

Title	確率的最適化を用いた絵画的迷路生成
Author(s)	池田, 心; 橋本, 隼一
Citation	情報処理学会論文誌, 53(6): 1625-1634
Issue Date	2012-06-15
Type	Journal Article
Text version	publisher
URL	http://hdl.handle.net/10119/10912
Rights	社団法人 情報処理学会, 池田心, 橋本隼一, 情報処理学会論文誌, 53(6), 2012, 1625-1634. ここに掲載した著作物の利用に関する注意: 本著作物の著作権は(社)情報処理学会に帰属します。本著作物は著作権者である情報処理学会の許可のもとに掲載するものです。ご利用に当たっては「著作権法」ならびに「情報処理学会倫理綱領」に従うことをお願いいたします。 Notice for the use of this material: The copyright of this material is retained by the Information Processing Society of Japan (IPSJ). This material is published on this web site with the agreement of the author (s) and the IPSJ. Please be complied with Copyright Law of Japan and the Code of Ethics of the IPSJ if any users wish to reproduce, make derivative work, distribute or make available to the public any part or whole thereof. All Rights Reserved, Copyright (C) Information Processing Society of Japan.
Description	

確率的最適化を用いた絵画的迷路生成

池田 心^{1,a)} 橋本 隼一^{1,†1}

受付日 2011年9月14日, 採録日 2011年12月16日

概要: 絵画的迷路とは、解くことで正解路が何らかの絵として浮かび上がる迷路のことであり、本論文では白黒2値画像が与えられた場合にそれを自動生成する手法について論じる。画像の縮尺を変更することで高速・厳密にこれを達成する手法はすでに提案されているが、本論文では厳密性を犠牲にすることで縮尺の変更を行わないで済む手法を提案する。具体的には、入力画像と正解路の食い違いを目的関数として定義し、それを最小化するように Simulated Annealing による確率的最適化を行う。さらに不正解部分の構成法についても新しい手法を提案し、これらをアンケートにより評価する。

キーワード: 絵画的迷路, 確率的最適化, 焼きなまし法, ダブルブリッジ近傍

Stochastic Optimization for Picturesque Maze Generation

KOKOLO IKEDA^{1,a)} JUNICHI HASHIMOTO^{1,†1}

Received: September 14, 2011, Accepted: December 16, 2011

Abstract: Picturesque maze is a category of maze, whose solution path shows up as a picture. In this paper we discuss how to make such a maze when a black-and-white image is given. Some methods have been proposed, and they change the scale of image for rapid and strict maze generation. We propose a method which keeps the scale in compensation for the strictness. This method defines an error function as the difference of the input image and a solution path, and minimizes the function stochastically by Simulated Annealing. We also propose a new method making the background of maze more entertaining. Finally the new methods are evaluated by a questionnaire survey.

Keywords: picturesque maze, stochastic optimization, simulated annealing, double bridge method

1. はじめに

迷路は子供から大人まで楽しめる古典的パズルの1つであり、これまで多くの種類の迷路が提案・公開されてきた[4]。それら種類の中で“絵画的迷路”とは、正解路が何らかの絵、たとえば動物・記号・言葉などとして浮かび上がる迷路のことである。絵画的迷路は、パズルとしての面白さだけでなく、絵を描くという芸術的な楽しさをもたらし、これらだけを集めた迷路本も複数出版されている[1], [10]。しかしこれらのアルゴリズムは公開されていないか、自動

的に生成されたものではないとされており[3]、学術的な研究は依然少なく、発展が期待されている。

Okamotoらは、白黒2値画像が与えられたときに、正解路が黒ピクセル部分と一致するように迷路を構成する問題を絵画的迷路構成問題として定式化した[3]。一般的な迷路の自動構成法や解法については多くの研究がなされているが[4]、絵画的迷路の構成法についてはOkamotoらの手法を含め数件にすぎない[3], [7], [8], [9]。これらの既存手法の特徴は、絵と正解路の一致が厳密に達成できてかつ高速である一方で、画像の縮尺を変更する必要がある点である。すなわち、横 C_{wd} ピクセル、縦 C_{hg} ピクセルの画像が与えられた場合、出力される迷路は小さい場合でも横 $2C_{wd}$ ピクセル、縦 $2C_{hg}$ ピクセルとなってしまう、ギザギザが生じることになる。

本論文では、正解路と黒ピクセル部分の多少の食い違い

¹ 北陸先端科学技術大学院大学
Japan Advanced Institute of Science and Technology, Nomi,
Ishikawa 923-1292, Japan

^{†1} 現在, 株式会社 KDDI 研究所

^{a)} kokolo@jaist.ac.jp

を許容することで、与えられた画像の縮尺を変更せずに済むような手法を提案する．具体的には、絵画的迷路構成問題を、食い違いの大きさを目的関数とする最小化問題として定式化しなおし、確率的最適化手法の1つである Simulated Annealing (SA) を用いて最適化する．縮尺を変更しない場合、入力画像と完全に一致する正解路は存在しないか、存在したとしてもその判定は特殊な場合を除いて NP 完全であり [3]、本論文では厳密解法を用いることは避け、確率的最適化を用いて満足化を図る．従来法と比べた場合、多少の食い違いが生じたとしても、縮尺を変更する必要がない分、なめらかな絵を浮かび上がらせることができるのではないかと考える．

Okamoto らは迷路の質を向上させるためのヒューリスティックとして、背景部分（正解路でない部分）の行き止まりの数を減らす手法を提案している [3]．本論文ではそれを拡張した手法についても論じる．最後に、正解路部分の構成法、背景部分の構成法それぞれについて、提案手法の有効性を確かめるためにアンケート調査を行う．

2. 絵画的迷路構成問題

2.1 定義

一口に迷路といっても非常に多くの種類が存在し、次元数、トポロジ、行き止まりの有無、正解路の唯一性、外観などいくつかの観点で分類される [4]．本論文では、最も一般的と思われる、2次元の格子状の全域木型迷路を対象とする．以下にその特徴をまとめる．

- 横 C_{wd} マス、縦 C_{hg} マスの長方形格子状の迷路である．
- 各マスは上下左右最大4マスと接する．その間に壁がある場合は「閉じている」と称し直接の行き来はできない．壁がない場合は「開いている」と称し直接の行き来ができる．
- 外周部のマスは、迷路の外部と壁で隔てられている．ただし、入口と出口が1つずつ存在してその部分だけ外部とつながっている．
- 正解路は、重複しないマスの配列 $P = \{p_1, \dots, p_n\}$ で定義される．ただし、
 - p_1 は入口マスであり、 p_n は出口マスである．
 - すべての p_i と p_{i+1} は、間に壁のない隣り合うマスでなければならない．
 - 逆に p_i と p_{i+j} , $j > 1$ は、隣り合っていないか、間に壁がなければならない．
- ワープ（離れた2マスを移動する仕組み）は設置されない．
- ループ（閉路）は存在しない．到達できないマスも存在しない．いいかえれば、任意のマス（たとえば入口）から任意のマス（たとえば出口）を結ぶ経路が必ず1つ定まる．

このうえで、解きたい絵画的迷路構成問題は以下の方

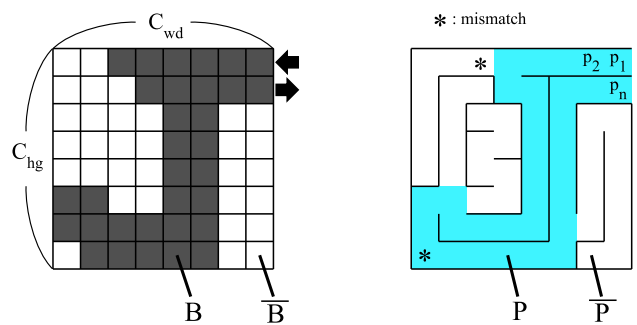


図 1 絵画的迷路構成問題．左のような画像から、右のような迷路を生成する．*は食い違い部分である

Fig. 1 Picturesque maze generation problem. Input image (left) and output maze (right). Mismatches are denoted by *.

に表せる．これは Okamoto らの論文の定義とは若干異なり、画像の縮尺変更を行わず、一方で食い違いを許容したものとしている．図 1 に例を示す．

- 黒白2値の、横 C_{wd} ピクセル、縦 C_{hg} ピクセルの画像が与えられたとする．黒ピクセル（マス）の集合を絵と呼び、 B で表す．白ピクセルの集合を背景と呼び、 $\bar{B}$ で表す．
- 入口マスと出口マスも与えられる．これらは通常黒ピクセル (B のメンバ) である．
- 目的は、指定した入口出口を持つ、横 C_{wd} マス、縦 C_{hg} マスの、良い迷路を構成することである．その正解路 P は、集合として B とほぼ等しいことが求められる．

2.2 絵画的迷路の質

前節の定義には「良い迷路」「ほぼ等しい」といった曖昧な言葉が含まれているが、これは絵画的迷路の質の評価には多くの観点がありえ、状況によって使い分けていく必要があると考えたためである．正解路と絵の食い違いピクセル数 $|(P \cap \bar{B}) \cup (\bar{P} \cap B)|$ というのは明解で有力な評価基準の1つであるが、これを最小化したものがつねに良い絵画的迷路であるとは限らない．たとえば、動物の絵を浮かび上がらせる場合、目の周囲の部分の食い違いは体の部分に比べて強く避けるべきであろうし、文字を浮かび上がらせる場合、c と o などを区別するためにはある部分が接続しているかどうかが重要になるだろう．いずれにせよ、絵画的迷路の見た目の綺麗さという意味では正解路 P の生成法が質を評価する際の中心となる．

一方で、迷路を解く際の難しさというのも迷路の質に関係し、これは主に背景部 $\bar{P}$ の生成法によって決まる．一般に、行き止まりの数が多すぎる迷路、つまり結果的に行き止まりまでの道のりが短すぎる迷路というのは簡単でつまらない迷路とされる．Okamoto らは単純な手法で背景部を生成したあと、行き止まりの数を減らすヒューリスティッ

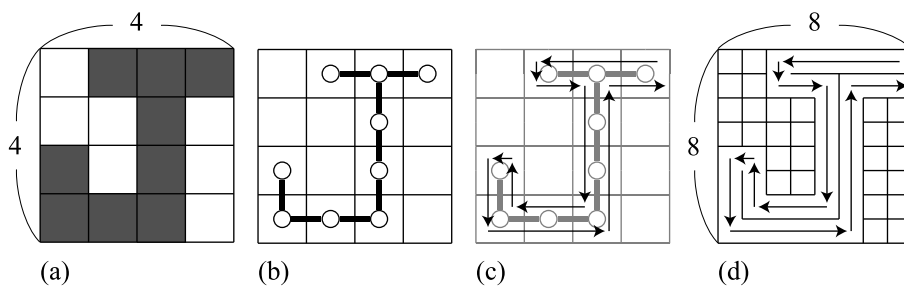


図 2 Okamoto らの手法
Fig. 2 Okamoto et al.'s algorithm.

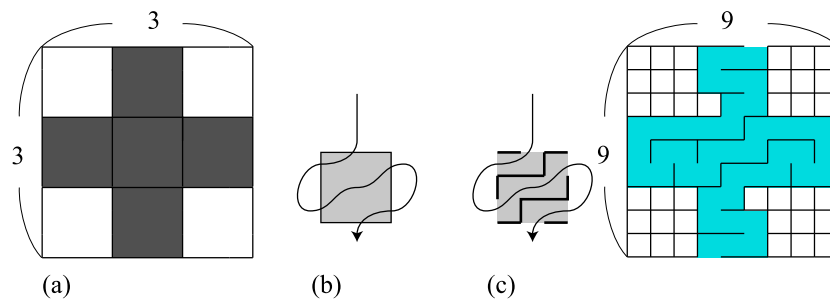


図 3 (a) 2×2 マスへの変換ではうまくいかない例, (b) その理由, (c) 池田の手法
Fig. 3 (a) An example the 2×2 methods fails, (b) the reason, (c) Ikeda's solution.

クを用いることで難しさという意味での質の向上を図っている [3].

本論文では、まず 3 章から 5 章まで正解路 P の生成法のみに着目して議論を進め、6 章ではじめて背景部 $\bar{P}$ の生成法について述べることにする。この 2 つは独立して動作し、どちらかだけを既存手法と組み合わせて使うことも可能であることに注意されたい。

3. 既存の正解路構成法

1 章で述べたように、絵画的迷路の構成法についての学術的論文は数少ないが、Okamoto らのもの [3] をきっかけとして近年研究がさかんになっている [7], [8], [9]。本章では、その正解路 P の生成法のみを抜き出して概要を述べ、特徴や課題を整理する。

3.1 Okamoto らの手法

図 2 を用いて Okamoto らの手法の概要を説明する。白黒 2 値画像が与えられる (図 2(a))。黒ピクセルは連結しており、また入口出口の指定はない。絵の部分 (黒ピクセル) を 1 つのノードとして、それらを結ぶ全域木を構成する (図 2(b))。この例では構成法は一意であるが、一般には複数あり、そのいずれでもよい。続いて、この全域木を伝う走査路を作成する (図 2(c))。最後に、元の画像の 1 ピクセルを 2×2 マスに置き換えたうえで、走査路が正解路となるように壁を取り除く (図 2(d))。

本手法はアルゴリズムとして華麗であり、また実装が容易で高速であり、与えられた絵と正解路にも食い違いが生

じない。一方で元の画像の 1 ピクセルが 2×2 マスになるという縮尺の変更が必要であり、これはいいかえれば、縮尺しないなら、もとの白黒 2 値画像で $(2x, 2y)$, $(2x+1, 2y)$, $(2x, 2y+1)$, $(2x+1, 2y+1)$ の 4 ピクセルがすべて同じ色であることを要求するということである。さらに、迷路としての面白さという点でいうと、入口と出口が隣り合っている必要があり、左手法 (入口から見て出口側の壁を伝って進む方法) を用いれば不正解部分に立ち入らずに無駄なく解けてしまうという大きな課題があった。

3.2 いくつかの発展手法

Okamoto らの手法を受け、簡単に解けるという課題を解決すべく、入口出口を指定できるような手法がいくつか提案された。入口出口を指定した場合、図 3(a) のような画像は、1 ピクセルを 2×2 マスにすることでは正解路を構成することができない。これは、1 ピクセルに 3 回の出入りが要求されるためである (図 3(b))。

池田は、1 ピクセルを 3×3 マスにすることでこの問題を解決しようとした [7] (図 3(c)) が、この手法では正解路がすべての必要なマスを含めることができない場合があるため、絵と正解路は若干食い違う。中井らは 1 ピクセルを 4×4 マスにすることで、絵と正解路を完全に一致させながら入口出口の指定を可能にしている [8] が、現実的な迷路の大きさと絵を綺麗に見せるのは 2×2 マスの場合よりもいっそう難しいといわざるをえない。濱田は、与えられた画像が 2-枝連結という条件を満たしている場合、 2×2 マスへの縮尺変換でも入口出口を指定できるアルゴリズム

を示した [9] が、この条件は人間にとって分かりにくく、絵画的迷路生成を支援するソフトウェアを作ること考えた際には若干不利な点であろう。

いずれにせよこれらの手法はすべて縮尺の変更を前提としており、本論文で提案する手法とはその点で大きく異なる。

4. Simulated Annealing を用いた正解路の生成

4.1 最適化問題への定式化

本論文では、縮尺の変更を行わずに、与えられた絵 B と正解路 P の食い違いをできるだけ軽度にするような最適化を行う。最適化問題は、探索変数・目的関数・制約条件の3要素からなるが、背景部分（不正解部分）の生成は別に考えるので、探索変数としては正解路 $P = \{p_i\}$ 、目的関数（評価値） $f(P)$ としてはその質だけを考えればよい。制約条件は、 P が入口出口を端点に持ち、重複がないことである。

B と正解路 P の食い違い部分のピクセル集合は $M_P = (P \cap \bar{B}) \cup (\bar{P} \cap B)$ と書ける。食い違いの個数 $|M_P|$ は目的関数として有力であるが、2.2 節で述べたようにピクセルごとに食い違いの影響度は異なるであろう。本論文では、ピクセル p の重み w_p を利用者が設定できるものとして、食い違いを重み付き加算した次式を目的関数として最小化することにする。

$$f(P) = \sum_{p \in M_P} w_p$$

具体的には、たとえば動物の目などを「きっちり描き分けてほしい箇所」として、あるいは絵 B から遠い部分を「できるだけ避けてほしい箇所」として、重要度（ペナルティ） w_p を大きくするなどの利用法が考えられる。また、絵の内容や見る人によって、輪郭が崩れることと絵の内部でピクセル抜けが生じることのどちらを重視するかは異なるため、好みに応じた制御も可能になる。このことについてはアンケート調査に関係して後述する。

4.2 Simulated Annealing を用いた最適化

本最適化問題の探索空間は非常に大きく、分枝限定法などの厳密解法を用いるのは適切でないと考える。確率的最適化手法は、最適解を得る保証こそないものの、厳密解法よりも短い時間で高質な解を得ることができるとされ、多くの問題で良好な性能が示されている [2], [11]。遺伝的アルゴリズム (Genetic Algorithm) はその中でも最も高質の解を得られるとされるものであるが、2つの解の情報を交換する交叉オペレータの生成が困難であると考え、より単純な、Simulated Annealing (SA, 焼きなまし法) を用いることとした。

本論文で利用する Simulated Annealing は、開始温度

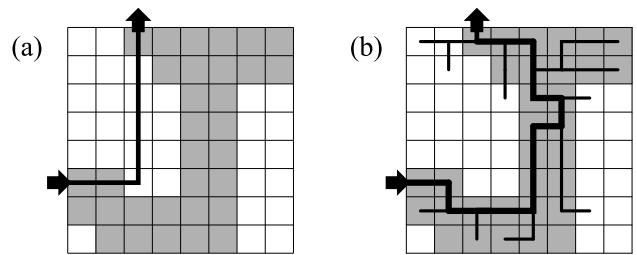


図 4 初期解の生成法. (a) 単純な方法, (b) 提案手法. 提案手法では出口からできるだけ絵の部分に枝を伸ばしてゆき、入口が見つかるまで続ける

Fig. 4 Initial path generation. (a) simple method, (b) our method which tries to expand path to the foreground as much as possible.

T_{start} , 減衰率 T_{decay} , 最終温度 T_{end} の3つのパラメータを持ち、以下の手順で行われる。

- (1) 制約を満たした初期解 P を1つ作成する。
- (2) 温度を $T = T_{\text{start}}$ に設定する。
- (3) 現在の解 P を少し変更した解 P' を生成する。
- (4) 評価値の改善量 $\text{gain} = f(P) - f(P')$ を計算する。
- (5) $\text{gain} \geq 0$ のとき、すなわち改悪していないとき、 P を P' で置き換え、(7)に進む。
- (6) $\text{gain} < 0$ のとき、 $e^{-\text{gain}/T}$ の確率で P を P' で置き換える。
- (7) 温度 T を T_{decay} 倍し、 $T > T_{\text{end}}$ なら(3)に戻る。そうでないなら終了する。

ステップ(6)がSAの特徴的な手順であり、温度 T が高い場合は $\text{gain} < 0$ であっても比較的高い確率で P を P' で置き換える、すなわち変更した解（近傍解と呼ぶ）を受諾する。温度は徐々に低くなり、改悪するような変更が低確率でしか保存されないようになる。このような挙動をすることでSAは局所解を乗り越え、局所探索法と同程度の単純さのまま質の高い解を得られるとされる [11]。なお温度が0のまま一定である場合にはSAは局所探索法と一致する。

実際にSAを構成するためには、パラメータの決定のほかに、ステップ(1)での初期解の生成法と、ステップ(3)での近傍解の生成法を決めなければならない。次節以降でその方法を説明する。

4.3 初期解の生成法

本論文の定式化によれば、解 $P = \{p_i\}$ の制約条件から、第1要素 p_1 を入口、最終要素 p_n を出口とするような重複のない配列をまず最初に1つ作る必要がある。解の質を気にしなければどこを通ってもよいので、その作業は容易である (図4(a))。しかし、できるだけ絵の部分を通るような経路を初期解とすることには意味があると思われるため、パラメータ α_{select} を用いて全域木を作るのと似た下記のようなアルゴリズムで経路を生成することとする。その

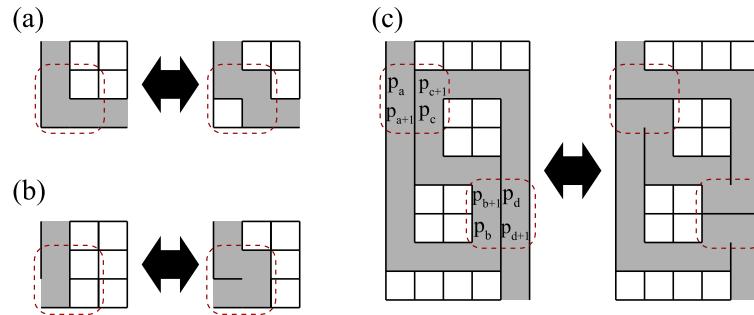


図 5 近傍解の生成法. (a) ずらす機能, (b) 増減する機能, (c) 訪問順を変更する機能

Fig. 5 Neighborhoods. (a) Flip, (b) Expand/Contract, (c) Double bridge.

動作イメージを図 4(b) に示す. なお本アルゴリズムは入力画像が非連結な場合でも適用可能であり, 必ず停止して初期解を生成できる.

- (1) 集合 G を出口ピクセルのみを持つ集合とする.
- (2) G のピクセル p' に接するピクセル $p \notin G$ を 1 つランダムに取り出す.
- (3) p が絵の部分でない場合つまり $p \in \bar{B}$ の場合, $\alpha_{\text{select}} w_p > 1$ なら, $1 - 1/\alpha_{\text{select}} w_p$ の確率で (2) に戻る.
- (4) p' を, p の親ピクセル $\text{parent}(p)$ として記録し, G に p を加える.
- (5) p が入口ピクセルでないなら, (2) に戻る. 入口ピクセルなら (6) に進む.
- (6) p から親をたどって出口ピクセルに至るまでの配列 $\{p, \text{parent}(p), \text{parent}(\text{parent}(p)), \dots\}$ を初期解とする.

4.4 近傍解の生成法

SA では, 現在保持する比較的良好な解を劇的に変更することは多くの場合適切ではない. 近傍オペレータは, 解の一部だけを変更し, ただしそれらの組合せで複雑あるいは広域の変更が可能ないように設計する必要がある. 本論文では, 「正解路をずらす」「正解路を増減する」「大域的な訪問順を変更する」という機能を分担する 3 つのオペレータを提案する. これらのオペレータはすべて双方向的で逆操作が可能である.

正解路をずらす機能は図 5(a) に示す近傍オペレータ (フリップと呼ぶ) が担う. これは, 連続する 3 つの正解路 p_i, p_{i+1}, p_{i+2} が 2×2 のマスの中にあり, 残り 1 つのマス p' が正解路でない場合に使うことができ, p_{i+1} の代わりに p' を正解路とする処理が行われる. この近傍のオペレータでは, 正解路は長くも短くもならない.

正解路を増減する機能は図 5(b) に示す近傍オペレータが担う. 連続する 4 つの正解路 $p_i, p_{i+1}, p_{i+2}, p_{i+3}$ が 2×2 のマスの中にある場合に使うことができ, p_{i+1} と p_{i+2} を正解路から削除することで減らす処理が行える. 増やす処理はこのちょうど逆である.

大域的な訪問順を変更する機能は図 5(c) に示す近傍オペレータ (ダブルブリッジと呼ぶ) が担う. これは, ある $a < b < c < d$ に対し, 正解路中の 4 点 $p_a, p_{a+1}, p_c, p_{c+1}$, および別の 4 点 $p_b, p_{b+1}, p_d, p_{d+1}$ がそれぞれ 2×2 のマスの中にあるような場合に使うことができ, p_{a+1} から p_b に至るまでの部分を p_{c+1} から p_d に至るまでと交換する処理が行われる. このように書くとき非常に大きな変更に見えるが, 図に示すとおり, 2×2 マスの部分の通路のつながり方を縦横切り替えるだけの処理であり, 評価値も変わらない. 評価値は変わらないものの, 本オペレータを用いることで解探索の自由度が増し, 結果として性能も向上することを後述する実験で示す. なお本オペレータは, やはり経路 (ハミルトン閉路) を探索することが必要である巡回セールスマン問題でしばしば用いられるものでもある [5]. 本オペレータは他の近傍オペレータに比べ計算量が大きいため, 実験では 1/10 の頻度で試みることにする.

5. 最適化実験

本章では, 前章で提案した正解路構成法についてその挙動を確かめるいくつかの実験を行う. 主に用いる問題図 (40×30 ピクセル) を図 6 右に示す. 元のシルエット (同左) は web 上のページ [6] から取得した. 開始温度 $T_{\text{start}} = 10.0$, 減衰率 $T_{\text{decay}} = 0.99995$, 最終温度 $T_{\text{end}} = 0.1$, 初期解生成時パラメータ $\alpha_{\text{select}} = 10.0$ とした. この温度設定の場合約 9 万回候補が生成され, 要する時間は標準的な PC で数秒である. 黒ピクセルの重みは 1.0, 黒ピクセルに接する白ピクセルの重みは 2.0, 接しない白ピクセルの重みは 100.0 とし, 大きなはみ出しを防ぐ設定とした.

5.1 収束の様子を見る実験

図 7 左は, 異なる乱数種を用いて 20 回試行した場合の, 温度を横軸, そこまでに発見した最良評価値を縦軸にとった収束曲線である. 横軸は対数のグラフになっているが, 温度は一定比率で減衰するため, 候補生成数という意味では通常のグラフである. 収束の様子を見ると, おおむね

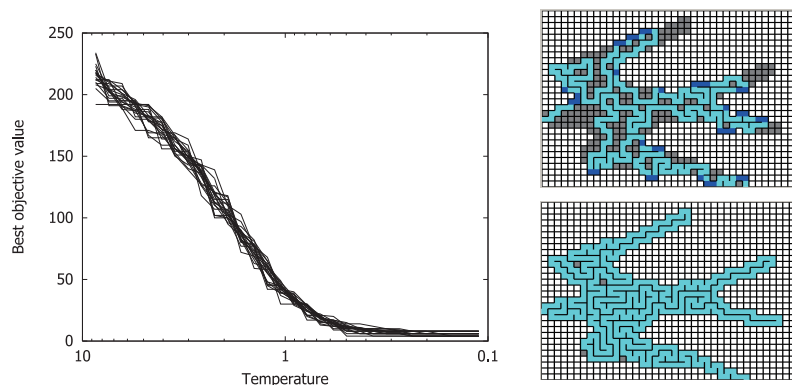


図 7 左, SA の収束の様子. 右上: 温度 5.0 のときのある解. 右下: 評価値 4 の解. 少し濃い点は経路にならないほうが望ましい箇所

Fig. 7 Error convergence (left), a solution at $T = 5.0$ (right-top), a solution whose evaluation value is 4 (right-bottom).

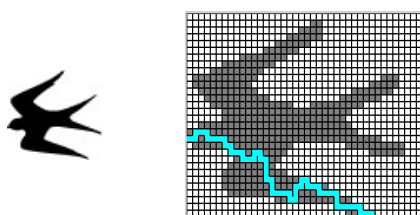


図 6 問題図. 左は利用したシルエット, 右はそれを 40×30 にしたものとの初期解

Fig. 6 Original silhouette (left), input image (40×30) and an initial path (right).

どの試行も同様の軌跡をたどり, 最終的な評価値は 4, 6, 8 のいずれかに収束していることが分かる. 図右上は温度 5.0 のとき (およそ 14,000 回候補生成後) の様子であり, まだはみ出し・欠損が非常に多い. 図右下は評価値が 4 になったときの様子であり, これはおそらく最適解 (の 1 つ) である. 最終の評価値の平均は 5.90, 標準偏差は 1.61 であり, 必ず最適解が得られるというわけではないものの, 実用に耐える良好な性能であると考えられる.

5.2 各提案手法の効果を見る実験

前節と同じ問題に対し, 提案手法のいくつかを用いないことで性能がどのように変化するかを調べる. 各々 20 回試行し, 調査項目は評価値の平均・標準偏差・最良評価値・最悪評価値の 4 つとする. 比較するのは以下の 5 つである.

- (a) 標準的な手法
- (b) 近傍の 1 つ, フリップを使わない場合
- (c) 近傍の 1 つ, ダブルブリッジを使わない場合
- (d) すべてのピクセルの重みを 1.0 とした場合
- (e) $\alpha_{\text{select}} = 0$ として, 絵の上とは限らず適当な初期解を用いた場合

結果は表 1 のとおりである. すべての場合で, 標準的な手法に比べて評価値がはっきり劣っていることが分かる. なお (d) ではすべてのピクセルの重みを 1.0 としているた

表 1 提案手法を使う場合と使わない場合の性能比較. なお最後のケースでは評価値ではなく食い違い数を評価した

Table 1 Effects of proposed methods. a–d: averaged error, e: averaged number of mismatches.

手法	評価値平均	標準偏差	最良	最悪
a (標準)	5.90	1.61	4	8
b	32.1	6.72	20	44
c	8.85	1.65	6	12
d	31.2	7.49	22	44
e	(9.00)	(4.27)	4	(18)

め, 食い違いが小さく評価される傾向にあり, それでも (a) に劣っているという点に留意されたい. その意味で, (d) と他の (b)(c) の評価値を比較することにはあまり意味がない.

(b) フリップ近傍を使わない場合の解の例を図 8 左に, (d) すべてのピクセルの重みを 1.0 とした場合の例を図右に示す. これらは評価値としては偶然似通ったものの, 傾向としては大きく異なる. (b) の場合, 大まかな形には問題ないものの, シルエット中の欠損を移動させて解消することが困難になっている. 一方 (d) または (e) の場合, 絵のくぼみの部分をまたぐような経路 (橋と呼ぶ. 図右の濃い部分) が生じることがしばしばある. このような経路は, 4.4 節で提案したようなオペレータでは解消することが非常に困難であり, これを生じさせないような工夫として, ピクセルの重み付け (4.2 節), できるだけ絵の上をたどる初期解生成 (4.3 節) を行っているのである.

5.3 スケーラビリティに関する実験

絵画的迷路の実用上は, 数分からせいぜい 10 分ほどで解けるサイズの迷路が中心になるだろうが, アルゴリズムが大きなサイズの迷路の場合でも破綻しないかどうかを確認しておくことは重要である. 本節では, 同じツバメの絵をもとに, 40×30 , 80×60 , 120×90 , 160×120 ピクセル

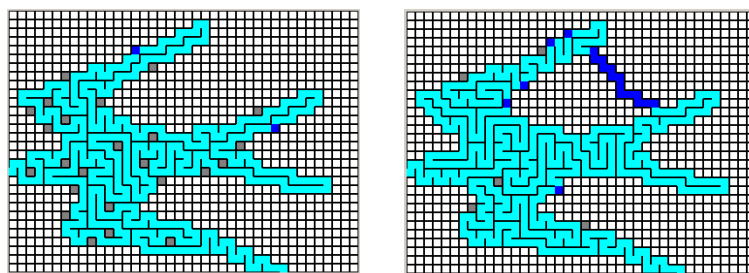


図 8 左, フリップを使わない場合の解, 右, 橋がかかる失敗

Fig. 8 Solution without flip operator (left), failure called bridge (right).

表 2 入力画像サイズと減衰率を変化させた場合の評価値. () 内は 1 ピクセルあたりのエラーの大きさ

Table 2 Averaged errors in 10 trials. Error per pixel is denoted in ().

サイズ	$T_{\text{decay}} = 0.999$	0.9999	0.99995	0.99998
40 × 30	40.0(0.0333)	29.9(0.0249)	28.2 (0.0235)	-
80 × 60	178.4(0.0371)	44.4(0.0093)	37.8 (0.0079)	32.1(0.0069)
120 × 90	1085.0(0.100)	81.8(0.0076)	57.5 (0.0053)	37.2(0.0034)
160 × 120	3063.0(0.160)	132.9(0.0069)	109.6 (0.0057)	57.5(0.0030)

の絵を B として用意し, これを減衰パラメータ T_{decay} の異なる SA で解かせることで, アルゴリズムの安定性を確認する. なおここで用いる絵は, 元画像を機械的に規定サイズの白黒 2 値画像として変換したものであり, 細すぎる部分や尖っている部分など, 迷路の正解路になりえないような部分もそのまま残している.

表 2 に, 各サイズ・各減衰率での評価値の 10 試行平均と, 1 ピクセルあたりのエラーの大きさ (平均評価値 / 総ピクセル数) を示す. 最も小さいサイズでは $T_{\text{decay}} = 0.9999$ でも十分であるが, サイズが大きくなるにつれ, よりゆっくりとした減衰が必要になることが分かる. 最も大きいサイズでは $T_{\text{decay}} = 0.99995$ と 0.99998 でも評価値に 2 倍近い違いが見られ, $T_{\text{decay}} = 0.99998$ の場合でも実行時間は 30 秒ほどであることから, さらにゆっくりとした減衰も有効であろうと推測できる. なお, 57.5 という平均値は満足のゆくものではなく, 実際, 48 という評価値が確認されている. 複数の SA の試行を別プロセスに割り当て並列実行するなどの工夫でばらつきを抑えることが実用上は有用だろう.

6. 不正解部分の構成法

前章までは, 与えられた絵 B とほぼ等しくなるように正解路 P を定める方法について述べた. 本章では, 正解路の確定後の, 不正解部分 $\bar{P}$ の構成法について, 既存手法 [3] を拡張した手法を提案する.

不正解部分 $\bar{P}$ は, 正解部 P からの脇道 (あるいは枝分かれ) で構成される. Okamoto らの手法は以下のとおりである.

(1) 正解路が与えられる (図 9(a)). P を 1 本の接続するグラフと見立て, $\bar{P}$ に含まれるマスをもノードと見立て

る. すべてのノードを含む全域木を構成し, エッジ部と交叉する壁を取り除く (図 9(b)).

(2) 隣接する 2 つの行き止まりマスがあれば, 片方の行き止まりを派生させている分岐部分に壁を形成し, 一方その 2 つのマスを隔てる壁を取り除く. これにより行き止まりの数が 1 つ減る (図 9(c)).

(3) 隣接する行き止まりがなくなるまで繰り返す (図 9(d)).

ステップ (1) で生成した迷路には短い行き止まりが非常に多い. このことは, 正解路部分と一見して様相が違うことが分かってしまうこと, 行き止まりを容易に回避することができてしまうことにつながり, 結果として迷路の面白さを減ずる結果になる. ステップ (2) の操作は行き止まりを減らすことでこれを解決しようとしている.

しかしこの手法は, 隣接する 2 つの行き止まりに対してのみに行われるため, まだ行き止まりが多い状況であっても終了してしまう場合がある (図 10(a)). そこで本論文では, 隣接する行き止まりがない場合でも行える次の手法を提案する.

(1) ステップ (1) は既存手法と同様.

(2) 行き止まりマス p を 1 つ選ぶ. 最大操作回数 $l \in [l_{\min}, l_{\max}]$ を定める.

(3) p の周囲には, 移動できる唯一のマス p' と, 壁が 3 枚あるはずである. 迷路外周以外の壁 w を 1 つ選ぶ.

(4) 壁 w を取り除き, p と p' の間に壁を設置する (図 10(b)). l を 1 減ずる.

(5) p' の周囲に移動できるマスが 2 つ以上ある場合 (図 10(c)), または $l = 0$ となった場合, ステップ (7) に進む.

(6) p に p' を代入し, w に先ほど設置した壁を代入して, ステップ (4) に戻る.

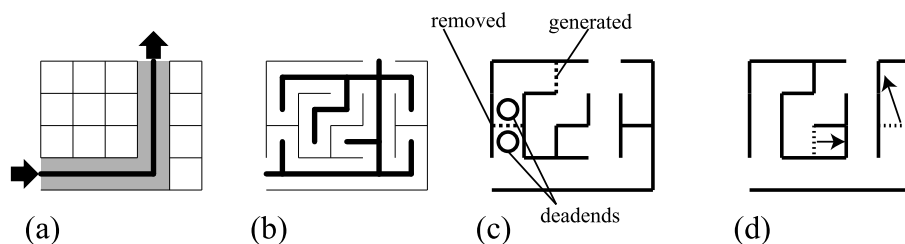


図 9 不正解部分構成の既存手法. (a) 正解路確定時点, (b) 全域木による不正解部の構成, (c) 行き止まりを減らす操作, (d) 完成図

Fig. 9 Conventional generation method of the background, (a) input, (b) temporal generation, (c) heuristic improvement, (d) output.

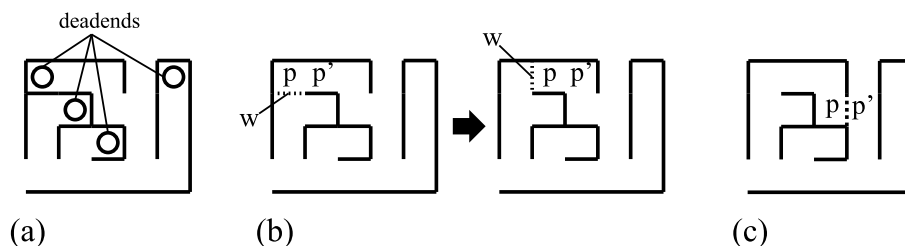


図 10 不正解部分構成の提案手法. (a) 隣接する行き止まりがない場合, (b) 壁の除去と設置, (c) 終了条件

Fig. 10 Proposed generation method of the background.

表 3 複数の手法による, 行き止まり数と平均長さ, 長さ 10~19 および 20 以上の行き止まり数. 10 試行平均

Table 3 Number of deadends and the averaged length to deadends (10 trials).

手法	deadend 数	平均長さ	長さ 10~19	長さ 20 以上
全域木作成時点	309.7	0.72	0.0	0.0
既存手法	134.1	4.00	10.3	1.7
提案手法 1 $l \in [10, 20]$	59.4	11.7	39.6	4.8
提案手法 2 $l \in [35, 70]$	31.3	24.5	4.8	22.1

(7) ステップ (2) に戻る. 十分な回数繰り返したら終了する.

本手法のステップ (2) からステップ (6) を 1 回行ったあとは, 行き止まり数に関して以下の 3 つの場合がある.

- 減る. 隣接する 2 つの行き止まりの片方を p として選び, 隔てる壁を w として選んだ場合.
- 増える. 選んだ行き止まり p が派生する分岐部までが l 以上だった場合.
- 変化しない.

1 回の操作で行き止まり数が変化しなくても, 行き止まりの箇所が移るため, 隣接する行き止まりが発生しやすくなり, 結果的にはより効果的に行き止まり数を減らすことができるだろうというのが本手法の思想である. これは, 4.4 節で述べたダブルブリッジ近傍の効果にも通じるところがある.

パラメータ $l_{\min}$, $l_{\max}$ は, 分岐から行き止まりまでの長さ (マス数) を適切な値に制御しようとするものである. $l_{\min}$ を大きくすれば行き止まり数が増えることを防ぐこと

ができるが, あまりに行き止まりまでの長さが長いと, 分岐の数が減ることになり, これも迷路の面白さを減ずる要因となりうるからである.

表 3 に, 既存手法および提案手法を用いた場合の行き止まり数と平均長さを示す. 用いた図は 5.1 節のものである. 既存手法は, 行き止まりを 4 割程度まで削減できているが, 平均長さは 4 と十分ではない. 提案手法は, パラメータに応じて, 行き止まり数を削減, 長さを制御できていることが分かる.

7. アンケート調査

本章では, 絵画的迷路の綺麗さ・面白さに関する予備的なアンケート調査の結果を報告する. 図 11 はそのアンケート用紙である. 実際には順番を変えた 3 通りのアンケート用紙をランダムに配布している. 回収数は 18 (右側は 17) である.

左側は, 正解路部分のみに着目してもらう. 上から, (a) 2×2 のギザギザのある絵を与えたもの, (b) 絵の内部にピクセル欠けが生じにくいように重み w_p を調整したもの,

【絵画的迷路についてのアンケート】

(123タイプ)

問1：純粋にシルエットの美しさで見たとき、以下の3つを5段階で評価してください。

問2：下記は正解部は同じです。迷路としての面白さをそれぞれ5段階で評価してください。

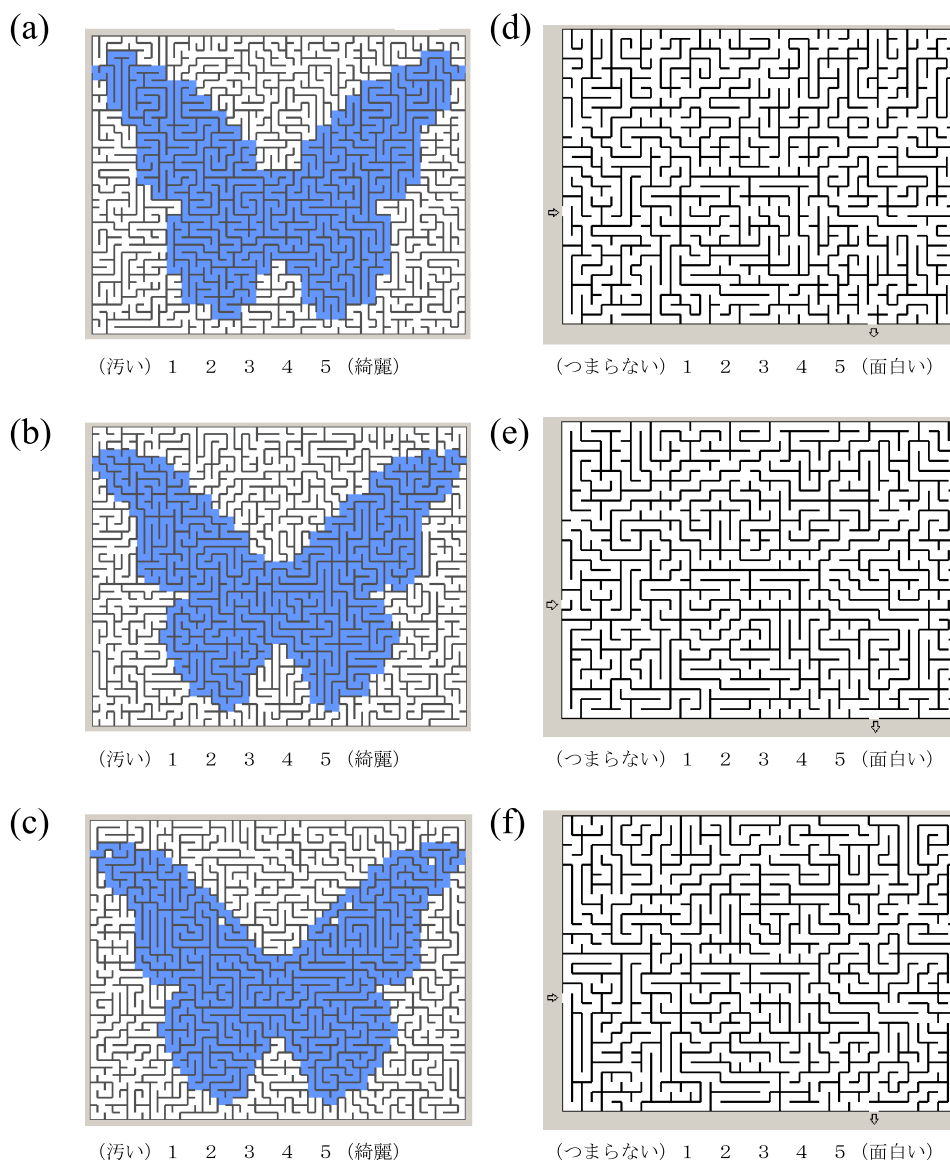


図 11 アンケート用紙の1つ。左が綺麗さ、右が面白さを問うもの

Fig. 11 Questionnaire sheet.

(c) 絵の輪郭に食い違いが生じにくいように重みを調整したものである。アンケート評価の平均点はそれぞれ (a) 2.89, (b) 3.56, (c) 3.94 であり、食い違いを許容して縮尺を変更しないという提案手法の有効性が示唆された。しかし (a) を最も良いとした人も 18 人中 4 人おり、食い違いによる違和感があるくらいならギザギザのほうが良い、という嗜好もあることが分かった。

右側は、正解路部分は固定で、上から (d) 既存手法で不正解部分を生成したもの、(e) 前章の提案手法 2 で生成したもの、(f) 前章の提案手法 1 で生成したもの、である。アンケート評価の平均点はそれぞれ (d) 3.18, (e) 3.12, (f) 3.12

であり、有意な差はみられなかった。しかしやはり、(e)(f) を最も良いとする人が 17 人中それぞれ 4 人おり、2 点以上の差をつける人も珍しくないことから、ここにも好みの差が見られる。いずれより詳細で大規模な調査を行い、手法の発展につなげたいと考えている。

8. おわりに

絵画的迷路は、迷路の中でも芸術性が高く、子供受けもしやすいものの 1 つであり、その自動生成アルゴリズムが研究・公開されれば保育所など社会への貢献も大きいと予想される。本論文では、まず正解路部分の生成について、

縮尺を変更せずに絵と正解路の食い違いを最小化するような最適化問題を定式化し, Simulated Annealing (SA) で比較的短時間に高質の解を得ることができることを示した. この手法ではピクセルの重要度を指定できることも強みであり, それらの効果を実験およびアンケート調査によって示した. さらに不正解部分についても行き止まり数や行き止まりまでの長さを制御するための手法を提案し, 利用者の好みを反映しやすいようにすることができた. 本手法は, 縮尺の変更がないこと, 入口出口の指定が可能なのほか, 非連結な画像・穴がある画像に対しても適用可能であり, 利用者にとって利便性が高いと考える. アルゴリズムとしての華麗さはないが, 実用的な絵画的迷路生成支援ソフトウェアを作るための嚆矢となったのではないかと考えている. 今後は, ソフトウェア公開により特に子供や保護者からのフィードバックを得て, 学術的および実用的な改善を図っていきたい.

参考文献

- [1] Conceptis Puzzles: Picture This! Mazes, Sterling, New York (2005).
- [2] Nagata, Y. and Kobayashi, S.: Edge Assembly Crossover: A High-power Genetic Algorithm for the Traveling Salesman Problem, *Proc. 7th International Conference on Genetic Algorithms*, pp.450–457 (1997).
- [3] Okamoto, Y. and Uehara, R.: How to Make a Picturesque Maze, *Proc. Canadian Conference on Computational Geometry (CCCG2009)*, pp.137–140 (2009).
- [4] Pullen, W.D.: Think Labyrinth! (online), available from <http://www.astrolog.org/labyrnth.htm> (accessed 2011-08-01).
- [5] Reinelt, R.: The Traveling Salesman: Computational Solutions for TSP Applications, *Lecture Notes in Computer Science*, Vol.840, Springer-Verlag (1994).
- [6] TopeconHeroes: Silhouette Design (online), available from <http://kage-design.com/> (accessed 2011-09-01).
- [7] 池田 心: 絵画的迷路生成のある拡張, 組合せゲーム・パズルミニプロジェクト第5回ミニ研究集会 (2010).
- [8] 中井亮平, 岡本吉央: 絵画的迷路生成アルゴリズムの改善, 京都大学数理解析研究所講究録, Vol.1691, pp.162–166 (2010).
- [9] 濱田浩気: 始点と終点を指定した絵画的迷路の作り方, 組合せゲーム・パズルミニプロジェクト第5回ミニ研究集会 (2010).
- [10] 望月士郎: 浮き出し迷路, Vol.1, 学研 (2006).
- [11] 山田武士, 中野良平: 確率的探索と確定的探索の組合せによるジョブショップスケジューリング問題の解法, 情報処理学会誌, Vol.37, No.4, pp.597–604 (1996).



池田 心 (正会員)

1975 年生. 1999 年東京大学理学部数学科卒業, 2000 年東京工業大学大学院総合理工学研究科知能システム科学専攻博士前期課程修了, 2003 年同博士後期課程修了, 博士 (工学). 同年京都大学学術情報メディアセンター助手, 2010 年北陸先端科学技術大学院大学准教授. ゲーム, 進化計算, エージェントシミュレーション, パズル等の研究に従事. 計測自動制御学会, 進化計算学会各会員.



橋本 隼一

1981 年生. 2005 年東京工業大学卒業, 2007 年北陸先端科学技術大学院大学情報科学研究科博士前期課程修了, 2011 年同博士後期課程修了, 博士 (工学). ゲーム AI, 木探索の研究に従事.