Chop-SAT: A New Method for Knowledge-Based Agent Decision Making

Thomas C. Henderson, Amelia Lessen, Ishaan Rajan, Tessa Nishida, and Kutay Eken

University of Utah, Salt Lake City UT 84112, USA,
tch@cs.utah.edu,
WWW home page: www.cs.utah.edu/~tch

Abstract. Logical agents base their action selection decisions on inferences made over a logical knowledge base. Given a propositional logic knowledge base expressed in Conjunctive Normal Form (CNF), the knowledge can be converted into a geometrical format, and subsequent analysis takes place as geometrical operations on the feasible region in that representation. We present two novel methods based on this approach in order to: (1) find SAT solutions for the knowledge base (i.e., a truth assignment to each logical variable which makes the CNF sentence true), and (2) find a reasonable approximation to the atom probabilities given the current set of information. This allows agents to determine the semantics (truth) of the world as well as to estimate the probability of truth.

1 Introduction and Background

Given a propositional calculus knowledge base represented in Conjunctive Normal Form (CNF), an agent can find out information about the world by finding new sentences that are entailed by the current knowledge. In this way, an agent can avoid danger and take appropriate actions. In order to do this, it may be useful to determine whether or not the CNF sentence has a solution. If so, the sentence is called *satisfiable*, otherwise it is *unsatisfiable*. This is called the SAT problem (for SATisfiability), and is in NP.

Another aspect of interest to the agent is the probability that a specific atom (logical variable) is true. For example, for the CNF $S = A \vee B$, then since there are three solutions $\{(0,1),(1,0),(1,1)\}$, the probability of A is 2/3, and the probability of B is 2/3. Note that this is the mean of the models (truth assignments to variables) which satisfy the sentence. One way to determine the atom probabilities is to solve the *Probabilistic SAT* (PSAT) problem [12, 7]. That is, given n logical variables, there are 2^n unique truth assignments (also called models) to the variables. The set of all models is called Ω, and ω_i is the model with binary assignments corresponding to the binary representation of $i-1$; e.g., ω_1 is all zero assignments - all false. The PSAT problem is to find a probability distribution, $\pi : \Omega \rightarrow [0,1]$ such that $\sum_{i=1}^{2^n} \pi(\omega_i) = 1$, and $\sum_{\omega_i \models C_j} \pi(\omega_i) = p_j$. The probability of an atom is then found as $Prob(A) = \sum_{\omega_i \models A} \pi(\omega_i)$ We have also previously described how to solve the Probabilistic Sentence Satisfiability Problem (PSSAT) [9] which in certain cases provides a PSAT solution (i.e., given independent variables). The methods we are proposing differ from standard methods in that we solve linear or nonlinear systems of equations rather than having to consider the full joint probability distribution over the variables (e.g., like Bayesian networks [13] or Markov Logic Networks [6]).

The *Chop-SAT* method has been proposed as an alternative way to answer these questions about SAT and PSAT [10, 11]. Previous work on the topic started with Gomory [8] who sought integer solutions for linear programs. Given the semantics of the literals in a disjunction, then a linear inequality can be formed summing x_i for atoms in the clause and $(1 - xi)$ for negated atoms in the clause and setting this to be greater than or equal to 1. Next, a $\{0, 1\}$ solution is sought resulting in an integer linear programming problem. If a non-$\{0, 1\}$ solution is found, Gomory proposed a way to separate (via a *cutting plane*) that solution from all integer solutions. This method has been used in finding lower complexity ways to provide theorems for proving the boundedness of polytopes, cutting plane proofs for unsatisfiable sentences, pseudo-Boolean optimization, etc. (see [1–5]).

Based on the *Chop-SAT* approach, the contributions here provide:

1. A method to determine whether a SAT solution exists, and
2. A method to determine an approximation to the atom probabilities.

2 Chop-SAT

A CNF sentence is the conjunction of a set of disjunctions where each disjunction is a literal (i.e., either an atom or the negation of an atom). A CNF sentence is then represented as $S = C_1 \wedge C_2 \wedge \ldots \wedge C_m$, where $C_i = L_{i,1} \vee L_{i,2} \vee \ldots \vee L_{i,k}$ where $L_{i,j} = a_p$ or $L_{i,j} = \neg a_p$, and a_p is an atom.

A CNF sentence, S, over n variables can be converted to a geometric problem as follows. Consider the hypercube of dimension n; call it H_n. Then each vertex of H_n represents a model for n variables. The vertexes also represent assignments of probability 0 or 1 for the truth of each variable. Every other point in H_n can be considered to give a probability on the interval $[0, 1]$ for each variable. E.g., the center of H_n represents a probability assignment of $1/2$ for each atom.

Next, consider a clause, C_i, of S with k literals. Since it is a disjunction, there is only one truth assignment over its literals which makes it false: namely, where the atom of each literal is assigned the value which makes the literal false. However, every atom not represented by a literal in C_i can take on either truth value and not change the truth of the clause. Thus, there is a sub-hypercube of dimension $n - k$, that is, an instance of H_{n-k}, whose vertexes are not solutions for S. It turns out that there is an $(n - 1)$-dimensional hyperplane which can be constructed so as to separate solutions from non-solutions for C_i. This chopping hyperplane can be positioned anywhere along the edges connecting H_{n-k} to the rest of H_n. Figure 1 shows an example of chopping one vertex ([1,0,1]) from the H_3.

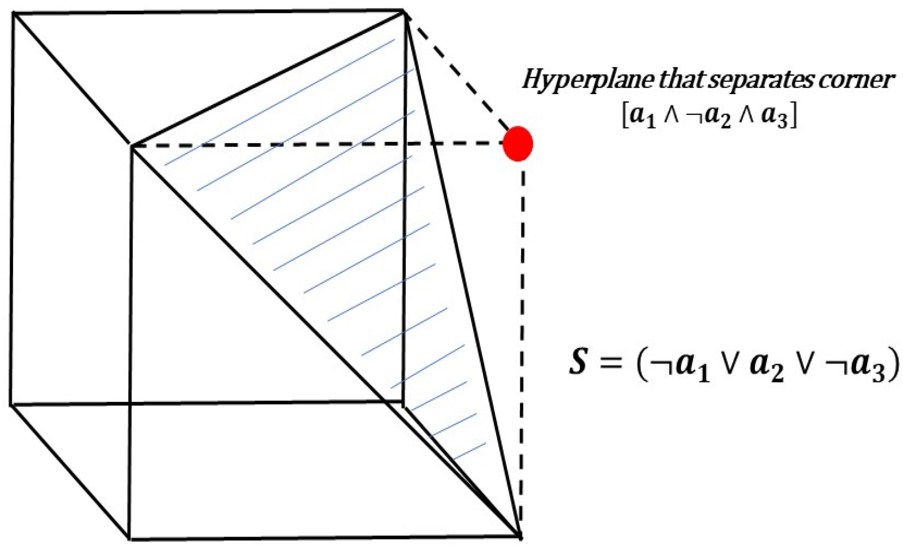

Fig. 1. Example of Chopping [1;0;1] from H_3.

Once all the chops are made (i.e., the intersection of the non-zero half-spaces defined by the hyperplanes), the resulting convex set is called the *feasible region*. Note that it is necessary to include hyperplanes which define the faces of H_n in order to keep the feasible region bounded. Figure 2 shows a simple 2-D example feasible region resulting from chopping off each corner. The hyperplanes in this case are lines in 2-D (each row provides the coefficients a, b, c of the standard form equation of a line: $ax + by + c = 0$):

```
-0.7071   -0.7071    1.0607
-0.7071    0.7071    0.3536
 0.7071   -0.7071    0.3536
 0.7071    0.7071   -0.3536
 1.0000         0         0
-1.0000         0    1.0000
      0    1.0000         0
      0   -1.0000    1.0000
```

where the first four are the corner cuts, and the last four are define the faces (sides) of the square.

3 Solving SAT

Given a feasible region, $\mathcal{F}$, corresponding to a CNF sentence (or knowledge base), it is possible to probe that region to determine if there is a SAT solution

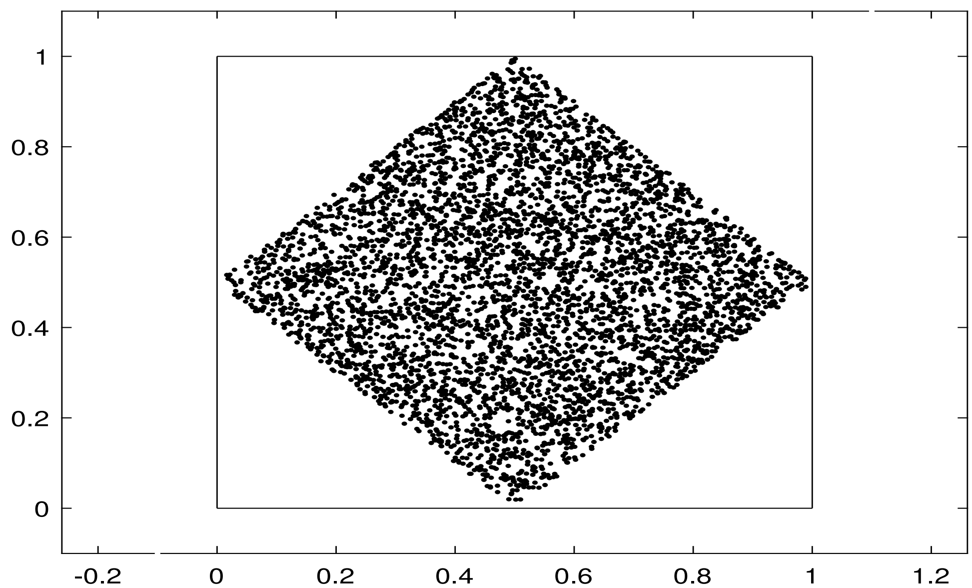

Fig. 2. Example 2-D Feasible Region Resulting from Chopping all Corners o H_2.

(i.e., a corner of H_n in $\mathcal{F}$). Note that the feasible region for any unsatisfiable sentence has no point farther than $\frac{\sqrt{n-2}}{2}$ from the center of H_n. Our approach takes advantage of the fact that the Maximal Volume Inscribed Ellipsoid (MVE) of a set of linear inequalities can be found in polynomial time. Moreover, this ellipsoid will, in general, have its major semi-axis aligned with the most elongated part of $\mathcal{F}$. In order to increase the volume, it is possible to move the hypercube face constraints outward to allow a larger volume for the ellipsoid. Figure 3 demonstrates this idea in 2-D where there is just one solution ([0;0]), and the sides of the square have been moved out 10 units.

To test the practicality of this approach, the following experiment was performed:

- A set of 1000 random knowledge bases over 20 variables were generated.
- The corresponding feasible region was determined for each KB.
- The MVE was found for each feasible region.
- The major semi-axis was found.
- Linear programming was used to find the point in the feasible region which projected to the extreme position along the major semi-axis.
- If the point is greater than $\frac{\sqrt{n-2}}{2}$ away from the center, then a solution is known to exist.

A solution was found in this way for every knowledge base.

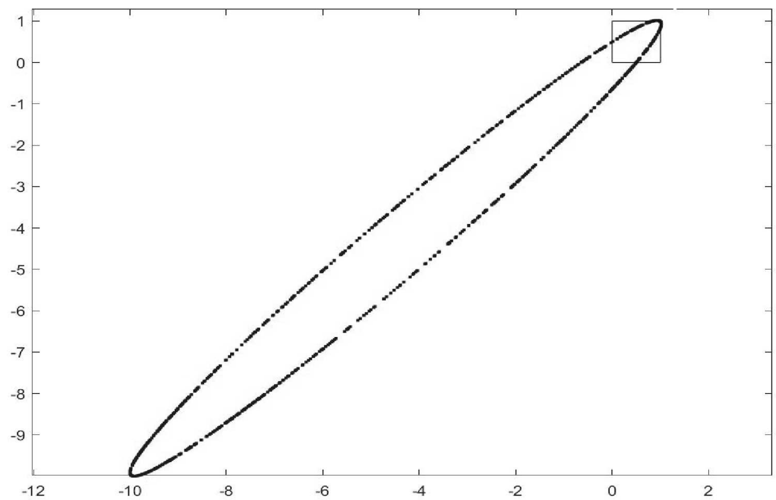

Fig. 3. Example 2-D MVE Showing Major Semi-Axis in Direction of Solution.

4 Solving for Atom Probabilities

Another important problem for an agent is to determine the probability of the state of the world. For example, in the Wumpus World, the agent will die if it enters a cell with a pit. Here we describe a method to determine such probabilities using the feasible region arising from a CNF sentence knowledge base.

For any given satisfiable knowledge base, the atom probabilities are just the average of the 0/1 truth assignments of the models which make satisfy the CNF sentence as described in the introduction. However, since it is too computationally costly to determine all possible solutions and take their average, we propose the following approximations that can be found within a feasible region $\mathcal{F}$:

- the analytic center of $\mathcal{F}$.
- the p-center of $\mathcal{F}$.
- the center of the MVE.
- mean of samples from $\mathcal{F}$.

Analytic Center of $\mathcal{F}$ The *analytic center* of $\mathcal{F}$ is defined as:

$$c = max_y \prod (a \cdot y_h)$$

where $y \in \mathcal{F}$, a is the hyperplane coefficient vector, and y_h is y with a final homogeneous coefficient of 1.

P-Center of $\mathcal{F}$ The p-center of the feasible region is defined as follows: for each hyperplane in the set of linear inequalities, find a point, p, in $\mathcal{F}$ that lies on the hyperplane; next, find the point on the opposite side of $\mathcal{F}$ in the direction of the normal of the hyperplane

Center of MVE of $\mathcal{F}$ The MVE provides both the center of the MVE as well as the directions of the semi-axes. Here, the center of the MVE is used to approximate the atom probabilities.

Mean of Sample Points from $\mathcal{F}$ In this approach, a number of samples are found in the feasible region and their average determined. For example, solve the following:

$$min_x f^T x$$

such that $x \in \mathcal{F}$ and f is taken as the positive and negative unit vector along each axis of the n-D space.

4.1 Experimental Study

For this study, 1000 random knowledge bases were generated with 5 atoms, a maximum of 10 clauses, and at most three literals in a clause. Figure 4 shows the Euclidean distance between the atom probability vector and the four proposed approximations: (1) the analytic center (has lowest error), (2) the p-center (next lowest error), (3) the Chop mean center (similar to p-center error), and (4) the ellipse center which has the most error. Error over all trials was 0.25, 0.52, 0.57, and 1.07, respectively.

Figure 5 shows how the centers cluster. Since the vectors are all 5-dimensional, they are converted to a 2-dimensional representation which maintains spatial coherence. As can be seen, the analytic centers spread similarly to the actual atom probabilities, the MVE centers form a tight central cluster, the p-centers skew somewhat away (up) from the actual atom probabilities, while the Chop mean centers skew a good bit away from the actual probabilities.

5 Wumpus World Experiment

To demonstrate the effectiveness of this approach, agent decision making was tested in the Wumpus World framework. Wumpus World was proposed by Yob [15], and has been used as a standard agent testbed for some time [14]. Wumpus world here is a 4x4 grid of cells; each cell may contain a pit with 20% probability, and if there is a pit it is the only thing in the cell. One cell contains some gold, and one cell contains a Wumpus (the gold may be co-located with the Wumpus). An agent starts in cell $(1, 1)$ and explores the grid in order find the gold. Each cell neighboring a pit has a breeze, and each cell neighboring the Wumpus has a stench. Figure 6 shows an example board with the agent in cell $(1, 1)$ with direction $\theta = 0$, with pits in cells $\{(1, 4), (4, 2), (4, 3)\}$, the Wumpus in cell $(2, 3)$, and the gold in cell $(4, 4)$.

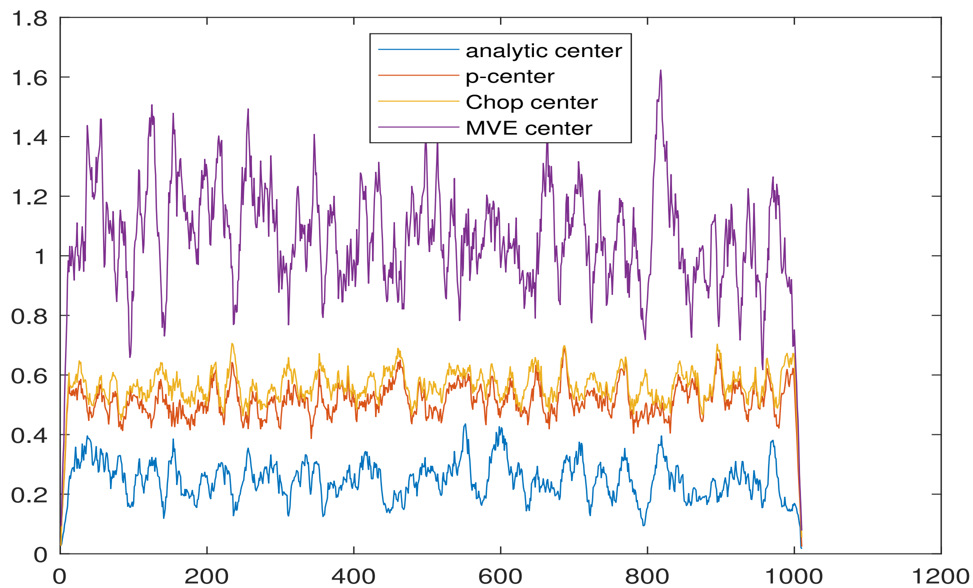

Fig. 4. Error of Approximation Methods: Analytic Center (blue), P-Center (red), Chop center (mustard), MVE center (purple).

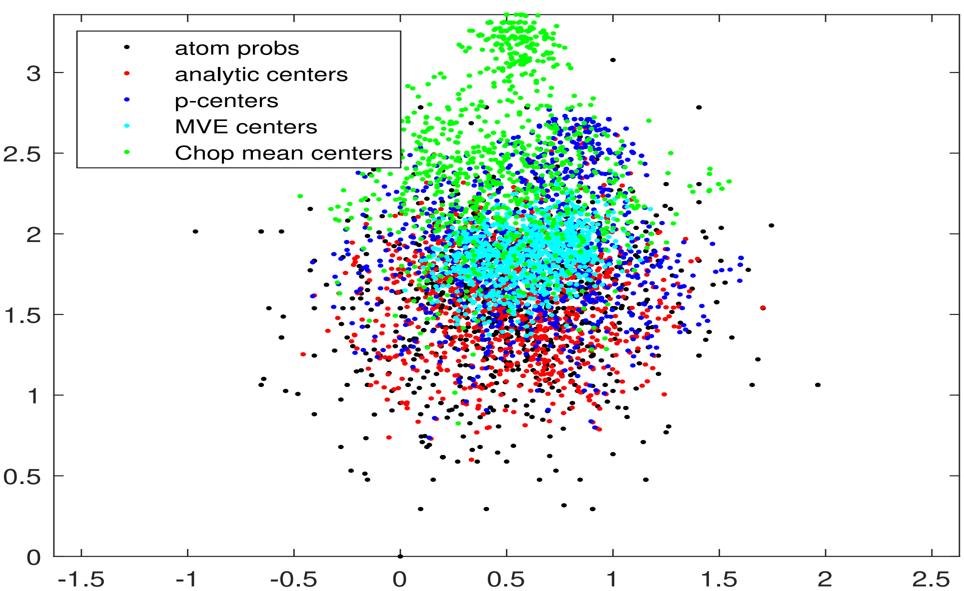

Fig. 5. The center distributions.

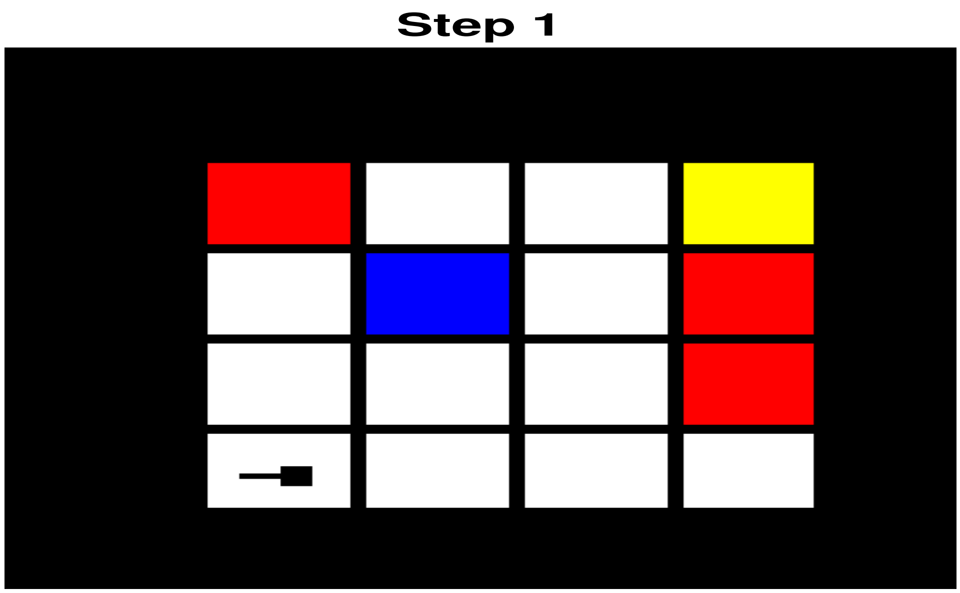

Fig. 6. An Example Wumpus World Board.

The agent has a set of percepts in a cell (represented as bits): (1) stench, (2) breeze, (3) glitter, (4) bump, and (5) scream. The glitter percept lets the agent know it's in the cell with the gold. The agent has a state (x, y, θ), where (x, y) is its location in the grid, and θ is its orientation $\theta \in \{0, 90, 180, 270\}$ degrees. The agent has a set of possible actions: (1) rotate left, (2) rotate right, (3) forward, (4) grab, (5) shoot, and (6) climb. If the agent moves into a cell with a pit or the Wumpus, it dies. The agent has one arrow and can shoot the Wumpus.

The agents here all share a Belief, Desire, Intention cognitive architecture with desires: (1) escape, (2) kill the Wumpus, and (3) explore. If it has the gold, its intention will be to escape with the gold. The agent's intention will be to kill the Wumpus if its location is known. Otherwise, its intention is to explore the grid. In order to explore, the agent selects the lowest probability risk unvisited cell. All agents have the same BDI architecture and only differ in how they compute the probabilities for risk: (1) a human produced algorithm to assign pit and Wumpus probabilities, (2) the analytic center, (3) Chop-SAT mean, and (4) Monte Carlo simulation based on statistics over sample boards satisfying the known percept information.

The experimental method is to generate 1,000 random Wumpus boards, run each agent type on the boards, and measure successful escape with the gold. Note that the Monte Carlo results will be very close to optimal and serves as the upper bound on success. The results are given in Table 1. As can be seen, the Chop-SAT based agent performed the best, and this is a good indication of its efficacy.

Table 1. Results of Wumpus World Experiment.

Number of Boards	**Human**	**Analytic Center**	**Chop-SAT**	**Monte Carlo**
1000	585	598	608	613.3

6 Conclusions and Future Work

The results of the experiments indicate that this geometric approach works well at least up to dimension 20, and that the analytic center is the best approximation to the actual atom probabilities on a random set of knowledge bases. However, the probabilities provided by the Chop-SAT method performed slightly better in the decision making experiment in Wumpus World, and both the analytic center and Chop-SAT methods performed better than a human developed risk probability algorithm.

Future work includes:

– performing SAT solver experiments in higher dimensions,
– using the atom probability approximation in more complex agent decision making situations and compare its effectiveness with other methods.

Acknowledgements

This work was supported in part by National Science Foundation award 2152454.

References

1. S.R. Buss and P. Clote. Cutting Planes, Connectivity, and Threshold Logic. *Archive for Mathematic Logic*, 35:33–62, 1996.
2. V. Chvatal. Edmonds Polytopes and a Hierarchy of Combinatorial Problems. *Discrete Mathematics*, 4:305–337, 1973.
3. V. Chvatal. Cutting Planes in Combinatorics. *European Journal of Combinatorics*, 6:217–226, 1985.
4. W. Cook, C.R. Coullard, and G. Turan. On the Complexity of Cutting-Plane Proofs. *Discrete Applied Mathematics*, 18:25–38, 1987.
5. J. Devriendt, S. Gocht, E. Demirovic, J. Nordstrom, and P.J. Stuckey. Cutting to the Core of Psuedo-Boolean Optimization: Combining Core-Guided Search with Cutting Planes Reasoning. In *Thirty-Fift AAAI Conference on Artificial Intelligence*. Elsevier, 2021.
6. P. Domingos and D. Lowd. *Markov Logic Networks*. Morgan and Claypool Publishers, Williston, VT, 2009.
7. G. Georgakopoulos, D. Kavvadias, and C.H. Papadimitriou. Probabilistic Satisfiability. *Journal of Complexity*, 4:1–11, 1988.
8. R.E. Gomory. Outline of an Algorithm for Integer Solution to Linear Programs. *Bulletin of the Americal Mathematical Society*, 64(5):275–278, 1958.

9. T.C. Henderson, R. Simmons, B. Serbinowski, M. Cline, D. Sacharny, X. Fan, and A. Mitiche. Probabilistic Sentence Satisfiability: An Approach to PSAT. *Artificial Intelligence*, 278:71–87, 2020.
10. Thomas C. Henderson, Amar Mitiche, Xiuyi Fan, and David Sacharny. Some Explorations in SAT. Technical Report UUCS-21-016, University of Utah, July 2021.
11. Thomas C. Henderson, David Sacharny, Amar Mitiche, Xiuyi Fan, Amelia Lessen, Ishaan Rajan, and Tessa Nishida. Chop-SAT: A New Approach to Solving SAT and Probabilistic SAT for Agent Knowledge Bases. In *International Conference on Agents and Artificial Intelligence*, Lisbon, Spain, February 2023.
12. N. Nilsson. Probabilistic Logic. *Artificial Intelligence Journal*, 28:71–87, 1986.
13. J. Pearl. *Probabilistic Reasoning in Intelligent Systems*. Morgan Kaufmann, San Mateo, CA, 1988.
14. Stuart Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach*. Prentice Hall Press, Upper Saddle River, NJ, 3rd edition, 2009.
15. G. Yob. Hunt the Wumpus? *Creative Computing*, September–October 1975.