2004－MPS－51　(2)
2004／9／13

文字数最大しりとり問題の解法

乾 伸雄, 品野 勇治, 小谷 善行
東京農工大学工学部情報コミュニケーション工学科

本論文では、しりとり全体に含まれる文字数を最長とする文字数最大しりとり問題をネットワークフロー問題としてモデル化し、 LP ベースの分枝限定法による解法および実験結果について述べる。単語数を最大にする最長しりとり問題に対して、問題を記述するための変数が最大単語長に比例して多くなる特徴を持つ。実験は実際の辞書に含まれる単語について行った。実験の結果、最長しりとり問題と同じく文字数最大しりとり問題は現実的な時間で解ける問題であることがわかった。

Solving the Maximum Character Length Shiritori Problem

Nobuo Inui, Yuji Shinano, Yoshiyuki Kotani
Department of Computer, Information and Communication Sciences,
Faculty of Engineering, Tokyo University of Agriculture and Technology

This paper describes the maximum character shiritori problem where a shiritori sequence with the maximum number of character is requested. We model this problem as a graph, propose a solution using LP-based branch-and-bound method and show experimental results. Against the longest shiritori problem where a shiritori sequence with the maximum number of words, this problem is characterized by the increased number of variables in proportion to the maximum length of words. We use actual words statistic in Japanese dictionaries for our experiments. From the results, the solution is easy to find as same as the longest shiritori problem.

1　はじめに

しりとりは、単語の末尾の音に対して同じ音で始まる単語を継いでいき、次の単語を出すことができなかったゲーム参加者が負けとなるゲームである。最長しりとり問題 [7] は、各プレイヤーが協調して長く続くように単語を出していく言葉遊びと捉えることができる。最長しりとり問題については LP ベースの分枝限定法による解法が示されている。これに対して、本論文では、しりとりとなる単語列で文字数を最大とする文字数最大しりとり問題に対し、 LP ベースの分枝限定法による解法を示し、実験的に問題の性質を明かにする。ここで、文字数最大しりとり問題を次のように定義する。

文字数最大しりとり問題
文字数最大しりとり問題は、文字数が最大となるしりとりをみつける問題である。

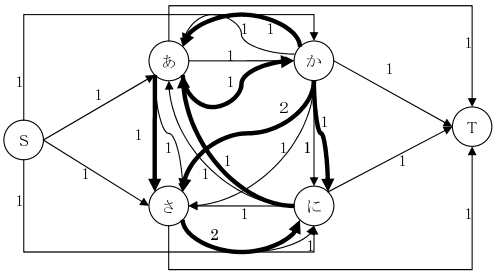
しりとりのような言葉遊びとして日常的なものであるが、コンピュータでしりとりを扱った研究として、連想しりとり [2] や二人完全情報ゲームとしてしりとりを扱った研究 [1] がある。しりとりは人間の発想力を高める題材として有望なものであり、本研究の成果は今後のアプリケーション開発に寄与すると考えている。

最長路問題としての文字数最大しりとり問題は、最長しりとり問題と同様に、与えられたグラフから最大オイラー路サブグラフを求める問題と等価であり、 NP- 困難な問題となる [3]。本研究の先行研究 [7] では、ひらがなを頂点、単語をアークとするグラフの条件を整数計画問題として記述し、実際のしりとりについての実験を行った。

2　文字数最大しりとり問題のモデル化

最長しりとり問題のモデルについては従来研究 [7] で行われているが、本論文でも同様のモデル化を行う。図 1に、最長しりとり問題と文字数最大しりとり問題のためのネットワークを示す。図 1で、 s および t は３章での解法で用いる仮の頂点であり、それぞれ、始点、終点を表す。それぞれの問題のネットワーク表現において、 s と t を結ぶフローを求めることによって、問題を解決する。両者の間の違いは、単語の長さを考慮しない最長しりとり問題に対し、文字数最大しりとり問題では単語の長さを考慮することにある。

本論文は、与えられた辞書において行われるしりとりをネットワーク表現でモデル化する。文字数最大しりとり問題を解くために、文字、単語の長さおよび単語数を扱う。単語の始めの文字または終りの文字になる文字集合を V、



文字数最大しりとり問題におけるネットワーク表現,各アークの数字は単語数
細線は単語長１，太線は単語長２のアークを示す

図 1: 文字数最大しりとり問題のネットワーク表現

単語の長さの集合を L とおく。ここでは、文字を数字に対応づけて表現し、文字数 n の集合を次のように表す。

$$V = \{1,2,\ldots,n\}, L = \{1,2,\ldots,l\}$$

これによって、$V \times V \times L$ によって表される集合は文字数最大しりとり問題を解くための単語情報を表す。あるひらがな $i \in V$ を始点、$j \in V$ を終点とし、長さが $k \in L$ であるアークを a_{ij}^k、その辞書などで既知な単語の数を f_{ij}^k と記述する。各々の集合を A,F で表す。これによって、$N = (G = (V \cup \{s,t\}, A), F)$ によって構成されるネットワークは図1のようになる。ただし、$f_{si}^1 = f_{it}^1 = 1, f_{si}^k = f_{it}^k = 0 (k > 1), i \in V, k \in L$ である。このとき、文字数最大しりとり問題は次のように定義される。

文字数最大しりとり問題のネットワークによるモデル化

x_{ij}^k をアーク a_{ij}^k におけるフローとする。有向ネットワーク $N = (G = (V \cup \{s,t\}, A), F)$ において、頂点 s から流量 1 のフローが流出し、頂点 t へも流量 1 のフローが流入する。このとき、フローが流れているアーク $(x_{ij}^k > 0)$ により誘導されるグラフは連結とする条件のもとで、$\sum_{i \in V \cup \{s\}} \sum_{j \in V \cup \{t\}} \sum_{k \in L} k x_{ij}^k, 0 \leq x_{ij}^k \leq f_{ij}^k$ を最大とするフローを求める問題。

ここでのフローは次のように特徴付けられる。

$$\sum_{j \in V \cup \{s,t\}} \sum_{k \in L} x_{ij}^k - \sum_{j \in V \cup \{s,t\}} \sum_{k \in L} x_{ji}^k = \begin{cases} 1: i = s \\ 0: i \neq s,t \\ -1: i = t \end{cases}$$

$$0 \leq x_{ij}^k \leq f_{ij}^k \forall i,j \in V, \forall k \in L$$

$f_{ij}^k = 0$ であるアークはしりとりで使うことができないので、本論文では表現を簡単にするためにこれらの条件を持つアークのことを「アークがない」と表現する。

3 文字数最大しりとり問題の解法

文字数最大しりとり問題は、最長しりとり問題と同じく、次の二つの条件を満す文字数が最大となるフローを求める問題として、整数計画問題に帰着できる。

1. 頂点 s から流量 1 のフローが流出し、頂点 t へ流量 1 のフローが流入する
2. フローが流れているアークにより誘導されるグラフが連結であること

本論文では、文字数最大しりとり問題のための緩和問題を繰り返し解く方法を提案する。これは、先のフローの条件 (1) は容易に記述できることを利用している。フローの条件 (1) で文字数を最大とする問題を解く場合、第二の条件は考慮されないので、連結していないグラフが得られる場合がある。このため、次のような問題の分割（分枝操作）を考える。

緩和問題 $(\mathbf{RP}_e)$ において、s から t への経路上の頂点集合を V_e とおく。このとき、この補集合 $V - V_e$ の要素である頂点にアークが存在する場合、連結ではない。このとき、問題は次のように分割（分枝操作）できる。

1. 頂点集合 V_e によって構成される文字数最大しりとり問題
2. 頂点集合 V_e から $V - V_e$ へのアークが存在する文字数最大しりとり問題

これによって、緩和問題 $(\mathbf{RP}_e), e = 0,1,\ldots$ は次のように定義できる。

$$(\mathbf{RP}_e) \quad \text{最大化} z = \sum_{i \in V \cup \{s\}} \sum_{j \in V \cup \{t\}} \sum_{k in L} k x_{ij}^k$$

条件

$$\sum_{j \in V} x_{sj}^1 = 1$$

$$\sum_{j \in V} \sum_{k \in L} x_{ij}^k - \sum_{j \in V} \sum_{k \in L} x_{ji}^k = 0 \quad \forall i \in V$$

$$\sum_{j \in V} x_{jt}^1 = 1$$

$$\sum_{i \in V_m} \sum_{j \in V - V_m} \sum_{k \in L} x_{ij}^k \geq 0 \qquad \forall m = 1,2,\ldots,e-1$$

$$0 \leq x_{ij}^k \leq f_{ij}^k, \quad \forall i \in V, \forall j \in V, \forall k \in L$$

$$0 \leq x_{sj}^1 \leq 1, \quad \forall j \in V$$

$$x_{sj}^k = 0, \quad \forall j \in V, \forall k \in L - \{1\}$$

$$0 \leq x_{jt}^1 \leq 1, \quad \forall j \in V$$

$$x^k_{jt} = 0, \quad \forall j \in V, k \in L - \{1\}$$
$$x^k_{ij} \in \mathbf{Z}, \quad \forall i \in V \cup \{s\},$$
$$\forall j \in V \cup \{t\},$$
$$\forall k \in L$$

緩和問題 ($\mathbf{RP}_0$) は、制約条件の係数が 1,0 の何れかであり、その任意の正方小行列の行列式の値が 1,0,-1 となるため、 Totally Unimodular Matrix(TU)[6] となり、線形計画問題として整数解が得られる。しかしながら、緩和問題 ($\mathbf{RP}_e$),$e > 1$ はその保証がないため、次の解を求めるアルゴリズムでは、線形計画問題で整数解が得られない場合、整数計画問題として解を得る。

Algorithm Making Maximum-Char Shiritori
begin
  $e := 0$; { 分枝の回数 }
  $x1 := \{\}$ {$x1$: 要素が x^k_{ij}である配列 }
  $y1 := 0$; { $y1$:s,t をつなぐしりとりの文字数 }
  $z1 := 0$; { $z1$: 目的関数の暫定値 }
  while true **do**
  begin
    (RP_e) を LP で解く;
    if ($\mathbf{RP}_e$) が整数解でない **then**
      ($\mathbf{RP}_e$) を IP で解く;
    if ($\mathbf{RP}_e$) に実行可能解がない **then**
      goto 1;
    $x = \{x^k_{ij} | i, j \in V, k \in L\}$ {($\mathbf{RP}_e$) の解 }
    $y =$ ($\mathbf{RP_e}$) の解の s,t をつなぐ文字数
    $z =$ ($\mathbf{RP_e}$) の解の目的関数値
    if z=y **then** { 解が連結であった場合 }
      begin
        if $y1 < y$ **then**
        begin
          $x1 := x$;
          $y1 := y$;
          $z1 := z$;
        end;
        goto 1;
      end
    else { 解が連結でない場合 }
      begin
        if $z < y1$ **then**
          goto 1;
        if $y1 < y$ **then**
        begin
          $x1 := x$;
          $y1 := y$;
          $z1 := z$;
        end;
        $e := e + 1$;
      end
  end
1: $x1$ よりしりとりを構成する
  { 文字数最大しりとりの文字数は、 $y1 - 2$ }
end;

アルゴリズム Making Maximum-Char Shiritori において、変数 x および $x1$ は、各アークの通過回数 x^k_{ij} を表す配列である。変数 z および $z1$ は、緩和問題 ($\mathbf{RP}_e$) の目的関数の値、つまり、求められたネットワークにおける文字数を表している。これは連結かどうかは関係なく計算される。これに対し、変数 $y, y1$ はそれぞれ $x, x1$ における s と t を結ぶしりとり上での文字数を表している。このしりとりの通る頂点が V_e となる。

4 実験

前章で述べた手法を評価するための実験を行った。実装は、 Windows XP 上の Cygwin(ver. 2.218) で gcc(ver. 3.2) を用い、整数計画法のソルバーとして、 GNU GLPK(ver. 4.2) を使った。

実験材料としては、広辞苑より名詞を抜き出したもの [4](広辞苑) を用いて行った。この辞書には、 192,687 の単語が含まれている。

実験は各アークの単語数を半減することで行った。半減は小数点以下切捨てとしている。表 1に結果を示す。最長しりとり問題は、従来研究 [7] では単語の長さを考慮していなかったが、本実験では文字数最大しりとり問題と同じく、文字数を考慮したモデル化によって解いている。表の各項目は次の内容を表している。

ALL: アークが存在すう頂点数, **InOut:** 出るアークと入るアークが存在する頂点数, **In:** 入るアークが存在する頂点数, **Out:** 出るアークが存在する頂点数, **アーク数:** 辞書中に単語数が 1 個以上のアーク数, **単語数:** 辞書中の総単語数, **IP:** 解くために必要な緩和問題の数, **TIME:** 解くためにかかった時間（秒）, **Len:** 作成されたしりとりに含まれる単語数, **Char:** 作成されたしりとりに含まれる文字数,

表 1の結果を見ると、特定の場合、緩和問題の数がふえていることが分る。従来研究では、緩和問題の数たかだか 4 程度が限界であったが、本実験では単語の長さごとにアークを作ったことが原因と考えられる。緩和問題が m 個できるということは、ネットワークを m 回空でない二つの部分ネットワークに分割できることを表しており、スパースなネットワークとなっていることを意味する。

結果的に、国語辞典を用いた場合、ひらがなで表される頂点数に対し、単語数が非常に多いため、本論文で用いた緩和問題による問題解決はうまく動作すると言えるが、頂点数が多くなったり、単語数の傾向がことなるネットワー

表 1: 各アークの単語数を半減した場合の結果 (広辞苑)

頂点数				アーク数	単語数	文字数最大しりとり問題				最長しりとり問題			
All	InOut	In	Out			IP	TIME	Len	Char	IP	TIME	Len	Char
68	68	68	68	23070	192687	1	55.672	83900	436106	1	51.797	86796	406971
68	66	67	67	15776	89345	1	20.109	36052	181824	1	19.86	37285	170750
68	66	67	67	9806	39709	2	5.892	14107	69432	2	5.438	14554	65612
68	64	65	67	5415	16600	2	1.218	4808	23139	3	1.218	4949	21842
64	54	56	62	2636	6416	15	0.641	1339	6402	6	0.36	1378	6124
61	33	34	60	1136	2255	54	7.531	310	1463	60	4.313	314	1368
51	14	15	50	433	714	1	0.047	48	217	1	0.031	48	199
34	4	8	30	132	184	1	0.031	6	27	1	0.032	6	25
16	0	4	12	32	38	1	0.016	1	6	1	0.03	1	4
6	0	2	4	6	6	1	0.031	1	5	1	0.031	1	4

クに対しては、緩和問題が多くなることで、結果を得るまでに時間が必要となる可能性がある。

紙面の都合で割愛するが、ランダムにアークを半減する実験、および、特定の文字数で最長しりとりを求める実験、特定の単語長で文字数最大しりとりを求める実験を行った。ランダムにアークを半減した場合、さらに多くの分枝操作がみられた。また、文字数、単語数に対する制約を入れたばあい、線形計画問題で整数解が得られない場合が存在した。このようなことから、従来研究 [7] のように、比較的緩和問題が少く、線形計画問題で整数解が得られるような場合は、しりとり問題でも容易に解ける場合であると言える。

5 おわりに

本論文では、最長しりとり問題の一般化として、頂点間に単語長ごとにアークを持つ文字数最大しりとり問題を定義し、この問題を解くために、LP ベースの分枝限定法による実験を行った。文字数最大しりとり問題の緩和問題は、最長しりとり問題の緩和問題に対し、定数倍の変数が増える問題であるが、実験結果では、それ以上の実行時間が係る問題となることがわかった。しかしながら、現存する国語辞典の見出し語の範囲では十分解ける問題であることもわかった。

謝辞：本研究の一部は科研費基盤研究 (B)(2) (No.15300269) の補助を受けている。

参考文献

[1] T.Ito, T.Tanaka, Z.Hu, M.Takeuchi: An Analysis of Word Chain Games, J.of IPSJ, Vol.43 No.10(2002)

[2] T.Kanasugi, K.Matsuzawa, K.Kasahara: Applications of ABOUT Reasoning to Solving Wordplays, TR.of IEICE, NLC96-31, pp.1-8(1996)

[3] H-J.Lai: Eulerian Subgraphs Containing given Edges, Discrete Mathematics, 230, pp.63-69(2001)

[4] I.Niimura(eds):Koujien Ver.4, Iwanami(1992)

[5] Institute for New Generation Computer Technology: Keitaiso Jisyo,ICOT Freeware No.33(1993)

[6] B. Doerr: Linear Discrepancy of Basic Totally Unimodular Matrics, The Electronic Journal of Combinatorics, 7, pp.1-4(2000)

[7] N. Inui, Y.Shinano, Y.Kounoike, Y.Kotani: Solving the Longest Shiritori Problem, 03-MPS-48, IPSJ(2003)