

シミュレーションで交通渋滞を考える

3年A組 村田 宏暁

指導教員 米田 隆恒

1. 要約

車に乗っているとよく渋滞に巻き込まれる。渋滞が発生すると、移動時間が長くなり、事故も増える。そこで私は交通のシミュレーターを作成し、円滑に交通を行うにはどのような運転、信号制御を行えば良いか研究を行った。本論文ではその研究の内容を紹介する。なお、今回のプログラムはすべて JAVA 言語で作成した。

キーワード 交通渋滞、シミュレーション、JAVA、グラフ

2. 研究の背景と目的

現在、車は非常にポピュラーな移動手段だが、交通渋滞などに巻き込まれると極端に時間がかかる。また、渋滞が増えると移動にかかる時間が長くなるだけでなく、事故が増えたり、燃費が悪くなり環境が汚染されるなど様々なデメリットがある。そこで私は、交通渋滞の発生や解消の仕組みを調べることで効果的な対策を立てられるのではないかと考え、コンピューターを用いたシミュレーションによってそれらを調べた。

一サイクル：信号が青になった瞬間から、次に信号が青になる瞬間までのこと。

オフセット：一サイクルの時間が等しい隣り合う交差点の間での、赤信号の始まる時刻のずれ。信号の一サイクルの時間を C (秒) とすると、 x (秒) のオフセットと $x+C$ (秒) のオフセットは同義である。

停止距離：緊急に停止する必要のある事象が起こってから、車が停止するまでに進んでしまう距離。

反応時間：ドライバーが状況を認識し、加速度として出力されるまでにかかる時間

3. 研究方法とその結果

(1) 用語の定義

本論文で使用する用語を次のように定義する。

渋滞： 一台の車が一つの信号の影響で二回以上止まることのある状態。

交通量： 単位時間あたりに道路に入ってくる車の数。

(2) シミュレーターの概要

今回開発したシミュレーターは以下のような道路を想定している。

- ・道は直線であり、一方通行である。

・歩行者や自転車などは存在せず、特殊自動車と二輪車を除く自動車のみが走行している。

また、[研究1]、[研究2]において車は以下のような動きをする。

- ・右折や左折をしない
- ・加速や減速をする場合は必ず指定した一定の加速度 $a(\text{km/h/s})$ で加速、減速する。
- ・車間距離と停止距離の差が一定値以下になると減速する。
- ・信号の停止線までの距離が一定の範囲内でその信号が赤のときも減速する。
- ・上の二つのどちらにも当てはまらず、加速しても最高速度を超えない場合は加速する。
- ・上の三つのいずれにも当てはまらない場合は加速する。

(3) 研究事項

今回、私は次の四つの研究を行った。

[研究1] 交通量を時間的に集中、分散させた際の、一定数の車が通過するまでの総所要時間の変化について

[研究2] オフセットを変化させた際の、一定の距離を通過するのに要する時間の変化について

[研究3] 文献[4]に記載されている車両挙動モデルと、それを少し改良した車両挙動モデルの検証

[研究4] 新たな車両挙動モデルの開発
次に、それぞれの研究について詳しく説明する。

[研究1]

シミュレーターの実行中に交通量を変化させることができるシミュレーターを作成し、信号が一つだけある一本の道路において、交通量を変化させた場合の車の挙動を観察した。その際、結果的に通過する車の数が同じであっても、その車間距離の違いによって総合的にかかる時間に差が出るのではないかと考えた。それを検証するために、シミュレーターの実行中に一度交通量が増えるようなプログラムを作成した。そして、交通量の変化量を変化させてシミュレーションを行い、初めの車が道路に進入してから最後の車が道路から出るまでの時間を記録し、グラフにした。また、渋滞が起きるかどうかで大きな変化が起こるのではないかと考え、渋滞が起こる最小の交通量に近い $8.5(\text{台/分})$ を中心に変化させるようにした。

[結果1]

図1は、まず $8.5 - \frac{x}{2}$ (台/分) のペースで 100 台の車を道路に入れた後、 $8.5 + \frac{x}{2}$ (台/分) のペースで車を 100 台道路に入れ、それらの車がすべて道路を通過しきるまでの時間を y 秒としたグラフである。

交通量の変化量と総所要時間との関係を表したグラフ

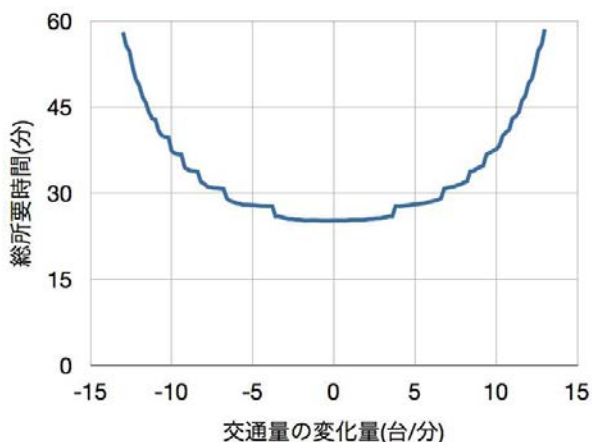


図 1

[考察 1]

仮説通り、結果的に通過する車の数が同じでも、すべての車が道路を抜けるまでの時間が変わった。また、交通量の変化が小さく、最大の交通量が少ない方がかかる時間は短くなった。そのことから、大規模な渋滞が起これるとその解消に時間がかかるため、結果的に通過するまでに要する時間が長くなったと考えられる。そのため、前方が混雑しており渋滞が起これそうな状態の時には、渋滞が起これる前に速度を落として車間距離を開けるなどして、後方の空いているところに交通量を分散することで全体が要する時間を短くすることができるのではないかと考えられる。しかし、前方が混雑しているかどうかを目視で把握することは難しいため、そういった情報をドライバーに伝達するシステムを作ることができれば、渋滞の解消に役立つのではないかと考えている。

[研究 2]

オフセットを適切に調整することで車が赤信号に出会う回数を大幅に減らすことができるのではないかと考え、シミュレーションを行おうと考えた。しかし、[研究 1]で作成したシミュレーターは道が一本と信号が一つしかなかったため、オフセットなどの複数の信号の関係について調べることができなかった。

そこで、道を格子状に並べることでできるシミュレーターを作成し、オフセットを変えながらシミュレーションを行い、それぞれについて一定の距離を移動するのにかかる時間の平均を求め、最も円滑に交通を行うことのできるオフセットを求めた。また仮説として、最適なオフセットを O (秒)、交差点の間隔を D (km)、最高速度を S (km/h)としたとき、

$$O = 3600 \frac{D}{S}$$

という式をたてた。これは、隣接する交差点同士の距離を最高速度で割ることにより、最高速度でこの交差点の間を走ったときにどの程度時間がかかるかを求めたものである。

[結果 2]

図 2 は交差点の間隔が 250m、最高速度がそれぞれ 30km/h、40km/h、50km/h、60km/h の場合のオフセット(x 軸)と 7.5km の道のりを通過するのにかかる時間(y 軸)の関連を調べたグラフである。

それぞれの最高速度について、通過時間が最小になるときのオフセットと、仮説でたてた式より得られた値を表にすると表1のようになり、誤差はいずれも2%未満と、非常に小さくなった。また、最高速度を40km/hとし、交差点の

間隔が125m、187.5m、250m、312.5m、375mのときのグラフ図3を作成し、同様に表2にした。こちらの表でも誤差はいずれも2%未満となったため、仮説でたてた式は正しいと言える。

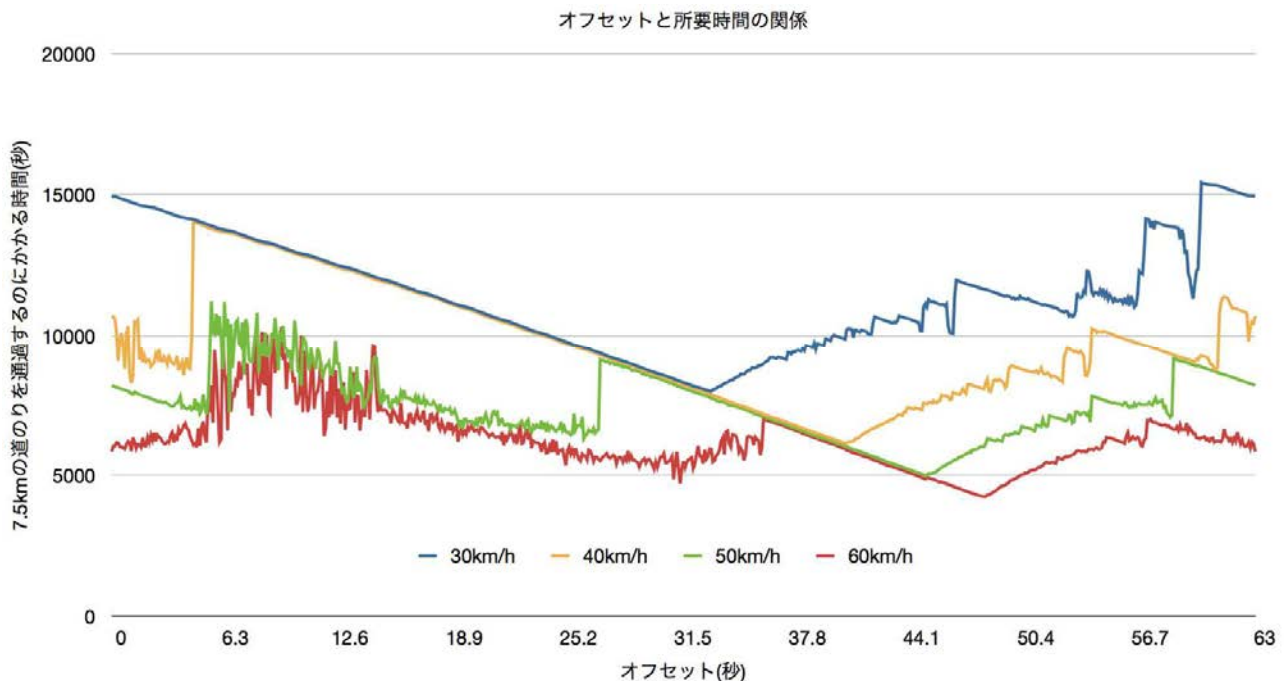


図2

制限速度	グラフより得られた最適なオフセット	仮説で立てた式より得られた最適なオフセット	右の二つの値の差	式の誤差
30km/h	29.88秒	30秒	-0.12秒	0.4%
40km/h	22.41秒	22.5秒	-0.09秒	0.4%
50km/h	17.73秒	18秒	-0.27秒	1.5%
60km/h	14.85秒	15秒	-0.15秒	1%

表1

オフセットと所要時間の関係

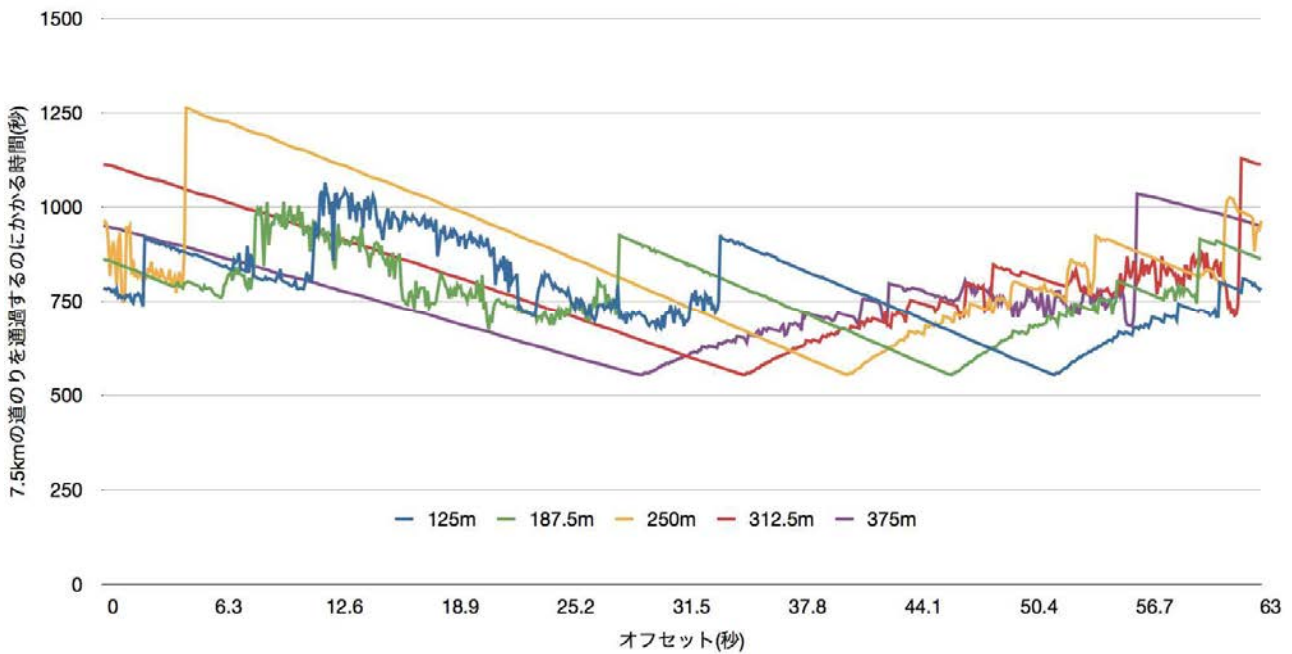


図 3

制限速度	グラフより得られた 最適なオフセット	仮説で立てた式より得られた 最適なオフセット	右の二つの値の差	式の誤差
125m	11.16秒	11.25秒	-0.09秒	0.8%
187.5m	16.74秒	16.875秒	-0.135秒	0.8%
250m	22.41秒	22.5秒	-0.09秒	0.4%
312.5m	27.99秒	28.125秒	-0.135秒	0.48%
375m	33.66秒	33.75秒	-0.09秒	0.27%

表 2

[考察 2]

仮説で立てた式がシミュレーションの実行結果とほぼ一致したが、この式は最高速度に至るまでにかかる時間を考慮していない。にも関わらず一致したのは、100m以上の距離を走る際には、そのような時間は誤差の範囲内であるからだと考えられる。また、実際に仮説で立てた式にしたがってオフセットを適切に調整した格

子状の道でシミュレーションを行ったところ、二つ目以降の信号ではほとんど車は止まらなかった。実際には右折や左折をする車や運転手による速度差などがあるため、これほど滑らかなにはならないと考えられるが、それでもオフセットを適切に調整することである程度の効果が期待できる。ただし、今回のシミュレーションでは一方通行の道についてしかシミュレーション

をしていないが、実際には一方通行の道は少なくほとんどの道に対向車線がある。また、オフセットは対向車線のものと同調し、信号の一サイクルの時間を C (秒) とするとき、片側の車線のオフセットが x (秒) ($0 \leq x$) だとすると、対向車線のオフセットは $nC - x$ (秒) (n は自然数) となる。さらに、上り線と下り線の制限速度が異なることは少なく、交差点間の距離も等しいので、最適なオフセットも一致する。最適なオフセットを O とすると $nC - O = O$ すなわち

$$C = \frac{2O}{n}$$

が成り立つ場合しか両車線のオフセットを最適なオフセットにすることはできない。また、オフセットを調整するためには、当然信号の一サイクルの時間が一致している必要があるので、それぞれの交差点で最適なオフセットに合わせて C の値を変えることはできない。すなわち、それぞれの交差点の位置と最高速度が確定した後でオフセットを調整しようとしても両車線のオフセットを共に最適なオフセットにすることができない場合が多いと考えられる。そのため、オフセットを適切に調整し、道路交通を円滑に行うためには、道路を建設する段階からそれを考慮した道路建設を行う必要がある。

[研究3]

ここまでのシミュレーターで使用していた車両の挙動モデルは、挙動そのものは人間の運転する車の挙動と大きな差はないものの、そこに

至るまでの計算式は人間のそれとはかけ離れており、どうしても誤差が出てしまう。

そこで、異なる車両挙動モデルを用いようと考え、それらの車両挙動モデルの精度を検証するために、一定の間隔で並んだ 500 台の車列の先頭の車両をキー操作に応じて加減速させ、後続の車両を検証したい車両挙動モデルに従って走行させるシミュレーターを作成した。これを用いて、文献[4]に記載されていた以下の三つの車両挙動モデルの検証を行った。

$P(x)$ は前から x 台目の車の車頭の位置、 $V(x)$ は前から x 台目の車の速度、 D は停止している際の望ましい車頭間距離、 $A(x)$ は前から x 台目の車の反応時間後の加速度、 a は感度を表す。また、すべてのモデルにおいて、加速度の上限と下限を定めている。

$$1 : A(n+1) = a\{V(n) - V(n+1)\}$$

$$2 : A(n+1) = a \frac{V(n+1)\{V(n) - V(n+1)\}}{P(n) - P(n+1)}$$

$$3 : A(n+1) = a \frac{V(n+1)\{V(n) - V(n+1)\}}{P(n) - P(n+1) - D}$$

[結果3]

$$1 : A(n+1) = a\{V(n) - V(n+1)\}$$

このモデルでは、車の、反応時間後の加速度は、前方を走行している車との相対速度に比例している。しかしこのモデルでは、 a を大きくすると、ある車両の速度変化が後方へ伝播していく際にその速度変化が増幅してしまい、増幅しない程度に a を小さくすると、後方の車両の

反応が遅れ、容易に衝突してしまう。また、実際には運転手は車頭間距離が近いほど前方の車との速度との違いに敏感になると考えられるが、このモデルでは車頭間距離によって加速度が変化しないため、それを反映できていない。

$$2 : A(n+1) = a \frac{V(n+1)\{V(n) - V(n+1)\}}{P(n) - P(n+1)}$$

このモデルは一つ目のモデルに加えて、自分自身の速度に比例し、車頭間距離に反比例するようになっている。このモデルは上のモデルと違い、車頭間距離が近いほど前方の車との速度の違いに敏感になる。しかし、衝突は比較的起こりにくくなっているものの、やはり a を大きくすると速度変化が増幅してしまい、増幅しない程度に a を小さくすると容易に衝突してしまう。

$$3 : A(n+1) = a \frac{V(n+1)\{V(n) - V(n+1)\}}{P(n) - P(n+1) - D}$$

このモデルは二つ目のモデルにおける車頭間距離を実際の車頭間距離と望ましい車頭間距離との差にしたものである。停止の直前までは順調に減速するが、前の車との車頭間距離が D より短く前の車の方が速度が遅い場合に、本来は減速するべきだが加速してしまう。

[考察3]

これらのモデルにはそれぞれ問題点があったが、それはこれらのモデルが速度変化を伴いながらも、停止などをせずに進んでいく場合を想定しているからだと考えている。実際に人間が

運転するときには、どのような状態であるかによって計算式を変えると考えられるため、このような単純なモデルでその挙動のすべてを説明することには無理があると考えている。また人間が運転している場合、自分の速度や相対速度、車間距離が同じであっても、気分や性格などの非常に細かい要因によって加速度が変化することが考えられるが、これらのモデルではそのような点についても再現できていないため、改善する必要がある。

[研究4]

参考文献[3]に記載されていたモデルは車両が順調に走行している場合を想定したモデルで、停止などをした場合には正しく動かない。そこで、条件分岐などを用いた、より複雑な式で車の動きを表現する車両挙動モデルを独自に開発した。このモデルは、前方の車両と同じ速度で理想的な車間距離を保って走行する状態を目標に車両の挙動を決定するものである。

[結果4]

このモデルでは前方を走行している車両が等速走行している場合の加速度の変化の様子は比較的人間に近いものになったが、加速度の大きさは非常に小さく、実際の車両のものとは違ったものになってしまった。また、前方の車両が加減速している場合の加速度は実際のものとはかけ離れてしまっている。

[考察4]

加速度の大きさが非常に小さくなってしまっている点についてはパラメーターや式の一部を適切に調整することで解決するのではないかと考えているが、それぞれが他の場合の動作にも関わっているため、安易に変更できず、現在は解決できていない。前方の車両が加減速している場合の挙動の精度が低いのは、このモデルは前方の車両が等速走行していることを想定して開発したことによるもので、前方の車両が等速走行していない場合の挙動を表現するための式を別に開発すれば解決できると考えられる。

4. 今後の課題

今回は研究1、2でシミュレーターを作成しシミュレーションを行い、交通渋滞の性質や、オフセットなどについて調べた。しかし、車両挙動モデルが正確さに欠けており、本論文に書いたような大まかなことはわかるが、複雑な式を用いて関係を表すなどの厳密なことは言えない。そのためモデルを正確なものにする必要があると考え、様々な車両挙動モデルの検証および開発を行ったが、まだ満足できる精度を持った車両挙動モデルは完成していないため、それを完成させたいと考えている。また、現在は格子状に並んでいる道などの単純な道でしかシミュレーションを行うことができないが、実際には曲がっていたり垂直でない角度で交わっていたりなど複雑な道も多いため、そのような道でのシミュレーションもできるようにしたい。こ

れらができれば、渋滞を少しでも少なくするためには、どのような運転をすればよいのか、どのような道を作れば良いのかがわかり、渋滞を減らすことができるのではないかと考えている。

さらに、リアルタイムで、現在の道路状況をもとに数分、数十分先の道路状況を予測し、ナビ等を通じてドライバーに指示を出すようなシステムを構築することができれば、渋滞の数を大きく減少させ、起こってしまった渋滞も素早く解消できるのではないかと考えている。

5. 参考文献

[1] 「やさしい JAVA 第二版」(高橋麻奈 著、風工舎 版、2002 年)

[2] 「碧色工房」

(<http://www.mm2d.net/applet-howto/>)

[3] 「Java Platform, Standard Edition 6 API 仕様」

(<http://docs.oracle.com/javase/jp/6/api/>)

[4] 「交通工学」(飯田恭敬 監修、北村隆一 編著、オーム社 版、2008 年)

6. 謝辞

サイエンス研究会物理班の活動において顧問の米田隆恒先生をはじめ、多くの先生方に多大なご指導を賜りました。また、先輩方にも多くのアドバイスをいただきました。この場で深く感謝申し上げます。