深層強化学習を用いた動的ネットワーク混雑課金

佐藤　公洋[1]・瀬尾　亨[2]・布施　孝志[3]

[1]学生会員　東京大学大学院　工学系研究科社会基盤学専攻（〒 113-8656 東京都文京区本郷 7-3-1）
E-mail: sato@trip.t.u-tokyo.ac.jp
[2]正会員　東京工業大学環境・社会理工学院准教授　土木・環境工学系（〒 152-8552 東京都目黒区大岡山 2-12-1）
E-mail: seo.t.aa@m.titech.ac.jp (Corresponding Author)
[3]正会員　東京大学大学院教授　工学系研究科社会基盤学専攻（〒 113-8656 東京都文京区本郷 7-3-1）
E-mail: fuse@civil.t.u-tokyo.ac.jp

交通渋滞を緩和する施策として，1 日の中での交通需要の変動に応じて課金額を設定する動的混雑課金の有用性が提唱されている．しかし，現実の道路ネットワークは大規模かつ複雑である上に，課金主体と道路利用者間に情報の非対称性があるため，最適な動的混雑課金額の決定は困難である．本研究では，深層強化学習を用いた，一般の大規模道路ネットワークにおいて観測可能なデータに基づき渋滞を解消する分散協調制御型の動的混雑課金額決定手法を提案する．具体的には，時空間的に分散した深層強化学習エージェントの実装を行い，エージェント間の協調のために「報酬の協調項」と「学習のオン/オフ」を導入することで，高速・高効率の学習を可能とする．Sioux Falls Network を用いた数値実験により，提案手法の有効性が確認された．

Key Words: *dynamic congestion toll, trial-and-error, deep reinforcement learning, day-to-day dynamics*

1. はじめに

都市部において，交通渋滞は交通安全や環境へのダメージのみならず，移動時間の浪費による経済的なダメージももたらす深刻な問題であり，交通渋滞を緩和・解消する適切な制御手法が求められている．ソフト面での制御手法の 1 つとして，1 日の中での交通需要の変動に応じて課金額を設定する動的混雑課金が注目されている (Vickrey[1)], Arnott et al.[2)], 赤松[3)], 桑原・赤松[4)], Zhu and Ukkusuri[5)], Qiu et al.[6)])．渋滞は主に交通需要の空間的・時間的な集中によって引き起こされるため，動的混雑課金では，ある時間帯に特定の道路を通行する道路利用者に適切な通行料を課金し，需要の集中を緩和する．また，道路ネットワークの形状が単純であり，道路利用者による出発時刻・経路選択のための経済計算が既知である場合は，最適な動的渋滞課金による渋滞の解消が理論的に可能であることが示されている [1)]．

しかし，現実の道路ネットワークは大規模かつ複雑であると共に，課金主体による道路利用者の経済計算の把握は困難であるという情報の非対称性が存在するため，最適な動的混雑課金額の理論的な決定は非常に困難である．情報の非対称性に対処するため，Trial-and-error による動的混雑課金額決定手法が提案されている (Yang et al. [7)], Seo and Yin [8)], 佐藤ら [9)])．これらの手法は，何らかの混雑課金を課した際の交通状態を観測し，観測データに基づく課金額調整を繰り返し行い，課金額を最適に近付けるものである．しかし，大規模なネットワークでは Trial-and-error に 1 年以上の長期間を要する場合があり，実用性の面で課題が残る．効率的な Trial-and-error のためのデータ駆動型アプローチとして，深層強化学習が考えられる [10)]．深層強化学習には，膨大な観測データを用いた傾向の自動抽出，および調整過程の自動最適化が可能であるというメリットがある．これらのメリットは，大規模な道路ネットワークにおける動的混雑課金額の最適化に有用である。

本研究では，深層強化学習を用いた，一般の大規模な道路ネットワークにおいて観測データ（例：旅行時間）のみに基づき課金額を日々決定する分散協調制御型の動的混雑課金額決定手法を構築する．具体的には，深層強化学習エージェントを時空間的に分散して実装し，エージェント間の協調のために「報酬の協調項」,「学習のオン/オフ」を提案する．また，出発時刻・経路が同時に日々選択される交通モデルを用いてシミュレーションを行い，提案手法の性質を検証する．

本稿の構成は以下の通りである．第 2 章にて制御手法を構築する．具体的には，交通モデルを定義し，深層強化学習を用いた分散協調制御型の動的混雑課金額決定手法を提案する．第 3 章にて制御手法をシミュレーション実験により検証する．具体的には，Sioux Falls Network を用いたシミュレーション，および単純な形状の交通

モデルを用いた他手法との比較を行う．第 4 章にて結論と今後の課題をまとめる．

2. 制御手法の構築

(1) 交通モデル

本研究では，一般の動的交通ネットワークを考える．具体的な定義は佐藤ら [9] を参考にした．具体的な交通ネットワークとしては，複数のノードと複数のリンクから構成され，任意のリンク上にボトルネックがあり，ボトルネック上でのみ Point Queue の待ち行列が形成されるとする．OD ペアは複数あり，OD 交通量は固定と仮定する．通常の出発時刻選択問題の設定で用いられるボトルネックモデル[1] を複数 OD のネットワークへと拡張したモデルである．道路利用者の出発時刻は，前日の経験を踏まえて日々決定する Day-to-day dynamics に基づくモデルとする．道路利用者の経路は，当日の情報に基づき決定されるとする．これらのモデルは，概ね既存の出発時刻選択問題等で用いられるもの[1),11),12)] と同様である．技術的な詳細は以下の通りである．

a) ネットワーク交通流モデル

ネットワーク交通流モデルは一般の複数 OD の Point Queue モデルとする．数学記法は以下の通りである．

j: 日数
t: 1 日の中での時刻
M: 1 日当たりの総旅行者数
t^*: 旅行者の勤務地への希望到着時刻
$T_{z,i}(t)$: 経路 z を通る場合の，ボトルネック i を出てから時刻 t に勤務地に到着するまでの所要時間
μ_i: ボトルネック i の単位時間当たりの容量
$a_{j,i}(t)$: j 日目におけるボトルネック i への流入率（ボトルネック流出時刻ベースで定義）
$N_{j,i}(t)$: j 日目, ボトルネック i における時刻 t の待ち行列台数（ボトルネック流出時刻ベースで定義）
$w_{j,i}(t)$: j 日目, ボトルネック i において，時刻 t に流出する旅行者のボトルネックでの待ち時間
$\tau_{j,i}(t)$: j 日目, ボトルネック i において，時刻 t に流出する旅行者に課される課金額
B_z: 経路 z 上の全ボトルネックの集合
Z_ζ: OD ペア ζ を結ぶ全経路の集合

j 日目，時刻 t におけるボトルネック i での待ち行列台数の変化を式 (1) のように表す．

$$\frac{\mathrm{d}N_{j,i}(t)}{\mathrm{d}t} = \begin{cases} 0 & (N_{j,i}(t)=0 \text{ and } a_{j,i}(t)<\mu_i) \\ a_{j,i}(t)-\mu_i & (\text{otherwise}) \end{cases} \tag{1}$$

ここで，t はボトルネック i からの流出時刻を示す．

j 日目，時刻 t におけるボトルネック i での待ち時間を式 (2) のように表す．

$$w_{j,i}(t) = \frac{N_{j,i}(t)}{\mu_i} \tag{2}$$

b) 旅行者のコスト

j 日目，時刻 t における経路 z での 1 人当たりの旅行者の一般化コスト $c_{j,z}(t)$ を式 (3) と定義する．

$$c_{j,z}(t) = \begin{cases} \sum_{i\in B_z}\tau_{j,i}(t_{\mathrm{l},z,i}(t)) + \\ \alpha\{t - t_{\mathrm{d},z}(t)\} + \beta\,(t^* - t) \quad (t < t^*) \\ \\ \sum_{i\in B_z}\tau_{j,i}(t_{\mathrm{l},z,i}(t)) + \\ \alpha\{t - t_{\mathrm{d},z}(t)\} + \gamma\,(t - t^*) \quad (t \geq t^*) \end{cases} \tag{3}$$

ここで，t は経路 z における勤務地への到着時刻，$t_{\mathrm{d},z}(t)$ は経路 z を通って時刻 t に勤務地に到着する旅行者の居住地からの出発時刻，$t_{\mathrm{l},z,i}(t)$ は経路 z を通って時刻 t に勤務地に到着する旅行者のボトルネック i からの流出時刻を示す．また，α は 1 人の旅行者における単位時間当たりの時間価値，β は希望到着時刻と比べて早く勤務地に到着する場合の単位時間あたりのコスト (早着コスト)，γ は希望到着時刻と比べて遅く勤務地に到着する場合の単位時間あたりのコスト (遅着コスト) を示す．一般的な出発時刻選択問題[1] で用いられるものと同等のモデルである．

c) 出発時刻・経路選択

旅行者の出発時刻選択は，多項ロジット選択モデルにより，過去の情報に基づき選択される Day-to-day dynamics [13] によるとする．旅行コストは，過去の経験コストの加重平均 [14) 15)] に基づく知覚旅行コスト [13] によるとする．OD ペア ζ 間で経路 z，出発時刻 t を選択する確率を $P^{\zeta}_{j,z}(t)$ とすると，式 (4) のように表される．

$$P^{\zeta}_{j,z}(t) = \frac{\exp(-\vartheta\bar{c}_{j,z}(t))}{\sum_{(z',t')}\exp(-\vartheta\bar{c}_{j,z'}(t'))} \tag{4}$$

ここで，ϑ はパラメータ，$\bar{c}_{j,z}(t)$ は j 日目，時刻 t，経路 z における知覚旅行コストである．

j 日目，時刻 t，経路 z における知覚旅行コストは式 (5)，式 (6) により定義する．

$$\bar{c}_{j,z}(t) = \frac{1}{\varsigma(\lambda)}\left(c_{j-1,z}(t) + \sum_{i=2}^{T_c}\lambda^{i-1}c_{j-i,z}(t)\right) \quad \forall z\in Z_\zeta, t\in T \tag{5}$$

$$\varsigma(\lambda) = \begin{cases} \sum_{i=1}^{T_c}\lambda^{i-1} & (0<\lambda<1) \\ 1 & (\lambda=0) \end{cases} \tag{6}$$

経路・時刻別の出発台数 $f_{j,z}(t)$ は式 (7) により算出される．

$$f_{j,z}(t) = M_\zeta \cdot P^{\zeta}_{j,z}(t) \qquad \forall z\in Z_\zeta, t\in T \tag{7}$$

ここで，M_ζ は OD ペア ζ 間の 1 日当たりの交通需要を示す．また，T は時刻の集合であり，ある範囲の連続す

る正の整数を示す．

また，Yu et al. [13] を基に，出発時刻・経路選択の Day-to-day dynamics に「限定合理性 (Bounded rationality)」を導入する．これは，参照元の選択の期待コストと，他の全選択における最小期待コストの間の差異が δ 以下の場合は翌日に出発時刻・経路選択を変更しないというものである．$j-1$ 日目に経路 z，出発時刻 t を選択した旅行者が，j 日目に経路と出発時刻を変更しない場合，期待コスト $\widehat{c}_{j,z}(t)$ は式 (8) を満たす．

$$\widehat{c}_{j,z}(t) - \widehat{c}_{j,z'}(t') \leq \delta \qquad \forall (z',t') \neq (z,t) \tag{8}$$

(2) 深層強化学習

一般に強化学習[16] は，予め制御エージェントが取れる行動の選択肢，およびシステムの状態に対する行動の望ましさを表す報酬を定義する．その上で，システムの状態を観測し，その状態と現在の学習内容に基づき行動を選択し，新たに実現したシステムの状態と行動に基づき報酬を獲得し，獲得した報酬に基づき学習内容を更新し，再度行動を選択する，という過程を繰り返し，システムの最適な状態を達成しようとするものである．深層強化学習は，強化学習に深層学習を組み合わせ，強化学習で扱う関数をニューラルネットワークで表現する手法である．本研究では，既往研究により提案された深層強化学習のアルゴリズムである Deep Deterministic Policy Gradient (DDPG) [17] を用いる．DDPG は，強化学習のアルゴリズムの 1 つである Deterministic Policy Gradient (DPG)[18] に Deep Q-Network (DQN)[19] の手法を組み合わせ，連続な行動を扱えるようにした Actor-Critic ベースの深層強化学習アルゴリズムである．また，DDPG では決定論的な方策を扱う．これは，ある状態に対して行動を一意に定めるものである．

DDPG の更なる詳細については，Lillicrap et al.[17] を参照されたい．

(3) 制御手法

本研究では，課金額の調整手法の具体的な実装形態として，分散制御型手法を提案する．これは，ボトルネック・時刻別の交通情報を状態，ボトルネック・時刻別の課金額更新を行動として設定するものである．分散制御型手法では，状態，および行動の次元が大幅に削減され，行動の探索に要する時間の短縮が期待される．

分散制御型手法では，課金額更新におけるボトルネック・時刻間の協調をとれるような手法が必要である．本研究では「報酬の協調項」，および「学習のオン/オフ」を提案する．

a) 状態，行動，報酬の定義

本節では，DDPG における状態，行動，報酬の定義を述べる．

j 日目，時刻 t，ボトルネック i における状態を式 (9)，式 (10) のように定義する．

$$\boldsymbol{s}_{j,i}(t) = \left(\frac{a_{j,i}(t)-\mu_i}{\mu_i}, \frac{w_{j,i}(t)}{\sum_{t\in T} w_{0,i}(t) / \sum_{t\in T} \varpi_{0,i,t}}, \frac{\tau_{j,i}(t)-\bar{\tau}_{j,i}}{\sum_{t\in T} w_{0,i}(t) / \sum_{t\in T} \varpi_{0,i,t}} \right) \tag{9}$$

$$\varpi_{j,i,t} = \begin{cases} 1 & (w_{j,i}(t) > 0) \\ 0 & (w_{j,i}(t) = 0) \end{cases} \tag{10}$$

ここで，$\sum_{t\in T} \varpi_{j,i,t}$ は，$w_{j,i}(t)$ $(t \in T)$ の内，正値のものの個数を表す．

提案手法を様々な交通ネットワークに適用するためには，DDPG における状態は交通状態の数値のオーダーによらないものである必要がある．よって，状態の各要素のスケーリングを行う．状態の 2 つ目の要素は，ボトルネック i におけるタイムスロット別の初期待ち時間の内，正値のものの平均で各時刻の待ち時間を割ったものある．また，状態の 3 つ目の要素は，j 日目，時刻 t，ボトルネック i における課金額と j 日目における課金額平均との差を，ボトルネック i におけるタイムスロット別の初期待ち時間の内，正値のものの平均で割ったものである．この定式化により，待ち時間が発生していない時間帯の長短によらない状態のスケーリングを行うことができる．

DDPG における行動は，Actor network の出力であり，時刻別の課金額の変動幅とする．Actor network の最終層への入力を y，最終層からの出力（行動）を b とすると，b は式 (11) のように表される．

$$b_{j,i}(t) = G \cdot \tanh y \quad (-G < b_{j,i}(t) < G) \tag{11}$$

ここで，G は行動の範囲を決定するパラメータである．なお，学習中（行動探索中）は b に探索ノイズを加えるものとする．

j 日目，時刻 t，ボトルネック i における課金額は式 (12) のように更新されるものとする．

$$\tau_{j,i}(t) \leftarrow \tau_{j,i}(t) + b_{j,i}(t) \tag{12}$$

本研究における DDPG の報酬は，下記 2 つの項の和の符号を反転させたものとする．

- 時刻別・ボトルネック別の待ち時間に基づく項
- 課金対象の全ボトルネック・全時刻での待ち時間平均に基づく項（報酬の協調項）

報酬の協調項により，時刻間・ボトルネック間で協調した総待ち時間の減少を図る．待ち時間には，スケーリング後の値を用いる．j 日目，ボトルネック i，時刻 t での報酬を $r_{j,i}(t)$ とすると，式 (13) のように表される．

$$r_{j,i}(t) = -\left\{ \frac{w_{j,i}(t)}{\sum_{t\in T} w_{0,i}(t) / \sum_{t\in T} \varpi_{0,i,t}} + \frac{1}{\#I} \sum_{i\in I} \frac{\bar{w}_{j,i}}{\sum_{t\in T} w_{0,i}(t) / \sum_{t\in T} \varpi_{0,i,t}} \right\} \tag{13}$$

報酬に協調項を入れる背景は次の通りである．荒井[20] では，学習途中で定常な政策（方策）を持たないエージェントが複数存在する系において自己の行動による状態遷移先を特定することは難しいことを指摘しており，また，この難しさに起因する問題を同時学習問題と呼んでいる．本研究でも，同様の問題が生じる．課金額調整を行うエージェントがボトルネック毎に学習を行う場合，個々のエージェントは自己の行動がどれほど自ら課金額調整を行うボトルネックの待ち時間の減少に寄与したかを特定し辛い．これは，あるボトルネックでの待ち時間増減の影響が他のボトルネックに影響し得ることに起因する．例えば，同一経路上に複数のボトルネックが存在し，下流側のボトルネックでの動的混雑課金により旅行者の出発時刻選択・経路選択が分散した場合，上流側・下流側の双方のボトルネックで待ち時間が減少し得る．また，同一の OD ペア間に複数の経路が存在する場合，他の経路のボトルネックにおける動的混雑課金に伴う経路選択確率の変動により待ち時間が増減し得る．そのため，エージェントに与える報酬をボトルネック・時刻別の待ち時間のみで設計すると，本来適切ではない課金額調整が良い行動であると判断されたり，逆に適切な課金額調整が悪い行動であると判断されたりすることで，学習の不安定化を引き起こす可能性がある．

本研究では，課金対象の全ボトルネックでの待ち時間に応じた負の報酬を各ボトルネックのエージェントに与えることで，他のボトルネックでの待ち時間を増加させる課金額調整を防ぎ，道路ネットワーク全域の総待ち時間の安定的な減少を図る．

b) 学習のオン/オフ

本研究では，課金額更新におけるボトルネック・時刻間の協調手法として「学習のオン/オフ」を提案する．これは，式 (14) のようにボトルネックでの待ち時間の移動平均が閾値を下回った時刻においては，深層強化学習を行わず，行動（ボトルネック・時刻別の課金額の増減幅）を 0 に固定するというものである．

$$\frac{1}{2n+1} \sum_{t'=t-n}^{t+n} w_{j,i}(t') < \mathrm{d}w \tag{14}$$

ここで，$\mathrm{d}w$ は微小の定数である．

学習のオン/オフを行う背景は次の通りである。混雑課金は，渋滞待ち時間による負の外部性を旅行者の経済計算に反映させることで交通渋滞を緩和する施策である．即ち，渋滞待ち時間によるコストを混雑課金により置き換えるものである．そのため，分散制御では待ち時間が発生しているボトルネック・時刻での課金額更新を重視する必要がある．本手法では，待ち時間がほとんど発生していないボトルネック・時刻での状態遷移を学習に用いないことで，制御エージェントの学習対象を基本的に待ち時間が発生しているボトルネック・時刻での課金額調整に絞り，探索空間を削減することで学習の効率化を図る．また，待ち時間がほとんど発生していないボトルネック・時刻で課金額の増減を行わない設定により，当該ボトルネック・時刻での課金額の過剰な増加を防ぐ．一方で，学習オフの基準を「ボトルネック・時刻別の待ち時間が微小の閾値未満の場合」にすると，ボトルネック別・時刻別の待ち時間が微小の閾値に達すると学習が即座にオンになる．また，学習オフの基準を「ボトルネック・時刻別の待ち時間の時刻間移動平均が 0 の場合」にすると，少しでも待ち時間が発生した時刻付近では学習がオフにならない．これらの場合には，学習のオン/オフが時刻によって細かく切り替わってしまう．その結果として，学習の不安定化を招く恐れがある．そこで，本研究では学習オフの基準を「ボトルネック・時刻別の待ち時間の時刻間移動平均が閾値未満の場合」とすることで，学習のオン/オフが時刻によって細かく切り替わらないようにし，学習の安定化を図る．

c) アルゴリズムのまとめ

以下に，アルゴリズムの流れを示す．フローチャートは**図–1** の通りである．

Step 0

日数を $j := 0$ と設定し，旅行者の出発率の初期状態を設定する．

Step 1 交通流状態の初期化段階

Step 1.1

日数を $j := j + 1$ と更新する．

Step 1.2

交通流における $w_{j,i}(t)$ および $a_{j,i}(t)$ を観測する．

Step 1.3

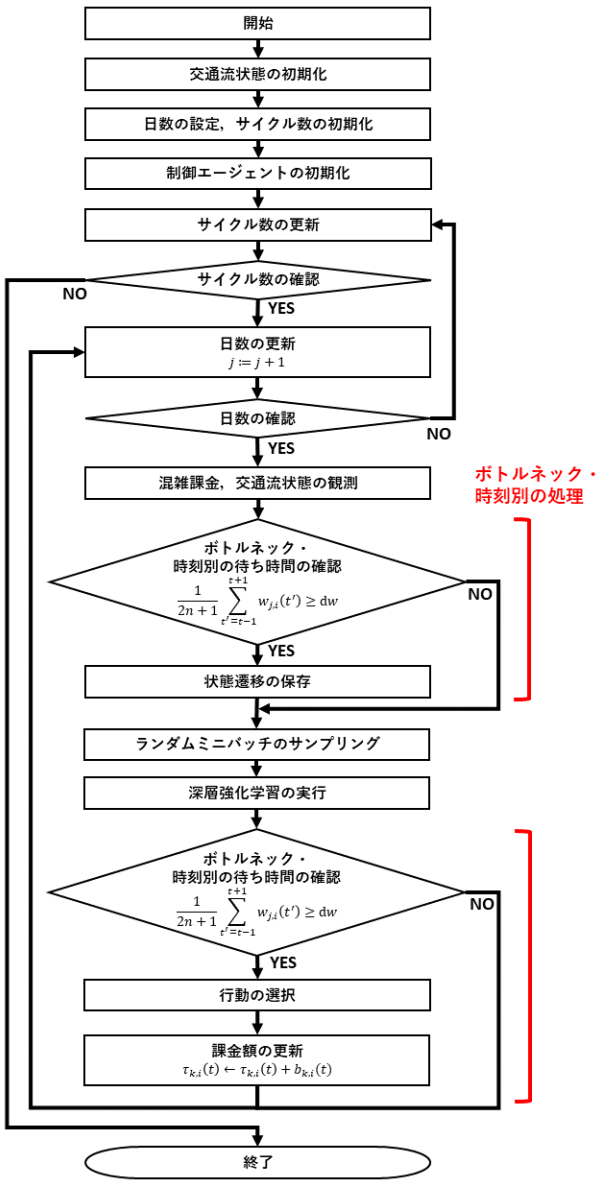
Day-to-day dynamics の収束確認を行う．e の設定値に対して，収束条件を満たす場合は Step 1.4 に進む．その他の場合，Step 1.1 に戻る．

Step 1.4

現在の状態を利用者均衡状態 (user equilibrium state) とみなす．日数を $j := 0$ にリセットし，サイクル数を 0 に設定する．

Step 2

制御エージェントを初期化する．



図–1 アルゴリズムのフローチャート

Step 3 Trial-and-error 段階

Step 3.1

サイクル数に 1 を足す．サイクル数が設定値以下なら Step 3.2 に進む．サイクル数が設定値より大きい場合はアルゴリズムを終了する．

Step 3.2

日数を $j := j+1$ と更新する．日数が設定値以下なら Step 3.3 に進む．日数が設定値より大きい場合は，$j := 0$ とリセットし，交通流状態を Step 1.4 の時点と同じにして Step 3.1 に戻る．

Step 3.3

課金額を $\tau_{j,i}(t) = \tau_{j-1,i}(t)$ と決定し，課金を実施する．

Step 3.4

交通流における $w_{j,i}(t)$ および $a_{j,i}(t)$ を観測する．

Step 3.5 学習のオン・オフ（行動の選択）

Step 3.5.1

ボトルネック・時刻別に待ち時間の移動平均を確認する．移動平均が閾値以上の場合は Step 3.5.2 に進む．移動平均が閾値未満の場合は Step 3.6 に進む．

Step 3.5.2

報酬 $r_{j,i}$ を算出し，状態遷移 $(s_{j,i}, b_{j,i}, r_{j,i}, s_{j+1,i})$ を保存する．

Step 3.6

保存された状態遷移からランダムミニバッチをサンプリングし，深層強化学習を行う．

Step 3.7 学習のオン・オフ（課金額の更新）

Step 3.7.1

ボトルネック・時刻別に待ち時間の移動平均を確認する．移動平均が閾値以上の場合は Step 3.7.2 に進む．移動平均が閾値未満の場合は Step 3.8 に進む．

Step 3.7.2

現在の方策に基づき，行動を選択する．

Step 3.7.3

行動 $b_{j,i}$ を実行する．即ち，課金額を $\tau_{j,i}(t) \leftarrow \tau_{j,i}(t) + b_{j,i}(t)$ と更新する．

Step 3.8

Step 3.2 に戻る．

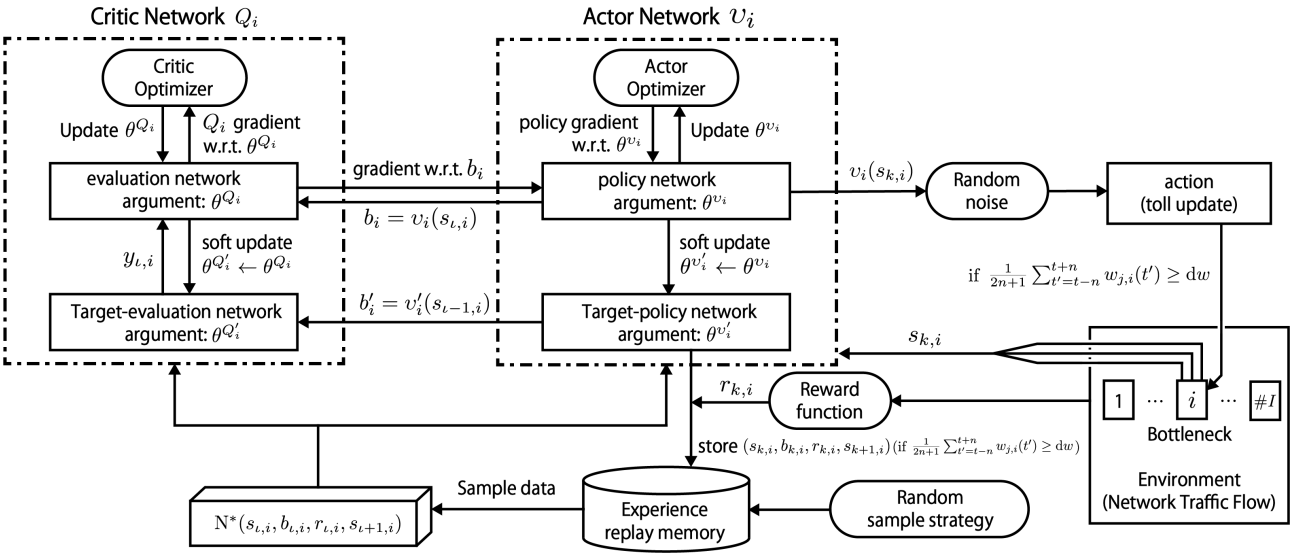
Guo et al. [21] を参考に，DP-DDPG のアーキテクチャを図–2 として示す．

3. 実験結果と考察

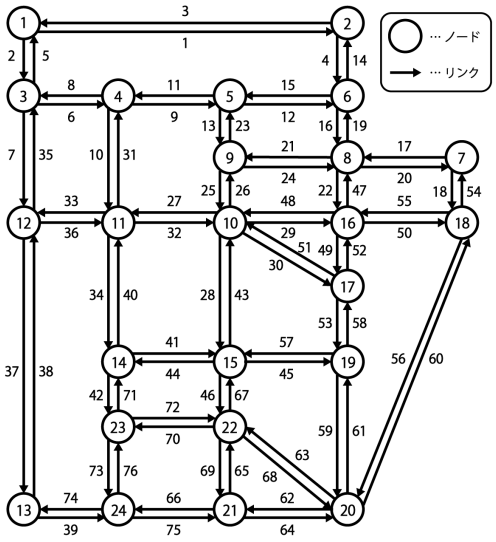
(1) 実験の目的

開発した制御手法の性質をシミュレーション実験により確認する．具体的には以下の内容を確認する．

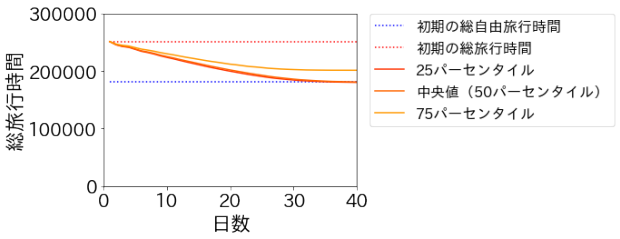
- Sioux Falls Network において，最適なエージェントパラメータを設定した上で DP-DDPG による深層強化学習を行い，学習済みエージェントの待ち時間減少性能を分析することで，複数の OD ペア，および複数の経路が存在する道路ネットワークへの対応可能性を検証する．なお，最適なエージェントパラメータは事前に総当たり的な試行により実験的に探索する．
- 単純なモデルである単一 OD・3 経路・経路当たり 1 ボトルネックの交通モデル（並列ボトルネックモデル）において，中央制御型 DDPG，DP-DDPG，完全分散制御型 DDPG の 3 手法による深層強化学習



図–2 DP-DDPG のアーキテクチャ



図–3 Sioux Falls Network の概要



図–4 Sioux Falls Network における総旅行時間の推移

を行い，同様のモデルにおいて Q 学習によるシミュレーションを行った佐藤ら [9] と学習済みエージェントの待ち時間減少性能を比較することで，DP-DDPG の優位性を検証する．

(2) Sioux Falls Network でのシミュレーション

本節では，**図–3** のような Sioux Falls Network[22] を用いてシミュレーションを行った．本研究の設定において，渋滞が発生し，かつ課金対象のボトルネックを有するリンクは 29，43，53，58 であった．また，渋滞が発生し，かつ課金対象でないボトルネックを有するリンクは 49，52，61 であった．旅行者の勤務地への希望到着時刻 t^* は全旅行者について等しいとした．Actor Network・Critic Network の学習率，および G の値について幾つかのパターンで学習を行い，下記の設定を最適と決定した．

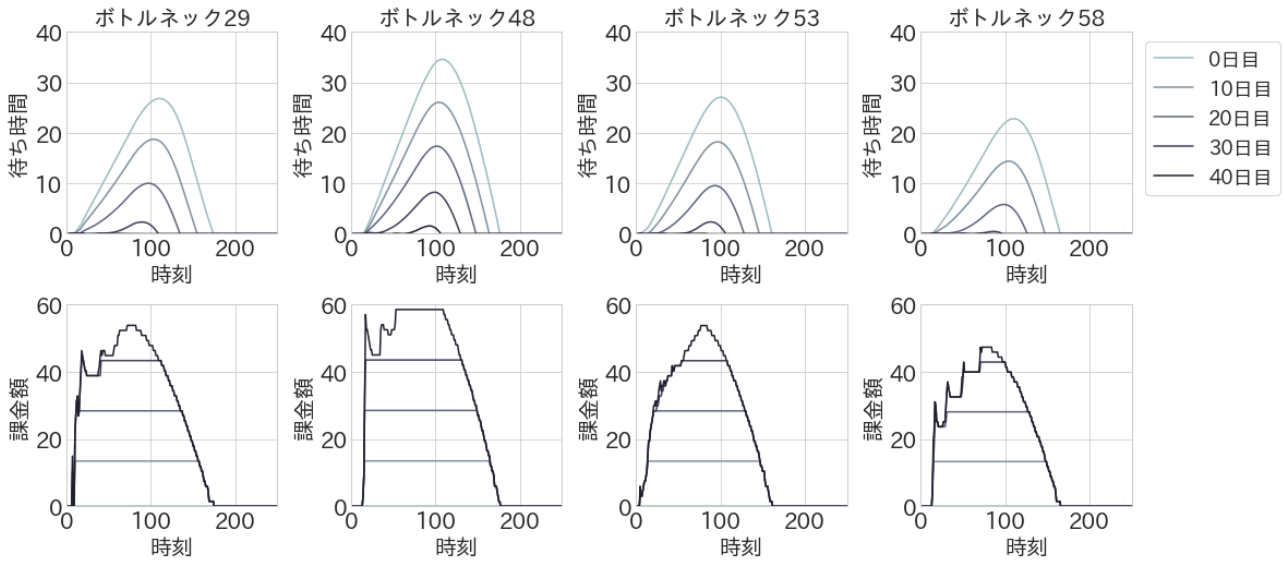
- Actor Network の学習率 $= 10^{-3}$
- Critic Network の学習率 $= 10^{-2}$
- $G = 1.5$

交通モデルのパラメータは，$\alpha = 1.0$，$\beta = 0.45$，$\gamma = 1.2$，$\vartheta = 0.015$，$\lambda = 0$，$\delta = 1$，$t \in \mathbb{N}$，$1 \leq t \leq 250$，$t^* = 75$ と設定した．

学習の際は 40 日を 1 サイクルとし，40 日経過後に「$\tau_{j,i}(t) = 0 \ \forall(j,i,t)$，かつ Day-to-day dynamics が収束している状態」に戻って学習を行うものとした (学習結果，即ち Actor Network・Critic Network におけるパラメータの更新結果は引き継ぐ)．そして，15 サイクルを 1 セットとし，10 セット学習を行った．

Sioux Falls Network における総旅行時間の 10 セット分の推移から第一四分位数，中央値，第三四分位数をとった図を**図–4** として示す．**図–4** より，約 40 日で総旅行時間が減少していることが読み取れる．

ボトルネック別の待ち時間分布・課金額分布の推移



図–5 Sioux Falls Network の各ボトルネックにおける待ち時間・課金額分布の推移

を図–5 として示す．図–5 より，4 箇所のボトルネック全てで，待ち時間分布に対応して課金額が増加し，待ち時間が減少していることが読み取れる．

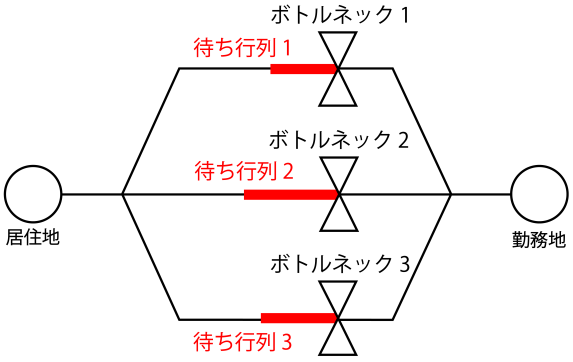
(3) 並列ボトルネックモデルでのシミュレーション

本節では，図–6 のような「並列ボトルネックモデル」を用い，下記の 4 手法におけるシミュレーション結果の比較を行う．並列ボトルネックモデルは，佐藤ら[9] における「複数ボトルネックモデル」と同等のモデルである．

- 中央制御型 DDPG
- 提案手法 (DP-DDPG)
- 完全分散制御型 DDPG
- Q 学習

各手法における状態，行動，報酬の概要は次の通りである．中央制御型 DDPG では，課金対象の全ボトルネックでの交通情報を状態とし，課金対象の全ボトルネックでの課金額更新を行動とする．報酬は，課金対象の全ボトルネックでの待ち時間に基づき定義する．なお，課金額分布は区分線形関数で表現するものとする．他の 3 手法では，課金対象の各ボトルネックでの交通情報を状態とし，課金対象の各ボトルネックでの課金額更新を行動とする．DP-DDPG（提案手法），および完全分散制御型 DDPG では，報酬は課金対象の各ボトルネックでの待ち時間に基づき定義する．Q 学習では，報酬は課金対象の各ボトルネックでの流入率に基づき定義する．更に，DP-DDPG，および Q 学習では，報酬に協調項を入れるものとする．

なお，中央制御型 DDPG では課金時間帯全体を等分して各時間帯での交通状態の平均をとり，深層強化学習の状態とする．また，課金額分布を区分線形関数で



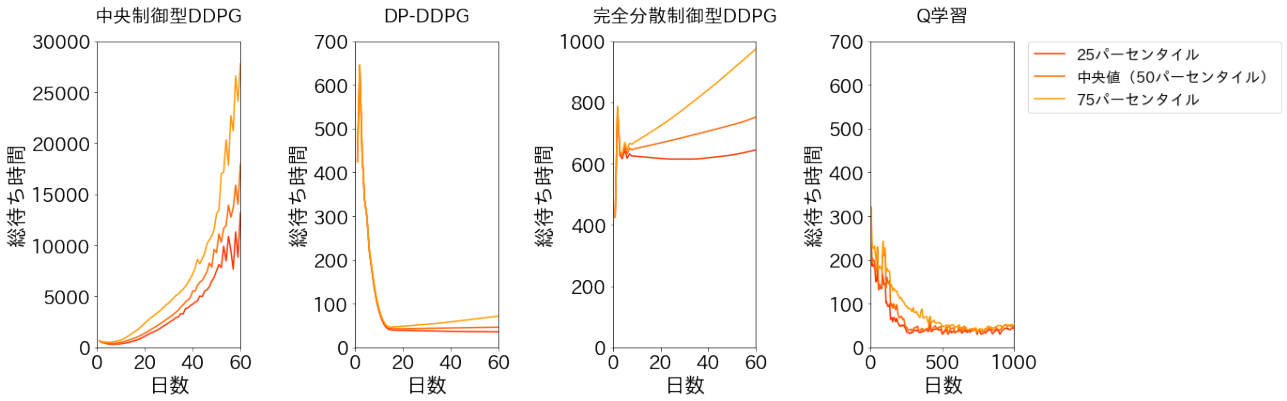
図–6 並列ボトルネックモデルの概要[9]

表現する．Q 学習については，佐藤ら[9] の実験結果を用いる．なお，佐藤ら[9] では Day-to-day dynamics として Replicator dynamics [23] を採用している点に留意されたい．中央制御型 DDPG，DP-DDPG，完全分散制御型 DDPG については，下記の設定で新たにシミュレーションを行った．

旅行者の勤務地への希望到着時刻 t^* は全旅行者について等しいとした．Actor Network・Critic Network の学習率，および G の値は下記の通りに設定した．

- Actor Network の学習率 $= 10^{-5}$
- Critic Network の学習率 $= 10^{-4}$
- $G = 0.5$

交通モデルのパラメータは，$\alpha = 1.0$，$\beta = 0.45$，$\gamma = 1.2$，$\vartheta = 0.05$，$\lambda = 0$，$\delta = 0.5$，$t \in \mathbb{N}$，$1 \leq t \leq 80$，$t^* = 30$ と設定した．学習の際は 60 日を 1 サイクルとし，60 日経過後に「$\tau_{j,i}(t) = 0 \ \forall (j,i,t)$，かつ Day-to-day dynamics が収束している状態」に戻って学習を行



図–7 各手法における総待ち時間の推移

うものとした (学習結果，即ち Actor Network・Critic Network におけるパラメータの更新結果は引き継ぐ) . そして，20 サイクルを 1 セットとし，20 セット学習を行った．

学習済みエージェントを学習時と同じ環境に適用した場合の，各手法における総待ち時間の推移を**図–7** として示す．**図–7** より，中央制御型 DDPG，もしくは完全分散制御型 DDPG を用いた場合は学習が適切に進行せず，学習済みエージェントが待ち時間を減少させないことが読み取れる．中央制御型 DDPG では，課金時間帯全体を n 等分して各時間帯での交通状態の平均をとり，強化学習の状態としている．また，課金額分布を区分線形関数で表現している．これにより，時刻別の交通状態の把握ができず，また最適な課金額分布とのずれが生じた結果，出発時刻・経路選択の Day-to-day dynamics が不安定化したと考えられる．

完全分散制御型 DDPG では，報酬の協調項，および学習のオン/オフを導入しなかったためにボトルネック・時刻間の協調が行われず，学習が適切に進行しなかったと考えられる．

DP-DDPG，および Q 学習については，課金額調整の基準や出発時刻・経路選択の Day-to-day dynamics の内容が異なるため，最適な課金額に近付くまでの日数を一概に比較することはできない．佐藤らの Q 学習では，課金額の更新は Day-to-day dynamics が収束した後に行っているため，最適な課金額に近付くまでの日数が長くなっている．一方で，佐藤らの Q 学習を用いた場合は待ち時間総和が 0 に近付いた後で振動しているが，DP-DDPG を用いた場合は待ち時間総和が 0 に近付いた後に振動しておらず，安定的に待ち時間総和を減少させていると言える．これは，佐藤らの Q 学習では課金額調整が離散値である一方，DP-DDPG では連続値であることに起因すると考えられる．

また，DP-DDPG，および Q 学習を用いた場合のシミュレーション結果より，報酬の協調項は適切な学習のために有用であると言える．

4. 結論と今後の課題

本研究では，深層強化学習を用いた，一般の大規模道路ネットワークにおいて観測可能なデータに基づき渋滞を解消する分散協調制御型の動的混雑課金額決定手法を提案した．本手法の特徴の 1 つとして，時刻・箇所別の交通状態観測，および課金額更新を行う分散協調制御の導入が挙げられる．これにより，大規模な道路ネットワークにおいて、高速かつ計算効率の良い学習が可能となる．また，行動を連続値として扱う DDPG を深層強化学習アルゴリズムとして用いることで，細かな課金額調整が可能となる．

Sioux Falls Network を用いた交通シミュレーションにより，提案手法が道路ネットワーク全域の総待ち時間を減少させることを確認した．また，中央制御型 DDPG，完全分散制御型 DDPG，Q 学習との実験結果の比較により，報酬の協調項，および学習のオン/オフを導入した DDPG ベースの分散協調制御の優位性を確認した．

今後の課題として，多数のボトルネックで動的混雑課金を行う場合のシミュレーション，報酬設計の更なる検討，学習済みエージェントを用いた転移学習の検討が考えられる．

謝辞: 本研究は JSPS 科学研究費補助金 20H02267 および国土交通省新道路技術会議の研究課題「学習型モニタリング・交通流動予測に基づく観光渋滞マネジメントについての研究開発」の助成を受けた．ここに謝意を表します．

参考文献

1) Vickrey, W.: Congestion theory and transport investment, *The American Economic Review*, Vol.59, No.2, pp.251–260, 1969.
2) Arnott, R., de Palma, A., and Lindsey, R.: Economics of a bottleneck, *Journal of Urban Economics*, Vol.27, pp.111–130, 1990.
3) 赤松隆: 交通流の予測・誘導・制御と動的なネットワーク配分理論, 土木計画学研究・論文集, Vol.13, pp.23–48, 1996.
4) 桑原雅夫, 赤松隆: 動的ネットワーク解析, 土木学会論文集, Vol.2000, No.653, pp.3–16, 2000.
5) Zhu, F. and Ukkusuri, S. V.: A reinforcement learning approach for distance-based dynamic tolling in the stochastic network environment, *Journal of Advanced Transportation*, Vol.49, pp.247–266, 2015.
6) Qiu, W., Chen, H., and An, B.: Dynamic electronic toll collection via multi-agent deep reinforcement learning with edge-based graph convolutional networks., *IJCAI*, pp. 4568–4574, 2019.
7) Yang, H., Meng, Q., and Lee, D.-H.: Trial-and-error implementation of marginal-cost pricing on networks in the absence of demand functions, *Transportation Research Part B: Methodological*, Vol.38, pp.477–493, 2004.
8) Seo, T. and Yin, Y.: Optimal pricing for departure time choice problems with unknown preference and demand: Trial-and-error approach, 2019.
9) 佐藤公洋, 瀬尾亨, 布施孝志: 強化学習を用いたデータ駆動型の動的混雑課金の最適化手法, 土木学会論文集 D3（土木計画学）, Vol.76, No.5, pp.I˙1273–I˙1285, 2021.
10) Pandey, V., Wang, E., and Boyles, S. D.: Deep reinforcement learning algorithm for dynamic pricing of express lanes with multiple access locations, *Transportation Research Part C: Emerging Technologies*, Vol.119, pp.102715, 2020.
11) 井料隆雅: 不安定な動的利用者均衡に対する安定化制御, 土木計画学研究発表会・講演集, 55, 2017.
12) Iryo, T., Smith, M. J., and Watling, D.: Stabilisation strategy for unstable transport systems under general evolutionary dynamics, *Transportation Research Part B: Methodological*, Vol.132, pp.136–151, 2020, 23rd International Symposium on Transportation and Traffic Theory (ISTTT 23).
13) Yu, Y., Han, K., and Ochieng, W.: Day-to-day dynamic traffic assignment with imperfect information, bounded rationality and information sharing, *Transportation Research Part C: Emerging Technologies*, Vol.114, pp.59–83, 2020.
14) Cascetta, E.: A stochastic process approach to the analysis of temporal dynamics in transportation networks, *Transportation Research Part B: Methodological*, Vol.23, No.1, pp.1–17, 1989.
15) Ouyang, Y.: Pavement resurfacing planning for highway networks: Parametric policy iteration approach, *Journal of Infrastructure Systems*, Vol.13, No.1, pp.65–71, 2007.
16) Sutton, R. and Barto, A.: *Reinforcement Learning: An Introduction*, The MIT Press, second edition, 2018.
17) Lillicrap, T. P., Hunt, J. J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., Silver, D., and Wierstra, D.: Continuous control with deep reinforcement learning, *arXiv preprint arXiv:1509.02971*, 2015.
18) Silver, D., Lever, G., Heess, N., Degris, T., Wierstra, D., and Riedmiller, M.: Deterministic policy gradient algorithms, *Proceedings of the 31st International Conference on International Conference on Machine Learning - Volume 32*, ICML'14, pp. I–387–I–395, JMLR.org, 2014.
19) Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., and Riedmiller, M. A.: Playing atari with deep reinforcement learning, *arXiv preprint arXiv:1312.5602*, 2013.
20) 荒井幸代: マルチエージェント強化学習 : 実用化に向けての課題・理論・諸技術との融合 (<特集>「マルチエージェント技術における新しい可能性」), 人工知能, Vol.16, No.4, pp.476–481, 2001.
21) Guo, S., Zhang, X., Zheng, Y., and Du, Y.: An autonomous path planning model for unmanned ships based on deep reinforcement learning, *Sensors*, Vol.20, No.2, pp.426, 2020.
22) Morlok, E., Schofer, J., Pierskalla, W., Marsten, R., Agarwal, S., Stoner, J., Edwards, J., LeBlanc, L., and Spacek, D.: *Development and Application of a Highway Network Design Model, Volumes 1 and 2*, Final Report: FHWA Contract Number DOT-PH-11, Northwestern University, Evanston, 1973.
23) Schuster, P. and Sigmund, K.: Replicator dynamics, *Journal of Theoretical Biology*, Vol.100, No.3, pp.533–538, 1983.

(2022. 3. 5 受付)

DYNAMIC NETWORK CONGESTION PRICING BASED ON DEEP REINFORCEMENT LEARNING

Kimihiro SATO, Toru SEO and Takashi FUSE

Dynamic congestion pricing has been proposed as one of useful schemes to eliminate traffic congestion, where toll is set according to changes in traffic demand throughout the day. However, an optimal dynamic congestion pricing is difficult because real road networks are large and complicated, and there is asymmetric information between pricing entities and road users. This paper proposes a dynamic congestion pricing method based on deep reinforcement learning (DRL), which eliminates traffic congestion based on observable data in general large-scale road networks. Specifically, DRL is implemented by a spatial-temporally distributed manner, and cooperation among DRL agents are established by novel techniques we call spatially shared reward and temporally switching learning. It enables a fast and computationally efficient learning. The numerical experiments using Sioux Falls Network showed that the proposed method works well.