

反復局所探索法に基づいた  
スケルトンパズルの生成アルゴリズム

学生番号 2014288  
氏名 西島善治  
提出年度 平成 29 年度

## 目次

1. はじめに .....	1
2. スケルトンパズル .....	2
2.1 ルール .....	2
2.2 解法 .....	2
2.3 関連するパズル .....	3
3. 提案手法 .....	5
3.1 定式化 .....	5
3.2 局所探索法 .....	5
3.3 反復局所探索法 .....	6
4. インスタンス生成実験 .....	9
4.1 実験 1 : 摂動の有意性 .....	9
4.2 実験 2 : 盤面の大きさと暫定解の更新回数との関係 .....	9
4.3 実験 3 : 辞書に使われる文字数と白マス数の割合との関係 .....	10
4.4 生成されたインスタンスの例 .....	11
5. まとめ .....	15
謝辞 .....	16
参考文献 .....	16
付録 .....	17

# 1. はじめに

本研究では、スケルトンパズル（英語名：criss cross）について取り上げる。今日、パズルは単なる娯楽にとどまらず、初等教育や脳トレなどでの活用が注目されている。スケルトンパズルは外国語の単語や歴史用語といった専門用語の学習や、地域学習への活用の可能性を秘めている[1]。ところがパズルのインスタンス（問題例）を作成することは、素人にとっては一筋縄では行かなく、難しい組み合わせ問題と考えられる。用語の学習や地域学習に用いる場合、異なるパズルを大量に生成する必要がある。そのためには、パズル制作の手間をなるべく軽減する必要がある。そこで本研究ではスケルトンパズルを、コンピュータを用いて自動生成することを考える。

スケルトンパズルを自動生成する方法は、従来から研究されている。従来手法では制約最適化問題に基づいた定式化を行い、アルゴリズムを適用することでそれを解くといったアプローチが取られている。[2]の手法では、入力として盤面の一辺の長さ  $n$  と辞書と呼ばれる単語の集合が与え、**厳密解法**に基づくソルバを用いてパズルを生成する手法が提案されている。しかし厳密解法によって制約最適化問題を解く場合、計算に膨大な時間がかかるため、従来手法では一辺の長さが 10 以上の大きな盤面の生成は現実的な時間内では困難であった。

以上を踏まえて、本研究では現実的な時間内でも大きい盤面の問題を自動生成する手法を提案する。具体的には**近似解法**である**反復局所探索法**に基づいたスケルトンパズルのインスタンス生成アルゴリズムである。このアルゴリズムをC言語によって実装し、計算実験を行った結果、一辺の長さが 30 のインスタンスをも生成することができたため、提案手法が従来手法より有望であることが確認された。

以下、2章でスケルトンパズルのルールと関連パズルについて説明し、3章では提案手法とその具体的なアルゴリズムを解説する。4章では開発したエンジンによるインスタンス生成実験を行い、5章でまとめを行う。

## 2. スケルトンパズル

### 2.1 ルール

スケルトンパズルでは白と黒に彩色された  $n \times n$  の盤面と、辞書と呼ばれる単語のリストが解答者に与えられる。縦もしくは横に連なる、長さ 2 以上の極大な白マスの区間をスロットと呼ぶ。解答者は、スロットの長さや既に入っている単語をヒントにし、リストに含まれるすべての単語をすべてのスロットに 1 対 1 に割り当てることを求められる。インスタンスの例を以下の図 1 に、その解答を図 4 に示す。

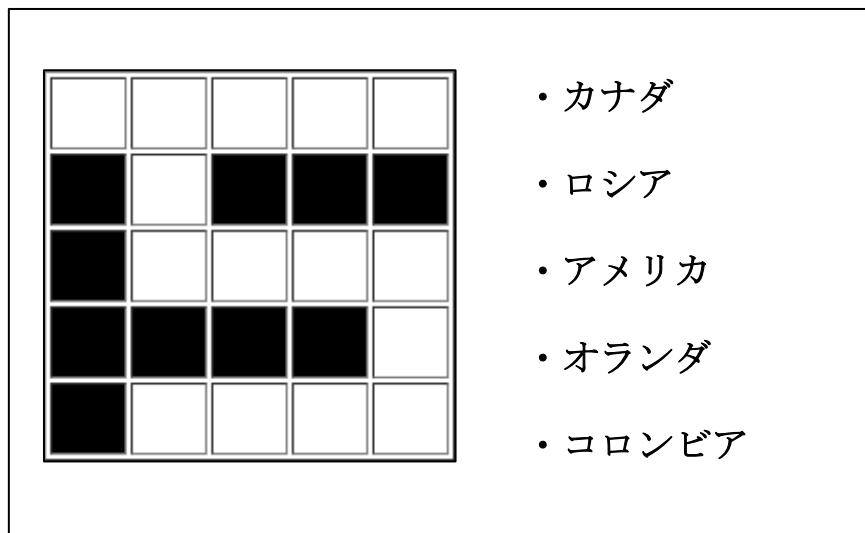


図 1: スケルトンパズルの問題例

### 2.2 解法

図 1 のインスタンスを例にスケルトンパズルの解法を示す。スケルトンパズルを解くためには、単語とスロットの長さの関係と既に配置されている文字の 2 つの情報を手掛かりにする必要がある。まず「コロンビア」という単語に注目すると、この他に 5 文字の単語はリストに存在しないため、配置するスロットを確定することができる。配置した様子を図 2 に示す。次に、3 文字のスロットは 2 つ存在するが一方にはスロットの 1 文字目に「ロ」が既に配置されている。これに注目することで、配置する単語をリストから「ロシア」と確定することができる。配置した様子を図 3 に示す。このような手法を繰り返すことにより、スケルトンパズルの解を導くことができる。最終的な解答を図 4 に示す。



図 2 : 1 つ目の単語を配置した様子



図 3 : 2 つ目の単語を配置した様子



図 4 : 解答

### 2.3 関連するパズル

スケルトンパズルに関連するパズルとして、ナンスケやクロスワードパズルなどが知られている。これらのパズルは雑誌や新聞に懸賞問題として掲載されることや、単語学習などの教育に使われることが多い。これらのパズルもスケルトンパズルと同様に、白と黒に彩色された盤面からなるが、使用する単語にそれぞれ異なる特徴がある。

ナンスケは、スケルトンパズルの辞書にあたる単語の集合がすべて数列になっており、解答者は与えられたすべての数列をすべてのスロットに当てはめることが求められる。ナンスケの問題例を図 5 に、その解答を図 6 に示す。

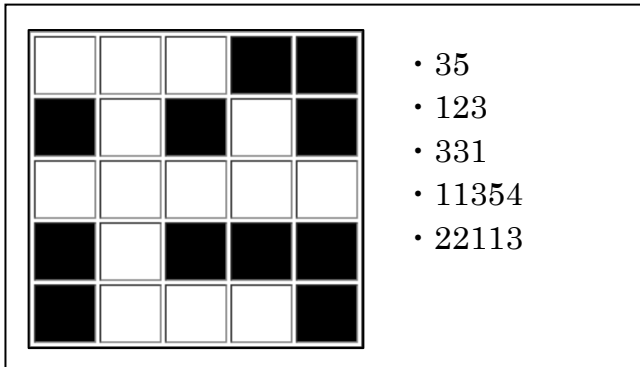


図 5 : ナンスケの問題例



図 6 : 図 5 の解答

クロスワードパズルは、カギと呼ばれるヒント文の集合が与えられる。解答者はカギを手掛かりにして単語を推定し、その単語をスロットに割り当てることが求められる。クロスワードパズルの問題例を図 7 に、その解答を図 8 に示す。

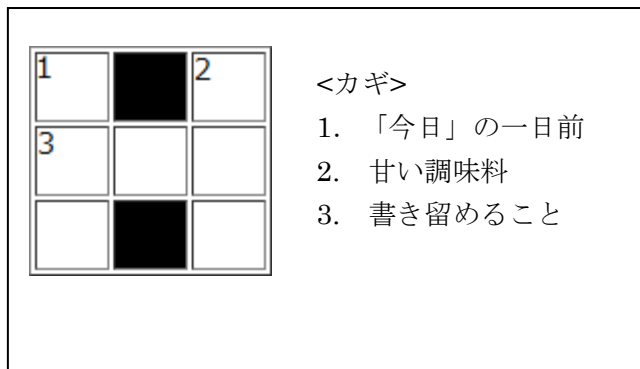


図 7 : クロスワードパズルの問題例

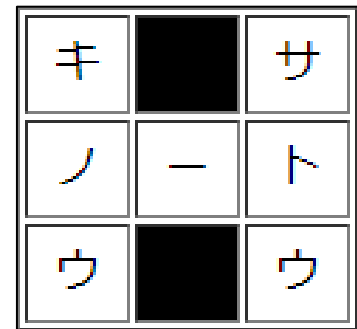


図 8 : 図 7 の解答

本論文が提案する手法ではスケルトンパズルだけでなく、これらの関連パズルの自動生成にも応用することができる。

## 3. 提案手法

### 3.1 定式化

スケルトンパズルのインスタンス生成問題を、以下のように定式化した。

#### スケルトンパズルのインスタンス生成問題

入力：整数  $n$ 、辞書  $D$

出力：辞書  $D$  に含まれる単語の  $n \times n$  盤面上への連結な配置で、白マスの個数を最大にするようなもの。（単語がカバーするマスを白、カバーしないマスを黒とする。）

インスタンスが満たすべき条件として、以下の性質を考慮する。

- ・白マスはすべて連結である。
- ・黒マスの数や配置に制限はない。
- ・リストに無い単語は現れない。
- ・解は唯一である。

このうち解の唯一性を考慮するのは、あるインスタンスが複数解を持つ場合、論理的な手順で解を一つに確定させることができなくなるためである。しかし、上記の問題を解いて得られるインスタンスが複数解を持つことは現実にはほとんどないため、本研究では考慮しないものとする。

### 3.2 局所探索法

局所探索法とは最適化問題の解法の一つであり、焼きなまし法やタブーサーチなど様々なメタヒューリスティクスの手法に組み込まれている。局所探索法の概略を以下に示す。

#### (局所探索法)

1. 初期解  $x$  を生成する。
2. 現在の解  $x$  の近傍  $N(x)$  に改善解  $y$  が存在するならば  $x \leftarrow y$  とし、2 の先頭に戻る。
3. 現在の解  $x$  を出力し、停止。

上記における解とは、単語の配置（すなわちインスタンス）である。また、解  $x$  の近傍  $N(x)$  は  $x$  に含まれない単語を加えて得られる解すべての集合とする。上記の枠組みをスケルトンパズルのインスタンス生成問題に適用する方法を以下に示す。

#### (スケルトンパズルのインスタンス生成問題における局所探索法)

1. まず  $n \times n$  のすべてのマスが黒マスの盤面を用意する。次に辞書にある単語から単語をランダムに一つ選び盤面上の配置可能な位置に配置し、これを初期解とする。

2. 現在の解の近傍を探索する。近傍解は現在の解に辞書からすでに使われていない単語を選び、すでに配置されている単語と交差するように追加したものである。近傍解に現在の解よりも白マスの個数が増加している解が見つかった場合、その解を現在の解とし、1に戻る。
3. 現在の解を出力とし停止。

局所探索法により出力される解は**局所最適解**である。局所最適解にたどり着いた場合、図9のように現在の解が最適解でなくても、これまで同様の探索範囲では改善解は存在しない。したがって、これ以上解を改善するためには最適解から抜け出す必要がある。局所最適解を抜け出す方法はいくつも知られているが、本研究では**反復局所探索法**を用いた。以下、反復局所探索法について述べる。

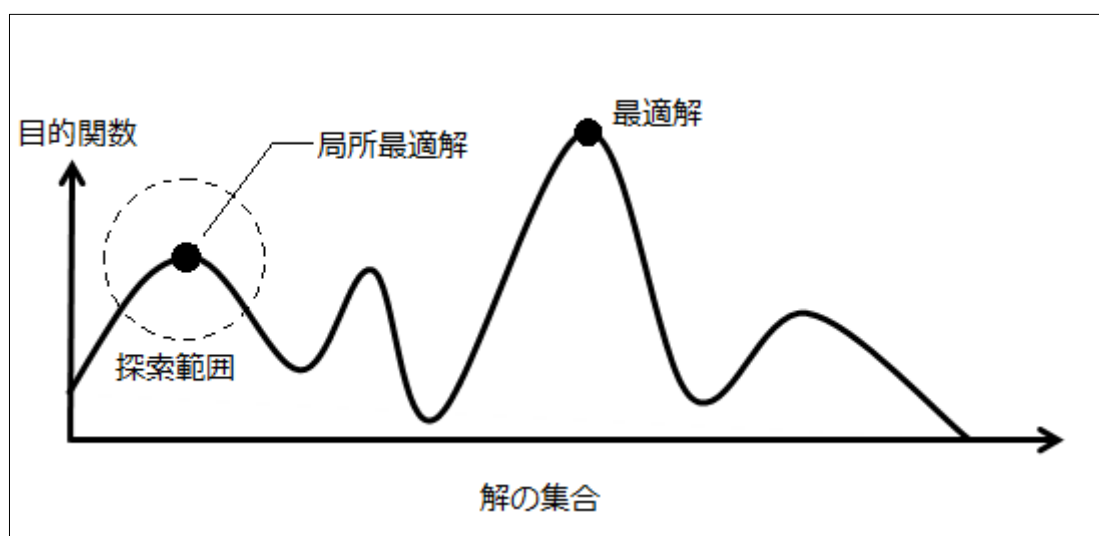


図9：局所最適解

### 3.3 反復局所探索法

反復局所探索法とは文字通り局所探索法を繰り返し行う枠組みである[3]。反復局所探索法の概略を以下に示す。

#### (反復局所探索法)

1. 解  $x_0$  を生成する。また暫定解  $x^*$  を、 $x^* \leftarrow x_0$  とする。
2. 終了条件（計算時間など）が満たされるまで以下を繰り返す。
  - ・  $x_0$  を初期解とした局所探索法を行い、得られた解を  $x$  とする。
  - ・  $f(x) \geq f(x^*)$  ならば暫定解を  $x^* \leftarrow x$  に更新（ $f$  は目的関数で、解  $x$  における白マス個数）。
  - ・ 暫定解  $x^*$  を用いて、次の初期解  $x_0$  を生成（**摂動**）。
3. 暫定解  $x^*$  を出力し、停止。



反復局所探索法を設計する上で大事なことは、探索の多様性と収斂性の両方を実現することである。提案した局所探索法に基づくアルゴリズムでは、現在の解から近傍解への移動は単純に単語を追加するというものであるため、収斂性は保たれている。またこの枠組みにおいて多様性は摂動によって実現される。これは局所探索法により局所最適解に到達した場合に、独自の操作を行うことで暫定解を強制的に移動させる手法である。なおこの際、解の目的関数値が改善する必要はない。摂動のイメージを表したものを図 10 に示す。

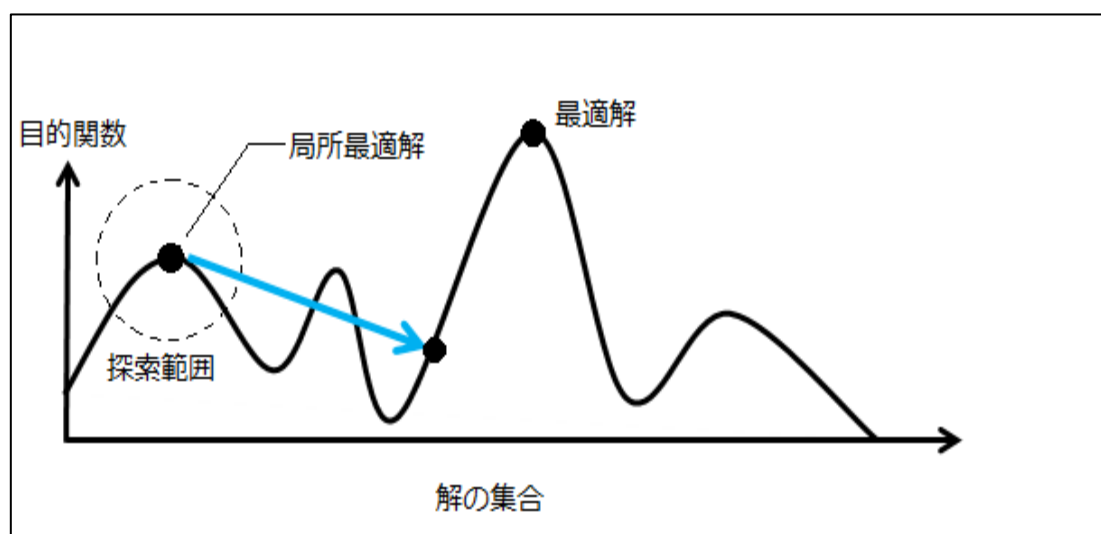


図 10：摂動のイメージ（青線が摂動を示す。）

本研究で提案する摂動のアルゴリズムは以下の通りである。

#### （暫定解の摂動）

1. 解が非連結になるまでランダムに選んだ単語を削除する。非連結になった場合、目的関数（白マスの数）が最大の連結成分のみを残す。
2. 残った連結成分を上下左右のいずれかの方向に確率的に平行移動する。
3. 得られた解を次の局所探索の初期解  $x_0$  とする。

以下、摂動のメカニズムを解説する。局所探索により局所最適解に到達した解が図 4 の形になったとする。まず、ランダムで単語選びパズルが非連結になるまで削除する。ここでは「コロンビア」、「アメリカ」の順で単語が削除された結果、パズルが非連結になったとする。このときの盤面を図 11 に示す。

	□			
	シ			
	ア			カ
				ナ
	オ	ラ	ン	ダ

図 11：単語削除後の盤面

非連結になった場合それぞれの連結成分に注目し、最も目的関数が大きい成分のみを残し、残りを削除する。図 11 の場合は、「ロシア」のみの連結成分よりも「カナダ」、「オランダ」が含まれる連結成分のほうが白マスの個数が多いのでこの成分のみを残し、残りの連結成分をすべて削除する。操作後の盤面を図 12 に示す。

				カ
				ナ
	オ	ラ	ン	ダ

図 12：操作後の盤面

				カ
				ナ
	オ	ラ	ン	ダ

図 13：平行移動後の盤面

次にランダムで上下左右いずれかの方向を選び可能ならば、盤面をその方向に平行移動する。上方向が選ばれたとして平行移動した様子を図 13 に示す。ここから次の局所探索を始め、暫定解よりも目的関数が改善した場合、暫定解を更新する。改善しなかった場合は摂動を適用する前の状態に戻す。反復局所探索法ではこのような操作を繰り返し行うことによって解を改善していく。

## 4. インスタンス生成実験

3章で定式化したスケルトンパズルのインスタンス生成問題をC言語により実装した。なお、プログラムの総行数は1600行程度である。このプログラムを用いて3つのインスタンス生成実験を行った。実験に使用したコンピュータのCPUはIntel® Core™ i7-4790(クロック周波数は3.60GHz)、メモリ容量は16GBである。以下実験方法とその結果をそれぞれ示す。

### 4.1 実験1：摂動の有意性

#### [方法]

2章で提案した摂動のメカニズムが有意であるかを確かめるために、摂動を適用する場合としない場合で出力される解の目的関数値を比較する実験を行った。 $n=10, 12, 14$ のそれぞれに対して10回ずつプログラムを実行し、120秒間でパズルを生成した。盤面に対する白マスの割合の平均を比較した。なお、入力には世界中の国に関する206語の辞書を用いた。

#### [結果]

実験結果を表1に示す。%が表すのはマスの総数 $n^2$ に対する白マスの割合である。摂動を適用することで、 $n=10, 12, 14$ すべての場合で白マスの割合が増加した。この結果は提案した摂動の手法が解の探索における多様性を実現していることを示している。

表1：実験1の結果

	平行移動あり	平行移動なし
$n=10$	63.1%	62.8%
12	59.7%	59.4%
14	57.0%	56.0%

### 4.2 実験2：盤面の大きさと暫定解の更新回数との関係

#### [方法]

これは、このプログラムが改善解を探索するときに、 $n$ の大きさがいかに探索時間に影響を与えるかを確かめる実験である。計算時間を120秒に設定し、辞書は含まれる単語数を60から140まで20刻みで5つ用意し、 $n=10, 15, 20, 25, 30$ の場合でパズルの生成を行った。それぞれの場合の暫定解の更新回数を調べた。

## [結果]

実験結果を図 14 に示す。すべての場合でパズルを生成することができた。従来手法[2]では 3600 秒間で  $9 \times 9$  のパズルを生成することが限界であったが、提案手法では 120 秒間で  $30 \times 30$  のパズルを出力することができた。さらに大きい盤面も時間内に初期解を生成することは可能だが、 $n=30$  付近で初期解の生成から 1 つ目の改善解に更新されるまでの時間が 120 秒に近づいていることから、単純に単語を追加して生成された解よりも優れた解を探索できるのは 120 秒間では  $30 \times 30$  までだということが確かめられた。また 140 語の辞書以外ですべて  $n=10$  の時が  $n=15$  の時よりも更新回数が少なくなっているのは、少ない更新回数で最適解に近い解や抜け出すことが不可能な局所最適解に到達したためと考えられる。実際に  $n=10$  ではどの辞書を用いた場合でも暫定解の最終更新時間は開始 60 秒以内であった。

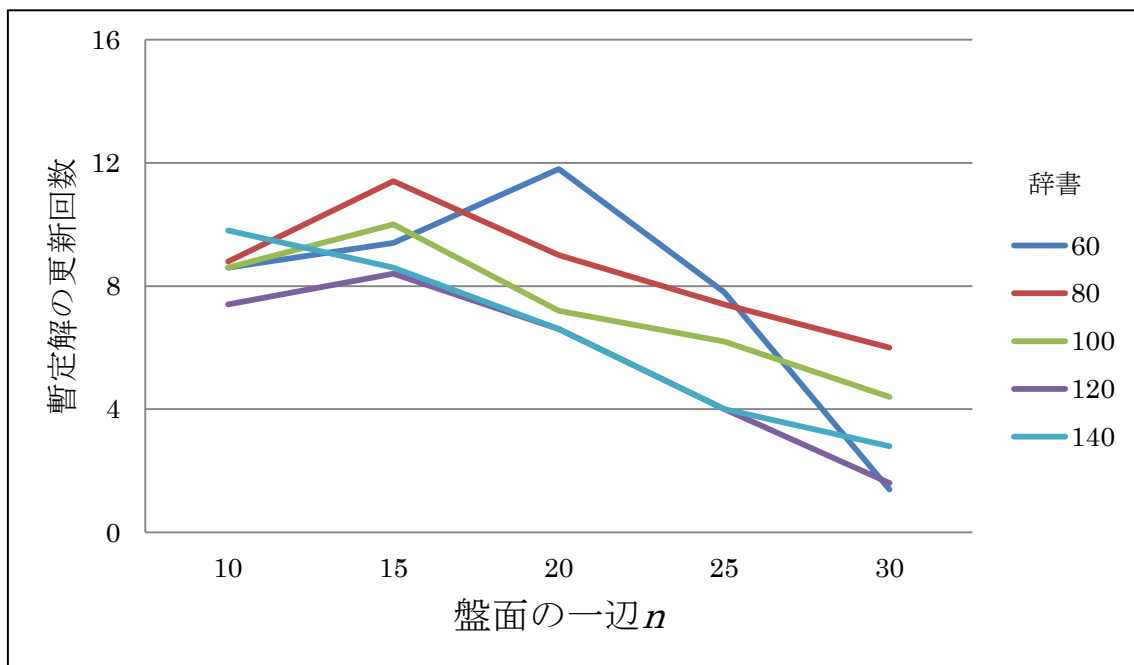


図 14 : 120 秒間での暫定解の更新回数

## 4.3 実験 3 : 辞書に使われる文字数と白マス数の割合の関係

### [方法]

これは、辞書の単語に使われる文字の種類の数や盤面の生成にどのような影響を与えるかを確かめる実験である。文字数が 2, 10, 26, 50 の 100 語の辞書を用意した。それぞれの文字数が意味するものとしては 10 がナンスケ、26 が英語スケルトンパズル、50 は日本語等のスケルトンパズルを想定している。また、文字数が最小の辞書として 0 または 1 のみで構成された単語のみを使ったものを用意した（以降、この辞書によって生成されたパズル

をバイナリスケルトンと呼ぶ。)。これらの辞書を使い  $n=10, 15, 20, 25, 30$  において、120 秒間でパズルの生成を 5 回行い、マス の総数  $n^2$  に対する白マスの割合の平均をみた。  
[結果]

実験結果を図 15 に示す。盤面の一边  $n$  が大きければ大きいほど、また単語に使われる文字の数が多ければ多いほど生成されたパズルの白マスの割合が小さかった。この実験によって日本語スケルトンパズルの自動生成はナンスケや英語スケルトンパズルの自動生成よりも難しい問題であることが確認された。

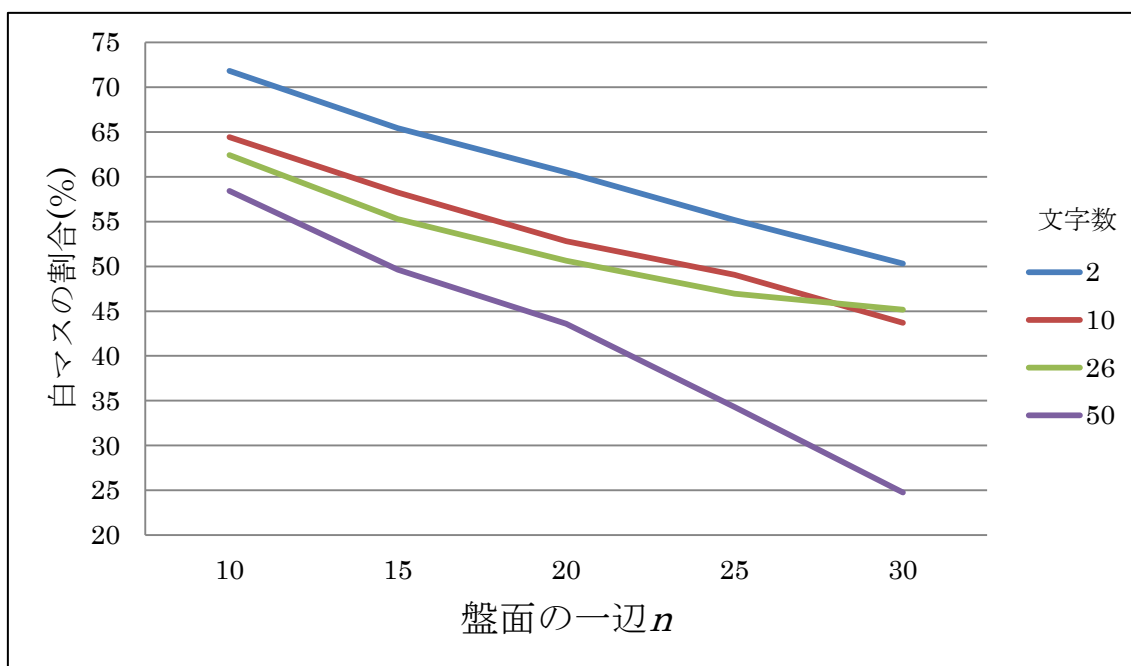


図 15 : マスの総数  $n^2$  に対する白マスの割合の平均

#### 4.4 生成されたインスタンスの例

以上で行った実験により生成されたインスタンスを 2 つ挙げる。図 16 は実験 3 で生成されたバイナリスケルトンのインスタンスである。このインスタンスは本論文で行った全実験において白マスの数が最大となったものである。探索を行った 120 秒間で暫定解の更新は 9 回行われた。白マスの割合は 72% である。

1	0	0	■	1	0	0	0	0	1
0	■	1	1	0	■	1	■	■	1
0	■	0	■	1	■	1	■	0	1
0	■	■	1	0	1	1	1	0	■
1	1	0	0	■	■	0	■	■	1
■	1	■	1	0	1	0	1	1	1
1	1	1	1	■	0	■	1	■	0
■	0	■	1	0	1	1	1	0	1
1	1	0	1	■	1	■	0	■	0
■	1	■	0	1	1	1	1	1	0

図 16 : バイナリスケルトン

次に、提案手法のアルゴリズムによる探索により解が改善していることを示す。実験 1 で生成されたインスタンスの初期解の例を図 17 に、最終的な解を図 18 に示す。図 17 の初期解は探索を開始してから約 1.5 秒で生成されたものである。使用されている単語は 22 語、白マスの割合は 45.4%である。それに対し最終的な解である図 18 では、使用されている単語は 25 語と 3 つ増加し、白マスの割合は 57.1%と 11.7%増加している。この解は探索を開始してから約 108 秒で更新されたものである。

			J					N	I	G	E	R	
C	U	B	A			U	K		T				
O			M		U		Y		A		C		
N			A	U	S	T	R	A	L	I	A		
G			I		A		G		Y		M		
			C				Y				B		
	A	R	A	B			Z		L	A	O	S	
	L					U	S				D		
	B						T		F		I		
	A			P			A		I	R	A	Q	
	N		G	A	B	O	N		J			A	
	I			L					I			T	
U	A	E		A								A	
			T	U	R	K	E	Y		P	E	R	U

图 17 : 初期解

B	E	L	A	R	U	S			Q		C		
A		E			A		K		A		U	S	A
H		B			E	G	Y	P	T		B		
A		A					R		A	R	A	B	
M	O	N	A	C	O		G		R				B
A		O		H			Y				N		O
S		N		A			Z		L		I		S
	A			D		U	S		A		G		N
	L				C		T		O		E		I
U	G	A	N	D	A		A	U	S	T	R	I	A
	E				N		N			O		R	
U	R	U	G	U	A	Y		K	E	N	Y	A	
	I				D					G		Q	
S	A	U	D	I	A	R	A	B	I	A			

図 18 : 最終的な解



## 5. まとめ

本論文では反復局所探索法を用いたスケルトンパズルの自動生成の方法を提案した。また実験により従来手法では $9 \times 9$ より大きいパズルを生成することが困難であったが、提案手法では $30 \times 30$ 以上の大きい盤面のパズルをより短時間で生成することが可能になった。

今後の課題を2つ挙げる。1つ目は更なる高速化である。筆者が実装したプログラムにはまだ改善の余地がある可能性があり、同じアルゴリズムでもより高速に動く実装が出来れば、同じ時間内でさらに良い解が探索できると考えられる。また、焼きなまし法や遺伝的アルゴリズムなど本研究とは別の近似解法アルゴリズムを適用することで、より良いパズルを生成できないか検討することも有益であろう。2つ目は辞書を自動生成できるようにすることである。提案手法では辞書を何らかの方法で用意しなければならないが、この手間を無くすことが望まれる。特に単語学習や地域振興への利用が期待されるスケルトンパズルは、特定のテーマに関する単語を集める必要がある。そこで、自然言語処理や情報検索・抽出といった技術を用いることで、テーマを与えるだけで自動的にそれに関する単語を集めるエンジンを開発することが望まれる。

また、提案手法をもとに開発したエンジンにより自動生成したパズルを搭載したアプリ、「オタルトンパズル」を公開した[4]。パズルには小樽市に関する単語のみが含まれており、小樽観光協会主催のイベント「小樽ゆき物語」にて観光振興の目的で本アプリを活用した。アプリのスクリーンショットを図19に示す。イベント当日にアプリが集客にどのような影響を与えたかについては岩崎拓人氏の論文を参照されたい[5]。

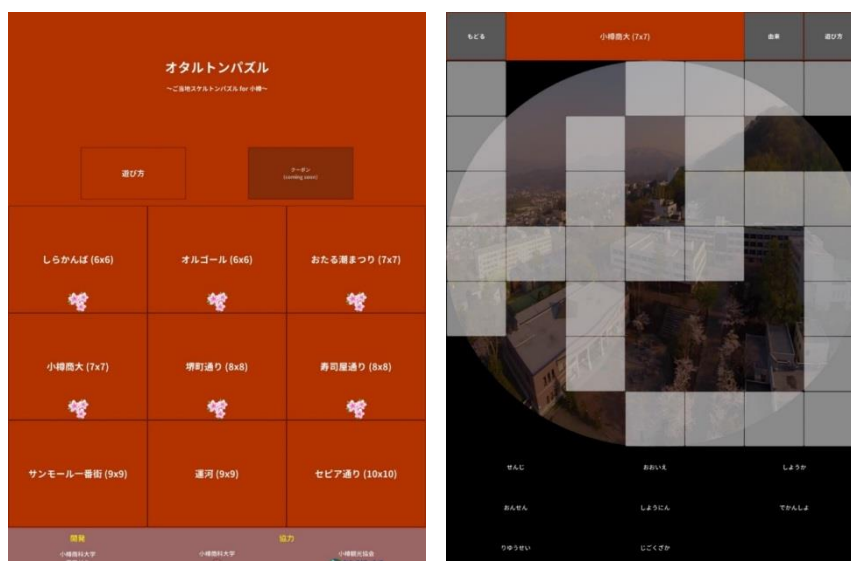


図19：オタルトンパズルのスクリーンショット

最後に付録として、提案手法により生成されたパズルを2つ掲載する。1つ目のパズルは北海道全179市町村の名前を入力した辞書を使用し、180秒間で生成されたものである。使用する単語は44語で、盤面に対する白マスの割合は56.89%である。2つ目のパズルは世界中の国々に関する206語の辞書を使用し、180秒間で生成されたものである。使用する単語は28語で盤面に対する白マスの割合は58.67%である。

## 謝辞

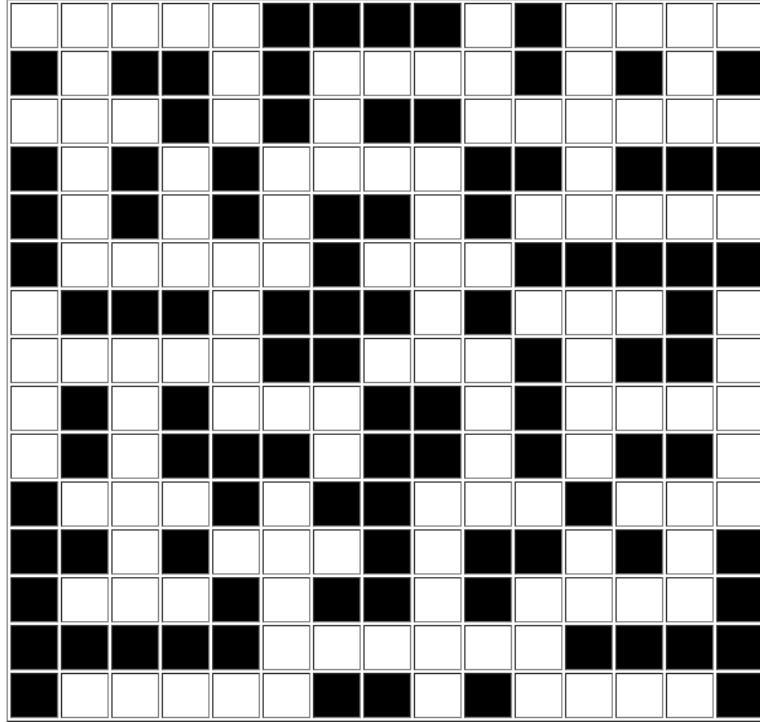
本論文は、多くの方々からのご協力を受け執筆することができました。ご指導いただいた先生、学生論文賞で貴重なご意見をくださった先生方に心より感謝の意を表します。

## 参考文献

- [1] 西勇樹, 小尻智子. 英語長文読解学習のためのクロスワード・パズルの自動生成手法. IEICE tech. rep., Education technology, 111(473):95-100, 2012.
- [2] 佐藤潤一, クロスワードパズル生成問題の新しい定式化. 小樽商科大学・商学部・卒業論文 2016.
- [3] 久保幹雄, J.P. ペドロソ: メタヒューリスティクスの数理. 2009, 共立出版.
- [4] オタルトンパズル.  
<https://play.google.com/store/apps/details?id=org.test.otaruton> (2017年12月21日確認)
- [5] 岩崎拓人, ご当地クイズアプリはユーザの現実行動に影響を与えるか. 小樽商科大学・商学部・卒業論文(予定) 2017.

付録

北海道の市町村パズル(44語)

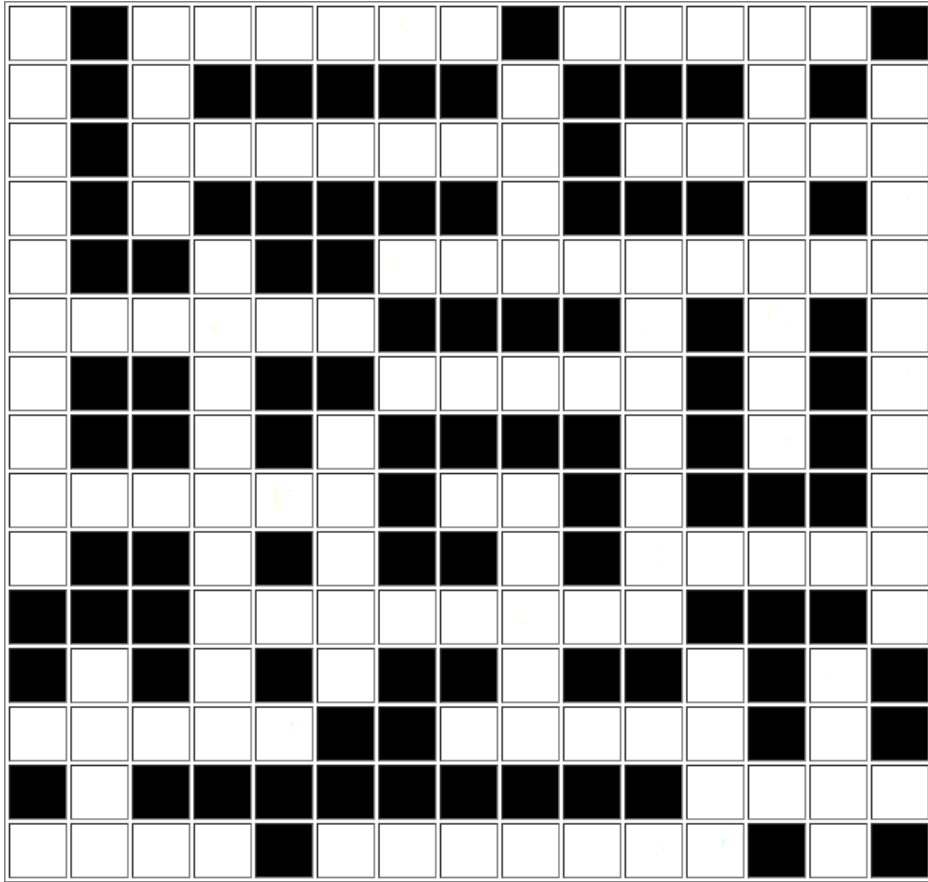


[2文字]	たいき (大樹町)	きよさと (清里町)	くんねつぷ (訓子府町)
だて (伊達市)	たかす (高鷲町)	しままき (島牧村)	しんひだか (新ひだか町)
にき (仁木町)	てしお (天塩町)	たきかわ (滝川市)	とまこまい (苫小牧市)
[3文字]	とうま (当麻町)	とうやこ (洞爺湖町)	のぼりべつ (登別市)
あつま (厚真町)	にせこ (ニセコ町)	びらとり (平取町)	ほろかない (幌加内町)
いけだ (池田町)	びえい (美瑛町)	べつかい (別海町)	[6文字]
えにわ (恵庭市)	ふらの (富良野市)	もせうし (妹背牛町)	おといねつぷ (音威子府村)
おびら (小平町)	やくも (八雲町)	[5文字]	なかさつない (中札内村)
くしろ (釧路市)	るもい (留萌市)	あかいがわ (赤井川村)	ひがしかぐら (東神楽町)
しみず (清水町)	[4文字]	いわみざわ (岩見沢市)	みなみふらの (南富良野町)
しべつ (士別市)	あしべつ (芦別市)	かみのくに (上ノ国町)	
しべつ (標津町)	あつけし (厚岸町)	かもえない (神恵内村)	
しやり (斜里町)	あばしり (網走市)	きようごく (京極町)	

解答

か	み	の	く	に					て		あ	ぼ	し	り
	な			せ		あ	つ	け	し		か		べ	
し	み	ず		こ		つ			お	と	い	ね	つ	ぶ
	ふ		し		し	ま	ま	き			が			
	ら		や		べ			よ		い	わ	み	ざ	わ
	の	ほ	り	べ	つ		と	う	ま					
あ				つ				ご		お	び	ら		と
し	ん	ひ	だ	か			や	く	も		ら			ま
べ		が		い	け	だ			せ		と	う	や	こ
つ		し				て			う		り			ま
	た	か	す		か			く	し	ろ		び	え	い
		ぐ		る	も	い		ん			に		に	
	ふ	ら	の		え			ね		た	き	か	わ	
					な	か	さ	つ	な	い				
	ほ	ろ	か	な	い			ぶ		き	よ	さ	と	

世界の国名パズル(28語)



[2文字]

UK (イギリス)

[4文字]

ARAB (アラブ首長国連邦)

CHAD (チャド)

CUBA (キューバ)

IRAN (イラン)

IRAQ (イラク)

LAOS (ラオス)

MALI (マリ)

[5文字]

CHINA (中国)

INDIA (インド)

KENYA (ケニア)

MALTA (マルタ)

NEVIS (セントクリストファー・ネイビス)

QATAR (カタール)

SAMOA (サモア)

SUDAN (スーダン)

[6文字]

BRUNEI (ブルネイ)

ISRAEL (イスラエル)

RWANDA (ルワンダ)

[7文字]

ALGERIA (アルジェリア)

AUSTRIA (オーストリア)

LEBANON (レバノン)

[8文字]

SLOVENIA (スロベニア)

TRINIDAD (トリニダード・トバゴ)

[9文字]

COSTARICA (コスタリカ)

INDONESIA (インドネシア)

[10文字]

KAZAKHSTAN (カザフスタン)

LUXEMBOURG (ルクセンブルク)

解答

L		I	S	R	A	E	L		M	A	L	T	A	
U		R						L				R		K
X		A	L	G	E	R	I	A		C	H	I	N	A
E		Q						O				N		Z
M			I			C	O	S	T	A	R	I	C	A
B	R	U	N	E	I					U		D		K
O			D			N	E	V	I	S		A		H
U			O		S					T		D		S
R	W	A	N	D	A		U	K		R				T
G			E		M			E		I	N	D	I	A
			S	L	O	V	E	N	I	A				N
	C		I		A			Y			I		C	
S	U	D	A	N			Q	A	T	A	R		H	
	B										A	R	A	B
M	A	L	I		L	E	B	A	N	O	N		D	