

## Self-Cooling Simulated Annealing (SCSA) Algorithm for Nonlinear Least Square's Data Analysis

Federico Costa<sup>1</sup>, Gustavo Quintero-Marquina<sup>2</sup>, Maximiliano Riddick<sup>3</sup>, Leandro Andrini<sup>4</sup>, Alejandra Vahnovan<sup>4</sup> and Cristián Huck-Iriart<sup>5\*</sup>

<sup>1</sup>TALP Research Center, Universitat Politecnica de Catalunya, C/ Jordi Girona 1-3, Barcelona, 08034, Spain

<sup>2</sup>Institute of Emerging Technologies and Applied Sciences (ITECA), UNSAM School of Science and Technology, CONICET. San Martín, Buenos Aires, Argentina

<sup>3</sup>Consolidated Nucleus in Pure and Applied Mathematics (NUCOMPA), National University of the Center of the Province of Buenos Aires, Tandil, Argentina

<sup>4</sup>La Plata Mathematics Center (CMaLP), Department of Mathematics, Faculty of Exact Sciences, National University of La Plata, 50th and 115th Street, La Plata, Argentina

<sup>5</sup>ALBA Synchrotron Light Source, Carrer de la Llum 2-26, Cerdanyola del Vallès, 08290 Barcelona, Spain

### ABSTRACT

A Self-Cooling Simulated Annealing (SCSA) algorithm is introduced for the optimization of nonlinear least-squares problems. In contrast to conventional simulated annealing techniques that require a predefined cooling schedule, the SCSA algorithm autonomously regulates system temperature based on the lowest figure of merit (ie.  $\chi^2$ ) achieved at each iteration. The algorithm incorporates two key enhancements to improve efficiency: the separation of linear and nonlinear parameters, which reduces the dimensionality of the stochastic search space, and an adaptive Gaussian sampling mechanism that dynamically updates parameter-specific variances based on recent optimization history. A thermal resistance parameter (K) regulates the cooling rate and can be adjusted according to problem complexity. Performance benchmarking against standard Monte Carlo and gradient-based methods demonstrates that SCSA offers greater robustness in avoiding local minima and provides reliable convergence across varying levels of optimization difficulty. These characteristics make the method broadly applicable to nonlinear data analysis and other complex optimization tasks.

### \*Corresponding author

Cristián Huck-Iriart, 5.ALBA Synchrotron Light Source, Carrer de la Llum 2-26, Cerdanyola del Vallès, 08290 Barcelona, Spain.

**Received:** July 08, 2025; **Accepted:** August 14, 2025, **Published:** August 25, 2025

**Keywords:** Monte Carlo, Data Analysis, Parametric Functions, Simulated Annealing

### Introduction

Through the application of mathematical models across different domains of knowledge, the objective is to discover the minimum set of parameters that completely describe the system under study and the laws that relate the values of these parameters to the results of any set of measurements in the system [1,2]. Furthermore, to the extent that the values of the parameters can only be obtained as a result of measurements, one may equivalently consider that theories impose some relationships between the results of some measurements. Theoretical relationships can be of two types, either functional, in which the values of the parameters are precisely related to the results of the measurements, or probabilistic, where the probabilistic description of the results of the measurements is essential to the theory [3].

By definition, the problem we will address in this paper is an inverse problem, i.e., given information about the values of some measured quantities, a theoretical relationship will be used to obtain information about the values of the set of parameters. That

is, the data are the results of some measurements, whose unknown is a set of parameter values [4,5]. For this reason, parameter estimation is very important within the field of systems modeling. In most cases, the systems are non-linear, and the methods commonly used for estimation are gradient search methods and Newton search methods that combine the principle of maximum likelihood. To make these estimates, when the data set is large, the implementation of codes or/and algorithms becomes necessary [6,7].

Least squares algorithms(5) are used in natural and social sciences to find the parameters of functions that best describe a set of observed data. It is the primary tool for solving curve fitting problems [8,9]. For a parametric function  $y(\vec{x}|\vec{\theta})$  where  $\vec{x} \in \mathbb{R}^N$  and  $\vec{\theta}$  are the model vector parameters  $\vec{\theta} = (\theta_1, \theta_2, \theta_3, \dots, \theta_M)$  the residual function to be minimized is defined by [10]:

$$Z(\vec{\theta}) = \sum_{i=1}^N w_i (y_i - y(x_i|\vec{\theta}))^2 \quad (1)$$

Where  $y_i$ ,  $w_i$  are the experimental data and its weight at the  $i$ -th point, respectively. In the particular case, when the uncertainty in the experimental point  $y_i$ , follows a normal distribution,  $w_i$  can be written as  $\sigma_i^{-2}$  where  $\sigma_i$  is the standard deviation of the normal

distribution corresponding to the error at the  $i$ -th point. Thus, in this condition  $z(\vec{\theta})$  become the well-known figure of merit  $\chi^2$ .

$$\chi^2(\vec{\theta}) = \sum_{i=1}^N \frac{(y_i - y(x_i; \vec{\theta}))^2}{\sigma_i^2} \quad (2)$$

It is often convenient to consider the reduced figure of merit defined by  $\chi^2_{\text{red}} = \chi^2/(N-M)$ , where the function is obtained by normalizing  $\chi^2$  by the degrees of freedom  $(N-M)$ . The minimization of the  $\chi^2$  function can be done by different methodologies, namely, grid-search, gradient-descendent methods in which the most popular algorithm was proposed by Marquardt & Levenberg, and stochastic search parameter thought Monte Carlo methods among others [11-16].

### Simulated Annealing for Data Analysis

The minimization of the figure of merit, in our case  $\chi^2$ , to obtain physical information from a mathematical model's parameter requires selecting the most efficient methodology. If the function is convex with a single minimum, almost any methodology solves the problem easily. However, for multiple minima, the solution requires more advanced methods, moreover, gradient-descendant relies on the calculation of the Hessian matrix which is not always possible and it could be trapped in a local minimum [17]. For these problems, Monte Carlo approaches based on Metropolis algorithm are usually employed. These types of algorithms are based on accept/reject simulation with a given probability [18]. For the Boltzmann distribution ( $B_K$ ), the probability to accept a new state in the  $k$ -th simulation is shown in the equation 3 where  $T$  is the temperature of the system (in analogy with a physical phenomena)

$$P(k|k-1) = \begin{cases} B_k & \text{if } \chi_k^2 > \chi_{k-1}^2 \text{ and } B_k > U(0,1) \\ 0 & \text{if } \chi_k^2 > \chi_{k-1}^2 \text{ and } B_k < U(0,1) \\ 1 & \text{otherwise} \end{cases} \quad (3)$$

Where  $U(0,1)$  is a random number with a uniform distribution between 0 and 1 and the Boltzmann distribution is defined by:

$$B_K \propto \exp(-\beta \cdot [\chi_k^2 - \chi_{k-1}^2]) \quad (4)$$

Where  $\beta = 1/k_b T$  and  $T$  is the temperature of the system (in analogy with a physical phenomena) and  $k_b$  is a constant. Simulated annealing methodology is based on the same accept/reject strategy but the artificial temperatures are introduced and gradually cooled, in analogy with the annealing technique in material science, where the cooling down rate is programmed by the user. Nevertheless, since the nature of the problem may vary from case to case, the cooling program may need adjustments. In this work, a self-cooling algorithm is proposed using as temperature a function of the best solution found in the  $k$ -th iteration thus, the cooling rate depends on the convexity of the particular problem [19,20]. Also, a linear/non-linear parameter decoupling is implemented. These two combined approaches aim to reduce the computation time. Despite we optimized the  $\chi^2$  function, the same routine could be used in other least squares problems.

### Linear and Non-Linear Parameters Decoupling

In order to reduce the parameter searching dimension space a decoupling is proposed based on previous works [21-22]. In this work, we consider  $\vec{\theta} = (\vec{\theta}^L, \vec{\theta}^{NL})$  where  $\vec{\theta}^L \in \mathfrak{R}^r$  represents the linear and  $\vec{\theta}^{NL} \in \mathfrak{R}^{M-r}$  nonlinear parameters.

$$y(x, \vec{\theta}) = \sum_{j=1}^r \theta_j^L f_j(x, \vec{\theta}^{NL}) \quad (5)$$

This strategy allows to compute by means of linear-least squares algorithms  $\vec{\theta}^L$  in each step, thus, the Monte Carlo searching space is reduced to  $M-r$  and the machine time with it. The minimization of  $\chi^2$  function by keeping fixed non-linear parameters is shown in the equation Eq. 6.

$$\frac{\partial \chi^2}{\partial \theta_j^L}(\vec{\theta}^L, \vec{\theta}^{NL}) = \dots = \frac{\partial \chi^2}{\partial \theta_r^L}(\vec{\theta}^L, \vec{\theta}^{NL}) = 0 \quad (6)$$

The resulting  $r$  linear equations can be written in a matrix form by the following expression:

$$(A^T \cdot A) \cdot \theta = A^T \cdot b \quad (7)$$

Where  $A \in \mathfrak{R}^{r \times M}$  such that  $A_{ij} = f_j(x_i; \vec{\theta}^{NL})$  and  $b \in \mathfrak{R}^{r \times 1}$  is such that  $b_i = \frac{y_i}{\sigma_i}$ . There are plenty of algorithms to invert the Matrix in equation 7. Nevertheless, some iterations could lead to problematic points when inverting equation 7. For these cases, the inverse of the matrix should be computed using its singular-value decomposition [23,24].

### Self-Cooling Simulated Annealing (SCSA) Algorithm

In the present algorithm, the temperature is regulated by an external bath which depends on the minimum  $\chi^2$  found in the given iteration  $k$ ,  $T_{\text{bath}} = T(\chi_{\text{min}}^2)$  with  $\chi_{\text{min}}^2 = \min(\chi_{1, \dots, \chi_k}^2)$ . At the beginning of the self-cooling procedure, the temperature of the system is probably high, since  $\chi^2$  can take large values when parameters are far away from the optimum solution. As the temperature selection will depend on the weight of the experimental data, we can consider using different functions in the temperature regulation process (see Appendix for some examples). In this work we simplified the problem to  $k_b T_{\text{bath}} = \chi_{\text{min}}^2$  but other functions may be tested (see Appendix for other options) depending on the data weight in equation Eq. 1. In order to avoid abrupt jumps in temperature, known as quenching, the inertia of the system to change temperature is included by eq. 8.

$$T_k = T_{k-1} + K \cdot (T_{\text{bath}} - T_{k-1}) \quad (8)$$

where,  $K \in (0, 1]$  represents the thermal conductivity of the system which ensure a smooth temperature decay, in other words smaller  $K$  values means larger resistance in analogy with a physical phenomena ( $K = 1/\Omega$  with  $\Omega$  the system resistance), however, quenching is given when  $K=1$ . Figure 1 shows the inertial response of the system when a change in the external bath temperature occurred. Notice that the equation 8 converges to  $\propto \exp(-K \cdot k)$  when  $K \rightarrow 0$ . The  $K$  constant must be chosen in function of the desired amount of simulation  $\Delta k$  to achieve the final temperature, were  $K \approx 5/\Delta k$ .

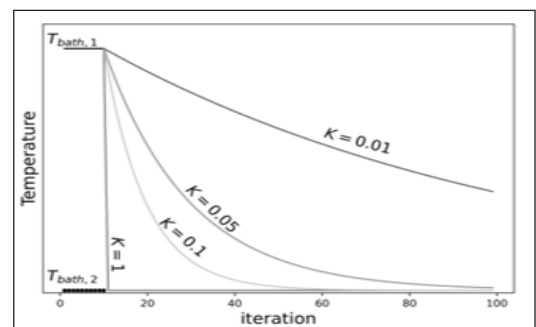
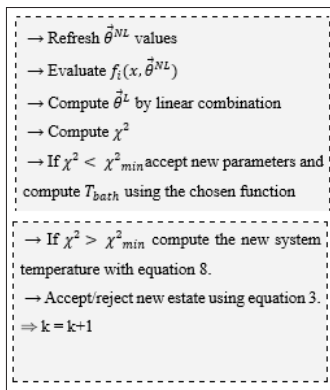


Figure 1: System' Temperature Decay with Different Conductivity Values "K"

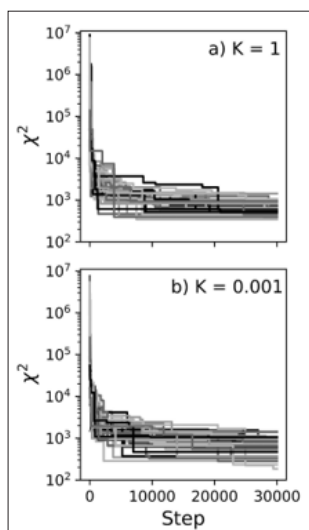
The Proposed Procedure is Detailed Pseudo-Algorithm at  $k$ -th Step



## Results

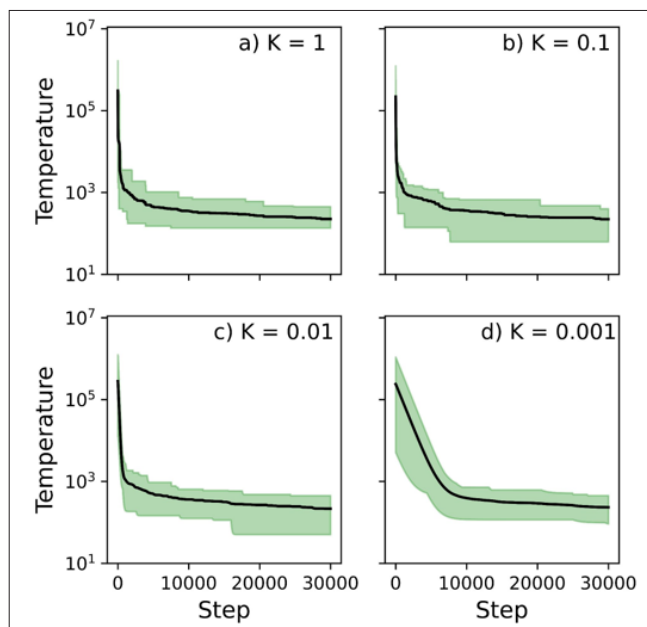
### Testing the Algorithm

To test the Algorithm, a simulated Small Angle X-ray Scattering (SAXS) pattern of a mix of lamellar bilayer and multilayer structures was employed (Eq. A1), the choice of the model was because it combines a series of trigonometric functions commonly found to test optimization algorithms(26). SAXS is a well-known technique to analyze several physicochemical and material science systems which usually employs highly nonlinear equations. However, the proposed algorithm is generic and it could be employed in a vast set of mathematical problems involving least squares procedures.. The algorithm was tested using Python 3.9 scripting and its libraries (Pandas, Numpy, Scipy, Seaborn and Matplotlib). In order to test the performance of the SCSA algorithm we employed a composition of non-linear models usually found in literature (see Supporting Information) with five nonlinear parameters and three linear parameters. Thirty nonlinear parameters initial values (seeds) were randomly chosen in a given range using a uniform distribution while parameters perturbation were done with an initial normal distribution with 50% of variance from the chosen range of interest (RI). The RI should be limited to values with physical meaning. For the experiments, different system thermal resistance (K) from 1 to 0.001 were tested to compute the temperature of the system with respect to the external thermal bath (Equation 8). Figure 2 shows the evolution of the  $\chi^2$  function for a fixed  $3 \times 10^4$  steps using  $K = 1$  and  $K = 0.001$  for thirty different seeds. For both experiments, the  $\chi^2$  drastically reduced its number during the first 2000-5000 steps while after that, changes were smoother [25].



**Figure 2:** Logscale  $\chi^2$  Evolution 30 Random Seeds with a)  $K = 1$  and b)  $K = 0.001$

For a deeper insight on the average behavior of the algorithm, the averaged system's temperature were plotted in logscale in function of the iteration step (Figure 3). Here, the line represents the mean value obtained from the tested seeds while the shadowing region represents the minimum and maximum values obtained at the  $k$ -th step ( $\Delta T_0$ ). In all cases the temperature decrease with steps, however, the decrease for  $K = 1$  (Fig. Figure 3a) was very big the first 5000 steps. The  $\Delta T_0$  becomes smoother with the decrease in the K value (Figure 3a to Figure 3d). For large K values (near 1) the system's temperature decreases abruptly and it may be trapped in a local minimum. Moreover, when temperature decreases gradually the probability to explore all possible states increases. To illustrate this concept, the acceptance rate is shown in Figure 4. Nevertheless, for simple problems with few minimas  $K = 1$  converge faster, therefore, low K values should be employed for more complex problems.

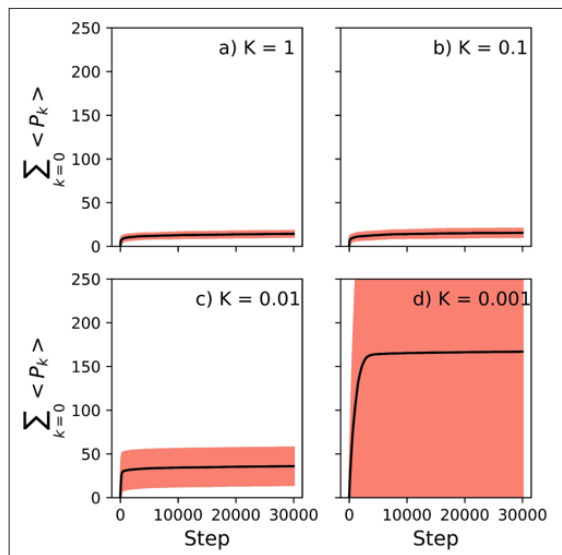


**Figure 3:** Logscale evolution of the averaged (black line) and the temperature range  $\Delta T_0$  defined between the minimum and maximum values found at  $k$ -th step (shaded region) of the system in function of parameter K a)  $K = 1$ , b)  $K = 0.1$ , c)  $K = 0.01$  and d)  $K = 0.001$

The acceptance rate was defined by the following expression

$$p = \sum_0^k \langle P(k|k-1) \rangle \quad (9)$$

Where  $\langle P(k|k-1) \rangle$  is the averaged probability to accept a change predefined in equation 4. This value gives a notion of the rate of the parameter acceptance during the algorithm. Nevertheless, the accepting rate will strongly depend on the data weight in equation 1.



**Figure 4:** Evolution of the Integrated Averaged Acceptance Rate P in Function of Parameter K a) K = 1, b) K = 0.1, c) K = 0.01 and d) K = 0.001

Figure 4 shows that the acceptance rate increased from K = 1 (Figure 4a) to K = 0.001 (Figure 4d). Moreover, the average plateau shifts to higher step values when the thermal resistance is reduced. Thus, the smooth temperate decay allows the system to explore a larger amount of states avoiding to be trapped in a local minimum. Under the current simulations, not only the plateau value in the averaged curves (continuous lines) reaches larger values for higher thermal resistance (smaller K values), but also it is reached at longer simulation steps.

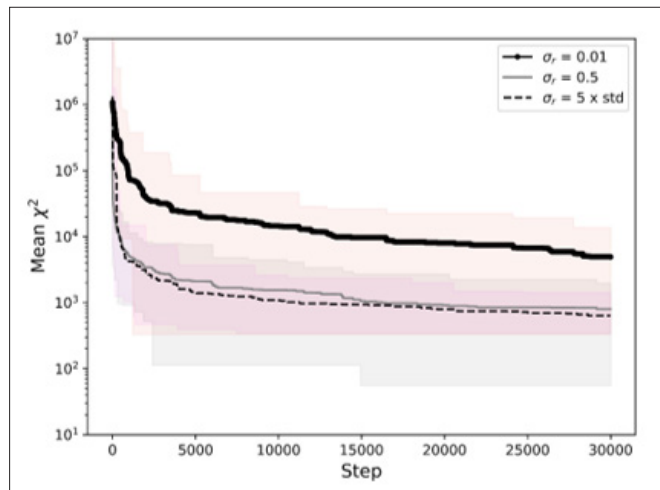
### Parameter Space Sampling

One of the issues for MC optimization is the starting guess seeds and the space of sampling. If this space is large the number of unproductive simulations will increase and with it the number of steps required to reach the minimum energy state. However, if the searching space is too narrow there is a possibility to fall into a local minima. Here, the updating of the Non-linear parameters was done with a random number generator with a normal distribution around the last accepted value according to the proposed algorithm (equation 3).

$$p_{k,j}^{NL} = p_{k,j_{opt}}^{NL} + \mathbf{N}(0, var_j) \quad (10)$$

Where  $\mathbf{N}(0, var_j)$  is the random number generator with a normal distribution with a variance  $var_j$ . Nevertheless, not all parameters showed the same impact over the figure of merit. Therefore, pre-assigned a searching area around a certain value will require previous information. In this work, a large variance was assigned then after a certain number of iterations the variance of the normal distribution of each parameter was updated taking five times the standard deviation from the last 5 to 10 accepted values. Thus, the sampling space was optimized, diminishing the computing cost. Figure 5 shows the logplot evolution of the  $\chi^2$  function vs step for a setted random gaussian distribution function for each nonlinear parameter compared with the one obtained with the standard deviation from the last 5 accepted values. Small variance means more steps to reach the optimum value, moreover, after 30,000 iterations the mean value was near one order of magnitude above the larger variance tested. However, the best results were

obtained by keeping the variance unfixed for each parameter. In the tested example the largest values were around 0.5 while the smallest near 0.01.



**Figure 5:** Mean  $\chi^2$  value in functions of the step for a set random distribution function reduced variance of 0.01, 0.5 and five times the standard deviation (std) obtained from the last five accepted values.  $\sigma_r = \sigma_j^{NL}/\theta_j^{NL}$

### Algorithm Performance

The SCSA algorithm was evaluated against commonly used optimization methods, including Monte Carlo with accept/reject (MCAR), gradient descent, and related approaches such as Newton-Raphson and Marquardt-Levenberg. To benchmark performance, we used a standard set of reference functions described in the Supporting Information, applied across models with varying levels of difficulty: two low-difficulty models (Figures s2 and s3), one medium-difficulty model (Figure s4), and one high-difficulty model (Figure s5). To ensure fair comparison in computation time, all algorithms were implemented in Python using standard libraries such as NumPy and SciPy. The tested methods included MCAR, SCSA1 ( $k = 1$ ) without thermal resistance, SCSA01 ( $k = 0.1$ ) with a preset thermal resistance, and SCSAV, which dynamically adjusts the variance of the Gaussian sampling based on search history, as defined by Equation 10. For SCSAV, the thermal resistance was also set to  $k=0.1$ . Gradient-based methods were also tested using random seed initializations: Gradient Descent (RSGD), Newton-Raphson (RSNR), and Marquardt-Levenberg (RSML). Performance was assessed using multiple metrics: the  $\chi^2$  goodness-of-fit, computation time normalized to the time registered from the slowest method, the Norm of Difference Between Estimated and True Parameters (NDETP), and the Norm of Difference Between the Mean of Estimated and True Parameters (NDEMP). All methods were tested using the same number of random seed initializations. For full metric definitions and experimental details, see Supporting Information [27-29].

The results show that gradient-based algorithms, particularly MCNR and MCLM, can achieve lower  $\chi^2$  values than other methods in low-complexity models. This indicates they can fit the data more precisely when the optimization landscape is relatively smooth. However, their performance declines sharply as complexity increases. These algorithms are prone to becoming trapped in local minima, especially in rugged parameter spaces with multiple attractors. This limitation leads to higher NDETP and NDEMP values, reflecting less accurate parameter estimates in more complex scenarios (see Tables S4, S6, S8, and S10).

In contrast, the Self-Cooling Simulated Annealing algorithms, SCSA1 and SCSA01, demonstrated robust and consistent performance across all levels of model complexity. Although they did not always achieve the lowest  $\chi^2$  values, they were more effective at avoiding local minima and consistently produced reliable global solutions. Their capacity for thorough exploration of the parameter space makes them particularly well-suited for complex optimization landscapes with multiple minima. On other hand, SCSAV, while capable of reaching acceptable  $\chi^2$  values, generally showed higher NDETP and NDEMP compared to SCSA1 and SCSA01. This is primarily due to the fact that some parameters contribute minimally to the  $\chi^2$  value, which allows for broader exploration in those dimensions. As a result, although the fit may appear adequate in terms of  $\chi^2$ , the estimated parameters can deviate more from the true values, especially in regions of low curvature in the figure of merit.

Computation time varied considerably across methods. Gradient descent methods like RSGD, RSNR, and RSML, though sometimes accurate, required significantly more time when multiple random seeds were used. These methods are faster only when initialized near optimal regions, which is rarely guaranteed in complex models. MCAR, while computationally efficient, was the least effective in reducing  $\chi^2$ .

## Discussion

The Self-Cooling Simulated Annealing (SCSA) algorithm offers notable advantages over traditional least-squares approaches for data analysis—and its flexibility makes it suitable for a broader range of optimization problems. In this method, the system temperature is dynamically adjusted based on the best minimum energy state found. Since initial seeds are typically far from the optimal solution, the algorithm starts at high temperatures, resulting in a high acceptance rate during the early steps (Figure 4). Although local minima may be encountered early, the gradual temperature decay allows the system to retain enough energy to escape them, encouraging deeper exploration of the parameter space (Figure 3). This strategy helps prevent premature convergence and improves the likelihood of locating global optima.

When multiple equivalent minima are present, users can select the one most aligned with the physical or experimental context by running a large number of seeds. In our case-study, we used  $3 \times 10^4$  steps, and the resulting average NL parameters values closely approached the optimal solutions (see Supporting Information), even with variations in the resistance parameter  $K$ . For simpler problems, a high  $K$  value (e.g.,  $K = 1$ ) ensures rapid cooling and convergence. In contrast, more complex problems benefit from slower temperature decay. However, smaller  $K$  values typically require more steps to achieve convergence.

Parameter correlation should be considered when applying constraints or restraints, as it can significantly influence optimization outcomes. The adaptive sampling mechanism of the SCSA algorithm reduces computation time by rapidly lowering the  $\chi^2$  value. However, this speed can increase the risk of becoming trapped in local minima. To address this, a variation of the method—referred to here as SCSAV—uses Gaussian random sampling with a dynamic (moving) variance for each parameter, as defined in Equation 10. In this approach, the variance is recalculated using the standard deviation from the last 5 to 10 accepted values, rather than the full distribution. This enhances sampling efficiency and maintains a flexible exploration range, which is especially beneficial for systems with many nonlinear parameters.

While SCSAV occasionally achieves lower  $\chi^2$  values compared to other methods, it generally yields higher NDETP and NDEMP values than SCSA01 and SCSA1. This is because the probability of reaching the global minimum depends on the normalized Gaussian function  $\prod N(p_{j, \text{opt}}^{\text{NL}} - p_{j, \text{true}}, \text{var}_j)$ , where  $p_{j, \text{true}}$  is the true parameter value and  $\text{var}_j$  is the adaptive variance. In problems where the curvature of the figure of merit function  $\partial^2(\chi^2)/\partial(p_{j, \text{opt}}^{\text{NL}})^2$  is steep (i.e., high convexity), the algorithm converges quickly to the true solution. In contrast, when convexity is low, the algorithm explores a broader parameter space, which increases the likelihood of larger deviations from the true parameters, leading to higher NDETP and NDEMP. Nevertheless, this trade-off allows SCSAV to adapt the sampling range dynamically based on the problem structure, offering flexibility in balancing exploration and convergence.

While gradient methods may offer faster run times with a favorable start, their efficiency drops sharply when multiple seeds are needed to ensure reliability. SCSA algorithms require fewer seeds to achieve similar or superior reliability, yielding favorable normalized computation times (Supporting Information). Additionally, interpreting NDETP and NDEMP as measures of parameter deviation from true values underscores that lower values (as achieved by SCSA) strongly indicate better global convergence and reduced seed-dependence. Overall, the SCSA algorithm strikes an effective balance between computational efficiency and optimization accuracy. Its dynamic temperature regulation allows it to escape local minima more readily, helping reduce both NDETP and NDEMP values. This makes the estimated parameters not only more accurate individually but also more reliable on average. While gradient descent-based methods can achieve excellent fits—particularly in simple cases such as Gaussian peak fitting—their tendency to get stuck in local minima and their sensitivity to initial conditions make them less reliable for complex optimization landscapes. In contrast, SCSA provides more stable and consistent results across varying levels of problem complexity, making it a strong candidate for nonlinear least-squares data analysis and beyond.

## Conclusions

The Self-Cooling Simulated Annealing (SCSA) algorithm presented herein introduces a dynamic temperature adjustment mechanism based on the best minimum found at each iteration. This approach promotes a high initial acceptance rate and allows the system to progressively focus on regions of the parameter space with improved solutions. The cooling rate is governed by a user-defined thermal resistance parameter, enabling flexible adaptation to problems of varying complexity.

The algorithm also employs a decoupling of linear and nonlinear parameters, effectively reducing the dimensionality of the Monte Carlo search and significantly improving computational efficiency. Additionally, the sampling distribution is autoregulated through variance updates derived from recent accepted values, allowing for more targeted exploration of the parameter space.

Benchmarking results indicate that SCSA delivers robust performance across a range of optimization challenges, particularly in complex or multimodal landscapes where traditional gradient-based and Monte Carlo methods tend to underperform. While the method was applied here to nonlinear least-squares data analysis, its core principles are generalizable to other inverse and optimization problems. The algorithm's ability to balance exploration and convergence positions it as a competitive and versatile tool in scientific computing.

## Acknowledgements

This work was funded by ANPCyT (PICT 2017-3150). CHI and LA are researchers of CONICET, Argentina.

## References

1. Li Y (2022) Mathematical Modeling Methods and their Application in the Analysis of Complex Signal Systems. *Adv Math Phys* <https://onlinelibrary.wiley.com/doi/10.1155/2022/1816814>.
2. Velten K, Schmidt DM, Kahlen K (2024) Mathematical Modeling and Simulation: Introduction for Scientists and Engineers <http://content.e-bookshelf.de/media/reading/L-24529340-643b322082.pdf>.
3. Feller W (1991) An Introduction to Probability Theory and Its Applications, <https://bitcoinwords.github.io/assets/papers/an-introduction-to-probability-theory-and-its-applications.pdf>.
4. Fernández-Martínez JL, Fernández-Muñiz Z, Pallero JLG, Pedruelo-González LM (2013) From Bayes to Tarantola: New insights to understand uncertainty in inverse problems. *J Appl Geophys* 98:62-72.
5. Tarantola A, Valette B (1981) Inverse problems = Quest for information. *J Geophys* 50:159 - 70.
6. (2024) University "St Kliment Ohridski"- Bitola, Faculty of Technology and Technical Science -Veles, 7000, Bitola, Republic of North Macedonia, Knights V, Prchkovska M, Mother Teresa University, Faculty of Computer Science, Informatics, 1000, Skopje, Republic of North Macedonia. From Equations to Predictions: Understanding the Mathematics and Machine Learning of Multiple Linear Regression. *J Math Comput Appl* 30: 1-8.
7. Leskovec J, Rajaraman A, Ullman JD (2020) Mining of Massive Data Sets. Cambridge University Press <http://infolab.stanford.edu/~ullman/mmds/book.pdf>.
8. Pelikan M, Goldberg DE, Cantu-Paz E (1999) BOA: The Bayesian Optimization Algorithm. 1: 525-532.
9. Pan WT (2012) A new Fruit Fly Optimization Algorithm: Taking the financial distress model as an example. *Knowl-Based Syst* 26: 69-74.
10. Björck A (1990) Least squares methods. In: Handbook of numerical analysis 110: 465-652.
11. Pedersen JS (1997) Analysis of small-angle scattering data from colloids and polymer solutions: modeling and least-squares fitting I. *Adv Colloid Interface Sci* <https://www.sciencedirect.com/science/article/abs/pii/S0001868697003126>.
12. Fayed HA, Atiya AF (2019) Speed up grid-search for parameter selection of support vector machines. *Appl Soft Comput* [https://www.researchgate.net/publication/331942512\\_Speed\\_up\\_gridsearch\\_for\\_parameter\\_selection\\_of\\_support\\_vector\\_machines](https://www.researchgate.net/publication/331942512_Speed_up_gridsearch_for_parameter_selection_of_support_vector_machines).
13. Ruder S (2017) An overview of gradient descent optimization algorithms. <http://arxiv.org/abs/1609.04747>.
14. Marquardt DW (1963) An Algorithm for Least-Squares Estimation of Nonlinear Parameters. *J Soc Ind Appl Math* 11: 31- 41.
15. Levenberg K (1944) A method for the solution of certain nonlinear problems in least squares. *Q Appl Math* <https://www.scrip.org/reference/referencespapers?referenceid=1987558>.
16. Hogg DW, Foreman-Mackey D (2018) Data Analysis Recipes: Using Markov Chain Monte Carlo. *Astrophys J Suppl Ser* <https://iopscience.iop.org/article/10.3847/1538-4365/aab76e>.
17. Zupanski M, Navon IM, Zupanski D (2008) The Maximum Likelihood Ensemble Filter as a non-differentiable minimization algorithm. *Q J R Meteorol Soc* <https://people.sc.fsu.edu/~inavon/nsf/Qj162.pdf>.
18. Metropolis N, Rosenbluth AW, Rosenbluth MN, Teller AH, Teller E (1953) Equation of State Calculations by Fast Computing Machines. *J Chem Phys* <https://bayes.wustl.edu/Manual/EquationOfState.pdf>.
19. Kirkpatrick S, Gelatt CD, Vecchi MP (1983) Optimization by Simulated Annealing. *Science* <https://www.2.stat.duke.edu/~scs/Courses/Stat376/Papers/TemperAnneal/KirkpatrickAnnealScience1983.pdf>.
20. Bertsimas D, Nohadani O Robust (2010) optimization with simulated annealing. *J Glob Optim* 48: 323-34.
21. da Silva Costa D, Huck-Iriart C, Kellermann G, Giovanetti LJ, Craievich AF et al. (2015) In situ study of the endotaxial growth of hexagonal CoSi<sub>2</sub> nanoplatelets in Si(001). *Appl Phys Lett* 30: 107-223.
22. Golub G, Pereyra V (2003) Separable nonlinear least squares: the variable projection method and its applications. *Inverse Probl* 19: R1-26.
23. van de Geer SA (2005) Least Squares Estimation. In: Encyclopedia of Statistics in Behavioral Science <https://onlinelibrary.wiley.com/doi/abs/10.1002/0470013192.bsa199>.
24. Golub GH, Reinsch C (2022) Singular Value Decomposition and Least Squares Solutions. In: Bauer FL, editor. *Linear Algebra* [http://link.springer.com/10.1007/978-3-662-39778-7\\_10](http://link.springer.com/10.1007/978-3-662-39778-7_10).
25. Altube MJ, Caimi LI, Huck-Iriart C, Morilla MJ, Romero EL (2021) Reparation of an Inflamed Air-Liquid Interface Cultured A549 Cells with Nebulized Nanocurcumin. *Pharmaceutics* <https://pubmed.ncbi.nlm.nih.gov/34575407/>.
26. Andrei N (2008) An Unconstrained Optimization Test Functions Collection. In: *Advanced Modeling and Optimization* <https://camo.ici.ro/journal/vol10/v10a10.pdf>.
27. Gratton S, Lawless AS, Nichols NK (2007) Approximate Gauss-Newton Methods for Nonlinear Least Squares Problems. *SIAM J Optim* <https://epubs.siam.org/doi/10.1137/050624935>.
28. Filliben J, Possolo A (2025) Statistical Reference Datasets, NIST Standard Reference Database 140. National Institute of Standards and Technology <http://www.itl.nist.gov/div898/strd/>.
29. StRD Nonlinear Least Squares Regression Datasets [https://www.itl.nist.gov/div898/strd/nls/nls\\_main.shtml](https://www.itl.nist.gov/div898/strd/nls/nls_main.shtml).

**Copyright:** ©2025 Cristián Huck-Iriart, et al. This is an open-access article distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.