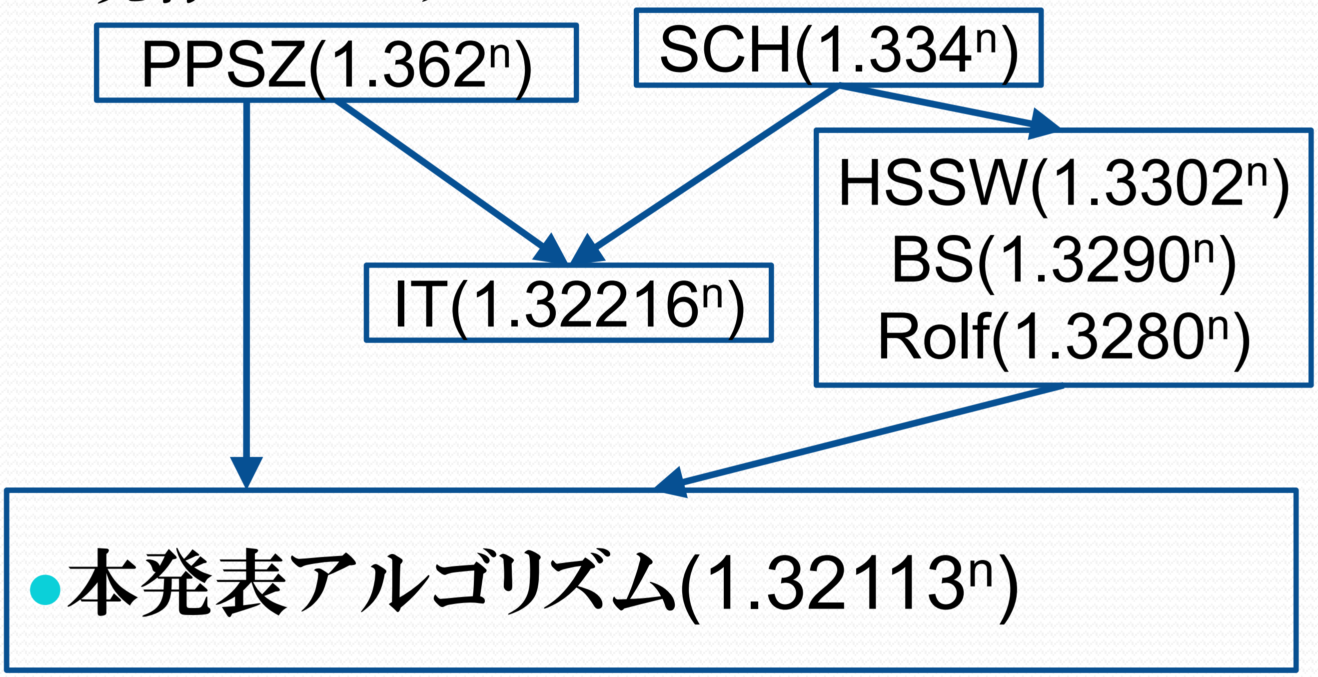
玉置 卓 (京都大学)

Joint work with: 岩間 一雄, 脊戸 和寿, 高井 唯史
ERATO湊離散構造処理系＋NEO 合同研究会, 2010/7/30

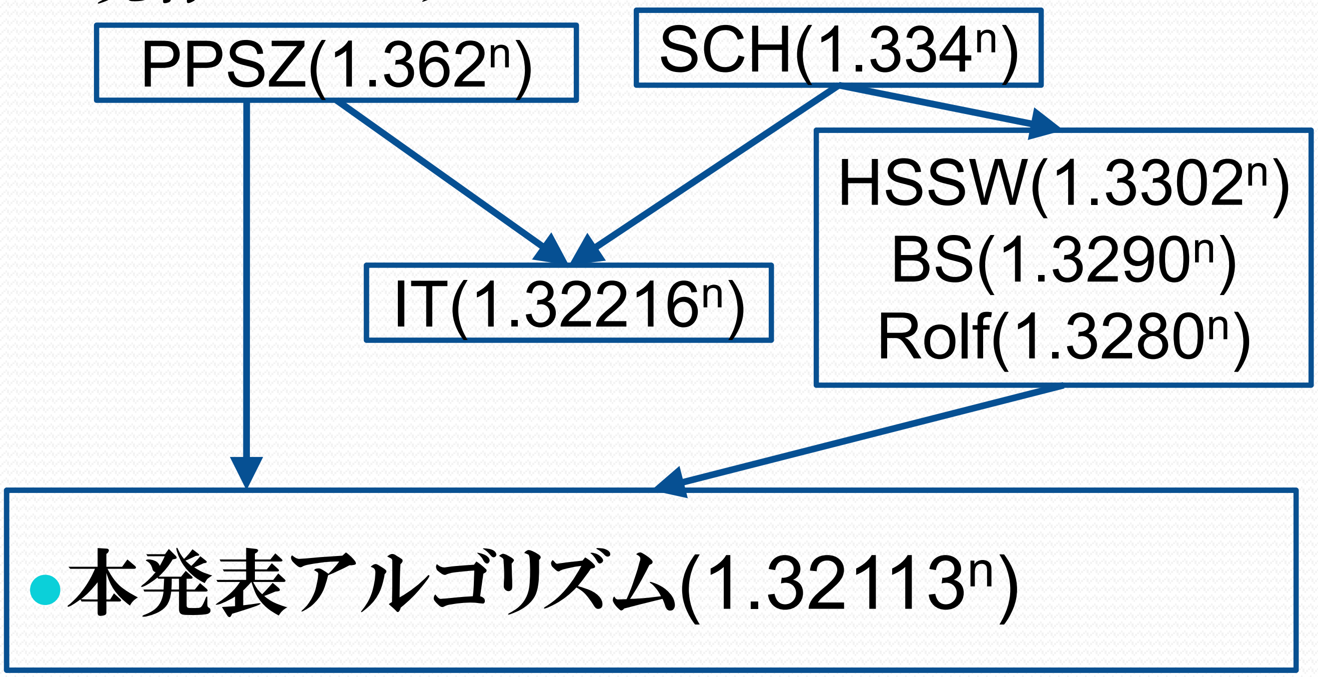
発表の流れ

- 背景と結果
  - 3SATの定義、既存結果と改良結果
- 既存アルゴリズム



発表の流れ

- 背景と結果
  - 3SATの定義、既存結果と改良結果
- 既存アルゴリズム



3-SAT(3充足可能性問題)

3CNF論理式(リテラルの数が3以下の節のみ)

$x1=1, x2=0, x3=0, x4=1, x5=0$

$$F=(x1 \lor x2 \lor x3)(x2 \lor \overline{x3} \lor x5)(\overline{x1} \lor \overline{x4} \lor x5)$$

同時に全て充足出来るか？

最も基本的なNP完全問題

3-SAT(3充足可能性問題)

3CNF論理式(リテラルの数が3以下の節のみ)

x1＝1,x2＝0,x3＝0,x4＝1,x5＝0

$$F=\underbrace{(\overset{1}{x1}\lor\overset{0}{x2}\lor\overset{0}{x3})}_{\text{true}}\underbrace{(\overset{0}{x2}\lor\overset{\overline{0}}{\overline{x3}}\lor\overset{0}{x5})}_{\text{true}}\underbrace{(\overset{\overline{1}}{\overline{x1}}\lor\overset{\overline{1}}{\overline{x4}}\lor\overset{0}{x5})}_{\text{false}}$$

同時に全て充足出来るか？

3-SAT(3充足可能性問題)

3CNF論理式(リテラルの数が3以下の節のみ)

$$x1=\overset{0}{\cancel{1}},\ x2=\overset{1}{\cancel{0}},\ x3=0,\ x4=1,\ x5=0$$

$$F=\underbrace{(\overset{1}{x1}\vee\overset{0}{x2}\vee\overset{0}{x3})}_{\text{true}}\underbrace{(\overset{0}{x2}\vee\overset{\overline{0}}{\overline{x3}}\vee\overset{0}{x5})}_{\text{true}}\underbrace{(\overset{\overline{1}}{\overline{x1}}\vee\overset{\overline{1}}{\overline{x4}}\vee\overset{0}{x5})}_{\text{false}}$$

同時に全て充足出来るか？

3-SAT(3充足可能性問題)

3CNF論理式(リテラルの数が3以下の節のみ)

$$x1=\overset{0}{\cancel{1}},\ x2=\overset{1}{\cancel{0}},\ x3=0,\ x4=1,\ x5=0$$

$$F=\underbrace{(\overset{\overset{0}{\cancel{1}}}{x1}\vee\overset{\overset{1}{\cancel{0}}}{x2}\vee\overset{0}{x3})}_{\text{true}}\underbrace{(\overset{\overset{1}{\cancel{0}}}{x2}\vee\overset{\overline{0}}{\overline{x3}}\vee\overset{0}{x5})}_{\text{true}}\underbrace{(\overset{\overset{\overline{0}}{\cancel{\overline{1}}}}{\overline{x1}}\vee\overset{\overline{1}}{\overline{x4}}\vee\overset{0}{x5})}_{\text{true}}$$

同時に全て充足出来るか？

充足可能！！

SATのアルゴリズム

- 2^n 時間は自明 (総当たり)
- 分枝限定法 (バックトラック法)
  - 決定性
  - 良い分岐ルール

実用的にはヒューリスティック
理論的解析は困難

- 局所探索法
  - 確率的
  - 良い初期値と近傍

分枝限定法

$$F$$

$$F' = F \mid x_1 = 0 \qquad F'' = F \mid x_1 = 1$$

$$F''' = F' \mid x_2 = 0 \qquad F'''' = F' \mid x_2 = 1$$

変数に0,1を代入して再帰的に判定

分枝限定法

変数の少ない式に分割
充足性は保存

$$F \in \mathrm{SAT}$$
$$\Updownarrow$$
$$(F \mid x_1 = 0) \in \mathrm{SAT} \ \text{ or } \ (F \mid x_1 = 1) \in \mathrm{SAT}$$

計算時間

$$T(n) = 2T(n-1) = \cdots = 2^n$$

分枝限定法

$(x_1 \vee x_2 \vee x_3)$ に注目すると

$$F \in \mathrm{SAT}$$
$$\Updownarrow$$
$$(F \mid x_1 = 1) \in \mathrm{SAT} \ \text{ or } \ (F \mid x_1 = 0, x_2 = 1) \in \mathrm{SAT}$$
$$\text{or } \ (F \mid x_1 = 0, x_2 = 0, x_3 = 1) \in \mathrm{SAT}$$

計算時間

$$T(n) = T(n-1) + T(n-2) + T(n-3) = \cdots = 1.839^n$$

Autarkness [MS85]

どちらかが成立　ϕ: 部分割り当て

1. $F|\phi$ は2-clauseを持つ
2. $F \in \mathrm{SAT} \Leftrightarrow F|\phi \in \mathrm{SAT}$

計算時間

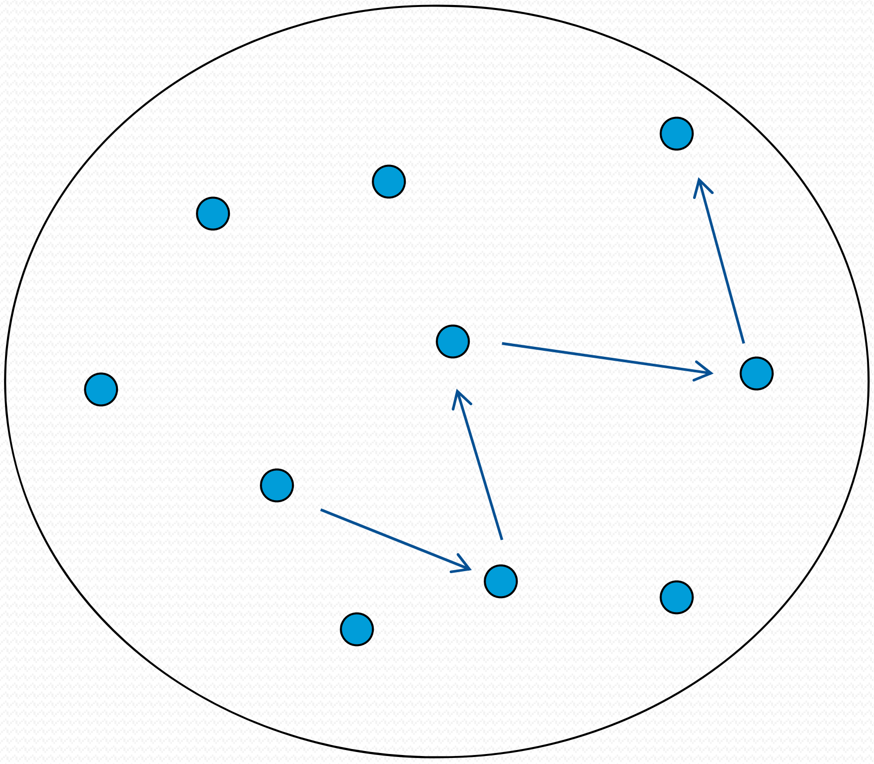
$$T(n) = \max\left\{\begin{matrix} T(n-1)+T(n-2), \\ T(n-2)+T(n-3)+T(n-4) \end{matrix}\right\}$$

$$= 1.618^n$$

局所探索法

- 適当な初期割り当て
- 近傍の最も良い点への移動
  を繰り返す

探索空間



割り当て

(0,0,0,0,0,0) ~ (1,1,1,1,1,1)

2^n 通り

近傍, 最も良い, の定義例

- 貪欲法
  - 近傍: 現在の割り当てを1ビットフリップ
  - 評価関数: 充足される節の数
- ランダムウォーク
  - 近傍: 現在の割り当てで充足されていない節の変数をひとつフリップ
  - 評価関数: ランダムに選ぶ

本研究の目標

- 3SAT の最悪時計算時間 c^n をできるだけ小さく
  (moderately exponential time algorithm)
  c=1.618 ?, c=1.4 ?, c=1.0001 ?
- $2^{o(n)}$ (例えば $2^{n/\log n}$) では解けないと信じられている

改良の歴史

c	type	年		
2	det.		自明な全探索	
1.839	det.	1979	B. Monien and E. Speckenmeyer	
1.769	det.	1983	E. Dantsin	
1.618	det.	1985	B. Monien and E. Speckenmeyer	
1.588	rand.	1997	R. Paturi et al.	
1.579	de			
1.497	de			
1.476	de			
1.473	de			
1.465	det.	2008	D. Scheder	
1.362	rand.	1998	R. Paturi et al.	PPSZアルゴリズム
1.334	rand.	1999	U. Schoning	SCHアルゴリズム
1.3302	rand.	2002	T. Hofmeister et al.	HSSWアルゴリズム
1.32971	rand.	2003	D. Rolf	Rolf1アルゴリズム
1.3290	rand.	2003	S. Baumer and R. Schuler	BSアルゴリズム
1.32793	rand.	2003	D. Rolf	Rolf2アルゴリズム
1.3238	rand.	2004	K. Iwama and S. Tamaki	
1.32266	rand.	2004	S. Tamaki	
1.32216	rand.	2005	D. Rolf(現在最速)	

SCH改良

SCH+PPSZ

1.32216^n

$\Downarrow$

1.32113^n

:本発表の結果

HSSW+PPSZ

関連結果: *k*-SATの上界 (一部)

- 一般の*k*

$$2^{n(1-1/k)}\ \text{[Paturi et al.,97]}$$

$$\left(\frac{2(k-1)}{k}\right)^n\ \text{[Shoening,99]}$$

- CNF-SAT ($k = n$) m: #clause

$$2^{n-\frac{n}{1+c\log n}}\ \text{[Schuler,03]}\ (m < n^c)$$

$$2^{n(1-\varepsilon_c)}\ \text{[Arvind, Schuler,03]}\ (m < cn)$$

関連結果: SAT の下界（チューリングマシン）

- $\mathrm{SAT} \notin \mathrm{DTISP}(n^c, n^d)$ [Fortnow, van Melkebeek 2000]
  $c < (\sqrt{5}+1)/2, 1 > d > 0 : \mathrm{const.}$
  - すべてのテープにランダムアクセス

- $\mathrm{SAT} \notin \mathrm{DTIME}(n^a)$ [van Melkebeek, Raz 2004]
  $a < \sqrt{(d+2)(d+1)}$
  - 入力テープにランダムアクセス
  - ワークテープ: *d*次元, 1本のみ

関連結果: 一般化SAT

- #SAT: 充足解の個数を求める (2-CNFでも #P完全)
  - k-CNF, kは定数, の場合 2^n より速く解ける

- MAX-SAT: 充足節数を最大化 (2-CNFでもNP困難)
  - 2-CNF の場合 2^n より速くとける
  - 3-CNF は open

- QBF-SAT: 変数にquantifier が付く

$$\exists x \forall y \exists z \ldots f(x, y, z, \ldots) = 1?$$

(2-CNFはP, 3-CNFだとPSPACE完全)
  - 3-CNF は2^n より速く解けないと予想されている

解けると, 節の長さに制限のないSAT が解けてしまう

本題に入る前に:
乱択アルゴリズムの計算時間とは

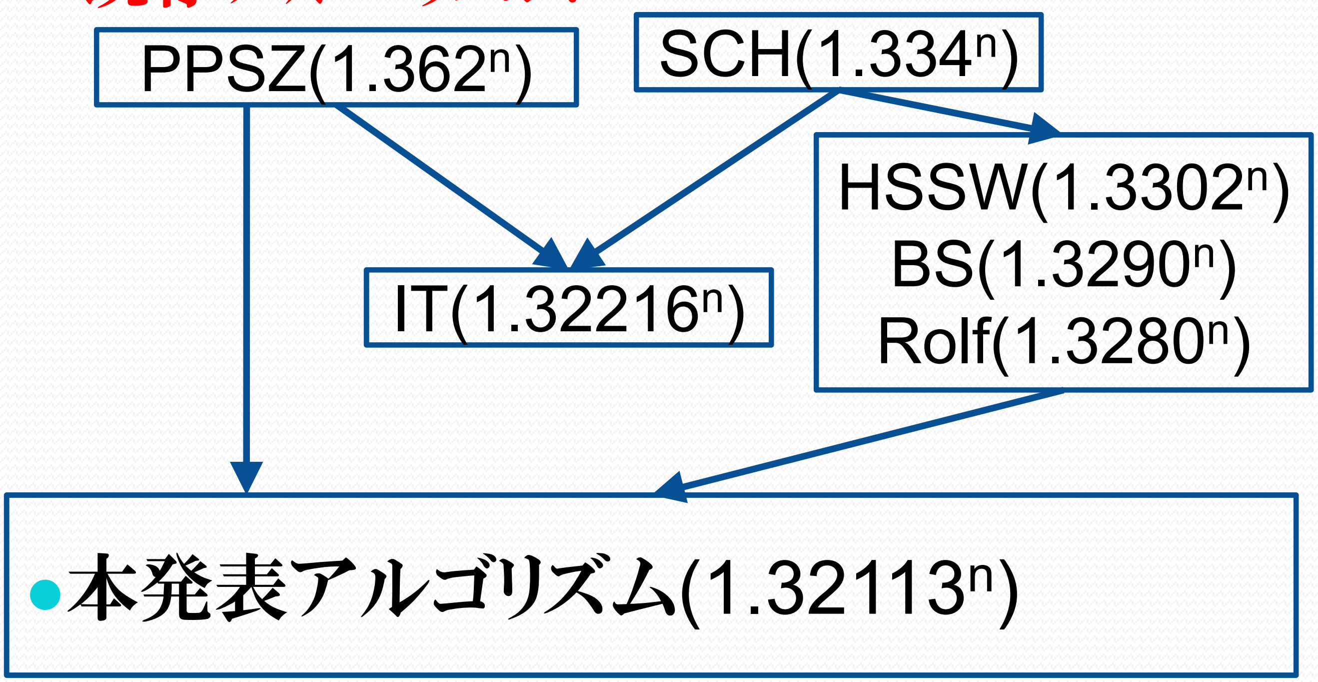
- 片側誤りの探索アルゴリズム ALG
  - F が充足可能: 確率p 以上で充足解を見つける
    - pは指数的に小さい (c^{-n}, $c>1$) ことが多い
  - F が充足不能: 常に充足不能と答える
  - ALG は多項式時間で終了

- $1/p \times T$ 回探索を行う
  失敗確率 $\leq (1-p)^{T/p} \leq e^{-T}$ (e は自然対数2.718…)

- ALG の計算時間は $O(1/p)$ と定義

発表の流れ

- ✓背景と結果
  - 3SATの定義、既存結果と改良結果
- 既存アルゴリズム



PPSZアルゴリズム[PPSZ98]

1. {1,2,…,n}のランダムな順列πを生成する。
2. $\{0,1\}^n$ のランダムな割り当て y を生成する。
3. $i=1$から nまで以下を繰り返す。

   もし　F　が単一節 $x_{\pi(i)}$ を含むなら　$z_{\pi(i)}=1$

   もし　F　が単一節 $\overline{x}_{\pi(i)}$ を含むなら　$z_{\pi(i)}=0$

   上記に当てはまらないなら $z_{\pi(i)}=y_{\pi(i)}$

   $F=F_{|x_{\pi(i)}=z_{\pi(i)}}$ とする。
4. Z がF を満たすならtrueを返す。

※単一節とは (x_i)のようにその変数だけからなる節

PPSZアルゴリズム

$F=(x1\vee\overline{x2}\vee\overline{x3})(x2\vee\overline{x3})(\overline{x1}\vee x3)(x1\vee x3)$

ランダムに順列 $\pi=(3,1,2)$ を取り、
ランダムに割当 $y=\{1,0,1\}$ と取ると

$F=(x1\vee\overline{x2}\vee\overline{x3})(x2\vee\overline{x3})(\overline{x1}\vee x3)(x1\vee x3)$

$z=\{\ ,\ ,1\}$

PPSZアルゴリズム

$F=(x1 \lor \overline{x2} \lor \overline{x3})(x2 \lor \overline{x3})(\overline{x1} \lor x3)(x1 \lor x3)$

ランダムに順列 $\pi=(3,1,2)$ を取り、
ランダムに割当 $y=\{1,0,1\}$ と取ると

$F=(x1 \lor \overline{x2} \lor \quad)(x2 \lor \quad)$

$z=\{1, \quad ,1\}$

PPSZアルゴリズム

$F=(x1 \vee \overline{x2} \vee \overline{x3})(x2 \vee \overline{x3})(\overline{x1} \vee x3)(x1 \vee x3)$

ランダムに順列 $\pi = (3,1,2)$ を取り、
ランダムに割当 $y = \{1,0,1\}$ と取ると

$F=$ $(x2 \vee \quad)$

$z=\{1,1,1\}$

zをFに代入
$F=(1 \vee 0 \vee 0)(1 \vee 0)(0 \vee 1)(1 \vee 1)$
zはFの解である

PPSZアルゴリズムとcritical-clause

$$F=(x1\vee \overline{x2}\vee \overline{x3})(x2\vee \overline{x3})(\overline{x1}\vee x3)(x1\vee x3)$$

$$(1\quad 0\quad 0)(1\quad 0)(0\quad 1)(1\quad 1)$$

- critical-clause

解を割り当てたときに一つの変数のみtrueの節

PPSZではこの節を利用して解を求めている。

(x2, x3 を先に求めれば x1 は自動的に決定)

PPSZの性質

- critical clause を多く含む式に対して高速
  - 充足解の個数が少ない場合に対応
  - Unique 3-SAT (解が1個) だと 1.3071^n
  - F: 少数の変数を固定するとunique 3SAT, でも高速

$$F(x)=1 \quad \text{for} \quad x \in S = \{00000, 10001, 10010, 10100, 11000\}$$

$$\Downarrow \quad x_1 = 0$$

$$F|_{x_1=0}(x)=1 \quad \text{for} \quad \text{only} \quad x = 0000$$

SCHアルゴリズム[Sch99]

局所探索アルゴリズム

入力された割当yの変数ビットを反転していく

F＝(x1∨x2∨x3)(x3∨x4∨x5)(x4∨x6∨x7)

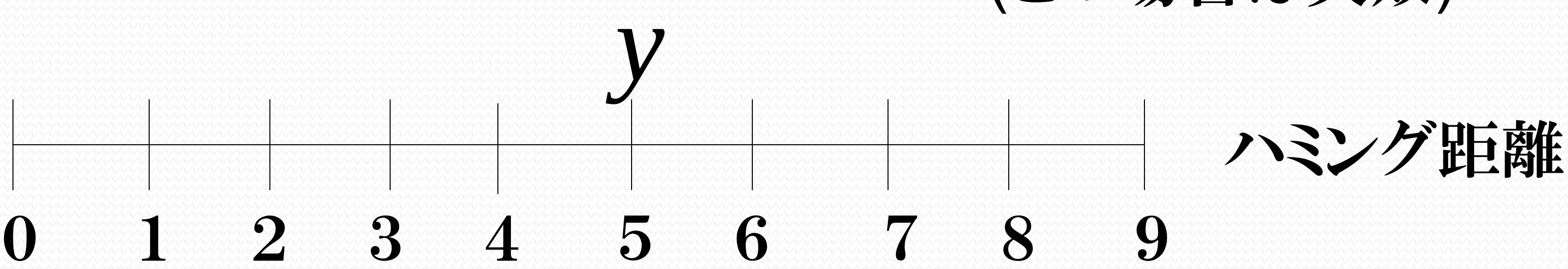
(x1∨x5∨x9)(x5∨x8∨x10)(x10∨x11∨x12)

y={0 1 0 0 1 0 0 0 0 }

ビットを変えるごとに、ハミング距離の増減を繰り返しながら解を目指す

一定数回繰り返して、解に到達したら成功。到達しなかったら失敗

(この場合は失敗)



z={1}^{n}

SCHアルゴリズム[Sch99]

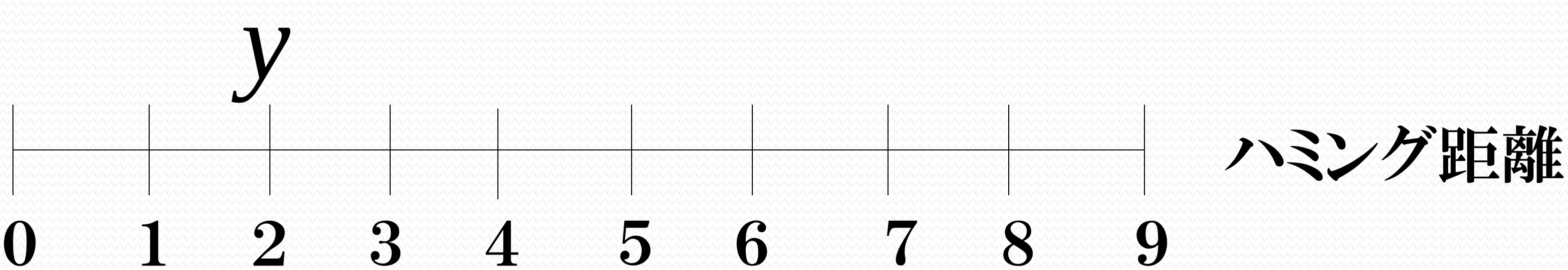
局所探索アルゴリズム

入力された割当yの変数ビットを適当に選び反転する
(充足されていない節からランダムに1個選ぶ)

y={ 0 1 1 0 1 1 0 0 1 }

成功！！

解に近い値からスタートすれば解を得やすくなる



$z=\{1\}^n$

SCHの成功確率[Sch99]

補題　初期値が解から j 離れているとき
　SCHが解を見つける確率 q_j は

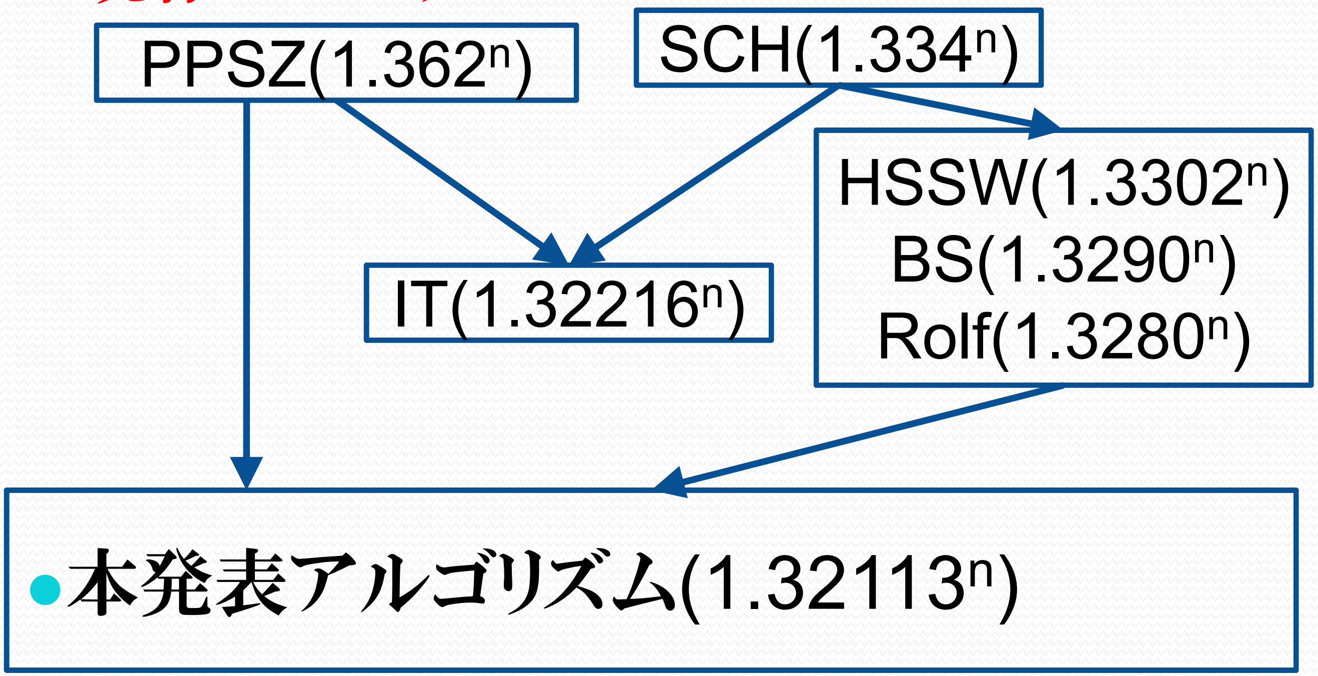
$$q_j \geq (1/2)^j$$

定理　初期値をランダムに選んだとき
　SCHが解を見つける確率 p は

$$p \geq (3/4)^n$$

発表の流れ

- ✓背景と結果
  - 3SATの定義、既存結果と改良結果
- 既存アルゴリズム



既存アルゴリズム

一様ランダム
な割当y

↓

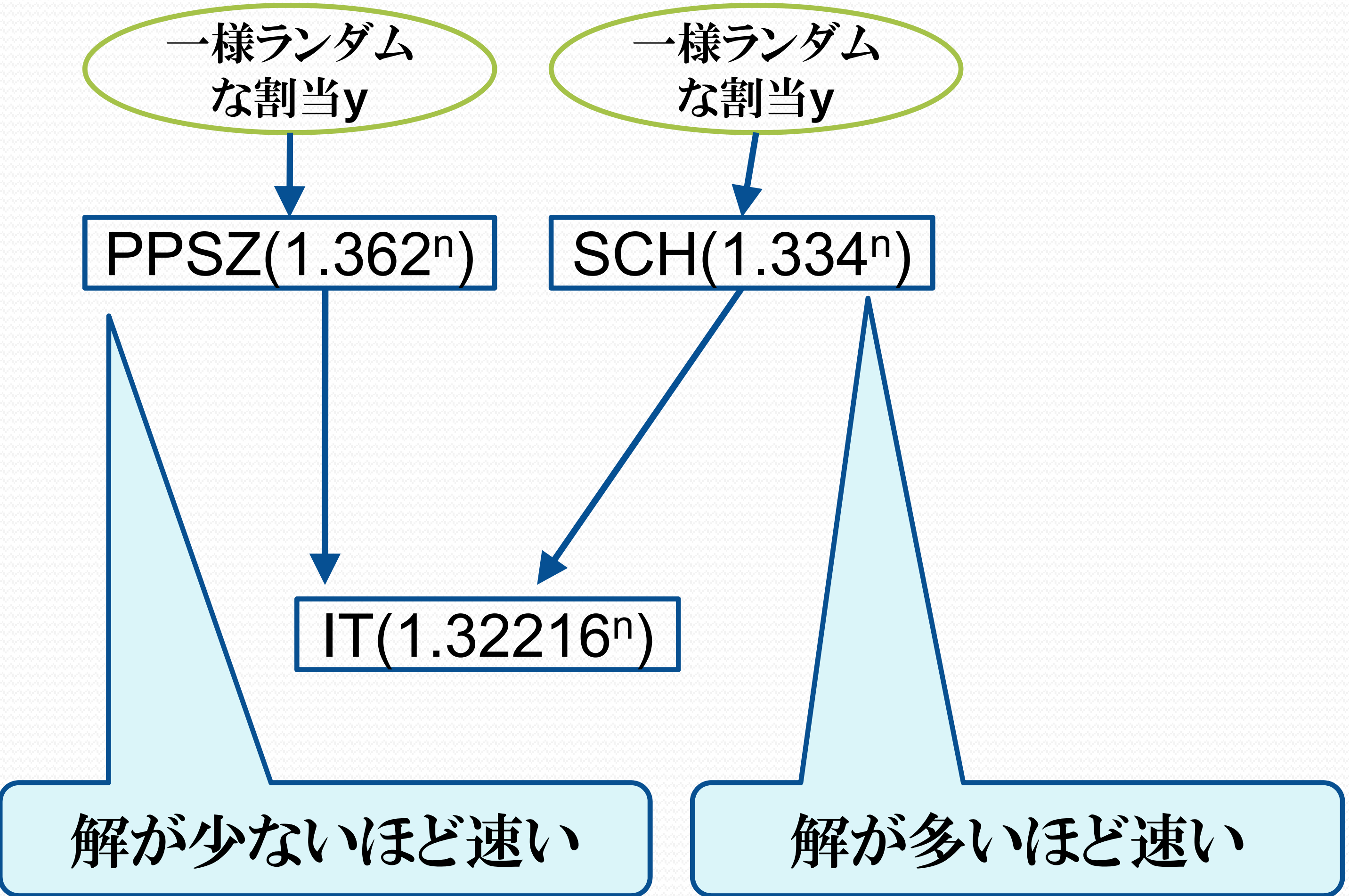
PPSZ(1.362^n)

一様ランダム
な割当y

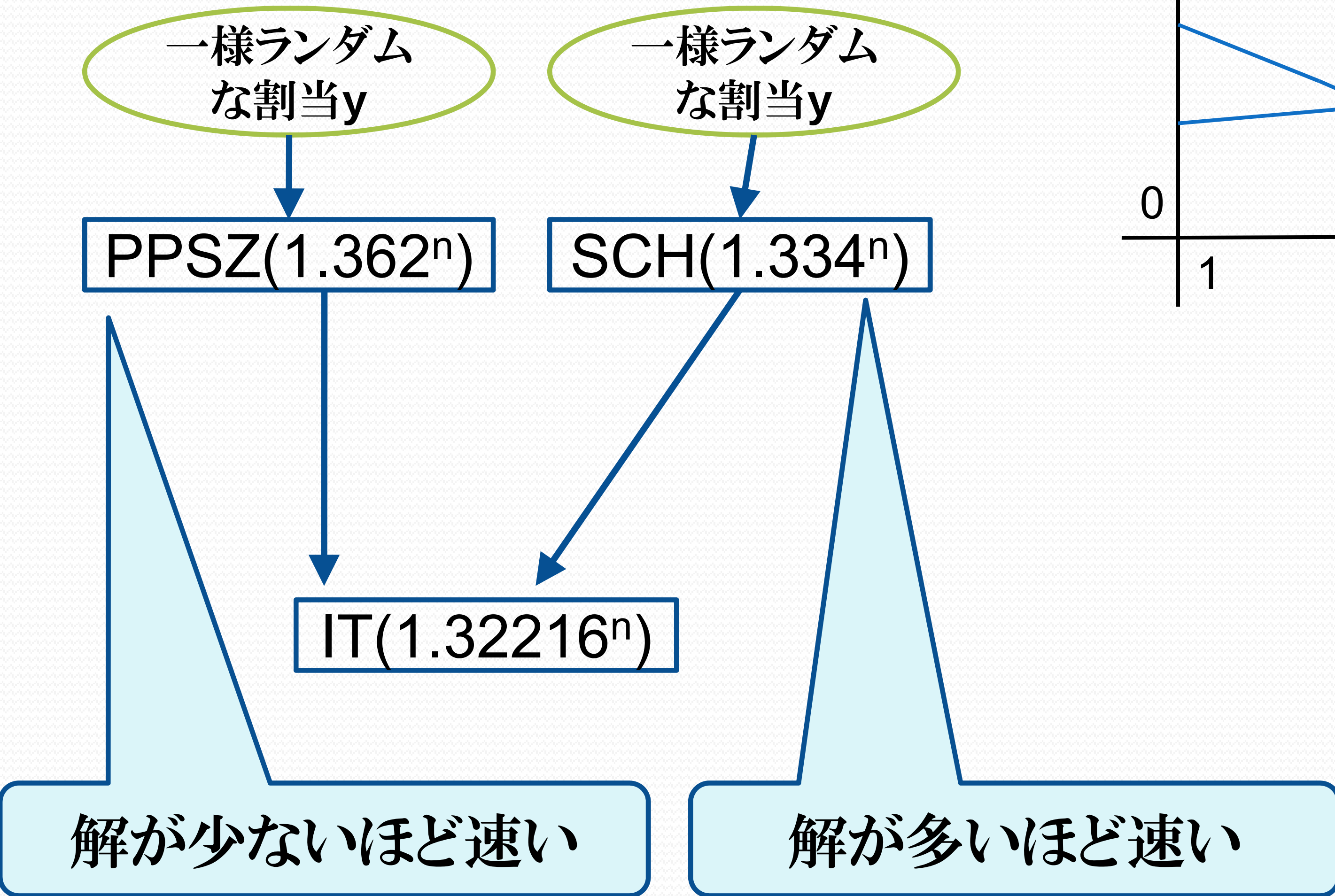
↓

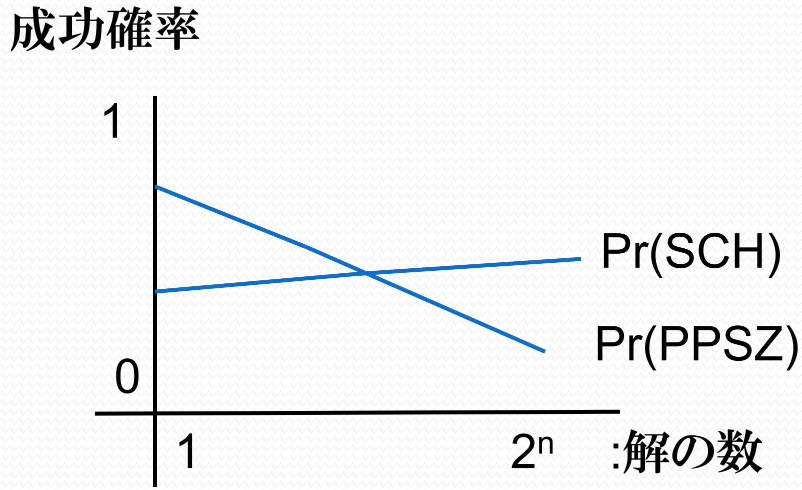
SCH(1.334^n)

既存アルゴリズム


一様ランダムな割当y
一様ランダムな割当y
PPSZ(1.362^n)
SCH(1.334^n)
IT(1.32216^n)
解が少ないほど速い
解が多いほど速い

既存アルゴリズム


一様ランダムな割当y
一様ランダムな割当y
PPSZ(1.362ⁿ)
SCH(1.334ⁿ)
IT(1.32216ⁿ)
解が少ないほど速い
解が多いほど速い


成功確率
1
0
Pr(SCH)
Pr(PPSZ)
1
2ⁿ
:解の数

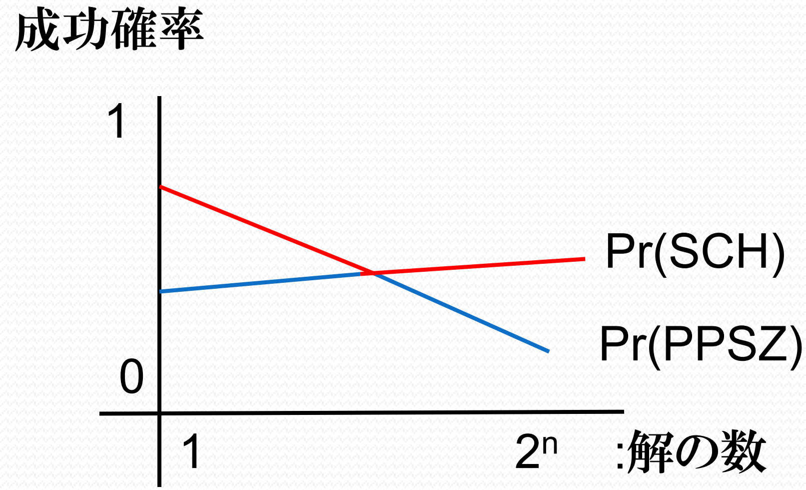
既存アルゴリズム

一様ランダムな割当y → PPSZ(1.362^n)

一様ランダムな割当y → SCH(1.334^n)

PPSZ(1.362^n) → IT(1.32216^n)

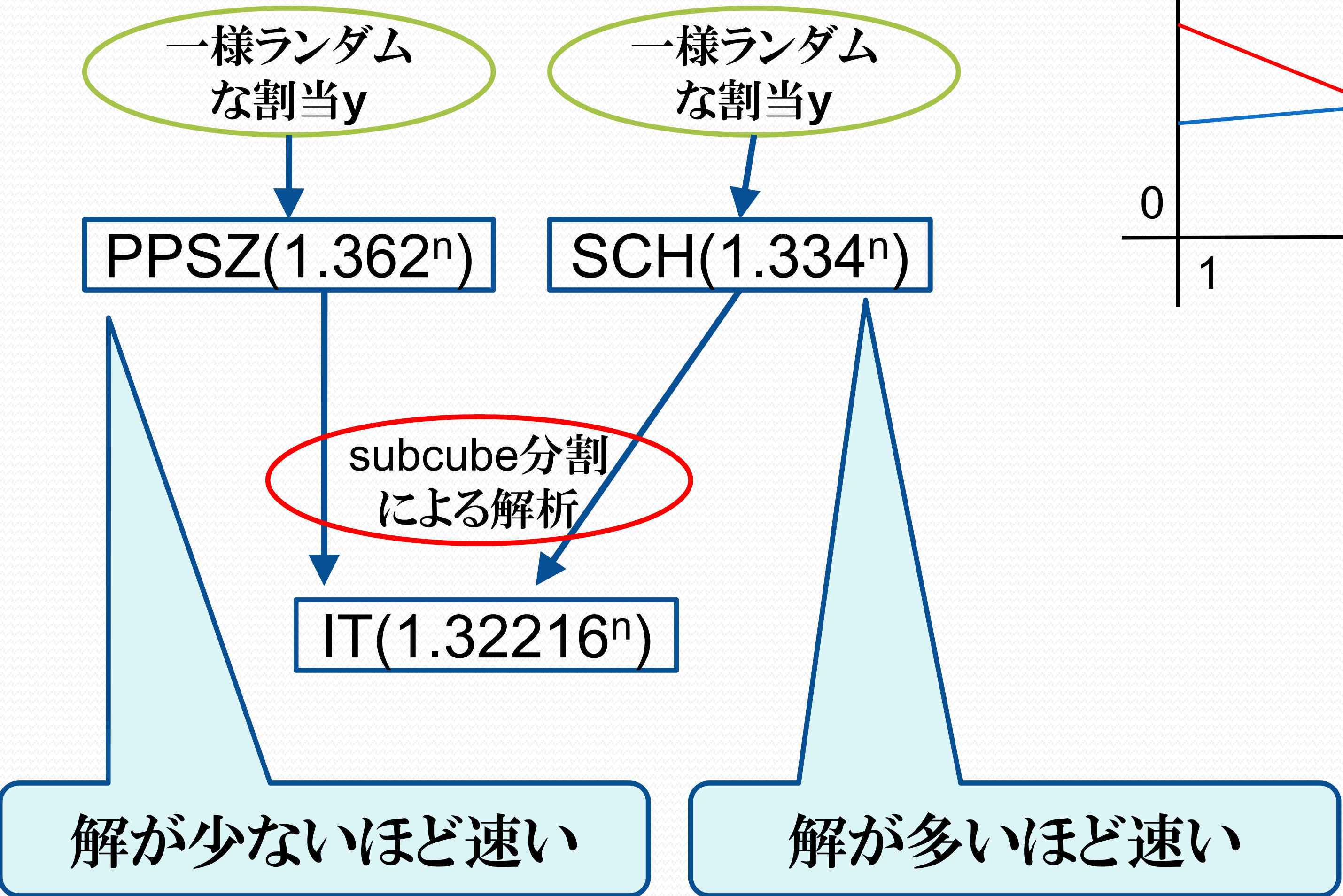
SCH(1.334^n) → IT(1.32216^n)



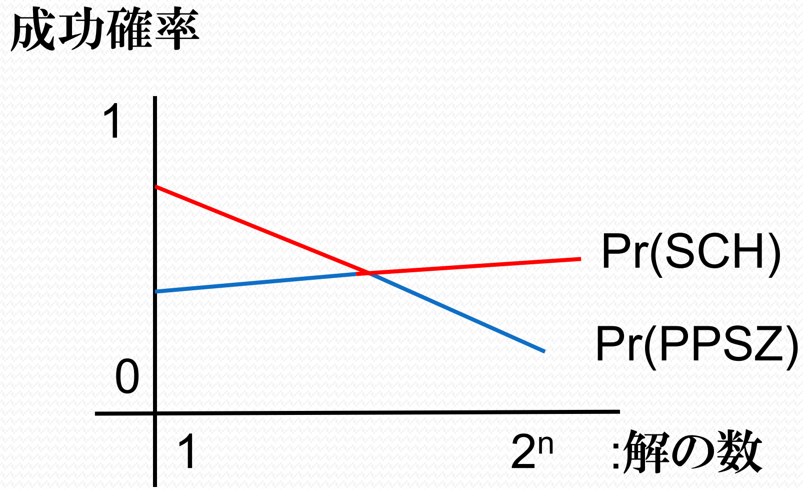
解が少ないほど速い

解が多いほど速い

既存アルゴリズム



一様ランダムな割当y
一様ランダムな割当y
PPSZ(1.362^n)
SCH(1.334^n)
subcube分割による解析
IT(1.32216^n)
解が少ないほど速い
解が多いほど速い



成功確率
1
0
Pr(SCH)
Pr(PPSZ)
1
2^n
:解の数

既存アルゴリズム

一様ランダムな割当y

PPSZ(1.362^n)

一様ランダムな割当y

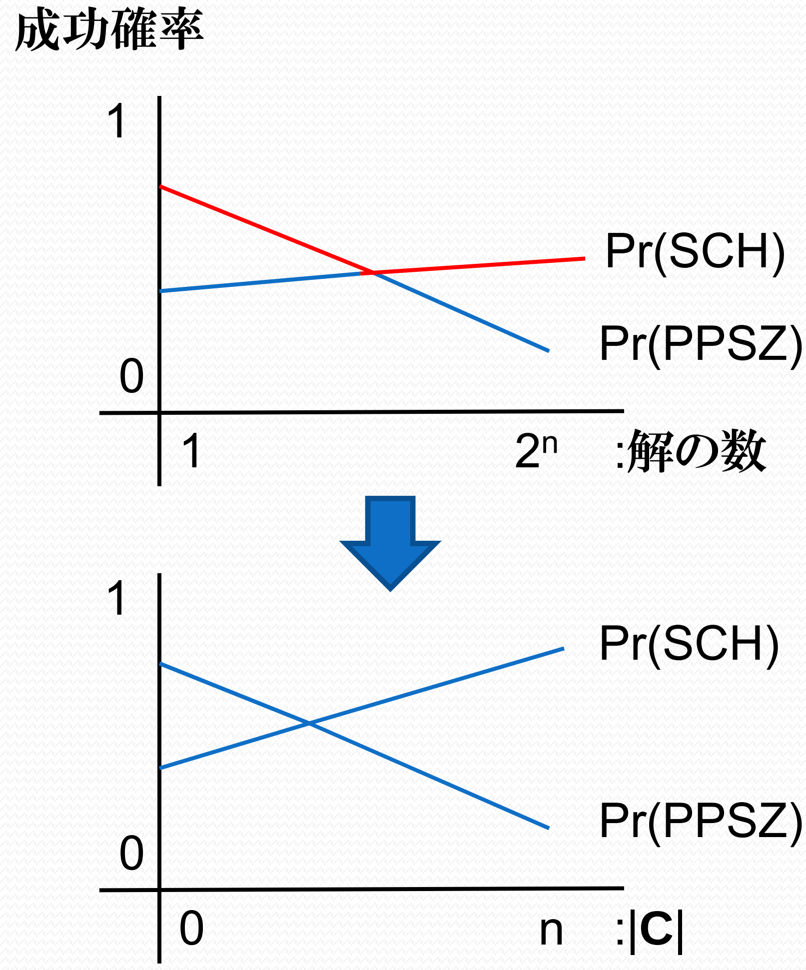
SCH(1.334^n)

subcube分割による解析

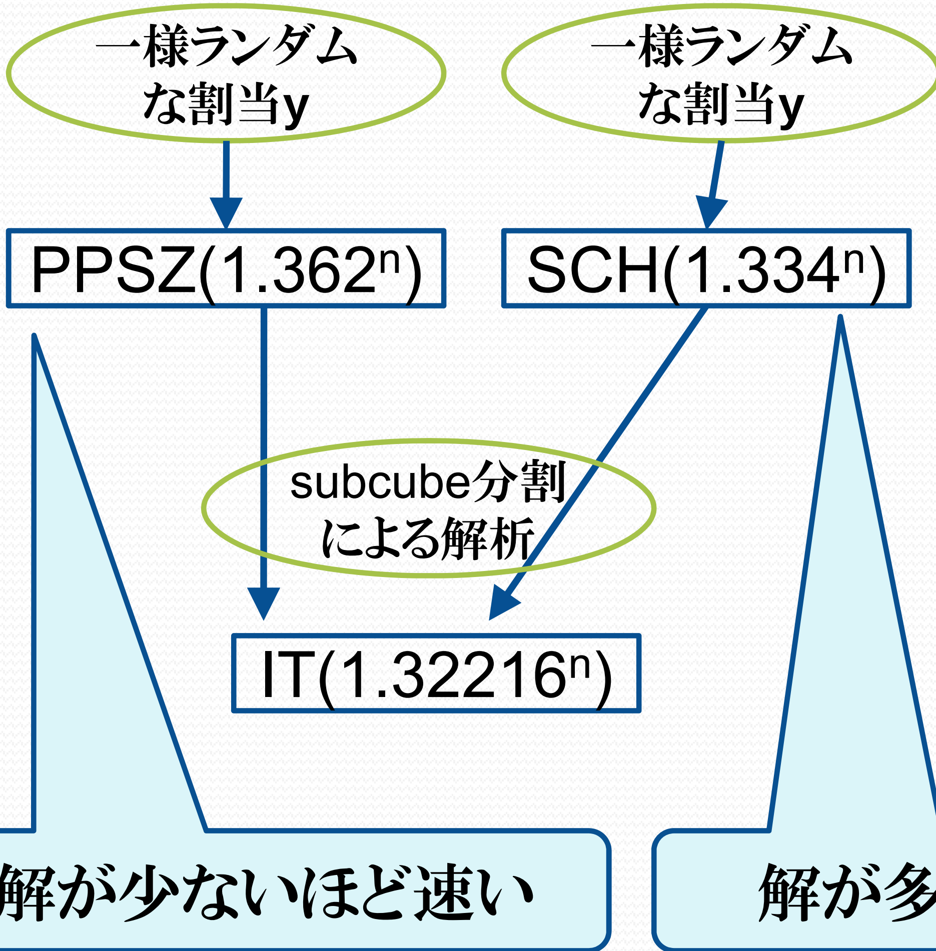
IT(1.32216^n)

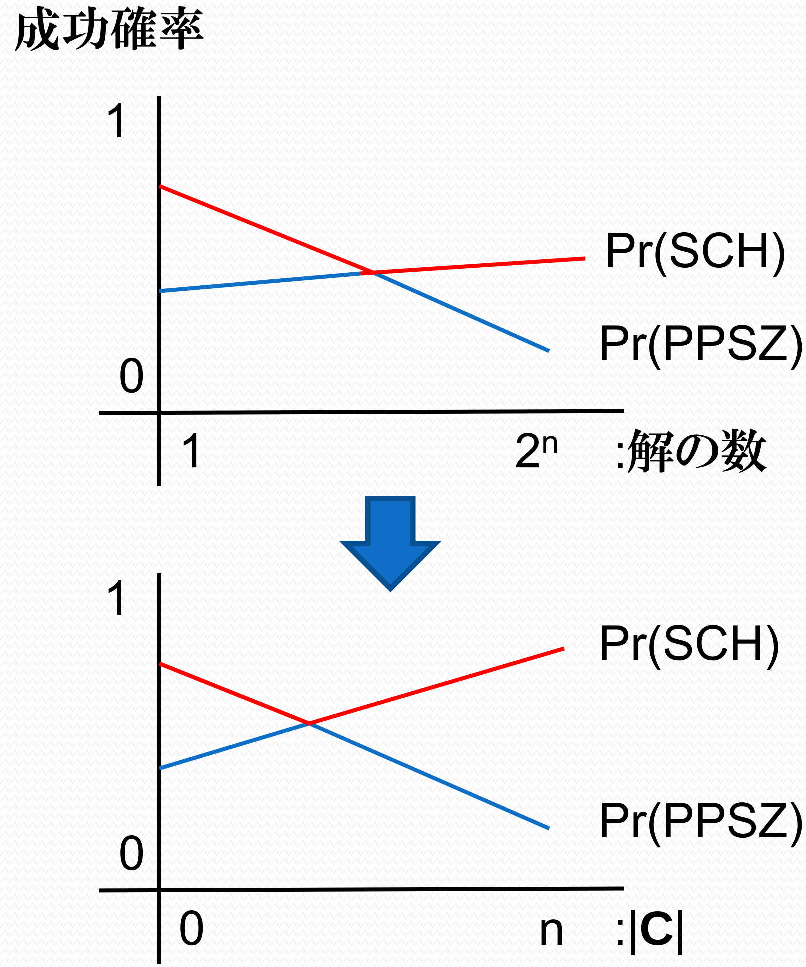
解が少ないほど速い

解が多いほど速い



既存アルゴリズム


一様ランダムな割当y
一様ランダムな割当y
PPSZ(1.362ⁿ)
SCH(1.334ⁿ)
subcube分割による解析
IT(1.32216ⁿ)
解が少ないほど速い
解が多いほど速い


成功確率
1
0
1
2ⁿ
:解の数
Pr(SCH)
Pr(PPSZ)
1
0
0
n
:|C|
Pr(SCH)
Pr(PPSZ)

解集合に基づくsubcube分割

- $\{0,1\}^n$をsubcubeに分割：各subcubeが解を一個含む

$\{0,1\}^4$

z_1=(0001)

z_3=(1100)

z_2=(0101)

z_4=(1110)

$\{0,1\}^4$

解集合に基づくsubcube分割

- $\{0,1\}^n$をsubcubeに分割：各subcubeが解を一個含む

$x_1=0$

$\mathbf{z_1}=(0001)$

$\mathbf{z_2}=(0101)$

$x_1=1$

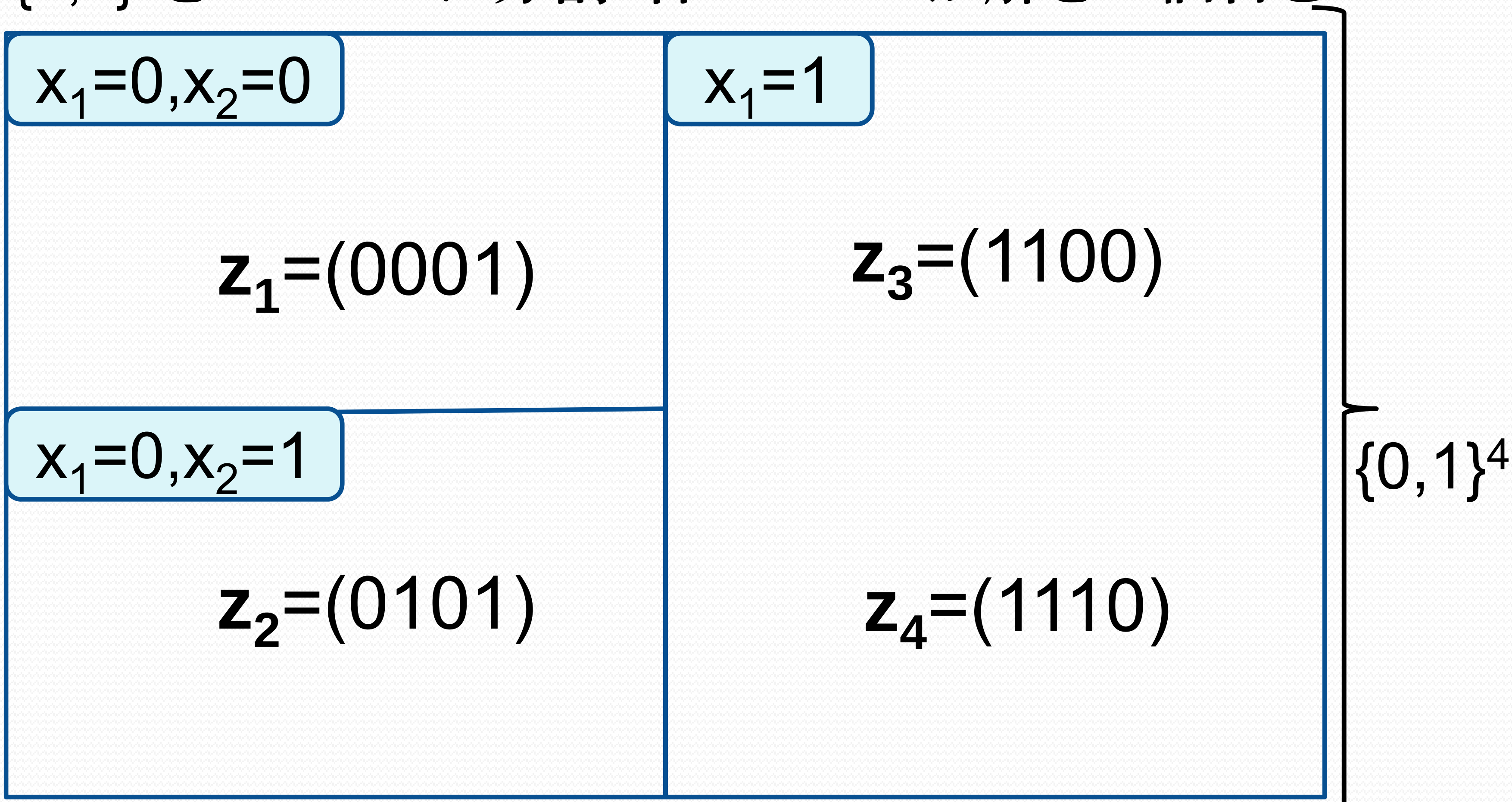
$\mathbf{z_3}=(1100)$

$\mathbf{z_4}=(1110)$

$\{0,1\}^4$

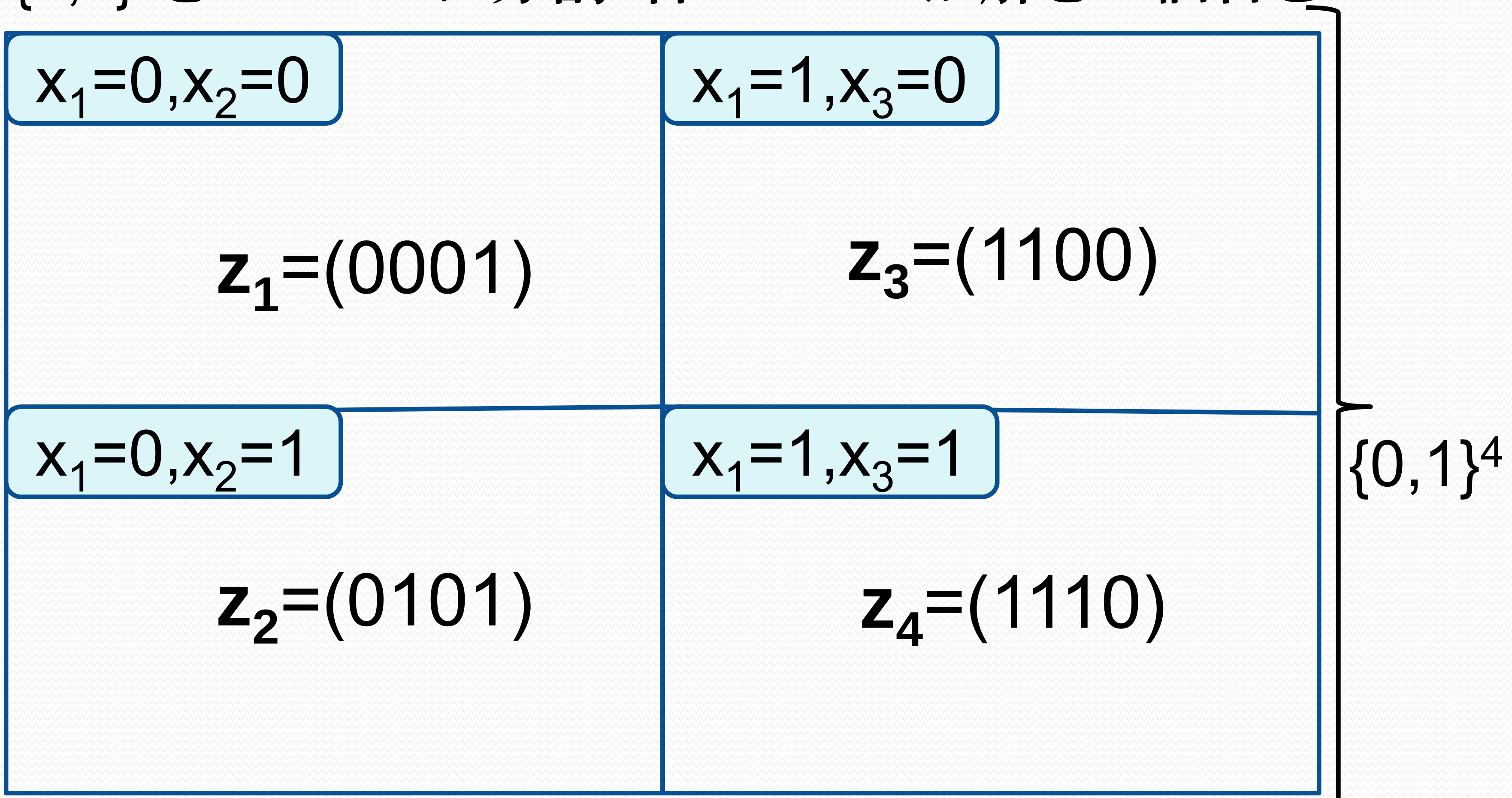
解集合に基づくsubcube分割

- $\{0,1\}^n$をsubcubeに分割：各subcubeが解を一個含む



解集合に基づくsubcube分割

- $\{0,1\}^n$をsubcubeに分割：各subcubeが解を一個含む



解集合に基づくsubcube分割の性質

Lemma 1
任意の充足可能なFにおいて、
解集合に基づくsubcubeの分割が存在する

※一意とは限らない
※実際に作る必要はない

Lemma 2
割当yがsubcube Cに属する
⇒解と同じ割当の変数が|C|個以上

|C|:subcubeを
構成する変数の数

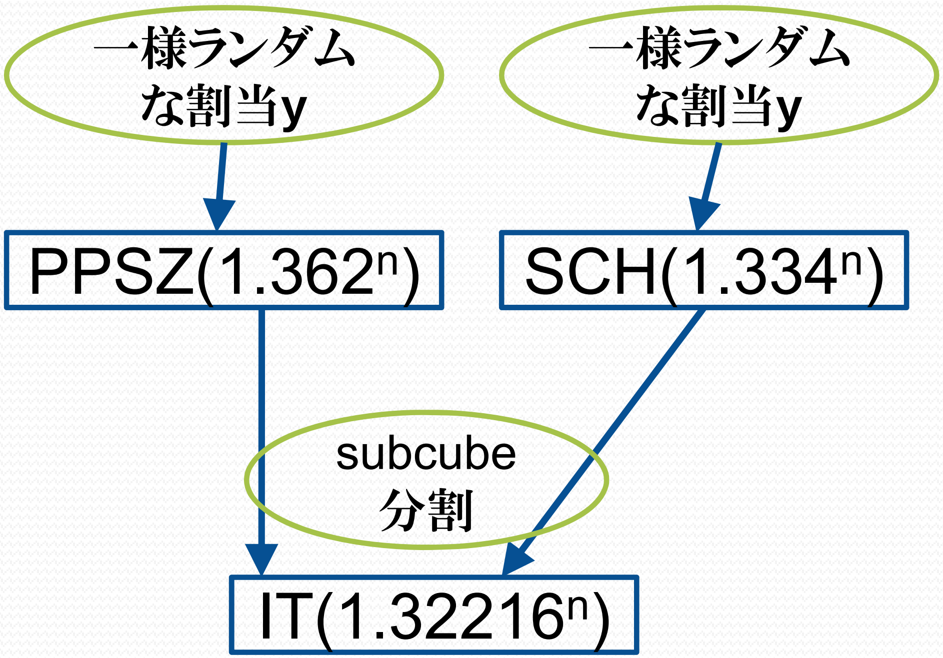
$x_2=0$

$|C|=1$

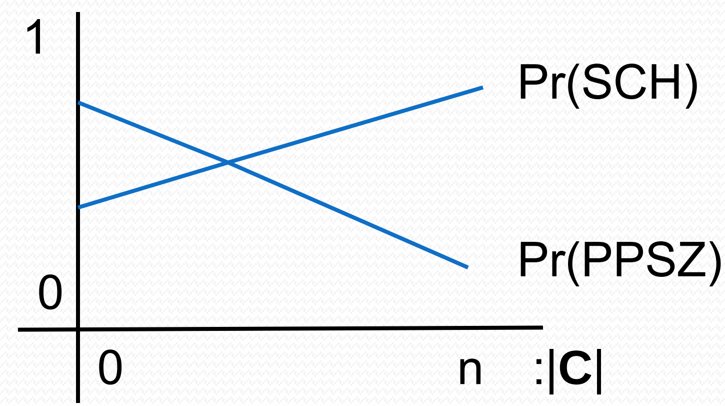
$z_1=(0001)$

$y=(1011)$
$y=(0010)$

ITアルゴリズムの解析



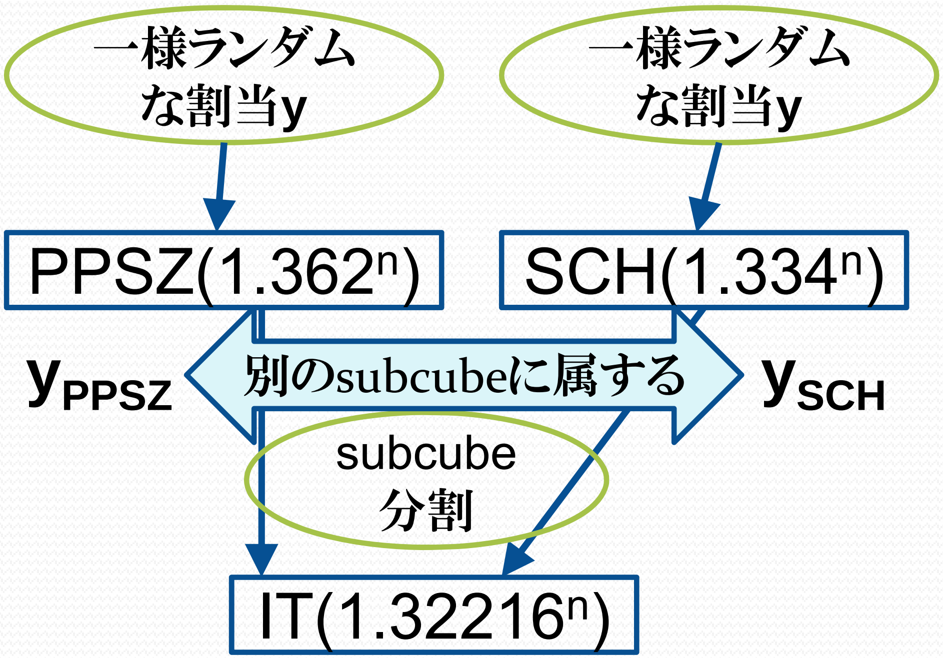
成功確率 y∈subcube Cの条件下



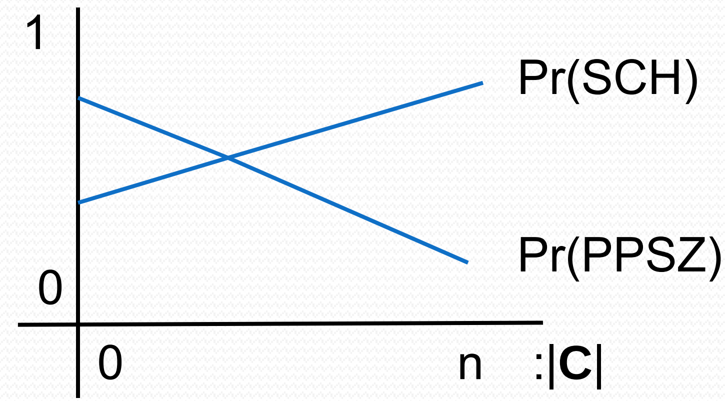
$$\Pr(PPSZ \mid C) \geq 2^{-0.38770n+0.51854|C|}$$

$$\Pr(SCH \mid C) \geq \left(\frac{3}{4}\right)^{n-|C|}$$

ITアルゴリズムの解析



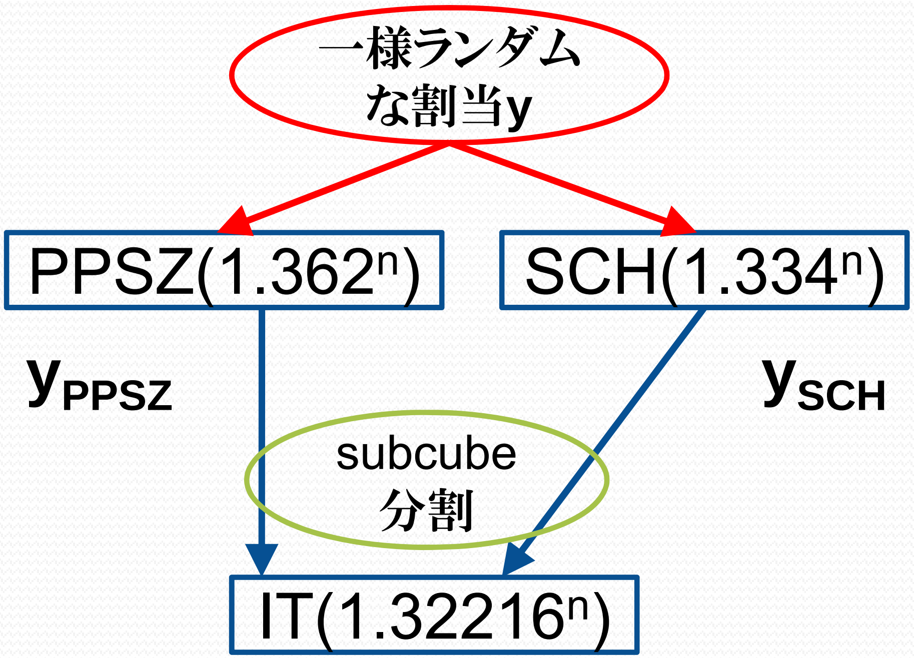
成功確率　y∈subcube Cの条件下



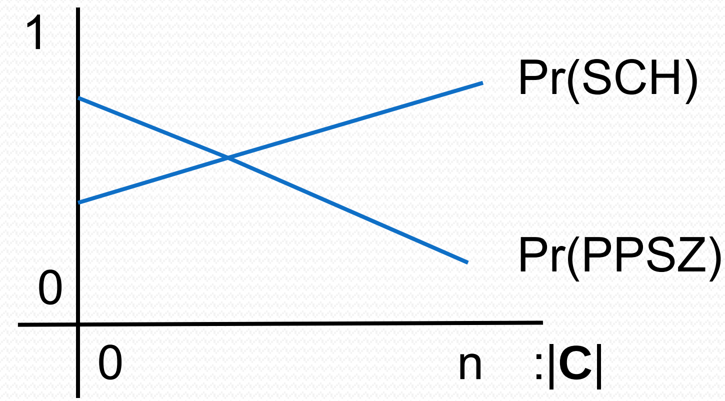
$$\Pr(PPSZ \mid C) \geq 2^{-0.38770n+0.51854|C|}$$

$$\Pr(SCH \mid C) \geq \left(\frac{3}{4}\right)^{n-|C|}$$

ITアルゴリズムの解析



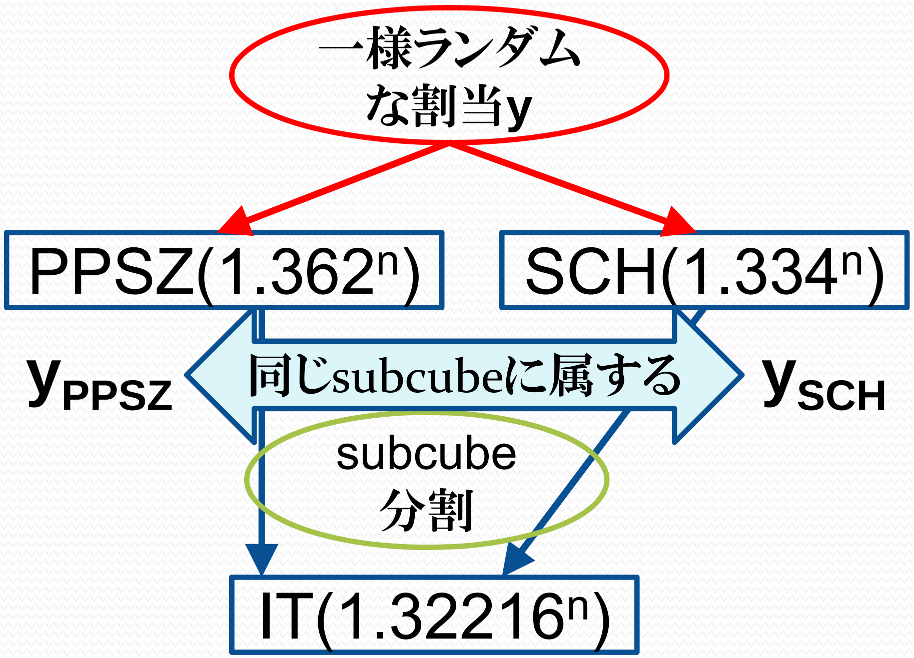
成功確率 y∈subcube Cの条件下



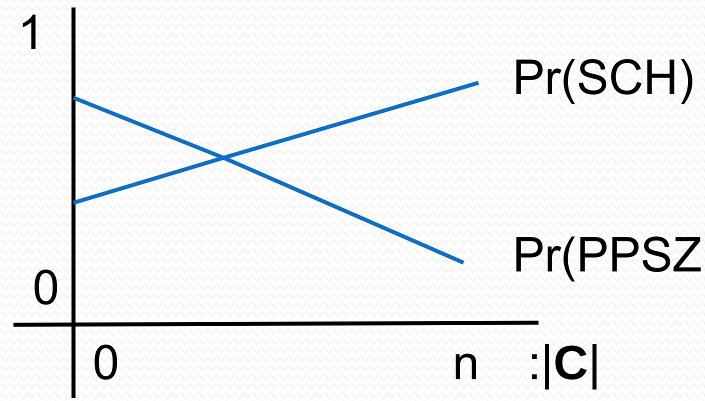
$$\Pr(PPSZ \mid C) \geq 2^{-0.38770n+0.51854|C|}$$

$$\Pr(SCH \mid C) \geq \left(\frac{3}{4}\right)^{n-|C|}$$

ITアルゴリズムの解析



成功確率　y∈subcube Cの条件下

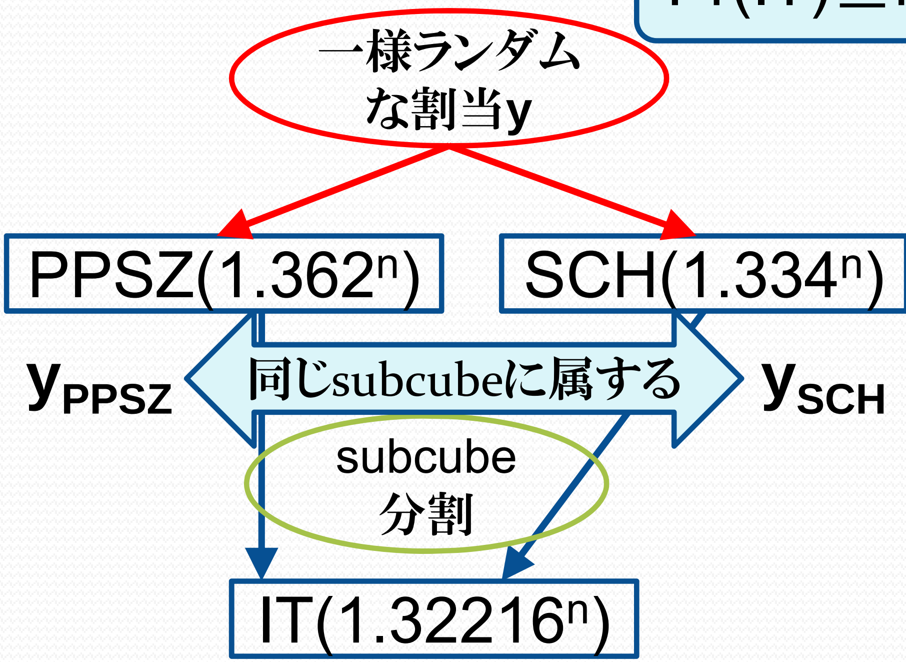


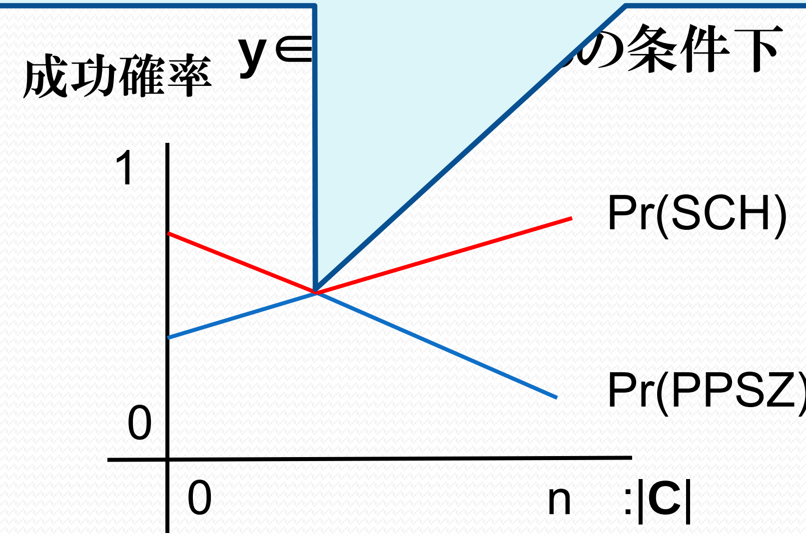
$$\Pr(PPSZ \mid C) \geq 2^{-0.38770n+0.51854|C|}$$

$$\Pr(SCH \mid C) \geq \left(\frac{3}{4}\right)^{n-|C|}$$

ITアルゴリズムの解析

$\Pr(IT) \geqq \min_{C} \max\{\Pr(SCH|\mathbf{C}), \Pr(PPSZ|\mathbf{C})\}$





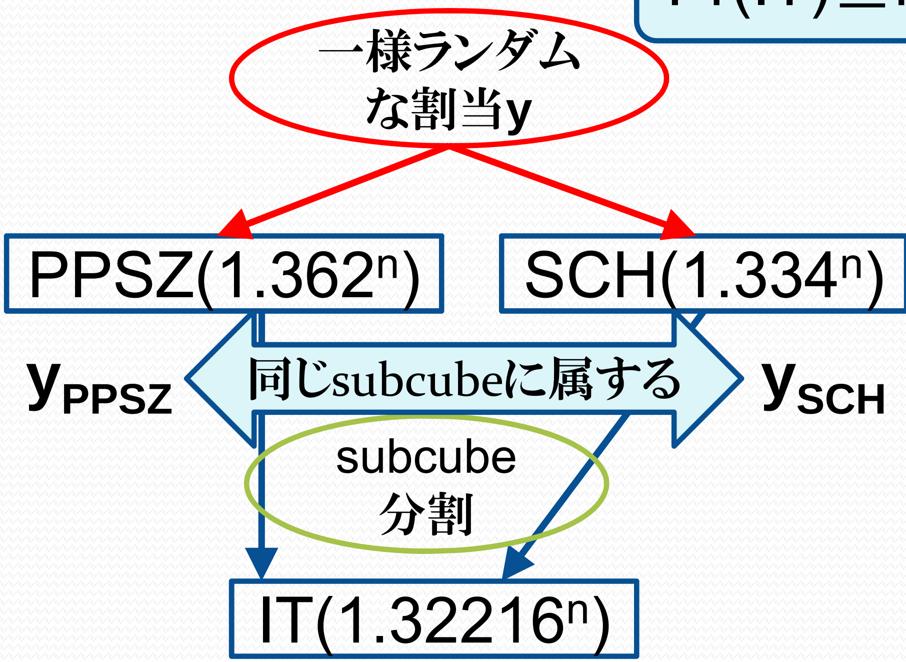
$$\Pr(PPSZ \mid C) \geq 2^{-0.38770n + 0.51854|C|}$$

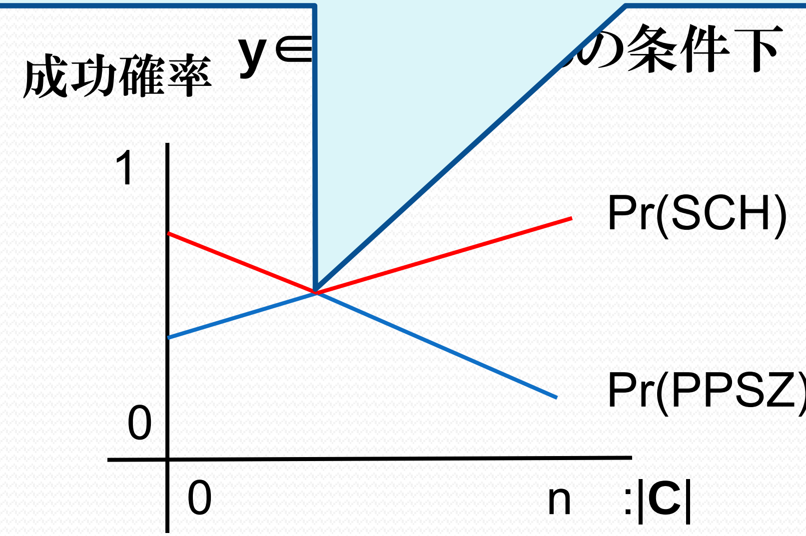
$$\Pr(SCH \mid C) \geq \left(\frac{3}{4}\right)^{n-|C|}$$

$$\Pr(IT) \geq \Omega(1.32216^{-n})$$

ITアルゴリズムの解析

$\Pr(IT) \geqq \min_C \max\{\Pr(SCH|\mathbf{C}), \Pr(PPSZ|\mathbf{C})\}$





$$\Pr(PPSZ \mid C) \geq 2^{-0.38770n + 0.51854|C|}$$

$$\Pr(SCH \mid C) \geq \left(\frac{3}{4}\right)^{n-|C|}$$

$$\Pr(IT) \geq \Omega(1.32216^{-n})$$

SCHアルゴリズム

初期割当yから局所探索を行う

Lemma 3 [Sch99]
初期割当yからスタートした時に解zを見つける確率は

$$\left(\frac{1}{2}\right)^{d(y,z)}$$

(d(y,z)は**yとzのハミング距離**)

ある分布に従うランダムな初期割当yからスタートした時の成功確率は

$$E_y\left[\left(\frac{1}{2}\right)^{d(y,z)}\right]$$

SCHアルゴリズム

y={y1,y2,y3,…,yn}
z={z1,z2,z3,…,zn}

一様ランダムな初期割当yの場合、yのそれぞれの変数は独立

$$E_y\left[\left(\frac{1}{2}\right)^{d(y,z)}\right]=E_{y1}\left[\left(\frac{1}{2}\right)^{d(y1,z1)}\right]\cdot E_{y2}\left[\left(\frac{1}{2}\right)^{d(y2,z2)}\right]\cdots E_{yn}\left[\left(\frac{1}{2}\right)^{d(yn,zn)}\right]$$

$$=\left(\frac{1}{2}\cdot\left(\frac{1}{2}\right)^1+\frac{1}{2}\cdot\left(\frac{1}{2}\right)^0\right)\cdots\left(\frac{1}{2}\cdot\left(\frac{1}{2}\right)^1+\frac{1}{2}\cdot\left(\frac{1}{2}\right)^0\right)=\left(\frac{3}{4}\right)^n$$

Theorem 4 [Sch99]

一様ランダムな初期割当yの場合の成功確率

$$\Pr(SCH)\geq\left(\frac{3}{4}\right)^n$$

SCHアルゴリズム

y={y1,y2,y3,…,yn}
z={z1,z2,z3,…,zn}

一様 … は独立

Lemma 5 [IT04]

一様ランダムな初期割当(y∈C)の場合の条件付き成功確率

$$\Pr(SCH \mid C) \geq \left(\frac{3}{4}\right)^{n-|C|}$$

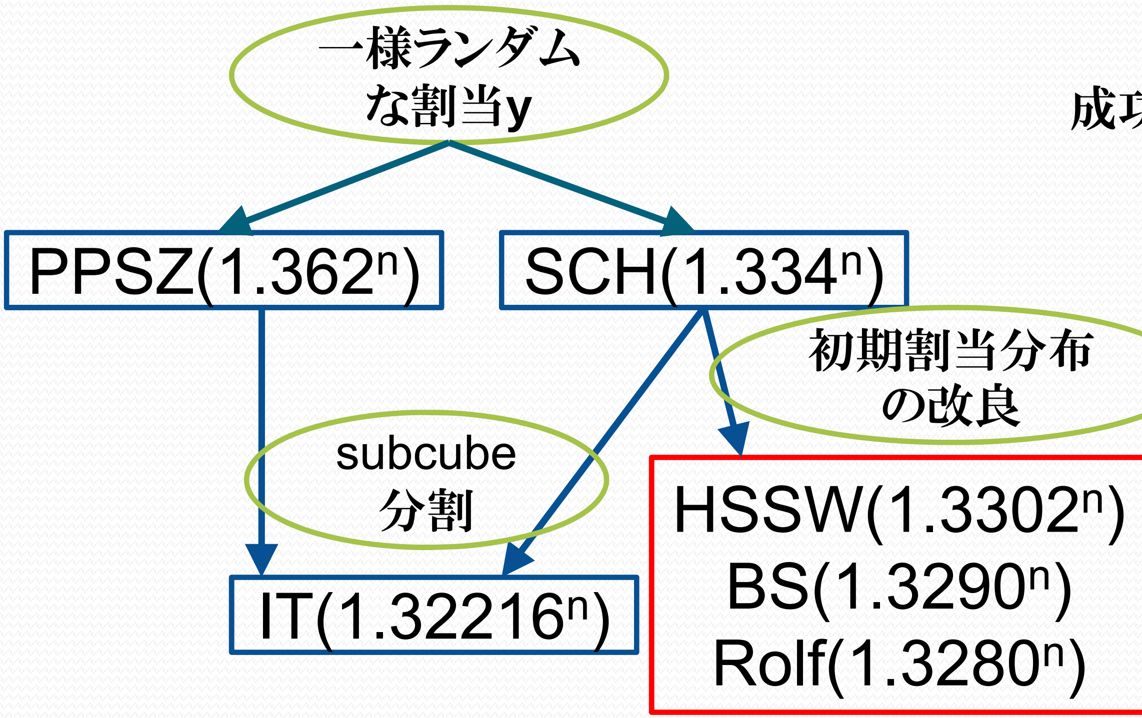
$$= \cdots = \left(\frac{3}{4}\right)^{n}$$

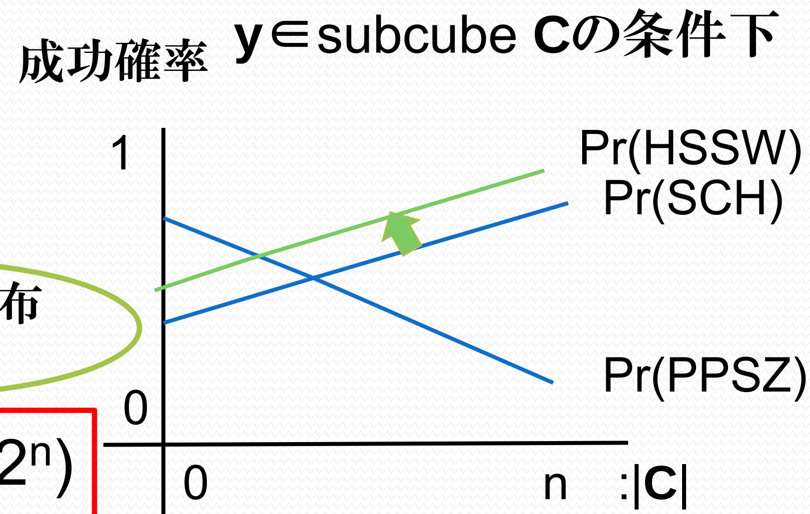
Theorem 4 [Sch99]

一様ランダムな初期割当yの場合の成功確率

$$\Pr(SCH) \geq \left(\frac{3}{4}\right)^{n}$$

SCH改良アルゴリズム

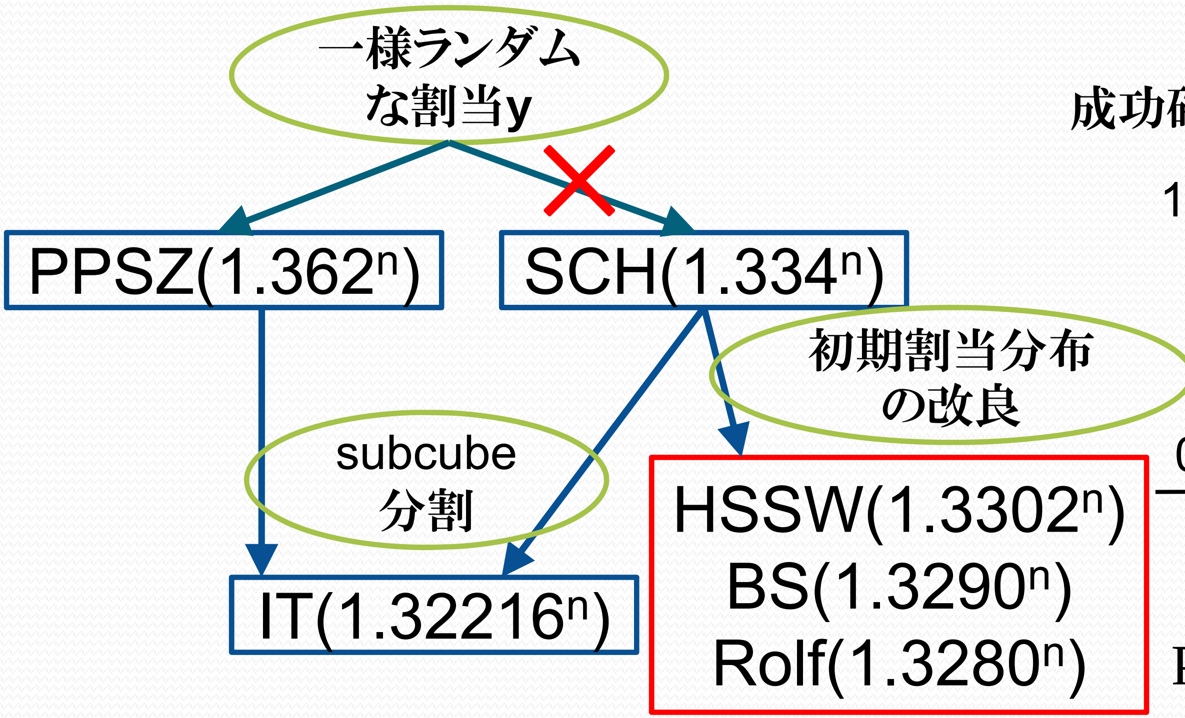


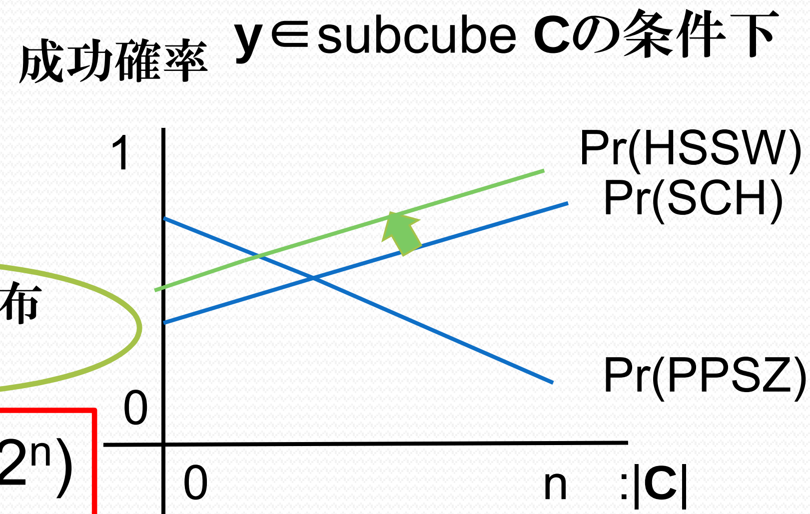


$$\Pr(PPSZ \mid C) \geq 2^{-0.38770n+0.51854|C|}$$

$$\Pr(SCH \mid C) \geq \left(\frac{3}{4}\right)^{n-|C|}$$

SCH改良アルゴリズム



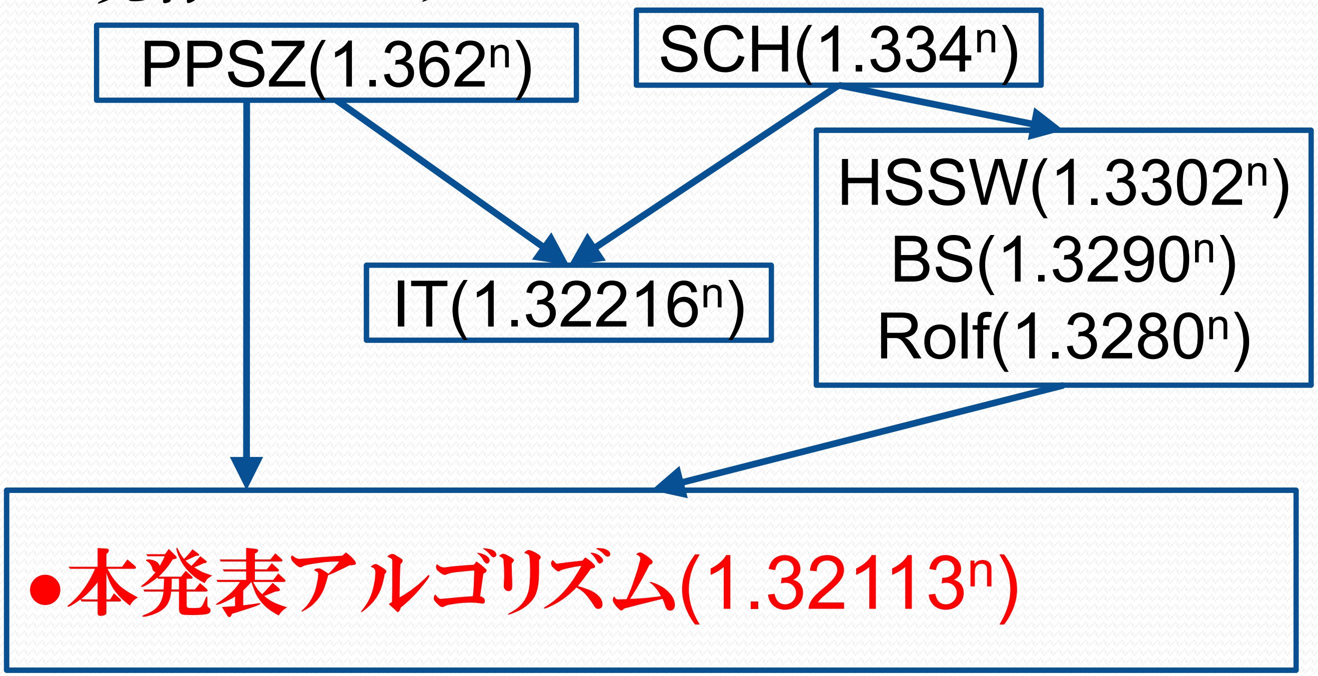


$$\Pr(PPSZ \mid C) \geq 2^{-0.38770n+0.51854|C|}$$

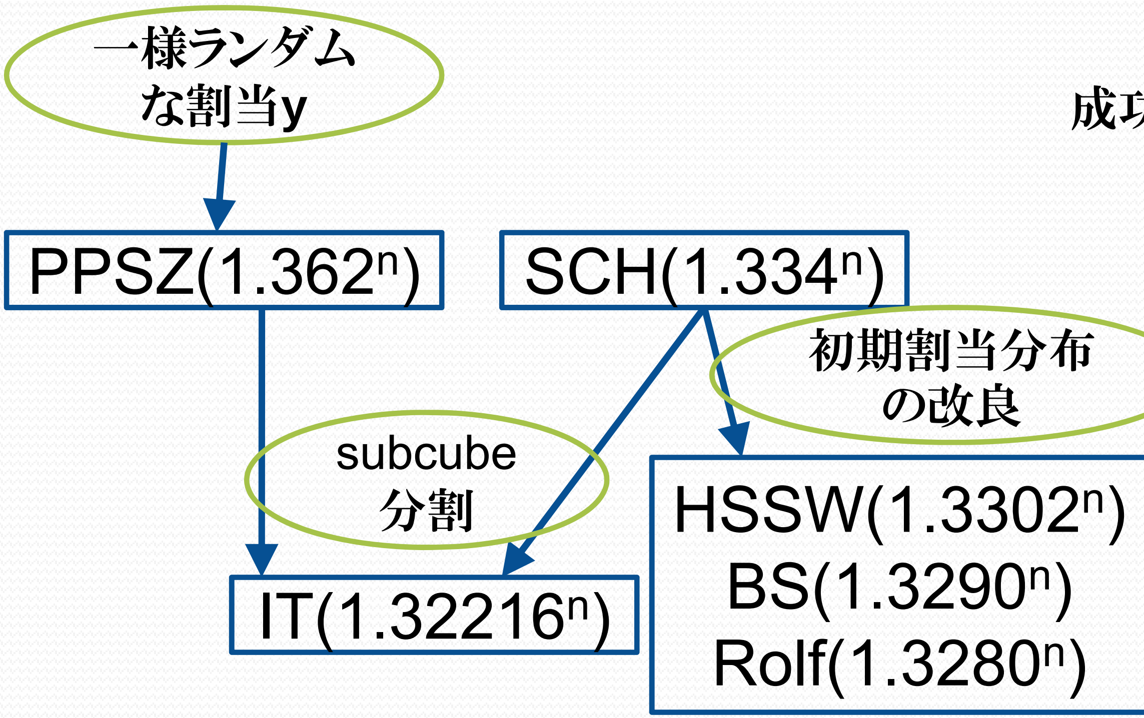
$$\Pr(SCH \mid C) \geq \left(\frac{3}{4}\right)^{n-|C|}$$

発表の流れ

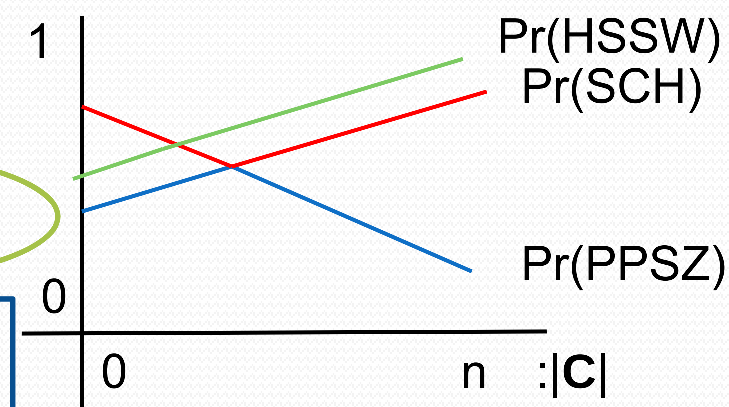
- ✓背景と結果
  - 3SATの定義、既存結果と改良結果
- 既存アルゴリズム



改良アイデア



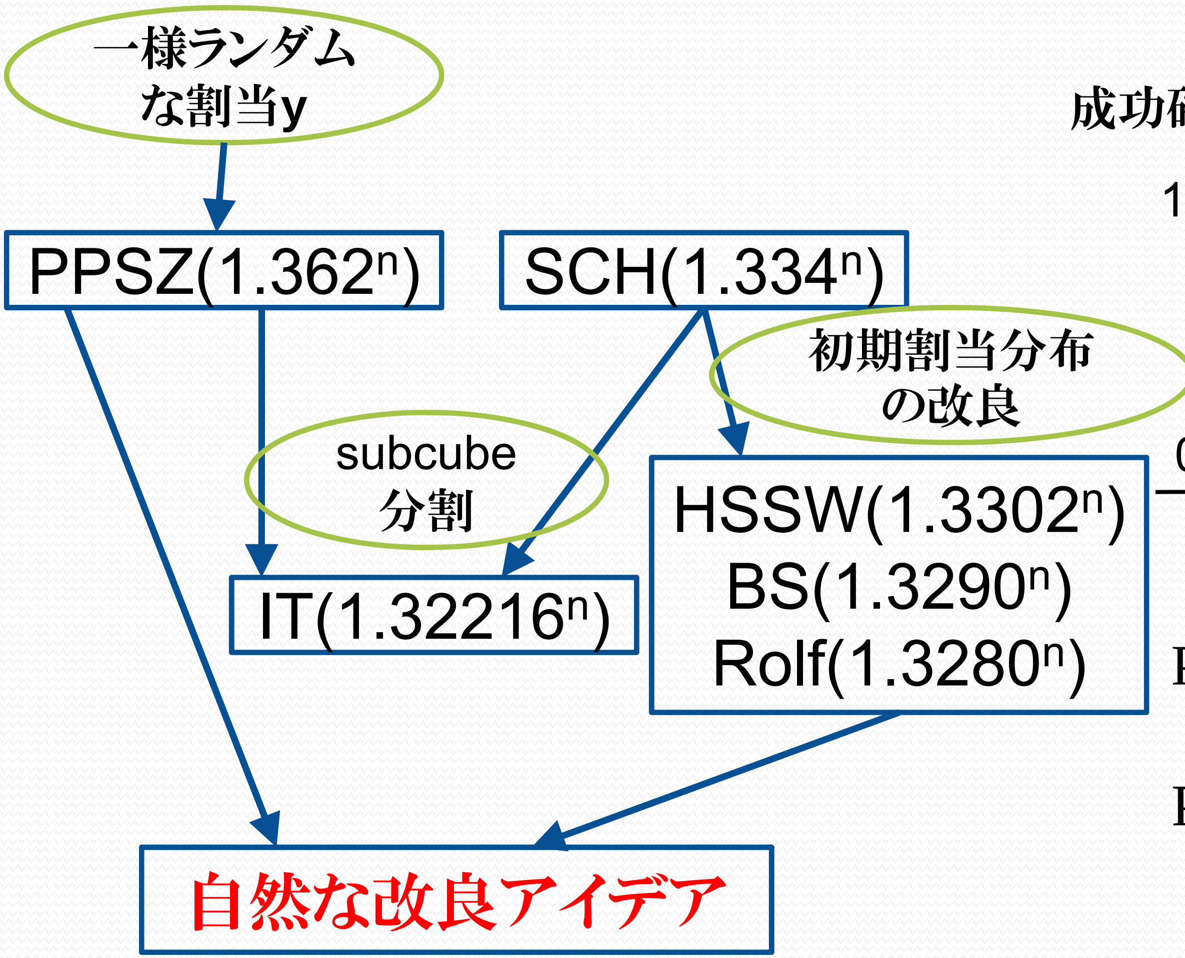
成功確率　**y∈subcube Cの条件下**

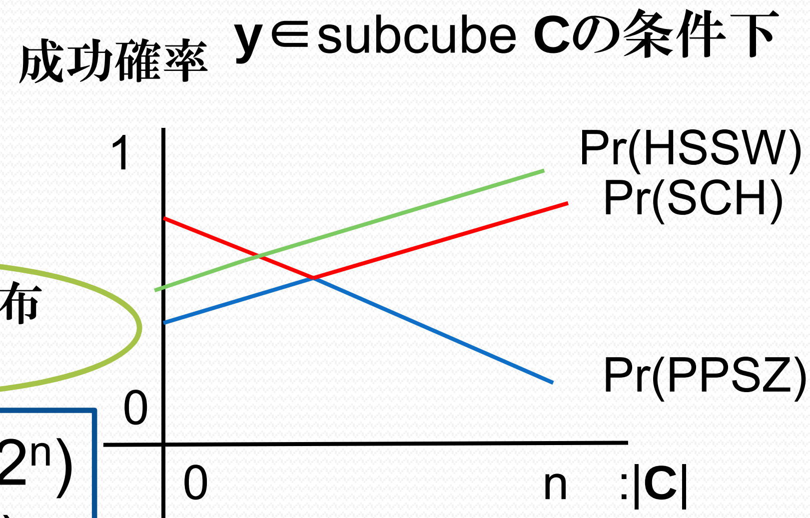


$$\Pr(PPSZ \mid C) \geq 2^{-0.38770n+0.51854|C|}$$

$$\Pr(SCH \mid C) \geq \left(\frac{3}{4}\right)^{n-|C|}$$

改良アイデア

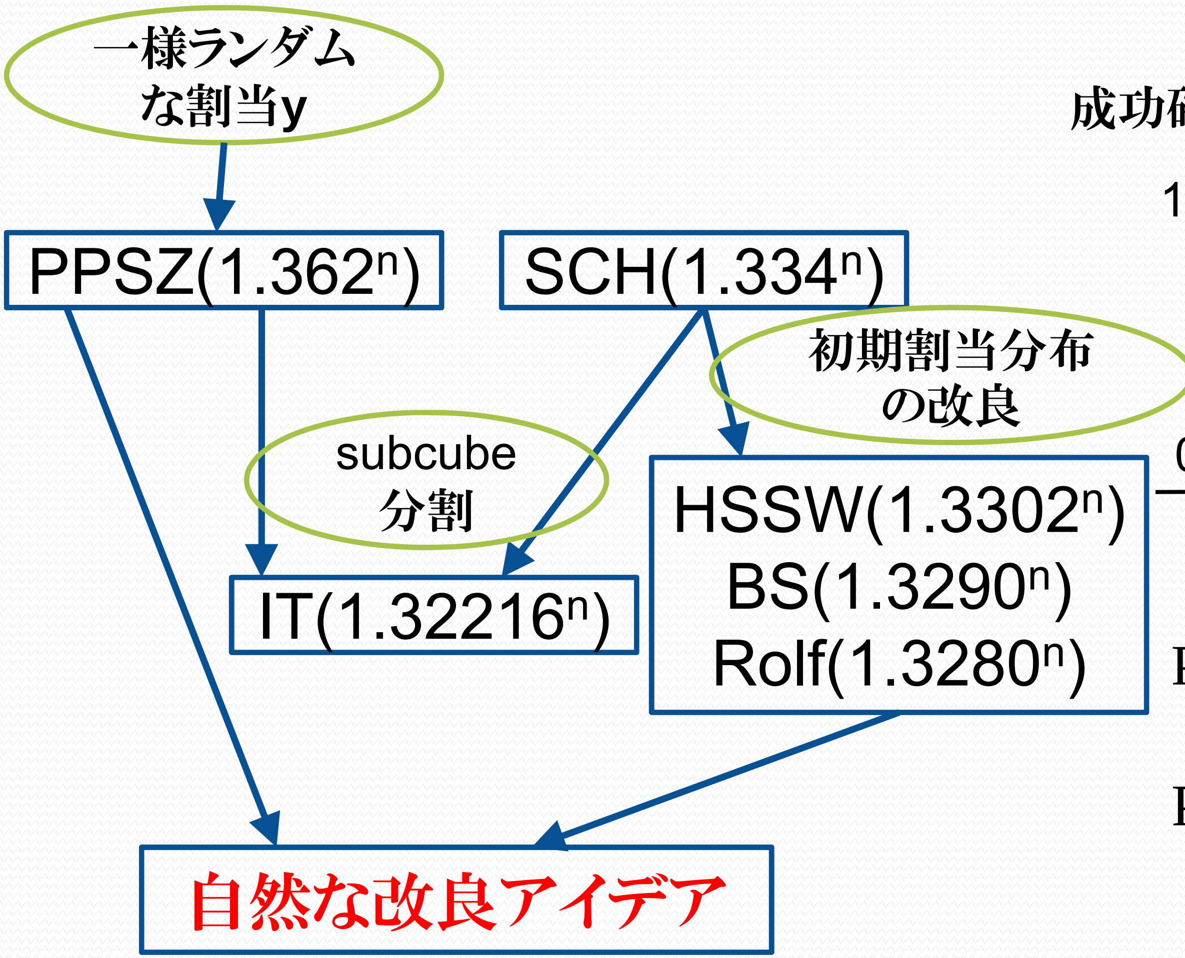


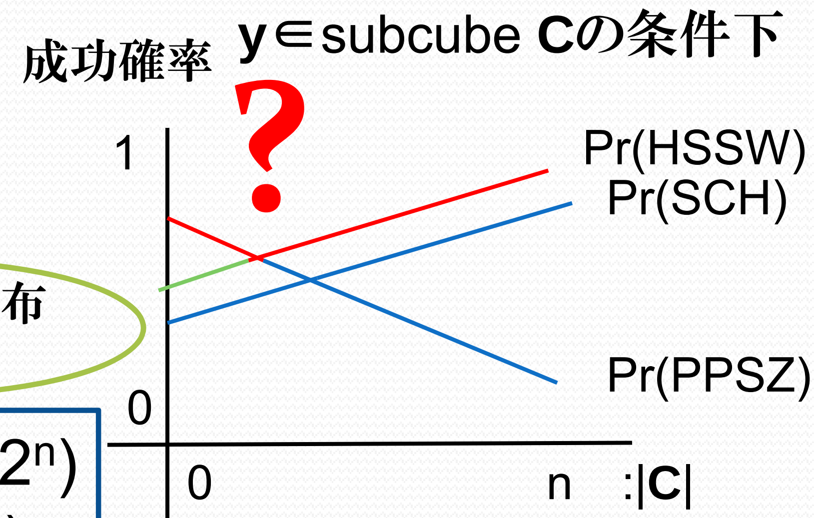


$$\Pr(PPSZ \mid C) \geq 2^{-0.38770n+0.51854|C|}$$

$$\Pr(SCH \mid C) \geq \left(\frac{3}{4}\right)^{n-|C|}$$

改良アイデア

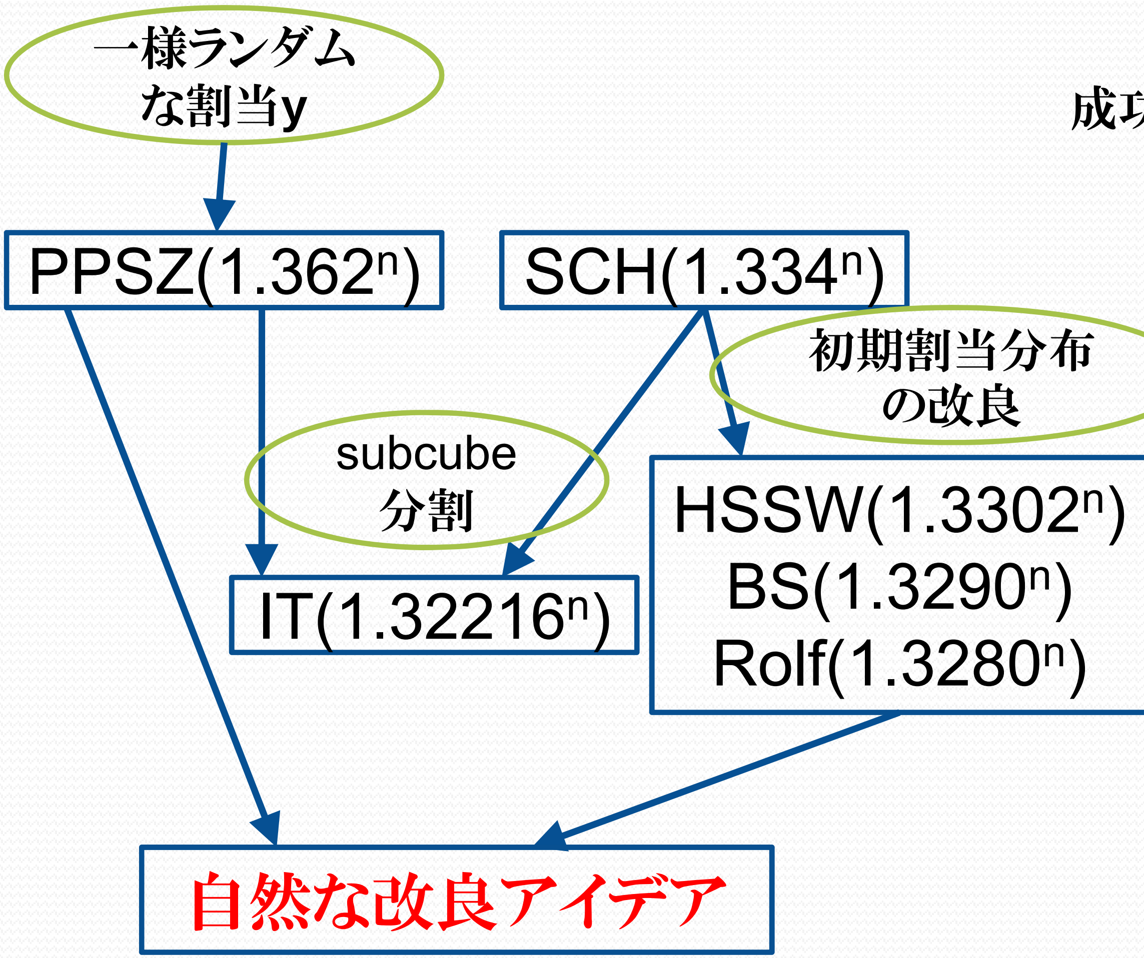


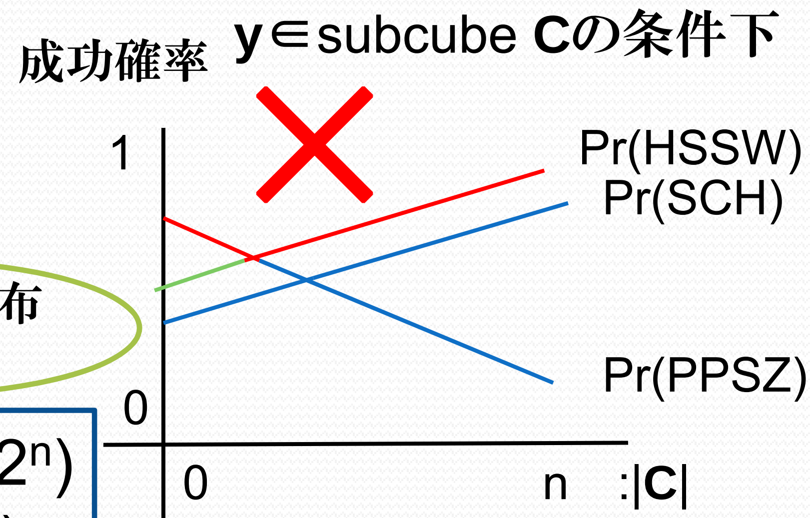


$$\Pr(PPSZ \mid C) \geq 2^{-0.38770n+0.51854|C|}$$

$$\Pr(SCH \mid C) \geq \left(\frac{3}{4}\right)^{n-|C|}$$

改良アイデア⇒失敗

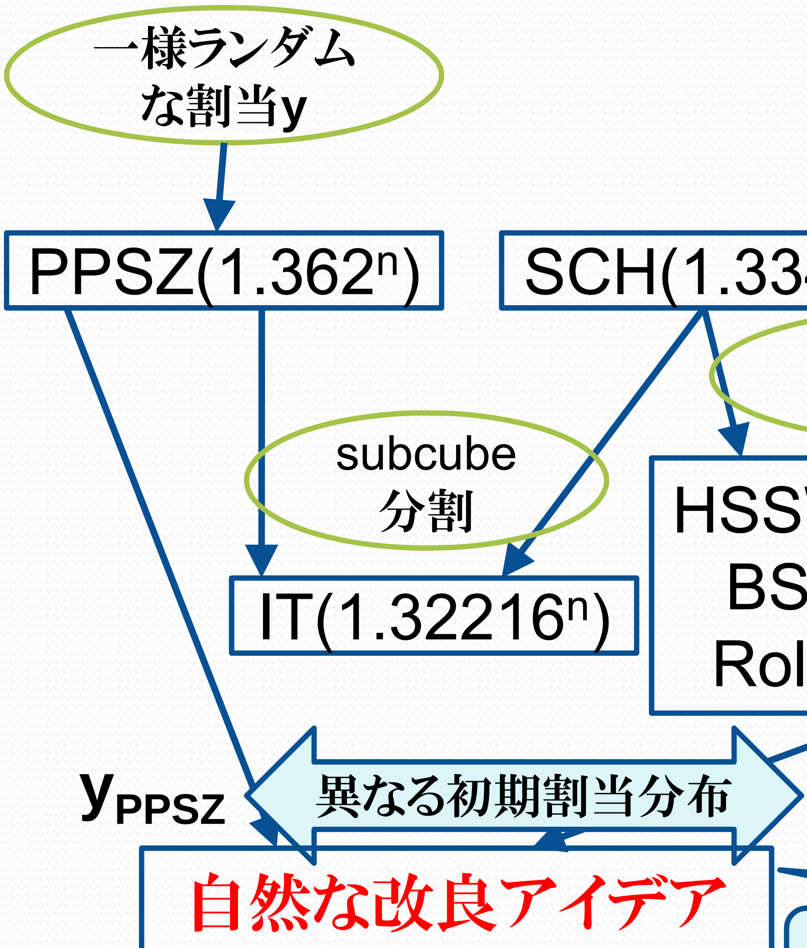


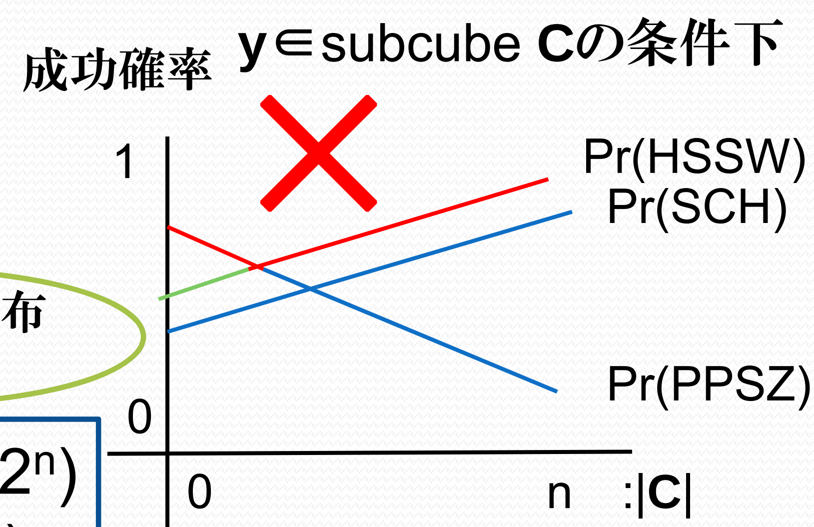


$$\Pr(PPSZ \mid C) \ge 2^{-0.38770n+0.51854|C|}$$

$$\Pr(SCH \mid C) \ge \left(\frac{3}{4}\right)^{n-|C|}$$

改良アイデア⇒失敗



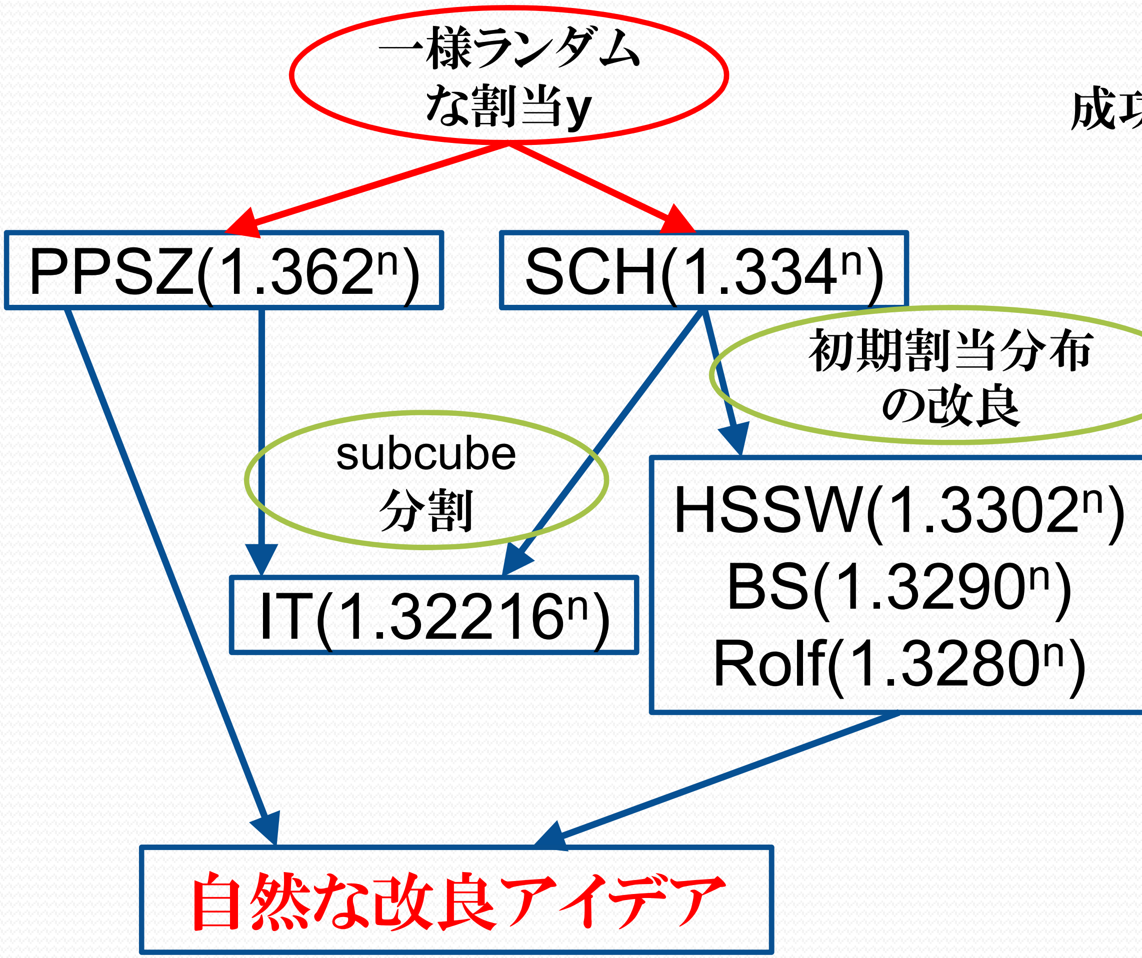


$$\Pr(PPSZ \mid C) \geq 2^{-0.38770n+0.51854|C|}$$

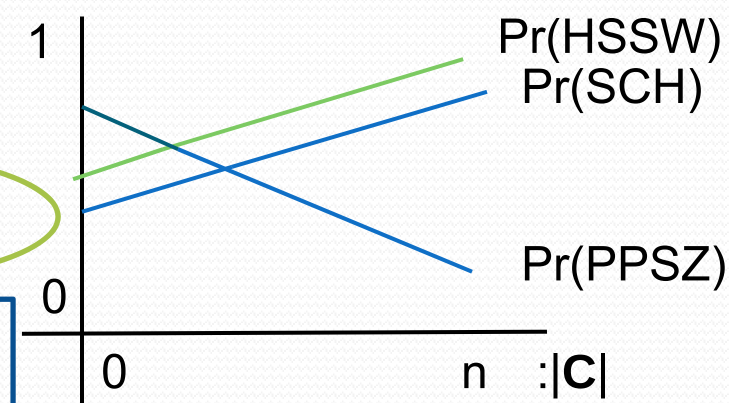
$$\Pr(SCH \mid C) \geq \left(\frac{3}{4}\right)^{n-|C|}$$

$y_{PPSZ}\in C$, $y_{HSSW}\in C'$となり、$|C|\neq|C'|$

改良アイデア修正版



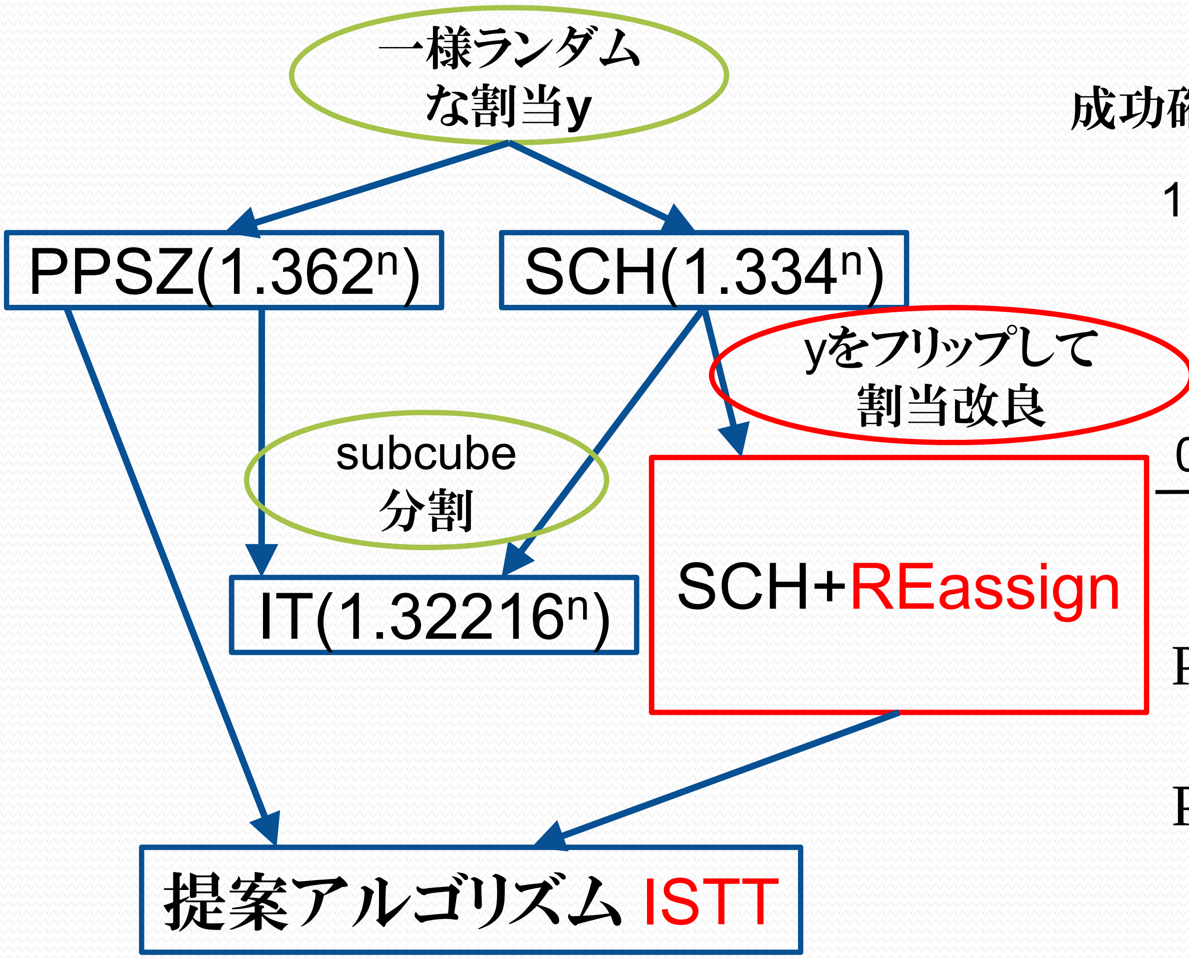
成功確率　y∈subcube Cの条件下

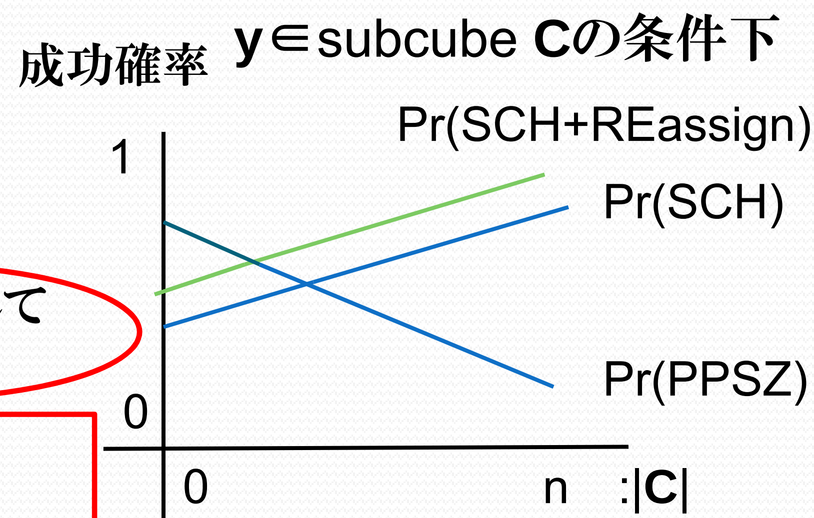


$$\Pr(PPSZ \mid C) \geq 2^{-0.38770n+0.51854|C|}$$

$$\Pr(SCH \mid C) \geq \left(\frac{3}{4}\right)^{n-|C|}$$

改良アイデア修正版

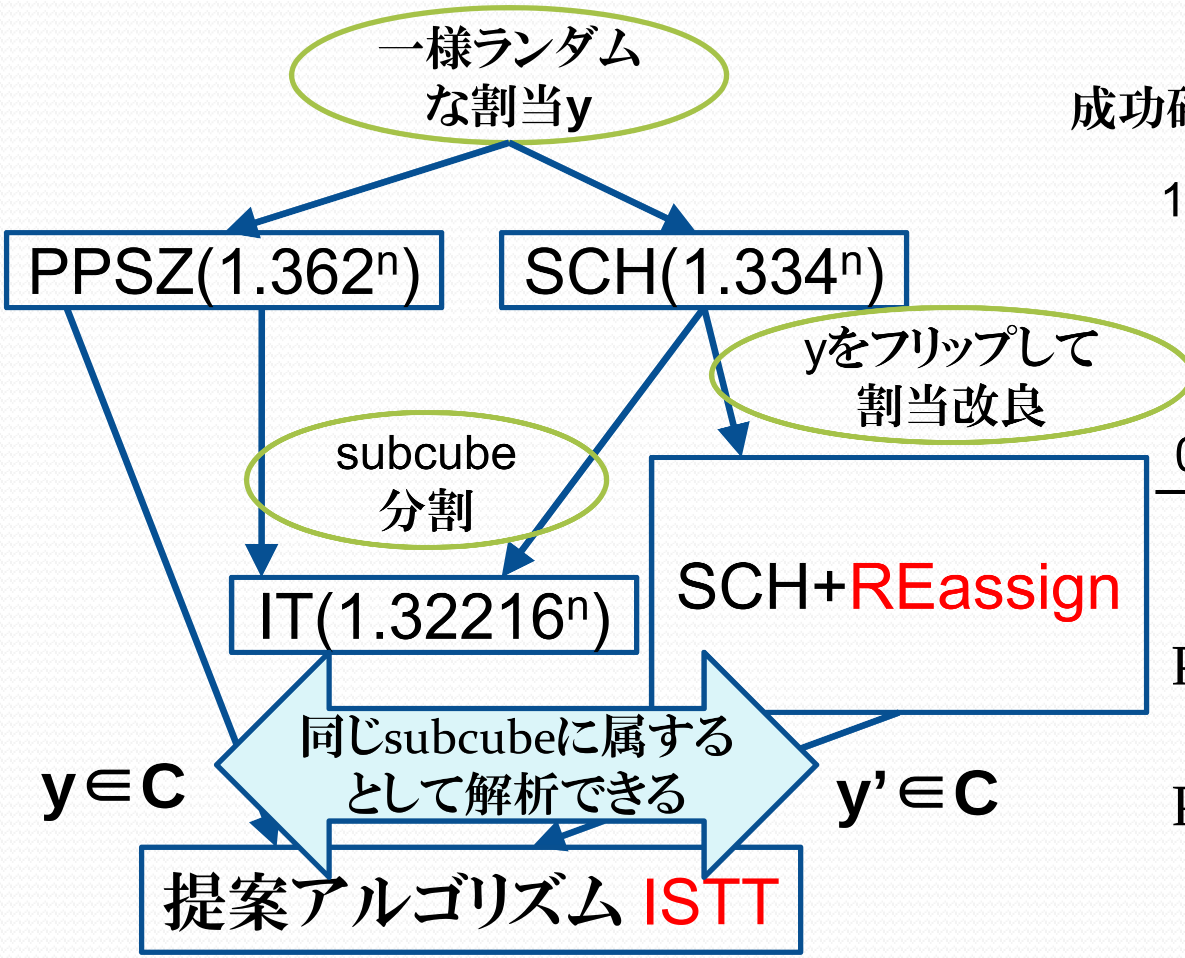


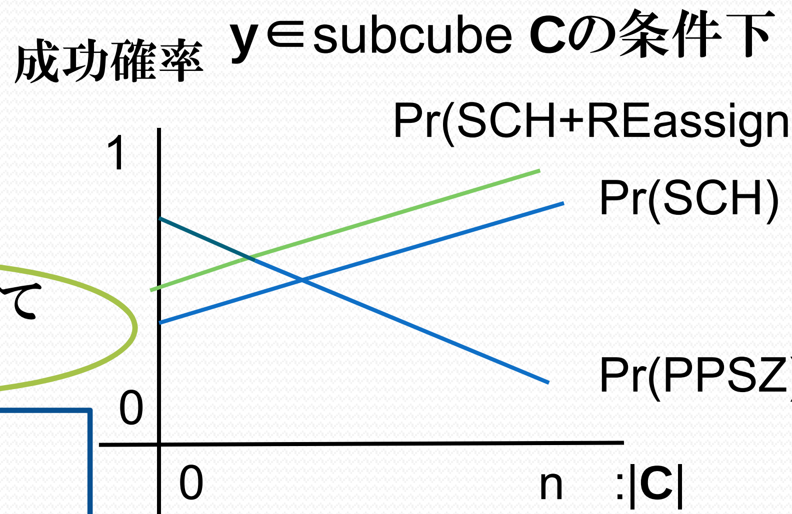


$$\Pr(PPSZ \mid C) \geq 2^{-0.38770n+0.51854|C|}$$

$$\Pr(SCH \mid C) \geq \left(\frac{3}{4}\right)^{n-|C|}$$

改良アイデア修正版



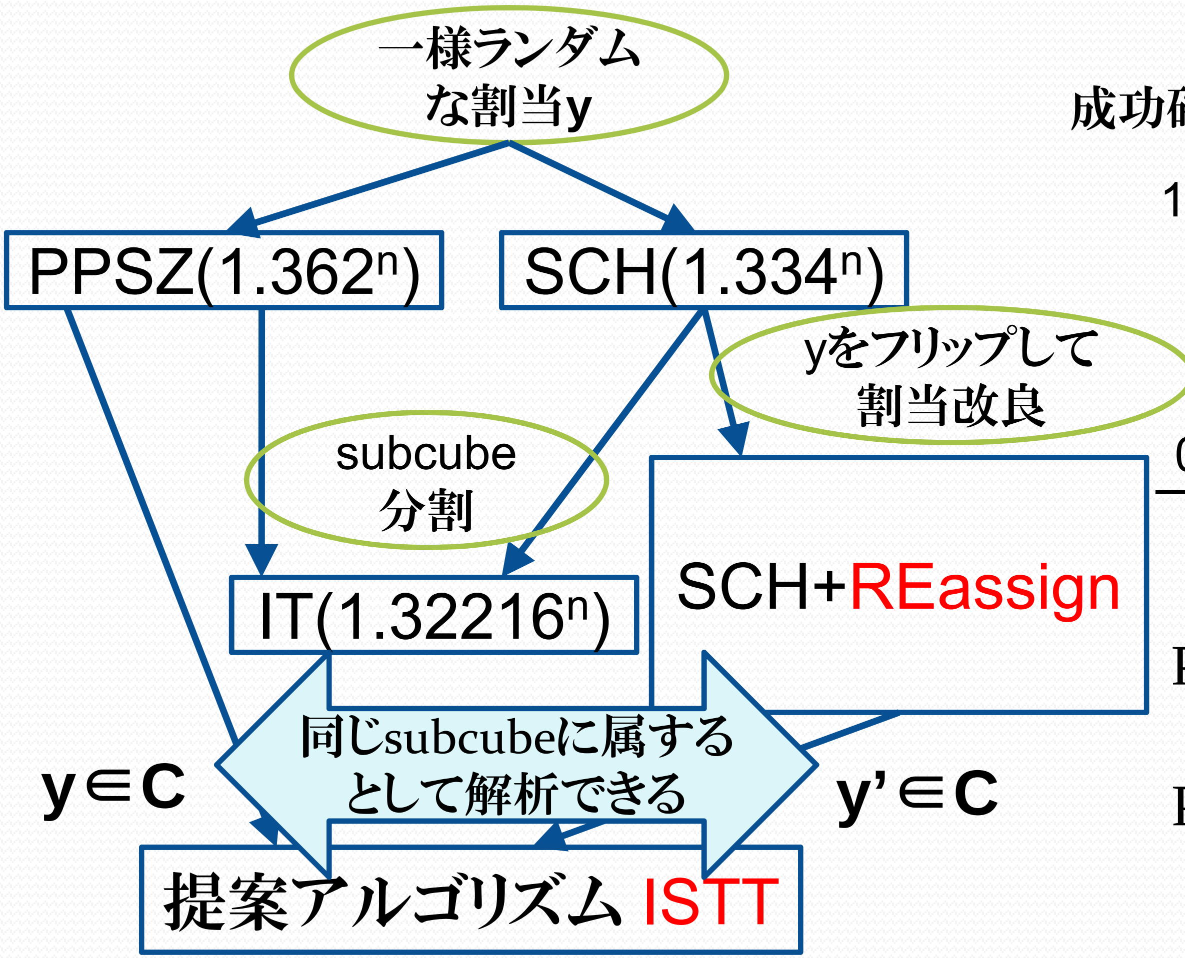


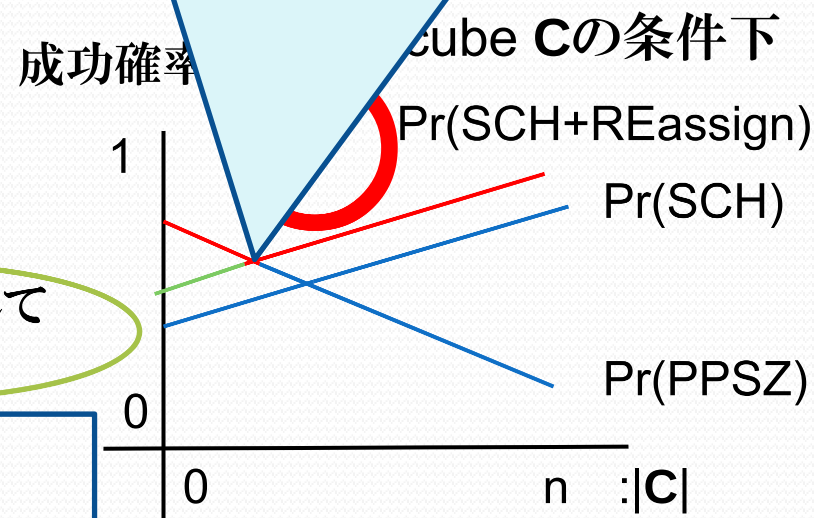
$$\Pr(PPSZ \mid C) \ge 2^{-0.38770n+0.51854|C|}$$

$$\Pr(SCH \mid C) \ge \left(\frac{3}{4}\right)^{n-|C|}$$

$$\Pr(ISTT) \geq \Omega(1.32113^{-n})$$

改良アイデア修正版

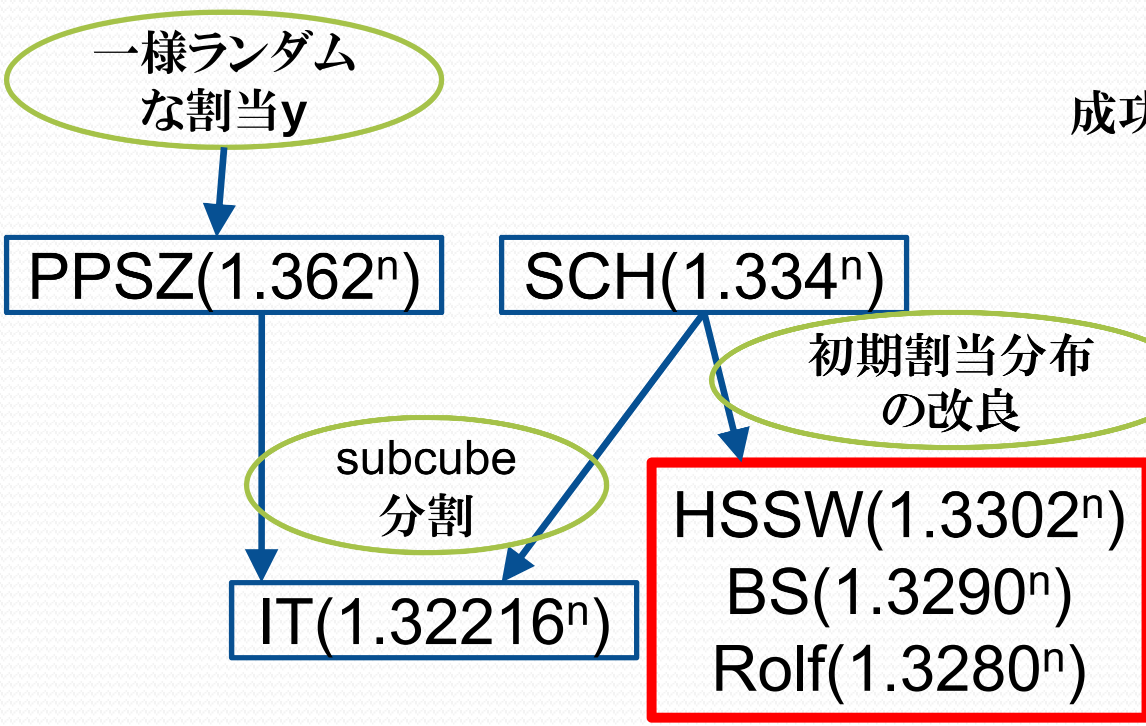




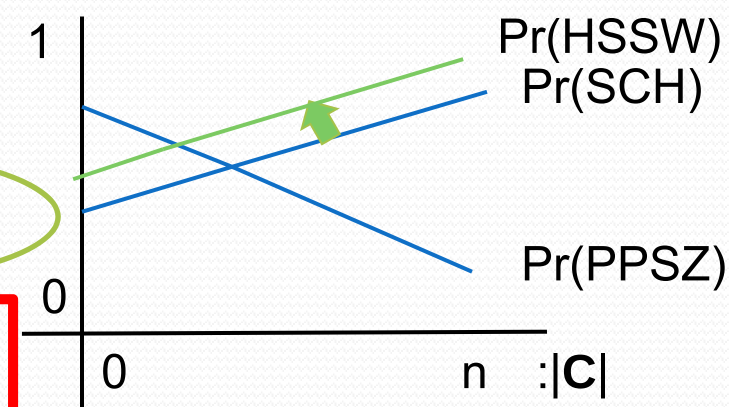
$$\Pr(PPSZ \mid C) \geq 2^{-0.38770n + 0.51854|C|}$$

$$\Pr(SCH \mid C) \geq \left(\frac{3}{4}\right)^{n-|C|}$$

SCH改良アルゴリズムの復習



成功確率　y∈subcube Cの条件下



$$\Pr(PPSZ \mid C) \geq 2^{-0.38770n+0.51854|C|}$$

$$\Pr(SCH \mid C) \geq \left(\frac{3}{4}\right)^{n-|C|}$$

HSSWアルゴリズム

SCHの初期割り当てをFの節を利用して改良

(x1∨x2∨x3)

(0	0	1)	4/21
(0	1	0)	4/21
(0	1	1)	2/21
(1	0	0)	4/21
(1	0	1)	2/21
(1	1	0)	2/21
(1	1	1)	3/21

{y1,y2,y3}={0,0,0}は解になり得ない

最適な割り当て確率

$$E_y\left[\left(\frac{1}{2}\right)^{d(y,z)}\right] \quad : \left(\frac{3}{4}\right)^3 \Rightarrow \left(\frac{3}{7}\right)$$

(0.421875) (0.4285714)

HSSWアルゴリズム

Fの極大独立集合Mの全ての節に対して割り当てる

$$M=\overbrace{(x1 \lor x2 \lor x3)(\overline{x4} \lor x6 \lor \overline{x7})(\overline{x5} \lor \overline{x8} \lor x10)\cdots}^{m\text{節}}$$

$(x1 \lor x2 \lor x3)$

$$\left.\begin{array}{llll} (0 & 0 & 1) & 4/21 \\ (0 & 1 & 0) & 4/21 \\ (0 & 1 & 1) & 2/21 \\ (1 & 0 & 0) & 4/21 \\ (1 & 0 & 1) & 2/21 \\ (1 & 1 & 0) & 2/21 \\ (1 & 1 & 1) & 3/21 \end{array}\right\}$$

最適な割り当て確率

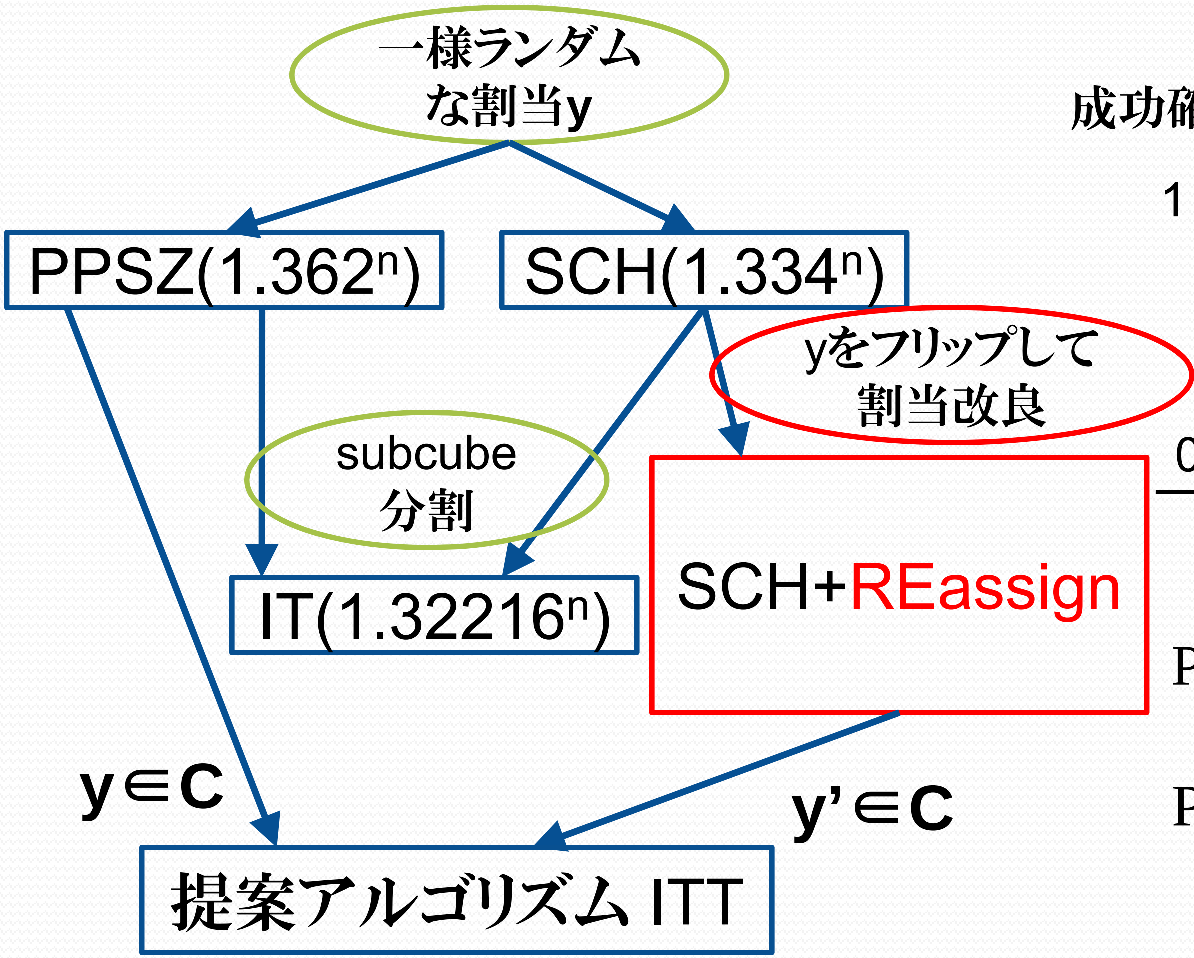
なり得ない

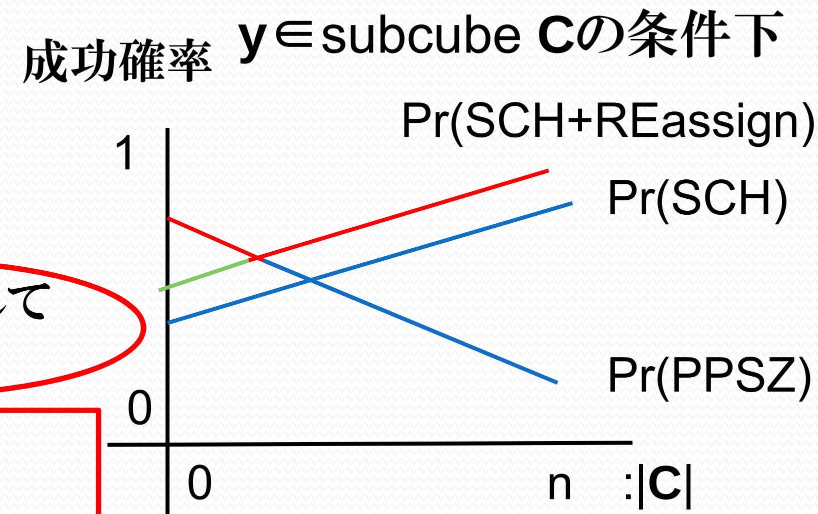
$$\Pr(BF) \geq \left(\frac{1}{6}\right)^m$$ [BS03]

この時のSCHの成功確率は

$$\Pr(SCH) \geq \left(\frac{64}{63}\right)^m \cdot \left(\frac{3}{4}\right)^n$$

アルゴリズムITT





$$\Pr(PPSZ \mid C) \ge 2^{-0.38770n+0.51854|C|}$$

$$\Pr(SCH \mid C) \ge \left(\frac{3}{4}\right)^{n-|C|}$$

REassignアルゴリズム

一様ランダムでの割り当て確率

(x1∨x2∨x3)

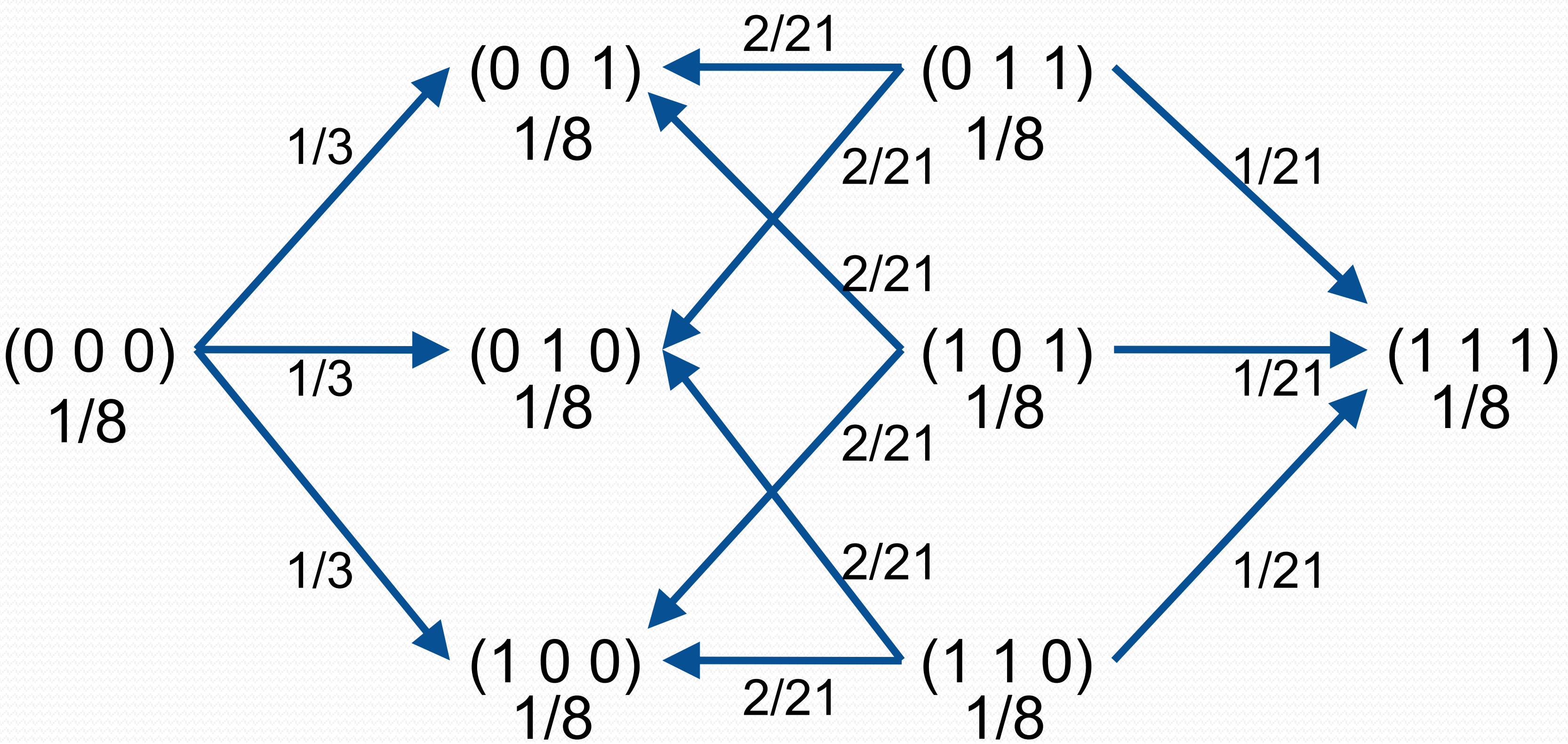
(0	0	0)1/8
(0	0	1)1/8
(0	1	0)1/8
(0	1	1)1/8
(1	0	0)1/8
(1	0	1)1/8
(1	1	0)1/8
(1	1	1)1/8

良い割り当て確率

(x1∨x2∨x3)

(0	0	1)4/21
(0	1	0)4/21
(1	0	0)4/21
(0	1	1)2/21
(1	0	1)2/21
(1	1	0)2/21
(1	1	1)3/21

REassignアルゴリズム



REassignアルゴリズム

(0 0 1)
4/21

(0 1 1)
2/21

(0 0 0)
0

(0 1 0)
4/21

(1 0 1)
2/21

(1 1 1)
3/21

(1 0 0)
4/21

(1 1 0)
2/21

SCHに良い初期割り当ての確率になる。
Reassignを行った後、SCHアルゴリズムを行う

REassignをしたSCHの$E_{y\in C}\left[\left(\frac{1}{2}\right)^{d(y,z)}\right]$を計算する

$$E_{y\in C}\left[\left(\frac{1}{2}\right)^{d(y,z)}\right] \ge \left(\frac{64}{63}\right)^{t1}\cdot\left(\frac{181}{192}\right)^{t2}\cdot\left(\frac{379}{384}\right)^{t3}\cdot\left(\frac{29}{32}\right)^{t4}\cdot\left(\frac{29}{32}\right)^{t5}$$

$$\cdot\left(\frac{63}{64}\right)^{t6}\cdot\left(\frac{63}{64}\right)^{t7}\cdot\left(\frac{117}{128}\right)^{t8}\cdot\left(\frac{63}{64}\right)^{t9}\cdot\left(\frac{4}{3}\right)^{|C|}\cdot\left(\frac{3}{4}\right)^{n}$$

$$m = t2+t3+t4+t5+t6+t7+t8+t9$$

$$|C| = t2+t3+2(t4+t5+t6)+3(t7+t8+t9)$$

$$0 \le t2,t3,t4,t5,t6,t7,t8,t9 \le m$$

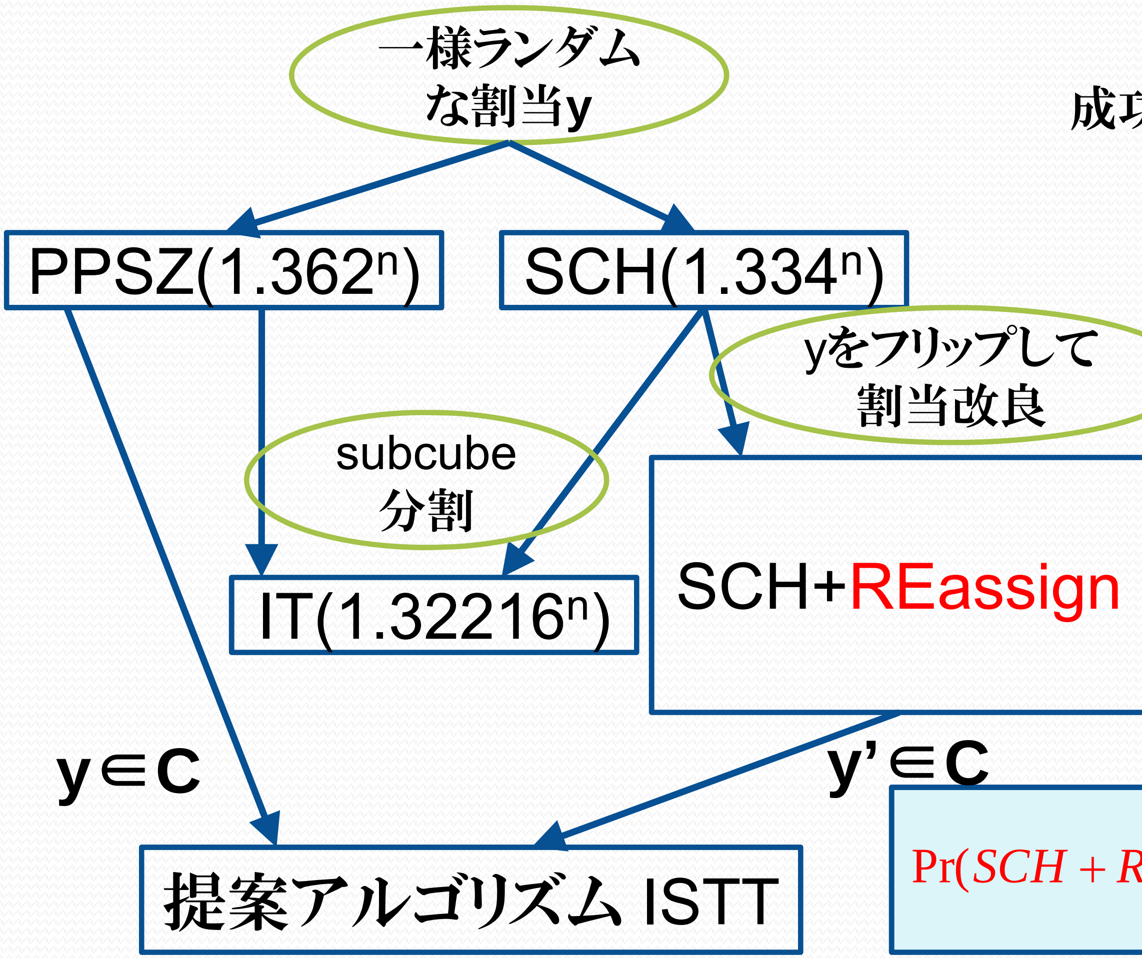
$$0 \le t2,t3 \le d$$

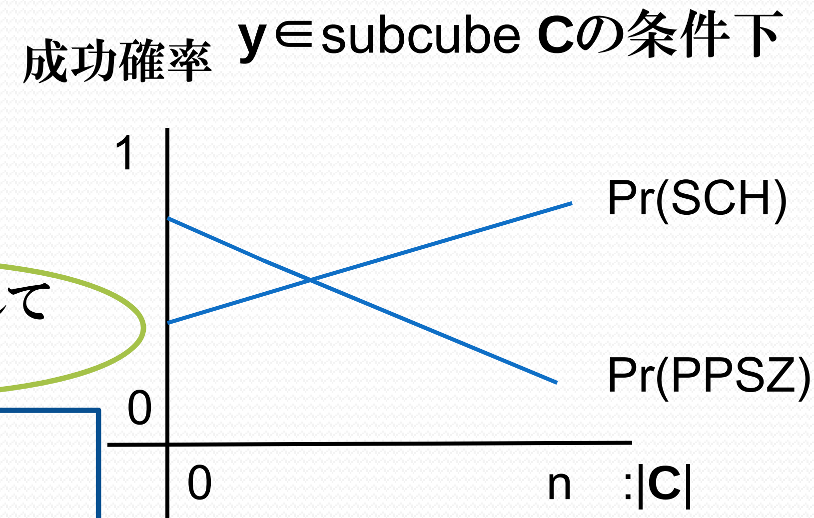
$$0 \le t4,t5,t6 \le \frac{d}{2}$$

$$0 \le t7,t8,t9 \le \frac{d}{3}$$

$$\Pr(SCH + REassign \mid C) \ge \left(\frac{64}{63}\right)^{m}\cdot\left(\frac{247}{191}\right)^{|C|}\cdot\left(\frac{3}{4}\right)^{n}$$

アルゴリズムITT

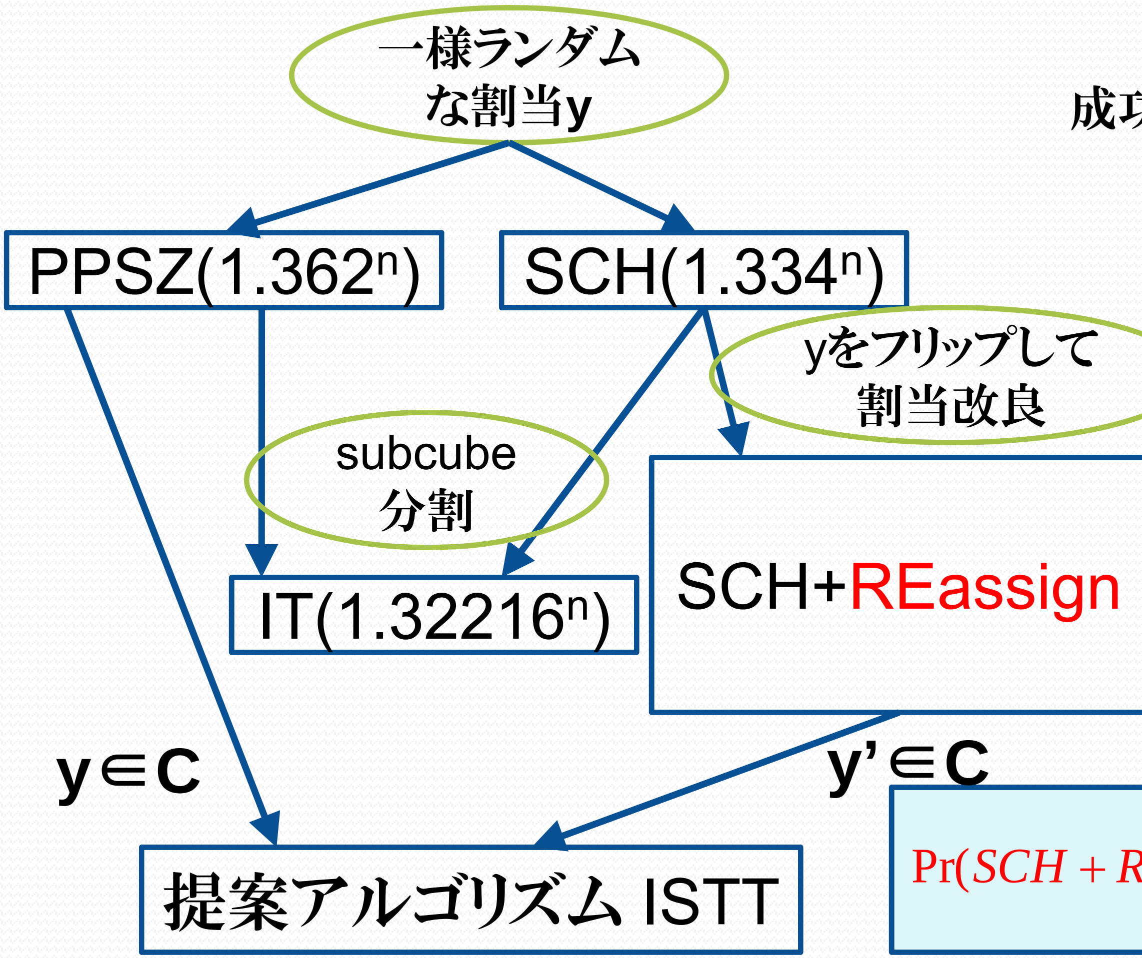


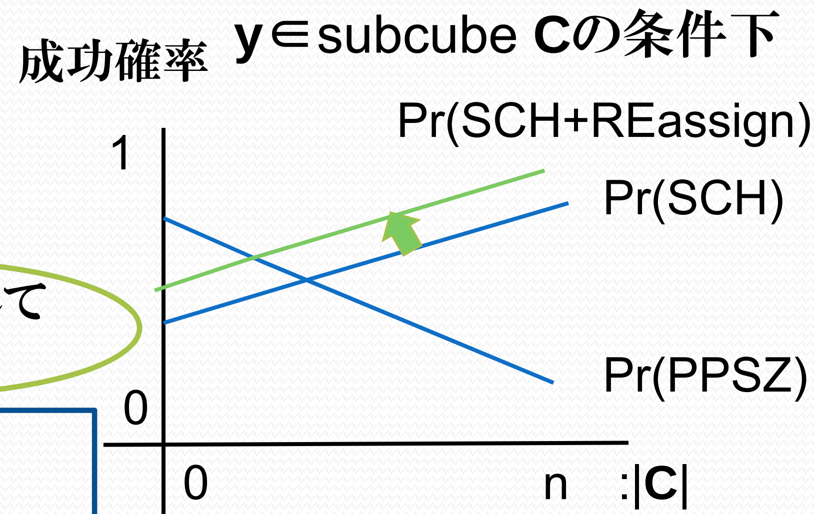


$$\Pr(PPSZ \mid C) \geq 2^{-0.38770n+0.51854|C|}$$

$$\Pr(SCH + REassign \mid C) \geq \left(\frac{64}{63}\right)^{m} \cdot \left(\frac{247}{191}\right)^{|C|} \cdot \left(\frac{3}{4}\right)^{n}$$

アルゴリズムITT



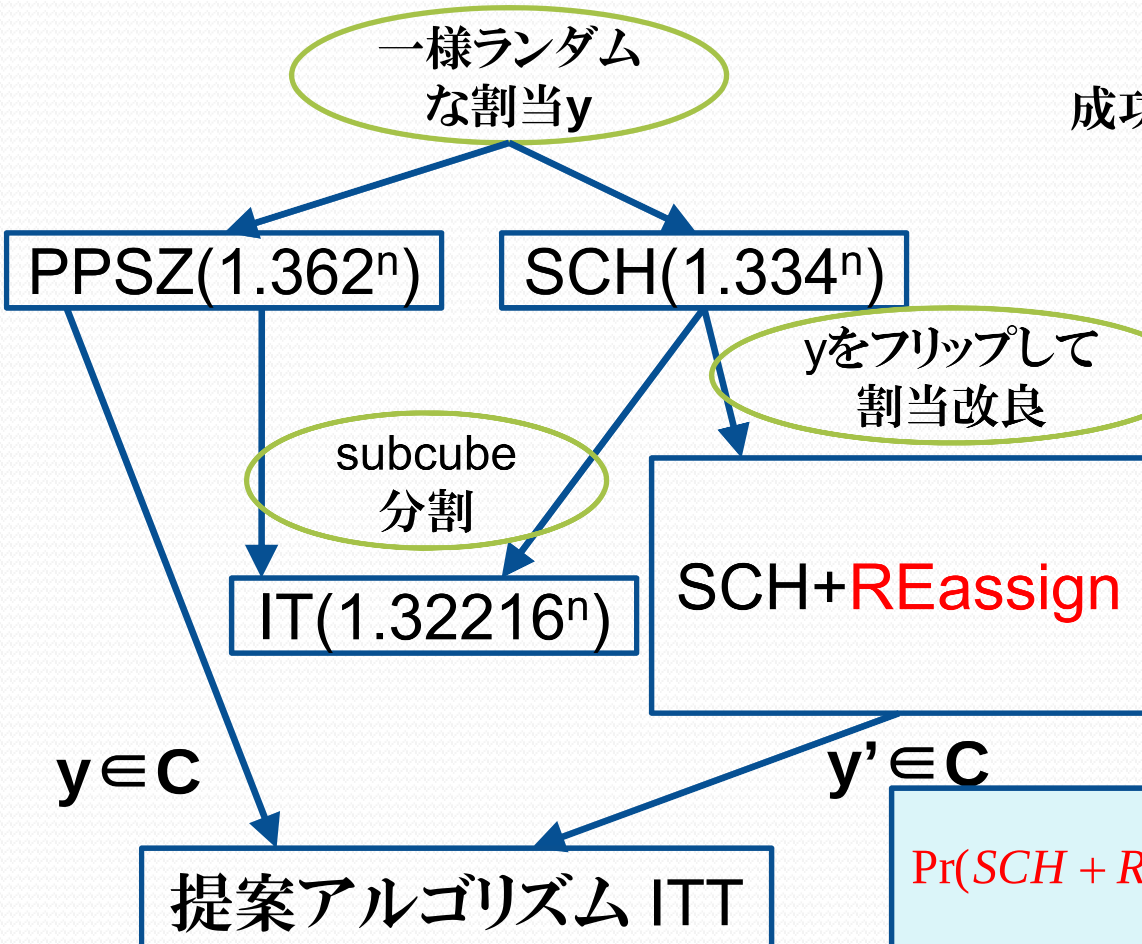


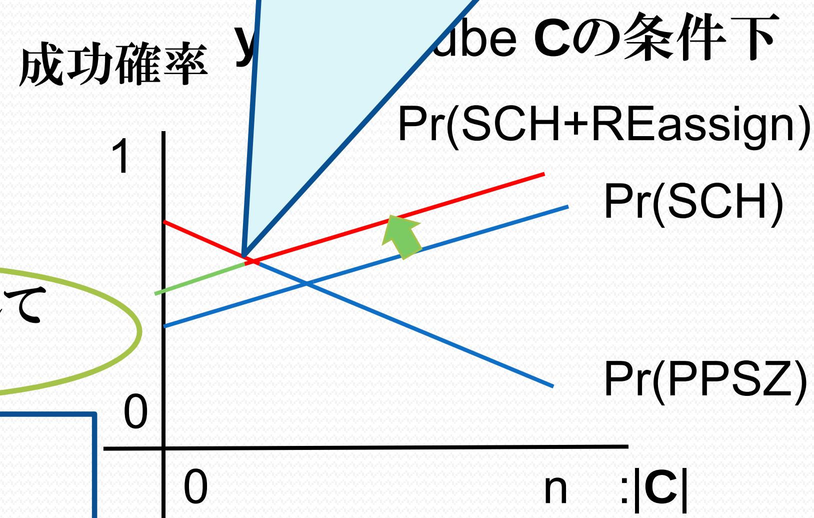
$$\Pr(PPSZ \mid C) \geq 2^{-0.38770n+0.51854|C|}$$

$$\Pr(SCH + REassign \mid C) \geq \left(\frac{64}{63}\right)^{m} \cdot \left(\frac{247}{191}\right)^{|C|} \cdot \left(\frac{3}{4}\right)^{n}$$

アルゴリズムITT

$$\Pr(ISTT) \ge \Omega(1.32113^{-n})$$





$$\Pr(PPSZ \mid C) \ge 2^{-0.38770n+0.51854|C|}$$

$$\Pr(SCH + REassign \mid C) \ge \left(\frac{64}{63}\right)^{m} \cdot \left(\frac{247}{191}\right)^{|C|} \cdot \left(\frac{3}{4}\right)^{n}$$

まとめ

PPSZ+BS？ PPSZ+Rolf？

2^n	自明な全探索	
⋮		
1.362^n	1998 R. Paturi, et.al.(PPSZアルゴリズム)	
1.334^n	1999 U. Schoning.(SCHアルゴリズム)	
1.3302^n	2002 T. Houfmeister, et.al. (HSSW)	
1.3290^n	2003 S. Baumer and R. Schuler. (BS)	(SCH改良)
1.3280^n	2003 D. Rolf. (Rolf)	
1.3238^n	2004 K. Iwama and S. Tamaki.	
1.32266^n	2005 S. Tamaki.	(SCH+PPSZ)
1.32216^n	2005 D. Rolf. (現在最速)	
1.32113^n	本結果 SCH+REassign(HSSW)+PPSZ	

ご清聴ありがとうございました