Modulo 計算に基づく重み付部分 MaxSAT 問題の基数制約符号化手法の改良

Improvement in CNF Encording of Cardinal Constraints for Weighting Partial MaxSAT based on modulo calculation

有村 寿高 [1]　長谷川隆三 [2]　藤田博 [2] 上村直輝 [1]

Toshitaka Arimura [1]，Ryuzo Hasegawa [2]，Hiroshi Fujita[2]，Naoki Uemura[1]

[1] 九州大学大学院システム情報科学府
[1]Graduate School of Information Science and Electrical Engineering
[2] 九州大学大学院システム情報科学研究院
[2] Faculty of Information Science and Electrical Engineering

Abstract:
Weighted Partial MaxSAT(WPMS) is a generalization of Satisfiability problem.Manyoptimization problems can be reduced to WPMS in polynomial time. So it is important to develop MaxSAT solvers.Cardinality constraints plays important role in solving MaxSAT. In this paper, we propose Weighted Modulo Totalizer(WMTO).WMTO is based on Modulo Totalizer(MTO) and Weighted Modulo Totalizer(WTO) .WMTO use less variables and clauses than them.Our experimental results show the effectiveness of this methods.

1. はじめに

重み付部分 MaxSAT 問題（Weighted Partial MaxSAT Problem:WPMS 問題）とは，CNF 式の各節に重みを与えられており，充足する節の重みの総和の最大値を求める問題である[3][4][5]. 様々な最適化問題は多項式時間で WPMS 問題に帰着可能であるため，このような問題を解く MaxSAT ソルバーを高速化することは重要である．WMPS 問題の解法として，SAT ソルバーを利用する手法がある．しかし，SAT ソルバーは CNF 式しか解釈できないため重みの情報を CNF 式に符号化する必要がある．これを基数制約符号化と言い，符号化のサイズが大きければ SAT ソルバーの推論に要する時間も増大する．そのため，符号化を効率化することで MaxSAT ソルバーの改良を行うことができる．符号化方式の一つとして重みの部分和と生成する変数を一対一対応させることで符号化後のサイズを削減する手法がある（Weighted Totalizer：WTO）[9].

本研究では modulo 計算に基づく WTO の改良手法を提案する．また既存の手法との比較実験を行い，提案する手法の有効性について評価する．

2．提案手法

2．1 既存手法

WTO は Totalizer(TO)[6]を改良した符号化方式である．TO では重みを一進数表現，つまり重みの数だけ新たに変数を増やして重みを CNF 式に変換している．しかし，この方法ではソフト説の重みに大きく依存するため重みの総和が大きい問題ではメモリアウトしやすい．WTO では一進数表現ではなく，重みの部分和 1 つにつき 1 つの変数を生成する．これにより, TO で大量に生成されていた変数や節を省略している．

また，WTO とは別方式で TO を改良した Modulo Totalizer(MTO)[7]がある．MTO は重みをある定数で割った商と剰余について一進数で表現することで生成する変数や節を削減している．

他にも Warner らによる符号化方式(Warners)[8]もある．これまでの研究で WPMS 問題では Warners による符号化が一番性能が良く，TO，MTO，WTO は実装時にメモリアウトしやすい．しかし，メモリアウトしなかったものに関しては WTO が Warners による符号化よりも解答時間が早いことが分かっている．このことから WTO の生成する節や変数を削減

することで性能向上を期待される．

2．2Weighted Modulo Totalizer

本研究では前述の WTO と MTO を組み合わせることでより生成する節を削減する Weighted Modulo Totalizer（WMTO）を提案する．

WMTO は重みをある定数で割った商と剰余に分ける．これを WTO と同様に商に関する部分和と剰余に関する部分和で管理することで生成する節と変数を削減することができる．

図 1 は WMTO 本体の計算手続きである．基本的な構造は MTO と同様である．WMTO は割る定数を p とした時 n 個の変数からなる系列 $\{i_1,\dots,i_n\}$ を入力にとり，出力として変数系列 $\{h_1,\dots,h_m|_p r_1,\dots,r_{p-1}\}$ と節を生成する．入力を受けつけると WTO は入力変数を 2 つに分け，WMTO を再帰的に呼び出す．そしてその結果 $\{i_1,\dots,i_{n/2}\}$ から変数系列 $\{f_1,\dots,f_\alpha|_p a_1,\dots,a_{p1}\}$ と節 Φ_1 を生成し，$\{i_{n/2+1},\dots,i_n\}$ から変数系列 $\{g_1,\dots,g_\beta|_p b_1,\dots,b_{p2}\}$ と節 Φ_2 を生成する．その後加算器（WMUA）にそれらの変数系列を渡し変数系列 $\{h_1,\dots,h_m|_p r_1,\dots,r_{p-1}\}$ と商と剰余それぞれの節 Φ と Φ'' を生成する．最後に $\{h_1,\dots,h_m|_p r_1,\dots,r_{p-1}\}$ と $\Phi_1\wedge\Phi_2\wedge\Phi\ \wedge\Phi''$ を出力し，計算を終了する．部分和の種類だけ変数が生成されるため m は最小の時は n/p 個，最大で重みの総和 w を定数 p で割った w/p 個となり，p1 と p2 はともに最大で p-1 個となる．

図 2，3 は WMTO の加算器の部分の計算手続きである．サイズ α +p1 の変数系列 $\{f_1,\dots,f_\alpha|_p a_1,\dots,a_{p1}\}$ とサイズ β +p2 の変数系列 $\{g_1,\dots,g_\beta|_p b_1,\dots,b_{p2}\}$ を受け取る．受け取った変数に対応する重みの部分集合から，新たな部分集合を商と剰余それぞれで計算し，そのサイズだけ変数系列 $\{h_1,\dots,h_{[m]}|_p r_1,\dots,r_{p-1}\}$ を生成する．また商と剰余それぞれで変数に関する節集合 Φ と Φ'' を生成する．W，W"はそれぞれ商と剰余で重み σ に対応する変数番号を返す関数である．

f_α と g_β の重さをそれぞれ $W(f_\alpha)$，$W(g_\beta)$ とすると Φ の各節は f_α と g_β の重みの和が σ になることと剰余で桁上りが起きた時には重み和が σ +1 になることを表している．また a_{p1} と b_{p2} の重さをそれぞれ $W''(a_{p1})$，$W''(b_{p2})$ とすると Φ'' の各節は a_{p1} と b_{p2} の重みの和を定数 p で割った剰余が mod(σ)になることと σ が定数 p よりも大きいとき桁上りがあることを表している．

$$
\begin{aligned}
&WMTO(\langle i_1,\dots,i_n\rangle)=\\
&(\langle h_1\dots h_{[n/p]}|_p\, r_1\dots r_{p-1}\rangle,\Phi_1\wedge\Phi_2\wedge\Phi\wedge\Phi'')\\
&\quad WMTO(\langle i_1,\dots,i_{[n/2]}\rangle)=\langle f_1\dots f_{[\alpha]}|_p\, a_1\dots a_{p1}\rangle,\Phi_1)\\
&\quad WMTO(\langle i_{[n/2+1]},\dots,i_n\rangle)\\
&\qquad =(\langle g_1\dots g_{[\beta]}|_p\, b_1\dots b_{p2}\rangle,\Phi_2)\\
&\quad WMA\left(\left\langle f_1\dots f_{[\alpha]}|_p\, a_1\dots a_{p1}\right\rangle,\left\langle g_1\dots g_{[\beta]}|_p\, b_1\dots b_{p2}\right\rangle\right)\\
&\qquad =(\langle h_1\dots h_{[n/p]}|_p\, r_1\dots r_{p-1}\rangle,\Phi,\Phi'')
\end{aligned}
$$

図 1:WMTO の本体

$$
\begin{aligned}
&MUA\left(\left\langle f_1\dots f_{[\alpha]}|_p\, a_1\dots a_{p1}\right\rangle,\left\langle g_1\dots g_{[\beta]}|_p\, b_1\dots b_{p2}\right\rangle\right)\\
&\qquad =(\langle h_1\dots h_{[n/p]}|_p\, r_1\dots r_{p-1}\rangle,\Phi\ ,\Phi''\ )\\
&\Phi''\ =\bigwedge_{i=0}^{m}\bigwedge_{j=0}^{n} w(a_i)+w(b_j)=\sigma\\
&\qquad (\bar{a}_i\vee\bar{b}_i\vee r_{f(\sigma)}\vee c)\quad(\sigma<p)\\
&\qquad (\bar{a}_i\vee\bar{b}_i\vee r_{f(mod(\sigma,p))})\quad(\sigma<p)\\
&\qquad (\bar{a}_i\vee\bar{b}_i\vee c)\quad(\sigma<p)
\end{aligned}
$$

図 2:WMUA 加算器（剰余）

$$
\begin{aligned}
&MUA\left(\left\langle f_1\dots f_{[\alpha]}|_p\, a_1\dots a_{p1}\right\rangle,\left\langle g_1\dots g_{[\beta]}|_p\, b_1\dots b_{p2}\right\rangle\right)\\
&\qquad =(\langle h_1\dots h_{[n/p]}|_p\, r_1\dots r_{p-1}\rangle,\Phi\ ,\Phi''\ )\\
&\Phi\ =\bigwedge_{i=0}^{\alpha}\bigwedge_{j=0}^{\beta} w(f_i)+w(g_j)=\sigma\\
&\qquad (\bar{f}_i\vee\bar{g}_i\vee h_{F(\sigma)})\\
&\qquad (\bar{f}_i\vee\bar{g}_i\vee\bar{c}\vee h_{F(\sigma)})
\end{aligned}
$$

図 3:WMUA 加算器（商）

また WTO では一回目の推論で得られた暫定の非充足節の重み総和 k よりも大きな部分に関しては変数を作成せず節を作ることで生成する節と変数を削減している．

3．実験

3. 1 実験環境

本研究の実験は以下の環境で行った.

- CPU
  Intel core i7 CPU@2.70GHZ ×8
- 使用メモリ
  8Gib
- OS
  ubuntu13. 10
- 問題セット

  MaxSAT Evaluation 2015 から

  Crafted 部門 319 問
  Industrial 部門 610 問
- 制限時間
  1800 秒

3. 2 実験

WTO を改良した WMTO がどれくらいの性能か調べるために比較実験を行った. 比較対象は TO, MTO, WTO, Warners である. また WMTO は重みを割る定数の値を重み総和の平方根とする$WMTO_{square}$と, 推論で最初に導いた充足しない節の重み和上限の平方根とした $WMTO_{ksquare}$ の二種類の方式で実験を行った.

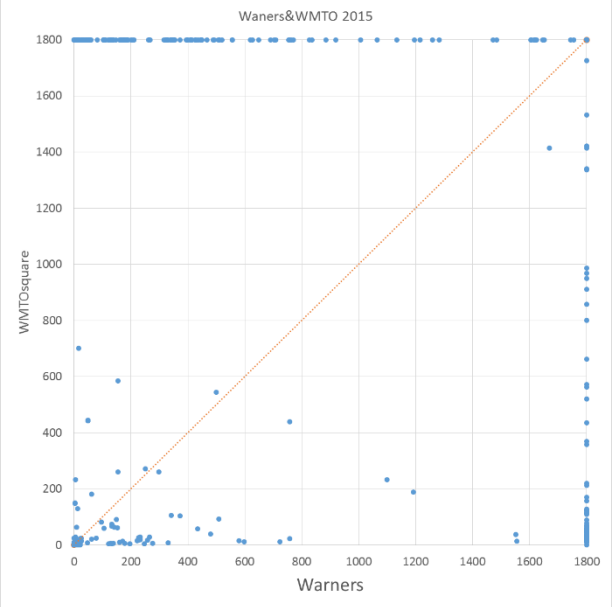
解答時間と平均時間を表 1 に示す. 数値の左の欄が解答数, 右の欄が平均解答時間 (単位:秒) である. 平均解答時間についてはメモリアウトしなかった問題での平均解答時間であり, タイムアウトを 1800 秒として計算した. WMTO は TO, WTO と比べるとどの部門においても解答数と改善した. MTO と比べると Crafted 部門では解答数で劣っているものの Industrial 部門では MTO よりも多くの問題を解いており, 平均解答時間もわずかに早くなった. WMTO の二つの方式を比較すると, $WMTO_{square}$ が $WMTO_{ksquare}$よりも解答数が多く, 平均解答時間も短かかった. WMTO は一度目の推論で求めた非充足節の重み総和 k よりも大きな重み和に関して変数を生成しないことで生成する節数を省略しているため, $WMTO_{ksquare}$が効果的だと考えられた. しかし, k の値は重み総和 W に比べ小さく, 商の部分での削減効率が悪くなったと考えられる.

Warners と比較すると, どの部門でも WMTO が解答数で劣っている. しかし, 平均解答時間を見るとはるかに WMTO が優れている. 図 4 は$WMTO_{square}$と Warners について比較したグラフである. グラフ上の 1 つの点が$WMTO_{square}$で解くことのできた 1 つの問題に対応しており, 横軸が Warners の解答時間, 縦軸が$WMTO_{spuare}$の解答時間であり, 点が対角線よりも右下にプロットされているほど$WMTO_{spuare}$が早い, ということである. グラフを見ると, 全体的に推論時間は改善されており, Warners では解くことの出来なかった問題 76 問を解くことができた. このことから, 問題によって Warners と WMTO を使い分けることでさらなる改善が期待できる.

表 2 は問題の重みの総和とその問題の解答数である. 表の値は左端の欄に示す値以下の重み総和を持つ問題の解答数を表している. 表 2 からは, 重みの総和が10^6以下の時は$WMTO_{square}$が WARN より多く問題を解いている. さらに10^8下の範囲でも10^7～10^8で解答数が負けているが合計では多くの問題を解いている. このことから10^8以下の範囲で$WMTO_{square}$が有効であると考えられる.

また, WMTO は重みの総和からの判断だけではなく問題の内容を分析することで効率化を期待できる. 割る定数の最適な値を見つけることが必要であり, どのような基準で定数を定めるべきか問題を分析する仕組みの開発が課題である.

図 4:Warners と WMTO の解答時間



4. おわりに

本研究では, 符号化手法の WTO に Modulo 計算を用いて改良し, WMTO を開発した. WTO の性能を大幅に改善することができ, Industrial 部門では MTO よりも多くの問題を解くことができた. Warners とは

表 1 : MaxSAT Evaluation 2015 WPMS 問題の解答数・平均解答時間

	Crafted		Industrial		Total	
	319	average(sec)	611	average(sec)	930	average(sec)
TO	40	1261.89	55	685.53	96	1027.48
MTO	118	956.38	98	579.94	217	816.99
WTO	95	871.49	160	312.40	256	584.35
WMTO_square	112	915.74	187	501.83	300	695.35
WMTO_ksquare	101	953.19	170	558.61	272	741.65
Warners	143	1060.52	244	1181.78	388	1140.19

表２:重みの総和と解答数

	合計	TO	MTO	WTO	WMTO_square	WMTO_ksquare	Warners
10	0	0	0	0	0	0	0
100	10	10	10	10	10	10	10
1000	18	17	17	17	17	17	17
10000	88	32	30	32	32	32	24
100000	95	23	64	44	65	65	62
1000000	133	8	55	39	70	67	66
10000000	149	5	31	15	59	63	91
100000000	118	0	9	9	45	17	18
1000000000	113	0	0	0	0	0	7
10000000000	42	0	0	0	0	0	11
100000000000	130	0	0	64	0	0	69
1000000000000	4	0	0	2	0	0	4
10000000000000	15	0	0	10	0	0	6
100000000000000	12	0	0	12	0	0	1
1000000000000000	1	0	0	1	0	0	1

解答数では劣っていたが平均解答実行時間では WMTO が優れており，WMTO が解けた問題を比較するとはるかに WMTO の方が早い時間で問題を解いていた．このことから WMTO で解くことのできる問題に関しては WMTO を用いるといったように適切に使い分けることで性能向上が期待できる．

また WPMS 問題の中には，ある重みがそれより小さい重みの総和より大きい可変進数 WPMS 問題などがある．このような特別な問題に対しては WMTO の割る値を総和の平方根ではなくそれぞれの重みとすることで桁上がりを無視することができ，さらなる節と変数の削減ができる．そのように問題を分析する機構や複数回 modulo 計算を行う機構の開発が課題である．

謝辞

本研究は科研究費（25330085，25330262）の助成を受けたものである．

参考文献

[１] 越村 三幸， 有村 寿高: 基数制約の SAT 符号化を用いた MaxSAT ソルバーの試作， 人工知能学会第 28 回全国大会， 1D4-OS-11a-4(2014)

[２] Miyuki Koshimura， Tong Zhang zhang， Hiroshi Fujita，Ryuzo Hasegawa． :QMaxSAT: A Partial Max-SAT Solversystem description． Journal on Satisfiability , Boolean Modeling and Computation 8 (2012) 95—100

[３] Chu Min Li and Felip Many`a. : MaxSAT， Hard and Soft Constraints， chapter 19， pages 613--631． In Biere et al. [4]， 2009.

[４] 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.

[５] 平山 勝敏，横尾 真．:*-SAT : SAT の拡張． 人工知能学会誌， 25(1):105--113， 2010.

[６] 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.

[７] 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.

[８] Joost P． Warners．:A linear-time transformation of linear inequalities into conjunctive normal form．Information Processing Letters， 68:63--69， 1998.

[９] 早田翔，長谷川隆三: 重み付 MaxSAT 問題における基数制約符号化手法の改良， 人工知能基本問題研究会資料， SIG-FPAI-B404-14