深層強化学習に基づく自動ネスティング手法

正会員 谷 口 智 之* 正会員 平 方 勝*
正会員 万 順 一**

Algorithm Based on Deep Reinforcement Learning for Irregular Shape Nesting Problem

by Tomoyuki Taniguchi, *Member* Masaru Hirakata, *Member*
Junichi Man, *Member*

Summary

An automatic nesting algorithm based on deep reinforcement learning for shipbuilding is proposed. Automatic nesting method is desired because of the large number of parts to be placed. The nesting problem in shipbuilding is very difficult because of the relatively small area where they can be placed and the variety of shapes that can rotate freely. Since it is very difficult to manually create effective placement rules, this paper proposes an algorithm to generate rules autonomously based on reinforcement learning. To apply reinforcement learning to the nesting problem, we organized the problem as a Markov chain process. Based on deep q network, a type of reinforcement learning, we used the components of a real ship's block to learn the network parameters. The parts are represented in pixel format. It is confirmed that the present method was superior to the conventional method, and the results were comparable to those of a skilled person. However, for unlearned members, the results are inferior to those of the conventional method. This problem can be solved by relearning including unlearned components.

1. 緒 言

船体を構成する様々な形状の部材は、主に長方形の鋼材から切り出されるが、複数の部材をできるだけ廃棄する面積が少なくなるよう、歩留まり率が良くなるように配置すること（以下、ネスティングと呼ぶ）が望まれる。造船の場合、部材点数が多いため、これらすべてを手作業で、かつ歩留まり率が良くなるように配置することは容易ではない。

ネスティング作業の自動化に関する研究は多く実施されている。例えば、Burkeらは配置ルールを記述した Bottom Left Fill (BLF)法¹⁾などがある。しかしながら、造船分野において、既存の自動ネスティングシステムによって出力された結果の一部もしくはすべてに対し、人手による修正が行われているのが現状である。これは、造船でのネスティング問題には、配置可能領域が部材の大きさに対して相対的に狭いこと、多様な部材が存在し、部材の自由回転が許容され、設計自由度が高いことなど造船特有の問題が存在するため

ある。一方で、このように多様な部材で構成される配置問題に対して有効な配置ルールを人手で設計することは非常に困難である。

そこで、本報では、造船分野におけるネスティング問題に対して、強化学習を用いて自律的に配置ルールを構築し、歩留まり率の良い配置を出力するネスティング手法を提案する。強化学習とは、長期的な目標を示す数値を最大化するように学習する手法であり、行動主体であるエージェントと、報酬および状態を出力する環境の2つで構成され、この報酬につながる行動方針を学ぶ。提案手法では、この報酬を歩留まり率とし、状態を部材の配置状況、行動を部材の配置位置と角度の組合せとして強化学習の枠組みを適用する。また、部材の形状や配置状況は画像のピクセルデータとして表現し、畳み込みニューラルネットワーク（CNN）を活用して部材形状を考慮する。具体的には、実際の船を構成する部材形状データを用いて、様々なネスティング問題を生成し、試行錯誤で最適な配置ルールを学習する。既存手法および人手による配置結果と比較することで、提案手法の有効性を検証する。

以下、2章では関連研究と本報の違いについて述べ、3章では、ネスティング問題の定式化について説明する。4章では、強化学習の枠組みでネスティング問題を捉え、その具体的な計算手順について説明する。5章では提案法と既存手法とを数値計算結果から比較・検討し、6章にてまとめとする。

* 国立研究開発法人 海上・港湾・航空技術研究所
海上技術安全研究所

** 株式会社大島造船所

2. 関連研究

自動ネスティングに関連する既存研究として、ヒューリスティックとメタヒューリスティックの大きく 2 つに分類される。まず、ヒューリスティックな手法として、Burke らは長方形の詰込み問題に対して、隙間を埋めるのに最も適した部材を選択し配置する Best Fit 法²⁾や、ある配置順に基づき最も下かつ左に部材を順次配置していく Bottom Left Fill (BLF)法¹⁾を提案しており、複雑な形状部材の配置問題に対して適用できるように拡張している。また、Umetani ら³⁾は多角形状の部材に対して、配置母材の長さを順次小さくする段階を設け、各段階で部材が重ならない解を求めていくことで最終的に歩留まり率を改善させる手法を提案しており、その有効性を確認している。これらの手法は広い範囲内に多数の部材を配置する問題や角度が 90° や 180° 単位に限定した問題に対しては短時間で良好な解が得られている。ただし、造船のように狭い範囲内に大きな部材を詰め込むような問題にはその有効性は十分には確認されていない。また、Hamada ら⁴⁾は、位置と角度のストリップパッキング問題と部材を選択するビンパッキング問題の 2 つをとらえ、分枝限定法を適用した自動ネスティング手法を開発している。

配置ルールを定めない手法としてメタヒューリスティックな手法を活用した研究例がある。奥本ら⁵⁾は擬似焼きなまし法 (SA) を組立定盤におけるブロック配置問題に適用し、実用的な配置が得られることを示している。山内ら⁶⁾は遺伝的アルゴリズム (GA) を用いた自動ネスティングシステムを開発し、実船舶の部材データで有効性を確認している。高原ら⁷⁾は局所探索、SA、GA、およびタブーサーチ法とランダム法を比較し、それぞれの優位性を確認している。ただし、これらの手法では、探索パラメータの設計が難しく、初期解によって得られる解の良さが変わり再現性に乏しいことや、解の良さと処理時間のバランスなど課題が残る。

処理時間や再現性の観点からヒューリスティックな手法が採用されることが多いが、上述の通り、狭い範囲かつ多様な部材に対して効果的な配置ルールを別途検討する必要がある。しかしながら、1 隻分の部品点数が数万以上あり、造船所ごとに使用する切断機が異なるため、配置可能領域が異なるなど、汎用的かつ効果的なルールを人手で設計することは困難である。そこで、本報では人手で設計するのではなく、各造船所でのネスティングに即した配置ルールを強化学習に基づき自律的に構築する方法について提案する。Mnih ら⁸⁾はネスティング問題ではないが、コンピュータゲームにおいて高スコアを出すルールを自律的に学習する深層強化学習について提案している。強化学習にてルールを構築する学習段階には計算コストがかかるものの、結果を出力するまでの処理時間は短い。本報では、この強化学習の適用範囲をネスティング問題へと拡張し、造船特有のネスティング問題に

適した手法として改良した新たな自動ネスティング手法を提案する。

3. ネスティング問題の定式化

最大幅 $B_{\max}$ 、最大長さ $L_{\max}$ の配置可能領域内に回転可能な任意形状の n 個の部材を配置する問題を取り扱う。配置された部材をすべて含むことができる最小の長方形の面積に対する全部材の総面積の比を歩留まり率と定義し、この歩留まり率が最大化するような配置を求めるネスティング問題を定式化する。

3.1 部材の表現方法

Fig. 1 に示すように点と線のベクトル形式で表現される部材を、0 と 1 のピクセル形式で表現し、部材同士の重なり判定や歩留まり率の計算、および後述する強化学習での入力として用いる。Hu ら⁹⁾はビットマップ形式にて取り扱うことでネスティング問題を効率的に解く手法を提案しており、本報でもこのような概念と同様の方法を導入する。

まず、配置可能領域内に設けられた幅 d の正方形格子を 1 ピクセルとし、ベクトル形式で表される部材は、その領域内部にある格子点に接するすべてのピクセルとして変換する。例えば Fig. 1 の左図に示すように、赤色の格子点では隣接する 4 つのピクセルに 1 が入り、右図にあるように部材内部のすべての格子点において計算してピクセル形式の部材表現を獲得する。また、回転については、ベクトル形式にて図心を中心に回転変換された後にピクセル形式への変換を行い、回転後の部材を表現する。本報では、 0° から 360° の回転角度を角度 Δ_d 毎に $r_{\max}$ 個に分割し、部材 i の角度 Δ におけるピクセルデータを Fig. 1 に示すように $P_i(u_p, v_p, w_p)$ と表す。 u_p, v_p は部材のローカルな座標系、 w_p は離散化された回転角度であり、 P_i は部材の幅 $b \times$ 長さ $l \times r_{\max}$ のマトリックスとなり、値は 0 または 1 となる。また、部材の 0° 方向は単体での歩留まり率が最小となる角度に調整する。

Fig. 1 Conversion of component representation from vector to pixel format.

Fig. 2 Yield rate calculation in pixel format.

3.2 歩留まり率の計算方法

ピクセル形式で表現された部材の配置結果に対して、部材同士の重なり判定と歩留まり率の計算を行う。まず、最大幅 $B_{\max}$ 、最大長さ $L_{\max}$ の配置可能領域を格子幅 d で離散化し、配置可能領域の幅のピクセル数が $b_{\max}$ 、長さのピクセル数が $l_{\max}$ であるとする。ここで、配置可能領域内のあるピクセルの幅方向と長さ方向の位置を (x, y) 、そこでのピクセル値を $I(x, y)$ とすると、部材が配置されていない領域内のピクセルでは $I(x, y) = 0$ 、部材が一つだけ配置された領域内のピクセルでは $I(x, y) = 1$ 、 m 個の部材が重なって配置された領域内のピクセルでは $I(x, y) = m$ と表すことができる。したがって、配置可能領域内の任意の位置 (x, y) において $I(x, y) > 1$ のときは重なりが存在し、配置可能領域内のすべての位置 (x, y) において $I(x, y) \leq 1$ のときは重なりが存在しないと判別できる。また、 n 個の部材が重なりなくすべて配置された場合、 $(x, y) = (1, 1)$ を起点としたすべての部材が含まれる最小の長方形の幅と長さのピクセル数が b, l であるとする、歩留まり率 Y は、次式で定義される。

$$Y = \frac{1}{bl} \sum_{x=1}^b \sum_{y=1}^l I(x, y) \quad (1)$$

例えば、Fig. 2 に示すように、緑色の部材 P_1 と黄色の部材 P_2 が配置されている場合、赤色の破線が最小の長方形となり、歩留まり率は $(19+19)/40=0.95$ となる。ただし、ピクセル形式で表現しているため、ベクトル形式にて算出される歩留まり率とは誤差があるが、適切な格子幅を選択すれば実用上微小であるとみなすことができる。

3.3 最適化問題としての定式化

n 個の部材を配置する場合、配置可能領域内における部材の位置 x_i, y_i と離散化された角度 q_i を設計変数とし、設計変数ベクトル $\mathbf{Z}$ を次のように表す。

$$\mathbf{Z} = \{\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_n\}, \mathbf{z}_i = (x_i, y_i, q_i) \quad (2)$$

ここに、 x_i, y_i は i 番目の部材の図心を代表点とした配置可能領域内での位置を示している。3.2 節で述べた配置可能領域内のピクセル値 I と歩留まり率 Y は $\mathbf{Z}$ の関数であることを考慮して $I(\mathbf{Z}, x, y)$, $Y(\mathbf{Z})$ と表し、すべての部材が重なりなく配置されることを制約条件とすると、ネスティング問題は次式のような最適化問題として定式化される。

$$\left. \begin{array}{l} \text{Find } \mathbf{Z} \\ \text{To maximize } Y(\mathbf{Z}) \\ \text{Subject to } \max \{I(\mathbf{Z}, x, y)\} \leq 1 \end{array} \right\} \quad (3)$$

4. 強化学習によるネスティング問題の解法

3 章で定式化したネスティング問題をマルコフ決定過程 (MDP) として捉えなおし、深層強化学習の枠組みで解く方法について説明する。ネスティングでは部材の形状を認識することが重要であるため、価値関数の近似方法に畳み込みニューラルネットワークを採用し、自律的に部材の形状からネスティングに必要な特徴量を獲得することを期待する。

4.1 マルコフ決定過程としてのネスティング問題

3 章のネスティング問題を n 個の部材の順番を決定する問題と、順番が与えられた上で配置位置を決定する問題との 2 層構造に分解して考える。本報では、前者にメタヒューリスティックな手法を適用し、後者はマルコフ決定過程として定義して深層強化学習を適用する。まず、MDP では状態集合 $S = \{s_1, s_2, \dots\}$ 、行動集合 $A = \{a_1, a_2, \dots\}$ 、状態遷移確率分布 $P(s_b, a_b, s_{b+1})$ 、即時報酬関数 $R(s_b, a_b, s_{b+1})$ で表される。 n 個の部材の順番が与えられ、 i 番目の部材を配置する場合、 i 番目から n 番目までの未配置の部材の形状と、 $i-1$ 番目までに配置された部材の位置と形状の情報を状態 s_{i-1} と表す。 i 番目の部材の配置位置と角度は行動 a_i と表す。このとき、 $i-1$ 番目までに配置された状態 s_{i-1} と行動 a_i から i 番目まで配置された次の状態 s_i へは部材を配置するだけなので、遷移確率 $P(s_{i-1}, a_b, s_i)$ は 1 となる。即時報酬は部材をすべて配置できた場合のみ、その時の歩留まり率を報酬として受け取るものと定義すれば、本報で取り扱うネスティング問題は MDP として定式化される。また、 n 個すべての部材が配置された状態、もしくは i 番目の部材を重なりなく配置できない状態 $s_{i-1, \text{Terminal}}$ に陥った場合は、どちらもそれ以上どのような行動が選択されても変化がない終端状態と見なせる。

4.1.1 状態の定義

状態 s_{i-1} は $i-1$ 番目までに配置された部材と未配置の部材の 2 つで構成される。Fig. 3 に状態 s_i の例を示す。これは、最大 10 個の部材を考慮するが、8 個の部材を配置する場合に、3 つが配置済みで 5 つが未配置であるときの状態 ($i=4$) を表した例である。

Fig. 3 Example of state representation in nesting problems.

$i-1$ 番目までに配置された部材は, Fig. 3 の左から 1 番目のピクセルデータのように部材の種類によらず, 0 と 1 で表現する. 未配置の部材は, Fig. 3 の左から 2 番目以降に最大 n 個の部材ピクセルデータとして表現する. 左から配置順に並べ, 残りはすべて 0 とし, 各部材は配置可能領域の中心に角度 0° で配置する.

以上のように, 状態は最大配置部材数 $n+1$ 枚の画像データで構成される. なお, 説明上, 状態 s_i を配置順 i のみで区別したが, 部材の形状の違いや組合せ, 順番なども変わるため, 状態の数は膨大な数となる.

4.1.2 行動の定義

i 番目の部材の配置位置と角度を行動 a_i と表す. 行動の候補として, 部材の図心における離散化された配置位置と角度の全組合せが考えられるが, 配置位置は, 幅のピクセル数が $b_{\max}$, 長さのピクセル数が $l_{\max}$ であるため, $b_{\max} \times l_{\max}$ 個の配置候補がある. 角度も $r_{\max}$ 個に離散化されているため, 取りうる行動の選択枝は $b_{\max} \times l_{\max} \times r_{\max}$ 個となり, 膨大な数となることが容易に想定され, 学習コストの面で不利となる. そこで, 本報では, 学習の効率性を考慮し, 行動の選択枝を減らすため, 以下 3 つの事前知識に基づくルールを導入する.

(1) 長さ方向 (y 方向) の配置位置は最小値とする.

BLF 法では部材を一つずつ, 配置可能な候補に対して最も下および左に配置 (y が最小および x が最小となるように配置) していく手法である. 本手法でも同様に長さ方向 (y 方向) の配置位置のみ最小値となるように自動決定する. つまり, 幅方向 (x 方向) と角度が決定されると, y 方向の配置位置は自動で決定され, 行動の選択枝は $b_{\max} \times r_{\max}$ 個に削減できる. 特に造船で用いられる母材となる長方形鋼材は縦長であることが多く, 切断機の性能によって最大幅には制限があるため, $b_{\max}$ が極端に大きくなることは考え難い.

(2) 配置可能領域内に部材が収まらない行動を省く.

上記(1)により, 行動は部材の図心位置の幅方向 (x 方向) と角度のみを考慮すればよいが, 部材の角度と形状によっては, 配置可能領域から部材の一部がはみ出るようになる. これは適切な配置ではないため, このような行動の選択枝は事前に省き, 選択されないようにする. また,

行動の選択枝がすべて省かれた場合は終端状態と見なす. 以上により, 行動の選択枝は最大 $b_{\max} \times r_{\max}$ 個となり, 解として不適切な選択枝を削減して, 学習の効率化が期待される.

4.1.3 報酬関数

ネスティング問題では, n 個すべての部材が重なりなく配置された状態 s_n での歩留まり率が良くなるように部材を配置することが目的である. そのため, すべての部材が配置された終端状態でのみ歩留まり率を計算でき, それを報酬と見なす. これら以外の状態では, 歩留まり率は未確定であるため, 報酬は与えず, 次のような報酬関数として定義する.

$$R(s_{i-1}, a_i, s_i) = \begin{cases} Y(Z) & \text{if } i = n \\ 0.0 & \text{otherwise} \end{cases} \quad (4)$$

式(4)の報酬関数は 0 から 1 の間の値を取りうる.

4.2 Deep Q Network に基づくネスティング手法

4.1 節で MDP として捉えたネスティング問題に対し, 強化学習を用いて最適解を求める手法について説明する. 部材形状を表すピクセルデータを入力として, 行動価値を出力する関数の学習法として, Mnih らが提案している Deep Q Network¹⁰⁾ を用いた. なお, 学習の効率性を考慮し, Double Deep Q Network¹¹⁾, Dueling Network¹²⁾ および Prioritized Experience Replay¹³⁾ を導入しているが, それらの計算手続きは文献に記載されているため, ここでの説明は省略する.

4.2.1 学習時のアルゴリズム

本提案方法の学習時の計算手続きを下記 Step0~Step4 に示す.

Step0: 線群で表される実部材データを読み込み, 3 章で述べたピクセル形式での表現に変換後, 同一板厚同一材質毎に部材を分け, 部材群 G_i を生成する. また, 行動価値を推定するニューラルネットワークを初期化する.

Step1: まず, 部材群とそこから取り出す個数 n_p をランダムに決定する. 選択された部材群 g_i の中から部材をランダムに n_p 個取り出す. 取り出された n_p 個の部材は 3 章で述べたように, すべて未配置の状態のピクセルデータとして表し, 初期状態 s_0 とする.

Step2: 4 章のルールに基づき, 状態 s_{i-1} における配置可能領域内に部材が収まらない行動を省いた選択枝の中から, Mnih ら¹⁰⁾ が用いていた Epsilon-Greedy 法に基づき行動 a_i を選択し, 行動に対応した配置位置・角度で配置し, 次の状態 s_i , 報酬 r_i を計算する. また, 行動履歴 (s_{i-1}, a_i, r_i, s_i) を最大優先度とともに保存する. この保存先を Replay memory と呼ぶ. 本報では初期状態から終端状態に至るまでを 1 episode とする.

Step3: Step2 が K 回実施されるごとに, Replay memory から Schaul らによって定義される優先度¹³⁾に基づきバッチサイズ分の履歴を取り出し, ニューラルネットワークのパラメータを更新する. この更新には Kingma らが提案する Adam¹⁴⁾ を用いた.

Step4: 実施した episode 数が M 回となるまで, Step1~Step3 を繰り返す.

4.2.2 推論時のアルゴリズム

4.2.1 節で述べた学習を実行した後、その学習済みのニューラルネットワークを用いて行動価値を推定し、最も行動価値が高い行動を選択し続けることで、最終結果を得ることができる。この行動価値が正しく正確に推定されていれば、十分な結果が得られるが、一般的には誤差を含むため、部材の変化に対するロバスト性に欠ける。そのため、本手法では下記のような探索アルゴリズムを導入し、結果の精度を高める。

- Step1: 現在の状態に基づき行動価値関数をニューラルネットワークで予測する。
- Step2: 予測結果に対して最も行動価値の高い上位 N_a 個の行動を選択する。
- Step3: N_a 個の行動から 1 つ行動 a_j を選択し、それ以降は学習したニューラルネットワークを用いて終端状態まで更新し、報酬 r_j を得る。
- Step4: Step3 を N_a 個すべてに対して実施し、その中から得られた報酬が最も大きい行動を選択し、状態を更新する。
- Step5: Step1～Step4 をすべての部材が配置終わるまで繰り返し実施する。

5. 計算結果

実船舶の部材データを用いて、4 章で述べた提案手法と従来のネスティング手法である BLF 法、および人間の熟練者による配置結果とを比較し、提案手法の有効性と汎化性能について考察する。

5.1 計算条件

実船舶の全部材データに対して、そのままの条件で計算を適用するのは、コストパフォーマンスの面から実用的ではない。そこで、ネスティング問題を一部簡略化し、それに即したニューラルネットワークの構成およびハイパーパラメータの調整を行った。

5.1.1 ネスティング問題の条件設定

通常、船の中央部や居住区等を構成する部材形状は長方形であることが多いが、船首や船尾部は曲がり部が多く、複雑な部材形状をしており、ネスティングの問題の難易度は高いと考えられる。そこで本報では、船首を構成する 1 つのブロック A の部材を対象に検証を行った。対象としたブロック A は、345 個の部材で構成される。ただし、小さな部品も含まれており、これらの部材は大きめの部材が配置された後にできた隙間に配置することが多いため、ここでは、ピクセルで表現された部材のピクセル数が 100 以上 (約 8m^2 以上の面積) であった 95 個の部材を対象を絞り込んだ。このピクセル数の閾値は、大きすぎると、中程度の大きさの部材まで除外され、熟練者と同等の配置結果を再現できない場合がある。そこで、今回対象とした熟練者の配置結果を再現できる最大数を事前に検討し、その結果ピクセル数を 100 と設定した。

また、今回除外される小さな部材の配置は、実用上は従来手法を活用して配置することを想定する。

今回用いたブロックに対して人間の熟練者により複数の母材に配置されていたが、一つの母材あたりに配置された部材数は、部材のピクセル数が 100 未満の小さな部材を除けば、すべて 10 個以下であった。そのため、本報では、最大配置可能数を 10 個と設定した。配置数は他のブロックに対しても同様の傾向がみられたため、特別に限定された条件ではない。切断機の許容幅や購入可能な鋼材の最大長を考慮し、適切な正方形格子幅を選択した結果、本報では、配置可能領域の幅 $b_{\max}$ のピクセル数を 16、長さ $l_{\max}$ のピクセル数を 76 とした。回転については、 0° から 360° までも 10° 刻みとして、回転角度の候補数を $r_{\max} = 36$ と設定した。正方形格子幅と角度刻みは細かいほど正確な最適配置を再現できるが、実用性と学習コストとのトレードオフを考慮して決定している。

また、穴のような内部まで考慮して配置する場合と内部への配置を考慮しない場合の 2 種類が考えられるが、今回のケースでは計算時間が 3 割ほど増加する上、優位な差がみられなかったため、内部への配置を考慮せずに学習を実行した。

5.1.2 ニューラルネットワークの構成

入力層は状態 s_{i-1} として配置済みのピクセルデータおよび最大配置部材数 10 個の未配置部材を表すピクセルデータをあわせた $16 \times 76 \times (1+10)$ のピクセルデータとし、出力層は $b_{\max}$ の幅における x 方向位置と角度の組合せに対応する各行動の価値 $Q(s_{i-1}, a_j), j = 1, \dots, (16 \times 36)$ とする。構築したニューラルネットワークのネットワーク構造の概要を Fig.4 に示す。図中の青矢印は畳み込みニューラルネットワーク、黒矢印は全結合層、緑色の矢印は Dueling Network¹²⁾ で定義される合算を示す。初めの中間層には $4 \times 4 \times 32$ のフィルタをストライド数 2 で適用し、活性化関数として ReLU¹⁵⁾ を用いた。2 つ目の中間層と 3 つ目の中間層は、 $4 \times 4 \times 64$ のフィルタをストライド数 2 で適用し、同様に ReLU を用いた。5 層目は Dueling Network¹²⁾ における value と advantage のユニット数 2048 の全結合層 2 つとし、こちらも ReLU を適用し、最後の出力層へ全結合とした。

Fig. 4 Network Architecture

Table. 1 Parameters

ハイパーパラメータ	値
バッチサイズ	128
リプレイメモリサイズ	1000000
ターゲットネットワーク更新頻度	1000 episodes
学習頻度 K	10 episodes
学習率	0.0005
初期探索時の $\square$	0.4
最終探索時の $\square$	0.01
学習開始時のリプレイメモリサイズ	20000

5.1.3 パラメータ

Table. 1 に示すパラメータ値を用いて、600000 episode の学習を行った。表中の探索 $\square$ は Epsilon-Greedy 法¹⁰⁾におけるランダムな行動をとる確率であり、最初の 300000 episode の間は、初期探索時の $\square=0.4$ から線形的に最終探索時の $\square=0.01$ まで降下し、それ以降は 0.01 で固定した。その他、Deep Q Network⁹⁾ にて必要なパラメータは Table. 1 の通りであるが、本提案手法で試行錯誤的に決定した値である。

5.2 学習に用いた部材に対する計算結果

BLF 法および提案手法 (DQN) によって配置された結果、および熟練者 (Expert) によって実際に配置された結果の比較を行う。Fig. 5 は、横軸が歩留まり率、縦軸の c0~c6 は実際に配置された部材を基に設定された全 7 ケースの配置問題の名称を示している。1 つのケースに対してランダムに配置順を変更した 20 個の問題を作成し、BLF 法と提案手法はともに 20 問すべてを配置して、その中から最も歩留まり率の良い解を選択している。

Fig. 5 に示すように、c0~c6 のすべてのケースにおいて提案手法は BLF 法と同等かそれよりも歩留まり率の高い結果が得られている。また、c0~c4 については熟練者の結果とほぼ同程度の歩留まり率の結果となっているが、c5 と c6 の結果は熟練者の結果よりも劣る結果となった。

Fig. 6 は c0~c6 の配置結果と歩留まり率を示している。各部材は紺色~黄色で色分けしており、紺色から順に配置していることを表している。まず、c0 と c2 の結果は、人の結果と提案手法が同様の配置を出力しているのに対し、BLF 法は歩留まり率の劣る解が出力されていることがわかる。これは 1 つずつ部材を配置する過程において、BLF 法は即時的な歩留まり率のみで配置ルールを構築していることに起因する。c2 のケースに代表されるように、BLF 法の結果にあるように 3 つ目の部材を斜めに配置すると、3 つの部材だけの即時的な歩留まり率は、斜めに配置しない場合よりもよくなるため、この配置を採用している。しかしながら、図に示すように 4 つ目の部材もそれに合わせて斜めに配置されるため最終的に劣った解となる。c0 のケースについても提案手法 (DQN) や Expert のように 2 つ目の部材をあえて縦長

に配置すれば最終的に良い解となるが、BLF 法では即時的な歩留まりで判断するため、このような解を求めることができていない。c1, c3, c4 については、熟練者の結果と提案手法の結果は異なるものの、ほぼ同程度か、ピクセル化の誤差を考慮しても熟練者の結果を上回る結果が得られている。一方、c5, c6 については熟練者の結果が最も良い配置結果となっている。これらは少しでも配置位置や角度がずれると歩留まり率が大きく悪くなるため、解としては非常に発見することが難しく、この差が生じたと考えられる。

以上より、熟練者よりも劣る解が一部存在するものの、従来法よりも良い結果かつ熟練者の結果に近い解を出力していることから、本提案手法の有効性が確認できる。

また、計算時間について、c6 で示す 10 個の部材をある配置順で配置する場合、BLF 法では約 0.5 秒、本手法では約 4.3 秒となった。使用した計算機の CPU は Intel(R) Core™ i7-6800K @ 3.40GHz である。BLF 法よりも提案手法のほうが計算時間は長い、実用上十分な計算速度といえる。

Fig. 5 Results for 7 nesting problems.

Fig. 6 Placement results and yield rate.

5.3 未学習の部材に対する計算結果

5.2 節で述べたように学習に用いた部材に対しては良好な結果が確認されたが、提案手法に限らず、一般的に機械学習において汎化性能にはしばしば課題が残る。本報においても汎化性能について検討するため、5.2 節までで用いたブロック A とは異なるブロック B を用いて配置結果の検証を行う。

Fig. 7 は未学習部材に対する BLF 法と提案手法の結果を Fig. 6 と同様に示している。c7~c9 については学習したことのない部材ではあるものの、BLF 法と同等かそれ以上に良い解を求めることができています。

Fig. 7 Placement results and yield rate for untrained parts.

Fig. 8 Retrained placement results and yield rate.

一方, c10~c12 のように BLF 法よりも劣る解も存在する. 特に c11 や c12 のように, 長方形形状で細長い部材の配置の際に, 提案手法では隙間が発生するような配置となっているため, 歩留まり率が悪くなっている. これは行動価値関数を予測するニューラルネットワークが正しくその価値を予測できなかったことが原因であり, 学習時に細長い部材が含まれていなかったことが要因と考えられる. そのため, これらの部材も含め, 改めて学習を実行することで, これらの配置は改善される. Fig. 8 は未学習であった部材を含めて改めて学習を実施し, それに基づき配置した結果を示す. この図より, BLF 法と同様かそれ以上の解となっており, 改めて学習を実行することで結果が大きく改善されることが確認される. 以上のように, 部材の形状の種類を増やして学習させることで, 結果が改善されるため, 造船で用いられることの多い形状を学習することにより, 本提案手法が造船において非常に有効な自動ネスティング手法になると考える.

6. 結言

本報では, 造船におけるネスティング問題に対して, 深層強化学習の一手法である Deep Q Network を用いた配置手法を新たに提案した. 部材をピクセル形式に変換し最適化問題として定義し, それが MDP 問題として取り扱うことが可能であることを示した. 実際のブロックの部材を対象として, 熟練者と従来法である BLF 法による配置結果とを比較した結果, 多くの例において BLF 法よりも最適な配置を求めることができ, 熟練者の結果に近い配置を求めることができ, 本提案手法の優位性が確認された. 一方で, 未学習の部材に対しては, 従来法よりも劣る結果が一部存在し, 汎化性能に対しては今後の課題である. ただし, これら未学習の部材も含め改めて学習を実行すれば, 結果が大きく改善されることも確認された. このため, 造船においてよく現れる形状の部材を学習することができれば, 実用上の問題はかなり軽減され, 本提案手法が造船において非常に有効な自動ネスティング手法となりうることを示した. 本報では造船部材のネスティングを対象としたが, 造船においては, 定盤レイアウトやスケジューリング問題などの配置問題もあるため, 適応対象の展開も期待できる.

謝 辞

本研究は株式会社大島造船所との共同研究によりなされた成果である. 関係各位に深く感謝する.

参 考 文 献

1) Burke, E., Hellier, R., Kendall, G., and Whitwell, G.: A New Bottom-Left-Fill Heuristic Algorithm for the

Two-Dimensional Irregular Packing Problem, *Operations Research*, Vol. 54, No. 3, pp. 587–601, 2006.

2) Burke, E. K., Kendall, G., and Whitwell, G.: A New Placement Heuristic for the Orthogonal Stock-Cutting Problem, *Operations Research*, Vol. 52, No. 4, pp. 655–671, 2004.

3) Umetani, S., Yagiura, M., Imahori, S. and Imamichi, T.: Solving the irregular strip packing problem via guided local search for overlap minimization, *International Transactions in Operational Research*, Vol. 16, pp. 661–683 (2009)

4) Hamada, K., Ikeda, Y., Tokumoto, H. and Hase, S.: Development of automatic nesting system for shipbuilding using the branch-and-bound method, *Journal of Marine Science and Technology*, Vol. 24, pp. 398–409 (2019)

5) Okumoto, Y. and Iseki, R.: Optimization of Block Allocation in Assembly Area Using Simulated Annealing Method, *Journal of the Japan Society of Naval Architects and Ocean Engineers*, Vol. 1, pp. 71–76, 2005 (in Japanese). 奥本泰久, 井関隆太郎: SA 法を応用した組立定盤内のブロック最適配置, *日本船舶海洋工学会論文集*, 第 1 巻, pp. 71–76, 2005.

6) Yamauchi, S. and Tezuka, K.: Automatic Nesting System by Use of Genetic Algorithm, *Journal of the Society of Naval Architects of Japan*, Vol. 178, pp. 707–712, 1995 (in Japanese). 山内重樹, 手塚研治: 遺伝的アルゴリズムを用いた自動ネスティングシステムの開発と実用化, *日本造船学会論文集*, Vol. 178, pp. 707–712, 1995.

7) Takahara, S. and Miyamoto, S.: Application of Meta-Heuristics to a Nesting Problem, *Transactions of the Society of Instrument and Control Engineers*, Vol. 33, No. 11, pp. 1081–1086, 1997 (in Japanese). 高原茂幸, 宮本定明: メタ戦略を用いたネスティング問題の解法, *計測自動制御学会論文集*, 第 33 巻, 第 11 号, pp. 1081–1086, 1997.

8) Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, G. M., Graves, A., Riedmiller, M., Fidjeland, K. A., Ostrovski, G., Petersen, S., Beattie, C., Sadik, A., Antonoglou, I., King, H., Kumaran, D., Wierstra, D., Legg, S. and Hassabis, D. Human-level control through deep reinforcement learning, *Nature*, Vol. 518, pp. 529–533, 2015.

9) Hu, Y., Fukatsu, S. and Hashimoto, H.: EFFICIENT OVERLAP DETECTION AND CONSTRUCTION ALGORITHMS FOR THE BITMAP SHAPE PACKING PROBLEM, *Journal of the Operations Research Society of Japan*, Vol. 61, pp. 132–150, 2018.

10) Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D. and Riedmiller, M.: Playing Atari with Deep Reinforcement Learning. *NIPS Deep*

Learning Workshop 2013.

- 11) Hasselt, H., Guez, A. and Silver, D.: Deep Reinforcement Learning with Double Q-learning, Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence 2016, pp. 2094–2100, 2016.
- 12) Wang, Z., Schaul, T., Hessel, M., Hasselt, H., Lanctot, M. and Freitas, N.: Dueling Network Architectures for Deep Reinforcement Learning, Proceedings of the 33rd International Conference on Machine Learning, 2016
- 13) Schaul, T., Quan, J., Antonoglou, I. and Silver, D.: Prioritized Experience Replay, International Conference on Learning Representations, 2016
- 14) Kingma, P. D. and Ba, L. J.: Adam: A Method for Stochastic Optimization, International Conference for Learning Representations, 2015.
- 15) Vinod, N. and Geoffrey, H.: Rectified Linear Units Improve Restricted Boltzmann Machines, Proceedings of the 27th International Conference on Machine Learning, 2010.