OR研究の最前線

最長しりとり問題とその解法

品野　勇治，乾　　伸雄

1. はじめに

本稿では，最長しりとり問題および文字数最大しりとり問題の解法，さらに，それらの数値実験について紹介します．本稿の内容は『OR研究の最前線』ではなく，ORを研究している方々にとってはオーソドックスな，最適解を求める解法の適用例です．そのため，解法そのものに目新しさはないと思います．最長しりとり問題は，とても分かりやすい問題で，かつ，「トリビアの泉」という視聴率の高いテレビ番組で取り上げられ，広く知られるようになりました．そして，どこにでもあるPCとフリーソフトウェアの利用により，実際に数十秒～数分程度で最適解が得られることから，解法の面白さを伝えやすい題材と言えます．ですから，研究分野を紹介するには良い題材かもしれません．『OR研究の最前線』の記事としては，期待を裏切るかもしれませんが，楽しんでいただければ幸いです．

テレビ番組では，「広辞苑に載っている言葉を使って作ることができる最も長いしりとり」を求めるという内容で紹介されました．この際のしりとりのルールは，一般なものであり，次の通りでした．

- 名詞だけを使用
- 長音は母音で次の単語へ繋ぐ
  -「コーヒー（こぉひぃ）」⇒「いるか」
- 「じ」=「ぢ」，「ず」=「づ」は同じ音として扱う
- 例えば，「橋（はし）」=「箸（はし）」は同じ単語として扱い，使用は1度
- 1度使用した単語は再び使用しない
  -「すいか」⇒「からす」⇒~~「すいか」~~
- 「しりとり」という単語から始め，単語の語尾に「ん」がついた時点で終了

本稿では，与えられた辞書の単語を使って，最も長い（最も単語数の多い）しりとりを一つ求める問題を**最長しりとり問題**と呼ぶことにします．ただし，テレビ番組で紹介された問題よりも少し問題を一般化し，任意の単語から開始して，繋ぐ単語がなくなった時点で終了する最も長いしりとりを作る問題とします．また，長さの定義を，しりとりに含まれる単語の文字数とした問題を**文字数最大しりとり問題**と呼ぶことにします．

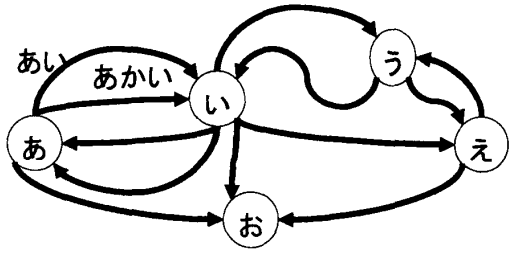
最長しりとり問題で求めたいものは，最大の単語数を持つしりとりです．ただし，広辞苑規模の辞書の場合，名詞だけを取り出しても15万語以上の単語があり，単純な列挙では現実的な時間内に最適解が得られません．最長しりとり問題は，与えられた条件を満足するような組合せ集合の中から，目的とする関数（**目的関数**）値を最大，あるいは，最小にする，ORの分野ではよく知られた**組合せ最適化問題**です．そこで，最適解を求めるための道具である分枝限定法[1, 2]を適用します．

2. 最長しりとり問題のモデル化

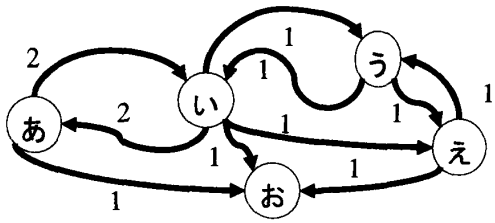
ここでは，最長しりとり問題に対する，グラフ表現によるモデル化と，ネットワーク表現によるモデル化を示します．

しりとりで単語のつながりをみるためには，各単語の開始文字と終了文字だけが問題であるので，開始文字または終了文字になる文字集合を V とおきます．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 $V=\{1,2,\cdots,n\}$ と表します．有向辺の集合 A は単語の集合であって，$A=B_{11}\cup B_{12}\cup\cdots\cup B_{nn}$ と表されます．ここで，B_{ij} は文字 i で始まって，文字 j で終わる単語の集合です．これによって得られる有向グラフを図1(a)に示します．図1(a)において，二項点を結ぶ路で，有向辺を一度だけ使うオイラー路が一つのしりとりを表現します．しりとりを構成する部分グラフのように，始点と終点が異なり，それ自身がオイラー路となるグラフは，**準オイラーグラフ**と呼ばれます．よって，最長しりとり問題は次のようにモデル

しなの　ゆうじ，いぬい　のぶお
東京農工大学大学院共生科学技術研究部
〒184-8588 小金井市中町2-24-16



(a)　単語が枝に対応する場合



(b)　単語数が枝に対応する場合

図1　しりとりのモデル

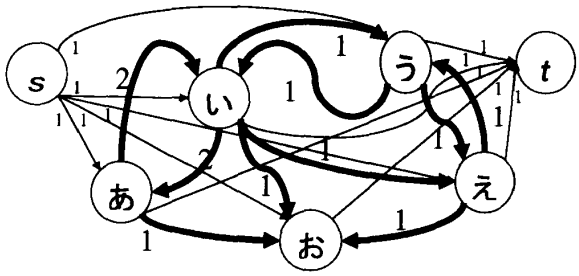
化されます．

最長しりとり問題のグラフ表現によるモデル
有向グラフ $G=(V, A)$ の準オイラー部分グラフ (Semi-Eulerian Subgraph) の中で有向辺の数が最大となるものを見つける問題．

しりとりの長さだけに着目すると，単語の違いは問題ではなく，二つの文字を結ぶ単語の数だけを扱えば十分です．このため，文字 i から文字 j への有向辺の集 B_{ij} に含まれる有向辺を一つの有向辺 a_{ij} で置き換えてできる有向辺の集合 $A'=\{a_{11}, a_{12}, \cdots, a_{nn}\}$ と，単語数 $f_{ij}=|B_{ij}|$ の集合 $F=\{f_{11}, f_{12}, \cdots, f_{nn}\}$ を用いたネットワークを考えます．ここで，f_{ij} は有向辺 a_{ij} の容量と考えます．このネットワークの例を図1(b)に示します．このネットワークにおいて，最長しりとり問題は次のようにモデル化されます．

最長しりとり問題のネットワーク表現によるモデル
ネットワーク $N=(G=(V, A'), F)$ において，任意の二頂点間を流れるフローの中で，各有向辺を流れるフロー量の総和を最大化する問題．

本稿では，このネットワーク表現に基づいた解法を説明します．まず，任意の二頂点間のフローを考える代わりに，図2に示すような補助ネットワークを作り，補助ネットワーク N' 上での $s-t$ フローを考えます．補助ネットワーク N' は，ネットワーク N に，スーパー・ソース s とスーパー・シンク t を付加したもの



s より出る，および t に入る有向辺の容量1，その他の有向辺の容量は辞書中の単語数で決定される．

図2　補助ネットワークによるグラフ表現

です．さらに，補助ネットワーク N' は，頂点 s から N のすべての頂点へ，および N のすべての頂点から頂点 t へ，容量1の有向辺を加えて構成します．補助ネットワーク N' 上で求まる頂点 s から頂点 t への最長しりとりの長さは，[本来の長さ+2]ですが，最終的に解が求まった後に，しりとりの長さを -2 すれば済むので，途中の議論では -2 を省略します．

この補助ネットワークにおいて一つのしりとりが構成されるためには，次の条件が満たされなければなりません．

フローとしての条件　頂点 s, t を除き，各頂点に入るフロー量と出るフロー量が等しい．頂点 s では出るフロー量が1，頂点 t では入るフロー量が1である．

解が連結グラフとなるための条件　フロー量が正の有向辺により誘導される部分グラフが連結である．

フローとしての条件の記述は容易ですが，連結グラフとなるための条件は，頂点数に対して指数本の制約式が必要となります．そこで，フローとしての条件だけを制約とする緩和問題から開始する分枝限定法を構成します．

3.　分枝限定法による解法

しりとり中の開始文字 i，終了文字 j の単語の出現回数を示す変数 x_{ij} を用いて，まず，フローとしての条件だけを考慮し，各有向辺に流れるフロー量の総和を最大化する緩和問題 ($\mathbf{RP}_0$) を考えます．

($\mathbf{RP}_0$)　最大化　$z_0=\sum_{i\in V\cup\{s\}, j\in V\cup\{t\}} x_{ij}$

条件　$\sum_{j\in V} x_{sj}=1,$

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

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

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

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

$$0 \le x_{jt} \le 1,\ \forall j \in V.$$

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ます．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り，各 f_{ij} が整数なので，問題（$\mathbf{RP}_0$）の最適基底解 x_{ij} は整数解として求まります．よって，仮に，（$\mathbf{RP}_0$）の解 $x_{ij}>0$ の有向辺により誘導される部分グラフが連結である場合には，最長しりとり問題は解けたことになります．

そこで，（$\mathbf{RP}_0$）の解 $x_{ij}>0$ の有向辺で誘導される部分グラフが，複数の連結成分に分かれた場合について考えます．（$\mathbf{RP}_0$）の制約条件より，頂点 s と頂点 t の両方を含む連結成分が必ず存在します．いま，s, t を含む連結成分の頂点集合を V_0^* とし，最長しりとり問題の許容解集合を二つに分割（**分枝操作**）する次の二つの子問題を定義します．

子問題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一つはある最長しりとり問題．

子問題2　頂点 $i \in V_0^*$ から頂点 $j \in V \setminus V_0^*$ への単語の遷移は一つもない最長しりとり問題．

この二つの子問題の最長しりとりのうち，長い方が元の最長しりとり問題の最適解となります．

子問題2は，（$\mathbf{RP}_0$）の頂点集合 $V \cup \{s, t\}$ を V_0^* に制限した問題を解くことに相当します．よって，新たな問題を解くことなく，（$\mathbf{RP}_0$）の解から，$x_{ij}, \forall i \in V_0^*, \forall j \in V_0^*$ を取り出せばしりとりを構成できます．このときの目的関数値 z_0' は，

$$z_0' = \sum_{i \in V_0^*,\, j \in V_0^*} x_{ij}$$

として求まります．z_0' は，許容解の目的関数値なので，最適値の下界値となります．

子問題1に関しては，「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一つはある」という条件

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

を（$\mathbf{RP}_0$）に加えた（$\mathbf{RP}_1$）を考えます（例として，図3を参照）．（$\mathbf{RP}_1$）を解き，同様の分枝操作を行います．以上を再帰的に繰り返し，アルゴリズムを構成します．$k(\ge 1)$ 回目の再帰で解く問題（$\mathbf{RP}_k$）は，次のようになります．

（$\mathbf{RP}_k$）　最大化　$z_k = \sum_{i \in V \cup \{s\},\, j \in V \cup \{t\}} x_{ij}$

条件　$\sum_{j \in V} x_{sj} = 1,$

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

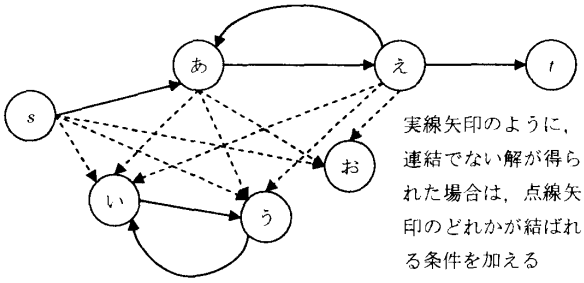


図3　分枝操作による追加条件の例

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

$$\sum_{\substack{j \in V_l^* \\ j \in V \setminus V_l^*}} x_{ij} \ge 1,\ l = 0, \cdots, k-1,$$

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

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

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

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

ここで，V_l^* は，問題（$\mathbf{RP}_{k-1}$）を解いて得られた最適解における，頂点 s, t を含む連結成分の頂点集合です．

（$\mathbf{RP}_k$）の整数条件を取り除くと，基底解の整数性は保証されませんので，（$\mathbf{RP}_k$）を整数計画問題として解く必要があります．このように，ここで提案している分枝限定法では，整数計画問題を緩和問題とし，親問題にさらに制約式を加えて解いています．分枝限定法によるアルゴリズムを次に示します．

Algorithm Making Longest Shiritori
begin
　k : $=0$;
　z^* : $=0$; {z^*：暫定値}
　{$\boldsymbol{x}^*$ へ暫定解を保存する}
　while true **do**
　begin
　　（$\mathbf{RP}_k$）を解く；
　　if（$\mathbf{RP}_k$）に実行可能解がない **then**
　　　goto 1 ;
　　if（$\mathbf{RP}_k$）の解で $x_{ij}>0$ の有向辺で
　　　構成されるグラフは連結グラフ **then**
　　　begin
　　　　if $z^* < z_k$ **then**
　　　　begin
　　　　　z^* : $=z_k$;
　　　　　z_k を導いた解を $\boldsymbol{x}^*$ へ保存する；
　　　　end ;
　　　　goto 1 ;

```
    end
  else
    begin
      if z_k < z* then
        goto 1;
      if z* < z'_k then
      begin
        z* : = z'_k;
        z'_k を導いた解を x* へ保存する;
      end;
      V*_k を抽出し，新しい制約条件を
      追加して，(RP_{k+1}) を作成する;
      k := k+1;
    end
  end
1: x* よりしりとりを構成する
  {しりとり長は，z*−2}
end;
```

ネットワーク表現により求まるフロー量 x_{ij} の値は，グラフ表現によるモデルで考えると，最長しりとりを構成するために使う文字 i から文字 j への有向辺の数を示します．そこで，この x_{ij} の値によりグラフを構成すれば，準オイラーグラフが構成できます．そこで，まず，構成された準オイラーグラフから，閉路を含まない s-t 路が抽出できるまで，閉路を取り除き記録しておきます．次に，閉路を含まない s-t 路を見つけたら，記録しておいた閉路を適当に加えてゆくことで，最長しりとりが構成できます．

4. 文字数最大しりとり問題

次に，文字数最大しりとり問題の解法について説明します．各単語の文字数が必要となるので，文字 i で始まり文字 j で終わる単語に適当に順序付けします．文字 i で始まり文字 j で終わる k 番目の単語の文字数を定数 c_{ij}^k とし，その単語をしりとりとして使うかどうかを示す 0-1 変数 x_{ij}^k を用いれば，最長しりとり問題の場合と同様の解法が構成できます．このとき，最初の緩和問題（$\mathbf{RP}_0$）は，次のようになります．

（$\mathbf{RP}_0$） 最大化 $z_0=\sum_{\substack{i\in V\cup\{s\},\, j\in V\cup\{t\}\\ 1\le k\le f_{ij}}} c_{ij}^k x_{ij}^k$

条件 $\sum_{\substack{j\in V\\ 1\le k\le f_{sj}}} x_{sj}^k=1,$

$\sum_{\substack{j\in V\\ 1\le k\le f_{ij}}} x_{ij}^k-\sum_{\substack{j\in V\\ 1\le k\le f_{ji}}} x_{ji}^k=0,\ \forall i\in V,$

$\sum_{\substack{j\in V\\ 1\le k\le f_{jt}}} x_{jt}^k=1,$

$0\le x_{ij}^k\le 1,\ \forall i\in V,\forall j\in V,\ 1\le k\le f_{ij},$

$0\le x_{sj}^k\le 1,\ \forall j\in V,1\le k\le f_{sj},$

$0\le x_{jt}^k\le 1,\ \forall j\in V,1\le k\le f_{jt}.$

この緩和問題（$\mathbf{RP}_0$）の変数の数は，辞書にある総単語数と等しくなります．もし，辞書に 10 万語～20 万語くらいの単語があったとしても，現在の線形計画問題ソルバであれば，（$\mathbf{RP}_0$）は比較的短時間に解けると思われます．最長しりとり問題と同様に，最初の緩和問題（$\mathbf{RP}_0$）では整数基底解が得られますが，以降の緩和問題においては，最長しりとり問題と同様の制約式が追加されるため，x_{ij} に対して 0-1 条件を追加した整数計画問題を解く必要があります．

ここでは，変数の数を減らすために，文字 i から始まり文字 j で終わる単語長 l の単語の数を f_{ij}^l，文字 i から始まり文字 j で終わる最長の単語の単語長を L_{ij} とし，辞書に対する前処理としてこれらの値を求めることにします．すると，文字 i から始まり文字 j で終わる単語長 l の単語をしりとりで使う回数を x_{ij}^l として，最長しりとり問題と同様の解法が構成できます．このとき，最初の緩和問題（$\mathbf{RP}_0$）は，次のようになります．

（$\mathbf{RP}_0$） 最大化 $z_0=\sum_{\substack{i\in V\cup\{s\},\, j\in V\cup\{t\}\\ 1\le l\le L_{ij}}} l x_{ij}^l$

条件 $\sum_{\substack{j\in V\\ 1\le l\le L_{sj}}} x_{sj}^l=1,$

$\sum_{\substack{j\in V\\ 1\le l\le L_{ij}}} x_{ij}^l-\sum_{\substack{j\in V\\ 1\le l\le L_{ji}}} x_{ji}^l=0,\ \forall i\in V,$

$\sum_{\substack{j\in V\\ 1\le l\le L_{jt}}} x_{jt}^l=1,$

$0\le x_{ij}^l\le f_{ij}^l,\ \forall i\in V,\forall j\in V,\ 1\le l\le L_{ij},$

$0\le x_{sj}^l\le 1,\ \forall j\in V,\ 1\le l\le L_{sj}$

$0\le x_{jt}^l\le 1,\ \forall j\in V,\ 1\le l\le L_{jt}.$

この場合もやはり，最長しりとり問題と同様に，最初の緩和問題（$\mathbf{RP}_0$）では整数基底解が得られます．しかし，それ以降の緩和問題においては，最長しりとり問題と同様の制約式が追加されるため，x_{ij}^l に整数変数としての条件を追加した整数計画問題を解く必要があります．

5. 数値実験結果

解法をC言語により実装しました．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として，GNU GLPK (ver. 4.2) を使用しました．実験に使用したPCは，Xeon 2.8 GHz, Dual Processer, 2 GB Memoryです．

テレビ番組で紹介された際には，最新の辞書を使いたいということで，広辞苑第5版を使用しました．その際，冒頭に述べたように名詞のみを抽出したり同じ音の単語を一つにするなどの条件を満たす辞書を作成しました．これらの処理を施した結果，辞書の総単語数は154,150語で，構成された最長しりとりの単語数は75,775語でした．このとき，緩和問題 ($\mathbf{RP}_0$) を解いて，連結なグラフが得られました．つまり，分枝限定法は最初の問題の緩和問題（線形計画問題）を解いただけで終了したわけです．辞書から有向グラフを構成すると，頂点数69に対して，有向辺数は154,150という頂点間に多重有向辺のあるグラフとなります．よって，可能な限り有向辺を使う準オイラー部分グラフを求めていることを考えると，直感的には部分グラフは連結グラフとして得られるように思えます．しかし，連結グラフが構成されるかどうかは実行してみないと確認できないことなので，分枝限定法が最も効果的に機能した良い例と考えられます．

本節では，広辞苑第4版を使用して，最長しりとり問題，文字数最大しりとり問題に対するいくつかの実験結果を示します．実験で使用するデータは，テレビ番組のものとは異なり広辞苑のCD-ROMから機械的に取り出して処理しました．そのため，基本的には名詞ですが，品詞が指定されていない単語や，1文字の単語も含まれています．また，同じ音の単語が複数存在し，拗音などの処理も異なります．その結果，辞書の総単語数は192,687語で，補助ネットワーク N' の頂点数は，s, t 頂点を除いて，68頂点です．補助ネットワーク N' の有向辺の数（定式化時の変数の数）は，表1のように扱う問題とモデル化により異なります．表1に示された変数の数は，s から各頂点への有向辺と各頂点から t への有向辺に対応する変数の数 (68+68=136) を含んでいます．これは実際に整数計画問題を解く際の変数の数を示すためです．

5.1 最長しりとり問題

数値実験としては，可能な開始文字をすべて指定して，最長しりとり問題を解きました．その際に，しりとりが終了する文字は，必ずしも「ん」とは限らないので終了文字も指定して，すべての組合せについて解いています．

表1 各問題とモデルにおける有向辺数（変数の数）

問題	最長しりとり問題	文字数最大しりとり問題	
定式化での変数の種類	整数変数	0-1変数	整数変数
変数の数	4,314	192,823	23,206

開始文字を変えた場合に，最も長い最長しりとりが作られた開始文字は，「あ」，「ほ」，「ひ」，「は」，「ぺ」，「へ」でした．つまり，これらの文字で始まる単語から開始したしりとりが最長で，その長さは86,796でした．また，そのときの終了文字は，すべて「ん」でした．最長しりとりが，最も短くなるのは「ん」から始めた場合で（この実験は，機械的に単語を抽出したので，品詞の指定されていない「ん」「んとす」が辞書データに含まれていました．しりとりとしては適当ではありませんが，ご容赦下さい），その長さは86,787でした．また，この場合の終了文字は，先の開始文字と同じ「あ」，「ほ」，「ひ」，「は」，「ぺ」，「へ」でした．しりとりを開始する文字を変えたとしても，最長しりとりの長さには9しか違いがありません．このことから，グラフ表現のモデルで考えると，いずれの開始文字から構成された最適解においても共通する，極大なオイラー閉路が構成されていると思われます．

分枝限定法によるアルゴリズムの実行時間は，平均1.32秒で，分散に関しては誤差の範囲でした．分散が小さいのは，いずれも最初の緩和問題 ($\mathbf{RP}_0$) における線形計画問題を解くだけで，分枝することなく最適解を得ているためです．さらに詳細な数値実験結果に関しては，文献[3]をご参照下さい．

5.2 文字数最大しりとり問題

文字数最大しりとり問題は，緩和問題の変数の数が最長しりとり問題と比較して，かなり大きくなります．特に，0-1変数による定式化では192,823変数の問題ですが，それでも，3,201秒（試行は1回）で最適解が得られました．こちらも，最初の緩和問題（線形計画問題）を解くことで連結グラフが得られ，ほぼ線形計画問題を一つ解いた時間です．参考までに，広辞苑に掲載されている単語をひらがな読みにした際，1単

語で最も長い単語長は29，得られたしりとり長は83,901，そのしりとりで構成される文字数は436,106でした．これを，整数計画問題として定式化した場合は23,206変数となるので，最適値は当然同じですが，実行時間は約56秒でした．

辞書から構成されるグラフ上で，各頂点間の有向辺の数が半分ずつ減るように単語数を減らし，再帰的に辞書群を構成しました．その辞書群に対して，数値実験を行うと，最初の緩和問題（$\mathbf{RP}_0$）では解が連結グラフとして求まらない場合もありました．しかし，緩和問題である整数計画問題（$\mathbf{RP}_k$）を繰り返し解いた回数は，行った数値実験の範囲では，最大でも217回でした．GLPKの整数計画問題ソルバは，双対単体法をベースにした単純な分枝限定法による実装です．本稿で示した分枝限定法の実装は，最初の緩和問題（$\mathbf{RP}_0$）では整数基底解が得られるので，それ以降の制約条件が加えられた子問題では，一般的に行われるように既に得られている基底解から解き直しました．このとき，実際には，最初の緩和問題（$\mathbf{RP}_0$）以外の整数計画問題による緩和問題（$\mathbf{RP}_k$）は極めて短時間で最適解が得られています．

6. おわりに

本稿で示したアルゴリズムでは，整数計画問題を緩和問題としています．本誌の過去のコラムに掲載された藤江氏による解説[4]にもあるように，ここ数年の整数計画問題ソルバの性能向上は目覚しいです．仮に，ある種の辞書において，緩和問題としての整数計画問題を何度も解く必要があったとしても，強力なソルバを利用することで現実的な時間内で解ける可能性は高いと思われます．解法の性質上うまく機能する場合もありますし，整数計画問題ソルバが強力になっていることを考えると，緩和問題として整数計画問題を利用することも，試みてみる価値があると思われます．

最後に，本稿の内容の一部に関しては他分野の方々向けのプレゼンテーション[5]や記事[6]もありますのでご覧いただき，もしよろしかったら研究普及活動の題材としてご利用いただければと思います．

謝辞 原稿執筆に当たり，兵庫県立大学の藤江哲也氏に原稿を読んでいただき，有意義なコメントをいただいたことに感謝いたします．

参考文献

[1] 茨木俊秀："組合せ最適化—分枝限定法を中心として—"，産業図書，1983.

[2] 今野浩，鈴木久敏編："整数計画法と組合せ最適化"，日科技連出版社，1982.

[3] 乾伸雄，品野勇治，鴻池祐輔，小谷善行："最長しりとり問題の解法"，「情報処理学会論文誌：数理モデル化と応用」(TOM)，掲載予定.

[4] 藤江哲也："整数計画問題に対する分枝カット法とカットの理論"，オペレーションズ・リサーチ，Vol. 48, No. 12, 935-940, 2003.

[5] URI : http://al.cs.tuat.ac.jp/~yshinano/shiritori/

[6] 品野勇治："最短・最長を探す：組合せ最適化，最も長いしりとりの作り方"，数学セミナー8月号，日本評論社，Vol. 43, No. 8, 36-41, 2004.