Abstract:

Exact Solution Counting for Artificial Intelligence based on Decomposition of Constraint Networks

Philippe Jégou, Hanan Kanso, and Cyril Terrioux

Aix-Marseille Université, CNRS, ENSAM, Université de Toulon, LSIS UMR 7296

Counting models in propositional logic (#SAT) and counting solutions for constraint satisfaction problems (#CSP) are challenging problems. They have numerous applications in AI, e.g. in approximate reasoning [1], in diagnosis [2], in belief revision [3], in probabilistic inference [4–7], in planning [8, 9], etc. However, these problems are extremely difficult from a theoretical point of view in terms of complexity because they are #P-hard [10], and moreover, we can claim that they are really hard considering Toda’s theorem which shows that $\text{PH} \subseteq \text{P}^\#P$ [11]. On a practical level, their resolution is also very difficult. So, given the theoretical (and practical) hardness of these problems, most works try to solve the problem by approximating the solution, that is to say, by offering bounds of the number of solutions (or models). Indeed, it is often difficult or impossible to solve these problems accurately i.e. to obtain the exact number of solutions. So, most works have been achieved by sampling the search space [12–15]. All these methods provide a lower bound of the number of solutions with a high-confidence interval obtained by randomly assigning variables until solutions are found. A possible drawback of these approaches is that they might find no solution within a given time limit due to inconsistent partial assignments. For large and complex problems, this results in zero lower bounds or it requires time-consuming parameter (e.g. sample size) tuning in order to avoid this problem. In contrast, by exploiting certain properties of instances, it is possible to provide exact methods that can be efficient w.r.t. theory and practice. We think especially about methods for which polynomial time algorithms may exist as this is the case for example when the constraint network representing the problem has tree-width bounded by a constant [16]. Such search methods that exploit the problem structure, provide time and space complexity bounds like the d-DNNF compiler [17], AND/OR graph search [15, 18] and BTD [19, 20].

So, we propose to improve this class of approaches, using last results about decomposition methods [21–24]. Moreover, our claim is that improving exact methods can also be useful to design better approximate methods. For this purpose, we propose a new algorithm, called #EBTD, which is dedicated to solving exactly #CSP. This algorithm which is based on tree-decomposition improves the previous works [20]. #EBTD ensures non trivial complexity bounds for time and space which are related to structural properties of the considered constraint network: the tree-width and the size of separators. Such an approach, of course, does not allow to improve the theoretical complexity, but it leads to a significant improvement in computation time. Indeed, the experiments we conducted show that this new approach solves more instances on different classes of benchmarks. Above all, the solving is generally faster than for the approaches of the state of the art, like for example sharpSAT [25].
References
