

確率的最適化による深層学習とマルチエレメント GA を用いた 道路交通信号パラメータの最適化

中村新之介^{†1} 植村匠^{†1} 上瀧剛^{†2} 内村圭一^{†2}

概要: 近年、交通渋滞を緩和する手法として交通信号制御による手法が注目されている。筆者らはマルチエレメント GA と交通流シミュレータを用いて交通信号パラメータを最適化し、渋滞を緩和する手法が提案したが、最適化処理に長い時間を要するため、実環境下での運用は現実的ではない問題を有していた。本研究では、交通流シミュレータの入出力関係を学習させた学習機械をシミュレータと置き換える手法を提案し、最適化処理の時間短縮を目指す。

キーワード: 交通信号制御, 深層学習, 確率的最適化, 交通流シミュレータ, マルチエレメント GA

Optimization of road traffic signal parameters using a multi-element GA and deep learning by stochastic optimization

Shinnosuke NAKAMURA^{†1} Takumi UEMURA^{†1} Gou KOUTAKI^{†2}
Keiichi UCHIMURA^{†2}

Abstract: In recent years, techniques have been attracting attention due to traffic signal control as a method to alleviate traffic congestion. Among them, Mitigation method congestion by optimizing the traffic signal parameters using multi-element GA and traffic simulator was proposed. But it takes a long time to the optimization process. Therefore, It is difficult to operating in a real environment. In this study, in order to shorten the optimization process time, we propose a method to replace the learning machine to the traffic flow simulator.

Keywords: Traffic Signal Control, Deep Learning, Stochastic Optimization, Traffic Simulator, Multi-Element GA

1. はじめに

近年、交通渋滞による経済的損失や環境汚染が問題となっている。経済的損失に関して国土交通省によると、日本では年間 12 兆円もの経済的損失が発生している[1]。また、環境問題に関しては、交通渋滞による損失時間は一人あたり年間約 30 時間に上り、必要以上に排気ガスが発生するため環境への影響が懸念されている。

この交通渋滞を解決する手法は大きく 3 つに分類される。1 つ目は道路拡張によって交通容量を増加させる手法である。しかしながら、工事のためのコストや工事による交通規制によって渋滞が生じるデメリットが発生する。2 つ目は電子掲示板やカーナビゲーションなどによって車両を動的に迂回路へ案内し、交通網への車両流入量を減少させる手法である。しかしながら、全ての車両に対応するには大きな経済的コストが発生するだけでなく、迂回路へ案内することによる更なる渋滞が発生する。3 つ目は交通量に応じた交通信号制御で、交通網への車両流入量を減少させる手法である。この手法は交通信号における制御パラメータの変更することで実現可能で、コストを抑えられるだけでなく、制御範囲内の全車両に対応可能な利点を有する

[2,3,4,5].

筆者らはこの交通信号制御による渋滞緩和手法の一つに実環境を再現した交通流シミュレータと、マルチエレメント遺伝的アルゴリズム(ME-GA)を用いた交通信号パラメータの最適化手法を提案している[5]。この手法によって最適化したパラメータを用いることで、実測した交通信号パラメータを用いた場合と比べると渋滞緩和が確認された。しかしながら、最適化のために繰り返し行う交通流シミュレート時間がボトルネックとなり、最適化までの処理時間が非常に長く実環境下での運用は現実的ではない課題を有していた。

本研究では、この問題を解決するために交通流シミュレータを、シミュレータの入出力関係を機械学習させた学習機械へ置き換える手法を提案し、従来手法で課題となっていた最適化処理時間の短縮を図る。

2. 関連手法

交通信号制御による渋滞緩和手法は幾つかあるが、その中でも近年実際に運用されている手法が GreenWave である[6,7]。GreenWave 法とは一方向の区間における信号機を車両郡が停止することが無いよう連続的に制御する手法であるが、GreenWave 区間の出入り口で渋滞が発生するなどの問題を有している。また、GreenWave 法を改良した手法に徐らの GreenSwirl 法が有り、これは GreenWave を渦巻き状

^{†1} 崇城大学
Sojo University.

^{†2} 熊本大学
Kumamoto University

に発生させた循環路と循環路を最適に利用可能とするための経路案内アルゴリズムを用いた手法で、GreenWave 法よりも良い渋滞緩和性能が確認されている[8]。しかしながら、循環路の設計や信号制御パターンは人手によって設計されており、広範囲における応用は困難である課題を有している。

広範囲な交通網の渋滞緩和に有効とされる手法として、西原らは実環境(図1に示すような熊本市内大江・渡鹿地区)を再現した交通流シミュレータと ME-GA を用いた交通信号パラメータの最適化手法を提案し、実環境での渋滞評価値よりも大幅な渋滞緩和を確認している[5]。この手法は以下の5つのステップにより構成されており、図2に処理フローを示す。

- Step 1 実世界の交通網と交通信号パラメータを交通流シミュレータへ設定する。
- Step 2 交通流シミュレーションにより出力された渋滞評価値をME-GAへの初期パラメータとして入力し、初期個体を生成する。
- Step 3 入力された渋滞評価値を元に、ME-GAの1世代分の更新として交通信号パラメータを設定個体数だけ出力する。
- Step 4 交通信号パラメータ毎の渋滞評価値を交通流シミュレーションにより取得する。
- Step 5 設定した世代数へ到達していなければ各渋滞評価値をME-GAへ入力し、Step 3へ移る。

Step1における交通信号パラメータは、Cycle, Sprit, Offsetの3つのパラメータで構成されており、Cycleとは信号機が「青, 黄, 赤」と1周期の変化に要する時間を表す。Offsetとは基準信号機に隣接する信号機の青信号が開始する時間をずらすための時間を表す。SpritはCycleにおける通行許可時間の割合を表す。

Step3における渋滞評価値に関しては、渋滞評価に使用している交通流シミュレータ Aimsun6.1 から出力される渋滞評価値の中から、Waitout, Inside, Total Travel Distance(TTD), Delay Timeの4つのパラメータを選択している。Waitoutは最適化範囲外の交差点において渋滞等が原因でその範囲内へ入ることが出来ない車両の台数を表す。Insideは最適化範囲内に存在する車両台数を表す。TTDはシミュレーション中に存在した全ての車両の走行距離の合計値を表す。Delay Timeは理想的な走行時間と実際の走行時間の差を1台あたりの平均時間で表す。

しかしながら、交通流シミュレーションを繰り返し行いながら最適解を得る手法であるため、シミュレーション時間がボトルネックとなり、交通信号パラメータの最適化に

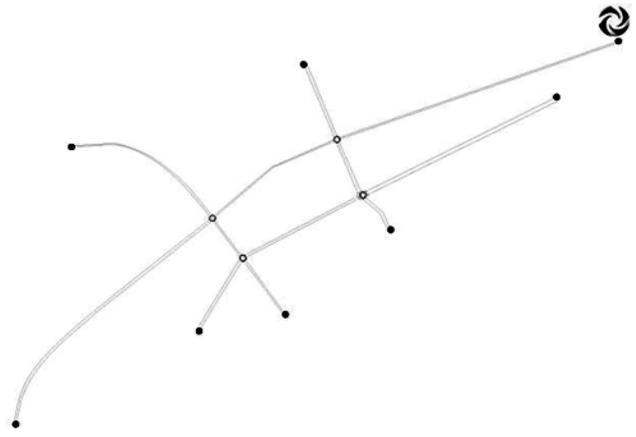


図1 熊本市 大江・渡鹿地域を再現した交通網

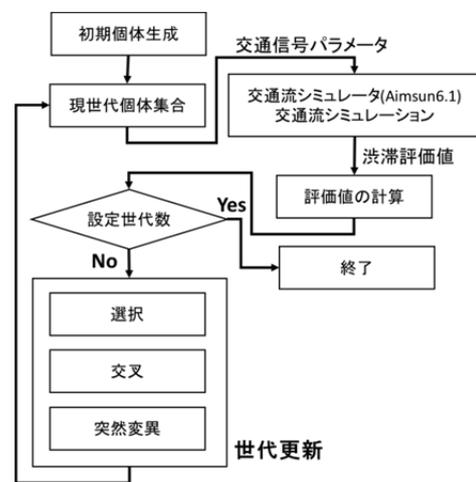


図2 従来手法における処理フロー

約19時間もの処理時間を要し、実環境下での運用は現実的ではないという課題を有している。

3. 提案手法

著者らは広範囲且つ全ての車両に対して対応可能である西原らの手法に着目し、課題とされているシミュレーション時間を短縮するために、図3に示すような交通流シミュレータの入出関係を学習させた学習機械をそのシミュレータと置き換えることによる処理時間短縮手法を提案している[9]。

文献[9]では、深層学習によるNeural Network(NN)を用いて学習機械を構築し、交通流シミュレータと置き換えた結果、処理時間の短縮と西原らと同等な渋滞緩和性能を確認した。しかしながら、交通流シミュレータの入出力関係との学習誤差を減少させることで更なる渋滞緩和が可能であると考えられた。

深層学習における誤差の減少に有効な手法として、近年注目されているのは、ニューロン重みの最適化手法である確率的最適化で、この手法を新たに取り入れることで学習誤差の減少を図る。

また、ME-GA における個体内遺伝子である Offset に対して、基準信号から遠い程 Offset が長くなる制約条件を新たに追加する。基準信号から遠い程 Offset は長くなることは一般的であるが、従来手法においてはこの点を考慮していないため、致死解が多く発生していると考えられる。文献[5]においてはシミュレータによる正確な渋滞評価値を出力していたため、この致死解は世代更新によって淘汰されている。しかし、文献[9]の手法においては、シミュレータを完全には再現していないため、誤って致死解に対して良い渋滞評価値を出力していると考えられる。よって、ME-GA の個体である Offset に対して上記の制約条件を新たに追加することで、致死解の発生を防ぎ、更なる渋滞緩和を目指す。

本研究では確率的最適化による深層学習を用いることでシミュレータとの誤差の減少と ME-GA に対して新たな制約を追加することで更なる渋滞緩和を目指す。

3.1 遺伝的アルゴリズム

3.1.1 マルチエレメント GA の概要[5]

一般的な遺伝的アルゴリズム(以下 GA)の大きな特徴として、問題適応範囲の多様性が挙げられる。しかし、ランダム性が強い事により、制約条件の多い複雑な最適化問題においては、その問題が与えられる環境に適応できない個体(致死解)が発生する。この問題を解決するために西原らは、どのような複雑な最適化問題においてもより最適解に近い解を導出する「マルチエレメント GA(ME-GA)」を提案した。この ME-GA は一般的な GA に対して以下の点が異なる。

・個体表現を多次元配列化

一般的な GA では 1 個の要素に対してのみ遺伝子操作を行うのが基本であるが、多くの要素を持った複雑な問題に対応するべく個体表現を多次元配列が図られた。

・遺伝子情報に最適化問題毎で制約条件

多次元配列化に伴い、個体内の各遺伝子が干渉し合うことで致死解が多く発生する問題が考えられる。よって個体内の遺伝子情報に対して「ある遺伝子が x なら、こちらの遺伝子は y でなくてはならない」といった制約を加えることで致死解の発生を防ぐ。

3.1.2 新たな制約条件

ME-GA の個体内遺伝子の一つである Offset への制約として、従来の処理フローに対して図 3 の赤色箇所を示すような修正処理を追加する。修正処理では、初期生成、従交叉、突然変異により Offset に関する致死解が発生した際に処理を行うが、その際に致死解を生む交差点に一番近い交差点の Offset 値を m として、 $m \sim 180$ 秒の乱数を代入することで修正する。乱数とする理由は多様性を維持するためである。

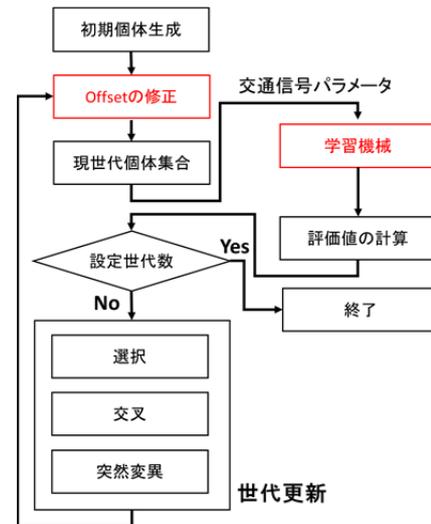


図 3 提案手法における処理フロー

3.2 学習機械

3.2.1 深層学習の学習手法

本研究で用いる学習手法には、近年において機械学習がよく用いられている画像認識などのコンペティションや、研究分野において高い性能が示されている深層学習による NN を用いた。文献[10]によると、従来の誤差逆伝搬法を用いた NN において、複雑な問題に対応するために多層 NN を構築したとしても、誤差伝搬の発散や過学習により、上手く学習を行うことが出来ないとされていた。この問題に対して Hinton らは、事前学習と呼ばれる多層の NN を 2 つの層に分離させ、それぞれの層で教師なし学習後、ネットワーク全体で教師あり又は教師なし学習を行う Deep Belief Network を提案し、多層 NN の問題点を克服した[11]。

深層学習の学習手法は Autoencoder (以下 AE)[12]、Restricted Boltzmann Machine (RBM)[13,14]、Convolution Neural Network[15,16,17] および Recurrent Neural Network[18,19]の 4 つに分類されている。

本研究では、一般的な NN と同様な構造をしており最も汎用性が高い期待される AE を選択する。またその AE の中でも最も高い性能が示されている Stack Denoising Autoencoder (以下 SDA)と呼ばれる学習手法を用いる。SDA を用いることで、学習後は入力を再現できるだけでなく、そのノイズを削除することができるかとされている[12]。

3.2.2 ニューロン重みの最適化手法

ニューロン重みの最適化手法に、確率的最適化を用いる。

これは、深層学習における確率的勾配降下法に代表されるような、目的関数が最大もしくは最小となるような目的関数内のパラメータを確率論的に最適化する手法で、近年における多くの機械学習の研究において中心的な手法とされている。またその中でも Adam と呼ばれる手法は、近年の機械学習において一般的に用いられている AdaGrad と RMSProp の利点を組み合わせた手法で、確率的最適化手法

の中でも少ないメモリで高い効率での最適化性能が確認されている[20].

本研究ではこの Adam による深層学習を用いることで学習誤差の減少を図る

3.2.3 Neural Network の構成

深層学習による特徴抽出後、教師データに対する識別を行うために誤差逆伝搬法を用いた教師あり学習を行う。本研究では山田らの手法に習い、教師データは[0, 1]の間で実数として正規化し、活性化関数としては次式に示すシグモイド関数を用いた[21].

$$\text{sigmoid}(z) = \frac{1}{1+e^z} \quad (1)$$

出力層における誤差関数は平均二乗誤差を用いた.

$$L = \frac{1}{2} \sum_k (T_k - O_k)^2 \quad (2)$$

k はニューロン数, T_k を教師データ, ニューロンからの出力値を O_k とする.

次に Adam によるパラメータ最適化手順を表 1 に示す. θ が求めたいパラメータで f が目的関数, g は目的関数から得られた勾配, t は学習回数を示す. また, Adam ではスーパーパラメータが決められており, 表 1 の上部に示している.

4. 実験

提案手法の有効性を示すために 2 つの実験を行う. 実験 1 では提案手法による誤差減少を確認するために, 誤差を文献[9]の手法と比較する. 実験 2 では提案手法を用いて交通信号パラメータの最適化を行い, 交通流シミュレータを用いての渋滞緩和性能の評価と, 最適化処理時間を文献[5]および文献[9]とで比較する.

4.1 実験環境

各手法における計算機の環境を表 2 に示す.

4.1.1 交通流シミュレータの設定

交通流シミュレータにおける交通網は, 文献[5]と同様に図 1 に示す範囲を Aimsun6 内に設定した. また, Aimsun6 内における交通量や各車両エージェントの設定も文献[5]に合わせた.

4.1.2 学習機械の設定

本実験で用いる学習機械の構成を表 3 に示す. 学習用データは入力データと教師データで構成されており, 入力データは ME-GA にて設定された範囲内でランダムに変化させた 10,000 データとし, この入力データをシミュレータへ入力して得られた渋滞評価値を教師データとしている. また, 汎化誤差計測用のデータとして学習用データとは別に 2,000 データを用意した. 次に対象となる交通網を文献[5]と同様の 4 交差点とするため, 学習機械に対する入力データは, 3 種類ある信号機パラメータが 4 交差点分の計 12 個の値となることから, 入力層の要素数は 12 個とした.

表 1 Adam のアルゴリズム

Adam : $\alpha=0.001$, $\beta_1=0.9$, $\beta_2=0.999$, $\varepsilon=10^{-8}$, β_1^t と β_2^t は β_1 , β_2 を t 乗したものを示す.

```

 $m_0 \leftarrow 0$ 
 $v_0 \leftarrow 0$ 
 $t \leftarrow 0$ 
while 学習回数 do
   $t \leftarrow t + 1$ 
   $g_t \leftarrow \nabla_{\theta} f_t(\theta_{t-1})$ 
   $m_t \leftarrow \beta_1 \cdot m_{t-1} + (1 - \beta_1) \cdot g_t$ 
   $v_t \leftarrow \beta_2 \cdot v_{t-1} + (1 - \beta_2) \cdot g_t^2$ 
   $\hat{m}_t \leftarrow m_t / (1 - \beta_1^t)$ 
   $\hat{v}_t \leftarrow v_t / (1 - \beta_2^t)$ 
   $\theta_t \leftarrow \theta_{t-1} - \alpha \cdot \hat{m}_t / (\sqrt{\hat{v}_t} + \varepsilon)$ 
end while

```

表 2 各実験における実験環境

種類	CPU	メモリ	OS
文献[5]	Intel i7 870 2.93GHz	8GB	Windows 7 Professional 64bit
文献[9]	AMD A8 3.0GHz	16 GB	Windows 8.1 Pro 64bit
提案手法	Intel i5 4690K 3.9GHz	8GB	Ubuntu 15.04 64bit

表 3 学習機械の設定

項目	特徴抽出時	判別時
学習用データ	10,000	共有
総層数	5	共有
入力層要素数	12	共有
中間層数	3	共有
中間層要素数	60 要素×4	共有
出力層要素数	なし	1
ミニバッチサイズ	なし	10
学習回数	30,000	表 4 参照

特徴抽出時は教師なし学習を行っており, 最終的な出力層を含むことが出来ないことから, 特徴抽出時の出力層は無しとする. 識別時は 4 つの渋滞評価値毎に学習機械を構築し, 各学習機械には 1 つの出力とする. 分割した理由は, 一つの学習機械で構成してしまうと, それぞれの渋滞評価値に対する値が他の出力値に影響するためである.

次に本研究で用いた学習回数とミニバッチサイズについて説明する. ミニバッチサイズとは, 確率的最適化におけるミニバッチ学習用のパラメータで, ミニバッチ学習では学習用データを 10~100 個の集合に小分けてパラメータの更新を行うが, 小分けする集合の大きさをミニバッチサイズと呼ぶ. また, ミニバッチ学習により判別時における学

習回数は、学習用データが1,000、バッチサイズが10であるため、図4~7の横軸および表4が示す学習回数は、従来でいう学習回数の100回に相当するパラメータの更新を行っている。

4.1.3 ME-GA の設定

ME-GA における設定を表5に示す。また、ME-GA の評価式を以下に示す。

$$F = \exp\left(\frac{V_{wo}}{C_w}\right) + \exp\left(\frac{V_{in}}{C_i}\right) + \exp\left(\frac{C}{C_d}\right) \quad (3)$$

$$C = \frac{t_{delay}}{TTD} \quad (4)$$

ここで、 V_{wo} は Waitout, V_{in} は Inside, t_{delay} は Delaytime, TTD は Total Travel Distance を示し、また、 C_w , C_i , C_d は各渋滞評価値に対する重みであり、それぞれ $C_w = 100$, $C_i = 500$, $C_d = 500$ とした。

4.2 実験 1

表6および表7に、それぞれ文献[9]の手法と提案手法における学習誤差を示す。これらの表から提案手法による誤差の減少が確認される。

次に各パラメータにおける学習過程を図4~7に示す。なお、グラフ上の三角と丸の記号はそれぞれ未知データとの誤差(汎化誤差:generalization error)と既知データでの誤差(訓練誤差:training error)を示している。学習回数を重ねるにつれて汎化誤差が増加し、過学習に陥っていることが確認されたため、実験2における渋滞評価値毎の学習機械に対するネットワークパラメータは、汎化誤差と訓練誤差の差が最小となる学習回数におけるパラメータを選択した。

4.3 実験 2

表8に文献[5], 文献[9]ならびに提案手法によって最適化した交通信号パラメータにおける渋滞評価値と、実測した

表4 渋滞評価値毎の学習機械における学習回数

渋滞評価値	学習回数 ^{※1}
Waitout	26
Inside	20
TTD	60
Delaytime	20

※1 学習回数はミニバッチ学習における学習回数を示す。

表5 ME-GA に対する設定

世代数	500
個体数	300
交叉率	100%
突然変異率	5%
交叉法	2点交叉
選択方法	ルーレット選択
評価式	式(3)

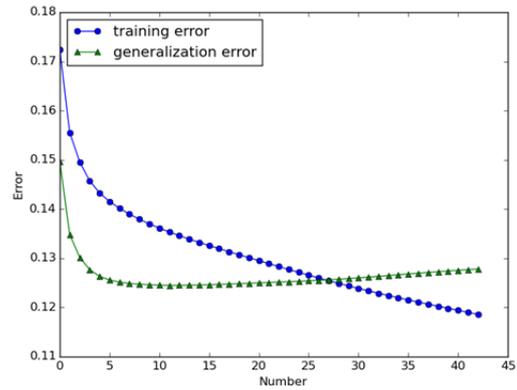


図4 Waitout における学習誤差

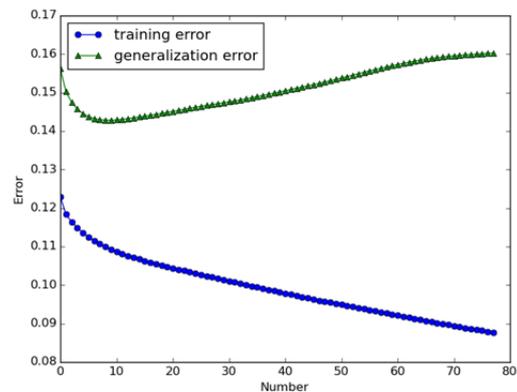


図5 Inside における学習誤差

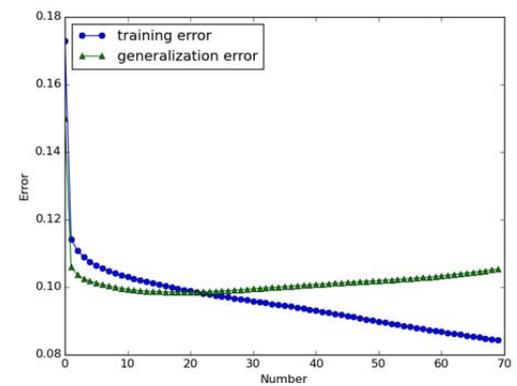


図6 TTD における学習誤差

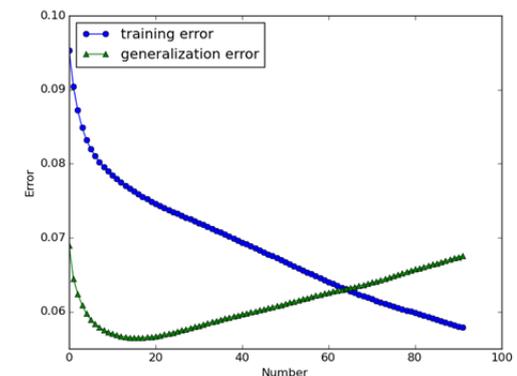


図7 Delaytime における学習誤差

表 6 文献[9]における学習誤差(%)

渋滞評価値	既知データ	未知データ
WaitOut	13.0	11.9
Inside	17.6	14.3
TTD	12.2	10.6
Delaytime	9.4	7.6

表 7 提案手法における学習誤差(%)

渋滞評価値	既知データ	未知データ
WaitOut	12.5	12.5
Inside	10.9	14.2
TTD	9.9	9.9
Delaytime	6.3	6.3

表 8 各種法における渋滞評価値

手法	WaitOut	Inside	GoneOut
文献[5]の手法	324	1003	4111
文献[9]の手法	521	847	4186
提案手法	326	960	4293
実測値	859	1122	3569

表 9 各種法における最適化処理時間

手法	処理時間
文献[5]	約 19 時間
文献[9]	約 15 分
提案手法	約 15 分

交通信号パラメータにおける渋滞評価値を示す。表 8 より提案手法を用いることで交通流シミュレータを用いた文献[5]および文献[9]の手法を上回る渋滞緩和性能が確認された。

また、表 9 に示す通り最適化処理時間は約 15 分となり、文献[9]とは同等であるが、文献[5]の約 99%もの処理時間の短縮が確認された。

5. まとめ

本研究では、文献[9]における、交通流シミュレータをシミュレータの入出力関係を学習させた学習機械へ置き換える処理時間短縮手法を改善するため、確率的最適化による深層学習を用いた学習誤差減少を図る手法と、ME-GA に対して新たな制約を追加する手法を提案した。

実験の結果、文献[5]に対しては処理時間が約 99%短縮され、渋滞評価値についても大幅な改善が見られた。また、文献[9]に対しては、処理速度についてはほぼ同等であったが、学習誤差の減少により渋滞評価値の大きな改善が見られた。

しかしながら、渋滞評価値毎の学習誤差を見てみると、

汎化誤差が訓練誤差を下回る場合が見られる。これは、学習用データよりも、汎化誤差計測用のデータが偏っているためであった。また、処理時間に関しては、文献[5]よりも大幅な処理時間短縮が確認されたが、交通情報取得のために広く用いられている VICS の更新時間が 5 分となっており[22]、その時間内に収めることが実運用性の向上に繋がると考えられる。

今後は、学習用データと汎化誤差計測用データの見直しと、さらなる処理時間短縮のため、ME-GA における個体毎の適応度計算を並列化することなどによる処理時間短縮を目指す。

参考文献

- [1] 国土交通省, "道路行政の業績計画書", <http://www.mlit.go.jp/road/ir/ir-perform/h18/07.pdf>.
- [2] Takahashi, S., Kazama, H., Fujikura, T., Nakamura, H., "Adaptive Search of an Optimal Offset for the Fluctuation of Traffic Flow using Genetic Algorithm", IEEJ Trans. on Industry Applications, Vol. 123, Issue 3, pp. 204-210, 2004.
- [3] 高橋聖, 風間洋, 藤倉智一, 中村英夫, "遺伝的アルゴリズムによる交通流量の変動に適応した最適信号機オフセットの探索", 電学論 D, Vol.123, No.3, pp.204-210, 2003.
- [4] Nishihara, T., Wijaya, IGPS., Matsumoto, S., Koutaki, G., Uchimura, K., Sugitani, H., and Ishigaki, S., "The Verification with Real-World Road Network on Optimization of Traffic Signal Parameters using Multi-Element Genetic Algorithms", ITS World Congress 2012, AP-00144, 2012.
- [5] 西原稔貴, I Gede Pasek Suta Wijaya, 松本駿太, 上瀧剛, 内村圭一, 杉谷浩, 石垣新一, "マルチエレメント GA による道路交通信号パラメータの最適化と実環境における検証", 信学会 ITS 研究会技術報告, Vol.111, No.441, pp.263-268, 2012.
- [6] Warberg, A., Larsen, J. and Jrgensen, R., "Green Wave Traffic Optimization", A Survey, Informatics and Mathematical Modeling, 2008.
- [7] Saki, M. and Nagatani, T., "Transition and saturation of traffic flow controlled by traffic lights, Physica A., "Statistical Mechanics and Its Applications, Vol.325, Issue 3-4, pp.531-546, 2003.
- [8] 除家興, 井川明, 柴田直樹, 伊藤実, "GreenSwirl:交通渋滞の緩和を目指した信号機制御および経路案内方式の提案と性能評価", 情報処理学会論文誌, Vol.57, No. 1, pp.66-78, 2016.
- [9] 中村新之介, 植村匠, 上瀧剛, 内村圭一, "深層学習とマルチエレメント GA を用いた交通信号パラメータの最適化,平成 27 年度電気・情報通信学会九州支部大会, pp.344, 2015.
- [10] 岡谷貴之, 斎藤真樹, "ディーブラーニング, コンピュータビジョン最先端ガイド 6", アドコム・メディア株式会社, 初版, 2013.
- [11] Hinton, G. E., Osindero, S. and Teh., Y.-W., "A fast Learning Algorithm for Deep Belief Nets", Neural Computation, Vol. 18, pp.1527-1544, 2006.
- [12] Vincent, P., Larochelle, H., Lajoie, L., Bengio, Y. and Manzagol, P.-A., "Stacked Denoising Autoencoders: Learning Useful Representations in a Deep Network with a Local Denoising Criterion", Journal of Machine Learning Research, Vol. 11, pp. 3371-3408, 2010.
- [13] Hinton, G. E., "A Practical Guide to Training Restricted Boltzmann Machines", Technical report, 2010.
- [14] Hinton, G. E., "Training Products of Experts by Minimizing Contrastive Divergence", Neural computation, Vol. 14, No. 8, pp. 1771-1800, 2002.
- [15] Hubel, D. H. and Wiesel, T. N., "Receptive Fields, Binocular

- Interactions, and Functional Architecture in the Cat's Visual Cortex", *Journal of Physiology*, Vol. 160, pp. 106-154, 1962.
- [16] Fukushima, K., and Miyake, S., "Neocognitron: A New Algorithm for Pattern Recognition Tolerant of Deformations and Shifts in Position", *Pattern Recognition*, Vol. 12, pp. 455-469, 1982.
- [17] Lecun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W. and Jackel, L. D., "Backpropagation Applied to handwritten Zip Code Recognition", *Neural Computation*, Vol. 1, No. 4, pp. 541-551, 1989.
- [18] Mkoloc, T., Karafiat, M., Cernocky, J., and Khudanpur, S., "Recurrent Neural Network Base Language Model", In *Proc. Interspeech*, Vol. 2, pp. 3, 2010.
- [19] K. P. Murphy. *Machine Learning. "A Probabilistic Perspective"*, MIT Press, 2012.
- [20] Diederik P. Kingma, Jimmy Lei Ba, "Adam: A method for stochastic optimization", *Proceedings of International Conference on Learning Representations 2015*, arXiv:1412.6980v8, 2015.
- [21] 山田浩次, 山田宗男, 堀場勇夫, 池谷和夫, 鈴木賢治, "アナログ出力ニューラルネットワークを用いた駐車率直接推定法", *情報処理学会論文誌*, Vol.36, No.3, pp. 627-635, 1995.
- [22] 一般社団法人 道路交通情報通信センター, "VICS", <http://www.vics.or.jp/index1.html>.