Applied Soft Computing 31 (2015) 339–347

Contents lists available at ScienceDirect

Applied Soft Computing

journal homepage: www.elsevier.com/locate/asoc

Bee colony optimization for the satisfiability problem in probabilistic logic

Tatjana Stojanović[a,*], Tatjana Davidović[b], Zoran Ognjanović[b]

[a] Faculty of Science, University of Kragujevac, R. Domanovića 12, 34000 Kragujevac, Serbia
[b] Mathematical Institute, Serbian Academy of Science and Arts, Kneza Mihaila 36 (P.O. Box 367), 11001 Belgrade, Serbia

ARTICLE INFO

Article history:
Received 14 October 2013
Received in revised form 15 October 2014
Accepted 8 March 2015
Available online 17 March 2015

MSC:
03B60
68T20
68T27

Keywords:
Probabilistic satisfiability
Conditional probability
Approximate probability
Meta-heuristics
Swarm intelligence

ABSTRACT

This paper presents the first heuristic method for solving the satisfiability problem in the logic with approximate conditional probabilities. This logic is very suitable for representing and reasoning with uncertain knowledge and for modeling default reasoning. The solution space consists of variables, which are arrays of 0 and 1 and the associated probabilities. These probabilities belong to a recursive non-Archimedean Hardy field which contains all rational functions of a fixed positive infinitesimal. Our method is based on the bee colony optimizationmeta-heuristic. The proposed procedure chooses variables from the solution space and determines their probabilities combining some other fast heuristics for solving the obtained linear system of inequalities. Experimental evaluation shows a high percentage of success in proving the satisfiability of randomly generated formulas. We have also showed great advantage in using a heuristic approach compared to standard linear solver.

1. Introduction

Working with uncertain knowledge has been well documented problem in mathematical logic and computer science, since the first works of Leibnitz and Bool. The unique solution to this problem has not been found yet, but there are many ideas and different variants of solutions for particular problems, that are used in artificial intelligence. Many of the formalisms for representing and reasoning with uncertainty are based on probabilistic logics [1–10]. These logics are extensions of classical logic with probabilistic operators. Satisfiability problem in these logics (PSAT) can be reduced to linear programming problem. However, solving it by any standard linear solver is inapplicable in practice due to the complexity of the problem. For example, the application of Fourier–Motzkin elimination procedure yields the exponential growth in the number of inequalities in the system. Therefore, the application of some other techniques for solving this problem, such as different types of meta-heuristics, could prove very useful.

Using meta-heuristics for solving satisfiability problems is not a new idea. The most interesting problems in propositional logic are satisfiability problem (SAT) and maximum satisfiability problem (MAX-SAT), i.e., the problem of determining the maximum number of clauses of a given Boolean formula in conjunctive normal form, that can be made true by an assignment of truth values to the variables. Several methods based on different heuristics have been developed for SAT and MAX-SAT. Many of those methods are based on local search procedure and some of them are presented in [11–14]. Heuristics based on swarm intelligence, like Ant Colony Optimization (ACO) or Bee Swarm Optimization (BSO), were also applied to SAT [15,16]. For this type of problem probabilistic approach is presented in [17]. Genetic Algorithm (GA) is another approach used for dealing with SAT and/or MAX-SAT [18,19], and it is also combined with some other heuristics [20]. In probabilistic logics presented by Nilson in [21], Fagin et al. [1] or by Rašković et al. [2], local search based heuristics [22], Tabu Search (TS) [23], GA [24,25], Variable Neighbourhood Search (VNS) [26], and combination of GA and VNS [27] were used for solving PSAT.

Here, we discus the satisfiability problem in approximate conditional probabilities logic described by Rašković et al. in [4]. We denote this version of satisfiability problem with CPSAT-ε. The main differences between PSAT and CPSAT-ε are:

- CPSAT-ε involves conditional probability operator on the contrary to PSAT.
- Probabilities of formulas in CPSAT-ε may take infinitesimal values, and not only real-values as in PSAT.

The first use of infinitesimal was by Leibniz, when he introduced differentials. Later, Robinson [28] showed how infinitely large and infinitesimal numbers can be rigorously defined and used. Characteristics of CPSAT-ε, given above, do not allow us to use the existing methods for PSAT that work with formulas containing real-valued unconditional probabilities only.

The logic, described in [4], enriches the propositional calculus with probabilistic operators which are applied to propositional formulas: $CP_{\geqslant s}(\alpha, \beta)$, $CP_{\leqslant s}(\alpha, \beta)$ and $CP_{\approx s}(\alpha, \beta)$, with the intended meaning that the conditional probability of α given β is "at least s", "at most s" and "approximately s", respectively. This way of knowledge representation and reasoning can be widely used. One of the most obvious examples would be the process of reaching diagnosis in medicine. Occurrence of some symptoms with adequate certainty represented in percents, leads to conclusion that a

[*] Corresponding author. Tel.: +381 642183742.
E-mail addresses: tanjat@kg.ac.rs (T. Stojanović), tanjad@mi.sanu.ac.rs (T. Davidović), zorano@mi.sanu.ac.rs (Z. Ognjanović).

http://dx.doi.org/10.1016/j.asoc.2015.03.017
1568-4946/© 2015 Elsevier B.V. All rights reserved.

specific disease is in question. In addition, in [4] is presented a method for modelling default reasoning with logic with conditional probabilities approximately close to 1. That type of non-monotonic reasoning supports reasoning with incomplete information and can be used in many areas such as medicine, legal reasoning, regulations, specifications of systems and software, etc.

To the best of our knowledge, CPSAT-ε for the logic described in [4] has no practically usable automated solver, and our main effort was to apply meta-heuristics, in particular Bee Colony Optimization (BCO), for solving CPSAT-ε in this logic. BCO is a meta-heuristic technique, which showed very good performance in solving hard optimization problems [29–37]. It is a stochastic, random-search technique that belongs to the class of population-based algorithms. This technique is inspired by the nectar collection process of honey bees from nature. Until now, BCO has not been applied to a class of problems involving search for a solution, only to the problems that already have some feasible solutions and one wants to improve them. The first application of BCO to SAT-type problems was not the only reason for selecting this method to deal with CPSAT-ε. Our experience with other meta-heuristics shows that better performance is achieved with population based methods since they allow the degradation in solution quality. Local search based methods are easily trapped in neighborhoods of the current best solution that may not lead to the global best, i.e., to the solution that satisfies the considered formula. On the other hand, evolutionary methods involve more randomness that may diversify the search and require more time to get to the desired final solution.

To address CPSAT-ε we used the improved variant of BCO, denoted by BCOi and proposed in [38]. The original BCO method is constructive: each bee starts from an empty solution and builds feasible solution through the algorithm steps belonging to the single iteration. Contrary to the constructive version, at the beginning of BCOi iteration some complete solutions are assigned to the bees. The role of each bee is to modify the assigned solution with an aim to improve its quality. Since the modification rules are highly problem dependent, the BCOi concept proposed in [38] needed significant adjustment to be applicable to CPSAT-ε problem. Constructive BCO in CPSAT-ε is not applicable since it is not possible to estimate the quality of final solution based on the partial solutions, i.e., truth values and probabilities assigned to a subset of variables. Our procedure begins with generating random potential solutions for each bee, and in the next steps it tries to improve them, using some additional methods and stochastic moves based on a roulette wheel, in an attempt to produce the real solution. The experimental results obtained by BCOi for CPSAT-ε were compared with those obtained using Fourier–Motzkin elimination procedure and thus demonstrated the superiority of BCOi method.

The rest of the paper is organized as follows. Section 2 gives a brief description of logic with approximate conditional probability. Section 3 briefly outlines the BCO algorithm, while Section 4 describes our implementation of BCOi for solving CPSAT-ε. Section 5 contains some experimental results. Section 6 is devoted to the conclusions.

2. Approximate conditional probabilities

In this section, we present the brief formal introduction to the CPSAT-ε (for a more detailed description see [4]).

The Hardy field $Q[\varepsilon]$ is a recursive non-Archimedean field which contains all rational functions of a fixed positive infinitesimal ε which belongs to a nonstandard elementary extension $^*R$ of the standard real numbers [39,28]. An element ε of $^*R$ is an infinitesimal if $|\varepsilon| < \frac{1}{n}$ for every natural number n. Some examples of infinitesimal are (in ascending order, if $\varepsilon > 0$): $\varepsilon^3 + \varepsilon^4$, $\varepsilon^2 - 5\varepsilon^6$, $\frac{\varepsilon}{100}$, 85ε, or negative infinitesimals: $-\varepsilon$, $-\varepsilon^2$, ... Field $Q[\varepsilon]$ contains all rational numbers. Let S be the unit interval of the $Q[\varepsilon]$ and $Q[0, 1]$ denote the set of rational numbers from $[0, 1]$.

The language of the logic with approximate conditional probability consists of: a countable set $\mathrm{Var} = \{p, q, r, \ldots\}$ of propositional letters, the classical connectives $\neg$, and $\wedge$, and binary probabilistic operators $(CP_{\leqslant s})_{s \in S}$, $(CP_{\geqslant s})_{s \in S}$, and $(CP_{\approx r})_{r \in Q[0,1]}$. The set For_C of classical propositional formulas is defined as usual. The set For_P^S of probabilistic propositional formulas is the smallest set Y containing all formulas of the forms:

- $CP_{\geqslant s}(\alpha, \beta)$ for $\alpha, \beta \in For_C$, $s \in S$,
- $CP_{\leqslant s}(\alpha, \beta)$ for $\alpha, \beta \in For_C$, $s \in S$ and
- $CP_{\approx r}(\alpha, \beta)$ for $\alpha, \beta \in For_C$, $r \in Q[0, 1]$,

and closed under the formation rules: if A belongs to Y, then $\neg A$ is in Y, and if A and B belong to Y, then $(A \wedge B)$ is in Y. Note that neither mixing of pure propositional formulas and probabilistic formulas, nor nested probabilistic operators are allowed. For example, $\alpha \wedge CP_{\geqslant s}(\alpha, \beta)$ and $CP_{\leqslant s}(\alpha, CP_{\geqslant r}(\beta, \gamma))$ are not well formed formulas, while $CP_{>0.5+\varepsilon}(p \wedge q \wedge r, p \vee r) \wedge CP_{\leqslant 0.8-\varepsilon^2}(\neg p \wedge r, \neg p)$ and $\neg CP_{\approx 0.75}((p \vee q) \rightarrow r, \neg p \wedge \neg q)$ are examples of correct formulas. The other classical connectives ($\vee$, $\rightarrow$, $\leftrightarrow$) can be defined as usual. In the rest of the paper $\pm A$ is either A or $\neg A$, while:

- $CP_{<s}(\alpha, \beta)$ denotes $\neg CP_{\geqslant s}(\alpha, \beta)$ for $\alpha, \beta \in For_C$, $s \in S$,
- $CP_{>s}(\alpha, \beta)$ denotes $\neg CP_{\leqslant s}(\alpha, \beta)$ for $\alpha, \beta \in For_C$, $s \in S$,
- $CP_{=s}(\alpha, \beta)$ denotes $CP_{\geqslant s}(\alpha, \beta) \wedge CP_{\leqslant s}(\alpha, \beta)$ for $\alpha, \beta \in For_C$, $s \in S$.

It should be noted that $CP_{\geqslant}$ and $CP_{\leqslant}$ are not interdefinable since the appropriate equivalence breaks down when the probability of the condition equals 0.

We can perform some easy transformations which will reduce the satisfiability problem to checking probabilistic formulas of simpler form. Let A be a probabilistic formula and $p_1, \ldots, p_n$ be the list of all propositional letters from A. An atom a of A is a formula $\pm p_1 \wedge \ldots \wedge \pm p_n$. Note that all pairs of different atoms are mutually exclusive. We use $At(A)$ to denote the set of all atoms from A, and n to denote the number of propositional letters from A. Obviously, $|At(A)| = 2^n$.

Using propositional reasoning it is easy to show that every probabilistic formula A is equivalent to a formula: $DNF(A) = \bigvee_{i=1}^{m} \bigwedge_{j=1}^{k_i} \pm X_{i,j}(p_1, \ldots, p_n)$ called a disjunctive normal form of A, where:

- $X_{i,j} \in \{CP_{\geqslant s}, CP_{\leqslant s}\}_{s \in S} \cup \{CP_{\approx r}\}_{r \in Q[0,1]}$,
- $X_{i,j}(p_1, \ldots, p_n)$ denotes that propositional formulas which are in the scope of the probabilistic operator $X_{i,j}$ are in the complete disjunctive normal form, i.e., propositional formulas are disjunctions of the atoms of A.

Let $\mu : At(A) \rightarrow S$ be a probability measure. We introduce the following abbreviations:

- x_i denotes the measure of the atom $a_i \in At(A)$, $i = 1, \ldots, 2^n$,
- $a_i \vDash \alpha$ means that the atom a_i appears in the complete disjunctive normal form of a classical propositional formula α,
- $\sum(\alpha)$ denotes $\sum_{a_i \in At(A): a_i \vDash \alpha} x_i$, and
- $C\sum(\alpha, \beta)$ denotes $\frac{\sum(\alpha \wedge \beta)}{\sum(\beta)}$.

$[\alpha]$ denotes the set of atoms that satisfy α. Since $[\alpha] = \cup_{a_i \in At(A): a_i \vDash \alpha}[a_i]$, different atoms are mutually exclusive (i.e., $[a_i] \cap [a_j] = \emptyset$ for $i \neq j$).

Now, we can easily define that the formula A is satisfiable if the following holds:

1. if $A \in For_C$ it is satisfiable if $(\exists a_i \in At(A)) a_i \vDash A$,
2. if A has a form $CP_{\leqslant s}(\alpha, \beta)$ it is satisfiable if either $\sum(\beta) = 0$ and $s = 1$ or $\sum(\beta) > 0$ and $C\sum(\alpha, \beta) \leqslant s$,
3. if A has a form $CP_{\geqslant s}(\alpha, \beta)$ it is satisfiable if either $\sum(\beta) = 0$ or $\sum(\beta) > 0$ and $C\sum(\alpha, \beta) \geqslant s$,
4. if A has a form $CP_{\approx r}(\alpha, \beta)$ it is satisfiable if either $\sum(\beta) = 0$ and $r = 1$ or $\sum(\beta) > 0$ and for every positive integer n, $C\sum(\alpha, \beta) \in [\max(0, r - \frac{1}{n}), \min(1, r + \frac{1}{n})]$,
5. $\neg A$ is satisfiable if A is not satisfiable,
6. $A \wedge B$ is satisfiable if A is satisfiable and B is satisfiable.

Therefore, for every conditional probabilistic formula ($\pm CP_{\geqslant s}(\alpha, \beta)$, $\pm CP_{\leqslant s}(\alpha, \beta)$, and $\pm CP_{\approx r}(\alpha, \beta)$) from $DNF(A)$ we can distinguish two cases:

1. the probability of β is zero, in which case