Simulated annealing

Juan Antonio Pérez Juárez
University of Guadalajara
Guadalajara, Jalisco
juan.perez0996@alumnos.udg.mx

I. INTRODUCTION

Simulated annealing (SA) is a widely used probabilistic optimization technique that has proven effective in solving complex optimization problems across various fields, including operations research, artificial intelligence, and engineering design. Its inherent flexibility allows it to tackle a diverse array of challenges, making it a valuable tool for researchers and practitioners alike.

The algorithm's ability to escape local optima while searching for global solutions positions as a competitive alternative to traditional optimization methods. This paper explores the theoretical foundations of simulated annealing, its algorithmic framework, and its applications in real-world scenarios. By examining the strengths and limitations of SA, we aim to provide insights into its potential for addressing contemporary optimization challenges.

II. ORIGIN

Simulated annealing actually has its origins in metallurgy. In metallurgy, annealing refers to the process of heating metal to a high temperature and then slowly cooling it in a controlled environment. The high heat gives the atoms in the metal the freedom to move around wildly. If the metal were to be cooled quickly, the atoms would suddenly come to rest wherever they were when the metal was hot, resulting in a random arrangement and a poor-quality result. But when the metal is cooled slowly, the atoms have time to gradually find the best possible orientation and align themselves into a nice crystal lattice. This is preferable to quick cooling; it makes the metal more ductile, reduces defects, and makes it significantly stronger.

The simulation can be performed either by a solution of kinetic equations for probability density functions, or by using a stochastic sampling method. The method is an adaptation of the Metropolis–Hasting's algorithm, a Monte Carlo method to generate sample states of a thermodynamic system.

III. OVERVIEW

The state s of some physical systems, and the function E(s) to be minimized, is analogous to the internal energy of the system in that state. The goal is to bring the system, from an arbitrary initial state s to a state with the minimum possible energy.

A. Basic Iteration

At each step, the simulated annealing heuristic considers some neighboring state s* of the current state s, and probabilistically decides between moving the system to state s* or staying in state s. These probabilities ultimately lead the system to move to states of lower energy. Typically this step is repeated until the system reaches a state that is good enough for the application, or until a given computation budget has been exhausted.

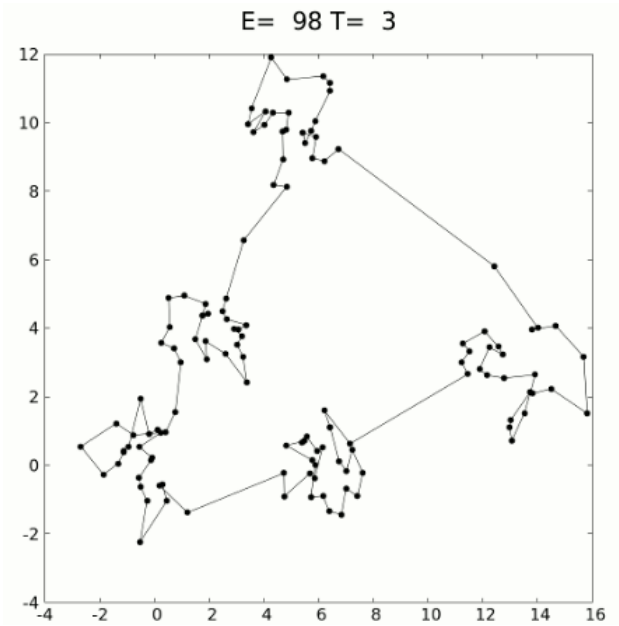

Fig. 1 Simulated annealing can be used to solve combinatorial problems, is applied to the travelling salesman problem to minimize the length of a route that connects all 125 points.

B. The neighbors of a state

Optimization of a solution involves evaluating the neighbors of a state of the problem, which are new states produced through conservatively altering a given state. For example, in the traveling salesman problem each state is typically defined as a permutation of the cities to be visited, and the neighbors of any state are the set of permutations produced by swapping any two of these cities. The well-defined way in which the states are altered to produce neighboring states is called a "move", and different moves give different sets of neighboring states. These moves usually result in minimal alterations of the last state, to progressively improve the solution through iteratively improving its parts (such as the city connections

in the traveling salesman problem). It is even better to reverse the order of an interval of cities.

C. Acceptance probabilities

The probability of making the transition from the current state s to a candidate new state s_{new} specified by an acceptance probability function $P(e, e_{new}, T$ that depends on the energies $e = E(s)\ and\ e_{new} = E(s_{new})$ of the two states, and on a global time-varying parameter T called the temperature. States with smaller energy are better than those with greater energy. The probability function P must be positive even when e_{new} is greater than e. This feature prevents the method from becoming stuck at a local minimum that is worse than the global one.

When T tends to zero, the probability $P(e, e_{\text{new}}, T)$ must tend to zero if $e_{\text{new}} > e$ and to a positive value otherwise. For sufficiently small values of T the system will then increasingly favor moves that go "downhill" and avoid those that go "uphill." With T=0 the procedure reduces to the greedy algorithm, which makes only the downhill transitions.

D. The annealing schedule

Simulated annealing is an optimization algorithm inspired by temperature variation in physical annealing processes. The algorithm starts with a high initial temperature, which is gradually reduced according to a user-defined schedule, ending at zero. This cooling process allows the algorithm to first explore the search space broadly and then focus on increasingly optimal regions, eventually settling into a local minimum using steepest descent. While theoretically, extending the annealing schedule increases the chance of finding a global optimum, in practice, the required time often exceeds that of a complete search, limiting its practical usefulness for guaranteed optimal solutions.

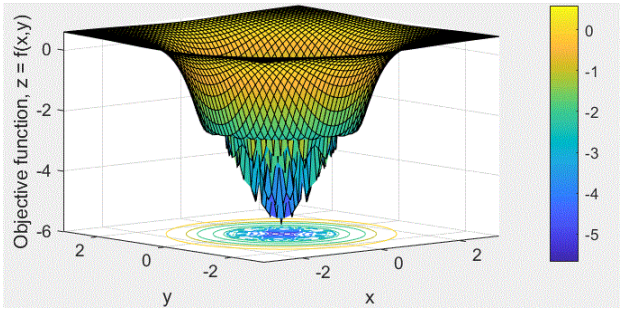

Fig. 2 Performance Test of Simulated Annealing Algorithm to Search for Global Optima of a Surface Function.

IV. SELECTING THE PARAMETERS

Simulated annealing has established itself as a powerful optimization technique, effectively addressing complex problems across various domains. By mimicking the physical process of annealing, this algorithm adeptly navigates the trade-off between exploration and exploitation, allowing it to escape local optima and converge toward global solutions. The adaptability of simulated annealing makes it suitable for a diverse range of applications, from combinatorial optimization to machine learning.

A. Transition probabilities

To investigate the behavior of simulated annealing on a particular problem, it can be useful to consider the transition probabilities that result from the various design choices made in the implementation of the algorithm. For each edge (s, s') of the search graph, the transition probability is defined as the probability that the simulated annealing algorithm will move to state s′ when its current state is s. This probability depends on the current temperature as specified by temperature(), on the order in which the candidate moves are generated by the neighbour() function, and on the acceptance probability function P(). (Note that the transition probability is not simply $P(e, e', T)$, because the candidates are tested serially.)

B. Acceptance probabilities

The acceptance probabilities in simulated annealing depend on three main functions: the neighbor function, the acceptance probability function, and the temperature schedule. Typically, the same acceptance probability function is used for various problems, while the other two are adjusted as needed.

Kirkpatrick et al. originally defined the acceptance probability so that moves to lower energy states are always accepted, and moves to higher energy states are accepted with a probability that decreases exponentially with the energy difference and temperature. This formula was based on analogies to physical systems and the Metropolis–Hastings algorithm. However, in simulated annealing, the neighbor function is often neither symmetric nor probabilistic, meaning the algorithm's transitions don't match those of a physical system, and its equilibrium distribution doesn't resemble thermodynamic equilibrium. Despite this, most implementations still use the original acceptance probability function.

C. Efficient candidate generation

When designing the candidate generator (neighbor function) for simulated annealing, it's important to consider that, after several iterations, the current state will typically have much lower energy than a random state. Therefore, it's generally best to generate candidate moves where the destination state's energy is similar to the current state's. This heuristic, which is central to the Metropolis-Hastings algorithm, helps avoid both extremely good and extremely bad moves—though very good moves are rare, so the heuristic remains effective.
More precisely, the heuristic suggests prioritizing candidate states where the acceptance probability is high. With the standard acceptance function, this means the energy difference between the current and candidate states

should be about the same as the temperature or less. In the traveling salesman example, this could mean swapping two random cities, but with a decreasing probability as their distance increases beyond the temperature value.

D. Barrier avoidance

When designing the candidate generator (neighbour function) for simulated annealing, you should also aim to reduce the number of "deep" local minima states or groups of states with much lower energy than their neighbors. These "closed catchment basins" can trap the algorithm for a long time, with the trapping probability and duration increasing with the basin's depth and size.

It is generally impossible to create a candidate generator that both avoids deep local minima and always prioritizes candidates with similar energy. However, even simple changes to the generator can significantly improve the efficiency of simulated annealing.

E. Cooling schedule

The cooling schedule in simulated annealing is inspired by a physical analogy, where a system is cooled slowly enough to maintain thermodynamic equilibrium. In theory, this means the probability distribution of the current state should always be close to equilibrium as the temperature decreases. However, in practice, the relaxation time the period required for the system to return to equilibrium after a temperature change varies greatly depending on the landscape of the energy function, the current temperature, and the candidate generator used to explore new states. These factors interact in complex ways and are typically treated as black box functions within the algorithm. As a result, it is impossible to determine the ideal cooling rate beforehand; instead, it must be empirically tuned for each specific problem. To address this challenge, adaptive simulated annealing algorithms have been developed, which adjust the cooling schedule dynamically based on the progress of the search. Other adaptive approaches, such as Thermodynamic Simulated Annealing, automatically set the temperature at each step according to the energy difference between states, following the laws of thermodynamics. This allows the algorithm to respond more effectively to the problem's unique characteristics and improve optimization performance.

F. Restarts

Sometimes it is better to move back to a solution that was significantly better than always moving from the current state. This process is called restarting of simulated annealing. To do this we set s and e to sbest and ebest and perhaps restart the annealing schedule. The decision to restart could be based on several criteria. Notable among these include restarting based on a fixed number of steps, based on whether the current energy is too high compared to the best energy obtained so far, restarting randomly, etc.

X. CONCLUSION

Simulated annealing has established itself as a powerful optimization technique, effectively addressing complex problems across various domains. By mimicking the physical process of annealing, this algorithm adeptly navigates the trade-off between exploration and exploitation, allowing it to escape local optima and converge toward global solutions. The adaptability of simulated annealing makes it suitable for a diverse range of applications, from combinatorial optimization to machine learning.

Despite its strengths, simulated annealing is not without limitations. The choice of cooling schedule and parameters significantly influences its performance, and finding the optimal configuration often requires empirical tuning. Additionally, while SA is robust, it may not always guarantee convergence to the best possible solution within a reasonable timeframe, particularly for highly complex or large-scale problems.

Future research directions could focus on enhancing the efficiency of simulated annealing through hybridization with other optimization techniques or the development of adaptive cooling schedules. Furthermore, exploring its integration with emerging technologies, such as quantum computing, could open new avenues for solving previously intractable problems.

In summary, simulated annealing remains a vital tool in the optimization landscape, offering a balance of simplicity and effectiveness. Its continued evolution and adaptation will likely ensure its relevance in tackling the increasingly complex challenges faced in various scientific and engineering fields.

REFERENCES

[1] Wikipedia contributors, "Simulated annealing," Wikipedia, Nov. 11, 2025. https://en.wikipedia.org/wiki/Simulated_annealing

[2] Macromoltek, Inc., «Machine learning and simulated annealing», Medium, 16 de agosto de 2018. https://macromoltek.medium.com/machine-learning-and-simulated-annealing-588b2e70d0cc (accedido 20 de noviembre de 2025).

[3] "(10) Performance test of simulated annealing algorithm to search global optima of a surface function | LinkedIn." https://www.linkedin.com/pulse/performance-test-simulated-annealing-algorithm-search-sarkar/.