Approximations for Boolean Satisfiability

João Lobo*
Universidade de São Paulo, São Paulo, Brazil
Marcelo Finger†
Universidade de São Paulo, São Paulo, Brazil

Keywords: approximation algorithms, boolean satisfiability, linear programming

The Boolean satisfiability problem is the most notorious and first ever discovered NP-Complete problem. To solve it, state-of-the-art methods still rely on techniques that yield exponential running time in the worst case. The goal of this work, is to study methods used on correlated problems in order to approach solving a SAT instance from an optimization perspective. In particular, linear programming methods have obtained good results when dealing with a closely related problem, the satisfiability problem in many valued logics [1] and, more specifically, the satisfiability problem in Łukasiewicz Logic [2].

We want to develop a family of optimization settings that are simpler to solve than the original SAT, but as the complexity of these simpler problems grows, the family converges to the original problem. Therefore, positively solving one of this easier problems should imply solving the original harder problem.

Stating the problem formally: For a given boolean formula $\varphi(\mathbf{x})$, in clausal normal form, where $\mathbf{x} = (x_1, ..., x_n) \in \{0,1\}^n$. For each boolean variable x_i, we introduce a real variable $z_i \in [0,1]$, thus giving us $\mathbf{z} = (z_1, ..., z_n) \in [0,1]^n$. Let $\mathcal{C}$ be the set of clauses of φ. For every clause c of φ, we produce the following constraint

$$\sum_{x_j \in c} \nu(x_j) \geq 1$$

where

$$\nu(x_j) = \begin{cases} z_j, & \text{if } x_j \text{ appears in } c \\ 1 - z_j, & \text{if } \neg x_j \text{ appears in } c \end{cases}.$$

Lastly, let

$$g_c(\mathbf{x}) = \sum_{x_j \in c} \nu(x_j) - 1.$$

This transforms our constraint into $g_c(\mathbf{x}) \geq 0$.

We claim to want to study the satisfiability of φ by studying the following program

$$\begin{aligned} \text{maximize} \ & \sum_{i=0}^{n} |z_i - 0.5| \\ \text{subject to} \ & g_c(x) \geq 0, \forall c \in \mathcal{C} \\ & 0 \leq z_i \leq 1, i = 1, ..., n \end{aligned} \tag{1}$$

*joaolobo@ime.usp.br
†mfinger@ime.usp.br

It was previously thought that any absolute value programming problem could be easily solved by noticing that $|x| = |x_i^+ - x_i^-| = x_i^+ + x_i^-$, where $x^-, x^+ \geq 0$, if and only if not both x_i^+ and x_i^- are nonzero.

However, in [3] a discussion was given about optimizing absolute value functions, highlighting that, in fact, that technique is only applicable to a maximization problem with nonpositive coefficients (or a minimization problem with nonnegative coefficients); otherwise there are local maxima that the simplex method will converge to [4].

Even though we are assured to not converge directly to an optimal solution (which in this context may or may not be integral), we aim to tinker with the procedure in order to influence it towards an integral solution incrementally.

References

[1] Hähnle, R. Many-valued logic and mixed integer programming. *Annals of Mathematics and Artificial Intelligence* 12:231—263, 1994. https://doi.org/10.1007/BF01530787

[2] Finger, M; Preto, S. Polyhedral semantics and the tractable approximation of Łukasiewicz infinitely-valued logic. *Journal of Logic and Computation* exad059, 2023. https://doi.org/10.1093/logcom/exad059

[3] Hill, T.W.; Ravindran, A. On programming with absolute-value functions. *J Optim Theory Appl* 17:181—183,1975. https://doi.org/10.1007/BF00933924

[4] SHANNO, David F.; WEIL, Roman L. “Linear” programming with absolute-value functionals. *Operations Research* 19.1:120–124,1971.