Solving Sudoku with Simulated Annealing

Pia Jakobus, Nicklas Riebsamen and Luke Ewig

Frankfurt Institute for Advanced Studies

Abstract

Simulated Annealing a minimization technique closely related to the Metropolis algorithm. Here, it is applied to the logic-based number-placement puzzle Sudoku. Starting from a random guess of the solution, it is possible to determine the correct solution by gradually selecting two numbers and exchanging them with a certain probability.

1. Metropolis algorithm and Simulated Annealing

Simulated annealing deals with finding a global minimum of a function E defined on some set S. In the context of Simulated Annealing, S is the state space and each $x \in S$ is a state of the system. $E(x)$ then is interpreted as the energy of the system in state x. According to the Boltzmann statistics, in thermodynamic equilibrium at temperature T the probability of observing energy $E(x)$ is given by

$$p(E(x)) \propto \exp\left(-\frac{E(x)}{T}\right).$$

In the usual Metropolis algorithm one defines a Markov chain on the state space by (1) determining a criterion for neighborhood of two states x and y and (2) transitioning from a state x to a random neighboring state y with probability

$$p_{xy} = \min\left(1, \exp\left(-\frac{\Delta E}{T}\right)\right) \qquad \Delta E = E(y) - E(x) \tag{1}$$

and staying in the state x otherwise. It then is guaranteed that iterating the Markov chain long enough results in a sample from the Boltzmann distribution. This is irrespective of the initial state.

In Simulated Annealing, the temperature T is lowered over time. This “cooling” of the system has the following intuition: Assuming discrete states and a single global minimum of E, for $T = 0$ the Boltzmann distribution has the form $p(E = E_{\min}) = 1$ and 0 else. By lowering the temperature, the probability of observing the state with the lowest energy rises.

2. Sudoku

	1	5			8		6	2
4		2				9		5
	7				3			
1				9				
					5	6	8	9
	3	8		6				7
	9			8	6	5	7	1
5	6		1		2	4		
			7			3		

Figure 1: Sudoku game

Sudoku is a logic-based number-placement puzzle. On a 9×9 grid consisting of 3×3 boxes, in some fields numbers are presented. The goal is to complete the grid so that in each row, column and block every number from 1 to 9 occurs exactly once. Figure 1 shows an example.

3. Application of Simulated Annealing to Sudoku

To solve a Sudoku by means of Simulated annealing, one needs

- a state space S,
- a neighborhood criterion and
- an energy function E where the minimum energy corresponds to a solved Sudoku.

We define the state space S as follows:

$$S = \{x \in \{1, \ldots, 9\}^{9\times 9} : \text{each row of } x \text{ is a permutation of } \{1, \ldots, 9\}\}$$

Thus, for $x \in S$, one needs only to ensure that each column and each box contains the numbers $1, \ldots, 9$. For the neighbourhood criterium we define $y \in S$ to be a neighbor of $x \in S$ if the only difference between x and y is that in y two number in the same row are switched.

It is not immediately clear which energy function one should choose. Our strategy consists of punishing mistakes in the following way:

$$E = E_1 + E_2 \tag{2}$$

$$E_i = \sum_{i,j=1}^{9} |\nu_{ij}^{(k)} - 1| \geq 0 \quad (i = 1, 2) \tag{3}$$

Here $\nu_{ij}^{(1)}$ is the multiplicity of the number j in column i and $\nu_{ij}^{(2)}$ is the multiplicity of the number j in box i. Since for a correct solution of a Sudoku, in every column and every box each number occurs exactly once, the energy takes a minimum $E = 0$ for a correct solution.

There are many cooling schedules for simulated annealing and the appropriate method always depends on the problem set. In our case, we used an exponential one where in each step the temperature is reduced by a factor α starting from an initial temperature T_0:

$$T_k = \alpha^k \, T_0.$$

4. Algorithm

Require: x, T_0, α
  $E \leftarrow \infty$
  $T \leftarrow T_0$
  for $k = 1, \ldots, 10^4$ **do**
    $T \leftarrow \alpha T$
    choose a row at random
    choose two (not fixed) fields a and b in that row at random
    $y \leftarrow x$ where a and b are switched
    $E' \leftarrow E(y)$
    if rand(0,1) $\leq \min\left(1, \exp\left(-\frac{E'-E}{T}\right)\right)$ **then**
      $E \leftarrow E'$
      $x \leftarrow y$
    end if
    if $E = 0$ **then**
      break
    end if
  end for

5. Results and interpretation

We used a freely available dataset of Sudoku games (`https://www.kaggle.com/bryanpark/sudoku`). First, we explored for which parameters of T_0 and α the algorithms performs best by running it on a subset of 100 Sudoku games. Figure 2 illustrates the performance in terms of the number of solved games (a total of 100) in dependence on the initial temperature for $\alpha = 0.99999$. One can see a maximum at $T_0 \approx 1$.

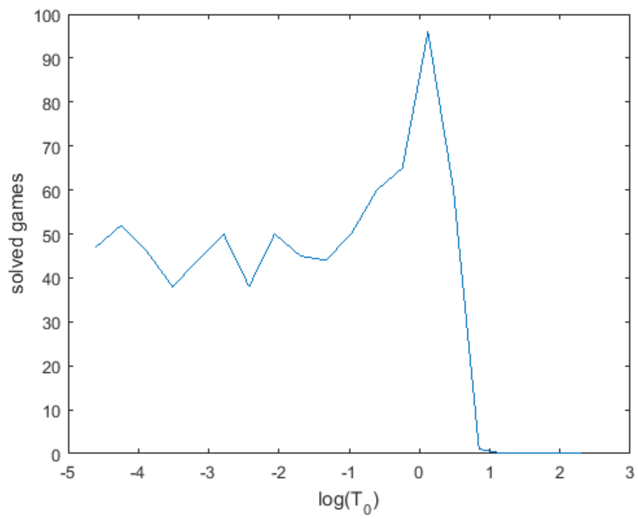

Figure 2: Solved Sudoku games in dependence on T_0

We also calculated the dependency of the number of solved games on α for $T_0 = 1$ (figure 3) and it looks like no cooling at all is required. This seems to be reasonable because we actually already know that $E = 0$ is the global minimum. We just need to determine $x \in S$ where the energy takes this minimum value. Therefore, continuing to explore the state space until the stopping criterion $E = 0$ is met might be smarter than reducing the temperature. This is somewhat different than the problem of finding the minimum value of E.

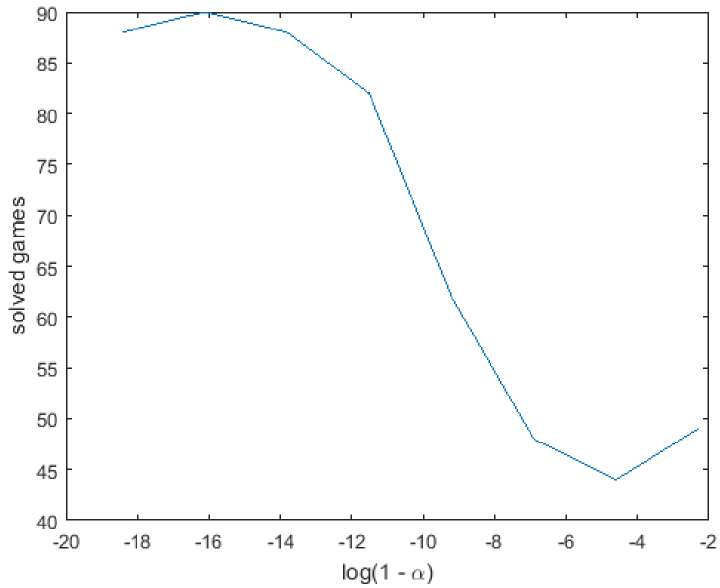

Figure 3: Solved Sudoku games in dependence on α

With these parameter values, Simulated annealing succeeds in solving Sudoku in a lot of cases. The solution of the Sudoku in figure 1 is shown in figure 4. Figure 5 illustrates the course of the energy as a function of the iteration number for the example of this Sudoku, using different initial states (figure 5). Simulated annealing seems to be fairly stable in finding the correct solution. One can see the special behaviour of the algorithm, namely the increase of the energy function and thereby the possibility to escape local minima which is ideal for complex functions with several local minima. However, it is striking that E rises only seldom.

3	1	5	9	4	8	7	6	2
4	8	2	6	7	1	9	3	5
6	7	9	5	2	3	8	1	4
1	5	6	8	9	7	2	4	3
7	2	4	3	1	5	6	8	9
9	3	8	2	6	4	1	5	7
2	9	3	4	8	6	5	7	1
5	6	7	1	3	2	4	9	8
8	4	1	7	5	9	3	2	6

Figure 4: Solved Sudoku game

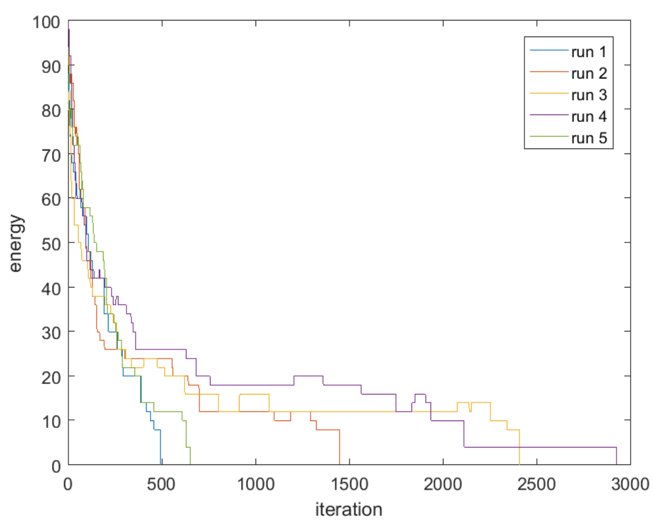

Figure 5: Energy as a function of time (iterations) for 5 runs of Simulated Annealing