Université Libre de Bruxelles

Institut de Recherches Interdisciplinaires
et de Développements en Intelligence Artificielle

Effect of Transformations of Numerical Parameters in Automatic Algorithm Configuration

A. Franzin, L. Pérez Cáceres, and T. Stützle

IRIDIA – Technical Report Series

Technical Report No.
TR/IRIDIA/2017-006

March 2017

IRIDIA – Technical Report Series
ISSN 1781-3794

Published by:
IRIDIA, *Institut de Recherches Interdisciplinaires et de Développements en Intelligence Artificielle*
Université Libre de Bruxelles
Av F. D. Roosevelt 50, CP 194/6
1050 Bruxelles, Belgium

Technical report number TR/IRIDIA/2017-006

Noname manuscript No.
(will be inserted by the editor)

Effect of Transformations of Numerical Parameters in Automatic Algorithm Configuration

Alberto Franzin · Leslie Pérez Cáceres · Thomas Stützle.

the date of receipt and acceptance should be inserted later

Abstract In this work, we study the impact of altering the sampling space of parameters in automatic algorithm configurators. We show that a proper transformation can strongly improve the convergence towards better configurations; at the same time, wrong assumptions about good parameter values, possibly based on misleading prior knowledge, may lead to wrong choices in the transformations and, hence, be detrimental for the configuration process. To emphasize the impact of the transformations, we study their effect on configuration tasks with a single parameter in different experimental settings.

1 Introduction

Automatic algorithm configuration has shown to be crucial for reaching high performing parameter settings of search algorithms [8] and a number of effective configurators are available [1, 9, 10, 13, 16, 5]. Modern configurators can handle a large number of parameters and different types of parameters including categorical, ordinal and numerical ones. In this article, we focus on the latter type of parameters.

Choosing the value of a numerical parameter for an algorithm can be difficult due to three intertwined issues: (i) the wide range of possible values, (ii) the possibility of parameter interactions, and (iii) the impact a variation of a parameter value has. Automatic algorithm configuration techniques relieve algorithm designers and practitioners of the first two issues. The third issue depends in part on the parameter representation and the sensitivity of each single

This research and its results have been made possible through funding from the COMEX project (P7/36) within the Interuniversity Attraction Poles Programme of the Belgian Science Policy Office. Thomas Stützle acknowledges support from the Belgian F.R.S.-FNRS, of which he is a Senior Research Associate.

IRIDIA, Université Libre de Bruxelles
50, Av. F. Roosevelt, 1050 Ixelles
E-mail: {afranzin,lperez,stuetzle}@ulb.ac.be

parameter during the configuration task. The representation issue is related to the question whether a parameter should be varied according to an additive or a multiplicative scale. For example, in a population based-algorithm it is intuitively clear that the sensitivity of the population size depends on its value: the effect of increasing the population size by a constant number of, say, ten has a very different impact if the increase is from 1 to 11, from 10 to 20, or from 100 to 110. Instead, it seems more intuitive to change population size by a specific factor. Choosing an additive or multiplicative scale is related to the question whether a parameter should be presented in a normal scale or in a logarithmic scale.

In this article, we study the effect of a transformation of the parameter values for automated algorithm configuration. We consider five transformations of a single parameter and study their effect under different configuration budgets. We show how the right transformation helps considerably to discover good parameter values; this impact is stronger for low configuration budgets. At the same time, we show how a wrong transformation, which may be chosen due to misleading prior knowledge, can instead be harmful.

The choice between the transformation has currently to be done by the user. The implementation of such a choice may depend on the particular configurator. For example, ParamILS [9] (as also GGA [1]) requires a discretization of numerical parameters and one possibility is therefore to discretize the numerical interval in normal, logarithmic, or another fashion. When using irace [13], one can apply a transformation in the wrapper script used to execute the experiments, which is also the way how we apply the transformations in this paper. SMAC [10] includes the possibility of exploring a parameter range in a logarithmic way (e.g. $1, 10, 100, \dots$) by specifying a transformation in the parameter definition, but with limited possibility (e.g. it is not possible to reflect the granularity, such as in the case of a parameter taking values $0.9, 0.99, 0.999, \dots$).

The article is structured as follows. In the next section, we define the transformations we studied. We describe our experimental setup in Section 3, report our experimental results in Sections 4 and 5, and discuss them in Section 6.

2 Parameter transformations

As baseline, we consider the parameter space without transformation, that is,

$$\mathcal{I}(x) \triangleq f(x \in [a, b]) \mapsto x \in [a, b] \tag{1}$$

where $[a, b]$ is the range of possible values for parameter x. This is the default behaviour of configurators such as SMAC or irace. This baseline is well-suited for parameters whose effect reflects an additive scale. For a multiplicative scale, we may apply a logarithmic transformation of the parameter:

$$\mathcal{L}og(x) \triangleq f(x \in [\log a, \log b]) \mapsto 10^x \in [a, b], \tag{2}$$

that is, we sample the logarithm of the parameter under consideration. For ease of representation, we consider base-10 exponentials and logarithms. (Note that this is also the transformation that is natively provided by SMAC.) However, it should be noted that the base chosen might have an impact on the configuration process, especially for (very) low configuration budgets. As for $a = 0$, $\log a$ is not possible, we use as lower bound the precision δ that we use when we sample real-valued parameters. For example, if $\delta = 4$ and the parameter range is $[0, 1]$, our sampling interval will be $[-4, 0]$.

The effect of a $\mathcal{L}og(x)$ transformation is a modification of the probability of sampling specific parameter ranges, favoring the lower part of the interval. For example, assuming uniform distributions, in a logarithmic range $[0, 4]$, corresponding to a final parameter value range of $[1, 10\,000]$, there is a probability of 0.25 of sampling a value between zero and one (three and four), which actually corresponds to a transformed parameter between one and ten (1 000 and 10 000) once rescaled, an interval whose probability of being hit in a non-transformed scale is 0.001 (0.9).

In case we are interested in exploiting the upper subrange of the interval, we define the reflection in the parameter range of the logarithmic transformation:

$$\mathcal{RL}og(x) \triangleq f(x \in [\log a, \log b]) \mapsto b - 10^x + a \in [a, b]. \tag{3}$$

This transformation is useful in the dual case of $\mathcal{L}og(x)$: consider a parameter that represents a probability that we want to take values very close to 1, with a precision of four decimal digits, resulting in a sampling interval $[-4, 0]$. As an example, with probability 0.25, we could sample a value between -4 and -3, which maps to the range $[0.9991, 1]$ after the transformation.

Milder transformations are also possible, for example the power function with exponent greater than 1, that in an interval $[0, 1]$ magnifies the area close to 0. Here, we consider a quadratic transformation

$$\mathcal{S}(x) \triangleq f(x \in [a, b]) \mapsto \left(\frac{x-a}{b-a}\right)^2 \cdot (b-a) + a \in [a, b], \tag{4}$$

exploring the lower subrange; by reflecting it in the parameter range we obtain

$$\mathcal{RS}(x) \triangleq f(x \in [a, b]) \mapsto b - \left(\frac{x-a}{b-a}\right)^2 \cdot (b-a) \in [a, b] \tag{5}$$

that explores the upper subrange.

Also powers with exponent between 0 and 1 can be useful to exploit the upper part of the interval; we consider the square root of the normalized sampled value

$$\mathcal{S}qrt(x) \triangleq f(x \in [a, b]) \mapsto \left(\frac{x-a}{b-a}\right)^{1/2} \cdot (b-a) + a \in [a, b]. \tag{6}$$

3 Materials

We evaluate the transformations of Section 2 using different variations of Simulated Annealing (SA) [11]. SA is a popular metaheuristic, which moves through solutions accepting them using a probabilistic acceptance criterion. In the original formulation SA uses the Metropolis criterion [14], which accepts a solution in a minimization problem with a probability given by

$$p = \begin{cases} 1, & \text{if } \Delta \leq 0 \\ \exp(-(\Delta/T_k)), & \text{otherwise} \end{cases} \tag{7}$$

where Δ is the difference between the cost of the new and the current solution and T_k is the temperature parameter after k cooling steps, that is, after k times the temperature has been reduced. A solution is always accepted if it is better than or equal to the current incumbent solution; if it is worse, it is accepted with a probability that depends both on the relative difference in terms of objective function values between the two solutions and on the stage of the execution. The temperature parameter is usually decreased during the search following a so-called cooling scheme; the number of solutions evaluated at a same temperature value T_k is called temperature length. The effect of the temperature parameter is to gradually transition from an initial exploratory search phase, in which more uphill moves are accepted, to a final exploitative search phase, in which almost only improving moves are accepted.

We consider SA in two different settings. In the first one, reported in Section 4, we study a fixed temperature variant of the original SA formulation [4, 6], where the temperature is held constant during the whole runtime. The cooling scheme is therefore replaced by the formula

$$T_{k+1} = T_k = T_0 \quad \forall\, k, \tag{8}$$

where T_0 is the initial temperature. The acceptance probability of a worsening move depends therefore only on Δ. Hence, the only numerical parameter to be considered is T_0.

In Section 5, we consider the original formulation of SA with a geometric cooling as

$$T_k = \alpha \times T_{k-1}, \tag{9}$$

where $\alpha \in [0, 1]$ controls the decrease of the temperature value[1].

Low values of α enforce a very quick decrease in the temperature, as shown in Figure 1; for example for values of α equal to 0.1 or 0.5 the acceptance probability becomes negligible already after a few cooling steps, making the acceptance of worsening solutions very unlikely. As α gets closer to 1, many more cooling steps are needed to decrease the acceptance probability of worsening moves. Traditionally, in the SA literature α is chosen close to 1, to ensure that the algorithm does not converge too quickly.

[1] Though traditionally α is chosen in $(0, 1)$, we include also the extreme values, where $\alpha = 0$ turns SA into a first improvement scheme, and $\alpha = 1$ into a fixed temperature schemeas in the experiments of Section 4.

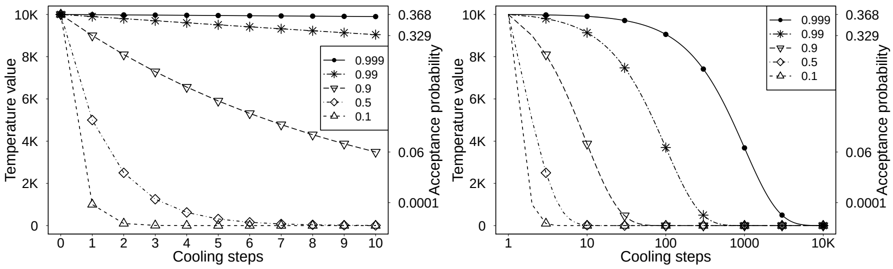

Fig. 1 Evolution of the temperature for different values of α, measured over 10 cooling steps (left plot) and 10 000 cooling steps (right plot, in logarithmic scale). On the right side is indicated the acceptance probability of a solution whose objective function value is worse than the optimal one by $\Delta = 10\,000$.

To study the impact of the various transformations, we consider two cases, one in which the best value of α is close to 1, and one in which the best value is close to 0; such settings are obtained by fixing the temperature length either as very small or very large. The other parameters are kept fixed: the initial temperature is given by the maximum gap between consecutive solutions observed during an initial random walk of 10 000 steps in the search space.

We evaluate the five transformations applying the SA algorithm to the Quadratic Assignment Problem (QAP) [3]. We use instances of size 100 facilities and locations, generated uniformly at random [15]; 25 instances are used for the tuning, and 25 for testing of the tuned parameter settings. For both tuning and testing we stop the SA algorithm after two seconds CPU time on an Intel Xeon E5-2680 CPU. The tuning is done with irace [13] using three different budgets, namely 125, 500 and 2 000 experiments per tuning. Each tuning is run 30 times; after each tuning we obtained a tuned parameter value and we then compute the average relative percentage deviation (ARPD) obtained for each tuned value on the 25 test instances. In Sections 4 and 5, we show the results obtained using four digits as precision for the sampling of the parameter values. In Appendix A we show the results obtained using two decimal digits as precision; some discussion about the use of a more limited precision is, however, given in Section 6.

4 Simulated Annealing with a fixed temperature

We allow the temperature T_0 to take any integer value in the area $[1, 10\,000\,000]$; this translates in an interval $[0, 7]$ for the logarithmic transformations $\mathcal{L}og(x)$ and $\mathcal{RL}og(x)$. In our instance set, the values that give the best results lie in a very narrow range close to the lower part of the interval, as shown in the plots of Figure 4; the best values are around 4 500. In this context, we expect the logarithmic transformation $\mathcal{L}og(x)$, and the quadratic one $\mathcal{S}(x)$, to outperform

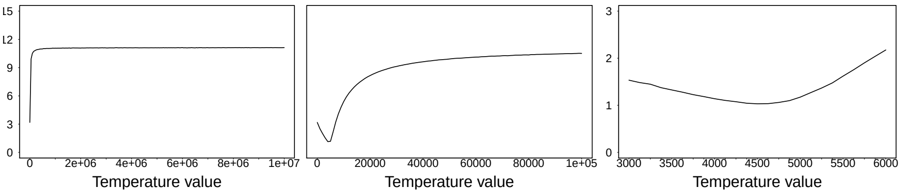

Fig. 2 Landscape of the results obtained by different fixed temperature values, in terms of ARPD (given on y-axis). Given are respectively, the full landscape, the landscape of the results obtained in the range $(0, 10\,000)$ (the lowest 1% of possible values) of the fixed temperature, and the subinterval close to optimal values are reported.

their counterparts, especially in the low budget case. The "good" parameter values lie in fact in the center of the exponent interval of $\mathcal{L}og(x)$.

The results of the five transformations are shown in Figure 3. The boxplots show the ARPD values as given by the thirty parameter values obtained form the thirty tunings. For each transformation we report three results, obtained respectively with 125, 500 and 2000 experiments per tuning to observe the convergence behaviour exhibited by each transformation.

As expected, the logarithmic transformation outperforms all the other ones, with very good results already with the smallest budget, followed by the quadratic transformation. The identity transformation needs a much larger budget than the logarithmic and square transformations to reach good results, while the remaining ones, which are thought to exploit the upper part of the interval, show, as expected, very poor results.

Figures 4 and 5 give boxplots of the parameter values found across the 30 tunings with 125 and 2000 experiments respectively. They show that the high performance reached when using transformations $\mathcal{L}og(x)$ and $\mathcal{S}(x)$ is because they allow to identify much more precisely the best parameter values, while transformations aimed to exploit the upper part of the range are indeed harmful. This effect is most visible for the smallest tuning budget level, as with a larger budget, also other transformation (e.g. the identity one) converge to the best values around 4500 (see Figure 2). Still, as shown in Figure 6, $\mathcal{S}(x)$ and especially $\mathcal{L}og(x)$ converge towards a very narrow area around the best value.

5 Geometric cooling in SA

5.1 Small temperature length

The first experiment uses a small temperature length, which enforces that the best parameter settings for α are close to 1. This is done by applying a cooling step after a number of moves that is equal to the size of the neighbourhood. The ARPD in dependence of the values for α is shown in Figure 7 for the

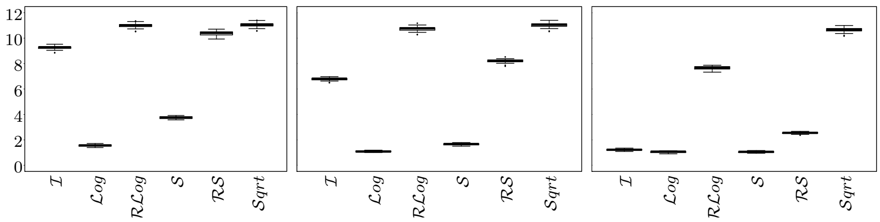

Fig. 3 ARPD values obtained by the different transformations. The three plots show the results obtained with a budget of 125, 500 and 2000 experiments, respectively.

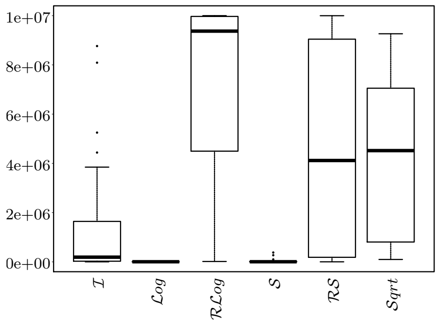

Fig. 4 Parameter values obtained by irace with a budget of 125 experiments.

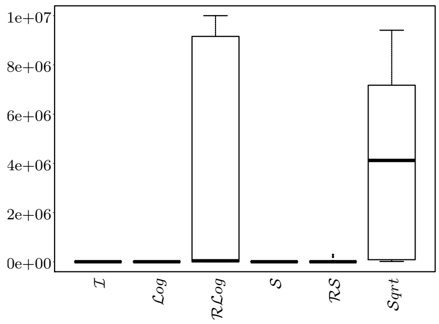

Fig. 5 Parameter values obtained by irace with a budget of 2000 experiments.

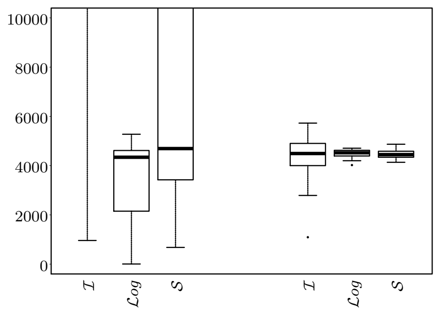

Fig. 6 Parameter values obtained by irace using budgets of 125 (left boxplots) and 2000 experiments (right boxplots) for $\mathcal{I}(x)$ $\mathcal{L}og(x)$ and $\mathcal{S}(x)$.

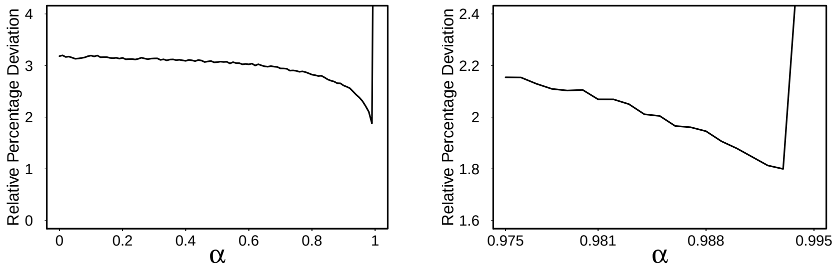

Fig. 7 ARPD in dependence of the value of parameter α for small temperature length; the left plot gives the full parameter range and the right plot the subinterval with the best-performing values.

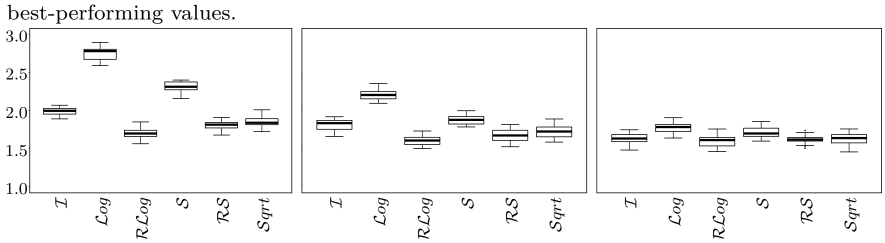

Fig. 8 ARPD (given on y-axis) obtained by the different transformations for a small temperature length. The three plots show the results obtained with a budget of 125, 500 and 2 000 experiments, respectively.

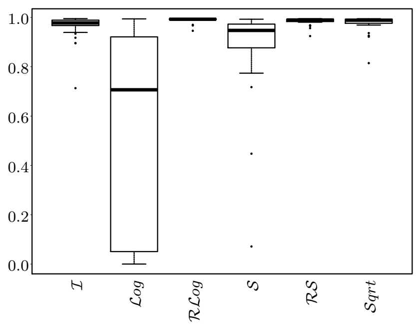

Fig. 9 Parameter values obtained by irace with a budget of 125 experiments for a small temperature length.

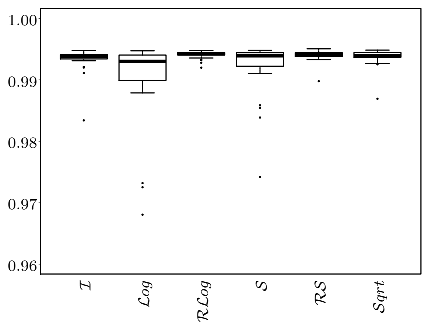

Fig. 10 Parameter values obtained by irace with a budget of 2 000 experiments for a small temperature length.

whole parameter range (left plot) and for the best-performing value (right plot). There is a slow but clear improvement as α grows, until $\alpha = 0.993$; for $\alpha = 0.994$ the results are already very bad, as shown by the spike in the plots for values close to $\alpha = 1$. This is an extreme condition, where a tiny variation can make a huge difference but also a good benchmark as the very best parameter values stem from a rather narrow range.

In Figure 8, we show the results obtained by each transformation with a budget of 125, 500 and 2 000 experiments per tuning. It is quite evident how the normal sampling is outperformed by the transformations aimed to exploit the region of the range close to 1, which are $\mathcal{RL}og(x)$ and $\mathcal{RS}(x)$. The transformations $\mathcal{L}og(x)$ and $\mathcal{S}(x)$ obtain worse results, as they tend to exploit the wrong range of values.

In Figures 9 and 10 we show the parameter values obtained by irace and a budget of, respectively, 125 and 2 000. As expected, transformations $\mathcal{RL}og(x)$, $\mathcal{RS}(x)$ and $\mathcal{S}qrt(x)$ show a quicker convergence to the best parameter values with respect to $\mathcal{I}(x)$, and even more so than $\mathcal{L}og(x)$ and $\mathcal{S}(x)$. In this case, differently from the previous experiment of Section 4, also the transformations that are expected to perform badly are able to converge to a narrow, though suboptimal, subrange of values, at least for the high budget case. Analyzing more in detail the behaviour of the transformations aimed to exploit the upper subrange of parameter values, $\mathcal{RL}og(x)$ converges very quickly to the best values, while $\mathcal{RS}(x)$ and $\mathcal{S}qrt(x)$ exhibit a somewhat slower convergence.

5.2 High temperature length

In the second experiment, we consider a high temperature length by applying a cooling step only after a number of moves that is 500 times the neighbourhood size. This setting of the temperature length enforces that the best parameter settings for α are close to 0. In Figure 11 we show the ARPD values in dependence of the parameter α over the full (left plot) and the best-performing (right plot) subrange. In this case, we have a smoother and relatively wider “good area” of parameter values than in the case of a short temperature length, roughly centered between $\alpha = 0.025$ and $\alpha = 0.035$.

Figure 12 shows the obtained results. As expected, the situation is reversed with respect to the previous situation. In this case, the $\mathcal{L}og(x)$ and $\mathcal{S}(x)$ transformations exhibit a slightly quicker convergence to the best values than the identity transformation, while the remaining transformations $\mathcal{S}qrt(x)$, $\mathcal{RS}(x)$ and $\mathcal{RL}og(x)$ obtain much worse results. These latter ones require a large number of experiments (here 2 000) to converge to good values. In Figures 13 and 14 we observe the convergence of the parameter values obtained by irace with a budget of, respectively, 125 and 2 000 experiments. As the experiment is designed to exploit the lower subrange, the most effective transformations are $\mathcal{L}og(x)$ and $\mathcal{S}(x)$. A more detailed examination shows that $\mathcal{L}og(x)$ and $\mathcal{S}(x)$ converge slightly faster than $\mathcal{I}(x)$ to the best values; however, here the differences are not very pronounced, as the range of very good values appears to be somewhat larger as for the case with low temperature length and the variability between similar values larger.

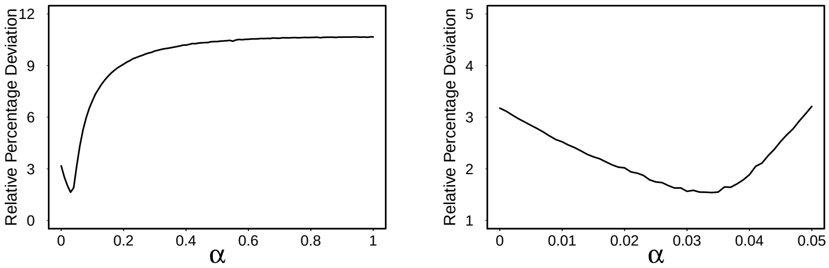

Fig. 11 ARPD in dependence of the value of parameter α for high temperature length; the left plot gives the full parameter range and the right plot the subinterval with the best-performing values.

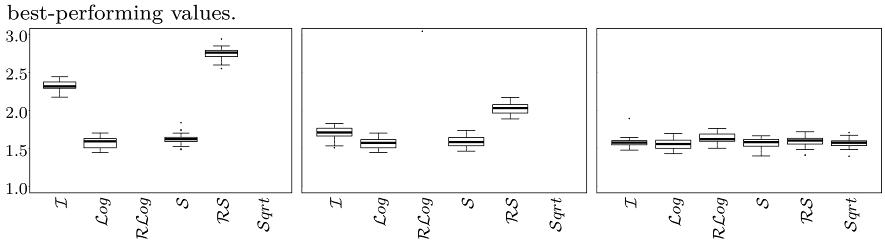

Fig. 12 ARPD (given on y-axis) obtained by the different transformations for a high temperature length. The three plots show the results obtained, respectively, with a budget of 125, 500 and 2 000 experiments.

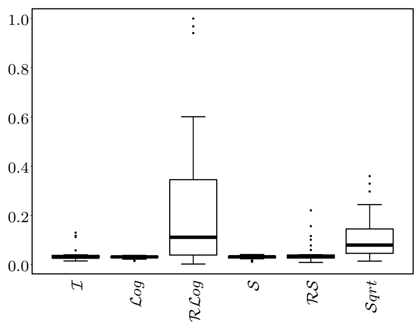

Fig. 13 Parameter values obtained by irace with a budget of 125 experiments for a high temperature length.

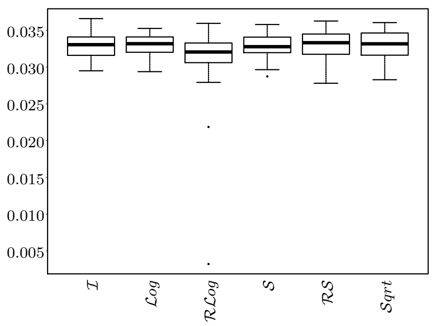

Fig. 14 Parameter values obtained by irace with a budget of 2 000 experiments for a high temperature length.

6 Discussion and conclusions

Parameter space transformations have been considered before, but their impact has never been studied explicitly in the optimization community, unlike in other research communities such as medicine [7, 12]. In Sections 4 and 5.2 we have shown two experiments in which the optimal parameter values lie in the lower part of the allowed range. In this case, the transformations that are designed to exploit that specific area, namely the logarithmic transformation $\mathcal{L}og(x)$ and the square transformation $\mathcal{S}(x)$, also show a faster convergence towards good values with respect to the identity transformation. Analogously, in Section 5.1 we see how their bounded reflections, namely the transformations $\mathcal{RL}og(x)$ and $\mathcal{RS}(x)$, outperform the identity transformation. The differences in convergence speed are more pronounced in the case of a large range of possible parameter values, as shown in Section 4, where the identity transformation needs eight times the budget needed by the logarithmic transformation to reach comparable results, even if only a single parameter is tuned.

These results illustrate the usefulness of knowledge about the parameter space in practice. Tuning a single parameter is a seemingly trivial task, but the choice of a right transformation can make this task even less computationally expensive. This is highly desirable, as the hardness of the tuning task grows strongly with the number of parameters.

However, a wrong transformation is detrimental for the tuning process. This is illustrated by the example in Section 5.2. Traditionally, the cooling coefficient α of the geometric cooling in Simulated Annealing is set to values close to 1, but in the example of Section 5.2 the best-performing values are actually close to zero. While this experiment may seem artificial, it is more important to note that in absolute terms, well-tuned parameter settings from the example in Section 5.2 perform better than well-tuned parameter settings from the example in Section 5.1. In fact, the average relevative percentage deviations obtained are 1.55 and 1.59, respectively; a pairwise Wilcoxon test shows that this small difference is actually statistically significant. In a sense, while the definition and application of transformations is often intuitive if the parameter landscape is known, our results show that inaccurate assumptions may lead to wrong choices of transformations, and be harmful for the quality of the final results.

The results obtained using a more limited precision (two decimal digits, reported in Appendix A) show how the quality of the results obtained is comparable to the quality obtained using four decimal digits. However, using a higher precision allows to obtain slightly lower deviations from the best known solutions. To this regard, it is interesting to observe the case of the experiment reported in Section 5.1, where for $\alpha = 0.993$ the best solutions are obtained, but for $\alpha = 0.994$ the results are very bad. This is a safe benchmark for the more coarse precision. Yet, all the transformations, including the identity one, are able to reach better results with four digits of precision, meaning that the transformations are robust enough to handle limit cases like the one reported.

References

1. C. Ansótegui, M. Sellmann, and K. Tierney. A gender-based genetic algorithm for the automatic configuration of algorithms. In *CP 2009*, volume 5732 of *LNCS*, pages 142–157. Springer, Heidelberg, Germany, 2009.

2. R. E. Burkard, E. Çela, P. M. Pardalos, and L. S. Pitsoulis. The quadratic assignment problem. In P. M. Pardalos and D.-Z. Du, editors, *Handbook of Combinatorial Optimization*, volume 2, pages 241–338. Kluwer Academic Publishers, 1998.
3. E. Çela. *The Quadratic Assignment Problem: Theory and Algorithms.* Kluwer Academic Publishers, Dordrecht, The Netherlands, 1998.
4. H. Cohn and M. J. Fielding. Simulated annealing: Searching for an optimal temperature. *SIAM Journal on Optimization*, 9(3):779–802, 1999.
5. A. E. Eiben and S. K. Smit. Parameter tuning for configuring and analyzing evolutionary algorithms. *Swarm and Evolutionary Computation*, 1 (1):19–31, 2011.
6. M. J. Fielding. Simulated annealing with an optimal fixed temperature. *SIAM Journal on Optimization*, 11(2):289–307, 2000.
7. N. Girerd, M. Rabilloud, P. Pibarot, P. Mathieu, and P. Roy. Quantification of treatment effect modification on both an additive and multiplicative scale. *PLOS ONE*, 11(4):1–14, 04 2016.
8. H. H. Hoos. Programming by optimization. *Communications of the ACM*, 55(2):70–80, February 2012.
9. F. Hutter, H. H. Hoos, K. Leyton-Brown, and T. Stützle. ParamILS: an automatic algorithm configuration framework. *JAIR*, 36:267–306, 2009.
10. F. Hutter, H. H. Hoos, and K. Leyton-Brown. Sequential model-based optimization for general algorithm configuration. In *LION 5*, volume 6683 of *LNCS*, pages 507–523. Springer, Heidelberg, Germany, 2011.
11. S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. Optimization by simulated annealing. *Science*, 220:671–680, 1983.
12. M. J. Knol, T. J. VanderWeele, R. H. H. Groenwold, O. H. Klungel, M. M. Rovers, and D. E. Grobbee. Estimating measures of interaction on an additive scale for preventive exposures. *European Journal of Epidemiology*, 26(6):433–438, 2011.
13. M. López-Ibáñez, J. Dubois-Lacoste, L. Pérez Cáceres, T. Stützle, and M. Birattari. The irace package: Iterated racing for automatic algorithm configuration. *ORP*, 3:43–58, 2016.
14. N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. Teller, and E. Teller. Equation of state calculations by fast computing machines. *Journal of Chemical Physics*, 21:1087–1092, 1953.
15. É. D. Taillard. Robust taboo search for the quadratic assignment problem. *Parallel Computing*, 17(4-5):443–455, 1991.
16. Z. Yuan, M. A. Montes de Oca, T. Stützle, and M. Birattari. Continuous optimization algorithms for tuning real and integer algorithm parameters of swarm intelligence algorithms. *SI*, 6(1):49–75, 2012.

A Results obtained with two decimal digits of precision

A.1 Simulated Annealing with a fixed temperature

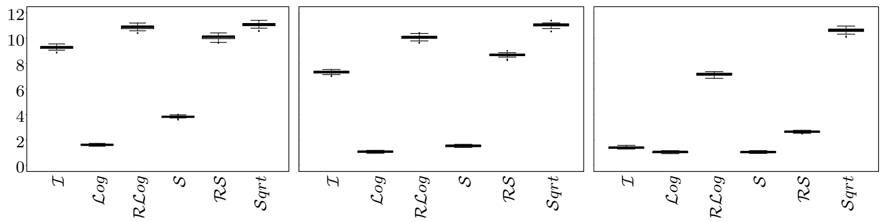

Fig. 15 Average Relative Percentage Deviation values obtained by the different transformations with respect to the best known solutions. The three plots show the results obtained, respectively, with a budget of 125, 500 and 2000 experiments.

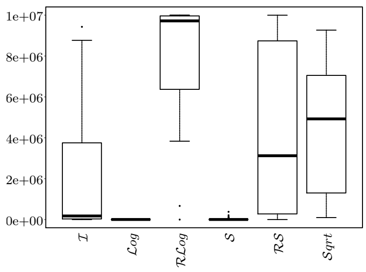

Fig. 16 Parameter values obtained by irace with a budget of 125 experiments.

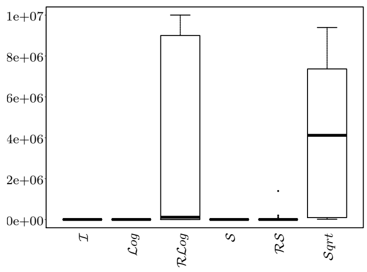

Fig. 17 Parameter values obtained by irace with a budget of 2000 experiments.

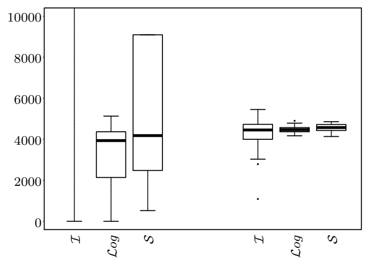

Fig. 18 Parameter values obtained by irace using budgets of 125 (left boxplots) and 2000 experiments (right boxplots) for $\mathcal{I}(x)$ $\mathcal{L}og(x)$ and $\mathcal{S}(x)$.

A.2 Geometric cooling with a short temperature length

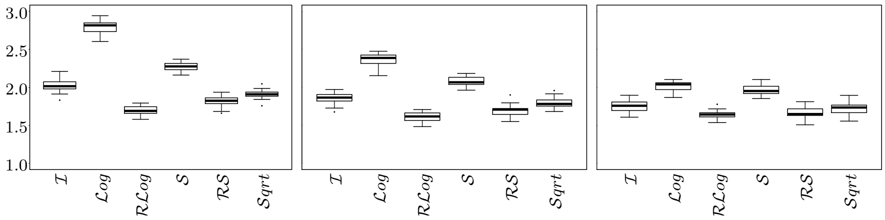

Fig. 19 Average Relative Percentage Deviation values obtained by the different transformations for a short temperature length with respect to the best known solutions. The three plots show the results obtained, respectively, with a budget of 125, 500 and 2000 experiments.

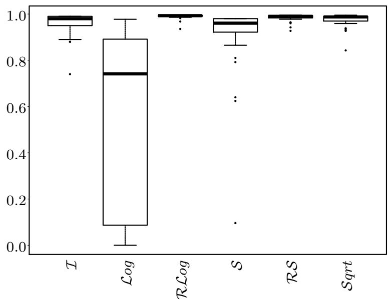

Fig. 20 Parameter values obtained by irace with a budget of 125 experiments for a short temperature length.

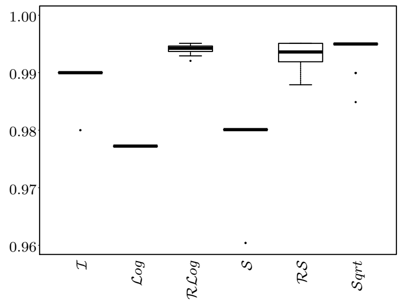

Fig. 21 Parameter values obtained by irace with a budget of 2000 experiments for a short temperature length.

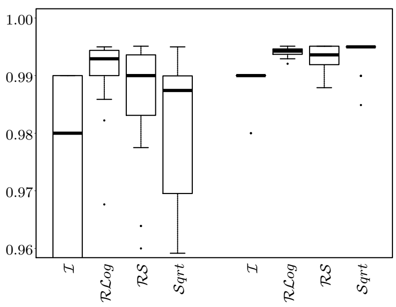

Fig. 22 Parameter values obtained by irace using budgets of 125 (left boxplots) and 2000 (right boxplots) experiments for a short temperature length.

A.3 Geometric cooling with a high temperature length

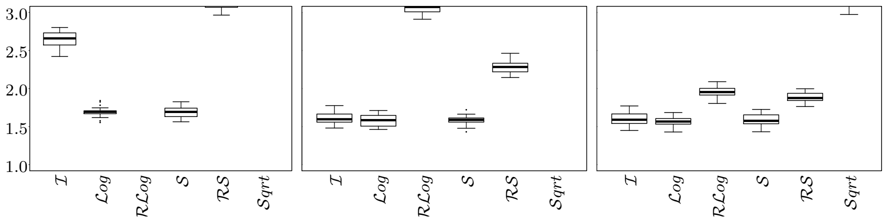

Fig. 23 Average Relative Percentage Deviation values obtained by the different transformations for a high temperature length with respect to the best known solutions. The three plots show the results obtained, respectively, with a budget of 125, 500 and 2000 experiments.

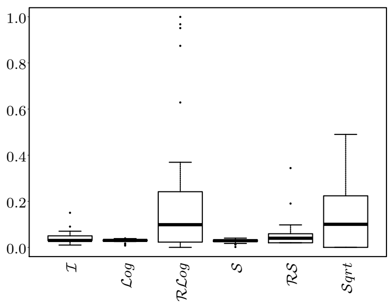

Fig. 24 Parameter values obtained by irace with a budget of 125 experiments for a high temperature length.

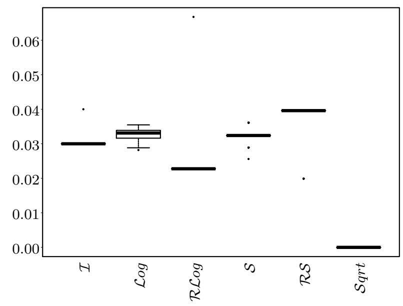

Fig. 25 Parameter values obtained by irace with a budget of 2000 experiments for a high temperature length.

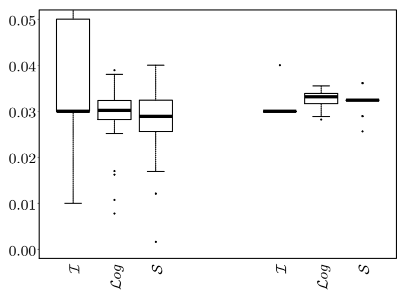

Fig. 26 Parameter values obtained by irace using budgets of 125 (left boxplots) and 2000 (right boxplots) experiments for a high temperature length.