

# On the empirical scaling of running time of Concorde-Find+Prove for solving RUE

Empirical Scaling Analyser

14th July 2017

## 1 Introduction

This is the automatically generated report on the empirical scaling of the running time of Concorde-Find+Prove for solving RUE.

## 2 Methodology

For our scaling analysis, we considered the following parametric models:

- $Poly[a, b](n) = a \times x^b$  (2-parameter Poly)
- $Exp[a, b](n) = a \times b^x$  (2-parameter Exp)
- $RootExp[a, b](n) = a \times b^{\sqrt{x}}$  (2-parameter RootExp)
- $ExpLogRoot[a, b](n) = a \times b^{\log(x) \cdot \sqrt{x}}$  (2-parameter ExpLogRoot)
- $RootExp1[a](n) = a \times 1.2281^{\sqrt{x}}$  (1-parameter RootExp1)
- $RootExpU1[a](n) = a \times 1.2526^{\sqrt{x}}$  (1-parameter RootExpU1)

Note that the approach could be easily extended to other scaling models. For fitting parametric scaling models to observed data, we used the non-linear least-squares Levenberg-Marquardt algorithm.

Models were fitted to performance observations in the form of Q75s of the distributions of running times over sets of instances for given  $n$ , the instance size. To assess the fit of a given scaling model to observed data, we used root-mean-square error (RMSE).

Due to the instances for which the running times are unknown, there is uncertainty about the precise location of the Q75s of the running time distributions at each such  $n$ , and we can only provide bounds on those Q75s instead. Closely following [1], we calculate these bounds based on the best and worst case scenarios, in which all instances with unknown running times are easiest or hardest, respectively. We note that these are not confidence intervals, since we can guarantee the actual Q75 running times to lie within them. We also calculate RMSEs and confidence intervals based on these bounds.

Closely following [2, 3], we computed 95% bootstrap confidence intervals for the performance predictions obtained from our scaling models, based on 1000 bootstrap samples per instance set and 1000 automatically fitted variants of each scaling model. In the following, we say that a scaling model is inconsistent with observed data if the bootstrap confidence interval for the observed data is disjoint from the bootstrap confidence interval for predicted running times; we say that a scaling model is consistent with observed data, if the interval for observed medians overlaps with, but is not fully contained

$n$	500	600	700	800
# instances	1000	1000	1000	1000
# running times	1000	1000	1000	1000
mean	29.97	50.98	87.51	143.6
coefficient of variation	1.29	1.267	1.222	1.889
Q(0.1)	3.1	5.89	9.73	14.88
Q(0.25)	7.89	13.41	22.58	33.61
median	18.88	29.02	52.09	76.9
Q(0.75)	38.44	60.72	105.9	159
Q(0.9)	70.76	124.5	202.3	318.8

$n$	900	1000	1100	1200
# instances	1000	1000	1000	1000
# running times	1000	1000	1000	1000
mean	217.6	292	466.9	556.8
coefficient of variation	2.075	1.483	1.733	1.699
Q(0.1)	22.85	32.95	44.44	52.54
Q(0.25)	54.91	72.57	95.71	125.7
median	106.9	157.4	217.8	275.8
Q(0.75)	220.9	332.1	496.2	611.4
Q(0.9)	455	660.9	1033	1238

$n$	1300	1400	1500
# instances	1000	1000	1000
# running times	1000	1000	1000
mean	774.6	1120	1514
coefficient of variation	1.684	3.594	2.372
Q(0.1)	79.4	103.3	140.3
Q(0.25)	165.4	221.5	295.9
median	371.9	499.6	654.1
Q(0.75)	875.6	1152	1567
Q(0.9)	1714	2104	3186

Table 1: Details of the running time dataset used as support data for model fitting.

within the bootstrap confidence interval for predicted running times; we say that a scaling model is strongly consistent with observed data, if the observed median is fully contained within the bootstrap confidence interval for predicted running times. Also, we define residue of a model at a given size as the observed point estimate less the predicted value.

### 3 Dataset Description

The dataset contains running times of the Concorde-Find+Prove algorithm solving 15 sets of instances of different sizes. We split the running times into two categories, support ( $n \leq 1500$ ) and challenge ( $n > 1500$ ). The details of the dataset can be found in Tables 1 and 2.

$n$	2000	2500	3000	3500
# instances	1000	100	100	100
# running times	1000	99	98	97
mean	6760	N/A	N/A	N/A
coefficient of variation	2.4	N/A	N/A	N/A
Q(0.1)	426.9	[790, 822]	$[3.87 \times 10^3, 4226]$	$[5.74 \times 10^3, 6570]$
Q(0.25)	1019	$[2.12 \times 10^3, 2234]$	$[8.74 \times 10^3, 9538]$	$[1.17 \times 10^4, 1.325 \times 10^4]$
median	2508	$[6.89 \times 10^3, 7899]$	$[1.95 \times 10^4, 2.064 \times 10^4]$	$[3.53 \times 10^4, 4.057 \times 10^4]$
Q(0.75)	6576	$[1.97 \times 10^4, 1.999 \times 10^4]$	$[4.71 \times 10^4, 5.145 \times 10^4]$	$[1.07 \times 10^5, 1.171 \times 10^5]$
Q(0.9)	$1.381 \times 10^4$	$[6.59 \times 10^4, 7.173 \times 10^4]$	$[9.07 \times 10^4, 9.745 \times 10^4]$	$[2.35 \times 10^5, 2.766 \times 10^5]$

Table 2: Details of the running time dataset used as challenge data for model fitting.

	Model	RMSE (support)	RMSE (challenge)
Concorde-Find+Prove	Poly. Model	$1.0341 \times 10^{-9} \times x^{3.8323}$	[36260, 41790]
	Exp. Model	$14.324 \times 1.0031^x$	$[3.6513 \times 10^5, 3.7054 \times 10^5]$
	RootExp. Model	$0.32189 \times 1.2449\sqrt{x}$	[9958, 15284]
	ExpLogRoot. Model	$1.7179 \times 1.0243^{\log(x)*\sqrt{x}}$	[37111, 42561]
	RootExp1. Model	<b><math>0.53319 \times 1.2281\sqrt{x}</math></b>	<b>[4625, 9691.8]</b>
	RootExpU1. Model	$0.25613 \times 1.2526\sqrt{x}$	[20098, 25540]

Table 3: Fitted models of the Q75s of the running times and RMSE values (in CPU sec). The models yielding more accurate predictions (as per RMSEs on challenge data) are shown in boldface.

## 4 Empirical Scaling of Solver Performance

We first fitted our parametric scaling models to the Q75s of the running times of Concorde-Find+Prove, as described in Section 2. The models were fitted using the Q75s of the running times for  $500 \leq n \leq 1500$  (support) and later challenged with the Q75s of the running times for  $2000 \leq n \leq 3500$ . This resulted in the models, shown along with RMSEs on support and challenge data, shown in Table 3. In addition, we illustrate the fitted models of Concorde-Find+Prove in Figure 1, and the residues for the models in Figure 2.

But how much confidence should we have in these models? Are the RMSEs small enough that we should accept them? To answer this question, we assessed the fitted models using the bootstrap approach outlined in Section 2. Table 4 shows the bootstrap intervals of the model parameters, and Table 5 contains the bootstrap intervals for the support data. Challenging the models with extrapolation, as shown in Table 6, it is concluded that the Poly model under-estimates the data, the Exp model over-estimates the data, the RootExp model fits the data very well, the ExpLogRoot model tends to over-estimate the data, the RootExp1 model tends to under-estimate the data, and the RootExpU1 model over-estimates the data (as also illustrated in Figure 1).

## 5 Conclusion

In this report, we presented an empirical analysis of the scaling behaviour of Concorde-Find+Prove on RUE. We found the Poly model under-estimates the data, the Exp model over-estimates the data, the RootExp model fits the data very well, the ExpLogRoot model tends to over-estimate the data,

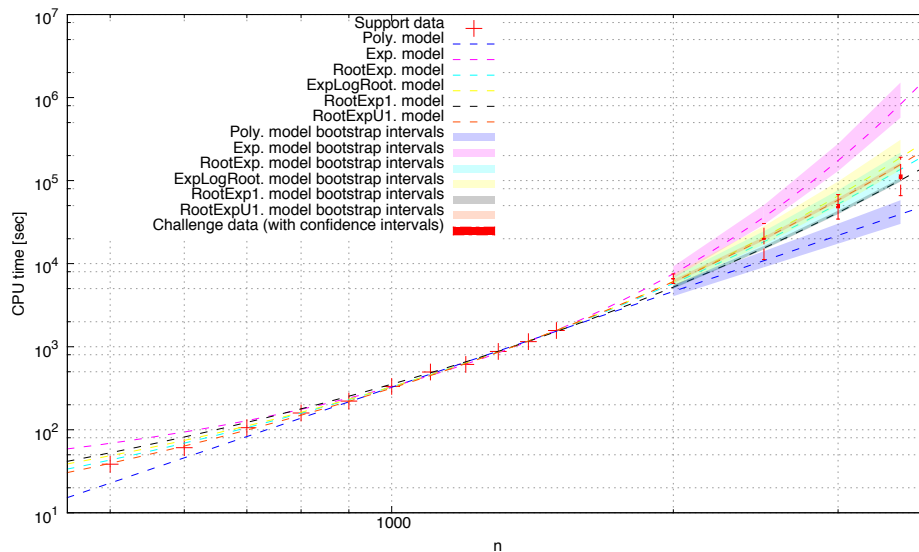


Figure 1: Fitted models of the Q75s of the running times. The models are fitted with the Q75s of the running times of Concorde-Find+Prove solving the set of RUE of  $500 \leq n \leq 1500$  variables, and are challenged by the Q75s of the running times of  $2000 \leq n \leq 3500$  variables.

Solver	Model	Confidence interval of $a$	Confidence interval of $b$
Concorde-Find+Prove	Poly.	$[6.4365 \times 10^{-11}, 6.9389 \times 10^{-9}]$	$[3.5616, 4.2193]$
	Exp.	$[10.312, 17.528]$	$[1.003, 1.0034]$
	RootExp.	$[0.15804, 0.51934]$	$[1.2275, 1.2701]$
	ExpLogRoot.	$[0.99002, 2.4869]$	$[1.0228, 1.0266]$
	RootExp1.	$[0.51083, 0.56279]$	
	RootExpU1.	$[0.24513, 0.2707]$	

Table 4: 95% bootstrap intervals of model parameters for the Q75s of the running times

the RootExp1 model tends to under-estimate the data, and the RootExpU1 model over-estimates the data.

## References

- [1] Jérémie Dubois-Lacoste, Holger H. Hoos, and Thomas Stützle. On the empirical scaling behaviour of state-of-the-art local search algorithms for the euclidean tsp. In *GECCO*. ACM, 2015.
- [2] Holger H Hoos. A bootstrap approach to analysing the scaling of empirical run-time data with problem size. Technical report, Technical Report TR-2009-16, University of British Columbia, 2009.
- [3] Holger H Hoos and Thomas 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.

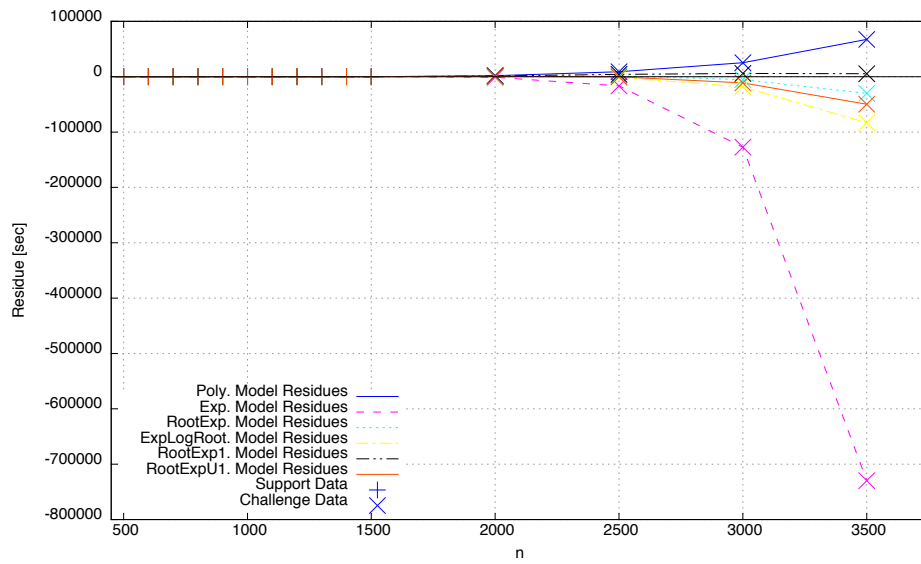


Figure 2: Residues of the fitted models of the Q75s of the running times.

Solver	$n$	Predicted confidence intervals		Observed median run-time	
		Poly. model		Point estimates	Confidence intervals
Concorde-Find+Prove	500	[16.11, 28.96]		38.44	[35.43, 41.87]
	600	[34.65, 55.56]		60.72	[57, 68.06]
	700	[66.28, 96.55]		105.9	[97.44, 113.6]
	800	<b>[116, 155.9]</b>		159	[147, 171.7]
	900	<b>[190.2, 238.1]</b> #		220.9	[207, 242.1]
	1000	<b>[296.2, 348.5]</b> #		332.1	[301.4, 384.4]
	1100	<b>[440.7, 491.9]</b>		496.2	[450.9, 530.8]
	1200	<b>[629.1, 678.6]</b>		611.4	[552.9, 667.3]
	1300	<b>[856.1, 923.6]</b> #		875.6	[782.9, 974.8]
	1500	<b>[1127, 1252]</b> #		1152	[1084, 1268]
1500	<b>[1448, 1663]</b> #		1567	[1441, 1750]	

Solver	$n$	Predicted confidence intervals		Observed median run-time	
		Exp. model		Point estimates	Confidence intervals
Concorde-Find+Prove	500	[56.18, 77.04]		38.44	[35.43, 41.87]
	600	[78.77, 103.7]		60.72	[57, 68.06]
	700	<b>[110.8, 139.7]</b>		105.9	[97.44, 113.6]
	800	<b>[155.5, 188.3]</b> #		159	[147, 171.7]
	900	<b>[218.6, 254.1]</b> #		220.9	[207, 242.1]
	1000	<b>[305.6, 342.8]</b> #		332.1	[301.4, 384.4]
	1100	<b>[428.1, 464.9]</b>		496.2	[450.9, 530.8]
	1200	<b>[592.1, 635.9]</b> #		611.4	[552.9, 667.3]
	1300	<b>[812.6, 876.9]</b> #		875.6	[782.9, 974.8]
	1400	<b>[1104, 1225]</b> #		1152	[1084, 1268]
1500	<b>[1496, 1712]</b> #		1567	[1441, 1750]	

Solver	$n$	Predicted confidence intervals		Observed median run-time	
		RootExp. model		Point estimates	Confidence intervals
Concorde-Find+Prove	500	<b>[33.45, 51.49]</b> *		38.44	[35.43, 41.87]
	600	<b>[55.6, 80.08]</b> *		60.72	[57, 68.06]
	700	<b>[88.68, 120.1]</b> *		105.9	[97.44, 113.6]
	800	<b>[136.8, 175.3]</b> *		159	[147, 171.7]
	900	<b>[206.2, 250.1]</b> *		220.9	[207, 242.1]
	1000	<b>[303.1, 350.1]</b> #		332.1	[301.4, 384.4]
	1100	<b>[436.4, 482.4]</b>		496.2	[450.9, 530.8]
	1200	<b>[613, 660.8]</b>		611.4	[552.9, 667.3]
	1300	<b>[836.6, 902.5]</b> #		875.6	[782.9, 974.8]
	1400	<b>[1117, 1240]</b> #		1152	[1084, 1268]
1500	<b>[1470, 1685]</b> #		1567	[1441, 1750]	

Solver	$n$	Predicted confidence intervals		Observed median run-time	
		ExpLogRoot. model		Point estimates	Confidence intervals
Concorde-Find+Prove	500	<b>[38.16, 57.6]</b> #		38.44	[35.43, 41.87]
	600	<b>[60.65, 86.16]</b> #		60.72	[57, 68.06]
	700	<b>[93.46, 125.5]</b> *		105.9	[97.44, 113.6]
	800	<b>[140.9, 179.4]</b> *		159	[147, 171.7]
	900	<b>[209.1, 252.5]</b> #		220.9	[207, 242.1]
	1000	<b>[303.7, 350.5]</b> #		332.1	[301.4, 384.4]
	1100	<b>[434.6, 480.2]</b>		496.2	[450.9, 530.8]
	1200	<b>[609, 656.4]</b> #		611.4	[552.9, 667.3]
	1300	<b>[832, 897.5]</b> #		875.6	[782.9, 974.8]
	1400	<b>[1115, 1237]</b> #		1152	[1084, 1268]
1500	<b>[1475, 1691]</b> #		1567	[1441, 1750]	

Solver	$n$	Predicted confidence intervals		Observed median run-time	
		RootExp1. model		Point estimates	Confidence intervals
	500	[50.54, 55.68]		38.44	[35.43, 41.87]
	600	[78.35, 86.32]		60.72	[57, 68.06]

Solver	$n$	Predicted confidence intervals		Observed median run-time	
		Poly. model		Point estimates	Confidence intervals
Concorde-Find+Prove	2000	[4062, 5556]		6576	[5772, 7515]
	2500	[ <b>9057</b> , <b><math>1.414 \times 10^4</math></b> ]		$[1.973 \times 10^4, 1.999 \times 10^4]$	$[1.12 \times 10^4, 3.041 \times 10^4]$
	3000	$[1.738 \times 10^4, 3.035 \times 10^4]$		$[4.707 \times 10^4, 5.145 \times 10^4]$	$[3.448 \times 10^4, 6.784 \times 10^4]$
	3500	$[3.005 \times 10^4, 5.815 \times 10^4]$		$[1.069 \times 10^5, 1.171 \times 10^5]$	$[6.589 \times 10^4, 1.911 \times 10^5]$
Solver	$n$	Predicted confidence intervals		Observed median run-time	
		Exp. model		Point estimates	Confidence intervals
Concorde-Find+Prove	2000	[ <b>6673</b> , <b>9267</b> ]		6576	[5772, 7515]
	2500	[ <b><math>2.944 \times 10^4</math></b> , <b><math>5.076 \times 10^4</math></b> ]		$[1.973 \times 10^4, 1.999 \times 10^4]$	$[1.12 \times 10^4, 3.041 \times 10^4]$
	3000	$[1.301 \times 10^5, 2.781 \times 10^5]$		$[4.707 \times 10^4, 5.145 \times 10^4]$	$[3.448 \times 10^4, 6.784 \times 10^4]$
	3500	$[5.728 \times 10^5, 1.52 \times 10^6]$		$[1.069 \times 10^5, 1.171 \times 10^5]$	$[6.589 \times 10^4, 1.911 \times 10^5]$
Solver	$n$	Predicted confidence intervals		Observed median run-time	
		RootExp. model		Point estimates	Confidence intervals
Concorde-Find+Prove	2000	[ <b>5070</b> , <b>6995</b> ]#		6576	[5772, 7515]
	2500	[ <b><math>1.505 \times 10^4</math></b> , <b><math>2.453 \times 10^4</math></b> ]#		$[1.973 \times 10^4, 1.999 \times 10^4]$	$[1.12 \times 10^4, 3.041 \times 10^4]$
	3000	[ <b><math>4 \times 10^4</math></b> , <b><math>7.663 \times 10^4</math></b> ]#		$[4.707 \times 10^4, 5.145 \times 10^4]$	$[3.448 \times 10^4, 6.784 \times 10^4]$
	3500	[ <b><math>9.878 \times 10^4</math></b> , <b><math>2.181 \times 10^5</math></b> ]#		$[1.069 \times 10^5, 1.171 \times 10^5]$	$[6.589 \times 10^4, 1.911 \times 10^5]$
Solver	$n$	Predicted confidence intervals		Observed median run-time	
		ExpLogRoot. model		Point estimates	Confidence intervals
Concorde-Find+Prove	2000	[ <b>5345</b> , <b>7403</b> ]#		6576	[5772, 7515]
	2500	[ <b><math>1.703 \times 10^4</math></b> , <b><math>2.83 \times 10^4</math></b> ]#		$[1.973 \times 10^4, 1.999 \times 10^4]$	$[1.12 \times 10^4, 3.041 \times 10^4]$
	3000	[ <b><math>4.963 \times 10^4</math></b> , <b><math>9.759 \times 10^4</math></b> ] +		$[4.707 \times 10^4, 5.145 \times 10^4]$	$[3.448 \times 10^4, 6.784 \times 10^4]$
	3500	[ <b><math>1.345 \times 10^5</math></b> , <b><math>3.103 \times 10^5</math></b> ] +		$[1.069 \times 10^5, 1.171 \times 10^5]$	$[6.589 \times 10^4, 1.911 \times 10^5]$
Solver	$n$	Predicted confidence intervals		Observed median run-time	
		RootExp1. model		Point estimates	Confidence intervals
Concorde-Find+Prove	2000	[5000, 5508]		6576	[5772, 7515]
	2500	[ <b><math>1.479 \times 10^4</math></b> , <b><math>1.629 \times 10^4</math></b> ]		$[1.973 \times 10^4, 1.999 \times 10^4]$	$[1.12 \times 10^4, 3.041 \times 10^4]$
	3000	[ <b><math>3.943 \times 10^4</math></b> , <b><math>4.344 \times 10^4</math></b> ]		$[4.707 \times 10^4, 5.145 \times 10^4]$	$[3.448 \times 10^4, 6.784 \times 10^4]$
	3500	[ <b><math>9.714 \times 10^4</math></b> , <b><math>1.07 \times 10^5</math></b> ] +		$[1.069 \times 10^5, 1.171 \times 10^5]$	$[6.589 \times 10^4, 1.911 \times 10^5]$
Solver	$n$	Predicted confidence intervals		Observed median run-time	
		RootExpU1. model		Point estimates	Confidence intervals
Concorde-Find+Prove	2000	[ <b>5804</b> , <b>6409</b> ]		6576	[5772, 7515]
	2500	[ <b><math>1.906 \times 10^4</math></b> , <b><math>2.104 \times 10^4</math></b> ]#		$[1.973 \times 10^4, 1.999 \times 10^4]$	$[1.12 \times 10^4, 3.041 \times 10^4]$
	3000	[ <b><math>5.582 \times 10^4</math></b> , <b><math>6.164 \times 10^4</math></b> ]		$[4.707 \times 10^4, 5.145 \times 10^4]$	$[3.448 \times 10^4, 6.784 \times 10^4]$
	3500	[ <b><math>1.5 \times 10^5</math></b> , <b><math>1.656 \times 10^5</math></b> ]		$[1.069 \times 10^5, 1.171 \times 10^5]$	$[6.589 \times 10^4, 1.911 \times 10^5]$

Table 6: 95% bootstrap confidence intervals for the Q75s of the running time predictions and observed running times on RUE. The instance sizes shown here are larger than those used for fitting the models. Bootstrap intervals on predictions that are weakly consistent with the observed data are shown in boldface, those that are consistent are marked by plus signs (+), those that are strongly consistent are marked by sharps (#), and those that fully contain the confidence intervals on observations are marked by asterisks (\*).