Bounded Continous-Time Satisfiability Solver

Hiroshi Yamashita[1], Hideyuki Suzuki[2], Zoltán Toroczkai[3], and Kazuyuki Aihara[1,4]

[1] Graduate School of Information Science and Technology, The University of Tokyo
7-3-1 Hongo, Bunkyo-ku, Tokyo, Japan
[2] Graduate School of Information Science and Technology, Osaka University
1-5 Yamadaoka, Suita-shi, Osaka, Japan
[3] Department of Physics, University of Notre Dame,
225 Nieuwland Science Hall, University of Notre Dame, Notre Dame, Indiana
[4] Institute of Industrial Science, The University of Tokyo
4-6-1 Komaba, Meguro-ku, Tokyo, Japan
Email: hiroshi_yamashita@sat.t.u-tokyo.ac.jp

Abstract—To tackle problems that can not be solved by current digital computers, many systems propose ideas from physics and neuroscience. The CTDS solver introduced by Ercsey-Ravasz and Toroczkai is one of such system. It solves the satisfiability problem by reducing it to a minimization of a time-varying target function. Although the possibility of an efficient electric circuit implementation of the solver has been shown, in terms of physical realizations, the solver has a problem of unbounded variations of the target function parameters. Here we propose a variant of the solver with bounded target function parameters. It includes several possible modifications of the solver in system parameter differences. We also show the basic characteristics of the solver, the upper and lower bounds of the target function parameters.

1. Introduction

Despite the great progress of digital computers, there remain many problems that currently can not be solved efficiently by them, such as those called NP-complete and NP-hard problems. Based on ideas originating from physics and neuroscience, many non-universal, i.e., problem specific systems are being developed to tackle such hard problems [1, 2]. The Boolean satisfiability (SAT) problem, which is a widely known NP-complete problem, is one such family of hard problems, also considered to be one of the most fundamental problems of computer science. The SAT problem is also important in application areas, such as planning and system verification.

1.1. Boolean Satisfiability Problem

A variable that takes *True* or *False* is called Boolean variable. Boolean variables and negated forms of them, such as $\neg x$, are called literals. A formula that is formed by literals and logical connectives, $\wedge$ (AND) and $\vee$ (OR), is called Boolean formula. Any Boolean formula can be transformed into conjunctive normal form (CNF). The CNF is a Boolean formula formed by clauses connected by $\wedge$, where a clause is a Boolean formula that is formed by literals connected by $\vee$. An assignment of truth values, *True* or *False*, to Boolean variables is called a truth assignment. A truth assignment satisfies a Boolean formula when evaluating it with the assignment results in $True$. The problem of finding a satisfying truth assignment of the input Boolean formula is called Boolean satisfiability (SAT) problem. The input is usually given as a CNF. Let $f = c_1 \wedge \cdots \wedge c_M$ be the input CNF, where $c_1, \ldots, c_M$ are clauses. Let $x_1, \ldots, x_N$ be Boolean variables that appear in f, and let $V = \{1, \ldots, N\}$ and $C = \{1, \ldots, M\}$ be the indices of variables and clauses, respectively.

1.2. Continuous-Time Dynamical System Solver

The authors of [3] proposed a continuous-time dynamical system (CTDS) for solving SAT problems. First, let us associate *True* and *False* to $+1$ and -1, respectively, and for $i \in V$, let us associate $s_i \in \{-1, +1\}$ to x_i. Let $c_{m,i}$ represent an appearance and the sign of x_i in c_m. In particular, $c_{m,i} = +1$ holds if x_i appears in c_m, $c_{m,i} = -1$ holds if $\neg x_i$ appears in c_m, and $c_{m,i} = 0$ holds if otherwise. Let $K_m = 2^{-k_m} \prod_{i \in V}(1 - c_{m,i} x_i)$, where k_m represents the number of literals in c_m. We can easily find that $K_m = 0$ holds when c_m is satisfied and $K_m = 1$ holds when c_m is unsatisfied. Then, we can reduce the SAT problem to a minimization of a target function such as $E = \sum_{m \in C} K_m$. Let us relax $s_i \in \{-1, +1\}$ to a continuous variable $s_i \in [-1, +1]$, and use the gradient descent method for the minimization of E. However, E is not necessarily convex and the trajectory of the minimization can be trapped in a local minimum.

It is proposed in [3] that we can avoid the issue of trapping by using a time-varying target function. Let $\mathcal{L} = \sum_{m \in C} a_m K_m^2$ be the target function, where $a_m > 0$ is a weight of c_m. The gradient descent for $\mathcal{L}$ is performed concurrently with growing a_m as follows:

$$\frac{\mathrm{d}\boldsymbol{s}}{\mathrm{d}t} = -\nabla \sum_{m \in C} a_m K_m^2, \tag{1}$$

$$\frac{\mathrm{d}a_m}{\mathrm{d}t} = a_m K_m, \tag{2}$$

where $\boldsymbol{s}$ is a vector of s_i. Because K_m for unsatisfied clauses are positive, the weights for such clauses grow exponentially. Due to the growth of these weights, this system can escape from the local minima and continue the search. In this paper, we call this system the CTDS solver for the SAT problem. It was also shown that the probability that the solver with a random initial condition does not find a satisfying assignment decays exponentially in time and that the decay rate for random SAT instances decreases only in polynomially with the problem size, while the cost of the simulation in a digital computer grows exponentially in size.

As shown in [4], the system could be implemented efficiently with electric circuits. However, it was also reported that the proposed prototype sometimes fails to solve instances, because of the limitation of the voltage for representing the weight variables.

The two parts of the CTDS solver, gradient descent and the change in the target function, have their own timescales. The balance of the two timescales is investigated in [5] by using a variant of the CTDS solver that explicitly represents the relative timescale of the two parts of the system;

$$\frac{\mathrm{d}\boldsymbol{s}}{\mathrm{d}t} = -\eta \nabla \sum_{m \in C} a_m K_m^2, \tag{3}$$

$$\frac{\mathrm{d}a_m}{\mathrm{d}t} = a_m (K_m - T), \tag{4}$$

where $T = \sum_{m \in C} a_m K_m$. In this system, the sum of the weight variables is kept constant because of the decreasing term T and we can tune the relative timescale by η. In [5], a natural time measure based on the limitation in physical realization is proposed for the evaluation of the choice of η and it is shown that a strongly biased relative timescale degrades the search performance.

The unbounded growth of the weight variable in the original system is problematic in terms of physical realization. The system of [5] avoids the unbounded growth, moreover, a_m may decay to be extremely small, so that in a physical realization, noise can destroy the information and randomize the dynamics, degrading performance. In other words, $-\log a_m$ can grow unboundedly. Therefore, we have to develop a system which avoids both the unbounded increases and decreases of $\log a_m$. In this paper, we propose a system that has a decay term which prevents such unbounded variations. We also show basic characteristics of the proposed system; the upper and lower bounds of $\log a_m$. Both bounds are important because the operating range in the analog circuit implementation will be determined by these bounds. We also show that several possible modifications of the system are identical to changing the parameters and the time variable in the system, and, as a consequence, the original system (1) and (2) and variant (3) and (4) are included in the system.

2. Bounded CTDS Solver

The proposed system is as follows:

$$\frac{\mathrm{d}\boldsymbol{s}}{\mathrm{d}t} = -\nabla \sum_{m \in C} a_m K_m^2, \tag{5}$$

$$\frac{\mathrm{d}b_m}{\mathrm{d}t} = (K_m - c) - \lambda b_m, \tag{6}$$

where $a_m = \exp(b_m)$. Parameters c and λ represent a threshold for K_m and decay rate of b_m, respectively, and they may depend on time. This inherits the important properties of the original system, namely, the improvement of $\boldsymbol{s}$ following the gradient descent direction and the exponential growth of the weights of unsatisfied clauses. If K_m is larger than the threshold, corresponding weight increases, and it decreases, otherwise. Although we also call b_m as weights, b_m can be negative. The last term of (6) pulls b_m to zero. In section 2.4, we will describe that this system can be derived from making the variance of $\log a_m$ constant in the system (3) and (4). The equivalent transformation results in a different system from (5) and (6) in multipliers, and in general we can imagine natural modifications of the system that multiplies a_m, b_m and the time variable in (5) and (6). However, the proposed system can represent such differences by changing the parameters c and λ and the time variable. The multiplier can be time-dependent. In the following, we show the actual parameter changes for time-dependent case and those for the time-independent case can be obtained by substituting $\frac{\mathrm{d}A}{\mathrm{d}t} = 0$ in them.

2.1. Multiplying a_m

First, we consider the following system obtained by multiplying a_m by A:

$$a'_m = A \exp(b'_m), \tag{7}$$

$$\frac{\mathrm{d}\boldsymbol{s}}{\mathrm{d}t} = -\nabla \sum_{m \in C} a'_m K_m^2, \tag{8}$$

$$\frac{\mathrm{d}b'_m}{\mathrm{d}t} = (K_m - c') - \lambda' b'_m. \tag{9}$$

By changing variables as $b_m = b'_m + \log A$, $a_m = \exp(b_m)$, $c = c' - \lambda \log A - \frac{\mathrm{d}\log A}{\mathrm{d}t}$, and $\lambda = \lambda'$, we reconstruct the system as follows:

$$\frac{\mathrm{d}\boldsymbol{s}}{\mathrm{d}t} = -\nabla\sum_{m\in C}\exp(b'_m + \log A)K_m^2 = -\nabla\sum_{m\in C} a_m K_m^2, \quad (10)$$

$$\frac{\mathrm{d}b_m}{\mathrm{d}t} = \left(K_m - c - \lambda\log A - \frac{\mathrm{d}\log A}{\mathrm{d}t}\right) - \lambda(b_m - \log A) + \frac{\mathrm{d}\log A}{\mathrm{d}t} \quad (11)$$

$$= (K_m - c) - \lambda b_m. \quad (12)$$

2.2. Multiplying b_m

Next we consider multiplying b_m by $1/A$:

$$a''_m = \exp\left(\frac{b''_m}{A}\right), \quad (13)$$

$$\frac{\mathrm{d}\boldsymbol{s}}{\mathrm{d}\tau} = -\nabla\sum_{m\in C} a''_m K_m^2, \quad (14)$$

$$\frac{\mathrm{d}b''_m}{\mathrm{d}\tau} = (K_m - c'') - \lambda'' b''_m. \quad (15)$$

In this case, changing variables as $b'_m = b''_m/A$, $a'_m = A\exp(b'_m)$, $c' = c''$, $\lambda' = A\lambda'' + \frac{\mathrm{d}A}{\mathrm{d}t}$, and $\mathrm{d}\tau = A\mathrm{d}t$ leads the following system:

$$\frac{\mathrm{d}\boldsymbol{s}}{\mathrm{d}t} = -A\nabla\sum_{m\in C} a''_m K_m^2 = -\nabla\sum_{m\in C} a'_m K_m^2, \quad (16)$$

$$\frac{\mathrm{d}b'_m}{\mathrm{d}t} = A\left(\frac{1}{A}\left((K_m - c') - \frac{1}{A}\left(\lambda' - \frac{\mathrm{d}A}{\mathrm{d}t}\right)Ab'_m\right) - \left(\frac{1}{A^2}\frac{\mathrm{d}A}{\mathrm{d}t}\right)Ab'_m\right), \quad (17)$$

$$= (K_m - c') - \lambda' b'_m. \quad (18)$$

Because this is equivalent to the system in section 2.1, we can also reconstruct the system (5) and (6).

2.3. Multiplying Time Variable

By changing the coefficient of the change speed in weight, we propose

$$a''_m = \exp(b''_m), \quad (19)$$

$$\frac{\mathrm{d}\boldsymbol{s}}{\mathrm{d}\tau} = -\nabla\sum_{m\in C} a''_m K_m^2, \quad (20)$$

$$\frac{\mathrm{d}b''_m}{\mathrm{d}\tau} = \frac{1}{A}((K_m - c'') - \lambda'' b''_m). \quad (21)$$

We can obtain a system equivalent to that in section 2.1 by changing variables as $b'_m = b''_m$, $a'_m = A\exp(b'_m)$, $c' = c''$, $\lambda' = \lambda''$, and $\mathrm{d}\tau = A\mathrm{d}t$ as follows:

$$\frac{\mathrm{d}\boldsymbol{s}}{\mathrm{d}t} = -A\nabla\sum_{m\in C} a''_m K_m^2 = -\nabla\sum_{m\in C} a'_m K_m^2, \quad (22)$$

$$\frac{\mathrm{d}b'_m}{\mathrm{d}t} = (K_m - c') - \lambda' b'_m. \quad (23)$$

2.4. CTDS that Keeps Variance of Weights

We can naturally derive the proposed system by keeping the variance of the $\log a_m$ of the CTDS solver.

First, we note that the original CTDS solver corresponds to the case $c = 0$ and $\lambda = 0$.

Let $S = \sum_{m\in C} a_m$, $b'_m = \log(a_m/S)$, and $T = \frac{\mathrm{d}\log S}{\mathrm{d}t} = \frac{1}{S}\sum_{m\in C}\frac{\mathrm{d}a_m}{\mathrm{d}t} = \sum_{m\in C}\exp(b'_m)K_m$. The following system proposed in [5], which is equivalent to the original system, keeps the average of the weights:

$$\frac{\mathrm{d}\boldsymbol{s}}{\mathrm{d}t} = -S\nabla\sum_{m\in C}\exp(b'_m)K_m^2, \quad (24)$$

$$\frac{\mathrm{d}b'_m}{\mathrm{d}t} = \frac{\mathrm{d}\log a_m}{\mathrm{d}t} - \frac{\mathrm{d}\log S}{\mathrm{d}t} = K_m - T. \quad (25)$$

This system corresponds to the case $c = T$ and $\lambda = 0$ except for that $\frac{\mathrm{d}\boldsymbol{s}}{\mathrm{d}t}$ is multiplied by S, which is described in section 2.1.

We can also keep the variance of the weights by applying the following transformation to the system (24) and (25). Let $b''_m = Bb'_m$ so that $\frac{1}{M}\sum_{m\in C}(b''_m - \mu)^2 = 1$ holds, where $\mu = \frac{1}{M}\sum_{m\in C} b''_m$. By substituting $\frac{\mathrm{d}b''_m}{\mathrm{d}t} = \frac{\mathrm{d}B}{\mathrm{d}t}b'_m + B\frac{\mathrm{d}b'_m}{\mathrm{d}t}$ to $0 = \frac{1}{M}\sum_{m\in C}(b''_m - \mu)(\frac{\mathrm{d}b''_m}{\mathrm{d}t} - \frac{\mathrm{d}\mu}{\mathrm{d}t}) = \frac{1}{M}\sum_{m\in C}(b''_m - \mu)\frac{\mathrm{d}b''_m}{\mathrm{d}t}$, we obtain the following equation:

$$0 = \frac{\mathrm{d}B}{\mathrm{d}t}\frac{1}{M}\sum_{m\in C}(b''_m - \mu)b'_m + \frac{B}{M}\sum_{m\in C}(b''_m - \mu)\frac{\mathrm{d}b'_m}{\mathrm{d}t} \quad (26)$$

$$= \frac{\mathrm{d}B}{\mathrm{d}t}\frac{1}{M}\sum_{m\in C}(b''_m - \mu)\frac{1}{B}(b''_m - \mu) + \frac{B}{M}\sum_{m\in C}(b''_m - \mu)(K_m - T) \quad (27)$$

$$= \frac{1}{B}\frac{\mathrm{d}B}{\mathrm{d}t} + BC, \quad (28)$$

where $C = \frac{1}{M}\sum_{m\in C}(b''_m - \mu)(K_m - T)$. Hence we obtain $\frac{\mathrm{d}B}{\mathrm{d}t} = -B^2C$ and the system

$$\frac{\mathrm{d}\boldsymbol{s}}{\mathrm{d}t} = -S\nabla\sum_{m\in C}\exp\left(\frac{b''_m}{B}\right)K_m^2, \quad (29)$$

$$\frac{\mathrm{d}b''_m}{\mathrm{d}t} = \frac{\mathrm{d}B}{\mathrm{d}t}b'_m + B\frac{\mathrm{d}b'_m}{\mathrm{d}t} \quad (30)$$

$$= -B^2Cb'_m + B(K_m - T) \quad (31)$$

$$= B\left((K_m - T) - Cb''_m\right). \quad (32)$$

Here the threshold is T and decay rate is C, and this can be also reduced to the proposed system (5) and (6) by using the previous reductions.

3. Characteristics of the Proposed System

In this section, we assume that c and λ are time-independent. To make the system bounded, we can fix the value of S and B of the systems in the previous section. We can also use fixed T or C to eliminate its calculation cost. We can learn about the characteristics and performance of such a system by investigating the case of time-independent parameters.

Table 1: The maximum value of b_m and the parameter c. Maximum value is taken for $m \in C$ and $t \in [0, 100]$, and the result for 500 trajectories with 50 instances are averaged.

c	$\max b_m(t)$
0.001	46.88
0.002	45.63
0.005	40.69
0.01	34.87
0.02	27.27
0.05	14.08

3.1. Upper Bound and Lower Bound of b_m

We assume that $\lambda > 0$ and $c > 0$. $\frac{db_m}{dt} > 0$ holds when $b_m < -c/\lambda$ and $\frac{db_m}{dt} < 0$ holds when $b_m > (1-c)/\lambda$. Thus, if the initial value of b_m is in $[-c/\lambda, (1-c)/\lambda]$, b_m remains in the same range. As we stated in the introduction, the maximum and minimum values of the b_m are important in terms of physical realizations of the solver.

3.2. Parameter c Determines $\max b_m$

Although the upper bound of b_m is shown, we found that the maximum value of b_m is usually smaller than that. It is because, during the increase of b_m, $\boldsymbol{s}$ escapes from the local minimum and then K_m becomes small so that $K_m - c < 0$.

We also found that, if c is small, this maximum value of b_m increases with the decrease of c. This can be explained as follows. Because we assumed that c is small, the increasing speeds of b_m for unsatisfied clauses are same. Let $c_{m'}$ be the clause that previously has a maximum b_m value. Because whether the local minima is destabilized or not is determined by $\mathcal{L}$ without a constant factor, the difference between the $b_{m'}$ and the new $\max b_m$ required to destabilize the local minimum is similar among the different c settings. If $b_{m'}$ is small, the required time for small b_m to become sufficiently larger than $b_{m'}$ is short. If the decrease of the the $b_{m'}$ during this period is smaller than the required difference, the new $\max b_m$ is larger than the previous $\max b_m$. The $\max b_m$ increases until it requires long time to destabilize the local minimum so that during this period $b_{m'}$ decreases more than the difference between $b_{m'}$ and the new $\max b_m$. Therefore, because the decreasing speed is proportional to c, $\max b_m$ is increased by using a small c.

We generated several 3-SAT instances (CNF all of whose clauses have exactly 3 literals) with 100 variables and 427 clauses and took 50 satisfiable ones from them. We ran the proposed system 10 times for each instance with varying parameter $c = 0.001, 0.002, 0.005, 0.01, 0.02, 0.05$ and $\lambda = 0.01$. For each obtained trajectory we calculated $\max_{m \in C, t \in [0,100]} b_m(t)$ and averaged it over 500 trajectories for each setting. From Table 1, we can see that the maximum value of b_m increases with the decrease of c.

4. Conclusion

To address the issue of unlimited maximum and minimum values of the weights in the CTDS solver, we proposed a variant of the system that adds a decay term in the dynamics of the weights. This system can represent its several natural modifications by changing parameters and time variables. We also showed the upper and lower bound of b_m, and that usually the maximum value of b_m increases if we decrease c.

We note that the maximum value of b_m can effect the search efficiency. If $\max b_m$ is increased, the number of clauses that have large b_m enough to affect the change in $\boldsymbol{s}$ increases. It means that the system can take many constraints into consideration concurrently, and the searching efficiency may increase. It is also reported in [5] that increasing η, which corresponds to increase $\max b_m$, results in a decrease of the searching time of the system to find the satisfying assignment.

In addition, we note that λ can be related to the solver's abillity to find solutions. Assume that there is a periodic orbit of the system. The stability of the periodic orbit is determined by whether a pertubation introduced at some point is expanded when it returns along the orbit to that point. Because the term $-\lambda b_m$ in (6) shrinks the pertubation, we can say that if we use large λ, periodic orbits are likely to be stable and the system is likely to be trapped in a limit cycle. Thus, we have to represent wide range of b_m if we use small λ. There should be a trade-off between the completeness and the value range.

In any way, theoretically and empirically, testing these arguments which are not sufficiently confirmed yet is left as a future work.

Acknowledgments

This work is supported in part by JST CREST (JPMJCR18K2 and JPMJCR14D2) and by a Grant-in-Aid for JSPS Fellows (17J02174). It was also supported in part by the National Science Foundation under Grants CCF-1644368 and 1640081, and by the Nanoelectronics Research Corporation, a wholly-owned subsidiary of the Semiconductor Research Corporation, through Extremely Energy Efficient Collective Electronics, an SRC-NRI Nanoelectronics Research Initiative under Research Task ID 2698.004 (Z.T.).

References

[1] L. Chen and K. Aihara, "Chaotic simulated annealing by a neural network model with transient chaos," *Neural Networks*, vol. 8, pp. 915–930, 1995.

[2] T. Inagaki *et al.*, "A coherent Ising machine for 2000-node optimization problems," *Science*, vol. 354, no. 6312, pp. 603–606, 2016.

[3] M. Ercsey-Ravasz and Z. Toroczkai, "Optimization hardness as transient chaos in an analog approach to constraint satisfaction." *Nature Physics*, vol. 7, pp. 966–970, 2011.

[4] X. Yin *et al.*, "Efficient analog circuits for Boolean satisfiability," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 26, no. 1, pp. 155–167, 2018.

[5] H. Yamashita, K. Aihara, and H. Suzuki, "Timescales of Boolean satisfiability solver using continuous-time dynamical system," 2019, submitted.