Image: section of the property of

制約を利用したSAT解法 - 九州大学における研究紹介

九州大学大学院 システム情報科学研究院 長谷川隆三 2013/7/24

アウトライン

- 1. Part1: モデル生成型SATソルバー
 - MiniMG
 - 学習節の分岐利用判定手法,削除戦略
- 2. Part2: MaxSATソルバー
 - QMaxSat
 - Modulo Totalizer
- 3. Part3: 未解決問題に挑戦
 - QG5
 - Ramsey Numbers

PART1: モデル生成型 SATソルバー

MGTPのモデル生成法

• モデル拡張
$$A_1 \wedge \cdots \wedge A_n \rightarrow B_1 \vee \cdots \vee B_m$$

$$M \models A_1 \wedge \cdots \wedge A_n \qquad M \not\models B_1 \vee \cdots \vee B_m$$

$$M \cup \{B_1\} \qquad \cdots \qquad | \qquad M \cup \{B_m\}$$

• モデル棄却 $A_1 \wedge \cdots \wedge A_n \rightarrow .$

$$M \models A_1 \land \cdots \land A_n$$

 \perp

• 単位反駁

単位簡約

Conflict節による枝刈り

C1: $\rightarrow a \lor b$.

C2: $\rightarrow c \lor d \lor e$.

C3: $\rightarrow p \lor q$.

C4: $p \rightarrow q$.

C5: $q \rightarrow p$.

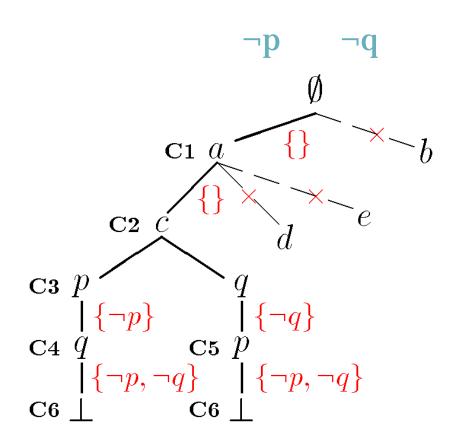
C6: $p \wedge q \rightarrow .$

• ¬p, ¬q

disjuct をlocal lemma として folding-up

• C1,C2

irrelevant な分岐を 刈り込む(証明濃縮)



分岐仮定による非極小モデルの棄却

$$S_{2} = \begin{cases} \rightarrow a \lor b \\ b \rightarrow a \end{cases}$$

$$\begin{bmatrix} a & b \\ [\neg b] & | \\ & 0 \\ M_{1} & \otimes \\ & M_{2} \end{cases}$$

(b) 無効な分岐仮定

 \Downarrow (Bry らの方法) モデル制約 $\{a \rightarrow\}$ を S_2 に付加

分岐補題による刈り込み

対称的分岐仮定の使用

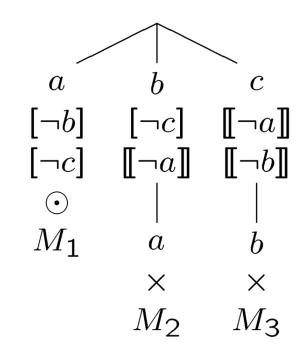
$$\begin{array}{c|c} A \lor B \\ \hline A & B \\ \llbracket \neg B \rrbracket & \llbracket \neg A \rrbracket \end{array}$$

$$\llbracket \neg A \rrbracket$$
: 分岐補題

Aの証明で、分岐仮定 $[\neg B]$ を使用していないとき生成.

$$\begin{bmatrix} a & b \\ [\neg b] & \llbracket \neg a \rrbracket \\ \odot & | \\ M_1 & a \\ \times \end{bmatrix}$$

$$S_3 = \left\{ \begin{array}{l} \rightarrow a \lor b \lor c \\ b \rightarrow a \\ c \rightarrow b \end{array} \right\}$$



MiniMGの概要

MiniMG

- □ モデル生成型定理証明系MGTPとMiniSatを組み合わせたSATソルバー
- □ MGTPの極小モデル生成機能 + MiniSatの枝刈能力(学習節等)の相乗効果を狙う
- 。命題特化のMGTP: Java版, C版

• 分岐仮定と選言バッファ

- 違反節の正選言によるn-way分岐 (MiniSatは2分岐)
- 正選言をto-be-givenリストとして保持(MiniSatは不要)

MiniMGの改良

- ◆正リテラルが問題により偏る ⇒ 性能低下 ← 学習節を分岐利用
- >分岐利用判定手法の提案
 - 出現数が少ない正リテラルを含む学習節
- 移動平均値と移動中間値(閾値の提案)
 - 。 直近のM個の学習節中の正リテラル出現数で閾値計算
 - 中間値:最大値と最小値の中間の値
- 移動区間 Mをどうするか?
 - 。固定值:100,1000,5000
 - · 非固定值:Restart間隔, 学習節削除(RDB)間隔

評価実験

- 実験環境
 - Linux, Intel Xeon X5260 3.3GHz, 8GB RAM
- ・ベンチマーク
 - SATCompetition2005、2007のIndustrial問題
 - SATRACE2006、2008の問題
 - □ 計303間
- 評価指標
 - □ 1問あたり制限時間900秒で解けた問題数

移動平均値を用いた分岐利用判定

移動区間	LBD値制限			
	3	4	5	
100	199(+2)	207(+10)	200(+3)	
1000	202(+5)	202(+5)	209(+12)	
5000	198(+1)	203(+6)	199(+2)	
Restart	202(+5)	209(+12)	198(+1)	
RDB	200(+3)	208(+11)	196(-1)	

表の数字 :解けた問題数

()の数字:提案手法未導入のMiniMGと比較した増減

青い数字 : 性能が低下 黒い数字 : 性能が向上

赤い数字 :特に性能向上(minisat2.0よりも良い結果)

移動中間を用いた分岐利用判定

移動区間	LBD値制限			
	3	4	5	
100	204(+7)	207(+10)	208(+11)	
1000	208(+11)	204(+7)	206(+9)	
5000	199(+2)	206(+9)	207(+10)	
Restart	203(+6)	207(+10)	204(+7)	
RDB	206(+9)	203(+6)	207(+10)	

表の数字 :解けた問題数

()の数字:提案手法未導入のMiniMGと比較した増減

青い数字 : 性能が低下 黒い数字 : 性能が向上

赤い数字 :特に性能向上(minisat2.0よりも良い結果)

分岐利用学習節の削除戦略

- **)** 残す度合い(戦略)を変化させ,影響を検証
 - □ 削除戦略評価のため、LBD制限はしない
- 1. 通常削除戦略
- 2. 積極的削除戦略
 - 。 含むすべて正リテラルの出現数が閾値以下なら残す
- 3. 平均的削除戦略
 - 。 含む正リテラルの出現数の平均が閾値以下なら残す
- 4. 消極的削除戦略
 - 。 一つでも閾値以下の正リテラルを含むなら残す
- 5. 非削除戦略
 - 。 分岐利用学習節は削除しない

移動平均値を用いた削除戦略

移動区間	通常削除	積極的削除	平均的削除	消極的削除	非削除
100	200(+3)	201(+4)	194(-3)	196(-1)	194(-3)
1000	203(+6)	208(+11)	194(-3)	191(-6)	195(-2)
5000	196(-1)	204(+7)	195(-2)	191(-6)	190(-7)
Restart	205(+8)	203(+6)	193(-4)	189(-8)	192(-5)
RDB	201(+4)	201(+4)	199(+2)	191(-6)	189(-8)

表の数字 :解けた問題数

()の数字:提案手法未導入のMiniMGと比較した増減

青い数字 : 性能が低下 黒い数字 : 性能が向上

赤い数字 :特に性能向上 (minisat2.0よりも良い結果)



移動中間値を用いた削除戦略

移動区間	通常削除	積極的削除	平均的削除	消極的削除	非削除
100	204(+7)	198(+1)	200(+3)	189(-8)	202(+5)
1000	199(+2)	200(+3)	200(+3)	191(-6)	201(+4)
5000	199(+2)	202(+5)	195(-2)	191(-6)	202(+5)
Restart	210(+13)	208(+11)	203(+6)	197(0)	195(-2)
RDB	206(+9)	202(+5)	193(-4)	190(-7)	197(0)

表の数字 :解けた問題数

()の数字:提案手法未導入のMiniMGと比較した増減

青い数字 : 性能が低下 黒い数字 : 性能が向上

赤い数字 :特に性能向上(minisat2.0よりも良い結果)

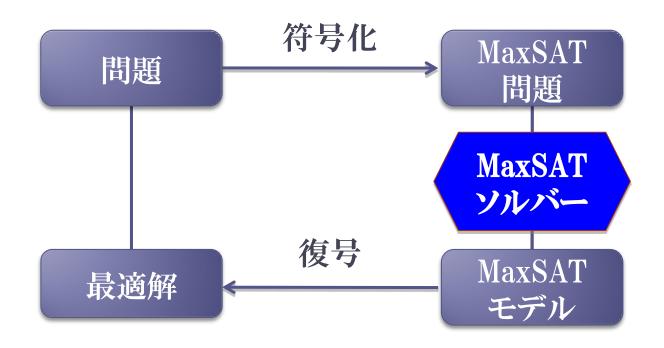


結果のまとめ

ソルバー	SAT	UNSAT	TOTAL
MiniMG-org	171	26	197
MiniSat2.0	170	36	206
MiniMG-ave-Restart-LBD4lim	179	30	209
MiniMG-ave-1000-LBD5lim	175	34	209
MiniMG-mid-Restart-orgDelete	178	32	210

PART 2: MAXSATソルバー

MaxSATを用いた問題解決



- 計画/ スケジューリング
- 論理回路設計、検証
- 最大クリーク,組合せ最適化,オークション

MaxSATソルバー

- 分枝限定法ベース: Clone, MaxSatz, IncMaxSatz
- SAT ベース:
 - 充足可能性ベース: SAT4J, ShinMaxSat, *QMaxSat*
 - 充足不能性ベース: msuncore, WPM1, PM2, pwbo

QMaxSAT: Q-dai MaxSAT Solver



MiniSat 2.0

MiniSat 2.2.0

Glucose 2.0

Bailleux50
Totalizer

Asin50
Half Sorting Networks

部分MaxSAT

- Hard節: 必ず充足すべき節
- Soft節: 必ずしも充足しなくてよい節
- 【目的】 全てのHard節を充足し、できるだけ多くの Soft節を充足するような変数割当を見つける
- 【出力】その変数割当で充足しているSoft節の数 (MaxSAT)
- 例 Hard: a \ b, c \ d Soft: ¬a, ¬b, ¬c, ¬d のMaxSAT解は2, モデルは{a,c},{a,d},{b,c},{b,d}.

基数制約とMaxSAT解

$$C_S = Hard \cup Soft$$

Hard =
$$\{H_1, ..., H_m\}$$
 Soft = $\{S_1, ..., S_n\}$

$$B = \{b_1, ..., b_n\}$$
 阻止変数(blocking variable)
$$C_S^B = Hard \cup Soft^B$$

$$Soft^B = \{S_1 \lor b_1, ..., S_n \lor b_n\}$$
基数制約

$$C_S^B = Hard \cup Soft^B$$

$$Soft^{B} = \{S_{1} \lor b_{1}, \dots, S_{n} \lor b_{n}\}$$

$$\min \left\{ k \mid C_{S}^{B} \cup CNF \left(\sum_{i=1}^{n} b_{i} \leq k \right), \ k \in \{0,1,\ldots,n\} \right\}$$

基数制約 $\sum_{i=1}^{n} b_i < k$ のSAT符号化

$$V_1$$
 V_2 \cdots V_n

$$C(b_1,\ldots,b_n,V_1,\ldots,V_n)$$

$$b_1$$

$$b_2$$

$$\cdots$$

$$b_n$$

もし m 個の b; が1に確定すれば,

V₁...V_mが1になる (左から m 個が1になる)

もしs個の biが0に確定すれば,

 $V_{n-s+1}...V_n$ が0になる (右から s 個が0になる)

$$\sum_{i=1}^{n} b_{i} < k \Leftrightarrow C(b_{1}, \dots, b_{n}, v_{1}, \dots, v_{k-1}, 0, \dots, 0)$$

K-Cardinality QMaxSat: v0.2

```
Solver S; bool first = true; C_S^B = Hard \cup Soft^B;
while (S.solve()) { // モデルM の発見
// M 中b<sub>i</sub>=1の個数kを計数する
                                      基数制約付加
 if (first) { first = false;
   C(b_1,...,b_n,V_1,...,V_{k-1},0,...,0)をC_s^Bに加える
                          k:現在の値
 } else {
   V_k = 0, V_{k+1} = 0, ..., V_{k'-1} = 0
                              k': 以前の値
最後のMを返す
```

Totalizer (Bailleux et al.)

$$TO(\langle i_1 \cdots i_n \rangle) = (\langle o_1 \cdots o_n \rangle, \Phi_1 \wedge \Phi_2 \wedge \Phi) \quad (n>1)$$

$$TO(\langle i_1 \cdots i_{\lfloor n/2 \rfloor} \rangle) = (\langle a_1 \cdots a_{\lfloor n/2 \rfloor} \rangle, \Phi_1)$$

$$TO(\langle i_{\lfloor n/2+1 \rfloor} \cdots i_n \rangle) = (\langle b_1 \cdots b_{n-\lfloor n/2 \rfloor} \rangle, \Phi_2)$$

$$UA (\langle a_1 \cdots a_{\lfloor n/2 \rfloor} \rangle, \langle b_1 \cdots b_{n-\lfloor n/2 \rfloor} \rangle) = (\langle o_1 \cdots o_n \rangle, \Phi)$$

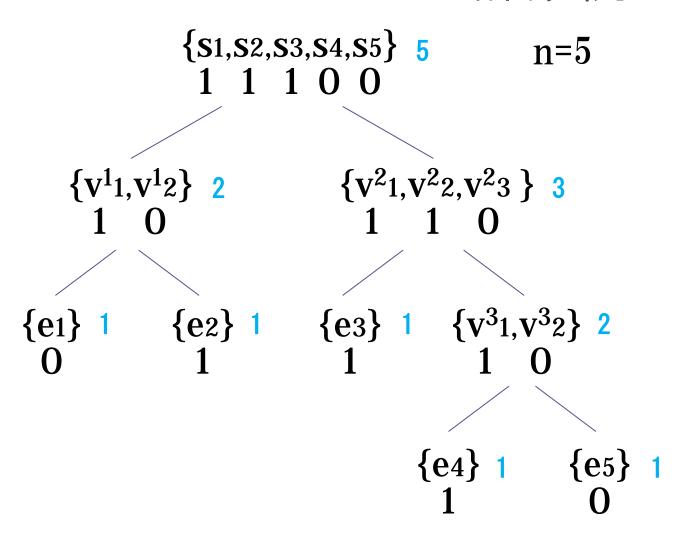
$$TO(\langle a_i \rangle) = \langle a_i \rangle \quad (n=1)$$

Unary Adder

$$UA(\langle \mathbf{a}_1 \cdots \mathbf{a}_m \rangle \langle \mathbf{b}_1 \cdots \mathbf{b}_n \rangle) = (\langle \mathbf{r}_1 \cdots \mathbf{r}_{m+n} \rangle, \Phi)$$

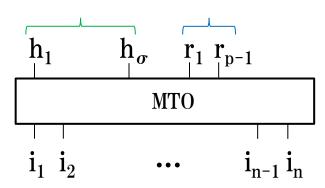
$$\Phi = \bigwedge_{i=0}^{m} \bigwedge_{j=0}^{n} (\bar{\mathbf{a}}_i \vee \bar{\mathbf{b}}_j \vee \mathbf{r}_{i+j}) \quad \mathbf{a}_0 = \mathbf{b}_0 = \mathbf{r}_0 = 1$$

Totalizerによる計数例



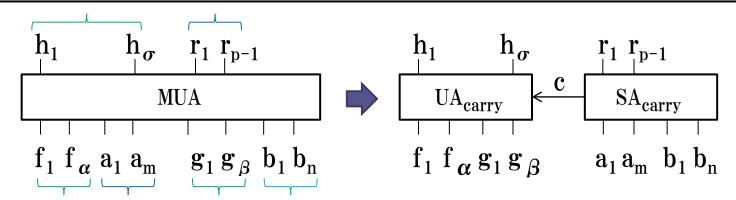
Modulo Totalizer(新提案)

$$\begin{split} & \operatorname{MTO}(\langle i_1 \cdots i_n \rangle, p) = (\langle h_1 \cdots h_{\lfloor n/p \rfloor} |_p \ r_1 \cdots r_{p-1} \rangle \rangle, \Phi_1 \wedge \Phi_2 \wedge \Phi) \quad (n \geqq p) \\ & \operatorname{MTO}(\langle i_1 \cdots i_{\lfloor n/2 \rfloor} \rangle, p) = (\langle f_1 \cdots f_{\alpha} |_p \ a_1 \cdots a_{p-1} \rangle \rangle, \Phi_1) \quad \alpha = \lfloor \lfloor n/2 \rfloor / p \rfloor \\ & \operatorname{MTO}(\langle i_{\lfloor n/2+1 \rfloor} \cdots i_n \rangle, p) = (\langle g_1 \cdots g_{\beta} |_p \ b_1 \cdots b_{p-1} \rangle \rangle, \Phi_2) \quad \beta = \lfloor \lceil n/2 \rceil / p \rfloor \\ & \operatorname{MUA}(\langle f_1 \cdots f_{\alpha} |_p \ a_1 \cdots a_{p-1} \rangle, \langle g_1 \cdots g_{\beta} |_p \ b_1 \cdots b_{p-1} \rangle) \\ & = (\langle h_1 \cdots h_{\lfloor n/p \rfloor} |_p \ r_1 \cdots r_{p-1} \rangle, \Phi) \\ & \operatorname{MTO}(\langle i_1 \cdots i_n \rangle, p) = \operatorname{TO}(\langle i_1 \cdots i_n \rangle) = \langle h_0 |_p \ r_1 \cdots r_n \rangle \quad (n \lessdot p) \quad h_0 = 0 \end{split}$$



Modulo Unary Adder (U:A)

$$\begin{split} \text{MUA}(\langle \mathbf{f}_{1} \cdots \mathbf{f}_{\alpha} \mid_{\mathbf{p}} \mathbf{a}_{1} \cdots \mathbf{a}_{\mathbf{m}} \rangle, \langle \mathbf{g}_{1} \cdots \mathbf{g}_{\beta} \mid_{\mathbf{p}} \mathbf{b}_{1} \cdots \mathbf{b}_{\mathbf{n}} \rangle) &= (\langle \mathbf{h}_{1} \cdots \mathbf{h}_{\sigma} \mid_{\mathbf{p}} \mathbf{r}_{1} \cdots \mathbf{r}_{\mathbf{p}-1} \rangle, \boldsymbol{\Phi}_{1} \wedge \boldsymbol{\Phi}_{2}) \\ \boldsymbol{\Phi}_{1} &= \bigwedge_{i=0}^{m} \bigwedge_{j=0}^{n} (\bar{\mathbf{a}}_{i} \vee \bar{\mathbf{b}}_{j} \vee \mathbf{r}_{i+j} \vee \mathbf{c}) \wedge (\bar{\mathbf{a}}_{i} \vee \bar{\mathbf{b}}_{j} \vee \mathbf{r}_{\mathsf{mod}\,(i+j,\mathbf{p})}) \wedge (\bar{\mathbf{a}}_{i} \vee \bar{\mathbf{b}}_{j} \vee \mathbf{c}) \\ &= \lim_{i=0}^{m} \sum_{j=0}^{n} (\bar{\mathbf{a}}_{i} \vee \bar{\mathbf{b}}_{j} \vee \mathbf{r}_{i+j} \vee \mathbf{c}) \wedge (\bar{\mathbf{a}}_{i} \vee \bar{\mathbf{b}}_{j} \vee \mathbf{r}_{\mathsf{mod}\,(i+j,\mathbf{p})}) \wedge (\bar{\mathbf{a}}_{i} \vee \bar{\mathbf{b}}_{j} \vee \mathbf{c}) \\ &= \lim_{i=0}^{m} \sum_{j=0}^{n} (\bar{\mathbf{a}}_{i} \vee \bar{\mathbf{b}}_{j} \vee \mathbf{r}_{i+j} \vee \mathbf{c}) \wedge (\bar{\mathbf{a}}_{i} \vee \bar{\mathbf{b}}_{j} \vee \mathbf{r}_{\mathsf{mod}\,(i+j,\mathbf{p})}) \wedge (\bar{\mathbf{a}}_{i} \vee \bar{\mathbf{b}}_{j} \vee \mathbf{c}) \\ &= \lim_{i=0}^{n} \sum_{j=0}^{n} (\bar{\mathbf{b}}_{i} \vee \bar{\mathbf{b}}_{j} \vee \mathbf{r}_{i+j} \vee \mathbf{c}) \wedge (\bar{\mathbf{a}}_{i} \vee \bar{\mathbf{b}}_{j} \vee \mathbf{r}_{\mathsf{mod}\,(i+j,\mathbf{p})}) \wedge (\bar{\mathbf{a}}_{i} \vee \bar{\mathbf{b}}_{j} \vee \mathbf{c}) \\ &= \lim_{i=0}^{n} \sum_{j=0}^{n} (\bar{\mathbf{b}}_{i} \vee \bar{\mathbf{b}}_{j} \vee \mathbf{r}_{\mathsf{mod}\,(i+j,\mathbf{p})}) \wedge (\bar{\mathbf{a}}_{i} \vee \bar{\mathbf{b}}_{j} \vee \bar{\mathbf{b}}_{i} \vee \bar{\mathbf{b}}_{j} \vee \bar{\mathbf{b}}_{i}) \wedge (\bar{\mathbf{b}}_{i} \vee \bar{\mathbf{b}}_{j} \vee \bar{\mathbf{b}}_{i} \vee \bar{\mathbf{b}}_{j} \vee \bar{\mathbf{b}}_{i}) \wedge (\bar{\mathbf{b}}_{i} \vee \bar{\mathbf{b}}_{i} \vee \bar{\mathbf{b}}_{i} \vee \bar{\mathbf{b}}_{i} \vee \bar{\mathbf{b}}_{i}) \wedge (\bar{\mathbf{b}}_{i} \vee \bar{\mathbf{b}}_{i} \vee \bar{\mathbf{b}}_{i} \vee \bar{\mathbf{b}}_{i} \vee \bar{\mathbf{b}}_{i} \vee \bar{\mathbf{b}}_{i}) \wedge (\bar{\mathbf{b}}_{i} \vee \bar{\mathbf{b}}_{i} \vee \bar{\mathbf{b}}_{i} \vee \bar{\mathbf{b}}_{i} \vee \bar{\mathbf{b}}_{i} \vee \bar{\mathbf{b}}_{i} \vee \bar{\mathbf{b}}_{i} \vee \bar{\mathbf{b}}_{i}) \wedge (\bar{\mathbf{b}}_{i} \vee \bar{\mathbf{b}}_{i} \vee \bar{\mathbf{b}}_{i} \vee \bar{\mathbf{b}}_{i} \vee \bar{\mathbf{b}}_{i} \vee \bar{\mathbf{b}}_{i} \vee \bar{\mathbf{b}}_{i}) \wedge (\bar{\mathbf{b}}_{i} \vee \bar{\mathbf{b}}_{i} \vee$$



Modulo Totalizerによる計数例

基数制約符号化方式のCNFサイズ比較

Encoding	#clauses	#aux. vars	decided
Totalizer v.0.1*	$O(n^2)$	$O(n\log n)$	unit prop.
K-Cardinality Totalizer v.0.2*	$O(n^{\bullet}k)$	$O(n\log k)$	unit prop.
Half Sorting Networks	$O(n\log^2 n)$	$O(n\log^2 n)$	unit prop.
K-Cardinality HS Networks	$O(n\log^2 k)$	$O(n\log^2 k)$	unit prop.
Pairwise Cardinality Network	$O(n\log^2 k)$	$O(n\log^2 k)$	unit prop.
Modulo Totalizer $(p=n^{1/2})$	$O(n^{3/2})$	O(nlogn)	unit prop.
Parallel binary counter	O(n)	O(<i>n</i>)	search
Warners	O(n)	O(<i>n</i>)	search

Totalizerを用いたQMaxSat

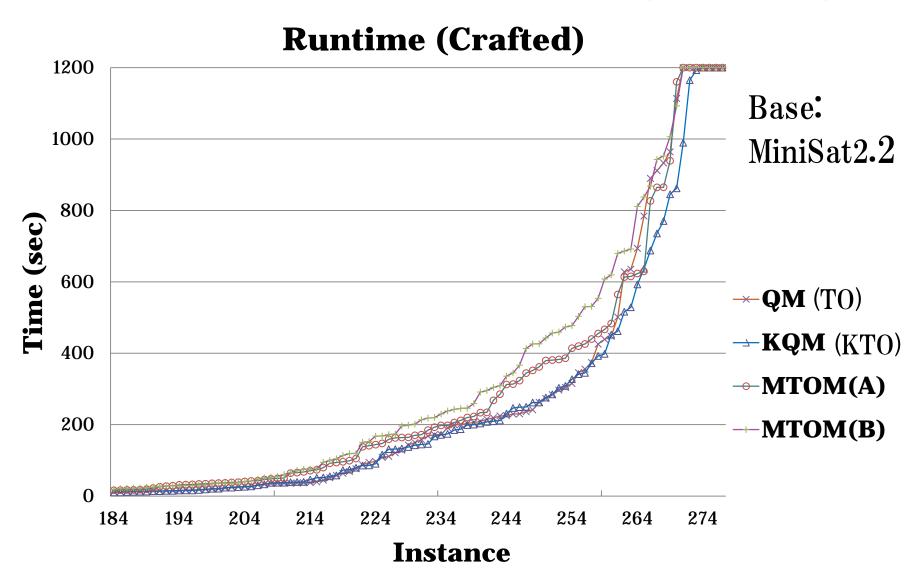
	Crafted	Industrial	
MiniSat2.0 + TO	280 (64.0) 2010,2011	379 (38.1) 2010	
MiniSat2.0 + KTO	281 (58.4)	385 (31.5) 2011	
MiniSat2.2.0 + KTO	274 (47.9)	402 (26.3)	
Glucose2.0 + KTO	275 (46.1)	410 (19.7) 2012	

Half Sorting Networksを用いたQMaxSat

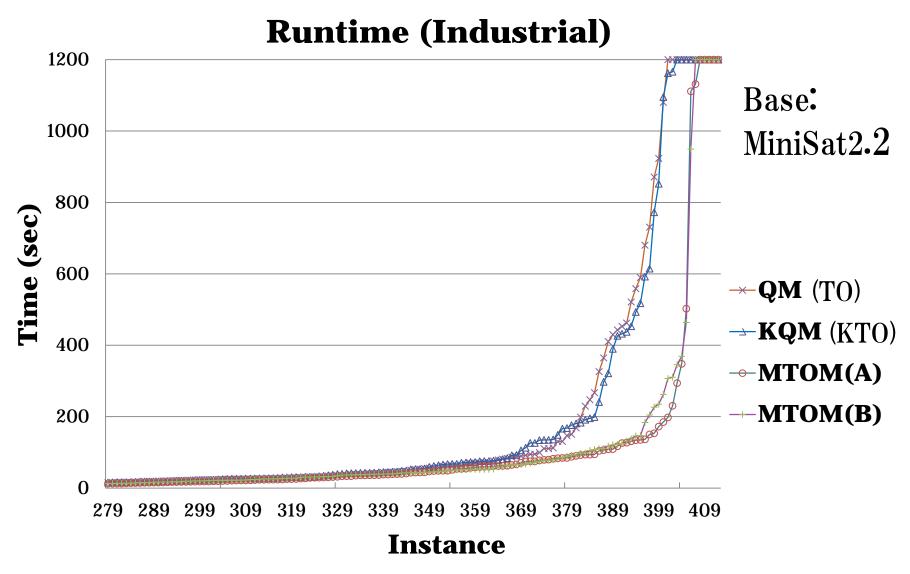
	Crafted	Industrial
MiniSat2.0 + HS	276 (64.7)	388 (38.6)
MiniSat2.0 + KHS	276 (59.3)	389 (35.0)
MiniSat2.2.0 + KHS	275 (70.1)	383 (27.8)
Glucose2.0 + KHS	270 (43.7)	408 (53.2)

解けたインスタンス数 (平均CPU時間:秒)

TOとMTOのMaxSAT性能比較(Crafted)



TOとMTOのMaxSAT性能比較(Industrial)



■

SINNminisat

- SAT Competition 2013
 - Core Solvers, Sequential, MiniSAT Hack-track, Application SAT+UNSAT
 - · 1位
- MiniSat2.2.0からの変更点
 - 。 True LBD [学習節評価値の向上]
 - ・LBDを最低の値として保持するのではなく、最新の値 を保持する。
 - ・LBDにてレベル0の変数ブロックを無視する
 - Agrgessive ReduceDB [重要な節だけ残す]
 - ・ 学習節の削除頻度を矛盾数の2乗に比例するように変更
- MiniSat2.2.0からの変更文字数:約635文字



ZENN

- SAT Competition 2013
 - Core Solvers, Sequential, Application SAT+UNSAT 3位
 - Core Solvers, Sequential, Application SAT 2位
- True LBD + Safe LBD [学習節の評価制度を更に向上]
 - ・ 評価値としてTrue LBDを利用し、加えて様々な種類の LBDから削除しない節を厳選する手法。
- Phase Shift [様々な問題に対応]
 - 。SAT,UNSATそれぞれの問題に強いリスタート戦略、節削除方法、アクティビティ上昇幅の操作などを組み合わせ実行する手法。
- 学習節の削除頻度はSINNminisatと同様

PART3: 未解決問題に挑戦



動機

- 高性能SATソルバーの力試し
 - MiniSat, Walksat, sharpSAT, SAT4J etc.
- より高機能・高能率なソルバーの開発
 - EPR provers (MGTP, CMGTP, MM-MGTP etc.)

- →未解決問題(大規模組合わせ問題)
 - QG(準群), Job shop scheduling, Ramsey数, etc.



未解決問題1:QG(準群の存在性)

N x N ラテン方陣

∧幂等: X * X = X

 $\land QG5$ 制約: ((Y * X) * Y) * Y = X

QG5_5 は存在→

より大きなNについて QG5_N は存在するか?

*	1	2	3	4	5
1	1	3	2	5	4
2	5	2	4	3	1
3	4	5	3	1	2
4	2	1	5	4	3
5	3	4	1	2	5

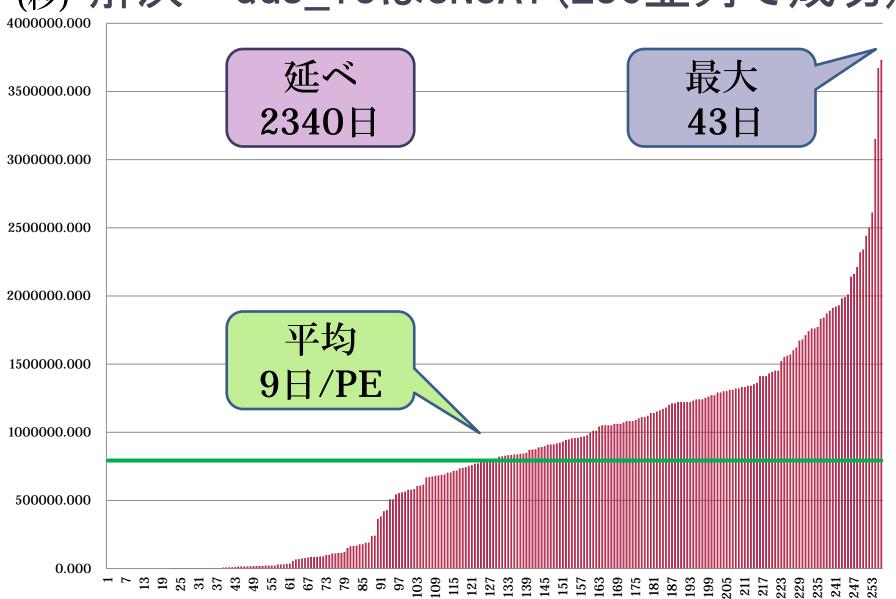


QG5_Nの従来記録

INT	#	#a]a	S/	Min	iSat2	Einst Danautad
N	#vars	#cls.	U	#conflicts	CPU time (s)	First Reported
9	729	28,540	U	17	0.015	H.Zhang (FALCON) 199x
10	1,000	43,636	U	47	0.031	M.Fujita (MGTP) 1993
11	1,331	64,054	S	40	0.031	
12	1,728	90,919	U	705	0.093	M.Fujita (MGTP) 1993
13	2,197	125,464	U	24,884	5.319	M.Stickel (DDPP) 1995
14	2,744	169,030	U	31,078	7.144	M.Stickel (DDPP) 1995
15	3,375	223,066	U	334,446	383.715	M.Stickel (DDPP) 1995
16	4,096	289,129	U	19,382,469	1,831,452.000	Y.Shirai,R.Hasegawa (CP,CMGTP) Oct. 1993,1995
16	4,096	289,129	U	12,983,551	54,280.600	H.Zhang (SATO) 1996
17	4,913	368,884	S	455,389,187	3,907,217.000	
18	5,832	464,104	U		43 days	H.Fujita,R.Hasegawa (MiniSat) Oct. 2009



(秒) 解決: QG5_18はUNSAT (256並列で成功)



手法: CMGTP/並列化

CMGTP

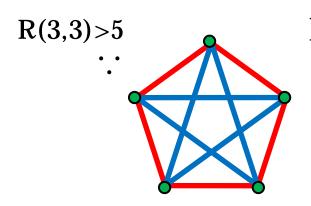
- □ ICOTプロジェクトの成果
- □ 制約MGTP: 対偶による負情報の伝搬を可能と する。DPの双方向伝搬に近づける。

• 並列化

- 。探索分割後、MiniSatによる個別探索
- ◦並列計算環境の利用(産総研検証施設)
 - · 112 * (Xeon 3.3GHz DualCore, 8 GB), etc.

未解決問題2:ラムゼー数

- ラムゼー数R(s, t)とは,以下を満たす 完全グラフの頂点数nの最小値
 - □ 2色でどのように辺彩色しても、必ず
 - 。第1色のみのクリークKs, あるいは,
 - 。第2色のみのクリークKtが存在



R(3,3) ≤ 6

R(3,3)=6



ラムゼー数の従来記録

ラムゼー数	記録保持者 発表年
R(3, 3) = 6	Greenwood and Gleason 1955
R(3, 9) = 35	Grinstead and Roberts 1982
$R(3, 10) \ge 40$	Exoo 1989c, Radziszowski and Kreher 1988
R(4, 5) = 25	McKay and Radziszowski 1995
$R(4, 6) \ge 37$	McKay and Radziszowski 1995, Exoo 2012
$R(4, 7) \geq 49$	Exoo 1989a, Mackey 1994
$R(4, 8) \geq 56$	Exoo 1998, Exoo 2002
$R(5, 5) \geq 43$	Exoo 1989b, McKay and Radziszowski 1995
$R(6, 6) \ge 102$	Kalbfleisch 1965, Mackey 1994

ツール:SCSat

(A Soft Constraint Guided SAT Solver)

- ・ソフト制約(ソフト節集合)
 - 充足しなくてもよい(最大充足も要請しない)
 - 。探索空間の削減が主目的
 - 。好ましい解に向けて探索を誘導
- ・ソフト節が原因でのコンフリクトは無視
 - 。ただし,原因ソフト節に反則点を付与.
- ・ソフト制約の段階的緩和
 - □ 求解の失敗(UNSAT)時,反則点の大きいソフト節たちの中から一部を削除し,探索をリスタート

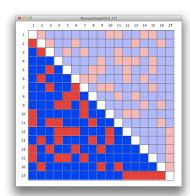
See Proc. SAT2013, LNCS 7962, 2013, pp.415-421



R(3,6,17)の各種ソフト制約下探索

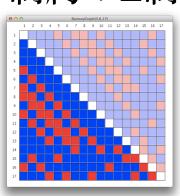
無制約

R(3,6,17) は0.01秒 で解ける

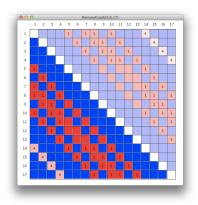


U制約+Z制約

R(3,5,13) の解を表す unitを付加 + Z制約緩和 で解ける

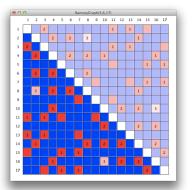


Z制約



e_ij ≡ e_kl (i<j, k<l) s.t. j-l=l-k の制約下では未解

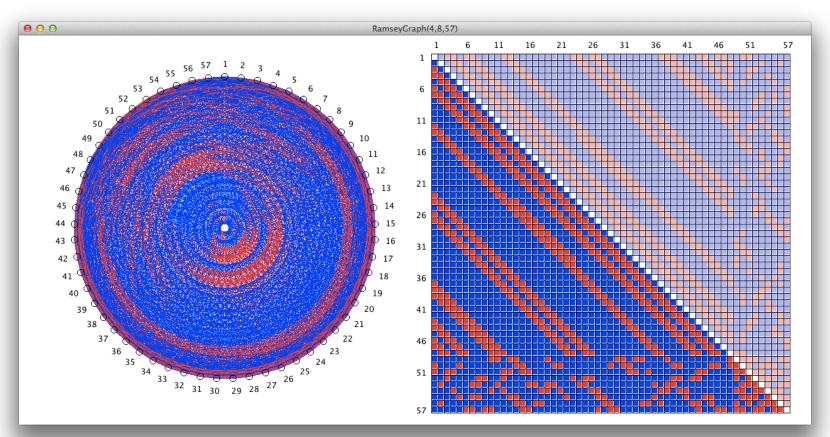
市松制約



checker board制約 下では未解

闡

新発見: RamseyGraph (4, 8, 57) - R(4, 8) の下界を58に更新 -



R(4,7,48)の解を利用

See arXiv:1212.1328, 2012