

基数制約の SAT 符号化を用いた MaxSAT ソルバーの試作

Building a MaxSAT Solver with SAT Encodings for Cardinality Constraints

越村 三幸*¹ 有村 寿高*²
Miyuki Koshimura Toshitaka Arimura

*¹九州大学大学院システム情報科学研究所 *²九州大学工学部
Faculty of Information Science and Electrical Engineering, Kyushu University School of Engineering, Kyushu University

We present a weighted partial MaxSAT solver QwMaxSAT which uses a normal SAT solver as an inference engine and CNF encodings of Boolean cardinality constraints. QwMaxSAT is a successor of QMaxSAT which was placed first in the partial MaxSAT (industrial) category from 2010 to 2012 Max-SAT Evaluations. The performance of QwMaxSAT heavily depends on that of its SAT solver and CNF encoding. We evaluate QwMaxSAT by solving MaxSAT instances taken from the 2013 Max-SAT Evaluation while change SAT solver and CNF encoding. In this evaluation, we use MiniSat 2.0/2.2.0 and glucose 2.0/3.0 as an inference engine, and four CNF encodings.

1. はじめに

命題論理の充足可能性判定問題 (SAT) は, 人工知能および計算機工学における基本的な問題で, NP 完全であることが最初に証明されたことでも知られている [4]. MaxSAT は SAT を最適化問題に拡張したもので, SAT が制約を満たす「解」を求めるのに対し, MaxSAT は制約を満たす「最適解」を求める [7, 11, 12].

本稿では, 重み付き部分 MaxSAT 問題を解くソルバー QwMaxSAT (Q-dai weighted partial MaxSAT ソルバー) の実装について述べる. QwMaxSAT は部分 MaxSAT ソルバー QMaxSAT[6] の後継で, 重み付きソフト節も扱えるように拡張されている. SAT ソルバーに基数制約を処理する機構を付加することにより実装されており, 基数制約の処理は SAT 符号化により実現されている.

このように実装されていることから, SAT ソルバーおよび基数制約の SAT 符号化が QwMaxSAT の性能を決する. 本研究は, SAT ソルバーおよび SAT 符号化の入れ替えが QwMaxSAT の性能にどのように影響するかを実験的に確かめ, 与えられた問題に対してどの符号化が適しているかを予想できるようにすることを目的とする.

2. QwMaxSAT

連言標準形 (以下, CNF:Conjunctive Normal Form) に対する重み付き部分 MaxSAT (以下, WPMS: Weighted Partial MaxSAT) 問題は, ハード節 (hard clause) と重み付きソフト節 (weighted soft clause) からなる節集合として与えられる. その目的は, 次のような命題変数への真偽値の割当を求めることである.

- 全てのハード節を充足する.
- 充足するソフト節の重みの総和が最大となる.

本稿では, 重み付きソフト節を通常の節 C と重み w (w は正整数) の対 (C, w) で表す.

ϕ をハード節の集合 H と重み付きソフト節の集合 S からなる WPMS 問題とする ($\phi = H \cup S$). また, S は n 個の

連絡先: 越村 三幸, 九州大学大学院システム情報科学研究所,
〒 819-0395 福岡市西区元岡 744, 092-802-3599

重み付きソフト節 (C_i, w_i) ($1 \leq i \leq n$) からなるものとする ($S = \{(C_1, w_1), \dots, (C_n, w_n)\}$). この時, n 個の新しい変数 b_i ($1 \leq i \leq n$)*¹を導入して, 新しい n 個の節 $C_i \vee b_i$ を作り, その集合を S^b と標記する ($S^b = \{C_1 \vee b_1, \dots, C_n \vee b_n\}$). H と S^b の節からなる節集合 $\phi^b (= H \cup S^b)$ のモデルの中で, $\sum_{i=1}^n w_i \cdot b_i$ を最小とするものが ϕ の WPMS 解となる.

ϕ^b は通常の節集合であり, 通常の SAT ソルバーでモデルを求めることができる. この性質を生かして, QwMaxSAT は SAT ソルバーを利用して WPMS 問題 ϕ を解く. その手順は次の通り.

まず, ϕ^b のモデル M_1 を SAT ソルバーを利用して求める. モデルがない場合は, ϕ の WPMS 解はない. モデル M_1 が見つかったら, M_1 の下で $\sum_{i=1}^n w_i \cdot b_i$ を計算する. その値を k_1 とする. 次に, $\sum_{i=1}^n w_i \cdot b_i$ を SAT 符号化し, それを $Card$ とする. $Card$ は, 変数 b_i ($1 \leq i \leq n$) と $\sum_{i=1}^n w_i \cdot b_i$ の結果を表す変数 s_i ($1 \leq i \leq m$) を含む節集合である.

次に求めたいモデルは, $\sum_{i=1}^n w_i \cdot b_i < k_1$ を満たすものなので, $Card$ にそのような制約を加える. その制約を SAT 符号化したものを $Card^{<k_1}$ と表す. $Card^{<k_1}$ は変数として s_i ($1 \leq i \leq m$) のみを含む節集合である. そして, $\phi^b \cup Card \cup Card^{<k_1}$ のモデル M_2 を SAT ソルバーを利用して求める. モデルがない場合, $\sum_{i=1}^n w_i \cdot b_i < k_1$ を満たすモデルがなかったことになるので, M_1 が WPMS 解である. モデル M_2 があった場合, M_2 の下で $\sum_{i=1}^n w_i \cdot b_i$ を計算し, その値を k_2 とする. そして, 制約の SAT 符号化 $Card^{<k_2}$ を作成し, $\phi^b \cup Card \cup Card^{<k_2}$ のモデル M_3 を SAT ソルバーを利用して求める.

以下同様にモデルが見つからなくなるまで繰り返す. $k_1 > k_2 > \dots > 0$ なので, この手続きは停止する. p (> 1) 回の SAT ソルバーの呼び出しの後に停止した場合, M_{p-1} が ϕ の WPMS 解である.

図 1 に QwMaxSAT の手続きを示す. 図中の solve(A) は SAT ソルバーの呼び出しを示す. 節集合 A にモデルがあれば, true を返すとともにモデルを M_j に設定する. モデルがなければ, false を返す.

*¹ b_i は阻止変数 (blocking variable) あるいは緩和変数 (relaxation variable) と呼ばれる.

- (1) $A = \phi^b$; // ϕ^b : 阻止変数を付加した問題
- (2) $j = 1$; // SAT ソルバーの呼び出しカウンタ
- (3) **while** (solve(A)) { // モデル M_j の探索
- (4) $k_j = \sum_{i=1}^n w_i \cdot b_i$; // モデル M_j の下での計算
- (5) **if** ($j == 1$) $A = A \cup \text{Card}$; // $\sum_{i=1}^n w_i \cdot b_i$ の SAT 符号化を A に加える
- (6) $A = A \cup \text{Card}^{<k_j}$; // 制約 $\sum_{i=1}^n w_i \cdot b_i < k_j$ の付加
- (7) $j = j + 1$;
- (8) }
- (9) **if** ($j > 1$) **return** M_{j-1} ;
- (10) **else return unsatisfiable**;

図 1: QwMaxSAT 手続き

3. 基数制約の SAT 符号化

n 個の命題変数 b_i ($1 \leq i \leq n$) について, $\sum_{i=1}^n b_i \# k$ なる制約を基数制約 (cardinality constraints) と呼ぶ. ここで, k は非負整数, 関係 $\#$ は $\{<, \leq, =, \geq, >\}$ のいずれかである. 本論文では, 各命題変数 b_i に正整数 w_i の係数をつけ, 関係を「 $<$ 」に限定した制約 $\sum_{i=1}^n w_i \cdot b_i < k$ を扱う*2.

現在, QwMaxSAT では基数制約の SAT 符号化が 4 つ組込んであり, 起動時のオプションで指定することができる*3.

- Warners による符号化法 [10]. w_i を二進数で符号化し, それぞれをトーナメント方式で二進加算して総和を求める機構を SAT 符号化している.
- Bailleux らによる符号化法で, 彼らは Totalizer と呼んでいる [3]. w_i を一進数で符号化, つまり w_i 個のビット列で符号化し, それぞれをトーナメント方式で一進加算して総和を求める機構を SAT 符号化している.
- Asín らによる符号化法で, 彼らは Cardinality Network と呼んでいる [1]. w_i を一進数で符号化し, トーナメント方式で併合ソートしていく機構を SAT 符号化している. ここで, 併合は奇数番目の要素の併合結果と偶数番目の要素の併合結果を, 下位の方から順次比較することにより行われる.
- Ogawa らによる符号化で, 彼らは Modulo Totalizer と呼んでいる [8]. w_i を正整数 p を基数とする modulo 数で符号化し, それぞれをトーナメント方式で modulo 加算して総和を求める機構を SAT 符号化している. Modulo の「商」と「剰余」の加算の符号化に, Bailleux らの符号化が用いられている.

表 1 は, 各符号化に要する命題変数と節の数のオーダを示している. w_{max} は w_i ($1 \leq i \leq n$) の最大値, w_{sum} は $\sum_{i=1}^n w_i$, k は図 1 における k_1 を表す. また, Ogawa のオーダは基数 $p = w_{sum}^{1/2}$ の時のものを示している.

QwMaxSAT がターゲットとする WPMS 問題には, ソフト節の重みが 32 ビット符号付き整数で表現できない程大きなものも存在する. そのような問題に現実的に対処できる符号化

*2 整数係数を導入しているため, この制約はむしろ, 線形擬似ブール制約 (linear pseudo-Boolean constraint) [9] の一種と呼んだ方がいいかもしれないが, 本稿では (広義の) 基数制約と呼ぶことにする

*3 Warners の符号化を除いて, それぞれのオリジナルの符号化では $w_i = 1$ ($1 \leq i \leq n$) の場合, つまり通常の基数制約の符号化を扱っている. ここでは, オリジナルを参考にして, w_i ($1 \leq i \leq n$) が正整数の場合の符号化を述べる.

表 1: 各符号化方式における変数と節の数のオーダ

符号化方式	変数の数	節の数
Warners	$O(n \cdot \log w_{max})$	$O(n \cdot \log w_{max})$
Bailleux	$O(w_{sum} \cdot \log k)$	$O(w_{sum} \cdot k)$
Asín	$O(w_{sum} \cdot \log^2 k)$	$O(w_{sum} \cdot \log^2 k)$
Ogawa	$O(w_{sum} \cdot \log w_{sum})$	$O(w_{sum}^{3/2})$

表 2: 半加算器の真偽表

入力		出力	
a	b	c	s
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

は, オーダの面では Warners のみである. そこで, Warners の符号化については, 幾つかの変種も試した. 本節の残りでは, それについて概説する.

上で述べたように, Warners の符号化は二進加算器の機構を SAT 符号化している. 二進加算器は, 半加算器と全加算器を組合わせて構成される. 紙面の都合上, ここでは半加算器のみの符号化を述べるが, 全加算器の符号化も同様である.

半加算器の真偽表を表 2 に示す. a と b が入力, s と c が出力で, s は和, c が桁上がり, である. 符号化では基本的に, 真偽表の 1 行につき, s 用の節と c 用の節の二つの節が作られる. 例えば, 1 行目からは, $a \vee b \vee \neg c$ と $a \vee b \vee \neg s$ が得られる. 前者は $(\neg a \wedge \neg b) \supset \neg c$ を, 後者は $(\neg a \wedge \neg b) \supset \neg s$ を節形式に書き換えたものである. ただ QwMaxSAT では, $\sum_{i=1}^n w_i \cdot b_i < k$ なる「 k 未満」という制約しか使わないので, このようにして得られる節のおおよそ半分は冗長*4である.

具体的には, 出力で 1 となる行のみから節を作れば十分である. 表 2 で 1 となるのは, c では最終行, s では 2 行目と 3 行目である. したがって, 半加算器の SAT 符号化は $\neg a \vee \neg b \vee c$, $a \vee \neg b \vee s$, $\neg a \vee b \vee s$ の三つの節の連言となる. このように出力が 1 となる組合せのみからの符号化を緩和方式と名付ける.

QwMaxSAT で扱う制約「 k 未満」を付加すると, 出力の c と s が共に 0 になることがある. そのような場合には, 入力

*4 ある節が冗長であるとは, その節がなくても正しい MaxSAT 解が得られることをいう.

表 3: 実験環境

CPU	Intel Xeon X5260 3.3GHz
メモリ容量	8GB
OS	Linux 2.6.32-5-amd64
コンパイラ	GCC 4.6.3
制限時間	30分 (1問あたり)

表 4: 評価に用いた例題 (2013 MaxSAT Evaluation より)

カテゴリ (略称)	問題数
部分 MaxSAT-Crafted (pms_crafted)	377
部分 MaxSAT-Industrial (pms_industrial)	627
重み付き部分 MaxSAT-Crafted (wpms_crafted)	340
重み付き部分 MaxSAT-Industrial (wpms_industrial)	396

の a と b は共に 0 でなければならないが, 緩和方式の節だけでは直接は分からない. 節の導出 (resolution) が必要となる. そこで, 「出力の c と s が共に 0 であれば, 入力 a と b は共に 0 になる」ことを表す節, $c \vee s \vee \neg a$ と $c \vee s \vee \neg b$ を, 緩和方式に加えた. これを方式 (A) と呼ぶことにする.

さらに, 「出力の c と s が共に 1 にはならない」ことを表す節 $\neg c \vee \neg s$ と, 「 c が 1 で s が 0 であれば, a と b は共に 1 でなければならない」ことを表す節 $\neg c \vee s \vee a$, $\neg c \vee s \vee b$ を方式 (A) に加えた. これを方式 (B) と呼ぶことにする.

4. 実験評価

QwMaxSAT で利用している SAT ソルバーと基数制約の SAT 符号化が性能にどの程度影響するかを評価するため, SAT ソルバーと符号化を入れ替えながら実験を行った. 利用した SAT ソルバーは MiniSat 2.0/2.2.0[5] と Glucose 2.0/3.0[2] でいずれも CDCL ソルバーである. Glucose は MiniSat 2.2.0 をベースとしているソルバーである. 表 3 に実験環境, 表 4 に評価に用いた例題を示す.

表 5 に実験結果を示す. 上から順に MiniSat2.0, MiniSat2.2.0, Glucose2.0, Glucose3.0 を推論エンジンとして用いた結果を示している. 表中の数字は 30 分の制限時間内に解けた問題数, 括弧内の数字は解けた問題の平均計算時間 (秒) を示している. なお, Ogawa の符号化の基数 p は $\lceil k^{1/2} \rceil$ とした. ここで, k は表 2 と同じである.

4.1 SAT 符号化の比較

部分 MaxSAT 問題 (pms_*) については Ogawa が SAT ソルバーによらず良い性能を示している. それに次ぐのが Bailleux である. Ogawa は Modulo の「商」と「剰余」部分に Bailleux の符号化を用いていることから, Bailleux による制約伝搬は QwMaxSAT の計算には適しているものと思われる.

一方, 重み付き部分 MaxSAT 問題 (wpms_*) については, Warners が良い性能を示している. これらの問題の重みの総

和は概ね大きいので, 符号化に要する節数の数が少なく済む Warners の性能が良かったと思われる. 他の符号化では, そもそも符号化中にメモリ不足になることが多かった. Warners (緩和), (A), (B) の三者を比較すると, 加算器の出力側からの値の伝搬を考慮した (A) と (B) の性能が考慮しない (緩和) より良かった.

Asín は, 平均的には良い性能を示さなかった. QwMaxSAT での利用には向いてないと思われる. Asín の併合ソートにおいて, 併合が奇数番目と偶数番目の併合に分けて行われるがこれがあまりよくない性能を招いたのでは, と考えており, より詳細な解析は今後の課題である.

以上より, ソフト節の重さの総和が小さい問題については Ogawa を利用し, 大きい問題については Warners(A) あるいは (B) の利用が望ましいと思われる.

4.2 SAT ソルバーの比較

Industrial 問題に対する性能は, MiniSat2.0, MiniSat2.2.0, Glucose2.0, Glucose3.0 の順に良くなっていく. SAT ソルバーのバージョン番号から分かるように, SAT ソルバーが新しくなるにつれて QwMaxSAT の性能も向上していることが分かる. 最新 SAT ソルバーの性能向上の恩恵を直接享受することを狙って実装された QwMaxSAT にとって, これは望ましい実験結果といえる. 特に, MiniSat2.0 と MiniSat2.2.0 の性能差, MiniSat2.2.0 と Glucose2.0 の性能差は顕著である.

Crafted 問題に対する性能は, Industrial 問題ほど顕著ではない. pms_crafted については, MiniSat2.0 を 2.2.0 に代えるといずれの SAT 符号化でも性能が劣化する. MiniSat2.2.0 から Glucose2.0 に代えると, 特に Warners (緩和) に対する性能向上が顕著であるが, Bailleux や Ogawa の性能を上回る程ではない. wpms_crafted については, Glucose3.0 の性能が一般的に良いように見える.

5. おわりに

WPMS ソルバー QwMaxSAT の性能は, 推論エンジンである SAT ソルバーと基数制約の SAT 符号化の性能によって決まる. 本論文では, 4 つの SAT ソルバーと 4 つの SAT 符号化を用いてそれらが QwMaxSAT の性能に及ぼす影響について定量的に調べた. Warners による符号化については, 冗長な節を加えた符号化も調べた.

SAT ソルバーについては, Glucose3.0 の性能が最も良かった. SAT 符号化については, ソフト節の重みの総和が小さいときには Ogawa の符号化が, 大きいときには Warners による符号化に冗長な節を加えたものが性能が良かった. 冗長な節の付加が性能向上をもたらしている結果は興味深く, さらなる探究が必要である. また実験結果を解析し, 問題にあった符号化を自動的に選択する機能を実装したいとも考えている.

謝辞 本研究は科研費 (25330085) の助成を受けたものです.

参考文献

- [1] Roberto Asín, Robert Nieuwenhuis, and Albert Oliveras. Cardinality Networks: a theoretical and empirical study. *Constraints*, 16:195–221, 2011.
- [2] Gilles Audemard and Laurent Simon. Predicting Learnt Clauses Quality in Modern SAT Solvers. In *Proceedings of 21st International Joint Conference on*

表 5: 実験結果

	(緩和)	Warners (A)	(B)	Bailleux	Asín	Ogawa
MiniSat2.0						
pms_crafted	245(314.33)	263(305.96)	272(288.03)	283(188.18)	278(295.13)	286(221.29)
pms_industrial	432(178.27)	457(176.08)	464(168.84)	453(117.56)	442(185.62)	486(130.22)
wpms_crafted	210(366.72)	232(346.84)	240(326.43)	36(1479.27)	67(1224.57)	203(311.77)
wpms_industrial	96(802.21)	162(496.72)	136(576.06)	44(1210.31)	46(1783.61)	79(801.13)
MiniSat2.2.0						
pms_crafted	236(287.81)	258(270.34)	269(241.47)	278(183.01)	261(278.29)	277(206.54)
pms_industrial	462(147.02)	502(138.94)	497(130.69)	496(102.57)	463(156.87)	529(108.15)
wpms_crafted	232(292.78)	244(285.86)	250(259.82)	46(1106.02)	66(1100.50)	222(257.71)
wpms_industrial	183(371.17)	248(281.24)	251(258.78)	49(1038.30)	50(1452.66)	91(628.70)
Glucose2.0						
pms_crafted	272(246.45)	262(242.35)	265(257.00)	278(167.79)	269(257.10)	281(145.67)
pms_industrial	505(132.74)	529(120.03)	526(129.48)	511(91.28)	493(140.28)	537(76.23)
wpms_crafted	277(242.00)	278(228.40)	279(244.10)	43(1084.75)	63(1097.76)	216(189.51)
wpms_industrial	240(279.31)	290(218.95)	293(232.44)	45(1036.54)	48(1440.80)	89(459.94)
Glucose3.0						
pms_crafted	278(229.86)	276(204.34)	277(209.47)	286(139.69)	279(223.76)	288(144.36)
pms_industrial	509(125.54)	533(105.81)	530(109.48)	511(78.18)	497(125.61)	537(77.42)
wpms_crafted	295(216.61)	295(191.18)	295(196.68)	40(998.79)	66(945.87)	218(190.71)
wpms_industrial	238(268.49)	302(186.75)	301(192.76)	47(850.04)	49(1274.03)	88(472.44)

解けた問題数 (平均計算時間: 秒)

Artificial Intelligence (IJCAI 2009), pages 399–404, 2009.

- [3] Olivier Bailleux and Yacine Boufkhad. Efficient CNF Encoding of Boolean Cardinality Constraints. In *Proceedings of 9th International Conference on Principles and Practice of Constraint Programming (CP 2003)*, pages 108–122, 2003.
- [4] Armin Biere, Marijn Heule, Hans van Maaren, and Toby Walsh, editors. *Handbook of Satisfiability*. IOS Press, 2009.
- [5] Niklas Eén and Niklas Sörensson. An Extensible SAT-solver. In *Proceedings of 6th International Conference on Theory and Applications of Satisfiability Testing (SAT 2003)*, pages 502–518, 2003.
- [6] Miyuki Koshimura, Tong Zhang, Hiroshi Fujita, and Ryuzo Hasegawa. QMaxSAT: A Partial Max-SAT Solver. *Journal on Satisfiability, Boolean Modeling and Computation*, 8:95–100, 2012.
- [7] Chu Min Li and Felip Manyà. *MaxSAT, Hard and Soft Constraints*, chapter 19, pages 613–631. In Biere et al. [4], 2009.
- [8] Toru Ogawa, YangYang Liu, Ryuzo Hasegawa, Miyuki Koshimura, and Hiroshi Fujita. Modulo Based CNF Encoding of Cardinality Constraints and Its Application to MaxSAT Solvers. In *Proceedings of IEEE 25th*

International Conference on Tools with Artificial Intelligence (ICTAI 2013), pages 9–17, 2013.

- [9] Olivier Roussel and Vasco Manquinho. *Pseudo-Boolean and Cardinality Constraints*, chapter 22, pages 695–733. In Biere et al. [4], 2009.
- [10] Joost P. Warners. A linear-time transformation of linear inequalities into conjunctive normal form. *Information Processing Letters*, 68:63–69, 1998.
- [11] 井上 克巳, 田村 直之. SAT ソルバーの基礎. 人工知能学会誌, 25(1):57–67, 2010.
- [12] 平山 勝敏, 横尾 真. *-SAT: SAT の拡張. 人工知能学会誌, 25(1):105–113, 2010.