MAX SAT に対する近似アルゴリズムの実際的評価

堀 邦彰　　浅野 孝夫
中央大学 理工学部 情報工学科

MAX SAT (最大充足化問題) とは, 重み付きクローズ集合が与えられたときに, 満たされるクローズの重みの和が最大となるような真偽割当を見つけ出す問題である. 本論文は今までに提案されているいくつかの近似アルゴリズムをプログラム化し, 計算機実験を通してその理論的性能と実際的性能を比較, 考察したものである.

Practical Performances of Approximation Algorithms for MAX SAT

Kuniaki Hori　Takao Asano
Department of Information and System Engineering, Chuo University

MAX SAT (the maximum satisfiability problem) is stated as follows: given a set of clauses with weights, find a truth assignment that maximizes the sum of the weights of satisfied clauses. In this paper, we implement several exisiting approximation algorithms and compare their theoretical and practical performances.

1 序論

本論文では MAX SAT (最大充足化問題) について議論していく. MAX SAT とは重み付きクローズ集合が与えられたときに, 満たされるクローズの重みの和が最大となるような真偽割当を見つけ出す問題である. この問題は各クローズが高々 2 つのリテラルしか含まずかつ同一の重みを持つという制限を付けた MAX 2-SAT においても, NP 困難であることが知られており, したがって MAX SAT も NP 困難である. そのためこの問題に対する近似アルゴリズムが数多く提案されてきた. Johnson が確率的方法に基づく 0.5-近似アルゴリズムを提案した [7]. それを Yannakakis がネットワークの手法を取り入れ改良し 0.75-近似アルゴリズムを提案する [9] と Goemans-Williamson も線形計画法に基づく方法で 0.75-近似アルゴリズムを提案した [5]. 一方で半正定値計画法に基づく方法で Goemans-Williamson は MAX 2-SAT に対する 0.878-近似アルゴリズムを提案し, Johnson のアルゴリズムと組み合わせることにより 0.7584-近似アルゴリズムを提案した [6]. Asano-Ono-Hirata は Yannakakis の 0.75-近似アルゴリズムと半正定値計画法に基づく Goemans-Williamson のアルゴリズムとを組み合わせることにより 0.765-近似アルゴリズムを提案した [2]. Asano-Hori-Ono-Hirata は Yannakakis のアルゴリズムを詳細化し Goemans-Williamson のアルゴリズムと組み合わせることにより 0.767-近似アルゴリズムを提案した [3]. そして, もっとも最近では Asano が Yannakakis の 0.75-近似アルゴリズムを更に詳細化することによって 0.770-近似アルゴリズムを提案している [1].

現在 MAX SAT に対する近似アルゴリズムの主な流れは半正定値計画法に基づくものである. しかしながら, 半正定値計画問題への定式化は変数の数を自乗に膨れあがらせるという欠点がある. したがって本論文では半正定値計画法以外の方法による近似アルゴリズムに注目し, プログラム化し計算機

実験をすることによりその性能を実際的に評価することを目的とする．各近似アルゴリズムの説明をする前にいくつかの定義および表記法を示す．

2 基本的な定義 および 表記法

MAX SAT のインスタンスは $(\mathcal{C}, w)$ によって定義される．ここで $\mathcal{C}$ はクローズ (clause) の集合であり，各クローズ $C \in \mathcal{C}$ は非負の重み $w(C)$ をもつリテラル (literal) の和である．リテラルとはブール変数 (以下，単に変数という) x_i の肯定形 x_i，あるいは否定形 $\bar{x}_i$ である．具体的には $(\mathcal{C}, w) = \{(x_1, 4), (x_2, 2), (\bar{x}_3, 6), (\bar{x}_1 \vee x_2, 8), (\bar{x}_2 \vee x_3, 2), (\bar{x}_1 \vee \bar{x}_2 \vee x_3, 6)\}$ といったようなインスタンスとなる．$X = \{x_1, x_2, \ldots, x_n\}$ を重み付きクローズ $(\mathcal{C}, w)$ の変数とする．ここで同一クローズには同じ変数はあらわれないと仮定する．すなわち $x_1 \vee x_1 \vee x_2$ のようなクローズはないと仮定する．各変数 $x_i \in X$ に対して，もし x_i が true ならば $x_i = 1$ とし，x_i が false ならば $x_i = 0$ とする．そうすることにより，$\bar{x}_i = 1 - x_i$ となり，$C_j \in \mathcal{C}$ は

$$C_j = C_j(\boldsymbol{x}) = 1 - \prod_{x_i \in X_j^+} (1 - x_i) \prod_{x_i \in X_j^-} x_i$$

という $\boldsymbol{x} = (x_1, x_2, \ldots, x_n)$ の関数と考えることができる．ここで X_j^+ は C_j に肯定形であらわれる変数の集合で X_j^- は否定形であらわれる変数の集合とする．$\bar{x}_{i_1} \vee x_{i_2} = \overline{x_{i_1} \wedge \bar{x}_{i_2}}$ であることを考えるとこの式は自然なものといえる．

こうして，任意の真偽割当 $\boldsymbol{x} \in \{0, 1\}^n$ に対し C_j は $C_j = C_j(\boldsymbol{x}) = 0$ または 1 となる．そして $C_j = 1$ のときにはクローズ C_j は満たされているといわれる．

真偽割当 $\boldsymbol{x}$ の値は

$$F(\boldsymbol{x}) = \sum_{C_j \in \mathcal{C}} w(C_j) C_j(\boldsymbol{x})$$

と定義される．すなわち真偽割当 $\boldsymbol{x}$ の値とは $\boldsymbol{x}$ によって $\mathcal{C}$ 内の満たされているクローズの重みの和である．こうして MAX SAT は $\boldsymbol{x}$ の値が最大となる $\boldsymbol{x}$ をみつける問題である．

A を MAX SAT に対するアルゴリズムとし $w^A(\mathcal{C})$ をインスタンス $\mathcal{C}$ に対してアルゴリズム A が作り出した真偽割当 $\boldsymbol{x}^A(\mathcal{C})$ の値とする．もし任意のインスタンス $\mathcal{C}$ に対して $w^A(\mathcal{C})$ が最適な真偽割当 $\boldsymbol{x}^*(\mathcal{C})$ の値 $w^*(\mathcal{C})$ の少なくとも α 倍であるならば，A は近似率 α をもつ近似アルゴリズムであるといわれる．近似率 α をもつ多項式時間アルゴリズム A は α-近似アルゴリズムであるといわれる．

多くの MAX SAT に対する近似アルゴリズムは確率的方法に基づいている．$p_i (0 \leq p_i \leq 1)$ を独立に各変数 $x_i \in X$ を true にする確率とし，$\boldsymbol{x}^p = (p_1, p_2, \ldots, p_n)$ をランダム真偽割当とする．そのときランダム真偽割当 $\boldsymbol{x}^p$ によってクローズ $C_j \in \mathcal{C}$ が満たされる確率は

$$C_j(\boldsymbol{x}^p) = 1 - \prod_{x_i \in X_j^+} (1 - p_i) \prod_{x_i \in X_j^-} p_i$$

となり，ランダム真偽割当 $\boldsymbol{x}^p$ の期待値は

$$F(\boldsymbol{x}^p) = \sum_{C_j \in \mathcal{C}} w(C_j) C_j(\boldsymbol{x}^p)$$

となる．確率的方法では，その値が少なくとも $F(\boldsymbol{x}^p)$ であるような真偽割当 $\boldsymbol{x}^q \in \{0, 1\}^n$ が存在することが知られており，実際に条件付き確率法によって多項式時間で求めることができる [2]．したがって次節から述べるアルゴリズムは求められたランダム真偽割当の期待値についての議論が主となっている．

3 Johnson の 0.5-近似アルゴリズム

Johnson の 0.5-近似アルゴリズムは確率的方法に基づいている．そのアルゴリズムは非常に単純であり，理解しやすい．各変数 x_i が true になる確率を 1/2 とするだけである．すなわち $\boldsymbol{x}^p = (1/2, 1/2, \ldots, 1/2)$ とする．そのときランダム真偽割当て $\boldsymbol{x}^p$ により $C_j \in \mathcal{C}$ が満たされる確率は

$$C_j(\boldsymbol{x}^p) = 1 - \prod_{x_i \in X_j^+} \left(1 - \frac{1}{2}\right) \prod_{x_j \in X_j^-} \frac{1}{2}$$

となる．そのときランダム真偽割当て $\boldsymbol{x}^p$ の期待値は次式となる．

$$F(\boldsymbol{x}^p) \geq 0.5W_1^* + 0.75W_2^* + 0.875W_3^* + 0.938W_4^* + 0.969W_5^* + \sum_{k \geq 6} \left(1 - \frac{1}{2^k}\right) W_k^*$$

ここで $\mathcal{C}_k$ は $\mathcal{C}$ 内の k 個のリテラルを含むクローズの集合とする．また，$W_k^* = \sum_{C_j \in \mathcal{C}_k} w(C_j) C_j(\boldsymbol{x}^*)$ とする．こうして $F(\boldsymbol{x}^*) = \sum_{k \geq 1} W_k^*$ である．

Johnson のアルゴリズムでは多くのリテラルを含むクローズに対して良い近似であることに注意されたい. したがってどのクローズも少なくとも k 個のリテラルを含むと仮定した場合, 近似率は $1-2^{-k}$ となる.

4 Goemans-Williamson の 0.75-近似アルゴリズム

MAX SAT は次のような整数計画問題として定式化できる.

$$
\begin{aligned}
\max \quad & \sum_{C_j \in \mathcal{C}} w(C_j) z_j \\
\text{subject to:} \quad & \sum_{x_i \in X_j^+} y_i + \sum_{x_i \in X_j^-} (1-y_i) \geq z_j \\
& \qquad\qquad\qquad {}^{\forall} C_j \in \mathcal{C} \\
& y_i \in \{0,1\} \qquad {}^{\forall} x_i \in X \\
& z_j \in \{0,1\} \qquad {}^{\forall} C_j \in \mathcal{C}
\end{aligned}
$$

この定式化において, 変数 $\boldsymbol{y} = (y_i)$ は変数 $X = \{x_1, x_2, \ldots, x_n\}$ に対応し, $\boldsymbol{z} = (z_j)$ はクローズ $\mathcal{C}$ に対応する. このときもし $x_i = 1$ ならば, またそのときに限り $y_i = 1$ である. 同様にもし C_j が満たされるならば, またそのときに限り $z_j = 1$ となる. こうしてはじめの制約式はクローズ C_j 内のリテラルの少なくとも 1 つが true ならばまたそのときに限りクローズ C_j は満たされるということを意味している. したがってこの定式化は MAX SAT に正確に対応していることがわかる.

Goemans-Williamson は変数 $\boldsymbol{y}$ と $\boldsymbol{z}$ に対する {0,1} 制約を $0 \leq y_i, z_j \leq 1$ と緩和することにより線形計画問題へと緩和することを考えた.

緩和問題に対する最適解を $(\boldsymbol{y}^\#, \boldsymbol{z}^\#)$ とする. そのとき Goemans-Williamson はまず $x_i^p = y_i^\#$ とすることを考えた. すなわちランダム真偽割当 $\boldsymbol{x}^\#$ を $\boldsymbol{x}^\# = \boldsymbol{y}^\# = (y_i^\#)$ とするわけである. そのとき k 個のリテラルを含むクローズ C_j が満たされる確率は

$$C_j(\boldsymbol{y}^\#) \geq \left(1 - \left(1 - \frac{1}{k}\right)^k\right) z_j^\#$$

となる. したがって, ランダム真偽割当 $\boldsymbol{x}^\# = \boldsymbol{y}^\#$ の期待値は次式となる.

$$
\begin{aligned}
F(\boldsymbol{x}^\#) \geq{} & W_1^* + 0.75W_2^* + 0.704W_3^* + 0.684W_4^* \\
& +0.672W_5^* + \sum_{k \geq 6} \left(1 - \left(1 - \frac{1}{k}\right)^k\right) W_k^*
\end{aligned}
$$

こうして線形計画法によって得られた解をそのままランダム真偽割当に用いた場合の近似率は $1 - 1/e$ (≈ 0.632120) であることがわかる. 線形計画法を用いたアルゴリズムでは少ないリテラルを含むクローズに対して良い近似であることに注意されたい. この性質は Johnson のアルゴリズムとちょうど反対の性質である. そこで Goemans-Williamson は線形計画法を用いて求められた解と Johnson のアルゴリズムによって求められた解の良い方を選ぶことを考えた. そのときの期待値は

$$
\begin{aligned}
F(\boldsymbol{x}^\#) \geq{} & 0.75W_1^* + 0.75W_2^* + 0.789W_3^* + 0.810W_4^* \\
& +0.820W_5^* + \sum_{k \geq 6} \beta_k W_k^*
\end{aligned}
$$

となる. ここで $2\beta_k = (1 - \frac{1}{2^k}) + (1 - (1 - \frac{1}{k})^k)$ である.

また, Goemans-Williamson は線形計画法によって得られた解 $\boldsymbol{y}^\#$ をそのままランダム真偽割当に用いるのではなく, 適切に定義した関数 $f : [0,1] \to [0,1]$ を用いて $x_i^p = f(y_i^\#)$ とすることによってランダム真偽割当を生成することを考えた. Goemans-Williamson が提案した関数は以下の 3 つの関数である.

$$1 - 4^{-y} \leq f_1(y) \leq 4^{y-1}$$

$$f_2(y) = \alpha + (1 - 2\alpha)y \qquad \left(2 - \frac{3}{\sqrt[3]{4}} \leq \alpha \leq \frac{1}{4}\right)$$

$$
f_3(y) = \begin{cases}
\frac{3}{4}y + \frac{1}{4} & \text{if } 0 \leq y \leq \frac{1}{3} \\
\frac{1}{2} & \text{if } \frac{1}{3} \leq y \leq \frac{2}{3} \\
\frac{3}{4}y & \text{if } \frac{2}{3} \leq y \leq 1
\end{cases}
$$

これらの関数 $f(y)(= f_1, f_2, f_3)$ は Goemans-Williamson の定義した次の性質 $\frac{3}{4}$ を満たしている.

$$1 - \prod_{x_i \in X_j^+} (1 - f(y_i^\#)) \prod_{x_i \in X_j^-} f(y_i^\#) \geq \frac{3}{4} z_j^\#$$

したがって Johnson のアルゴリズムと合わせることなく単独で近似率 0.75 を実現している. また, 関数 $f_2(y)$ で $\alpha = 1/4$ とすることにより Johnson のアルゴリズムと線形計画問題によって得られた解の良い方を選ぶといった方法と同じ期待値を得ることができる. しかしながら, 各変数 x_i を独立に設定できるという点で優れている.

5 Yannakakis の 0.75-近似アルゴリズム

Yannakakis は Johnson の 0.5-近似アルゴリズムの改良を考えた. Johnson のアルゴリズムはすべてのクローズが少なくとも 2 個のリテラルを含む場合 0.75-近似アルゴリズムとなる. したがって Johnson のアルゴリズムでの 1 個のリテラルからなるクローズの扱いを変えることを考えた. 簡単に説明すると x_1 のようなクローズに対しては $x_1^p = 3/4$ とするわけである. しかしながら $x_1^p = 3/4$ とすることにより $\bar{x}_1 \in \mathcal{C}$ の場合に $\Pr(\bar{x}_1 = 1) = 1/4$ となってしまったり, リテラルを 2 つ以上含むクローズが満たされる確率が 3/4 未満になってしまうということが生じる. そういったことを避けるために Yannakakis はクローズ集合を表現するネットワークを構築しネットワークアルゴリズムを駆使することによりそれらの不具合を解消した. 具体的には $x \vee y$ というクローズを $(\bar{x}, y), (\bar{y}, x)$ の 2 本の枝に対応させるなどのことを行なうことにより, ネットワークを構築する. そして, 制約付き最大フローを流すことによって, 問題を引き起こすクローズ集合をクラス分けしたのである. そして, 変数の集合を P, P', Q, Z に分け $x_i \in P'$ に対しては $x_i^p = 3/4$, $x_i \in (P - P') \cup Q$ に対しては $x_i^p = 5/9$, $x_i \in Z$ に対しては $x_i^p = 1/2$ とした. それにより

$$F(\boldsymbol{x}^p) \geq 0.75W_1^* + 0.75W_2^* + 0.75W_3^* + 0.766W_4^* + 0.763W_5^* + \sum_{k \geq 6} \left(1 - \left(\frac{3}{4}\right)^k\right) W_k^*$$

の近似率を達成した.

また Asano-Hori-Ono-Hirata は 3 段階からなる Yannakakis のアルゴリズムを 4 段階へと詳細化することにより

$$F(\boldsymbol{x}^p) \geq 0.75W_1^* + 0.75W_2^* + 0.775W_3^* + 0.789W_4^* + 0.810W_5^* + \sum_{k \geq 6} \left(1 - \left(\frac{3}{4}\right)^k\right) W_k^*$$

の近似率を達成した. 3, 4, 5 個のリテラルを含むクローズに対する近似率が改善されたのである.

Yannakakis のアルゴリズムにおいて重要な概念に**等価** (equivalent) という概念がある. それは同じ変数を持つ 2 つのインスタンス $(\mathcal{C}, w)$ と $(\mathcal{C}', w')$ 上で定義されており, もし全ての真偽割当において $(\mathcal{C}, w)$ と $(\mathcal{C}', w')$ が同じ値ならばそのとき 2 つのインスタンス $(\mathcal{C}, w)$ と $(\mathcal{C}', w')$ は等価であるという. また Asano-Hori-Ono-Hirata はその概念の拡張である**強等価** (strongly equivalent) を導入した. それは任意のランダム真偽割当に対して $(\mathcal{C}, w)$ と $(\mathcal{C}', w')$ が同じ期待値ならば強等価であるというものである. 次の補題は Yannakakis のアルゴリズムおよびその詳細化アルゴリズムにおいて重要な役割を果たしている.

補題 1 以下の全てのクローズは同じ重みを持つとする. このとき $\mathcal{A} = \{\bar{x}_i \vee x_{i+1} | i = 1, \ldots, k\}$ と $\mathcal{A}' = \{x_i \vee \bar{x}_{i+1} | i = 1, \ldots, k\}$ は強等価である. ただし $k+1 = 1$ とする. また, $\mathcal{B} = \{x_1\} \cup \{\bar{x}_i \vee x_{i+1} | i = 1, \ldots, l\}$ と $\mathcal{B}' = \{x_i \vee \bar{x}_{i+1} | i = 1, \ldots, l\} \cup \{x_{l+1}\}$ は強等価である.

6 実験結果

前節までに説明したアルゴリズムに関して実験を行なった. 使用機種は Ultra Sparc-1 143MHz Memory 128MB である. 実験データは 2nd DIMACS Implementation Challenge の SAT インスタンスクラス jnh を加工したものであり, GRASP の実験に使われたものである [8]. 更にそれらのインスタンスに手を加えインスタンスはすべて 100 変数 800 から 950 のクローズから構成されるようにした. データは 7 クラスあり, 1 クラス 44 インスタンス, 計 308 個のインスタンスを用いた. DATA0 はクローズの重みは 1 から 1000 までの整数値, 2 から 14 個のリテラルを含む. DATA1 は DATA0 のデータに 1 個のリテラルからなるクローズを 50 個追加したものである. DATA2 は DATA1 の重みをすべて 1 にし, 同様に DATA3 は DATA0 の重みをすべて 1 にしたものである. DATA4 は DATA1 の 3, 4, 5 個のリテラルを含むクローズに対し 900 から 1000 の重みを偏らせて与え, その他には 1 から 100 までの重みを与えた. 同様に DATA5 は DATA1 の 8 個以上のリテラルを含むクローズに重みを偏らせて与えた. 最後 DATA6 は DATA1 のデータから 1, 2, 3 個のリテラルを含むクローズのみを抜きだしたものである. また正確な近似率を求めるため, 最適解は XPRESS-MP を用い整数計画問題に従い分枝限定法により得た. 図 1 は各アルゴリズムによって得られたランダム真偽割当の期待値のクラスごとの平均値である. どのクラスにおいても期待値が最

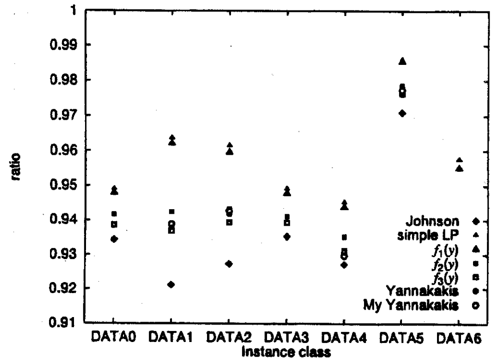

図 1. ランダム真偽割当の期待値の近似率

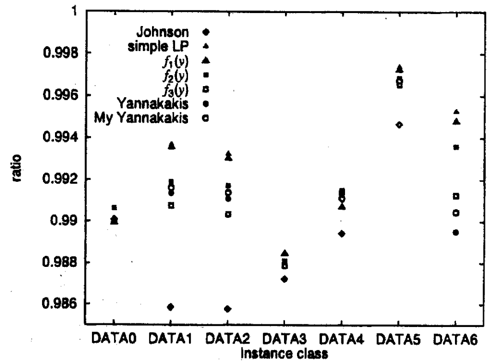

図 2. 真偽割当の値の近似率 (丸めは $x_1, x_2, \ldots$)

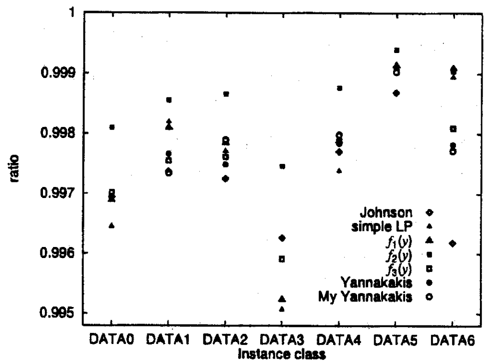

図 3. 真偽割当の値の近似率 (丸めは Greedy)

も良かったアルゴリズムは線形計画法の解をそのままランダム真偽割当に用いるものであった. 一方, 期待値の平均値が最も悪かったのは Johnson のアルゴリズムであった. 図 1 には, 表示していないが, 特に MAX 3-SAT のインスタンスである DATA6 に対する Johnson のアルゴリズムの期待値の平均値は 0.8091 と, 他に比べ非常に小さなものとなっていた.

次に, ランダム真偽割当から {0,1} 真偽割当を求めることを考える. ランダム真偽割当の期待値以上の真偽割当を求める方法に, 条件付き確率法がある. それはランダム真偽割当 $\boldsymbol{x}^p$ を用いて, $x_1, x_2, \ldots$ という順に真偽割当を定めていくものである. $\boldsymbol{x}^p = (p_1, p_2, \ldots, p_n)$ の p_1 を 0 または 1 にした $\boldsymbol{x}^p_{10} = (0, p_2, \ldots, p_n)$ と $\boldsymbol{x}^p_{11} = (1, p_2, \ldots, p_n)$ を考える. もし p_1 以外の p_i を定数だと考えたならば $F(\boldsymbol{x}^p) = \sum_{C_j \in C} w(C_j) C_j(\boldsymbol{x}^p)$ は p_1 の線形関数となる. したがって $\boldsymbol{x}^p_{10}$ または$\boldsymbol{x}^p_{11}$ で最大値をとる. $\boldsymbol{x}^p_{10}$ と $\boldsymbol{x}^p_{11}$ で大きい方を $\boldsymbol{x}^q_1$ とする. そのとき $\boldsymbol{x}^q_1$ の期待値は $\boldsymbol{x}^p$ の期待値 $F(\boldsymbol{x}^p)$ 以上になっている. x_1 の値を $\boldsymbol{x}^q_1$ の $x_1 \in \{0,1\}$ に固定し, 同様に$\boldsymbol{x}^p_{20} = (x_1, 0, p_3, \ldots, p_n)$ と $\boldsymbol{x}^p_{21} = (x_1, 1, p_3, \ldots, p_n)$ を考え再帰的に行なうことによって少なくとも $F(\boldsymbol{x}^p)$ の値の真偽割当 $\boldsymbol{x}^q \in \{0,1\}^n$ を得ることができる. 条件付き確率法によって真偽割当をもとめ, その値をとったものが図 2 である. 図 1 と比べると, 近似率は格段に良くなっていることがわかる. MAX 3-SAT のインスタンスである DATA6 の Johnson のアルゴリズムにいたっては, 図 1 には示すことができなかったが, ランダム真偽割当の期待値 0.80911, 丸め後の真偽割当の値 0.97502 と, 平均値で 0.1659 もの差が生じている. また, ランダム真偽割当から真偽割当に丸めるだけで, 近似率は少なくとも 0.01 程度は良くなっていた.

この丸めの方法はランダム真偽割当の期待値以上の真偽割当を求めるということのみを目的としている. よりよい真偽割当を求めることを考えていないわけである. したがって, よりよい真偽割当を求めるために, 真偽割当を求めるときの丸めの順序を変えてみることを考える. 丸め方は $x_1, x_2, \ldots$ という順序である必要はなく, どのような順序で丸めたとしてもランダム真偽割当の期待値以上の値は得られるはずである. したがって, 現在まだ丸められていない変数 x^p_i の中で, それを丸めたときに得られる値が最も大きいものを Greedy に選び丸めることを考える. 図 3 に結果を示す. 真偽割当の値の平均値が, 少なくとも 0.995 以上という結果であり, Greedy に丸めた効果は非常に大きなものであった.

更に, Greedy に丸めることの良さを示すために, 比較対象として $x_1, x_2, \ldots$ と丸めた解に対して局

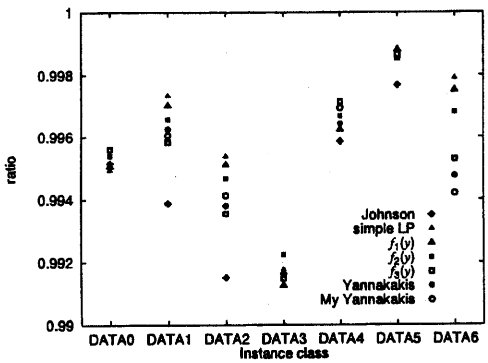

図 4. 真偽割当の値の近似率 (局所改善)

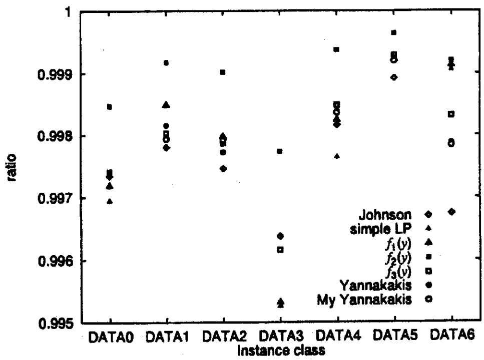

図 5. 真偽割当の値の近似率 (局所改善)

所改善を行なったものを考える. 局所改善の方法は $\boldsymbol{x} \in \{0,1\}^n$ に対して $||\boldsymbol{x}-\boldsymbol{y}||_2 = 1$ であるようなすべての $\boldsymbol{y} \in \{0,1\}^n$ を調べ $f(\boldsymbol{y}) > f(\boldsymbol{x})$ ならば現在の真偽割当 $\boldsymbol{x}$ を $\boldsymbol{y}$ に置き換えるといったものである. したがって得られた真偽割当 $\boldsymbol{x}$ は $||\boldsymbol{x}-\boldsymbol{y}||_2 = 1$ であるようなすべての $\boldsymbol{y}$ に対して $f(\boldsymbol{x}) \geq f(\boldsymbol{y})$ をみたす極大真偽割当である. $x_1, x_2, \ldots$ と丸めた解に対して局所改善を行なったものが図 4 である. 局所改善により, 平均で 0.00417 程度の改善が行なわれていた. DATA6 の Johnson のアルゴリズムに対する真偽割当が 0.0134 と, 最も改善されている.

Greedy に丸めた真偽割当との比較を行なうと $x_1, x_2, \ldots$ と丸めた後に局所改善を行なった結果よりも Greedy に丸めた結果の方が, 平均で 0.00232 程度良いという結果を得た. ここで重要なことは局所改善を行なうためにかかる時間は, 最悪の場合, 多項式時間で抑えることができないという点である. Greedy に丸めるときにかかる時間は多項式時間で抑えることができる. したがって, Greedy に丸めることは近似率が良いというだけでなく, 計算時間の点でも優れているのである.

さらに追加実験として, Greedy に丸めた真偽割当に対しても局所改善を行なった. 結果を図 5 に示す. 平均で 0.000295 と改善の度合は非常に小さくなっている. また, 最大でも 0.000620 の 改善であった. この結果は, やはり Greedy に丸めることの良さを示しているものである. 局所改善後の結果についても, $x_1, x_2, \ldots$ の順に丸めた真偽割当に対する局所改善によって得られた解の最悪の近似率は 0.9830 であったのに対して, Greedy に丸めた真偽割当に対する局所改善は最悪でも近似率 0.9912 となっており, Greedy に丸めることは有効であるといえるであろう.

以上の実験結果に対し考察を行なう. まず, 各アルゴリズムに関して考察を行なう. Johnson の 0.5-近似アルゴリズムは, 期待値についていえば, $x_i^p = 1/2$ としてだけなのでクローズに含まれるリテラルの個数と密接な関係をもっていることは明らかであり, 実際に期待値はクローズに含まれるリテラル数の荷重平均と良い相関をもっていた. したがって, MAX 3-SAT のインスタンスである DATA6 に対しては 0.80911 と小さく, DATA5 に対しては 0.97107 と大きくなっていた. また, 1 個のリテラルからなるクローズが含まれるインスタンスとそうでないインスタンス (その他のクローズの重みは等しい) とを比べると近似率で 0.01 程度の差が生じていた.

線形計画問題に基づく Goemans-Williamson のアルゴリズムでは, 線形計画によって得られたランダム真偽割当の期待値が非常に良く, 近似率 0.75 を達成するために Johnson のアルゴリズムと組み合わせているといった点で, 期待値ではかえって損をしている. しかしながら提案されている関数 $1-4^{-y} \leq f_1(y) \leq 4^{y-1}$ (実際には $y \leq 0.5$ のとき $f(y) = 1-4^{-y}$, $y > 0.5$ のとき $f(y) = 4^{y-1}$ としている) が非常によく線形計画問題のみによって得られるランダム真偽割当 $\boldsymbol{y}$ を追従しており, この実験に用いたインスタンスに対して, 近似率 0.75 を保証しているアルゴリズムの期待値としては最も良いアルゴリズムとなっていた. 興味深い結果としては, DATA5 に関するものである. 線形計画問題に基づくアルゴリズムでは, 多くのリテラルを含むクローズに対して近似が悪くなるのだが, 多くのリ

テラルを含むクローズに対して大きな重みを与えたDATA5 に対しても線形計画問題に基づくアルゴリズムは他のアルゴリズムよりも良い結果を得ていた．原因としては，近似率の解析の際に相加相乗平均の関係式を用いている点があげられる．

Yannakakis のアルゴリズムおよびその詳細化アルゴリズムに関しては，今回実験に用いたインスタンスではその期待値において大きな違いはみられなかった．詳細化により 3 から 5 個のリテラルを含むクローズが満たされる確率が大きくなっているわけだが，重みを 3 から 5 個のリテラルに偏らせたインスタンスに対しても大きな違いはみられなかった．原因としては Yannakakis のアルゴリズムによってクラス分けされるクローズ集合が，10 から 25 個程度と，全体のクローズ数に対して非常に少なかったということが考えられる．しかしながら，近似率でわずか 0.001 程度ではあるが，詳細化アルゴリズムの方がよい近似率をもつ解を作り出していた．その原因はクラス分けされたクローズ集合の数と関係が深いであろうと予想したものの，実際大きな相関はみられなかった．

次に，ランダム真偽割当の丸めに関して考察を行なう．注目してほしい点は，丸めた後の真偽割当の値は期待値の善し悪しに完全には対応してはいないといった点である．特に，Greedy に丸めた結果では，線形計画法の解をそのままランダム真偽割当に用いるものよりも，関数 $f_2(y)$ を用いたものが優れているといった興味深い結果を得ていた．しかしながら，$x_1, x_2, \ldots$ の順に丸めたものに関しては，どちらかというと期待値の良さがそのまま反映されているようであった．Greedy に丸めた結果に対して興味深い点といえば，1 つのリテラルを含むクローズがない DATA0, DATA3 で，線形計画法によって得た解をそのままランダム真偽割当とした場合が，もっとも悪くなっているといった点である．1 つのリテラルからなるクローズに対する近似率が 1 と非常に高い線形計画法によって得られた解の期待値の性質を，反映しているようである．逆に，DATA0, DATA3 に対して，Johnson のアルゴリズムの近似率は 0.75 となる．線形計画法によって得られた解をそのままランダム真偽割当として用いた場合，その近似率は $1 - 1/e$ であり，DATA0, DATA3 では，Johnson のアルゴリズムの方が近似率は良いことになる．$x_1, x_2, \ldots$ に丸めた結果においては，DATA3 にその性質があらわれていないが，Greedy に丸めた結果ではその性質があらわれているようである．

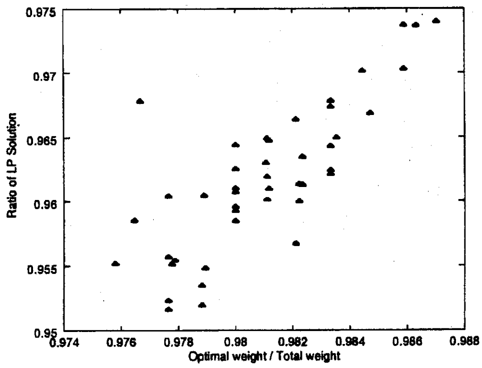

図 6. 総重みに対する最適解の割合と LP の期待値との近似率の相関

インスタンスと近似率の関係で興味深い結果を得たので紹介する．それは，クローズの総重みに対する最適解の割合が線形計画問題によって得られるランダム真偽割当の近似率と非常に良い正の相関をもっていたというものである．特に重みをすべて 1 にした 1 個だけのリテラルからなるクローズを含むインスタンスの集合に対しては良かった．(図 6) ただし，多くのリテラルを含むクローズに重みを偏らせたインスタンスについては，それほど目立った相関は見られなかった．

以下，計算機実験に関して工夫した点を述べる．Greedy に丸めを行なう，あるいは局所改善を行なうときに，その真偽割当の値，もしくは期待値を求めるといったことを繰り返し行なわなければならない．どちらの場合にも各クローズは高々 k 個のリテラルを含むとすると 1 個のクローズの値を計算する時間は $O(k)$ 必要で m クローズを計算する，すなわち，$f(\boldsymbol{x})$ を計算する時間は $O(km)$ となる．Greedy に丸めを行なった場合，1 つの変数を 0 または 1 に丸めるために他のまだ丸められていない変数に対しても丸めを行なうため，全く工夫をしない場合は $O(kmn^2)$ の計算時間が必要となる．計算時間を削減するために次のように $f(\boldsymbol{x})$ の計算を行なった．Greedy に丸めを行なうあるいは局所改善を行なうときには，必ずそれまでの $f(\boldsymbol{x})$ が求められているはずである．丸めあるいは改善を行なう際，変化するのは真偽割当 $\boldsymbol{x}$ 中の 1 つの変数 x_i のみである．したがって x_i が使われているクローズの

値のみを計算すれば十分である. 変数 x_i はクローズ集合 C の中で平均的には $O(km/n)$ 個のクローズによって使われており, 一回の $f(\boldsymbol{x})$ に必要な平均計算時間は $O(km/n)$ ということになる. したがって Greedy に丸めを行なう場合の平均計算時間は $O(kmn)$ となる.

最後に, 今回実験したインスタンスに対する結果として, 各アルゴリズムによって得られた真偽割当の中から最も良い解を選んだとすると, 最悪でも 0.9949 の近似を達成していた.

7 結論

本論文では, Johnson の 0.5-近似アルゴリズム, Goemans-Williamson の線形計画問題に基づく 0.75-近似アルゴリズム, Yannakakis の 0.75-近似アルゴリズムおよびその詳細化アルゴリズムをプログラム化しその理論的性能と実際的性能を比較した. またプログラムを実際に組むことにより, 発見的手法ではあるが理論的には気付かないであろう丸め方についての議論を行なうことができた. また, インスタンスと各アルゴリズムの近似率の関係等, 実際に実験を行なわないと分からないであろういくつかの興味深い結果を得た.

理論的性能と実際的性能の比較に関しては, すべてのアルゴリズムで理論的性能よりもはるかに良い結果を得ていた. 特に線形計画法に基づくものが良い近似をしていることがわかった.

アルゴリズムの解析に用いられている理論的性能評価は, すべて確率的手法に基づき期待値によって行なわれている. したがって丸め方について議論するということがほとんどない. 本論文では, 単純ではあるが Greedy に丸めるといった手法を提案した. そして, 実際に実験を行なうことにより, その有効性も示した.

しかしながら, 今回実験したデータは, 基本的には GRASP で用いられたデータを加工したものであり, 重みの違いを除くと基本的には 88 個のデータしか扱っていない. したがって, この論文で得られた結果は偏った結果となっている可能性も高い. しかしながら, 一般性の高いインスタンスをつくるのは困難であり, 今後の課題である.

現在 MAX SAT の近似アルゴリズムの主流となっている半正定値計画問題に基づくアルゴリズムは変数の数が自乗に膨れるという欠点はあるものの, 線形計画問題に基づくアルゴリズムの性能から察するに非常に良いものであることが予想される. 一方で近似率の悪い単純なアルゴリズムにおいてもランダム真偽割当の丸め方によっては良い性能が得られていることを考えると, 丸め方についての取り組みも今後の課題である.

謝辞

本研究は一部, 文部省科学研究費補助金, 中央大学理工学研究所, 電気通信普及財団の援助のもとで行なわれたものである.

参考文献

[1] T. Asano, Approximation algorithms for MAX SAT: Yannakakis vs. Goemans-Williamson. *Proc. 5th Israel Symposium on Theory of Computing and Systems*, 1997, pp.24-37.

[2] T. Asano, T. Ono and T. Hirata, Approximation algorithms for the maximum satisfiability problem. *Nordic Journal of Computing*, 3 (1996), pp.388-404.

[3] T. Asano, K. Hori, T. Ono, and T. Hirata, A refinement of Yannakakis's algorithm for MAX SAT, *Information Processing Society of Japan, SIGAL-TR-54-11*, 1996, pp.81-88.

[4] T. Asano, K. Hori, T. Ono, and T. Hirata, Hybrid Approaches to MAX SAT, *Information Processing Society of Japan, SIGAL-TR-59-11*, 1997, pp.1-8.

[5] M. X. Goemans and D. P. Williamson, New 3/4-approximation algorithms for the maximum satisfiability problem. *SIAM J. Disc. Math.*, 7 (1994), pp.656-666.

[6] M. X. Goemans and D. P. Williamson, Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *Journal of the ACM*, 42 (1995), pp.1115-1145.

[7] D. S. Johnson, Approximation algorithms for combinatorial problems, *Journal of Comput. and Sys. Sci.*, 9 (1974), pp.256-278.

[8] M. G. C. Resende, L. S. Pitsoulis, and P. M. Pardalos, Approximate Solution of Weighted MAX SAT Problems using GRASP, http://www.research.att.com/~mgcr/doc/gmaxsat.ps.Z

[9] M. Yannakakis, On the approximation of maximum satisfiability. *Proc. 3rd SODA*, 1992, pp.1-9 (and also *J. Algorighms*, 17 (1994), pp.475-502).