Chapitre 1

Complexité des Problèmes de Satisfaction de Contraintes Booléennes

1.1. Introduction

Un problème de satisfaction de contraintes booléennes, encore appelé problème de satisfaisabilité, consiste, étant donné un ensemble de contraintes définies sur des variables booléennes, à décider s'il existe une assignation de valeurs aux variables satisfaisant toutes les contraintes (et éventuellement à déterminer une telle assignation). Souvent une telle assignation n'existe pas et, dans ce cas, il est naturel de chercher une assignation satisfaisant un nombre maximal de contraintes, ou minimisant le nombre de contraintes non satisfaites.

Un exemple d'un problème de satisfaction de contraintes booléennes est le problème connu sous le nom de SAT, consistant à décider si une formule propositionnelle (exprimée comme une conjonction de disjonctions) est satisfaisable ou non. SAT a été le premier problème montré *NP*-complet par Cook [COO 71] et Levin [LEV 73] et est resté depuis un problème central dans l'étude de la *NP*-difficulté des problèmes d'optimisation [GAR 79]. La *NP*-complétude de SAT assure qu'aucun algorithme pour ce problème ne peut être efficace au pire cas, sous l'hypothèse $P \neq NP$. Néanmoins, il existe, en pratique, de nombreux algorithmes efficaces pour résoudre le problème SAT.

Chapitre rédigé par Cristina BAZGAN.

Les problèmes de satisfaisabilité ont des applications directes en recherche opérationnelle, intelligence artificielle et architecture des systèmes. Par exemple, en recherche opérationnelle, le problème de coloration de graphes, peut être modélisé comme une instance de SAT. Pour décider si un graphe avec n sommets peut être coloré avec k couleurs, nous considérons $k \times n$ variables booléennes, x_{ij}, $i = 1, \ldots, n$, $j = 1, \ldots, k$, où x_{ij} prend la valeur vrai si et seulement si le sommet i se voit attribuer la couleur j. Hoos [HOO 98] a étudié l'efficacité de diverses modélisations du problème de coloration de graphes comme un problème de satisfaisabilité quand on applique un algorithme de recherche locale spécifique sur l'instance du problème de satisfaisabilité obtenue. Le problème de l'arbre de Steiner, largement étudié en recherche opérationnelle, intervient dans des applications de conception de réseaux et routage. Dans [JIA 95], les auteurs ont réduit ce problème à un problème consistant à trouver une assignation maximisant le nombre de contraintes satisfaites. Certains problèmes d'ordonnancement ont été résolus en passant par une modélisation en termes de problème de satisfaisabilité [CRA 94]. Tester diverses propriétés de graphes ou hypergraphes est également un problème qui se ramène à un problème de satisfaisabilité. En intelligence artificielle, une application intéressante est le problème de planification qui peut être représenté comme un ensemble de contraintes tel que toute assignation satisfaisante correspond à un plan valide (voir [KAU 92] pour une telle modélisation). D'autres applications en intelligence artificielle sont : l'apprentissage à partir d'exemples, la détermination de la cohérence d'un système de règles d'une base de connaissances, la construction d'inférences dans une base de connaissances. Dans la conception de circuits électriques, on souhaite généralement construire un circuit avec certaines fonctionnalités (décrites par une fonction booléenne) satisfaisant diverses contraintes motivées par des considérations technologiques, de fiabilité ou de disponibilité comme par exemple : minimiser le nombre de portes utilisées, minimiser la profondeur du circuit, ou n'utiliser que certains types de portes.

Les problèmes de satisfaisabilité ont également d'autres applications en raisonnement automatique, vision par ordinateur, bases de données, robotique, conception assistée par ordinateur. Gu, Purdom, Franco et Wah ont écrit un article de synthèse [GU 97] citant de nombreuses applications de problèmes de satisfaisabilité (environ 250 références).

Face à un problème de satisfaisabilité, on peut soit l'étudier du point de vue théorique (établir sa complexité exacte ou d'approximation, bâtir des algorithmes qui garantissent une solution exacte ou approchée), soit le résoudre du point de vue pratique. Parmi les méthodes les plus efficaces pour la résolution pratique des problèmes de satisfaisabilité citons : la recherche locale, la recherche tabou, le recuit simulé. Pour plus de détails, on pourra se reporter à [GU 97] et [GEN 99] qui proposent une synthèse de la plupart des algorithmes pratiques de résolution pour les problèmes de satisfaisabilité.

Dans ce chapitre, nous présentons les principaux résultats de complexité exacte et d'approximation pour les problèmes de satisfaisabilité selon le type de fonctions booléennes intervenant dans les contraintes. Notre but n'est pas de présenter de manière exhaustive tous les résultats existant dans la littérature mais d'identifier les problèmes les plus étudiés et d'introduire les concepts et algorithmes de base. La plupart des problèmes de satisfaisabilité sont difficiles. Il est donc intéressant, à la fois du point de vue théorique et pratique, d'identifier des cas particuliers qui sont plus faciles. Nous avons choisi de présenter les cas particuliers les plus étudiés : les instances planaires, les instances avec un nombre borné d'occurrences de chaque variable et les instances denses. Plusieurs problèmes d'optimisation se modélisent comme un problème de satisfaisabilité avec une contrainte globale additionnelle sur l'ensemble des solutions réalisables. En particulier, le problème MIN BISECTION, dont la complexité d'approximation n'est toujours pas établie, se modélise comme un problème de satisfaisabilité où l'ensemble des solutions réalisables est l'ensemble des assignations équilibrées (avec autant de variables fixées à 0 que à 1). Nous présentons également quelques résultats obtenus sur les problèmes de satisfaisabilité sous cette contrainte globale.

Pour le lecteur souhaitant approfondir la complexité des problèmes de satisfaisabilité, citons dans la littérature, la monographie de Creignou, Khanna et Sudan [CRE 01] où on peut trouver les preuves de la plupart des résultats importants dans ce domaine et qui couvre en dehors des résultats présentés ici, d'autres aspects comme la complexité de comptage, la complexité de représentation des fonctions ainsi que d'autres problèmes de satisfaisabilité. Notons également l'existence d'un compendium électronique, de Crescenzi et Kann [CRE 95b], qui regroupe les résultats connus de complexité d'approximation pour les problèmes d'optimisation, en particulier pour les problèmes de satisfaisabilité.

Le chapitre est structuré comme suit. Dans la Section 1.2 nous introduisons les types de fonctions booléennes que nous allons utiliser et définissons les problèmes de décision et d'optimisation considérés. Dans la Section 1.3 nous étudions les problèmes de décision et dans la Section 1.4 les problèmes de maximisation et minimisation. Nous considérons ensuite quelques instances particulières des problèmes de satisfaisabilité : instances planaires (Section 1.5.1), denses (Section 1.5.2), avec un nombre borné d'occurrences de chaque variable (Section 1.5.3). Nous présentons également la complexité des problèmes de satisfaisabilité quand l'ensemble des solutions réalisables est restreint aux assignations équilibrées (Section 1.6). Nous clôturons notre chapitre par une brève conclusion (Section 1.7).

1.2. Préliminaires

Une instance d'un problème de satisfaction de contraintes booléennes est un ensemble de m contraintes $C_1, \ldots, C_m$ définies sur un ensemble de n variables $x_1, \ldots, x_n$.

Une contrainte C_j est l'application d'une fonction booléenne $f : \{0,1\}^k \rightarrow \{0,1\}$ à un sous-ensemble de variables $x_{i_1}, \ldots, x_{i_k}$ où $i_1, \ldots, i_k \in \{1, \ldots, n\}$. Cette contrainte est également notée $f(x_{i_1}, \ldots, x_{i_k})$. Une assignation $x_i = v_i$, pour $i = 1, \ldots, n$, où $v_i \in \{0,1\}$ satisfait la contrainte $f(x_{i_1}, \ldots, x_{i_k})$ si et seulement si $f(v_{i_1}, \ldots, v_{i_k}) = 1$.

Un littéral est une variable x_i (littéral positif) ou bien sa négation $\bar{x}_i$ (littéral négatif).

EXEMPLE 1.1.– Quelques exemples de fonctions booléennes utilisées pour définir des contraintes :

– $T(x) = x, F(x) = \bar{x}$

– *OR*$_i^k(x_1, \ldots, x_k) = \bar{x}_1 \vee \ldots \vee \bar{x}_i \vee x_{i+1} \vee \ldots \vee x_k$, où $i \leq k$ représente le nombre de littéraux négatifs dans la disjonction

– *AND*$_i^k(x_1, \ldots, x_k) = \bar{x}_1 \wedge \ldots \wedge \bar{x}_i \wedge x_{i+1} \wedge \ldots \wedge x_k$, où $i \leq k$ représente le nombre de littéraux négatifs dans la conjonction

– *XOR*$^k(x_1, \ldots, x_k) = x_1 \oplus \ldots \oplus x_k$ où $\oplus$ représente l'opérateur "ou exclusif" $(0 \oplus 0 = 0, 1 \oplus 0 = 1, 0 \oplus 1 = 1, 1 \oplus 1 = 0)$

– *XNOR*$^k(x_1, \ldots, x_k) = \overline{x_1 \oplus \ldots \oplus x_k}$

Une contrainte peut également se représenter comme une formule booléenne qui peut avoir diverses formes. Une formule est sous forme normale conjonctive (CNF) si elle est de la forme $c_1 \wedge \ldots \wedge c_m$ où chaque c_t est une clause disjonctive, c'est-à-dire de la forme $\ell_{t_1} \vee \ldots \vee \ell_{t_p}$ où $\ell_{t_i}, i = 1, \ldots, p$ sont des littéraux. Une formule est sous forme normale disjonctive (DNF) si elle est de la forme $c_1 \vee \ldots \vee c_m$ où chaque c_t est une clause conjonctive, c'est-à-dire de la forme $\ell_{t_1} \wedge \ldots \wedge \ell_{t_p}$ où $\ell_{t_i}, i = 1, \ldots, p$ sont des littéraux. Une formule kCNF (ou kDNF) est une formule CNF (ou DNF) dont chaque clause contient au plus k littéraux.

Observons que si chaque contrainte d'un problème de satisfaisabilité est représentée par une formule CNF, l'ensemble des contraintes du problème est lui-même représentable par une formule CNF correspondant à la conjonction des formules précédentes.

On distingue différents problèmes de satisfaisabilité selon le type de fonctions booléennes utilisées pour définir les contraintes. Soit $\mathcal{F}$ un ensemble fini de fonctions booléennes. Un *$\mathcal{F}$-ensemble de contraintes* est un ensemble de contraintes qui n'utilisent que des fonctions booléennes appartenant à $\mathcal{F}$. Une assignation satisfait un $\mathcal{F}$-ensemble de contraintes si et seulement si elle satisfait chaque contrainte de l'ensemble de contraintes.

1.2.1. *Problèmes de satisfaction de contraintes : versions décision et optimisation*

Nous définissons dans cette partie les classes de problèmes que nous allons étudier. Il s'agit des versions décision et optimisation des problèmes de satisfaisabilité.

La version de décision d'un problème consiste à déterminer si ce problème admet au moins une solution ; sa version recherche consiste à trouver une solution s'il en existe. La version optimisation d'un problème consiste à chercher une solution qui maximise ou minimise une fonction adéquate.

DÉFINITION 1.1.– *Le problème de satisfaisabilité* SAT($\mathcal{F}$) *consiste à décider s'il existe une assignation qui satisfait un* $\mathcal{F}$*-ensemble de contraintes. Le problème de recherche associé au problème de décision* SAT($\mathcal{F}$) *consiste à trouver une assignation qui satisfait un* $\mathcal{F}$*-ensemble de contraintes si une telle assignation existe ou bien répondre "non" sinon.*

Dans ce chapitre nous allons voir qu'à chaque fois qu'on peut résoudre le problème de décision SAT($\mathcal{F}$) en temps polynomial, on peut également trouver une solution pour les instances satisfaisables et donc résoudre le problème de recherche associé en temps polynomial.

Il est usuel de distinguer certaines variantes de SAT($\mathcal{F}$) où chaque fonction de $\mathcal{F}$ dépend d'au plus (ou d'exactement) k variables. Ces variantes sont notées kSAT($\mathcal{F}$) (ou EkSAT($\mathcal{F}$)).

Nous présentons ensuite quelque problèmes classiques de décision ainsi que le problème de satisfaisabilité SAT($\mathcal{F}$) correspondant :

– SAT est le problème consistant à décider si un ensemble de clauses disjonctives définies sur n variables booléennes est satisfaisable. Il correspond au problème SAT($\mathcal{F}$) où $\mathcal{F}$ est l'ensemble des fonctions OR_i^k, pour $k \leq n$.

– CONJ est le problème consistant à décider si un ensemble de clauses conjonctives définies sur n variables booléennes est satisfaisable. Il correspond au problème SAT($\mathcal{F}$) où $\mathcal{F}$ est l'ensemble des fonctions AND_i^k, pour $k \leq n$.

– LIN2 est le problème consistant à décider si un ensemble d'équations linéaires définies sur n variables booléennes est satisfaisable. Il correspond au problème SAT($\mathcal{F}$) où $\mathcal{F}$ est l'ensemble des fonctions XOR^k, $XNOR^k$, pour $k \leq n$.

– 2SAT est la version de SAT où chaque clause disjonctive comporte au plus 2 littéraux, et il correspond à 2SAT($\mathcal{F}$) où $\mathcal{F}$ est l'ensemble des fonctions OR_i^k, pour $k \leq 2$.

– E3SAT est la version de SAT où chaque clause disjonctive a exactement 3 littéraux, et il correspond à SAT($\{OR_0^3, OR_1^3, OR_2^3, OR_3^3\}$).

DÉFINITION 1.2.– *Le problème de maximisation* MAX SAT($\mathcal{F}$) *consiste à déterminer une assignation qui satisfait un nombre maximum de contraintes d'un* $\mathcal{F}$*-ensemble de contraintes.*

Par exemple, le problème MAX CUT, qui consiste à partitionner l'ensemble des sommets d'un graphe en deux parties telles que le nombre d'arêtes dont les extrémités appartiennent à des parties différentes soit maximum, peut se formuler comme un problème de type MAX SAT($\{XOR^2\}$) comme suit. Considérant un graphe G, instance de MAX CUT, on associe à chaque sommet i une variable x_i et à chaque arête (i, j) de G la contrainte $XOR^2(x_i, x_j)$.

DÉFINITION 1.3.– *Le problème de minimisation* MIN SAT DELETION($\mathcal{F}$) *consiste à déterminer une assignation qui minimise le nombre de contraintes non satisfaites d'un* $\mathcal{F}$*-ensemble de contraintes, ce qui correspond au nombre minimum de contraintes à enlever tel que les contraintes restantes sont satisfaites.*

MIN SAT DELETION($\mathcal{F}$) permet de modéliser naturellement certains problèmes de minimisation. Par exemple, le problème s-t MIN CUT dans un graphe non-orienté, qui consiste à partitionner l'ensemble des sommets d'un graphe en deux parties telles que s et t appartiennent à des parties différentes et telles que le nombre d'arêtes dont les extrémités appartiennent à des parties différentes soit minimum, peut se formuler comme un problème de type MIN SAT DELETION($\{XNOR^2\} \cup \{T, F\}$) comme suit. Considérant un graphe G, instance de s-t MIN CUT, on associe à chaque sommet i une variable x_i et à chaque arête (i, j) de G la contrainte $XNOR^2(x_i, x_j)$. De plus on ajoute les contraintes $T(x_s)$ et $F(x_t)$.

REMARQUE 1.–

1) Les problèmes MAX SAT($\mathcal{F}$) et MIN SAT DELETION($\mathcal{F}$) sont clairement reliés. En effet, considérant une instance I de MAX SAT($\mathcal{F}$) avec m contraintes, une solution optimale pour l'instance I de MAX SAT($\mathcal{F}$) de valeur $opt_{MaxSAT(\mathcal{F})}(I)$ est également une solution optimale de l'instance I du problème MIN SAT DELETION($\mathcal{F}$) de valeur $opt_{MinSATDeletion(\mathcal{F})}(I) = m\text{-}opt_{MaxSAT(\mathcal{F})}(I)$. Ainsi, les complexités exactes de MAX SAT($\mathcal{F}$) et MIN SAT DELETION($\mathcal{F}$) coïncident. En revanche, les complexités d'approximation des deux problèmes peuvent être très différentes comme nous allons le voir par la suite.

2) Dans la littérature, on définit également le problème MIN SAT($\mathcal{F}$) qui consiste à determiner une assignation minimisant le nombre de contraintes satisfaites. Par exemple, dans le compendium de Crescenzi et Kann [CRE 95b], le problème MIN SAT consiste à déterminer une assignation minimisant le nombre de clauses satisfaites d'un ensemble de clauses disjonctives. Observons que MIN SAT($\mathcal{F}$) est équivalent du point de vue exact et approximation) à MIN SAT DELETION($\mathcal{F}'$), où $\mathcal{F}'$ est l'ensemble des fonctions complémentaires aux fonctions de $\mathcal{F}$. Par exemple, trouver une assignation qui minimise le nombre de contraintes satisfaites parmi les

contraintes $x_1 \vee x_2, x_1 \vee \bar{x}_3, x_2 \vee x_3, \bar{x}_1 \vee \bar{x}_2$ est équivalent à trouver une assignation qui minimise le nombre de contraintes non satisfaites parmi les contraintes $\bar{x}_1 \wedge \bar{x}_2, \bar{x}_1 \wedge x_3, \bar{x}_2 \wedge \bar{x}_3, x_1 \wedge x_2$. Ainsi le problème MIN SAT est équivalent à MIN SAT DELETION($\mathcal{F}$) où les contraintes sont des clauses conjonctives (problème appelé dans la suite MIN CONJ DELETION). Dans la suite, nous considérons uniquement le problème MIN SAT DELETION($\mathcal{F}$).

1.2.2. *Types de contraintes*

La complexité de SAT($\mathcal{F}$) ainsi que les complexités exactes et d'approximation de MAX SAT($\mathcal{F}$) et MIN SAT DELETION($\mathcal{F}$) dépendent des types des fonctions booléennes de l'ensemble $\mathcal{F}$. Dans cette partie, nous décrivons les types de fonctions booléennes qui ont été les plus étudiées et dont nous allons discuter dans la suite du chapitre.

Une fonction booléenne f est :

– *0-valide* si $f(0, \ldots, 0) = 1$

– *1-valide* si $f(1, \ldots, 1) = 1$

– *Horn* si elle peut s'exprimer comme une formule CNF ayant au plus un littéral positif dans chaque clause

– *anti-Horn* si elle peut s'exprimer comme une formule CNF ayant au plus un littéral négatif dans chaque clause

– *affine* si elle peut s'exprimer comme une conjonction d'équations linéaires sur le corp de Galois $GF(2)$, c'est-à-dire comme une conjonction d'équations de type $x_{i_1} \oplus \ldots \oplus x_{i_p} = 0$ ou $x_{j_1} \oplus \ldots \oplus x_{j_q} = 1$.

– *bijonctive* si elle peut s'exprimer comme une formule 2CNF

– *2-monotone* si elle peut s'exprimer comme une formule DNF de la forme $x_{i_1} \wedge \ldots \wedge x_{i_p}$ ou $\bar{x}_{\ell_1} \wedge \ldots \wedge \bar{x}_{\ell_q}$ ou $(x_{i_1} \wedge \ldots \wedge x_{i_p}) \vee (\bar{x}_{\ell_1} \wedge \ldots \wedge \bar{x}_{\ell_q})$. Remarquons qu'une fonction 2-monotone est à la fois Horn et anti-Horn.

– *fermée au complément* si pour toute assignation v on a $f(v) = f(\bar{v})$, où $\bar{v}$ est l'assignation complémentaire à v

On étend les notions précédentes à un ensemble $\mathcal{F}$ de fonctions dès lors que toutes les fonctions de $\mathcal{F}$ ont la propriété requise. Par exemple, si chaque fonction de $\mathcal{F}$ est Horn alors l'ensemble $\mathcal{F}$ est Horn et le problème SAT($\mathcal{F}$) est appelé HORN SAT. Les formules Horn interviennent en intelligence artificielle pour développer des systèmes experts ou formaliser des bases de connaissances. Elle représentent également le fondement logique de Prolog.

La notation utilisée dans la littérature pour le problème SAT($\mathcal{F}$) quand $\mathcal{F}$ est affine est LIN2. k-LIN2 est le problème kSAT($\mathcal{F}$) où $\mathcal{F}$ est affine et Ek-LIN2 est la variante de k-LIN2 où chaque équation dépend exactement de k variables. Une instance de LIN2 est *0-homogène* (respectivement *1-homogène*) si toutes ses équations linéaires ont leur termes libres égaux à 0 (respectivement 1).

MONOTONE-SAT et MONOTONE-kSAT sont les variantes de SAT et kSAT où toute clause ne contient que des littéraux positifs ou toute clause ne contient que des littéraux négatifs.

Nous allons considérer dans ce chapitre d'autres variantes de SAT($\mathcal{F}$) :

– le problème NAE3SAT est SAT($\{f\}$), où f est d'arité 3 et $f(x_1, x_2, x_3) = 1$ si et seulement si les 3 variables n'ont pas la même valeur, plus précisément, $f(0,0,0) = f(1,1,1) = 0$ sinon f prend la valeur 1.

– le problème 1IN3SAT est SAT($\{g\}$), où g est d'arité 3 et $g(x_1, x_2, x_3) = 1$ si et seulement si exactement une des 3 variables a la valeur 1, plus précisément, $g(1,0,0) = g(0,1,0) = g(0,0,1) = 1$, sinon g prend la valeur 0.

REMARQUE 2.– Pour certaines variantes du problème SAT($\mathcal{F}$), l'ensemble des contraintes peut être représenté de façon équivalent par une formule mettant en conjonction ces contraintes. Dans les problèmes d'optimisation correspondants MAX SAT($\mathcal{F}$) et MIN SAT DELETION($\mathcal{F}$), nous n'utilisons que la formulation sous forme d'un ensemble de contraintes afin de pouvoir compter le nombre de contraintes satisfaites.

Nous présentons ensuite quelques variantes de problèmes d'optimisation utilisées dans la suite du chapitre :

– MAX SAT, consiste, étant donné un ensemble de clauses disjonctives définies sur n variables booléennes, à trouver une assignation maximisant le nombre de clauses satisfaites. MAX SAT correspond donc au problème MAX SAT($\mathcal{F}$) où $\mathcal{F}$ est l'ensemble des fonctions OR_i^k, pour $k \leq n$.

– MIN SAT DELETION, consiste, étant donné un ensemble de clauses disjonctives définies sur n variables booléennes, à trouver une assignation minimisant le nombre de clauses non satisfaites. MIN SAT DELETION correspond donc au problème MIN SAT DELETION($\mathcal{F}$) où $\mathcal{F}$ est l'ensemble des fonctions OR_i^k, pour $k \leq n$.

– MAX CONJ, consiste, étant donné un ensemble de clauses conjonctives définies sur n variables booléennes, à trouver une assignation maximisant le nombre de clauses satisfaites. MAX CONJ correspond donc au problème MAX SAT($\mathcal{F}$) où $\mathcal{F}$ est l'ensemble des fonctions AND_i^k, pour $k \leq n$.

– MIN CONJ DELETION, consiste, étant donné un ensemble de clauses conjonctives définies sur n variables booléennes, à trouver une assignation minimisant le nombre de clauses non satisfaites. MIN CONJ DELETION correspond donc au problème MIN SAT DELETION ($\mathcal{F}$) où $\mathcal{F}$ est l'ensemble des fonctions AND_i^k, pour $k \leq n$.

– MAX LIN2, consiste, étant donné un ensemble d'équations linéaires définies sur n variables booléennes, à trouver une assignation maximisant le nombre d'équations satisfaites. MAX LIN2 correspond donc au problème MAX SAT($\mathcal{F}$) où $\mathcal{F}$ est l'ensemble des fonctions *XOR*^{k}, *XNOR*^{k}, pour $k \leq n$.

– MIN LIN2 DELETION, consiste, étant donné un ensemble d'équations linéaires définies sur n variables booléennes, à trouver une assignation minimisant le nombre d'équations non satisfaites. MIN LIN2 DELETION correspond donc au problème MIN SAT DELETION($\mathcal{F}$) où $\mathcal{F}$ est l'ensemble des fonctions *XOR*^{k}, *XNOR*^{k}, pour $k \leq n$.

– Les problèmes MAX kSAT, MAX EkSAT, MAX kCONJ, MAXEkCONJ, MAX k-LIN2, MAX Ek-LIN2 ainsi que les versions MIN DELETION correspondantes sont définis de manière similaire sur des clauses ou équations de taille (au plus) k.

1.3. Complexité des problèmes de décision

Nous étudions dans cette section la complexité des problèmes de décision SAT($\mathcal{F}$) selon le type de fonctions de l'ensemble $\mathcal{F}$.

SAT est le premier problème montré *NP*-complet par Cook [COO 71] et Levin [LEV 73]. On peut réduire facilement SAT à kSAT, $k \geq 3$ ce qui implique la *NP*-complétude de kSAT, pour $k \geq 3$. Par contre, 2SAT est polynomial [COO 71].

THÉORÈME 1.1.– 2SAT *est résoluble en temps polynomial.*

Preuve : Soit I une instance de 2SAT avec m clauses $C_1, \ldots, C_m$ et n variables $x_1, \ldots, x_n$. Nous allons construire un graphe orienté G_I avec $2n$ sommets $v_1, \bar{v}_1, \ldots, v_n, \bar{v}_n$, où v_i (resp. $\bar{v}_i$) correspond à x_i (resp. $\bar{x}_i$). Pour un littéral ℓ_i (resp. $\bar{\ell}_i$), notons par w_i (resp. $\bar{w}_i$) le sommet correspondant. Ainsi, si $\ell_i = x_i$ alors $w_i = v_i$ et $\bar{w}_i = \bar{v}_i$ et si $\ell_i = \bar{x}_i$ alors $w_i = \bar{v}_i$ et $\bar{w}_i = v_i$. Chaque clause constituée d'un seul littéral ℓ est remplacée par la clause équivalente $\ell \vee \ell$. Pour chaque clause $\ell_1 \vee \ell_2$, équivalente aux implications logiques $\bar{\ell}_1 \Rightarrow \ell_2$ et $\bar{\ell}_2 \Rightarrow \ell_1$, introduisons dans G_I les arcs $(\bar{w}_1, w_2)$ et $(\bar{w}_2, w_1)$. Observons alors que si dans G_I il existe un chemin de w_i à w_j alors il existe aussi un chemin de $\bar{w}_j$ à $\bar{w}_i$.

Considérons une assignation de valeurs de vérité pour les sommets de G_I. Cette assignation correspond à une assignation de $x_1, \ldots, x_n$ satisfaisant I si et seulement si :

(a) pour chaque i, v_i et $\bar{v}_i$ ont des valeurs complémentaires.

(b) aucun arc (w_p, w_q) n'est tel que w_p a la valeur 1 et w_q a la valeur 0 (sinon l'implication logique $\ell_p \Rightarrow \ell_q$ serait fausse).

Nous allons justifier ensuite que I est satisfaisable si et seulement si dans G_I, aucun sommet v_i n'est dans la même composante fortement connexe que $\bar{v}_i$.

Supposons que I est satisfaisable et qu'il existe un sommet v_i appartenant à la même composante fortement connexe que $\bar{v}_i$. Soit une assignation pour $x_1, \ldots, x_n$ satisfaisant I. Cette assignation induit une assignation des valeurs de vérité pour les sommets de G_I satisfaisant (a). Comme v_i appartient à la même composante fortement connexe que $\bar{v}_i$, il existe dans G_I un chemin de v_i à $\bar{v}_i$ et de $\bar{v}_i$ à v_i. L'un de ces deux chemins a nécessairement pour extrémité initiale un sommet évalué à 1 et pour extrémité terminale un évalué à 0. Il contient donc un arc (w_p, w_q) tel que w_p a pour valeur 1 et w_q a pour valeur 0 ce qui contredit (b) et donc le fait que I soit satisfaisable.

Supposons maintenant qu'aucun sommet v_i n'est dans la même composante fortement connexe que $\bar{v}_i$. Nous allons construire une assignation sur les sommets telles que (a) et (b) sont satisfaites. Déterminons d'abord les composantes fortement connexes de G_I en utilisant l'algorithme linéaire de Tarjan [TAR 72]. Construisons ensuite le graphe réduit de G_I, noté G_I^r, dont les sommets sont les composantes fortement connexes et où l'on crée un arc d'une composante S_1 à une composante S_2 s'il existe un arc d'un sommet de S_1 vers un sommet de S_2. Notons $\bar{S}_i$ la composante fortement connexe contenant les littéraux complémentaires aux littéraux de S_i. Évidemment si S_1 est un prédécesseur de S_2 alors $\bar{S}_2$ est un prédécesseur de $\bar{S}_1$. L'algorithme de Tarjan génère les composantes fortement connexes en ordre topologique inverse, plus précisément si S_1 est générée avant S_2, alors S_1 ne peut être un prédécesseur de S_2.

Nous allons définir ensuite des valeurs de vérité pour les sommets de G_I^r ; un sommet de G_I aura alors la valeur de vérité de la composante à laquelle il appartient. Nous répétons l'algorithme suivant tant que possible : considérons la première composante S dans l'ordre topologique inverse, qui n'a pas de valeur de vérité et attribuons la valeur 1 à S et la valeur 0 à la composante $\bar{S}$. Évidement (a) est satisfait. Pour justifier que (b) est satisfait il faut montrer qu'il n'existe pas d'arc d'un sommet correspondant à un littéral de valeur 1 vers un sommet correspondant à un littéral de valeur 0. S'il existait un arc d'un sommet w_1 de valeur 1 appartenant à la composante S_1 vers un sommet w_2 de valeur 0 appartenant à la composante S_2 alors dans G_I^r il existe un arc de S_1 (valuée 1) à S_2 (valuée 0) et de $\bar{S}_2$ (valuée 1) à $\bar{S}_1$ (valuée 0). Cela contredit la manière dont nous avons attribué les valeurs 1 aux composantes car dans un ordre topologique inverse S_2 est avant S_1 et $\bar{S}_1$ est avant $\bar{S}_2$ et donc une au moins des composantes S_2 ou $\bar{S}_1$ devrait avoir la valeur 1. □

Tester la satisfaisabilité d'une formule Horn a été montré polynomial par Jones et Laaser [JON 77] et la complexité de l'algorithme polynomial a été améliorée par Dowling et Gallier [DOW 84] et Minoux [MIN 88].

THÉORÈME 1.2.– HORN SAT *est resoluble en temps polynomial.*

Preuve : Considérons une instance I de HORN SAT. Si I ne contient pas de clause unitaire, chaque clause contient au moins un littéral négatif et il suffit de fixer toutes les variables à 0 pour obtenir une assignation satisfaisante. Si I contient au moins une clause unitaire, on utilise le principe de résolution unitaire qui consiste à appliquer itérativement les deux règles suivantes :

1) Si une clause est constituée d'un littéral positif x_i (ou d'un littéral négatif $\bar{x}_i$), alors fixer x_i à 1 (ou à 0) et supprimer la clause.

2) Tant qu'il existe une clause contenant au moins une variable fixée, alors la formule peut être réduite ainsi :

(a) supprimer toute clause contenant un littéral positif x_i où x_i a été fixée à 1 (ou un littéral négatif $\bar{x}_i$ où x_i a été fixée à 0) car cette clause va être automatiquement satisfaite indépendamment des valeurs des autres littéraux de la clause.

(b) dans toute clause effacer tout littéral positif x_i tel que x_i a été fixée à 0 (ou tout littéral négatif $\bar{x}_i$ tel que x_i a été fixée à 1) car un tel littéral ne va jamais satisfaire cette clause.

Si en appliquant (b) on efface tous les littéraux d'une clause alors la formule n'est pas satisfaisable.

Après avoir appliqué les règles 1 et 2, trois cas sont possibles :

– I est déclarée non satisfaisable en 2(b).

– I est satisfaisable car toutes ses clauses ont été supprimées par l'application de 1 et 2(a).

– l'assignation partielle obtenue définit une sous-instance I' qui ne contient pas de clause unitaire. I est donc satisfaisable en fixant à 0 les variables non fixées par l'assignation partielle.

□

Évidemment un algorithme semblable au précédent peut être établi pour décider si SAT($\mathcal{F}$) est satisfaisable quand $\mathcal{F}$ est anti-Horn et dans le cas positif trouver une assignation satisfaisante. Chacun de ces deux algorithmes fonctionne également quand $\mathcal{F}$ est 2-monotone.

Quad $\mathcal{F}$ est affine, SAT($\mathcal{F}$) est également résoluble en temps polynomial en utilisant l'élimination de Gauss :

THÉORÈME 1.3.– LIN2 *est résoluble en temps polynomial.*

Donc SAT($\mathcal{F}$) est résoluble en temps polynomial quand chaque fonction de $\mathcal{F}$ est une clause disjonctive de taille au plus 2 (ou plus généralement quand chaque fonction de $\mathcal{F}$ est 2CNF), quand $\mathcal{F}$ est Horn ou anti-Horn et quand $\mathcal{F}$ est affine. Existe-t'il d'autres cas particuliers pour lesquels SAT($\mathcal{F}$) est résoluble en temps polynomial ? Schaefer [SCH 78] a établi une caractérisation de la complexité des problèmes de décision en fonction du type de contraintes, qui montre que les seuls cas où SAT($\mathcal{F}$) est résoluble en temps polynomial sont les cas précédents ainsi que le cas trivial où $\mathcal{F}$ est 0 ou 1-valide. Dans ce dernier cas, une des deux assignations triviales (l'assignation 0 pour chaque variable ou l'assignation 1 pour chaque variable) est une solution réalisable. Par exemple, MONOTONE-SAT est résoluble en temps polynomial car il rentre dans ce dernier cas.

THÉORÈME 1.4.– *[Théorème dichotomique pour* SAT($\mathcal{F}$) *[SCH 78]]*
Étant donné un $\mathcal{F}$-ensemble de contraintes, le problème SAT($\mathcal{F}$) *est dans P si $\mathcal{F}$ satisfait une des conditions suivantes, et* SAT($\mathcal{F}$) *est NP-complet autrement.*

- *$\mathcal{F}$ est 0-valide (1-valide)*
- *$\mathcal{F}$ est Horn (anti-Horn)*
- *$\mathcal{F}$ est affine*
- *$\mathcal{F}$ est bijonctive*

1.4. Complexité et approximation des problèmes d'optimisation

Dans cette section nous présentons d'abord un algorithme polynomial pour résoudre MAX SAT($\mathcal{F}$) quand $\mathcal{F}$ est 2-monotone. Ensuite, nous mettons en évidence quelques méthodes classiques qui permettent d'établir des résultats positifs d'approximation pour MAX SAT($\mathcal{F}$). Nous citons également d'autres résultats positifs et négatifs existant dans la littérature sur MAX SAT($\mathcal{F}$) et MIN SAT DELETION($\mathcal{F}$).

1.4.1. *Problèmes de maximisation*

Si un problème SAT($\mathcal{F}$) est *NP*-difficile alors le problème MAX SAT($\mathcal{F}$) correspondant est aussi *NP*-difficile. Cependant, il existe des problèmes de maximisation qui deviennent difficiles même si les problèmes de décision correspondants sont faciles. Ainsi, MAX 2SAT est *NP*-difficile [GAR 74], MAX HORN SAT est *NP*-difficile [KOH 94] même si 2SAT et HORN SAT admettent des algorithmes polynomiaux. Néanmoins, dans certains cas, MAX SAT($\mathcal{F}$) est polynomial. Un premier cas trivial est celui où $\mathcal{F}$ est 0 ou 1-valide, toutes les contraintes étant alors nécessairement satisfaites.

Nous avons vu dans la section précédente que SAT($\mathcal{F}$) est polynomial quand $\mathcal{F}$ est 2-monotone (en utilisant l'algorithme pour $\mathcal{F}$ Horn ou anti-Horn). En fait, on peut établir un résultat plus fort qui permet de déterminer en temps polynomial une assignation maximisant le nombre de contraintes satisfaites.

THÉORÈME 1.5.– *[Creignou [CRE 95a], Khanna, Sudan, Williamson [KHA 97b]]*
MAX SAT($\mathcal{F}$) *est polynomial quand chaque fonction de $\mathcal{F}$ est une fonction 2-monotone.*

Preuve : On considère le problème équivalent MIN SAT DELETION($\mathcal{F}$) que l'on réduit au problème s-t MIN CUT dans un graphe orienté. Considérons une instance I du problème MAX SAT($\mathcal{F}$) sur n variables et avec m contraintes, chaque fonction de $\mathcal{F}$ étant une fonction 2-monotone de type

1) $x_{i_1} \wedge \ldots \wedge x_{i_p}$
2) $\bar{x}_{\ell_1} \wedge \ldots \wedge \bar{x}_{\ell_q}$
3) $(x_{i_1} \wedge \ldots \wedge x_{i_p}) \vee (\bar{x}_{\ell_1} \wedge \ldots \wedge \bar{x}_{\ell_q})$

On construit un graphe orienté $G_I = (V, A)$ où V contient 2 sommets spéciaux F, T, un sommet x_i pour chacune des n variables x_i et un sommet v_j pour une contrainte C_j de type 1, un sommet $\bar{v}_j$ pour une contrainte C_j de type 2 et deux sommets v_j et $\bar{v}_j$ pour une contrainte C_j de type 3. Pour construire l'ensemble des arcs on procède comme suit :

– pour une contrainte C_j de type 1, on crée un arc de coût ∞ de x_{i_k} à v_j pour $k = 1, \ldots, p$ et un arc de coût 1 de v_j à T.

– pour une contrainte C_j de type 2, on crée un arc de coût ∞ de $\bar{v}_j$ à x_{ℓ_k} pour $k = 1, \ldots, q$ et un arc de coût 1 de F à $\bar{v}_j$.

– pour une contrainte C_j de type 3, on crée un arc de coût ∞ de x_{i_k} à v_j pour $k = 1, \ldots, p$, un arc de coût ∞ de $\bar{v}_j$ à x_{ℓ_k} pour $k = 1, \ldots, q$ et un arc de coût 1 de v_j à $\bar{v}_j$.

Nous justifions ensuite que la valeur d'une coupe minimale de F à T correspond à une assignation avec un nombre minimum de contraintes non satisfaites. Rappelons que la valeur d'une coupe engendrée par une partition (A, B) avec $F \in A$ et $T \in B$ est la somme des coûts des arcs dont l'extrémité initiale appartient à A et l'extrémité terminale appartient à B.

Étant donnée une coupe C^* de valeur minimale de F à T, considérons l'assignation qui attribue 0 (respectivement 1) aux variables qui se trouvent dans la même partie que F (respectivement T). Si un arc de coût 1 de v_j à T, correspondant à une contrainte de type 1, fait partie de la coupe C^*, alors au moins une des variables $x_{i_1}, \ldots, x_{i_p}$ est fixée à 0 car sinon les sommets correspondant à ces variables sont tous dans la même partie que T dans la coupe C^* et alors en mettant v_j du coté T de la coupe on obtiendrait une coupe de valeur inférieure à la valeur de la coupe C^*, ce qui contredit le fait que C^* est une coupe de valeur minimale. Ainsi la contrainte C_j n'est pas satisfaite. De la même manière on peut justifier que si un arc de coût 1 de F à $\bar{v}_j$, correspondant à une contrainte de type 2, fait partie de la coupe C^*, alors la contrainte C_j correspondante n'est pas satisfaite. De plus, si un arc de coût 1 de v_j à $\bar{v}_j$, correspondant à une contrainte de type 3, fait partie de la coupe C^*, alors au moins une parmi les variables $x_{i_1}, \ldots, x_{i_p}$ est fixée à 0 et au moins une parmi les variables $x_{\ell_1}, \ldots, x_{\ell_q}$ est fixée à 1 et donc la contrainte C_j correspondante n'est pas satisfaite.

Considérons maintenant une assignation pour $x_1, \ldots, x_n$ qui minimise le nombre de contraintes non satisfaites. La valeur de la coupe suivante est égale au nombre de contraintes non satisfaites par l'assignation précédente :

– placer les sommets correspondant aux variables fixées à 0 (respectivement 1) dans cette assignation dans la même partie que F (respectivement T).

– placer le sommet v_j correspondant à une contrainte C_j de type 1 dans la partie de T (ou F) si C_j est satisfaite (ou non satisfaite).

– placer le sommet $\bar{v}_j$ correspondant à une contrainte C_j de type 2 dans la partie de F (ou T) si C_j est satisfaite (ou non satisfaite).

– si C_j est une contrainte de type 3, si $x_{i_1} \wedge \ldots \wedge x_{i_p}$ est satisfaite mettre v_j dans la partie de T sinon dans la partie de F et si $\bar{x}_{\ell_1} \wedge \ldots \wedge \bar{x}_{\ell_q}$ est satisfaite placer $\bar{v}_j$ dans la partie de F sinon dans la partie de T.

$\square$

Ainsi, MAX SAT($\mathcal{F}$) est résoluble en temps polynomial quand chaque fonction de $\mathcal{F}$ est une fonction 0-valide, 1-valide ou 2-monotone. Le théorème de classification pour MAX SAT($\mathcal{F}$) établit que les cas précédents sont les seuls cas pour lesquels le problème est facile.

THÉORÈME 1.6.– *[Théorème de classification pour* MAX SAT($\mathcal{F}$) *[CRE 95a, KHA 97b]]* MAX SAT($\mathcal{F}$) *est dans P si $\mathcal{F}$ est 0-valide ou 1-valide ou 2-monotone et* MAX SAT($\mathcal{F}$) *est APX-complet autrement.*

Dans la suite nous allons établir quelques algorithmes d'approximation pour un problème difficile, MAX SAT. Un premier algorithme d'approximation très simple a été proposé par Johnson [JOH 74].

THÉORÈME 1.7.– *[JOH 74]* MAX SAT *est approximable à un facteur $\frac{1}{2}$ près.*

Preuve : Considérons une instance avec m clauses $C_1, \ldots, C_m$ sur n variables $x_1, \ldots, x_n$, dont la valeur optimale est noté *opt*. L'algorithme consiste, pour chaque variable x_i, à considérer $x_i = 1$ avec la probabilité $\frac{1}{2}$ et $x_i = 0$ avec la probabilité $\frac{1}{2}$. Cet algorithme fournit une approximation à un facteur $\frac{1}{2}$ près. Soit W la variable aléatoire qui représente le nombre de clauses satisfaites, alors l'espérance de cette variable aléatoire est :

$$E(W) = \sum_{j=1}^{m} P(C_j \text{ est satisfaite}) = \sum_{j=1}^{m} (1 - \frac{1}{2^{|C_j|}}) \geq \frac{m}{2} \geq \frac{opt}{2}.$$

En utilisant la méthode de l'espérance conditionnelle proposée par Erdös et Selfridge [ERD 73], on peut transformer cet algorithme en un algorithme déterministe avec la même garantie de performance comme suit :

Nous allons fixer des valeurs aux variables dans l'ordre $x_1, \ldots, x_n$. Supposons qu'on a fixé les valeurs $b_1, \ldots, b_i$ aux variables $x_1, \ldots, x_i$. Calculons $E(W|x_1 = b_1, \ldots, x_i = b_i, x_{i+1} = 0)$ et $E(W|x_1 = b_1, \ldots, x_i = b_i, x_{i+1} = 1)$ et soit $x_{i+1} = 0$ si $E(W|x_1 = b_1, \ldots, x_i = b_i, x_{i+1} = 0) \geq E(W|x_1 = b_1, \ldots, x_i = b_i, x_{i+1} = 1)$ et $x_{i+1} = 1$ sinon. Comme

$$E(W|x_1 = b_1, \ldots, x_i = b_i) = \frac{1}{2}E(W|x_1 = b_1, \ldots, x_i = b_i, x_{i+1} = 1) + \frac{1}{2}E(W|x_1 = b_1, \ldots, x_i = b_i, x_{i+1} = 0)$$

alors

$$\max\{E(W|x_1 = b_1, \ldots, x_i = b_i, x_{i+1} = 1), E(W|x_1 = b_1, \ldots, x_i = b_i, x_{i+1} = 0)\} \geq E(W|x_1 = b_1, \ldots, x_i = b_i)$$

L'assignation trouvée à la fin $x_1 = b_1, \ldots, x_n = b_n$ a la valeur égale à

$$E(W|x_1 = b_1, \ldots, x_n = b_n) \geq E(W) \geq \frac{opt}{2}.$$

□

En utilisant la méthode de l'arrondi aléatoire, Goemans, Williamson [GOE 94] ont amélioré le résultat précédent.

THÉORÈME 1.8.– *[GOE 94]* MAX SAT *est approximable à un facteur* $1 - \frac{1}{e} \approx 0,632$ *près.*

Preuve : Soit I une instance de MAX SAT avec m clauses $C_1, \ldots, C_m$ sur n variables $x_1, \ldots, x_n$. L'algorithme est le suivant :

1) Formuler MAX SAT comme un programme linéaire en variables 0-1. On associe à chaque variable booléenne x_i une variables 0-1 y_i et à chaque clause C_j une variable z_j telle que z_j va prendre la valeur 1 si et seulement si C_j est satisfaite. Soit $C_j^+ = \{i : x_i \in C_j\}$ et $C_j^- = \{i : \bar{x}_i \in C_j\}$. Alors le programme linéaire associé à MAX SAT est :

$$(Sat) \left\{ \begin{array}{l} \max \sum_{j=1}^m z_j \\ \sum_{i \in C_j^+} y_i + \sum_{i \in C_j^-} (1 - y_i) \geq z_j \qquad (j = 1, \ldots, m) \\ y_i \in \{0, 1\} \quad (i = 1, \ldots, n), \qquad z_j \in \{0, 1\} \quad (j = 1, \ldots, m) \end{array} \right.$$

2) Résoudre le problème relaxé (P).

$$(P) \left\{ \begin{array}{l} \max \sum_{j=1}^m z_j \\ \sum_{i \in C_j^+} y_i + \sum_{i \in C_j^-} (1 - y_i) \geq z_j \qquad (j = 1, \ldots, m) \\ y_i \in [0, 1] \quad (i = 1, \ldots, n), \qquad z_j \in [0, 1] \quad (j = 1, \ldots, m) \end{array} \right.$$

Soit (y^*, z^*) la solution optimale trouvée.

3) Considérons l'assignation : $x_i = 1$ avec la probabilité y_i^* et $x_i = 0$ avec la probabilité $1 - y_i^*$.

Soit W la variable aléatoire qui représente le nombre de clauses satisfaites. Alors l'espérance de cette variable aléatoire est :

$$E(W) = \sum_{j=1}^{m} P(C_j \text{ est satisfaite}) = \sum_{j=1}^{m}(1 - \Pi_{i\in C_j^+}(1 - y_i^*)\Pi_{i\in C_j^-} y_i^*)$$

Nous allons montrer que $1 - \Pi_{i\in C_j^+}(1 - y_i^*)\Pi_{i\in C_j^-} y_i^* \geq (1 - \frac{1}{e})z_j^*$.

Pour cela, montrons d'abord que pour toute solution (y, z) de (P) et toute clause C_j avec k littéraux, on a

$$1 - \Pi_{i\in C_j^+}(1 - y_i)\Pi_{i\in C_j^-} y_i \geq c_k z_j,$$

où $c_k = 1 - (1 - \frac{1}{k})^k$.

Dans (*Sat*), l'inégalité correspondant à C_j est :

$$\sum_{i\in C_j^+} y_i + \sum_{i\in C_j^-}(1 - y_i) \geq z_j \iff$$

$$|C_j^+| + |C_j^-| - \sum_{i\in C_j^+} y_i - \sum_{i\in C_j^-}(1 - y_i) \leq k - z_j \iff$$

$$\sum_{i\in C_j^+}(1 - y_i) + \sum_{i\in C_j^-} y_i \leq k - z_j.$$

Sachant que l'on a l'inégalité classique $\frac{a_1+\ldots+a_k}{k} \geq \sqrt[k]{a_1 \ldots a_n}, \forall a_1, \ldots, a_k \geq 0$, on a

$$1 - \Pi_{i\in C_j^+}(1 - y_i)\Pi_{i\in C_j^-} y_i \geq 1 - \left(\frac{\sum_{i\in C_j^+}(1 - y_i) + \sum_{i\in C_j^-} y_i}{k}\right)^k$$

$$\geq 1 - \left(\frac{k - z_j}{k}\right)^k = 1 - \left(1 - \frac{z_j}{k}\right)^k$$

Considérons la fonction $f(x) = 1 - (1 - \frac{x}{k})^k$. On peut vérifier facilement que f est concave, $f(0) = 0$ et $f(1) = 1 - (1 - \frac{1}{k})^k = c_k$. Sachant que si f est concave, alors pour montrer que $f(x) \geq ax + b$, pour $x \in [u, v]$, il est suffisant de montrer que $f(u) \geq au + b$ et $f(v) \geq av + b$, on en déduit que $f(x) \geq c_k x$, pour $x \in [0, 1]$.

Ainsi,

$$1 - \Pi_{i \in C_j^+}(1 - y_i)\Pi_{i \in C_j^-} y_i \geq 1 - \left(1 - \frac{z_j}{k}\right)^k \geq c_k z_j$$

Comme $c_1(= 1) > c_2(= \frac{3}{4}) > \ldots > c_k > \ldots > 1 - \frac{1}{e}$, on a :

$$1 - \Pi_{i \in C_j^+}(1 - y_i^*)\Pi_{i \in C_j^-} y_i^* \geq (1 - \frac{1}{e})z_j^*$$

pour tout $j = 1, \ldots, m$.

Ainsi, nous obtenons finalement

$$E(W) \geq \sum_{j=1}^{m}(1 - \frac{1}{e})z_j^* = (1 - \frac{1}{e})opt_P \geq (1 - \frac{1}{e})opt_{Sat}.$$

En utilisant la méthode de l'espérance conditionnelle proposée par Erdös et Selfridge [ERD 73], on peut transformer cet algorithme en un algorithme déterministe avec la même garantie de performance comme dans le Théorème 1.7. □

Goemans et Williamson [GOE 94] ont amélioré ensuite l'algorithme précédent pour MAX SAT.

THÉORÈME 1.9.– *[GOE 94]* MAX SAT *est approximable à un facteur* $\frac{3}{4}$ *près.*

Preuve : L'algorithme consiste à attribuer à un bit b la valeur 0 avec la probabilité $\frac{1}{2}$ et la valeur 1 avec la probabilité $\frac{1}{2}$. Si $b = 0$ on applique l'algorithme de Johnson et si $b = 1$ on applique l'algorithme de l'arrondi aléatoire précédent.

Pour une clause C_j de taille k, soit W_j la variable aléatoire qui indique si la clause est satisfaite.

$$E(W_j) = \frac{1}{2}[E(W_j|b = 0) + E(W_j|b = 1)]$$

$$E(W_j|b = 0) = 1 - \frac{1}{2^k} \geq (1 - \frac{1}{2^k})z_j^*$$

$$E(W_j|b = 1) = 1 - (1 - \frac{1}{k})^k \geq (1 - (1 - \frac{1}{k})^k)z_j^*$$

Donc, $E(W_j) \geq \frac{3}{4}z_j^*$ et $E(W) = \sum_{i=1}^{m} E(W_j) \geq \frac{3}{4}opt_{Sat}$.

Utilisation de la méthode de l'espérance conditionnelle permet de retrouver un algorithme déterministe avec la même garantie de performance. □

Le résultat précédent n'est pas le meilleur connu dans la littérature concernant l'approximation de MAX SAT. Asano et Williamson [ASA 00] ont établi un algorithme d'approximation à un facteur 0,784 près pour MAX SAT. L'algorithme de Johnson pour MAX SAT [JOH 74] établit également une approximation à un facteur $\frac{2^k-1}{2^k}$ près pour MAX EkSAT, $k \geq 2$. Une autre méthode qui a permis d'obtenir de meilleurs résultats d'approximation pour MAX SAT et ses variantes consiste à modéliser le problème comme un programme semi-défini et d'utiliser l'arrondi aléatoire [GOE 95]. Ainsi, en suivant cette méthode pour la version MAX 2SAT, Feige et Goemans [FEI 95] ont obtenu le meilleur algorithme d'approximation qui donne une approximation à 0,931 près. Du coté négatif, Papadimitriou et Yannakakis [PAP 88] ont montré que MAX kSAT, $k \geq 2$ est *MAX SNP*-difficile, ce qui implique qu'il n'a pas de schéma d'approximation en temps polynomial. Ultérieurement, Håstad [HÅS 97] a montré que même la version MAX EkSAT, $k \geq 3$, n'est pas approximable à un facteur $(\frac{2^k-1}{2^k} - \varepsilon)$ près, pour tout $\varepsilon > 0$ et que MAX E2SAT n'est pas approximable à un facteur $(\frac{21}{22} - \varepsilon)$ près, pour tout $\varepsilon > 0$, si *P*≠*NP*.

En utilisant également la relaxation de la programmation entière et l'arrondi aléatoire, Trevisan [TRE 96] a montré que MAX kCONJ, $k \geq 2$, est approximable à un facteur $\frac{1}{2^{k-1}}$ près. MAX CONJ est aussi difficile à approximer que MAX INDEPENDENT SET [CRE 96], c'est-à-dire qu'il n'est pas approximable à un facteur $\frac{1}{m^{1-\varepsilon}}$ près, pour tout $\varepsilon > 0$ si *NP*≠ *ZPP*, où m est le nombre de contraintes.

L'algorithme de Johnson pour MAX SAT [JOH 74] peut être appliqué également pour MAX LIN2 et MAX kLIN2, $k \geq 2$, fournissant une approximation à un facteur $\frac{1}{2}$ près. Håstad [HÅS 97] a montré que même la version MAX E3LIN n'est pas approximable à un facteur $(\frac{1}{2} - \varepsilon)$ près, pour tout $\varepsilon > 0$ et que MAX E2LIN n'est pas approximable à un facteur $(\frac{11}{12} - \varepsilon)$ près, pour tout $\varepsilon > 0$, si *P*≠*NP*.

1.4.2. *Problèmes de minimisation*

Considérons le problème MIN SAT DELETION($\mathcal{F}$). Compte tenu de l'équivalence de ce problème avec MAX SAT($\mathcal{F}$) du point de vue complexité exacte, les cas polynomiaux pour MIN SAT DELETION($\mathcal{F}$) sont exactement les mêmes que pour MAX SAT($\mathcal{F}$), à savoir quand $\mathcal{F}$ est 0-valide, 1-valide et 2-monotone.

Considérons maintenant la complexité d'approximation. Un théorème de classification a été également établi pour MIN SAT DELETION($\mathcal{F}$) par Khanna, Sudan et Trevisan [KHA 97a]. Ce théorème est beaucoup plus complexe que les théorèmes de classification pour SAT($\mathcal{F}$) et MAX SAT($\mathcal{F}$).

Klauck [KLA 96] a montré que MIN 3SAT DELETION n'est pas approximable à un facteur $n^{1-\varepsilon}$ près, pour tout $\varepsilon > 0$ si *P*≠ *NP*, où n est le nombre de variables. Par contre, MIN 2SAT DELETION est approximable à un facteur $O(\log n \log \log n)$ près [KLE 97] et il n'a pas de schéma d'approximation.

MIN kCONJ DELETION, $k \geq 2$, est approximable à un facteur $2(1 - \frac{1}{2^k})$ près [BER 96], et *MAX SNP*-difficile [KOH 94] et donc il n'a pas de schéma d'approximation. MIN 2CONJ DELETION est approximable à un facteur de 1,103 près et il n'est pas approximable à $\frac{7}{6} - \varepsilon$ près pour tout $\varepsilon > 0$ si *P*≠ *NP* et MIN 3CONJ DELETION est approximable à un facteur de 1,213 près et il n'est pas approximable à $\frac{15}{14} - \varepsilon$ près pour tout $\varepsilon > 0$ si *P*≠ *NP* [AVI 02]. MIN CONJ DELETION est aussi difficile à approximer que MIN VERTEX COVER [CRE 96], c'est-à-dire qu'il est approximable à un facteur 2 près et il n'a pas de schéma d'approximation.

Le problème MIN E2-LIN2 DELETION a été montré *MAX SNP*-difficile dans [GAR 93] et n'admet donc pas de schéma d'approximation en temps polynomial. En revanche, il est approximable à un facteur $O(\log n)$ près [GAR 93]. Les problèmes MIN Ek-LIN2 DELETION sont extrêmement difficiles à approximer pour tout $k \geq 3$. En fait, ils ne sont pas approximables en temps polynomial à un facteur $n^{\Omega(1)/\log \log n}$ près, à moins que *P* = *NP* [DIN 98]. Un premier algorithme polynomial avec un facteur d'approximation sous-linéaire, $O(n/\log n)$, a été établi pour le problème général MIN Ek-LIN2 DELETION [BER 02].

1.5. Instances particulières de problèmes de satisfaction de contraintes

Certains problèmes d'optimisation deviennent plus faciles à approximer lorsqu'on se restreint à des instances particulières. Dans cette partie nous allons étudier divers types d'instances particulières de problèmes d'optimisation : les instances planaires, denses, avec un nombre borné d'occurrences de chaque variable.

1.5.1. *Instances planaires*

On parle en général d'instances planaires d'un problème quand le problème est défini sur un graphe. Dans le cas des problèmes de satisfaisabilité, il existe une manière naturelle d'associer un graphe à un tel problème.

DÉFINITION 1.4.– *Etant donnée une instance I d'un problème de satisfaction de contraintes booléennes, m contraintes $C_1, \ldots, C_m$ définies sur n variables booléennes $x_1, \ldots, x_n$, le graphe associé $G_I = (V, E)$ est un graphe biparti défini ainsi :*

– $V = \{x_1, \ldots, x_n\} \cup \{C_1, \ldots, C_m\}$ où x_i est le sommet associé à la variable x_i et C_j est le sommet associé à la contrainte C_j.

– $E = \{(x_i, C_j) : x_i$ apparaît dans $C_j\}$.

DÉFINITION 1.5.– *Une instance d'un problème de satisfaisabilité est planaire si le graphe associé est planaire.* PLANAR A *est le problème* A *réduit aux instances planaires, où* A *est un problème de décision ou d'optimisation.*

La complexité des instances planaires a été étudiée depuis longtemps. Par exemple Lichtenstein a montré dans [LIC 82] que PLANAR 3SAT reste *NP*-difficile et Dyer et Frieze [DYE 86] ont montré que PLANAR 1IN3SAT reste *NP*-difficile. Plus généralement, on peut montrer [CRE 01] que pour chaque $\mathcal{F}$-ensemble de contraintes, si SAT($\mathcal{F}$) est *NP*-complet, alors PLANAR SAT($\mathcal{F} \cup \{F, T\}$) est aussi *NP*-complet. De plus, si l'ensemble $\mathcal{F}$ n'est pas fermé par complément, alors PLANAR SAT($\mathcal{F}$) est *NP*-complet quand SAT($\mathcal{F}$) est *NP*-complet. Un exemple de problème SAT($\mathcal{F}$) où $\mathcal{F}$ est fermé par complément est NAE3SAT. Kratochvil et Tuza [KRA 00] ont montré que PLANAR NAE3SAT est polynomial tandis que NAE3SAT est *NP*-difficile.

En ce qui concerne la complexité d'approximation, Hunt et al. [HUN 94] ont donné un schéma d'approximation en temps polynomial pour PLANAR MAX kSAT($\mathcal{F}$) pour tout ensemble $\mathcal{F}$ ce qui implique, par exemple, que PLANAR MAX 3SAT a un schéma d'approximation. Khanna et Motwani [KHA 96] ont généralisé le résultat précédent en montrant que PLANAR MAX SAT et plus généralement PLANAR MAX SAT($\mathcal{F}$) ont un schéma d'approximation.

Avant d'expliquer l'idée de ce dernier schéma, définissons la notion de graphe t-extérieur planaire.

DÉFINITION 1.6.– *Un graphe 1-extérieur planaire est un graphe planaire qui admet une représentation dans le plan où tous les sommets apparaissent sur la face extérieure. Un graphe t-extérieur planaire est un graphe planaire qui a une représentation dans le plan telle qu'en effaçant les sommets sur la face extérieure on obtient un graphe $(t-1)$-extérieur planaire.*

THÉORÈME 1.10.– *[KHA 96]* PLANAR MAX SAT($\mathcal{F}$) *a un schéma d'approximation.*

Preuve : Soit I une instance de PLANAR MAX SAT($\mathcal{F}$) avec n variables et m contraintes et soit $G_I = (V, E)$ le graphe associé à I. Puisque $|V| = n + m$, le graphe G_I est t-extérieur planaire où $t \leq n + m$. Soit $L_1, \ldots, L_t$ les ensembles de sommets tels que L_t correspond à la face extérieure et chaque L_i est la face extérieure obtenue en enlevant les sommets des ensembles $L_t, \ldots, L_{i+1}$.

Considérons une assignation optimale pour I et soit n_i le nombre de contraintes satisfaites correspondant aux sommets appartenant à L_i. On partitione les faces $L_1, \ldots, L_t$ en $p + 1$ groupes $S_0, \ldots, S_p$ (où p va être déterminé en fonction de l'erreur maximale ε avec laquelle on veut trouver une solution) où chaque groupe S_r est l'union des faces L_i où i est égal à $3r$, $3r + 1$ ou $3r + 2$ modulo q et $q = 3(p + 1)$. En utilisant le principe des tiroirs on peut déduire qu'il existe un groupe S_j tel que

$\sum_{L_i \in S_j} n_i \leq \frac{opt(I)}{p+1}$. Ce groupe va être déterminé en essayant toutes les possibilités et en choisissant la meilleure solution. Quand on choisit S_j, on efface les sommets des faces avec un indice égal à $3j+1$ modulo q, séparant ainsi le graphe en une famille de graphes disjoints $(q-1)$-extérieur planaires, $G_1, G_2, \ldots, G_\ell$ telle que la somme totale des n_i correspondant est au moins $(1 - \frac{1}{p+1})opt(I)$. Un graphe k-extérieur planaire a une largeur d'arbre d'au plus $3k-1$ ([BOD 98]). En utilisant la programmation dynamique on peut établir un algorithme polynomial qui fournit une solution optimale pour les graphes avec une largeur d'arbre bornée, en particulier pour les graphes $G_1, G_2, \ldots, G_\ell$. Comme la somme des valeurs des solutions optimales obtenues pour chaque G_t va être au moins égale à la somme totale des n_i correspondant, lorsqu'on choisit $p = \lceil \frac{1}{\varepsilon} - 1 \rceil$ on obtient une approximation à un facteur $(1-\varepsilon)$ près. $\square$

1.5.2. *Instances denses*

Il existe deux types d'instances denses étudiées dans la littérature : les instances uniformément denses et les instances denses en moyenne.

DÉFINITION 1.7.– *Une instance d'un problème de* MAX kSAT($\mathcal{F}$) *ou* MIN kSAT DELETION($\mathcal{F}$) *sur* n *variables est* uniformément α-dense *si pour chaque variable, le nombre total d'occurrences de la variable et de sa négation est au moins* αn^{k-1} *et il est* α-dense en moyenne *si le nombre de contraintes est d'au moins* αn^k.

DÉFINITION 1.8.– *Un ensemble d'instances est* uniformément dense *s'il existe une constante* $\alpha > 0$ *telle que chaque instance est uniformément* α*-dense et un ensemble d'instances est* dense en moyenne *s'il existe une constante* $\alpha > 0$ *telle que chaque instance est* α*-dense en moyenne.*

Donc, un ensemble d'instances uniformément dense est dense en moyenne mais le contraire n'est pas vrai.

Arora, Karger et Karpinski [ARO 95] ont débuté l'étude systématique de la complexité d'approximation d'instances denses de problèmes d'optimisation. Ils ont montré que les instances denses en moyenne (et uniformément denses) de MAX kSAT, MAX CUT, MAX DICUT, DENSE kSUBGRAPH et plus généralement de tout problème de MAX kSAT($\mathcal{F}$) ont un schéma d'approximation en temps polynomial. Arora, Karger et Karpinski ont remarqué que les optimum des instances denses en moyenne des problèmes de MAX kSAT($\mathcal{F}$) sont "grands" ($\Omega(n^k)$ où n est le nombre des variables) et, que dans ce cas, une approximation additive implique une approximation relative. L'idée de base est de représenter les problèmes comme des programmes mathématiques en nombres entiers d'un certain type [ARO 95], puis d'appliquer des résultats généraux d'approximation pour ces programmes pour obtenir une approximation additive.

Les instances denses de problèmes de minimisation ont été également étudiées. Dans [ARO 95], Arora, Karger et Karpinski ont établi des schémas d'approximation en temps polynomial pour les instances uniformément denses de problèmes de minimisation suivants : MIN BISECTION, MIN kCUT. Pour ces derniers problèmes ils ont utilisé des idées supplémentaires par rapport aux problèmes de maximisation car les valeurs des solutions optimales des instances denses des problèmes de minimisation peuvent être proches de zéro et dans ce cas une approximation additive ne fournit pas forcément une approximation relative.

Bazgan, Fernandez de la Vega [BAZ 99] ont initié l'étude systématique des instances denses des versions minimisation des problèmes de satisfaisabilité par le problème MIN E2-LIN2 DELETION. Plus exactement, ils ont montré [BAZ 99] que les instances uniformément denses de MIN E2-LIN2 DELETION ont un schéma d'approximation en temps polynomial. Dans [BAZ 02, BAZ 03] Bazgan, Fernandez de la Vega et Karpinski ont généralisé le résultat obtenu pour MIN E2-LIN2 DELETION aux deux problèmes MIN kCONJ DELETION, $k \geq 2$ et MIN Ek-LIN2 DELETION, $k \geq 3$ qui appartiennent à MIN kSAT DELETION($\mathcal{F}$).

Le schéma d'approximation en temps polynomial pour les instances uniformément denses de ces problèmes de MIN kSAT DELETION($\mathcal{F}$) est constitué de deux algorithmes (comme dans [ARO 95] pour MIN BISECTION). Le premier garantit une bonne solution quand l'optimum du problème est $\Omega(n^k)$, le second garantit une bonne solution quand l'optimum du problème est $O(n^k)$. Quand l'optimum est grand, l'idée consiste à écrire le problème comme un programme entier d'un certain type puis à utiliser la méthode de [ARO 95] qui fournit une solution avec une erreur additive de l'ordre $O(n^k)$. Quand l'optimum est petit, l'idée de l'algorithme est de réaliser un échantillonnage exhaustif dans un graphe ou hypergraphe et de prendre comme solution la meilleure obtenue en "complétant" chaque possibilité de placer les variables. L'algorithme obtenu est un algorithme aléatoire qui peut être dérandomisé comme dans [ARO 95].

Certains problèmes d'optimisation portant sur des variables booléennes n'admettent pas de schéma d'approximation en temps polynomial sur les instances uniformément denses. Un exemple d'un tel problème est MIN 2SAT DELETION. En fait, on peut rendre uniformément denses les instances de MIN 2SAT DELETION, sans changer la valeur de l'optimum, en ajoutant des copies disjointes des variables originales, puis en ajoutant toutes les conjonctions ayant exactement une variable originale et une copie. Comme MIN 2SAT DELETION n'a pas de schéma d'approximation en temps polynomial, les instances uniformément denses de MIN 2SAT DELETION n'ont pas de schéma d'approximation en temps polynomial.

Soulignons que les instances denses en moyenne de MIN kCONJ DELETION et MIN Ek-LIN2 DELETION, $k \geq 2$, sont aussi difficiles à approximer que les instances générales de ces problèmes [BAZ 03]. L'idée est de construire une réduction du cas général vers le cas particulier en doublant le nombre de variables et considérant toutes les clauses ou équations sur exactement k variables.

En conclusion, les problèmes de MAX kSAT($\mathcal{F}$) ont un schéma d'approximation pour les instances uniformément denses ainsi que pour les instances denses en moyenne, par contre la plupart des problèmes de MIN kSAT DELETION($\mathcal{F}$) qui ont un schéma d'approximation pour les instances uniformément denses restent aussi difficiles à approximer pour les instances denses en moyenne qu'en général.

1.5.3. *Instances avec un nombre borné d'occurrences*

Certains problèmes de décision restent *NP*-complets même dans le cas où chaque variable n'apparaît qu'un nombre borné de fois.

Notons par EtOCC-EkSAT la variante de EkSAT où chaque clause contient exactement k littéraux et chaque variable apparaît exactement t fois positivement ou négativement.

THÉORÈME 1.11.– 3SAT *reste NP-complet même quand chaque variable apparaît au plus 3 fois dont au moins une fois positivement et au moins une fois négativement.*

Preuve : L'idée est de réduire 3SAT à ce cas particulier en remplaçant une variable x qui apparaît $k \geq 3$ fois par k copies $x_1, \ldots, x_k$ et s'assurer que ces k copies prennent la même valeur de vérité en ajoutant les clauses $(\bar{x}_1 \vee x_2) \wedge (\bar{x}_2 \vee x_3) \wedge \ldots \wedge (\bar{x}_k \vee x_1)$. □

Dans le théorème précédent il est important que chaque variable apparaisse au plus (et pas exactement) 3 fois et chaque clause ait au plus (et pas exactement) 3 littéraux car sinon le problème devient polynomial.

THÉORÈME 1.12.– *[[PAP 94], Problème 9.5.4 (b)]* EkOCC-EkSAT*,* $k \geq 2$*, est polynomial.*

Preuve : Soit I une instance de E3OCC-E3SAT avec n variables $x_1, \ldots, x_n$ et n clauses $C_1, \ldots, C_n$. Nous construisons un graphe biparti $G = (V_1, V_2, E)$, où $V_1 = \{x_1, \ldots, x_n\}$, $V_2 = \{C_1, \ldots, C_n\}$ et on crée une arête entre x_i et C_j si et seulement si la clause C_j contient x_i ou $\bar{x}_i$. Le graphe biparti k-régulier ainsi construit contient un couplage parfait $M = \{(x_{i_1}, C_{j_1}), \ldots, (x_{i_n}, C_{j_n})\}$ (en raison du théorème de König-Hall [HAL 34]) qui peut être trouvé en utilisant par exemple l'algorithme de Ford-Fulkerson. L'assignation suivante obtenue à partir de M, satisfait I : considerer $x_{i_\ell} = 1$ si C_{j_ℓ} contient x_{i_ℓ} et $x_{i_\ell} = 0$ si C_{j_ℓ} contient $\bar{x}_{i_\ell}$, pour $\ell = 1, \ldots, n$. □

Tovey [TOV 84] a montré que E4OCC-E3SAT est *NP*-difficile et MAX E4OCC-E3SAT est APX-difficile et Berman, Karpinski, Scott [BER 03c] ont montré que ces résultats restent vrais même pour les variantes de ces problèmes où chaque variable apparaît exactement deux fois positivement et deux fois négativement. Dubois [DUB 90] a montré la *NP*-difficulté de E6OCC-E4SAT et E11OCC-E5SAT. Dans [KRA 93], Kratochvil, Savicky et Tuza ont défini la fonction $f(k)$ comme étant le plus grand t telle que toute instance de EtOCC-EkSAT est toujours satisfaisable et ils ont montré que si $t > f(k)$ alors EtOCC-EkSAT est *NP*-difficile. De plus, $f(k+1) \leq 2f(k)+1$ et $f(k) \geq \lfloor \frac{2^k}{ek} \rfloor$. Berman, Karpinski et Scott [BER 03b] ont montré que si $t > f(k)$ alors MAX EtOCC-EkSAT est *APX*-difficile et ils ont également amélioré certaines bornes pour la fonction f. Plus exactement, $f(5) < 9$ et $f(6) \geq 7$. Dans [KAR 01, BER 03a, BER 03b, BER 03c] on peut trouver certaines bornes inférieures et supérieures d'approximation pour divers problèmes, par exemple pour MAX 3OCC-E2SAT, MAX 3OCC-E2-LIN2 et MIN 3OCC-E3-LIN2 DELETION.

1.6. Problèmes de satisfaisabilité sous contraintes globales

Les contraintes de nature globale apparaissent naturellement dans certains problèmes d'optimisation. Par exemple, MIN BISECTION est le problème MIN CUT sous la contrainte que les deux parties séparées par la coupe doivent être de taille égale. Il est connu que MIN CUT est polynomial tandis que MIN BISECTION est *NP*-difficile. Plusieurs problèmes d'optimisation comme par exemple MAX BISECTION et MIN BISECTION peuvent se formuler comme des problèmes de satisfaction de contraintes booléennes où une solution réalisable est une solution avec autant de variables fixées à 0 que de variables fixées à 1. Par exemple, pour une instance de MIN BISECTION représentée par un graphe G avec n sommets et m arêtes on considère n variables booléennes $x_1, \ldots, x_n$ et m contraintes en associant à chaque arête (i, j) de G la contrainte $x_i \oplus x_j = 0$. Ainsi, MIN BISECTION est le problème consistant à trouver une solution avec autant de variables à 0 qu'à 1 tel qu'un nombre minimum parmi les m contraintes précédentes soit non satisfaites.

DÉFINITION 1.9.– *Une assignation est appelée équilibrée si le nombre de variables fixées à 1 est le même que le nombre de variables fixées à 0.* BALANCED A *est la variante du problème* A *où les solutions réalisables sont des assignations équilibrées et* A *est un problème de décision ou d'optimisation.*

Les complexités exacte et d'approximation des versions équilibrées des problèmes de satisfaisabilité de décision et d'optimisation ont été étudiées par Bazgan et Karpinski dans [BAZ 05]. De manière générale, si un problème est difficile, sa version équilibrée reste difficile. En revanche, plusieurs problèmes triviaux deviennent difficiles.

Plus précisément, si SAT($\mathcal{F}$) est *NP*-complet alors BALANCED SAT($\mathcal{F}$) est aussi *NP*-complet [BAZ 05]. Il est facile de voir que MONOTONE-EkSAT est trivial car l'assignation 1 pour chaque variable (si la formule est composée seulement de littéraux positifs) ou l'assignation 0 pour chaque variable (si la formule est composée seulement de littéraux négatifs) est une assignation satisfaisante. En revanche, BALANCED MONOTONE-EkSAT est *NP*-complet, pour tout $k \geq 2$ ([BAZ 05]). Comme spécifié dans le Théorème 1.3, Ek-LIN2, pour tout $k \geq 2$ est polynomial. Dans le cas équilibré, la situation est différente car pour $k = 2$ le problème reste polynomial, par contre pour $k \geq 3$ le problème devient *NP*-complet même pour un ensemble d'équations linéaires 0-homogène ou 1-homogène [BAZ 05].

Les versions équilibrées des problèmes de maximisation ont été également étudiées. Comme dans le cas des problèmes de décision, on peut montrer que MAX SAT($\mathcal{F}$) est E-réductible à BALANCED MAX SAT($\mathcal{F}$) ([BAZ 05]). Il s'ensuit que la version équilibrée est au moins aussi difficile à approximer que la version générale. De plus, BALANCED MAX MONOTONE-EkSAT est *APX*-difficile, pour tout $k \geq 2$ [BAZ 05]. BALANCED MAX SAT est approximable à un facteur $(1 - \frac{1}{e})$ près [SVI 01] et BALANCED MAX 2SAT est approximable à un facteur 0,66 près [BLA 02] et aléatoirement approximable à un facteur $\frac{3}{4}$ près [HOF 03]. BALANCED MAX MONOTONE-EkCONJ, $k \geq 2$, n'a pas de schéma d'approximation si *NP* $\not\subseteq \cap_{\delta>0}$ *BTIME*(2^{n^δ}) [BAZ 05]. BALANCED MAX E2-LIN2 dans le cas 1-homogène, qui correspond à MAX BISECTION est *APX*-difficile [PAP 88, HÅS 97] et BALANCED MAX E2-LIN2 dans le cas 0-homogène, qui correspond à BALANCED MAX UNCUT n'a pas de schéma d'approximation si *NP* $\not\subseteq \cap_{\delta>0}$ *BTIME*(2^{n^δ}) [BAZ 05]. De plus, BALANCED MAX Ek-LIN2 est *APX*-difficile, pour tout $k \geq 3$ même dans le cas 0-homogène ou 1-homogène [BAZ 05]. Également, utilisant la technique PCP (Probabilistically Checkable Proof), Holmerin [HOL 02] a montré, que le problème BALANCED MAX E4-LIN2 dans le cas 0-homogène n'est pas approximable à un facteur 0,912 près et Holmerin et Khot [HOL 03] ont montré que le problème BALANCED MAX E3-LIN2 dans le cas 0-homogène n'est pas approximable à un facteur $(\frac{3}{4} - \varepsilon)$ près, pour tout $\varepsilon > 0$. Récemment, Holmerin et Khot [HOL 04] ont montré que le problème BALANCED MAX E3-LIN2 dans le cas 0-homogène n'est pas approximable à un facteur $(\frac{1}{2} - \varepsilon)$ près, pour tout $\varepsilon > 0$, si *NP* $\not\subseteq \cap_{\delta>0}$ *DTIME*(2^{n^δ}), obtenant ainsi le meilleur résultat de non-approximabilité pour ce problème car il est facilement approximable à un facteur $\frac{1}{2}$ près.

Pour les problèmes de minimisation, on peut établir le même résultat que pour les problèmes de maximisation : BALANCED MIN SAT DELETION($\mathcal{F}$) est au moins aussi difficile à approximer que MIN SAT DELETION($\mathcal{F}$), pour tout ensemble $\mathcal{F}$. BALANCED MIN MONOTONE-EkSAT DELETION, $k \geq 2$, n'a pas de schéma d'approximation si *P*$\neq$*NP* and BALANCED MIN MONOTONE-EkCONJ DELETION, $k \geq 2$, n'a pas de schéma d'approximation si *NP* $\not\subseteq \cap_{\delta>0}$ *BTIME*(2^{n^δ}) [BAZ 05]. Holmerin et Khot [HOL 03] ont établi une borne inférieure pour une généralisation de MIN

BISECTION. Plus précisément, ils ont montré que BALANCED MIN E3-LIN2 DELETION, même dans les cas 0-homogène et 1-homogène, n'est pas c-approximable, pour toute constante $c > 1$, si $P \neq$ *NP*. BALANCED MIN E2-LIN2 DELETION dans le cas 1-homogène correspond à BALANCED MIN UNCUT qui a été montré *APX*-difficile [GAR 93]. BALANCED MIN E2-LIN2 DELETION dans le cas 0-homogène est MIN BISECTION. La complexité d'approximation de MIN BISECTION n'est pas établie. Le meilleur algorithme approche le problème à un facteur $O(\log n^2)$ près [FEI 00]. Récemment, Khot [KHO 04] a établi que si *NP* $\not\subseteq \cap_{\delta>0}$ *BTIME*(2^{n^δ}) alors MIN BISECTION n'a pas de schéma d'approximation en temps polynomial. BALANCED MIN Ek-LIN2 DELETION, pour $k \geq 4$, dans les cas 0-homogène et 1-homogène n'a pas de schéma d'approximation si $P \neq$ *NP* [BAZ 05].

1.7. Conclusion

Les problèmes de satisfaisabilité restent des problèmes centraux en théorie de la complexité. Ce chapitre montre le progrès théorique considérable réalisé sur l'étude des problèmes de satisfaisabilité pendant les dernières décennies. Nous disposons actuellement d'une caractérisation presque complète de la complexité exacte et d'approximation de ces problèmes ainsi que de diverses instances particulières.

1.8. Bibliographie

[ARO 95] ARORA S., KARGER D., KARPINSKI M., « Polynomial time approximation schemes for dense instances of NP-hard problems », *Proceedings of 27th Annual ACM Symposium on the Theory of Computing*, p. 284–293, 1995, Also published in Journal of Computer and System Sciences 58, 1999, 193–210.

[ASA 00] ASANO T., WILLIAMSON D. P., « Improved approximation algorithms for MAX SAT », *Proceedings of the 11th ACM-SIAM Symposium on Discrete Algorithms*, p. 96-105, 2000.

[AVI 02] AVIDOR A., ZWICK U., « Approximating MIN 2-SAT and MIN 3-SAT », *Proceeding of 13th Annual International Symposium on Algorithms and Computation, 465-475*, vol. LNCS 2518, Springer-Verlag, 2002, Also published in Theory of Computing Systems 38(3), 2005, 329–345.

[BAZ 99] BAZGAN C., FERNANDEZ DE LA VEGA W., « A polynomial time approximation scheme for dense MIN 2SAT », CIOBANU G., PAUN G., Eds., *Proceedings of the 12th International Symposium on the Fundamentals of Computation Theory*, LNCS 1684, Springer-Verlag, p. 91–99, 1999.

[BAZ 02] BAZGAN C., FERNANDEZ DE LA VEGA W., KARPINSKI M., « Approximability of Dense Instances of Nearest Codeword Problem », PENTTONEN M., MEINECHE SCHMIDT E., Eds., *Proceedings of the 8th Scandinavian Workshop on Algorithm Theory*, LNCS 2368, Springer-Verlag, p. 298–307, 2002.

[BAZ 03] BAZGAN C., FERNANDEZ DE LA VEGA W., KARPINSKI M., « Polynomial time approximation schemes for dense instances of the minimum constraint satisfaction », *Random Structures and Algorithms*, vol. 23(1), p. 73–91, 2003.

[BAZ 05] BAZGAN C., KARPINSKI M., « On the Complexity of Global Constraint Satisfaction », *Proceeding of 16th Annual International Symposium on Algorithms and Computation*, vol. LNCS 3827, Springer-Verlag, p. 624–633, 2005.

[BER 96] BERTSIMAS D., TEO C.-P., VOHRA R., « On dependent randomized rounding algorithms », *Proceeding of the 5th International Integer Programming and Combinatorial Optimization Conference*, vol. LNCS 1084, Springer-Verlag, p. 330–344, 1996.

[BER 02] BERMAN P., KARPINSKI M., « Approximation Hardness of Bounded Degree MIN-CSP and MIN-BISECTION », *Proceeding of the 29th International Colloquium on Automata, Languages and Programming*, vol. LNCS 2380, Springer-Verlag, p. 623-632, 2002.

[BER 03a] BERMAN P., KARPINSKI M., Improved Approximation Lower Bounds on Small Occurrence Optimization, ECCC Technical Report, TR03-008, 2003.

[BER 03b] BERMAN P., KARPINSKI M., SCOTT A., Approximation Hardness and Satisfiability of Bounded Occurrence Instances of SAT, ECCC Technical Report, TR03-022, 2003.

[BER 03c] BERMAN P., KARPINSKI M., SCOTT A., Approximation Hardness of Short Symmetric Instances of MAX-3SAT, ECCC Technical Report, TR03-049, 2003.

[BLA 02] BLASER M., MANTHEY B., « Improved Approximation Algorithms for Max 2Sat with Cardinality Constraint », *Proceedings of the 13th Annual International Symposium on Algorithms and Computation*, LNCS 2518, Springer-Verlag, p. 187–198, 2002.

[BOD 98] BODLAENDER H. L., « A partial k-arboretum of graphs with bounded treewidth », *Theoretical Computer Science*, vol. 209, p. 1–45, 1998.

[COO 71] COOK S., « The complexity of theorem-proving procedures », *Proceedings of the 3rd Annual ACM Symposium on Theory of Computing*, p. 151–158, 1971.

[CRA 94] CRAWFORD J., BAKER A., « Experimental results on the application of satisfiability algorithms to scheduling problems », *Proceedings of the 12th National Conference on IA*, p. 1092–1097, 1994.

[CRE 95a] CREIGNOU N., « A dichotomy theorem for maximum generalized satisfiability problems », *Journal of Computer and Systeme Sciences*, vol. 51(3), p. 511–522, 1995.

[CRE 95b] CRESCENZI P., KANN V., A compendium of NP optimization problems, http ://www.nada.kth.se/∼viggo/problemlist/compendium.html, 1995.

[CRE 96] CRESCENZI P., SILVESTRI R., TREVISAN L., « To weight or not to weight : where is the question ? », *Proceedings of the 4th Israeli Symposium on Theory of Computing and Systems*, p. 68–77, 1996.

[CRE 01] CREIGNOU N., KHANNA K., SUDAN M., *Complexity Classifications of Boolean Constraint Satisfaction Problems*, SIAM Monographs on Discrete Mathematics and Applications, 2001.

[DIN 98] DINUR I., KINDLER G., SAFRA S., « Approximating-CVP to within almost-polynomial factors is NP-hard », *Proceeding of the 39th IEEE Annual Symposium on Foundations of Computer Science*, p. 99–109, 1998.

[DOW 84] DOWLING W. F., GALLIER J. H., « Linear time algorithms for testing the satisfiability of propositional Horn formulae », *Journal of Logic Programming*, vol. 3, p. 267–284, 1984.

[DUB 90] DUBOIS O., « On the r,s-SAT satisfiability problem and a conjecture of Tovey », *Discrete Applied Mathematics*, vol. 26, p. 51–60, 1990.

[DYE 86] DYER M., FRIEZE A., « Planar 3DM is NP-complete », *Journal of Algorithms*, vol. 7, p. 174–184, 1986.

[ERD 73] ERDŐS P., SELFRIDGE J., « On a combinatorial game », *Journal of Combinatorial Theory A*, vol. 14, p. 298–301, 1973.

[FEI 95] FEIGE U., GOEMANS M. X., « Approximating the value of two prover proof systems, with applications to MAX 2SAT and MAX DICUT », *Proceedings of the 3rd Istrael Symposium on Theory of Computing and Systems*, p. 182–189, 1995.

[FEI 00] FEIGE U., KRAUTHGAMER R., « A polylogarithmic approximation of the Minimum Bisection », *Proceedings of the 41st Annual Symposium on Foundations of Computer Science*, p. 105–115, 2000.

[GAR 74] GAREY M. R., JOHNSON D. S., STOCKMEYER L., « Some simplified NP-complete problems », *Proceedings of the Conference record of 6th Annual ACM Symposium on Theory of Computing*, p. 47–63, 1974, Also published in Theoretical Computer Science, 1, 1976, 237–267.

[GAR 79] GAREY M. R., JOHNSON D. S., *Computers and Intractability / A Guide to the Theory of NP-Completeness*, W.H. Freeman & Company, San Francisco, 1979.

[GAR 93] GARG N., VAZIRANI V., YANNAKAKIS M., « Approximate Max-flow Min-(multi)cut Theorems and Their Applications », *Proceedings of the 25th Annual ACM Symposium on the Theory of Computing*, p. 698–707, 1993, Also published in SIAM Journal on Computing, 25, 1996, 235–251.

[GEN 99] GENT I., WALSH T., The search for Satisfaction, Rapport, 1999, Internal Report, Dept. of Computer Science, University of Strathclyde.

[GOE 94] GOEMANS M., WILLIAMSON D., « New 3/4-approximation algorithms for Max Sat », *SIAM Journal on Discrete Mathematics*, vol. 7, p. 656–666, 1994.

[GOE 95] GOEMANS M., WILLIAMSON D., « Improved Approximation Algorithms for Maximum Cut and Satisfiability Problems Using Semidefinite Programming », *Journal of the ACM*, vol. 42, p. 1115–1145, 1995.

[GU 97] GU J., PURDOM P., FRANCO J., WAH B., « Algorithms for the Satisfiability problem : a survey », *DIMACS Series on Discrete Mathematics and Theoretical Computer Science*, American Mathematical Society 35, p. 19–151, 1997.

[HAL 34] HALL P., « On representations of subsets », *Journal London Math. Soc.*, vol. 10, page 26, 1934.

[HÅS 97] HÅSTAD J., « Some optimal inapproximability results », *Proceedings of the 29th Annual ACM Symposium on the Theory of Computing*, p. 1–10, 1997.

[HOF 03] HOFMEISTER T., « An Approximation Algorithm for MAX-2-SAT with Cardinality Constraint », *Proceedings of the 11th Annual European Symposium on Algorithms*, LNCS 2832, Springer-Verlag, p. 301–312, 2003.

[HOL 02] HOLMERIN J., PCP with Global Constraints - Balanced Homogeneous Linear Equations, manuscript, 2002.

[HOL 03] HOLMERIN J., KHOT S., « A strong inapproximability result for a generalization of Minimum Bisection », *Proceedings of the 18th IEEE Conference on Computational Complexity*, p. 371–378, 2003.

[HOL 04] HOLMERIN J., KHOT S., « A new PCP Outer Verifier with Applications to Homogeneous Linear Equations and Max-Bisection », *Proceedings of the 36th Annual ACM Symposium on Theory of Computing*, p. 11–17, 2004.

[HOO 98] HOOS H., Stochastic Local Search - Methods, Models, Applications, http ://www.cs.ubc.ca/spider/hoos/publ-ai.html, 1998, PhD thesis, TU Darmstadt.

[HUN 94] HUNT III H., MARATHE M., RADHAKRISHNAN V., RAVI S., ROSENKRANTZ D., STEARNS R., « Approximation Schemes using L-reductions », *Proceedings of the 14th Conference on Foundations of Software Technology and Theoretical Computer Science*, LNCS 880, Springer-Verlag, p. 342–353, 1994.

[JIA 95] JIANG Y., KAUTZ H., SELMAN B., « Solving problems with hard and soft constraints using a stochastic algorithm for Max-Sat », *First International Joint Workshop on Artificial Intelligence and Operation Research*, p. 1–15, 1995.

[JOH 74] JOHNSON D. S., « Approximation algorithms for combinatorial problems », *Journal of Computer and System Sciences*, vol. 9, p. 256–278, 1974.

[JON 77] JONES N. D., LAASER W. T., « Complete problems for deterministic polynomial time », *Theoretical Computer Sciences*, vol. 3, p. 107–117, 1977.

[KAR 01] KARPINSKI M., « Approximating bounded degree instances of NP-hard problems », *Proceedings of the 13th Symposium on Fundamentals of Computation Theory*, LNCS 2138, Springer-Verlag, p. 24–34, 2001.

[KAU 92] KAUTZ H., SELMAN B., « Planning as Satisfiability », *Proceedings of the 10th ECAI*, p. 359–363, 1992.

[KHA 96] KHANNA S., MOTWANI R., « Towards a syntactic characterization of PTAS », *Proceedings of the 28th Annual ACM Symposium on Theory of Computing*, p. 329–337, 1996.

[KHA 97a] KHANNA S., SUDAN M., TREVISAN L., « Constraint Satisfaction : The Approximability of Minimization Problems », *Proceedings of the 12th Annual IEEE Conference on Computational Complexity*, p. 282–296, 1997.

[KHA 97b] KHANNA S., SUDAN M., WILLIAMSON D., « A Complete Classification of the Approximability of Maximization Problems Derived from Boolean Constraint Satisfaction », *Proceedings of the 29th Annual ACM Symposium on Theory of Computing*, p. 11–20, 1997.

[KHO 04] KHOT S., « Ruling Out PTAS for Graph Min-Bisection, Densest Subgraph and Bipartite Clique », *Proceedings of the 45th Annual IEEE Symposium on Foundations of Computer Science*, p. 136–145, 2004.

[KLA 96] KLAUCK H., « On the hardness of global and local approximation », *Proceedings of the 5th Scandinavian Workshop on Algorithm Theory*, LNCS 1097, Springer-Verlag, p. 88–99, 1996.

[KLE 97] KLEIN P., PLOTKIN S., RAO S., TARDOS E., « Approximation Algorithms for Steiner and Directed Multicuts », *Journal of Algorithms*, vol. 22, n˚2, p. 241-269, 1997.

[KOH 94] KOHLI R., KRISHNAMURTI R., MIRCHANDANI P., « The Minimum Satisfiability Problem », *SIAM Journal on Discrete Mathematics*, vol. 7, p. 275–283, 1994.

[KRA 93] KRATOCHVIL J., SAVICKY P., TUZA Z., « One more occurence of variable makes satisfiability jump from trivial to NP-complete », *SIAM Journal on Computing*, vol. 22(1), p. 203–210, 1993.

[KRA 00] KRATOCHVIL J., TUZA Z., « On the complexity of bicoloring clique hypergraphs of graphs (extended abstract) », *Proceedings of the 11th Annual ACM-SIAM Symposium on Discrete Algorithms*, p. 40–41, 2000.

[LEV 73] LEVIN L. A., « Universal sorting problems », *Problems of Information Transmission*, vol. 9, p. 265–266, 1973.

[LIC 82] LICHTENSTEIN D., « Planar formulae and their uses », *SIAM Journal on Computing*, vol. 11(2), p. 329–343, 1982.

[MIN 88] MINOUX M., « LTUR : A simplified linear-time unit resolution algorithm for Horn formulae and computer implementation », *Information Processing Letters*, vol. 29, p. 1–12, 1988.

[PAP 88] PAPADIMITRIOU C., YANNAKAKIS M., « Optimization, Approximation, and Complexity Classes », *Proceedings of the 20th Annual ACM Symposium on Theory of Computing*, p. 229–234, 1988, Also published in Journal of Computer and System Sciences, 43, 1991, 425–440.

[PAP 94] PAPADIMITRIOU C., *Computational Complexity*, Addison-Wesley Publishing Company, 1994.

[SCH 78] SCHAEFER T., « The complexity of satisfiability problems », *In Conference Record of the 10th Annual ACM Symposium on Theory and Computing*, p. 216–226, 1978.

[SVI 01] SVIRIDENKO M. I., « Best possible approximation algorithm for MAX SAT with cardinality constraint », *Algorithmica*, vol. 30(3), p. 398–405, 2001.

[TAR 72] TARJAN R., « Depth first search and linear graph algorithms », *SIAM Journal on Comput.*, vol. 1(2), p. 146–160, 1972.

[TOV 84] TOVEY C., « A simplified satisfiability problem », *Discrete Applied Mathematics*, vol. 8, p. 85–89, 1984.

[TRE 96] TREVISAN L., « Parallel Approximation Algorithms by Positive Linear Programming », *Proceedings of the 4th European Symposium on Algorithms*, LNCS 1136, Springer-Verlag, p. 62–75, 1996, Also published in Algorithmica, 21, 72–88, 1998.

Chapitre 2

Index