卒業論文

ポケモンつなげるもん♪
―最長しりとり問題を整数計画法で解く―

愛知教育大学 初等教育教員養成課程 数学選修
s2070268 佐藤　一生

1 はじめに

しりとりは，日本人なら誰でもやった事のあるゲームだと思う．特別な道具を必要とせず，頭の中でできるこのゲームは，何か手持ち無沙汰になったときに最適なゲームである．「る」で終わる言葉で常に返してみたり，何か制限をつけてやってみたり，遊び方は様々だが，遊んでいるとふと疑問に思うことがある．果たして，しりとりはどのくらい続けられるものなのかという問いである．そこで，今回，卒業論文として「最長しりとり問題」という問題を取り上げ，数学的にしりとりを出来るだけ長く続けたときにどのくらいの長さになるのか考察してみた．

本論文では，しりとりを有向グラフやネットワークとして捉え，線形計画問題として解く．そこで，まず第 2 章でしりとりの大まかなルールと最長しりとり問題を定義したあと，第 3 章でグラフ理論の重要な定理を証明する．そして，第 4 章では最適化問題を解くために有効な緩和法と限定分枝法の原理について説明する．第 5 章では，整数計画問題を解くために重要な完全ユニモジュラー行列と線形計画問題を解く際に有効な単体法の原理について説明する．以上の諸定理を揃えたあと，第 6 章で最長しりとり問題を整数計画問題として定式化し，最長しりとり問題を解くためのアルゴリズム及び最適解を得るしりとり列をつくるアルゴリズムを提示する．第 7 章では実際に小規模な辞書で最長しりとり問題を解き，その最適値や最適解について考察する．

2 最長しりとり問題とは

2.1 しりとりのルール

ゲームとしてのしりとりは，一般には回答順序の決められた複数人で行われ，回答が求められた人は前の人の言った単語の末尾の文字で始まる単語を回答する．一般的に行われているしりとりには様々なルールがあるが，本論文では以下のルールを採用する．

(1) 名詞だけを使用する．
(2) 1 度使用した単語は再び使用できない．
(3) どの文字から始めてもよい．
(4) 「ん」で終わったり，次に続く単語が無くなったら終了する．
(5) 「ー (長音符)」で終わる単語は長音符の前の文字を末尾の文字とする．
(6) 小さい「ぁぃぅぇぉ」や「ゃゅょ」で終わる単語は末尾の文字をそれぞれ「あいうえお」「やゆよ」とする．
(7) 「゛ (濁点)」や「゜ (半濁点)」の付く文字はそれらが付かない文字と区別しない．

2.2 最長しりとり問題

はじめに，最長しりとり問題を定義する．各単語の最初の文字を**開始文字**，最後の文字を**終了文字**，また単語の集合を**辞書**と呼ぶ．この論文で考える最長しりとり問題は，以下のような問題である．

問題 2.1. (**最長しりとり問題**)
与えられた辞書において，しりとりとなる最も単語数の多い単語列を作成せよ．ただし，しりとりの長さは，しりとりに現れる単語数として定義する．

3 グラフとネットワーク

3.1 グラフやネットワークに関する用語の定義

本論文では，最長しりとり問題を有向グラフやネットワークフローの問題に読みかえて考えていく．そこで，グラフやネットワークに関するいくつかの用語を定義する．

定義 3.1. (**無向グラフ，有向グラフ**)
無向グラフは，**頂点**と呼ばれる要素の有限集合 V，**辺**と呼ばれる要素の有限集合 E，関数 $\Psi : E \to \{X \subseteq V \mid \#X = 2\}$ の 3 つの組 (V, E, Ψ) で定義される．一方，**有向グラフ**は 2 つの有限集合 V, E と関数 $\Psi : E \to V \times V$ の 3 つの組 (V, E, Ψ) で定義される．特に，有向グラフにおいて E の要素は**有向辺**と呼ばれる．そして，無向グラフと有向グラフを総称して**グラフ**と呼ぶ．

通常，グラフ G の頂点集合や辺集合を $V(G), E(G)$ と表し，$G = (V(G), E(G))$ とも書く．グラフ $G = (V(G), E(G))$ に対して，$V(H) \subset V(G)$ かつ $E(H) \subset E(G)$ となるグラフ $H = (V(H), E(H))$ を G の**部分グラフ**と呼ぶ．また，有向グラフ D について，任意の辺 $e = (u, v)$ を $e = \{u, v\}$ と置き換えた無向グラフ G を D の**基礎グラフ**という．

グラフ G の 2 頂点 $u, v \in V(G)$ について，$\Psi(e) = \{u, v\}$ または (u, v) となる $e \in E(G)$ が存在するとき，u と v は**隣接している**という．特に有向グラフにおいて $\Psi(e) = (u, v)$ となる e は u を**出て** v に**入る**といい，$e = e_{uv}$ と表す場合もある．このとき，$s(e) = u, t(e) = v$ と表すことにする．従って，$\Psi(e) = (s(e), t(e))$ となる．また，$\Psi(e) = \{u, v\}$ または (u, v) について $u = v$ のとき，e を**ループ**といい，$\Psi(e) = \Psi(e')$ となる e と e' を合わせて**多重辺**という．グラフ G がループや多重辺を含まないとき，G は**単純グラフ**と呼ばれる．

次に，G の頂点と辺の有限個の交互系列

$$W = (v_1, e_1, v_2, \ldots, v_k, e_k, v_{k+1}), \quad v_i \in V(G), e_i \in E(G)$$

を考える．

定義 3.2. (**歩道，小道，閉路**)

系列 W について，すべての $i = 1, \dots, k$ で $\Psi(e_i) = (v_i, v_{i+1})$ (無向グラフのときは $\Psi(e_i) = \{v_i, v_{i+1}\}$) が成り立つとき，$W$ は G の**歩道**という．このとき，どの $1 \leq i < j \leq k$ でも $e_i \neq e_j$ ならば，W を G の**小道**といい，さらに $v_1 = v_{k+1}$ ならば W を G の**閉路**という．

定義 3.3. (**オイラー (閉) 路，(準) オイラーグラフ**)

小道 W に $E(G)$ のすべての元があらわれるとき，W を G の**オイラー路**といい，G を**準オイラーグラフ**という．さらに W が閉路になっているとき，W を G の**オイラー閉路**といい，G を**オイラーグラフ**という．

G の部分グラフ H が $E(H)$ のすべての元があらわれる小道を持つとき，H を**準オイラー部分グラフ**，さらにそれが閉路になっているとき H を**オイラー部分グラフ**という．また，オイラー路 (またはオイラー閉路) P について，P の最初の頂点 v_1 を**始点**，最後の頂点 v_{k+1} を**終点**という．

小道 W について，すべての $1 \leq i < j \leq k+1$ で $v_i \neq v_j$ となるグラフ

$$P = (\{v_1, \dots, v_{k+1}\}, \{e_1, \dots, e_k\})$$

を v_1 から v_{k+1} への**道**という．

定義 3.4. (**連結，連結成分**)

無向グラフ G の任意の 2 頂点 $v, w \in V(G)$ について v から w への道が存在するとき，G は**連結**であるといい，G は**連結グラフ**と呼ばれる．また，G の極大な連結部分グラフを G の**連結成分**という．有向グラフ D についてはその基礎グラフ G が連結になっているとき，D は**連結**であるという．

定義 3.5. (**次数**)

無向グラフ G の頂点 $v \in V(G)$ について，**次数** $d_G(v)$ を

$$d_G(v) = \#\{e \in E(G) \mid v \in \Psi(e)\}$$

と定義する．また，有向グラフ D の頂点 $v \in V(D)$ について，**出次数** $d_D^+(v)$ と**入次数** $d_D^-(v)$ を

$$d_D^+(v) = \#\{e \in E(D) \mid s(e) = v\}$$
$$d_D^-(v) = \#\{e \in E(D) \mid t(e) = v\}$$

と定義する．特に，$d(v) = 0$ (有向グラフの場合は $d^+(v) = d^-(v) = 0$) のとき，v を**孤立点**と呼ぶ．

定義 3.5. より，有向グラフ D の基礎グラフ G の任意の頂点 v について，その次数 $d_G(v)$ は

$$d_G(v) = d_D^+(v) + d_D^-(v)$$

となることがわかる．

定義 3.6. (**ネットワーク，フロー**)

ネットワークは，有向グラフ D，関数 $c: E(D) \to \mathbf{R}_+$，**入口** $s \in V(D)$，**出口** $t \in V(D)$ の 4 つの組 (D, c, s, t) で定義される．このとき，$e \in E(D)$ について，$c(e)$ を e の**容量**という．また，関数 $f: E(D) \to \mathbf{R}_+$ が

(1) $\forall v \in V(D) \setminus \{s, t\}$ に対して

$$\sum_{u \in V(D)} f(e_{vu}) - \sum_{u \in V(D)} f(e_{uv}) = 0$$

(2) $\forall v, u \in V(D)$ に対して

$$0 \leq f(e_{uv}) \leq c(e_{uv})$$

を満たすとき，f を**フロー**といい，$f(e)$ を e を流れる**フロー量**と呼ぶ.

3.2 オイラーグラフ

この節では，オイラーグラフに関する定理を証明する．初めに，後の定理の証明で使う補題を証明する．

補題 3.7.

無向グラフ G の任意の頂点 v について，$d(v) \geq 2$ のとき，G には閉路がある．

Proof. G にループや多重辺があるときはそこで既に閉路となり明らかなので，G が単純グラフであるときを考える．G の任意の頂点を v_0 とし，v_0 と隣接する任意の頂点 v_1 を選ぶ．同様に頂点 v_i $(i \geq 1)$ に隣接している v_{i-1} 以外の任意の頂点 v_{i+1} を選び，歩道を作っていく．このとき，$d(v) \geq 2$ より，任意の頂点には必ず隣接する頂点があることが保証される．すると，G の頂点は有限なので，この操作を繰り返すとやがて同じ頂点が 2 回選ばれることになる．そこでその頂点を始点として同じ歩道を辿り，次にその頂点が出てきた時その歩道は閉路となっている． □

補題 3.7. の証明を考えると，有向グラフについての以下の系も容易に証明できる．

系 3.8.

有向グラフ D の任意の頂点 v について，$d^+(v) \geq 1$ のとき，D には閉路がある．

Proof. D にループや多重辺があるときはそこで既に閉路となり明らかなので，D が単純グラフであるときを考える．D の任意の頂点を v_0 とし，$s(e_0) = v_0$ となる辺 e_0 を選び，$v_1 = t(e_0)$ とおく．同様に頂点 v_i $(i \geq 1)$ に対して，$s(e_i) = v_i$ となる辺 e_i を選び，$v_{i+1} = t(e_i)$ とおき，歩道を作っていく．このとき，D は単純グラフと仮定したので，$v_{i+1} = v_{i-1}$ となることはない．また，$d^+(v) \geq 1$ より，$s(e) = v$ となる辺 e は必ずあり，任意の頂点には必ず隣接する頂点があることが保証される．すると，D の頂点は有限なので，この操作を繰り返すとやがて同じ頂点が 2 回選ばれ

ることになる．そこでその頂点を始点として同じ歩道を辿り，次にその頂点が出てきた時その歩道は閉路となっている． $\square$

命題 3.9.

連結グラフ G がオイラーグラフであるための必要十分条件は，グラフ G の任意の頂点 v について，次数 $d(v)$ が偶数であることである．

Proof. まず十分性を証明する．G のオイラー閉路を P とする．定義 3.5. より，P がある頂点を 1 回通過するごとにその頂点の次数には 2 が加えられる．P は G のオイラー閉路なので，G のすべての辺はそれぞれ 1 回ずつ P に含まれる．よって，P が通過する回数の 2 倍がその頂点の次数となる．よって十分性は証明された．

次に必要性を $\#E(G)$ に関する帰納法で示す．G の任意の頂点 v について，$d(v)$ が偶数であるとする．G が連結グラフであることから G の各頂点は孤立していないので，$d(v) \geq 2$ といえる．$\#E(G) = 1$ のとき，$d(v) \geq 2$ より G はループのみのグラフとなり，明らかにオイラーグラフになる．$\#E(G) \geq 2$ のとき，$\#E(G) > \#E(G')$ かつ $\forall v' \in V(G')$ について $d(v')$ が偶数になっているグラフ G' はオイラーグラフになっていると仮定する．$d(v) \geq 2$ であることから，補題 3.7. より G には閉路が存在する．その閉路を C とすると，$E(G) = E(C)$ のときは明らかに G はオイラーグラフなので，そうでない時を考える．グラフ H を $H = (V(G), E(G) \setminus E(C))$ とする．H の孤立点を除く任意の頂点 v_h について

$$d_H(v_h) = d_G(v_h) - \#\{e \in E(C) \mid v_h \in \Psi(e)\}$$

が成り立つ．仮定より $d_G(v_h)$ は偶数で，C が閉路であることから $\#\{e \in E(C) \mid v_h \in \Psi(e)\}$ も偶数になる．従って，$d_H(v_h)$ も偶数となる．さらに，$\#E(H) < \#E(G)$ より，帰納法の仮定から H の各連結成分にはオイラー閉路が存在する．G の連結性から H の各連結成分は C と少なくとも 1 つの頂点を共有しているので，C 上の任意の 1 頂点を始点として，C の辺を辿り，H の孤立点でない頂点が現れるとその頂点を含む H の連結成分のオイラー閉路を辿り，再びその頂点に戻ってくる．そしてそこから再び C の辺を辿る．この操作を H のすべての連結成分について繰り返し，始点へ戻る．この閉路を P とする．また，G の連結性から，H の孤立点は C に含まれている．以上から P はオイラー閉路となり，G はオイラーグラフとなる (図 1 参照)．よって，必要性は証明された． $\square$

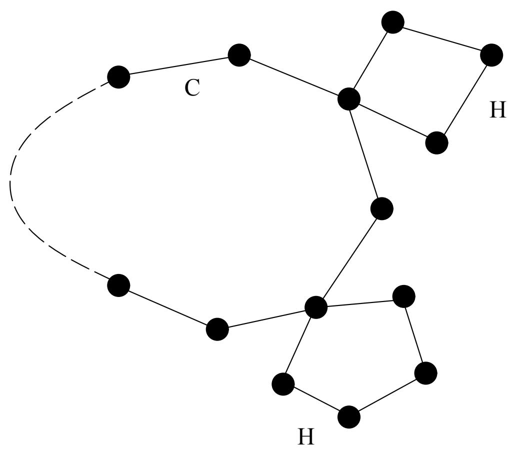

図 1　命題 3.9.

命題 3.9. の証明を考えると，以下の系も容易に証明できる．

系 3.10.

連結有向グラフ D が有向オイラーグラフであるための必要十分条件は，D の任意の頂点 v について $d^+(v) = d^-(v)$ が成り立つことである．

Proof. まず十分性を証明する．D のオイラー路を P とする．命題 3.9. の十分性の証明と同様に考えると，P がある頂点を 1 回通過するごとに入次数と出次数にそれぞれ 1 が加えられるので，$d^+(v) = d^-(v)$ が成り立つ．よって，十分性は証明された．

次に必要性を $\#E(D)$ に関する帰納法で示す．D の任意の頂点 v について，$d^+(v) = d^-(v)$ が成り立つとする．G が連結グラフであることから D の各頂点は孤立していないので，$d^+(v) \geq 1$ かつ $d^-(v) \geq 1$ といえる．$\#E(D) = 1$ のとき，$d^+(v) \geq 1$ かつ $d^-(v) \geq 1$ より D はループのみのグラフとなり，明らかにオイラーグラフになる．$\#E(D) \geq 2$ のとき，$\#E(D) > \#E(D')$ かつ $\forall v' \in V(D')$ について $d^+(v) \geq 1$ かつ $d^-(v) \geq 1$ が成り立っているグラフ D' はオイラーグラフになっていると仮定する．$d^+(v) \geq 1$ かつ $d^-(v) \geq 1$ であることから，系 3.8. より G には閉路が存在する．その閉路を C とすると，$E(D) = E(C)$ のときは明らかに D はオイラーグラフなので，そうでない時を考える．グラフ H を $H = (V(D), E(D) \setminus E(C))$ とする．H の孤立点を除く任意の頂点 v_h について

$$d^+_H(v_h) = d^+_G(v_h) - \#\{e \in E(C) \mid s(e) = v_h\}$$
$$d^-_H(v_h) = d^-_G(v_h) - \#\{e \in E(C) \mid t(e) = v_h\}$$

が成り立つ．仮定より $d^+_G(v_h) = d^-_G(v_h)$ が成り立ち，C が閉路であることから

$$\#\{e \in E(C) \mid s(e) = v_h\} = \#\{e \in E(C) \mid t(e) = v_h\}$$

も成り立つ．従って，$d^+_H(v_h) = d^-_H(v_h)$ も成り立つ．さらに，$\#E(H) < \#E(D)$ より，帰納法の

仮定から H の各連結成分にはオイラー閉路が存在する．D の連結性から H の各連結成分は C と少なくとも 1 つの頂点を共有しているので，C 上の任意の 1 頂点を始点として，C の辺を辿り，H の孤立点でない頂点が現れるとその頂点を含む H の連結成分のオイラー閉路を辿り，再びその頂点に戻ってくる．そしてそこから再び C の辺を辿る．この操作を H のすべての連結成分について繰り返し，始点へ戻る．この閉路を P とする．また，G の連結性から，H の孤立点は C に含まれている．以上から P はオイラー閉路となり，D はオイラーグラフとなる．よって，必要性は証明された． □

命題 3.11.

連結有向グラフ D が有向準オイラーグラフであるための必要十分条件は，D の任意の頂点 $v \in V(D)$ について，2 頂点 s, t を除き $d^+(v) = d^-(v)$ が成り立ち，2 頂点 s, t については

$$d^+(s) - d^-(s) = d^-(t) - d^+(t) = 1$$

が成り立つことである．

Proof. まず十分性を証明する．D が有向準オイラーグラフであるとする．D のオイラー路の始点を s，終点を t とする．t から出て s に入る辺 e_{ts} を加えたグラフを D^* とすると，D^* は有向オイラーグラフとなる．従って，系 3.10. より D^* の任意の頂点 $v_{D^*} \in V(D^*)$ について $d^+(v_{D^*}) = d^-(v_{D^*})$ が成り立つ．よって，辺 e_{ts} を加える前のグラフ D でも，任意の頂点 $v \in V(D)$ について，2 頂点 s, t を除き $d^+(v) = d^-(v)$ が成り立ち，2 頂点 s, t については

$$d^+(s) - d^-(s) = d^-(t) - d^+(t) = 1$$

が成り立つ．よって，十分性は証明された．

次に必要性を証明する．D の任意の頂点 $v \in V(D)$ について，2 頂点 s, t を除き $d^+(v) = d^-(v)$ が成り立ち，2 頂点 s, t については

$$d^+(s) - d^-(s) = d^-(t) - d^+(t) = 1$$

が成り立つとする．t から出て s に入る辺 e_{ts} を加えたグラフを D^* とすると，D^* の任意の頂点 $v_{D^*} \in V(D^*)$ について，$d^+(v_{D^*}) = d^-(v_{D^*})$ が成り立つ．よって系 3.10. より，D^* は有向オイラーグラフとなる．従って，辺 e_{ts} を加える前のグラフ D は有向準オイラーグラフとなる．よって，必要性は証明された． □

3.3 有向準オイラーグラフとネットワーク

この節では，有向準オイラーグラフとネットワークの関係性について述べる．有向グラフ D について，同じ組み合わせの頂点を結ぶ有向辺の集合 $B_{uv} = \{e \in E(D) \mid \Psi(e) = (u, v)\}$ を改めて 1 つの有向辺 a_{uv} に置き換え，有向グラフ D' を作る．さらに，a_{uv} に容量 $c(a_{uv})$ を $c(a_{uv}) := \#B_{uv}$

として与え，$V(D')$ に入口 s と出口 t を加えたネットワーク $N = (D', c, s, t)$ を考える．このとき，以下の定理が成り立つ．

定理 3.12. (**有向準オイラーグラフとネットワーク**)
D が有向準オイラー部分グラフ H を持つとき，関数

$$f(a_{uv}) := \#\{e \in E(H) \mid \Psi(e) = (u, v)\}$$

は N のフローになっている．即ち，関数 f は

(1) $\forall v \in V(D') \setminus \{s, t\}$ に対して

$$\sum_{u \in V(D')} f(a_{vu}) - \sum_{u \in V(D')} f(a_{uv}) = 0$$

(2) $\forall v, u \in V(D')$ に対して

$$0 \leq f(a_{uv}) \leq c(a_{uv})$$

を満たす．

Proof. まず (1) について示す．f の定義より，D' の s, t を除く各頂点 v について

$$\sum_{u \in V(D')} f(a_{vu}) = \#\{e \in E(H) \mid s(e) = v\} = d^+_H(v)$$
$$\sum_{u \in V(D')} f(a_{uv}) = \#\{e \in E(H) \mid t(e) = v\} = d^-_H(v)$$

が成り立つ．H が有向準オイラーグラフであることから，命題 3.11. より，$d^+_H(v) = d^-_H(v)$ が成り立つ．よって (1) が成り立つ．

次に (2) について示す．H が D の部分グラフであることから

$$\#\{e \in E(H) \mid \Psi(e) = (u, v)\} \leq \#\{e \in E(D) \mid \Psi(e) = (u, v)\}$$

が成り立つ．従って，$f(a_{uv}) \leq c(a_{uv})$ が成り立つ．また，f の定義より $0 \leq f(a_{uv})$ が成り立つ．よって (2) が成り立つ． □

4 緩和法と分枝限定法

4.1 緩和法の原理

緩和法の原理を述べる前に，いくつかの用語を定義する．

最適化問題とは，ある**目的関数** $\Phi : \mathbf{R}^n \to \mathbf{R}$ について，与えられた**制約条件** $x \in X \subset \mathbf{R}^n$ の範囲内で目的関数値 $\Phi(x)$ を最大化または最小化する $x_0 \in X$ を求める問題のことである．このと

き，Φ に代入される $x \in X$ を**実行可能解**，実行可能解を制限する領域 X を**実行可能領域**という．また，$\Phi(x)$ を最大化または最小化する $x^* \in X$ を**最適解**，その値 $\Phi(x^*)$ を**最適値**という．特に，目的関数が線型写像として表され、制約条件の集合が一次方程式や一次不等式によって定義されている場合には，その問題は**線形計画問題**という．さらに，実行可能解が整数解のみの場合，その問題は**整数計画問題**という．

次のような最適化問題 (P) を考える．

$$
\begin{array}{lll}
(P) & \text{最大化} & \Phi(x) \\
 & \text{条　件} & x \in X \ (\subseteq \mathbf{R}^n)
\end{array}
$$

この最適化問題を解くのが困難な場合，(P) の制約条件を緩めて，ある集合 $X_0 \supseteq X$ に対して以下のような**緩和問題** (P') を考える．

$$
\begin{array}{lll}
(P') & \text{最大化} & \Phi(x) \\
 & \text{条　件} & x \in X_0 \ (\supseteq X)
\end{array}
$$

この緩和問題 (P') が (P) に比べて容易に解けるとする．問題 (P) と緩和問題 (P') の間には明らかに次の関係が成り立つ．

定理 4.1. (**緩和法の原理**)
問題 (P) と緩和問題 (P') の間には次の関係が成り立つ．

(1)　(P') が実行可能解を持たないときは (P) も実行可能解を持たない．
(2)　(P') の最適解を $\bar{x}$ としたとき，$\bar{x} \in X$ なら $\bar{x}$ は (P) の最適解である．
(3)　(P') の最適解を $\bar{x}$ とし，(P) の最適解を x^* とすると，$f(\bar{x}) \geq f(x^*)$ である．

4.2　分枝限定法の原理

この節では，**分枝限定法**の原理について説明する．
最適化問題

$$
\begin{array}{lll}
(P_0) & \text{最大化} & \Phi(x) \\
 & \text{条　件} & x \in X_0
\end{array}
$$

について，この問題を直接的にうまく解く方法が見つかっていないとする．このとき，適当な条件を設定して (P_0) の実行可能領域 X_0 をいくつかの部分領域 $X_1, X_2, \ldots$ に分割し，その各々に対応する問題

$$
\begin{array}{lll}
(P_k) & \text{最大化} & \Phi(x) \\
 & \text{条　件} & x \in X_k
\end{array}
$$

を考える．この (P_k) を (P_0) の**子問題**という． この一群の子問題 (P_k) $(k=1,2,\dots)$ を解いて得られた解の中で，$\Phi(x)$ の値が最大のものを見つければ，問題 (P_0) の最適解が得られることは明らかである．子問題 (P_k) がまだ難しく直接解けないときは，X_k を再びいくつかの領域に分けて，(P_k) をさらにいくつかの問題に分解していく．この手順を繰り返し適用すれば，いつかは直接解ける問題に到達し，それらの解を統合すれば (P_k) の最適解が得られ，結局 (P_0) が解ける．

問題 (P_k) が直接解けない場合，その実行可能領域 X_k を

$$X_k = \bigcup_{l=p}^{q} X_l \tag{4.1}$$

となるいくつかの部分集合 $X_p,\dots,X_q$ に分割して，(P_k) をその一群の子問題

$$\begin{array}{lll} (P_l) & \text{最大化} & \Phi(x) \\ & \text{条　件} & x \in X_l \end{array}$$

に置き換えるのが**分枝操作**である．普通 $X_p,\dots,X_q$ は $l_1 \neq l_2$ ならば $X_{l_1} \cap X_{l_2} = \phi$ となるように設計する．このとき，各子問題 (P_l) の最適値を z_l^*，最適解を x_l^* とする．L を $\max\limits_{p \le l \le q} z_l^* = z_L^*$ を満たす添字とすると，(4.1) より (P_k) の最適値 z_k^* と最適解 x_k^* は

$$\begin{aligned} z_k^* &= z_L^* \\ x_k^* &= x_L^* \end{aligned}$$

となる．また，$X_l = \phi$ の場合には $z_l^* = -\infty$ とし，x_l^* は意味を持たないものとする．

一般に，(P_k) の実行可能領域は X_k は X_0 に制限を加えた $X_k = X_0 \cap B_k$ なる形で与えられている．このとき，$\bigcup\limits_{l=p}^{q} B_{k,l} \supseteq B_k$ なる $B_{k,p},\dots,B_{k,q}$ を用いて，X_l を

$$X_l = X_k \cap B_{k,l} = X_0 \cap (B_k \cap B_{k,l})$$

とすれば (4.1) を満たす分割が得られる．

問題 (P_k) を直接解くことは難しくても，その最適値 z_k^* の**上界値**が得られれば探索を制御する有効な情報となる．そのために，前節で定義した緩和問題

$$\begin{array}{lll} (P_k') & \text{最大化} & \Phi(x) \\ & \text{条　件} & x \in X_k' \ (\supseteq X_k) \end{array}$$

が大きな役割を果たす．今，問題 (P_0) の少なくとも 1 つの実行可能解 $\tilde{x}$ が得られているとする．$\tilde{x}$ に対応する目的関数値を $\tilde{z} = \Phi(\tilde{x})$ とすると，$\tilde{z}$ は (P_0) の最適値の**下界値**となる．従って，もし (P_k') の最適値 $\bar{z}_k$ が

$$\bar{z}_k \le \tilde{z}$$

を満たせば，定理 4.1. より問題 (P_k) のいかなる実行可能解も $\tilde{x}$ より良い目的関数値を与えないことが結論される．従って，この場合には (P_k) をこれ以上分枝することは無意味である．このような場合，(P_k) は**限定操作**によって見切られて**分枝停止**になったという．また，(P'_k) の最適解 $\bar{x}_k$ が

$$\bar{x}_k \in X_0$$

となったとすると，定理 4.1. より $\bar{x}_k$ は (P_k) の最適解となる．従って，このような場合も (P_k) を分枝する必要はない．

以上より，以下の定理が導き出せる．

定理 4.2. (**分枝限定法の原理**)

最適化問題

$$\begin{array}{lll} (P_0) & \text{最大化} & \Phi(x) \\ & \text{条　件} & x \in X_0 \end{array}$$

について，以下の手順に従うことによって最適解が得られる．

$\tilde{z} := -\infty$, $\mathcal{N} = \{(P_0)\}$, $p = 1$ とする．
$\mathcal{N} = \phi$ なら　へ．そうでなければ　へ．
$\mathcal{N}$ の中から問題を選び出し，それを (P_k) とする．また，$\mathcal{N}$ から (P_k) を取り除く．
緩和問題 (P'_k) を定め，それを解いて解 $\bar{x}_k$ と目的関数値 $\bar{z}_k$ を求める．
$\bar{x}_k$ が (P_0) の実行可能解かつ $\bar{z}_k > \tilde{z}$ なら　へ．$\bar{x}_k$ が (P_0) の実行可能解でないかつ $\bar{z}_k > \tilde{z}$ なら　へ．$\bar{z}_k \leq \tilde{z}$ なら　へ．
$\tilde{x} := \bar{x}_k$, $\tilde{z} := \bar{z}_k$ と値を更新し，　へ．
(P_k) の子問題 $(P_p), \ldots, (P_q)$ を作り，$\mathcal{N} := \mathcal{N} \cup \{(P_p), \ldots, (P_q)\}$, $p := q + 1$ として　へ．
$\tilde{z} = -\infty$ なら (P_0) には実行可能解が存在しない．$\tilde{z} > -\infty$ なら $\tilde{x}$ が (P_0) の最適解，$\tilde{z}$ が (P_0) の最適値である．

定理 4.2. の $\tilde{x}$ は**暫定解**，$\tilde{z}$ は**暫定値**と呼ばれ，その時点までに見つかった最も良い解とその値を示す．従って，　で得られた $\tilde{x}$ が (P_0) の最適解となる．

5　整数計画法

5.1　完全ユニモジュラー

定義 5.1. (**完全ユニモジュラー行列**)

行列 A の任意の小行列式が $0, +1, -1$ のいずれかであるとき，A は**完全ユニモジュラー**であるといい，A を**完全ユニモジュラー行列**と呼ぶ．

特に，A の要素 a_{ij} は $a_{ij} \in \{0, -1, +1\}$ となる．

定理 5.2. (**完全ユニモジュラーの条件**)

整数行列 $A=(a_{ij})\in \mathbf{Z}^{m\times n}$ について，任意の $R\subseteq\{1,\dots,m\}$ に対して，すべての $j=1,\dots,n$ で

$$\sum_{i\in R_1} a_{ij} - \sum_{i\in R_2} a_{ij} \in \{-1,0,1\} \tag{5.1}$$

となるような分割 $R=R_1\cup R_2$ $(R_1\cap R_2=\phi)$ が存在するとき，A は完全ユニモジュラーである．

Proof. k についての帰納法で A の任意の $k\times k$ 小行列式は $0,+1,-1$ のいずれかになることを証明する．$k=1$ のとき，求める小行列式は A の要素と等しいので定義 5.1. より成り立つ．

次に $k>1$ とし，$B=(b_{ij})_{i,j\in\{1,\dots,k\}}$ を A の $k\times k$ 部分行列とする．$\det(B)=0$ のとき，題意を満たすので $\det(B)\neq 0$ の場合を考える．B の第 i 行と第 j 列を取り除いて得られる行列を B_{ij} と書くとする．$\det(B)$ を第 i 行に関して余因子展開して整理すると

$$\begin{aligned}
\det(B) &= (-1)^{i+1}b_{i1}\det(B_{i1})+\cdots+(-1)^{i+n}b_{in}\det(B_{in})\\
1 &= (-1)^{i+1}b_{i1}\frac{\det(B_{i1})}{\det(B)}+\cdots+(-1)^{i+n}b_{in}\frac{\det(B_{in})}{\det(B)}\\
&= \frac{1}{\det(B)}\begin{pmatrix} b_{i1} & \cdots & b_{in}\end{pmatrix}\begin{pmatrix}(-1)^{i+1}\det(B_{i1})\\ \vdots \\ (-1)^{i+n}\det(B_{in})\end{pmatrix}
\end{aligned}$$

となる．また，$j\neq i$ のとき

$$\begin{aligned}
&\frac{1}{\det(B)}\begin{pmatrix} b_{j1} & \cdots & b_{jn}\end{pmatrix}\begin{pmatrix}(-1)^{i+1}\det(B_{i1})\\ \vdots \\ (-1)^{i+n}\det(B_{in})\end{pmatrix}\\
=&(-1)^{i+1}b_{j1}\frac{\det(B_{i1})}{\det(B)}+\cdots+(-1)^{i+n}b_{jn}\frac{\det(B_{in})}{\det(B)}
\end{aligned}$$

となり，これは B の第 i 行を第 j 行で置き換えた行列の余因子展開なので，0 となる．よって

$$\frac{1}{\det(B)}B\begin{pmatrix}\det(B_{11}) & \cdots & (-1)^{1+n}\det(B_{1n})\\ \vdots & & \vdots\\ (-1)^{i+1}\det(B_{i1}) & \cdots & (-1)^{i+n}\det(B_{in})\\ \vdots & & \vdots\\ (-1)^{n+1}\det(B_{n1}) & \cdots & \det(B_{nn})\end{pmatrix}=E$$

が成り立ち，両辺に左から B^{-1} を掛ければ

$$B^{-1}=\frac{1}{\det(B)}\begin{pmatrix}\det(B_{11}) & \cdots & (-1)^{1+n}\det(B_{1n})\\ \vdots & & \vdots\\ (-1)^{i+1}\det(B_{i1}) & \cdots & (-1)^{i+n}\det(B_{in})\\ \vdots & & \vdots\\ (-1)^{n+1}\det(B_{n1}) & \cdots & \det(B_{nn})\end{pmatrix}$$

が得られる．帰納法の仮定から，$\det(B_{ij}) \in \{0, +1, -1\}$ であるので，$B^* := (\det B)B^{-1}$ は，要素が $0, +1, -1$ のいずれかの行列となる．

B^* の第 1 行 b_1^* は，第 1 要素が 1 の単位横ベクトル e_1 を用いて

$$b_1^* = e_1 B^* = \det(B) e_1 B^{-1} \tag{5.2}$$

と書けるので，$b_1^* B = \det(B) e_1$ となる．そこで $R := \{i \mid b_{1i}^* \neq 0\} = \{i \mid |b_{1i}^*| = 1\}$ とする．すると $\#R \leq k$ であり，さらに，$j = 2, \ldots, k$ に対して

$$\sum_{i \in R} b_{1i}^* b_{ij} = (b_1^* B)_j = (\det(B) e_1)_j = 0$$

が得られる．ここで

$$\begin{aligned} r_+^+ &:= \#\{i \in R \mid b_{1i}^* = 1 \text{ かつ } b_{ij} = 1\} \\ r_+^- &:= \#\{i \in R \mid b_{1i}^* = -1 \text{ かつ } b_{ij} = 1\} \\ r_-^+ &:= \#\{i \in R \mid b_{1i}^* = 1 \text{ かつ } b_{ij} = -1\} \\ r_-^- &:= \#\{i \in R \mid b_{1i}^* = -1 \text{ かつ } b_{ij} = -1\} \end{aligned}$$

とすると，R の取り方から

$$0 = \sum_{i \in R} b_{1i}^* b_{ij} = r_+^+ - r_+^- - r_-^+ + r_-^-$$

となる．即ち $r_+^+ + r_-^- = r_+^- + r_-^+$ となり，$j = 2, \ldots, k$ ならば $\#\{i \in R : b_{ij} \neq 0\} = r_+^+ + r_+^- + r_-^+ + r_-^-$ は偶数となる．

一方，仮定 (5.1) より，すべての j に対して

$$\sum_{i \in R_1} b_{ij} - \sum_{i \in R_2} b_{ij} \in \{-1, 0, 1\} \tag{5.3}$$

となる分割 $R = R_1 \cup R_2$ $(R_1 \cap R_2 = \phi)$ が存在する．ここで

$$\begin{aligned} r_1^+ &:= \#\{i \in R_1 \mid b_{ij} = 1\} \\ r_1^- &:= \#\{i \in R_1 \mid b_{ij} = -1\} \\ r_2^+ &:= \#\{i \in R_2 \mid b_{ij} = 1\} \\ r_2^- &:= \#\{i \in R_2 \mid b_{ij} = -1\} \end{aligned}$$

とすると，$j = 2, \ldots, k$ ならば $r_1^+ + r_1^- + r_2^+ + r_2^- = \#\{i \in R \mid b_{ij} \neq 0\}$ は偶数であるので，

$$\begin{aligned} \sum_{i \in R_1} b_{ij} - \sum_{i \in R_2} b_{ij} &= r_1^+ - r_1^- - r_2^+ + r_2^- \\ &= r_1^+ + r_1^- + r_2^+ + r_2^- - 2(r_1^- + r_2^+) \end{aligned}$$

も偶数になる．ところが，(5.3) が仮定されているから，$j = 2, \dots, k$ で

$$\sum_{i \in R_1} b_{ij} - \sum_{i \in R_2} b_{ij} = 0 \tag{5.4}$$

とならなければいけない．さらに，$j = 1$ の場合でも

$$\sum_{i \in R_1} b_{i1} - \sum_{i \in R_2} b_{i1} = 0$$

となったすると

$$\sum_{i \in R_1} \sum_{j=1}^{k} b_{ij} = \sum_{i \in R_2} \sum_{j=1}^{k} b_{ij}$$

が成り立つ．即ち R_1 の行の和は R_2 の行の和と等しくなり，$\det(B) = 0$ となるから B が正則であるという仮定に反することになる．従って，(5.3) から

$$\sum_{i \in R_1} b_{i1} - \sum_{i \in R_2} b_{i1} \in \{1, -1\} \tag{5.5}$$

であり，特に $R \neq \phi$ と分かる．$y = (y_1, \dots, y_m)$ を

$$y_i := \begin{cases} 1 & (i \in R_1 \text{ のとき}) \\ -1 & (i \in R_2 \text{ のとき}) \\ 0 & (i \notin R \text{ のとき}) \end{cases}$$

とおけば，(5.4),(5.5) はまとめて $yB \in \{e_1, -e_1\}$ と表現できる．すると (5.2) より

$$b_1^* = \det(B) e_1 B^{-1} \in \{\det(B) y, -\det(B) y\}$$

となる．y と b_1^* はともに要素が $0, +1, -1$ のみのベクトルであり，$R \neq \phi$ であるので，$b_{1i}^* \neq 0$ となる i について考えれば $|\det(B)| = 1$ が得られる． □

定義 5.3. (**ベクトルの大小関係**)

列ベクトル $a, b \in \mathbf{R}^n$ の任意の成分 a_i, b_i について，$a_i \leq b_i$ が成り立つとき，$a \leq b$ と表す．

有限個の変数を用いて記述される線形の目的関数を，それらの変数を用いた有限個の線形の制約のもとで，最大化あるいは最小化するのが線形計画問題である．従って，実行可能解の集合は，有限個の半空間の共通集合となる．なお，このような集合は多面体と呼ばれ，以下のように定義される．

定義 5.4. (**多面体**)

ある行列 $A \in \mathbf{R}^{m \times n}$ とあるベクトル $b \in \mathbf{R}^m$ を用いて $P = \{x \in \mathbf{R}^n \mid Ax \leq b\}$ と書ける集合 P を $\mathbf{R}^n$ の**多面体**という．A と b の要素がすべて有理数ならば，P は**有理数多面体**と呼ばれる．

定義 5.5. (**面，頂点，基底解**)

$P := \{x \mid Ax \leq b\}$ を空でない多面体とする．$\delta := \max\{cx \mid x \in P\}$ が有限となる非零ベクトル $c \in \mathbf{R}^n$ に対して，$\{x \mid cx = \delta\}$ は P の支持超平面と呼ばれる．P の**面**は，P 自身，あるいは P の複数の支持超平面と P の共通部分として定義される．$\{x\}$ が面となる点 x は，P の**頂点**と呼ばれる．このような点 x は系 $Ax \leq b$ の**基底解**とも呼ばれる．また，他のどの面も含まない面を**極小な面**と呼ぶ.

定義 5.6. (**整数包，整数多面体**)

多面体 $P = \{x \mid Ax \leq b\}$ について，P に含まれる整数ベクトル全体の凸包を $P_I = \{x \mid Ax \leq b\}_I$ と書き，P_I を P の**整数包**という．また，$P = P_I$ であるとき，P を**整数多面体**という．

多面体 P の極小な面について，以下の命題が成り立つことが知られている ([2],P.59, 系 3.9).

命題 5.7.

$P = \{x \in \mathbf{R}^n \mid Ax \leq b\}$ を多面体とする．すると P の極小な面はいずれも次数が $n - \mathrm{rank}(A)$ となる．有界な多面体の極小な面は次元が 0 の頂点である．

定理 5.8. (**完全ユニモジュラーと整数多面体**)

整数行列 A が完全ユニモジュラーならば，任意の整数ベクトル b に対して多面体 $\{x \mid Ax \leq b, x \geq 0\}$ は整数多面体である．

Proof. A を $m \times n$ 行列とし，$P := \{x \mid Ax \leq b, x \geq 0\}$ とする．P は

$$P = \left\{ x \;\middle|\; \begin{pmatrix} A \\ -E \end{pmatrix} x \leq \begin{pmatrix} b \\ 0 \end{pmatrix} \right\}, \quad E \text{ は } n \text{ 次単位行列}$$

とも書けるので，命題 5.7. より P の極小な面は P の頂点となる．

A が完全ユニモジュラーであるとする．b を整数ベクトル，x を P の任意の頂点とする．すると x は $\begin{pmatrix} A \\ -E \end{pmatrix}$ のある $n \times n$ 正則部分行列 A' を用いて $A'x \leq b'$ と書ける $\begin{pmatrix} A \\ -E \end{pmatrix} x \leq \begin{pmatrix} b \\ 0 \end{pmatrix}$ の部分系の $A'x = b'$ に対する解となる．A は完全ユニモジュラーなので，$|\det(A')| = 1$ であり，従って，クラーメルの公式より，$x = (A')^{-1}b'$ は整数ベクトルとなる．P の任意の頂点は整数ベクトルと分かったので，$P = P_I$ となり，P は整数多面体である． □

また，有理数多面体 P について，以下の定理が成り立つことが知られている ([2],P.115, 定理 5.12).

定理 5.9.

P が有理数多面体のとき，以下の命題 (1) – (7) は互いに同値である．

(1) P は整数多面体である．

(2) P の各面は整数ベクトルを含む.

(3) P の極小な各面は整数ベクトルを含む.

(4) 各支持超平面は整数ベクトルを含む.

(5) 各有理数支持超平面は整数ベクトルを含む.

(6) $\max\{cx \mid x \in P\}$ が有限となる各ベクトル c に対して，$\max\{cx \mid x \in P\}$ は整数ベクトルで達成される.

(7) $\max\{cx \mid x \in P\}$ が有限となる各整数ベクトル c に対して，$\max\{cx \mid x \in P\}$ は整数多面体である.

以上の定理より，整数計画問題において，制約条件を記述している制約式の係数行列が完全ユニモジュラーのとき，最適解は整数解で与えられることが分かる.

5.2 単体法

この節では，線形計画問題を解く方法として，単体法を紹介する．線形計画問題

$$
\begin{array}{lll}
(P) & \text{最大化} & f = c_1x_1 + \cdots + c_nx_n \\
 & \text{条　件} & a_{11}x_1 + \cdots + a_{1n}x_n \leq b_1, \\
 & & \qquad\vdots \\
 & & a_{m1}x_1 + \cdots + a_{mn}x_n \leq b_m, \\
 & & x_1 \geq 0, \ldots, x_n \geq 0
\end{array}
$$

について

$$
\begin{aligned}
&\lambda_1 = b_1 - (a_{11}x_1 + \cdots + a_{1n}x_n) \\
&\qquad\qquad\vdots \\
&\lambda_m = b_m - (a_{m1}x_1 + \cdots + a_{mn}x_n) \\
&\lambda_1 \geq 0, \ldots, \lambda_m \geq 0
\end{aligned}
$$

を満たす変数 $\lambda_1, \ldots, \lambda_m$ を用いて，(P) の制約条件の不等式を等式に変換することができる．つまり，(P) を解くことは，線形計画問題

$$
\begin{array}{lll}
(P') & \text{最大化} & f = c_1x_1 + \cdots + c_nx_n \\
 & \text{条　件} & a_{11}x_1 + \cdots + a_{1n}x_n + \lambda_1 = b_1, \\
 & & \qquad\vdots \\
 & & a_{m1}x_1 + \cdots + a_{mn}x_n + \lambda_m = b_m, \\
 & & x_1 \geq 0, \ldots, x_n \geq 0, \\
 & & \lambda_1 \geq 0, \ldots, \lambda_m \geq 0
\end{array}
$$

を解くことと同値である．このような変数 $\lambda_1, \ldots, \lambda_m$ のことを**スラック変数**という．そこで以下はスラック変数も通常の変数と同様に扱い，制約条件が等式で与えられている線形計画問題を解く

ことを考える．

制約条件が等式で与えられている線形計画問題

$$
\begin{array}{lll}
(P) & \text{最大化} & f = c_1x_1 + \cdots + c_nx_n \\
 & \text{条　件} & a_{11}x_1 + \cdots + a_{1n}x_n = b_1, \\
 & & \qquad\vdots \\
 & & a_{m1}x_1 + \cdots + a_{mn}x_n = b_m, \\
 & & x_1 \geq 0, \ldots, x_n \geq 0
\end{array}
$$

は，以下の手順に従うことで最適解を求めることができる．

$r = \mathrm{rank}(A)$ とすると，r 個の変数 $x_1, \ldots, x_r$ は残りの変数 $x_{r+1}, \ldots, x_n$ で表すことができる．計算の結果

$$
\begin{aligned}
x_1 &= b'_1 - (a'_{1r+1}x_{r+1} + \cdots + a'_{1n}x_n), \\
&\ \ \vdots \\
x_r &= b'_r - (a'_{rr+1}x_{r+1} + \cdots + a'_{rn}x_n), \\
b'_1 &\geq 0, \ldots, b'_r \geq 0
\end{aligned}
\tag{5.6}
$$

になったとする．これらを f に代入して整理すると

$$
\begin{aligned}
f &= c_1(b'_1 - (a'_{1r+1}x_{r+1} + \cdots + a'_{1n}x_n)) + \cdots + c_r(b'_r - (a'_{rr+1}x_{r+1} + \cdots + a'_{rn}x_n)) \\
&\qquad\qquad + c_{r+1}x_{r+1} + \cdots + c_nx_n \\
&= c_1b'_1 + \cdots + c_rb'_r \\
&\qquad - (c_1a'_{1r+1} + \cdots + c_ra'_{rr+1} - c_{r+1})x_{r+1} - \cdots - (c_1a'_{1n} + \cdots + c_ra'_{rn} - c_n)x_n
\end{aligned}
$$

となる．ここで

$$
\begin{aligned}
\gamma_0 &= c_1b'_1 + \cdots + c_rb'_r \\
\gamma_{r+1} &= -(c_1a'_{1r+1} + \cdots + c_ra'_{rr+1} - c_{r+1}) \\
&\ \ \vdots \\
\gamma_n &= -(c_1a'_{1n} + \cdots + c_ra'_{rn} - c_n)
\end{aligned}
$$

とおくと，f は

$$
f = \gamma_0 + \gamma_{r+1}x_{r+1} + \cdots + \gamma_nx_n
$$

と表すことができる．

$\gamma_{r+1}, \ldots, \gamma_n$ がすべて非正であるとき　へ．$\gamma_{r+1}, \ldots, \gamma_n$ の中に正のものがあるとき　へ．

$x_{r+1}, \ldots, x_n$ が増加すると f の値は γ_0 より減少するから，最適解は

$$
x_1 = b'_1, \ldots, x_r = b'_r, x_{r+1} = 0, \ldots, x_n = 0
$$

であり，最適値は $f = \gamma_0$ である．よって (P) は解けたので終了する．

$\gamma_j = \max\{\gamma_i \mid i = r+1, \dots, n\}$ とする．x_j 以外のすべての x_k を 0 とおくと

$$x_k = b'_k - a_{kj}x_j, \qquad k = 1, \dots, r$$

となる．

$a'_{1j}, \dots, a'_{rj}$ のすべてが非正のとき　へ．$a'_{1j}, \dots, a'_{rj}$ の中に正のものがあるとき　へ．

x_j はいくらでも大きい正の値を取れるから

$$f = \cdots + r_j x_j + \cdots$$

を考えると f の最大値は $+\infty$ となってしまうので最適解は存在しない．よって終了する．

$a'_{1j}, \dots, a'_{rj}$ の中の正のものを a'_{kj} とすると

$$x_k = b'_k - a'_{kj}x_j \geq 0$$
$$0 \leq x_j \leq \frac{b'_k}{a'_{kj}}$$

となる．正の a'_{kj} に対する $\dfrac{b'_k}{a'_{kj}}$ のうち最小の値を $\theta = \dfrac{b'_i}{a'_{ij}}$ とする．つまり

$$\theta = \min\left\{ \frac{b'_m}{a'_{kj}} \,\middle|\, a'_{kj} > 0 \right\}$$

とおくと，$0 \leq x_j \leq \theta$．ただし i は $1, \dots, r$ のうちの 1 つである．このとき

$$\begin{aligned}
&x_{r+1} = 0, \dots, x_{j-1} = 0, x_j = \theta, x_{j+1} = 0, \dots, x_n = 0, \\
&x_i = b'_i - a'_{ij}\theta = 0, \\
&x_k = b'_k - a'_{kj}\theta, \quad k = 1, \dots, r
\end{aligned}$$

とおくと，この解に対して f の値は

$$\gamma_0 + \gamma_j\theta \geq \gamma_0$$

となる．即ち，f の値は増加している．

$\dfrac{b'_i}{a'_{ij}} = \theta$ となる添字 $i \in \{1, \dots, r\}$ をとる．独立変数 $x_{r+1}, \dots, x_n$ のうち x_j を

$$x_j = \frac{b'_i}{a'_{ij}} - \left(\frac{a'_{ir+1}}{a'_{ij}}x_{r+1} + \cdots + \frac{1}{a'_{ij}}x_i + \cdots + \frac{a'_{in}}{a'_{ij}}x_n \right)$$

によって置き換え，x_i を新たな独立変数に加え，代わりに x_j を従属変数にする．従って，連立方程式 (5.6) はこの新たな独立変数によって

$$x_1 = b''_1 - (a''_{1r+1}x_{r+1} + \cdots + a''_{1i}x_i + \cdots + a''_{in}x_n),$$
$$\vdots$$

$$x_j = b''_j - (a''_{jr+1}x_{r+1} + \cdots + a''_{ji}x_i + \cdots + a''_{jn}x_n),$$
$$\vdots$$
$$x_r = b''_r - (a''_{rr+1}x_{r+1} + \cdots + a''_{ri}x_i + \cdots + a''_{rn}x_n)$$

と表され，このとき新たな目的関数は

$$f = \gamma'_0 + \gamma'_{r+1}x_{r+1} + \cdots + \gamma'_i x_i + \cdots + \gamma'_n x_n$$

と表される．

以下　へ戻って繰り返す．

線形計画問題の解は多面体の頂点で与えられるが，以上の手順を要約すると次のような意味を持っていることがわかる．　で与えられた点

$$x_1 = b'_1, \ldots, x_r = b'_r, x_{r+1} = 0, \ldots, x_n = 0$$

は，実際には制約条件を満たす多面体の 1 つの頂点であり，$f = \gamma_0$ はその頂点における値を与えている．　で与えられた点もまた多面体の頂点であり，その点における f の値が前の値 γ_0 以上になっていることを示している．上の手順を繰り返すことは，さらに他の頂点を求め，その点における前のもの以上の f の値を計算することを意味する．多面体は有限個の頂点を持っているから上の手順は有限回で終了し，f の最大値が得られる．

以上の手順を**単体法**といい，線形計画問題の多くはこの手順で解くことができる．

6　最長しりとり問題の定式化

6.1　グラフ表現によるモデル化

まず，与えられた辞書における単語間の関係を有向グラフとして表現し，1 つのしりとりをオイラー路に対応付け，辞書から構成される最大準オイラー部分グラフを求める問題としてモデル化する．

開始文字または終了文字になる文字集合を V とおく．ここでは，文字を数字に対応づけて表現し

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

と表す．また，文字 i で始まり文字 j で終わる単語の集合を B_{ij} とし，互いに素な集合の和

$$A = B_{11} \cup B_{12} \cup \cdots \cup B_{nn}$$

を考えると，(V, A) は有向グラフとなる．すると，しりとりはこの有向グラフ $G = (V, A)$ の準オイラー部分グラフとみなせ，最長しりとり問題は次のように表現される．

問題 6.1. (**グラフ表現による最長しりとり問題**)

有向グラフ $G = (V, A)$ の準オイラー部分グラフの中で有向辺の数が最大となるものを見つけよ．

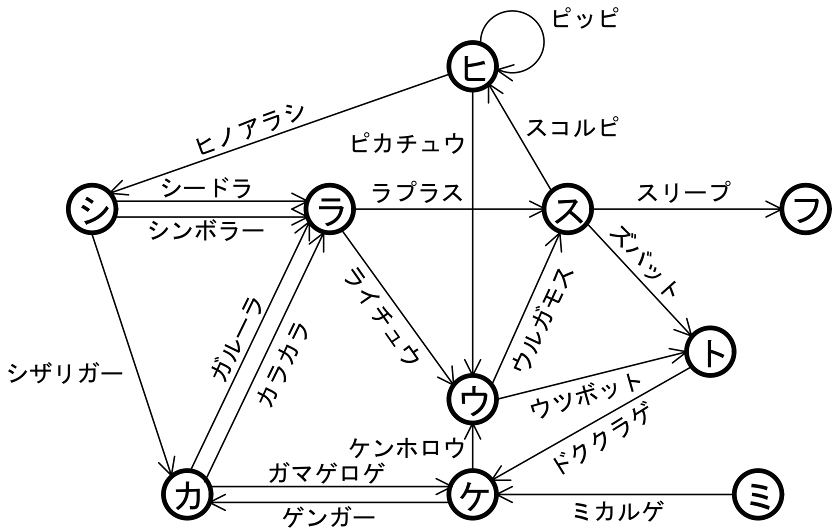

図 2　グラフ表現による最長しりとり問題

6.2　ネットワーク表現によるモデル化

しりとりの長さに着目する際には，単語の違いは問題ではなく，2 つの文字を結ぶ単語の数だけを扱えば十分である．そこで，同じ組み合わせの文字を結ぶ有向辺を改めて容量を持った 1 つの有向辺と考え，それらで構成されるネットワークにおいてフロー量の総和を最大化する問題として最長しりとり問題を考える．

文字 i から文字 j への有向辺の集合 B_{ij} を新たに 1 つの有向辺 a_{ij} と考え，その集合 A' を

$$A' = \{a_{11}, a_{12}, \ldots, a_{nn}\}$$

と表す．また，容量 F を $f_{ij} = \#B_{ij}$ を有向辺 a_{ij} の容量とし，容量の集合 F を

$$F = \{f_{11}, f_{12}, \ldots, f_{nn}\}$$

と表す．さらに，頂点集合 V に入口 s と出口 t を付加し，頂点 s から V のすべての頂点へ，及び V のすべての頂点から頂点 t へ容量 1 の有向辺を加える．すると，最長しりとり問題は次のように表現される．

定義 6.2. (**ネットワーク表現によるモデル**)

ネットワーク $N = (G = (V, A'), F, s, t)$ において，s - t 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せよ．

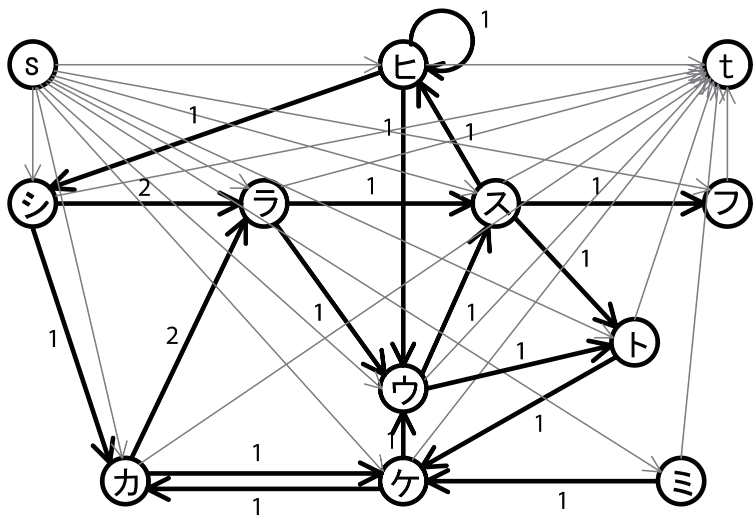

図 3　ネットワーク表現によるモデル

6.3　整数計画問題による定式化

ここでは，ネットワーク表現によってモデル化した最長しりとり問題を線形計画問題 (P) として定式化する．変数 x_{ij} をしりとりに含まれる開始文字 i，終了文字 j の単語の数，つまり i, j 間のフロー量とする．問題 (P) の条件は

1. **(フローとなるための条件)** 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 では出ていくフロー量が 1 多く，頂点 t では入ってくるフロー量が 1 多い．
2. **(連結であるための条件)** 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．

という 2 つの条件の記述からなる．本論文では，この問題 (P) の条件 2 を除き，フローとしての条件だけを考慮した各有向辺を流れるフロー量の総和を最大化する緩和問題 (RP_0) を考える．(RP_0) は以下のように表される．

$$
\begin{aligned}
(RP_0) \quad \text{最大化} \quad & z_0 = \sum_{\substack{i \in V \cup \{s\} \\ j \in V \cup \{t\}}} x_{ij} \\
\text{条　件} \quad & \sum_{j \in V \cup \{t\}} x_{ij} - \sum_{j \in V \cup \{s\}} x_{ji} = 0 \quad \forall i \in V, \\
& \sum_{j \in V} x_{sj} = 1, \\
& \sum_{j \in V} x_{jt} = 1, \\
& 0 \leq x_{ij} \leq f_{ij}, \quad \forall i \in V, \forall j \in V, \\
& 0 \leq x_{sj} \leq 1, \quad \forall j \in V,
\end{aligned}
$$

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

(RP_0) は (P) の緩和問題となっているので，その目的関数の値 z_0 は最長しりとり問題の上界値を与える．ここで，(RP_0) の制約条件を

$$
\begin{aligned}
(RP_0) \quad & \text{最大化} \quad z_0 = \sum_{\substack{i \in V \cup \{s\} \\ j \in V \cup \{t\}}} x_{ij} \\
& \text{条　件} \quad \sum_{j \in V \cup \{t\}} x_{ij} - \sum_{j \in V \cup \{s\}} x_{ji} \leq 0 \quad \forall i \in V, \\
& \qquad \sum_{j \in V} x_{sj} \leq 1, \\
& \qquad \sum_{j \in V} x_{jt} \leq 1, \\
& \qquad -\left(\sum_{j \in V \cup \{t\}} x_{ij} - \sum_{j \in V \cup \{s\}} x_{ji} \right) \leq 0 \quad \forall i \in V, \\
& \qquad -\sum_{j \in V} x_{sj} \leq -1, \\
& \qquad -\sum_{j \in V} x_{jt} \leq -1, \\
& \qquad x_{ij} \leq f_{ij}, \quad \forall i \in V, \forall j \in V, \\
& \qquad -x_{ij} \leq 0, \quad \forall i \in V, \forall j \in V, \\
& \qquad x_{sj} \leq 1, \quad \forall j \in V, \\
& \qquad -x_{sj} \leq 0, \quad \forall j \in V, \\
& \qquad x_{jt} \leq 1, \quad \forall j \in V, \\
& \qquad -x_{jt} \leq 0, \quad \forall j \in V.
\end{aligned}
$$

と書き直す．行列 $A = (a_{ij})$ について，A の第 k 行目の成分 a_{kj} を，$0 \leq k \leq n+2$ のときは

$$a_{kj} = \begin{cases} 1 & ((k-1)n+1 \leq j \leq kn \text{ のとき}) \\ -1 & (j \leq (k-1)n,\ kn+1 \leq j \text{ かつ } j = k \pmod{n} \text{ のとき}) \\ 0 & (\text{その他のとき}) \end{cases}$$

$n+3 \leq k \leq 2n+4$ のときは

$$a_{kj} = \begin{cases} -1 & ((k-1)n+1 \leq j \leq kn \text{ のとき}) \\ 1 & (j \leq (k-1)n,\ kn+1 \leq j \text{ かつ } j = k \pmod{n} \text{ のとき}) \\ 0 & (\text{その他のとき}) \end{cases}$$

$2n+5 \leq k \leq 2n^2+6n+4$ のときは

$$a_{kj} = \begin{cases} 1 & (k = j \text{ のとき}) \\ 0 & (k \neq j \text{ のとき}) \end{cases}$$

とする．また，列ベクトル $b=(b_i)$ を

$$
b_k = \begin{cases} 0 & (1 \le k \le n, n+3 \le k \le 2n+2, n^2+2n+5 \le k \le 2n^2+2n+4, \\ & \quad 2n^2+3n+5 \le k \le 2n^2+4n+4, 2n^2+5n+5 \le k \le 2n^2+6n+4 \text{ のとき}) \\ 1 & (n+1 \le k \le n+2, 2n^2+2n+5 \le k \le 2n^2+3n+4, \\ & \quad 2n^2+4n+5 \le k \le 2n^2+5n+4 \text{ のとき}) \\ -1 & (2n+3 \le k \le 2n+4 \text{ のとき}) \\ f_{ij} & (2n+5 \le k \le n^2+2n+4 \text{ のとき}) \end{cases}
$$

とすると，(RP_0) の条件式は列ベクトル

$$
x = {}^t\begin{pmatrix} x_{11} & \cdots & x_{nn} & x_{s1} & \cdots & x_{sn} & x_{1t} & \cdots & x_{nt} \end{pmatrix}
$$

を用いて $Ax=b$ と表すことができる．ここで，$R \subseteq \{1,\ldots,n^2+2n\}$ について，分割 $R = R_1 \cup R_2$ $(R_1 \cap R_2 = \phi)$ を，n が奇数のときは

$$
R_1 = \left\{ x \;\middle|\; x \equiv 0 \pmod 2 \right\}
$$
$$
R_2 = \left\{ x \;\middle|\; x \equiv 1 \pmod 2 \right\}
$$

n が偶数のときは

$$
R_1 = \left\{ x \;\middle|\; x \equiv 0, \ldots, \frac{2n}{4}, 2n \cdot \frac{3}{4}+1, \ldots, 2n-1 \pmod{2n} \right\}
$$
$$
R_2 = \left\{ x \;\middle|\; x \equiv \frac{2n}{4}+1, \ldots, 2n \cdot \frac{3}{4} \pmod{2n} \right\}
$$

つまり

$$
A = \left(\begin{array}{c|c|c|c||c|c|c} R_1 & R_2 & R_2 & R_1 & R_1 & \ldots & R_1 \end{array} \right)
$$
$$
\begin{array}{ccccccc} \frac{n}{2}\text{列} & \frac{n}{2}\text{列} & \frac{n}{2}\text{列} & \frac{n}{2}\text{列} & \frac{n}{2}\text{列} & & \frac{n}{2}\text{列} \end{array}
$$

と定めると

$$
\sum_{i \in R_1} a_{ij} = \sum_{i \in R_2} a_{ij} \in \{-1, 0, 1\}
$$

が成り立ち，定理 5.2. より A は完全ユニモジュラーになる．よって，各 f_{ij} が整数なので，定理 5.8. より (RP_0) の最適解は整数解として求まる．よって，仮に (RP_0) の解 $x_{ij}>0$ の有向辺により誘導されるグラフが連結である場合には，x_{ij} は (P) の最適解となる．つまり，最長しりとり問題は解けたことになる．

そこで，(RP_0) の解 $x_{ij} > 0$ の有向辺により誘導されるグラフが，複数の連結成分に分かれた場合について考える．(RP_0) の制約条件により，頂点 s と頂点 t の両方を含む連結成分が必ず存在する．今，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 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 つの問題の最長しりとりのうち，長いほうがもとの最長しりとり問題の最適解である．

(2) は，(RP_0) の頂点集合 $V \cup \{s, t\}$ を V_0^* に制限した問題に相当する．よって，新たな問題を解くことなく，(RP_0) の解から x_{ij}, $\forall i \in V_0^*, \forall j \in V_0^*$ を取り出せば構成できる．このときの目的関数の値 z'_0 は

$$z'_0 = \sum_{\substack{i \in V_0^* \\ j \in V_0^*}} x_{ij}$$

として求まる．z'_0 は，(P) の許容解の目的関数の値なので，最長しりとり問題の下界値を与える．

(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 1 つはある」という条件

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

を (RP_0) に加えて得られる問題 (RP_1) を考える．そして，(RP_1) を解き，同様の分子操作を行う．以上を再帰的に繰り返しアルゴリズムを構成する．k 回目の再帰で解く問題 (RP_k) は次のようになる．

$$
\begin{array}{lll}
(RP_k) & \text{最大化} & z_0 = \displaystyle\sum_{\substack{i \in V \cup \{s\} \\ j \in V \cup \{t\}}} x_{ij} \\
& \text{条　件} & \displaystyle\sum_{j \in V} x_{sj} = 1, \\
& & \displaystyle\sum_{j \in V \cup \{t\}} x_{ij} - \sum_{j \in V \cup \{s\}} x_{ji} = 0 \quad \forall i \in V, \\
& & \displaystyle\sum_{j \in V} x_{jt} = 1, \\
& & \displaystyle\sum_{\substack{i \in V_l^* \\ j \in V \setminus V_l^*}} x_{ij} \geq 1 \quad l = 0, \ldots, k-1, \\
& & 0 \leq x_{ij} \leq f_{ij}, \quad \forall i \in V, \forall j \in V, \\
& & 0 \leq x_{sj} \leq 1, \quad \forall j \in V, \\
& & 0 \leq x_{jt} \leq 1, \quad \forall j \in V,
\end{array}
$$

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

V_l^* は，(RP_l) の解における s, t を含む連結成分の頂点集合である．(RP_k) においては，整数条件を取り除くと最適解の整数性は保証されない．

6.4　アルゴリズム

この節では，最長しりとり問題を解決するアルゴリズムを示す．

Algorithem Making Longest Shiritori
begin
　$k := 0$;
　$z^* := 0$; $\{z^*$: 暫定値 $\}$
　$\{\mathbf{x}^*$へ暫定解を保存する $\}$
　while true **do**
　begin
　　(RP_k) を解く;
　　if (RP_k) に実行解がない **then**
　　　goto 1;
　　if (RP_k) の解で $x_{ij} > 0$ の有向辺で構成されるグラフは連結 **then**
　　begin
　　　if $z^* < z_k$ **then**
　　　begin
　　　　$z^* := z_k$;
　　　　z_k を導いた解を $\mathbf{x}^*$ 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 $\mathbf{x}^*$ 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: $\mathbf{x}^*$ よりしりとりを構成する;
{ しりとりの長さは $z^* - 2$}
end.

(P) の最適解が求められれば，そこから容易に実際のオイラー路，即ちしりとり列を求めることができる．オイラー路をリストとして抽出するアルゴリズムを次に示す．各変数は次のことを表す．

$\mathbf{x}$: 最適解となる単語数 x_{ij} のベクトル表現
$\mathbf{x} = (x_{11}, \ldots, x_{nn}, x_{s1}, \ldots, x_{sn}, x_{1t}, \ldots, x_{nt})$
$\mathbf{l}_i$: i 番目のしりとり列のベクトル表現
$\mathbf{l}_i = (l^i_{11}, \ldots, l^i_{nn}, l^i_{s1}, \ldots, l^i_{sn}, l^i_{1t}, \ldots, l^i_{nt})$
L : $\mathbf{l}_i$ の集合
$IN[1 \ldots n]$: 頂点がチェックされたかどうかを表す配列

Algorithem Making Route of Shiritori($\mathbf{x}$)
begin
$L := \{\}$;
$ExtractAllCycle(\mathbf{x}, L)$;
{$\mathbf{x}$ には，s から t への道が残る }
$\mathbf{x}$ より，しりとりを表す頂点のリスト r を作成;
$r_{path} := r$ の頂点集合;
foreach $i \in V$ **do** $IN[i] =$ **false**;
foreach $j \in r_{path}$ **do**
if $IN[j] = false$ **then**
begin
foreach $\mathbf{l}_i \in L$ **do**
if $l^i_{jk} > 0$ or $l^i_{kj} > 0$ **then**
begin
$\mathbf{l}_i$ を頂点のリストに変換し r へ挿入する;
$L := L \setminus \{\mathbf{l}_i\}$
end;
$IN[j] = ture$;

end;
r を出力する;
end.

関数 *ExtractAllCycle* は，$\mathbf{x}$ より同じ頂点を 2 度以上通らない閉路 $\mathbf{l}$ を取り出し，L に追加する関数である．この閉路 $\mathbf{l}$ は縦型探索を使って求められる．アルゴリズム中のベクトル $\mathbf{d} = (d_1, \ldots, d_n)$ は，個々の頂点に入る単語の数を保持するベクトルである．

Function $\mathit{ExtractAllCycle}(\mathbf{x}, L)$
begin
foreach $i \in V$ **do** $d_i = \sum_{j \in V} x_{ij}$;
$k = 1$;
while $\max_{i \in V} d_i > 1$ **do**
begin
foreach $\imath \in V$ **do**
while i を起点として長さ k の閉路 $\mathbf{l}$ が $\mathbf{x}$ にある **do**
begin
$\mathbf{l}$ を L に追加する;
$\mathbf{x}$ より $\mathbf{l}$ で使われた単語を減らす;
$\mathbf{d}$ より $\mathbf{l}$ で使われた単語を減らす;
end;
$k := k + 1$;
end
end.

例えば，辞書 D(図 4)

ウツボット	ウルガモス	ガマゲロゲ	カラカラ	ガルーラ	ゲンガー	ケンホロウ
シードラ	シザリガー	シンボラー	スコルピ	ズバット	スリープ	ドククラゲ
ピカチュウ	ピッピ	ヒノアラシ	ミカルゲ	ライチュウ	ラプラス	

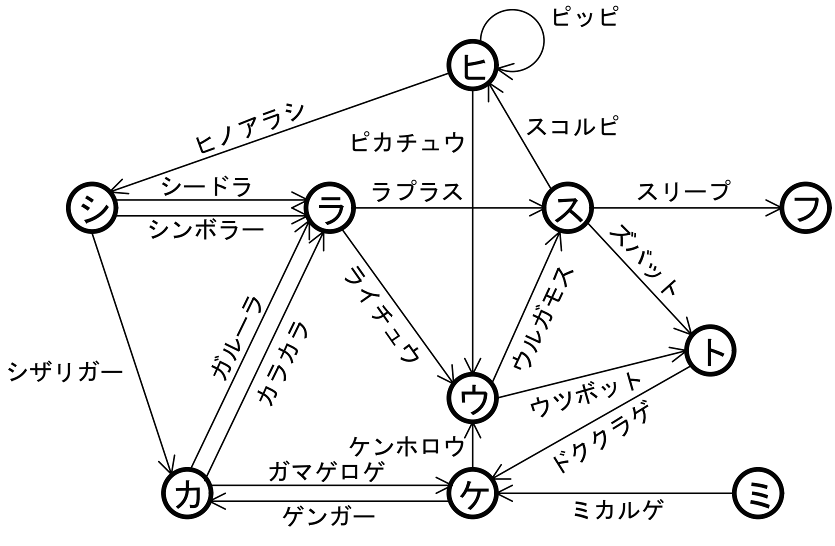

図 4　辞書 D のグラフ表現

について上記のアルゴリズムを用いると，まず，**Algorithem** Making Longest Shiritori で (RP_k) が解かれ，最適解

ウツボット	ウルガモス	ガマゲロゲ	ガルーラ	ゲンガー	ケンホロウ	シードラ
シザリガー	スコルピ	スリープ	ドククラゲ	ピッピ	ヒノアラシ	ライチュウ
ラプラス						

と最適値の 15 が得られる (図 5).

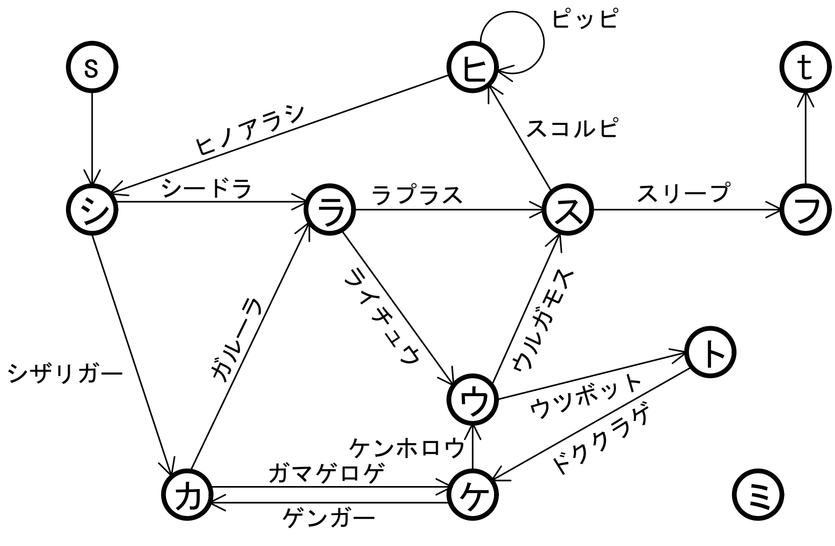

図 5　**Algorithem** Making Longest Shiritori による最適解

この最適解に対して，**Algorithem** Making Route of Shiritori($\mathbf{x}$) を用いると，まず，関数 *ExtractAllCycle* で同じ頂点を 2 度以上通らない閉路の集合 L(図 6)

ヒ→ヒ　カ→ケ→カ　ケ→ウ→ト→ケ　シ→ラ→ス→ヒ→シ

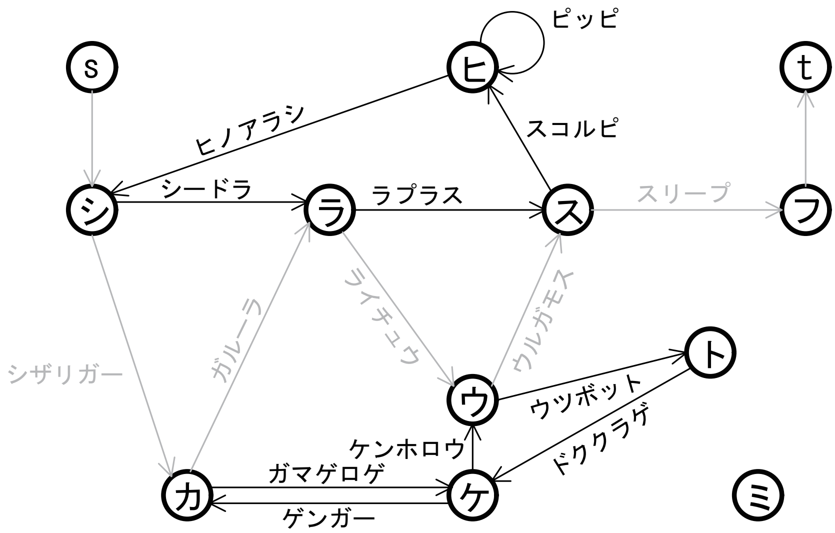

図 6　同じ頂点を 2 度以上通らない閉路の集合 L

と s から t への道 p(図 7)

s →シ→カ→ラ→ウ→ス→プ→ t

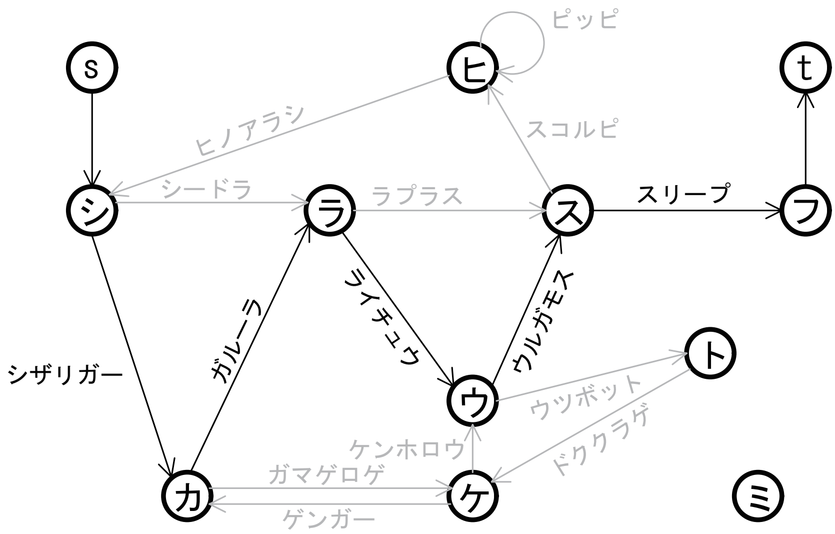

図 7　s から t への道 p

が得られる．そして，道 p をたどりながら，L に含まれる頂点が現れるたびにその閉路をたどっていくことにより，最適値を得るしりとり列

シードラ	ラプラス	スコルピ	ピッピ	ピクシー	シザリガー	ガマゲロゲ
ケンホロウ	ウツボット	ドククラゲ	ゲンガー	ガルーラ	ライチュウ	ウルガモス
スリープ						

を得ることができる．

7 実験

7.1 実験方法

実験は任天堂から発売されている Nintendo DS 用ソフト，ポケットモンスター ブラック＆ホワイトまでに出てくるポケットモンスター全 646 匹を辞書として用いて行った．実装は，Windows XP 上の ActivePerl を用いて，整数計画法のソルバとして，GNU GLPK を使用した．実験に使用した PC は，Core 2 Duo 1.66GHz，1GB Memory である．最長しりとり問題の最適値を求めるとともに，実際のしりとり列を作成し，さらに関数 $ExtractAllCycle$ で取り出した閉路と，そのときに残る s から t への道についても求めた．ただし，今回得られたしりとり列は一例であり，単語の入れ替えや閉路を辿る順番の入れ替えで別のしりとり列をつくることもできる．しかし得られた目的関数値が最適値であることは保証されており，この値を超える解は得られない．

7.2 実験結果

最適値：305

取り出された閉路：

長さ 1 の閉路

イ→イ	イ→イ	カ→カ	キ→キ	ス→ス	タ→タ	チ→チ	ト→ト	ト→ト	ヒ→ヒ
ホ→ホ	ホ→ホ	マ→マ	ミ→ミ	ラ→ラ					

長さ 2 の閉路

ア→ク→ア	ア→ツ→ア	ア→ト→ア	ア→ラ→ア	オ→タ→オ	オ→ト→オ
カ→ケ→カ	カ→シ→カ	キ→コ→キ	キ→ナ→キ	キ→フ→キ	ク→ト→ク
ク→ト→ク	ク→フ→ク	ク→マ→ク	コ→シ→コ	シ→タ→シ	シ→マ→シ
ス→ト→ス	タ→マ→タ	チ→ノ→チ	ト→ム→ト	ト→ラ→ト	ト→ラ→ト
ハ→フ→ハ	ハ→マ→ハ	ハ→メ→ハ	フ→ヤ→フ	マ→ム→マ	マ→リ→マ
リ→ル→リ					

長さ 3 の閉路

ア→ウ→キ→ア	ア→タ→キ→ア	ア→ト→ス→ア	ア→ム→ク→ア	イ→ク→ナ→イ

イ→テ→タ→イ	イ→ル→ラ→イ	ウ→チ→フ→ウ	ウ→ト→ケ→ウ	オ→チ→ホ→オ
オ→メ→ケ→オ	オ→メ→フ→オ	カ→キ→メ→カ	カ→モ→ホ→カ	カ→ユ→シ→カ
カ→ラ→タ→カ	キ→コ→タ→キ	キ→ナ→ラ→キ	キ→ヒ→ハ→キ	コ→ト→チ→コ
コ→ヒ→ハ→コ	コ→フ→サ→コ	コ→ヤ→マ→コ	シ→タ→ホ→シ	シ→チ→ム→シ
シ→ヒ→ユ→シ	シ→マ→ラ→シ	ナ→ロ→ム→ナ	ニ→マ→ミ→ニ	

長さ 4 の閉路

ア→マ→キ→リ→ア	ウ→ハ→モ→ラ→ウ	ウ→ム→ト→ラ→ウ	オ→シ→タ→ネ→オ
オ→チ→ム→ル→オ	カ→キ→リ→ト→カ	カ→ラ→ス→ク→カ	カ→ラ→ス→フ→カ
ク→ナ→サ→タ→ク	ク→ユ→コ→ヨ→ク	コ→ト→ス→メ→コ	コ→ト→ツ→ホ→コ
コ→ヘ→ニ→モ→コ	ス→ハ→ユ→ラ→ス	ス→フ→ネ→ト→ス	

長さ 5 の閉路

ア→ト→ス→ク→ル→ア	ア→ホ→フ→タ→ロ→ア	オ→ト→ス→ク→ル→オ
ケ→ソ→ノ→ス→ミ→ケ	タ→ツ→ヤ→ミ→チ→タ	

長さ 6 の閉路

イ→ム→ナ→シ→ミ→ウ→イ	オ→リ→レ→ハ→フ→テ→オ	カ→ラ→ス→ヒ→フ→テ→カ

長さ 7 の閉路

コ→ト→ス→ナ→ル→ハ→ホ→コ

長さ 9 の閉路

ア→モ→ユ→オ→ロ→ト→ヒ→ニ→ソ→ア

長さ 11 の閉路

イ→セ→メ→マ→ヨ→リ→ラ→ス→ミ→フ→ネ→イ

残った s-t 間の道：

ヘ→ハ→ホ→マ→チ→キ→サ→タ→ケ→ク→ル→ン

得られたしりとり列：

ヘイガニ	ニョロゾ	ソーナノ	ノコッチ	チコリータ	ダンゴロ	ロトム	ムシャーナ	ナエトル	ルカリオ
オニゴーリ	リリーラ	ラティアス	スカンプー	フリーザー	サニーゴ	ゴースト	ドククラゲ	ゲンガー	カイリュー
ユキワラシ	ジグザグマ	マスキッパ	バオッキー	キバニア	アリゲイツ	ツンベアー	アーボック	グレイシア	アズマオウ
ウソッキー	キリンリキ	キャタピー	ピッピ	ビッパ	ハヤシガメ	メラルバ	ハリテヤマ	マグマッグ	クヌギダマ
マッスグマ	マグマラシ	ジャノビー	ピチュー	ユクシー	シキジカ	カブルモ	モジャンボ	ホーホー	ポッポ
ホイーガ	カイリキー	キャモメ	メブキジカ	カラナクシ	シザリガー	カイオーガ	ガマゲロゲ	ケンホロウ	ウツボット
ドクロッグ	クチート	トゲチック	クロバット	ドレディア	アーマルド	ドードー	トサキント	ドテッコツ	ツボツボ
ホエルコ	ゴニョニョ	ヨルノズク	クルマユ	ユキメノコ	ゴローニャ	ヤンヤンマ	マグカルゴ	コアルヒー	ビブラーバ

ハネッコ	コロボーシ	シュバルゴ	ゴーリキー	キノココ	コジョフー	ファイヤー	ヤナップ	ブーバー	バオップ
ブーピッグ	クラブ	ブラッキー	ギャロップ	フィオネ	ネオラント	ドガース	スバメ	メグロコ	コジョンド
トゲキッス	スピアー	アイアント	ドダイトス	ズバット	トリデプス	ズガイドス	スコルピ	ヒヤップ	フシデ
ディアルガ	カポエラー	ラティオス	スリーパー	ハクリュー	ユンゲラー	ラブカス	スリープ	プロトーガ	ガラガラ
ラプラス	スカタンク	グライガー	カラカラ	ライコウ	ウパー	バシャーモ	モンジャラ	ラグラージ	シママ
マニューラ	ライボルト	ドーミラー	ラムパルド	ドッコラー	ラフレシア	アバゴーラ	ランプラー	ラルトス	スターミー
ミネズミ	ミミロップ	フシギダネ	ネイティ	イーブイ	イシズマイ	イルミーゼ	ゼニガメ	メガヤンマ	マッギョ
ヨーテリー	リザード	トドゼルガ	カモネギ	キマワリ	リーフィア	アメタマ	マンキー	キモリ	リングマ
マリルリ	リグレー	レディバ	バネブー	フワンテ	テッポウオ	オオスバメ	メリープ	フリージオ	オニスズメ
メノクラゲ	ケイコウオ	オノンド	ドードリオ	オムナイト	ドリュウズ	ストライク	グランブル	ルギア	アゲハント
トルネロス	ズルッグ	クルミル	ルージュラ	ラクライ	イトマル	ルリリ	リオル	ルンパッパ	ハスボー
ポポッコ	ゴビット	トロピウス	スワンナ	ナックラー	ラッキー	ギラティナ	ナットレイ	イワーク	クサイハナ
ナゲキ	キレイハナ	ナッシー	シェイミ	ミズゴロウ	ウインディ	イノムー	ムーランド	トドグラー	ライチュウ
ウリムー	ムクホーク	クレセリア	アグノム	ムウマ	マルノーム	ムクバード	ドゴーム	ムンナ	ナマケロ
ロゼリア	アーボ	ポカブ	ブースター	ダーテング	グラエナ	ナゾノクサ	サメハダー	タブンネ	ネイティオ
オドシシ	シェルダー	タネボー	ポッタイシ	ジャローダ	ダイケンキ	キバゴ	コラッタ	ダルマッカ	ガルーラ
ラッタ	ダークライ	イシツブテ	ディグダ	ダゲキ	キルリア	アギルダー	タマタマ	マクノシタ	タマザラシ
ジュペッタ	ダグトリオ	オムスター	タマンタ	タッツー	ツタージャ	ヤミラミ	ミノムッチ	チェリム	ムチュール
ルクシオ	オオタチ	チャーレム	ムウマージ	ジラーチ	チョロネコ	ゴルバット	ドジョッチ	チェリンボ	ホエルオー
オタチ	チャオブー	フシギソウ	ウソハチ	チョンチー	チラチーノ	ノクタス	スボミー	ミカルゲ	ケムッソ
ゾロア	アチャモ	モグリュー	ユキノオー	オタマロ	ロズレイド	トゲピー	ビクティニ	ニャルマー	マダツボミ
ミツハニー	ニョロモ	モココ	ゴンベ	ペリッパー	ハトーボー	ポッチャマ	マラカッチ	チョボマキ	キノガッサ
サンダー	タマゲタケ	ケッキング	グレッグル	ルナトーン					

7.3　考察

実験で得られた最適値は 305 で，辞書に含まれる単語の 47.2 %を占めた．また，その後に続く単語のない単語を辞書から除くと単語の数は 564 で，最適値はその数の 54.1 %を占めた．閉路の長さも最大で 11 となり，予想していたよりも短い長さ閉路で分割できることが分かった．しりとり列の最後の文字は「ン」であり，通常行われているしりとりに近い結果が得られた．しかし，今回用いた辞書は「ン」で終わる単語が多く，それがも影響しているのかもしれない．また，今回の実験では (PR_0) を解くだけで最適解が得られた．他の辞書でも試したが，大抵の場合 (RP_0) を解くだけで最適解が得られ，再帰的に問題を解く必要がある辞書は少なかった．

8　おわりに

しりとりという身近なゲームも，数学的な切り口で見ると，興味深い性質がたくさんあることが分かった．文字と単語の集合を有向グラフとして表すことによって，しりとりはオイラー路，即ち一筆書きで表すことができる．さらに有向グラフはネットワークに置き換えて考えることができ，最長しりとり問題を整数計画問題として解くことができる．一見単純そうに見える問題も，よく考えてみると実は奥が深く，探求しようと思えばいくらでも探求できる可能性を秘めていることが分かった．

今回，最長しりとり問題の長さは単語数として定義したが，他の文献を見てみると，長さを文字数として定義している問題もあった．その定義の仕方では，2 単語よりも 1 単語のほうが長い場合

があり，全く別の最長しりとり問題になっていた．また，最長しりとり問題を整数計画問題として解くのではなく，局所探索による解法も文献に載っていた．この方法では厳密な解を求めることはできないようであったが，最長しりとり問題を別の解法で解くこともできる可能性があることが分かった．

9 補足

ポケットモンスター図鑑 (全 646 匹)

フシギダネ	フシギソウ	フシギバナ	ヒトカゲ	リザード	リザードン	ゼニガメ	カメール	カメックス	キャタピー
トランセル	バタフリー	ビードル	コクーン	スピアー	ポッポ	ピジョン	ピジョット	コラッタ	ラッタ
オニスズメ	オニドリル	アーボ	アーボック	ピカチュウ	ライチュウ	サンド	サンドパン	ニドラン♀	ニドリーナ
ニドクイン	ニドラン♂	ニドリーノ	ニドキング	ピッピ	ピクシー	ロコン	キュウコン	プリン	プクリン
ズバット	ゴルバット	ナゾノクサ	クサイハナ	ラフレシア	パラス	パラセクト	コンパン	モルフォン	ディグダ
ダグトリオ	ニャース	ペルシアン	コダック	ゴルダック	マンキー	オコリザル	ガーディ	ウインディ	ニョロモ
ニョロゾ	ニョロボン	ケーシィ	ユンゲラー	フーディン	ワンリキー	ゴーリキー	カイリキー	マダツボミ	ウツドン
ウツボット	メノクラゲ	ドククラゲ	イシツブテ	ゴローン	ゴローニャ	ポニータ	ギャロップ	ヤドン	ヤドラン
コイル	レアコイル	カモネギ	ドードー	ドードリオ	パウワウ	ジュゴン	ベトベター	ベトベトン	シェルダー
パルシェン	ゴース	ゴースト	ゲンガー	イワーク	スリープ	スリーパー	クラブ	キングラー	ビリリダマ
マルマイン	タマタマ	ナッシー	カラカラ	ガラガラ	サワムラー	エビワラー	ベロリンガ	ドガース	マタドガス
サイホーン	サイドン	ラッキー	モンジャラ	ガルーラ	タッツー	シードラ	トサキント	アズマオウ	ヒトデマン
スターミー	バリヤード	ストライク	ルージュラ	エレブー	ブーバー	カイロス	ケンタロス	コイキング	ギャラドス
ラプラス	メタモン	イーブイ	シャワーズ	サンダース	ブースター	ポリゴン	オムナイト	オムスター	カブト
カブトプス	プテラ	カビゴン	フリーザー	サンダー	ファイヤー	ミニリュウ	ハクリュー	カイリュー	ミュウツー
ミュウ	チコリータ	ベイリーフ	メガニウム	ヒノアラシ	マグマラシ	バクフーン	ワニノコ	アリゲイツ	オーダイル
オタチ	オオタチ	ホーホー	ヨルノズク	レディバ	レディアン	イトマル	アリアドス	クロバット	チョンチー
ランターン	ピチュー	ピィ	ププリン	トゲピー	トゲチック	ネイティ	ネイティオ	メリープ	モココ
デンリュウ	キレイハナ	マリル	マリルリ	ウソッキー	ニョロトノ	ハネッコ	ポポッコ	ワタッコ	エイパム
ヒマナッツ	キマワリ	ヤンヤンマ	ウパー	ヌオー	エーフィ	ブラッキー	ヤミカラス	ヤドキング	ムウマ
アンノーン	ソーナンス	キリンリキ	クヌギダマ	フォレトス	ノコッチ	グライガー	ハガネール	ブルー	グランブル
ハリーセン	ハッサム	ツボツボ	ヘラクロス	ニューラ	ヒメグマ	リングマ	マグマッグ	マグカルゴ	ウリムー
イノムー	サニーゴ	テッポウオ	オクタン	デリバード	マンタイン	エアームド	デルビル	ヘルガー	キングドラ
ゴマゾウ	ドンファン	ポリゴン２	オドシシ	ドーブル	バルキー	カポエラー	ムチュール	エレキッド	ブビィ
ミルタンク	ハピナス	ライコウ	エンテイ	スイクン	ヨーギラス	サナギラス	バンギラス	ルギア	ホウオウ
セレビィ	キモリ	ジュプトル	ジュカイン	アチャモ	ワカシャモ	バシャーモ	ミズゴロウ	ヌマクロー	ラグラージ
ポチエナ	グラエナ	ジグザグマ	マッスグマ	ケムッソ	カラサリス	アゲハント	マユルド	ドクケイル	ハスボー
ハスブレロ	ルンパッパ	タネボー	コノハナ	ダーテング	スバメ	オオスバメ	キャモメ	ペリッパー	ラルトス
キルリア	サーナイト	アメタマ	アメモース	キノココ	キノガッサ	ナマケロ	ヤルキモノ	ケッキング	ツチニン
テッカニン	ヌケニン	ゴニョニョ	ドゴーム	バクオング	マクノシタ	ハリテヤマ	ルリリ	ノズパス	エネコ
エネコロロ	ヤミラミ	クチート	ココドラ	コドラ	ボスゴドラ	アサナン	チャーレム	ラクライ	ライボルト
プラスル	マイナン	バルビート	イルミーゼ	ロゼリア	ゴクリン	マルノーム	キバニア	サメハダー	ホエルコ
ホエルオー	ドンメル	バクーダ	コータス	バネブー	ブーピッグ	パッチール	ナックラー	ビブラーバ	フライゴン
サボネア	ノクタス	チルット	チルタリス	ザングース	ハブネーク	ルナトーン	ソルロック	ドジョッチ	ナマズン
ヘイガニ	シザリガー	ヤジロン	ネンドール	リリーラ	ユレイドル	アノプス	アーマルド	ヒンバス	ミロカロス
ポワルン	カクレオン	カゲボウズ	ジュペッタ	ヨマワル	サマヨール	トロピウス	チリーン	アブソル	ソーナノ
ユキワラシ	オニゴーリ	タマザラシ	トドグラー	トドゼルガ	パールル	ハンテール	サクラビス	ジーランス	ラブカス
タツベイ	コモルー	ボーマンダ	ダンバル	メタング	メタグロス	レジロック	レジアイス	レジスチル	ラティアス
ラティオス	カイオーガ	グラードン	レックウザ	ジラーチ	デオキシス	ナエトル	ハヤシガメ	ドダイトス	ヒコザル
モウカザル	ゴウカザル	ポッチャマ	ポッタイシ	エンペルト	ムックル	ムクバード	ムクホーク	ビッパ	ビーダル
コロボーシ	コロトック	コリンク	ルクシオ	レントラー	スボミー	ロズレイド	ズガイドス	ラムパルド	タテトプス

トリデプス	ミノムッチ	ミノマダム	ガーメイル	ミツハニー	ビークイン	パチリス	ブイゼル	フローゼル	チェリンボ
チェリム	カラナクシ	トリトドン	エテボース	フワンテ	フワライド	ミミロル	ミミロップ	ムウマージ	ドンカラス
ニャルマー	ブニャット	リーシャン	スカンプー	スカタンク	ドーミラー	ドータクン	ウソハチ	マネネ	ピンプク
ペラップ	ミカルゲ	フカマル	ガバイト	ガブリアス	ゴンベ	リオル	ルカリオ	ヒポポタス	カバルドン
スコルピ	ドラピオン	グレッグル	ドクロッグ	マスキッパ	ケイコウオ	ネオラント	タマンタ	ユキカブリ	ユキノオー
マニューラ	ジバコイル	ベロベルト	ドサイドン	モジャンボ	エレキブル	ブーバーン	トゲキッス	メガヤンマ	リーフィア
グレイシア	グライオン	マンムー	ポリゴン Z	エルレイド	ダイノーズ	ヨノワール	ユキメノコ	ロトム	ユクシー
エムリット	アグノム	ディアルガ	パルキア	ヒードラン	レジギガス	ギラティナ	クレセリア	フィオネ	マナフィ
ダークライ	シェイミ	アルセウス	ビクティニ	ツタージャ	ジャノビー	ジャローダ	ポカブ	チャオブー	エンブオー
ミジュマル	フタチマル	ダイケンキ	ミネズミ	ミルホッグ	ヨーテリー	ハーデリア	ムーランド	チョロネコ	パルダス
ヤナップ	ヤナッキー	バオップ	バオッキー	ヒヤップ	ヒヤッキー	ムンナ	ムシャーナ	マメパト	ハトーボー
ケンホロウ	シママ	ゼブライカ	ダンゴロ	ガントル	ギガイアス	コロモリ	ココロモリ	モグリュー	ドリュウズ
タブンネ	ドッコラー	ドテッコツ	ローブシン	オタマロ	ガマガル	ガマゲロゲ	ナゲキ	ダゲキ	クルミル
クルマユ	ハハコモリ	フシデ	ホイーガ	ペンドラー	モンメン	エルフーン	チュリネ	ドレディア	バスラオ
メグロコ	ワルビル	ワルビアル	ダルマッカ	ヒヒダルマ	マラカッチ	イシズマイ	イワパレス	ズルッグ	ズルズキン
シンボラー	デスマス	デスカーン	プロトーガ	アバゴーラ	アーケン	アーケオス	ヤブクロン	ダストダス	ゾロア
ゾロアーク	チラーミィ	チラチーノ	ゴチム	ゴチミル	ゴチルゼル	ユニラン	ダブラン	ランクルス	コアルヒー
スワンナ	バニプッチ	バニリッチ	バイバニラ	シキジカ	メブキジカ	エモンガ	カブルモ	シュバルゴ	タマゲタケ
モロバレル	プルリル	ブルンゲル	ママンボウ	バチュル	デンチュラ	テッシード	ナットレイ	ギアル	ギギアル
ギギギアル	シビシラス	シビビール	シビルドン	リグレー	オーベム	ヒトモシ	ランプラー	シャンデラ	キバゴ
オノンド	オノノクス	クマシュン	ツンベアー	フリージオ	チョボマキ	アギルダー	マッギョ	コジョフー	コジョンド
クリムガン	ゴビット	ゴルーグ	コマタナ	キリキザン	バッフロン	ワシボン	ウォーグル	バルチャイ	バルジーナ
クイタラン	アイアント	モノズ	ジヘッド	サザンドラ	メラルバ	ウルガモス	コバルオン	テラキオン	ビリジオン
トルネロス	ボルトロス	レシラム	ゼクロム	ランドロス	キュレム				

※「ニドラン♂」「ニドラン♀」「ポリゴン 2」「ポリゴン Z」はそれぞれ「ニドランオス」,「ニドランメス」,「ポリゴンツー」,「ポリゴンゼット」として処理した.

参考文献

[1] 乾信雄, 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, 第 46 号, pp105-117, 2005.

[2] B. コルテ,J. フィーゲン, 組合せ最適化 理論とアルゴリズム, シュプリンガー・フェラアーク東京, 2005.

[3] 榎本彦衛, グラフ学入門, 日本評論社, 2005.

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

[5] 杉山昌平, 初等数学シリーズ 4, 線形計画, 朝倉書店, 1972.