コンビニ探索問題の競合比解析

伊奈 林太郎

京都大学 大学院情報学研究科
ina@kuis.kyoto-u.ac.jp

概 要 未来に起きる現象にうまく対処する方法を考えるには, 入力全体へ最初からアクセス可能なわけではない問題に対するアルゴリズム (オンラインアルゴリズム) を考える必要がある. オンラインアルゴリズムの性能を評価する方法の一つに, 入力全体へ最初からアクセス可能だった場合の理想のアルゴリズム性能と比較する競合比解析がある. 本稿では, オンラインアルゴリズムの性能評価の例として, コンビニを探す問題を解くアルゴリズムの競合比を求める.

1 はじめに

問題を解決するための多くのアルゴリズムは, 入力全体へ最初からアクセス可能な場合の解決手段を提供する. しかし, 実世界の多くの問題では未来に起きる未知の状況に対してどう対処するかを考えなければならない. この場合, 入力は時系列に従って一つずつやってきて, ある時点までに得た情報を元に最善の行動をしなければならない.

このような未来の状況変化にも対処するようなアルゴリズムをオンラインアルゴリズムと言う. これと対比するとき, 従来の入力全体へ最初からアクセス可能な問題に対するアルゴリズムはオフラインアルゴリズムと呼ばれる.

オンラインアルゴリズムの性能を評価する方法の一つとして, 競合比解析がある. 競合比解析では, 入力がすべて与えられていると仮定した場合の理想のアルゴリズム (最適オフラインアルゴリズム) によるコストと, オンラインアルゴリズムのコストを比較する.

本稿では, 第 2節で競合比の定義について説明する. 第 3節でオンラインアルゴリズムで解くべき具体的な問題を提示し, 第 4節でこの問題に対するアルゴリズムを 3 つ提案する. 第 5節ではこれらのアルゴリズムの競合比解析を行なう.

Competitive analysis of convenience store search problem.
INA Lintaro, Graduate School of Informatics, Kyoto University, 2008-08-03.

2 競合比

入力全体へ最初からアクセス可能だった場合の理想のアルゴリズム (最適オフラインアルゴリズム) の性能に対してどれだけ劣るか, という観点でオンラインアルゴリズムの性能を評価する. つまり「なんでも知っている神様だったらこれだけでできる」ということに対して, 神様に比べてどれくらい効率が悪いかを測る.

定義 2.1. 最適オフラインアルゴリズムのコストを C_{OPT}, あるアルゴリズムのコストを C とする. ある定数 c が存在して

$$C \leq rC_{\mathrm{OPT}} + c$$

が成り立つとき, r をそのアルゴリズムの競合比と呼ぶ.

3 問題

問題

ある見知らぬ町の駅に降り立ったところ, 東西に街路が延びていた. コンビニを探したいのだが, 西に行ったら良いのか東に行ったら良いのか分からない.

(1) コンビニを見つけるために歩く距離がなるべく短いアルゴリズムを設計せよ.
(2) 設計したアルゴリズムの競合比を求めよ.

4 アルゴリズム

アルゴリズム 1: 一方向へひたすら歩く

コンビニが見つかるまで東に歩き続ける, あるいは西に歩き続ける, というアルゴリズム.

アルゴリズム 2: 向きを変える度に少し先まで歩く

東へ 1 メートル歩き, コンビニが見つからなければ西に向きを変えて 2 メートル歩き, 見つからなければ東に向きを変えて 3 メートル歩き, ... という具合に l メートル歩いたあと向きを変えて $l+1$ メートル歩くというアルゴリズム.

アルゴリズム 3: 向きを変える度に歩く距離を 2 倍にする

東へ 1 メートル歩き, コンビニが見つからなければ西に向きを変えて 2 メートル歩き, 見つからなければ東に向きを変えて 4 メートル歩き, ... という具合に l メートル歩いたあと向きを変えて $2l$ メートル歩くというアルゴリズム.

5 競合比解析

アルゴリズムのコストは歩く距離である. 最適オフラインアルゴリズムでは, 一番近いコンビニが東西どちらにあるかは既知で, そのコンビニのある方角へ最短距離で行くことができる. 駅の位置からそのコンビニまでの距離を d とすると, $C_{\mathrm{OPT}} = d$ である.

アルゴリズム 1

一方向にひたすら歩くアルゴリズムのコスト C_1 は, 西に一つだけコンビニがあったときに東に歩き続けてしまった場合, いつまで経ってもコンビニは見つからないので, 最悪の場合 $C_1 = \infty$ である. ゆえに

$$C_1 \le \infty = \infty \cdot C_{\mathrm{OPT}} + 0$$

となり, 競合比は ∞ で, 最悪コンビニに辿り着けない場合すらある.

アルゴリズム 2

最初の地点から東に d メートル行ったところにコンビニがある場合, 最後に東に向きを変えてから $2d-1$ メートル歩いたところでコンビニに到達することになる. 最初の地点から西に d メートル行ったところにコンビニがある場合, 最後に西に向きを変えてから $2d$ メートル歩いたところでコンビニに到達することになる. コストは向きを変えてから次に向きを変えるまでの距離の総和なので,

$$\begin{aligned} C_2 &\le \sum_{k=1}^{2d} k = \frac{1}{2} \cdot 2d(2d+1) \\ &= d(2d+1) \\ &= (2d+1)C_{\mathrm{OPT}} + 0 \end{aligned}$$

となり, 競合比は $2d+1$ で, コンビニまでの距離に比例していくらでも大きくなる.

アルゴリズム 3

向きを変える度に歩く距離を 2 倍にするのではなく, x 倍 $(x > 1)$ にする場合に一般化して考え, この一般化されたアルゴリズムのうちで $x = 2$ のものが競合比最小であることを示す.

準備 n 回目の方向転換の後にコンビニに到達するとして, k 回目の方向転換を行なった地点から最初の地点までの距離を $\bar{d}_k$, $k+1$ 回目の方向転換を行なった地点から最初の地点までの距離を d_k とする. つまり k 回目の方向転換から $k+1$ 回目の方向転換までに歩く距離は $d_k + \bar{d}_k$ である.

このとき, 一般化されたアルゴリズムの定義と d_k, $\bar{d}_k$ の定義より, 以下の式が成り立つ.

$$\begin{cases} d_0 = 1 \quad \bar{d}_0 = 0 \\ d_k + \bar{d}_k = x(d_{k-1} + \bar{d}_{k-1}) = x^k \\ \qquad\qquad = (x-1)\displaystyle\sum_{i=0}^{k-1}(d_i + \bar{d}_i) + 1 \\ \bar{d}_k = d_{k-1} \end{cases}$$

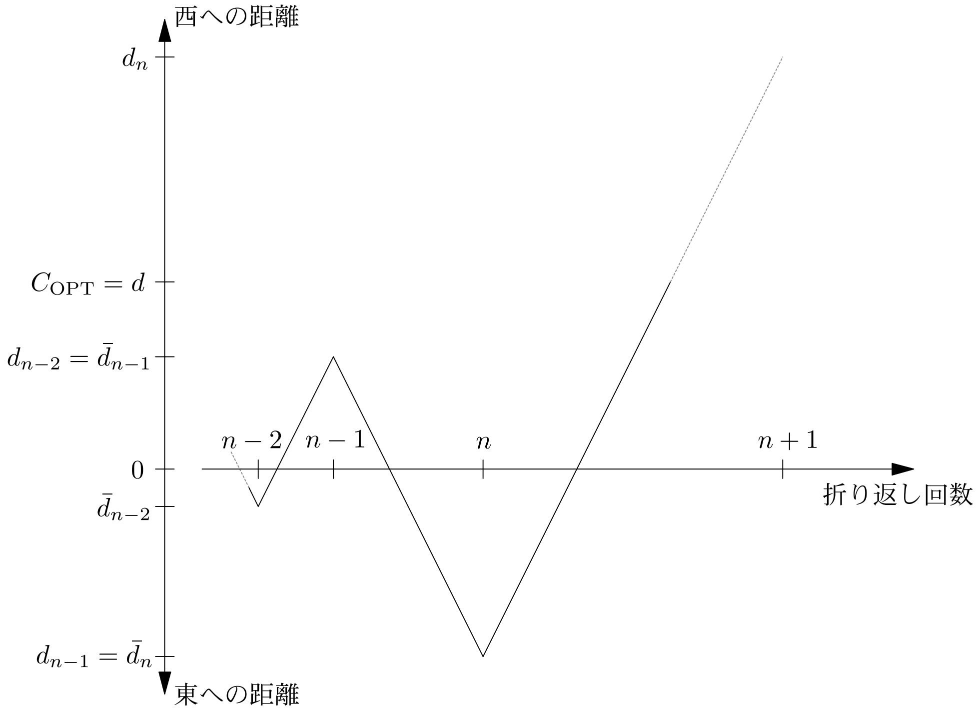

図 1. 方向転換の回数と距離

また, $d_k + \bar{d}_k = x(d_{k-1} + \bar{d}_{k-1})$ と $\bar{d}_k = d_{k-1}$ から, d_k に関する 2 つの 3 項間漸化式が導ける.

$$\begin{cases} d_k + d_{k-1} = x(d_{k-1} + d_{k-2}) \\ \qquad\qquad\;\; = x^{k-1}(d_1 + d_0) \\ d_k - xd_{k-1} = -(d_{k-1} - xd_{k-2}) \\ \qquad\qquad\;\; = (-1)^{k-1}(d_1 - xd_0) \end{cases}$$

$d_1 = x - 1$, $d_0 = 1$ より,

$$d_k = \frac{x^{k+1} - (-1)^{k+1}}{x+1}$$

となる. このことからさらに, $d_{k-1} = \bar{d}_k$ より,

$$\begin{aligned} d_k &= xd_{k-1} + \frac{(-1)^k x}{x+1} - \frac{(-1)^{k+1}}{x+1} \\ &= xd_{k-1} + (-1)^k = x\bar{d}_k + (-1)^k \end{aligned}$$

が言える. 結局,

$$x\bar{d}_k - 1 \leq d_k \leq xd_{k-1} + 1$$

である.

コストの計算 歩く距離の合計は, n 回目の方向転換までに歩く距離, そこから最初の地点までの距離, 最初の地点からコンビニまでの距離 $(= d)$ の和なので,

$$C_3 = \sum_{k=0}^{n-1}(d_k + \bar{d}_k) + \bar{d}_n + d$$

がコストである.

ここで, $d_{n-2} < d$ であることに注意する (さもなければ $n-2$ 回目の方向転換のあとで折り返すことなくコンビニに到達したはずであり, d が d_{n-2} に近いほど悪いケースである) と,

$$\begin{aligned} d_{n-2} + \bar{d}_{n-2} + \frac{x-1}{x^2}\bar{d}_n &< d + \bar{d}_{n-2} + \frac{x-1}{x^2}\bar{d}_n \\ \frac{1}{x^2}(d_n + \bar{d}_n) + \frac{x-1}{x^2}\bar{d}_n &< d + \bar{d}_{n-2} + \frac{x-1}{x^2}\bar{d}_n \end{aligned}$$

と変形できる. 両辺に $\frac{x-1}{x^2}d$ を足せば, $d_n + \bar{d}_n = (x-1)\sum_{k=0}^{n-1}(d_k + \bar{d}_k) + 1$ より,

$$\begin{aligned} &\frac{x-1}{x^2}\left\{\sum_{k=0}^{n-1}(d_k + \bar{d}_k) + \bar{d}_n + d\right\} + \frac{1}{x^2} \\ &\qquad < d + \bar{d}_{n-2} + \frac{x-1}{x^2}(\bar{d}_n + d) \\ &\frac{x-1}{x^2}C_3 + \frac{1}{x^2} < \frac{x^2+x-1}{x^2}d + \bar{d}_{n-2} + \frac{x-1}{x^2}\bar{d}_n \end{aligned}$$

従って, $x\bar{d}_k - 1 \leq d_k \leq xd_{k-1} + 1$ より

$$\begin{cases} \bar{d}_{n-2} \leq \frac{d_{n-2}+1}{x} < \frac{d+1}{x} \\ \bar{d}_n = d_{n-1} \leq xd_{n-2} + 1 < xd + 1 \end{cases}$$

となることから,

$$\begin{aligned}
C_3 &< \frac{x^2+x-1}{x-1}d + \frac{x^2}{x-1}\bar{d}_{n-2} + \bar{d}_n - \frac{1}{x-1} \\
&< \frac{x^2+x-1}{x-1}d + \frac{x}{x-1}(d+1) + \bar{d}_n - \frac{1}{x-1} \\
&= \frac{x^2+2x-1}{x-1}d + \bar{d}_n + 1 \\
&< \frac{x^2+2x-1}{x-1}d + xd + 2 \\
&= \frac{2x^2+x-1}{x-1}d + 2 \\
&= \frac{2x^2+x-1}{x-1}C_{\mathrm{OPT}} + 2
\end{aligned}$$

となる.

競合比を最小にする x 競合比を最小にする x は, 関数 $f(x) = \frac{2x^2+x-1}{x-1}$ を最小にする x の値である.

$$f'(x) = \frac{2x(x-2)}{(x-1)^2}$$

より, 関数 $f(x)$ の増減表は表 1のようになり, $x > 1$ では $x = 2$ で最小値 $f(2) = 9$ をとる.

x	1	$\cdots$	2	$\cdots$
$f'(x)$		$-$	0	$+$
$f(x)$		$\searrow$	9	$\nearrow$

表 1. $f(x)$ の増減表

つまり, 一般のアルゴリズムのうち $x = 2$ のものが競合比を最小にする.

競合比 以上の議論により, $x = 2$ のとき

$$C_3 < 9 \cdot C_{\mathrm{OPT}} + 2$$

である. よって, 競合比は 9 未満 で, このアルゴリズムでは競合比は定数で抑えられる.

6 おわりに

本稿では, オンラインアルゴリズムの性能評価手法である競合比解析の具体例として, コンビニ探索問題に対するアルゴリズムの設計と競合比の計算を行なった.

参考文献

1. 岩間一雄. アルゴリズム・サイエンス: 出口からの超入門. アルゴリズム・サイエンス シリーズ 2. 共立出版, 2006.