Benchmark of Model RB: Model Counting Competition 2020

Minghao Yin and Yiyuan Wang
School of Computer Science and Information Technology
Northeast Normal University
Changchun, China
ymh@nenu.edu.cn,yiyuanwangjlu@126.com

Ke Xu
State Key Lab of Software Development Environment,
Beihang University
Beijing, China
kexu@nlsde.buaa.edu.cn

***Abstract*—In previous works, we proposed a new type of random CSP model, called Model RB, which was proved to have both exact phase transitions and many hard instances. In Model Counting Competition 2020 (MC2020), we submit 70 SAT instances based on Model RB.**

I. Benchmark Introduction

It is well known that, to overcome the trivial asymptotic insolubility of the previous random CSP models, Xu and Li [1] proposed a popular CSP model, i.e. Model RB, which is a revision to the standard Model B. It was proved that the phase transitions from solubility to insolubility do exist for Model RB as the number of variables approaches infinity. Moreover, the threshold points at which the phase transitions occur are also known exactly.

It has already been shown, both theoretically [2] and experimentally [3], [4], that Model RB abounds with hard instances in the phase transition region. Based on Model RB, we provided a series of benchmarks (e.g. BHOSLIB) which have been widely used to test the performance of various discrete optimization problems such as clique, vertex covering and set covering problems. Moreover, these benchmarks have also been used in different algorithm competitions such as SAT, MaxSAT, PB, and Answer set programming competitions. Based on Model RB, we can propose a simple random SAT model as follows:

- First generate n disjoint sets of boolean variables, each of which has cardinality n^{α} (where $\epsilon > 0$ is a constant), and then for every set, generate a clause which is the disjunction of all variables in this set, and for every two variables x and y in the same set, generate a 2-clause $\neg x \vee \neg y$.
- Randomly select two different disjoint sets and then generate without repetitions $pn^{2\alpha}$ clauses of the form $\neg x \vee \neg z$ where x and z are two variables selected at random from these two sets respectively (where $0 < p < 1$ is a constant);
- Run Step 2 (with repetitions) for another $rn\ln n$-1 times (where $r > 0$ is a constant);

It is easy to see that to satisfy an instance generated by the model above, exactly one variable from every disjoint set can take value 1, and if this is the case and no random clause is violated by this assignment, then the instances is satisfiable. To hide a satisfying assignment, we first select a variable at random from each disjoint set to form a set of n variables which take value 1, and then in the above process of generating random clauses (Step 2), no clause is allowed to violate this hidden assignment.

We name the instances according to these three values. For instance, frb35-17-2.cnf represents that 35 is the value of n, 17 indicates the value of n^{α} and 2 is the seed value. We submit **70 representative satisfiable problem instances** as our benchmark for the MC2020, which is mainly divided into two parts.

- In the first part, we submit 30 satisfiable instances which are located in the exact phase transition point of Model RB (frb35-17-1∼frb35-17-5, frb40-19-1∼frb40-19-5, frb45-21-1∼frb45-21-5, frb50-23-1∼frb50-23-5, frb53-24-1∼frb53-24-5 and frb56-25-1∼frb56-25-5)[1], which have already used into testing the performance of many SAT algorithms in some SAT competitions. The corresponding parameters are set as follow: $n = 35$ or 56, $\alpha = 0.8$, $r = 0.8/(ln4 - ln3)$, and $p = 0.25$.
- In the second part, according to the parameter settings in Table 1 of [5], we apply four combinations of parameters: 1) $\alpha = 0.7$, $r = 2.3$, and $p = 0.2$; 2) $\alpha = 0.8$, $r = 1.5$, and $p = 0.3$; 3) $\alpha = 0.9$, $r = 2.1$, and $p = 0.3$; 4) $\alpha = 1$, $r = 2$, and $p = 0.35$. We set n to 30 and 59. Following these rules, we generate 40 instances which have many satisfiable assignments, including frb30-10-1∽frb30-10-5, frb30-15-1∼frb30-15-5, frb30-21-1∼frb30-21-5, frb30-30-1∼frb30-30-5, frb59-17-1∼frb59-17-5, frb59-26-1∼frb59-26-5, frb59-39-1∼frb59-39-5 and frb59-59-1∼frb59-59-5.

References

[1] Xu K., Li W. Exact Phase Transitions in Random Constraint Satisfaction Problems. Journal of Artificial Intelligence Research. 2000, vol. 12, pp. 93–103.

[2] Xu K.,Li W. Many Hard Examples in Exact Phase Transitions. Theoretical Computer Science, 2006, vol. 355, pp. 291–302.

[3] Xu K., Boussemart F., Hemery F., et al. A simple model to generate hard satisfiable instances. Proceedings of the 19th international joint conference on Artificial intelligence. 2005, pp. 337-342.

[1] http://sites.nlsde.buaa.edu.cn/ kexu/benchmarks/benchmarks.htm

[4] Xu K., Boussemart F., Hemery F., et al. Random Constraint Satisfaction: Easy Generation of Hard (Satisfiable) Instances. Artificial Intelligence. 2007, vol. 171, pp. 514–534.

[5] Huang P., Yin M., Xu K. Exact phase transitions and approximate algorithm of #CSP. Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence. 2011, pp. 1790–1791.