I482F 実践的アルゴリズム特論
(14) 乱択アルゴリズム：乱択による近似

岡本 吉央

okamotoy@jaist.ac.jp

北陸先端科学技術大学院大学

2011 年 7 月 2 日

"最終更新：2011/07/02 12:01"

乱択アルゴリズム (randomized algorithm) とは？

乱数を使用するアルゴリズム

乱数を使用することのメリット

- アルゴリズム設計と実装が単純になる場合がある
- アルゴリズムの高速化が可能になる場合がある
- アルゴリズムの近似性能が上がる場合がある

乱数を使用することのデメリットとその解消

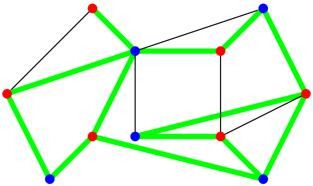
- アルゴリズムに望む性質が常に保証されるわけではなくなる
  - 例：停止しなくなる場合があるかもしれない
  - 解消法：「高い確率」で性質が保証されることを目指す
- アルゴリズムが乱数を作る必要が出てくる
  - 参考：擬似乱数生成 (pseudo-random number generation)
  - 解消法 (？)：その部分はブラックボックスとして扱う

1. 乱択による近似：最大カット問題
2. 乱択による近似：最大独立集合問題
3. 今日のまとめ

最大カット問題

定義：最大カット問題

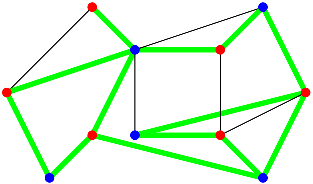
- 入力：無向グラフ $G = (V, E)$
- 出力：G の頂点への赤と青の色割当 (彩色である必要はない)
- 目的：2 端点が 2 色で塗られている辺 (2 色辺) の数の**最大化**



2 色辺の数 $= 13$

単純な乱択アルゴリズム 1

1. 各頂点に対して**独立に**赤か青の色を割り当てる
2. 得られた色割当を出力して，停止



2 色辺の数 $= 13$

疑問

2 色辺の数はどれくらい大きいのか？

2 色辺の数の期待値 (1)

- グラフの辺集合 $E = \{e_1, \ldots, e_m\}$ とする
- $X = 2$ 色辺の数 (確率変数)
- 辺 e_i に対して，確率変数 X_i を次で定義

$$X_i = \begin{cases} 1 & (e_i \text{ は } 2 \text{ 色辺である}) \\ 0 & (e_i \text{ は } 2 \text{ 色辺でない}) \end{cases}$$

- このとき，

$$X = X_1 + X_2 + \cdots + X_m$$

2 色辺の数の期待値 (2)

- $X_i = 1 \Leftrightarrow e_i$ の端点の一方が赤，他方が青
- $\Pr(X_i = 1) = \frac{1}{2}$
- $E[X_i] = 0 \cdot \Pr(X_i = 0) + 1 \cdot \Pr(X_i = 1) = \frac{1}{2}$
- したがって，

$$
\begin{aligned}
E[X] &= E[X_1 + X_2 + \cdots + X_m] \\
&= E[X_1] + E[X_2] + \cdots + E[X_m] \\
&= \frac{1}{2} + \frac{1}{2} + \cdots + \frac{1}{2} = \frac{m}{2}
\end{aligned}
$$

期待値が分かると何がよいか？ (1)

- 次の命題が成り立つ

命題 1

$$\Pr\left(2 \text{ 色辺の数} < \frac{m}{2}\right) \leq \frac{m}{m+2} = 1 - \frac{2}{m+2}$$

- 次のようなアルゴリズムを考える

繰返しアルゴリズム 2

1. 単純な乱択アルゴリズムを T 回繰返し実行
2. T 回の中で最も 2 色辺の数が大きかった色割当を出力して停止

- このアルゴリズムの出力する 2 色辺の数が $\frac{m}{2}$ 未満である確率は？

$$\Pr\left(T \text{ 回とも } 2 \text{ 色辺の数} < \frac{m}{2}\right)$$

$$= \Pr\left(1 \text{ 回の実行で } 2 \text{ 色辺の数} < \frac{m}{2}\right)^T \leq \left(1 - \frac{2}{m+2}\right)^T$$

期待値が分かると何がよいか？ (2)

役立つ不等式

任意の実数 x に対して，$1 + x \leq e^x$

- したがって，

$$\Pr\left(T \text{ 回とも 2 色辺の数 } < \frac{m}{2}\right) \leq e^{-\frac{2}{m+2}T}$$

- $T \geq \frac{m+2}{2}m$ とすると，

$$\Pr\left(T \text{ 回とも 2 色辺の数 } < \frac{m}{2}\right) \leq \frac{1}{e^m}$$

- ゆえに，

$$\Pr\left(\text{アルゴリズム 2 の出力する 2 色辺の数 } \geq \frac{m}{2}\right) \geq 1 - \frac{1}{e^m}$$

- すなわち，
アルゴリズム 2 は高確率で 2 色辺の数が $\frac{m}{2}$ 以上の色割当を出力

命題 1 の証明 (1)

2 色辺の数を X とすると，

$$
\begin{aligned}
\frac{m}{2} = \mathrm{E}[X] &= \sum_{i=0}^{m} i \cdot \Pr(X = i) \\
&= \sum_{i=0}^{m/2-1} \underbrace{i}_{\leq m/2-1} \cdot \Pr(X = i) + \sum_{i=m/2}^{m} \underbrace{i}_{\leq m} \cdot \Pr(X = i) \\
&\leq \left(\frac{m}{2} - 1\right) \sum_{i=0}^{m/2-1} \Pr(X = i) + m \sum_{i=m/2}^{m} \Pr(X = i) \\
&= \left(\frac{m}{2} - 1\right) \Pr\left(X < \frac{m}{2}\right) + m \Pr\left(X \geq \frac{m}{2}\right) \\
&= \left(\frac{m}{2} - 1\right) \Pr\left(X < \frac{m}{2}\right) + m \left(1 - \Pr\left(X < \frac{m}{2}\right)\right) \\
&= -\left(\frac{m}{2} + 1\right) \Pr\left(X < \frac{m}{2}\right) + m
\end{aligned}
$$

命題 1 の証明 (2)

したがって

$$\begin{aligned} \left(\frac{m}{2}+1\right)\Pr\left(X<\frac{m}{2}\right) &\leq \frac{m}{2} \\ \therefore \quad \Pr\left(X<\frac{m}{2}\right) &\leq \frac{m}{m+2} \end{aligned}$$

$\square$

乱択による近似：最大カット問題 (まとめ)

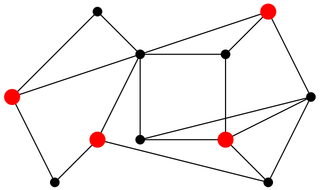
- アルゴリズム 1：一様分布に従って色を割り当てる
  - 2 色辺の数の期待値が計算可能
  - 2 色辺の数が期待値未満になる確率は 1 未満
- アルゴリズム 2：アルゴリズム 1 を T 回繰り返す
  - T 回とも 2 色辺の数が期待値未満になる確率は T に関して指数関数的減少
  - 高確率で，T 回中一度は 2 色辺の数が期待値以上になる

1. 乱択による近似：最大カット問題
2. 乱択による近似：最大独立集合問題
3. 今日のまとめ

独立集合

定義：独立集合

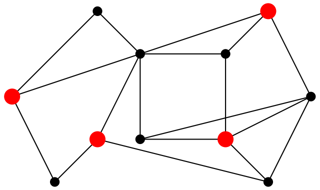
無向グラフ $G = (V, E)$ **の独立集合** (independent set) **とは**
G **の頂点部分集合** S **で，その中のどの** 2 **頂点も隣接していないもの**



最大独立集合問題

定義：最大独立集合問題

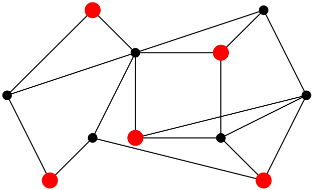
- 入力：無向グラフ $G = (V, E)$
- 出力：G の独立集合 S
- 目的：S の要素数の最大化



最大独立集合問題

定義：最大独立集合問題

- **入力：無向グラフ** $G = (V, E)$
- **出力：**G **の独立集合** S
- **目的：**S **の要素数の最大化**



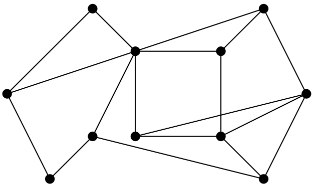
要素数 $= 5$

最大独立集合問題に対する乱択アルゴリズム

入力：無向グラフ $G = (V, E)$

1. $S \leftarrow \emptyset$ (S **が最終的に出力する独立集合となる**)
2. G **の頂点を一様分布に従って一列に並べ，** $v_1, v_2, \ldots, v_n$ **とする**
3. **各** $i = 1, 2, \ldots, n$ **に対して，次を行う**
   1. v_i **の隣接頂点がどれも** S **の要素でないとき，** S **に** v_i **を追加する**
   2. **それが成り立たないとき，何もしない**
4. S **を出力して，停止**

一様分布に従って一列に並べる	$\equiv$	$n!$ **の並べ方をそれぞれ確率** $\frac{1}{n!}$ **で選ぶ**

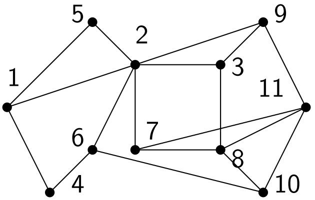


最大独立集合問題に対する乱択アルゴリズム

入力：無向グラフ $G = (V, E)$

1. $S \leftarrow \emptyset$ (S **が最終的に出力する独立集合となる**)
2. G **の頂点を一様分布に従って一列に並べ，**$v_1, v_2, \dots, v_n$ **とする**
3. **各** $i = 1, 2, \dots, n$ **に対して，次を行う**
   1. v_i **の隣接頂点がどれも** S **の要素でないとき，**S **に** v_i **を追加する**
   2. **それが成り立たないとき，何もしない**
4. S **を出力して，停止**

一様分布に従って一列に並べる	$\equiv$	$n!$ **の並べ方をそれぞれ確率** $\frac{1}{n!}$ **で選ぶ**

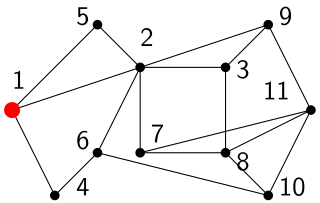


最大独立集合問題に対する乱択アルゴリズム

入力：無向グラフ $G = (V, E)$

1. $S \leftarrow \emptyset$ (S **が最終的に出力する独立集合となる**)
2. G **の頂点を一様分布に従って一列に並べ，** $v_1, v_2, \ldots, v_n$ **とする**
3. **各** $i = 1, 2, \ldots, n$ **に対して，次を行う**
    1. v_i **の隣接頂点がどれも** S **の要素でないとき，** S **に** v_i **を追加する**
    2. **それが成り立たないとき，何もしない**
4. S **を出力して，停止**

一様分布に従って一列に並べる	$\equiv$	$n!$ の並べ方をそれぞれ確率 $\frac{1}{n!}$ で選ぶ

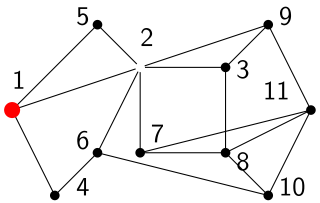


最大独立集合問題に対する乱択アルゴリズム

入力：無向グラフ $G = (V, E)$

1. $S \leftarrow \emptyset$ (S が最終的に出力する独立集合となる)
2. G の頂点を一様分布に従って一列に並べ，$v_1, v_2, \ldots, v_n$ とする
3. 各 $i = 1, 2, \ldots, n$ に対して，次を行う
   1. v_i の隣接頂点がどれも S の要素でないとき，S に v_i を追加する
   2. それが成り立たないとき，何もしない
4. S を出力して，停止

一様分布に従って一列に並べる	$\equiv$	$n!$ の並べ方をそれぞれ確率 $\frac{1}{n!}$ で選ぶ

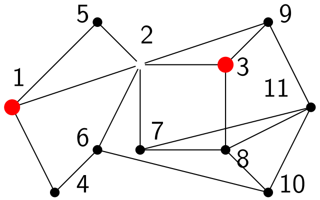


最大独立集合問題に対する乱択アルゴリズム

入力：無向グラフ $G = (V, E)$

1. $S \leftarrow \emptyset$ (S が最終的に出力する独立集合となる)
2. G の頂点を一様分布に従って一列に並べ，$v_1, v_2, \ldots, v_n$ とする
3. 各 $i = 1, 2, \ldots, n$ に対して，次を行う
   1. v_i の隣接頂点がどれも S の要素でないとき，S に v_i を追加する
   2. それが成り立たないとき，何もしない
4. S を出力して，停止

一様分布に従って一列に並べる ≡ $n!$ の並べ方をそれぞれ確率 $\frac{1}{n!}$ で選ぶ

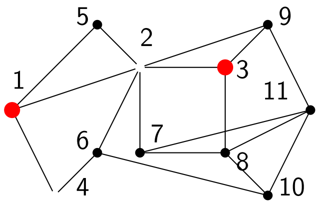


最大独立集合問題に対する乱択アルゴリズム

入力：無向グラフ $G = (V, E)$

1. $S \leftarrow \emptyset$ (S **が最終的に出力する独立集合となる**)
2. G **の頂点を一様分布に従って一列に並べ，** $v_1, v_2, \ldots, v_n$ **とする**
3. **各** $i = 1, 2, \ldots, n$ **に対して，次を行う**
   1. v_i **の隣接頂点がどれも** S **の要素でないとき，** S **に** v_i **を追加する**
   2. **それが成り立たないとき，何もしない**
4. S **を出力して，停止**

一様分布に従って一列に並べる	$\equiv$	$n!$ の並べ方をそれぞれ確率 $\frac{1}{n!}$ で選ぶ

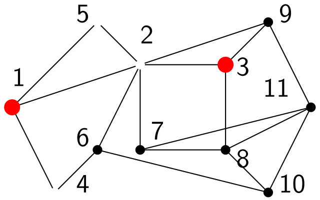


最大独立集合問題に対する乱択アルゴリズム

入力：無向グラフ $G = (V, E)$

1. $S \leftarrow \emptyset$ (S **が最終的に出力する独立集合となる**)
2. G **の頂点を一様分布に従って一列に並べ，** $v_1, v_2, \ldots, v_n$ **とする**
3. **各** $i = 1, 2, \ldots, n$ **に対して，次を行う**
    1. v_i **の隣接頂点がどれも** S **の要素でないとき，** S **に** v_i **を追加する**
    2. **それが成り立たないとき，何もしない**
4. S **を出力して，停止**

一様分布に従って一列に並べる	$\equiv$	$n!$ **の並べ方をそれぞれ確率** $\frac{1}{n!}$ **で選ぶ**

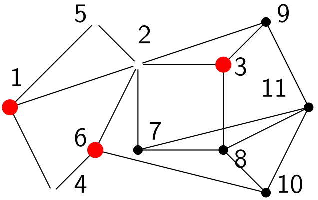


最大独立集合問題に対する乱択アルゴリズム

入力：無向グラフ $G = (V, E)$

1. $S \leftarrow \emptyset$ (S が最終的に出力する独立集合となる)
2. G の頂点を**一様分布に従って**一列に並べ，$v_1, v_2, \ldots, v_n$ とする
3. 各 $i = 1, 2, \ldots, n$ に対して，次を行う
    1. v_i の隣接頂点がどれも S の要素でないとき，S に v_i を追加する
    2. それが成り立たないとき，何もしない
4. S を出力して，停止

一様分布に従って一列に並べる ≡ $n!$ の並べ方をそれぞれ確率 $\frac{1}{n!}$ で選ぶ

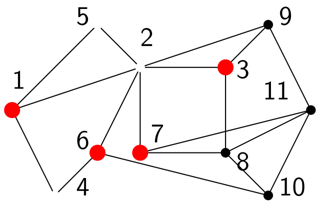


最大独立集合問題に対する乱択アルゴリズム

入力：無向グラフ $G = (V, E)$

1. $S \leftarrow \emptyset$ (S **が最終的に出力する独立集合となる**)
2. G **の頂点を一様分布に従って一列に並べ，** $v_1, v_2, \ldots, v_n$ **とする**
3. **各** $i = 1, 2, \ldots, n$ **に対して，次を行う**
   1. v_i **の隣接頂点がどれも** S **の要素でないとき，** S **に** v_i **を追加する**
   2. **それが成り立たないとき，何もしない**
4. S **を出力して，停止**

一様分布に従って一列に並べる	$\equiv$	$n!$ の並べ方をそれぞれ確率 $\frac{1}{n!}$ で選ぶ

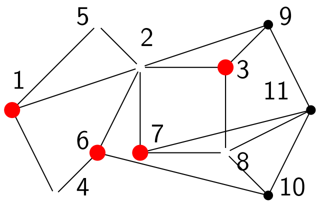


最大独立集合問題に対する乱択アルゴリズム

入力：無向グラフ $G=(V,E)$

1. $S \leftarrow \emptyset$ (S が最終的に出力する独立集合となる)
2. G の頂点を一様分布に従って一列に並べ，$v_1, v_2, \ldots, v_n$ とする
3. 各 $i=1,2,\ldots,n$ に対して，次を行う
   1. v_i の隣接頂点がどれも S の要素でないとき，S に v_i を追加する
   2. それが成り立たないとき，何もしない
4. S を出力して，停止

一様分布に従って一列に並べる	$\equiv$	$n!$ の並べ方をそれぞれ確率 $\frac{1}{n!}$ で選ぶ

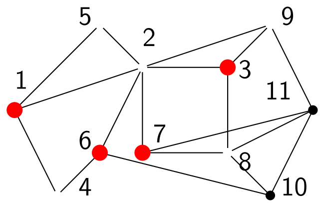


最大独立集合問題に対する乱択アルゴリズム

入力：無向グラフ $G = (V, E)$

1. $S \leftarrow \emptyset$ (S **が最終的に出力する独立集合となる**)
2. G **の頂点を一様分布に従って一列に並べ，** $v_1, v_2, \ldots, v_n$ **とする**
3. **各** $i = 1, 2, \ldots, n$ **に対して，次を行う**
    1. v_i **の隣接頂点がどれも** S **の要素でないとき，** S **に** v_i **を追加する**
    2. **それが成り立たないとき，何もしない**
4. S **を出力して，停止**

一様分布に従って一列に並べる	$\equiv$	$n!$ **の並べ方をそれぞれ確率** $\frac{1}{n!}$ **で選ぶ**

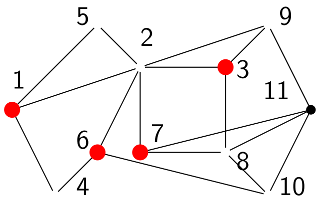


最大独立集合問題に対する乱択アルゴリズム

入力：無向グラフ $G = (V, E)$

1. $S \leftarrow \emptyset$ (S **が最終的に出力する独立集合となる**)
2. G **の頂点を****一様分布に従って****一列に並べ，** $v_1, v_2, \ldots, v_n$ **とする**
3. **各** $i = 1, 2, \ldots, n$ **に対して，次を行う**
   1. v_i **の隣接頂点がどれも** S **の要素でないとき，** S **に** v_i **を追加する**
   2. **それが成り立たないとき，何もしない**
4. S **を出力して，停止**

$$\boxed{\text{一様分布に従って一列に並べる}} \equiv \boxed{n!\ \text{の並べ方をそれぞれ確率}\ \tfrac{1}{n!}\ \text{で選ぶ}}$$

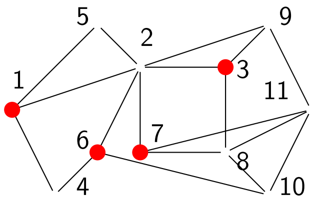


最大独立集合問題に対する乱択アルゴリズム

入力：無向グラフ $G = (V, E)$

1. $S \leftarrow \emptyset$ (S が最終的に出力する独立集合となる)
2. G の頂点を一様分布に従って一列に並べ，$v_1, v_2, \ldots, v_n$ とする
3. 各 $i = 1, 2, \ldots, n$ に対して，次を行う
    1. v_i の隣接頂点がどれも S の要素でないとき，S に v_i を追加する
    2. それが成り立たないとき，何もしない
4. S を出力して，停止

一様分布に従って一列に並べる	$\equiv$	$n!$ の並べ方をそれぞれ確率 $\frac{1}{n!}$ で選ぶ



出力される独立集合の要素数の期待値 (1)

- **アルゴリズムで並べた頂点の順序を** $v_1, v_2, \ldots, v_n$ **とする**
- $X =$ **出力される独立集合の要素数 (確率変数)**
- **頂点** v_i **に対して，確率変数** X_i **を次で定義**

$$X_i = \begin{cases} 1 & (v_i \text{ は出力の要素である}) \\ 0 & (v_i \text{ は出力の要素でない}) \end{cases}$$

- **このとき，**

$$X = X_1 + X_2 + \cdots + X_n$$

出力される独立集合の要素数の期待値 (2)

- $X_i = 1 \Leftarrow v_i$ **のすべての隣接点** v_j **に対して，** $i < j$
- $\Pr(v_i$ **のすべての隣接点** v_j **に対して，** $i < j) = \dfrac{1}{d_i + 1}$
  (**ただし，** d_i **は** v_i **の次数**)
- $\therefore \Pr(X_i = 1) \geq \dfrac{1}{d_i + 1}$
- $\therefore \mathrm{E}[X_i] = 0 \cdot \Pr(X_i = 0) + 1 \cdot \Pr(X_i = 1) \geq \dfrac{1}{d_i + 1}$
- **したがって，**

$$\begin{aligned} \mathrm{E}[X] &= \mathrm{E}[X_1 + X_2 + \cdots + X_n] \\ &= \mathrm{E}[X_1] + \mathrm{E}[X_2] + \cdots + \mathrm{E}[X_n] \\ &\geq \frac{1}{d_1 + 1} + \frac{1}{d_2 + 1} + \cdots + \frac{1}{d_n + 1} \end{aligned}$$

- **たとえば，** $d_1 = d_2 = \cdots = d_n = 9$ **だったら，**

$$\mathrm{E}[X] \geq \frac{n}{10}$$

期待値が分かると何がよいか？

たとえば，$d_1 = d_2 = \cdots = d_n = 9$ だったら

- $\Pr(X < n/10) < 1$ を証明できる
- よって，繰返しアルゴリズムを考えると，
  高確率で期待値以上の要素数を持つ独立集合が出力される

詳細は演習問題 (流れは最大カット問題と同様)

1. 乱択による近似：最大カット問題
2. 乱択による近似：最大独立集合問題
3. 今日のまとめ

今日のまとめ

乱択アルゴリズム

乱数を使用するアルゴリズム

- アルゴリズム設計と実装が単純になる場合がある
- アルゴリズムの近似性能が上がる場合がある

乱択による近似

- 最大カット問題
- 最大独立集合問題

設計・解析の方法論

- 01 確率変数 (標示変数，indicator variable) の利用
- 期待値の計算
- 反復による成功確率の増幅

参考文献

- **乱択アルゴリズムについて**
  - **玉木久夫．乱択アルゴリズム．共立出版，**2008**．**
  - **結城浩．数学ガール/乱択アルゴリズム．ソフトバンククリエイティブ，**2011**．**
  - R. Motwani, P. Raghavan. Randomized Algorithms. Cambridge University Press, 1998.
  - M. Mitzenmacher, E. Upfal. Probability and Computing. Cambridge University Press, 2005. (**日本語訳：小柴健史，河内亮周訳．確率と計算．共立出版，**2009**．**)