HOKKAIDO UNIVERSITY

Title	命題論理におけるβ節充足可能性問題とその解法
Author(s)	大柳, 俊夫; 山本, 雅人; 大内, 東
Citation	電気学会論文誌 C 電子・情報・システム部門誌, 114(7), 796-804
Issue Date	1994-07
Doc URL	https://hdl.handle.net/2115/64534
Type	journal article
File Information	114_796.pdf

Hokkaido University Collection of Scholarly and Academic Papers : HUSCAP

論　文

命題論理におけるβ節充足可能性問題とその解法

非会員　大柳俊夫（札幌医科大）
非会員　山本雅人（北海道大）
正　員　大内東（北海道大）

An Algorithm for β Clause Satisfiability Problem in Propositional Logic

Toshio Ohyanagi, Non-member(Sapporo Medical University), Masahito Yamamoto, Non-member, Azuma Ohuchi, Member (Hokkaido University)

The satisfiability problem (SAT) is one of the important problems in the field of computer science, since it is well known *NP-complete* problem. The problem is well studied and many valuable results are reported in the literature. We have already proposed IEMSAT algorithm as one of SAT algorithms. Recently, J. N. Hooker made a study of *resolution* for a set of β clauses. A β clause is a clause that has an additional quantitative condition. The condition is that a β clause is true if and only if at least number of β literals in the clause are true. He also referred to β SAT problem that is SAT consists of β clauses. He, however, did not propose efficient algorithm for it as yet.

This paper intends to present β SAT algorithm (IEMBSAT) that is an extended variant of IEMSAT. Some relations between a β clause and a clause are also clarified. Some computational experiments are made to show the effectiveness of IEMBSAT.

キーワード：命題論理充足可能性問題 (SAT)，β 節，β 節 充足可能性問題 (β SAT)，0-1 整数計画問題

1. まえがき

命題論理の充足可能性問題 (SAT) は，その計算複雑度が NP-完全である代表的な問題として，情報処理の分野における基本的な問題の一つにあげられている。この問題は，与えられた命題論理の基礎節集合中のすべての基礎節を真とする原子論理式への真偽の割り当てが存在するか否かを決定するものである。

この問題に対するこれまでの研究は，Davis-Putnam の方法に代表される記号的方法[(1)(2)] と数理計画法に基づく定量的方法 [(3)-(10)] に大別される。前者は，古典的な記号論理学として 1960 年代頃盛んに行われた研究である。これに対して後者は，主に OR の分野の研究者によって近年行われつつあるもので，著者らも定量的方法として，SAT を 0-1 整数計画問題に定式化した場合の特徴を利用した，陰的列挙法[(11)]に基づく SAT アルゴリズム (IEMSAT) を提案している [(10)]。

最近になって Hooker は，基礎節に少なくともある決まった個数のリテラルが真であるという量的な条件を付加し，そのような基礎節 (β 節) の集合における導出に関する研究を行っている[(6)]。さらに，β 節 充足可能性問題 (以下，β SAT と呼ぶ) についても言及している。しかし，β SAT に対する効率の良いアルゴリズムは提案されておらず，また Davis-Putnam の方法に代表される記号的方法の中にも β SAT を扱うアルゴリズムは見当たらない。

本論文では，まずβ 節 と基礎節の関係を明らかにし，β SAT を 0-1 整数計画問題へ定式化する新しい方法を示す。つぎにその定式化に基づき，IEMSAT で用いている諸定理を見直してβ 節 を扱うことを可能にした，β SAT アルゴリズム (IEMBSAT) を提案する。そして，ランダム SAT[(10)]と同様な方法で作成するランダムβ SAT と SAT アルゴリズムのベンチマークテストとして最近用いられている n-Queen 問題を使った数値実験を行い，IEMBSAT の有効性を検証する。

以下，2 章でβ 節 の定義を述べ，基礎節との関係を明らかにする。そして，β SAT とβ SAT を 0-1 整数計画問題へ定式化する方法を示す。3 章では IEMSAT をβ SAT に適用できるようにした IEMBSAT を説明する。そして 4 章で数値実験の方法と実験結果を述べ，IEMBSAT の有効性について考察する。

2. β SAT と 0-1 整数計画問題

<2·1>　β 節　一般に，リテラル l_j $(j=1,\cdots,n)$ の選言からなる基礎節 $C=l_1\vee,\cdots,\vee l_n$ に対し，β 節 とその真偽は次のように定義される。

定義 2.1 (β 節 とその真偽)

基礎節 C に量的な条件を付加したものをβ 節 もしくはβ 次数の節と呼び，$C(\beta)$ と表す。そしてβ 節 $C(\beta)$ において，

少なくともβ個のリテラルが真のとき，そのβ 節 は真であるという。

このβ 節 により，論理的な条件に量的な条件を加えた次の3つの形式の条件が表現できる。

1. 節中のn 個のリテラルのうち少なくとも k個が真である。
2. 節中のn 個のリテラルのうち高々k個が真である。
3. 節中のn 個のリテラルのうちちょうどk個が真である。

1.は，β 節 の定義より $l_1 \vee \cdots \vee l_n(k)$ となる。2.の条件は，「節中の n 個のリテラルのうち少なくとも $n-k$個が偽である。」と言い換えることができるので，β 節 $\neg l_1 \vee \cdots \vee \neg l_n(n-k)$ となる。そして3.の条件中の「ちょうどk個が真」の部分が「少なくとも k個が真かつ高々k個が真」と言い換えることができるので，2つのβ 節 $l_1 \vee \cdots \vee l_n(k), \neg l_1 \vee \cdots \vee \neg l_n(n-k)$ となる。

<2·2>　β 節 と基礎節の関係　　β 節 の定義から明らかなように，基礎節は β 節 においてβが1である特別な場合であり，このことからβ 節 は基礎節を一般化したものと考えることができる。

また，β 節 と基礎節の関係として，β 節 はそれと等価な基礎節集合に変換できることをHookerは示唆している。そして例として，β 節 $l_1 \vee l_2 \vee l_3(2)$ を取り上げ，このβ 節 が充足可能のとき，またそのときに限り基礎節集合 $\{ l_1 \vee l_2, l_2 \vee l_3, l_1 \vee l_3 \}$ は充足可能であることを説明している[6]。しかし，β 節 から基礎節集合への一般的な変換手続きは説明されおらず，また，著者の知る限りそのような変換方法はいままで明らかにされていない。

β 節 $(l_1 \vee \cdots \vee l_n(\beta))$ を基礎節集合へ変換する一般的な方法を検討した結果，図1の変換アルゴリズムが得られた。この変換アルゴリズムがβ 節 と等価な基礎節集合を生成することは次の定理による。証明は付録に示す。

定理 2.1 (等価変換に関する定理)

β 節 $l_1 \vee \cdots \vee l_n(\beta)$ が充足可能なとき，またそのときに限り変換アルゴリズムによって生成される${}_nC_{\beta-1}$個の基礎節はすべて充足可能である。

以上より，β 節 と基礎節の関係は以下のようにまとめることができる。

Step 0: 求める基礎節集合 Sを空にする。
Step 1: n個のリテラルから$\beta-1$個リテラルを選ぶ。
Step 2: n 個のリテラルから選ばれた$\beta-1$ 個のリテラルを除き，$n-\beta+1$ 個のリテラルで基礎節を作り Sへ加える。
Step 3: n 個のリテラルから$\beta-1$ 個のリテラルを選ぶ他の組合せが存在すればその組合せを1つ選びStep 2へ戻る。さもなければ，終了する。

図1. β 節 から基礎節集合への変換アルゴリズム
Fig. 1. A transformation algorithm from a β clause to a set of clauses.

- β 節 は基礎節を一般化したものである。
- β 節 はそれと等価な基礎節集合に変換できる。

<2·3>　β SAT　　β SAT は，β 節 集合，

$$BS = \{C_1(\beta_1), C_2(\beta_2), \cdots, C_m(\beta_m)\} \quad \cdots\cdots(1)$$

として与えられる。ここで，C_i $(i=1,\cdots,m)$ はリテラル l_j $(j=1,\cdots,n)$ の選言からなる基礎節である。各リテラル l_jは，原子論理式 p_j またはその否定$\neg p_j$である。

β SAT は BS中のすべてのβ 節 を真とする原子論理式 p_j $(j=1,\cdots,n)$ への真偽の割り当てが存在するか否かを調べる問題で，そのような割り当てが存在するとき BSは充足可能であるという。以下に BSの例を示す。

[例題]

$$BS = \{p_1(1), p_1 \vee p_3 \vee p_4 \vee \neg p_5(2), \\ p_1 \vee \neg p_3 \vee p_5(2), \neg p_1 \vee p_2 \vee p_5 \vee \neg p_6(3), \\ \neg p_2 \vee p_4 \vee \neg p_6(2), p_2 \vee \neg p_4 \vee p_6(1)\} \quad \cdots\cdots(2)$$

は，6個のβ 節 からなり，各β 節 は原子論理式 $p_1,\cdots,p_6$ またはその否定の選言である。この例では，後に示すように，$(p_1,p_2,p_3,p_4,p_5,p_6)$ = (真, 真, 偽, 真, 真, 偽) とすると6個のβ 節 がすべて真となり，BSは充足可能と判定される。

<2·4>　β SAT の 0–1 整数計画問題としての定式化

β SAT を 0–1 整数計画問題へ定式化する方法として，

1. Hookerの定式化[6]よりも用いる変数の数が少ない，
2. IEMSAT で用いている定式化の拡張になっている，

という特徴をもつ新しい方法を示す。

(1) 式のあるβ 節 $(C_i(\beta_i))$ に対し，一次不等式，

$$a_{i1}x_1 + a_{i2}x_2 + \cdots + a_{in}x_n \geq \beta_i - n(a_{i\cdot}) \quad \cdots\cdots(3)$$

を考える。ここで，x_jはβ 節 中の原子論理式p_jが真のとき1，偽のとき0となる0-1変数，a_{ij}は C_i中で原子論理式 p_jが現れるとき1，$\neg p_j$が現れるとき-1，いずれも現れないとき0となる係数である。また，$n(a_{i\cdot})$ は C_i中の負のリテラルの個数である。すると，$C_i(\beta_i)$ と (3) 式の間には，$C_i(\beta_i)$ を真とする原子論理式 $p_j(j=1,\cdots,n)$ への真偽の割り当てが存在するとき，またそのときに限り (3) 式を満足する $x_j(j=1,\cdots,n)$ への0,1の割り当てが存在する，という関係が成立する。

次に，0–1変数 x_0を導入して (3) 式の左辺に項βx_0を加えた一次不等式，

$$\beta x_0 + a_{i1}x_1 + a_{i2}x_2 + \cdots + a_{in}x_n \geq \beta_i - n(a_{i\cdot}) \quad \cdots\cdots(4)$$

を考える。ここで，$\beta = \max(\beta_1,\beta_2,\cdots,\beta_m)$ である。この不等式は，x_0の値を1にすると他の0–1変数の値によらず満足される。また，x_0の値を0にすると (3) 式と等しくなる。

以上で述べたβ 節 と一次不等式の関係を，(1) 式のすべてのβ 節 に対して適用し，x_0の値を0にすることを目的とすると，0–1整数計画問題

$$\begin{aligned}&\min \quad x_0\\&\text{sub.to}\ \beta x_0 \boldsymbol{e} + \boldsymbol{A}\boldsymbol{x} \geqq \boldsymbol{b},\\&\qquad x_0 \in \{0,1\}, \mathbf{x} \in \{0,1\}^n\end{aligned} \quad \cdots\cdots\cdots\cdots\cdots\cdots (5)$$

が得られる。ここで，$\boldsymbol{e}$はすべての成分が1であるm列ベクトル，$\boldsymbol{A}$および$\boldsymbol{b}$は，

$$\boldsymbol{A} = \begin{pmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{m1} & \cdots & a_{mn} \end{pmatrix}, \boldsymbol{b} = \begin{pmatrix} \beta_1 - n(a_{1\cdot}) \\ \vdots \\ \beta_m - n(a_{m\cdot}) \end{pmatrix} \quad \cdots\cdots\cdots\cdots\cdots\cdots (6)$$

である。この問題において，x_0の最小値が0でそのとき$\boldsymbol{A}\boldsymbol{x} \geqq \boldsymbol{b}$が成り立つことがわかり，$\beta$ SATと(5)式の問題の間には，『(1)式のβ SATが充足可能であればまたそのときに限り，$x_0 = 0$となる実行可能解が(5)式に存在する。』という関係が成立する。

本論文で提案するIEMBSATは，(5)式の0–1整数計画問題の$x_0 = 0$となる実行可能解を探索するアルゴリズムである。

例えば(2)式のβ SATを0–1整数計画問題として定式化すると，

$$\begin{aligned}&\min \quad x_0\\&\text{sub.to}\ 3x_0 + x_1 \geqq 1,\\&\qquad 3x_0 + x_1 + x_3 + x_4 - x_5 \geqq 1,\\&\qquad 3x_0 + x_1 - x_3 + x_5 \geqq 1,\\&\qquad 3x_0 - x_1 + x_2 + x_5 - x_6 \geqq 1,\\&\qquad 3x_0 - x_2 + x_4 - x_6 \geqq 0,\\&\qquad 3x_0 + x_2 - x_4 + x_6 \geqq 0,\\&\qquad x_j \in \{0,1\}, \forall j \in \{0,\cdots,6\}\end{aligned} \quad \cdots\cdots\cdots\cdots\cdots\cdots (7)$$

となる。

3. β SAT アルゴリズム

<3·1> 用語および諸記号の定義 2つの添え字集合M, Nをそれぞれ，$M = \{1,\cdots,m\}$, $N = \{1,\cdots,n\}$とする。(5)式に対して$x_j \in \{0,1\}, \forall j \in N$を満たすベクトル$\boldsymbol{x} = (x_1,\cdots,x_n)^t$を解と呼ぶ。さらに，(5)式中の制約条件$\beta x_0 \boldsymbol{e} + \boldsymbol{A}\boldsymbol{x} \geqq \boldsymbol{b}$を満足する解を実行可能解と呼ぶ。また$N$の部分集合$J$に対し，$x_j, j \in J$の値を0または1に固定し，$x_j, j \in N - J$は変数のままであるベクトル$\boldsymbol{x}$を(5)式の部分解，もしくは単に部分解と呼ぶ。部分解$\boldsymbol{x}$が与えられたとき，値が0または1に固定されている変数の添え字集合を$J = J(\boldsymbol{x})$と表す。部分解$\boldsymbol{x}$において，値が固定されていない変数$x_j, j \in N - J(\boldsymbol{x})$を$\boldsymbol{x}$に対応する自由変数，もしくは単に自由変数という。自由変数のすべてを0または1に固定したものを完備解といい，特にすべての自由変数を0に固定したものを0–完備解と呼ぶ。

また，実行可能解を探索する途中で調べる第k番目の部分解$\boldsymbol{x}^k$に対して，

$$\begin{aligned}&P_k : \min \quad x_0\\&\qquad \text{sub.to}\ \beta x_0 + \sum_{j \in N - J^k} a_{ij} x_j \geqq b_i^k,\ \forall i \in M^k,\\&\qquad\qquad x_0, x_j \in \{0,1\},\ \forall j \in N - J^k,\\&\text{ただし}, J^k = J(\boldsymbol{x}^k), b_i^k = b_i - \sum_{j \in J^k} a_{ij} x_j^k \text{である。}\end{aligned} \quad \cdots\cdots\cdots\cdots\cdots\cdots (8)$$

を，(5)式の第k部分問題，もしくは単に部分問題と呼ぶ。ここでM^kは元問題の制約式番号集合Mの部分集合である。なお(5)式の問題は，$M^1 = M, J = \phi$である第1部分問題と考えP_1と呼ぶ。

さらに，P_k中の各制約式$i \in M^k$に対し，

$$d_i = |\{j | a_{ij} = 1 \wedge x_j^k = 1 \text{ もしくは } a_{ij} = -1 \wedge x_j^k = 0, j \in J(\boldsymbol{x}^k)\}| \quad \cdots\cdots (9)$$

を，制約式の充足度と呼ぶ。すると各制約式の右辺は，

$$b_i^k = \beta_i - d_i - |\{j | a_{ij} = -1, j \in N - J^k\}| \quad \cdots\cdots (10)$$

となる。

<3·2> 諸定理 IEMBSATでは，以下の諸定理を用いて$x_0 = 0$となる実行可能解の探索を行う。

本論文で示す諸定理は，IEMSATで用いている定理[10]を(5)式のβ SATの定式化に基づき見直したもので，β SATの特別な形であるSATにも適用可能な一般化したものとなっている。特に定理3.4と3.5は，IEMSATで用いている制約式の削除に関する2つの定理を(9)式の充足度を導入して一般化したものである。各定理の証明は付録に示す。

定理 3.1 (実行可能解の存在に関する定理)

部分問題P_kにおいて，$b_i^k \leqq 0, \forall i \in M^k$または$M^k = \phi$ならば，部分解$\boldsymbol{x}^k$の0–完備解は，$x_0 = 0$となる実行可能解である。

定理 3.2 (実行可能解の非存在に関する定理)

部分問題P_kにおいて，$s_i < 0$となる$i \in M^k$が存在すれば，$\boldsymbol{x}^k$の完備解の中に$x_0 = 0$となる実行可能解は存在しない。ただし，

$$s_i = \sum_{j \in N - J^k} \max\{0, a_{ij}\} - b_i^k \quad \cdots\cdots\cdots\cdots (11)$$

である。

定理 3.3 (自由変数の値に関する定理)

$\tilde{\boldsymbol{x}}^k$を$x_0 = 0$となる実行可能な$\boldsymbol{x}^k$の完備解とする．このとき，$i \in R, h \in N - J^k$に対して，$a_{ih} = 1$ならば$\tilde{x}_h^k = 1, a_{ih} = -1$ならば$\tilde{x}_h^k = 0$でなくてはならない。ただし，

$$R = \{i \in M^k | s_i = 0\} \quad \cdots\cdots\cdots\cdots\cdots\cdots (12)$$

である。

系 3.1 (実行可能解の非存在に関する系)

$\boldsymbol{x}^k$の完備解で，$x_0 = 0$となる実行可能解は，$x_j = 1, \forall j \in Q_+^k$; $x_j = 0, \forall j \in Q_-^k$をみたさなくてはならない。したがって，$Q_+^k \cap Q_-^k \neq \phi$ならば$\boldsymbol{x}^k$の完備解で実行可能なものは存在しない。ただし，

$$\begin{aligned}Q_+^k &= \{h \in N - J^k | a_{ih} = 1, i \in R\},\\Q_-^k &= \{h \in N - J^k | a_{ih} = -1, i \in R\}\end{aligned} \quad \cdots\cdots (13)$$

である。

```
S1:   節集合 BSを 0-1 整数計画問題として定式化し, 元問題とする;
S2:   部分問題の集合 P ← φ;
S3:   k← 1;
S4:   P_k← 元問題;
S5:   P_kより削除可能な制約式を削除する (定理 3.5);
S6:   while (P_kの 0-完備解が実行可能でない (定理 3.1)) do begin
S7:       if (P_kに実行可能解が存在する可能性がある (定理 3.2, 系 3.1)) then
S8:           if (値を固定しなければならない自由変数が存在する (定理 3.3)) then begin
S9:               Q ← 部分解にそれらの変数を加えた部分問題 (定理 3.4);
S10:              Q より削除可能な制約式を削除する (定理 3.5);
S11:              P ← P ∪ {Q}
S12:          end else begin
S13:              自由変数を 1 つ選択する (yとする);
S14:              Q_1 ← y = 0 を部分解に加えた部分問題 (定理 3.4);
S15:              Q_2 ← y = 1 を部分解に加えた部分問題 (定理 3.4);
S16:              Q_1および Q_2より削除可能な制約式をそれぞれ削除する (定理 3.5);
S17:              P ← P ∪ {Q_1} ∪ {Q_2}
S18:          end;
S19:      if (P ≠ φ) then begin
S20:          k← k + 1;
S21:          P より部分問題 P_kを 1 つ選択し, P ← P - {P_k} とする
S22:      end else
S23:          BSは充足不能と判定され終了
S24:  end;
S25:  BSは充足可能と判定され終了
```

図 2. 陰的列挙法に基づくβ SAT アルゴリズム.
Fig.2. A β SAT algorithm based on Implicit Enumeration Method.

定理 3.4 (制約式の除去に関する定理 (1))
部分問題 P_kの自由変数の値を固定して新しい部分問題を生成する際に, ある制約式において係数が 1 の自由変数の値を 1 に固定するか, もしくは係数が−1 の自由変数を 0 に固定した結果, その制約式の充足度が元のβ 節 の次数と等しくなれば, 新しい部分問題においてその制約式を取り除くことができる。

定理 3.5 (制約式の除去に関する定理 (2))
部分問題 P_kのある自由変数 x_qに対し,

$$a_{iq} = 1 \text{ もしくは } a_{iq} = 0 \quad \forall i \in M^k \quad \ldots\ldots\ldots (14)$$

または,

$$a_{iq} = -1 \text{ もしくは } a_{iq} = 0 \quad \forall i \in M^k \quad \ldots\ldots\ldots (15)$$

であれば, x_qの値を (14) 式が成り立てば 1, (15) 式が成り立てば 0 に固定することができる。そして固定した結果, 各制約式の充足度が元のβ 節 の次数と等しくなれば, その制約式は取り除くことができる。

<3·3> IEMBSAT 前節で示した定理および系に基づく IEMBSAT を図 2 に示す。

まず, S1 で節集合 BSを (5) 式の問題に定式化する。その際, BS中に含まれる可能性がある $p \vee \neg p$ のトートロジーを含む節を制約式として表すことはできないが, p または$\neg p$ のいずれか一方が真となるので, 充足度を 1 増加させ, pに対応する 0-1 変数を含めないようにする。そして, S2〜S5 で元問題を P_1とするための処理を行い, S6〜S24 の **while** ループで $x_0 = 0$ なる解の探索を行う。

この **while** ループの部分は探索の一般形を示しており, S21 の部分問題の選択方法によって種々の探索戦略が可能となる。例えば, 部分問題として P の中で加えられた順序が最も古い問題を選択すると幅優先探索となり, 最も新しく加えられた問題を選択すると深さ優先探索となる。

<3·4> 例題 IEMBSAT による充足可能性判定の例を示す。例として, (2) 式の基礎節集合を考える。

まず S1 により, この問題を 0-1 整数計画問題として定式化すると (7) 式となる (P_1)。そして S5 で, 定理 3.5 に基づき制約式の削除を試みるが, 削除可能な制約式は存在しないので P_1は (7) 式のままとなる。以下 S6〜S24 で $x_0 = 0$ なる実行可能解の探索を行う。その探索の過程を以下に示す。

(1) $k = 1$ のとき: $\boldsymbol{b}^1 = (b_1^1, b_2^1, b_3^1, b_4^1, b_5^1, b_6^1)^t = (1,1,1,1,0,0)^t$ であるから, 定理 3.1 より 0-完備解は実行可能解ではないことになる。そこで $s_i, \forall i \in M^1$を調べると, $\boldsymbol{s} = (s_1, s_2, s_3, s_4, s_5, s_6) = (0,2,1,1,2,2)$ となり, $s_i \geqq 0, \forall i \in M^1$であるから, 定理 3.2 よりこの部分問題に実行可能解が存在する可能性があることが判明する。またこの結果 $s_1 = 0$ であることから $R = \{1\}$ となり, 定理 3.3 より x_1の値を 1 に固定しなければならないことになる。そこで, $x_1 = 1$ とし, それ以外は変数のままである部分解に対応する部分問題を生成する。この際, $x_1 = 1$ に固定することにより各制約式の充足度が$\boldsymbol{d} = (d_1, d_2, d_3, d_4, d_5, d_6)^t = (1,1,1,0,0,0)^t$ となり, $\beta = (1,2,2,3,2,1)^t$であるから, 定理 3.4 より P_1中の第 1 制約式が削除可能となる (S9)。

この結果生成される部分問題は,

$$
\begin{array}{lll}
\min & x_0 & \\
\text{sub.to} & 3x_0 + x_3 + x_4 - x_5 & \geqq 0, \\
& 3x_0 - x_3 + x_5 & \geqq 0, \\
& 3x_0 + x_2 + x_5 - x_6 & \geqq 2, \\
& 3x_0 - x_2 + x_4 - x_6 & \geqq 0, \\
& 3x_0 + x_2 - x_4 + x_6 & \geqq 0, \\
& x_j \in \{0,1\}, \forall j \in \{0,2,3,4,5,6\} &
\end{array}
\quad \cdots\cdots\cdots\cdots\cdots\cdots (16)
$$

となり，この問題をPに加える(S11)。

次に，$P \neq \phi$であるから新しい部分問題を1つ選択する。ここでは部分問題として選択可能なものは1つであるからP_2として(16)式の問題を選択し，Pからこの問題を取り除く(S21)。なお，P_1からP_2を生成する際に制約式を除去したので$M^2 = \{2,3,4,5,6\}$である。

(2) $k = 2$のとき: P_2において$\boldsymbol{b}^2 = (b_2^2, b_3^2, b_4^2, b_5^2, b_6^2) = (0,0,2,0,0)^t$であるから，定理3.1より0-完備解は実行可能解ではないことになる。そこで$s_i, \forall i \in M^1$を調べると，$\boldsymbol{s} = (s_2, s_3, s_4, s_5, s_6) = (2,1,0,1,2)$となり，$s_i \geq 0$, $\forall i \in M^2$であるから，定理3.2よりこの部分問題に実行可能解が存在する可能性があることが判明する。またこの結果$s_4 = 0$であることから$R = \{2,5,6\}$となり，定理3.3よりx_2とx_5の値を1，x_6の値を0に固定しなければならないことになる。そこで，これらの値を固定し，それら以外は変数のままである部分解に対応する部分問題を生成する。この際，値の固定により各制約式の充足度が$\boldsymbol{d} = (d_2, d_3, d_4, d_5, d_6)^t = (1,2,3,1,1)^t$となり，定理3.4より$P_2$中の第2,3,5制約式が削除可能となる(S9)。さらに削除後の2本の制約式においてx_4の係数がともに1であることから，$x_4 = 1$に固定すると，$\boldsymbol{d} = (d_2, d_5)^t = (2,2)^t$となり，定理3.5より両方の制約式をともに除去できる。この結果生成される部分問題は，

$$
\begin{array}{lll}
\min & x_0 & \\
\text{sub.to} & x_0 \in \{0,1\} &
\end{array}
\quad \cdots\cdots\cdots\cdots\cdots\cdots\cdots (17)
$$

となり，これをPに加える。

次に，$P \neq \phi$であるから新しい部分問題を1つ選択する。ここでは部分問題として選択可能なものは1つであるからP_3として(17)式の問題を選択し，Pからこの問題を取り除く(S21)。なお，P_2からP_3を生成する際に制約式を除去したので$M^3 = \phi$である。

(3) $k = 3$のとき: P_3において$M^3 = \phi$であるから，定理3.1より0-完備解$\boldsymbol{x} = (1,1,0,1,1,0)^t$は実行可能解となる。したがって，(7)式の$\beta$ SATは充足可能と決定され，その際の原子論理式への真偽の割り当ては，$(p_1, p_2, p_3, p_4, p_5, p_6) =$ (真, 真, 偽, 真, 真, 偽) となる。

4. 数値実験と考察

<4·1> 実験方法とテスト問題 本論文で提案するIEMBSATの有効性を検証するためにランダムβ SATとn-Queen問題を用いた数値実験を行う。

β SATを解く方法としては，現在のところ，

- IEMBSATを用いる，
- β SATをSATに変換して，IEMSATなどのSATを解くアルゴリズムを用いる，
- 一般の0-1整数計画問題に対する陰的列挙法(IEM)を用いる，

ことが考えられる。そこで，IEMBSAT，IEMSATならびにIEMのプログラムをIRIS Indigo^2 (主記憶: 96 MB, 85 MIPS) 上でC言語を用いて開発した。この際，探索戦略は縦型探索とした。

ランダムβ SATは，

1. あるリテラルが1つのβ節に出現する確率が等しい，
2. 同じβ節に同一のリテラルは出現しない，
3. 同じβ節に相補対は出現しない，
4. 各β節のリテラルの数は6とする，
5. 各β節の次数は{1,2,3}の値のいずれかの値で，その値は等確率で出現する，

という基準で，リテラル数(30,50,70の3種類)と節数(100,150の2種類)の組合せに対して乱数の種を変えて各10題作成した。このように作成したランダムβ SATは，IEMBSATならびにIEMにとって特に有利な形式とはなっていないと考えられる。さらに，β節を基礎節集合に変換するアルゴリズムを用いて，すべてのランダムβ SATをSATに変換した。この変換で得られるSATはランダムβ SATの生成基準の4.と5.から変換後各節に3つ以上のリテラルが含まれることになるので，IEMSATにとって特に有利な形式の問題とはなっていないと考えられる。

そして，IEMBSATとIEMでランダムβ SATを解く場合と，変換して得られたSATをIEMSATで解く場合を比較するために，CPU時間と実行可能性を調べた部分問題の数(繰り返し回数と呼ぶ)を調べる実験を行なった。さらに，変換して得られたSATはIEMBSATでも解くことができるので，同様の実験を行った。

また，SATアルゴリズムのベンチマーク問題として用いられるn-Queen問題[12]に対して，β SATとして定式化してIEMBSATとIEMで解く場合とSATとして定式化してIEMSATで解く場合を比較した。なお，n-Queen問題のβ SATおよびSATによる定式化方法を付録に示す。

<4·2> 結果と考察 表1に，リテラル数と節数の各組み合せ(L,C)に対し，IEMBSATによる計算時間がもっとも長かった問題(上段)ともっとも短かった問題(中段)の実験結果，ならびに10題の問題を解いた結果の平均(下段)を示す。S/Uの欄は上段および中段がその問題の充足可能性(S:充足可能，U:充足不能)下段が10題中充足可能であった問題数を示す。IEMBSAT，IEMSATおよびIEMの欄において，timeは問題を解くために要したCPU時間，iterは繰り返し回数を表す。また，IEMBSATの欄のtime'は変換して得られたSATをIEMBSATで解くために要したCPU時間，IEMSATの欄のclsは問題変換の結果得られたSATの節数をそれぞれ表す。なお，CPU時間の単位は秒で入出力を除いた時間であり，小数点以下第3桁で四捨

表1. ランダムβ SATの実験結果

Table 1. Computational results for random β SAT problems.

(L,C)	S/U	IEMBSAT			IEMSAT			IEM	
		time	iter	time'	time	iter	cls	time	iter
(30,100)	S	0.27	2820	1.38	1.52	2873	719	0.35	2588
	S	0.01	215	0.16	0.16	346	727	0.04	256
	6	0.14	1496	0.76	0.76	1508	734	0.20	1433
(30,150)	U	0.61	4026	3.04	2.87	4087	954	0.88	4076
	U	0.09	611	0.57	0.53	692	1062	0.13	625
	0	0.19	1252	1.04	1.00	1282	1090	0.27	1255
(50,100)	S	0.38	5315	0.34	0.32	894	724	0.86	5694
	S	0.00	46	0.01	0.02	53	736	0.01	53
	10	0.06	782	0.12	0.12	351	714	0.12	798
(50,150)	S	17.78	159032	59.76	58.92	95158	1137	32.40	153153
	S	0.95	8730	6.36	6.25	11955	1049	1.95	8968
	7	4.61	39961	20.73	20.14	33586	1112	8.49	39705
(70,100)	S	0.02	336	0.32	0.31	1066	753	0.05	302
	S	0.00	72	0.01	0.02	56	845	0.01	72
	10	0.00	105	0.05	0.05	172	726	0.02	104
(70,150)	S	2.10	22871	19.05	18.52	38631	1173	5.02	22996
	S	0.00	46	0.01	0.01	41	1035	0.00	38
	10	0.47	5153	3.46	3.40	7526	1098	1.18	5327

表2 n-Queen問題の実験結果

Tabel 2 Computational results for n-Queen problems.

n	L	IEMBSAT			IEMSAT			IEM	
		C	time	iter	C	time	iter	time	iter
8	64	58	0.00	45	736	0.02	96	0.00	48
20	400	154	0.01	113	12560	0.18	335	0.24	308
30	900	234	0.02	82	43240	0.69	444	0.20	114
50	2500	394	0.03	105	203400	4.76	889	5.00	957
100	10000	794	0.13	302	1646800	91.79	1363	24.52	1141
200	40000	1594	4.17	10698	—	—	—	31.38	389
300	90000	2394	1.53	586	—	—	—	171.01	736
500	250000	3994	5.86	985	—	—	—	2338.00	2902

五入した値である。また，β SAT を SAT に変換した問題を解く場合には変換時間は含めていない。IEMBSAT で変換したSATを解いたときの繰り返し回数は，探索の方法がIEMSAT と同じため，IEMSAT による場合と同じである。

また，n-Queen 問題を解いた結果を表 2 に示す。Q およびLの欄はクイーンの数およびリテラルの数，Cの欄は節の数，time および iter の欄は，CPU 時間および繰り返し回数をそれぞれ表す。CPU時間の単位は秒で入出力を除いた時間である。IEMSAT の欄の — は，計算機のメモリー不足のため問題を解くことができなかったことを意味する。なお，これらの問題はすべて充足可能である。

まず，IEMBSAT と IEMSAT の実験結果を比較すると，

1. ランダムβ SAT を SAT へ変換した場合の節数の増加は7倍程度である，
2. ランダムβ SAT を IEMBSAT で解いた場合，変換した問題を IEMSAT で解く場合に比べて CPU 時間の平均で2〜7倍程度高速である，
3. 変換した問題を IEMBSAT で解く場合は IEMSAT で解く場合に比べて，CPU 時間の平均で最大4%程度遅くなっている，
4. 繰り返し回数はほぼ等しい場合が多いが，(L, C) = (50, 100) の上段の結果のように IEMBSAT のほうが IEMSAT に比べて5倍以上多いことや，逆に (L,C) = (70,100) の上段の結果のように IEMSAT が IEMBSAT に比べて3倍程度多いことがある，
5. n-Queen 問題を SAT として定式化した場合はβ SAT として定式化した場合に比べ，節数が極端に増加する，
6. 計算時間および解くことが可能な問題のサイズが両者で顕著に異なる，

ことがわかる。

1. の節数の増加は，変換方法の組合せ的な性質に起因するもので，変換による節の増加は避けられないものである．なおテスト問題で7倍程度の増加であったことは，1つの基礎節中のリテラル数が6で，次数が {1,2,3} を等確率となるようにしていることから推測できる妥当な値である。

また 2. の結果から，β 節 を基礎節に変換して解くよりはそのまま IEMBSAT で解いたほうが計算効率が良く，変

換した問題をIEMBSATで解いた場合にも計算効率の極端な悪化がないことを示す3.の結果と合せると，IEMBSATはβ節の次数によらず有効な方法と考えられる。

4.の繰り返し回数に関する結果は，組合せ問題に特有の現象と考えられるが，IEMBSATの繰り返し回数がIEMSATに比べて多くても，扱う節の数が少なくて済むためにCPU時間で比べると速くなっている場合が多く，逆にIEMSATの繰り返し回数が多いときには，(L,C) = (70,100)の上段の結果のように15倍のCPU時間を要することがあり，β節を扱うにはIEMBSATが有利であると考えられる。

5.と6.の結果から，n-Queen問題をSATとして定式化して解くよりも，節の数の増加が抑えられるβSATとして定式化してIEMBSATで解いたほうがよいことが明らかである。なお，n=200の場合がn=100や300の場合に比べて繰り返し回数が極端に増加しCPU時間が長くなっていることは，実行可能解を1つ見つけた時点で計算を打ち切るために生じる現象と考えられる。

また，IEMBSATとIEMの実験結果を比較すると，

1. ランダムβ SATの場合の繰り返し回数はほぼ等しいが，n-Queen問題では極端に異なる場合がある，
2. ランダムβSATの場合，すべての問題をIEMBSATが高速に解いており，CPU時間の平均で1.5～2.5倍高速である，
3. n-Queen問題の場合，すべてのサイズをIEMBSATが高速に解いており，問題のサイズが大きくなるとその差が顕著になる，

ことがわかる。

1.の繰り返し回数については，IEMBSATで行なう制約式の除去のために，IEMBSATとIEMにおける変数の選択に違いが生じ，このため繰り返し回数が異なることになる。

また2.のCPU時間については，本実験で用いたランダムβSATでは，問題のサイズがそれほど大きくないために2倍前後の比となっているが，3.の結果と合わせると，問題のサイズが大きくなるとIEMBSATの有効性が顕著に現れると考えられる。

以上の実験結果と考察から，本論文で提案するIEMBSATは，β SATを解く方法としてかなり有効な方法であると考えられる。

5. むすび

本論文では，

1. β節と基礎節の関係の検討，
2. β SATを0-1整数計画問題へ定式化する方法の考案，
3. 2.の定式化に基づきβ SATを解くアルゴリズムIEMBSATの開発，

を行った。本論文で提案したIEMBSATは，SATに対するIEMSATをβ SATに適用できるようにしたものである。

そして，数値実験を行った結果，

- β SATとして定式化される問題は，SATに変換してIEMSATで解くよりもIEMBSATで解くほうがよいこと，
- IEMBSATとIEMを比較すると，問題のサイズによらずIEMBSATはβ SATを高速に解いており，特に問題のサイズが大きくなるとIEMBSATで解くほうが顕著に有利であること，

が明かとなり，本論文で提案したIEMBSATの有効性が検証された。

(平成5年8月24日受付，平成6年4月12日再受付)

文　献

(1) M. Davis and H. Putnam : A Computing Procedure for Quantified Theories, *J. ACM*, Vol.7, pp.201-215 (1960).

(2) C. Chang and R. C. Lee: *Symbolic Logic and Mechanical Theorem Proving*, Academic Press, Orland (1973) (長尾, 辻井訳:コンピュータによる定理の証明, 日本コンピュータ協会, 東京 (1983)).

(3) R. G. Jeroslow: *Logic-Based Decision Support: Mixed Integer Model Formulation*, North-Holland, Amsterdam (1989).

(4) J. N. Hooker: A Quantitative Approach to Logical Inference, *Decision Support Systems*, Vol.4, pp.45-69 (1988).

(5) J. N. Hooker: Resolution vs. Cutting Plane Solution of Integer Problems: Some Computational Experience, Operations Research Letters, Vol. 7, No. 1, pp.1-7(1988)

(6) J. N. Hooker: Generalized Resolution and Cutting Planes, *Ann. of Oper. Res.*, Vol. 12, pp.217-239(1988).

(7) J. N. Hooker and C. Fedjki: Branch-and-Cut Solution of Inference Problems in Propositional Logic, *Ann. of Mathematics and Artificial Intelligence*, Vol.1, pp.123-139(1990).

(8) A. P. Kamath, N. K. Karmarkar, K. G. Ramakrishman and M. G. C. Resende: Computational Experience with an Interior Point Algorithm on Satisfiability Problem, *Ann. of Oper. Res.*, Vol.25, pp.43-58 (1990).

(9) 今井, 大前: 0-1整数計画問題における実数ベクトルの整数ベクトルへの丸め, 情報処理学会研究会資料, 92-AL-27-4 (1992).

(10) 大柳, 山本, 大内 : 陰的列挙法に基づくSATアルゴリズム, 情報処理学会論文誌, Vol.34, No.12, pp.2464-2473(1992).

(11) E. Balas: An Additive Algorithm for Solving Linear Programs with Zero-One Variables, *Oper. Res.*, Vol.13, pp.517-546 (1965).

(12) B. Selman, H. Levesque and D. Mitchell: A New Method for Solving Hard Satisfiability Problems, *Proc. of AAAI'92*, pp.440-446 (1992).

付　録

定理 2.1 の証明　　β 節 $l_1 \vee \cdots \vee l_n(\beta)$ が充足可能とする。するとこのβ節において少なくともβ個のリテラルが真であることになる。一方，${}_nC_{\beta-1}$個の各基礎節は，それぞれ元の n 個のリテラルから$\beta-1$個のリテラルを取り除いて作られている。したがって，各基礎節には少なくとも1個の真のリテラルが存在することになる。よって，変換アルゴリズムによって生成される${}_nC_{\beta-1}$個の基礎節はすべて充足可能である。

逆に変換アルゴリズムによって生成される${}_nC_{\beta-1}$個の基礎節がすべて充足可能とする。すると各基礎節に少なくとも1つは真となるリテラルが存在することになる。そこで，${}_nC_{\beta-1}$個の基礎節からある節を一つ取りだし，その節で真

となっているリテラルをl'_1とする。次にl'_1を含まない基礎節 ($_{n-1}C_{\beta-2}$個存在する) を考え，そこからある節を一つ取りだし，その節で真となっているリテラルをl'_2とする。そして，l'_1とl'_2を含まない基礎節 ($_{n-2}C_{\beta-3}$個存在する) を考える。このような手続きを$\beta-1$回繰り返すと，真となっているリテラル$l'_1,\cdots,l'_{\beta-1}$と，リテラル$l'_1,\cdots,l'_{\beta-1}$を含まない基礎節が得られることになる。ここで，リテラル$l'_1,\cdots,l'_{\beta-1}$を含まない節は，$_{n-\beta-1}C_{\beta-\beta}=1$個となる。この基礎節も充足可能であるから，リテラル$l'_1,\cdots,l'_{\beta-1}$以外の$n-\beta+1$個のリテラルのうち少なくとも1つは真でなければならない。したがって，$_nC_{\beta-1}$個の基礎節がすべて充足可能とすると，少なくともβ個のリテラルが真となっていることになり，β節は充足可能となる (証明終)。

定理3.1の証明　部分問題P_kの制約条件は(8)式の通りで，$\boldsymbol{x}^k$の0-完備解に対し，$\beta x_0+\sum_{j\in N-J^k}a_{ij}x_j=\beta x_0\geqq 0,\forall i\in M^k$, である．したがって，$b_i^k\leqq 0,\forall i\in M^k$であればすべての制約式を満足する。つまり，$\boldsymbol{x}^k$の0-完備解は$x_0=0$となる$P_k$の実行可能解となる。また$M^k=\phi$のときは，制約式は存在しないので自明である (証明終)。

定理3.2の証明　$\boldsymbol{x}^k$の完備解の中に実行可能解$\tilde{\boldsymbol{x}}^k$が存在するとする。すると，元問題(5)式の制約条件は，

$$\begin{aligned}&\beta x_0+\sum_{j\in N}a_{ij}\tilde{x}_j^k\\&=\beta x_0+\sum_{j\in J^k}a_{ij}\tilde{x}_j^k+\sum_{j\in N-J^k}a_{ij}\tilde{x}_j^k\\&=\beta x_0+b_i+\sum_{j\in N-J^k}a_{ij}\tilde{x}_j^k-b_i^k\\&\leqq\beta x_0+b_i+s_i\end{aligned}\quad\cdots\cdots\cdots(18)$$

となる．ここで，$x_0=0$とすると，$s_i<0$なる制約式に対して，

$$\beta x_0+\sum_{j\in N}a_{ij}\tilde{x}_j^k\leqq x_0+b_i+s_i<b_i\quad\cdots\cdots\cdots(19)$$

となり，$\tilde{\boldsymbol{x}}^k$が実行可能解であることと矛盾する (証明終)。

定理3.3の証明　$\tilde{\boldsymbol{x}}^k$に対して，元問題(5)式の制約条件は，

$$\begin{aligned}&\beta x_0+\sum_{j\in N}a_{ij}\tilde{x}_j\\&=\beta x_0+\sum_{j\in J^k}a_{ij}\tilde{x}_j^k+\sum_{j\in N-J^k-\{h\}}a_{ij}\tilde{x}_j^k+a_{ih}\tilde{x}_h^k\\&\leqq\beta x_0+b_i-b_i^k+\sum_{N-J^k-\{h\}}\max(0,a_{ij})+a_{ih}\tilde{x}_h^k\\&=\beta x_0+b_i+s_i-\max(0,a_{ih})+a_{ih}\tilde{x}_h^k\\&=b_i+a_{ih}\tilde{x}_h^k-\max(0,a_{ih})\end{aligned}\quad\cdots\cdots\cdots(20)$$

となる。したがって，$\tilde{\boldsymbol{x}}^k$が実行可能解であるためには，$a_{ih}\tilde{x}_h^k-\max(0,a_{ih})\geqq 0$となる必要がある。よって，$a_{ih}=1$ならば$\tilde{x}_h^k=1$, $a_{ih}=-1$ならば$\tilde{x}_h^k=0$，でなくてはならない (証明終)。

系3.1の証明　定理3.3より自明である (証明終)。

定理3.4の証明　部分問題P_kの制約式は，一般に，

$$\beta x_0+\sum_{j\in N_{i+}^k}x_j-\sum_{j\in N_{i-}^k}x_j\geqq\beta_i-d_i-|N_{i-}^k|\quad\cdots\cdots\cdots(21)$$

という形式となっている。ここで，N_{i+}^kおよびN_{i-}^kは係数がそれぞれ1および-1である自由変数の添え字集合である。まず，係数が1の自由変数 (添え字をqとする) の値を1に固定する場合を考えると，充足度の定義(9)式より，固定後の制約式の充足度は1増加する。仮定より固定後その制約式の充足度と基礎節の次数は等しくなることから(21)式の右辺は$1-|N_{i-}^k|$である。一方，固定後の(21)式の左辺は，$x_0=0,x_j=0(\forall j\in N_{i+}^k-\{q\}),x_q=1,x_j=1(\forall j\in N_{i-}^k)$，のとき最小値$1-|N_{i-}^k|$となり，この値は右辺と一致する。また，係数が-1の自由変数 (添え字をqとする) の値を0に固定する場合を考えると，同様に充足度は1増加することになる。したがって，(21)式の右辺は$1-|N_{i-}^k|$である。一方(21)式の左辺は，$x_0=0,x_j=0(\forall j\in N_{+}^k),x_q=0,x_j=1(\forall j\in N_{-}^k-\{q\})$，のとき最小値$-(|N_{-}^k-\{k\}|)=1-|N_{-}^k|$となり，この値は右辺と一致する。

以上より，値の固定後その制約式は他の自由変数の値によらず常に成り立つことになり，新しい部分問題において取り除くことができる (証明終)。

定理3.5の証明　(14)式が成り立つ場合，自由変数x_qを含むすべての制約式でその係数は1である。したがって，x_qを含むすべての制約式の充足度は，x_qの値を1に固定することで一様に1増加させることができる。同様に(15)式が成り立つ場合，自由変数x_qを含むすべての制約式でその係数は-1である。したがって，x_qを含むすべての制約式の充足度は，x_qの値を0に固定することで一様に1増加させることができる。そして固定後の充足度が基礎節の次数と等しければ，定理3.4の証明で示した通り，その制約式は新しい部分問題において取り除くことができる (証明終)。

n-Queen問題の定式化　n-Queen問題は，$n\times n$の正方マス目上にn個のクイーンをお互いに取合わないように配置する問題である。SATおよびβ SATによる問題の定式化では，クイーンを置くとき真，置かないとき偽とする，n^2個のマス目に対応した原子論理式を用いる。

(1) SATとしての定式化: 条件

1. 各行に少なくとも1個のクイーンを置く，
2. どの2つのクイーンも，同じ行，同じ列，対角位置には置かない，

を基礎節として表す。例えば，4-queen問題において第1行の原子論理式が$p_{11},p_{12},p_{13},p_{14}$のとき，条件の1.は基礎節$p_{11}\vee p_{12}\vee p_{13}\vee p_{14}$として表すことができる。また，2.の同じ行に関する条件は，$\neg p_{11}\vee\neg p_{12}$, $\neg p_{11}\vee\neg p_{13}$, $\neg p_{11}\vee\neg p_{14}$, $\neg p_{12}\vee\neg p_{13}$, $\neg p_{12}\vee\neg p_{14}$, $\neg p_{13}\vee\neg p_{14}$の6つの基礎節として表される。同様の定式化を，他の行，列および対角位置について行い，4-queen問題をSATとして表現する。

(2)β SATとしての定式化: 条件

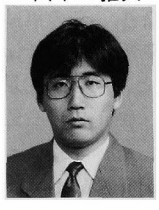
1. 各行と各列にちょうど1個のクイーンを置く，
2. 対角位置に高々1個のクイーンを置く，

をβ節として表す。これらの条件は，<2.1>節で示したようにβ節として表すことができる。

大柳　俊夫

(非会員) 1962 年生。1985 年北海道大学工学部電気工学科卒業。札幌医科大学保健医療学部講師。システム工学，遺伝情報解析の研究に従事。情報処理学会，日本 OR 学会，生物物理学会，ACM 各会員。

山本　雅人



(非会員) 1968 年生。1991 年北海道大学工学部情報工学科卒業。1993 年北海道大学大学院工学研究科情報工学専攻修士課程修了。現在，同博士課程在学中。システム工学の研究に従事。情報処理学会，人工知能学会各会員。

大内　東

(正員) 1945 年生。1974 年北海道大学大学院工学研究科博士課程修了。工学博士。北海道大学工学部情報工学科教授。システム情報工学，応用人工知能システム，医療システムの研究に従事。情報処理学会，人工知能学会，電子情報通信学会，計測自動制御学会，日本 OR 学会，医療情報学会，病院管理学会，IEEE-SMC 各会員。