Empirical Scaling Analyser: An Automated System for Empirical Analysis of Performance Scaling

Zongxu Mu
Department of Computer Science
University of British Columbia
zongxumu@cs.ubc.ca

Holger H. Hoos
Department of Computer Science
University of British Columbia
hoos@cs.ubc.ca

ABSTRACT

The time complexity of problems and algorithms, i.e., the scaling of the time required for solving a problem instance as a function of instance size, is of key interest in theoretical computer science and practical applications. This paper presents an automated tool – Empirical Scaling Analyser (ESA) – that is designed to perform empirical scaling analysis. The methodological approach underlying ESA was introduced by Hoos [1], then applied to analysing a complete TSP solver [2] and later extended and applied to analysing several prominent SAT solvers [3]. ESA is broadly applicable to analysing different kinds of algorithms as long as running time data can be collected on sets of problem instances of various sizes. It is particularly well suited for the analysis of the empirical time complexity of evolutionary algorithms and other heuristic procedures for solving $\mathcal{NP}$-hard problems.

Categories and Subject Descriptors

F.2 [**Theory of Computation**]: Analysis of Algorithms and Problem Complexity

Keywords

Empirical analysis; scaling analysis

1. INTRODUCTION & METHODOLOGY

In theoretical computer science, time complexity is arguably the most important aspect in analysing and understanding problems and algorithms. The time complexity of an algorithm describes the time required for solving a problem instance as a function of instance size and is traditionally studied mostly by means of theoretical analysis. For instance, the Boolean satisfiability problem (SAT) and the travelling salesman problem (TSP) are two $\mathcal{NP}$-hard problems, for which the best algorithms have exponential time-complexity in the worst case. Empirical analysis of time complexity has seen increasing interest, because it permits predicting the running times of high-performance algorithms in practice and may also provide useful insights into their behaviour.

A significant advance in the methodology for studying the empirical scaling of algorithm performance with input size was realised by Hoos [1]. His method uses standard numerical optimisation approaches to automatically fit scaling models, which are then challenged by extrapolation to larger input sizes. Most importantly, it uses a bootstrap resampling approach to assess the models and their predictions in a statistically meaningful way. Hoos & Stützle used this approach to characterise the scaling behaviour of Concorde, a state-of-the-art complete algorithm for the travelling salesperson problem (TSP), demonstrating that the scaling of Concorde's performance on 2D Euclidean TSP instances with n cities agrees well with a square-root-exponential model of the form $a \cdot b^{\sqrt{n}}$ [2].

Recently, we applied the same empirical methodology to study the scaling behaviour of prominent high-performance SAT solvers [3]. Moreover, we proposed a useful extension to the methodology: Noting that observed running time statistics are also based on measurements on sets of instances, we calculate bootstrap percentile intervals for those, in addition to the point estimates used in the original approach. This way, we capture dependency of these statistics on the underlying sample of instances.

In this work, we present an automated tool – the Empirical Scaling Analyser (ESA) – that can perform the analysis described above. To perform the scaling analysis, ESA needs input data containing the sizes of the instances studied and the running times a given algorithm requires for solving these instances. From further information about the algorithm and instance distributions and a given LaTeX template, ESA automatically generates a technical report with detailed empirical scaling analysis results and interpretation. This report contains tables and figures that users can easily read (see Section 2 for examples). ESA is not limited to fitting and assessing a single scaling model, but can deal with multiple models simultaneously.

2. EXAMPLE USE CASE & OUTPUT

In this section, we use the scaling of the median running time of WalkSAT/SKC [4] as an example to illustrate the use of ESA. WalkSAT/SKC is one of the most widely known high-performance SAT solvers based on stochastic local search (SLS). The target problem instances we consider are uniform random 3-SAT instances from the phase transition region, which are widely believed to represent one of the most challenging benchmarks for SAT.

GECCO '15 July 11-15, 2015, Madrid, Spain

ACM ISBN 978-1-4503-3488-4/15/07.
DOI: http://dx.doi.org/10.1145/2739482.2764898

Solver	Model	n	Predicted confidence intervals: Poly. model	Predicted confidence intervals: Exp. model	Observed median running time (sec): Point estimates	Observed median running time (sec): Confidence intervals
WalkSAT/SKC	Poly$[a, b]$ $a \in [2.58600 \times 10^{-12}, 8.63869 \times 10^{-10}]$ $b \in [2.80816, 3.76751]$	600	$[\mathbf{0.054}, \mathbf{0.081}]$	$[0.067, 0.104]$	0.056	$[0.050, 0.070]$
		700	$[\mathbf{0.083}, \mathbf{0.146}]$*	$[0.137, 0.264]$	0.121	$[0.105, 0.145]$
		800	$[\mathbf{0.122}, \mathbf{0.238}]$*	$[0.277, 0.664]$	0.180	$[0.132, 0.209]$
		900	$[\mathbf{0.170}, \mathbf{0.373}]$*	$[0.565, 1.676]$	0.267	$[0.222, 0.323]$
		1000	$[\mathbf{0.229}, \mathbf{0.557}]$*	$[1.151, 4.200]$	0.385	$[0.327, 0.461]$

Table 3: 95% bootstrap confidence intervals for median running time predictions and observed running times on random 3-SAT instances. The instance sizes shown here are larger than those used for fitting the models. Bootstrap intervals on predictions that agree with the observed point estimates are shown in boldface, and those that fully contain the confidence intervals on observations are marked with asterisks (*).

n	200	250	300	350	400	450	500
# Instances	601	589	633	558	579	572	578
mean	0.0065	0.0167	0.0479	0.0743	0.2162	0.2634	2.1713
coefficient of variation	1.9323	2.7076	7.1479	4.6358	8.1654	6.2329	17.9680
Q(0.1)	0.0006	0.0011	0.0016	0.0022	0.0035	0.0050	0.0066
Q(0.25)	0.0010	0.0019	0.0032	0.0043	0.0076	0.0101	0.0144
median	0.0021	0.0045	0.0075	0.0109	0.0182	0.0241	0.0365
Q(0.75)	0.0057	0.0121	0.0210	0.0298	0.0536	0.0867	0.1292
Q(0.9)	0.0157	0.0364	0.0599	0.0891	0.2392	0.3534	0.4375

n	600	700	800	900	1000
# Instances	572	636	584	592	593
mean	2.5027	3.3031	2.7717	15.5353	30.1594
coefficient of variation	13.3185	7.8551	5.1294	6.3333	5.4317
Q(0.1)	0.0124	0.0184	0.0268	0.0359	0.0540
Q(0.25)	0.0240	0.0395	0.0550	0.0801	0.1190
median	0.0564	0.1083	0.1797	0.2668	0.3845
Q(0.75)	0.2014	0.4775	0.7455	1.3348	1.8264
Q(0.9)	1.0791	2.0195	3.3366	8.3035	14.4725

Table 1: Details of the support data for model fitting and the challenge data.

		Model	RMSE (support)	RMSE (challenge)
WalkSAT/SKC	Exp. Model	$6.89157 \times 10^{-4} \times 1.00798^{n}$	0.0008564	0.7600
	Poly. Model	$\mathbf{8.83962 \times 10^{-11} \times n^{3.18915}}$	**0.0007433**	**0.03142**

Table 2: Fitted models of median running times and RMSE values (in CPU sec). Model parameters are shown with 6 significant digits, and RMSEs with 4 significant digits; the models yielding more accurate predictions (as per RMSEs on challenge data) are shown in boldface.

After collecting running times of WalkSAT/SKC on these 3-SAT instances, we fed the data into ESA, which then generated a report containing the details of the dataset and the analysis results. Table 1 provides the details of the dataset, as generated by ESA based on the input data. Table 2 contains the fitted models and their corresponding RMSEs, and Table 3 shows bootstrap intervals of the model parameters and predicted running times. The fitted models and the bootstrap intervals of the predicted running times are also illustrated in Figure 1, which clearly shows which models fit the given data well. From these results, ESA automatically determines that a polynomial model describes our data well, whereas an exponential model is not consistent with the given data. In other words, the analysis yields the surprising result that the median running times of WalkSAT/SKC scale polynomially with instance size. Using ESA, empirical scaling results like this are easy to obtain, for arbitary algorithms and problem instance distributions.

3. CONCLUSION

In this work, we presented Empirical Scaling Analyser (ESA), an automated tool that performs empirical scaling analysis on the running time of an algorithm. The underlying

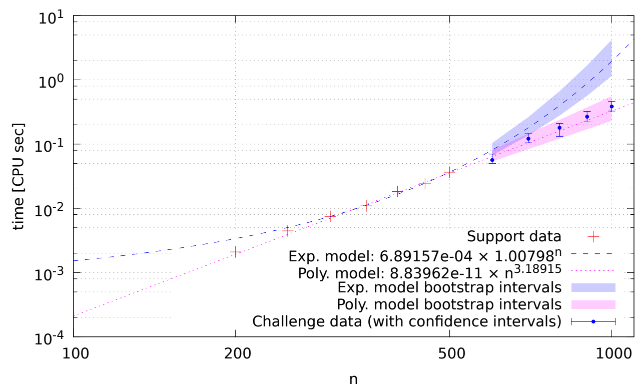

Figure 1: Fitted models of the median running times of WalkSAT/SKC. Both models are fitted with the median running times of WalkSAT/SKC solving the SAT instances from the set of 1200 random instances of 200, 250, ..., 500 variables, and are challenged by median running times of 600, 700, ..., 1000 variables.

methodology was first proposed by Hoos [1] and applied to analysing prominent algorithms for the TSP [2] and SAT [3]. ESA makes it possible to carry out such analyses quickly, efficiently and automatically. We believe that the results thus obtained are useful and interesting for research and practical applications concerning state-of-the-art algorithms for difficult computational problems. ESA is especially useful for the analysis of evolutionary algorithms and other heuristic procedures for solving $\mathcal{NP}$-hard problems. A web-based version of ESA is available at http://www.cs.ubc.ca/labs/beta/Projects/ESA/esa-online.html.

4. REFERENCES

[1] H. H. Hoos. A bootstrap approach to analysing the scaling of empirical run-time data with problem size. Technical Report TR-2009-16, Dept. of Computer Science, University of British Columbia, 2009.

[2] H. H. Hoos and T. Stützle. On the empirical scaling of run-time for finding optimal solutions to the travelling salesman problem. *European Journal of Operational Research*, 238(1):87–94, 2014.

[3] Z. Mu and H. H. Hoos. On the empirical time complexity of random 3-SAT at the phase transition. To appear in *Proceedings of IJCAI*, 2015.

[4] B. Selman, H. A. Kautz, and B. Cohen. Noise strategies for improving local search. In *Proceedings of AAAI'94*, pp. 337–343, 1994.