

# 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評価値算出の高速化

## Faster calculation of evaluation values for facility layout optimization considering multiple AMR collisions

○学 望月洋彰 (名古屋大) 杉山俊一郎 (名古屋大)  
正 鈴木達也 (名古屋大) 正 奥田裕之 (名古屋大)

Hiroaki Mochizuki, nagoya University, mochizuki.hiroaki.j0@s.mail.nagoya-u.ac.jp

Shunichiro Sugiyama, nagoya University

Tatsuya Suzuki, nagoya University

Hiroyuki Okuda, nagoya University

This study proposed a direct cost prediction model for the facility layout optimization problem of an environment in which multiple autonomous mobile robots work, combining CNN and A\* algorithm instead of path planning algorithm as a layout evaluation method. The proposed method improves the computational efficiency over the search algorithm and enables collision-aware cost prediction, and the combination of the A\* algorithm achieves stable cost prediction. We also discuss how the accuracy of the proposed method changes as the input data is varied.

**Key Words:** navigation, autonomous mobile robot, facility layout design

### 1 緒言

我が国における少子高齢化の進行は深刻な社会課題となっており、特に製造業においては、工場の有能な働き手の減少が問題視されている。この人材不足を解消するとともに、生産性の向上および品質の安定化を図るために、ロボット技術を活用し工場の自動化が進められている。その一環として、自律走行ロボットである AMR (Autonomous Mobile Robot) や AGV (Automated Guided Vehicle) を活用し、部品搬送の自動化が行われている [1]。

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衝突の可能性が高まり工場全体のコスト増加につながる [2]。施設配置はレイアウトとも呼ばれ、工場の生産性に影響を与えることから、facility layout problem (FLP) の分野では施設配置を最適化するアルゴリズムが多く提案されている [3]。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) であり、これは評価集団に対して、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。

複数のエージェントが活動する環境ではエージェントの移動量が工場の効率に影響するため、移動量に着目して GA における評価基準を定めると、移動量を算出するためには各エージェントの経路を算出し総和をとる必要がある。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、経路計画アルゴリズムとして CBS (Conflict-Based Search) がよく使用されている [4]。しかし CBS は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ると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。そこで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が、強化学習と模倣学習を組み合わせた PRIMAL [5] では、障害物が増えて密度が上がると探索解が増え、計算精度および計算速度が落ちてしまう。

このように機械学習においても経路計画を行うと解の候補が増えることで計算時間が増加してしまう。そこで、経路計画に代わり深層学習による直接的なコスト予測モデルが実現できれば、経

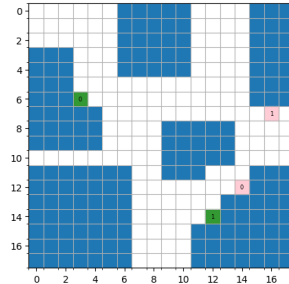


Fig.1: map data

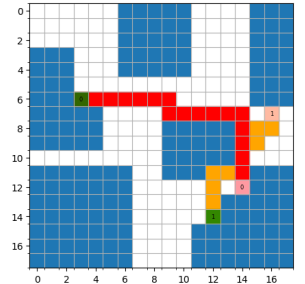


Fig.2: path map

路計画にかかる計算時間を短縮できるのではないかと考えた。そして画像認識に優れた CNN (畳み込みニューラルネットワーク) ではレイアウトの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の関係性を学習できると考え、本研究では、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。

### 2 問題設定

本研究では、図 1 のようなグリッドマップを考え、施設と各エージェントのスタートゴールを配置する。図 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。このようなマップは無向グラフ  $G = (V, E)$  とエージェント集合  $A = \{a_1, \dots, a_k\}$  によって表される。時刻  $t \in \mathbb{N}$  におけるエージェント  $a_i$  の位置を  $\pi_i[t]$  とする。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、次時刻において隣接する頂点に移動するか待機するか行動をとることができる。 $a_i$  の経路を  $\pi_i = (\pi_i[0], \pi_i[1], \dots, \pi_i[T])$  とすると条件を満たした経路の組み合わせ  $\pi = (\pi_1, \dots, \pi_k)$  を求める。エージェント同士は衝突を回避するために頂点上での衝突の禁止 ( $\pi_i[t] \neq \pi_j[t]$ ) とすれ違いの禁止 ( $(\pi_i[t] \neq \pi_j[t+1]) \vee (\pi_i[t+1] \neq \pi_j[t])$ ) を制約として課す。このような制約を満たす  $\pi$  を求めることで図 2 のような経路を算出する。ここで各エージェント  $a_i$  の目的地  $g_i$  に対して時刻  $T$  で  $\pi_i[T] = g_i$  となるような  $\pi$  について各エージェントの合計移動時間  $\sum_{a_i \in A} T_i$  をコストと定義し、マップの効率性の指標として用いる。

## 2.1 従来の MAPF 解法

### 2.1.1 A\*アルゴリズム

従来から広く使用されている経路計画アルゴリズムとして A\* アルゴリズムがあげられる。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も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み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 (1) 式で計算される。

$$f(n) = g(n) + h(n) \quad (1)$$

$g(n)$  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) 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。

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

A\* アルゴリズムはシングルエージェントにおける経路計画手法であるため、A\* アルゴリズムのみでは各エージェントの衝突回避を考慮することができない。そこで A\* のようなシングルエージェントの経路計画と組み合わせることで MAPF を解くアルゴリズムが提案されている。

### 2.1.2 CBS (Conflict-Based Search) [4]

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経路計画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。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。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 CT 上の各ノードでは、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。このノードを CT ノードと呼ぶ。高レベルの探索は以下のように進行する。

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。

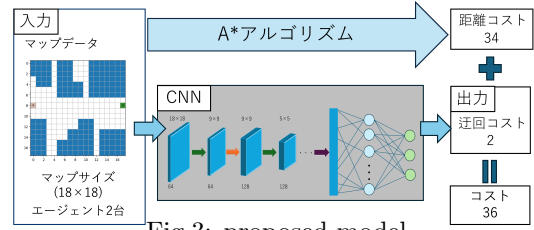


Fig.3: proposed model

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。

CBS は最適解が保証されており、経路計画アルゴリズムとしてよく使用されているが、狭所の多いマップやエージェントの密集した条件では衝突回避制約が増え、探索する解が増えることで全体の計算時間が増加してしまう傾向がある。

## 3 CNN を用いたマルチエージェントの移動コスト推定

経路計画をしてコストを算出すると特定のレイアウトにおいて計算時間が増加してしまうといった問題があった。そこで本研究ではコスト計算における計算時間を短縮するために経路計画に代わり、CNN を用いたコスト推定モデルを提案する。

### 3.1 モデル概要

本研究で作成するコスト推定モデルを図 3 に示す。入力マップから直接コストを予測させると予測値にばらつきが出てしまい誤差が増えるという問題があった。そこでコスト全体を予測するのではなく衝突によるコスト増加のみを予測させることで CNN が予測する情報が減り、予測精度が安定するのではないかと考え、本研究では CNN と A\* アルゴリズムを組み合わせたコスト予測モデルを提案する。ここで提案手法で用いる最適コスト、距離コスト、迂回コストについて説明する。

#### 最適コスト

最適コストは、CBS によって計算された最適経路におけるコストであり、マップに対するコストの正解として使用する。

#### 距離コスト (Distance Cost, DC)

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距離コストと呼ぶ。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回避による迂回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可能性がある。

#### 迂回コスト (Circumvention Cost, CC)

最適コストと距離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を迂回コストと呼ぶ。本研究ではこの迂回コストを予測することで最適コストを推定する。

### 3.2 コスト予測までの流れ

提案手法では、マップ入力に対して CNN によって迂回コストを推定し、A\* アルゴリズムによって求まる距離コストとの和を取ることでマップのコストを推定する。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、ラベルとして迂回コストを学習させる
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迂回コストを推定する
3. テストデータについて各エージェントの距離コストを A\* アルゴリズムで算出する
4. CNN による出力と距離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

### 3.3 CNN (畳み込みニューラルネットワーク)

CNN 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。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、AlexNet[6]、VGGNet[7] 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。

### 3.3.1 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。

### 3.3.2 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。

### 3.3.3 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。

### 3.3.4 活性化関数

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、線形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、(3) 式で表される。

$$y = \max(0, x) \quad (3)$$

### 3.3.5 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。

$$L_{\text{MSE}}(\hat{y}, y) = \frac{1}{K} \sum_{n=1}^K (\hat{y} - y)^2 \quad (4)$$

### 3.3.6 勾配降下法

推定データと正解データ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、フィルタの重みや全結合層の重みを更新している。重みを  $\omega$ 、誤差関数  $E(\omega)$  とするとき、勾配降下法では  $\nabla E$  を用いて式 (5) で更新を行う。  $t$  は反復回数、 $\epsilon$  は学習係数を表す。

$$\omega^{(t+1)} = \omega^{(t)} - \epsilon \nabla E \quad (5)$$

勾配降下法をそのまま適用しようとすると訓練データの数が増えるため、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)[8] や SGD を応用した adam (adaptive moment)[9] が最適化アルゴリズムとして用いられている。

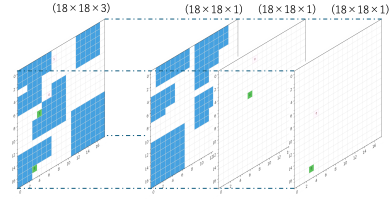


Fig.4: スタートゴールマップ (SG)

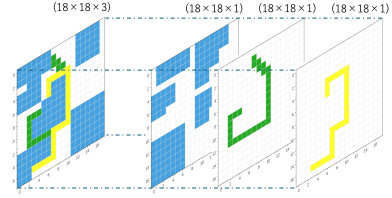


Fig.5: 経路マップ (P)

## 4 迂回コスト推定のための入力情報

本研究では、進化的アルゴリズムを使用して施設の重複がないマップデータを生成する。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り、以下の手順でマップデータを生成する。まず、任意の施設形状を作成し、ランダムに施設や各エージェントのスタート地点とゴール地点を配置したマップデータを作成する。次に、施設やスタート地点とゴール地点に重複がないかを確認し、重複がある場合は再配置を行う。その後、配置したマップ集団から配置位置などの要素を抽出し、集団内の様々なマップからこれらの要素を組み合わせる新たなマップを作成する。また、一定確率で突然変異を発生させ、位置などをランダムに変更し、集団には存在しない新たな配置を行う。このような流れで施設の重複がないマップデータを効率的に生成する。

本研究では、CNN に入力するためのマップデータを、障害物の位置や各エージェントのスタート・ゴール位置を含む  $H \times W \times C$  次元のマトリクスとして表現する。学習タスクを簡略化するため、各種情報を異なるチャンネルに分割し、それぞれを CNN に入力させることで学習を行う。具体的には、図 4 に示すように、障害物マップ、各エージェントのスタートマップ、ゴールマップをそれぞれ個別のチャンネルとして配置し、これらを重ね合わせることで一つの入力データを構成する。このように情報を分離した形で入力することで、ネットワークが各情報を効果的に認識し、学習・予測を行うことが可能となる。今回使用する障害物マップ  $O(i, j, 1)$ 、スタートマップ  $S(i, j, 1)$ 、ゴールマップ  $G(i, j, 1)$  は以下のように定義される。

$$O(i, j, 1) = \begin{cases} 1 & \text{障害物がある場合} \\ 0 & \text{通行可能な場合} \end{cases}$$

$$S(i, j, 1) = \begin{cases} 1 & \text{各エージェントのスタート位置} \\ 0 & \text{それ以外の位置} \end{cases}$$

$$G(i, j, 1) = \begin{cases} 1 & \text{各エージェントのゴール位置} \\ 0 & \text{それ以外の位置} \end{cases}$$

またこのようなマップに対し各エージェントの衝突を考慮するために異なる入力データ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。ここでエージェントの経路情報があると CNN が衝突をしやすいレイアウトを判断しやすくなると考え、各エージェントの A\*によって求められる経路情報を含む A\*経路マップ  $P_k(i, j, 1)$  を作成した。

$$P_k(i, j, 1) = \begin{cases} 1 & \text{エージェント k の A*経路に含まれる場合} \\ 0 & \text{それ以外の位置} \end{cases}$$

スタート・ゴールマップの代わりに、各エージェントの A\*探索によって得られた経路マップ  $P_1, \dots, P_k$  を入力に加え、図 5 に示すような入力マップを作成する。図 4 のような障害物とスター

Table 1: Map for each method

名称	マップ情報	マップサイズ
CNN(P)	$O + P_1 + P_2$	$18 \times 18 \times 3$
DC+CC(SG)	$O + S + G$	$18 \times 18 \times 3$
DC+CC(P)	$O + P_1 + P_2$	$18 \times 18 \times 3$

Table 2: valuation index

名称	平均誤差	計算時間 (s)
CBS	0	3.093
CNN(P)	0.89	0.605
DC(A*)	0.41	0.015
DC+CC(SG)	0.44	0.495
DC+CC(P)	0.31	0.495

トゴールの情報を含む入力データをスタートゴールマップ (SG), 図5のような障害物と各エージェントのA\*経路を含むマップを経路マップ (P) として表す. これらの入力データを用いて学習を行い, 異なる種類の入力データを持つ2つのCNNモデルを構築する. 各入力データからCNNによって予測される迂回コストをCC(SG), CC(P)とし, これらをDCと組み合わせることで最終的なコスト予測を行う. また, 入力データから直接コストを予測するモデルも構築し, 経路マップを入力データとしてコストを予測させる. A\*を用いずに迂回コストではなく直接コストを予測するモデルをCNN(P)とし, 提案手法との比較を行う.

## 5 提案手法の評価

### 5.1 評価時のモデル設定

今回モデルは畳み込み層が3層で, 入力から順にフィルタ数は64, 128, 256である. 全結合層は2層ありユニット数は1024, 512であり, 最後に出力層が続く. 畳み込みにおけるフィルタサイズを3, プーリング層のフィルタサイズを2, ストライドはそれぞれ2として作成した. 誤差関数としてMSE, バッチサイズは32, エポック数30, 最適化アルゴリズムとしてadamを用いた.

今回のモデル実装においてマップ情報を簡略化するためにマップサイズを $(18 \times 18)$ , エージェントは2台に固定する. 各手法と対応する入力マップについては, 表1にまとめる. 訓練データ, 検証データ, テストデータとして22887, 2543, 1922データのマップデータを作成して学習, 予測に用いる.

### 5.2 評価結果と考察

学習データと同様に生成したテストデータ, 1922データを用いて予測を行った結果をまとめる. 直接的なコスト予測モデルCNN(P)と距離コスト, 提案手法における予測コスト分布を図6, 7, 8, 9に示す. 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各手法の計算時間を表2にまとめる.

本研究の提案手法では, CBSと比較して計算時間を短縮することができた. CBSでは, 探索空間が増加するにつれて計算時間が指数関数的に増大する. 一方, CNNによる予測はエージェントの台数や衝突回避制約に依存せず, 主に入力データのサイズとデータ数に依存する. そのため, 探索空間の増大による計算時間の増加という問題が, 提案手法には存在しない. この特性により, 提案手法はCBSよりも計算時間を短縮できた. 現段階では, エージェントの台数を2台に固定した条件で検証を行ったが, エージェント数を変化させた場合の予測を行うことで, 提案手法がCBSに比べてスケーラビリティの問題を抱えないことを示すことができると考えられる. また, CNN(P)と提案手法を比較した結果, 提案手法の方が平均誤差が小さく, 全体的として安定したコスト予測が可能となった. 提案手法では, 距離コストを導入することで, CNNは迂回によるコスト増加のみを学習すればよく, コスト全体を直接予測する必要がない. そのため距離コストと組み合わせた提案手法は, 平均誤差が減少し, 安定したコスト予測を実現した. さらに, 提案手法における入力マップの違いによる予測精度の比較を行うと, A\*経路マップを加えたモデルでは, スタート・ゴール情報のみを用いた場合と比較して平均誤差が減少した. このことから, 経路情報が予測精度の向上に寄与することが確認できた. A\*経路の時点で衝突がないことが経路マップに明示されているため, 本来迂回コストが0であるケー

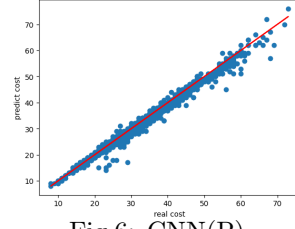


Fig.6: CNN(P)

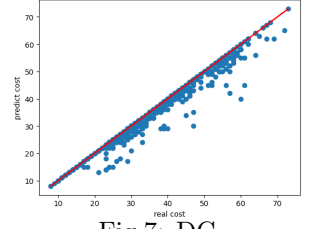


Fig.7: DC

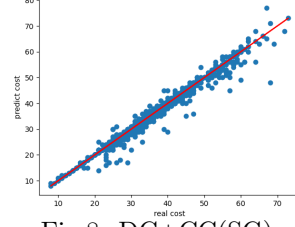


Fig.8: DC+CC(SG)

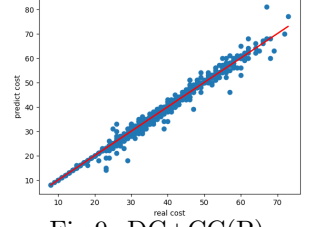


Fig.9: DC+CC(P)

スにおいて過大な迂回コストを予測する誤りが減少し, 結果として平均誤差の低減につながったと考えられる.

## 6 結言

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\*とCNNを組み合わせることでCBSよりも高速に, そしてCNN(P)よりも安定して予測させることに成功した. また, 入力データに経路マップのような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も可能にした. 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, 今後は2台ではなくさらに台数を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検証していきたいと考えている.

## 参考文献

- [1] 2023年版 世界AGV / AMR市場の現状と将来展望. 矢野経済研究所, 2023.
- [2] JA Tompkins, JA White, YA Bozer, EH Frazelle, JMA Tanchoco, and J Trevino. Facility planning, 1996.
- [3] 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*, Vol. 94, pp. 957–977, 2018.
- [4] Guni Sharon, Roni Stern, Ariel Felner, and Nathan R Sturtevant. Conflict-based search for optimal multi-agent pathfinding. *Artificial intelligence*, Vol. 219, pp. 40–66, 2015.
- [5] Guillaume Sartoretti, Justin Kerr, Yunfei Shi, Glenn Wagner, TK Satish Kumar, Sven Koenig, and Howie Choset. Primal: Pathfinding via reinforcement and imitation multi-agent learning. *IEEE Robotics and Automation Letters*, Vol. 4, No. 3, pp. 2378–2385, 2019.
- [6] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, Vol. 25, , 2012.
- [7] Karen Simonyan. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.
- [8] Herbert Robbins and Sutton Monro. A stochastic approximation method. *The annals of mathematical statistics*, pp. 400–407, 1951.
- [9] Diederik P Kingma. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.