特　集

∗-SAT: SATの拡張

∗-SAT: Extentions of SAT

平山 勝敏　神戸大学大学院海事科学研究科
Katsutoshi Hirayama　Graduate School of Maritime Sciences, Kobe University
hirayama@maritime.kobe-u.ac.jp, http://www.edu.kobe-u.ac.jp/fmsc-hrymlab/

横尾 真　九州大学大学院システム情報科学研究院
Makoto Yokoo　Graduate School of Information Science and Electrical Engineering, Kyushu University
yokoo@is.kyushu-u.ac.jp, http://lang.is.kyushu-u.ac.jp/˜yokoo/

keywords: MaxSAT, QSAT, parallel SAT, distributed SAT

1. はじめに

SAT ソルバーの研究が近年大きく発展したことを受けて，SAT の枠組みを多方面に拡張する試みが盛んになりつつある．本稿では，SAT の定式化や解法を拡張する試み（本稿では総称して ∗-SAT とよぶ）を概観する．

まず，SAT を最適化問題に拡張した MaxSAT を紹介する．MaxSAT は理論計算機科学の分野でも古くから研究されているが，SAT ソルバーの技法を利用した新しい厳密解法がここ数年で相次いで提案された．本稿では，それらの最新の厳密解法の概要を述べる．

次に，SAT に限量記号を導入した QSAT（Quantified SAT）の概要を述べる．通常の SAT ではすべての命題変数が存在限量化されていると解釈できるが，QSAT では新たに全称限量化された命題変数が導入される．全称限量化された命題変数は，意思決定者が制御できない敵対者や自然の選択を表し，これらの命題変数の値に関わらず制約を満足する解が存在するかどうか判定することが求められる．

また，SAT の解法を拡張する試みとして並列 SAT（parallel SAT），および，分散環境で SAT を解く分散 SAT（distributed SAT，または，DisSAT）の概要を述べる．並列 SAT とは，マルチコア CPU や PC クラスタなどの並列処理を実現するハードウエアを利用して SAT を高速に解く解法であり，近年そのような環境が比較的容易に構築できるようになったことから，最近注目されるようになった．一方，分散 SAT とは，問題の構成要素が複数の自律的なエージェントに分散されている SAT である．これを解くには，一般に集中制御のないピアツーピア型のエージェント間通信プロトコル（分散アルゴリズム）が必要になる．

以下，2 章で MaxSAT，3 章で QSAT，4 章で並列 SAT，5 章で分散 SAT に関して，それぞれの定式化と代表的なアルゴリズムの概要を述べ，6 章で本稿をまとめる．

2. MaxSAT

SAT の問題例が充足不能の場合，どの程度充足不能なのか知りたいというのは自然な欲求だと思われる．この欲求に答えるべく SAT を拡張したものが MaxSAT である．

MaxSAT については，主に理論計算機科学の分野で，近似精度保証のある近似解法に関する理論的な研究 [Karloff 97] が活発に行われてきたが，最適解を求める保証のある厳密解法を追求した研究は少なく，大規模な問題例に対して厳密解法を適用することは実質的に不可能であった．

しかし，ここ数年の間に，SAT ソルバーの技法を用いて MaxSAT の厳密解法を高速化する試みが多数提案されている．さらに，厳密解法の性能を競う MaxSAT evaluations という競技会 [Argelich 08, Heras 08a] が 2006 年から毎年開催されるようになり，ベンチマーク問題例が徐々に整備されるとともに，競技会に参加した解法の実験結果が公開されるようになった．このような背景のもとに，現時点では，数百変数程度の問題例がある程度現実的な時間で厳密に解けるようになりつつある．

本章では，MaxSAT の厳密解法に関する最近の話題を取り上げる．

2·1 定式化／例題

入力は，各節に非負の重みが定義された CNF 形式の命題論理式（CNF 式）であり，目的は，違反となる節の重み和を最小にする（満たされる節の重み和を最大にする）ような命題変数への真偽値の割り当てを求めることである．

各節の重みがすべて 1 である問題は**重みなし** MaxSAT（unweighted MaxSAT)，一方，1 以外の重みをもつ節を含む問題は**重みつき** MaxSAT（weighted MaxSAT）とよばれる．また，必ず満たさなければならない節には特別な重み’⊤’ を設定し，特に**ハード節**（hard clause）とよぶ．一方，場合によっては満たさなくてもよい節には非負の重み

を与え，特に**ソフト節**（soft clause）とよぶ．一般に，ハード節とソフト節の両方を含む問題は**部分** MaxSAT（partial MaxSAT）とよばれ，そのうち，すべてのソフト節の重みが 1 であるものは**重みなし部分** MaxSAT（unweighted partial MaxSAT），1 以外の重みをもつソフト節を含むものは**重みつき部分**（weighted partial MaxSAT）とよばれる．なお，SAT の場合と同様，各節が高々k 個のリテラルからなることを明示したい場合には Max$k$SAT と表記する．

MaxSAT は典型的な NP 困難問題である．SAT では，2SAT は P，3SAT が NP 完全なのに対し，MaxSAT では，重みなし Max2SAT でさえ NP 困難である．

前述の MaxSAT evaluations には多くのベンチマーク問題例が投稿されており，最新の MaxSAT evaluation 2009 では，重みなし MaxSAT 738 例，重みつき MaxSAT 309 例，重みなし部分 MaxSAT 1500 例，重みつき部分 MaxSAT 932 例が使用された．それらはランダムに生成された問題例だけでなく，パズルやグラフの問題例（最大カット，最小頂点カバー，最大クリークなど），実用的な問題例（スケジューリング，組合せオークション，回路設計，バイオインフォマティクスの問題など）も含んでいる．以下では，そのうちのいくつかについて MaxSAT への符号化法を述べる．詳しくは文献 [Heras 08a] 等を参照されたい．

§1 最大カット問題

頂点集合を V，枝集合を E とするグラフ $G=(V,E)$ に対し，頂点集合 V をある集合 U とその補集合 $V\setminus U$ に分けるとする．その際，$v_i\in U$ かつ $v_j\in V\setminus U$ となるような枝 (v_i,v_j) の集合が一意に決まるが，そのような枝の集合をカット（cut）といい，その集合のサイズをカットのサイズという．最大カット問題とは，与えられたグラフに対し，カットのサイズが最大となる頂点集合 U を求める問題である．

この問題は次のようにして重みなし MaxSAT に符号化できる．まず，グラフの各頂点 v_i に 1 つの命題変数 x_i を定義し，x_i が真のときは $v_i\in U$，偽のときは $v_i\in V\setminus U$ とする．次に，グラフの各枝ができるだけカットに含まれるようにするために各枝 (v_i,v_j) に対して重みが 1 となる 2 つの節 $(x_i\vee x_j,1)$ と $(\neg x_i\vee\neg x_j,1)$ を定義する．この重みなし MaxSAT を解けば最大カット問題の解が得られる．

§2 最小頂点カバー問題と最大クリーク問題

グラフ $G=(V,E)$ に対し，任意の枝 $(v_i,v_j)\in E$ について $v_i\in U$ または $v_j\in U$ を満たす頂点集合 $U\subseteq V$ をグラフ G の頂点カバー（vertex covering）といい，その集合のサイズを頂点カバーのサイズという．最小頂点カバー問題とは，与えられたグラフに対し，サイズが最小となる頂点カバーを求める問題である．

この問題は次のようにして重みなし部分 MaxSAT に符号化できる．まず，グラフの各頂点 v_i に 1 つの命題変数 x_i を定義し，x_i が真のときは $v_i\in U$，偽のときは $v_i\in V\setminus U$ とする．次に，グラフの各枝の少なくとも一方の頂点が頂点カバーに含まれるようにするために各枝 (v_i,v_j) に対してハード節 $(x_i\vee x_j,\top)$ を定義し，また，頂点カバーのサイズを最小化するために各頂点 v_i に対してソフト節 $(\neg x_i,1)$ を定義する．この重みなし部分 MaxSAT を解けば最小頂点カバー問題の解が得られる．

一方，グラフ G に対して，その部分グラフで完全な（部分グラフ内のすべての 2 頂点のペア間に枝が存在する）ものをクリーク（clique）といい，その頂点数をクリークのサイズという．最大クリーク問題とは，与えられたグラフに対し，サイズが最大となるクリークを求める問題である．グラフ G に対する最大クリーク問題は，G の補グラフ $\bar{G}$ に対する最小頂点カバー問題と等価である．従って，上と同様な方法で，最大クリーク問題も重みなし部分 MaxSAT に符号化できる．

§3 組合せオークション

組合せオークションとは異なる種類の財が同時に販売されるオークションで，一般に，財の集合 G，および，入札の集合 B よりなる．各入札 $b_i\in B$ は，G の部分集合 $G_i\subseteq G$ と入札額 u_i のペアであり，G_i で指定される財一式を金額 u_i で購入するという入札者 i の意思を表す．オークション主催者は，入札の集合 B より一般に複数の入札を選んで落札集合とする．その際，落札集合内の任意の 2 つの入札が同一の財を選択しないという条件を満たしつつ，落札額の合計を最大にする．

このオークション主催者の問題は次のように重みつき部分 MaxSAT に符号化できる．まず，各入札 b_i につき 1 つの命題変数 x_i を定義し，x_i が真かつそのときのみ，その入札が落札集合に含まれるものとする．次に，各入札 b_i に対して，それが落札集合に含まれれば u_i だけ増益となることを意味するソフト節 (x_i,u_i) を定義する．さらに，落札集合内の任意の 2 つの入札が同一の財を選択しないよう，$G_i\cap G_j\neq\emptyset$ なる任意の 2 つの入札 b_i と b_j に対して $(\neg x_i\vee\neg x_j,\top)$ というハード節を定義する．この重みつき部分 MaxSAT を解けば，この問題に対する解が得られる．

2·2 アルゴリズム

ここ数年，MaxSAT の厳密解法を高速化するための手法が相次いで提案されており，現時点では，数百変数程度の問題例がある程度現実的な時間で厳密に解けるようになりつつある．最近の厳密解法は，その基本的な特徴から，SAT ソルバーを用いた解法と分枝限定法に基づく解法（branch-and-bound solver）に分けることができる．以下では，それらの概要を述べる．

§1 SAT ソルバーを用いた解法

この解法の特徴は，任意の系統的 SAT ソルバーを繰り返し適用して MaxSAT の最適解を求めることである．その基本的なアイデアを重みなし MaxSAT を用いて説明

する．

重みなし MaxSAT の節集合を $C = \{C_1, C_2, \ldots, C_m\}$ として，各節 C_i に新しい命題変数 b_i を用意し，$C' = \{C_1 \vee b_1, C_2 \vee b_2, \ldots, C_m \vee b_m\}$ という節集合を作る．この命題変数 b_i は**阻止変数**（blocking variable）とよばれる．容易にわかる通り，C' において阻止変数 b_i を真（1）にすれば，C_i を満たさなくとも C' 全体は充足可能である．よって，もとの節集合 C に対する重みなし MaxSAT は,「C' を充足しつつ真（1）となる阻止変数の数を最小にせよ」という最適化問題と等価である．形式的には,

$$\min\{k \mid C', \sum_{i=1}^{m} b_i \leq k, \, k \in \{0, 1, \ldots, m\}\}$$

を解くことになる．この 2 つ目の制約条件 $\sum_{i=1}^{m} b_i \leq k$ は基数制約（cardinality constraint）と呼ばれ，これを CNF 式として符号化する方法がいくつか提案されている [Bailleux 03, Sinz 05]．よって，$\mathrm{CNF}(\cdot)$ を CNF 式を返す任意の符号化法とし,

$$\min\{k \mid C' \cup \mathrm{CNF}(\sum_{i=1}^{m} b_i \leq k), \, k \in \{0, 1, \ldots, m\}\} \quad (1)$$

を解けばよい．この 1 つ目の制約条件 $C' \cup \mathrm{CNF}(\sum_{i=1}^{m} b_i \leq k)$ は，k の値を固定すれば通常の SAT となる．

SAT4Jmaxsat [Berre 09] は，式 (1) の最適化問題を解くにあたり，k の値を次第に減少させながら対応する $C' \cup \mathrm{CNF}(\sum_{i=1}^{m} b_i \leq k)$ を系統的 SAT ソルバーで繰り返し解く解法である．すなわち，適当な上界値より出発し，ある k に対する $C' \cup \mathrm{CNF}(\sum_{i=1}^{m} b_i \leq k)$ を系統的 SAT ソルバーで解き，それが充足可能な限り k の値を下げ続ける．この過程で SAT が充足不能になった場合には一つ前の充足可能な SAT の解を最適解とする．SAT4Jmaxsat は，ソフト節にのみ阻止変数を追加することで重みなし部分 MaxSAT にも適用できる．さらに，先の基数制約の部分を疑似ブール制約（pseudo boolean constraint）に置き換え，その CNF 式への符号化法 [Eén 06] を利用することにより重みつき部分 MaxSAT にも対応できる．

上の方法では，すべてのソフト節にあらかじめ阻止変数を用意するので，ソフト節の数が多くなると繰り返しの各回における SAT の探索空間が非常に大きくなる．Fu と Malik は，阻止変数をあらかじめ用意するのではなく必要に応じて追加することでこの問題に対処している [Fu 06]．彼らの解法では，阻止変数のない元の問題から出発し，SAT が充足不能だった場合にその原因となった節の集合— **充足不能コア**（unsatisfiable core）—を特定し，その中のソフト節にのみ阻止変数を追加する．

Fu と Malik の解法では，一つの節に複数の阻止変数が追加されることがあるが，阻止変数の役割を考えるとこれは無駄である．そこで，充足不能コアに阻止変数を追加する際，各節につき追加される阻止変数の数が高々一つとなるよう改良することが提案されている [Marques-Silva 08]．また，Fu と Malik の解法は重みつきの問題を解くことができない．そこで，彼らの解法を拡張し，重みつき部分 MaxSAT まで扱えるようにすることも提案されている [Ansótegui 09].

§2　分枝限定法に基づく解法

分枝限定法は，組合せ最適化問題を解く汎用的な厳密解法の一つである．目的関数値の最小化を目指す組合せ最適化問題に対して，分枝限定法では次の分枝操作（branching operation）と限定操作（bounding operation）により，最適解が存在する空間を絞り込む．

分枝操作　探索空間を複数の部分探索空間に再帰的に分割する．その際，各部分探索空間における目的関数値の最小値の下界値を何らかの方法で推定する．

限定操作　事前，あるいは，探索中に得た最適解の候補（暫定解）を用いて最適解の目的関数値（最適値）の上界値を計算する．分枝操作で生成されたある部分探索空間における下界値がこの上界値以上であれば，その部分探索空間に最適解は存在しないため，その空間の探索を打ち切る．

分枝限定法の基本部分は，最新の系統的 SAT ソルバーの原型となった DPLL アルゴリズム [Davis 62] と概略同じである．従って，最新の系統的 SAT ソルバーと同様の技法を使って，MaxSAT の分枝限定法を高速化しようというのは自然なアイデアと言える．事実，最近の MaxSAT evaluations に参加しているほとんどのソルバーは分枝限定法に基づくものであり，しかも，その高速化には，MaxSAT の節集合を等価でより簡略化されたものに置き換えるための推論規則が重要な役割を果たす [Argelich 08]．ここでは，そのような推論規則を使った分枝限定法を 2 つ紹介する．

MaxSatz [Li 07] は，MaxSAT evaluation 2006 の重みなし MaxSAT 部門で 1 位になった分枝限定法に基づく解法である．MaxSatz の主要な特徴の一つは，DPLL アルゴリズムの単位伝播（unit propagation）に相当する 6 つの推論規則を導入したことである．これらの推論規則は，分枝限定法の各探索ノードにおける CNF 式を，重みなし MaxSAT として等価で空節 □（empty clause）を含む CNF 式に置き換える．ここで，重みなし MaxSAT として等価とは違反となる節の数が任意のモデルで同じになることをいう．例えば，文献 [Li 07] で提案された推論規則 3 は，節集合 $C = \{x_1, x_2, \neg x_1 \vee \neg x_2, \ldots\}$ を $C' = \{\Box, x_1 \vee x_2, \ldots\}$ に置き換える．これらの推論規則は空節を顕在化させるため，分枝限定法における下界値計算に非常に有効である．また，重みなし MaxSAT として等価な CNF 式に置き換えるため，置き換えた後の式を探索木の子ノードにそのまま引き継げるというメリットもある．

MaxSatz はその後も改良が続いており，重みつき部分 MaxSAT にも対応した W-MaxSatz というアルゴリズムが開発されている．W-MaxSatz の詳細を記した論文はな

いようだが，著者のホームページよりプログラムのソースコードが入手できる．

MiniMaxSAT [Heras 08b] は，MaxSAT evaluation 2007 の重みなし部分 MaxSAT と重みつき部分 MaxSAT の 2 部門で 1 位になった分枝限定法に基づく解法であり，ハード節とソフト節の両方を含む重みつき部分 MaxSAT にも対応している．MiniMaxSAT では，分枝限定法の探索ノードにおける行き詰まりを，その原因がハード節の違反に由来する**ハードな矛盾**（hard conflict）と下界が上界を超えたことに由来する**ソフトな矛盾**（soft conflict）に分け，前者の場合には最新の系統的 SAT ソルバーと同様にバックジャンプ法（backjumping）と節学習（clause learning）を行ない，後者の場合には通常の年代順バックトラック法（chronological backtracking）を行なう．また，分枝限定法の各探索ノードにおける重みつき CNF 式を，空節を含む等価な重みつき CNF 式に置き換える．この置き換えは，重みつきの空節を導く反駁木を構成し，その葉より順次ソフト節を融合（resolution） [Larrosa 08] して行われる．よって，MaxSatz の推論規則に比べて，コストはかかるがより柔軟な置き換えができる．

3. QSAT

本章では，Quantified SAT (QSAT)，もしくは，Quantified Boolean Formulae (QBF) と呼ばれる，SAT の拡張について概説する．一言で言えば，通常の SAT では，命題変数はすべて存在限量化されていると考えられるが，QSAT では，一部の命題変数が全称限量化されていることを仮定する．この仮定のもとで，通常の SAT と同様に，与えられた CNF 式が充足可能かどうかを判定し，充足可能な場合はそれを充足する命題変数への真偽値の割り当てを求めることが目的である．

なぜ，このような全称限量化された命題変数を考える必要があるのだろうか? 通常の SAT は，問題を解こうとしている意思決定の主体が一人のみの場合を想定していると考えることができる．この主体を問題解決エージェントと呼ぶことにする．意思決定者は一人なので，世の中の命題変数すべては問題解決エージェントがコントロールでき, コントロールできない事象は，すべて定数として固定されていると見なすことができる．これに対して QSAT では，問題解決エージェント以外に，他の意思決定者が存在することを仮定する．他の意思決定者は，問題解決エージェントに対して，敵対的である可能性があり，問題解決を妨害するように振る舞うかも知れない．このような他の意思決定者を敵対エージェントと呼ぶことにする．

このような状況を考える必要がある応用領域として，二人ゲームや敵対者がいる場合のプランニング，偶発的な事態に備えたプランニング (contingency planning) 等がある．

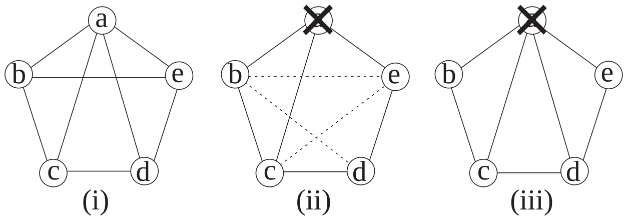

図 1 偶発的な事態に備えたプランニング (通信ネットワークの構築)

3·1 定式化／例題

以下，偶発的な事態に備えたプランニングの具体例を示そう．以下の目的を満たすように，通信ネットワークの設計と敷設のプランを作成することを考える．全体で 5 つの通信ノードが存在し，2 つのノード間を結ぶ通信路 (エッジ) を，最大で 9 本敷設可能であるとする．最終的に得られたネットワークでは，任意の二つのノードが，直接的，もしくは間接的にエッジを介して接続可能である必要があり，かつ，最大 2 つのノードが故障した場合でも，ノード間の接続経路が維持される必要がある．例えば，最終的なネットワークを図 1 (i) のように構成すれば，任意の 2 つのノードが故障しても接続関係は維持される．この場合は，8 本のエッジのみを利用している．また，さらに追加の要求条件として，エッジを敷設している間に，最大 1 つのノードが故障する可能性があり，その場合は，故障したノードを除いて，残りのノード間で，最大 2 つのノードが故障した場合でも，ノード間の接続経路が維持されるように，ネットワークを構成する必要がある．

例えば，図 1 (ii) 中の，実線のエッジを敷設した後に，ノード a が故障した場合，プランを変更して点線のエッジを敷設する必要がある．この場合，エッジは合計 9 本必要となる．また，もし図 1 (iii) のように 7 本のエッジを敷設した後に，ノード a が故障したとすると，残りの 2 本のエッジのみでは，目的を達成するように最終的なネットワークを構成することは不可能となる．このような事態が生じないように，エッジを敷設する順序を慎重に決定することが必要となる．

この問題は，プランを設計する問題解決エージェントと敵対エージェントが存在し，敵対エージェントがエッジの敷設中に，任意の一つのノードを故障させることが可能である状況と考えることができ，QSAT として表現することができる．

さて，CNF 式の節の一つが $x \vee y \vee z$ で，x が全称限量化された命題変数である場合，敵対エージェントは CNF 式を満足させないように値を選択すると考えられる．そうすると x の値は当然，偽となるので，結局，問題解決エージェントは節 $y \vee z$ を満足させる必要があることになる．このように考えると，単純にすべての節から，全称限量化された命題変数を取り除いて，残りの問題を通常の SAT として解けば良いように思われるが，これは正確ではな

い．以下の例を考えよう．$F(x, y) = (x \vee y) \wedge (\neg x \vee \neg y)$ という CNF 式を考える．ここで，x が全称限量化された命題変数だとすると，単純に x および $\neg x$ を各節から取り除くと，残りは $y \wedge \neg y$ となり，明らかにこの式は充足不能である．

上記の結論は正しいだろうか? まず問題解決エージェントが y の真偽値を決定し，次に敵対エージェントが，y の真偽値に依存して x の真偽値を決定するなら，敵対エージェントは y が真なら x を真にし，y が偽なら x を偽にすることにより，$F(x, y)$ を偽とすることが可能であり，F が充足不能という結論は正しい．

一方，まず敵対エージェントが x の真偽値を決定し，次に問題解決エージェントが，x の真偽値に依存して y の真偽値を決定するなら，問題解決エージェントは x が真なら y を偽にし，x が偽なら x を真にすることにより，$F(x, y)$ を真とすることが可能であり，F は充足可能となってしまう．

この例で示されているように，QSAT では限量化された命題変数の順序 (シーケンス) に重要な意味があり，シーケンスで後の命題変数は，前の命題変数の真偽値に依存して，値を決定することができる．これはゲーム理論における完全情報の二人ゲームにおける手番に相当し，自分の手 (命題変数の真偽値) を決める際には，それまでの相手の手を参照して決めることが可能である．よって，単純に全称限量化された命題変数を取り除いてよいのは，その命題変数がシーケンスの最後尾に置かれている場合のみである．

また，命題変数の値の割り当ては，シーケンスの前の命題変数の割り当てに依存するため，充足可能な QSAT の解は，SAT の解のように単純な命題変数への値の割り当てとしては表現することができず，x が真なら y を偽にする等の，条件付きのルールとなる．命題変数への値の割り当てで分岐する探索木を考えれば，SAT の解は探索木中の根節点から葉節点に至る一つのパスとして表現できるのに対して，QSAT の解は，全称限量化された命題変数に関しては真／偽の両方の分岐を含み，存在限量化された命題変数に関しては片方の分岐のみを含む部分木として表現される．

以下，QSAT の正確な定義を示す．QSAT は命題変数を**存在限量子**（existential quantifier）∃，あるいは，**全称限量子**（universal quantifier）∀ で限量化することにより SAT を一般化した問題である．各限量子の意味は次の通りである．

- $\exists x F$ が充足可能：式 F を真とするある x の値が存在する．
- $\forall x F$ が充足可能：任意の x の値について式 F は真となる．

QSAT は式 (2) のように，各命題変数を限量化する限量子を定義するシーケンス Q と，CNF 形式の命題論理式 F を用いて QF と定義される．

$$\exists x_1 \forall x_2 \exists x_3 (x_1 \vee x_2) \wedge (x_2 \vee \neg x_3) \tag{2}$$

限量子のシーケンスは，式 (3) のように，n 個の限量子 $Q_i(\exists, \forall)$ と命題変数 x_i のペアで構成されるシーケンスのことである．

$$Q_1 x_1 \cdots Q_n x_n \tag{3}$$

QSAT は通常の SAT を含むため，明らかに通常の SAT より難しく，通常の SAT が NP 完全であるのに対して，QSAT は PSPACE 完全となる．クラス PSPACE とは，チューリングマシンによって多項式領域で（多項式の長さのテープを用いて）解ける問題のクラスであり，クラス PSPACE はクラス NP を含む．PSPACE 完全はクラス PSPACE 中で最も難しい (クラス PSPACE の任意の問題を表現可能な) 問題を意味する．

存在限量化された命題変数に関して，

- 左の分岐が真なら成功（論理式は真）
otherwise, 右の分岐が真なら成功（論理式は真）
otherwise，失敗．
- 成功の場合，まだ完全に展開されていない，直前に値を決定した全称限量化された命題変数にバックトラックする（そのような命題変数がなければ終了，論理式は真）．
- 失敗の場合, まだ完全に展開されていない，直前に値を決定した存在限量化された命題変数にバックトラックする（そのような命題変数がなければ終了，論理式は偽）．

全称限量化された命題変数に関して，

- 左の分岐が偽なら成功（論理式は偽）
otherwise，右の分岐が偽なら成功（論理式は偽）
otherwise, 失敗．
- 成功の場合，まだ完全に展開されていない，直前に値を決定した存在限量化された命題変数にバックトラックする（そのような命題変数がなければ終了，論理式は偽）．
- 失敗の場合, まだ完全に展開されていない，直前に値を決定した全称限量化された命題変数にバックトラック する（そのような命題変数がなければ終了，論理式は真）．

図 2　QSAT に対する DPLL ベースのアルゴリズムの概要

3·2 アルゴリズム

QSAT の代表的な解法として，通常の SAT で用いられる DPLL アルゴリズム [Davis 62] に基づくアルゴリズムがある．このアルゴリズムは，命題変数への真偽値の割り当てで分岐する探索木を探索するアルゴリズムである．以下，存在限量化された命題変数での分岐に関して，以降の論理式が真となった場合に成功, 偽となった場合に失敗とする．一方，全称限量化された命題変数での分岐に関して，論理式が偽となった場合に成功, 真となった場合に失敗とする．QSAT に対する DPLL ベースのアルゴリズムの概要を図 2 に示す．

通常の SAT を解く DPLL アルゴリズムでは，分岐する命題変数の選択の順序は任意であるが，QSAT ではシーケンス中の順序を維持する必要がある．連続する存在限量化された命題変数，もしくは連続する全称限量化された命題変数に関しては，順序の選択は任意であるが，その他の命題変数の順序を入れ替えることはできない．

DPLL ベースのアルゴリズムとして，QuBE [Giunchiglia 01] や Quaffle [Zhang 02] がある．また，DPLL ベースのアルゴリズムを高速化するためのテクニックとして，通常の SAT と同様の単位伝播，純リテラル規則等が用いられている．また，前述のように，シーケンスの最後尾の全称限量化された命題変数は取り除くことが可能である．さらに，通常の SAT と同様，学習節の追加による高速化等が提案されている [Giunchiglia 03].

その他, 融合を一般化した Q **融合**（Q-resolution）[Büning 95] に基づくアルゴリズムである Quantor [Biere 04]，スコーレム化を用いたソルバーである skizzo [Benedetti 05] 等が提案されている．

QSAT ソルバーの性能は，シーケンス中の，全称限量子と存在限量子の切り替えの回数に大きく依存する．この切り替えの回数が数回程度であれば，QSAT を SAT に符号化して，既存の SAT ソルバーで解くことも十分現実的である．様々なテクニックを導入した QSAT ソルバー [Sabharwal 06] により，10 数回の限量子の切り替えが生じる，10 万変数程度の問題が現実的な時間で解けるという報告がなされている．

4. 並　列　SAT

最新の系統的 SAT ソルバーは，何百万変数からなる問題例を数分で解くことができる [Eén 03]．そのため，SAT ソルバーに対する期待は非常に大きく，ますます多くの実用的な問題例が SAT に符号化される一方で，ここ何年かは SAT ソルバーの性能は劇的には向上しておらず，SAT 競技会の問題例でも未解決のままに終わる例が増えていると言われている [Hamadi 09].

一方で，マルチコア CPU や PC クラスタなどの技術が進み, 近年, 並列処理を行なう環境が比較的容易に構築できるようになった．このような背景をもとに，並列 SAT ソルバーに対する期待が徐々に高まりつつある．SAT 競技会の一つである SAT レースでは，2008 年に並列ソルバー部門ができ，3 つのソルバーが参加した．本章では並列 SAT ソルバーの概要を紹介する．

4·1　アルゴリズム

既存の並列 SAT ソルバーは，分割統治型解法（divide-and-conquer solver）とポートフォリオ型解法（portfolio-based solver）の 2 つに分けることができる．

§1　分割統治型解法

分割統治型解法の基本的なアイデアは，系統的 SAT ソルバーの探索木を複数の部分探索木に再帰的に分割し，複数のプロセスでそれらを並列に解くことにより求解の効率を上げようというものである．通常，分割はオンラインで行われ，あるプロセスが自分の探索木において，まだ探索されていない部分探索木を空いている別のプロセスに割り当てるという要領で行われる．また，複数のプロセスで学習節を共有すると全体の探索効率が向上することが知られている．

既存の並列 SAT ソルバーの多くはこの分割統治型解法に分類される．PSATO [Zhang 96] はそのようなソルバーの一つであり，他のプロセスに割り当てる部分探索木を**誘導パス**（guiding path）を用いて表現することを提案した．その後，探索空間を分割するだけでなく，プロセス間で学習節を共有する多くの並列 SAT ソルバーが提案されている [Chrabakh 06, Feldman 05, Gil 08, Schubert 09, Sinz 01].

§2　ポートフォリオ型解法

一方，ポートフォリオ型解法の基本アイデアは，ある一つの問題例に対して，異なる振舞いをする複数の解法をそれぞれ別プロセスとして並列に実行し，そのうちの一つが結果を得た時点で全体を終了させるというものである．その際，複数のプロセスが実行時に何らかの情報を交換して「協力」すると，全体の探索効率が向上する．グラフ彩色問題などの組合せ問題に対してこのアイデアを適用した研究は幾つか存在するが [Clearwater 91, Hogg 93]，SAT では，ポートフォリオ型解法が注目されるようになったのは比較的最近になってからである．

ManySAT [Hamadi 09] は，SAT レース 2008 の並列ソルバー部門で 1 位となったポートフォリオ型解法である．ManySAT では，最新の系統的 SAT ソルバーの一つである MiniSAT [Eén 03] の様々な戦略のうち，リスタートポリシー，変数選択，値選択，および，節学習の各項目についていくつかの方法を用意し，それらを適宜組み合わせた異なる 4 つのソルバーをマルチコア CPU 上で並列に実行する．また，ソルバー間でサイズ 8 以下の学習節を共有する．

Multisat [Inoue 06] は，系統的 SAT ソルバーである Satz [Li 97] と Chaff [Moskewicz 01]，および，確率的 SAT ソルバーである WalkSAT [Selman 96] を並列に実行し，さらに互いに学習節を共有することを許したポートフォリオ型解法である．Multisat は，系統的 SAT ソルバーと確率的 SAT ソルバーという全く異なるソルバーを組合せて協調させることが，特にプランニングやスケジューリングなどの問題例で有効であることを示した．

MiniWalk [Kroc 09] は，系統的 SAT ソルバーである MiniSAT [Eén 03] と確率的 SAT ソルバーである WalkSAT [Selman 96] の 2 つのソルバーをマルチコア CPU 上で並列実行するポートフォリオ型解法である．MiniSAT

は，探索木における現在の探索ノードを共有メモリに書き込み，一方，WalkSAT は，共有メモリを参照して MiniSAT の現在の探索ノードに矛盾しない値への反転のみを行う．すなわち，MiniSAT の現在の探索ノードでまだ値の確定していない命題変数については任意の値反転を行い，値の確定している命題変数についてはその確定した値への反転のみを行う．MiniWalk は，最適値が 1 に近い重みなし MaxSAT で特に有効であると報告されている．

5. 分　散　SAT

本章では，問題の構成要素が複数の自律的なエージェントに分散された分散 SAT とそれを解くアルゴリズムの概要を紹介する．まず，並列 SAT と分散 SAT の違いを明確にしよう．

4 章で説明した通り，並列 SAT では SAT を並列処理により高速に解くことを目指す．すなわち，並列 SAT では，SAT の求解プロセスを任意に分割することができ，基本的には処理の並列性を生かして SAT を高速に解く．

一方，分散 SAT では，複数のエージェントに分散された SAT を解くことを目指す．すなわち，分散 SAT では，マルチエージェントシステムにおける問題等を想定し，複数の自律的なエージェントに問題の構成要素が分散されている SAT を（問題を一か所に集めずに）エージェント同士が協調して解く．一般に，マルチエージェントシステムでは，システムの頑健性や柔軟性，および，各エージェントのプライバシーやセキュリティ等を重視するため，分散 SAT ソルバーは集中制御のないピアツーピア型のエージェント間通信プロトコル（分散アルゴリズム）であることが望ましい．

5·1 定式化／例題

分散 SAT には 2 つの定式化が存在する．

一つは，SAT の節集合が分割され，各部分をそれぞれ 1 つのエージェントが管理するというものである．本稿では，**節集合分割型の分散** SAT（clause-set partitioned distributed SAT）とよぶ．文献 [Adjiman 06, Amir 05, 岡本 05] はこの定式化による．

一方，SAT の命題変数集合が分割され，各部分をそれぞれ 1 つのエージェントが管理するという定式化もある．これを**変数集合分割型の分散** SAT（variable-set partitioned distributed SAT）とよぶ．この場合，命題変数については，それを管理するエージェントが一意に決まるが，節については，そこに現れる命題変数を管理する複数のエージェントで共同管理することになる．文献 [Hirayama 02, Hirayama 05] はこの定式化による．また，分散 SAT を一般化した分散制約充足問題（distributed constraint satisfaction problem） [Yokoo 98] でも，基本的に後者の定式化が用いられる．

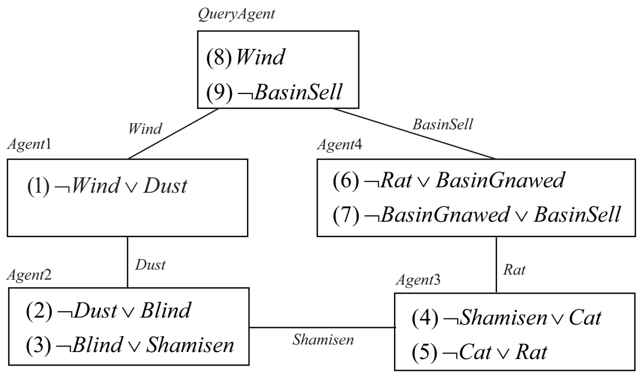

図 3　分散知識ベースシステム

ただし，両者は互いに変換可能である．従って，解法を設計する際にはどちらの定式化を前提としてもよい．

図 3 は，それぞれのエージェントがローカルな知識ベースを管理する分散知識ベースシステムの例である．この例は，節集合分割型の分散 SAT であり，各エージェントが管理する節集合を四角で囲んでグラフの頂点とし，また，任意の 2 つのエージェントの節集合がある命題変数を共有するならば，対応する 2 つの頂点を枝で結んでその共有変数の名前を枝のラベルとした．

この例では，プライバシーやセキュリティ等の理由，また，知識ベース自体の移転が物理的に困難であるという理由から，どのエージェントも全体の知識にはアクセスできないという前提がある．この前提のもとで，各共有変数の値がそれをもつエージェント間で一致し，かつ，すべての節を充足させるような全変数への真偽値の割り当てが存在するか否かを，エージェントが互いに情報を部分的に交換しながら判定する．

なお，各共有変数の値がそれをもつエージェント間で一致するという制約は CNF 式で記述できる．これを関連するエージェントに与えれば変数集合分割型の分散 SAT になる．例えば，図 3 では，エージェント 1 と 2 の間の共有変数 $Dust$ について，$Dust_1 \vee \neg Dust_2$ と $\neg Dust_1 \vee Dust_2$ という 2 つの節を作って両者に与え，共有変数 $Dust$ を，エージェント 1 は $Dust_1$，エージェント 2 は $Dust_2$ としそれぞれ別の変数として管理すればよい．

5·2 アルゴリズム

分散 SAT ソルバーの研究はまだ非常に少ないというのが現状である．よって，本稿では少し視点を広げて，分割に基づく推論（partition-based reasoning），分散型結論発見（distributed consequence finding），および，分散制約充足アルゴリズムの研究から関連するものを挙げる．

§1 分割に基づく推論

Amir と McIlraith は，知識の量が膨大で一カ所では管理できないという動機から，大規模知識ベースを複数の知識ベースに分割して解くことを提案している [Amir 05]．彼らの解法では，分割された複数の知識ベース（節集合）

上での結論発見および充足可能性判定を行う．結論発見とは，与えられた公理集合から隠れた結論を発見することであり，充足可能性判定よりも困難な問題である．ただし，彼らの解法では，図 3 に相当するグラフが木構造であることを前提にしており，グラフが木構造でない場合には，閉路除去を行う特別な手続きが必要になる．

Amir と McIlraith の研究に始まる分割に基づく推論というアプローチは，一見，分散 SAT と類似しているが，設計者がアルゴリズムにとって都合が良いように分割を構成できる点で，分散 SAT ソルバーが本来目指しているものとは少し異なる．

§2 分散型結論発見

一方，複数の知識ベースがもともと地理的に分散しており，それを所与として複数のエージェントが協調的に結論発見を行うという研究も存在する．

岡本と井上は，分散された公理集合群よりエージェントが協調的に結論発見を行う手法を提案している [岡本 05]．また，Adjiman らも，同様の設定で結論発見を行う手法を提案し，セマンティック Web のデータ管理システムに適用している [Adjiman 06].

§3 分散制約充足アルゴリズム

分散 SAT を一般化した分散制約充足問題では，分散制約充足問題を解く分散制約充足アルゴリズムが多数提案されており [Yokoo 00a]，その性能評価の一部として変数集合分割型の分散 SAT が利用されることがある．ただし，分散制約充足アルゴリズムの多くは基本的に 1 エージェントが 1 変数のみをもつケースを想定するため，分散 SAT のような 1 エージェントが一般に多くの節をもつようなケースでは相応の工夫が必要になる．文献 [Hirayama 02, Hirayama 05, 横尾 00b] では，各エージェントが複雑な局所問題をもつ場合の分散制約充足アルゴリズム／分散 SAT アルゴリズムが提案されている．

6. ま と め

本稿では，SAT の枠組みを拡張する試みとして，MaxSAT，QSAT，並列 SAT，そして，分散 SAT に関する研究の概要を紹介した．

もちろん，SAT を拡張する試みはこれ以外にも存在する．紙面の都合と筆者の力量不足のため本稿では紹介することはできなかったが，独立した決定手続きをもつ背景理論を SAT に組込んだ satisfiability modulo theories については，今回の特集号に岩沼氏と鍋島氏による解説 [岩沼 10] がある．また，SAT のモデルを全て列挙するモデル列挙（model enumeration），モデルの数を数えるモデル計数（model counting，もしくは，#SAT）については，同じく今回の特集号に長谷川氏，藤田氏，越村氏による解説 [長谷川 10] があるので，あわせて参照していただきたい．

◇ 参 考 文 献 ◇

[Adjiman 06] Adjiman, P., Chatalic, P., Goasdoué, F., Rousset, M.-C., and Simon, L.: Distributed Reasoning in a Peer-to-Peer Setting: Application to the Semantic Web, *Journal of Artificial Intelligence Research*, Vol. 25, pp. 269–314 (2006)

[Amir 05] Amir, E. and McIlraith, S.: Partition-Based Logical Reasoning for First-Order and Propositional Theories, *Artificial Intelligence*, Vol. 162, No. 1–2, pp. 49–88 (2005)

[Ansótegui 09] Ansótegui, C., Bonet, M. L., and Levy, J.: Solving (Weighted) Partial MaxSAT Through Satisfiability Testing, in *Proceedings of the 12th International Conference on Theory and Applications of Satisfiability Testing (SAT-2009)*, pp. 427–440 (2009)

[Argelich 08] Argelich, J., Li, C.-M., Manyà, F., and Planes, J.: The First and Second Max-SAT Evaluations, *Journal on Satisfiability, Boolean Modeling and Computation*, Vol. 4, pp. 251–278 (2008)

[Bailleux 03] Bailleux, O. and Boufkhad, Y.: Efficient CNF Encoding of Boolean Cardinality Constraints, in *Proceedings of the 9th International Conference on Principles and Practice of Constraint Programming (CP-2003)*, pp. 108–122 (2003)

[Benedetti 05] Benedetti, M.: sKizzo: a Suite to Evaluate and Certify QBFs, in *Proceedings of 20th International Conference on Automated Deduction (CADE-2005)*, pp. 369–376 (2005)

[Berre 09] Berre, D. L.: SAT4J: Bringing the Power of SAT Technology to the Java Platform, in *http://www.sat4j.org/* (2009)

[Biere 04] Biere, A.: Resolve and Expand, in *Proceeding of the Seventh International Conference on Theory and Applications of Satisfiability Testing (SAT-2004)*, pp. 59–70 (2004)

[Büning 95] Büning, H. K., Karpinski, M., and Flögel, A.: Resolution for Quantified Boolean Formulas, *Information and Computation*, Vol. 117, No. 1, pp. 12–18 (1995)

[Chrabakh 06] Chrabakh, W. and Wolski, R.: GridSAT: a System for Solving Satisfiability Problems Using a Computational Grid, *Parallel Computing*, Vol. 32, No. 9, pp. 660–687 (2006)

[Clearwater 91] Clearwater, S. H., Huberman, B. A., and Hogg, T.: Cooperative Solution of Constraint Satisfaction Problems, *Science*, Vol. 254, No. 5035, pp. 1181–1183 (1991)

[Davis 62] Davis, M., Logemann, G., and Loveland, D.: A Machine Program for Theorem-proving, *Communications of the ACM*, Vol. 5, No. 7, pp. 394–397 (1962)

[Eén 03] Eén, N. and Sörensson, N.: An Extensible SAT-Solver, in *Proceedings of 6th International Conference on Theory and Applications of Satisfiability Testing (SAT-2003)*, pp. 502–518 (2003)

[Eén 06] Eén, N. and S örensson, N.: Translating Pseudo-Boolean Constraints into SAT, *Journal on Satisfiability, Boolean Modeling and Computation*, Vol. 2, pp. 1–26 (2006)

[Feldman 05] Feldman, Y., Dershowitz, N., and Hanna, Z.: Parallel Multithreaded Satisfiability Solver: Design and Implementation, *Electronic Notes in Theoretical Computer Science*, Vol. 128, pp. 75–90 (2005)

[Fu 06] Fu, Z. and Malik, S.: On Solving the Partial MAX-SAT Problem, in *Proceedings of the 9th International Conference on Theory and Applications of Satisfiability Testing (SAT-2006)*, pp. 252–265 (2006)

[Gil 08] Gil, L., Flores, P., and Silveira, L. M.: PMSat: a Parallel Version of MiniSAT, *Journal on Satisfiability, Boolean Modeling and Computation*, Vol. 6, pp. 71–98 (2008)

[Giunchiglia 01] Giunchiglia, E., Narizzano, M., and Tacchella, A.: QUBE: a System for Deciding Quantified Boolean Formulas Satisfiability, in *Proceedings of the First International Joint Conference on Automated Reasoning (IJCAR-2001)*, pp. 364–369 (2001)

[Giunchiglia 03] Giunchiglia, E., Narizzano, M., and Tacchella, A.: Backjumping for Quantified Boolean Logic Satisfiability, *Artificial Intelligence*, Vol. 145, No. 1–2, pp. 99–120 (2003)

[Hamadi 09] Hamadi, Y., Jabbour, S., and Sais, L.: ManySAT: a Parallel SAT Solver, *Journal on Satisfiability, Boolean Modeling and Computation*, Vol. 6, pp. 245–262 (2009)

[長谷川 10] 長谷川 隆三, 藤田 博, 越村 三幸：モデル列挙とモデル計数, 人工知能学会誌, Vol. 25, No. 1 (2010)

[Heras 08a] Heras, F., Larrosa, J., deGivry, S., and Schiex, T.: 2006 and 2007 Max-SAT Evaluations: Contributed Instances, *Journal on*

Satisfiability, Boolean Modeling and Computation, Vol. 4, pp. 239–250 (2008)

[Heras 08b] Heras, F., Larrosa, J., and Oliveras, A.: MiniMaxSat: An Efficient Weighted Max-SAT Solver, *Journal of Artificial Intelligence Research*, Vol. 31, pp. 1–32 (2008)

[Hirayama 02] Hirayama, K. and Yokoo, M.: Local Search for Distributed SAT with Complex Local Problems, in *Proceedings of the First International Joint Conference on Autonomous Agents & Multi-Agent Systems (AAMAS-2002)*, pp. 1199–1206 (2002)

[Hirayama 05] Hirayama, K. and Yokoo, M.: The Distributed Breakout Algorithms, *Artificial Intelligence*, Vol. 161, No. 1–2, pp. 89–115 (2005)

[Hogg 93] Hogg, T. and Williams, C. P.: Solving the Really Hard Problems with Cooperative Search, in *Proceedings of the Eleventh National Conference on Artificial Intelligence (AAAI-1993)*, pp. 231–236 (1993)

[Inoue 06] Inoue, K., Soh, T., Ueda, S., Sasaura, Y., Banbara, M., and Tamura, N.: A Competitive and Cooperative Approach to Propositional Satisfiability, *Discrete Applied Mathematics*, Vol. 154, No. 16, pp. 2291–2306 (2006)

[岩沼 10] 岩沼 宏治, 鍋島 英知：SMT: 個別理論を取り扱う SAT, 人工知能学会誌, Vol. 25, No. 1 (2010)

[Karloff 97] Karloff, H. and Zwick, U.: A 7/8-approximation Algorithm for MAX 3SAT?, in *Proceedings of the 38th Annual Symposium on Foundations of Computer Science (FOCS-1997)*, pp. 406–415 (1997)

[Kroc 09] Kroc, L., Sabharwal, A., Gomes, C. P., and Selman, B.: Integrating Systematic and Local Search Paradigms: A New Strategy for MaxSAT, in *Proceedings of the 21st International Joint Conference on Artificial Intelligence (IJCAI-2009)*, pp. 544–551 (2009)

[Larrosa 08] Larrosa, J., Heras, F., and Givry, de S.: A Logical Approach to Efficient Max-SAT Solving, *Artificial Intelligence*, Vol. 172, No. 2–3, pp. 204–233 (2008)

[Li 97] Li, C. M. and Anbulagan, : Heuristics Based on Unit Propagation for Satisfiability Problems, in *Proceedings of the 15th International Joint Conference on Artificial Intelligence (IJCAI-1997)*, pp. 366–371 (1997)

[Li 07] Li, C. M., Manyà, F., and Planes, J.: New Inference Rules for Max-SAT, *Journal of Artificial Intelligence Research*, Vol. 30, pp. 321–359 (2007)

[Marques-Silva 08] Marques-Silva, J. and Planes, J.: Algorithms for Maximum Satisfiability using Unsatisfiable Cores, in *Proceedings of Design, Automation and Test in Europe (DATE-2008)*, pp. 408–413 (2008)

[Moskewicz 01] Moskewicz, M. W., Madigan, C. F., Zhao, Y., Zhang, L., and Malik, S.: Chaff: Engineering an Efficient SAT Solver, in *Proceedings of the 38th Design Automation Conference (DAC-2001)*, pp. 530–535 (2001)

[岡本 05] 岡本 孝之, 井上 克巳：メッセージ通信を用いた分散型結論発見, 電子情報通信学会技術研究報告, AI2004-75, pp. 25–30 (2005)

[Sabharwal 06] Sabharwal, A., Ansotegui, C., Gomes, C. P., Hart, J. W., and Selman, B.: QBF Modeling: Exploiting Player Symmetry for Simplicity and Effciency, in *Proceeding of the 9th International Conference on Theory and Applications of Satisfiability Testing (SAT-2006)*, pp. 382–395 (2006)

[Schubert 09] Schubert, T., Lewis, M., and Becker, B.: PaMiraXT: Parallel SAT Solving with Threads and Message Passing, *Journal on Satisfiability, Boolean Modeling and Computation*, Vol. 6, pp. 203–222 (2009)

[Selman 96] Selman, B., Kautz, H., and Cohen, B.: Local Search Strategies for Satisfiability Testing, *Cliques, Coloring and Satisfiability: the Second DIMACS Implementation Challenge*, Vol. 26, pp. 521–532 (1996)

[Sinz 01] Sinz, C., Blochinger, W., and Küchlin, W.: PaSAT–Parallel SAT-Checking with Lemma Exchange: Implementation and Applications, in *LICS 2001 Workshop on Theory and Applications of Satisfiability Testing* (2001)

[Sinz 05] Sinz, C.: Towards an Optimal CNF Encoding of Boolean Cardinality Constraints, in *Proceedings of the 11th International Conference on Principles and Practice of Constraint Programming (CP-2005)*, pp. 827–831 (2005)

[Yokoo 98] Yokoo, M., Durfee, E. H., Ishida, T., and Kuwabara, K.: The Distributed Constraint Satisfaction Problem: Formalization and Algorithms, *IEEE Transactions on Knowledge and Data Engineering*, Vol. 10, No. 5, pp. 673–685 (1998)

[Yokoo 00a] Yokoo, M. and Hirayama, K.: Algorithms for Distributed Constraint Satisfaction: A Review, *Autonomous Agents and Multi-agent Systems*, Vol. 3, No. 2, pp. 189–211 (2000)

[横尾 00b] 横尾 真, 平山 勝敏：複雑な局所問題に対応する分散制約充足アルゴリズム, 人工知能学会誌, Vol. 15, No. 2, pp. 348–354 (2000)

[Zhang 96] Zhang, H., Bonacina, M. P., and Hsiang, J.: PSATO: a Distributed Propositional Prover and Its Application to Quasigroup Problems, *Journal of Symbolic Computation*, Vol. 21, No. 4–6, pp. 543–560 (1996)

[Zhang 02] Zhang, L. and Malik, S.: Conflict Driven Learning in a Quantified Boolean Satisfiability Solver, in *Proceedings of International Conference on Computer-Aided Design (ICCAD-2002)*, pp. 442–449 (2002)

〔担当委員：××○○〕

19YY 年 MM 月 DD 日　受理

著 者 紹 介

平山　勝敏（正会員）

1990 年 大阪大学基礎工学部制御工学科卒．1992 年 同大大学院基礎工学研究科博士前期課程修了．1995 年 同大大学院基礎工学研究科博士後期課程修了．博士（工学）．1995 年 神戸商船大学助手．1997 年 同講師．2001 年 同助教授．2003 年 神戸大学海事科学部助教授（神戸大学と神戸商船大学の統合による）．2007 年 神戸大学大学院海事科学研究科准教授．1999 年–2000 年 カーネギーメロン大学ロボティクス研究所客員研究員（文部省在外研究員）．マルチエージェントシステム，制約充足，組合せ最適化に関する研究に従事．電子情報通信学会，情報処理学会，人工知能学会，日本オペレーションズリサーチ学会，AAAI 各会員．

横尾　　真（正会員）

1986 年 東京大学大学院 電気工学専門課程修了．同年 NTT に入社．2004 年より九州大学大学院 システム情報科学研究院 教授．博士 (工学)．マルチエージェントシステム，制約充足問題に関する研究に従事．1992 年，2002 年 人工知能学会論文賞，1995 年 情報処理学会坂井記念特別賞，2004 年 ACM SIGART Autonomous Agent Research Award，2005 年 日本ソフトウェア科学会論文賞，2006 年 学士院学術奨励賞 受賞．