Stochastic Algorithms Assessment Using Performance Profiles

Lino Costa
Department of Production and Systems Engineering, University of Minho
Campus de Gualtar
4710-057 Braga, Portugal
lac@dps.uminho.pt

Isabel Espírito Santo
Department of Production and Systems Engineering, University of Minho
Campus de Gualtar
4710-057 Braga, Portugal
iapinho@dps.uminho.pt

Pedro Oliveira
Instituto de Ciências Biomédicas Abel Salazar, Universidade do Porto
Largo Prof. Abel Salazar, 2
4099-003 Porto, Portugal
pnoliveira@icbas.up.pt

ABSTRACT

Optimization with stochastic algorithms has become a relevant approach, specially, in problems with complex search spaces. Due to the stochastic nature of these algorithms, the assessment and comparison is not straightforward. Several performance measures have been proposed to overcome this difficulty. In this work, the use of performance profiles and an analysis integrating a trade-off between accuracy and precision are carried out for the comparison of two stochastic algorithms. Traditionally, performance profiles are used to compare deterministic algorithms. This methodology is applied in the comparison of two stochastic algorithms - genetic algorithms and simulated annealing. The results highlight the advantages and drawbacks of the proposed assessment.

Track Name: Genetic Algorithms

Categories and Subject Descriptors

G.1.6 [**Numerical Analysis**]: Optimization—*Global optimization, Nonlinear optimization, Unconstrained optimization*; D.2.8 [**Software Engineering**]: Metrics—*performance measures*

General Terms

Performance

Keywords

Performance measures, stochastic algorithms, performance profiles

1. INTRODUCTION

Stochastic algorithms are often used in the optimization field. Due to their stochasticity, the performance analysis and the comparison between algorithms are not straightforward. Therefore, a relevant issue is the analysis of the behavior of these algorithms when solving distinct classes of optimization problems. Consequently, this implies the design of experiments and the use of statistical techniques to make fair comparisons between algorithms. Note that for stochastic algorithms, two goals need to be attained: (i) to achieve a good approximation to the optimum - accuracy and (ii) to reduce the variability of the solutions produced - precision. From an optimization point of view, it is desired to maximize the probability of obtaining good solutions and to minimize the variability of the solutions.

Therefore, assessing the quality of a stochastic algorithm commonly implies a large number of experimental comparisons with other stochastic or even deterministic algorithms. Such comparisons always assume the use of performance metrics. Because of their stochastic nature, they are also statistical, and their computation requires experiments to be conducted in order to obtain sufficient performance data.

On the other hand, the "No Free Lunch" theorem states that it is not possible to find a single algorithm that behaves better for all the problems [12]. Therefore, the comparative study and the identification of trade-offs between the algorithms performance, in terms of accuracy and precison, are of great interest. This study can assist the decision of selecting the best algorithms for each particular problem.

In the context of the comparison of deterministic optimization algorithms, Dolan and Moré [4] proposed a method based on performance profiles to compare solvers. This method provides graphical representations of the distribution of performance measures over a problem set. However, as far as we know, with the exception of the work of Barreto *et al.* [2], the application of the performance profiles to stochastic algorithms has not been studied, although it has been applied without foreseeing its full implications.

In recent years, there has been a growing interest on the use of statistical techniques in the analysis of stochastic algorithms. There are some authors that have applied parametric statistical tests to compare the performance of algorithms [3, 11]. Recently, García *et al.* [5] have studied the application of parametric and non-parametric tests to the comparison of evolutionary algorithms. They suggest and show the advantages of using non-parametric tests for the performance comparison of evolutionary algorithms.

In this paper, we intend to extend and study the applicability of performance profiles to the context of stochas-

GECCO'11, July 12–16, 2011, Dublin, Ireland.

tic algorithms. Moreover, we propose a measure assisted by graphical representation to identify the compromises between accuracy and precision of stochastic algorithms. The emphasis of the paper is on the benchmarking process in the context of stochastic algorithms, based on the concepts of performance profiles.

The paper is structured as follows: in section 2 several performance measures are presented; section 3 presents the application of the performance measures to the results obtained with two stochastic algorithms; and in section 4, some conclusions and future work are discussed.

2. PERFORMANCE MEASURES

In stochastic algorithms, the solutions over a given number of independent runs show a certain variability. Typically, for test problems, it is possible to perform multiple runs and select the best solution obtained. Moreover, the optimal solution of the problem is often known *a priori*. This knowledge can be seen as a reference target and used in the assessment of the performance of the algorithms.

For those problems where the optimal solution is known (this is the case of most test problems), it is possible to define a success criterion based on the distance to the optimum, in terms of the objective function value. A run is considered successful if the solution obtained is within a neighborhood defined by a given tolerance. The successful rate is computed as the percentage of runs terminating with success. It should be noted that this measure can only be computed if the optimal solution is known. Moreover, this measure has several drawbacks since it depends on the tolerance defined and does not include any information regarding the distribution of the obtained solutions. Clearly, it is convenient to take the randomness associated to these solutions into account. Thus, statistical procedures must be adopted.

2.1 Statistical analysis

The data obtained from several independent runs of a stochastic algorithm applied to an optimization problem allows the computation of several statistics. The mean best solution measure is defined as the average of the objective function values of the solutions obtained over all runs. It is also possible to compute other statistics such as the median, the first and third quartiles, the best and the worst solutions. Diversity measures can also be computed such as standard deviation and interquartile range. These measures do not require the knowledge of the optimal solution. Graphical representations such as boxplots or histograms can help the visualization of the distribution of the results over all the runs.

Therefore, algorithm comparison implies the use of specific statistic tests to establish when the differences on the algorithms performance are truly statistically significant. Parametric tests can be used such as the t-test for independent samples when two algorithms are compared or analysis of variance when three or more algorithms are being compared. The applicability of these tests is subject to certain conditions namely, independence of runs, normality of data and homocedasticity. In general, the distribution of the results does not satisfy these requirements. Furthermore, for most real world problems, the evaluation of the objective function and constraints may imply high computational times. In this situation, due to limited time resources, it is not possible to perform a large number of runs in order to obtain a "representative" sample, so non-parametric tests are preferable [5].

2.2 Performance profiles

The statistical analysis of algorithms performance for multiple problems is more difficult. Note that optimization problems have different features that influence the results. However, it is possible to compute the cumulative distribution of some measures. This is the case of the performance profiles [4] that were proposed to compare the performance of deterministic algorithms over a set of distinct optimization problems. These performance profiles can be extended to the context of stochastic algorithms with some adaptations. A brief description of the performance profiles follows.

Let $\mathcal{P}$ and $\mathcal{S}$ be the set of problems and the set of solvers in comparison, respectively, and let $m_{p,s}$ be the performance metric required to solve problem $p \in \mathcal{P}$ by solver $s \in \mathcal{S}$. The comparison is based in performance ratios defined by

$$r_{p,s} = \frac{m_{p,s}}{\min\{m_{p,s} : s \in \mathcal{S}\}}$$

and the overall assessment of the performance of a particular solver s is given by

$$\rho_s(\tau) = \frac{1}{\text{total number of problems}}\{\text{size}\{p \in \mathcal{P} : r_{p,s} \leq \tau\}\}.$$

For $\tau = 1$, $\rho_s(\tau)$ gives the probability that the solver s will win over the others in the set. Thus, for $\tau = 1$, the uppermost curve shows the algorithm with the highest percentage of problems with the best metric value. However, for large values of τ, the $\rho_s(\tau)$ measures the solver robustness. Overall, the highest the ρ_s values, the better the solver is. Also, for solver s that performs the best on a problem p, $r_{p,s} = 1$. If $r_{p,s} = 2$, it means that the m-fold improvement by solver s on problem p is twice the best value found by another solver on the same problem p.

Dolan and Moré [4] used the computing time, $t_{p,s}$, required to solve a problem p by a solver s to evaluate the performance of the solvers. They suggested other measures that can be used instead; however, not all the measures have an absolute zero and the performance profiles may loose the original meaning. In order to maintain the same principles, in this paper it is used an m-fold improvement as suggested in [1]. We remark that this measure can give non-positive values, so we used instead a function

$$\mathcal{M}(f) = \left| \frac{f_{\text{stats}} - f^*}{f_{\text{worst}} - f^*} \right|$$

to define a metric, given by

$$m_{p,s}(f) = \begin{cases} \delta & \text{if } \mathcal{M}(f) \leq \delta \\ \mathcal{M}(f) + \delta & \text{otherwise,} \end{cases}$$

being δ a small positive parameter to prevent $m_{p,s}(f) = 0$, since in this case no performance ratios $r_{p,s}$ could be computed. Thus, we guarantee that the original meaning of the Dolan and Moré [4] performance profiles is maintained. f_{stats} represents a statistic computed for objective function values obtained in several runs (e.g., median, 1st quartile, 3rd quartile) and f_{worst} the worst obtained value over the runs. f^* denotes the best known value of the objective function f for the problem under consideration.

The overall performance of algorithms on a set of optimization problems can be assessed by the performance pro-

files. However, it is also desirable to inspect, for each problem, the trade-off between accuracy and precision of the algorithms. The ideal algorithm should have small variance in order to obtain a good approximation to the optimum (accuracy) and minimize the risk of being far from it (precision). In general, the algorithm performance for a given problem corresponds to a compromise between these two goals. Thus, it is crucial to compute measures that represent accuracy *versus* precision. For this purpose, it is possible to compute the difference between the medians and the interquartile range for the measures used in the the performance profiles. These measures are, in general, preferable than mean and standard deviation due to the non symmetry of the distributions. Accuracy and precision measures can be plotted in order to perceive the different compromises.

3. NUMERICAL RESULTS

In this section, previously described performance measures are used to compare the results obtained with two stochastic algorithms on a set of test problems. The emphasis is not on the solvers but on the benchmarking process, so we used two commercial available solvers. The set of test problems was mostly based on a collection of problems, arriving from quite different contexts.

3.1 Stochastic Algorithms

In the experiments, two stochastic algorithms have been used: genetic algorithms and simulated annealing. We have chosen the commercial implementations of these algorithms by MatLab (MatLab is a registered trademark of the MathWorks, Inc.): the `ga` and `simulannealbnd` commands of the global optimization toolbox, version 3.0. The goal is to have a set of results to test the performance measures. The two solvers were used with the default options without any experimental work in order to fine tune the algorithms parameters. Therefore, results may be different if the parameters are changed or if different versions of these solvers are used. The maximum number of objective function evaluations was set to 1,200 for both algorithms. The initial population of the genetic algorithm is generated at random. Simulated annealing starts the search from an initial guess of the optimum. Next, a brief description of these two stochastic algorithms is provided.

None of the algorithms imposes any condition to the continuity or convexity of the search space and both require only information on the objective function and constraints, and no derivative or other auxiliary knowledge is necessary. There are some important differences between these algorithms.

Genetic algorithms (GAs) are population based algorithms with search procedures that mimic the natural evolution of the species in the natural systems [6].

Simulated Annealing (SA), a combinatorial optimization algorithm first proposed by Kirkpatric *et al.* [8], was inspired on statistical mechanics.The annealing scheme refers to the sequence of temperatures and rearrangements until an equilibrium is reached at a given temperature [7]. The cooling mechanism is used as an analogy for optimization where, for instance, in a minimization problem, slight uphill movements are allowed.

3.2 Test problems

We use a set of benchmark minimax problems to test the performance of the two stochastic algorithms. This type of problems appears in many engineering areas, such as optimal control, engineering design, discrete optimization, Chebyshev approximation and game theory applications. The general form of a minimax problem is

$$\underset{x\in\mathbb{R}^n}{\text{minimize}}\, f(x),$$

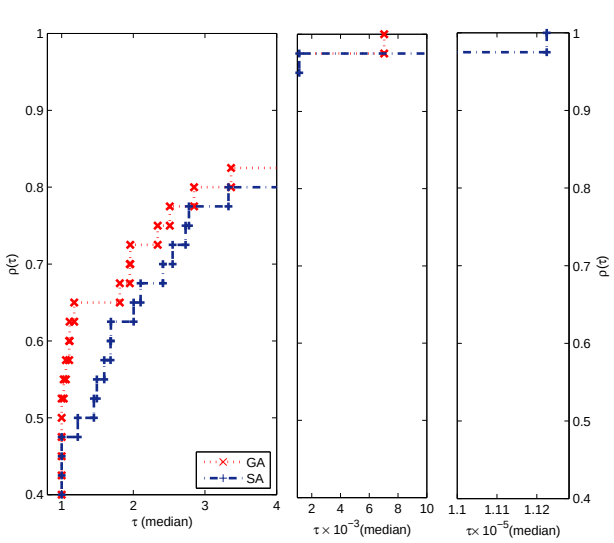
where $f(x) = \max F_j(x)$, $F_j : \mathbb{R}^n \to \mathbb{R}$, $j = 1, \ldots, m$ are continuously differentiable functions. Some of these problems are described in full detail in [9], and others in [10]. The characteristics of the problems are summarized in Table 1 that lists the name of the problem, the number of functions of the minimax problem (m), the number of decision variables (n) and the known optimum value. The problems were coded in MatLab and can be obtained from the corresponding author. For each problem, 100 independent runs of each algorithm were performed.

Table 1: Test problems

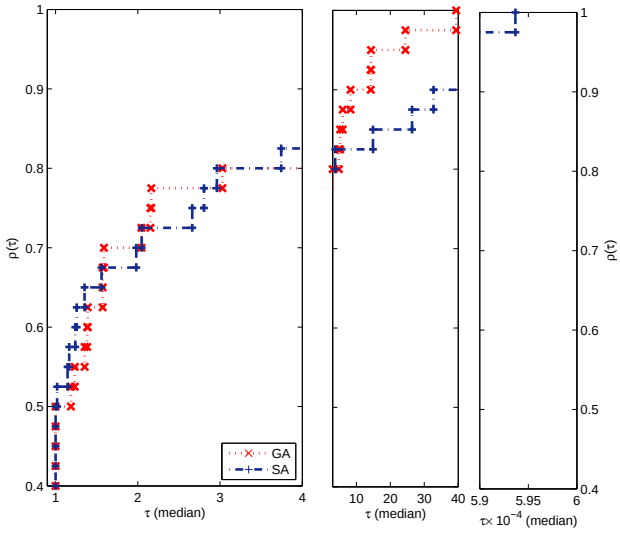
No	Problem	m	n	Optimum value
1	CB2	3	2	1.9522245
2	Rosen-Suzuki	4	4	-44
3	S Xu	5	7	247
4	H-P Schwefel	2	2	0
5	Maxl	20	20	0
6	Spiral	2	2	0
7	OET6	21	4	$0.20160753 \times 10^{-2}$
8	Crescent	2	2	0
9	DEM	2	2	-3
10	QL	3	2	7.2
11	CB3	3	2	2
12	LQ	2	2	-1.4142136
13	MXHILB	50	50	0
14	WF	3	2	0
15	EVD52	6	3	3.5997193
16	Davidon 2	20	4	115.70644
17	OET5	21	4	0.26359735^{-2}
18	Polak 1	2	2	2.7182818
19	Hald-Madsen 1	2	2	0
20	Wong 1	5	7	680.63006
21	Watson	31	20	$0.14743027 \times 10^{-7}$
22	Polak 3	10	11	3.8872
23	Polak 6	4	4	-44
24	PBC3	21	3	$0.42021427 \times 10^{-2}$
25	Bard	15	3	$0.50816327 \times 10^{-1}$
26	Kowalik-Osborn	11	4	$0.80843684 \times 10^{-2}$
27	GAMMA	61	4	$0.12041887 \times 10^{-6}$
28	EXP	21	5	$0.12237125 \times 10^{-3}$
29	PBC1	30	5	$0.22340496 \times 10^{-1}$
30	EVD61	51	6	$0.3490504926 \times 10^{-1}$
31	Transformer	11	6	0.19729063
32	Filter	41	9	$0.61852848 \times 10^{-2}$
33	Wong 2	9	10	24.306209
34	Wong 3	18	20	133.72828
35	Polak 2	2	10	54.59815
36	Osborne 2	65	11	$0.48027401 \times 10^{-1}$
37	Shor	10	5	22.600162
38	Maxquad	5	10	-0.8414083
39	Gill	3	10	9.7857721
40	No. of active faces	21	20	0

3.3 Comparison using performance profiles

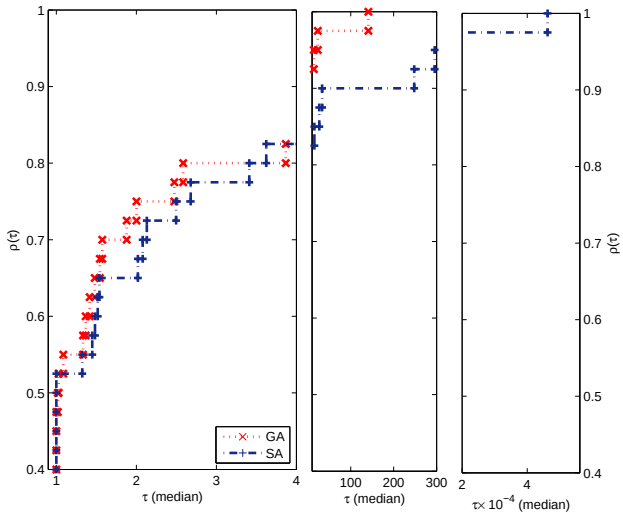
To compare the overall performance of the algorithms, performance profiles, as described in previous section, were used. Figures 1 and 2 show the performance profiles on the median and on the minimum (best solution), respectively. In order to investigate the effect of the number of runs performed in the shape of the performance profiles, we depict the profiles for 10, 30, 80 and 100 independent runs (Figures 1(a), 1(b), 1(c), 1(d), 2(a), 2(b), 2(c) and 2(d)). As



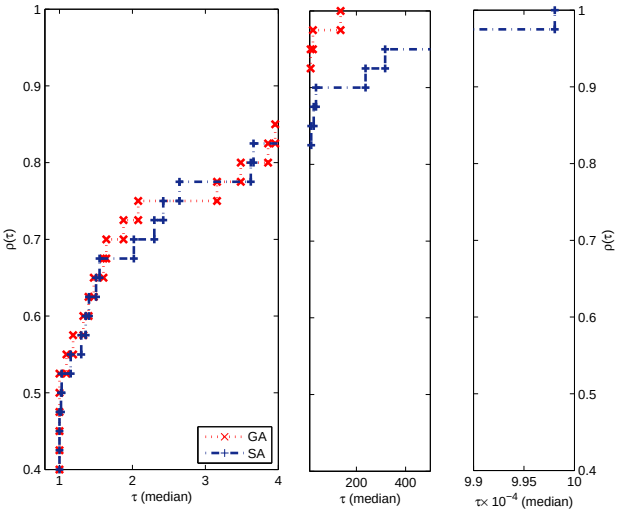
(a) 10 runs



(b) 30 runs



(c) 80 runs



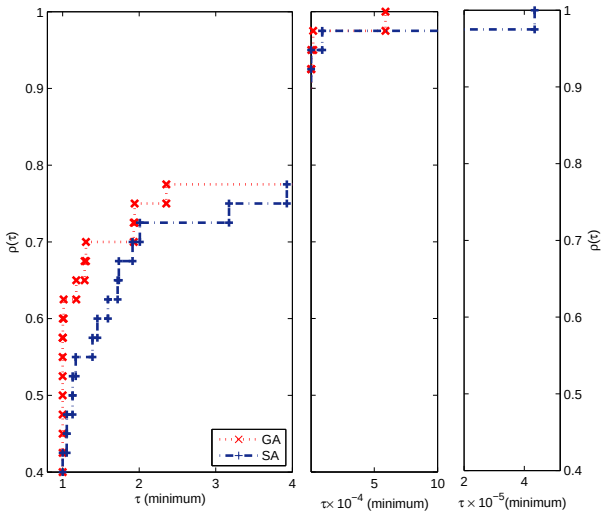
(d) 100 runs

Figure 1: Profiles for median.

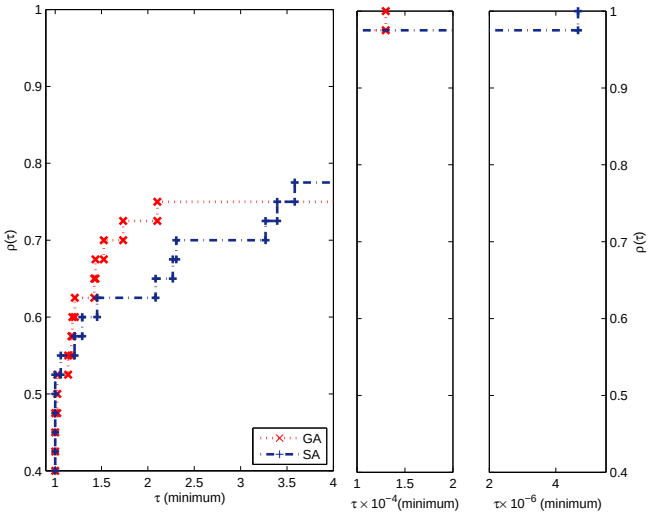
expected, it is clear that the number of runs influences the shape of the performance profiles. For instance, according to Figures 1(a) and 2(a) for $\tau = 1$, it seems that GA achieves the best approximation to the optimum in a large majority of the problems (60% and 65%, respectively). With SA this value is less than 50% in both cases. Figures 1(b) to 1(d) contradict this statement. In fact, the algorithms performance is very similar as it can be seen in Figures 1(c) and 1(d) (both algorithms achieved a similar performance on about 50% of the problems). Also, Figure 2(d) seems to indicate a larger difference between the algorithms performance when compared with Figures 2(b) and 2(c). It should be noted that, for $\tau = 1$ (please refer to Figure 2(a)), GA solves 65% of the problems with the closest approximation to the optimum. Recall, that in the performance profiles, in $\tau = 1$, $\rho_s(1)$ gives the probability that the solver will win over all the others. This is an important issue and, as can be observed, an insufficient number of runs can lead to misleading conclusions.

Additional information can be obtained for other values of τ. For instance, for $\tau = 2$, it can be observed that SA solves more than 65% of the problems and GA about 70% of the problems (Figures 1(b), 1(c), 1(d), 2(b), 2(c) and 2(d)). These observations highlight the influence of the number of runs in the performance profiles. It is clear that a small number of runs can compromise the analysis and the conclusions of the comparison.

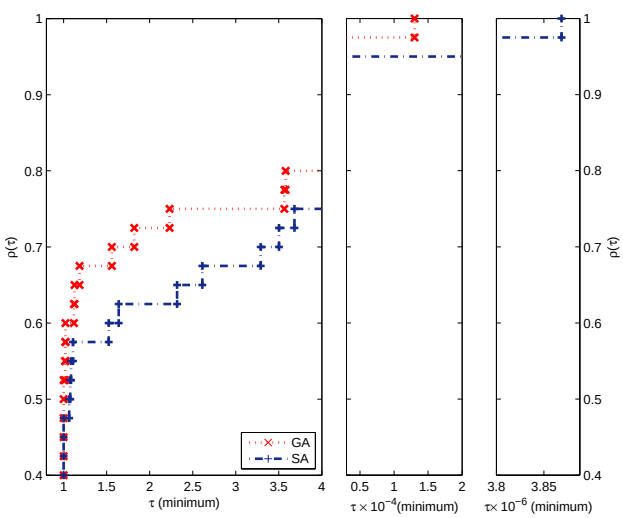
The robustness of algorithms can be assessed for large



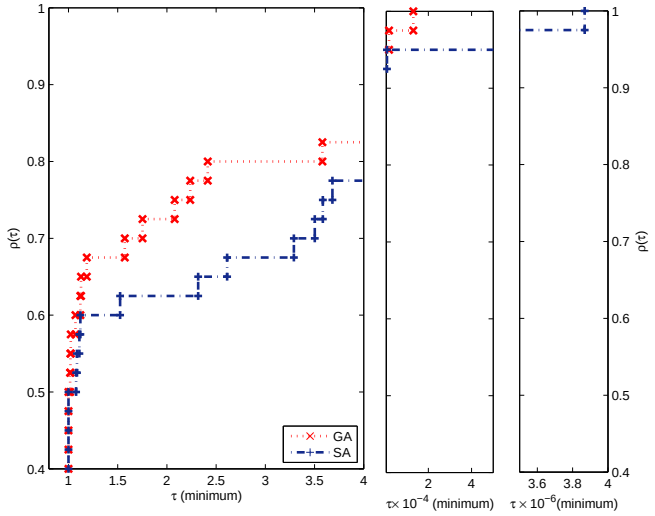
(a) 10 runs.



(b) 30 runs.



(c) 80 runs.



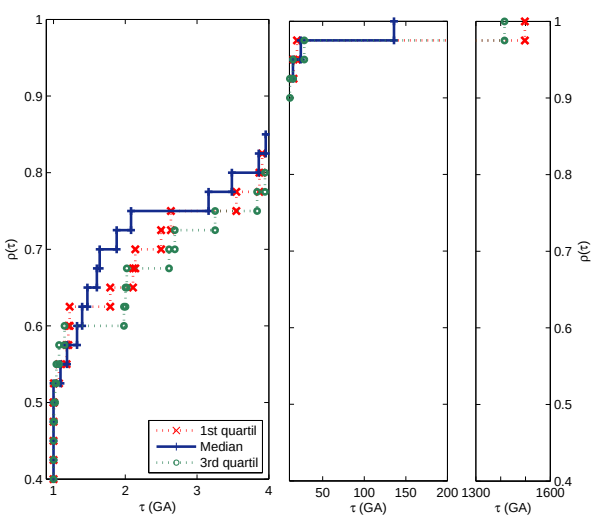
(d) 100 runs.

Figure 2: Profiles for minimum.

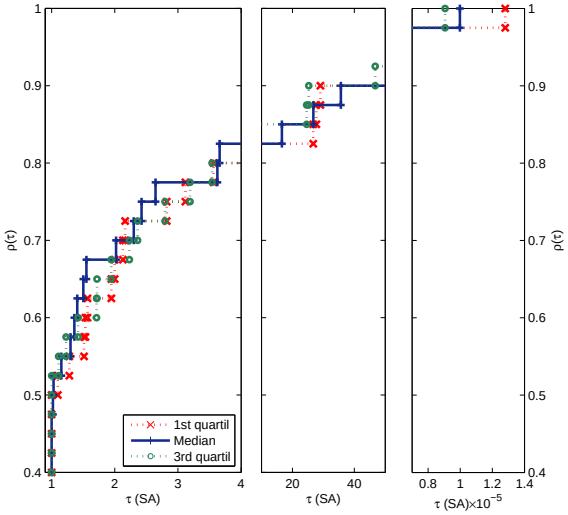
values of τ. For 10 runs (Figures 1(a) and 2(a)), it can be observed that GA solves all problems for $\tau \approx 6,000$ and $\tau \approx 50,000$, respectively. On the other hand, SA solves all problems for $\tau \approx 112,000$ and $\tau \approx 400,000$, respectively. This relation is consistent for 30, 80 and 100 runs. In Figure 1 the values of τ where GA and SA solve all the problems are in the order of 10^1 and 10^4, respectively, and in Figure 2 these values are in the order of 10^4 and 10^6.

In Figures 1 and 2, we analyze the performance from two distinct points of view. With test problems, it may be interesting to analyze the performance of an algorithm based on the best solution found over all runs ("peak" performance). In this sense, an algorithm is said to be "better" than other if it found the best approximation to the optimal solution over all the runs. This idea contrasts with an analysis based on location measures. For instance, based on Figure 2(d), we can conclude that GA outperforms SA; on the contrary, Figure 1(d), based on the median, shows that the two algorithms are similar in performance. Therefore, performance based on extreme values should be read with care, if not at all avoided.

This analysis should also include the information regarding the variability. For real world problems, with computationally expensive objective evaluations, it is more important to have an algorithm that exhibits a higher average performance and small variability, than an algorithm that has a "high peak" performance. It is crucial to have an algorithm that carries the lowest risk of missing a "suitable solution" due to the reduced number of runs that can be performed (in most cases, a single run).



(a) Genetic Algorithm.



(b) Simulated Annealing.

Figure 3: Profiles for Stochastic Algorithms.

Figure 3 shows the performance profiles for first, second and third quartiles for each algorithm. It can be observed that the curves for these metrics cross each other. These can be explained by the different variability of the distributions of the results for each problem, reinforcing the care with which performance profiles must be read in the case of stochastic algorithms.

In Figure 4(a), we inspect, for each problem, the trade-off between accuracy and precision. For this purpose, the difference between the medians and the interquartile range for the measures used in the the performance profiles are plotted for each problem. In general, the algorithms performance for a given problem corresponds to a compromise between these two goals. The quadrants of the graph define regions that allow the comparison of the results in terms of accuracy and precision as follows:

- 1st quadrant - SA is more accurate and more precise than GA;
- 2nd quadrant - SA is less accurate and more precise than GA;
- 3rd quadrant - SA is less accurate and less precise than GA;
- 4th quadrant - SA is more accurate and less precise than GA.

For any problem that belongs to the 3rd quadrant, GA is preferable to SA because GA is better in terms of accuracy and precision than SA. Conversely, for any problem in the 1st quadrant, SA is preferable to GA because SA is better in terms of accuracy and precision than GA. The 2nd and 4th quadrants define regions of indifference. Thus, problems that belong to these quadrants can be solved by GA or SA with different compromises between accuracy and precision, according to the specificity of the problem at hand.

For illustrative purposes, and to reinforce the results plotted in Figure 4(a), we show the boxplots for some of the problems. Figure 4(b) shows the boxplot of the distribution of the objective function values for problem 16. The median of the results of GA is greater than the median of the results of SA, i.e., GA is less accurate than SA. In terms of variability, it can be observed the larger interquartile range of the results from SA when compared with the results from GA.

The boxplot of the distribution of the objective function values for problem 24 is shown in Figure 4(c). Here, the median of the results of GA is clearly lower than the median of the results of SA, i.e., GA is more accurate than SA. In terms of variability, it can also be seen the larger interquartile range of the results from SA when compared with the results from GA.

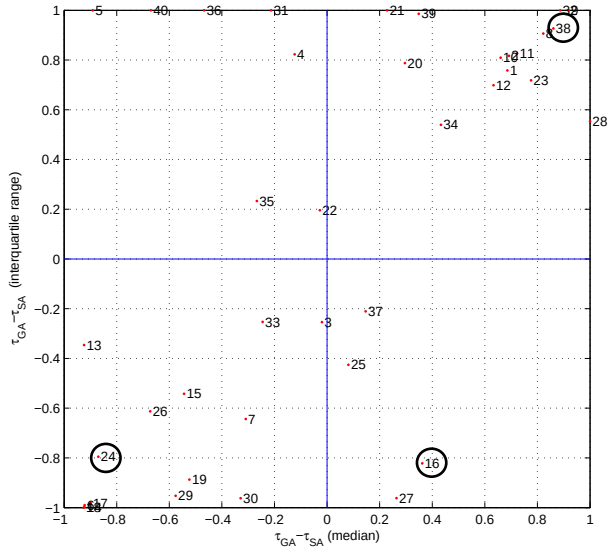
Finally, Figure 4(d) is the boxplot of the distribution of the objective function values for problem 38. Here, the results of SA are more accurate and precise than the results from GA.

The results show that, in the context of stochastic algorithms, the choice of the performance measure in building up the performance profiles must be thought with caution.

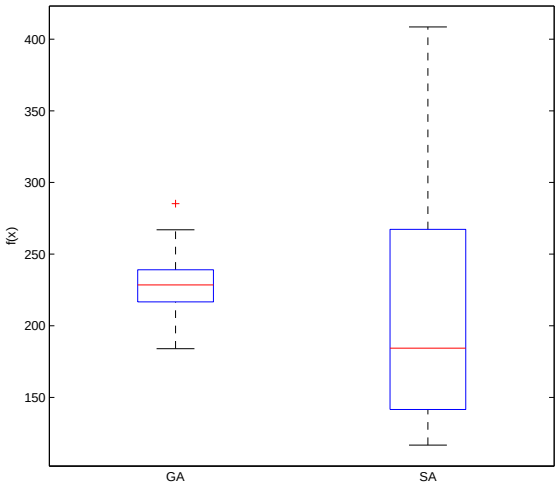
4. CONCLUSIONS

In this work, we show that the number of runs influences the shape of the performance profiles. The stability of the profiles seem to be achieved for a number of runs over 30, becoming stable for an even larger number of runs.

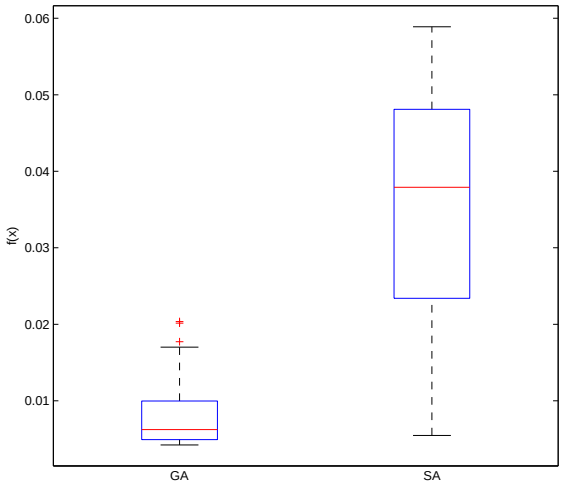
The use of performance measures based on the best solutions is not sustainable, given the stochastic nature of the algorithms and the non-symmetric distribution of the results with the presence of very large extreme values. Performance comparison must include some kind of measure in terms of of accuracy and precision, allowing the enhancement of the bi-objective nature of the assessment.



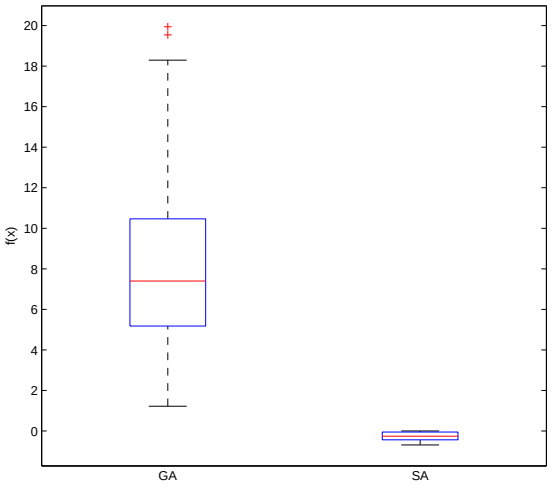
(a) Accuracy versus precision.



(b) Boxplot for problem 16.



(c) Boxplot for problem 24.



(d) Boxplot for problem 38.

Figure 4: Performance analysis by problem.

Future work will include the development of statistical comparisons taking into consideration the dependence on performance measures and also on the set of problems.

5. ACKNOWLEDGMENTS

The authors would like to thank FCT - Fundação para a Ciência e a Tecnologia (Portuguese Foundation for Science and Technology) that supported in part this work.

6. REFERENCES

[1] M. M. Ali, C. Khompatraporn, and Z. B. Zabinsky. A numerical evaluation of several stochastic algorithms on selected continuous global optimization test problems. *J. of Global Optimization*, 31:635–672, April 2005.

[2] A. M. Barreto, H. S. Bernardino, and H. J. Barbosa. Probabilistic performance profiles for the experimental evaluation of stochastic algorithms. In *Proceedings of the 12th annual conference on Genetic and evolutionary computation*, GECCO '10, pages 751–758, New York, NY, USA, 2010. ACM.

[3] A. Czarn, C. MacNish, K. Vijayan, R. Turlach, and R. Gupta. Statistical exploratory analysis of genetic algorithms. *IEEE Trans. Evol. Comput.*, 8(4):405–421, 2004.

[4] E. Dolan and J. Moré. Benchmarking optimization software with performance profiles. *Math. Programming*, 91(2):201–213, 2002.

[5] S. García, D. Molina, M. Lozano, and F. Herrera. A study on the use of non-parametric tests for analyzing the evolutionary algorithms' behaviour: a case study on the cec'2005 special session on real parameter optimization. *Journal of Heuristics*, 15:617–644, 2009.

[6] D. Goldberg. *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley, Reading, Massachusetts, 1989.

[7] L. Ingber. Adaptive simulated annealing (asa): Lessons learned. *Control and Cybernetics*, 25(1):33–54, 1996.

[8] S. Kirkpatrick, C. D. Gelatt, Jr, and M. P. Vecchi. Optimization by Simulated Annealing. *Science*, 220:671–680, 1983.

[9] L. Lukšan and J. Vlček. Test problems for nonsmooth unconstrained and linearly constrained optimization. Technical report, TR 798, ICS, Academy of Science of the Czech Republic, January, 2000.

[10] Y. Petalas, K. Parsopoulos, and M. Vrahatis. Memetic particle swarm optimization. *Annals of Operations Research*, 156:99–127, 2007.

[11] I. Rojas, J. González, H. Pomares, J. Merelo, P. Castillo, and G. Romero. Statistical analysis of the main parameters involved in the design of a genetic algorithm. *IEEE Trans. Syst. Man Cybern. Part C*, 32(1):31–37, 2002.

[12] D. Wolpert and W. Macready. No free lunch theorems for optimization. *IEEE Trans. Evol. Comput.*, 1(1):67–82, 1997.