

平成 24 年度

修士論文

指導教授 古川康一

流言の流布と鎮静のモデル化に関する研究

- シミュレーション・アプローチ -

(Research on Modeling Circulation and Calming-down of Wild
Rumors-Simulation Approach-)

嘉悦大学大学院

ビジネス創造研究科

g11008 植村康平

目次	
第 1 章 研究の目的と概要	1
第 2 章 これまでの研究	4
第 3 章 流言とは	5
第 4 章 システム・シミュレーション	8
第 5 章 エージェントシミュレータ Artisoc の概要	10
第 6 章 シミュレーション・モデル	11
6.1 登場人物	12
6.2 エージェント同士が隣接した時の情報伝達	13
6.3 コネクション・ネットワークあり情報伝達	14
6.4 終了条件	15
第 7 章 実験	16
7.1 実験 1：コネクション・ネットワークありでの情報伝達の実験	16
7.1.1 実験 1.1：コネクション・ネットワークありでの最小鎮静者数	16
7.1.2 実験 1.2：コネクション・ネットワークありでの情報伝達での鎮静化時間	18
7.2 実験 2：エージェント同士が隣接した時の情報伝達の実験	20
第 8 章 考察	21
8.1 最小初期鎮静者数の予測	21
8.1.1 実験 1.1 に対する最小初期鎮静者数の予測関数	21
8.1.2 実験 2.1 に対する最小初期鎮静者数の予測関数	24
8.2 鎮静化までの所要ステップ数の分布－実験 1.2 の考察	25
第 9 章 おわりに	29
謝辞	32
参考文献	33
付録 1 実験 1.1 初期扇動者別の最小初期鎮静者の推定実験における終了ステップ数と不偏分散	35
付録 2 実験 1.2 初期扇動者 40 人、初期鎮静者 57 人での繰り返し実験による鎮静化ステップ数の分布の推定実験	37
付録 3 表 4 実験 1.2, 表 4 エージェント数 1000 初期扇動者 5 人、初期鎮静者 8 人での、繰り返し実験による鎮静者数が 0 になるまでのステップ数の推定	40
付録 4 実験 1.2, エージェント数 1000 初期扇動者 5 人、初期鎮静者 8 人での、繰り返し実験の平均と不偏分散	43
付録 5 実験 2.1 エージェント同士が隣接した時の情報伝達での、	

初期扇動者別の最小初期鎮静者の推定実験における終了ステップ数と 不偏分散	46
付録 6 Artisoc によるシミュレーション・プログラム	49
付録 7 シミュレーション・プログラムの説明	79

第1章 研究の目的と概要

本修士論文の目的は、コミュニケーションによって情報が流布することで発生する流言の発生メカニズムとその影響を評価することであり、流言の結果引き起こされる風評被害への対策を提言することを狙いとしている。

流言によって引き起こされる風評は巨額の経済損失をもたらすことがあり、具体的には原発事故のあった際に、福島原発から約 100km 離れた距離にある会津地方も風評被害による観光客の激減が起き、福島県産の農作物も敬遠される事態が発生している。

この為、流言の発生メカニズムや影響を調査することが必要であるが、流言は実際に調査することは難しい。そのため、本論文では、流言の実際の調査は行わずに、シミュレーションによって流言の流布のし方を調べる手法をとることとした。すなわち、流言を発生させる扇動者と、流言は鎮静化させようとする鎮静者、どちらでもない一般人を仮定して、システム・シミュレーションの技法によって流言が流布する社会モデルをコンピュータ上で構築し、その構築したシミュレータによって初期扇動者数に対して、鎮静化させるために必要な初期鎮静者数を求めること、鎮静化までのステップ数の分散を求める。

このモデルでは、流言を扇動する扇動者と流言を鎮静化させようとする鎮静者、および扇動者と鎮静者の影響を受ける一般人を仮定したシミュレーション・モデルを作成して実験を行う。

シミュレーション・モデルはマルチエージェント・シミュレーションとする。このモデルでは人をエージェントとしてシミュレーション空間内でランダムに移動させ、エージェント同士が接触することで流言が流布すると同時に、扇動者あるいは鎮静者と接触した一般人の間にコネクション・ネットワークを設定し、そのネットワークを通してのより直接的な流言の流布を可能にしている。

この情報を伝えるエージェントは、エージェント内のパラメータの組み合わせによってエージェントの振る舞いに変化を持たせ、他のエージェントに情報を伝えるモードを変化させるモデルを作成した。

エージェント・システムとは、複数の自律的に活動する要素であるエージェントが、互いに協調・競合することにより動くシステムである。エージェントは属性集合で行動を記述でき、この記述された行動をネットワークでつながれた他のエージェントとの相互関係を記述することで、エージェントの振る舞いが決まる。

この流言が流布するシミュレーション・モデルでは、人をエージェントとし

て捉え、各エージェントが自律的に他のエージェントに情報を送る様子を再現する。

このシミュレーション・モデルは、シミュレートするためのソフトウェアである既存のエージェントシミュレータ上に構築した。

流言のシミュレーション・モデルで使用するシミュレータとしては、エージェントアプローチでのシミュレーションを記述でき、世界的に広く使われている各種のシミュレータを検討した結果、**Artisoc** を利用した。

Artisoc は構造計画研究所が開発した複雑系シミュレータである。シミュレーション・モデルを **GUI** の操作と **VisualBasic** に似たわかりやすいプログラム言語と、結果を **GUI** によってマップやグラフを表示する点が特徴であり、この利点によってシミュレーション・モデルの作成時間を割くことができるので **Artisoc** を利用することとした。

流言が流布するモデルでは情報が伝わっていくモデルを作成し、そのモデルを基本モデルとして、流言が流布する要因を取り入れることで流言モデルを作成した。

この基本モデルでは、シミュレーション空間内に人であるエージェントが複数あり、最初に一人だけ情報を持っている。シミュレータは、1 ステップごとにすべてのエージェントをランダムに移動させ、移動した際に口コミを聞いた人が聞いていない人の隣に来たときに口コミを聞いていない人に口コミを伝え、このステップを繰り返すことによって多くの人へ口コミを伝えていく様子を再現している。

この基本モデルに、情報の強度、エージェント同士の接続とエージェントの種類として登場人物を加える。情報の強度は、エージェントが情報を受け取った時、その情報によって状態を変える、情報の度合を表すパラメータである。情報の強度は個々のエージェントが持ち、このパラメータはエージェントが情報を発信する際に、相手に与える影響の度合いを決定する。

情報の発信は、エージェントが同じ相手に数回接触することで接続を作り、接続を作った相手に情報を確率的に伝える。これは現実での人が属しているグループによる情報伝達を再現している。

情報の伝達は登場人物によってその伝達強度を変える。登場人物は、流言の強度を高める扇動者と流言の強度を弱める鎮静者、流言に影響される一般人を用意する。この登場人物の組み合わせによって、接触したエージェントからの影響の強さを決定する。

エージェントが情報を受け取る時に、ぶつかった相手が扇動者であれば自エージェントの情報の強度を高め、鎮静者であれば自エージェントの情報の強

度を弱めることで、エージェントが互いに接触したとき、閾値によって登場人物のタイプが変化する 1 度の接触による流言の伝播と、エージェントがコネクションを作り、コネクション網による伝播の 2 通りの流言伝播を再現した。

第2章 これまでの研究

流言が広まっていく社会モデルでは、人同士の会話によって情報が広まっていくシミュレーション・モデルであるが、情報が広まっていくモデルとしては、伝染病の伝播を表す確率モデルがあり、空間内で特定のオブジェクトの状態が変わり、それが連鎖することで伝染病の伝わり方を再現する SIR(Susceptible Infected Recovered) モデルなどの数値モデルを使った研究や、Small-WorldNetworkなどのネットワークモデルを用いた研究がある。

これらの研究は、数値モデルは、微分方程式を用いたモデルであり、個々の異なる人物の振る舞いを再現できない。ネットワークモデルは、人のコネクションがあらかじめ繋がれており、そのコネクションの中で情報を伝えていくモデルである。このモデルでは、情報をやり取りする人が固定されている。

一方、本研究では、マルチエージェント・モデルを用いてシミュレーション・モデルを作成した。このモデルでは、エージェントに複数のパラメータを設定し、このパラメータの組み合わせで、個々の異なる人物の振る舞いを決定し、エージェントがシミュレーション内でランダムに情報をやり取りすることができる点が特徴である。このエージェント・モデルによって、流言を扇動する人と鎮静化させる人とどちらでもない一般人を再現し、これらの登場人物が動的に情報を伝える相手を変え、情報をやり取りさせることができる点が、本研究での特徴である。

第3章 流言とは

流言はコミュニケーションによって情報が流布するがその過程で情報が歪むことで、誤った情報が加速度的な速さで拡大していく。このような流言が発生する理由としては、コミュニケーションの過程で、伝言ゲームのように情報を送受信する際に正しく伝えることができなかったり、情報を聞く際に認知バイアスにより情報を歪曲して認識したりして、事実と異なる情報が流布すること、および、意図的に歪曲した情報を発信することが考えられる。

少しの情報の歪みが伝達していき、大きな出来事になることもある。その一つの事例として、小さな地域で発生した銀行の取り付け騒ぎから考えてみたい。

1973年3月に豊川信用金庫で取り付け騒ぎがあった。これは、一人の高校生が、電車内で友人に「信用金庫なんて危ないわよ」と話し、その雑談を聞いた他者が、電話で友人に「信用金庫がつぶれる」と話し、伝わっていくことで豊川信用金庫がつぶれるという具体的な信用金庫の名前まで上がったデマに変化して、その地域の多くの人に伝わってしまったために、銀行にそのデマ聞いた人集まってしまい、取り付け騒ぎという大きな事件となった。そして、最終的には、日本銀行から現金輸送車で現金を豊川信用金庫に送ることで解決された。

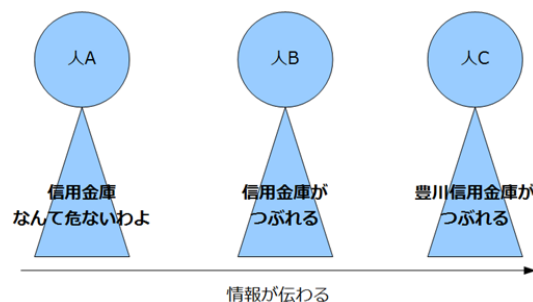


図1 風評が伝わっていく過程

この事例のように、流言はコミュニケーションによって情報が流布するがその過程で情報が歪むことで、誤った情報が加速度的な速さで拡大していく。このような流言の発生メカニズムはコミュニケーションの過程で、伝言ゲームのように情報を送受信する際に正しく伝えることができない場合や情報を聞く際に認知バイアスにより情報を歪曲して認識し、その歪曲して認識した情報がコミュニケーションによって流布すること、意図的に歪曲した情報を発信し、コミュニケーションを通して流布することによって流言となることが考えられる。

認知バイアスとは、情報を受け取ったとき人は、その情報について認知し演

算推論・発想推論・帰納推論を行う。しかしその推論は論理的に正しかったり、規範的であったりする推論ルールには必ずしも則らないで、正しくない推論を行ってしまうこともある。

(a) 帰納推論

帰納推論においては、確証バイアスと呼ばれるバイアスがしばしばみられる。確証バイアスとは、自分が持っている仮説を確かめるときに、その仮説に合致する証拠を重視する傾向をいう。

(b) 発想推論

人は推論を行う際、無意識のうちに経験則を多用している。その理由は、人は情報を処理する能力に限界があるため、入ってくる情報が多い場合、全て処理していると、すばやく判断することができないからである。

このとき、アブダクションという、ある事実があるとき、それを説明するために仮説を使い一般知識を用いた説明を行う推論バイアスが働く。

(c) 演繹推論

三段論法について、その結論が常識的な知識と一致しているときには妥当であると結論されやすくなる推論バイアスである。

以上から人は情報を受けたときに、認知バイアスによって情報を変化させる場合があると思われる。

このような流言が成立するには、情報の話題性を高め、より多くの人へ流言を流布させてしまう「扇動者」と、流言となった情報を衰退させる「鎮静者」がいることが考えられる。

「扇動者」と「鎮静者」は、情報を受け入れた時に、認知バイアスの種類によって個々人より情報の受け入れ方が異なる。この時に個々人で異なった情報を発信するようになるが、この発言の影響度で「扇動者」と「鎮静者」の2首里の登場人物に分かれる。

2種類の登場人物の違いは、「扇動者」は情報に過剰に反応する人や、意図的に流言を発生させる人といった流言を流布させるすべての人であり、「鎮静者」は流言となった情報に興味がない人や意図的に流言を減衰させるすべての人である。

扇動者と鎮静者の割合によって流言が発生する状況が変わり、流言がどのような状況で発生するのかを調べることで、流言発生メカニズムの一端を捉えることができると考えられる。

このような流言を素早く見つけ、流言が広がる前の適切なタイミングで、扇動者数に対して、必要最低限の数の鎮静者によって流言を鎮静化させることで、風評被害の拡大を防ぐことができる。

本論文では、この情報伝達をシステム・シミュレーションによって再現し、流言の発生と流布の観察を行う。流言が流布するシミュレーション・モデルは既存のシミュレータ Artisoc によって作成し、扇動者と鎮静者の割合を変え、システム・シミュレーションを行う。Aritsoc の詳細は、後述する。

デマ・風評をシステム・シミュレーションにより再現するためには、人同士がランダムに出会い、情報を伝える伝達と、社会の人的なネットワークを介した情報伝達の両者を再現する必要がある。これは社会で人が持っている、親戚関係、インターネットでのリンク、移動経路上での出会いといった関係により結びつけられたネットワークであり、これらの関係の中でデマ・風評を伝達するシステム・シミュレーションを作成する。このモデルで人同士がランダムに出会い、情報伝達は、シミュレーション・モデル内で人をランダムウォークさせ、人同士が接触したときに情報を伝達させる。

人的なネットワークでの情報伝達では、接触したエージェント同士がコネクション・チャンネルを築き、その後、エージェントがコネクションを通して確率的に情報を発信するモデルであり、人のコネクションの中での情報の流布を再現した。

第4章 システム・シミュレーション

システム・シミュレーションとは、ある一つのシステム（数人の間の共同作業から大きな一つの社会システムまでの様々なシステム）において、起こりうる事象あるいは事柄を、そこでの状態の時間経過をコンピュータ上で記述していくことによって将来を予測する、コンピュータ上での実験的な手法である。システム・シミュレーションのメリットは実際の実験と比べ、時間、費用、危険性により、実際に行うことが難しい出来事を模擬的に再現できることである。

システム・シミュレーションでは、微分方程式や待ち行列理論などの数式だけで結論づける従来手法で解明することのできなかった複雑な事象においても、時々刻々と変化する事象を、現実と同じ変化をコンピュータ上で詳細に記述していくことにより、コンピュータという仮想空間上で実験を行う手法である。そして、確率的に変動する複雑な現象を再現するために、多種多様なパラメータを組み合わせ、最適なパラメータを探すことが可能である。

システム・シミュレーションでは、再現する対象をシステムと呼び、様々な要素があつまって、何らかの目標に対して機能化されているもの考える。例として、工場であれば作業者、資材、工具、搬送機などの要素から構成されるシステムである。

モデルとはある現実の状況、対象を概念的な解釈で抽象化した仮想の対象物である。モデルは、数式、記号、あるいは言葉で表し、実体、プロセス、属性、およびそれらのシステムの目的を再現するための対象であり、プログラムで記述される。

このようなシステム・シミュレーションの例としては、工場のラインでの製品の流れを待ち行列理論によりモデル化し、最も製品を効率的に生産できることを目的にした、工場でのラインを再現したシミュレーションや、セルオートマトンという単純な格子状のセルからなるモデルを用いて自動車の流れを再現し、道路の移動効率を予測することを目的にした交通流のシミュレーションなどがある。自動車の教習者が運転動作を学ぶことを目的にし、物理モデルを用いることで自動車の挙動を再現したドライブ・シミュレーションもその例である。また、自動車などの工業製品において、負荷や強度から耐久性を見ることを目的とし、条件を満たす製品の素材や寸法を決定するために、物理モデルを要することで、製品の耐久性を確認するためのシミュレーションもある。さらに、天気、気象の予測を目的にし、物理現象をモデル化した天気・気象のシミュレーションもその一つである。

本研究での、流言のシミュレーション・モデルはマルチエージェント・システムにより実現する。マルチエージェント・システムとは、複数の自律的に活動する要素であるエージェントが、互いに協調・競合することにより動くシステムである。

エージェントは属性で行動を記述でき、この記述された行動をネットワークでつながれた他のエージェントとの関係や各エージェントの位置などの周りの環境が変化するシステムとの相互関係を記述することで、マルチエージェント・システムとして機能する。

現在、研究を進めているシミュレーション・モデルでは、人をエージェントとし、そのエージェントが自律的に他のエージェントに情報を送ることを再現する。具体的には、このネットワークをエージェント・システムにより実現する。本研究でのエージェントは、エージェントを人物として扱い、このエージェントが他のエージェントに流言を伝える。

マルチエージェント・システムを使う理由としては、個々の人物のふるまいを変化させる多様なパラメータを設定できる点と流言を伝える人物が固定されていなく、人物が状況によって流言を伝える相手を動的に選択させることができる点が重要であると考えられる。

第 5 章 エージェントシミュレータ Artisoc の概要

このシミュレーション・モデルは、シミュレートするためのソフトウェアであるシミュレータ上に構築する。我々は Swarm や StarLogo、Artisoc といったエージェントアプローチでシミュレータを構築できるシミュレータを検討した。

Swarm はアメリカの Santa Fe Institute で教育を目的に開発されたシミュレータであり、ObjectiveC および Java 言語のクラスライブラリーとなっている。このシミュレータは、汎用的なプログラム言語をつかうため自由度が高いこと。実行速度が速いことが特徴である。

StarLogo は Massachusetts Institute of Technology によって開発されたシミュレータであり、2 次元の GUI と習得しやすい LEGO 言語により、プログラミングの経験が少なくても容易にシミュレーション・モデルを構築することができるが、シミュレーション・モデルによって可能なことが少ないことが欠点である。

Artisoc は構造計画研究所が開発した複雑系シミュレータである。シミュレーション・モデルを GUI の操作と VisualBasic に似たわかりやすいプログラム言語で構築でき、結果を GUI によってマップやグラフによって表示できる点が特徴である。

流言のシミュレーション・モデルで使用するシミュレータとしては、エージェントアプローチでのシミュレーションを記述でき、容易にシミュレーション・モデルを構築でき、この利点によってシミュレーション・モデルの作成時間を短縮することができるので、Artisoc を利用することとした。

流言が流布するモデルでは口コミが伝わっていくモデルを基本モデルとして、流言が流布する要因と要素を取り入れることで流言モデルを作成した。

第6章 シミュレーション・モデル

流言はコミュニケーションによって情報が流布するがその過程で情報が歪むことで、誤った情報が加速度的な速さで拡大していく。

このような流言が成立するには、情報をこの扇動者は情報の話題性を高め、より多くの人へ流言を流布させてしまう「扇動者」と、流言となった情報を減衰させる「鎮静者」がいることが考えられ、この扇動者と鎮静者の割合によって流言が発生する状況が変わり、流言がどのような状況で発生するのかを調べることで、流言発生メカニズムの一端を捉えることができると思われるので、システム・シミュレーションによってその状況の再現を行い、流言の発生と流布の観察を行う。流言が流布するシミュレーション・モデルは Artisoc によって作成し、扇動者と鎮静者の割合を変え、システム・シミュレーションを行う。

基本モデルとして、エージェントがランダムウォークによって移動し、情報を持っているエージェントが他のエージェントに接触した際に情報が伝わることで流言が流布する簡単なシミュレーション・モデルを作成し、このシミュレーション・モデルを発展させることで流言流布のシミュレーション・モデルを作成した。

流言が流布するシミュレーション・モデルでは、人をエージェントとして、そのエージェントが自律的に他のエージェントに情報を送ることを再現する。

この基本モデルでは、シミュレーション空間内に人であるエージェントが複数あり、1 ステップごとにすべてのエージェントをランダムに移動させ、移動した際に情報を伝えた人が聞いていない人の隣に来たときに情報を聞いていない人に情報を伝え、このステップを繰り返すことによって多くの人へ情報を伝えていく様子を再現している。

流言モデルでは、エージェントの内部パラメータに情報の強度である「流言信念強度」、エージェント同士のコネクションとエージェントの種類として登場人物を加える。

情報の強度は、エージェントが情報を受け取った時、その情報によって状態を変える、情報の度合を表すパラメータである。情報の強度は個々のエージェントが持ち、このパラメータはエージェントが情報を発信する際に、相手に与える影響の度合いを決定する。

登場人物は、流言の強度を高める扇動者と流言の強度を弱める鎮静者、流言に影響される一般人を用意する。この登場人物の組み合わせによって、接触したエージェントからの影響を決定する。エージェントが情報を受け取る時に、

ぶつかった相手が扇動者であれば自エージェントの情報の強度を高め、鎮静者であれば自エージェントの情報の強度を弱める。自エージェントは、情報の強度がある閾値を超えて大きくなるとそのタイプが扇動者になり、ある閾値以下になればそのタイプが鎮静者となる。

6.1 登場人物

流言が発生する要因は、流言はコミュニケーションによって情報が流布するが、その過程で情報が歪むことで、誤った情報が加速度的な速さで拡大する。このような流言の発生メカニズムはコミュニケーションの過程で、伝言ゲームのように情報を送受信する際に正しく伝えることができない場合や、情報を聞く際に認知バイアスにより情報を歪曲して認識し、その歪曲して認識した情報がコミュニケーションによって流布すること、意図的に歪曲した情報を発信され、コミュニケーションを通して流布することによって流言となることが考えられる。

このような流言が成立する流布過程には、情報をこの扇動者は情報の話題性を高め、より多くの人へ流言を流布させてしまう扇動者と、流言となった情報を衰退させる鎮静者がいることが考えられる。

「扇動者」は情報に過剰に反応する人や、意図的に流言が発生させる人といった流言を流布させるすべての人であり、「鎮静者」は流言となった情報に興味がない人や意図的に流言を減衰させるすべての人である。

扇動者が流布させた流言を鎮静者によって鎮静化させることができる。よって扇動者の数の割合ごとに、流言を衰退させるために必要な鎮静者の数を扇動者の関数として決定するのが本論文の目的である。

流言モデルでのエージェントは鎮静者、扇動者、一般人の 3 種類のエージェントを設定しており、この登場人物によって流言の伝達処理が異なっている。エージェントの登場人物はシミュレーションの初期設定で 3 人の登場人物の割合を決める。初期の扇動者および鎮静者の割合を、それぞれ初期扇動者割合、初期鎮静者割合と呼ぶ。

これらのエージェントによる流言伝達パターンは、出会い時の会話による流言伝達と、電話や電子メールといった人とのつながりによる流言の伝達パターンの 2 種類を再現している。この 2 種類の情報伝達は、以後、「エージェント同士が隣接した時の流言の伝達」と「コネクション・ネットワークありでの流言の伝達」と呼ぶ。

6.2 エージェント同士が隣接した時の情報伝達

エージェント同士が隣接した時の情報伝達は、人が出会い会話することによる流言の伝達を再現している。

この流言伝達は個々のエージェントがシミュレーション空間内で、前方 180 度の範囲でランダムに進行方向を決定し、1 の距離を移動させることでランダムウォークをさせる。この時に移動したエージェントの周り 1 の距離にある他のエージェントがあるとき、移動したエージェントが流言の伝達処理を行う。

流言の伝達は周りのエージェントの種類によって異なり、扇動者であれば、エージェント内部のパラメータである流言信念強度に加算を行い、鎮静者であれば減算する。一般人場合は何もしない。このパラメータは上限閾値を超えれば扇動者、下限閾値をより低くなれば鎮静者、上限閾値と下限閾値の間の値で一般人の状態になる。この加減算されるパラメータを流言伝達強度と呼ぶ。



図 2 エージェント同士が隣接した時の流言伝達

6.3 コネクション・ネットワークありでの情報伝達

人とのコネクション・ネットワークありでの流言の伝達は、接触した個々のエージェント記憶し、その記憶した複数のエージェントと接触に伴う会話による流言伝達と同じ処理を行う。

会話での流言伝達の違いは個々のエージェントが記憶したエージェントと情報伝達を行うことであり、流言伝達を行うエージェント同士が離れていても流言を伝えられる点と処理の回数を接触に伴う会話による流言伝達の処理よりも多くする点である。この処理では接触に伴い関係を持ったエージェント同士が電話やメールなどで離れた場所での流言伝達を再現しており、会話よりも人とのコネクション・ネットワークありでの流言の伝達が多いと考察できるので、会話による流言流布よりも伝達処理を増やした。またこの回数のパラメータを伝達ネットワークによる伝達頻度と言う。

伝達ネットワークによる情報伝達では、エージェントが同じ相手に接触することでコネクションを作り、コネクションを作った相手に情報を確率的に伝える。これは現実で人が属している集団の中でのコミュニケーションを再現し

ている。

伝達ネットワークによる情報伝達はぶつかったエージェントがコネクション・ネットワークを築き、ぶつけられた相手の状態によって自エージェントの状態を変化させ、コネクションを持ったエージェント同士が確率的に情報を発信する情報伝達であるが、エージェントが属している複数の集団に伝わることで、一つエージェントから複数の別の集団に属しているエージェントへ流言が流布することを再現した。

流言の伝達処理で加減算された流言信念強度のパラメータは、エージェントの登場人物を変化させる。

流言信念強度が上限しきい値を超えれば扇動者、下限閾値より低くなれば鎮静者、上限閾値と下限閾値の間の値の時は一般人に登場人物の種類を決定させる。

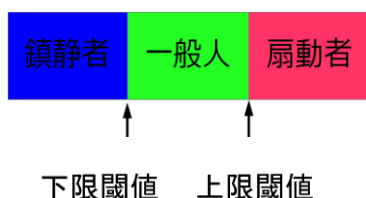


図 3 流言信念強度の閾値

上述の 2 種類の処理のいずれか(あるいは両方)を繰り返し行い、この流言の伝達の処理ごとにエージェントが受けた流言によって、エージェントの種類を煽動、鎮静、どちらでもないエージェントに変化させる。

結果の表示は、図 4 において、2 次元空間でのエージェントの位置を点で表し、点の色はエージェント種類を表している。各登場人物の数は、折れ線グラフで表示 3 種類のエージェントと扇動者と鎮静者数の 10 ステップ間の移動平均の折れ線グラフで表示した。

またシミュレーションの終了時には、テキストボックスに、ステップ数と移動平均の推移、処理が終了した時の、終了した理由と、その時のステップ数、扇動者と鎮静者の数、最大鎮静者数と最大鎮静者数を表示する。

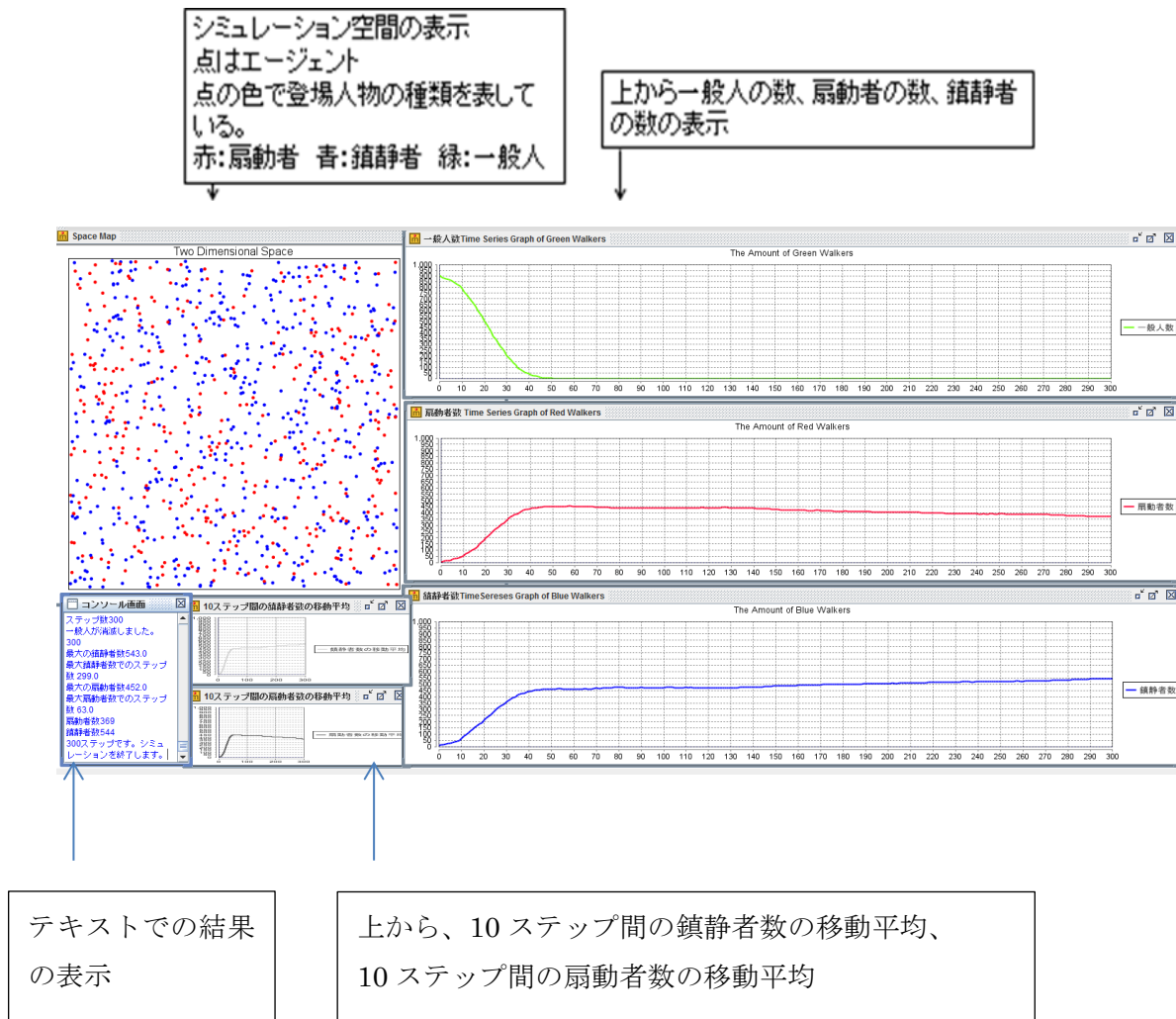


図 4 結果の表示

6.4 終了条件

この流言の流布モデルの終了条件は、エージェントの登場人物の割合が定常状態になるか、すべてエージェントが鎮静者となるか、扇動者の数が発散した時に終了する。この実験での定常状態の定義は、扇動者数が鎮静者数の 10 ステップの移動平均が、扇動者数と鎮静者数それぞれの最大数の半分を下回り、かつその値が減少し続けている場合とする。発散は上限ステップまでに定常条件で呈しなかった場合と登場人物の数の割合が連続して 10 ステップ変わらない場合を発散とした。

第7章 シミュレーション実験

シミュレーション実験は、第6章で述べた2つの流言の伝達モードごとに行い、それらの効果を確かめた。8.1節では、「7.1. コネクション・ネットワークありでの情報伝達」の場合での実験の詳細を述べ、8.2節では、「7.2 エージェント同士が隣接した時の情報伝達」の場合の実験の詳細を述べる。

7.1 実験1：コネクション・ネットワークありでの情報伝達の実験

実験は、扇動者数に対して、実験を10回繰り返して、10回ともすべて鎮静化する初期鎮静者数を求める実験(実験1.1)と、エージェントの数と登場人物の割合を固定して実験を繰り返し行い、鎮静者数が0になるまでの、ステップ数の分散を求める実験(実験1.2)を行った。

前者は扇動者を減少させる、扇動者数に対する鎮静者数を求めるが、この目的は、各扇動者数に対して、鎮静化するために必要な最小鎮静者数を求め、その扇動者数ごとの各最小鎮静者から、扇動者数に対する最小鎮静者数を扇動者数の関数として求めることが目的である。

このため実験は一般人を900人に固定して、扇動者数を5人、10人、20人、40人、80人で、各扇動者数の実験で、鎮静化する鎮静者数を求めるために、実験の扇動者数と同じ数の鎮静者数を初期値とし、結果が最大の扇動者数の半分以下になった場合を鎮静化とみなして、鎮静化しなければ鎮静者数を一人ずつ増やして、10回連続して流言が鎮静化する扇動者数に対する鎮静者数を求めた。

後者は、結果の分散を求めるため、同じ条件で実験を繰り返し、扇動者が0になるまでのステップ数の分散を求め、その分布の仕方を求めた。この実験では、すべて同じ条件で実験を繰り返した。パラメータはエージェント数を900人で扇動者を5人、鎮静者が8人、残りのエージェントを一般人とした。実験は、鎮静化した場合が100件になるまで実験を続け、それらのケースでの扇動者数がなくなるまでのステップ数から分散を求め、そのステップ数の分散がどのような分布の仕方を考察した。

7.1.1 実験1.1：コネクション・ネットワークありでの最小鎮静者数

実験1.1は、扇動者数に対する最小鎮静者数を扇動者数の関数として求めることが目的である。

実験は一般人を 900 人に固定して、扇動者数を 5 人、10 人、20 人、40 人、80 人で、この各扇動者に対する最小鎮静者数は、同じ鎮静者数で 10 回繰り返す実験を行い、すべて鎮静化した場合の鎮静者数を扇動者に対する最小鎮静者数として求めた。またこの実験では扇動数間の分散と扇動者数 40 人での最小鎮静者数とその分散を求めた。

最少鎮静者数を求めるために、実験の扇動者数と同じ数の鎮静者数を初期値として実験を行い、鎮静化しなければ鎮静者の数を一人ずつ増やしていくことで、鎮静化する扇動者数に対する最小鎮静者数を求めた。この実験での鎮静化は、このシミュレータの発散条件である、扇動者の最大数の半分の数で、扇動者が減少している場合を鎮静化とした。

この実験でのパラメータは、エージェントの強度が 1、閾値が 3、コネクション・ネットワークでの情報伝達の回数を 2 回とした。この設定では、

実験 1 の実験結果は初期扇動者数ごとに、10 回すべて鎮静化する最小鎮静者数を表 1 と図 5 に示し、各初期扇動者数別最小初期鎮静者数の終了ステップ数とその平均と不偏分散を付録 1 に記した。

初期扇動者数	最小初期鎮静者数
5	9
10	16
20	27
40	53
80	99

表 1 初期扇動者別最小初期扇動者数

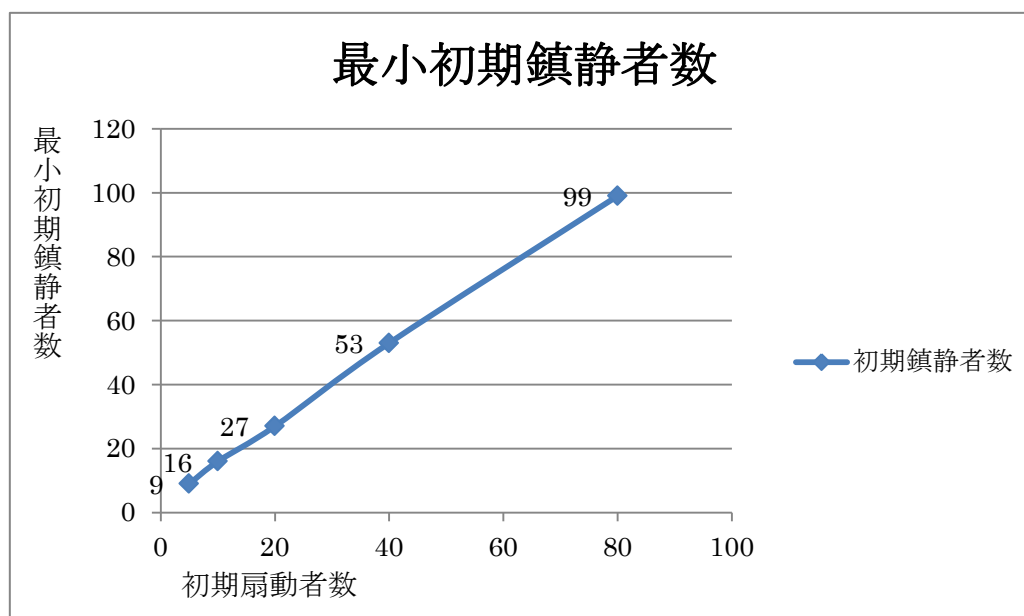


図 5 初期扇動数別最小初期鎮静者数

初期扇動者 40 人で 100 回繰り返し実験を行い、すべて鎮静化する最小初期鎮静者数は 57 人になった。その場合の終了ステップ数と終了ステップ数の分散を付録 2 に記した。

実験 1.1 の考察は、9 章に記す。

7.1.2 実験 1.2 : コネクション・ネットワークありでの情報伝達での鎮静化時間

鎮静化に要する時間の分布を求めるために、同じ条件で実験を繰り返し、扇動者が 0 になるまでのステップ数の平均と分散を求めた。

この実験では、すべて同じ条件で実験を繰り返して行った。パラメータはエージェント数を 1,000 人で扇動者を 5 人、鎮静者が 8 人、残りのエージェントを一般人とした。エージェントの強度が 1、閾値が 3.2 にした。

扇動者数の変化が、最大時の扇動者数から、終了条件で与えた扇動者の半減ポイントを通る直線であることを仮定して、鎮静者数が 0 になるまでのステップ数を予測した。

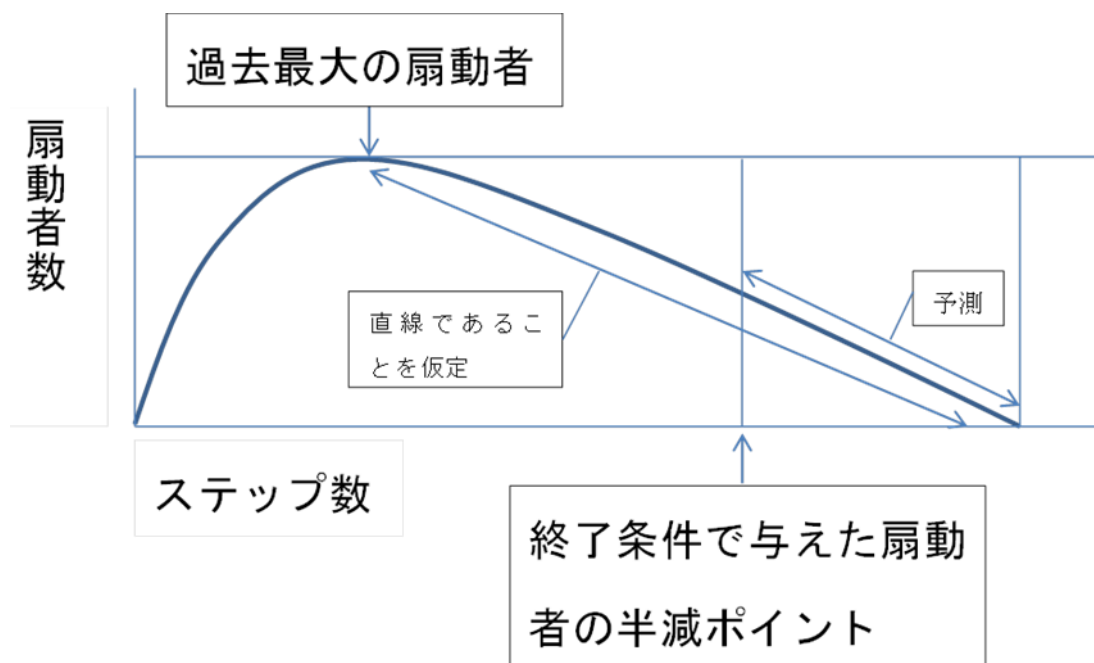


図 6 鎮静者数が 0 になるまでのステップ数の予測

実験は、鎮静化した場合の実験結果が 100 回となるまで続けた。

実験 1.2 の実験結果は、付録 3 で、最大鎮静者数とその時のステップ数、シミュレーションが終了時の扇動者数とその時のステップ数、扇動者数が 0 になるステップ数の予測を記述した。この実験での平均と標本分散は、表 2 の値になった。

平均	277.6460887
不偏分散	23392.13

表 2 実験 2 の平均と不偏分散

7.2 実験 2：エージェント同士が隣接した時の情報伝達の実験

エージェント同士が隣接した時の情報伝達での実験(実験 2.1)は、コネクション・ネットワークでの情報伝達の機能を省き、人が出会っただけでの情報伝達を再現する。この実験で省く、コネクション・ネットワークでの情報伝達は、メールや SNS などの情報伝達を再現している。すなわち、この実験では情報メディアによる影響を排除した場合での情報伝達を再現することで、情報メディアによる流言への影響を調査することを目的にしている。(表 3)(図 7)

実験方法は、コネクション・ネットワークでの情報伝達がある場合と比較したいので、コネクション・ネットワークありでの情報伝達がある実験 1.1 を行う。したがって実験 1.1 でのシミュレータの人とコネクション・ネットワークでの情報伝達の機能だけを省き、その他の条件と設定は同じ条件で実験を行い、実験 1.1 での実験結果と比較した。実験結果は付録 5 に記す。

初期扇動者数	最低初期鎮静者数
5	9
10	17
20	28
40	55
80	101

表 3 実験 2 の初期扇動者別最小初期扇動者数

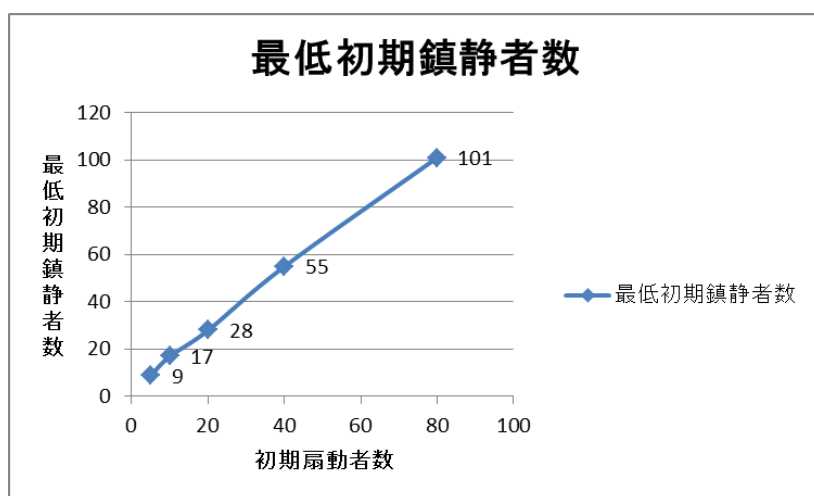


図 7 実験 2 の初期扇動者別最小初期扇動者数

第 8 章 考察

8.1 最小初期鎮静者数の予測

8.1.1 実験 1.1 に対する最小初期鎮静者数の予測関数

実験 1.1 は、図 5 の線グラフを見ると、初期扇動者 5 人から 80 人まではほぼ直線になっており、初期扇動者に対する流言を鎮静化させるのに必要とする最小初期鎮静者は初期扇動者数の 1 次関数として求められると思われるので、初期扇動者に対する最小初期鎮静者数その関数を最小二乗法で求めた。

最小二乗法は、測定した数値が、1 次関数などの特定の関数で近似するとき、その測定値の組 $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$ と推定値の誤差の二乗和が最小になるように近似関数を求める方法である。

求めたい 1 次関数の式を、

$$y = ax + b$$

と仮定して、a を次式

$$a = \frac{n \sum_{k=1}^n x_k y_k - \sum_{k=1}^n x_k \sum_{k=1}^n y_k}{n \sum_{k=1}^n x_k^2 - \left(\sum_{k=1}^n x_k \right)^2}$$

で求め、b を次式

$$b = \frac{\sum_{k=1}^n x_k^2 \sum_{k=1}^n y_k - \sum_{k=1}^n x_k y_k \sum_{k=1}^n x_k}{n \sum_{k=1}^n x_k^2 - \left(\sum_{k=1}^n x_k \right)^2}$$

で求める。

上の a と b を求めるエクセルでの処理を図 7 に示した。

	A	B	C	D	E	F
1						
2						
3		初期扇動者数	初期鎮静者数			
4		x	y	xy	xx	
5		5	9	81	45	25
6		10	16	256	160	100
7		20	27	729	540	400
8		40	53	2809	2120	1600
9		80	99	9801	7920	6400
10	合計	155	204	13676	10785	8525
11						
12	データの個数	5				
13						
14	a	1.199193548	$= (B12 * E10 - C10 * B10) / (B12 * F10 - B10^2)$			
15	b	3.625	$= (F10 * C10 - E10 * B10) / (B12 * F10 - B10^2)$			
16						

図 8 Excel での a と b の計算

この計算によって

$$a=1.199193548$$

$$b=3.625$$

になり、一次関数の式は、

$$f(x) = 1.2x + 3.63$$

となった。ここで有効数字は 3 桁とした。

ここで求めた関数が実際に測定した結果を良く近似するかを確かめるため、初期扇動者数ごとの最小初期鎮静者のシミュレーションによる実測値と、求めた関数による最小初期鎮静者数の推定値を表 3 に並べて表示し、図 7 にその散布図を描画した。

初期扇動者数	最小初期鎮静者数	最小初期鎮静者を求める 初期扇動者数の関数による 最小初期鎮静者数		
X	Y	Y	誤差	
0		3.625		
5	9	9.620967742	0.620968	0.385601
10	16	15.61693548	0.383065	0.146738
20	27	27.60887097	0.608871	0.370724
40	53	51.59274194	1.407258	1.980375
80	99	99.56048387	0.560484	0.314142
160		195.4959677	標準偏 差	0.894089

表 4 初期鎮静者に対す最小初期鎮静者数の測定値と関数による推定値の比較

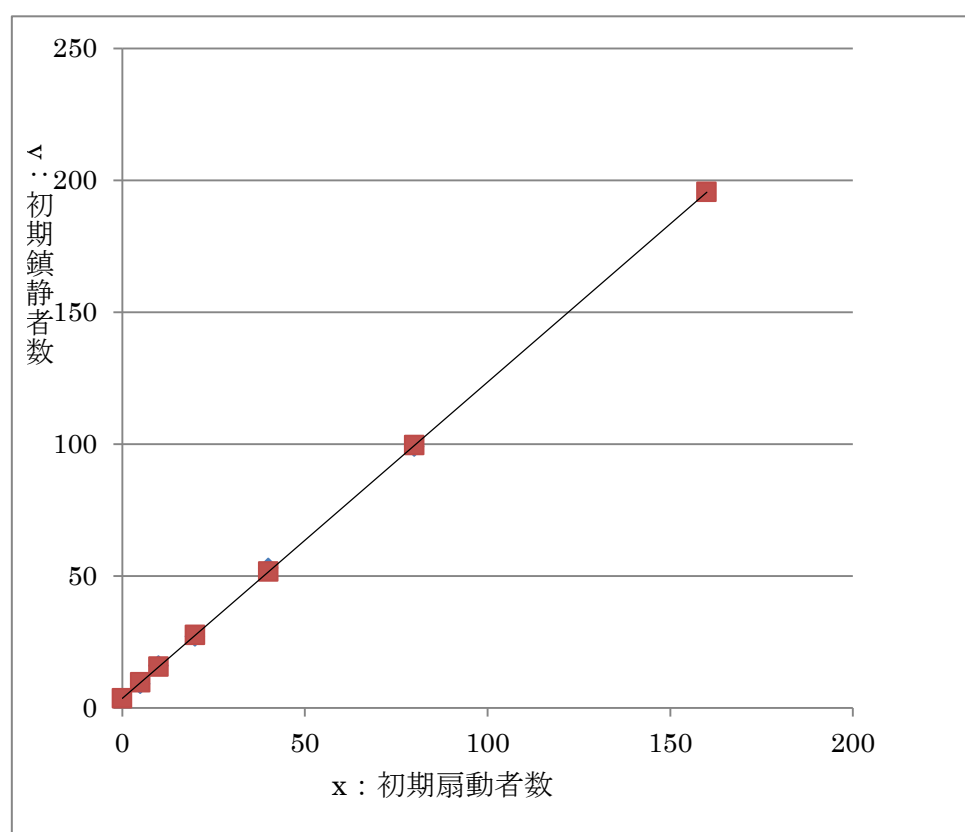


図 9 表 3 の初期鎮静者数と初期扇動者の関数による初期鎮静者数の散布図

図 8 から求められた関数による推定初期鎮静者数は、実際に測定した最小初期鎮静者数を良く近似し、推定最小初期鎮静者数は、初期扇動者数の増加に比例して直線的に増え、流言を鎮静化させる鎮静者が、扇動者数に対して僅か 1.2 倍の鎮静者がいれば鎮静化することが示された。

8.1.2 実験 2.1 に対する最小初期鎮静者数の予測関数

コネクション・ネットワークでの情報伝達を省いたシミュレーション・モデルによる実験を、実験 1 と同じ条件で行った。その結果得られた最小初期鎮静者数の予測式は、以下のとおりである。

$$g(x) = 1.22x + 3.88$$
$$(f(x) = 1.2x + 3.63)$$

予測式を見ると、コネクション・ネットワークによる情報伝達の無い方が、最少初期鎮静者数を若干多く設定しないと、鎮静化しない。図 9 にその違いを示す。

次に終了ステップ数の分散を比較すると、表 4 に示すように、コネクション・ネットワークでの情報伝達を省いたモデルによる実験のほうが、省いていないモデルによる実験と比べて分散が大きい。この分散の差は、人コネクション・ネットワークありでの情報伝達は、一度であったエージェントと継続的に情報伝達を行うが、コネクション・ネットワークでの情報伝達を省くと、情報伝達はエージェントの周りに他のエージェントに来た時のみ、情報伝達処理を行うので、エージェント同士がすべてランダムに情報伝達を行うので、分散の値が高くなったことが考察できる。

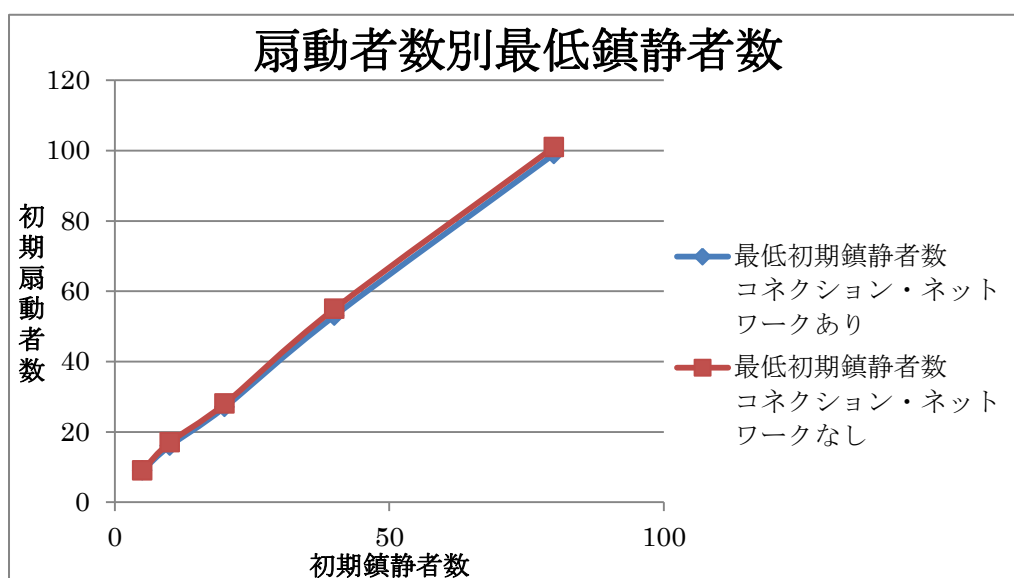


図 10 コネクション・ネットワークでの情報伝達の有無による、最小初期鎮静者数の変化

初期扇動者数	不偏分散	
	コネクション・ネットワークあり	コネクション・ネットワークなし
5	1160.444444	4519.666667
10	435.2888889	497.6888889
20	1370.666667	3580
40	2196.044444	2398.911111
80	586.8444444	1916.933333

表 5 コネクション・ネットワークでの情報伝達の有無による、終了ステップ数の分散の違い

8.2 鎮静化までの所要ステップ数の分布—実験 1.2 の考察

鎮静化までの所要ステップ数の分布を求めるため、同じ条件で実験を繰り返し、扇動者が 0 になるまでのステップ数の分散を求めた。結果を考察するために付録 4 の鎮静化するステップ数を昇順に並び替え、0 ステップ数から 1000 ステップ数までの結果を 50 ステップ単位でヒストグラム作成を行い、そのヒストグラムの散布図と実験で測定した平均と分散でのガンマ関数の散布図の近似を確かめた。

ガンマ関数は、以下の式で与えられる。

$$f(x) = x^{k-1} \frac{e^{-x/\theta}}{\Gamma(k) \theta^k} \quad \text{for } x > 0$$

この式とガンマ関数を計算するために母数 θ と k は、平均($E^{\wedge}(X)$)と分散($V^{\wedge}(X)$)で求めることができ、ガンマ分布の確率変数を X とするとき、

$$E(X) = k\theta,$$

$$V(X) = k\theta^2$$

となる。

実験 2 の結果である分散と平均から求めて、求めた θ と k をガンマ関数に代入して、データ区間に対する値の計算を行い、データ区間内の鎮静化回数とデータ区間に対するガンマ関数で求めた値を、データ区間内の鎮静化回数とデータ区間に対するガンマ関数で求めた値の確率分布を求めた。(表 5)(表 6)(図 10)

標本分散 $V^{\wedge}(X)$	23392.13331		
標本平均 $E^{\wedge}(X)$	277.6460887		
k: 形状母数	$E(X) = k \theta$	k	3.295439
θ : 尺度母数	$V(X) = k \theta^2$	θ	2346.078

表 6 実験 1.2 の標本分散と標本平均

データ区間	実験1.2の 確率分布	ガンマ分布の 確率分布
0	0	0
50	0	0.037075512
100	0	0.100541789
150	0.07	0.140869443
200	0.24	0.150615438
250	0.29	0.138862171
300	0.12	0.116574378
350	0.1	0.091735792
400	0.07	0.068852423
450	0.04	0.049842539
500	0	0.035066879
550	0	0.024108763
600	0.02	0.016262215
650	0.01	0.010795368
700	0	0.00706934
750	0.02	0.004575315
800	0	0.002931046
850	0	0.001860898
900	0	0.001172106
950	0.01	0.000733044
1000	0.01	0.000455541

表 7 エージェント数 913、扇動者数 5、鎮静者数 8 での 50 ステップごとの鎮静化回数の頻度の確率分布と $k=3.295$ 、 $\theta=2346.078$ でのガンマ分布の確率分布

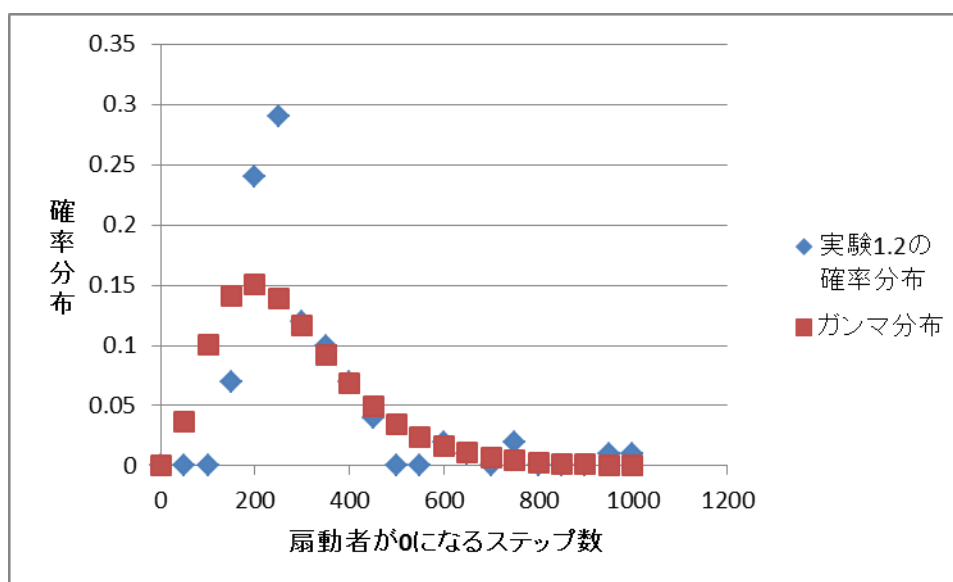


図 11 エージェント数 913、扇動者数 5、鎮静者数 8 での 50 ステップごとの鎮静化回数の頻度の確率分布と $k=3.295$ 、 $\theta=2346.078$ でのガンマ分布の確率分布

図 10 を見ると、データ区間の前半部分は、散布図の測定した頻度とガンマ分布での頻度の点がずれているが、データ区間で頻度が減少する 300 付近からガンマ分布に従うことが観察できる。

ガンマ分布は、モノや人の寿命の分布の予想に利用される。本研究のモデルは、扇動者の流言を扇動化させるが、扇動化は扇動者の数で扇動化の状態を定義できるので、扇動者が無くなるまでの時間を、流言の寿命とすることができる。

よって、扇動者数が 0 になるまでのステップ数が、ガンマ分布に従うということは、流言が鎮静化するまでの期間は、 $k=3.295$ 、 $\theta=2346.078$ のガンマ分布に従って流言が鎮静化するが、正規分布と違い、ガンマ分布は尾が長いので、鎮静化するまでステップ数が長くなることがある。この実験での図 11 のガンマ分布でも、300 ステップあたりから、尾を引いている。

鎮静化までのステップ数がガンマ分布に従うということは、鎮静化までの時間が長くなることがあるため、流言が鎮静化するまでの時間を予測することは難しいといえる。

以上の考察から、実験 1.1 では、初期扇動者数に対して、鎮静化させるために必要な初期鎮静者数を初期扇動者数の関数を $f(x)=1.20x+3.63$ として求めることができた。つまり、流言を鎮静化させるには、扇動者数に対して、鎮静者数を扇動者数に対して 1.20 倍の人を集めて、流言を鎮静化させる情報を発信してもらうことで、流言を鎮静化させることができる。

実験 1.2 では、鎮静者によって扇動者がなくなるまでのステップ数が、 $k=3.295$ 、 $\theta=2346.078$ のガンマ分布に近似することから。鎮静者によって流言を鎮静化作用とした場合に、流言は $k=3.295$ 、 $\theta=2346.078$ のガンマ分布に従って、流言が鎮静化することが明らかになった。

情報の伝達方法による相違は、最小初期鎮静者数に関しては、エージェント同士が隣接した時の情報伝達では、扇動者数に対して 1.20 倍と 3.63 人の鎮静者数が必要となるが、コネクション・ネットワークありの場合では、その比率が 1.22 倍と 3.88 人と異なり、コネクション・ネットワークによる情報伝達の無い方が、最少初期鎮静者数を若干多く設定しないと、鎮静化しない。一方、鎮静化に要するステップ数を比較すると、エージェント同士が隣接した時の伝達では、その分散がコネクション・ネットワークありの場合よりも格段に大きくなるので、人と人のコネクションによって鎮静化させる情報を発信するほうが、より安定して鎮静効果を期待できることを提言することができる。

第9章 おわりに

本修士論文の目的は、コミュニケーションによって情報が流布することで発生する流言の発生メカニズムとその影響を評価することであり、流言の結果引き起こされる風評被害への対策を提言することを狙いとして、流言は実際に調査することは難しいことから流言の実際の調査は行わずに、シミュレーションによって流言の流布のし方を調べる手法をとることとした。

すなわち、流言を発生させる扇動者と、流言は鎮静化させようとする鎮静者、どちらでもない一般人を仮定して、システム・シミュレーションの技法によって流言が流布する社会モデルをコンピュータ上で構築し、その構築したシミュレータによって初期扇動者数に対して、鎮静化させるために必要な初期鎮静者数を求め、さらに、鎮静化までのステップ数の分散を求めた。

本論文では、流言を扇動する扇動者と流言を鎮静化させようとする鎮静者、および扇動者と鎮静者の影響を受ける一般人を仮定したシミュレーション・モデルを作成して実験を行なった。

シミュレーション・モデルはマルチエージェント・シミュレーションとした。このモデルでは人をエージェントとしてシミュレーション空間内でランダムに移動させ、エージェント同士が接触することで流言が流布すると同時に、扇動者あるいは鎮静者と接触した一般人の間にコミュニケーションチャンネルを設定し、そのチャンネルを通してのより直接的な流言の流布を可能にしている。この情報を伝えるエージェントは、エージェント内のパラメータの組み合わせによってエージェントの振る舞いに変化を持たせ、他のエージェントに情報を伝えるモードを変化させるモデルを作成した。

流言が流布するモデルでは情報が伝わっていくモデルを作成し、そのモデルを基本モデルとして、流言が流布する要因を取り入れることで流言モデルを作成した。この基本モデルに、情報の強度、エージェント同士のコネクションとエージェントの種類として登場人物を加える。情報の強度は、エージェントが情報を受け取った時、その情報によって状態を変える、情報の度合を表すパラメータである。情報の強度は個々のエージェントが持ち、このパラメータはエージェントが情報を発信する際に、相手に与える影響の度合いを決定する。

情報の伝達は登場人物によってその伝達強度を変える。登場人物は、流言の強度を高める扇動者と流言の強度を弱める鎮静者、流言に影響される一般人を用意する。この登場人物の組み合わせによって、接触したエージェントからの影響を決定する。

エージェントが情報を受け取るときに、ぶつかった相手が扇動者であれば自エージェントの情報の強度を高め、鎮静者であれば自エージェントの情報の強度を弱めることで、エージェントが互いに接触したとき、閾値によって登場人物のタイプが変化する1度の接触による流言の伝播と、エージェントがコネクションを作り、コネクション網による伝播の2通りの流言伝播を再現した。実験では、この研究での目的である扇動者数に対する、鎮静聖化させるために必要な最小鎮静者数を求めるための実験1とシミュレーションの結果の分散を求めるための実験2を行った。

実験1は一般人を900人に固定して、この各扇動者に対する最小鎮静者数を求める実験を行った。この実験での最小鎮静者は同じ鎮静者数で10回繰り返し実験を行い、すべて鎮静化した鎮静者数を扇動者に対する最小鎮静者数として求めた。またこの実験では扇動数別の分散と扇動者数40人での最小鎮静者数とその分散を求めた。

最少鎮静者数を求めるために、実験の扇動者数と同じ数の鎮静者数から実験を行い、鎮静化しなければ鎮静者の数を増やしていくことで、鎮静化する扇動者数に対する最小鎮静者数を求めた。

実験1は、図5の線グラフを見ると、初期扇動者5人から80人までほぼ直線になっており、初期扇動者に対する流言を鎮静化させる初期鎮静者を求める初期扇動者は1次関数として求められると思われるので、初期扇動者に対する最小初期鎮静者数を最小二乗法によって近似した1次関数を求めることができた。

実験1.2では、結果の分散を求めるため、同じ条件で実験を繰り返し、扇動者が0になるまでのステップ数の分散を求め、結果は標本分散が8787.62になった。またこの実験では、0ステップ数から1000ステップ数までの結果を50ステップ単位でヒストグラム作成して、そのヒストグラムを散布図にした。この散布図に近似する関数を探したが、確率密度変数であるガンマ分布に近似しているように見えたので、実験で測定した、平均と分散から、ガンマ関数を用いて分布図を作成して、測定した結果とガンマ関数によって求めた値が近似しているか確かめた。データ区間の前半部分は、散布図の測定した頻度とガンマ分布での頻度の点がずれているが、データ区間で頻度が減少する300付近からガンマ分布に従うことが考察できた。

以上の考察から、実験1.1では、初期扇動者数に対して、鎮静化させるために必要な初期鎮静者数を初期扇動者数の関数を $f(x)=1.20x+3.63$ として求めることができた。つまり、流言を鎮静化させるには、扇動者数に対して、鎮静者数を扇動者数に対して1.2倍の人を集めて、流言を鎮静化させる情報を発信してもらうことで、流言を鎮静化させることができる。

実験 1.2 では、鎮静者によって扇動者がなくなるまでのステップ数が、 $k=3.33$ 、 $\theta=2346.08$ のガンマ分布に近似することから。鎮静者によって流言を鎮静化作用とした場合に、流言は $k=3.23$ 、 $\theta=2346.08$ のガンマ分布に従って、鎮静化するが、鎮静化までのステップ数がガンマ分布に従うということは、鎮静化までの時間が長くなることがあるため、流言が鎮静化するまでの時間を予測することは難しいことが明らかになった。

情報の伝達方法による相違は、最小初期鎮静者数に関しては、エージェント同士が隣接した時の情報伝達では、扇動者数に対して 1.20 倍と 3.63 人の鎮静者数が必要となるが、コネクション・ネットワークありの場合では、その比率が 1.22 倍と 3.88 人となり、コネクション・ネットワークによる情報伝達の無い方が、最少初期鎮静者数を若干多く設定しないと、鎮静化しない。

一方、鎮静化に要するステップ数を比較すると、エージェント同士が隣接した時の伝達では、その分散が人と人のコネクション・ネットワークありの場合よりも格段に大きくなるので、人と人のコネクションを使って鎮静化させる情報を発信するほうが、より安定して鎮静効果を期待できることを提言することができる。

この実験は単純なモデルである。このシミュレーション・モデルに、機能を加えて、様々なシーンでの流言の流布を再現することができる。

本研究を進めるにあたって、多くのケースでの情報流布を想定して、時間があれば、その想定したケースでのシーンをシミュレータに加えて実験を行いたかった。実験では、シミュレータでの登場人物の種類を増やして、実験を行う予定であった。登場人物は、扇動者であれば、有名人や強い政治家などの、強い影響力を持つ扇動者や鎮静者を設定して、その影響力の強い人物によって他の人物が、流言を扇動や鎮静させるなどの、エージェントのふるまいを変化させることで、影響力のある人が流言に与える影響を再現すること、などである。

シミュレーションの中で複数の情報を伝達させ、複数の情報を受け取ったエージェントが、その情報のパラメータの設定によって流言を発生や流言の強度を変化させ、情報や流言が複数ある場合の実験など、本研究で作成したシミュレータに機能を加えて、様々なケースでの流言流布を再現して、シミュレータでの実験結果を精緻化していくことが可能である。

また、現実での流言の流布と、この研究での流言の流布の整合性を検証することや実験のステップ数と現実の時間軸の関係など、検証できていないことがある。

私がお世話になった嘉悦大学大学院の、これから入学する後輩に、私の研究を引きつぎ、本研究で作成したシミュレータを精緻化と、本研究モデルと現実

での流言流布の整合性を検証していただきたい。

謝辞

本研究を進めるにあたり、終始丁寧にご指導をいただいた指導教授の古川康一教授と南憲一教授、滑川光裕教授、中村修教授に感謝いたします。

古川康一教授には、研究での実験と考察の方法の細部までご指導していただきました。ここで感謝させていただきます。

南憲一教授には、貴重な時間を割いて毎週行っていた私の研究指導に参加していただき、貴重なアドバイスを頂いたことを感謝します。

滑川光裕教授には、多大な協力と励ましをいただきました。

中村修教授には、研究の学内発表の際、発表が不慣れな私に的確なご指導をしていただきました。

古川康一教授、南憲一教授、滑川光裕教授、中村修教授に感謝の意を表します。ありがとうございました。

参考文献

- 1.共著:古川園智樹、石元龍太郎、小林慶太、笠井賢紀、赤松正教、井庭崇(2004)
「社会ネットワークの形成過程シミュレーション〜マルチエージェント・モデルによる表現と拡張〜」情報処理学会研究報告 2004-ICS 136 (14).8
- 2.共著:有澤誠、斉藤鉄也「モデルシミュレーション技法」 共立出版株式会社
1997.7
- 3.共著:森戸晋、相沢りえ子、貝原俊也 (1998) 「Visual SLAM によるシステム
シミュレーション」 構造計画研究所
- 4.西田豊明(2002) 「エージェント工学」 オーム社
- 5.船津衛 (2010) 「コミュニケーション・入門：心の中からインターネットまで」 有斐閣
- 6.山下元 (2010) 「ファジィ理論 基礎と応用」 共立出版
- 7.坂匂正敏 (1989) 「ファジィ理論の基礎と応用」 森下出版
- 8.道又爾 (2003) 「認知心理学―知のアーキテクチャを探る」 有斐閣アルマ
- 9.太田信夫(1991) 「認知心理学：理論とデータ」 誠信書房
- 10.共著:並河悠介、北栄輔、菲翟、侃沈(2007)
「行動ファイナンス理論に従うエージェントの市場取引への影響について」
情報処理学会誌: 数理モデル化と応用 Vol.48 No.SIG 6(TOM 17)
- 11.竹村和久(2006) 「リスク社会における判断と意思決定」 日本認知学会招待
論文 Cognitive Studies, 13(1), 17-31.
- 12.共著:山影進、服部正太 (2002) 「コンピュータのなかの人工社会―マルチ
エージェントシミュレーションモデルと複雑系」 構造計画研究所
- 13.共著:エプスタイン、J.M.、アクステル、R. 服部正太、木村香代子 (1999)
「人工社会―複雑系とマルチエージェント・シミュレーション」 構造計画研究
所
- 14.兼田敏之(2005) 「知的エージェントで見る社会 1、社会デザインのシミュ
レーション&ゲーミング」 共立出版
- 15.兼田敏之 (2010) 「人工社会の可能性 3artisoc で始める歩行者エージェント
シミュレーション原理・方法論から安全・賑わい空間のデザイン・マネジメン
トまで」 構造計画研究所
- 16.上田完次 (2007) 「創発とマルチエージェントシステム」 培風館
- 17.共著:湯田聡夫、小野直亮、藤原義久 (2006) 「ソーシャル・ネットワーキ
ング・サービスにおける人的ネットワークの構造(事例分析,<特集>ネットワー

ク生態学～生命現象から社会文化現象の新しいパースペクティブ～」情報処理学会論文誌 47(3) 865-874

18.共著:松尾豊、篠田孝祐、中島秀行 (2006) 「中心性に着目した合理エージェントのネットワーク形成」 人工知能学会論文誌 21 巻 1 号 N

19.河内一樹 (2008) 「流言の空間伝播モデルの進行波解数理解析研究所講究録」第 1597 巻

20.森田浩 (2010) 「図解入門よくわかる最新実験計画法の基本と仕組み―実験の効率化とデータ解析の全手法を解説」 秀和システム

21.共著:星野直人、関庸一 (2007) 「Excel で学ぶ理論と技術実験計画法入門」ソフトバンククリエイティブ

付録 1 実験 1.1,初期扇動者別の最小初期鎮静者の推定実験における終了ステップ数と不偏分散

一般人 900 人

初期鎮静者 9 人

初期扇動者 5 人

	終了ステップ数	平方和	
	109	64	
	143	676	
	183	4356	
	144	729	
	91	676	
	104	169	
	84	1089	
	104	169	
	71	2116	
	137	400	
平均	117	10444	不偏分散 1160.444

初期扇動者 10 人

初期鎮静者 16 人

	終了ステップ数	平方和	
	113	368.64	
	103	84.64	
	100	38.44	
	115	449.44	
	80	190.44	
	73	432.64	
	123	852.64	
	58	1281.64	
	79	219.04	
	94	0.04	
平均	93.8	3917.6	不偏分散 435.2889

初期扇動者 20 人

初期鎮静者 27 人

終了ステップ数	平方和
---------	-----

	98	529	
	122	1	
	120	1	
	70	2601	
	149	784	
	109	144	
	126	25	
	70	2601	
	162	1681	
	184	3969	
平均	121	12336	不偏分散 1370.667

初期扇動者 40 人

初期鎮静者 53 人

	終了ステップ数	平方和	
	121	31.36	
	130	11.56	
	190	4019.56	
	98	817.96	
	148	457.96	
	86	1648.36	
	113	184.96	
	111	243.36	
	107	384.16	
	162	1253.16	
平均	126.6	9052.4	不偏分散 1005.822

初期扇動者 80 人

初期鎮静者 99 人

終了ステップ数	平方和
133	948.64
62	1616.04
74	795.24
117	219.04
126	566.44
116	190.44

	76	686.44	
	100	4.84	
	118	249.64	
	100	4.84	
平均	102.2	5281.6	不偏分散 586.8444

付録 2 実験 1.2, 初期扇動者 40 人、初期鎮静者 57 人での繰り返し実験による鎮静化ステップ数の分布の推定実験

初期扇動者 40

人

初期鎮静者 57

人

	ステップ 数	平方和
1	112	8.41
2	102	166.41
3	84	954.81
4	108	47.61
5	78	1361.61
6	62	2798.41
7	136	445.21
8	93	479.61
9	129	198.81
10	80	1218.01
11	84	954.81
12	83	1017.61
13	130	228.01
14	211	9235.21
15	132	292.41
16	102	166.41
17	80	1218.01
18	67	2294.41
19	123	65.61
20	75	1592.01

21	98	285.61
22	115	0.01
23	82	1082.41
24	102	166.41
25	77	1436.41
26	96	357.21
27	92	524.41
28	73	1755.61
29	105	98.01
30	77	1436.41
31	109	34.81
32	93	479.61
33	224	11902.81
34	99	252.81
35	159	1944.81
36	122	50.41
37	141	681.21
38	159	1944.81
39	96	357.21
40	133	327.61
41	129	198.81
42	101	193.21
43	61	2905.21
44	144	846.81
45	148	1095.61
46	123	65.61
47	68	2199.61
48	89	670.81
49	83	1017.61
50	86	835.21
51	72	1840.41
52	212	9428.41
53	128	171.61
54	122	50.41

55	162	2218.41
56	94	436.81
57	125	102.01
58	72	1840.41
59	135	404.01
60	84	954.81
61	123	65.61
62	208	8667.61
63	107	62.41
64	82	1082.41
65	87	778.41
66	147	1030.41
67	102	166.41
68	70	2016.01
69	110	24.01
70	149	1162.81
71	163	2313.61
72	168	2819.61
73	275	25632.01
74	178	3981.61
75	112	8.41
76	95	396.01
77	137	488.41
78	229	13018.81
79	80	1218.01
80	274	25312.81
81	112	8.41
82	94	436.81
83	95	396.01
84	106	79.21
85	110	24.01
86	75	1592.01
87	91	571.21
88	97	320.41

89	115	0.01
90	110	24.01
91	143	789.61
92	81	1149.21
93	130	228.01
94	75	1592.01
95	131	259.21
96	57	3352.41
97	102	166.41
98	112	8.41
99	91	571.21
100	106	79.21
平均	114.9	179231
	分散	1810.414

付録 3 実験 1.2,表 4 エージェント数 1000 初期扇動者 5 人、初期鎮静者 8 人での、繰り返し実験による鎮静者数が 0 になるまでのステップ数の推定

番号	最大 扇動者数	最大扇動者数 時の ステップ数	終了時の 扇動者数	終了時の ステップ数	扇動者数が 0 になる ステップ数の予測
1	401	66	187	202	320.8411
2	318	43	149	150	244.3373
3	388	47	184	180	299.9608
4	339	39	155	142	228.7663
5	335	47	157	140	222.0281
6	357	44	169	150	245.2872
7	288	46	133	130	202.0774
8	353	39	164	133	214.5661
9	312	39	142	119	185.8235
10	305	47	142	143	226.6319
11	323	48	152	144	229.3333

12	393	50	190	158	259.0837
13	311	43	143	132	207.756
14	400	47	197	246	439.1182
15	177	42	71	85	113.8019
16	411	50	197	217	370.7336
17	270	42	121	100	147.1007
18	369	49	174	183	302.5692
19	364	51	174	176	290.4737
20	266	43	126	134	215.9
21	309	43	144	109	166.6
22	385	43	182	209	357.8276
23	322	48	150	155	248.314
24	233	40	106	83	118.8898
25	322	45	152	149	241.9882
26	368	46	151	151	224.0645
27	302	40	141	117	184.4348
28	303	48	143	140	222.225
29	307	45	146	177	296.7019
30	275	43	126	123	190.651
31	430	49	250	300	648.6111
32	357	48	169	138	218.9043
33	359	43	173	180	307.4247
34	379	47	179	150	242.185
35	238	39	111	113	177.6772
36	286	45	128	104	151.7975
37	311	43	143	125	194.7976
38	286	43	131	119	183.2323
39	338	45	160	149	242.4831
40	283	52	128	117	170.6774
41	389	46	187	196	334.8614
42	327	43	148	124	190.9721
43	396	56	187	224	374.3158
44	286	41	131	103	155.4
45	193	44	88	101	148.7714

46	211	44	91	96	135.4333
47	262	54	120	128	190.5352
48	410	53	197	255	441.8263
49	368	43	175	153	252.7409
50	247	47	118	134	213.5814
51	376	50	181	180	300.6667
52	193	51	137	126	309.4821
53	431	77	320	300	942.8829
54	368	48	175	169	278.715
55	334	43	151	144	227.3388
56	213	36	95	101	153.3305
57	350	48	165	126	195.5676
58	398	57	189	216	359.7847
59	369	56	170	165	258.1156
60	235	44	110	97	143.64
61	330	50	154	140	218.75
62	360	49	169	135	211.0942
63	391	53	187	217	367.3333
64	382	48	178	184	302.6667
65	275	42	129	118	185.1507
66	317	52	148	138	213.3136
67	220	41	97	90	128.6423
68	365	53	174	153	244.0995
69	326	43	151	124	193.8914
70	259	44	117	127	195.3873
71	316	42	150	128	205.7108
72	273	38	128	134	218.7448
73	378	44	175	150	241.3793
74	343	43	164	157	261.4469
75	405	52	197	201	342.1202
76	330	46	157	120	187.1561
77	395	52	190	227	389.1951
78	412	45	200	241	425.9057
79	332	56	157	153	240.0229

80	309	43	132	108	156.4746
81	358	45	167	155	251.178
82	263	43	118	108	160.8966
83	330	54	148	128	188.1758
84	299	45	139	121	187.025
85	359	47	170	174	288.2328
86	380	44	183	216	375.7766
87	356	41	169	143	235.1818
88	368	42	179	181	312.6455
89	440	54	324	300	987.1034
90	279	45	131	109	165.6486
91	443	47	281	300	738.8457
92	315	50	145	118	176
93	420	46	220	300	579.4
94	423	73	239	300	594.8533
95	381	43	182	153	253.603
96	302	42	137	142	225.0303
97	369	40	171	178	297.1818
98	443	56	276	300	703.2575
99	358	50	170	194	324.2128
100	411	61	195	254	428.2361

付録 4 実験 1.2, エージェント数 1000 初期扇動者 5 人、初期鎮静者 8 人での、繰り返し実験の平均と不偏分散

番号	扇動者数が 0 になる ステップ数	平方和
1	113.8018868	26844.92
2	118.8897638	25203.57
3	128.6422764	22202.14
4	135.4333333	20224.47
5	143.64	17957.63

6	147.1006711	17042.11
7	148.7714286	16608.68
8	151.7974684	15837.88
9	153.3305085	15454.36
10	155.4	14944.11
11	156.4745763	14682.54
12	160.8965517	13630.45
13	165.6486486	12543.43
14	166.6	12331.23
15	170.6774194	11442.3
16	176	10331.93
17	177.6771654	9993.786
18	183.2322581	8913.971
19	184.4347826	8688.348
20	185.1506849	8555.4
21	185.8235294	8431.382
22	187.025	8212.182
23	187.1560694	8188.444
24	188.1758242	8004.928
25	190.5352113	7588.305
26	190.6510067	7568.144
27	190.972067	7512.386
28	193.8914286	7014.843
29	194.797619	6863.869
30	195.3873239	6766.504
31	195.5675676	6736.884
32	202.0774194	5710.624
33	205.7108434	5174.68
34	207.7559524	4884.631
35	211.0942408	4429.148
36	213.3136095	4138.668
37	213.5813953	4104.285
38	214.5661376	3979.08
39	215.9	3812.579

40	218.7448276	3469.359
41	218.75	3468.749
42	218.9042553	3450.603
43	222.0280899	3093.362
44	222.225	3071.497
45	224.0645161	2870.985
46	225.030303	2768.421
47	226.6319018	2602.447
48	227.3387978	2530.824
49	228.7663043	2389.233
50	229.3333333	2334.122
51	235.1818182	1803.214
52	240.0228571	1415.508
53	241.3793103	1315.279
54	241.9882353	1271.483
55	242.185	1257.489
56	242.4831461	1236.433
57	244.0994764	1125.375
58	244.3372781	1109.477
59	245.287234	1047.095
60	248.3139535	860.3742
61	251.1780105	700.5592
62	252.7409326	620.2668
63	253.6030151	578.0694
64	258.1155779	381.4409
65	259.0837438	344.5606
66	261.4469274	262.4128
67	278.7150259	1.142627
68	288.2328042	112.0785
69	290.4736842	164.5472
70	296.7018634	363.1225
71	297.1818182	381.6447
72	299.9607843	497.9456
73	300.6666667	529.947

74	302.5692308	621.163
75	302.6666667	626.0293
76	307.4247312	886.7675
77	309.4821429	1013.534
78	312.6455026	1224.959
79	320.8411215	1865.811
80	324.212766	2168.455
81	334.8613861	3273.59
82	342.1201923	4156.91
83	357.8275862	6429.073
84	359.784689	6746.75
85	367.3333333	8043.802
86	370.7336449	8665.293
87	374.3157895	9345.031
88	375.7766497	9629.607
89	389.195122	12443.19
90	425.9056604	21980.9
91	428.2361111	22677.35
92	439.1182266	26073.25
93	441.8262911	26955.14
94	579.4	91055.42
95	594.8532609	100620.4
96	648.6111111	137615
97	703.257485	181145.1
98	738.845679	212705.1
99	942.8828829	442540
100	987.1034483	503329.7

付録 5 実験 2.1, エージェント同士が隣接した時の情報伝達での、初期扇動者別の最小初期鎮静者の推定実験における終了ステップ数と不偏分散

一般人 900 人

初期扇動者 5 人

初期鎮静者 9 人	終了ステップ 数	平方和		
	164	2209		
	109	64		
	129	144		
	170	2809		
	269	23104		
	145	784		
	133	256		
	202	7225		
	166	2401		
	158	1681		
	平均 164.5	40677	不偏分 散	1160.444

初期扇動者 10 人				
所金鎮静者 14 人	終了ステップ 数	平方和		
	88	33.64		
	102	67.24		
	88	33.64		
	137	1866.24		
	112	331.24		
	98	17.64		
	135	1697.44		
	106	148.84		
	78	249.64		
	88	33.64		
	平均 103.2	4479.2	不偏分 散	435.2889

初期扇動者 20 人			
初期鎮静者 28 人	終了ステップ 数	平方和	

	172	2601	
	117	16	
	265	20736	
	111	100	
	115	36	
	164	1849	
	75	2116	
	112	81	
	183	3844	
	92	841	
平均	140.6	32220	不偏分散 1370.667

初期扇動者 40 人

初期鎮静者 53
人

終了ステップ 数	平方和	
106	424.36	
127	0.16	
209	6789.76	
74	2766.76	
95	998.56	
85	1730.56	
101	655.36	
216	7992.36	
131	19.36	
113	184.96	
平均	125.7	21562.2
		不偏分散 1005.822

初期扇動者 80 人

初期鎮静者 101
人

終了ステップ 数	平方和
99	10.24
106	14.44

	225	15079.84	
	117	219.04	
	71	973.44	
	116	190.44	
	85	295.84	
	111	77.44	
	107	23.04	
	83	368.64	
平均	112	17252.4	不偏分散 586.8444

付録6 Artisoc によるシミュレーション・プログラム

```
//=====
===/
//                                     /
// artisoc                             /
//                                     /
//=====
===/
```

Version = 3.0;

```
//-----
// Component Tree
//-----
UNIVERSE {
    Space TwoDimensionalSpace(0,0),Square_2D(100, 100, Loop, North, 1, !Map_3D){
        AgtType Walker(0, 0)[913] {
            Dim ID(0, 0) As Integer = (0);
            Dim X(0, 0) As Double = (0.0);
```

```

    Dim Y(0, 0) As Double = (0.0);
    Dim Layer(0, 0) As Integer = (0);
    Dim Direction(0, 0) As Double = (0.0);
    Dim WalkerColor(0, 0) As Integer = (0);
    Dim WithInfo(0, 0) As Integer = (0);
    Dim WithInfo2(0, 0) As Integer = (0);
    Dim WithInfo3(0, 0) As Integer = (0);
    Dim Count(0, 0) As Integer = (0);
    Dim Ty(0, 0) As Double = (0.0);
    Dim Jotai(0, 0) As Double = (0.0); //流言信念強度
    Dim Color(0, 0) As Integer = (0);
    Dim Catch(0, 0) As Double = (0.0);
    Dim Hai(0, 0) <1> [913] As Integer = (913*0); //エージェントの組み合わせ
}
}

Dim WalkerCount(0, 0) As Integer = (0);
Dim StaggerLevel(0, 0) As Integer = (0);
Dim WalkersWithInfo(0, 0) As Integer = (0);
Dim WalkersWithInfoA(0, 0) As Integer = (0);
Dim WalkersWithInfo2(0, 0) As Integer = (0);
Dim WalkersWithInfo2A(0, 0) As Integer = (0);
Dim WalkersWithInfo3(0, 0) As Integer = (0);
Dim WalkersWithInfo3A(0, 0) As Integer = (0);
Dim ManType(0, 0) As Integer = (0);
Dim TY(0, 0) As Double = (0.0);
Dim Jotai(0, 0) As Double = (0.0);
Dim Color(0, 0) As Integer = (0);
Dim Day(0, 0) As Integer = (0);
Dim st(0, 0) As Integer = (0);
Dim st1(0, 0) As Integer = (0);
Dim e(0, 0) As Integer = (0);
Dim IdoHA(0, 0) <1> [913] As Integer = (913*0);
Dim IdoHB(0, 0) <1> [913] As Integer = (913*0);
Dim IdoHkaiA(0, 0) As Integer = (0);
Dim IdoHeikinA(0, 0) As Integer = (0);

```

```

    Dim IdoHeikinA1(0, 0) As Integer = (0);
    Dim IdoHkaiB(0, 0) As Integer = (0);
    Dim IdoHeikinB(0, 0) As Integer = (0);
    Dim IdoHeikinB1(0, 0) As Integer = (0);
    Dim sendosha(0, 0) As Integer = (0);
    Dim thinseisha(0, 0) As Integer = (0);
    Dim kyodo(0, 0) As Double = (0.0);
    Dim hkai(0, 0) As Integer = (0);
    Dim SMax(0, 0) As Double = (0.0);
    Dim TMax(0, 0) As Double = (0.0);
    Dim SStep(0, 0) As Double = (0.0);
    Dim TStep(0, 0) As Double = (0.0);
}

```

```

//-----
// Agent Variable Initial Data
//-----
Initial_Value {
    UNIVERSE.STAGGERLEVEL = 1(0);
    UNIVERSE.WALKERSWITHINFO = 1(0);
}

```

```

//-----
// Agent Rule
//-----
#begin_rule UNIVERSE
Univ_Init{

```

```

    Dim i As Integer
    Dim j As Integer
    Dim obj As Agt
    Dim x As Double
    Dim myAgtSet As AgtSet

```

```

For i = 0 To Universe.WalkerCount - 1
    obj = CreateAgt(Universe.TwoDimensionalSpace.Walker)
Next i

MakeAgtSet(myAgtSet, Universe.TwoDimensionalSpace.Walker)
RandomPutAgtSetCell(myAgtSet, False)

Universe.sendosha = 5 //初期扇動者数
x = 3
Universe.thinseisha = Universe.sendosha + x //初期鎮静者数
Universe.kyodo = 1 //流言伝達強度
Universe.hkai = 0

Universe.SMax = 0
Universe.TMax = 0

Universe.WalkersWithInfoA = 0
Universe.WalkersWithInfo2A = 0
Universe.WalkersWithInfo3A = 0

Universe.WalkersWithInfo = Universe.sendosha
Universe.WalkersWithInfo2 = Universe.thinseisha
Universe.WalkersWithInfo3 = Universe.WalkerCount - Universe.sendosha -
Universe.thinseisha

For j= 0 to 9
    Universe.IdoHA(j) = 10
    Universe.IdoHB(j) = 10
Next j

```



```
}
```

```
Univ_Step_Begin{
```

```
}
```

```
Univ_Step_End{
```

```
    Dim col As AgtSet
```

```
    Dim obj As Agt
```

```
    Dim IH As integer
```

```
    Dim IdoGen As integer
```

```
    Dim IdoA As integer
```

```
    Dim IdoB As Integer
```

```
    Dim sp As integer
```

```
    Dim MaShikii Double
```

```
    Dim MiShikii Double
```

```
    Universe.WalkersWithInfo = 0
```

```
    Universe.WalkersWithInfo2 = 0
```

```
    Universe.WalkersWithInfo3 = 0
```

```
    MakeAgtSet(col, Universe.TwoDimensionalSpace.Walker)
```

```
    MaShikii = 3
```

```
    MiShikii = -3
```

```
    For Each obj In col
```

```
        If obj.Jotai > MaShikii Then
```

```
            obj.Ty = 1
```

```
            obj.WalkerColor = COLOR_RED
```

```
        Else
```

```
            If obj.Jotai < MiShikii Then
```

```
                obj.Ty =2
```

```
                obj.WalkerColor = COLOR_BLUE
```

```
            Else
```

```

obj.Ty= 3
    obj.WalkerColor = COLOR_GREEN
End If
End If

If obj.WalkerColor == COLOR_BLUE Then //鎮静者
    Universe.WalkersWithInfo2 = Universe.WalkersWithInfo2 +
1
    End if
If obj.WalkerColor == COLOR_RED Then //扇動者
    Universe.WalkersWithInfo = Universe.WalkersWithInfo + 1
End If
If obj.WalkerColor == COLOR_GREEN Then //一般人
    Universe.WalkersWithInfo3 = Universe.WalkersWithInfo3 +
1
    End If
Next obj

If Universe.WalkersWithInfo == Universe.WalkersWithInfoA Then
If Universe.WalkersWithInfo2 == Universe.WalkersWithInfo2A Then
If Universe.WalkersWithInfo3 == Universe.WalkersWithInfo3A Then
    Universe.st1 = Universe.st1 + 1
    If Universe.st1 == 10 Then

        ExitSimulationMsg( "登場人物の割合が変わらなくなるまでのス
テップ数 " & GetCountStep())
        End If
        End If
        End If
        End If
        Universe.WalkersWithInfoA = Universe.WalkersWithInfo

If Universe.WalkersWithInfo == CountAgtSet(col) Then
    ExitSimulationMsg ( "すべてのエージェントが扇動者になるまでの

```

```

ステップ数 " & GetCountStep())
    End If
    If Universe.WalkersWithInfo2 == CountAgtSet(col) Then
        ExitSimulationMsg( "すべてのエージェントが鎮静者になるまでの
ステップ数 " & GetCountStep())
    End If

    If Universe.WalkersWithInfo3 == CountAgtSet(col) Then
        ExitSimulationMsg ( "すべてのエージェントが一般人になるまでの
ステップ数 " & GetCountStep())
    End If

    PrintLn ("ステップ数" &GetCountStep())

    If Universe.WalkersWithInfo == 0 Then
        ExitSimulationMsg ( "扇動者が消滅するまでのステップ数 " &
GetCountStep())
    End If

    If Universe.WalkersWithInfo2 == 0 Then
        ExitSimulationMsg ( "鎮静者が消滅するまでのステップ数 " &
GetCountStep())
    End If

    If Universe.WalkersWithInfo3 == 0 Then
        PrintLn ( "一般人が消滅しました。 " & GetCountStep())

    End If

    If GetCountStep() > -1 Then //現在の鎮静者数を数える。
        IH = 0
        sp = GetCountStep()

```

IH = sp mod 10 //IH の変数を作る

Universe.IdoHA(IH) = Universe.WalkersWithInfo

Universe.IdoHB(IH) = Universe.WalkersWithInfo2

IdoGen = 0

IdoA = 0

IdoB = 0

For IdoGen to 9 // 10 ステップ前から現在までの扇動者数を数える。

IdoB = IdoB + Universe.IdoHB(IdoGen)

Next IdoGen

Universe.IdoHeikinA = IdoA / 10

Universe.IdoHeikinB = IdoB / 10

If Universe.IdoHeikinA > Universe.SMax Then //扇動者数の過去最大
数を求める

Universe.SMax = Universe.IdoHeikinA

Universe.SStep = GetCountStep()

End If

If Universe.IdoHeikinB > Universe.TMax Then //鎮静者数の過去最大
数を求める

Universe.TMax = Universe.IdoHeikinB

Universe.TStep = GetCountStep()

End If

If Universe.IdoHeikinA < Universe.IdoheikinA1 Then

Universe.IdoHKaiA = Universe.IdoHKaiA + 1

If Universe.IdoHkaiA > 0 Then

If Universe.IdoHkaiA > 4 Then

println("扇動者数の移動平均が 5 ステップ以上連続で減少しました。

" & GetCountStep())

End If

```

        If Universe.SMax > (Universe.IdoHeikinA*2) Then
            println("扇動者数" & Universe.WalkersWithInfo)
            println("最大扇動者数 " & Universe.SMax)
            println("最大扇動者数でのステップ数 " & Universe.SStep)
            ExitSimulationMsg ("扇動者数が減少しており、扇動者数がその
最大数の半分なるまでのステップ数 " & GetCountStep())
        End If
    End If
else
    Universe.IdoHkaiA = 0
End If
Universe.IdoHeikinA1 = Universe.IdoHeikinA

If Universe.IdoHeikinB < Universe.IdoheikinB1 Then
    Universe.IdoHKaiB = Universe.IdoHKaiB + 1
    If Universe.IdoHkaiB > 0 Then
        If Universe.IdoHkaiB > 4 Then
            println("鎮静者数の移動平均が 5 ステップ連続で減少するまでのス
テップ数。 " & GetCountStep())
        End If
        If Universe.TMax > (Universe.IdoHeikinB*2) Then
            println("鎮静者数" & Universe.WalkersWithInfo2)
            println("最大鎮静者数 " & Universe.TMax)
            println("最大鎮静者数でのステップ数 " & Universe.TStep)
            ExitSimulationMsg ("鎮静者数が減少しており、鎮静者数がその
最大数の半分なるまでのステップ数 " & GetCountStep())
        End If
    End If
else
    Universe.IdoHkaiB = 0
End If
Universe.IdoHeikinB1 = Universe.IdoHeikinB
End If

```

```

If GetCountStep() == 300 Then

```

```

        println("最大の鎮静者数" & Universe.TMax)
            println("最大鎮静者数でのステップ数 " & Universe.TStep)
        println("最大の扇動者数" & Universe.SMax)
            println("最大扇動者数でのステップ数 " & Universe.SStep)
        println("扇動者数" & Universe.WalkersWithInfo)
        println("鎮静者数" & Universe.WalkersWithInfo2)
            ExitSimulationMsg ("300 ステップです。シミュレーションを終了します。
")
        End If
    }

Univ_Finish{

}

#end_rule UNIVERSE

```

```

//-----
// Agent Rule
//-----

#begin_rule UNIVERSE.TWODIMENSIONALSPACE.WALKER
Agt_Init{
    Dim obj As Agt
    Dim Shoki As AgtSet
    Dim a as integer
    MakeAgtSet(Shoki, Universe.TwoDimensionalSpace.Walker)
        My.Jotai = 0
        My.Ty = 3 //MyTy=3 は一般人。
        My.WalkerColor = COLOR_GREEN
        If My.ID < Universe.sendosha Then
            My.Ty = 1 //MyTy=1 は扇動者。
        My.Jotai = 3.2
    End If
End If
}

```

```

        My.WalkerColor = COLOR_RED
    Else
        If My.ID < Universe.sendosha + Universe.thinseisha Then
            My.Ty = 2 //MyTy=2 は鎮静者。
            My.Jotai= -3.2
            My.WalkerColor = COLOR_BLUE
        End If
    End If

    Turn(360*Rnd())
}

Agt_Step{

    Dim SurroundingAgents As AgtSet
    Dim obj As Agt
    Dim i As Integer
    Dim j As Integer
    Dim k As Integer
    Dim n As Integer
    Dim m As Integer
    Dim a As Double
    Dim b As Double
    Dim c As Integer
    Dim d As Integer
    Dim e As Integer
    Dim col1 As AgtSet
    Dim z As Integer
    Dim kae As AgtSet
    Dim hind As Integer

    Turn(Universe.StaggerLevel*(Rnd() - 0.5))
    Forward(1.0)

    MakeAllAgtSetAroundOwnCell(SurroundingAgents, 1, False)

```

For Each obj In SurroundingAgents

a=0

i=obj.ID

j=My.ID

If obj <> My Then

// エージェント同士が隣接した時の情報伝達

If My.Ty == 3 Then

If obj.Ty == 3 Then

My.hai(i)= 9

End If

If obj.Ty == 1 Then

a = a + Universe.kyodo //流言信念強度を加減算

My.hai(i)= 7

End if

If obj.Ty == 2 Then

a = - Universe.kyodo //流言信念強度を加減算

My.hai(i) = 8

End If

If obj.Ty == 10 Then

a = a + Universe.kyodo //流言信念強度を加減算

My.hai(i) = 7

End If

If obj.Ty == 11 Then

a = a - Universe.kyodo //流言信念強度を加減算

My.hai(i) = 8

End If

End If

If My.Ty == 1 Then

If obj.Ty ==1 Then

a = a +(Universe.kyodo*0.65)

My.hai(i)= 1

End If

If obj.Ty == 2 Then


```

        a = a -(Universe.kyodo*0.65)
        My.hai(i)=2
    End If
    If obj.Ty== 3 Then
        My.hai(i) = 3
    End If
        If obj.Ty == 10 Then
            a = a + Universe.kyodo //流言信念強度を加減算
            My.hai(i) = 1
        End If
        If obj.Ty == 11 Then
            a = a - Universe.kyodo //流言信念強度を加減算
            My.hai(i) = 2
        End If
    End If

If My.Ty == 2 Then
    If obj.Ty == 1 Then
        a = a +(Universe.kyodo*0.65)
        My.hai(i)= 4
    End If
    If obj.Ty==2 Then
        a = a -(Universe.kyodo*0.65)
        My.hai(i)= 5
    End if
    If obj.Ty == 3 Then
        My.hai(i)= 6
    End If
        If obj.Ty == 10 Then
            a = a + Universe.kyodo //流言信念強度を加減算
            My.hai(i) = 5
        End If
        If obj.Ty == 11 Then
            a = a - Universe.kyodo //流言信念強度を加減算
            My.hai(i) = 6

```

```

        End If
    End If
End If
My.Catch = My.Catch + a
next obj

```

// エージェント同士が隣接した時の情報伝達で、登場人物の組み合わせが変化したので、現在の組み合わせに直す。

```

MakeAgtSet(kae, Universe.TwoDimensionalSpace.Walker)

```

```

For Each obj in kae
    z = obj.id
    If obj <> My Then
        If obj.hai(z) > 0 Then

```

```

            If My.Ty == 1 Then
                If obj.Ty == 1 Then
                    my.hai(z) = 1
                End If
                If obj.Ty == 2 Then
                    my.hai(z) = 2
                End If
                If obj.Ty == 3 Then
                    my.hai(z) = 3
                End If
            End If

```

```

        If My.Ty == 2 Then
            If obj.Ty == 1 Then
                my.hai(z) = 4
            End If
            If obj.Ty == 2 Then
                my.hai(z) = 5
            End If
            If obj.Ty == 3 Then
                my.hai(z) = 6
            End If
        End If
    End If
End For

```

```

    End If
End If

If My.Ty == 3 Then
    If obj.Ty == 1 Then
        my.hai(z) = 7
    End If
    If obj.Ty == 2 Then
        my.hai(z) = 8
    End If
    If obj.Ty == 3 Then
        my.hai(z) = 9
    End If
End If
End If
End If
next obj

//コネクション・ネットワークでの伝達処理
MakeAgtSet(coll, Universe.TwoDimensionalSpace.Walker)
    Universe.e = 0 //いない
For Each obj in coll
    b = 0
    k = obj.Id
    for hind = 0 to Universe.hkai
        If obj.hai(k) == 1 Then
            b = b + (Universe.kyodo*0.65)*0.65
        End If
        If obj.hai(k) == 2 Then
            b = b - (Universe.kyodo*0.65)*0.65
        End If
        If obj.hai(k) == 3 Then
        End If
        If obj.hai(k) == 4 Then
            b = b + (Universe.kyodo*0.65)*0.65

```

```

End If
If obj.hai(k) == 5 Then
    b = b - (Universe.kyodo*0.65)*0.65
End If
If obj.hai(k) == 6 Then
End If
If obj.hai(k) == 7 Then
    b = b + (Universe.kyodo*0.65)
End If
If obj.hai(k) == 8 Then
    b = b - (Universe.kyodo*0.65)
End If
If obj.hai(k) == 9 Then
End If

If obj.hai(k) == 10 Then
    b = b + (Universe.kyodo*0.65)
End If
If obj.hai(k) == 11 Then
    b = b - (Universe.kyodo*0.65)
End If
Next hind
//エージェント同士が隣接した時の情報伝達で加算された変数 Catch とコネクション・ネットワークで加算された b を加算する。
obj.jotai = obj.jotai + obj.Catch + b
Next obj
Universe.e = 0
#end_rule UNIVERSE.TWODIMENSIONALSPACE.WALKER

```

```

//-----
// Simulation Definition Division
//-----

```

```

Single_Simulation {
    Step_Max = 0;
    Time_Max = 0;
    End_Condition = "";
    Exec_Wait = 1;
    Exec_Order = Random;
    Exec_Order_Random_First_Step_Only = No;
    Random_Seed = 0;
    Redraw_Timing = Step;
    Redraw_Interval = 1;
    Garbage_Interval = 0;
}

//-----
// Simulation Loop Definition Division
//-----
Repeat_Simulation {
    Value_Change = None;
    Initial_Value_Change_None {
        Repeat_Max = 1;
    }
}

//-----
// 2 Dimension Map Display Definition Division
//-----
Map_Output {
    Map_Space_Name = "Universe.TwoDimensionalSpace";
    Map_Name = "Space Map";
    Title = "Two Dimensional Space";
    Disable = No;
    Background_Color = 255, 255, 255;
    Position_x = -9;
}

```

```

Position_y = -9;
Size_x = 652;
Size_y = 544;
Remarks = Yes;
Legend_Pos = Top;
Font_Name = "Dialog";
Font_Char_Set = 81;
Font_Size = 12.0;
Font_Style = 0;
Font_Color = 0, 0, 0;
Font_Background_Color = 255, 255, 255;
FastDraw = No;
Draw_Timing = Simulation_Step;
Draw_Interval = 1;
Axis_Label = "", "";
Draw_Range = 0, 0, 100, 100;
Ruled_Line = No;
Text_Display = No;
Text_Display_Variable_Name = "";
Chess_Type_Display_Position = No;
Background_Transparent = No;
Target_Layer = 0;
Coordinate_Display = No;
Space_Kind = Square_2D;
BG_Pict = No;
BG_Type = 0;
BG_Var_Name = "";
BG_File_Name = "";
Origin_Position = 2;
Output_Group_Name = "";
Output {
    Map_Element_Name = "Walker";
    Map_Element = Agent_Variable;
    Output_Expression = "Universe.TwoDimensionalSpace.Walker";
    Draw_Line = No;

```

```

    Collection_Name = "";
    Line_Type = 0;
    Line_Arrow = 0;
    Line_Width = 0;
    Line_Color = 0, 0, 0;
    Line_Type_Variable = "";
    Line_Arrow_Variable = "";
    Line_Width_Variable = "";
    Line_Color_Variable = "";
    Agent_Color = "Universe.TwoDimensionalSpace.Walker.WalkerColor";
    Marker_Id = 1;
    Effective_Figures = 0;
    Format_Id = 0;
    Agent_Image_Direction = No;
    Icon_Type = 0;
    Icon_Var_Name = "";
    Icon_Transparent = No;
    Icon_Enlarg_Reduce = No;
    Icon_File_Name = "";
    Num_Display = No;
    Num_Var_Name = "";
    Agent_Image_Magnify_by_value = Yes;
    Agent_Image_Magnify_Value = 1.0;
    Agent_Image_Magnify_Variable_Name = "";
    Agent_Image_Transparency_by_value = Yes;
    Agent_Image_Transparency_Value = 0.0;
    Agent_Image_Transparency_Variable_Name = "";
}
}

```

```

//-----
// Time Graph Output Definition Division
//-----
Time_Graph {

```

```

Font_Name = "Dialog";
Font_Char_Set = 81;
Font_Size = 12.0;
Font_Style = 0;
Font_Color = 0, 0, 0;
Font_Background_Color = 255, 255, 255;
Draw_Timing = Simulation_Step;
Draw_Interval = 1;
Graph_Name = "扇動者数 Time Series Graph of Red Walkers";
Title = "The Amount of Red Walkers";
Disable = No;
Background_Color = 255, 255, 255;
Remarks = Yes;
Legend_Pos = Top;
FastDraw = No;
Position_x = 494;
Position_y = 242;
Size_x = 1106;
Size_y = 260;
Axis_Label = "", "";
Draw_By_JFree_Chart = Yes;
Scale = 0, 0.0, 0, 1000.0, 10, 50.0;
ScaleAuto = Yes, No;
Output_Group_Name = "";
Output {
    Graph_Element_Name = "扇動者数";
    Line_Color = 255, 0, 51;
    Line_Width = 2;
    Marker_Id = 0;
    Output_Expression = "Universe.WalkersWithInfo";
    Expression_Value_Display = No;
    Effective_Fraction_Figures = 0;
    Graph_Element_Color = (255, 0, 51);
}
}

```



```

//-----
// Time Graph Output Definition Division
//-----
Time_Graph {
    Font_Name = "Dialog";
    Font_Char_Set = 81;
    Font_Size = 12.0;
    Font_Style = 0;
    Font_Color = 0, 0, 0;
    Font_Background_Color = 255, 255, 255;
    Draw_Timing = Simulation_Step;
    Draw_Interval = 1;
    Graph_Name = "鎮静者数 TimeSerieses Graph of Blue Walkers";
    Title = "The Amount of Blue Walkers";
    Disable = No;
    Background_Color = 255, 255, 255;
    Remarks = Yes;
    Legend_Pos = Top;
    FastDraw = No;
    Position_x = 491;
    Position_y = 503;
    Size_x = 1112;
    Size_y = 261;
    Axis_Label = "", "";
    Draw_By_JFree_Chart = Yes;
    Scale = 0, 0.0, 1000, 1000.0, 100, 50.0;
    ScaleAuto = Yes, No;
    Output_Group_Name = "";
    Output {
        Graph_Element_Name = "鎮静者数";
        Line_Color = 0, 0, 255;
        Line_Width = 2;
        Marker_Id = 0;
    }
}

```

```

        Output_Expression = "Universe.WalkersWithInfo2";
        Expression_Value_Display = No;
        Effective_Fraction_Figures = 0;
        Graph_Element_Color = (0, 0, 255);
    }
}

//-----
// Time Graph Output Definition Division
//-----
Time_Graph {
    Font_Name = "Dialog";
    Font_Char_Set = 81;
    Font_Size = 12.0;
    Font_Style = 0;
    Font_Color = 0, 0, 0;
    Font_Background_Color = 255, 255, 255;
    Draw_Timing = Simulation_Step;
    Draw_Interval = 1;
    Graph_Name = "一般人数 Time Series Graph of Green Walkers";
    Title = "The Amount of Green Walkers";
    Disable = No;
    Background_Color = 255, 255, 255;
    Remarks = Yes;
    Legend_Pos = Top;
    FastDraw = No;
    Position_x = 492;
    Position_y = -5;
    Size_x = 1108;
    Size_y = 245;
    Axis_Label = "", "";
    Draw_By_JFree_Chart = Yes;
    Scale = 0, 0.0, 1000, 1000.0, 100, 50.0;
    ScaleAuto = Yes, No;
}

```

```

Output_Group_Name = "";
Output {
    Graph_Element_Name = "一般人数";
    Line_Color = 102, 255, 0;
    Line_Width = 2;
    Marker_Id = 0;
    Output_Expression = "Universe.WalkersWithInfo3";
    Expression_Value_Display = No;
    Effective_Fraction_Figures = 0;
    Graph_Element_Color = (102, 255, 0);
}
}

```

```

//-----
// Time Graph Output Definition Division
//-----
Time_Graph {
    Font_Name = "Dialog";
    Font_Char_Set = 81;
    Font_Size = 12.0;
    Font_Style = 0;
    Font_Color = 0, 0, 0;
    Font_Background_Color = 255, 255, 255;
    Draw_Timing = Simulation_Step;
    Draw_Interval = 1;
    Graph_Name = "10 ステップ間の扇動者数の移動平均";
    Title = "";
    Disable = No;
    Background_Color = 255, 255, 255;
    Remarks = Yes;
    Legend_Pos = Top;
    FastDraw = No;
    Position_x = 179;
    Position_y = 646;
}

```

```

Size_x = 326;
Size_y = 119;
Axis_Label = "", "";
Draw_By_JFree_Chart = Yes;
Scale = 0, 0.0, 1000, 1000.0, 100, 50.0;
ScaleAuto = Yes, No;
Output_Group_Name = "";
Output {
    Graph_Element_Name = "扇動者数の移動平均";
    Line_Color = 255, 0, 0;
    Line_Width = 2;
    Marker_Id = 0;
    Output_Expression = "Universe.IdoHeikinA";
    Expression_Value_Display = No;
    Effective_Fraction_Figures = 0;
    Graph_Element_Color = (51, 51, 51);
}
}

//-----
// Time Graph Output Definition Division
//-----
Time_Graph {
    Font_Name = "Dialog";
    Font_Char_Set = 81;
    Font_Size = 12.0;
    Font_Style = 0;
    Font_Color = 0, 0, 0;
    Font_Background_Color = 255, 255, 255;
    Draw_Timing = Simulation_Step;
    Draw_Interval = 1;
    Graph_Name = "10 ステップ間の鎮静者数の移動平均";
    Title = "";
    Disable = No;

```

```

Background_Color = 255, 255, 255;
Remarks = Yes;
Legend_Pos = Top;
FastDraw = No;
Position_x = 178;
Position_y = 518;
Size_x = 329;
Size_y = 131;
Axis_Label = "", "";
Draw_By_JFree_Chart = Yes;
Scale = 0, 0.0, 1000, 1000.0, 100, 50.0;
ScaleAuto = Yes, No;
Output_Group_Name = "";
Output {
    Graph_Element_Name = "鎮静者数の移動平均";
    Line_Color = 0, 255, 0;
    Line_Width = 2;
    Marker_Id = 0;
    Output_Expression = "Universe.IdoHeikinB";
    Expression_Value_Display = No;
    Effective_Fraction_Figures = 0;
    Graph_Element_Color = (204, 204, 204);
}
}

```

```

//-----
// Control Panel Definition Division
//-----
Control_Panel {
    Display_Area = 0, 0, 100, 100;
}

```

```

//-----

```

```

// Space Initial Value Definition Division
//-----
Space_Initial {
}

//-----
// Other Miscellaneous Data Definition Division
//-----
Others_Data {
    Indent = 4;
    Rule_Colored = Yes;
    Window_Hide_Run = 1;
    Debug_Position_x = 75;
    Debug_Position_y = 75;
    Debug_Size_x = 400;
    Debug_Size_y = 200;
    Console_Position_x = -3;
    Console_Position_y = 513;
    Console_Size_x = 195;
    Console_Size_y = 252;
    Componen_Tree_Position_x = 3;
    Componen_Tree_Position_y = 2;
    Componen_Tree_Size_x = 250;
    Componen_Tree_Size_y = 400;
    Console_Background_Color = 16777215;
    Console_Foreground_Color = 255;
    Console_DisplayLines_Max = 500;
    Font_Name = "Dialog";
    Font_Char_Set = 0;
    Font_Size = 12.0;
    Font_Style = 0;
    Font_Color = 0, 0, 0;
    Font_Background_Color = 0, 0, 0;
}

```

```

//-----
// Network Information Definition Division
//-----
Network_Info {
    Global_Space_Area = 0, 0, 100, 100;
    Local_Space_Info {
    }
}

//-----
// Window Location Definition Division
//-----
Window_Rectangle {
    Window_Name = Main_Window, -8, -8, 1616, 876;
    Window_Name = Output_Setting_Window, 568, 293, 480, 290;
    Window_Name = Artisoc_Help_Window, 100, 172, 693, 400;
}

//-----
// Help Favorite Division
//-----
Help_Favorite {
}

```

付録 7 シミュレーション・プログラムの説明

7.1 このシミュレーションについて

扇動者と鎮静者が一般人に流言を伝達することで、情報を扇動化または沈静化させ流言をシミュレートするモデルである。

「二次元空間」上を、情報を持った扇動者（赤）と鎮静者が一般人（緑）と隣接すると、情報は伝達され、隣接エージェントの内部パラメータである流言

信念強度が加減算され、隣接エージェントの流言信念強度の値が上限閾値を超えると扇動者(赤)になり、下限閾値を下回れば鎮静者、上限閾値と下限閾値の間で一般人になる。

7.2 設定

本モデルでは 100×100 の二次元空間上で登場人物が移動しています、空間の大きさを変える手順は次の通りである。

「ツリー」上の「空間」選択した状態で、「空間」を右クリックし、メニューから「プロパティ」を選択すると、「空間プロパティ」ウィンドウが表示され、そのウィンドウ内の「空間の大きさ X」、「空間の大きさ Y」に、適当な空間の大きさを入力して「了解」をクリックすると空間の大きさを設定される。

エージェント数は、コンポーネントツリー上の「空間」をクリック選択し、マウスの右ボタンを押し、メニューから「エージェント型の追加」を選択する。

「エージェントプロパティ」ウィンドウが表示され、そのウィンドウ内のエージェント数に必要なエージェント数を入力します。設定が完了したら「了解」をクリックするとエージェント数が設定される。

各登場人物の人数を変える手順は、扇動者数では `Univ_Init` 部の変数 `Universe.Sendosha` に扇動者の人数を代入する。

鎮静者数は、同じく `Univ_Init` 部の変数 `Universe.Thinsei` に鎮静者の人数を代入する

一般人は、エージェント数から、扇動者数と鎮静者数を引いた人数になる。

本論文の各登場人物の人数は、扇動者数が 5 人、鎮静者数が 8 人、一般人が 900 人である。

流言伝達強度の設定は、変数 `Universe.Kyodo` に適当な値を代入する。

本研究モデルの流言伝達強度は 1 である。

閾値の設定は上限閾値では `Univ_Step_End` 部の変数 `MaShikii` に上限閾値を代入する。

下限閾値は、同じく `Univ_Step_End` 部の変数 `MiShikii` に下限閾値を代入する。

本モデルでは上限閾値が 3、加減閾値が -3 に設定している。

7.3 Univ_Init 部での処理の説明

`Univ_init` はシミュレーションが開始された最初に 1 回実行するルールを記述する。


```

Univ_Init{
    Dim i As Integer
    Dim j As Integer
    Dim obj As Agt
    Dim x As Double
    Dim myAgtSet As AgtSet

    For i = 0 To Universe.WalkerCount - 1
        obj = CreateAgt(Universe.TwoDimensionalSpace.Walker)
    Next i
    MakeAgtSet(myAgtSet, Universe.TwoDimensionalSpace.Walker)
    RandomPutAgtSetCell(myAgtSet, False)

    Universe.sendosha = 5 //初期扇動者数
    x = 3
    Universe.thinseisha = Universe.sendosha + x //初期鎮静者数
    Universe.kyodo = 1 //流言伝達強度
    Universe.hkai = 0
    Universe.SMax = 0
    Universe.TMax = 0
    Universe.WalkersWithInfoA = 0
    Universe.WalkersWithInfo2A = 0
    Universe.WalkersWithInfo3A = 0

    Universe.WalkersWithInfo = Universe.sendosha
    Universe.WalkersWithInfo2 = Universe.thinseisha
    Universe.WalkersWithInfo3 = Universe.WalkerCount - Universe.sendosha - Universe.thinseisha

    For j = 0 to 9
        Universe.IdoHA(j) = 10
        Universe.IdoHB(j) = 10
    Next j
}

```

エージェント集合 `obj` を作成する。`i` が 0 からすべてのエージェントの個数 `Universe.WalkerCount-1` まで、`CreateAgtSet` によって、エージェントの型 `Universe.TwoDimensionalSpace.Walker` のエージェント一個生成して、エージェント集合 `obj` に代入する。

`MakeAgtSet(myAgtSet, Universe.TwoDimensionalSpace.Walker)` は、すべてのエージェント種別型変数 `Universe.TwoDimensionalSpace.Walker` のすべてのエージェントをエージェント集合関数 `MyAgtSet` に格納する。

`RandomPutAgtSetCell(myAgtSet, False)` は、エージェント集合関数 `MyAgtSet` のエージェントをシミュレーション空間内にランダムに配置する。次に初期の登場人物の数を設定と流言伝達強度と流言信念強度の設定を行う。

Universe.Sendosha は初期扇動者数。Universe.TInseisha は初期鎮静者数。
Universe.Kyodo は流言伝達強度の強さ。

Universe.WalkersWithInfoA=0 は、10 ステップ間の扇動者数の移動平均、
Universe.WalkersWithInfo2A=0 は、10 ステップ間の鎮静者数の移動平均、
Universe.WalkersWithInfo3A=0 は、10 ステップ間の一般人の数の移動平均であるが、すべての登場人物の 10 ステップ間の移動平均数を 0 にする。

Universe.WalkersWithInfo は扇動者数の変数、Universe.WalkersWithInfo2 は鎮静者数の変数、Universe.WalkersWithInfo3 は一般人の数の変数、ここでの処理で、初期の各登場人物の数を代入する。

7.4 Univ.Step_End 部での処理

Univ.Step_End はシミュレーションのステップの最後に実行するルールを記述する。

Univ_Step_End{

```
Dim col As AgtSet
Dim obj As Agt
Dim IH As integer
Dim IdoGen As integer
Dim IdoA As integer
Dim IdoB As Integer
Dim sp As integer
Dim MaShikii As Double
Dim MiShikii As Double

Universe.WalkersWithInfo = 0
Universe.WalkersWithInfo2 = 0
Universe.WalkersWithInfo3 = 0
MakeAgtSet(col, Universe.TwoDimensionalSpace.Walker)
```

```
MaShikii = 3
MiShikii = -3
```

```
For Each obj In col
  If obj.Jotai > MaShikii Then
    obj.Ty = 1
    obj.WalkerColor = COLOR_RED
  Else
    If obj.Jotai < MiShikii Then
      obj.Ty = 2
      obj.WalkerColor = COLOR_BLUE
    Else
      obj.Ty = 3
      obj.WalkerColor = COLOR_GREEN
    End If
  End If
```

```

If obj.WalkerColor == COLOR_BLUE Then //鎮静者
    Universe.WalkersWithInfo2 = Universe.WalkersWithInfo2 + 1
End if
If obj.WalkerColor == COLOR_RED Then //扇動者
    Universe.WalkersWithInfo = Universe.WalkersWithInfo + 1
End If
If obj.WalkerColor == COLOR_GREEN Then //一般人
    Universe.WalkersWithInfo3 = Universe.WalkersWithInfo3 + 1
End If
Next obj

```

エージェント集合型 Col を作成する。流言信念強度を表す変数 Jotai が閾値によって、個々のエージェントの種類と 2 次元グラフに表す個々エージェントに色を付け、個々のエージェントを色に処理を分岐させ、現在の各登場人物数を表す各変数 Universe.WalkersWithInfo、Universe.WalkersWithInfo2、Universe.WalkersWithInfo3 に、エージェントを一個ずつ入れていくことで、各登場人物数を数え、その数えた値をエージェント集合変数 obj に代入する。

```

If Universe.WalkersWithInfo == Universe.WalkersWithInfoA Then
If Universe.WalkersWithInfo2 == Universe.WalkersWithInfo2A Then
If Universe.WalkersWithInfo3 == Universe.WalkersWithInfo3A Then
    Universe.st1 = Universe.st1 + 1
If Universe.st1 == 10 Then

    ExitSimulationMsg( "登場人物の割合が変わらなくなるまでのステップ数 " & GetCountStep())
End If
End If
End If
End If
    Universe.WalkersWithInfoA = Universe.WalkersWithInfo

If Universe.WalkersWithInfo == CountAgtSet(col) Then
    ExitSimulationMsg ( "すべてのエージェントが扇動者になるまでのステップ数 " & GetCountStep())
End If
If Universe.WalkersWithInfo2 == CountAgtSet(col) Then
    ExitSimulationMsg( "すべてのエージェントが鎮静者になるまでのステップ数 " & GetCountStep())
End If

If Universe.WalkersWithInfo3 == CountAgtSet(col) Then
    ExitSimulationMsg ( "すべてのエージェントが一般人になるまでのステップ数 " & GetCountStep())

```

```

If Universe.WalkersWithInfo3 == CountAgtSet(col) Then
  ExitSimulationMsg ( "すべてのエージェントが一般人になるまでのステップ数 " & GetCountStep())
End If

```

```

PrintLn ("ステップ数" & GetCountStep())

```

```

If Universe.WalkersWithInfo == 0 Then
  ExitSimulationMsg ( "扇動者が消滅するまでのステップ数 " & GetCountStep())
End If

```

```

If Universe.WalkersWithInfo2 == 0 Then
  ExitSimulationMsg ( "鎮静者が消滅するまでのステップ数 " & GetCountStep())
End If

```

```

If Universe.WalkersWithInfo3 == 0 Then
  PrintLn ( "一般人が消滅しました。 " & GetCountStep())

```

```

End If

```

扇動者か鎮静者が消滅して停止条件になった場合の終了処理。一般人が消滅したときはメッセージだけ表示する。

```

If GetCountStep() > -1 Then //現在の鎮静者数を数える。
  IH = 0
  sp = GetCountStep()
  IH = sp mod 10 //IHの変数を作る

  Universe.IdoHA(IH) = Universe.WalkersWithInfo
  Universe.IdoHB(IH) = Universe.WalkersWithInfo2
  IdoGen = 0
  IdoA = 0
  IdoB = 0

  For IdoGen to 9 // 10ステップ前から現在までの扇動者数を数える。
    IdoB = IdoB + Universe.IdoHB(IdoGen)
  
```

```

Next IdoGen
  Universe.IdoHeikinA = IdoA / 10
  Universe.IdoHeikinB = IdoB / 10

```

登場人物数の 10 ステップ間の移動平均を求める処理。

```

If Universe.IdoHeikinA > Universe.SMax Then //扇動者数の過去最大数を求める
    Universe.SMax = Universe.IdoHeikinA
    Universe.SStep = GetCountStep()
End If
If Universe.IdoHeikinB > Universe.TMax Then //鎮静者数の過去最大数を求める
    Universe.TMax = Universe.IdoHeikinB
    Universe.TStep = GetCountStep()
End If

```

各登場人物数の今までの最大数を求める処理。

```

If Universe.IdoHeikinA < Universe.IdoheikinA1 Then
    Universe.IdoHKaiA = Universe.IdoHKaiA + 1
    If Universe.IdoHKaiA > 0 Then
    If Universe.IdoHKaiA > 4 Then
        println("扇動者数の移動平均が5ステップ以上連続で減少しました。 " & GetCountStep())
    End If
    If Universe.SMax > (Universe.IdoHeikinA*2) Then
        println("扇動者数 " & Universe.WalkersWithInfo)
        println("最大扇動者数 " & Universe.SMax)
        println("最大扇動者数でのステップ数 " & Universe.SStep)
        ExitSimulationMsg ( "扇動者数が減少しており、扇動者数とその最大数の半分なるまでのステップ数 " & GetCountStep())
    End If
    End If
else
    Universe.IdoHKaiA = 0
End If
    Universe.IdoHeikinA1 = Universe.IdoHeikinA

If Universe.IdoHeikinB < Universe.IdoheikinB1 Then
    Universe.IdoHKaiB = Universe.IdoHKaiB + 1
    If Universe.IdoHKaiB > 0 Then
    If Universe.IdoHKaiB > 4 Then
        println("鎮静者数の移動平均が5ステップ連続で減少するまでのステップ数。 " & GetCountStep())
    End If
    If Universe.TMax > (Universe.IdoHeikinB*2) Then
        println("鎮静者数 " & Universe.WalkersWithInfo2)
        println("最大鎮静者数 " & Universe.TMax)
        println("最大鎮静者数でのステップ数 " & Universe.TStep)
        ExitSimulationMsg ( "鎮静者数が減少しており、鎮静者数とその最大数の半分なるまでのステップ数 " & GetCountStep())
    End If
    End If
else
    Universe.IdoHKaiB = 0
End If
    Universe.IdoHeikinB1 = Universe.IdoHeikinB
End If

```

登場人物数移動平均が、最大数の半数以下で、その登場人物数が減少している場合の処理。

```

If GetCountStep() == 300 Then
  println("最大の鎮静者数" & Universe.TMax)
  println("最大鎮静者数でのステップ数" & Universe.TStep)
  println("最大の扇動者数" & Universe.SMax)
  println("最大扇動者数でのステップ数" & Universe.SStep)
  println("扇動者数" & Universe.WalkersWithInfo)
  println("鎮静者数" & Universe.WalkersWithInfo2)
  ExitSimulationMsg ("300ステップです。シミュレーションを終了します。")
End If
}

```

300 ステップ間停止状態にならなかった場合停止条件になるが、その停止条件での処理。

7.5 Agt_Init 部での処理

Agt_Init はエージェントが生成される時に 1 回実行するルールを記述する。

```

Agt_Init{
  Dim obj As Agt
  Dim Shoki As AgtSet
  Dim a as integer
  MakeAgtSet(Shoki, Universe.TwoDimensionalSpace.Walker)
  My.Jotai = 0
  My.Ty = 3 //MyTy=3は一般人。
  My.WalkerColor = COLOR_GREEN
  If My.ID < Universe.sendosha Then
    My.Ty = 1 //MyTy=1は扇動者。
    My.Jotai = 3.2
    My.WalkerColor = COLOR_RED
  Else
    If My.ID < Universe.sendosha + Universe.thinseisha Then
      My.Ty = 2 //MyTy=2は鎮静者。
      My.Jotai = -3.2
      My.WalkerColor = COLOR_BLUE
    End If
  End If

  Turn(360*Rnd())
}

```

MakeAgtSet でエージェント集合変数 Shoki を作成する。

0 から扇動者数まで、エージェントの流言信念強度を表す Jotai を 3.2、エージェントの登場人物表す変数 Ty を 1 に、歩行者の色を赤に設定。

扇動者数から鎮静者数までのエージェントをエージェントの流言信念強度を表す Jotai を -3.2、エージェントの登場人物表す変数 Ty を 2、エージェントの色を青に設定。残りのエージェントは登場人物を表す変数は 3、歩行者の色を緑に設定した。

Turn(360 *Rnd())はエージェントが進む向きを 360 の単位で決定する。

7.6 Agt_Step 部での処理

Agt_Step はシミュレーションのステップ毎に実行するルールを記述する。

Agt_Step{

```
Dim SurroundingAgents As AgtSet
Dim obj As Agt
Dim i As Integer
Dim j As Integer
Dim k As Integer
Dim n As Integer
Dim m As Integer
Dim a As Double
Dim b As Double
Dim c As Integer
Dim d As Integer
Dim e As Integer
Dim col1 As AgtSet
Dim z As Integer
Dim kae As AgtSet
Dim hind As Integer

Turn(Universe.StaggerLevel*(Rnd() - 0.5))
Forward(1.0)

MakeAllAgtSetAroundOwnCell(SurroundingAgents, 1, False)
For Each obj In SurroundingAgents
a=0
    i=obj.ID
    j=My.ID
    If obj <> My Then
```

Turn(Universe.StaggerLevel*(Rnd() - 0.5))

Forward(1.0)

エージェントを-0.5 から 0.5 * 現在の進行方向でエージェントの進行方向を変える。

MakeAllAgtSetAroundOwnCell(SurroundingAgents, 1, False)

For Each obj In SurroundingAgents

a=0

i=obj.ID

j=My.ID

If obj <> My Then

MakeAllAgtSetAroundeOwnCell で個々のエージェントの周り 1 の距離ある他のエージェントをだけ探し、他のエージェントが周りにあるエージェントをエージェント集合変数 SurroundingAgents に代入する。

SurroundingAgents に代入された、エージェントに処理を与える。

```
i=obj.ID
j=My.ID
If obj <> My Then

// エージェント同士が隣接した時の情報伝達
If My.Ty == 3 Then
If obj.Ty == 3 Then
My.hai(i)= 9
End If
If obj.Ty == 1 Then

a = a + Universe.kyodo //流言信念強度を加減算
My.hai(i)= 7
End if
If obj.Ty == 2 Then
a = - Universe.kyodo //流言信念強度を加減算
My.hai(i) = 8
End If
If obj.Ty == 10 Then
a = a + Universe.kyodo //流言信念強度を加減算
My.hai(i) = 7
End If
If obj.Ty == 11 Then
a = a - Universe.kyodo //流言信念強度を加減算
My.hai(i) = 8
End If
End If

If My.Ty == 1 Then
If obj.Ty ==1 Then
a = a +(Universe.kyodo*0.65)
My.hai(i)= 1
End If
If obj.Ty == 2 Then
a = a -(Universe.kyodo*0.65)
My.hai(i)=2
End If
If obj.Ty== 3 Then
My.hai(i) = 3
End If
```



```

    If obj.Ty == 10 Then
        a = a + Universe.kyodo //流言信念強度を加減算
        My.hai(i) = 1
    End If
    If obj.Ty == 11 Then
        a = a - Universe.kyodo //流言信念強度を加減算
        My.hai(i) = 2
    End If
End If

If My.Ty == 2 Then
    If obj.Ty == 1 Then
        a = a +(Universe.kyodo*0.65)
        My.hai(i)= 4
    End If
    If obj.Ty==2 Then
        a = a -(Universe.kyodo*0.65)
        My.hai(i)= 5
    End if
    If obj.Ty == 3 Then
        My.hai(i)= 6
    End If
    If obj.Ty == 10 Then
        a = a + Universe.kyodo //流言信念強度を加減算
        My.hai(i) = 5
    End If
    If obj.Ty == 11 Then
        a = a - Universe.kyodo //流言信念強度を加減算
        My.hai(i) = 6
    End If
End If
End If
End If
My.Catch = My.Catch + a
next obj

```

エージェント同士が隣接した時の情報伝達を再現している。エージェントの周りに来たエージェントと、自エージェントの種類によって処理を分け、**a** を流言信念強度で、加算して、エージェントの組み合わせを表す変数に **Hai** に組み合わせを表す数値を代入する。またエージェントの組み合わせは配列で表した。

```
MakeAgtSet(kae, Universe.TwoDimensionalSpace.Walker)
```

```
For Each obj in kae
```

```
z = obj.id
```

```
If obj <> My Then
```

```
  If obj.hai(z) > 0 Then
```

```
    If My.Ty == 1 Then
```

```
      If obj.Ty == 1 Then
```

```
        my.hai(z) = 1
```

```
      End If
```

```
      If obj.Ty == 2 Then
```

```
        my.hai(z) = 2
```

```
      End If
```

```
      If obj.Ty == 3 Then
```

```
        my.hai(z) = 3
```

```
      End If
```

```
    End If
```

```
    If My.Ty == 2 Then
```

```
      If obj.Ty == 1 Then
```

```
        my.hai(z) = 4
```

```
      End If
```

```
      If obj.Ty == 2 Then
```

```
        my.hai(z) = 5
```

```
      End If
```

```
      If obj.Ty == 3 Then
```

```
        my.hai(z) = 6
```

```
      End If
```

```
    End If
```

```
    If My.Ty == 3 Then
```

```
      If obj.Ty == 1 Then
```

```
        my.hai(z) = 7
```

```
      End If
```

```
      If obj.Ty == 2 Then
```

```
        my.hai(z) = 8
```

```
      End If
```

```
      If obj.Ty == 3 Then
```

```
        my.hai(z) = 9
```

```
      End If
```

```
    End If
```

```
  End If
```

```
End If
```

```
next obj
```

このシミュレーションではエージェントの登場人物を表す **Ty** が変化するが、ここでエージェントの組み合わせを表す変数 **Hai** を現在のエージェントの **Ty** で、エージェントの組み合わせで変数 **Hai** に組み合わせを表す数値を代入する。

```

MakeAgtSet(col1, Universe.TwoDimensionalSpace.Walker)
Universe.e = 0 // いない
For Each obj in col1
    b = 0
    k = obj.Id
    for hind = 0 to Universe.hkai
        If obj.hai(k) == 1 Then
            b = b + (Universe.kyodo*0.65)*0.65
        End If
        If obj.hai(k) == 2 Then
            b = b - (Universe.kyodo*0.65)*0.65
        End If
        If obj.hai(k) == 3 Then
        End If
        If obj.hai(k) == 4 Then
            b = b + (Universe.kyodo*0.65)*0.65
        End If
        If obj.hai(k) == 5 Then
            b = b - (Universe.kyodo*0.65)*0.65
        End If
        If obj.hai(k) == 6 Then
        End If
        If obj.hai(k) == 7 Then
            b = b + (Universe.kyodo*0.65)
        End If
        If obj.hai(k) == 8 Then
            b = b - (Universe.kyodo*0.65)
        End If
        If obj.hai(k) == 9 Then
        End If

        If obj.hai(k) == 10 Then
            b = b + (Universe.kyodo*0.65)
        End If

        If obj.hai(k) == 11 Then
            b = b - (Universe.kyodo*0.65)
        End If
    Next hind
// エージェント同士が隣接した時の情報伝達で加算された変数 Catch とコネクション・ネットワークで加算された b を加算する。
obj.jotai = obj.jotai + obj.Catch + b
Next obj
Universe.e = 0

```

この部分での処理はコネクション・ネットワークでの伝達処理を再現している。

エージェントの組み合わせを表す Hai に代入された値によって、処理を分岐させ、b に情報伝達強度を加算する。

$\text{obj.jotai} = \text{obj.jotai} + \text{obj.Catch} + b$

流言信念強度を表す変数 Jotai に、1 ステップ前の Jotai とエージェント同士が隣接した時の情報伝達で加算された変数 Catch とコネクション・ネットワークで加算された b を加算する。