- The optimal values for the parameters depend mainly on the problem and even the instance to deal with and on the search time that the user wants to spend in solving the problem.
- A universally optimal parameter values set for a given metaheuristic does not exist.
- There are two different strategies for parameter tuning: the *off-line* and the *online*
- Off-line: the values of different parameters are fixed before the execution of the metaheuristic, whereas in the online approach, the parameters are controlled and updated dynamically or adaptively during the execution of the metaheuristic.

Off-Line Parameter Initialization and Experimental Design

- Usually, metaheuristic designers tune one parameter at a time, and its optimal value is determined empirically. In this case, no interaction between parameters is studied.
- This sequential optimization strategy (i.e., one-by-one parameter) do not guarantee to find the optimal setting even if an exact optimization setting is performed.
- To overcome this problem, we use experimental design with the following concepts:
  - Factors that represent the parameters to vary in the experiments.
  - Levels that represent the different values of the parameters, which may be quantitative (e.g., mutation probability) or qualitative (e.g., neighborhood).

Off-Line Parameter Initialization and Experimental Design

- Let us consider n factors in which each factor has k levels, a full factorial design needs n^k experiments. Then, the “best” levels are identified for each factor.
- Hence, the main drawback of this approach is its high computational cost especially when the number of parameters (factors) and their domain values are large, that is, a very large number of experiments must be realized.
- However, a small number of experiments may be performed by using Latin hypercube designs, sequential design, or fractional design...
- In off-line parameter initialization, the search for the best tuning of parameters of a metaheuristic in solving a given problem may be formulated as an optimization problem.
- Hence, this meta-optimization approach may be performed by any (meta)heuristic, leading to a meta-metaheuristic (or meta-algorithm) approach. Meta-optimization may be considered a hybrid scheme in metaheuristic design.

Experimental Design

- Once the goals are defined, the selection of the input instances to perform the evaluation must be carefully done.
- The structure associated with the input instances may influence significantly the performance of metaheuristics. Two types of instances exist: real-life and constructed instances

Real-life instances:

- They represent practical instances of the problem to be solved.
- If available, they constitute a good benchmark to carry out the performance evaluation of a metaheuristic.
- For some problems, it is difficult to obtain real-life instances for confidentiality reasons. In fact, most of the time, those data are proprietary and not public.
- For other problems, it is difficult to obtain a large number of real-life instances for financial reasons. For instance, in computational biology and bioinformatics, the generation of some genomic or proteomic data has a large cost. Also, collecting some real-life instances may be time consuming.

Experimental Design

- Once the goals are defined, the selection of the input instances to perform the evaluation must be carefully done.
- The structure associated with the input instances may influence significantly the performance of metaheuristics. Two types of instances exist: real-life and constructed instances

Constructed instances:

- Many public libraries of “standard” instances are available on Internet.
- They contain well-known instances for global optimization, combinatorial optimization, and mixed integer programs such as OR-Library38, MIPLIB39, DIMACS challenges40, SATLIB for satisfiability problems, and the TSPLIB41 (resp. QAPLIB) for the traveling salesman problem (resp. the quadratic assignment problem).

Experimental Design

- In addition to some real-life instances, those libraries contain in general synthetic or randomly generated instances.
- A disadvantage of random instances is that they are often too far from real-life problems to reflect their structure and important characteristics.
- The advantage of synthetic data is that they preserve the structure of real-life instances. Using a synthetic program, different instances in size and structure may be generated.
- Evaluating the performances of a given metaheuristic using only random instances may be controversial. For instance, the structure of uniformly generated random instances may be completely different from real-life instances of the problem, and then the effectiveness of the metaheuristic will be completely different in practice.
- **Example 1.41 from the book.** Distance matrix uniform between 0 and 20 → any tour is good enough

Experimental Design

- The selection of the input instances to evaluate a given metaheuristic may be chosen carefully. The set of instances must be diverse in terms of the size of the instances, their difficulties, and their structure.
- It must be divided into two subsets: the first subset will be used to tune the parameters of the metaheuristic and the second subset to evaluate the performance of the search algorithms.
- The calibration of the parameters of the metaheuristics is an important and tricky task. Most metaheuristics need the tuning of various parameters that influence the quality of the obtained results. The values of the parameters associated with the used metaheuristics must be same for all instances.
- A single set of the parameter values is determined to solve all instances. No fine-tuning of the values is done for each instance unless the use of an automatic off-line or online initialization strategy.
- Indeed, this will cause an overfitting of the metaheuristic in solving known and specific instances. The parameter values will be excellent to solve the instances that serve to calibrate the parameters and very poor to tackle other instances. The robustness of the metaheuristic will be affected to solve unknown instances.

Experimental Design

- If you do tuning for each instance then the time to determine the parameter values of the metaheuristic to solve a given instance must be taken into account in the performance evaluation.
- In this case, different parameter values may be adapted to different structures and sizes of the instances.
- Up to this point, we talked about the experimental desing which is the first step.
- Now we move to the second step which is measurement.

Measurement

- In the second step, the performance measures and indicators to compute are selected.
- After executing the different experiments, some statistical analysis will be applied to the obtained results.
- In exact optimization methods, the efficiency in terms of search time is the main indicator to evaluate the performances of the algorithms as they guarantee the global optimality of solutions.
- To evaluate the effectiveness of metaheuristic search methods, other indicators that are related to the quality of solutions have to be considered.
- Performance indicators of a given metaheuristic may be classified into three groups solution quality, computational effort, and robustness.
- Other qualitative criteria such as the development cost, simplicity, ease of use, flexibility (wide applicability), and maintainability may be used.

Measurement: Quality of solutions

- Performance indicators for defining the quality of solutions in terms of precision are generally based on measuring the distance or the percent deviation of the obtained solution to one of the following solutions

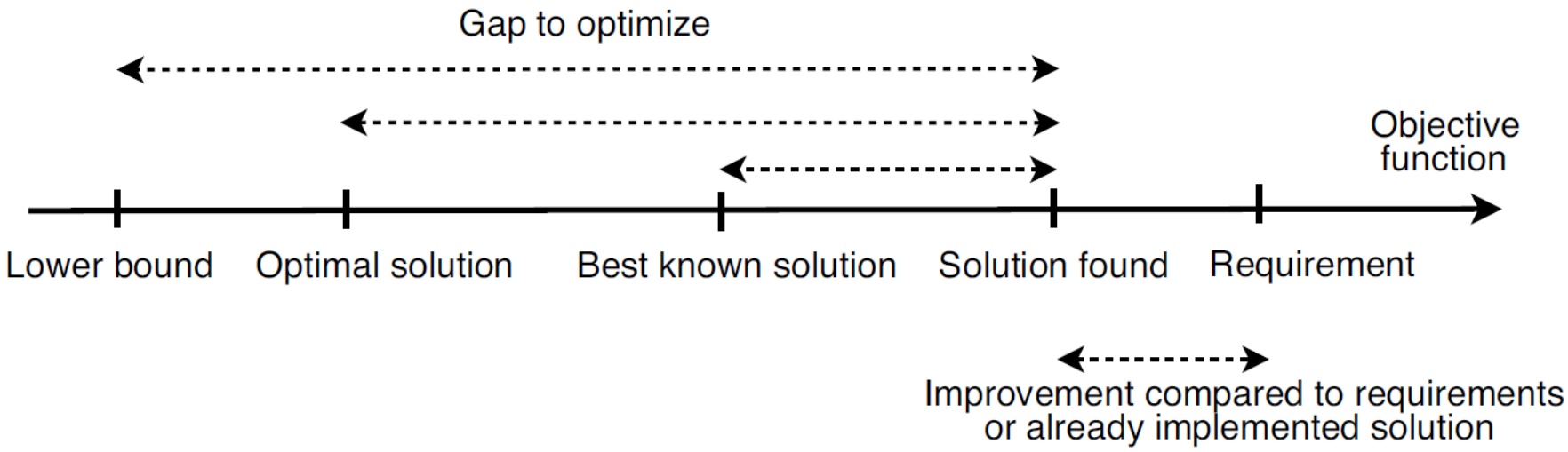


FIGURE 1.27 Performance assessment of the quality of the solutions. We suppose a minimization problem.

Measurement: Quality of solution

- **Global optimal solution:** The use of global optimal solutions allows a more absolute performance evaluation of the different metaheuristics. The absolute difference may be defined as

$$|f\ (s) -\ f\ (s^*)|\ or\ \frac{|f\ (s) -\ f\ (s^*)|}{f(s^*)}$$

where s is the obtained solution and s^* is the global optimal solution.

- Since those measures are not invariant under different scaling of the objective function, the following absolute approximation may be used:

$$|f\ (s)\ -\ f\ (s^*)|/|f_{worst}\ -\ f\ (s^*)|$$

Note: For some problems, it is difficult to find the worst solution.

Measurement: Quality of solution

- The global optimal solution may be found by an exact algorithm or may be available using “constructed” instances where the optimal solution is known a priori (by construction).
- Built-in optimal solutions have been considered for many academic problems:
  - traveling salesman problem,
  - graph partitioning problem ,
  - Steiner tree problem,
  - vertex packing, and
  - maximum clique.
- Unfortunately, for many complex problems, global optimal solutions could not be available.
- There are also some statistical estimation techniques of optimal values in which a sample of solutions is used to predict the global optimal solution

Measurement: Quality of solution

- **Lower and upper bounds:** For optimization problems where the global optimal solution is not available, tight lower bounds may be considered as an alternative to global optimal solutions.
- For some optimization problems, tight lower bounds are known and easy to obtain.

Example 1.43 Simple lower bound for the TSP. The Held–Karp (HK) 1-tree lower bound for the symmetric TSP problem is quick and easy to compute [371]. Given an instance (V, d) where V is the set of n cities and d the distance matrix. A node $v_0 \in V$ is selected. Let r be the total edge length of a minimum spanning tree over the $n-1$ cities ($v \in V - \{v_0\}$). The lower bound t is represented by the r value plus the two cheapest edges incident on v_0.

$$t = r + \min\{d(v_0, x) + d(v_0, y) : x, y \in V - \{v_0\}, x \neq y\}$$

Indeed, any TSP tour must use two edges e and f incident on the node v_0. Removing these two edges and the node v_0 from the tour yields a spanning tree of $V - \{v_0\}$. Typically, the lower bound t is 10% below the global optimal solution.

Measurement: Quality of solution

- Different relaxation techniques may be used to find lower bounds such as the classical continuous relaxation and the Lagrangian relaxation.
- In continuous relaxation for IP problems, the variables are supposed to be real numbers instead of integers.
- In Lagrangian relaxation, some constraints multiplied by Lagrange multipliers are incorporated into the objective function.
- If the gap between the obtained solution and the lower bound is small, then the distance of the obtained solution to the optimal solution is smaller.
- In the case of null distance, the global optimality of the solution is proven.
- In the case of a large gap (e.g., > 20%), it can be due to the bad quality of the bound or the poor performance of the metaheuristic.

Measurement: Quality of solution

- **Best known solution:** For many classical problems, there exist libraries of standard instances available on the Web. For those instances, the best available solution is known and is updated each time an improvement is found.
- **Requirements or actual implemented solution:** For real-life problems, a decision maker may define a requirement on the quality of the solution to obtain.
- This solution may be the one that is currently implemented. These solutions may constitute the reference in terms of quality.

Measurement: Computational Effort

- The efficiency of a metaheuristic may be demonstrated using a theoretical analysis or an empirical one. In theoretical analysis, the worst-case complexity of the algorithm is generally computed.
- In general, reporting the asymptotic complexity is not sufficient and cannot tell the full story on computational performances of metaheuristics.
- The average-case complexity, if it is possible to compute, is more practical.
- In empirical analysis, measures related to the computation time of the metaheuristic used to solve a given instance are reported.
- The meaning of the computation time must be clearly specified: CPU time or wall clock time, with or without input/output and preprocessing/postprocessing time.

Measurement: Computational Effort

- The main drawback of computation time measure is that it depends on the computer characteristics such as the hardware (e.g., processor, memories:RAM and cache, parallel architecture), operating systems, language, and compilers on which the metaheuristic is executed.
- Some indicators that are independent of the computer system may also be used, such as the number of objective function evaluations.
  - It is an acceptable measure for time-intensive and constant objective functions.
  - Using this metric may be problematic for problems where the evaluation cost is low compared to the rest of the metaheuristics or is not time constant in which it depends on the solution evaluated and time.
  - This appears in some applications with variable length representations (genetic programming, robotics, etc.) and dynamic optimization problems.

Measurement: Computational Effort

Different stopping criteria may be used: time to obtain a given target solution, time to obtain a solution within a given percentage from a given solution (e.g., global optimal, lower bound, best known), number of iterations, and so on.

Measurement: Robustness

- There is no commonly acceptable definition of robustness.
- Different alternative definitions exist for robustness. In general, robustness is insensitivity against small deviations in the input instances (data) or the parameters of the metaheuristic. The lower the variability of the obtained solutions the better the robustness.
- In the metaheuristic community, robustness also measures the performance of the algorithms according to different types of input instances and/or problems.
- The metaheuristic should be able to perform well on a large variety of instances and/or problems using the same parameters. The parameters of the metaheuristic may be overfitted using the training set of instances and less efficient for other instances.
- In stochastic algorithms, the robustness may also be related to the average/deviation behavior of the algorithm over different runs of the algorithm on the same instance.

Measurement: Statistical Analysis

- Once the experimental results are obtained for different indicators, methods from statistical analysis can be used to conduct the performance assessment of the designed metaheuristics.
- While using any performance indicator (e.g., the quality of solutions c_i obtained by different metaheuristics M_i or their associated computational efforts t_i), some aggregation numbers that summarize the average and deviation tendencies must be considered.
- Then, different statistical tests may be carried out to analyze and compare the metaheuristics. The statistical tests are performed to estimate the confidence of the results to be scientifically valid (i.e., determining whether an obtained conclusion is due to a sampling error).
- The selection of a given statistical hypothesis testing tool is performed according to the characteristics of the data (e.g., variance, sample size)

Measurement: Statistical Analysis

- Under normality conditions, the most widely used test is the paired t-test. Otherwise, a nonparametric analysis may be realized such as the Wilcoxon test and the permutation test.
- For a comparison of more than two algorithms, ANOVA models are well-established techniques to check the confidence of the results.
- Multivariate ANOVA models allow simultaneous analysis of various performance measures (e.g., both the quality of solutions and the computation time).
- Kolmogorov–Smirnov test can be performed to check whether the obtained results follow a normal distribution…

Measurement: Statistical Analysis

- These different statistical analysis procedures must be adapted for nondeterministic (or stochastic) algorithms.
- Indeed, most of the metaheuristics belong to this class of algorithms.
- Many trials (at least 10, more than 100 if possible) must be carried out to derive significant statistical results.
- From this set of trials, many measures may be computed: mean, median, minimum, maximum, standard deviation, the success rate that the reference solution (e.g., global optimum, best known, given goal) has been attained, and so on.
- The success rate represents the number of successful runs over the number of trials.

Measurement: Ordinal Data Analysis

- In comparing n metaheuristics for a given number of m experiments (instances, etc.), a set of ordinal values o_k $(1 \leq k \leq m)$ are generated for each method.
- For a given experiment, each ordinal value o_k denotes the rank of the metaheuristic compared to the other ones $(1 \leq o_k \leq n)$.
- Some ordinal data analysis methods may be applied to be able to compare the different metaheuristics.
- Those ordinal methods aggregate m linear orders O_k into a single linear order O so that the final order O summarizes the m orders O_k.

Reporting

- The interpretation of the results must be explicit using the defined goals and considered performance measures.
- In general, it is not sufficient to present the large amount of data results using tables. Some visualization tools to analyze the data are welcome to complement the numerical results.
- Indeed, graphical tools such as deviation bars (confidence intervals, box plots) and interaction plots allow a better understanding of the performance assessment of the obtained results.
- Interaction plots represent the interaction between different factors and their effect on the obtained response (performance measure).
- Box plots illustrate the distribution of the results through their five-number summaries: the smallest value, lower quartile (Q1), median (Q2), upper quartile (Q3), and largest value.
- They are useful in detecting outliers and indicating the dispersion and the skewness of the output data without any assumptions on the statistical distribution of the data.

Reporting

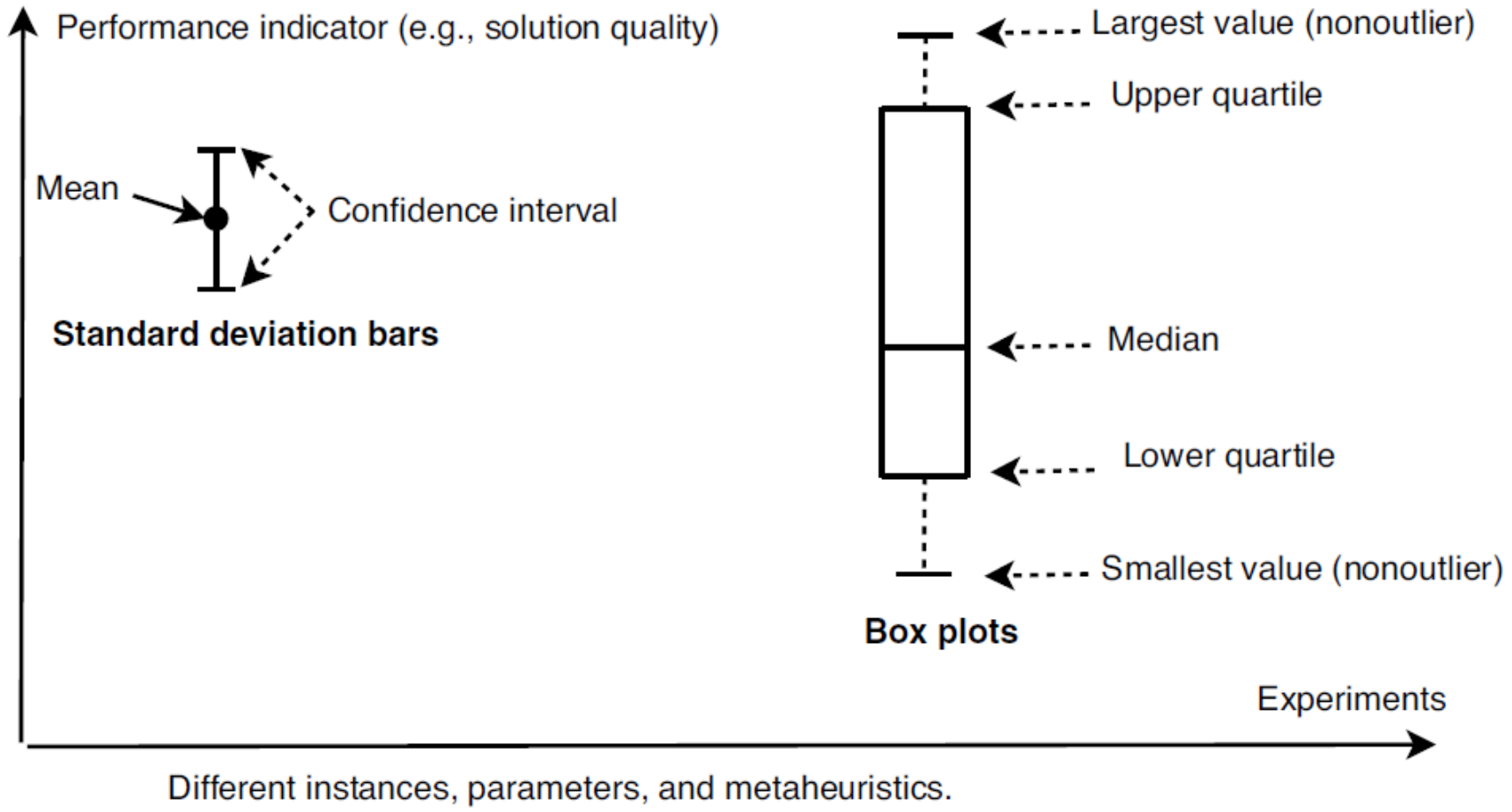
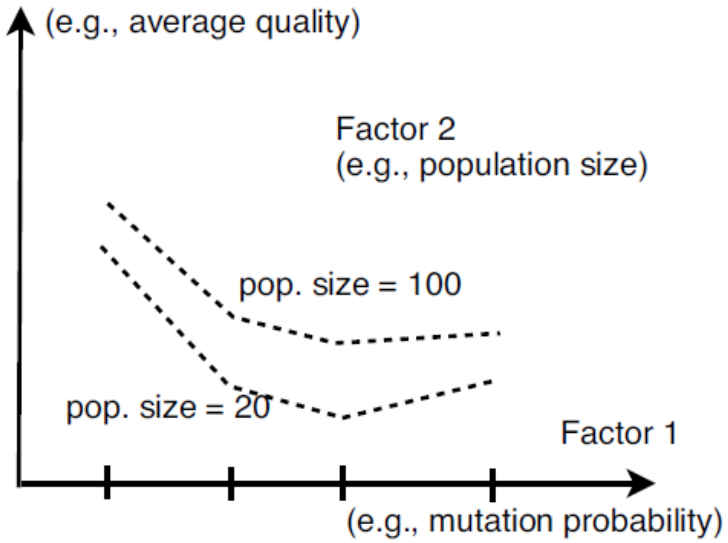


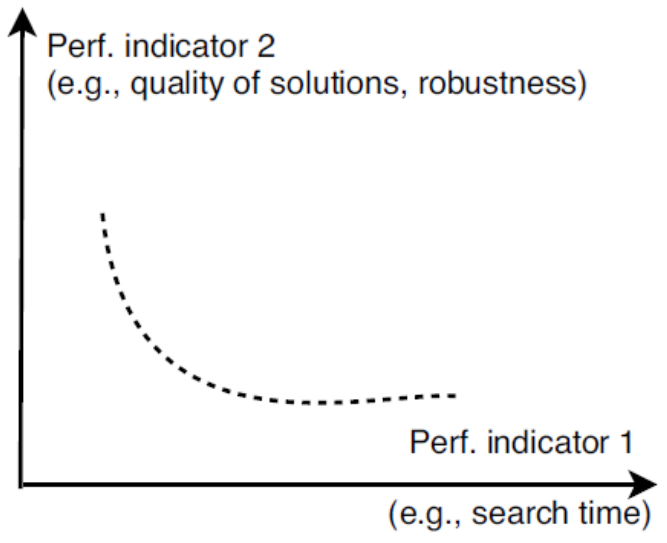
FIGURE 1.30 Some well-known visualization tools to report results: deviation bars, confidence intervals.

Reporting

- Moreover, it is important to use scatter plots to illustrate the compromise between various performance indicators.
- For instance, the plots display quality of solutions versus time, or time versus robustness, or robustness versus quality.



(a) Interaction plots



(b) Trade-off scatter plots

Reporting

- Other plots measure the impact of a given factor on a performance indicator: time versus instance size and quality versus instance size.
- Indeed, analyzing the relationship between the quality of solution, the search time, the robustness, and the size/structure of instances must be performed in a comprehensive way.
- Other visualization tools may be used such as half-normal plots and histograms.
- It would also be interesting to report negative results on applying a given metaheuristic or a search component to solve a given instance, problem, or class of problems.
- Indeed, most of the time only positive results are reported in the literature. From negative results, one may extract useful knowledge.

Conclusion

- A metaheuristic must be well documented to be reproduced.
- The program must be described in detail to allow its reproduction. If possible, making available the program, the instances, and the obtained results (complete solutions and the different measures) on the Web will be a plus.
- The different used parameters of the metaheuristic must be reported.
- Using different parameters in solving the different instances must also be reported.