

クロスワードパズル生成問題の新しい定式化

A New Formulation of the Crossword Puzzle Generation Problem

佐藤 潤一, 原口 和也

Junichi Sato, Kazuya Haraguchi

小樽商科大学 商学部 社会情報学科

Department of Information and Management Science, Faculty of Commerce, Otaru University of Commerce

1 はじめに

本研究では、クロスワードパズル生成問題の新しい定式化を提案する。クロスワードパズルは、各マスが白もしくは黒に彩色された $n \times n$ 盤面と、カギと呼ばれるいくつかのヒント文から成る。1つの行もしくは列において、連結した、2つ以上の白マスの極大系列をスロットという。カギはスロットと1対1対応し、その対応するスロットに割り当てべき語に関するヒントを与える。プレイヤーはすべてのスロットについて、対応するカギが示す語を割り当ててを求められる。

従来、クロスワードパズル生成問題は、辞書（語の集合）と彩色された $n \times n$ 盤面が与えられ、辞書に含まれる語のみを用いて、すべてのスロットを埋めることを問う探索問題として取り扱われてきた。本研究では、辞書と盤面の次数 n のみが入力として与えられ、語の配置のみならず、盤面の彩色をも問う最適化問題として定式化する。

2 背景

パズルは単なる娯楽にとどまらず、脳トレブームや初等教育への導入など学習支援ツールとしての注目を集めており、クロスワードパズルも例外ではない [2]。しかし従来の定式化は、この用途のためのクロスワードパズルの生成に適さない。もし、辞書はあらゆる語を含むことができ、いくらでも大きくできるという条件であれば、任意に彩色した盤面を与えたとしても、適当な探索エンジンを用いてパズルを生成することができるであろう。しかし学習の対象領域に属する語は一般に限られており、辞書は大きくなり得ない。そしてそのような辞書に対して解を持つような盤面を与えるのは自明ではない。辞書が小さい場合は、語の配置のみならず、盤面の彩色を同時に探索する必要がある。

3 定式化

定式化の方針は、すべての語の先頭と最後尾にそれぞれ黒マスを表す文字 ■ を追加した上で、 $(n+2) \times (n+2)$ 盤面に語を「うまく」配置する、というものである。ただし外周のマスはすべて ■ が割り当てられるようにし、内側の $n \times n$ 盤面をパズルとして得る。

任意の自然数 p, p' ($p < p'$) について、 $[p] = \{1, \dots, p\}$, $[p, p'] = \{p, \dots, p'\}$ とする。任意の $i, j \in [n+2]$ について、 i 行 j 列のマスを (i, j) とする。与えられた辞書 D を m 語の集合 $D = \{w_1, \dots, w_m\}$ とする。それぞれの語 $w_k \in D$ は、 ℓ_k 文字の系列 $w_k = (w_{k,1}, \dots, w_{k,\ell_k})$ である。文字の集合を $A = [q] \cup \{\blacksquare\}$ とすると、上記の方針より $w_{k,1} = w_{k,\ell_k} = \blacksquare$, $w_{k,2}, \dots, w_{k,\ell_k-1} \in A \setminus \{\blacksquare\}$ であり、本来の語 w_k は $(w_{k,2}, \dots, w_{k,\ell_k-1})$ である。

提案する定式化では、語の横向き配置を表す 0-1 変数 $\{x_{k,i,j}\}$ 、語の縦向き配置を表す 0-1 変数 $\{y_{k,i,j}\}$ 、および文字の割当を表す 0-1 変数 $\{z_{a,i,j}\}$ を用いる。このうち $x_{k,i,j}$ は次のように定義される；すべての語 $w_k \in D$ およびマス $(i, j) \in [2, n+1] \times [n+3-\ell_k]$ について、

$$x_{k,i,j} = \begin{cases} 1 & \text{語 } w_k \text{ を横向きに、} \\ & (i, j) \text{ を先頭に配置する、} \\ 0 & \text{語 } w_k \text{ を横向きに、} \\ & (i, j) \text{ を先頭に配置しない。} \end{cases}$$

縦向き配置の語の配置を表す $y_{k,i,j}$ も同様に定義されるが、紙面の都合上省略する。一方 $z_{a,i,j}$ は次のように定義される；すべての文字 $a \in A$ およびマス $(i, j) \in [n+2] \times [n+2]$ について、

$$z_{a,i,j} = \begin{cases} 1 & \text{文字 } a \text{ を } (i, j) \text{ に割り当てる、} \\ 0 & \text{文字 } a \text{ を } (i, j) \text{ に割り当てない。} \end{cases}$$

これらの変数を用いて、クロスワードパズル生成問題を以下のように定式化する。

クロスワードパズル生成問題

最大化 $f(x, y, z)$

制約条件 回数制約, 外周制約, 隣接制約,
文字一致制約, 語交差制約.

目的関数 $f(x, y, z)$ には、パズルの良さを評価する関数を設定する。本研究では、盤面の彩色によって解を評価する；隣接する白マスの対につき +1 点、隣接する黒マスの対につき -1 点、鉤状に連なる白マスの3つ組につき +1 点。制約条件の概要は以下の通りである。(回数

制約) どの語も, 2 回以上配置することはできない. またどのマスにも, 必ず 1 つの文字を割り当てなければならない. さらに横方向と縦方向のそれぞれについて, 高々 1 つの語しか配置できない. (外周制約) 1 行目, $n+2$ 行目, 1 列目, $n+2$ 列目にあるマスには, 必ず文字 ■ を割り当てなければならない. (隣接制約) 任意のマス (i, j) について, 隣接するマスすべてに ■ を割り当てるならば, (i, j) にも ■ を割り当てなければならない. (文字一致制約) 語 w_k を横向きに, (i, j) を先頭に配置するならば, $(i, j), (i, j+1), \dots, (i, j+\ell_k-1)$ に割り当てる文字は, それぞれ $w_{k,1}, w_{k,2}, \dots, w_{k,\ell_k}$ と一致しなければならない. 縦向きの配置も同様である. (語交差制約) 横に並ぶ任意の 3 つのマス $(i, j), (i, j+1), (i, j+2)$ について, (i, j) に文字 ■ を割り当て, $(i, j+1), (i, j+2)$ にそれ以外の文字を割り当てるならば, 必ずある語を横向きに, (i, j) を先頭に配置しなければならない. 縦に並ぶ 3 つのマスについても同様である.

上記の目的関数および制約条件は, すべて変数に関する 1 次式で記述することができる. したがってこの問題を整数線形計画問題や制約最適化問題として取り扱うことができる. なお変数の個数は $O((m+q)n^2)$, 制約式の本数は $O(mn^3)$ で, いずれも入力サイズの多項式オーダーである.

4 生成実験

制約最適化問題として定式化した問題を IBM ILOG CPLEX CP Optimizer (ver. 12.6.0)¹ によって解き, 計算中に得られた解の中で最良のものをパズルとして出力する. 計算時間の上限は 10 分とする. 計算を行ったワークステーションの CPU は Intel[®] Core^(TM) i7-4770 (クロック周波数は 3.40GHz), メインメモリの容量は 8GB である.

実験 1: 既成のパズルの再現. 既成のパズルの解に含まれる語の集合を辞書として与えたとき, 提案手法がこのパズルを作ることができるかどうかを観察する. 次数 $n \in \{6, \dots, 10\}$ のそれぞれに対して, 雑誌やウェブで公開されている 3 問についてパズルの再現を試みた. 結果を表 1 に示す. 提案手法は, $n \leq 7$ のすべて, および $n = 8$ の 1 問についてパズルを構成することができた.

既成のパズルは盤面の彩色とともに与えられるため, 従来の定式化を用いて生成問題を解くこともできる. 論文 [1] のモデル m_3 によって定式化し, 同じく CPLEX によって解いたところ, 瞬時にパズルを構成することができた. クロスワードギバー (ver. 3.2)² はシェアウエ

表 1: 実験 1 の結果. 各エントリは成功回数を示す.

	$n = 6$	7	8	9	10
提案手法 (辞書のみ)	3	3	1	0	0
従来手法 (辞書 + 彩色)	3	3	3	3	3
クロスワードギバー	2	3	2	2	1

表 2: 実験 2 の結果.

	$n = 6$	7	8	9	10
白マスの割合 (%)	69	67	64	65	60
スロット数	7	12	17	20	22
連結成分数	1	4	3	2	7
隣り合う黒マス対の数	5	2	11	12	24

アで, 提案手法同様, 語の配置と盤面の彩色を探索する. そのアルゴリズムはヒューリスティクスに基づいていると考えられ, 既成のパズルという解があるにも関わらず, パズルを生成できないという出力が一部でなされた.

実験 2: 独自辞書を用いたパズル生成. 北海道の地名に関する辞書 (120 語) を作成し, これに含まれる語を解として持つパズルを生成する. 結果を表 2 に示す. 入力として盤面の彩色を必要とするため, 従来の定式化をそのまま適用することはできない. またクロスワードギバーはいずれの回数においてもパズルを生成することはできなかった.

一般の日本語のクロスワードパズルでは, すべての白マスが連結する (連結成分数が 1), 黒マス同士は隣接しない (隣り合う黒マス対の数が 0) など, 盤面の彩色はいくつかの条件を満たすことが知られている. このような条件を制約条件として記述することも可能だが, そもそも与えられた辞書に対し, このような条件を満たす彩色が必ず存在するとは限らない. とはいえ, 連結成分の数と隣り合う黒マス対の数はできるだけ小さいことが望ましく, そのようなパズルを実用的な時間内で生成できるようにすることが今後の課題である.

参考文献

- [1] A. Beacham et al. Constraint programming lessons learned from crossword puzzles. In *Proceedings of AI'01*, pages 78–87, 2001.
- [2] 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.

¹<http://www-01.ibm.com/software/commerce/optimization/cplex-optimizer/>

²<http://katahiromz.web.fc2.com/xword/>