

令和 4 年度 卒業論文

複数 AMR の衝突を考慮した 施設レイアウト最適化のための 高速評価手法の提案

指導教官 鈴木 達也 教授
奥田 裕之 准教授
本田 康平 助教

名古屋大学 工学部
機械・航空宇宙工学科
機械知能学講座
モビリティシステムグループ

学部 4 年 望月 洋彰

Copyright © Hiroaki Mochizuki

2025 年 2 月 18 日

目次

第 1 章	序論	1
1.1	研究背景	1
1.2	従来研究	2
1.2.1	施設配置問題 (FLP)	2
1.2.2	経路計画アルゴリズム	3
1.3	研究目的	5
1.4	本論文の構成	5
第 2 章	基礎理論	7
2.1	CBS (Conflict-Based Search)	7
2.1.1	A*アルゴリズム	7
2.1.2	高レベルの探索	8
2.2	CNN (畳み込みニューラルネットワーク)	9
2.2.1	概要	9
第 3 章	レイアウト評価のための CNN モデル構成	13
3.1	問題設定	13
3.1.1	マップデータ設定	13
3.1.2	コストの定義	13
3.2	提案手法	14
3.2.1	モデル概要	14
3.2.2	データセット	15
3.2.3	データ生成	15
3.2.4	入力データ	16
3.2.5	モデル設定	18

第 4 章	実行結果	21
4.1	実行結果	21
4.1.1	コスト分布	21
4.1.2	誤差分布	23
4.1.3	各手法の評価指標	25
第 5 章	考察	27
5.1	CNN モデル適応による結果比較	27
5.2	学習画像による変化	28
5.2.1	狭所マップ	28
5.2.2	経路マップ	29
5.3	まとめの考察	29
第 6 章	まとめ	31
6.1	結論	31
6.2	今後の展望	31
	参考文献	35

目次

1.1	日本の産業用ロボットの稼働台数 [1]	2
1.2	FLP 問題に対するメタヒューリスティック手法の使用分布 [2]	3
1.3	効率的な経路と非効率な経路の一例	3
1.4	Neural A*のネットワーク構成	4
2.1	探索過程	9
2.2	CNN の構造	10
2.3	畳み込み	10
2.4	最大プーリング	11
3.1	マップデータ	14
3.2	モデル概要	15
3.3	チャンネルイメージ	17
3.4	狭所マップ	17
3.5	A*マップ	17
4.1	A*コストのコスト分布	22
4.2	手法 1 のコスト分布	22
4.3	手法 2 のコスト分布	22
4.4	手法 3 のコスト分布	23
4.5	手法 4 のコスト分布	23
4.6	A*コストの誤差分布	24
4.7	手法 1 の誤差分布	24
4.8	手法 2 の誤差分布	24
4.9	手法 3 の誤差分布	25

4.10	手法 4 の誤差分布	25
------	----------------------	----

表目次

3.1	CNN モデルと各入力データ	18
3.2	各手法	18
3.3	モデル設定	18
3.4	ネットワークパラメータ	19
4.1	CNN モデルと各学習データ	26

第 1 章

序論

1.1 研究背景

我が国では少子高齢化が深刻な社会課題となっており、あらゆる業界で働き手の減少が懸念されている。製造業では特に工場の有能な働き手が減少が問題視されている。そのような人材不足を解消し、さらに生産性向上、品質の安定化を図るためにロボットによる生産ラインや一部作業を自動化する取り組みが行われている。国際ロボット連盟(IFR) [1] によると、図 1.1 のように生産を自動化する産業用ロボットは年々増加している。このような背景のもとファクトリーオートメーション (FA) が推進され、FA 装置機器を導入することにより工場全体の自動化が進められている。その FA の一環として AMR(Autonomous Mobile Robot) や AGV(Automated Guided Vehicle) といった自立走行ロボットを導入し部品の搬送などを自動化する企業が増えている。

AMR や AGV といったエージェントが移動するにあたって荷物を指定された目的地に運ぶ必要があるが、施設の配置によっては、極端な狭所や作業場の密集などが生まれ、エージェントの迂回やエージェント同士の衝突の可能性が高まる。それによって工場全体の生産性の低下や製造コスト増加といった問題が発生する。工場内の施設配置は、運用の全体的な効率に貢献し、総運用コストを 50 %まで削減することができる [3]。そのため、適切な施設配置を選択することで、それらの余分なコストの削減、また工場全体のパフォーマンス最大化につなげることができる。

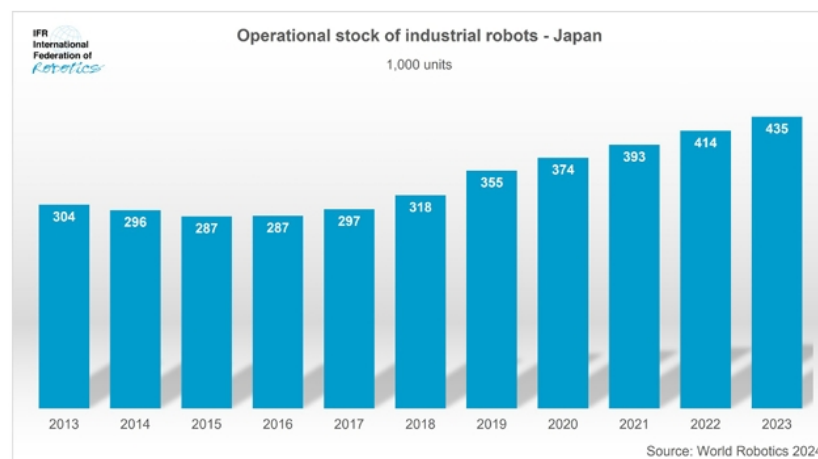


図 1.1 日本の産業用ロボットの稼働台数 [1]

1.2 従来研究

1.2.1 施設配置問題 (FLP)

施設配置問題は、しばしば facility layout problem (FLP) と呼ばれ、施設の配置が与えられたときに、施設間の移動コストや施設の配置による生産性の変化を最小化するような施設配置を求める問題である。

施設配置はレイアウトとも呼ばれ、FLP では最適なレイアウトを見つけることが目的とされるが、この問題は複雑で一般的には NP 困難であることが知られている。そのため現実的な時間でそこそこ良い解を求める手法や、個別設備に関する細かな配置制約を満たす解を求める手法が主に研究されている [4]。図 1.2 に示すように、FLP でもっとも用いられている最適化アルゴリズムは、GA (遺伝的アルゴリズム) である。GA は、自然選択と遺伝子操作を模倣したアルゴリズムであり、評価集団に対して、適応度の評価、選択、交叉、突然変異といった操作を繰り返すことで最適解を探索するアルゴリズムである。広範囲な探索空間を効率的に探索し、局所最適解に陥りにくいが、集団サイズが大きい場合や多数の世代を必要とする場合では計算時間が長くなるという特徴もある。

GA のような最適化アルゴリズムでは、どのレイアウトが最適なのかを示す評価基準が必要となる。そこで、互いに衝突せずに移動させたときのエージェントの合計移動時間をコストと定義する。同一のレイアウトでも図 1.3 の非効率な経路のように余分な迂回をすることでエージェントの移動距離の増加や衝突回避のために待機するエージェントの増加によって移動時間が増えてしまう。そのため効率的な経路のような必要最低限の経路を通った移動時間をそのレイアウトが持つコストとして、レイアウトごとに比較することでコストを最小にする最適レイアウトを見つける。

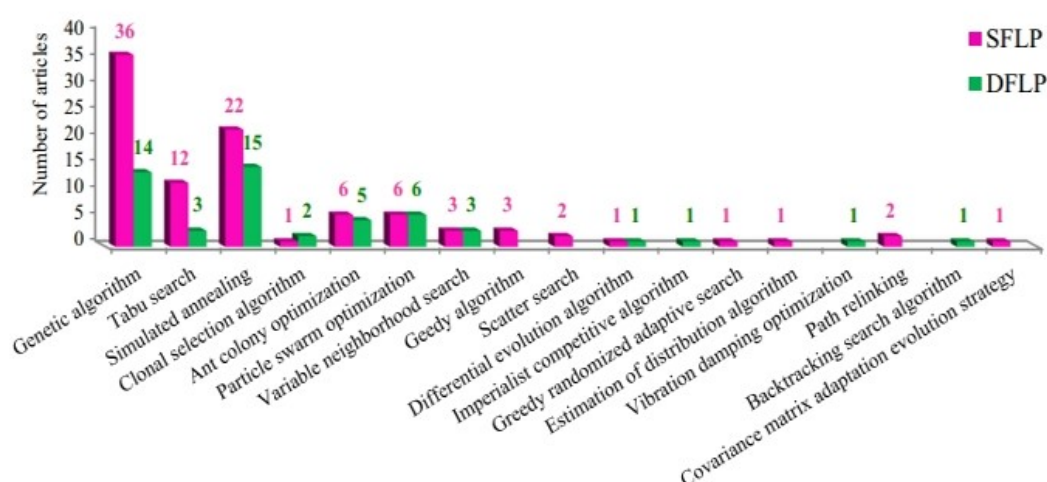


図 1.2 FLP 問題に対するメタヒューリスティック手法の使用分布 [2]

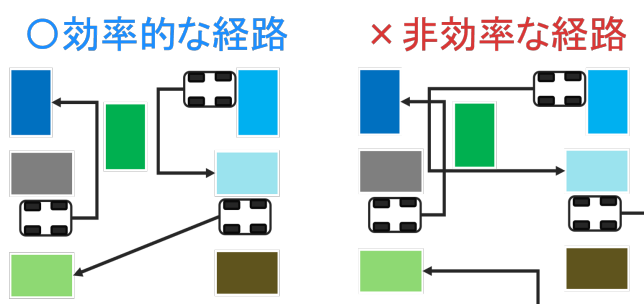


図 1.3 効率的な経路と非効率的な経路の一例

1.2.2 経路計画アルゴリズム

障害物のある環境において、与えられたスタート地点からゴール地点に至るまでの障害物に衝突しない最短の移動経路を出力する問題は経路計画問題と呼ばれる。

グラフ上の複数エージェントに対して互いに衝突しない経路を計算する問題はマルチエージェント経路計画問題 (Multi-Agent Path Finding: MAPF) と呼ばれ、様々な研究が行われている [5]。この MAPF においてよく使用されているアルゴリズムとして Sharon らによる CBS(Conflict-Based Search) がある [6]。CBS アルゴリズムでは、A* アルゴリズムのような単一エージェントの経路計画と経路同士の衝突管理の二つのアルゴリズムを交互に実行することによって、各エージェントの衝突のない経路を探索することができる。CBS は多数エージェント環境においても探索解の最適性が保証されているといった特徴を持つ。その一方でマップサイズやエージェントの数が増え、解の候補が多くなりすぎることによって計算時間が増加するといった傾向がある。またレイアウト最適化問題は、レイアウト集団も多く多数の世代交代も行うため、GA などの最適化アルゴ

リズムと組み合わせて何万ものレイアウトに対して CBS で計算させると計算時間がかかり過ぎてしまい、解が現実的な時間で求まらないことがある。最適解よりも実時間で解が求まることの方が重要であるため、PP(Prioritized Planning)[7] や PIBT (Priority Inheritance with Backtracking)[8] のような準最適解であっても早く計算できるアルゴリズムも MAPF では研究されている。

このように準最適解であっても解が速く求まることが必要となるため、機械学習技術を活用することでさらに計算時間を早めようとする研究が進められている。Guillaume Sartoretti らによる PRIMAL[9] では、強化学習と模倣学習を組み合わせることで、マルチエージェント環境における経路計画を学習する手法が提案された。PRIMAL は、従来の CBS (Conflict-Based Search) などの最適化アルゴリズムと比較して高速に経路計画を行うことが可能であるが、障害物が増えて密度が上がると計算精度および計算速度が落ちてしまう。このように強化学習は計算コストが高く、大規模なマップやエージェント数が増加すると解の探索に膨大な時間を要するという課題がある。

このように強化学習、模倣学習を用いた活用した経路計画の高速化は広く研究されている。一方で深層学習単体、特に CNN を用いたコスト予測に関する研究は限られている。谷合らによる Neural A* [10] では、CNN によってガイドマップを作成し、A* アルゴリズムの探索範囲を制限することで、最適性を維持しながら計算時間を短縮する手法が提案された。これによって CNN ではマップの経路に必要な部分について特徴抽出を行うことができ、経路という非線形問題に対してもうまく学習が可能であることがわかる。しかし Neural A*では単一エージェントの経路計画についてのみ行われており、マルチエージェントでの経路計画は行われていない。

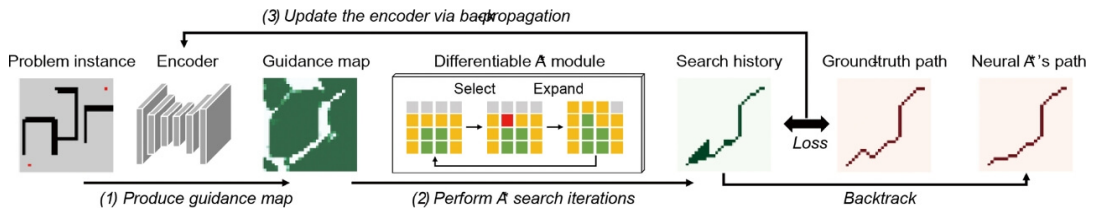


図 1.4 Neural A*のネットワーク構成

このように CNN 単体で経路計画問題を解いている研究は少なく、マルチエージェントに拡張されていない。これは、CNN による予測が推定値にとどまり、最適性が保証されないことや MAPF のような複雑な問題に対して高精度な予測が困難であると考えられているためである。しかし、CNN は画像認識やに優れており、画像データからの特徴抽出が得意であることからマップデータから障害物や各エージェント位置の空間的な関係を理解し、エージェントの経路や衝突位置などを予測することができるのではないかと考えた。neural A*のように CNN による特徴抽出は経路という非線形な問題に対しても適用可能であることがわかっているため、マルチエージェントにも適用可能であると考えられる。また、従来の経路計画アルゴリズムではエージェント数が増えることで計算時間が指

数関数的に増大するが，CNN では学習することで局所特徴を抽出し，パラメータを共有することで予測を高速化できる．そのため，CNN による高精度なコスト予測が実現できれば，計算時間を大幅に短縮でき，MAPF のような問題に対してより迅速なコスト予測が可能となる．

1.3 研究目的

そこで本研究では，これらの傾向を踏まえ，マルチエージェント環境において CNN を用いた直接的なコスト予測モデルを構築し，その有効性を検証することで，従来の経路計画手法と比較した場合の計算効率の向上を目指す．

1.4 本論文の構成

本論文は以下の様に構成される．

- 第 1 章では，研究背景・研究目的を述べた．
- 第 2 章では，基礎理論として CBS と CNN について述べる．
- 第 3 章では，本研究のモデル構成と実行条件について述べる．
- 第 4 章では，予測結果についてまとめる．
- 第 5 章では，結果を踏まえた考察を述べる．
- 第 6 章では，本研究の結論と今後の展望を述べる．

第 2 章

基礎理論

本章では、経路計画アルゴリズムとして用いられる CBS と本研究の予測モデルで使用する CNN について述べる。2.1 節では CBS の概要を述べるとともに、CBS で使用される二つの探索アルゴリズムについて説明する。本研究で CNN 予測の補助として使用する A*アルゴリズムについて 2.1.1 節で説明する。2.2 節では本研究の予測モデルに使用する CNN の概要と各層の要素について説明する。

2.1 CBS (Conflict-Based Search)

CBS は低レベルの探索と呼ばれるシングルエージェントの経路計画と、高レベルの探索と呼ばれる経路同士の衝突管理を行う二つの探索アルゴリズムから構成されるアルゴリズムである。CBS はこの二つを交互に実行することで多数エージェント条件においても解を見つけることができる。高レベルの探索では、低レベルの探索によって得られた経路の組み合わせに衝突があるかを判断する。衝突がなければそれを解とし、衝突がある場合には衝突を避けるように制約を課す。低レベルの探索では、高レベルが課す制約に従いながら、解候補を作成する。この低レベルの探索は、単なるシングルエージェントによる経路計画であるため、A*などによるアルゴリズムによって探索が行われる。そこで低レベルの探索で用いられる A*と高レベルの探索について説明する。

2.1.1 A*アルゴリズム

A*アルゴリズム (A*) はヒューリスティック関数によって、単一エージェントの最短経路を探索するアルゴリズムである。A*ではノードと呼ばれる状態を展開し、探索を行っていく。各ノードにはエージェントの現在位置、ゴール位置、コスト、親ノードなどの情報が含まれる。展開されたノードの中から最も有望であるノードを推定し、そのノードを優先して探索するため、ヒューリスティック関数を利用しないブラインド探索と比較して高速に最適解を見つけられることが多い。探索は以下のようなアルゴリズムである。

1. 初期状態をオープンセットに加える.
2. オープンセットにあるノードの中でゴール状態までの見積りが最小であるノードを展開する.
 - (a) 展開したノードをオープンセットから取り除き、クローズドセットにその状態を加える.
 - (b) 展開した状態がゴール状態かどうかを調べる. ゴール状態である場合は解を出力して終了する.
 - (c) 目標状態でない場合、展開した状態から1つの行為で遷移可能な状態のうち、クローズドセットにないすべての状態をオープンセットに加える.
3. オープンセットが空になった場合は解なしとして終了する.

オープンセットは未展開のノード、クローズドセットは展開済のノードが保存されるデータ構造である. これにより、一度探索したノードは再選択されることなく、効率的な探索が可能となる. あるノード n からゴール状態までのコスト $f(n)$ は (2.1) 式で計算される.

$$f(n) = g(n) + h(n) \quad (2.1)$$

$g(n)$ は初期状態からノード n までの経路にかかったコストである. $h(n)$ は状態からゴール状態までの距離を推定するヒューリスティック関数である. ヒューリスティック関数には (2.2) 式のような2次元平面上でのゴール (x_g, y_g) までのユークリッド距離がよく使用される.

$$h(n) = \sqrt{(x_n - x_g)^2 + (y_n - y_g)^2} \quad (2.2)$$

A*は単一エージェントの最短経路問題に強く、最小限のノードのみを展開するため効率的かつ最適な解を発見できる. しかし、A*のみではエージェント同士の衝突を考慮していないため、高レベルの探索を用いることで多数エージェント環境においても解を見つけることができるようになる.

2.1.2 高レベルの探索

高レベルの探索では、constraint tree(CT) と呼ばれるツリーを探索し、衝突のない解を見つけるまで繰り返す. 高レベルの探索の各ノードでは、エージェントのパス、総コスト、制約 (a_i, v, t) (時刻 t でエージェント a_i に頂点 v にいることを禁止) または (a_i, v_1, v_2, t) (時刻 t でエージェント a_i に頂点 v_1 から v_2 に移動することを禁止) を持つ. 高レベルの探索は図 2.1 のように進行する.

1. ある CT ノードを指定すると、低レベルの探索が呼び出される.
2. 低レベルの探索でエージェントごとに最短パスが返される.
3. 低レベルの探索で返されたパスが衝突しているかを確認し、衝突がない場合はそれを解とする

4. 衝突がある場合には衝突したエージェント数だけの制約を追加した子ノードを作成する
5. コストの低いノードから探索し解が見つかるまで繰り返す.

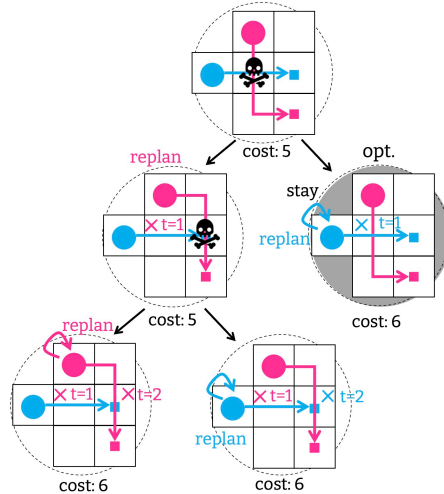


図 2.1 探索過程

2.2 CNN（畳み込みニューラルネットワーク）

2.2.1 概要

CNN（畳み込みニューラルネットワーク）は、学習可能な畳み込み層を含んだ4層以上から構成されるディープニューラルネットワークのことで、画像認識タスクの多くに用いられている。図 2.2 に CNN の構造の全体像を示す。CNN は畳み込み層、プーリング層よりなる特徴抽出部と全結合層からなる識別部によって構成される。CNN には様々なモデルが提案されており、LeNet[11], AlexNet[12], VGGNet[13] などがある。以下で CNN を構成する層と各部の要素について示す。

畳み込み層

畳み込み層は、その時点までに順伝搬で生成されたチャンネル分の特徴マップに対して画像における畳み込みと同様の処理を行い、フィルタに応答した特徴マップを出力する層である。

畳み込みとは、正方形のフィルタを用いたスライディングウィンドウ処理を通して、入力画像の各参照画素の近傍においてフィルタ値と参照画素値の積和計算をおこない、フィルタの値に沿って入力画像を変換する処理である。

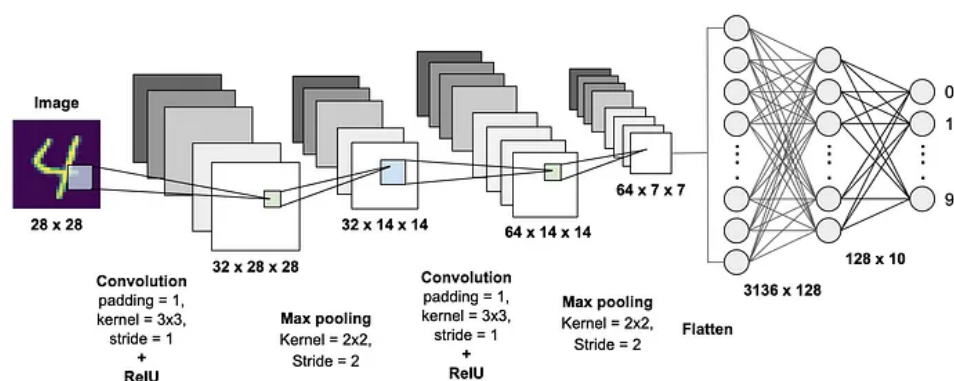


図 2.2 CNN の構造

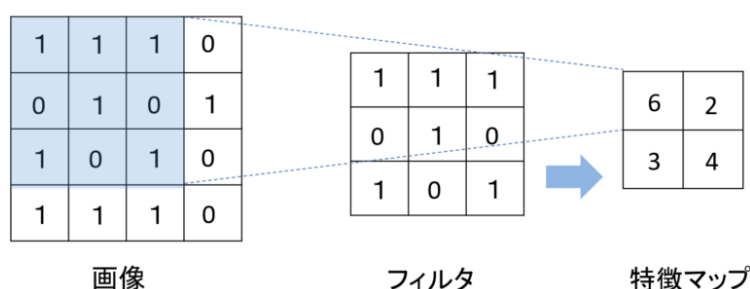


図 2.3 畳み込み

プーリング層

プーリング層では、局所範囲内の代表値のみからダウンサンプリングして集約した特徴マップを得るプーリング処理を行い、特徴マップの解像度を下げる層である。前層の畳み込みで得られた特徴量マップに対して、プーリングを行うことでマップ上の各画素がとらえる情報の対応範囲が広がる効果がある。プーリングには、最大プーリングや平均プーリングなどいくつか手法がある。最大プーリングではサイズによって定義された正方形領域を特徴マップに重ね、参照した領域内における最大値を取得し、正方形領域を移動させそれぞれの最大値を取り込むことで画像のダウンサンプリングを行う。平均プーリングでは参照した領域における平均値を取得しダウンサンプリングを行っている。

全結合層

畳み込み層、プーリング層によって得られた特徴マップを平滑化した際における一つの要素をニューロンと呼び、その全ニューロン同士が線形結合で接続している層を全結合層と呼ぶ。入出力の線形変換を通して特徴の抽出や次元の削減、上昇なども行う層である。全結合内の最終層を出力層と呼ぶ。出力層のニューロンは分類したいクラス数に設定する必要がある。

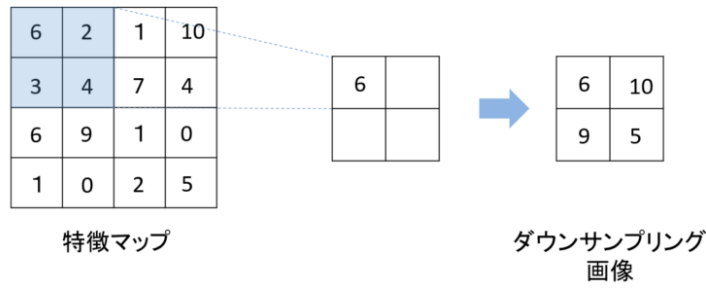


図 2.4 最大プーリング

活性化関数

線形層とプーリング層では、全体表現も線形となり非線形性がないため表現に限界が来てしまう。そこで、線形層の出力後に後処理として softmax や ReLU といった活性化関数が適用化され、活性化が行われる。これを適用することで各フィルタにおける応答の強い空間をより強調することができ、全結合層だけでは表現できない予測問題も学習できる。ReLU (Rectified Linear Unit) は、入力が 0 より大きければその値を出力とし、0 よりも小さければ出力を 0 とするものであり、(2.3) 式で表される。

$$y = \max(0, x) \quad (2.3)$$

誤差関数

CNN では出力する推定値と理想の出力値との違いを数値化し、パラメータを更新することで未知のデータに対応できるようになっている。この数値化として誤差関数が使用され、分類問題では交差エントロピー誤差 (Cross Entropy, CE)、回帰問題では、平均二乗誤差 (Mean Squared Error, MSE) が主に使用される。本研究では平均二乗誤差を誤差関数として使用しており予測値 \hat{y} と理想値 y を用いて以下の式で表される。

$$L_{\text{MSE}}(\hat{y}, y) = \frac{1}{K} \sum_{n=1}^K (\hat{y} - y)^2 \quad (2.4)$$

勾配降下法

学習では、推定データと正解データの誤差関数を最小にするようにフィルタの重みや全結合層の重みなどを更新していく必要がある。この誤差を無くして予測精度を最大化させるために機械学習においては勾配降下法が用いられる。重みを ω ，誤差関数 $E(\omega)$ とするとき、勾配降下法では ∇E を用いて式 (2.5) で更新を行う。 t は反復回数、 ϵ は学習係数を表す。

$$\omega^{(t+1)} = \omega^{(t)} - \epsilon \nabla E \quad (2.5)$$

勾配降下法をそのまま適応しようとするとう訓練データの数が増大であるため、計算時間の増大や誤差関数の形状によっては局所解に陥り最適解が算出できないといった問題が発生する。この問題の解決法として確率的勾配降下法 (SGD)[14] が用いられる。勾配降下法では一回のパラメータ更新に全データを使用していたのに対して SGD ではランダムにピックアップした 1 つのデータを使用し、局所解に陥らないようになっている。現在ではそれに加えて SGD での学習の振動性や並列化といった欠点を軽減するために SGD を応用した adam(adaptive moment)[15] が主にオプティマイザーとして用いられている。

第 3 章

レイアウト評価のための CNN モデル構成

本章では、提案手法である CNN を用いたマップに対するコスト予測モデルの概要について説明する。3.1 節ではマップデータ設定と最適化するコストについて説明する。3.2 節では、具体的なモデルの概要について説明し、実行に使用したデータセットとモデル設定について説明する。

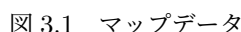
3.1 問題設定

3.1.1 マップデータ設定

本研究では、図 3.1 のようなグリッドマップデータを使用する。図 3.1 において青のマスは障害物を表し、緑のマスはスタート、ピンクのマスはゴールを示している。このように最適経路計画問題では、各エージェントのスタートゴールの位置、エージェントの数、障害物の位置、マップデータの大きさの条件が与えられている。これらの条件を含んだマップデータを以下ではマップと呼ぶ。

3.1.2 コストの定義

各エージェントは上下左右の 4 方向のみ移動が可能であり、1 ステップで 1 マス進むか待機するか行動をとることができる。この各エージェントのゴールまでのステップの総和をコストとする。ここで本研究のモデルで使用する最適コスト、A*コスト、コスト増分について説明する。



本研究において各エージェントの衝突のない最適経路によるコストを最適コストと呼ぶ。最適コストは、CBS によって計算されたコストであり、マップに対するコストの正解として使用する。

A*で計算した衝突を考慮しない各エージェントのコストの総和を A*コストと呼ぶ。これは A*によって単一エージェントでの経路計画を行い、各エージェントのコストを計算した後にすべてのエージェントのコストの和をとることで算出される。ここで算出されるコストは衝突を考慮していない A*によるコストであるため、CBS に比べ計算時間が速く算出することができるが、CBS で計算した経路で衝突が発生した場合に最適コストよりもコストが小さく出力されてしまう。

コスト増分とは、最適コストと A*コストの差として表れる、衝突や迂回によって発生するコスト増加のことである。本研究ではこのコスト増分をモデルによって予測させることで最適コストを推定する。

3.2.1 モデル概要

14

よって求まる A*コストを組み合わせることでマップの最適コストを予測するものである。具体的な学習から予測までの流れを示す。

1. 作成した CNN モデルに入力としてマップデータと、ラベルとしてコスト増分を学習させる
2. テストデータの新たなマップを CNN に入力することによってコスト増分を予測する
3. テストデータについて各エージェントの A*コストを A*アルゴリズムで算出する
4. CNN による出力と A*コストを組み合わせることでマップのコストを予測する

このような流れでマップデータの最適コストを推定する。

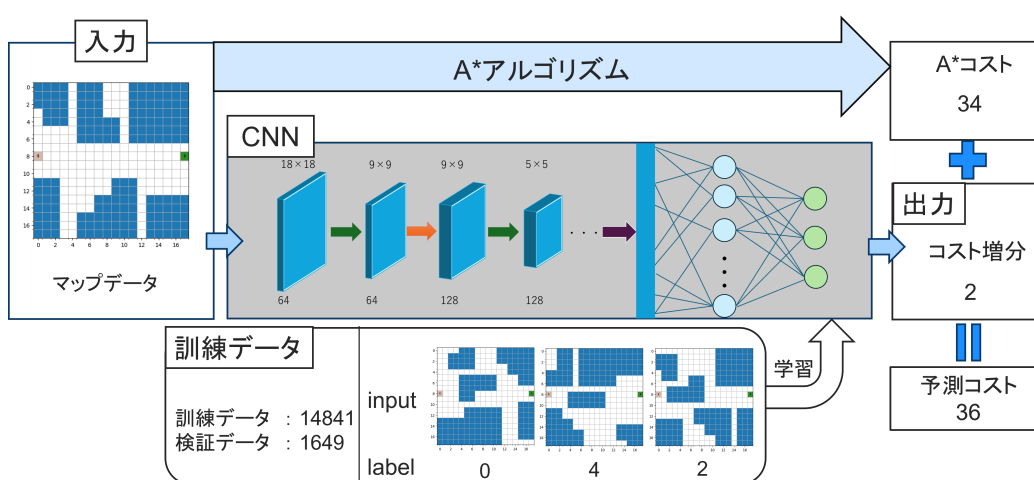


図 3.2 モデル概要

3.2.2 データセット

今回のモデル実装においてレイアウトデータのサイズは (18×18) 、エージェントは 2 台に固定する。訓練データとして 22887 データ、訓練に用いる検証データとして 2543 データ、テストデータとして 1922 データのレイアウトデータを作成して学習、予測に用いる。具体的なデータ生成法に関しては 3.2.3 節で説明する。提案手法では A*を共に計算するため A*によって経路が存在しないマップについては事前に判明する。そのため A*で経路が存在しなかったマップについてはデータ群から除外して学習、予測を行う。

3.2.3 データ生成

今回データを生成するにあたって進化的アルゴリズムを使用し、施設の重複のないマップデータを作成した。進化的アルゴリズムは生物の進化の過程を模倣した手法である。具

体的な生成手順を以下に示す.

1. 任意の施設形状を作成する
2. 初期集団としてランダムに施設や各エージェントのスタートゴールを配置したマップデータを作成する
3. 施設やスタートゴールに重複がないか調べ重複がある場合は再配置する
4. 配置したマップ集団から配置位置などの要素を抜き出し, 集団内の様々なマップから要素を組み合わせ, 子データとして新たなマップを作成する
5. ある一定の割合で突然変異として, 集団にはないランダムな配置を行う

このようなアルゴリズムを繰り返すことでマップデータを生成することができる.

3.2.4 入力データ

CNN に入力する際に, マップデータを障害物の位置, スタートゴールの位置を含む $\bigcirc \times \bigcirc \times$ 次元のマトリックスにする必要がある. そこで本研究では学習タスクを簡略化するために, 利用可能な情報を異なるチャンネルに分けて CNN に認識させ学習を行う. 具体的に説明する. 図 3.3 のように障害物, 各エージェントのスタート位置, ゴールの位置といった情報を表すマトリックスをそれぞれのチャンネルに配置し, まとめて一つのマップデータとする. このように各情報を一つのマップデータとして作成することで学習, 予測を行う. 今回使用するマップは以下の通りである.

- 障害物マップ (OBS)
障害物のある位置を 1, 通行可能な位置を 0 とするマトリックス
- スタートマップ (S)
各エージェントのスタート位置を 1, それ以外の位置を 0 とするマトリックス
- ゴールマップ (G)
各エージェントのゴールの位置を 1, それ以外の位置を 0 とするマトリックス

またこれらのマップに加えて各エージェントの衝突を考慮するために学習の補助となるような情報を加えることで予測精度の変化がみられるかについても検証する. 今回の問題では CNN が衝突回避のためのコスト増分を予測できることが必要になってくる.

狭所では衝突が発生しやすく, コスト増分が起りやすいと考え, 狭所に値を加えた狭所マップを作成した. また, エージェントの経路情報があると CNN が衝突をしやすレイアウトを判断しやすくなると考え, A*によって求められる経路情報を含む A*マップを作成した. 図 3.2.4, 3.2.4 に狭所マップと A*マップについて示す. 青いマスは 1, 緑のマスは 0.5, それ以外の白いマスは 0 を値として持つ. また, マップの詳しい情報は以下の通りである.

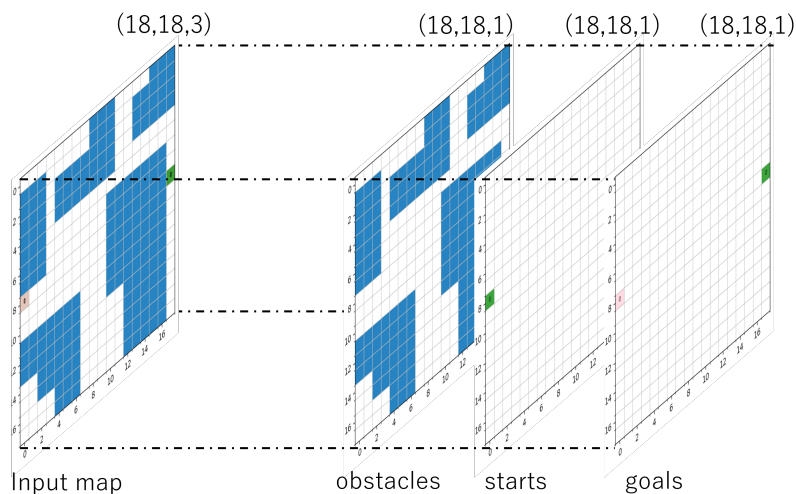


図 3.3 チャンネルイメージ

- 狭所マップ (NAR)
障害物のある位置を 1，障害物同士の間が 1 マスである位置に 0.5，それ以外の位置を 0 とするマトリックス
- 経路マップ (P)
各エージェントの A*によって得られた通行経路に 1 を加えたマトリックス

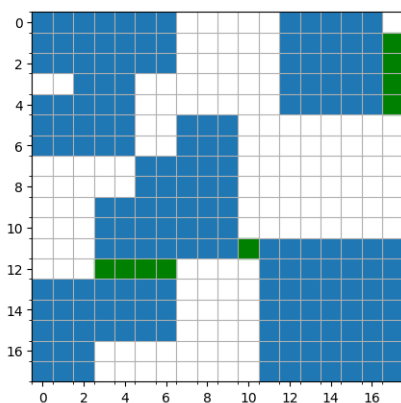


図 3.4 狭所マップ

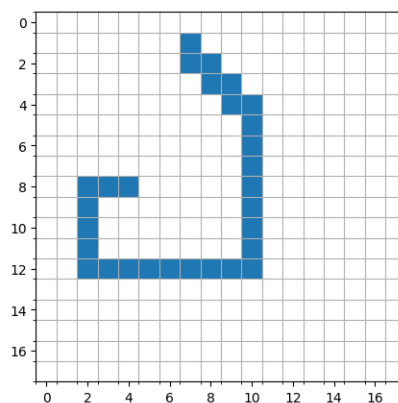


図 3.5 A*マップ

今回はこのマップデータを組み合わせたデータについて学習を行い，表 3.3 のように入力データの種類の異なる CNN モデルを 4 つ作成する．作成モデルをそれぞれ CNN(OBS+S+G), CNN(NAR+S+G), CNN(OBS+P), CNN(NAR+P) と呼ぶ．

作成したモデルを表 3.2 のように A*コストと組み合わせることでそれぞれの提案手法を手法 1, 2, 3, 4 と呼び，コストを予測させる．

表 3.1 CNN モデルと各入力データ

名称	マップ情報	マップサイズ
CNN(OBS+S+G)	障害物マップ + スタートマップ + ゴールマップ	$18 \times 18 \times 3$
CNN(NAR+S+G)	狭所マップ + スタートマップ + ゴールマップ	$18 \times 18 \times 3$
CNN(OBS+P)	障害物マップ + 経路マップ	$18 \times 18 \times 3$
CNN(NAR+P)	狭所マップ + 経路マップ	$18 \times 18 \times 3$

表 3.2 各手法

名称	予測モデル
手法 1	A*コスト + CNN(OBS+S+G)
手法 2	A*コスト + CNN(NAR+S+G)
手法 3	A*コスト + CNN(OBS+P)
手法 4	A*コスト + CNN(NAR+P)

3.2.5 モデル設定

今回作成したモデルの設定を表 3.3, ネットワークパラメータを表 3.4 に示す. conv は畳み込み層, pool はプーリング層, dense は全結合層を表す. 今回のモデルではプーリング層では最大プーリングを使用し, 活性化関数として各層で ReLU を用いている. 出力層では 1 ユニットの線形出力を行い, 誤差関数として平均二乗誤差を用いる.

表 3.3 モデル設定

誤差関数	平均二乗誤差
batchsize	32
epoch	30
optimizer	adam

表 3.4 ネットワークパラメータ

名称	項目	値
conv1	フィルタ数	64
conv1	フィルタサイズ	3
pool1	フィルタサイズ	2
pool1	ストライド	2
conv2	フィルタ数	128
conv2	フィルタサイズ	3
pool2	フィルタサイズ	2
pool2	ストライド	2
conv3	フィルタ数	256
conv3	フィルタサイズ	3
pool3	フィルタサイズ	2
pool3	ストライド	2
dense1	ユニット数	1024
dense2	ユニット数	512
output	ユニット数	1

第 4 章

実行結果

本章では、前章で説明した提案手法であるコスト予測モデルを用いてコスト予測を行った結果について述べる。A*コストのみによる手法と各提案手法の予測結果についてまとめる。

4.1 実行結果

学習データと同様に生成したテストデータ、1648 データを用いて予測を行った結果をまとめる。今回の結果として各手法における予測コスト分布と誤差分布を示す。また予測値の性能評価指標として、最適コストと予測値の差である誤差の絶対値を平均した平均誤差と最大誤差、各手法の計算時間を表 4.1.3 にまとめた。

A*は計算時間は早いですが、衝突を考慮していないため最適コストと差異が生じやすい。そのため提案手法によって結果にどれほどの変化を生じさせたかについて A*コストのみのコスト推定と CNN を加えた提案手法を比較するために A*コストのみによるコスト推定についても結果をまとめる。A*コストのみによるコスト推定は、A*コストとして以下ではまとめる。

4.1.1 コスト分布

最適コストに対する各手法における予測値をコスト分布として以下に示す。横軸にマップの最適コスト、縦軸に予測コストをとる。赤の直線は最適コスト=予測コストとなる直線であり、青点である予測値が直線に沿うほど予測が真値に近づくことを表す。

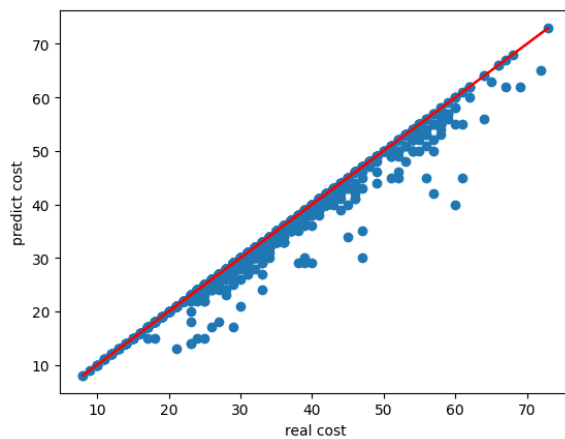


図 4.1 A*コストのコスト分布

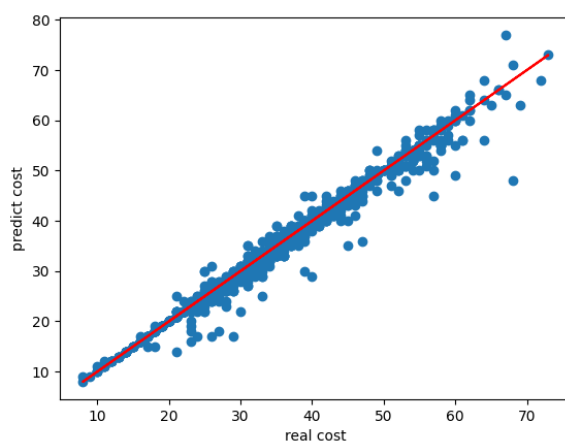


図 4.2 手法 1 のコスト分布

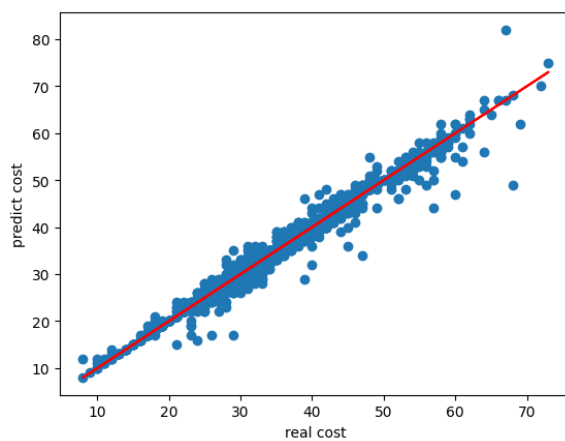


図 4.3 手法 2 のコスト分布

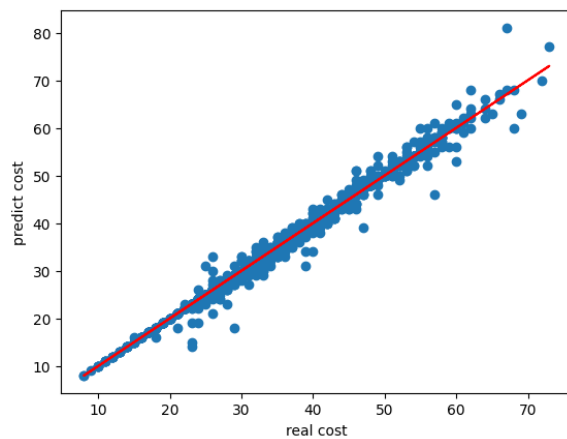


図 4.4 手法 3 のコスト分布

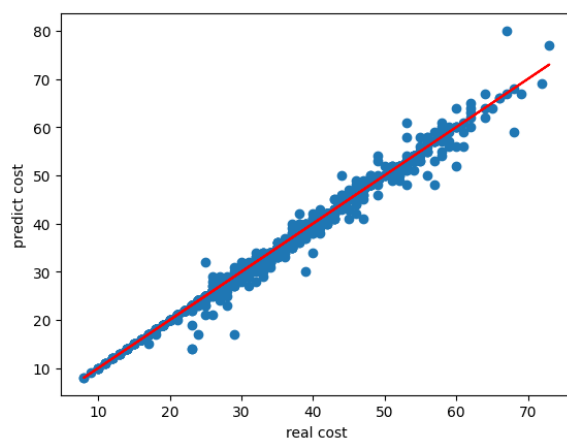


図 4.5 手法 4 のコスト分布

4.1.2 誤差分布

最適コストから予測値を引いたものを誤差として、各手法における誤差の分布をヒストグラムとして以下に示す。誤差が正の時は予測値が最適コストよりも小さく出力され、誤差が負の時は予測値が最適コストよりも大きく見積もってしまったことを示している。

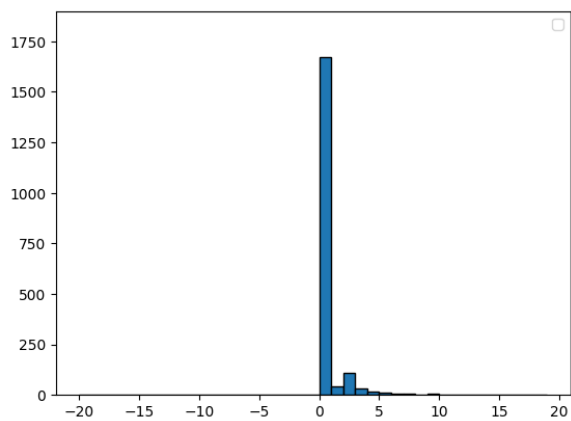


図 4.6 A*コストの誤差分布

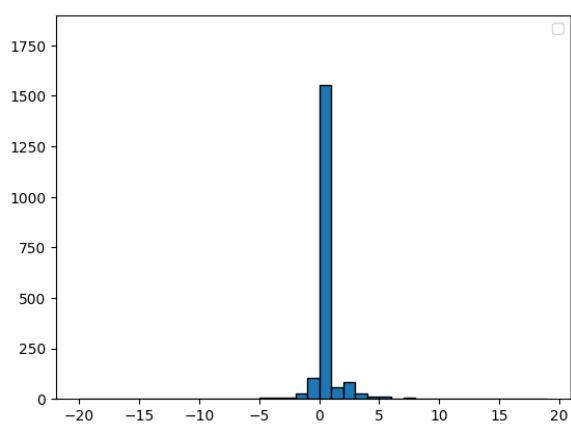


図 4.7 手法 1 の誤差分布

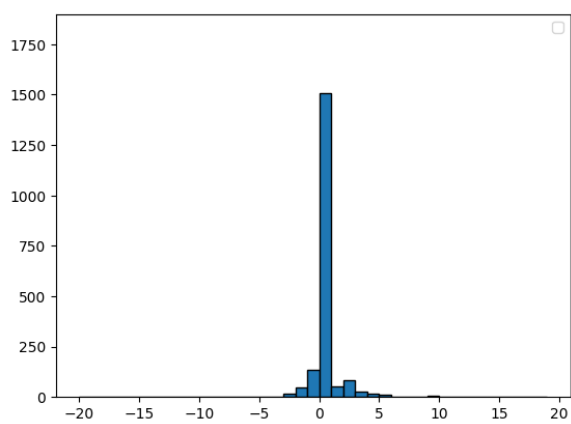


図 4.8 手法 2 の誤差分布

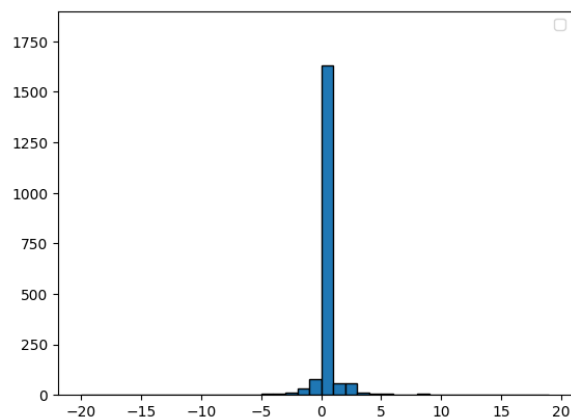


図 4.9 手法 3 の誤差分布

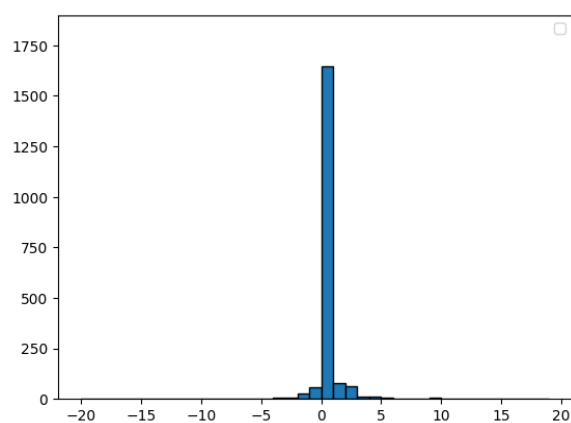


図 4.10 手法 4 の誤差分布

4.1.3 各手法の評価指標

各手法において予測値と最適コストとの平均誤差，最大誤差，計算時間について表 4.1 にまとめる．

表 4.1 CNN モデルと各学習データ

名称	平均誤差	最大誤差	計算時間 (s)
CBS	0	0	3.09375
A*コスト	0.41	20	0.015625
手法 1	0.44	12	0.495625
手法 2	0.47	18	0.555625
手法 3	0.31	14	0.495625
手法 4	0.30	13	0.465625

第 5 章

考察

本章では、前章での結果に対する考察を行う。

5.1 CNN モデル適応による結果比較

まず、CNN モデルを導入することによる結果の比較を行う。CBS と提案手法を比較するとテストデータに対してコストの計算時間が CBS では約 3.09s であるのに対して提案手法では 0.5s となり計算時間を短縮することができた。今回の実行条件ではエージェント台数が 2 台でありマップサイズも小さいため、衝突が少なく CBS での計算時間が速い。しかし、そのような条件でも今回の提案手法のほうが計算時間が速く、提案手法の有用性が示せたのではないかとと思われる。CBS では探索解が増えることで計算時間が指数関数的に増加する。一方で CNN による予測はエージェントの台数や衝突回避制約によらず入力データとデータ数に依存し、狭所や衝突回避制約が増えても計算時間が増加しないため、CBS における探索解が増えると計算時間が伸びるというスケーラビリティ問題が提案手法にはない。そのため CBS よりも提案手法の方が計算時間が短縮できたと考えられる。今回のような台数の少ない条件においても計算時間を短縮することができたため、台数の多い条件では、CBS がさらに計算時間がかかると予想され、台数が増えるほど提案手法は計算時間を短縮できると考えられる。また、手法 4 に対しては、予測値の分布をみても最適コストに沿って近い値を予測できており、提案手法によって最適コストをある程度正しく予測できていることがわかる。これより CNN を用いた提案手法でも最適コストが予測が可能であり、台数の多い条件では CBS よりもさらに計算速度を向上させる見込みがあると考えられる。

A*コストのみの手法と A*コストと CNN モデルを組み合わせた提案手法を比較すると提案手法において平均誤差は減少、最大誤差も減少し、CNN モデルを適応させることで予測精度が向上していることが読み取れる。A*コストのみでは衝突を考慮していないが、CNN モデルを適応することによって衝突を考慮しコストをうまく予測できていると考えられる。また A*コストのようなベースコストがあることで CNN モデルが予測するコス

トレンドがある程度限定され、予測による誤差が発生しにくくなり、また誤差による影響が大きくなる。そのためこのように A*コストと組み合わせる提案手法では最適コストの予測精度が向上されていると考えられる。

5.2 学習画像による変化

学習画像を変化させた場合における各手法の予測精度の変化について考察する。

5.2.1 狭所マップ

狭所マップによる違いを比較するために、手法 1 と手法 2、手法 3 と手法 4 の結果について比べる。

手法 1 と手法 2 を比較すると、狭所マップを追加したことで平均誤差は増加し、最大誤差も増加していることがわかる。一方で手法 3 と手法 4 を比較すると、狭所マップを追加したことで平均誤差が減少し、最大誤差も減少している。誤差分布を比べるとどちらもグラフの外形にほとんど変化はないことが読み取れる。手法 1 と手法 2 について、平均誤差と一致率が悪化した原因としては、狭所に値を加えてしまったことでマップからコスト増分に関する情報が得られなくなってしまったためではないかと考えられる。CNN では特徴を学習する際に畳み込みやプーリングによって特徴の抽出を行うが、この際に狭所マップのように障害物のほかに狭所の値が入っていることで、障害物と同じように狭所も抽出されて実際は通行可能なのに障害物のように学習してしまい、予測精度が悪化したと考えられる。

一方で手法 3 と手法 4 について、誤差が改善したのは、狭所マップと経路マップが重なることでコストの予測に必要な狭所と不要な狭所を学習することができたからと考えられる。CNN では入力データに対してフィルタをかけていく際、重なっている部分にはより強くフィルタがかかるため、狭所マップと経路マップのようにデータが重なっている部分があることでコスト増加に影響する形状について学習がうまく進み予測精度が改善されたと考えられた。

これより入力に使用するマップデータは組み合わせ次第で結果に良い影響と悪い影響のどちらも与えることが分かった。また狭所による予測精度への影響が少なかったことに関してはエージェントが 2 台と少なく、狭所によるコスト増加があまりないため、予測に影響が少なくなったのではないかと考えられる。それに加え、狭所があるマップでもエージェント同士が同時刻で通過しない場合にはコスト増加は発生しないため、狭所情報によるコスト増分への影響が少なく、予測精度の変化があまりなかったと思われる。

5.2.2 経路マップ

経路マップによるモデルの変化を比較するために手法 1 と手法 3, 手法 2 と手法 4 について比較する. 手法 1 と手法 3, 手法 2 と手法 4 を比較すると, どちらも平均誤差と最大誤差が減少していることがわかる. またコスト分布や誤差分布をみても経路マップを入力画像に加えることで全体的に誤差が減少したことがわかる. これより経路マップはコスト予測に大きく影響を与えることが読み取れる. 経路マップは A*アルゴリズムによる各エージェントの経路情報が入っているため, 各エージェントの経路情報がうまく学習に寄与したのではないかと考えられる. コスト増加というのは, 基本的に衝突が発生した場合に発生する. 経路マップでは最初の A*経路情報が含まれているため, 最初の A*で経路が衝突しているかどうかを判断することができる. 最初の A*経路が衝突しているかしていないかの情報があることで CNN は衝突の有無やコスト増加を予測できたのではないかと考えられる. つまり CNN の予測精度向上のためには衝突位置や経路情報が必要なのではないかとも考えられる.

CNN での特徴抽出では, フィルタを用いて周りの値も取り込んで特徴を抽出するため, 経路という直線的な情報が障害物マップと組み合わせてフィルタリングされることで衝突のしやすい位置を特定し, コスト増加予測がうまくいったのではないかと考えられる.

5.3 まとめの考察

以上より A*と CNN を組み合わせることによって, エージェントの衝突を考慮したコスト予測ができることが分かった. 今回 CBS よりも計算時間を短縮し, A*コストと組み合わせることで精度良くコスト予測ができたと考えられる. CBS では狭所が多く探索解が増えると計算時間が増加するが, CNN には狭所が多くとも予測時間に影響はないためこのように短縮ができたと考えられる.

提案手法では入力に使用するマップデータは組み合わせ次第で予測結果に良い影響と悪い影響のどちらも与えるため, 入力データには適切に学習させられるデータを選択する必要があることが分かった. 経路マップのように経路といったコスト増加に直接的に関係してくる入力データや衝突位置を推定できるような情報を加えられると CNN で学習する際に衝突を考慮することができ, 予測精度を向上させることができるのではないかと考えられる. そのため, 経路マップのような経路だけでなく, A*によるエージェントの衝突位置のみを含むマップや経路の時間経過を表せるように値に勾配をかけたマップなどを作成し, 学習, 予測を行って, どの要素が適切な予測のために必要かさらなる検証をしていきたいと考えている.

また狭所マップのようにコスト増分予測に必要な箇所と不要な箇所のあるデータでは必要な箇所が強調されるようなマップデータを組み合わせることで予測精度を向上させるこ

とができると分かった。今回の狭所マップでは障害物マップと同じマップに狭所の値が入ったため、CNN の精度に影響があまりなく、誤学習が発生してしまったと考えられる。そのため狭所のみを別のマップとして入力データに加えることで、予測精度にどれほどの影響が起こるかについても検証したい。今回狭所として、障害物同士の間が1 マスの時のみに値を加えたが、それぞれのマスにおける障害物との距離に重みをつけることや逆に障害物のないオープンスペースに値を加えるといったマップを作成し、狭所マップと比較することでどの要素がコスト増分を予測する際に重要なのかを検証していきたいと感じた。

また提案手法ではモデルが学習する際にエージェント数を固定して学習を行っているため、今回学習したモデルはエージェントが2 台の時のみ使用可能なモデルとなり、エージェント数が変化する場合には新たに学習を行う必要がある。エージェント台数が変化したときのモデルの適応性については考慮できていないため、今後の課題としてエージェント数が変化しても適応できるモデルの構築が必要であると感じた。

第 6 章

まとめ

6.1 結論

本研究では、多数 AMR の衝突を考慮したレイアウト評価値算出を高速化するために経路計画アルゴリズムに代わるコスト予測モデルを提案した。A*コストと CNN を組み合わせることで経路計画アルゴリズムである CBS よりも計算時間が速く、高い精度で予測させることに成功した。また、入力データに学習の補助となるマップデータを加えることで予測精度を向上させることに成功した。

6.2 今後の展望

今回はエージェント台数を 2 台に固定して予測を行ったが、2 台ではなくさらに台数を増やし 4 台、6 台と増やすことで予測精度にどれほど影響があるかについても検証していきたい。2 台について予測ができたことからエージェントの衝突について学習することができることがわかったため、台数を増やすことで複数の衝突を考慮した予測もできるかどうかについても検証を行っていきたいと考えている。今回では経路マップが予測精度向上に大きく役立ったが、エージェント数が増加した際にどのような情報マップを加えると予測精度が向上するかについても検証し、よりモデルをスケーラブルなものにしていきたいと考えている。

謝辞

本研究を進めるにあたり，名古屋大学大学院工学研究科機械システム工学専攻鈴木達也教授，同奥田裕之准教授，本田康平助教には丁寧かつ熱心なご指導を賜りました．心より感謝致します．また，研究のあらゆる面において，鈴木研究室の杉山俊一郎先輩には惜しみないご協力を賜りました．最後に，鈴木研究室の卒業された先輩，博士課程の方々，修士課程の方々，そして共に励まし合い研究に取り組んだ同輩の皆様のおかげで大変充実した研究生生活を送ることができました．各位のご指導，ご協力に対し心より御礼申し上げ，本論文の締めくくりとさせていただきます．

参考文献

- [1] 国際ロボット連盟. Operational stock of industrial robots - Japan. https://ifr.org/downloads/press2018/2024-SEP-24_IFR_press_release_World_Robotics_2024_-_Japan.pdf.
- [2] Hasan Hosseini-Nasab, Sepideh Fereidouni, Seyyed Mohammad Taghi Fatemi Ghomi, and Mohammad Bagher Fakhrazad. Classification of facility layout problems: a review study. *The International Journal of Advanced Manufacturing Technology*, No. 94, p. 957–977, 2018.
- [3] J.A. Tompkins, J.A. White, Y.A. Bozer, E.H. Frazelle, J.M. Tanchoco, and J. Trevino. *Facilities planning*. Wiley, 1996.
- [4] Miguel F Anjos and Manuel VC Vieira. Mathematical optimization approaches for facility layout problems: The state-of-the-art and future research directions. *European journal of operational research*, Vol. 261, No. 1, pp. 1–16, 2017.
- [5] Weizhe Chen, Zhihan Wang, Jiaoyang Li, Sven Koenig, and Bistra Dilkina. No panacea in planning: Algorithm selection for suboptimal multi-agent path finding. *arXiv preprint arXiv:2404.03554*, 2024.
- [6] Guni Sharon, Roni Stern, Ariel Felner, and Nathan R. Sturtevant. Conflict-based search for optimal multi-agent pathfinding. *Artificial Intelligence*, Vol. 219, , 2015.
- [7] David Silver. Cooperative pathfinding. *First Artificial Intelligence and Interactive Digital Entertainment Conference*, Vol. 1, No. 1, 2005.
- [8] Keisuke Okumura, Manao Machida, Xavier Défago, and Yasumasa Tamura. Priority inheritance with backtracking for iterative multi-agent path finding. *Artificial Intelligence*, Vol. 310, , 2022.
- [9] Guillaume Sartoretti, Justin Kerr, Yunfei Shi, Glenn Wagner, T. K. Satish Kumar, Sven Koenig, and Howie Choset. Primal: Pathfinding via reinforcement and imitation multi-agent learning. *IEEE Robotics and Automation Letters*, Vol. 4, , 2019.
- [10] 谷合竜典, 米谷竜. 機械学習を活用した経路計画技術. *OMRON TECHNICS*, Vol. 53, No. 2, pp. 36–45, 2021.

- [11] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, Vol. 86, No. 11, pp. 2278–2324, 1998.
- [12] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, Vol. 25, , 2012.
- [13] Karen Simonyan. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.
- [14] Herbert Robbins and Sutton Monro. A stochastic approximation method. *The annals of mathematical statistics*, pp. 400–407, 1951.
- [15] Diederik P Kingma. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

