6F-04

オンライン独立頂点集合問題に対する一般化されたオンラインアルゴリズムの性能評価

武富史郎　宮崎修一　岩間一雄　Magnús M. Halldórsson*
京都大学　情報学研究科

1 はじめに

オンライン独立頂点集合問題とは，各時点で1つの頂点 v_i と、v_i に接続する枝が与えられ，その頂点を独立頂点として選ぶかどうかをその時点で決定し，最終的に大きなサイズの独立頂点集合を得ることを目的とする問題である．従来の設定のように解を1つしか持つことを許さないオンラインアルゴリズムを用いた場合には，グラフの総頂点数を n として，競合比の上下限は $n-1$ となる [1]．Halldòrsson は，解を同時に複数個保持することを許すオンラインアルゴリズムを導入し，競合比の上下限が $\frac{n}{4}$ になることを示した [1].

上記のアルゴリズムは，解を複数個保持できるが，頂点が複数の解にまたがって使われることを禁止している．本研究ではその制約も取り払い，頂点の重複も許して同時に多項式個の解を保持することを許した場合を考察した．研究の結果，a を整数として，n 個頂点を受けとった段階で $2^{a-1}n$ 個の解を持つことを許したアルゴリズムの競合比の上限が $\frac{n-1}{(1-o(1))a}$，k を整数として，n 個頂点を受けとった段階で n^k 個の解を持つことを許したアルゴリズムの競合比の下限が $\Omega((\frac{n}{k})^{\frac{1}{2}-\varepsilon})$（$\varepsilon$ は任意の正の定数）となることを示した.

2 定義

本節では，オンライン独立頂点集合問題および $r(n)$ オンラインアルゴリズムの定義を与える.

2.1 オンライン独立頂点問題

本問題は頂点が次々に与えられ，次の頂点を受けとる前に現在の頂点を独立頂点集合（IS）として選ぶか否かを決定する問題である．この問題における i 番目の要求，行動，目標はそれぞれ次の通りである.

要求: 頂点 v_i 及び v_i と $v_j (j < i)$ との枝の有無.

行動: v_i を独立頂点集合に入れるか否か.

目標: 最終的にできるグラフ上でできるだけ頂点の多い IS を構成する.

2.2 $r(n)$ オンラインアルゴリズム

オンライン独立頂点集合問題に対する $r(n)$ オンラインアルゴリズムとは，i ステップ目において $r(i)$ 種類の解を同時に保持することができるものを言う．ただし，i ステップ目に持っている解は以下の (i) または (ii) のどちらかを満たさなければならない.

(i) $i-1$ ステップ目で持っていた解.

(ii) $i-1$ ステップ目で持っていた解に頂点 v_i を加えたもの.

3 $2^{a-1}n$ オンラインアルゴリズムの上限

定理 1. a を任意の正の整数とする．オンライン独立頂点問題に対する $2^{a-1}n$ オンラインアルゴリズムで競合比が高々 $\frac{n-1}{(1-o(1))a}$ であるものが存在する.

証明. まず上記の競合比を実現する $2^{a-1}n$ オンラインアルゴリズムの動作を説明する．このアルゴリズムは i 個目の頂点が到着した時点で新たに 2^{a-1} 個の解を持つことが許される．このアルゴリズムは $i-a+1$ 個目の頂点から i 個目の頂点までの a 個で形成されるすべての IS を，この 2^{a-1} 個の解で保持していくものである．アルゴリズムがこの様な解の持ち方をできるということは帰納法により証明できる.

任意の入力を考え，その最適解を L，そのコストを l とする．オンラインアルゴリズムは連続した a 個の範囲内にある IS は全て解として持つことができるので，これら l 個の頂点が等間隔に散らばっている場合がアルゴリズムにとって最も都合の悪い入力である.

On the analysis of generalized online algorithm for online independent set problem
Shiro Taketomi, Shuichi Miyazaki, Kazuo Iwama, Magnús M. Halldórsson
School of Informatics, Kyoto University
*現アイスランド大学

グラフの総頂点数を n とし, L に含まれる頂点同士の距離を x とする. (すなわち $1+ix(i=0,1,\cdots)$ 番目の頂点が L に含まれる.) 総頂点数が n 個, L のサイズが l という関係から $(l-1)x+1 \leq n$ が得られる. 従って $l \leq \frac{n-1}{x}+1$ となる. 次に, 間隔 x で L の頂点を並べたとき, a 個の連続した頂点のうち L の頂点が入る個数を s とすると $a \leq sx$ が成り立つ. アルゴリズムはコスト $s(\geq \frac{x}{a})$ 以上の解を出力することができるのでこのアルゴリズムの競合比は高々 $\frac{l}{s} \leq \frac{n-1}{(1-\frac{1}{l})a}$ となる. l が十分に大きいとき $\frac{l}{s} \leq \frac{n-1}{(1-o(1))a}$ とかける. □

4 n^k オンラインアルゴリズムの下限

定理 2. k を定数とするとき n^k オンラインアルゴリズムの競合比は少なくとも $CR = \Omega((\frac{N}{k})^{\frac{1}{2}-\varepsilon})$ である. ただし ε は任意の正の定数である.

証明. 上記の競合比を与えるアドバーサリーの戦略について述べる. まず与えられた頂点数 n から t,δ,l を次の 3 式

- $log2k+(1+\delta)logl < l^\delta$
- $\frac{1}{t}2^{1-t}log\frac{2}{1-t} < l^\delta$
- $n = 2kl^{1+\delta}(1^t+2^t+\cdots((2l+1)(t+1))^t)$

を満たすように決定し, $s_i = i^t kl^\delta$ とする.

アドバーサリーは頂点を 1 つずつアルゴリズムに与えるが, 各 i に対し, $2l(s_1+\cdots+s_{i-1})+1$ 個目から $2l(s_1+\cdots+s_i)$ 個目を与えるまでの間をラウンド i と呼ぶことにする. n 個の頂点を与えるのに必要なラウンド数を N とすると $N=(2l+1)(t+1)$ となる. 各頂点の接続状況は以下の通りである.

まず, 各 i に対してラウンド i に与えられる $2ls_i$ 個の頂点の間には枝は全くない. すなわち, これら $2ls_i$ 個は独立頂点集合となっている. また, 異なるラウンドで与えられた 2 つの頂点の枝の接続は以下のようにする. アドバーサリーは各ラウンドの終了時にそのラウンドで与えた頂点に good または bad のラベルを与える. ラウンド i で与える頂点は, $i-1$ ラウンド以前の全ての bad 頂点に枝を持ち, good 頂点には枝を持たないものとする.

次に, 各ラウンド終了時における good 頂点と bad 頂点との決め方について説明する. ラウンド i の終了時にアルゴリズムが保持している解の個数は高々 $(2l(s_1+\cdots+s_i))^k$ 個である. これらの解のうち, ラウンド i の $2ls_i$ 個の頂点のうち s_i 個以上を選んでいるものの集合を A とする. この時以下の補題が成り立つ.

補題 1. ラウンド i の $2ls_i$ 個のうち, 以下の条件を満たすように ls_i 個を bad にする決め方が存在する.
条件:$\forall a \in A$ に対し, a に含まれているラウンド i の頂点 (s_i 個以上) のうち少なくとも 1 つは bad である.

証明. 略

$2ls_i$ 個の頂点のうち, 上記の補題を満たすように ls_i 個を good, 残りの ls_i 個を bad に決定する. このとき, 各ラウンドの good 頂点 ls_i 個と, 最終ラウンドの $2ls_N$ 個は IS をなすので最適解のコストは $ls_1+ls_2+\cdots+ls_{N-1}+2ls_N$ 以上となる. また, オンラインアルゴリズムは bad 頂点を一度選んでしまったら, その後の頂点を選ぶことはできない. したがって, 各段階で good 頂点のみを選んでいったものが大きな解を得ることができるわけだが, 各ラウンド終了後のアドバーサリーの戦略により, 各ラウンドでは高々 s_i 個の good 頂点しか選ぶことができない. (最終ラウンドは $2ls_N$ 個すべてを選ぶことができる.) 従ってオンラインアルゴリズムの得る解のコストは高々 $s_1+s_2+\cdots+s_{N-1}+2ls_N$ である. ここで上記 s_i,l,t,δ の決め方より $s_1+s_2+\cdots+s_{N-1} \geq 2ls_N$ が成り立つので, 競合比は $\frac{ls_1+ls_2+\cdots+ls_{N-1}+2ls_N}{s_1+s_2+\cdots+s_{N-1}+2ls_N} \geq \frac{l+1}{2}$ となる. ここで $N=(2l+1)(t+1)$, $n=2l(s_1+\cdots+s_N)$ であるので, $n < 2^{2+t}(1+t)^t k(l+\frac{2+t}{2+2t})^{2+t+\delta}$ となる.

これより l を求めると $l > (\frac{N}{Ck})^{\frac{1}{2+t+\delta}} - \frac{2+t}{2+2t}$ (ただし $C=2^{2+t}(1+t)^t$) となり競合比は $\Omega((\frac{N}{k})^{\frac{1}{2+t+\delta}})$ となる. t,δ を小さくすることにより競合比は $\Omega((\frac{N}{k})^{\frac{1}{2}-\varepsilon})$ と書ける. □

5 おわりに

本稿では, $2^{a-1}n$ オンラインアルゴリズムの上限と n^k オンラインアルゴリズムの下限を示した. 多項式個の解を持つ場合, $\frac{n}{logn}$ の上下限が予想されるのでそれを証明することが今後の課題である.

参考文献

[1] Magnús M. Halldórsson, "On-Line Independent Set Problem," Unpublished, Sep 1996.