セル状オートマトンによる交通流の解析

～基礎と応用～

平成 16 年 2 月 18 日

数理工学研究室
板東　昭秀

目次

1：序文

現代社会の我々の移動や輸送手段にはほとんど自動車が使われていて、その自動車に対する依存度は年々高まってきている。しかし、その反面自動車の数が増えたことにより、渋滞や交通事故の数も増えてきており、その問題の解決が現在、様々な方法で行われようとしている。その方法のひとつとして、交通流シミュレーションがある。現実の交通は非常に規模が大きいため、実際に道路を用いて交通流の解析を行うのはコスト、時間的に効率がよくない。ある条件下で繰り返し実験を行うには交通流シュミュレーションは非常に効率的であるといえる。今回の研究では、あるセルが隣り合うセルとの相互作用を繰り返すことにより、その後のセルの状態を決めていくセル状オートマトンモデルの性質を用いた交通流モデルとして有名な Nagel-Schreckenberg(NS)モデルを使って交通の自然渋滞の様子を解析し、さらにこのモデルに信号という概念を加え、得られた結果から、信号によりどのように渋滞が緩和されるのか、また、最適な信号サイクルとはどれであるのかを考察する。

2：セル状オートマトンモデル

2.1 セル状オートマトンモデルの歴史

セル状オートマトンモデルは、自己増殖するシステムを作るために、1940 年代に数学者のウラムとフォン・ノイマンにより考案されたもので、その後ウォルフラムらにより体系化されていった自己複製に関する最初の計算モデルである。

2.2 セル状オートマトンとは

あるセルの次のステップの状態は、そのセルの周りの状態によって決まり、
簡単なセル間の相互作用から複雑な現象を再現でき、様々な社会現象や自然現象の解析への応用ができる。

セル状オートマトンの特徴として次のようなことが挙げられる。

- 同じ大きさの均一なセルから成り立っている
- 複雑な組織構造を簡単な同一の規則で生成できる
- 近傍から次のセルの状態を定義するため各セル同士が相互作用している
- 並列計算構造なので、計算コストが削減できる

2.3 セル状オートマトンの基本原理

横一列の１次元セル状オートマトンを考える。１次元モデルの場合、次の時間ステップでの状態を決定する際、その対象となるセルの範囲をどこまでにするかが重要になる。この場合は、2 状態 3 近傍のセル状オートマトンを考える。セルの現在の位置を i 、時間のステップを t とし、このときのセルの状態を a_i^t とすると、次のステップの状態は以下のように表せる。

$$a_i^{t+1} = F(a_{i-1}^t, a_i^t, a_{i+1}^t) \quad \text{.....(2.1)}$$

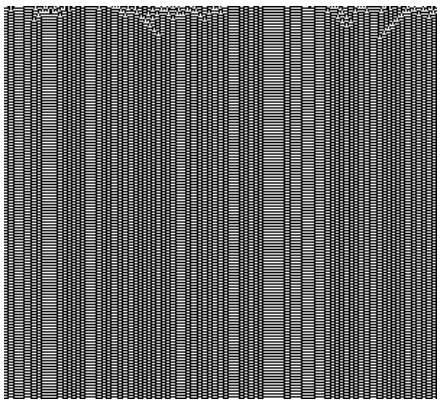
２状態３近傍の１次元セル状オートマトンの場合、パターンとしては、111, 110, 101, 100, 011, 010, 001, 000 の 2^3=８通り。それによって定められる次のセルの値の決め方は、0 または１の 2 通りずつある。したがって、このようなセル状オートマトンのルールは全部で 2^8=256 通りある。

t	{111}	{110}	{101}	{100}	{011}	{010}	{001}	{000}
	↓	↓	↓	↓	↓	↓	↓	↓
t+1	0	1	0	1	1	0	1	0

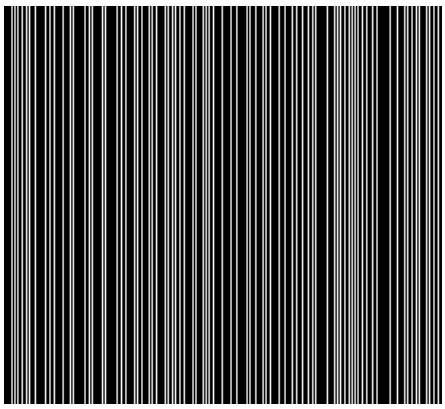
図.2.1

256 通りの各ルールには、ルールナンバーといわれる名称がつけられる。図 2.1 のルールの場合、2 進数 01011010 を 10 進数に変換すると 90 となる。したがって、ルールナンバーは 90 となる。そして、それぞれのルールナンバーによって違うセルの状態は次の 4 つに分類することができる。

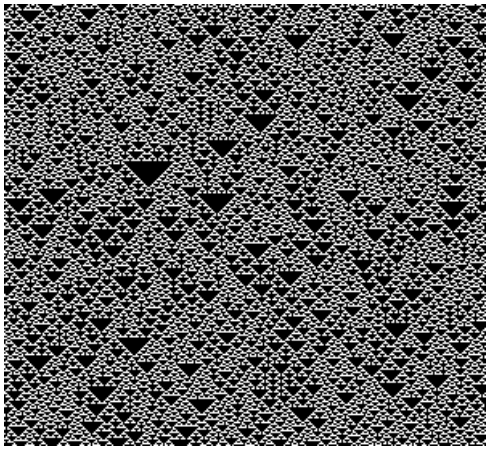
- クラス１・・・一様なパターン
- クラス２・・・局所的パターン。ある安定した周期に落ちつく
- クラス３・・・非周期的に変化するパターン。カオス的。
- クラス４・・・局所的非周期パターン。予測のつかない複雑な変化。カオスの縁。



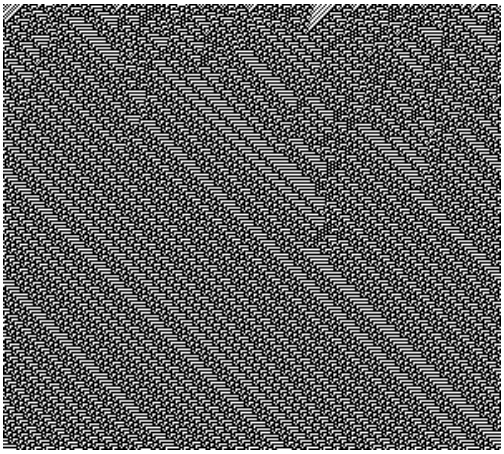
class1 :ルールナンバー37



class2:ルールナンバー12



class3:ルールナンバー22



class4:ルールナンバー41

class1,class2 は秩序状態であるため、新しい変化は生み出さない。Class3 はランダムすぎて意味のある情報は得られない。最も不可解な規則群で、秩序状態とカオス状態の間にある class4 のような状態こそが現実世界で見られる複雑な現象を引き起こす源だと wolfram は考えた。

2.2　セル状オートマトンで表す交通流

セル状オートマトンの 256 通りあるルールのうちで、wolfram のルール 184 が交通流の自然渋滞モデルとして広く知られているものである。

まず、現実での交通の流れを考えてみる。車の流れは非常に複雑で、一本の追い越しのない道路を考えてみても、スムーズに車が流れている時もあれば、渋滞している場合もある。その原因は様々な現実の要因が絡んでいるのだが、次のようなルールで単純化してみる。

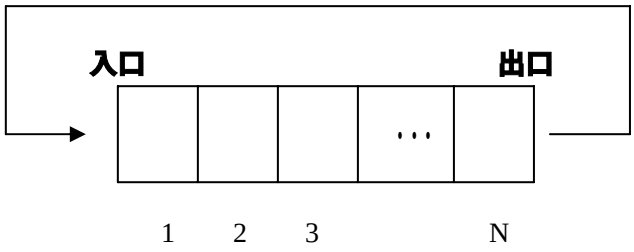
道路を 1 次元のセルに置き換え、左のセルから 1，2、・・・Nと番号付ける。また、車のいるセルを「1」、いないセルを「0」で表すことにする。そして、進行方向を右に取り、車は時間 1 ステップで、1 セル分動けるとする。前に車がいればそのセルにとどまり、前のセルに車がなければ進む。

＜周期境界条件＞

次に道路の境界条件を考える。最初に、セルに分割さらされた道路の上に M 個車を置く、最初のステップを t0 とし、次のステップ t1 の時、車は前方に障害がなければ進み、前方のセルが詰まっていればそのステップの間はその場のセルに留まることになる。この規則を全車に適用してアップデイトを繰り返し、車はいくつもの均一のセルに分けられた道路上を一方向に進んでいくことになる。

しかし、このまま道路の境界条件を定めなければ、このアップデイトを繰り返していくうち最後の N 番目のセルに車が着た際、このN番目の車は前方のセルがないために、次のステップでどう移動すればいいのか判断ができなくなる。そこでこの境界条件として周期境界条件を考える。

周期境界は、入口と出口が連結しているもので、左端の 1 番目のセルと右端のN番目のセルが円環状に繋がっていると考えることができる。つまり、アップデイトでN番目のセルの境界を越えた車は 1 番目のセルに車がいなければ、再び入口から 1 番目のセルに移動できることになる。



以上のような条件を前提に、ルール 184 の発展の様子を見てみる。

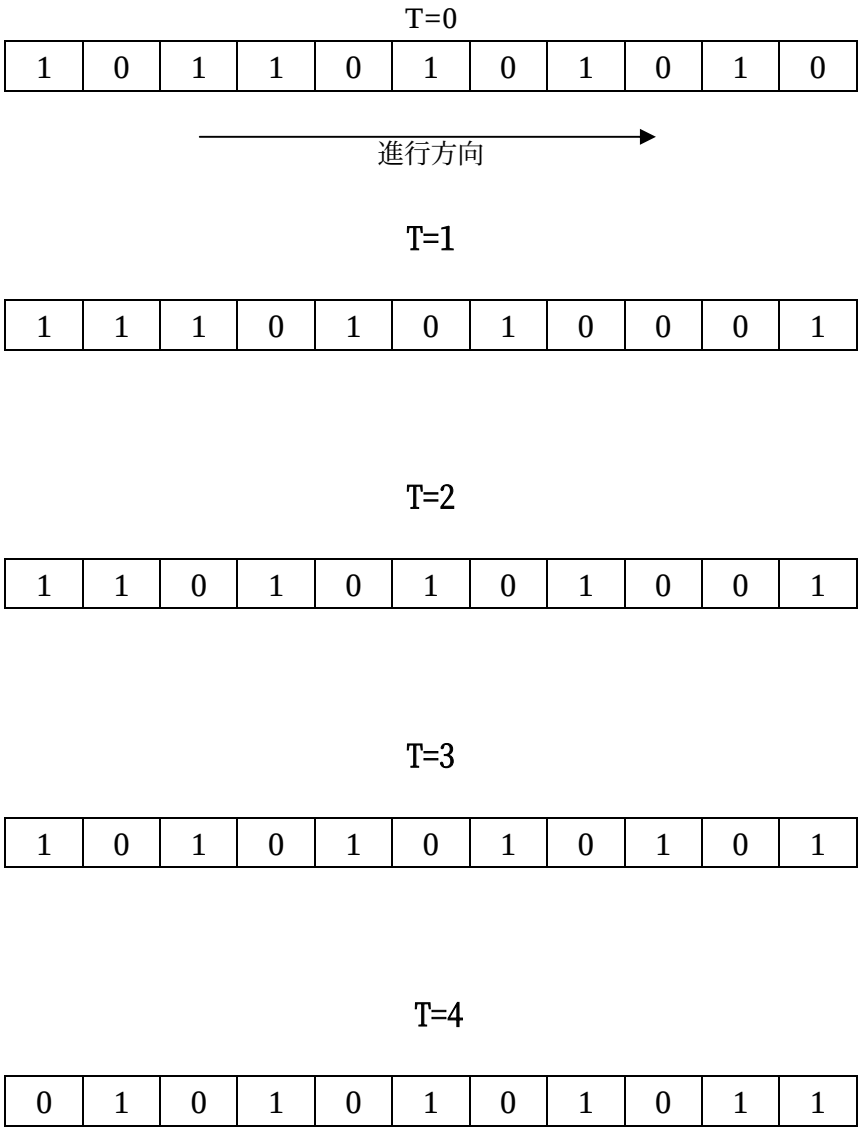


図 2.2

図 2.2 を見ると、車（1）は前のセルが空いている場合（0）のみ前のセルに進み、前の車（1）がある場合はその場のセルにとどまっている。単純なルールだが、きちんと渋滞が後ろに伝播していくのがわかる。

3：Nagel-Schreckenberg の交通流モデル

3.1 Nagel-Schreckenberg の交通流モデルとは

2で上げたルール 184 のような 2 状態 3 近傍のセル状オートマトンでは、一つ先までのセルの状態のみによってでしか自分の状態を決定することができないので、速度を 2 状態（0 か 1）しかとることができない。このモデルでは、あまりに単純化され過ぎていて現実の問題に対応しにくい。そこで、この交通流モデルの原型といえる CA に 0~5 までの加速、減速の概念を追加し、さらに車の揺らぎとしてランダム化したものが 1992 年に発表された Nagel-Schreckenberg の交通流モデルである。

Nagel-Schreckenberg の交通流モデルは以下のルールを使って、車の流れを制御する。

- 加速過程：車は１時間ステップで最高速度Ｖmax になるまで加速していく
- 減速過程：車は前方の車との距離が詰まってくると減速していく
- ランダム化：車は上であげた過程より、v=0 以外の場合にある確率で速度を変える
- 走行：以上で決めた速度の分だけ車は前方に移動する

まず「1」の数が保存するセル状オートマトンを考える。
ルール 184 のモデルと同じように道路を一つ一つのセルに分け、道路の数をＮ個とする。そこにM個車を入れる。そして、これまでと同様に、車がまったくない場合は「0」、道路が車で埋まっている場合は「１」と考える。

M 個のセルにＮ個「１」が入っている。セルは右から左に１，２、・・Ｌと番号付けて、左端のセルと右端のセルはつながっていると考える。車は一方向だけに進み、次のセルが空いていれば車は進む。もし、次のステップが車で埋まっていればその場に留まり、追越しはしないものとする。

3.2　交通流の基本量

交通流を考えていく上での基本量は、速度v、密度ρ、流速F である。これらはそれぞれ次の式で与えられる。

・密度

密度はρ＝N／Mと表される。つまりρ＝0 の場合、道路に車がない状態で、ρ＝1 の場合には道路が車で埋まっていることになる。この密度をNのかわりにする。

$$\rho = \frac{M}{N} \text{……..(3.1)}$$

・速度

平均速度については各車の速度、つまり、1 ステップでのセルを移動量を足しそれを車の数で割ることで得られる。

$$\bar{v} = \frac{1}{M} \sum_{i=1}^{n} v[i] \text{ …….(3.2)}$$

・流速

流速は交通流を見る上で一番重要な値であり、密度ρに平均速度$\bar{v}$をかけることで得られる。

$$F = \frac{1}{N} \sum_{i=1}^{n} v[i] = \frac{M}{N} \cdot \frac{1}{M} \sum_{i=1}^{n} v[i] = \rho \cdot \bar{v} \quad \text{……..(3.3)}$$

3.3 交通流モデルのシミュレーション

これまで Nagel-Schreckenberg の交通流モデルについてのルールや式を挙げてきたが、ここではそれらを前提として実際に、交通流モデルのシミュレーションを行う。

図 3.1 セル数 100 の道路に車 15 台を入れステップ 100 で回した。

Initial Car Locations

4 15 17 26 30 48 56 58 60 71 79 82 90 92 95

```
  0  1  2  3  4  5   1 2 3  4  5       6  7  8  9
 9  0  1  2  3  4  5   1 2 3   4  5       6  7  8
 8  9  0  1  2  3  4  5 1 2  3   4  5       6  7
 7  8  9  0  1  2  3  4 5 1  2  3   4  5       6
6  7  8  9  0  1  2  3 4 5 1   2  3   4  5
   6  7  8  9  0  1  2 3 4 5  1   2  3   4  5
      6  7  8  9  0  1 2 3 4   5  1   2  3   4  5
         6  7  8  9  0 1 2 3  4   5  1   2  3   4  5
 5        6  7  8  9 0 1 2  3  4   5  1   2  3   4
 4  5        6  7  8 9 0 1 2   3  4   5  1   2  3
   4  5        6  7 8 9 0 1  2   3  4   5  1   2  3
 3   4  5        6 7 8 9 0  1  2   3  4   5  1   2
 2  3   4  5        6 7 8 9 0   1  2   3  4   5  1
   2  3   4  5       6 7 8 9  0   1  2   3  4   5  1
 1   2  3   4  5    6 7 8  9  0   1  2   3  4   5
 5  1   2  3   4  5  6 7 8   9  0   1  2   3  4
   5  1   2  3   4  5 6 7  8   9  0   1  2   3  4
 4   5  1   2  3   4 5 6 7  8   9  0   1  2   3
 3  4   5  1   2  3  4 5 6  7  8   9  0   1  2
   3  4   5  1   2  3 4 5 6   7  8   9  0   1  2
 2   3  4   5  1   2 3 4 5  6   7  8   9  0   1
```

図 3.1

渋滞の様子を、一目でわかりやすいように車を数値で表しグラフ化したのが図 3.1 である。前方に車がいなければ加速していき、しだいに車が詰まってくると渋滞が始まり、その渋滞が車の進行方向とは逆に伝播していくのが観察できる。

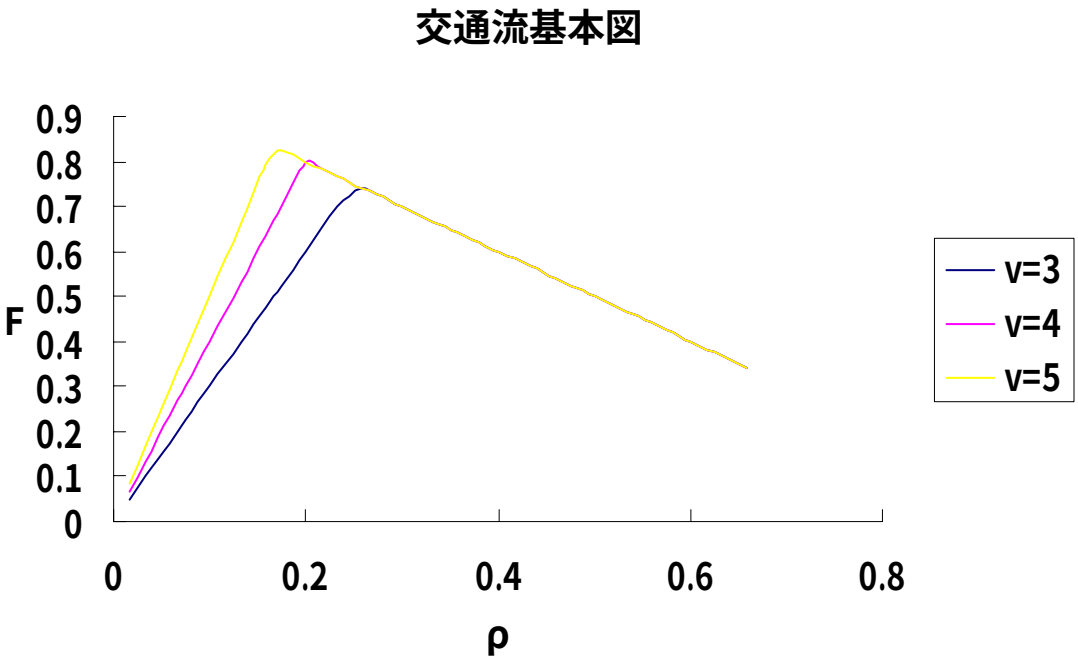


図 3.2

3.4 考察

図 3.1 では渋滞の様子を一目でわかるように表したが、図 3.2 では密度と流速に着目し、より深く渋滞の発生の様子について見てみる。設定条件は道路数 N=2000,速度はそれぞれ $V_{max}3$, $V_{max}4$, $V_{max}5$ でステップ数は 20000 である。グラフを見ると流速は交通密度の増加とともに従って減少、つまり渋滞していく。しかし、渋滞の始まりは設定したそれぞれの最高速度によって違う。$V_{max}5$ の曲線は $V_{max}3$ に比べて早く渋滞が起こっているのが見て取れ、最高速度が早ければその分早く渋滞が発生しているのがわかる。交通流の流速は密度と密接な関係があり、密度がある値を超えると、流速は下がっていく。流速が高いときが一番効率がよく、その値は設定する最高速度の値によって変わってくる。

4. Nagel-Schreckenberg の交通流モデルの発展

4.1 目的

この章では、3 章で挙げた Nagel-Schreckenberg の交通流モデルが信号のない高速道路での交通のシミュレーションに使われていることから、新たにこのモデルに信号の概念を付け加え、それにより現実での普段の交通の解析に近づけることにする。まず、最初に信号のない場合とある場合ではどのように効率が違うのかを調べる。次に、速度条件を同じに設定し、信号のサイクル数を変えてグラフ化し、その速度において、どの信号サイクル数が最適なのかを考察する。最後に、それぞれの最高速度毎に信号サイクル数を変え、速度によって最適なサイクル数を調べ考察する。

4.2 信号のルール

- 信号のサイクル数を p とする。この p の値のステップ数毎に停止、走行を繰り返す。
- 信号は $N/2$ の場所に設置し、そこに車が到達すると信号による制御を受ける。
- 車は信号が赤であれば、前方のセルが空いていようと車は速度 v=0 となりその場のセルにとどまる。信号が青であれば、車はそれまでの速度の規則通りに走行できる。

・車は、信号が青の間は、決められた最高速度 V_{max} まで加速していくことができる。

ここでは、信号の場所は $N/2$ の場所に設置してあるが、道路は2章で挙げた周期境界条件で、円環状に繋がっていると考えられる。そのため、どこに配置しようが、車が信号から受ける影響は結果的に変わらない。以上のような信号のルールをNagel-Schreckenbergの交通流モデルに追加し、信号による交通流への影響を見る。

次に実際に信号により交通の流れが制御されている様子を図4.1で示す。

```
N:    70 Vmx:        5 MX: 100
N:    70 M:     1 RHO:  0.01429 Vmx:   5 MX:        0
P:     5  R: 0.000000
  : 0.357,2.500                          234 5  678            9     0     1
  : 0.300,2.100                          23 4  5678                9     0     1
  : 0.257,1.8001                         2 3  45678                     9     0
  : 0.257,1.800     1                     2  345678                          9     0
  : 0.257,1.800    0     1                  234567 8                              9
  : 0.257,1.800   9     0     1             23456 7  8
  : 0.257,1.800        9     0     1        2345 6  7   8
  : 0.300,2.100             9     0     1   234 5  6   7    8
  : 0.357,2.500                  9     0   123 4  5   6    7     8
  : 0.400,2.800                       9   012 3  45       6     7     8
  : 0.357,2.500                          901 2  345           6     7     8
  : 0.300,2.100                          90 1  2345                6     7     8
  : 0.257,1.8008                         9 0  12345                     6     7
  : 0.257,1.800     8                     9  012345                          6     7
  : 0.257,1.800    7     8                  901234 5                              6
  : 0.257,1.800   6     7     8             90123 4  5
  : 0.257,1.800        6     7     8        9012 3  4   5
  : 0.300,2.100             6     7     8   901 2  3   4    5
  : 0.357,2.500                  6     7   890 1  2   3    4     5
  : 0.400,2.800                       6   789 0  12       3     4     5
  : 0.357,2.500                          678 9  012           3     4     5
  : 0.300,2.100                          67 8  9012                3     4     5
  : 0.257,1.8005                         6 7  89012                     3     4
  : 0.257,1.800     5                     6  789012                          3     4
```

図4.1

条件設定は道路のセル数100、ステップ数100、信号サイクル数5である。道路のセル数Nの1/2の場所において設定したサイクル数通りに信号が5ステップの間隔で青､赤が切り替わり交通の流れが制御されているのがわかる。

4.3 信号の有無による渋滞の変化

ここでは、信号を設置しない場合と設置した場合との渋滞の発生する様子の違いをグラフをもとに考察する。設定条件は道路数 N=2000,速度を V_{max}=5 としてステップ数は LP=20000 で信号の　サイクル数は 10 とした。

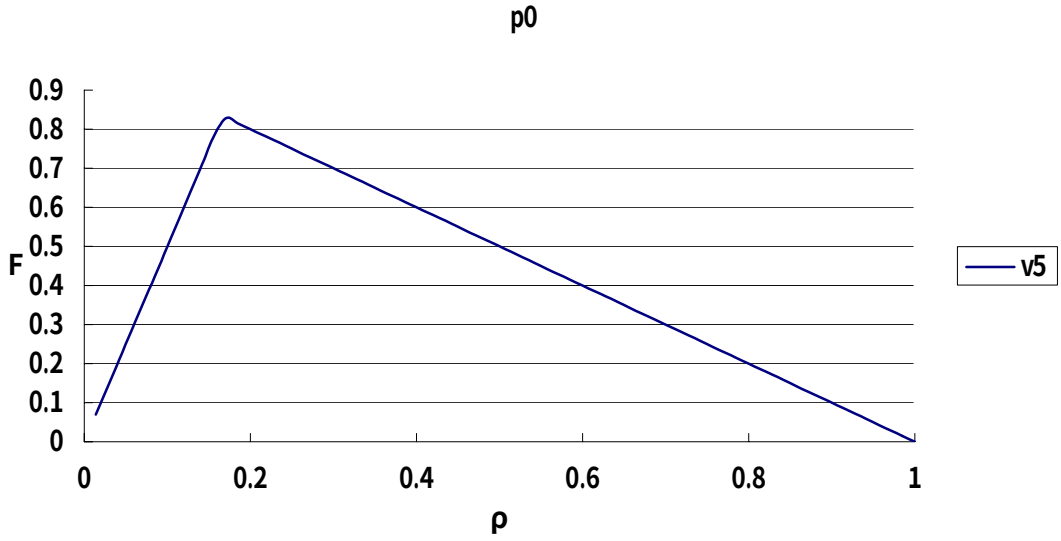


図 4.2

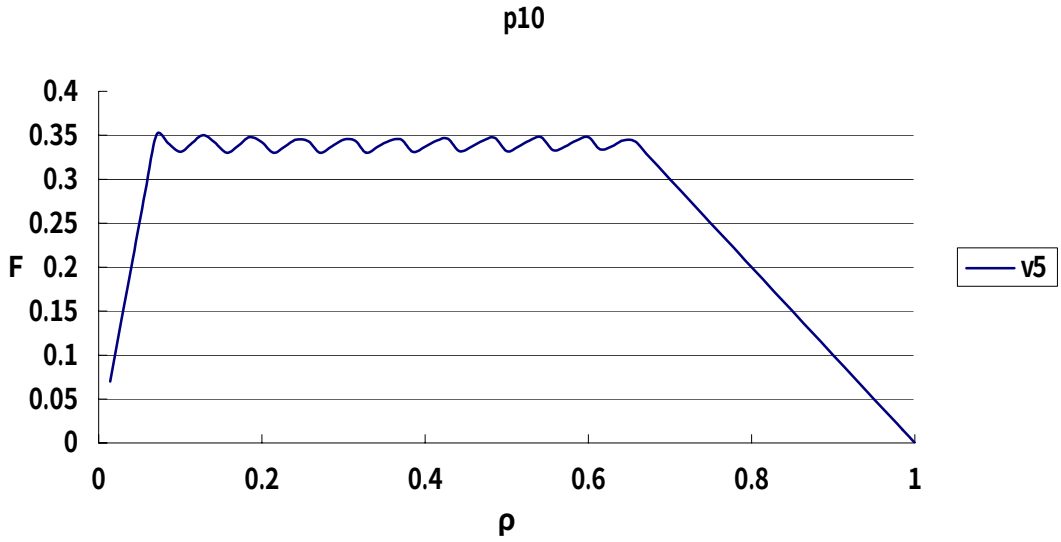


図 4.3

図 4.1 は信号サイクル 0、つまり、まったく信号がない状態でのグラフである。車の密度約 0.2%あたりまで流速が上がっているが、そこから渋滞が発生し、流速が下がり続けていく。これと同じ条件で、信号を設置し、信号サイクル 10 で交通の流れを制御したものが、図 4.2 である。信号を設置しなかった図 1 は、早くに渋滞が発生しているが、信号により交通が制御された、図 4.2 では、ある程度ピーク状態のまま高密度まで極端な渋滞が発生せずに推移していることがわかる。

4.4　信号サイクルの違いによる交通流の変化

4.3 で信号を設置することで渋滞の発生が抑制されていることが観測できたのだが、現実には必要以上に信号を設置している場合もあるはずである。その場合、交通の流れにどのような影響がでているのかを速度条件をある一定の値 $V_{max}=5$ に設定し、信号サイクルの値を変えていくことで、同一速度化での交通効率の違いを考察する。

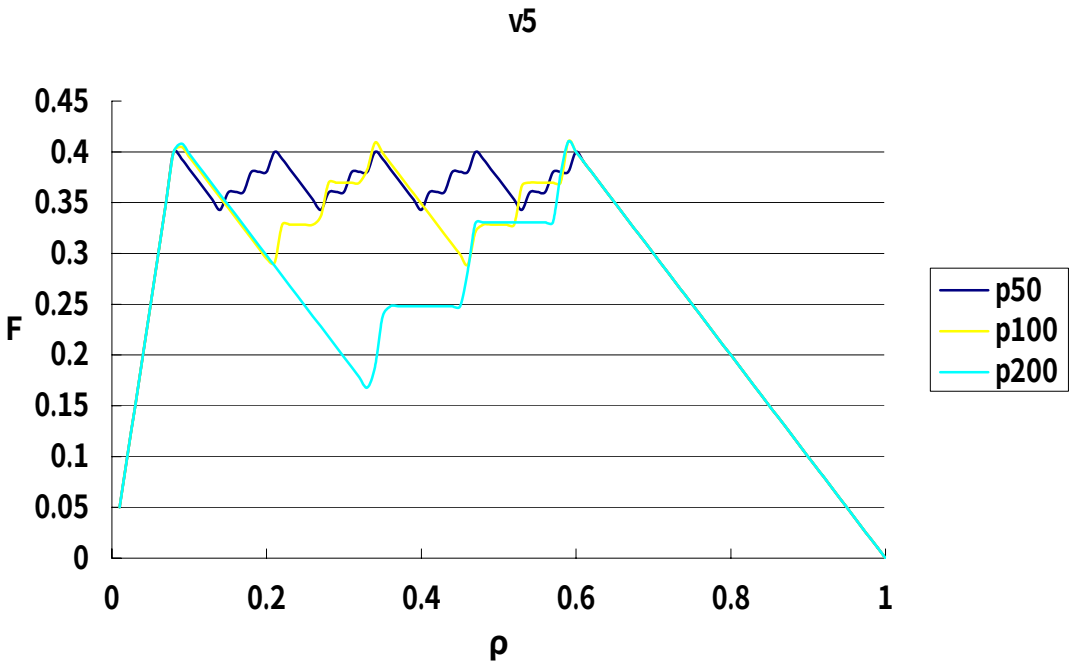


図 4.4

まず、信号サイクルを多くして、グラフ化してみた。信号サイクルはそれぞれ $p=50, p=100, p=200$ である。図 4.4 を見てわかるように、流速の最高値はそれぞれ 0.4 前後であるが、信号サイクル 200,100 では信号による渋滞の発生が顕著に見て取れる。これは、交通効率を上げるために設置した信号が、あまりに信号サイクルが大きすぎると、その機能を成さず、信号サイクルのいたずらな増加は、例えそれにより最高流速が上がったとしても、交通効率の点から見て現実的でないことを意味している。一方、信号サイクル数 50 を見てみると、信号による渋滞の発生は見られるが、流速の減少はさほど大きな値ではなく、十分許容範囲であるといえる。これ以降のシミュレーションは、この結果をもとに、大きな信号サイクルは考慮せず現実的な 10~100 のサイクル数で行うことにする。

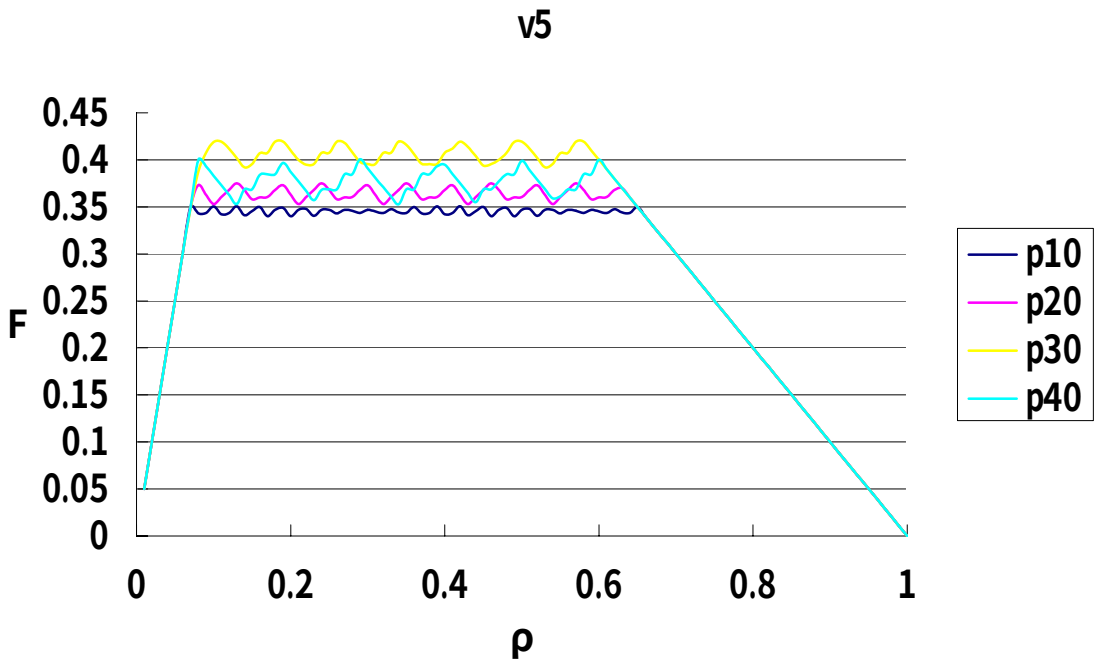


図 4.5

速度以外の設定条件は、道路数 $N=2000$,ステップ $LP=20000$ で、信号サイクルはそれぞれ、 $p=10, p=20, p=30, p=40$ である。この中で一番交通がスムーズに流れているのは信号サイクル 10 の時であるが、交通の効率、流速から見ると V_{max}=5 では交通サイクル 30 が一番適した信号サイクルであるといえる。

4.5 速度毎の信号サイクル数の違いによる交通流の変化

4.4 では一定の速度で信号サイクルを変化させた。ここではさらに各速度毎の信号サイクルの違いによる交通流への影響を観測し、それぞれの速度において信号サイクルが交通効率にどう影響しているのかを考察する。条件はすべて、道路数 N 2000,LP 20000 で設定した。

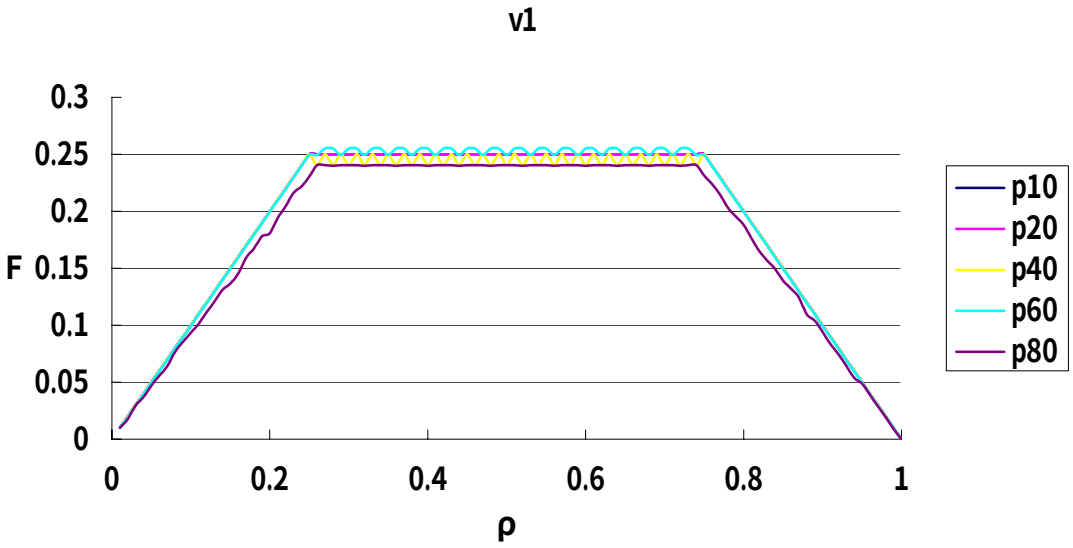


図 4.6

図 4.6 は車の速度を 1 とした、それぞれ信号サイクルは p10,p20,p40,p60,p80 でグラフを出している。どの信号サイクルも密度 0.25%付近で流速が最大になっており、信号サイクルの違いによる交通流への影響はほとんどないことがわかった。

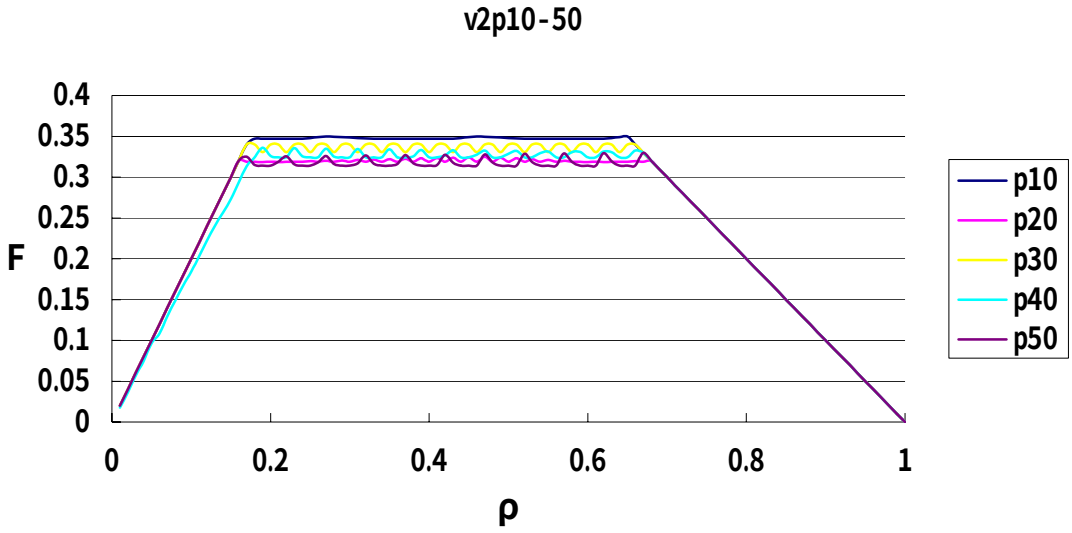


図 4.7

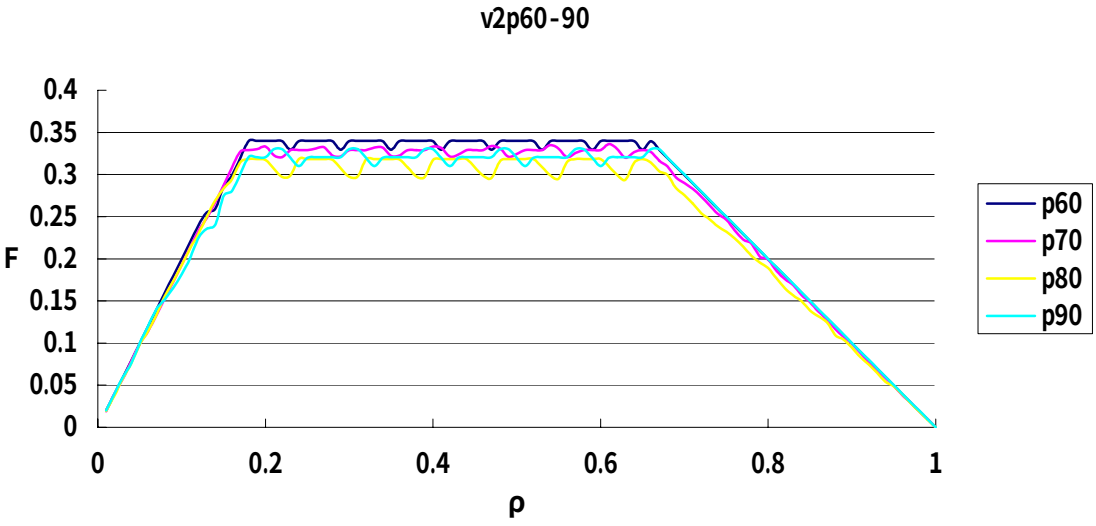


図 4.8

次に最高速度 2 でのグラフを図 4.7 に示す。まだ各信号サイクルで研著な違いは見られないが、それでも少しずつ各サイクルの最高流速の差が出ている。この中で一番、最高流速が高いのが、信号サイクル 10 の 0.35 で、これがこの速度において適した信号サイクルであるといえる。

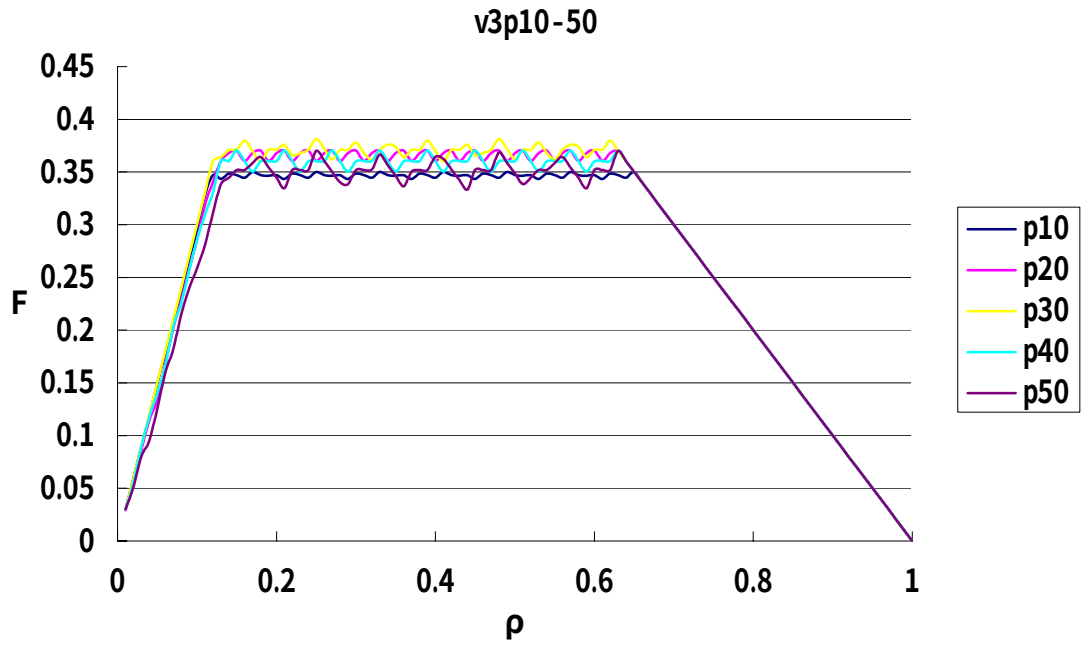


図 4.9

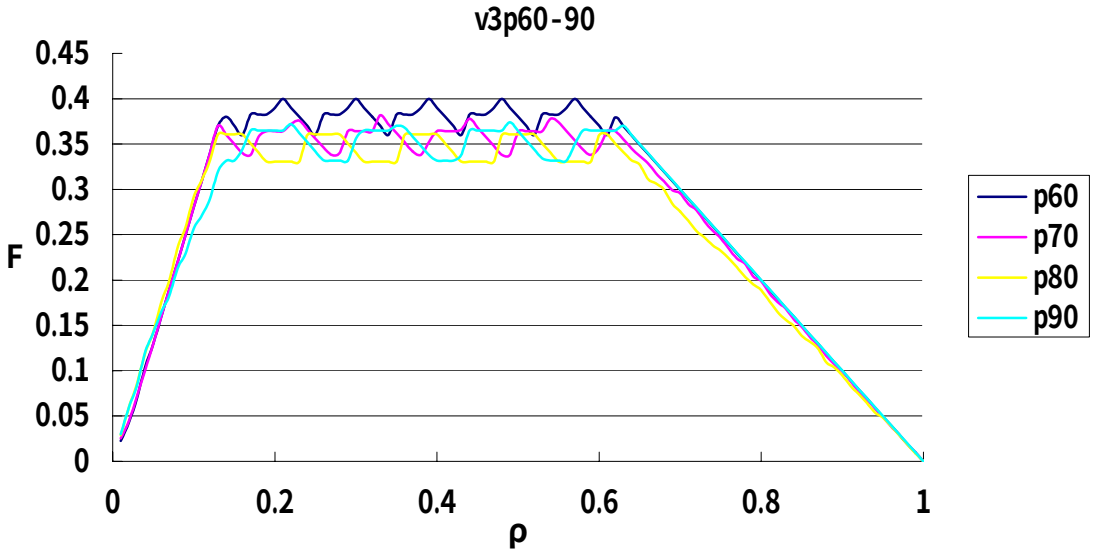


図 4.10

次に最高速度 3 での、シミュレーションを行う、信号サイクル数は 10～90 まで 10 サイクルの間隔で行い、図 4.7 にサイクル数 10~50、図 4.8 にサイクル数 60~90 でグラフ化して示した。まず、図 4.7 のサイクル数 10~50 については、どれも似たような流速に収まっている。図 4.9 のサイクル数 60~90 になってくると、しだいに差が現れ、サイクル数 60 が全体を通して、一番高い流速の値を示している。

最後に最高速度 4 での信号サイクル毎のシミュレーションを行ってみる。図はサイクル数 10~50, 60~90 で二つに分けて示す。

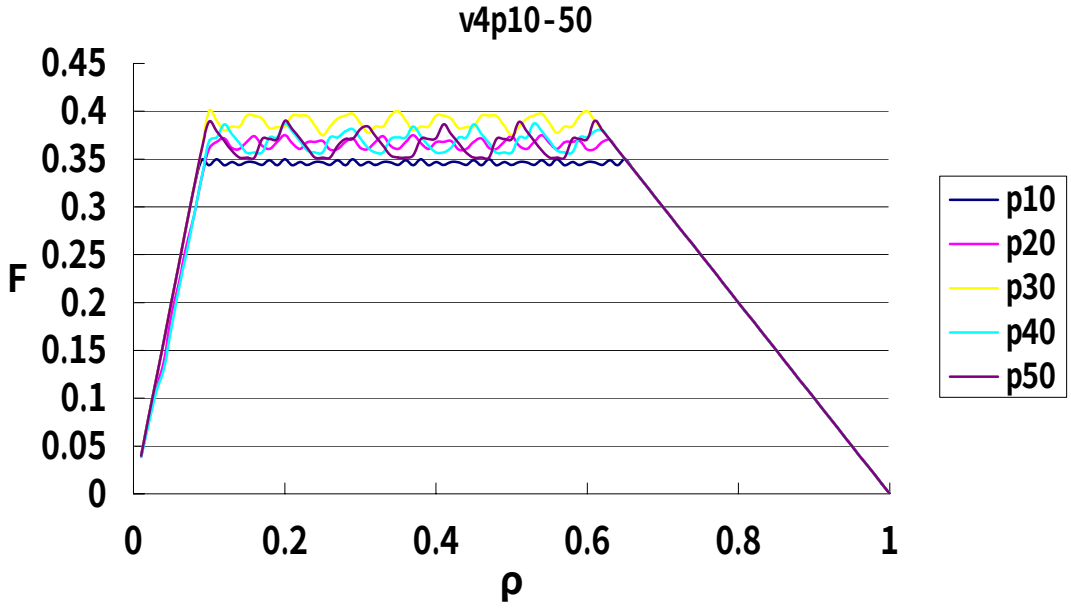


図 4.11

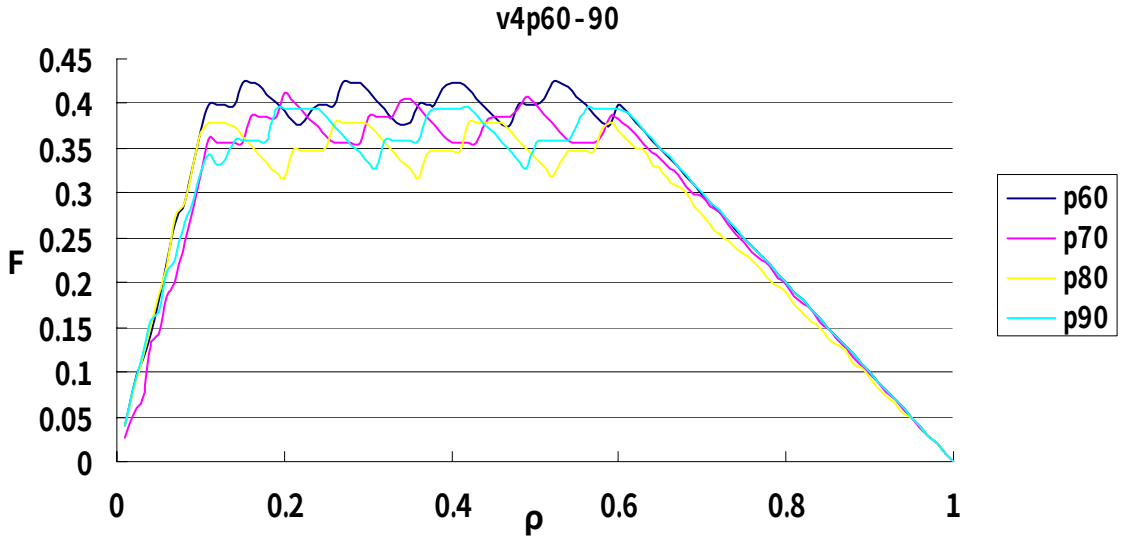


図 4.12

最高速度 4 でも、速度 3 の時と同じように、信号サイクル 60 の最高流速の値が一番高く 0.423 となっており、この速度での最適なサイクル数であるといえる。

4.6 考察

まず、4.3 において信号を設置することにより渋滞の発生を抑制し、交通の流れをスムーズにすることができる事が示された。それを前提に、決められた速度において信号サイクルを変えて、それによって交通の流れがどのように制御されるのかというシミュレーションを行い、順に最高速度を変えて、速度毎のグラフを出した。まず、最高速度 1 では、どの信号サイクルでも交通効率に大きな変化は見られなかった。最高速度 2 からしだいにサイクル数による差があらわれ、速度を上げていくに従って、車の平均流速の差が顕著に表れてきた、このことは速度が速ければ、それだけ渋滞の発生が早く、信号による影響をより強く受けることを意味している。最高速度 2 での最適な信号サイクル数は 10,最高速度 3 と 4 ではサイクル数 60、そして、最高速度 5 においてはサイクル数 30 が適した値となった。これらの結果から、信号による交通の流速を制御することで極端な渋滞の発生を抑制することでき、さらに、信号には速度毎にそれぞれ最適な信号サイクル数があることがわかった。つまり、現実においても道路によって条件や状況は様々であるが、それに応じた最適な信号サイクル数が存在し、それを見つけ出し、サイクル数をその値にすることによって、効率のよいスムーズな交通が実現できるということである。

5 まとめ

本稿では、交通流の解析の方法として Nagel-Schreckenberg の交通流モデルを使用した。最初に、そのモデルの原型である一次元セル状オートマトンのルール 184 の交通モデルを中心にセル状オートマトンについて紹介した。セル状オートマトンは、単純なルールで複雑な現象を表せることから、現在、様々な解析に使用されている。256 個あるオートマトンのルールの中で、交通モデルとしてルール 184 があり、このモデルに加速と減速を加え、速度の揺らぎとしてランダム化させたものが Nagel-Schreckenberg の交通流モデルである。

最後に 4 章で、この Nagel-Schreckenberg の交通流モデルに、さらに信号という概念を付け加え、交通流の制御を目的とした、シミュレーションを行ってきた。信号をかけることにより、最高流速は約半分まで落ちるが、極端な渋滞の発生は抑えられ、結果的にスムーズな交通ができることが証明された。また、一定の速度下で信号サイクルを変えるとにより、その速度における最適な信号サイクル数を解析し、さらに、速度毎に最適なサイクル数を導き出した。この結果から、信号を使い交通を制御することで渋滞を緩和することができ、速度に応じた信号サイクル数が適用された道路では、交通がスムーズに行えるという結論に至った。しかし、逆にサイクル数が適していないと信号はかえって交通効率を悪くするということにもなる。現実においても、その道路の交通量や制限速度に応じた信号サイクルがあり、渋滞の多い道路の中には、信号のサイクル数が適していないために渋滞が起こっている所もあるはずである。

現実の交通における渋滞の要因は多数存在するが、その渋滞を解消する方法も様々なものが存在する。今回の研究では、その一つとして最高速度毎の最適な信号のサイクルの考察を行った。今後の展望としては、現在一次元で行っている解析を 2 次元に広げ、車線を増やして、追越しも考慮にいれた、より現実的なシミュレーションを行いさらに深く交通流の研究を行いたい。

6.謝辞

本研究に際して、終始適切熱心なご指導をいただいた全卓樹教授には心から深く御礼を申し上げます。

7.参考文献

P.M.Simon and K.Nagel: “A Simplified Cellular Automaton Model for City Traffic”
Henryk Fuks:Exact results for deterministic cellular automata traffic models

８：付録資料

解析に使用したプログラム

```
#include<stdio.h>
#include<math.h>
#include<stdlib.h>

#define LP 2000        /*ステップ数*/
#define MX 1000
#define VM 5           /*最高速度*/

#define P 2000         /*信号サイクル数*/


int x[MX];
int v[MX];

float ranp()
{
        int ii,jj;
        float x,y,z;

        jj=181727;
        y=jj;
        z=rand()%jj;
        x=z/y;

        return x;
}

int sortarr(int N, int M)
{
        int i,j;
        int tmp;

        for(i=1;i<=M-1;i++)
```

```
        {
                for(j=i+1;j<=M;j++)
                {
                        if(x[j]<x[i])
                        {
                                tmp=x[j]; x[j]=x[i];x[i]=tmp;
                        };
                };

        };
        return 0;
}


int initarr(int N,int M)
{
        int i,j,it,tmp;

        for (i=1;i<=M;i++)
        {
        it=0;
        while (it==0)
        {
                it=1;
                tmp=rand()%N;

                for (j=1;j<i;j++)
                { if (x[j]==tmp) it=0;};

                if(it==1) x[i]=tmp;
        };
        };

        /*
        printf("Before sorting¥n");
        for (i=1;i<=M;i++) printf(" %d",x[i]);ptintf("¥n");
```

```
        */
        sortarr(N,M);


        for (i=1;i<=M;i++)
        v[i]=0;


        return 0;
}


int outparr(int N, int M,float *flx)
{

        int i,j;
        int it;
        float avv,avf;



avf=0;
for (i=1;i<=M;i++)
{
        avf=avf+v[i];
};
avv=avf/M;
avf=avf/N;
    *flx=avf;



return 0;
}

int updtarr(int N,int M,int lt)
```

```
{
        int i,j;
        int it;
        int dfx;
        int MLIT;
        int dqx;

        float rq;

        for (i=1; i<=M; i++)
        {
                j=i%M+1;
                dfx= (x[j]-x[i]+N)%N;
                if ( dfx-1<= v[i])
                {
                        v[i]=dfx-1;
                }
                else
                {
                        v[i]=v[i]+1;
                        if(v[i]>VM)v[i]=VM;
                };
        };




        if (lt!=1)
        {
                for (i=1;i<=M;i++)
                {
                        MLIT=N/2;               /*信号の位置*/
                        dqx=MLIT-x[i];
                        if( dqx<0) dqx=N;
                        if (v[i]>dqx)
                        {
```

```
                        v[i]=dqx;
                    };
            };
};

/* Update position */

for (i=1; i<=M; i++)
{
        x[i]=(x[i]+v[i])%N;
};
return 0;
}


int main()
{
        int N,M;
        int l,lm,lmx;
        int ii,jj,lt;

        float rho,flx;

        FILE *fp1;

        fp1= fopen("ファイル名","w");
        N=1000;
        printf("N: %5d Vmx: %8d MX: %3d¥n",N,VM,MX);

        lmx=70; /*プロッﾄする密度*/

        for (lm=1; lm<=lmx; lm++)
        {

                M=lm*N/lmx;
                rho=(M*1.0)/(N*1.0);
```

```
                printf("N:  %5d  M:  %5d  RHO:  %8.5f  Vmx:  %3d
MX: %8d¥n",N,M,rho,VM);

                printf("P: %5d   R: %f¥n",P);
                /* 初期化 */

                initarr(N,M);
                outparr(N,M,&flx);

                /* 信号 */
                jj=0;
                for (l=1; l<=LP; l++)
                {
                        if ( l%P ==0) {jj++; };
                        if ( jj%2==0) { lt=1;} else {lt=0; };
                        /*
                        lt=1;
                        */
                        updtarr(N,M,lt);
                        outparr(N,M,&flx);
                };

                printf(         "  %f  %f  %f¥n",rho,flx,flx/rho);
                fprintf(fp1,"  %f  %f  %f¥n",rho,flx,flx/rho);
        };

        fclose(fp1);
}
```