2004－MPS－48　(4)
2004／3／1

最大長しりとり問題の解法

乾 伸雄, 品野 勇治, 鴨池 祐輔, 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、最長しりとり問題をネットワークの問題として定義し、整数計画問題に帰着した定式化を行う。この定式化では、変数の数が頂点に対して、指数オーダーで増加するため、事実上、整数計画問題として直接的に解くことは難しい。そのため、緩和問題を設定し、 LP ベースの分枝限定法によって解決した。これによって、１３万語程度の辞書から最長しりとりを Xeon2.8GHz プロセッサの PC を使って 1 秒足らずで作成することができた。また、本論文では、局所探索による解法と比較し、問題の困難さを実験的に調べた。さらに、様々なインスタンスにおける解を分析することで、最長しりとり問題の性質を調べた。

Solving the Longest Siritori Problem

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

This paper describes the definition of the longest Shiritori problem as a problem of network and the solution which uses the the integer problem. This formulation requires a large number of variables in proportion to the exponential order. Against this issue, we propose a solution based on the LP-based branch-and-bound method, which gradually solves the relaxation problems. This method is able to calculate the longest Shiritori sequences for 130 thousand words dictionary within a second in Xeon2.8GHz PC. In this paper, we compare the performances for the local-heuristic search and investigate the results for a variety of instances to explore the longest Shiritori problem.

1　はじめに

しりとりは，通常 N 人ゲームで，前の人の言った末尾の文字で始まる言葉を言い，つないでいくゲームである。連想によるしりとりの研究 [2] では、人間の言葉から関係のある概念を検索して出力する。言葉の選び方は無作為であり、連想しずらい言葉が出たときなど、ゲームが早く終了してしまう。しりとりを二人完全情報ゲームの枠組みでモデル化し、単語を枝とする AND/OR 木によって、与えられた辞書におけるしりとりが、「先手必勝」か「後手必勝」かを判断する研究 [1] もある。本論文では、このようなしりとりを使ったシステムへの基礎研究として、最長のしりとり単語列を求める手法を提案し、実験的に実際の単語から作られたしりとりを分析する。

最長のしりとりを求める問題は、最長経路問題に帰着できる。このアルゴリズムは、 Directed Acyclic Graph について、頂点数 n に対し、$O(n^2)$ であることが知られている。また、 Directed Cyclic Graph については、 Liao-Wong Algorithm や Bellman-Ford Algorithm などが知られているが、しりとり問題は、ポジティブサイクルを持つ問題のため、これらのアルゴリズムを適用することは困難である。

本論文では、線形計画法の枠組みで問題をとき、条件を追加することで、最長しりとを作成する方法を述べる。また、実験的に国語辞典記載の単語によるしりとりの分析をおこなう。

2　最長しりとり問題の定義

最長しりとり問題とは、単語の集合が与えられたときに、最も多くの単語を使ったしりとりを作成する問題であり、次のように定義する。

最長しりとり問題
しりとり問題は、ある単語から始めて、その単語の末尾の文字で始まる単語を提示していき、次に提示する単語がない場合に終了する。本稿では、単語は全てひらがな表記されているものとして、次に提示する単語がなくなった時点でしりとりは終了する。最長しりとり問題は、与えられた辞書において、最も長い単語列を作成する問題と定義する。

この最長しりとり問題をとくために、グラフ表現を導入する。単語の先頭あるいは末尾の文字を頂点に対応づけ、文字数 n の頂点集合を $V = \{1, 2, \ldots, n\}$ とする。各単語はアークに対応付け、アーク集合を、 $A = B_{11} \cup B_{12} \cup$

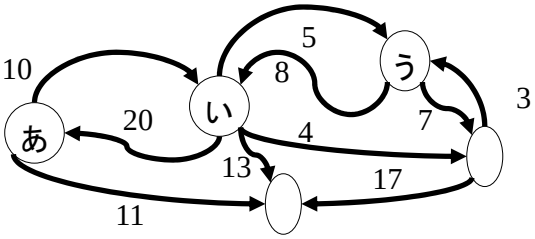

図 1: しりとりのモデル

$\ldots \cup B_{nn}$ とする。ここで、B_{ij} は、文字 i で始まって、文字 j で終る単語の集合とする。

F を単語数の集合とし、A の代りに次の E,F を用いる。ここで、e_{ij} は文字 i から j へのアークであり、f_{ij} は、アーク e_{ij} に対応する単語の数 (辞書で与えられる単語の上限) である。

$E = \{e_{11}, e_{12}, \ldots, e_{nn}\}$
$F = \{f_{11}, f_{12}, \ldots, f_{nn}\}$ where $f_{ij} = |B_{ij}|$

しりとり問題のネットワーク表現
遷移ネットワーク $N = (V, E, F)$ において、V は頂点集合であり、E は二つの頂点を結ぶアークである。重み F は各アークの最大の通過回数を表す。

最長しりとり問題は、全てのアークの重みが等しい場合の最大オイラー路サブグラフ (Largest Eulerian Subgraph) を求める問題に帰着される。一般に最大オイラー路サブグラフを求める問題は、最大のハミルトン路を求める問題と等しく、NP- 完全な問題となる [4]。

3 LP ベース分枝限定法による解法

最長経路問題の整数計画法（線形計画法）による記述の先行研究としては [3] があげられる。本研究も同様な方法で最長しりとり問題を、先頭が文字 i、末尾が j である求めたい単語の数を示す変数 x_{ij} を用いて、整数計画問題として記述する。f_{ij} は辞書で与えらる単語の上限を表す。この記述は、有向グラフにおけるオイラー路の条件である、

(a) 各頂点に入る枝の数と出る枝の数が等しい（始点では出ていく数が一つおおく、終点では入ってくる数が一つおおい）

(b) グラフが連結であること

という二つ条件からなる。解法では、(a) を初期の緩和問題とし、(b) の条件を満すように、条件を加えていく。

問題を解く際には、遷移ネットワークにスーパー・ソース s と，スーパー・シンク t、および，頂点 s から全ての頂点へと，全ての頂点から頂点 t へ重み 1 のアークを加えたネットワークを構成する。

最長しりとり問題を直接的にとくことが困難なので，まず (a) のフローとしての条件だけを考慮し，各アークを流れるフローの総和を最大化する緩和問題 $(\mathbf{RP}_0)$ を解く。

$(\mathbf{RP}_k)$

$$\text{最大化} \quad z_k = \sum_{i \in V \cup \{s\}} \sum_{j \in V \cup \{t\}} x_{ij}$$

$$\text{条件} \quad \sum_{j \in V} x_{sj} = 1$$

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

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

$$\sum_{\substack{i \in V_l^*, \\ j \in V \setminus V_l^*}} x_{ij} \geq 1, \quad \begin{array}{l} if\ k > 0, \\ l = 0, \ldots, k-1 \end{array}$$

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

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

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

$$x_{ij} \in \mathbf{Z}, \quad \forall i \in V \cup \{s\}, \quad \forall j \in V \cup \{t\}$$

(a) の条件しか考えない問題 $(\mathbf{RP}_0)$ は，最長しりとり問題の緩和問題となっているので，その目的関数値 z_0 は最長しりとり問題の上界値を与える。問題 $(\mathbf{RP}_0)$ の制約条件は，よく知られた整数定理の成り立つ制約条件である。つまり，各 f_{ij} が整数なので，線形計画問題を解くことで，各 x_{ij} は，必ず整数値として求まる。

$(\mathbf{RP}_0)$ の解 $x_{ij} > 0$ のアークで構成されるグラフが，複数の連結成分に分かれた場合、最長しりとり問題の許容解を２つに分割 (**分枝操作**) する以下の問題を考える。

(1) 頂点 $i \in V_0^*$ から頂点 $j \in V \setminus V_0^*$ への単語の遷移が少なくとも１つはある最長しりとり問題

(2) 頂点 $i \in V_0^*$ から頂点 $j \in V \setminus V_0^*$ への単語の遷移は１つもない最長しりとり問題

この２つの問題において，いずれか長いしりとりを構成したものが最適解である。

(2) の場合は、すでに求まった解から最適解が構成できる。(1) に関しては，「頂点 $i \in N_0^*$ から頂点 $j \in V \setminus V_0^*$ への単語の遷移が少なくとも１つはある」という条件

$$\textstyle\sum_{i \in V_0^*, j \in V \setminus V_0^*} x_{ij} \geq 1$$

を $(\mathbf{RP}_0)$ に加えた $(\mathbf{RP}_1)$ を考える。$(\mathbf{RP}_1)$ に対して，$(\mathbf{PR}_0)$ と同様の操作を行う。以上を再帰的に繰り返しアルゴリズムを構成する。

表 1: 文字頻度による最長しりとり長 (任意の品詞)

単語数	最長の長さ	割合	IP 適用回数	計算時間 (秒)
137,335	56,519	41%	1	0.53
68,417	27,718	41%	1	0.49
33,497	13,339	40%	1	0.47
16,079	6,183	38%	1	0.39
7,406	2,654	36%	1	0.30
3,219	1,016	32%	1	0.20
1,265	303	24%	1	0.16
444	70	16%	1	0.13
124	15	12%	1	0.11

表 2: 文字頻度による最長しりとり長 (名詞だけ)

単語数	最長の長さ	割合	IP 適用回数	計算時間 (秒)
192,687	86,788	45%	2	1.36
95,255	42,236	44%	1	1.20
46,595	20,077	43%	1	1.02
22,351	9,115	41%	3	0.84
10,361	3,792	37%	1	0.5
4,535	1,372	30%	1	0.27
1,853	401	22%	1	0.17
697	91	13%	1	0.12
231	18	8%	1	0.11

4 実験

国語辞典 (広辞苑 4 版) より 2 種類の方法で取り出した単語で行った実験と、擬似的な設定で網羅的に行った実験の結果について示す。実験は、 MS-Windows 上の Cygwin で gcc を用いて行い、線形計画法（および整数計画問題）のソルバーとして、 GNU GLPK を用いた。実験に使用した PC は、 Xeon 2.8GHZ, Dual Processer, 2GB Memory である。

4.1 国語辞典における最長しりとり問題

実験に用いたデータは、広辞苑より、見出し語が清音で始まる単語を取り出し、重複を省いたデータ（名詞以外の品詞も含む）と全名詞を抽出したデータである。また、全てひらがなでの繋がりを考えるため、末尾の記号を削除したり、旧仮名遣い、片仮名、拗音、促音を対応するひらがなに変換している。例えば、拗音などの「ょ」などを「よ」などに変換する処理を行った。この結果、しりとりで使われる文字は 70 種類となった。

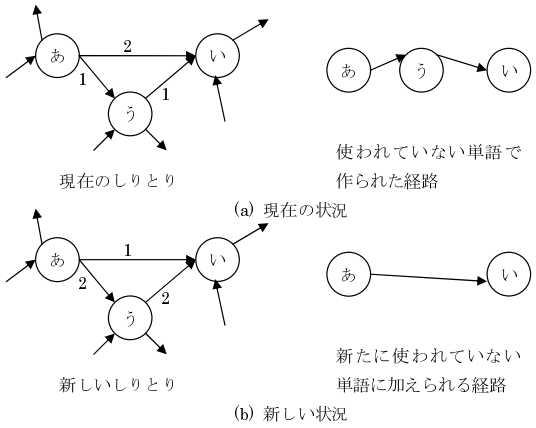

図 2: 局所探索における経路の交換

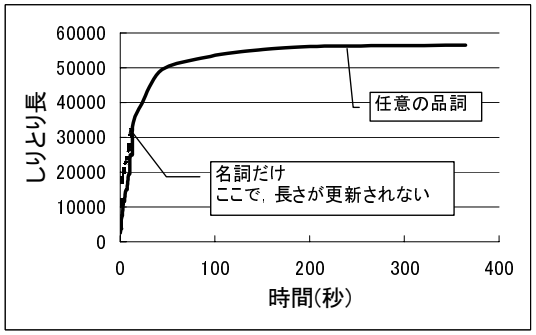

図 3: 探索によるしりとり長

国語辞典を用いた実験では、多くの問題で $(\mathbf{RP}_0)$ を解くだけで解を求められた。表 1、表 2に実行結果を示す。実行時間は GNU GLPK と分枝操作を行っている時間で、経路を生成する時間は含まれていない。表 1、表 2は、同じ始点と終点を持つ単語の数を 1/2 （小数点以下切捨て）した辞書で得られる最長しりとり問題での結果も示している。文字数（遷移ネットワークでの頂点数）は基本的に変らないが、単語数が減ることで、実行時間は短くなる。ある程度多く単語数がある場合、辞書中の単語数で最長しりとりで使われる単語数は 30% 〜 40% であることがわかる。これを多いとみるか少ないとみるかは議論が残る所であるが、普通の人間にはとても作成できるものではない、一種の感動を人間に与えることが出来る長さであるといえる。なお、一般のグラフにおいては、 2/3 以下であろうことが推測されている [5]。

局所探索でも、しりとりを作成可能である。ここでの局所探索は、基本的に短い部分経路を長い経路で置き換える（山登り法）ことで、しりとりを延す方法である (図 2を参照)。実際の実装では、最近置き換えた経路を置き換えないという、タブー探索的なヒューリステリックを導入している。探索の過程を図 3に示す。解を得るのに 365.7 秒を必要とした。他のデータ（全名詞見出し語）では最適解

表 3: 始点・終点ごとの最長しりとり長

最長しりとり長	始点・終点の種類数	始点の種類数	終点の種類数	例
56519	51	3	17	あ・ぐ, は・ご
56518	197	13	26	も・ぺ, ふ・げ
56517	389	29	35	む・ざ, ま・ぴ
56516	424	37	53	ろ・び, り・ざ
56515	449	38	65	ろ・で, わ・く
56514	504	38	53	わ・れ, よ・の
56513	395	35	44	わ・た, ま・お
56512	194	25	35	り・せ, む・ひ
56511	52	9	17	ろ・ほ, い・は
56510	5	1	5	る・あ, る・へ

表 4: 網羅的な探索における IP の適用回数

問題の大きさ	IP 適用回数				
	1	2	3	4	6
3	8	0	0	0	0
4	291	0	0	0	0
5	29869	(4,0)	0	(0,4)	0
6	11,338,759	(5903,0)	(3315,0)	(20,558)	(0,4)

() 内は,(制約を加えて最適解が発見された数, 条件が厳しく前に求まった解が最適だった数) を示す

を得ることができなかった。このような局所探索の欠点として、ある程度以上の長さの経路が発見されると、それを改善することが難しくなっていくことが挙げられる。しかしながら、このような局所探索によっても、線形計画法で得られた解を得られる場合があることから、国語辞典より最長しりとりを得る問題は、問題の性質として、最長の解が得られやすい問題であると考えることができる。

表 3に最長しりとりの始めの文字と終りの文字を明示的に指定した場合の各最長しりとりに対する始点と終点の種類数を示す。最長しりとりの長さは、どの始点終点を選んでも、あまり変化がない。同じ文字を一度しか通らず、どの文字から出ても同じ文字に戻ってこれる経路のことを輪と呼ぶと、この結果は、広辞苑から作成されたしりとりが輪になっている部分がほとんどであり、その輪にどのように入るかで、最長の長さがきまることを示している。最小の長さの輪より抽出していった場合、残った経路の長さは 29 であった。

4.2 網羅的な実験

最長しりとり問題において、回転、反転、文字番号の入れ換え等の反復を避けて、文字数 3 〜 6 までの範囲を網羅的に解いた。この条件のもとで、分枝操作された回数を表 4に示す。この表より、殆どの場合において、分枝操作をすることなく連結グラフが得られていることが分る。しかしながら、文字数が増えることによって、分枝操作の回数が増える傾向にあることもわかる。また、 IP の適用回数が少ないときは、より改善された解が得られやすく、増えると、追加される条件が厳しく、解が存在しなくなることがわかる。

5 おわりに

本論文では、最長しりとり問題を定義し、その解法を示した。今後、結果を人間の学習システム等に活用することを考えている。

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] M.Fischetti, J-J.Salazar-Gonzalez, P.Toth: The Genralized Traveling Salesman Problem and Orienteering Problems in The Generalized Traveling Salesman Problem and its Variations, Kluwer Academic Publisher(2002)

[4] S.Nakayama, S.Masuyama: A Parallel Algorithm for Solving the Longest Path Problem in Outerplannar Graphs, IEICE Transaction D-I, Vol.J78-D-I, No.6, pp.563-568(1995)

[5] Dengxin Li, Deying Li, J.Mao: On Maximum number of Edges in a spanning Eulerian Subgraph, Discrete Mathematics, 274, pp.299-302(2004)

[付録：最長路の例]

1: あヴェマリあ, あンモニあ, あンぺあ, あンプリファイあ, あンフェあ,
6: あンパイあ, あンティオキあ, あンダルシあ, あンダーウェあ, あンタキあ,
.....
56511: ようすもの, のあざみ, みあつめ, めいしいれ, れいいき,
56516: きえつく, くいら, らいはる, るモンど