RACSAM
Rev. R. Acad. Cien. Serie A. Mat.
VOL. **98** (1), 2004, pp. 55–63
Ciencias de la Computación / Computational Sciences

Satisfiability and matchings in bipartite graphs: relationship and tractability

Belaid Benhamou

Abstract. Satisfiability problem is the task to establish either a given CNF logical formula admits a model or not. It is known to be the canonical NP-complete problem. We study in this note the relationship between matchings in bipartite graphs and the satisfiability problem, and prove that some restrictions of formulae including the known r-r-SAT[1] class are trivially satisfiable. We present an algorithm which finds a model for such formulas in polynomial time complexity if one exists or, failing this, proves in polynomial time complexity that the current formula is not an element of the restricted class.

Satisfacibilidad y equiparación en grafos bipartitos: relaciones y tratabilidad

Resumen. El problema de la satisfacibilidad consiste en establecer si una fórmula lógica CNF dada admite o no un modelo. Se conoce como el problema canónico NP completo. En este trabajo se estudia la relación entre las equiparación en grafos bipartitos y el problema de la satisfacibilidad y se demuestra que algunas restricciones de las fórmulas, entre ellas la clase conocida r-r-SAT, son satisfacibles de forma trivial. Se presenta un algoritmo que encuentra, si existe, un modelo para tales fórmulas en tiempo polinomial y, en su defecto, demuestra, también en tiempo polinomial, que la fórmula actual no es un elemento de la clase restringida.

1. Introduction

Cook [6] has shown that 3-SAT, the boolean satisfiability problem restricted to instances with exactly three variables per clause, is NP-complete. However, some restrictions of the SAT problem are Known to be tractable. Among them 2-SAT and Horn-SAT[2] instances. Also, we can find in [1, 19, 18, 23, 15] different good algorithms which recognize Horn-Renaming-SAT[3] instances and in [13, 20, 7] methods which check their satisfiability in a linear time complexity. More general classes are the Q-Horn class introduced in [8, 10, 9] which mixes both 2-SAT and Horn-SAT instances and the Galo and Scutella hierarchy[12]. Genisson and Rauzy showed in [21] that most of the previous restrictions are polynomial for the Davis and Putnam procedure.

Horowitz and Sahni [16] discussed the importance of finding possible restrictions under which a problem remain NP-complete. Tovey in [22] reduced 3-SAT to 3-SAT where each variable appears in at most

Presentado por Luis M. Laita.
Recibido: December 15, 2003. *Aceptado*: October 13, 2004.
Palabras clave / Keywords: SAT, Bipartite graphs.
Mathematics Subject Classifications: 68T20.

[1] r-s-SAT: denotes the class of instances with exactly r variables per clause and at most s occurrences per variable.
[2] Horn-SAT: is the set of SAT instances which contain only horn clauses.
[3] Horn-renaming-SAT: is the set of instances which become Horn-SAT after renaming some variables.

four clauses (notation 3-4-SAT), he showed that the stronger restriction is 3-4-SAT and 3-3-SAT or r-r-SAT in general is in fact trivial.

In this note we discuss the relationship between the Satisfiability problem and matchings in bipartite graphs. We give a new representation for the satisfiability problem by using bipartite graphs. We show that some restricted instances of the SAT problem form a polynomial class and prove that the r-r-SAT class of Tovey is a sub-class of that polynomial class . Let (S, V_S) denote a SAT instance where S is the set of clauses and V_S the set of variables occurring in S. We prove that the instance (S, V_S) is trivially satisfiable when the cardinal of each subset of clauses C of S is less than or equal to the one of its corresponding variable subset V_C (i.e. if $\mid C \mid$ [4] $\leq \mid V_C \mid$ for all $C \subseteq S$, then (S, V_S) is trivially satisfiable). We present an algorithm which either finds a model, if one exists, or failing this, finds a subset $C \subseteq S$ such that $\mid C \mid > \mid V_C \mid$ with a worst case complexity of order $\theta(\mid S \mid^2)$. This proves that the class of SAT instances $\mathcal{I} = \{(S, V_S) / \mid C \mid \leq \mid V_C \mid$ for all $C \subseteq S\}$ is in P. Further, we show that each instance $(S, V_S) \in$ r-r-SAT satisfies the condition $\mid A \mid \leq \mid V_A \mid$ for all $A \subseteq S$ of the class $\mathcal{I}$. Thus, r-r-SAT is a subclass of $\mathcal{I}$.

The rest of this note is organized as follows. Section 2**.** gives the relationship between Boolean Satisfiability problem and matching theory in bipartite graphs. In Section 3**.**, we show that the class $\mathcal{I}$ is in P and prove that r-r-SAT is a subclass of $\mathcal{I}$. In Section 4**.** we study an extended class $\mathcal{J}$ which contains the class $\mathcal{I}$. Some experiment on random SAT instances are done in Section 5**.** to estimate both classes $\mathcal{I}$ and $\mathcal{J}$. Section 6**.** concludes the work.

2. Satisfiability and matchings in bipartite graphs

First we give some graph theory definitions that we shall use to prove our results on the Satisfiability problem. A graph is a statement $G = (X, E)$ in which X is the set of vertices and E the set of edges drawn between pairs of vertices of X. For any subset A of vertices in G, we define the neighbor set of A in G to be the set of all vertices adjacent to vertices in A; this set is denoted by $N_G(A)$.

A subset $M \subseteq E$ is called a matching in G if its elements are links and no two of them are adjacent in G. A matching M saturates a vertex v, and v is said to be M-saturated, if some edge of M is incident with v; otherwise, v is M-unsaturated. A subset X of vertices is M-saturated if only if each vertex in X is M-saturated. The matching is perfect when all the vertices in G are M-saturated; it is said maximum if G has no matching $M^{'}$, such that $\mid M^{'} \mid > \mid M \mid$. An M-alternating path in G is a path whose edges are alternatively in E/M and in M. An M-augmenting path is an M-alternating path whose origin and terminus are M-unsaturated.

Definition 1 *Let X and Y be subsets of vertices in a graph G, then $G = (X, Y, E)$ is a bipartite graph if only if the following conditions hold :*

1. *The pair (X, Y) defines a bipartition of the vertex set in G,*
2. *Each edge is drawn between two vertices of the different subsets of vertices, i.e $E = \{e/e = (x, y), x \in X\ and, y \in Y\}$.*

Suppose, now that G is a bipartite graph with a bipartition (X, Y), in many applications one wishes to find a matching of G that saturates every vertex in X; an example is the personnel assignment problem which consists in assigning in a company n workers to n jobs, such that each worker is assigned to exactly one job among those he is qualified for.

We show in the sequel how to express SAT instances as bipartite graphs and how to adapt the notion of matchings to prove their satisfiability.

Proposition 1 *Each SAT instance (S, V_S) where S is the set of clauses and V_S its set of variables is represented by a bipartite graph.*

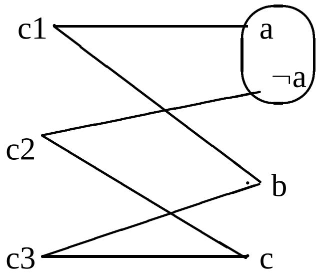
[4]$\mid S \mid$: is the cardinal of the set C

PROOF. Consider the graph $G = (S, V_S, E)$ in which S is the set of vertices corresponding to the clauses (called clause-vertices) and V_S the set of vertices corresponding to the variables (called variable-vertices) and E the set of edges drawn between clause vertices and its variable vertices. There will be a link between a vertex $c \in S$ and a vertex $v \in V_S$ if and only if the variable v appears in the clause c (by 'appears' we mean that it or its negation is in the clause). By construction, G is a bipartite graph.

Definition 2 *The bipartite graph $G = (S, V_S, E)$ corresponding to a SAT instance (S, V_S) is its SAT-graph*

Definition 3 *A matching in a SAT-graph $G = (S, V_S, E)$ is a set of edges defined between the clause-vertices of S and the variable-vertices of V_S such that no variable is saturated in both parities positive and negative.*

Example 1 *Let's $(S, V_S) = \{c_1 = a \vee b, c_2 = \neg a \vee c, c_3 = b \vee c\}$ be a SAT instance where S=$\{c_1, c_2, c_3\}$ and V_S=$\{a, b, c\}$. Its SAT-graph is the following:*



The literals a and $\neg a$ are in a potatoe because they define the same variable vertex. The set $M = \{(c_1, b), (c_2, \neg a), (c_3, c)\}$ defines a matching for the previous SAT-graph which saturates the set of clause vertices S, but the set M=$\{(c_1, a), (c_2, \neg a), (c_3, c)\}$ is not a matching, since it saturates both a and $\neg a$ at the same time.

Definition 4 *A matching M in a SAT-graph G=(S, V_S, E) satisfies the corresponding instance (S, V_S) if only if each clause in S has at least one of its variable-vertices M-saturated.*

See that the matching M=$\{(c_1, b), (c_2, \neg a)\}$ satisfies (is a model of) the instance of the example 1, but does not saturate the set of clauses, since c_3 is not M-saturated.

We prove in the following that to find a matching in a SAT-graph $G = (S, V_S, E)$ which satisfies (S, V_S) is not an easy problem and is in fact NP-complete.

Theorem 1 *The problem of deciding existence of a matching in a SAT-graph $G = (S, V_S, E)$ which satisfies the instance (S, V_S) is NP-complete.*

PROOF. The proof consists in reducing the satisfiability problem decision of an instance (S, V_S) to the problem of existence of a matching in the SAT-graph $G = (S, V_S, E)$ which satisfies (S, V_S). The reduction consists in representing each clause in S of the instance (S, V_S) by a clause vertex in the corresponding SAT-graph $G = (S, V_S, E)$, and then draw edges from the clause vertex to each of its variables (variables vertices in the SAT-graph). Clearly, there will be at most $r \mid S \mid$ edges if r is the maximum size of the clauses in S. Thus, we get a polynomial reduction whose complexity is $\mathcal{O}(r \mid S \mid)$ in the worst case. On other hand it is clear that proving satisfiability of (S, V_S) is equivalent to showing existence of a matching in the corresponding SAT-graph $G = (S, V_S, E)$ which satisfies (S, V_S). The last problem is then NP-complete (QED).

Proposition 2 *Let (S, V_S) be a SAT instance and $G = (S, V_S, E)$ its corresponding SAT-graph, if G has a matching which saturates S then the instance (S, V_S) is satisfiable.*

PROOF. Suppose that M is a matching in G which saturates S, then each clause vertex in S is linked with a different variable vertex of V_S. Thus, a trivial model of the instance (S, V_S) is to take each variable of the matching in its parity positive-negative as it appeared in the clause it is matched with.

Remark 1 *A matching of a SAT-graph $S = (S, V_S, E)$ which saturates S satisfies the instance (S, V_S), but the converse is in general false.*

3. A tractable restriction of SAT, the class $\mathcal{I}$

Let us now use the notion of matching in SAT-graphs to prove that some instances are trivially satisfiable. Before doing this, we summarize a main theorem of Berge [2] which we will use to prove some results on satisfiability.

Theorem 2 *A matching M of a graph G is a maximum matching if and only if G contains no M-augmenting path.*

PROOF. See Berge [2].

Now we give a result on the satisfiability problem which is based on the property of Matchings in bipartite graphs given by P. Hall [14].

Theorem 3 *Let (S, V_S) be a SAT instance, if $\mid V_C \mid \geq \mid C \mid$ for each $C \subseteq S$ then (S, V_S) is satisfiable.*

PROOF. Let $G = (S, V_S, E)$ be the SAT-graph corresponding to the instance (S, V_S), we shall prove that under the condition: $\mid V_C \mid \geq \mid C \mid$ for each $C \subseteq S$ (1), there exists a matching saturating the clause vertex set S. Conversely, suppose that G satisfies the condition (1), but G contains no matching saturating all the clause vertices of S. We shall then obtain a contradiction. Let $M^{'}$ be a maximum matching in G; by our supposition, $M^{'}$ does not saturate all the vertices in S. Let c be an $M^{'}$-unsaturated clause vertex in S, and let Z denote the set of all vertices connected to c by $M^{'}$-alternating paths. Since $M^{'}$ is a maximum matching, it follows from theorem 2 that c is the only $M^{'}$-unsaturated clause vertex in Z. Set $C = Z \cap S$ and $V_C = Z \cap V_S$, clearly, the clauses in C/c are matched under $M^{'}$ with the variables in V_C since every variable vertex in V_C is connected to the clause c by an $M^{'}$-alternating path and $M^{'}$ is a maximum matching. Therefore $\mid V_C \mid = \mid C \mid -1$, then $\mid V_C \mid \leq \mid C \mid$ and this contradict the assumption (1), QED.

The above proof provides the basis of a good algorithm (see figure 1) for finding a model for SAT instances (S, V_S) satisfying the condition of the previous theorem (i.e. $\mid V_C \mid \geq \mid C \mid$ for each $C \subseteq S$), or failing this, shows in a polynomial time complexity that the condition of theorem 3 doesn't hold.

The method uses the same principle as the search method for matchings in bipartite graphs [17]. The basic idea is very simple. We start with an arbitrary partial matching M. If M saturates all the clause-vertices in S , then it is a model of (S, V_S), since each clause in S will be satisfied when we take in M each variable in its parity positive-negative as it appears in the linked clause of M. If not, we choose an M-unsaturated clause-vertex c in S and systematically search for an M-augmenting path in the graph $G = (S, V_S, E)$ with the origin c (Edmonds in [11], suggested the first method). The search method of figure 1, finds such a path P if one exists; in this case $M^{'} = M \triangle^{5} E(P)$ is a larger matching and saturates more clauses in S. We then repeat the procedure with $M^{'}$ instead of M.

If such a path does not exist, the set Z of all vertices which are connected to the clause-vertex c by M-alternating paths is found. Then (as in the proof of the previous theorem) $C = Z \cap S$ satisfies the condition $\mid V_C \mid < \mid C \mid$ which violates the conditions of the theorem 3.

Correctness and Complexity: correctness of the algorithm of figure 1 is guaranteed by theorem 3. This algorithm can run at most $\mid S \mid$ times the step 3 before finding either a set $C \subseteq S$ such that $\mid V_C \mid < \mid C \mid$ or

[5] $\triangle$: is the symmetric difference

Procedure (1) model-search((S, V_S): a Sat instance represented by its Sat-graph)
Input: an instance (S, V_S) represented by its Sat-graph.
Output: a model for (S, V_S), otherwise a set $A \subseteq S$ such that $\mid V_A \mid < \mid A \mid$.

1. Start with an arbitrary matching M.
2. If M saturates every clause vertex in S, then it is a solution, stop. Otherwise, let c be an M-unsaturated clause in S. Set $C = \{c\}$ and $T = \emptyset$.
3. If $\mid V_C \mid = \mid T \mid$ then $\mid V_C \mid < \mid C \mid$, since $\mid T \mid = \mid C \mid -1$, stop the condition of theorem 3 doesn't hold. Otherwise, let $v \in V_C/T$.
4. If v is M-saturated (linked with $c^{'}$), let $(c^{'}, v) \in M$, replace C by $C \cup \{c^{'}\}$ and T by $T \cup \{v\}$ and go to step 3. Otherwise, let P be an M-augmenting (c, v)-path. Replace M by $M^{'} = M \triangle E(P)$ and go to step 2.

Figure 1. Procedure 1: matching saturation algorithm

an M-augmenting path, and since the initial matching (step 2) can be augmented at most $\mid S \mid$ times before the solution (S is M-saturated) is found, it is clear that the method is a good algorithm and its complexity in the worst case is $\mathcal{O}(\mid S \mid^2)$. Then We proved the following result:

Proposition 3 *The problem of deciding satisfiability of a SAT instance in the class* $\mathcal{I} = \{(S, V_S) / \mid V_C \mid \geq \mid C \mid$*, for each* $C \subseteq S\}$ *is polynomial (notation* $\mathcal{I} \in P$*).*

PROOF. The algorithm of figure 1 finds a model in a polynomial time complexity when the instance is in $\mathcal{I}$. Otherwise, the same algorithm proves in a polynomial time complexity that the instance is not in $\mathcal{I}$.

Now we show that the r-r-SAT class is a sub-class of the class $\mathcal{I}$. Before doing this, we need to prove the following proposition:

Proposition 4 *Let* (S, V_S) *be a SAT instance,* $c \in S$, $v \in V_S$, $size(c)$ *the size of the clause* c *(i.e. the number of variable in the clause) and* $occ(v)$ *the number of occurrences of the variable* v *in* S *.*
If $\mathcal{I}^{'} = \{(S, V_S)/min_{c \in S}[size(c)] \geq max_{v \in V_S}[occ(v)]\}$*, then* $\mathcal{I}^{'} \subset \mathcal{I}$*.*

PROOF. We prove that under the condition: $min_{c \in S}[size(c)] \geq max_{v \in V_S}[occ(v)]$, each subset $C \subseteq S$ satisfies the condition $\mid V_C \mid \geq \mid C \mid$ (the condition of theorem 3). Set $min_{c \in S}[size(c)] = r$ and let C be a subset of S, then

$$\sum_{c \in C} size(c) \geq r \mid C \mid \qquad (1);$$

On other hand,

$$\sum_{v \in V_C} occ(v) \geq \sum_{c \in C} size(c) \quad (2);$$

because the clauses of C are formed by the variables of V_C which can eventually appear in clauses of S/C. We deduce from (1) and (2) the following:

$$\sum_{v \in V_C} occ(v) \geq r \mid C \mid \qquad (3);$$

As $\forall\ v \in V_C$, we have $1 \leq occ(v) \leq r$, then we get the following:

$$\sum_{v \in V_C} occ(v) \leq r \mid V_C \mid \qquad (4);$$

From (3) and (4) we deduce: $r \mid V_C \mid \geq r \mid C \mid$, and then $\mid V_C \mid \geq \mid C \mid$, QED.

Remark 2 *The instances of* $\mathcal{I}^{'}$ *are trivially satisfiable, since* $\mathcal{I}^{'} \subset \mathcal{I}$*.*

We show now that the class $\mathcal{I}$ includes Tovey's r-r-SAT class:

Proposition 5 *r-r-SAT* $\subset \mathcal{I}^{'} \subset \mathcal{I}$

PROOF. The class r-r-SAT is a restriction of the class $\mathcal{I}^{'}$ to instances whose clauses are exactly of size r and the maximum of occurrences of the variables is r. Clearly the condition $\{min_{\, c\in S}[size(c)] \geq max_{v\in V_S}[occ(v)]\}$ of proposition 4 holds for each instance of r-r-SAT, thus r-r-SAT is a sub-class of $\mathcal{I}^{'}$ which is itself a sub-class of the class $\mathcal{I}$.

4. A tractable extension of the class $\mathcal{I}$, the class $\mathcal{J}$

Until now, we identified polynomial classes in which each variable is matched with a different and a single clause. Here, we extend the algorithm of figure 1. by considering matchings which satisfy the set of clauses, rather than the only ones which saturate it.

Example 2 *Let's* $(S, V_S) = \{c_1 : a \vee b, c_2 : \neg a \vee b, c_3 : \neg b \vee a\}$ *be a SAT instance where* S=$\{c_1, c_2, c_3\}$ *and* V_S=$\{a, b, c\}$*.*

We can see that the matching M=$\{(c_1, a), [c_2, b)\}$ satisfies all the clauses of S, it is then a model of the instance (S, V_S). However, there is no matching which saturates the set of clauses S. It is then more important to search matchings statisfying the set of clauses than searching only the ones saturating it.

Eventhough the problem of existance of matchings in a SAT-graph which satisfy the set of clauses is NP-complete in general case, there are some restrictions for which it remains polynomial. That is what we will show in the sequel by generalizing the procedure of figure 1. as follows.

Procedure (2) model-search((S, V_S): a Sat instance represented by its Sat-graph)
Input: an instance (S, V_S) represented by its Sat-graph.
Output: a model for (S, V_S), otherwise a set $A \subseteq S$ such that $\mid V_A \mid < \mid A \mid$.

1. Start with an arbitrary matching M.
2. If M satisfies every clause vertex in S, then it is a solution, stop. Otherwise, let c be an M-unsatisfied clause in S. Set $C = \{c\}$ and $T = \emptyset$.
3. If $\mid V_C \mid = \mid T \mid$ then $\mid V_C \mid < \mid C \mid$, since $\mid T \mid = \mid C \mid -1$, stop the condition of theorem 3 doesn't hold. Otherwise, let $v \in V_C / T$.
4. If v is M-saturated (linked with $c^{'}$), let $(c^{'}, v) \in M$, replace C by $C \cup \{c^{'}\}$ and T by $T \cup \{v\}$ and go to step 3. Otherwise, let P be an M-augmenting (c, v)-path. Replace M by $M^{'} = M \triangle E(P)$ and go to step 2.

Figure 2. Procedure 2: matching satisfaction algorithm

The extend algorithm of figure 2 is obtained from the first algorithm (figure 1) by changing the saturation check of step 2 by a check of satisfiability.

Proposition 6 *If* $\mathcal{J}$ *is the set of instances shown to be satisfiable by the extended algorithm, then* $\mathcal{I} \subset \mathcal{J}$

PROOF. Procedure (2) is an extension of procedure (1), since it computes matchings which satisfy the set of clauses and which include the ones saturating it. Procedure (1) deals only with matchings which saturate the set of clauses, thus $\mathcal{I} \subset \mathcal{J}$.

Proposition 7 $\mathcal{I} \subset \mathcal{J} \in P$

PROOF. It is easy to see that the algorithm of figure 2 has the same complexity as the one of figure 1 which is of order $\mathcal{O}(|\ S\ |^2)$. As $\mathcal{I}$ had been shown to be a subset of $\mathcal{J}$, the result holds.

5. Experiments

We give in this section an estimation of the classes $\mathcal{I}$ and $\mathcal{J}$ on random SAT instances. Both procedures (1 and 2) are tested on 3-SAT random generated instances where the number of variables is fixed to 200 and where the ratio ($\frac{c}{v}$) of the number of clauses c to the number of variables varies from 0 to 4 (by a step of 0.5). A sample of 200 random instances are tested on each point. It is well known that the hard instances are those having ratios close to the critical value 4.25 ($\frac{c}{v} = 4.25$) corresponding to the transition phase (see [3]). Problems having ratios less than 4.25 are all most satisfiable and those having greater ratios are all most unsatisfiable. Only satisfiable instances are considered in our experiments ($\frac{c}{v} \leq 4$).

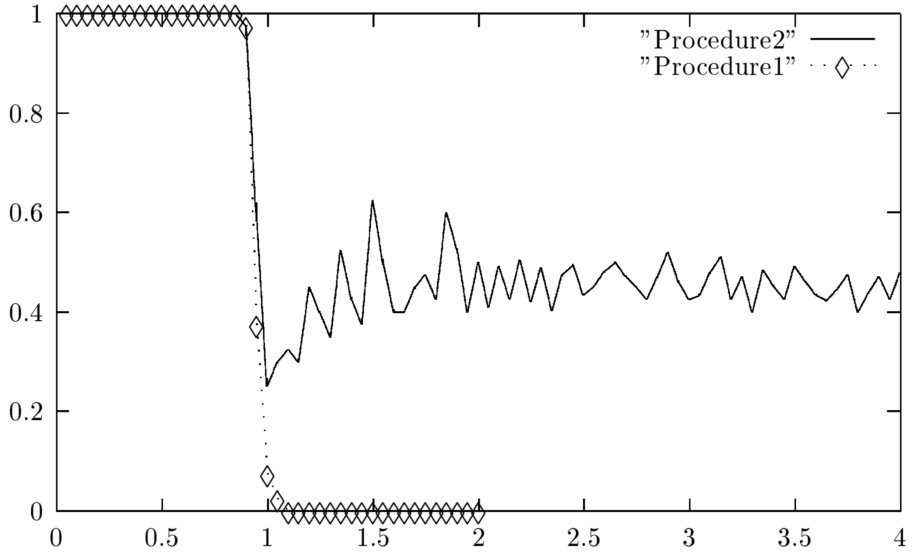


Figure 3. Evaluation of both classes $\mathcal{I}$ and $\mathcal{J}$.

Figure 3 shows the results. The curves represent the rate of solved instances with respect to the ratio ($\frac{c}{v}$) for both methods encoded by procedure 1 and procedure 2. Each curve expresses the success probability of the corresponding method in solving an instance. We can see that the procedure 1 solves only the instances whose ratios are less than 1. The class $\mathcal{I}$ is limited to the instances satisfying $\frac{c}{v} \leq 1$. However, procedure 2 solves about 50 percent of instances whose ratios is greater than 1 in addition to that instances solved by the procedure 1. Procedure 2 is an interesting extension of the procedure 1, giving the extended class $\mathcal{J}$ which includes $\mathcal{I}$. It will be important to use the procedure 2 to define new SAT local search methods. We are looking to do that in future.

6. Conclusion

We proved in this note that SAT instances can be seen as bipartite graphs, the satisfiability problem is equivalent to matching search in a SAT-graph. Matchings properties are used to prove results on the satisfiability problem tractability. We proved that some SAT instances are trivially satisfiable and the class r-r-SAT of

Tovey is a sub-class of these instances. Two polynomial model search algorithms are given. These methods find a model if one exists, otherwise they prove efficiently that the instance is not an element of the considered sub-class. Further investigations will consist in providing a new local SAT search algorithm by adapting the matching satisfiability algorithm (procedure 2) of figure 2. It is important to implement a such method and compare it to other existing methods of local search like GSAT [4] or Tabu search [5].

Acknowledgement. To the memory of Claire Rauzy, her advises and help in the graph theory had been very useful for this work. Many thanks to the reviewers for their good remarks.

References

[1] Aspvall B. (1980), *Recognizing Disguised NR(1) Instances of the Satisfiability problem* Journal of Algorithms. **1**, pages 97–103.

[2] Berge C. (1957), *Two theorems in Graph Theory*, In Proc. Nat. Ac. Sciences, USA 43, pages 842–844

[3] Mitchell D., Selman B. and Levesque H. (1992), *Hard and Easy Distributions of SAT Problems*, In Proc. AAAI-92, USA San Jose, pages 459–465.

[4] Selman B., Levesque H. and Mitchell D., (1992), *A new method for solving hard satisfiability problems,* In Proc. AAAI-92, USA San Jose, pages 440–446.

[5] Mazure B., and Sais L. (1997), *Tabu search for SAT,* In Proc. AAAI-97/IAAI-97, AAAI Press, pages 281–285.

[6] Cook S.A. (1971), *The complexity of Theorem Proving Procedures*, Third Annual ACM Sump. On Th. of computing, pages 151–158.

[7] Dowling W.F. and Gallier J.H. (1984), *Linear-time Algorithms for testing the Satisfiability of Propositional Horn Formulae*, Journal of Logic Programming, **3**, pages 267–284.

[8] Boros E., Hammer P.L. and Sun X. (1994), *Recognition of q-Horn formulae in linear time*, Journal of Discreet Applied Mathematics, **55** , pages 21–32.

[9] Boros E., Crama Y., Hammer P.L. and Saks M. (1994), *A complexity index for satisfiability problems*, SIAM Journ. Comp., **23**, pages 45–49.

[10] Boros E., Crama Y. and Hammer P. L. (1990), *Polynomial-time inference of all implications for Horn and related formulae*, Annals of Mathematics and Artificial Intelligence, **1**, pages 21–32.

[11] Edmonds J. (1965), *Paths, trees and flowers*, Canad. J. Math, **17**, pages 449–467.

[12] Gallo G. and Scutella M.G. (1988), *Polynomial Solvable Satisfiable Problems*, Information Processing Letters, **29** ,pages 221–227.

[13] Ghallab M. and Escalada-Imaz E. (1991), *A linear control algorithm for a class of rule-based systems*, Journal of Logic Programming, **11**, pages 117–132.

[14] Hall P. (1935), *On representative of subsets*, J. of London Math. Soc, **10**, pages 26–30.

[15] Hebrard J.J. (1994), *A linear algorithm for renaming a set of Horn set,* Theoretical Computer Sciences, **124** , pages 343–350.

[16] Horowitz E. and Sahni S. (1978), *Fundamentals of Computer Algorithms*, Computer Science Press, Rockville, MD.

[17] Kuhn H.W. (1955), *The Hungarian method for the assignment problem*, Naval Res. Logist. Quart, **2**, pages 83–87.

[18] Lindhorst G. and Shahrokhi F. (1989), *On renaming a set of clauses as a Horn set* , Information Processing Letters, **30**, pages 289–293.

[19] Mannila H. and Mehlorn K. (1985), *A fast algorithm for renaming a set of clauses as a Horn set*, Information processing Letter, **29**, pages 269–272.

[20] Minoux M. (1988), *LTUR: A simplified Lineal-Time Unit Resolution Algorithm for Horn Formulae and its Computer Implementation*, J. Information Processing Letters, **29**, pages 1–12.

[21] Rauzy A. and Genisson R. (1996), *Sur les relations algorithmiques entre classes polynomiales du problème SAT et des problèmes de satisfaction de contraintes*, RIFIA, Rennes France.

[22] Tovey Craig A. (1984)", *A Simplified NP-complete Satisfiability Problem*, Discrete Applied Mathematics, **8**, pages 85–89.

[23] Chandru V., Coulard C.R., Hammer P.L., Montanez M. and Sun X., *On renamable Horn functions* , In Annals of Mathematics and Artificial Intelligence, **1**. J.C Baltzer AG, Scientific Publishing Company, Basel Switzerland 30.

B. Benhamou
Laboratoire des Sciences de l'Information et des Systèmes
LSIS - UMR CNRS 6168
Université de Provence
Centre de Mathématiques et d'Informatique
39 rue F.J. Curie 13453 cedex 13, Marseille, France
`Belaid.Benhamou@cmi.univ-mrs.fr`