

Optimus User Manual

Nicholas Andre G. Johnson and Aleksandr B. Sahakyan

16 August 2018

Introduction

For a complex model, where the unknown parameters cannot be determined by conventional linear or non-linear fitting techniques, Monte Carlo methods for optimisation, based on the random sampling of the parameter space, is the method of choice. In such a case, the quality of the solution found following some optimisation protocol depends on that protocol's ability to effectively explore the parameter space. Many existing methods perform well for certain models, but fail in other models due in part to inefficient exploration of the parameter space. In this manual, we present Optimus, a Universal Monte Carlo optimisation engine in R with acceptance ratio annealing, replica exchange and adaptive thermoregulation. It can universally interface with any model definition and efficiently optimise the model's parameters by consistently exploring the parameter space effectively. Optimus can execute either a simulated annealing procedure or a replica exchange procedure, depending on the desires of the user.

This User Manual will begin with a brief overview of Monte Carlo optimisation and the common temperature simulated annealing framework. It will then proceed with a presentation of Optimus' acceptance ratio annealing procedure, its adaptive thermoregulation feature, and its replica exchange procedure. After an explanation of how to download the Optimus R package from GitHub and install it locally, 4 Tutorials will be presented to illustrate how users should employ Optimus and to demonstrate its flexibility as an optimisation engine. Finally, an "Advanced User Manual" section will be included in which all possible input parameters to Optimus will be outlined and the output format will be specified.

Briefly on Monte Carlo and Temperature Simulated Annealing Procedures

Let us assume that our model is a certain function $m()$ that performs operations on the inputted K coefficient set and returns the observable object $O = m(K)$. Our task is to optimise K , the set of coefficients, so that the error metric $u(O)$ that measures the violations of the target O^{trg} set by the model-generated O is minimal. In a Monte Carlo optimisation procedure, we can define a pseudo-energy $e(u)$ of the system as a function of $u()$, where lower values of the pseudo-energy e correspond to better candidate solutions K . In order to find a better set of K , we need to alter it by a certain rule $r()$ that, if repeated many times, enables the sampling of the parameter space for K . One can then evaluate the pseudo-energies before and after the alterations:

$$\begin{aligned} E_1 &= e(u(m(K))) \\ E_2 &= e(u(m(r(K)))) \end{aligned}$$

We then accept or reject the move, meaning we accept the new set of $r(K)$ coefficients as the new K or revert back to the previous K , guided by the following acceptance probability, as postulated in a Metropolis criterion:

$$\begin{aligned} p_{accept} &= \min(1, e^{\frac{-\Delta E}{T}}) \\ \Delta E &= E_2 - E_1 \end{aligned}$$

where T is the pseudo temperature, that should be always greater than 0. For a given $\Delta E > 0$ energy difference, one would have different stringency for accepting the move depending on the value of the pseudo-temperature T . Therefore, in case the Monte Carlo simulation were to drive the K set to a state where any further moves would increase the pseudo-energy great enough for the moves to be always rejected, then

one could overcome that and further sample the other values in the parameter space by increasing the pseudo-temperature.

In molecular dynamics simulation, where the objective is to sample the free energy surface populated by a molecule, one way to overcome the barriers is by using the technique known as simulated annealing, where we anneal the temperature gradually from some higher value to a lower value during the course of the simulation. The parts of the simulation where the pseudo-temperature is higher allows relatively unconstrained exploration of the search space whereas those parts with a lower pseudo-temperature limit the search to a more local area of the parameter space. Multiple cycles of this annealing procedure can be executed to increase the overall sampling. In the same manner, we can anneal the pseudo-temperature in the Monte Carlo procedures for model optimisation.

Acceptance Ratio Annealing Procedure

A significant limitation of pseudo-temperature simulated annealing is that a given scheme of temperature annealing might be efficient for some models or pseudo-energy metrics, but not efficient for others. A temperature at a given point of the annealing cycle that is designed to be quite permissive in terms of accepting the move, can actually not be permissive depending on the value and scale of the ΔE energy difference, as can be seen in the equation for p_{accept} . Furthermore, even within the single model optimisation procedure, the pseudo-energy metric can shift into a value range that does not match with the selected temperature scheme anymore, leading to similar problems (meaning poor sampling of the parameter space). This can often be the case when the pseudo-energy of the system does not exhibit a smooth dependency on K , loosely meaning that similar values of K do not necessarily produce similar pseudo-energy values.

To this end, in general cases where we do not deal with energies and temperatures that display the smoothness emulating real physical systems (such smoothness is often found in molecular dynamics or Monte Carlo simulations of molecules), we need to anneal a more robust metric for crossing different barriers. As such a metric, Optimus, when in its default mode, uses the acceptance ratio.

In a given annealing cycle, Optimus constructs a linear target acceptance ratio schedule for each step based on an initial target acceptance ratio, a final acceptance ratio and the number of iterations in each cycle for a given optimisation run (all of which can be specified as inputs). Once the optimisation process begins, Optimus calculates an observed acceptance ratio at the end of each STATWINDOW (a fixed number of steps which can be specified by the user) by calculating the fraction of the accepted moves from all the past trials in the current STATWINDOW. Thereafter, Optimus compares the observed acceptance ratio with the target acceptance ratio based on the annealing schedule and determines whether and how to alter the system pseudo-temperature (adaptive thermoregulation) to align the observed acceptance ratio with the target ratio at the end of the following STATWINDOW. Thus, by employing acceptance ratio annealing and adaptive thermoregulation, Optimus is able to methodically explore the parameter space for K even when no smooth relationship exists between K and the system pseudo-energy.

Adaptive Thermoregulation

All decisions governing system pseudo-temperature alterations are made by a Temperature Control Unit (TCU) that interfaces directly with Optimus on the backend (note that the TCU is completely encapsulated such that modifications can be easily made in future should they be needed). This section articulates the exact protocol followed by the current TCU.

The initial system temperature is specified as an input argument. At the end of each STATWINDOW, if the observed acceptance ratio is within a fixed value T.DELTA (specified as an input argument) of the target acceptance ratio based on the annealing schedule, the TCU will make no change to the current system pseudo-temperature. If the observed acceptance ratio is less than the ideal ratio and outside the range of T.DELTA, the TCU will increase the system pseudo-temperature by a value T.ADJUST (the initial value of

T.ADJSTEP is specified as an input argument). Similarly, if the observed acceptance ratio is greater than the ideal ratio and outside the range of T.DELTA, the TCU will reduce the temperature by a value T.ADJSTEP.

If the observed acceptance ratio has been below the ideal acceptance ratio for TSCLnum (an integer input argument) subsequent STATWINDOWS, T.ADJSTEP will be increased by a factor T.SCALING (an input argument). Similarly, T.ADJSTEP will also be increased by a factor T.SCALING if the observed acceptance ratio is greater than the ideal acceptance ratio for TSCLnum subsequent STATWINDOWS. T.ADJSTEP is reset to its original input value whenever a series of subsequent observed acceptance ratios being greater than/less than ideal acceptance ratios is broken. If ever the TCU subtracts T.ADJSTEP from the current temperature and the result is a negative value, the system pseudo-temperature is set to T.MIN (an input argument). The final feature of the TCU is that although the initial system pseudo-temperature is specified by the user, if multiple annealing cycles are employed, the initial pseudo-temperature for acceptance ratio annealing cycles after the first cycle is inferred from the decisions of the TCU on previous cycles.

This collection of decision rules that comprise the TCU result in pseudo-temperature alterations that cause the observed acceptance ratios during Optimus optimisation runs to follow the ideal acceptance ratios remarkably well. Moreover, as will be highlighted in the Tutorials, large temperature alterations are often required to align the observed acceptance ratios with the ideal ratios, a task which Optimus excels at whereas other protocols would have difficulty.

Replica Exchange Procedure

Optimus additionally supports replica exchange as an optimisation mode which can be selected in place of acceptance ratio annealing, provided that the user has access to multiple processors (ideally at least 4, and preferably 8 or more). The inspiration for this additional mode was taken from Replica Exchange Molecular Dynamics (REMD) simulations. REMD simulation is a technique employed to obtain equilibrium sampling of a molecule (for instance, a new protein whose properties one desires to characterize), usually at low temperatures. Let $T = \{T_1, T_2, \dots, T_n\}$ be a set of n distinct temperatures for which $T_1 < T_2 < \dots < T_n$. In REMD, n replicas of Monte Carlo simulation for a given molecule are initialized at each $T_i \in T$. Note that each temperature T_i corresponds to a slightly different energy landscape for the examined molecule. A key fact that underlies REMD simulations is that many configurations in a given energy landscape corresponding to T_i are likely also found in the energy landscapes corresponding to T_{i-1} and T_{i+1} . If molecule configurations in adjacent replicas are allowed to exchange, the simulation will be able to overcome energy barriers at the various temperature replicas and thoroughly explore the parameter space. Moreover, for molecule configuration x_n in replica T_i and configuration x_m in replica T_{i+1} , it has been proven that equilibrium sampling will occur if an exchange happens with the following probability (Sugita and Okamoto 1999):

$$p_{REMD} = \min(1, e^{-\Delta})$$

$$\Delta = (T_{i+1} - T_i)(E(x_n) - E(x_m))$$

where $E(x)$ represents the energy associated with the molecule configuration x . Thus, REMD simulation can be executed by repeating the following two steps for a chosen number of iterations (Sugita and Okamoto 1999):

- 1) Simulate n replicas of Monte Carlo Optimisation at distinct temperatures simultaneously and independently
- 2) Randomly select two configurations in adjacent replicas and exchange them with probability p_{REMD}

Optimus extends this approach to apply to arbitrary optimisation problems with two primary modifications. Firstly, due to the aforementioned robustness of utilizing acceptance ratio as a controlling metric rather than pseudo-temperature in arbitrary systems, Optimus initializes n replicas with different target acceptance ratios as opposed to different temperatures and uses the previously described TCU for adaptive thermoregulation. Secondly, given that Optimus is only concerned with finding an Optimal solution and is not concerned with equilibrium sampling of the parameter space (as equilibrium sampling has no meaning for non physical

systems), after two candidate configurations are selected for exchange, they are necessarily exchanged (this can be viewed as setting $p_{REMD} = 1$ in the above procedure). By relaxing the equilibrium sampling criterion, the parameter space can be more extensively explored. This approach produced good results (as illustrated in the tutorials) and is a viable alternative to the acceptance ratio annealing mode of Optimus when the user has access to good computing resources.

Installation Instructions

Installing Optimus locally for immediate use requires only an R client and a connection to the internet. After opening an R client, execute the commands below to install Optimus. Note that the latest version of Rtools is required for this installation to work. If it is not available locally and the installation is attempted with RStudio, a prompt will appear to download and install Rtools. After following those instructions, restart the RStudio session before reattempting the Optimus installation.

```
install.packages("devtools")      # install devtools
library(devtools)                 # load devtools
install_github("SahakyanLab/Optimus") # install Optimus
library(Optimus)                  # load Optimus
```

References

Sugita, Yuji, and Yuko Okamoto. 1999. "Replica-Exchange Molecular Dynamics Method for Protein Folding." *Chemical Physics Letters* 314 (November): 141–51. doi:10.1016/S0009-2614(99)01123-9.