

Combining a Unit Propagation with Genetic Algorithms to Solve max-SAT Problems

Dalila BOUGHACI¹, Habiba DRIAS², and Belaid BENHAMOU³

^{1,2} Department of Computer Science, University of sciences and technology
BP.32- EL-ALIA Bab-Ezzouar, 16111, Algiers, ALGERIA, Fax:/Phone (+213) 21 24 78 92
dalila_info@yahoo.fr

³LSIS - UMR CNRS 6168, Domaine Universitaire de Saint-Jérôme
Avenue Escadrille Normandie-Niemen, 13397 MARSEILLE CEDEX 20, FRANCE

Abstract

In this paper, we propose and investigate a new strategy for max-SAT problems. The proposed approach consists of a unit propagation strategy combined with a genetic algorithm. Based on the two strategies, we hope to propose an exact solver for max-SAT. The principal goal is to reduce the run-time of the unit propagation – based exact method. The latter is applied on partial assignments with few unspecified variables (generated by the genetic algorithm) what gives a very weak effort of the unit propagation compared to the genetic algorithm effort.

keywords: genetic algorithms, unit propagation, max-SAT, exact method, metaheuristics, hybridization.

1. Introduction

The Satisfiability problem is a core problem in both computational complexity theory and artificial intelligence discipline. Its wide application to the domain of artificial intelligence in automatic reasoning and other domains as VLSI and graph theory motivates the huge interest shown for this problem. Formally, given a collection of m clauses¹ $C = \{ C_1, C_2, \dots, C_m \}$ involving n Boolean variables x_1, x_2, \dots, x_n . The SAT problem is to decide whether an assignment of values to variables exists such that all clauses are simultaneously satisfied. The maximum satisfiability problem (max-SAT) is the optimization variant of SAT. Given a propositional formula in conjunctive normal form CNF², the max-SAT is to find a variable assignment that maximizes the number of satisfied clauses. The decision variants of both SAT and max-SAT are NP-Complete [3, 6].

Many algorithms have been proposed and important progress has been achieved. These algorithms can be divided into two main classes: exact and incomplete algorithms.

- *Complete or exact algorithms:* dedicated to solve the decision version of SAT problem. The well-known algorithms are based on the Davis-Putnam-Loveland procedure [4]. Satz[11] is a famous example of a complete algorithm. A branch and bound algorithm based on Davis-Putnam-Logeman-Loveland procedure DPLL is one of the most competitive exact algorithm for max-SAT[15, 14]. However, BnB³ algorithm can handle relatively small instances with moderate degree of constrainedness defined by [14] as the ratio of the number of clauses to the number of variables of max-SAT.

- *Incomplete algorithms:* they are mainly based on local search and evolutionary algorithms. Tabu search [12, 1], simulated annealing [8], genetic algorithms [5], GRASP [13] and recently mimetic algorithm [2] are examples of incomplete algorithms for SAT. These meta-heuristics are a good approach for finding a near solution of very large instances, in particular for unsatisfiable or unknown instances.

In general, an exact method is guaranteed to obtain an optimal solution but its run-time often increases with the instance size. For that, only small sized instances can be practically solved. To solve larger instances, the only possibility is to use a heuristics approach that can find a near-optimal solution of very large instances in limited time but the optimality is not guaranteed. Recently, there have been very different attempts to combine the exact and heuristics approaches [10, 14, 15].

Aiming at solving difficult max-SAT optimally, we propose in this work, a novel hybrid approach consisting of a genetic algorithm which is an incomplete algorithm combined with an exact strategy of a unit propagation.

¹ A clause is a disjunction of literals. A literal is a variable or its negation.

² A formula in conjunctive form CNF is a conjunction of clauses.

³ BnB: A branch and bound algorithm based on Davis-Putnam-Logeman-Loveland procedure DPLL.

We notice that the exact and approached methods don't use the same representation for their search space. Their hybridization are not obvious since the exact method works on partial assignments incrementally completed by a resolution process (unit propagation in our case) whereas the approached method explores the whole of all the possible assignments. Note that a partial assignment is an assignment where there are unspecified variables (not assigned again). In [10] authors propose a very interesting resolution framework for SAT problems. They introduced a third value of truth « undetermined » in the goal to improve the efficiency of the resolution. based on this framework, we try to extend the genetic algorithms in order to take in account the partial assignments. In the other hand, we add a unit propagation strategy to the extended genetic algorithms to complete the partial configuration created by the genetic algorithms. Consequently, some notions and logical rules must be redefined.

The paper starts with a brief review of the classical genetic evolutionary algorithms. Section 3 presents our proposed approach based on the combination of a genetic algorithm and a unit propagation strategy. Finally, conclusion and future work are explained in section 4.

2. Classical Genetic Algorithms for max-SAT

Genetic algorithms [7, 5] are an evolutionary meta-heuristic that have been used for solving difficult problems. They have been applied to complex optimization problems with remarkable success in some cases. Their behavior mimics the process of a natural evolution. A population initially made of candidate solutions representing individuals improves towards another population of individuals with higher quality along a process repeating a finite number of times, sequentially reproduction between individuals, and mutation of chromosomes and selection of better individuals. The goal is to create a very fit individual.

- **An Individual representation** An individual is represented by a binary chain X (n Vector). Each of whose components x_i receives the value 0 (False) or 1 (True). It is defined as a possible configuration verifying the problem constraints and satisfying the goal that consists in finding an assignment of truth values to the n variables that maximizes the number of satisfied clauses.

- **A crossover:** is an operator used to construct a new population (children) by combining several individuals.

- **A mutation:** is an operator used to generate a modified individual. An elementary mutation consists in flipping one of the variables of the individual.

- A genetic algorithm outline for max-SAT

Input: an instance of satisfiability ;

Output: an assignment of variables that maximizes the number of satisfied clauses;

Begin

Generate at random the initial population;

While (the maximum number of generations is not reached and the optimal solution is not found) **do**

Begin

Repeat

Select two individuals; generate at random a number R_c from $[0, 100]$;

If ($R_c >$ crossover rate) then apply the crossover; generate at random R_m from $[0, 100]$;

While ($R_m <$ mutation rate) *do Begin*

Choose at random a chromosome from the individual obtained by the crossover and flip it;

Generate at random R_m from $[0, 100]$; *End* ;

Evaluate the new individual;

End repeat;

Replace the bad individuals of the population by the fittest new ones.

End;

Return the best chromosome

End.

The algorithm, depicted above, operates as follows. From a population of points (parents), the algorithm constructs a new population (children) in combining several parents (crossover) and applying some random modifications (mutation: consists in flipping a variable). The selection phase chooses the best points among parents and children to produce the next population for the

next iteration. Usually, the genetic algorithms converges, ie, the population has the tendency to lose its diversity, so it loses its efficacy; it is why the convergence is often used like stop criteria. However, the premature convergence of genetic algorithms is an inherent characteristic that makes them incapable of searching numerous solutions of the problem domain why it is frequent to stop searching after a certain number of generations.

3. The Proposed Approach GA-UP

The proposed approach can be viewed as a combination between a population-based global technique and an exact method made by the fit individuals of the population.

Like in a standard genetic algorithm, the population of the new approach is initialized at random or using a heuristic but the representation of an individual is not the same. The individual in the GA-UP (Genetic Algorithm with Unit Propagation) represents a partial truth assignment. It considers three truth values (True, False, Undetermined).

To form a new population for the next generation, higher quality individuals are selected using the objective function described below. The selection phase is identical inform to that used in the classical genetic algorithm selection phase. Once two parents have been selected, their chromosomes are combined and the classical operators of crossover and mutation are applied to generate new individuals. The fit individual (a partial truth assignment) is sent to the unit propagation strategy in order to obtain a complete truth assignment (a solution).

- The Unit Propagation

Unit propagation is the most powerful strategy for SAT [11, 14]. It recursively sets literals in unit clauses to True. It forces the variable in a unit clause to take value that satisfies the clause immediately and ignores the other value completely. The result is a simplified formula since all the clause containing the literal equal to the forced value of the variable can be satisfied (removed) and the negated literal can also be removed from all clauses. However, the run-time often increases with the instance size. In our proposed approach, the procedure UP is applied on the partial assignments (fit individuals of the genetic algorithm) with few unspecified variables. The objective here is to reduce the run-time of an exact method or to improve the quality of the heuristic solution of the genetic algorithms.

- Objective Function

In order to measure the quality of individuals, we use the objective function defined as follow:
In the case of a complete assignment: the goal consists in maximizing the number of satisfied clauses. In the case of a partial assignment: the evaluation function computes the sum of satisfied clauses and also the number of undetermined clauses. The objective consisting in maximizing the number of satisfied clauses. So the fit current individual is the one that augments the number of satisfied clauses. In the case in witch two individuals have the same value, we choose the individual that augments the number of undetermined clauses that in order to minimize the number of unsatisfied (false) clauses.

- A new Individual representation

An individual is represented by a chain X (n Vector). Each of whose components x_i receives the value 0 (False) or 1 (True) or -1 (Undetermined). It is defined as a partial assignment of variables. We define a complete assignment as an assignment of variables with the classical true or false values (all the variable are specified at true or false). A partial assignment is an assignment where undetermined variables appeared (variables not specified again, having the value -1). The set of the truth values is {1, 0, -1} for { True, False, Undetermined}.

The introduction of the third value, need a new interpretation rules. In [10] give some rules. In our work, we consider the following rules:

$$\neg T = F, \quad \neg F = T, \quad \neg U = U, \quad T \vee U = T, \quad F \vee U = U, \quad U \vee U = U.$$

where T: True: 1, F: False: 0, U: Undetermined (unspecified): -1 ,

\vee : OR , \neg : NOT.

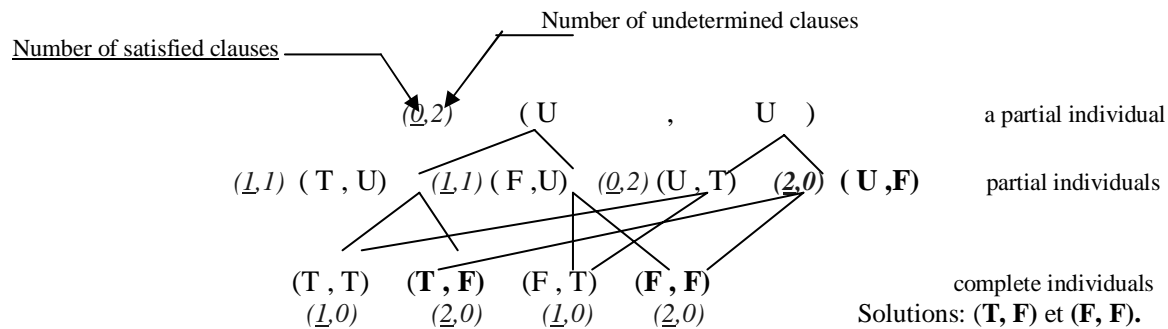
- **The new mutation:** is a **move** to generate modified individuals. It consists in changing the value of one of the variables of the individuals. The possible values are {1, 0, -1}.

$$1 \rightarrow \{0, -1\}, \quad 0 \rightarrow \{1, -1\}, \quad -1 \rightarrow \{0, 1\} \quad \rightarrow \text{is the operator move}$$

- Example:

Let consider the following formula F: $(a \vee \neg b) \wedge (\neg a \vee \neg b)$

The initial individual (U,U). The resolution graph can be given as:



- GA- UP algorithm outline for max-SAT

Input: F: an instance of satisfiability ;

Output: an assignment of variables that maximizes the number of satisfied clauses;

Begin

Generate at random an initial population of individuals;

While (the maximum number of generations is not reached and the optimal solution is not found) **do**

- Apply at first a **genetic algorithm** process and produce a fit individual;

- Let consider V the fit individual,

- F' is a simplified instance corresponding to V (all assigned variables in V are replaced in F to obtain F')

- Complete the partial assignment V by applying a **Unit propagation strategy** on F'

End;

Return the best complete assignment (solution)

End;

4. Conclusion and Future Work

In this paper, we have presented a new method for max-SAT so-called GA-UP. It is a combination between a genetic algorithm and a unit propagation strategy. Our objective is to determine how effective the new approach is in comparison with the classical genetic algorithm. Actually, we are working on the implementation and tests results. As soon work, we plan to improve our study by providing experimental results concerning several DIMACS difficult Benchmarks taken from SATLIB[9].

REFERENCES

1. D.Boughaci H.Drias "Efficient and Experimental Meta-heuristics for MAX-SAT Problems". *LNCS, Volume 3503 / 2005*, WEA 2005: pp:501-512.
2. D. Boughaci, H Drias and B. Benhamou, "Solving Max-SAT Problems Using a memetic evolutionary Meta-heuristic", in *proceedings of 2004 IEEE CIS 2004*, pp480-484,2004.
3. S.Cook . "The Complexity of Theorem Proving Procedures", in *the Proceedings of the 3rd Annual ACM Symposium on the Theory of Computing*, 1971.
4. M.Davis, G.Putnam , and D.Loveland. "A machine program for theorem proving", *communication of the ACM*, 394-397, Jul 1962.
5. J.Frank, "A study of genetic algorithms to find approximate solutions to hard 3CNF problems", in *proceedings of Golden West international conference on artificial intelligence*,1994.
6. MR.Garey, S.Johnson. "Computer and intractability, a guide to the theory of NP-Completeness, Freeman company Sanfranscso, 1978.
7. D. E Goldberg,."Genetic Algorithms in search, Optimization & Machine Learning". Wokingham, Addison-Wesley. 1989)
8. P.Hansen P; B.Jaumard. "Algorithms for the Maximum Satisfiability", *journal of computing* 44-279-303, 1990.
9. H. H. Hoos and T. Stutzle. SATLIB: An online resource for research on SAT. In I. Gent, H. van Maaren, and T. Walsh, editors, *SAT2000: Highlights of Satisfiability Research in theYear 2000*, Frontiers in Artificial Intelligence and Applications, pages 283–292. Kluwer Academic, 2000.
10. F. Lardeux, F. Saubion and J.K Hao. "Une vision trivaluée pour les problèmes SAT et MAX-SAT », *JFPC 2005*.
11. CMLi and. Anbulagan, "Heuristic based on unit propagation for satisfiability", in *Proceedings of CP 97*, springer-Verlag, LNCS 1330, pp 342-356, Austria, 1997.
12. B.Mazure, L.Sais and E.Greoire,." A Tabu search for Sat", in *proceedings of AAAI 1997*.
13. PM.Pardalos, L.Pitsoulis and MGC. Resende. "A Parallel GRASP for MAX-SAT Problems. PARA96 Workshop on Applied Parallel Computing in Industrial Problems and Optimization", Lyngby. Denmark August 18-21, 1996.
14. Z.Xing, W.Zhang, MaxSolver: An Efficient Exact Algorithm for (Weighted) Maximum Satisfiability, *Artificial Intelligence 2005*, to appear.
15. H. Zhang, H. Shen "Exact Algorithms for MAX-SAT", *Electronic Notes in Theoretical Computer Science* 86, N°1, 2003.