F-051

歩行者を加味した深層強化学習による信号制御手法の提案
Signal control by deep reinforcement learning taking pedestrians into account

村田　顕祐† 清　雄一† 田原　康之† 大須賀　昭彦†
Akimasa Murata Yuichi Sei Yasuyuki Tahara Akihiko Ohsuga

1. はじめに

現代の 交通環境では車両や歩行者などが入り混じる環境であり, 環境に適した信号機が用いられることによって交通制御がなされている. それに対し, 強化学習手法の１つである Deep Q-Network(DQN) [1] を用いることによって信号の制御方針を学習させ, リアルタイムの交通変化に対応可能な信号機の作成を目指した研究が行われている. それらの研究の多くは車両に重きを置いていることから, 我々の先行研究 [2] では歩行者の存在に着目した制御を試みた. 過去の隣接信号の制御情報を含めることによって歩行者の交通量の変化に対応可能な交通制御を目指し, 歩行者の道路上での待機時間の削減が可能であることを示した. しかし, 隣接信号の情報のみでは, 歩行者の規模や集団の移動先など, 人流の把握が困難であるため適切な交通制御に繋がらない.

本研究では歩行者に対する制御に重きをおき, 環境の変化に対応可能な交通制御を行うことを目的とする. その上で, 環境内の車両及び歩行者の一定時間の環境における情報を取得し, LSTM を用いたネットワークによって時間的な変化を利用した制御の可否及びその精度を検証していく.

深層強化学習を用い, 現在の環境を基に制御を行う信号機である DTC, 環境の過去の情報を利用して制御する信号機である LTC を作成し, その精度を評価することとした. 歩行者の交通量が大きく変化する 3 つの交通環境において, 信号を順番に切り替える信号機である KTC と比較を行った. その結果, 歩行者が多い環境において, KTC に比べ DTC は歩行者の待機時間を約 77%減少させ, LTC は約 82%減少させた. 車両の待機時間の増加も確認されたが, DTC は車両と歩行者全体の待機時間を約 28%の削減しており, 環境の状態に応じた制御が行えていると判断できる.

2. 関連研究

2.1 DQN

強化学習手法の 1 つに Q 学習がある. Q 学習はある状態での行動にどのくらいの価値があるかを Q テーブルといわれる表を用いて扱う学習方法である. Q テーブル上の値である Q 値から, ある状態での行動の価値を定め, それに従って次の行動を選択していく. Q 値は行動を行うごとに, 状態毎の行動の価値とその後の行動の価値から更新がなされる. しかし, Q 値を推定するための Q テーブルの作成は, 状態数や行動数が多くなるにつれて複雑化してしまう. DQN はこの Q 値の推定にニューラルネットワークを取り入れものである. Q 値の推定のための関数の代用としてニューラルネットワークを用いることで, 安定した高精度な推定を行うことができる[1].

2.2 LSTM

LSTM は RNN(Recurrent Neural Network)の発展として 1995 年に提案され[3], 改善がなされてきた学習手法である[4]. RNN は再帰的なネットワーク構造を持ち, 時系列データの取扱いに適した学習方法であるが, 長期的なデータの取り扱いに対して勾配が消失する問題が生じる. それに対して, LSTM は RNN と異なり, 長期的な依存関係も学習することができるため時系列データの学習により適している.

2.3 既存の交通制御研究

強化学習によって学習をさせた信号機による交通制御を行う研究は, 交通シミュレータを用いて行われている場合が多い. 主に使用されているシミュレータとして, ドイツ航空宇宙センターによって提供されている Simulation of Urban MObility(SUMO)[5] が挙げられる. 交通制御を行う上で使用される手法としては, 交通網の車両の有無によって得たテンソルを利用することによって, 信号機をエージェントとした学習を行うものである[6][7]. 取得した交通状態を表すテンソルをニューラルネットワークに渡すことにより, 行動を選択する. 行動による状態の変化に応じて行動価値が決定され, それを利用することによりネットワークの更新を行っていく. また, 同様の学習方法を用いたマルチエージェントシステムによる学習に対し隣接信号機の Q 値を転移させることによって Q 値を更新していく協調的なシステム[8]や, LSTM を導入した研究もなされている[9].

3. 提案手法

3.1 概要

本研究では, 道路上に車両及び歩行者が存在し, その交通量が大きく変化する環境において信号制御を行うものとする. 交通シミュレータである SUMO を用いることによって交通量が変化する環境を作成し, そこから環境の交通状態を取得する. その結果を基に深層強化学習を行うものとする. 交通シミュレーションにおいては青信号, 黄信号の持続時間を一定に設けており, その合計時間をkとする. 信号機は時間 kごとに交通状態（車両, 歩行者の位置情報）の取得, 行動の選択, 強化学習における報酬の計算などを行う.

また, 以下の記述において時間k毎に増加する値をステップtとし, 取得した値の一覧を表 3.1 で示す.

表 3.1: 記号一覧

記号	意味
s_t	状態配列
w_t	停止中における待ち時間
r_t	報酬の値
a_t	選択された行動
M	経験メモリ

3.2 状態表現

信号機は時間k毎に環境に存在する車両，歩行者の位置情報を取得する．信号付近の道路を予め定めた距離に分割し，分割範囲内を移動または停止している車両数及び 歩行者数に応じて配列を作成することにより環境の状態を表現する．さらに，車両の右左折時の可否を判断可能とするために交差点の横断歩道上に存在する歩行者数を配列に加えるものとしている．ステップtにおける環境の状態配列s_tは，道路の分割範囲 jでの車両または歩行者の数をN_t^jとした 104 の要素数を持つ配列として，$s_t = \{N_t^1, N_t^2, \cdots, N_t^{104}\}$と表される．

道路の分割範囲は信号から遠くなるほど広くなるようにしており，特に歩行者の分割範囲は，車両に比べて細分化することとしている（図 3.2）．これは環境内での歩行者一人一人の大きさが車両に比べ小さいことから，分割範囲内における距離に応じた歩行者の密集度を明確化させることを目的としている．

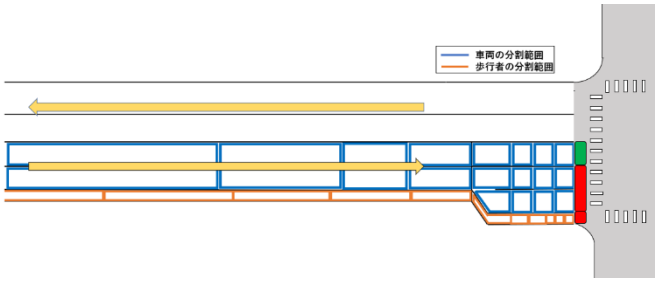

図 3.2 道路の分割範囲

3.3 行動選択

信号機は時間k毎に環境から得た状態配列s_tを用いて行動を選択する．行動は，現在の信号パターンを維持または図 3.3 に示す信号パターンのいずれかに変更をするものとしている．交通量が少ない環境においてはP_0, P_1のみでも単純な制御は可能であるが，歩行者の交通量が多い環境においては車両の右左折が制限される場合があるため，P_2, P_3を設けることで対応可能となるようにしている．また，車両または歩行者のどちらか一方の交通量が極端に多くなる場合を想定し，P_4, P_5, P_6を設けている．

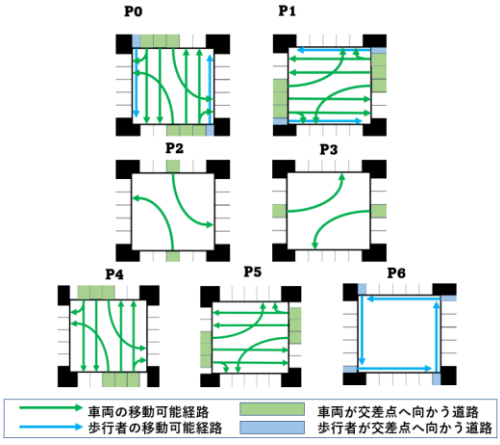

図 3.3 信号パターン

3.4 報酬設計

本研究における深層強化学習時の報酬の値は，主に車両及び歩行者の交差点付近での待機時間によって決定される．時間t毎に停止中である車両，歩行者の待機時間が取得される．ステップtにおける車両の待機時間をCW_t，歩行者の待機時間をPW_tとして報酬関数R_tは以下の式 3.4 で表される．

$$R_t = (CW_{t-1} - CW_t,) + \alpha(PW_{t-1} - PW_t) + \beta EM_t \quad \cdots \quad (3.4)$$

ここでαおよびβは任意の定数である．定数αは，環境内での歩行者に対する制御の重要度を調整するために設けている．また，式中のEM_tは，前のステップ$t-1$から現在のステップtまでに車両が起こした急ブレーキ数によって与えられる値である．信号の変化のタイミングで横断歩道上に歩行者が存在することによって車両が急ブレーキをかける場合が生じる．急ブレーキは事故を引き起こす基となると捉え，その数を抑制する信号制御を可能とすることを目的として，急ブレーキ数に応じて報酬の値を減少させる数値となるようにEM_tを設けた．

3.5 ネットワーク

本研究で用いるネットワークは，5 層の全結合層で構成される．また，時間的な情報を用いた交通制御を行わせるために LSTM を含めたネットワークを用意した(図 3.5)．このネットワークは前述の全結合層のみで構成されるネットワークの 1 層目を LSTM 層に変更したものである．

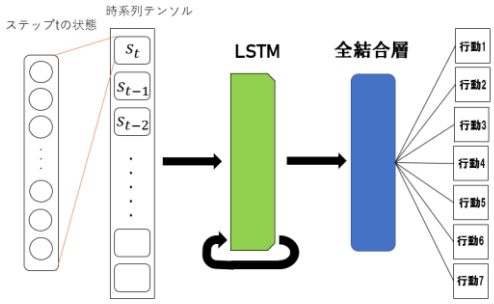

図 3.5 ネットワークのイメージ

このネットワークにおいては数ステップ間の状態配列s_tを要素としたテンソルを入力としており，環境の状態に応じて 3 章 2 節で示した信号パターンのいずれかを出力として得る．

3.6 学習アルゴリズム

学習のアルゴリズムは，以下の Algorithm1 及び Algorithm2 となる．Algorithm1 は DQN による学習を行う流れを示しており，Andrea Vidali が公開を行っているアルゴリズム[7]が基となっている．Algorithm2 は，LSTM を含めた学習の流れを示している．時間k毎に現在の状態，前ステップの状態，報酬，信号が選択した行動を保存し，一定ステップが終了した後に経験再生によってネットワークが更新される．

Algorithm2 においては Steven Kapturowski らの手法[10]を取り入れている．LSTM に利用においては短期記憶の初期状態が重要な要素となるが，シミュレーションや学習時にこ

の初期状態が初期化されてしまう問題が生じる. そのため, ステップtごとに LSTM 層の初期状態を経験に保存し, 学習の際に利用することとしている. 数ステップ分の時系列テンソルを一定ステップ間学習には使用せずにネットワークに通すようにし, LSTM の初期状態がシミュレーション時に時系列テンソルを取得した際に近い状態になるようにした.

Algorithm 1 Learning Algorithm

```
for episode = 1 to N do
  while step < max_step do
    get traffic states and waiting_time: s_t, w_t
    estimate reward : r_t
    update old adjacent signal action
    if len(M) > max_size − 1 then
      delete M[0]
    end if
    append experience = (s_t, a_t, r_t, s_{t−1})
    select action with ϵ−greedy : a_t
    yellow phase and green phase
    update s_{t−1} = s_t, a_{t−1} = a_t
  end while
  update target network
  repeat training epochs do
    get batch size experiences
    update network parameters
  end repeat
end for
```

Algorithm 2 Learning Algorithm using time series data

```
for episode = 1 to N do
  while step < max_step do
    get traffic states and waiting_time: s_t, w_t
    estimate reward : r_t
    get hidden state : h_{t−2}
    que and pop time step states : ls_t
    if ls_t > (max_ls_size) − 1 then
      if len(M) > max_size − 1 then
        delete M[0]
      end if
      append experience = (ols_t, oa_t, r_t, ls_t, h_t)
      select action with ϵ−greedy : a_t
    else
      select action randomly
    end if
    yellow phase and green phase
    update ols_t = ls_t, oa_t = a_t
  end while
  repeat training epochs do
    get batch size experiences
    burn in process
    update network parameters
  end repeat
end for
```

4. 実験

4.1 実験環境

本研究において, SUMO を用いたシミュレーションの実施を行い, 信号機の精度の評価を行った. シミュレーションは, SUMO 上における 4000step を 1episode とし, 100episode まで行った.

シミュレーション及び学習を行う環境として図 4.1 に示す十字路を用意した. この環境においては車両と歩行者が進行する道路は分かれており, 交差点上の横断歩道を除く地点で衝突をすることはない.

シミュレーション時, 1episode が実行される段階でその episode における車両, 歩行者の生成数が決定される. 1step 毎に車両, 歩行者が一定の確率に従い生成され, 同時に決定される目的地に移動を開始する. 車両は全 episode を通して同数になるように生成数が定められているが, 歩行者においては一定 step 毎に生成数が増減するように設定した. これにより時間帯によって歩行者数に大きく差が生じるようにし, 信号機が交通量の変化に対応可能かを検証することとした.

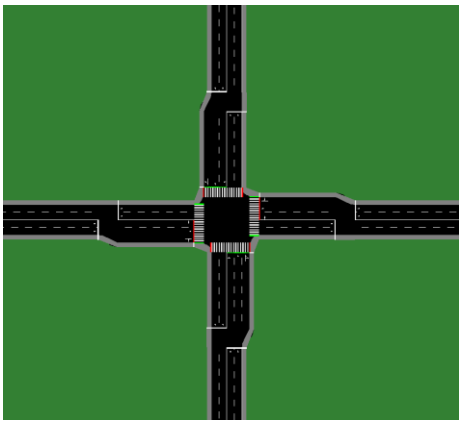

図 4.1　シミュレーション環境

4.2 評価実験

4.2.1 実験内容

学習させた信号機エージェントの性能の評価を行うために, 100episode までのシミュレーションを行い, 車両及び歩行者の待機時間の平均値を算出することによって評価を行った. 実験においては車両に対して歩行者の生成数が 3 倍, 4 倍, 5 倍となる交通パターンを用意し, それぞれ低密度, 中密度, 高密度交通環境と称し評価を行うものとする.

性能の評価においては学習を行った信号機である DTC 及び LTC と KTC と称した信号機による比較を行った. DTC は, DQN によって現在の当該交差点の状態を利用して制御を行う信号機であり, LTC は LSTM を用いたネットワークによって当該交差点とその数ステップ前の状態を利用して制御を行う信号機である. KTC は一定時間ごとに定められた順に信号の色を切り替える信号機であり, 交通環境に関わらず安定した制御を行う.

4.2.2 実験結果

以下の表 4.2, 表 4.3,表 4.4 に実験結果を示す. なお, 待機時間はシミュレータ上の時間を表すものであり, 値が小さい程適切に環境に対して交通制御が行えているものとする.

表 4.2: 車両の待機時間

制御手法	低密度 $\times 10^2 s$	中密度 $\times 10^2 s$	高密度 $\times 10^2 s$
DTC	128.1	147.6	178.3
LTC	571.9	629.2	792.4
KTC	64.8	67.1	69.3

表 4.3: 歩行者の待機時間

制御手法	低密度 $\times 10^2 s$	中密度 $\times 10^2 s$	高密度 $\times 10^2 s$
DTC	34.7	42.8	67.0
LTC	23.6	51.4	43.9
KTC	153.4	228.3	259.1

表 4.4: 全待機時間

制御手法	低密度 $\times 10^2 s$	中密度 $\times 10^2 s$	高密度 $\times 10^2 s$
DTC	162.7	190.5	245.3
LTC	595.8	680.7	836.3
KTC	218.2	295.4	328.4

5. 考察

実験結果である表 4.2 から表 4.4 に対して学習させた信号機の精度に対する評価, 考察を行う.

表 4.2 より, 安定した制御を行う KTC に対し, 学習させた信号機である DTC は各密度帯で平均して約 124%, LTC は約 888%の待機時間が増加していた. それに対し, 表 4.3 より KTC に比べ, 各密度帯で平均して DTC は約 77%, LTC は約 82%の歩行者の待機時間を減少していた. このことから学習させた信号機は歩行者の待機時間を削減するための交通制御を行っていることがわかる. また, 表 4.4 より, DTC は, KTC に比べて各密度帯で平均して約 28%の待機時間を減少させていた. この結果から, DTC は歩行者に重きを置いた制御を行った上で車両を含めた全体の信号による待機時間の改善に繋げていく制御を行っていると判断することができる.

表 4.3 より学習させた 2 つの信号機では, LTC の方が歩行者の多い高密度帯での待機時間が短いことがわかるが, 表 4.2 より LTC の車両の待機時間が大幅に増加していることがわかる.これは LTC がより歩行者の待機時間の削減を重視した制御を行ったことが要因であると言える. 3 章 4 節より報酬関数内に定数αを設けることによって環境内での歩行者の重要度の調整を行っている. この値が高いことにより, 歩行者の待機時間の減少に繋がる行動をとり続けることが報酬の値を増加させる単純な方法であると学習してしまっただと考えられる. そのため歩行者だけでなく車両を含めた双方の待機時間の減少に繋げる報酬設計が必要であると言える. また今回用意した道路環境の存在においても車両を含めた全体の待機時間が増加する要因となったと推測できる. 用意した環境は十字路のみの環境であることから, 進行してくる車両や歩行者の規模の把握が容易な環境であると言える. そのような環境においては現在の環境の状態を用いる DTC の方が環境の状況を反映した制御が行いやすいと判断できる. それに対し, LTC は交通環境の時間的な変化をデータとして扱っていることから, 車両や歩行者の進行方向をその場の交通状態だけでは判断が難しいような複雑な道路環境において, より良い制御に繋げていけるのでないかと考えられる.

6. おわりに

6.1 本研究のまとめ

本研究では, 車両と歩行者が入り混じる環境において, 深層強化学習によって学習させた信号機によって交通制御を行うことを目指した. 信号機の学習においては現在の環境の状態を用いるネットワークと過去の状態を含めた時系列データを用いるネットワークによって行い, 学習を行った信号機の制御の可否及びその精度を検証した.

その結果, 歩行者の交差点上での待機時間の削減を行うことが可能であると確認され, 特に現在の環境の状態を用いた信号機においては車両を含めた総合的な待機時間の減少に繋がった.

6.2 今後の課題

今後は学習及び実験環境の拡張を行い, 大規模交通網においての交通制御が可能となる信号機の作成を行っていく.

また, 時系列情報を用いた信号機による制御の改善を行っていく. 環境から取得する状態の増減, 行動の選択に用いる過去の状態の調整によって適切な制御を試みていく. その上で十字路のような単純な構造のみではなく, 複雑な道路環境において, 歩行者集団の規模や進行方向などの人流を利用した制御を行うようにする. そのようにして環境の状態により適した対応を可能とする信号機の実現に繋げていきたいと考えている.

謝辞

本研究は JSPS 科研費 JP21H03496, JP22K12157 の助成を受けたものです.

参考文献

[1] Volodymyr Mnih, Koray Kavukcuoglu, Davit Silver, Alex Graves, loannis Antonoglou, Daan Wierstra, Matin Riedmiller, "Playing Atari with Deep Reinforcement Learning", arXiv preprint arXiv:1312.5602,2013
[2] [2] 村田顕祐, 清雄一, 田原康之, 大須賀昭彦, “歩行者を加味した深層強化学習による信号制御手法の提案”, IEICE-AI2021-9, IEICE-121, no.298, pp.46-51
[3] Sepp Hochreiter, Jurgen Schmidhuber, "LONG SHORT-TERM MEMORY", Neural Computation 9(8): 1735-1780, 1997
[4] Felix A. Gers, Jurgen Schmidhuber and Fred Cummins, "Learning to forget: continual prediction with LSTM", Neural Computation 12(10), 2451-2471,2000
[5] SUMO(Simulation of Urban MObility) https://www.dlr.de/ts/en/desktopdefault.aspx/tabid-9883/16931_read-41000(2022/06/10 参照)
[6] Elise van der Pol, Frans A. Oliehoek, "Coordinated Deep Reinforcement Learners for Traffic Light Control", NIPS'16 Workshop on Learning, Infererence and Control of Muiti-Agent System
[7] Andrea Vidali, "Simulation of a traffic light scenario controlled by Deep Reinforcement Learning agent", https://github.com/AndreaVidali/Deep-QLearning-Agent-for-Traffic-Signal-Control(2022/06/10 参照)
[8] Hongwei Ge, Yumei Song, Chunguo Wu, Jiankang Ren, Guozhen Tan, "Cooperative Deep Q-Learning With Q-Value Transfer for Multi-Intersection Signal Control", 2019 IEEE Access 2907618
[9] Chung-Jae Choe, Seungho Baek, Bongyoung Woon, and Seung-Hyun Kong, "Deep Q Learining with LSTM for Traffic Light Control", 2018 24th Asia-Pacific Conference on Communications(APCC)
[10] Steven Kapturowski, Georg Ostrovsk, John Quan, Remi Menos, Will Dabney, “Recurrent Experience Replay In Distributed Reinforcement Learning”, ICLR 2019