確率アルゴリズムに基づいた支配集合問題の近似解法

梅田徳之*，富田悦次，三木哲也
電気通信大学 電気通信学部 情報通信工学科
(* 現在，エスエムジー（株）)

あらまし ボルツマンマシンにおけるアニーリング法の温度制御を廃し，さらに任意の局所解からの再探索を行う新しい確率アルゴリズム手法を提唱する．これを，支配集合問題の近似解抽出に適用し，いくつかのランダムグラフに対する比較実験により評価を行って，良好な結果を得た．

A Randomized Algorithm for the Dominating Set Problem

Noriyuki Umeda, Etsuji Tomita and Tetsuya Miki
Department of Information and Communication Engineering,
The University of Electro-Communications

Abstract We propose a new method of a randomized algorithm which is based upon Boltzmann machines. Then we apply it for the dominating set problem and show its effectiveness by computer simulations for random graphs.

1　はじめに

配置問題や符号理論などに関係した幾つかの問題が，無向グラフ中の支配集合抽出問題としてモデル化され，これまでに多くの研究がなされてきている（[1]，[2] 等）．しかし，支配集合抽出問題はＮＰ困難であり，大規模な問題に対して厳密最適解を求めることは困難とみなされる．このため，この様な問題に対してはボルツマンマシン上でアニーリングを適用した近似解法が考案されているが [3]，高精度の解を得るためには，複雑なパラメータ設定を要し，また，ゆるやかなアニーリングを行う必要があり，そのために長実行時間が必要となることが多い．

そこで，アニーリング法における温度制御を廃し，その代りに，局所解から抜け出し易くするための“緩和項”を導入した確率アルゴリズムが提案されている [4]．本稿では，それを基にして，その探索領域をより広く木状になるようにした手法を新たに提唱し，それを支配集合問題に適用してアルゴリズム RaTreeDS とし，その有効性を実験的に確認した．

2　支配集合問題

グラフ $G=(V,E)$ の節点部分集合 $V' \subseteq V$ が支配集合であるとは，任意の節点 $u \in V-V'$ に対して少なくとも１つの節点 $v' \in V'$ が隣接（支配）していることをいう．また，支配集合問題とは，与えられたグラフ G に対して要素数が最小の支配集合を求めることである．

3　ボルツマンマシンによる支配集合抽出

3.1　アニーリング法

支配集合問題をボルツマンマシン上でアニーリング法により解くために，グラフ $G=(V,E)$ の節点部分集合 $V' \subseteq V$ に隣接する節点集合を $\Gamma(V')$，$Cover = V' \cup \Gamma(V')$ とおいて，次のエネルギー関数の最小化問題として考える．

$$Energ(V') \stackrel{\text{def}}{=} \lambda(|V|-|Cover|)+|V'| \quad (\lambda \leq 1) \tag{1}$$

これは，V' が支配集合を構成するときに，$|V|-|Cover|=0$ となり，更に‘最小’支配集合を構成するときに最小値をとる．ここで，節点集合 V' を $|V|$ 次元の２値ベクトル

$$\boldsymbol{x}(V') \stackrel{\text{def}}{=} (x_1, x_2, \ldots, x_i, \ldots, x_{|V|}) \tag{2}$$

$$(x_i = 1 \ \text{ if } x_i \in V', x_i = 0 \ \text{ otherwise})$$

で表現したものを‘状態’という．また，

$$\mathcal{N}(\boldsymbol{x}, i) \stackrel{\text{def}}{=} (x_1, x_2, \ldots, 1-x_i, \ldots, x_{|V|}) \tag{3}$$

で定義される和集合 $\bigcup_{i=1}^{|V|}\mathcal{N}(\boldsymbol{x},i)$ を，$\boldsymbol{x}$ の‘近傍’と呼ぶ．解の抽出は2状態間の確率的遷移の繰り返し（状態遷移過程）により行うが，1回での状態遷移先は近傍に含まれる状態内に限定する。

状態遷移を受け入れる確率（受入確率）は，状態間のエネルギー差

$$\Delta Energ(i) = Energ(\mathcal{N}(\boldsymbol{x},i)) - Energ(V') \tag{4}$$

を用いて，シグモイド関数

$$sigm(-\frac{\Delta Energ(i)}{T}) \tag{5}$$
$$= (1 + exp(\frac{\Delta Energ(i)}{T}))^{-1}$$

の値とする．(T は温度パラメータ)．

なお，以下では $|V|$ 回の状態の更新（だだし，状態遷移が受入られない場合も含む）を 1 iteration とする．

アニーリング法では，温度パラメータを適切に制御して状態を収束させ，その最終時点の状態を解として採用する．この様なアニーリング法による支配集合抽出アルゴリズムを SADS と呼ぶ．

3.2 確率アルゴリズム RaDS

アニーリング法における適切な温度制御は難しい問題であり，長時間実行を要することともなる．

そこで，先ず文献 [4] の手法を直接的に本問題に適用し，温度を固定化して制御を不要にした，支配集合抽出のための確率アルゴリズム RaDS を提案する．

ここでは，文献 [4] と同様に，温度制御を廃した代りに，以下の式で表される“極の緩和”を用いることで，節点数が減少する変化に対して状態遷移確率を高くし，1つの極に滞在し続ける確率を低くする．極の緩和を実現するために2状態間のエネルギー差に緩和項を加えて受入れ確率を計算する．i 番目の節点についての緩和項は，次式 (6) で表すものとする．

$$Rel(\boldsymbol{x},i) = -\lambda \cdot x_i \tag{6}$$

従って，状態遷移の受入確率は

$$sigm(\frac{\Delta Energ(i) + Rel(\boldsymbol{x},i)}{T}). \tag{7}$$

4 確率アルゴリズム RaTreeDS

アニーリング法では収束した時点の解を目的の解とするのに対し，RaDS では任意の時点で解の候補をより良いものに更新していく．

ここで，途中解を保持するならば，局所解を有限個記憶しておき，何らかの条件で新たな局所解に推移することにより，特定の局所解に囚われず，幅広く解空間を探索出来ることが期待できる．

また，アルゴリズム RaDS で用いている極の緩和に，大域的情報を含めることにより，エネルギー差のみによらない，すなわち，局所的な変化のみに囚われない探索を実現する．この考え方で探索を進めたとき，局所解から新たな局所解を見付けていく過程が木の様に枝分かれするため，本アルゴリズムを確率アルゴリズム RaTreeDS と呼ぶ．

4.1 解の再探索

アルゴリズム RaDS では，固定した温度を用い，最終状態への収束を目標としない，いわば‘粗い’探索を続行する．

アルゴリズム RaTreeDS ではこの性質を有効に利用する．先ず，途中の局所解を保存していくことにより，特定の条件で別の局所解に推移することを可能にする．この実現のためには，温度制御が無いという点を有利に活かせる．すなわち，アニーリング法でこの推移を実現する場合には，推移後の再探索を始めるときの温度パラメータの制御が不可欠であり，難しい問題を残してしまう．しかし，アルゴリズム RaDS では温度パラメータが固定であるため，この探索の拡張が容易に行なえる．また，粗い探索を生かした，解の近傍に囚われない探索の拡張により，より幅広い解空間を探索可能にすることで，より多くの良い近似解を見つける可能性が生じる．

4.2 極の緩和と大域的情報

ボルツマンマシンでは，温度が一定の場合には極（局所解）から抜け出す確率より

も，極に落ち込む確率が高いので，極に滞在する確率も非常に高くなる．一方，アルゴリズムRaDSでは，解の候補を更新していくので，同じ極に長く滞在することは無駄になる．そこで，節点数が減少するような状態に遷移する確率を高くして，一つの極に滞在し続ける確率を低くする極の緩和を取り入れている．アルゴリズムRaTreeDSでは，極の緩和に更に大域的な情報を含めることにより，拡張を図った．

式(7)において$\Delta Energ(i)$は前回の探索と現在の探索における解状態のエネルギー差を示し，緩和項は差をとらない．ここに着目し，差分をとらない緩和項には，局所的な値にならざるを得ない$\Delta Energ(i)$とは異なって，全体的な解状態を表す大域的な情報を含めることが可能である．

$$Rel(\boldsymbol{x}, i) = -\lambda \cdot x_i \cdot \frac{|Cover|}{|V|} \qquad (8)$$

従って，アルゴリズムRaTreeDSにおいては，上記の大域的な情報を含む緩和項(8)を用いることにより，グラフ上で支配されている節点数が大きいほど極の緩和を大きくし，グラフ上で支配されている節点数が少ないほど極の緩和の影響を小さくした．以上を踏まえて確率アルゴリズムRaTreeDSの特徴をまとめると，次のようになる．

- 特定の局所解に囚われず，任意の局所解から新たな探索が出き，幅広い解空間を探索が可能．
- 極の緩和に大域的な情報を含むことにより，局所的な変化のみに囚われない探索が可能．

4.3 アルゴリズムの実働化

以上のような概念によるRaTreeDSの実働化において，途中の解の更新は次のように行う．

即ち，先ず途中解の保存対象は，探索過程において，それまでで最良の解サイズを持つ近似解全てとする．また，再探索は既存の解が検出されたときに行うこととする．ここで，再探索する解の決定方法はランダムとした．

アルゴリズムRaTreeDSでは，最初に，節点集合V'および解候補の支配集合の集合DS_{min}を初期化する．状態の更新は節点の登録順に行い，これを$iter_{max} \cdot |V|$回繰り返す．DS_{min}の更新は同精度より良い解が抽出された場合に適用し，アルゴリズム終了時のDS_{min}を解集合とする．なお，DS_{min}中の支配集合（同一サイズ）のサイズを$||DS_{min}||$で表す．

```
program RaTree(G = (V, E), T, iter_max)
begin
  x(V') := x(φ)
  DS_min := {V}
  while iter < iter_max · |V| do
    for i := 1 to |V| do
      if sigm(-(ΔEnerg(i)+Rel(x,i))/T)
        > random[0,1)
      then
        x(V') := N(x, i)(近傍解)
      fi
      j := i の隣接節点 (ランダムに選定)
      if sigm(-(ΔEnerg(j)+Rel(x,j))/T)
        > random[0,1)
      then
        x(V') := N(x, j)
      fi
      RaTreeJump1
    od
    iter := iter + 1
  od
end

procedure RaTreeJump1
begin
  if |Cover| = |V| then
    if ||DS_min|| > |V'|
    then
      DS_min := {V'}
      (新たなサイズの解を保存)
    else
      if ||DS_min|| = |V'|
      then
        if 現在の解が既存の解である
        then
          既存の解中の任意の解を
          次探索の出発解とする
```

```
        else
          DS_min := DS_min ∪ {V'}
          (解を追加保存する)
        fi fi fi
    fi
end
```

図1：アルゴリズム RaTreeDS

5 計算機実験

計算機シミュレーションにより，前記アルゴリズム RaTreeDS の評価を行った．アルゴリズム SADS，RaDS，RaTreeDS に対するパラメータ設定は，表1のとおりとした（α は温度逓減倍率，n はイテレーション数）．使用計算機は CPU Pentium II 266MHz の AT 互換機，使用言語は Java 2 である．

表2の解精度は，各5個のグラフそれぞれに5回試行した，計25回試行の平均値，表3の解精度は，各グラフに対し5回試行の平均値である．この対象における解の精度は，1つ上げるだけでも非常に困難である．このことを考慮すると，RaTreeDS は他のアルゴリズムに比べ，より有意な精度の解を抽出しているといえる．特に支配集合問題の性質上，枝存在確率が低いほど時間を要して問題が難しいので，この点で改善が達成されている点は有効である．

表3は300節点における支配数と抽出した解個数の詳細を示したものである．ここで特に，支配数8においても RaTreeDS ではどのグラフに対しても解が抽出されており，RaTreeDS の解（支配数）の精度は，SADS，RaDS に比べて，ほぼ同程度以上となっている．また，支配数9に対しては，RaTreeDS が全面的に多くの解個数を得ている．以上より，RaTreeDS は解精度，解個数の両面において SADS, RaDS よりも優れているといえる．

6 むすび

今後の課題として，遺伝アルゴリズム (GA) 等，他のアルゴリズムとの比較，及び融合手法の開発等が挙げられる．ランダムグラフ以外の対象についての実験も重要である．

表1: 実験パラメータ

Algorithm	SADS	RaDS	RaTreeDS
T:temperature	5.9-0.1	0.15	0.15
λ	0.4	0.5	0.5
α	0.96	-	-
iteration	$n/10$	$10n$	$10n$
total iteration	$10n$	$10n$	$10n$

表2: 各アルゴリズムの解精度

節点数	枝存在確率 (%)	SADS 支配数	RaDS 支配数	RaTreeDS 支配数
100	25	6.36	6.32	**6.28**
	50	3.80	3.80	3.80
	75	2.00	2.00	2.00
200	25	7.84	7.88	**7.84**
	50	4.00	4.00	4.00
	75	3.00	3.00	3.00
300	25	8.96	8.68	**8.60**
	50	4.88	4.92	**4.60**
	75	3.00	3.00	3.00

表3: サイズ毎の解個数
（節点数 300, 枝存在確率 25 %）

支配数	SADS 解個数	RaDS 解個数	RaTreeDS 解個数
9	160.4	719.6	1492.2
	130.8	520.4	961.4
	104.2	320.0	558.6
	115.6	586.8	700.0
	81.6	328.8	652.0
8	0.0	2.6	2.6
	0.2	0.4	0.8
	0.0	0.4	0.2
	0.2	0.4	0.2
	0.0	0.0	0.6

参考文献

[1] T.W. Haynes, "Domination in graphs: A brief overview", J. Comb. Math. Comb. Comput. vol.24,pp.225-237 (1997).

[2] 梅田徳之，富田悦次，三木哲也，"Dominating Set 問題に対する分枝限定アルゴリズムとその実験的評価"，人工知能全大，pp.253-255 (1998).

[3] E. Aarts and J. Korst, "Simulated Anneling and Boltzmann Machines", Wiley (1989).

[4] 山田義朗，富田悦次，高橋治久，"近似最大クリークを抽出する確率アルゴリズムとその実験的評価"，信学論 (D-I), vol.J76-D-I, pp.46-53 (1993).