A Study of Pure Random Walk Algorithms on Constraint Satisfaction Problems with Growing Domains*

Wei Xu[1] and Fuzhou Gong[2]

[1] School of Automation and Electrical Engineering,
University of Science and Technology Beijing, Beijing 100083, China
xuwei_225@163.com

[2] Institute of Applied Mathematics, Academy of Mathematics and System Science,
Chinese Academy of Sciences, Beijing 100190, China
fzgong@amt.ac.cn

Abstract. The performances of two types of pure random walk (PRW) algorithms for a model of constraint satisfaction problems with growing domains (called Model RB) are investigated. Threshold phenomenons appear for both algorithms. In particular, when the constraint density r is smaller than a threshold value r_d, PRW algorithms can solve instances of Model RB efficiently, but when r is bigger than the r_d, they fail. Using a physical method, we find out the threshold values for both algorithms. When the number of variables N is large, the threshold values tend to zero, so generally speaking PRW does not work on Model RB.

Keywords: constraint satisfaction problems, Model RB, random walk, local search algorithms.

1 Introduction

Constraint satisfaction problems (CSPs) arise in a large spectrum of scientific disciplines, such as computer science, information theory, and statistical physics [19, 17, 14]. A typical CSP instance involves a set of variables and a collection of constraints. Variables take values in a finite domain. Constraints contain a few variables and forbid some of their joint values. A solution is an assignment satisfying all the constraints simultaneously. Given a CSP instance, two fundamental scientific questions are to decide the existence of solutions and to find out a solution if it exists. Examples of CSPs are Boolean formula satisfiability (SAT), graph coloring, variants of SAT such as XORSAT, error correction codes, etc.

Random models of CSPs play a significant role in computer science. As instance generators, they provide instances for benchmarking algorithms, help to inform the design of algorithms and heuristics, and provide insight into problem hardness. Classical random CSP models were proposed and denoted

* Partially supported by NSFC 61370052 and 61370156.

J. Chen, J.E. Hopcroft, and J. Wang (Eds.): FAW 2014, LNCS 8497, pp. 276–287, 2014.

by A, B, C and D respectively [20, 12], and many alternatives also appeared [1, 27, 25, 11, 9, 10].

Model RB is a typical CSP model with growing domains. It was proposed by Xu and Li [27] to overcome the trivial insolubility of the classical model B, and was proved to have exact satisfiability phase transitions. The instances generated in the phase transition region of Model RB are hard to solve [28, 29] and have been widely used in various kinds of algorithm competitions. Model RB develops a new way to study CSPs, especially CSPs with large domains, thus has gotten considerable attention [14, 32, 15, 31, 18, 3, 13, 26, 16].

Algorithm analysis is a notoriously difficult task. The current rigorous results mostly deal with algorithms that are extremely simple, such as backtrack-free algorithms, which assign variables one by one without backtracking [5, 4]. Pure Random Walk (PRW) algorithm is a process that consists of a succession of random moves. It is relatively simple and has been intensively studied on the k-SAT problem [2, 21, 22, 6–8, 23, 24]. On k-SAT, A frequently studied PRW algorithm (Algorithm 2 in the following) is called Walksat. Another reason for PRW algorithm being studied is that random walk is a part of many local search algorithms [19].

In this paper, we study two types of PRW algorithms on Model RB. By experimental methods, threshold phenomenons on performance of these two PRW algorithms are found, just like that of Walksat on k-SAT. Moreover, by a physical method we locate the thresholds for both algorithms, which are $\frac{1-p}{p}\frac{1}{k\ln N}$, with N being the total number of variables, k the number of variables per constraints, p the portion of forbidden joint values per constraints.

This paper is organized as follows. We first give the definition of Model RB and its main properties in Section 2. In Section 3, we show the threshold behaviors of PRW algorithms by experiments, and also show the different performances before and after the thresholds. In Section 4, we use a physical method to calculate the thresholds for both algorithms. We finally give some concluding remarks in Section 5.

2 Model RB

Both classical and revised models of CSPs can be found in [14]. Here we give the definition of Model RB. Let $k \geq 2$ be an integer. Let $r > 0$, $\alpha > 0$, $0 < p < 1$ be real numbers. Let N be the number of variables and $V = \{\sigma_1, \sigma_2, \cdots, \sigma_N\}$ the set of variables. Each variable takes values from a domain $D = \{1, 2, \cdots, N^\alpha\}$. Each constraint involves k variables and an associated incompatible-set, which is a subset of the Cartesian product D^k. Elements in incompatible-set are called incompatible (forbidden) joint values. Model RB(N, k, r, α, p) is a probability space defined by the folowing steps to generate its instances.

1. We select with repetition $rN \ln N$ constraints independently at random. Each constraint is formed by selecting without repetition k out of N variables independently at random.

2. For each constraint, we form an incompatible-set by selecting without repetition $pN^{\alpha k}$ elements from D^k independently at random.

A solution is an assignment which satisfies all the constraints. That is to say, the joint values in a solution dose not belong to any incompatible-sets of the constraints. The set of all solutions, denoted by $\mathcal{S}$, is a subset of D^N. Let X be the number of solutions, $X = |\mathcal{S}|$. It is easy to see that in model RB, the expectation of X is

$$\mathbb{E}(X) = N^{\alpha N}(1-p)^{rN \ln N}.$$

Let

$$r_{cr} = -\frac{\alpha}{\ln(1-p)}.$$

If $\alpha > \frac{1}{k}$ and $0 < p < 1$ are two constants, and k and p satisfy the inequality $k \geq \frac{1}{1-p}$, then

$$\lim_{n\to\infty} \Pr(X > 0) = \begin{cases} 1, r < r_{cr}, \\ 0, r > r_{cr}. \end{cases}$$

Thus, Model RB has exact satisfiability phase transitions, see [27, 31].

3 Performance of Pure Random Walk on Model RB

In this section, we study the performance of PRW algorithms on Model RB. By experiments, we find that PRW algorithms exhibit threshold phenomenons, and have different performances before and after the thresholds.

3.1 Pure Random Walk Algorithms

We concentrate on two types of PRW algorithms, called Algorithm 1 and Algorithm 2 respectively. In Algorithm 1, we randomly reassign a variable from conflict set. In algorithm 2, we randomly select an unsat-constraint (unsatisfied constraint), then randomly select one of its variable to reassign it.

Algorithm 1

1. Pick up a random assignment. Set up a maximum number of steps.
2. Let conflict set be the set of all variables that appear in a constraint that is unsatisfied under the current assignment.
   (a) If the conflict set is empty, terminate the algorithm, output the current assignment.
   (b) Otherwise, randomly select a variable in the conflict set, reassign it a value.
3. Repeat step 2, until the maximum number of steps, then output *fail*.

Algorithm 2

1. Pick up a random assignment. Set up a maximum number of steps.
   (a) If current assignment satisfies all constraints, terminate the algorithm, output the current assignment.
   (b) Otherwise, randomly select an unsat-constraint, and randomly select a variable in the constraint, reassign it a value.
2. Repeat step 2, until time of repeating has gotten to the maximum step number, output *fail*.

3.2 Threshold Behavior

Both Algorithm 1 and Algorithm 2 exhibit threshold phenomenons, as shown in Figure 1. The probability of getting a solution drops from 1 to 0 dramatically. Every point in Figure 1 is averaged over 10 runs, and the maximum number of steps is 2000.

The same threshold phenomenon has been found for Walksat on k-SAT problem [7], with a conjectured threshold value $\alpha = 2^k/k$.

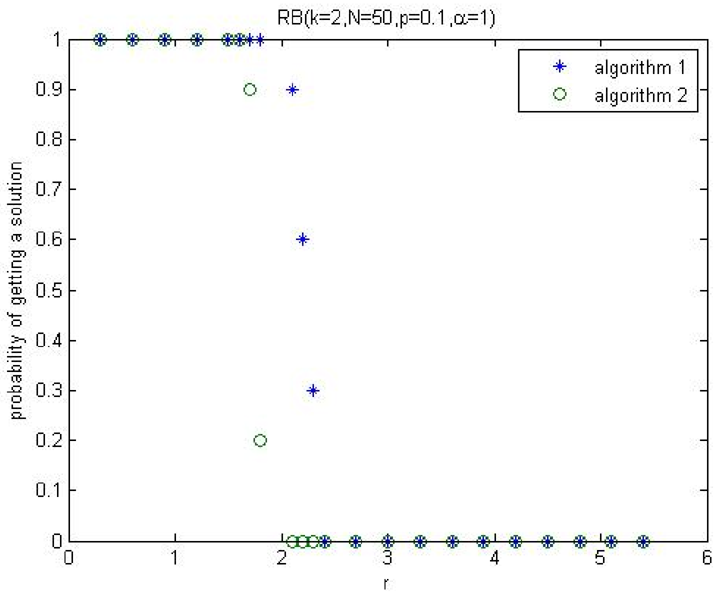

Fig. 1. Probability of getting a solution by PRW algorithms on model RB

3.3 Before Threshold

When $r < r_d$ (r_d is the threshold value) and r is very small, algorithms can find a solution in a short time. If at each step, the number of unsat-constraints decreases by $O(1)$, then the solving time will be $O(N \ln N)$. Figure 2 shows the average number of running steps, each point is averaged over 100 runs. Figure 3 shows the average number of running steps divided by $N \ln N$. So when r is very small, the solving time is in an order of $O(N \ln N)$.

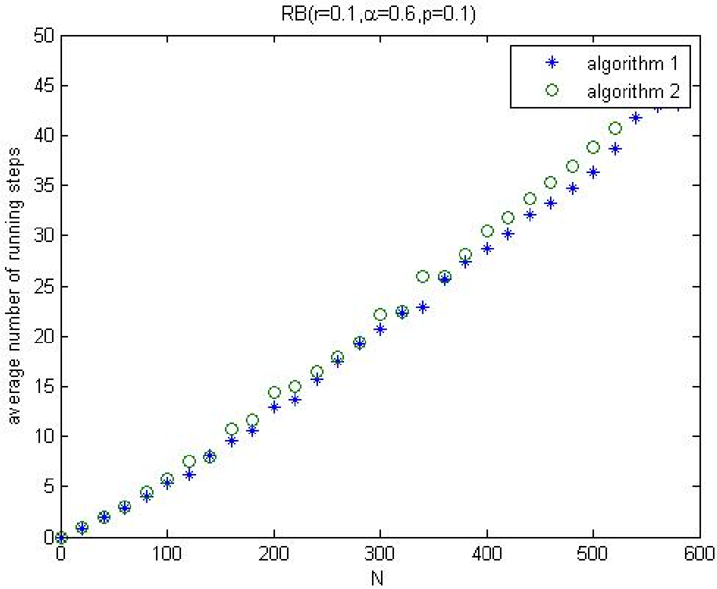

Fig. 2. Average number of running steps

3.4 After Threshold, Algorithm 2

When $r > r_d$, variables will be reassign values again and again, the number of unsat-constraints will fluctuate around some plateau value for a long time, see Figure 4. (Experiments on Algorithm 1 are similar.) The number of unsatisfied clauses exhibit a distribution, see Figure 5. This is the same as Walksat on k-SAT.

Two simple but not rigorous interpretations are as follows. First, the chosen constraint is optimized to be satisfied, but when variables contained in the constraint are reassigned values again for other chosen constraints, the optimization was destroyed. Second, the value of the reassigned variable is optimized, but when variables connected to the reassigned variable are reassigned values, the optimization was destroyed. So when r is big, for example $r > r_d$, optimization (or effect of each reassignment) cannot be retained, and algorithms fail.

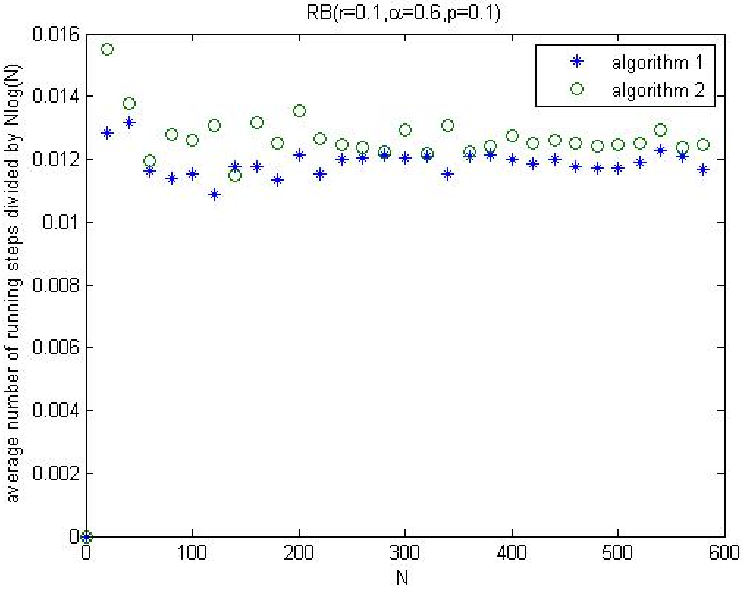

Fig. 3. Average number of running steps divided by $Nlog(N)$

4 Analysis Based on an Approximation

The main method used in this section is from theoretical physics, which has been used on k-SAT and other problems by Semerjian et al [21, 22, 6]. It is not a rigorous method, since an approximation is utilized, but remarkable results have been gotten on k-SAT problem and XORSAT problem with this method. For more background and correctness of this method, we refer to [21].

Approximation. At each step, before a reassignment, we treat the situation at that time as a typical situation, featured by the number of its unsat-constraints. A typical situation featured by M_0 means that, $M = rN \ln N$ constraints are randomly selected (as step 1 of Model RB definition), then M_0 unsat-constraints are randomly chosen from the M constraints. Then the solving process becomes a Markov chain using number of unsat-constraints as its state space, and the transition probability from M_0 to M_0' is the probability that the typical situation featured by M_0 have M_0' unsat-constraints after a step (a reassignment).

First, we should give the transition probability from M_0 to M_0'. We will choose a variable to reassign from typical situation featured by M_0. Let $p(Z_1)$ be the probability that a variable with Z_1 unsat-constraints will be chosen, which depends

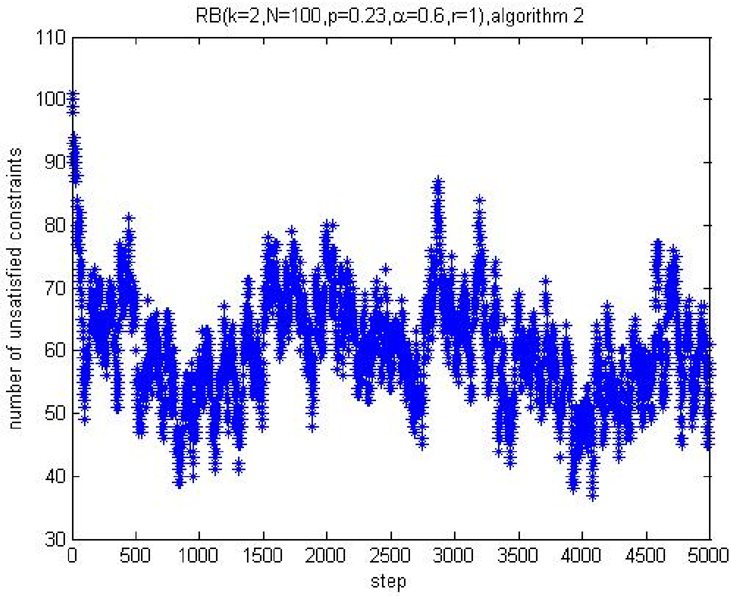

Fig. 4. Number of unsat-constraints, Algorithm 2

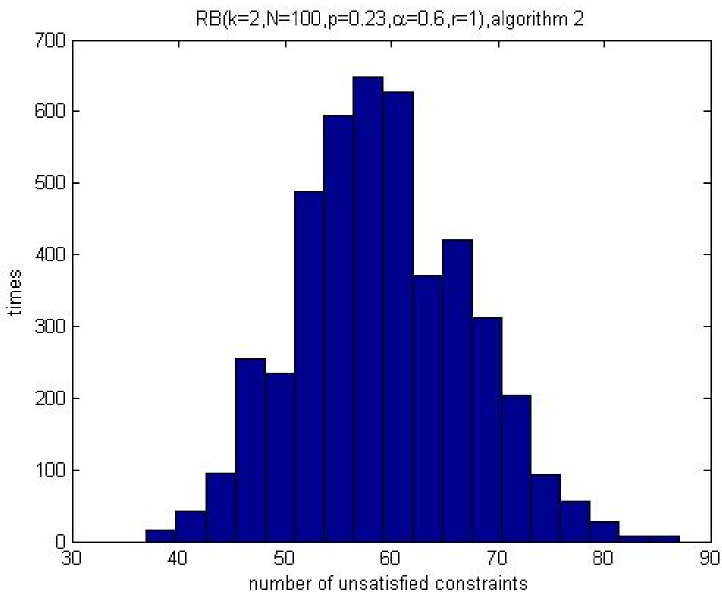

Fig. 5. Histogram of the number of unsat-constraints, where data at the first 500 step were omitted

on algorithms. The probability that Z_2 of Z_1 unsat-constraints become satisfied after reassignment is

$$\binom{Z_1}{Z_2}(1-p)^{Z_2}(p)^{Z_1-Z_2} \triangleq p(Z_2),$$

Z_2 obeys the binomial distribution. When $M_0 > 0$, the probability that Z_3 constraints become unsatisfied from satisfied is

$$\binom{M-M_0}{Z_3}\left(\frac{pk}{N}\right)^{Z_3}\left(1-\frac{pk}{N}\right)^{M-M_0-Z_3} \triangleq p(Z_3), \tag{1}$$

because each of $M-M_0$ feasible satisfied constraints connects to the reassignment variable with probability $\frac{k}{N}$, and each connecting constraint becomes unsatisfied with probability p. When $M_0 = 0$, $Z_3 = 0$ with probability 1.

Then the transition probability from M_0 to M_0' is

$$A_{M_0',M_0} = \sum_{Z_1=0}^{M_0}\sum_{Z_2=0}^{Z_1}\sum_{Z_s=0}^{M-M_0} p(Z_1)p(Z_2)p(Z_3)1_{M_0'-M_0+Z_2-Z_3}$$

where

$$1_X = \begin{cases} 1, \text{ if } X = 0, \\ 0, \text{ otherwise.} \end{cases}$$

The initial distribution, i.e. the probability that at time 0 the typical situation is featured by M_0, is

$$\Pr[M_0, 0] = \binom{M}{M_0} p^{M_0}(1-p)^{M-M_0}. \tag{2}$$

Iteratively, the probability that at time $T+1$ the typical situation is featured by M_0' is

$$\Pr[M_0', T+1] = \sum_{M_0=0}^{M} A_{M_0'M_0}\Pr[M_0, T].$$

Criterion. If $\Pr[0,T]$ is 0 (almost) in a long time, it is in the phase after the threshold; if $\Pr[0,T]$ becomes positive from 0 in polynomial time, it is in the phase before the threshold. Denote the average fraction of unsat-constraints at time $T = tM$ by $\varphi(t)$,

$$\varphi(t) = \frac{1}{M}\sum_{M_0=0}^{M} M_0 \Pr[M_0, T = tM].$$

In the beginning, $\Pr[0,T] = 0$, but if $\varphi(t)$ become 0 in polynomial time, then $\Pr[0,T]$ will become positive from 0; if $\varphi(t)$ is always positive, then the number of unsat-constraints will fluctuate around some plateau value, $\Pr[0,T]$ will always be 0. So the criterion is whether $\varphi(t)$ becomes 0 in polynomial time when $\Pr[0,T] = 0$.

4.1 Analysis on Algorithm 1

When N is large, we might as well say $\frac{d\varphi}{dt} = (\varphi(t+\frac{1}{M}) - \varphi(t))/(1/M)$, then

$$\begin{aligned}\frac{d\varphi}{dt} &= \sum_{M_0'=0}^{M} M_0' \left(\sum_{M_0=0}^{M} A_{M_0' M_0} \Pr[M_0, T] \right) - \sum_{M_0=0}^{M} M_0 \Pr[M_0, T] \\ &= \sum_{M_0=0}^{M} \Pr[M_0, T] \left(\sum_{M_0'=0}^{M} A_{M_0' M_0} (M_0' - M_0) \right) \\ &= \sum_{M_0=0}^{M} \Pr[M_0, T] \, (\mathbb{E}(Z_3) - \mathbb{E}(Z_2)), \qquad (3)\end{aligned}$$

where $\mathbb{E}(Z_3)$ is the average number of constraints becoming unsatisfied from satisfied, referring to (1),

$$\mathbb{E}(Z_3) = \frac{k}{N}(M - M_0)p; \qquad (4)$$

$\mathbb{E}(Z_2)$ is the average number of constraints becoming satisfied from unsatisfied,

$$\mathbb{E}(Z_2) = E(Z_1)(1-p). \qquad (5)$$

According to Algorithm 1, we randomly select a variable to reassign from the conflict set. When $M_0 > 0$,

$$\mathbb{E}(Z_1) = \frac{M_0 k}{N}\beta, \qquad (6)$$

where $\beta = \mathbb{E}(\frac{1}{X})$; X is the fraction of not empty variables (connecting to at least an unsat-constraint), when we throw M_0 unsat-constraints to N variable. $\mathbb{E}(Z_1)$ is at least 1 and

$$\mathbb{E}(Z_1) \to 1, \ as \ M_0/M \to 0. \qquad (7)$$

When $\Pr[0, T] = 0$, from (3)(4)(5)(6), we have

$$\begin{aligned}\frac{d\varphi}{dt} &= \frac{krN \ln N}{N}(1-\varphi)p - \sum_{M_0=1}^{M} \Pr[M_0, T] \frac{k}{N} M_0 \beta (1-p) \\ &= -(1-p) + kr \ln N (1-\varphi)p - \sum_{M_0=1}^{M} \Pr[M_0, T] (\frac{k}{N} M_0 \beta - 1)(1-p). \qquad (8)\end{aligned}$$

Let

$$r_d = \frac{1-p}{p} \frac{1}{k \ln N}.$$

For $r < c \cdot r_d$, where $c < 1$ is a constant, from (8) we have

$$\frac{d\varphi}{dt} < -(1-c)(1-p).$$

From (2), we have $\varphi(t=0)=p$. So when $r < c\cdot r_d$, φ becomes 0 before $t=\frac{p}{(1-c)(1-p)}$. For $r > c\cdot r_d$, where $c>1$ is a constant, when φ is near 0, by (7) we can see the last term of (8) is near 0, so $\frac{d\varphi}{dt}>0$, φ is always positive.

Thus, the threshold value of Algorithm 1 is $r_d=\frac{1-p}{p}\frac{1}{k\ln N}$.

4.2 Analysis on Algorithm 2

For Algorithm 2, when $M_0>0$, the probability that a variable with Z_1 unsat-constraints was chosen is

$$p(Z_1)=\binom{M_0-1}{Z_1-1}\left(\frac{k}{N}\right)^{Z_1-1}\left(1-\frac{k}{N}\right)^{M_0-Z_1}$$

because we randomly select an unsat-constraint, then each of other M_0-1 unsat-constraints connects to the variable with probability $\frac{k}{N}$. Therefore

$$\mathbb{E}(Z_1)=1+(M_0-1)k/N. \qquad (9)$$

Similarly, for Algorithm 2, when $\Pr[0,T]=0$, from (3)(4)(5)(9) we know

$$\begin{aligned}\frac{d\varphi}{dt}&=\sum_{M_0=1}^{M}\Pr[M_0,T]\left(\frac{k}{N}(M-M_0)p-(1-p)(1+(M_0-1)\frac{k}{N})\right)\\&=-(1-p)+kr\ln Np+\frac{k}{N}(1-p)-k\varphi rlnN.\end{aligned}$$

Sloving this first-order linear differential equation with the initial condition $\varphi(t=0)=p$, we get

$$\varphi(t)=p+\frac{1-p-k(1-p)/N}{rk\ln N}(e^{-rk\ln Nt}-1).$$

Sloving equation $\lim_{t\to\infty}\varphi(t)=0$ of variable r, we have

$$r=(1-\frac{k}{N})\frac{1-p}{p}\frac{1}{k\ln N}\triangleq r'_d.$$

For $r<cr'_d$, where $c<1$ is a constant, $\lim_{t\to\infty}\varphi(t)<0$. Function $\varphi(t)$ decreases and becomes 0 before $t=\frac{p}{(1-c)(1-p)(1-k/N)}$. For $r>cr'_d$, where $c>1$ is a constant, $\lim_{t\to\infty}\varphi(t)>0$.

Thus, the threshold value on Algorithm 2 is

$$r'_d\approx r_d=\frac{1-p}{p}\frac{1}{k\ln N}.$$

4.3 A Note

The estimates of threshold values from numerical simulations are always larger than the calculated one. Taking RB($k=2, N=350, p=0.2, \alpha=0.5$) as an example, the calculated value is $r_d=r'_d=0.34$, but the simulated value is 0.43 for Algorithm 1, and 0.51 for Algorithm 2. However, the simulated values always fall into the region $(r_d, 2r_d)$, so the theoretically calculated values r_d and r'_d reveal the positions of the real threshold values successfully.

5 Conclusion

We have studied performances of pure random walk (PRW) algorithms on a model of random constraint satisfaction problem with growing domains called Model RB. The same threshold behaviors of PRW are shown on Model RB, just like that of Walksat on k-SAT.

From our results, we find that PRW algorithms are more suitable for k-SAT than for Model RB. Taking 3-SAT as an example, Walksat can solve 3-SAT until clause density 2.7, which is not small relative to its satisfiability threshold value of 4.26. But for Model RB, PRW can work until $\frac{1-p}{p}\frac{1}{k\ln N}$, which is very small (tending to 0) relative to its satisfiability threshold value of $-\frac{\alpha}{\ln(1-p)}$ (a constant). This may be due to the fact that the instances of Model RB have large domain size, and a large domain size leads to more constraints and more unsat-constraints, while PRW algorithms can not deal with instances with many unsat-constrains.

In another recent paper, we found out that a backtrack-free algorithm can solve Model RB until a positive constant proportion of $-\frac{\alpha}{\ln(1-p)}$ [30], while it can barely solve k-SAT. Therefore, CSPs with large domain size (such as Model RB) and CSPs with small domain size (such as k-SAT) may have different properties, and different strategies (such as PRW and backtrack-free search) may have different effects on them.

References

1. Achlioptas, D., Kirousis, L., Kranakis, E., Krizanc, D., Molloy, M., Stamatiou, Y.: Random constraint satisfaction: a more accurate picture. In: Smolka, G. (ed.) CP 1997. LNCS, vol. 1330, pp. 107–120. Springer, Heidelberg (1997)
2. Alekhnovich, M., Ben-Sasson, E.: Linear Upper Bounds for Random Walk on Small Density Random 3-cnfs. SIAM J. Comput. 36(5), 1248–1263 (2006)
3. Alphonse, É., Osmani, A.: A model to study phase transition and plateaus in relational learning. In: Železný, F., Lavrač, N. (eds.) ILP 2008. LNCS (LNAI), vol. 5194, pp. 6–23. Springer, Heidelberg (2008)
4. Broder, A.Z., Frieze, A.M., Upfal, E.: On the Satisfiability and Maximum Satisfiability of Random 3-CNF Formulas. In: Proc of SODA, pp. 322–330 (1993)
5. Chao, M., Franco, J.: Probabilistic Analysis of Two Heuristics for the 3-Satisfiability Problem. SIAM J. Comput. 15(4), 1106–1118 (1986)
6. Cocco, S., Monasson, R., Montanari, A., Semerjian, G.: Analyzing search algorithms with physical methods. In: Percus, A., Istrate, G., Moore, C. (eds.) Computational Complexity and Statistical Physics, pp. 63–106. Oxford University Press (2006)
7. Coja-Oghlan, A., Frieze, A.: Analyzing Walksat on random formulas. In: Proc. of ANALCO, pp. 48–55 (2012)
8. Coja-Oghlan, A., Feige, U., Frieze, A., Krivelevich, M., Vilenchik, D.: On smoothed k-CNF formulas and the Walksat algorithm. In: Proc. of SODA, pp. 451–460 (2009)
9. Fan, Y., Shen, J.: On the phase transitions of random k-constraint satisfaction problems. Artif. Intell. 175, 914–927 (2011)
10. Fan, Y., Shen, J., Xu, K.: A general model and thresholds for random constraint satisfaction problems. Artif. Intell. 193, 1–17 (2012)

11. Gao, Y., Culberson, J.: Consistency and random constraint satisfaction problems. J. Artif. Intell. Res. 28, 517–557 (2007)
12. Gent, I., Macintype, E., Prosser, P., Smith, B., Walsh, T.: Random constraint satisfaction: flaws and structure. Constraints 6(4), 345–372 (2001)
13. Jiang, W., Liu, T., Ren, T., Xu, K.: Two hardness results on feedback vertex sets. In: Atallah, M., Li, X.-Y., Zhu, B. (eds.) FAW-AAIM 2011. LNCS, vol. 6681, pp. 233–243. Springer, Heidelberg (2011)
14. Lecoutre, C.: Constraint Networks: Techniques and Algorithms. John Wiley & Sons (2009)
15. Liu, T., Lin, X., Wang, C., Su, K., Xu, K.: Large Hinge Width on Sparse Random Hypergraphs. In: Proc of IJCAI, pp. 611–616 (2011)
16. Liu, T., Wang, C., Xu, K.: Large hypertree width for sparse random hypergraphs. J. Comb. Optim. (2014), doi 10.1007/s10878-013-9704-y
17. Mezard, M., Montanari, A.: Information, Physics and Computation. Oxford University Press (2009)
18. Richter, S., Helmert, M., Gretton, C.: A stochastic local search approach to vertex cover. In: Hertzberg, J., Beetz, M., Englert, R. (eds.) KI 2007. LNCS (LNAI), vol. 4667, pp. 412–426. Springer, Heidelberg (2007)
19. Rossi, F., Van Beek, P., Walsh, T. (eds.): Handbook of Constraint Programming. Elsevier (2006)
20. Smith, B.M., Dyer, M.E.: Locating the Phase Transition in Binary Constraint Satisfaction Problems. Artif. Intell. 81, 155–181 (1996)
21. Semerjian, G., Monasson, R.: Relaxation and Metastability in the Random Walk SAT search procedure. Phys. Rev. E 67, 066103 (2003)
22. Semerjian, G., Monasson, R.: A Study of Pure Random Walk on Random Satisfiability Problems with Physical Methods. In: Giunchiglia, E., Tacchella, A. (eds.) SAT 2003. LNCS, vol. 2919, pp. 120–134. Springer, Heidelberg (2004)
23. Schöning, U.: A probabilistic algorithm for k-SAT based on limited local search and restart. Algorithmica 32, 615–623 (2002)
24. Schöning, U.: A probabilistic algorithm for k-SAT and constraint satisfaction problems. In: Proc. of FOCS, pp. 410–414 (1999)
25. Smith, B.: Constructing an asymptotic phase transition in random binary constraint satisfaction problems. Theoret. Comput. Sci. 265, 265–283 (2001)
26. Wang, C., Liu, T., Cui, P., Xu, K.: A note on treewidth in random graphs. In: Wang, W., Zhu, X., Du, D.-Z. (eds.) COCOA 2011. LNCS, vol. 6831, pp. 491–499. Springer, Heidelberg (2011)
27. Xu, K., Li, W.: Exact Phase Transitions in Random Constraint Satisfaction Problems. J. Artif. Intell. Res. 12, 93–103 (2000)
28. Xu, K., Li, W.: Many Hard Examples in Exact Phase Transitions. Theoret. Comput. Sci. 355, 291–302 (2006)
29. Xu, K., Boussemart, F., Hemery, F., Lecoutre, C.: Random Constraint Satisfaction: Easy Generation of Hard (Satisfiable) Instances. Artif. Intell. 171, 514–534 (2007)
30. Xu, W.: An analysis of backtrack-free algorithm on a constraint satisfaction problem with growing domains (in Chineses). Acta Mathematicae Applicatae Sinica (Chinese Series) (accepted, 2014)
31. Zhao, C., Zheng, Z.: Threshold behaviors of a random constraint satisfaction problem with exact phase transitions. Inform. Process. Lett. 111, 985–988 (2011)
32. Zhao, C., Zhang, P., Zheng, Z., Xu, K.: Analytical and Belief-propagation Studies of Random Constraint Satisfaction Problems with Growing Domains. Phys. Rev. E 85, 016106 (2012)