マルチエージェントシステムに基づく建物内避難時間の簡易予測法の提案

-目次-

1章　序論

1.1　研究背景----------1
1.2　避難シミュレーションに関する既往研究----------1
1.3　マルチエージェントシステムについて----------3
1.4　既往研究と本研究の比較----------3
1.5　研究目的----------3
1.6　本論の構成----------4

2章　避難シミュレーションモデルの概要

2.1　シミュレータ artisoc academic 3.0 について----------5
2.2　対象モデルの設定
2.2.1　case1（部屋数：2，室出口位置：中央）----------6
2.2.2　case2（部屋数：4，室出口位置：端）----------7
2.2.3　case3（部屋数：4，室出口位置：case2 と case4 の中間）----------7
2.2.4　case4（部屋数：4，室出口位置：中央）----------8
2.3　エージェントのルール設定
2.3.1　hito エージェント----------8
2.3.2　wall エージェント----------10
2.3.3　exit エージェント----------10
2.3.4　dummy エージェント----------10
2.3.5　adjust エージェント----------10
2.3.6　counter エージェント----------11

3章　数値例題

3.1　case1（部屋数：2，室出口位置：中央）の結果考察----------12
3.2　case2（部屋数：4，室出口位置：端）の結果考察----------15
3.3　case3（部屋数：4，室出口位置：case2 と case4 の中間）の結果考察----------17
3.4　case4（部屋数：4，室出口位置：中央）の結果考察----------18

4章　避難時間簡易予測式の提案

4.1　パラメータ毎のシミュレーション結果の考察
4.1.1　室出口幅----------23
4.1.2　室出口の位置----------24
4.1.3　避難者数----------24
4.2　簡易予測式の構築----------25
4.3　定性的な分析によるアプローチ----------29

5章　結論----------31

謝辞----------32

参考文献----------33

付録----------36

1章　序論

1.1　研究背景

2011 年に発生した東北地方太平洋沖地震により，東日本は甚大な被害を受けた．この東日本大震災は，地震の被害だけでなく，その地震から引き起こされた津波による被害も大きく，現在も東日本一帯で余震が続いている．その影響もあり近年では，防災に対する意識も高まり，高い防災性能を持った設備が備え付けられた建築物が多くなってきた．

そのようなハード面での対策が数多くなされていく中で，ソフト面である避難行動の検討も重要となってくる．実際に，これまでされてきた研究の中でも，避難完了時間や避難開始時間が避難率に大きく影響しているという結果が挙がっており，「新・建築防災計画指針」の中でも，避難時間に関する計算について詳しく述べられている．このように，様々な防災計画を考えるうえで避難時間という要素は非常に重要であり，避難時間を予測する必要性も高くなってきている．

しかし，避難時間を検証・予測するために避難訓練を頻繁に行うことは，人手や施設利用の側面から考えても極めて困難である．そのため近年では，避難行動を予測するための手法の一つであるコンピュータ・シミュレーションを用いた研究が数多くなされている．

1.2　避難シミュレーションに関する既往研究

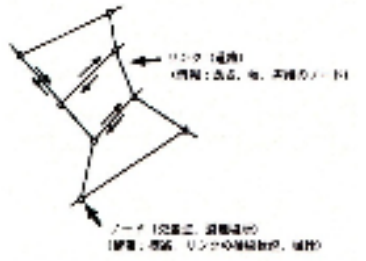
避難シミュレーションは，対象地区の扱いによって

(1) ネットワーク型：実際の市街地パターンや街路パターンを，ノードとリンクからなるネットワークで表現する

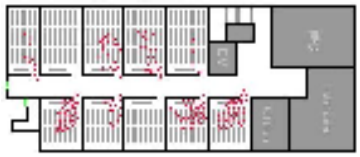
(2) メッシュ型：実際の市街地パターンや街路パターンを，メッシュで表現する

(3) ブロック型：実際の市街地パターンや街路パターンを，そのまま利用する

の 3 つに分類できる．



(1) ネットワーク型
（文献 1 より引用）



(2) メッシュ型
（文献 7 より引用）

(3) ブロック型
（文献 10 より引用）

図 1　対象地区の扱いによる分類

(1) ネットワーク型

今村・鈴木・谷口ら[1]は，過去に北海道南西沖地震の被害を受けた北海道奥尻島青苗地区でのアンケート調査を基に避難シミュレーションモデルを開発し，シミュレーション結果とアンケート結果を比較検討することにより，開発したモデルの妥当性を検証した．さらにその後，今村・鈴木ら[2]によって，住民の津波避難に対する意識・意向を取り込んだ改良モデルを開発し，アンケート結果との妥当性を検証した．金・高井ら[3]は，韓国のソウル市江南区を対象に，集中豪雨を想定した避難シミュレーション手法を構築し，これに基づき集中豪雨の被害を最小化するためのソフト的施策を提案した．円谷・西野ら[4]は，H13 名古屋～長久手線・本郷交差点エリアベンチマークデータセットをモデルの検証データとして使用した．そして，「避難ポテンシャル」を導入し，避難行動を「現在位置より低い避難ポテンシャルを有する位置への移動」と仮定することにより，道路ネットワーク，危険要因，安全要因を考慮した避難行動の高速な予測を可能とした

(2) メッシュ型

高瀬・円満・佐野・渡辺ら[5]は，設計者が普段使っている CAD とパソコンを使って簡易に，また設計変更にも柔軟に対応できるシミュレーションを行うことが可能になるシミュレーションシステムの開発を行った．尾崎・長名ら[6]は，マルチエージェントを用いた避難シミュレーションによって，オフィス内での避難行動を考慮したレイアウト案の評価が行うことができるシステムを提案した．城田・堤ら[7]は，実際の大学の建物を対象としたモデルを構築し，人の流れを円滑にし，避難時さらには通常時の移動時間を短縮できるような教室配置の検証を行った．

(3) ブロック型

渡辺・近藤ら[8]は，津波防災まちづくりに関するソフト的施策の効果を避難者数で評価できる津波避難シミュレーションモデル開発のため，徳島県鞆浦区を対象に，避難開始時間を変えてシミュレーションを行い，避難が完了した人数で評価を行った．安福ら[9]は，実際にある大学キャンパスを対象に，Social Force モデルによる避難行動を組み込んだ地図情報データベースが適用可能な広域避難シミュレーションを開発し，被害状況の変化に対してリアルタイムに避難行動を行うための避難行動のモデル化および空間のモデル化手法を提案した．玉川・大畑ら[10]は，北海道釧路市中心部の津波浸水危険地域を対象にシミュレーションを行い，現況の評価，施設を増強した場合の効果，問題点の検討を行った．

1.3　マルチエージェントシステムについて

異なったルールを持たせた複数のエージェントモデルを用いて，複数かつある一定以上のエージェントを多数設定することで，各々のエージェントの相互作用をシミュレーションするシステム．同じ特徴を持った複数のノードの相互作用によるシミュレーションとは異なり，様々な個性を持ち，一見予測不可能な人間社会などのモデルや群衆の動きを表現することに適している．

1.4　既往研究と本研究の比較

既往研究の避難シミュレーションモデルの多くは，避難速度について，一定もしくは年齢によって一定であり，周囲の状況に応じた速度変化についてあまり考慮されていない．また，対象地区についても，実際の地区や建物を対象としたものがほとんどである．そこで，本研究で使用するモデルは，仮想的なフロアモデルで構築し，避難行動をよりリアルなものに近付けるため，前方の滞留によって避難速度と避難方向を変化させる．

そこで本研究では，群集流動シミュレーションに適した正方形メッシュによるモデルを用いる．

1.5　研究目的

前節で述べた既往研究からもわかるように，既往研究のほとんどは，実存の建物や地区を対象とし，比較と検証を目的としたものである．そのため，研究で得られた分析結果や検証結果を他の地区や建物へ直接適用することが困難である．そこで，本研究では，様々な建物に適用できる避難時間の簡易予測式を提案することを目的とする．ここで，本研究の中で想定している災害は地震災害である．

はじめに，マルチエージェントシステムによる群集流動シミュレーションを用いることで，避難行動をよりリアルに再現したモデルを構築する．そして，ある一定の形状を持った階層と廊下を設定し，室出口の位置や幅，避難者数をパラメータとして変化させながらケーススタディを行う．最後に，シミュレーションによって得られた結果を比較・分析することで，避難時間の簡易予測式を提案する．

1.6　本論の構成

1 章では, 本研究の背景・目的, 既往研究, マルチエージェントシステムについて述べる.

2 章では, 本研究で用いるソフトウェアの概要とシミュレーションモデルの設定やエージェントのルールについて述べる.

3 章では，各モデルにおいてシミュレーションで得られた結果を分析する.

4 章では，前章で得られた各データの分析結果からパラメータ毎に考察し，簡易予測式を提案する．また，定性的な分析よる簡易予測式も提案し，考察を行う.

5 章では，本研究についてのまとめと今後の課題について述べる.

謝辞・参考文献

付録　　付 1　Universe のルールプログラム

　　付 2　各エージェントのルールプログラム

2 章　避難シミュレーションモデルの概要

2.1　artisoc academic 3.0 について [11)]

本研究で使用する artisoc academic 3.0 は，構造計画研究所が開発したソフトウェアである．文献 11 に基づいて artisoc academic 3.0 についての概要を説明する．Visual Basic に準拠したルール書式で，便利な関数があらかじめ組み込まれているため，マルチエージェントシステムを容易に構築し．シミュレートを行うことができる特徴を持つ．

このソフトウェアの中で，人工社会のモデルは，

・「エージェント」と呼ばれる行動主体

・個々のエージェントの属性（性質や役割）を表す「変数」

・エージェントが行動する（他のエージェントと関係する）「ルール」

・エージェントが行動（相互作用）する「空間」ないし「場」

・シミュレーションやモデル全体に関わる「変数」や「ルール」

を基本的な要素にしている．

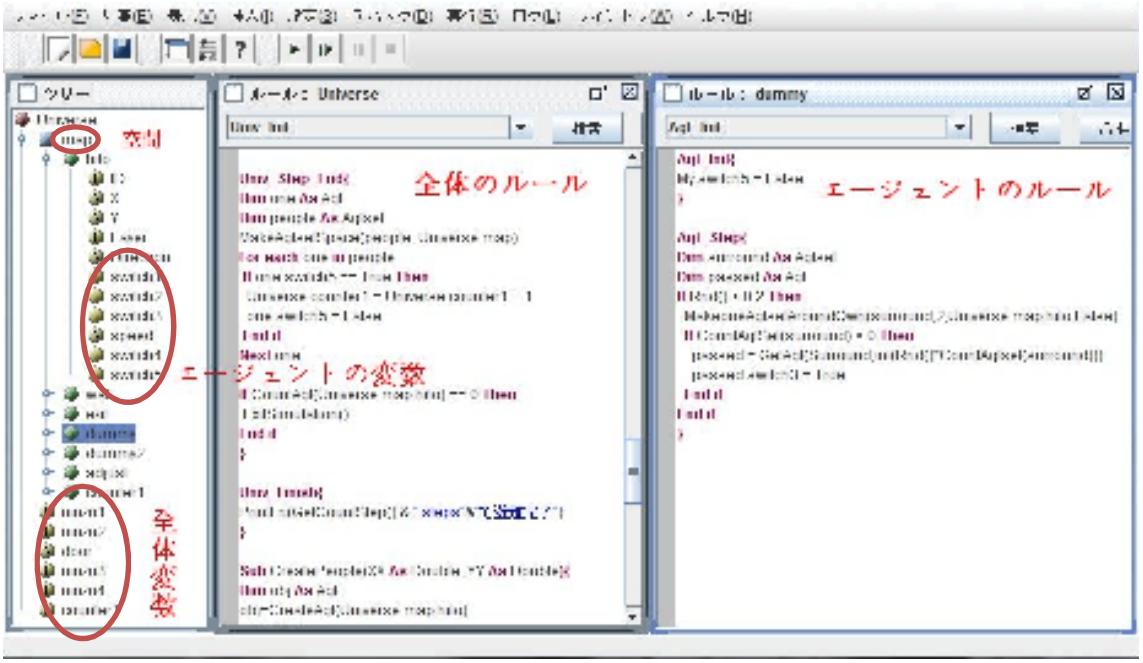


図 2　artisoc 起動画面

2.2　対象モデルの設定

artisoc academic 3.0 では，複数マスのセルで分割された空間において一定数の step を繰り返すことでシミュレーションを行っている．本システムでは文献 7 を参考に，1 マスを約 0.5m×0.5m とし，3step を 1 秒として時間を表現している．そして，対象とするモデルは，幅 4m の廊下が中央に通った，幅 20m・奥行 80m の長細い階層を想定する．

文献 12 より，避難完了時間には，「避難者数」・「室出口幅」・「室出口の位置」が関係してくると推測できる． そこで，部屋の数や室出口の位置を変更したモデルをいくつか提案し，「室出口幅」や「総避難者数」に関しては，モデル内のコントロールパネルによって任意に変更する．また，室出口幅は各部屋同じ広さとする．各モデルの特徴を示した表を下に示し，各モデルの概略を以下で説明する．

表 1　各モデルの特徴

	部屋の数	室出口の位置
case1	2	部屋の中央
case2	4	部屋の端
case3	4	case2 と case4 の中間
case4	4	部屋の中央

2.2.1　case1（部屋数：2，室出口位置：中央）

case1 モデルでは，部屋の数は廊下の上下に 1 部屋ずつの計 2 部屋配置し，室出口の位置は部屋の中央に設ける．

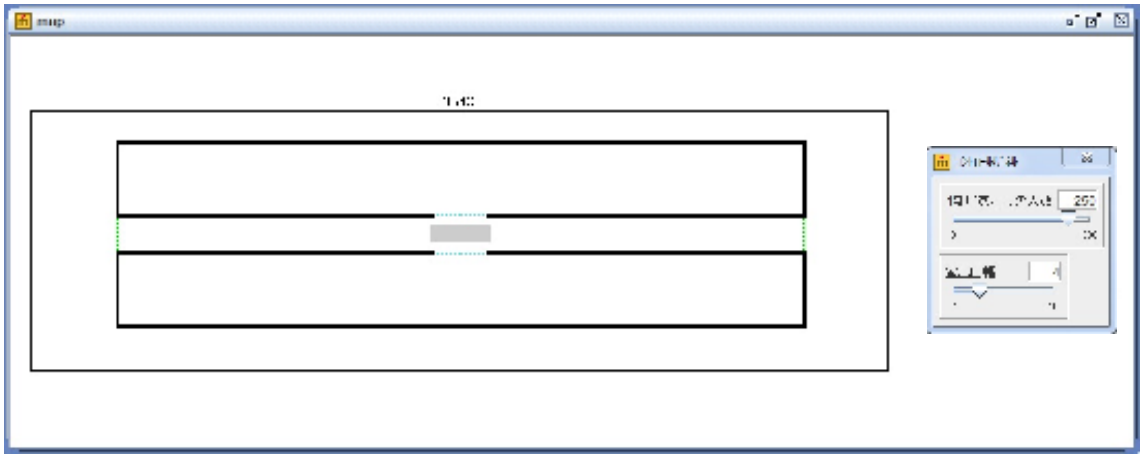


図 3　case1 概略図

2.2.2　case2（部屋数：4，室出口位置：端）

case2 モデルでは，部屋の数は廊下の上下に 2 部屋ずつの計 4 部屋配置し，室出口の位置は部屋の端に設ける．

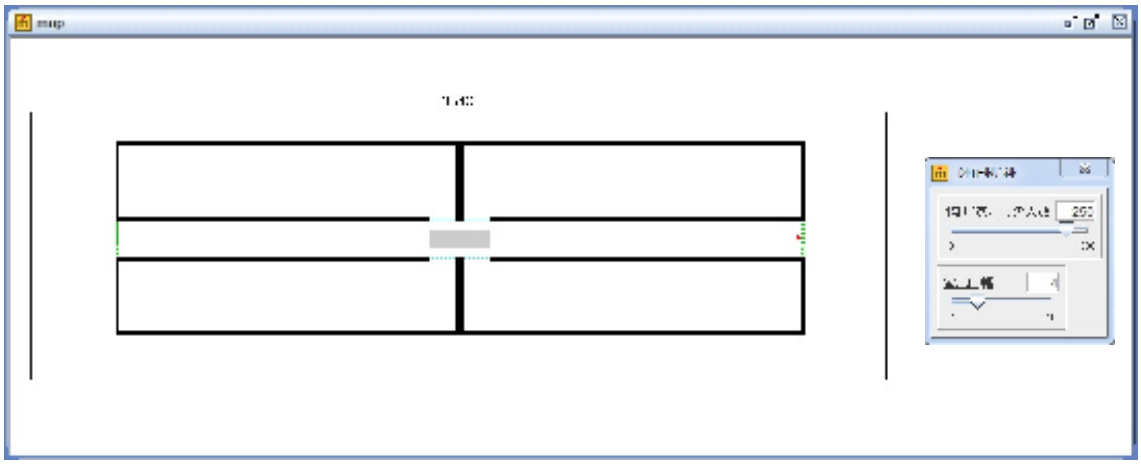


図 4　case2 概略図

2.2.3　case3（部屋数：４，室出口位置：case2 と case4 の中間）

case3 モデルでは，部屋の数は廊下の上下に 2 部屋ずつの計 4 部屋配置し，室出口の位置は case2 と case4 の中間に設ける．

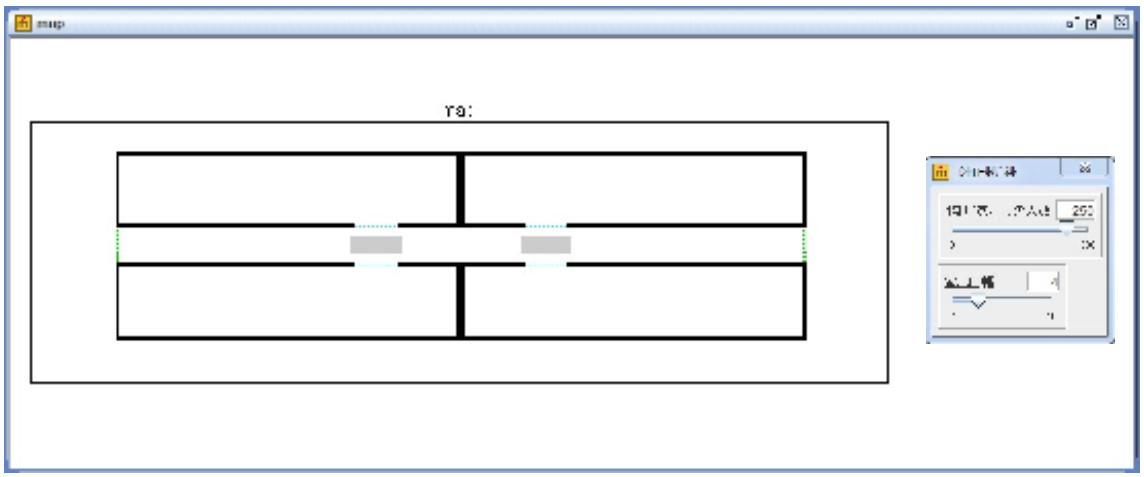


図 5　case3 概略図

2.2.4　case4（部屋数：４，室出口位置：中央）

case4 モデルでは，部屋の数は廊下の上下に 2 部屋ずつの計 4 部屋配置し，室出口の位置は部屋の中央に設ける．

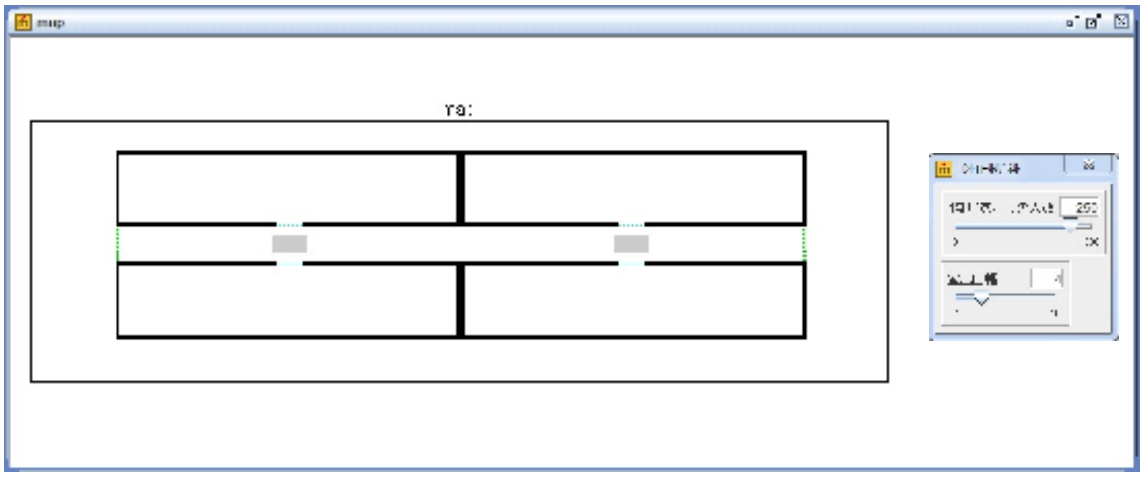


図 6　case4 概略図

2.3　エージェントのルール設定

本研究で使用するモデルには，

・hito エージェント　　・dummy エージェント
・wall エージェント　　・adjust エージェント
・exit エージェント　　・counter エージェント

の計 6 種類のエージェントが存在する．各エージェントのルールについて以下で説明する．

2.3.1　hito エージェント

本研究では地震災害を想定しているため，建物の外に出ることで避難完了とする．
はじめに hito エージェントは，各部屋に同じ人数配置され，その配置はランダムである．そして，廊下を目指し．室出口を通って廊下へ出た後，近い方の出口へ向かう．hito エージェントの目的地の切り替えは，hito エージェントが持つブール型変数（True と False の 2 つのみの変数）switch3 を False から True にすることで行う．

避難速度については，前方に人がいない状態を標準速度（1.2m/s）とし，前方の滞留によって減速する．また，前方に壁や滞留があった場合，それらを避けて進む．これらの避難行動に関するフローチャートを図 7 に記す．

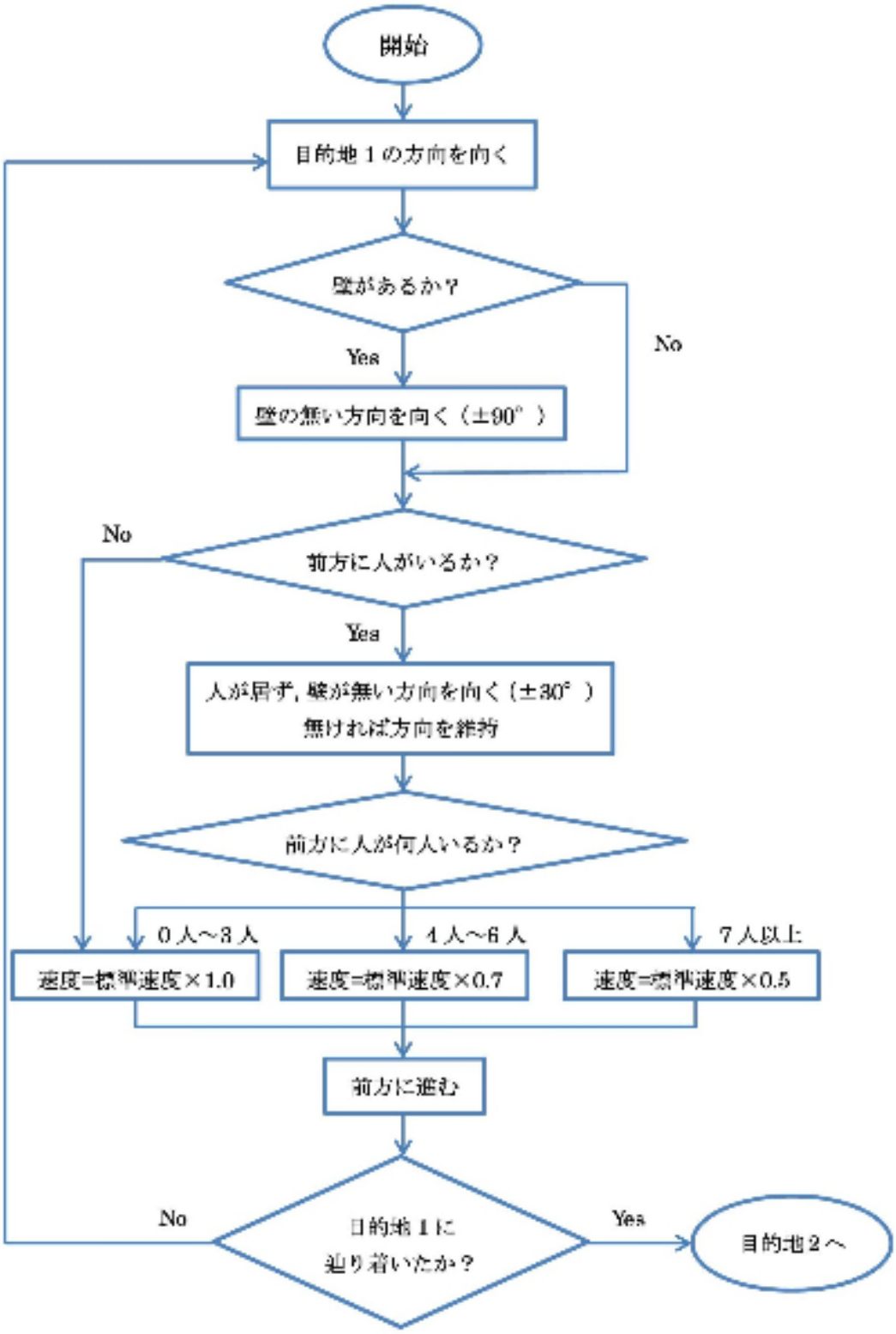


図 7　hito エージェントのフローチャート

2.3.2　wall エージェント

wall エージェントによって建物のレイアウトを表現する．建物の外壁に関しては，あらかじめ用意してある Excel の外部ファイルを初期設定に入力することで配置する．

2.3.3　exit エージェント

廊下の両側に配置し，避難が完了したという意味合いで exit エージェントまでたどり着いた hito エージェントを消す役割を果たす．exit エージェントによって全ての hito エージェントが消されたとき，シミュレーションが終了する．

2.3.4　dummy エージェント

hito エージェントが exit エージェントのある出口を目指す際に誘導する役割を果たす．dummy エージェントは向かい合う部屋の室出口位置の間に挟まれる形で配置され，hito エージェントははじめに dummy エージェントを目指し，その後 exit エージェントを目指す．

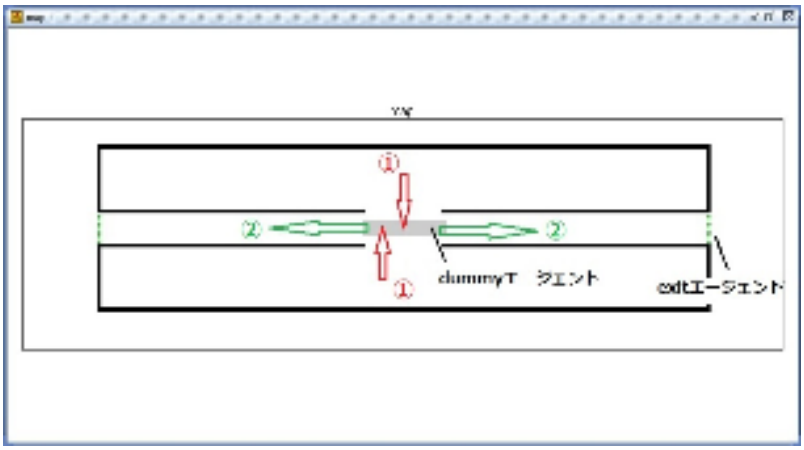


図 8　dummy エージェントの説明

2.3.5　adjust エージェント

大人数のシミュレーションを行った場合に，稀に図のように hito エージェントが廊下に出ることなく，部屋の隅に向かい残ってしまう現象が起こる．この現象を防ぐため，部屋の廊下側の隅に adjust エージェントを配置し，部屋の隅に残った hito エージェントを消す役割を果たす．wall エージェントの一部と同様，Excel の外部ファイルを初期設定に入力することで配置する．

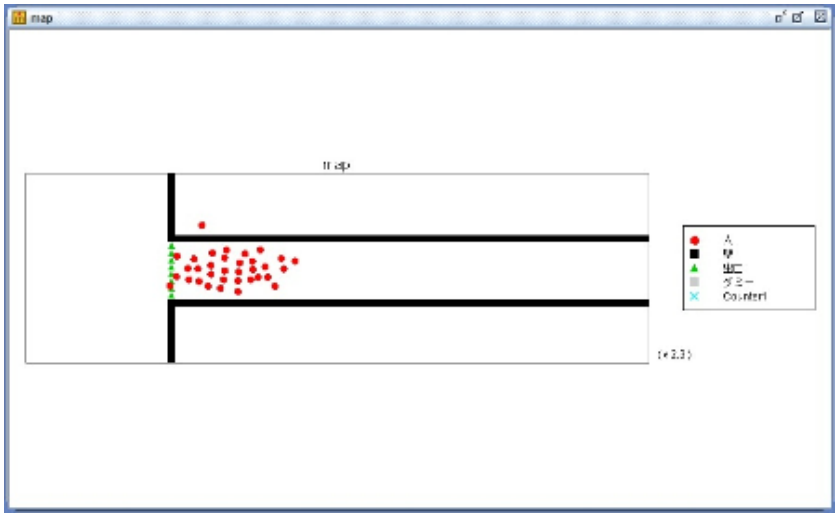


図 9　adjust エージェントの説明

2.3.6　counter エージェント

この counter エージェントを通過することによって，hito エージェントが持つブール型変数 switch5 が False から True に切り替わり，変数 switch5 が True となっている hito エージェントを数えることで，部屋から廊下へ避難した hito エージェントの数を時系列のグラフに出力する．重複を回避するため，数え終わった hito エージェントの変数 swtich5 は True から False に戻しておく．

3 章　数値例題

各モデルのシミュレーションの様子や結果について，以下で説明する．
また，

T(秒)：避難者全員が避難し終えるのにかかる時間，避難完了時間

N_r：部屋の数

P(人)：各部屋の避難者数の合計，総避難者数

p(人)：1 部屋あたりの避難者数

B(m)：室出口幅の合計

b(m)：1 部屋あたりの室出口幅

とする．

シミュレーション結果については，1 つのケースに対し 10 回以上試行した結果の平均値で算出し，excel の表やグラフで表す際に，単位を step から秒に変換した．

3.1　case1（部屋数：2，室出口位置：中央）の結果考察

case1 について，P=600 人，B=8m の場合でのシミュレーションの様子を図 10～図 15 に，また，室出口を通過した人数と避難が完了した人の人数の時系列グラフを図 16，図 17 に，パラメータ P，B を変えて行ったシミュレーション結果を表 2，図 18 に示す．

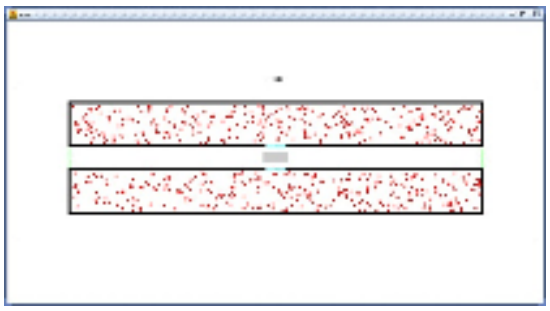


図 10　case1 のシミュレーションの様子（開始から 0 秒後）

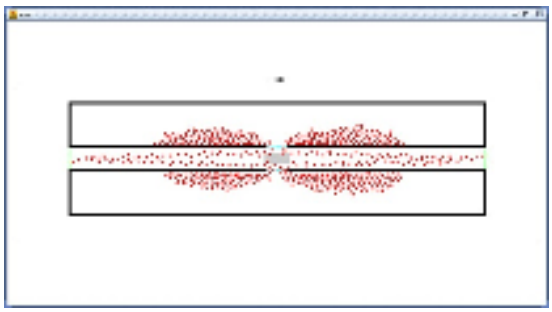


図 11　case1 のシミュレーションの様子（開始から 35 秒後）

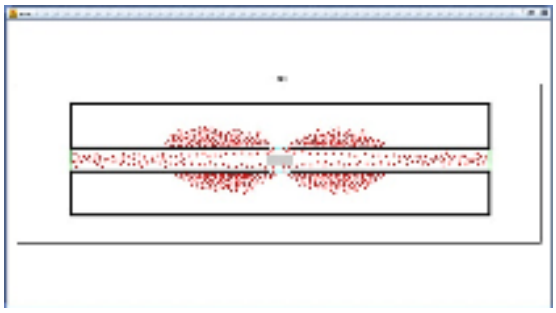


図 12　case1 のシミュレーションの様子
（開始から 56 秒後）

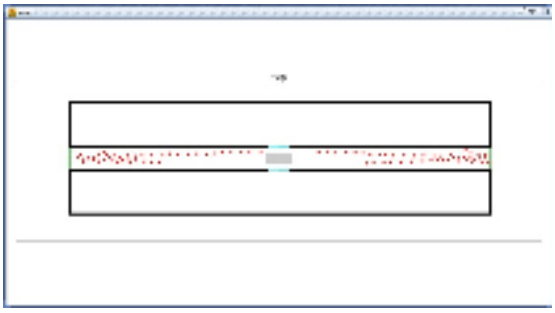


図 13　case1 のシミュレーションの様子
（開始から 187 秒後）

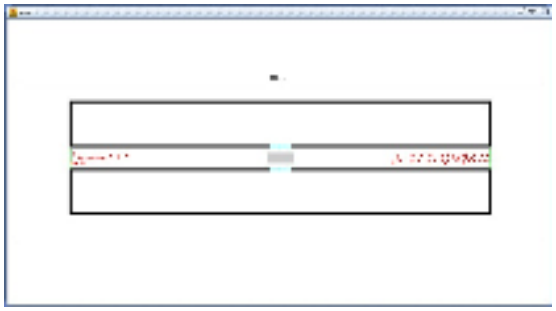


図 14　case1 のシミュレーションの様子
（開始から 201 秒後）

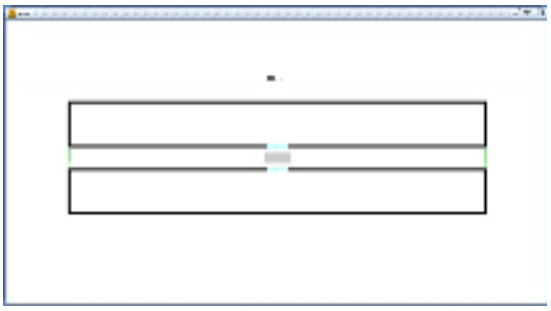


図 15　case1 のシミュレーションの様子
（開始から 231 秒後）

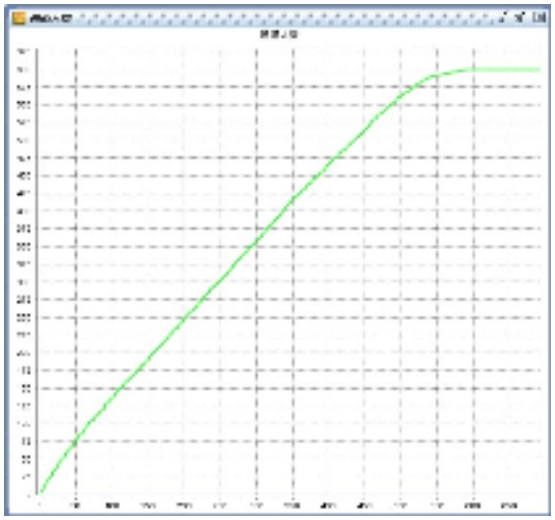


図 16　case1 の時系列グラフ
（室出口通過人数）

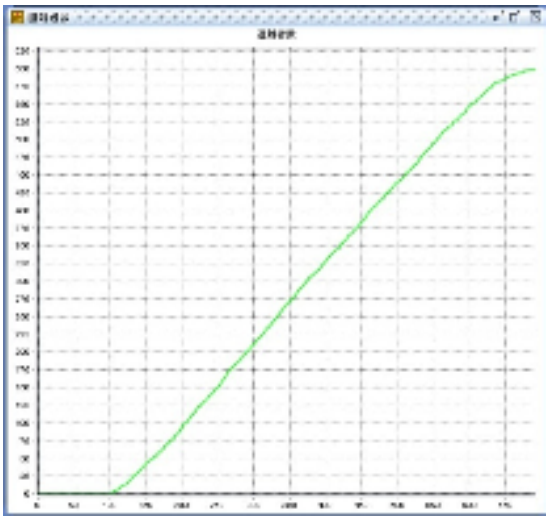


図 17　case1 の時系列グラフ
（避難完了人数）

表 2　case1 でのシミュレーション結果

case1									
		避難者数の合計 P(人)							
		P=100	P=200	P=300	P=400	P=500	P=600	P=800	P=1000
室出口幅合計 B(m)	2	113.7	180.3	240.1	317.5	380.5	453.6	583.3	715.9
	3	95.3	147.5	200	241.3	286.5	325.3	424.5	519.3
	4	92.7	133.1	175.2	208.0	246.9	286.0	373.8	445.2
	6	81.6	121.7	149.4	187.6	212.4	249.1	318.0	394.1
	8	78.6	109.3	135.8	170.9	191.6	227.2	284.6	343.1
	10	76.9	108.3	133.2	162.3	184.1	208.4	261.1	313.4
	12		104.6		152.5		204.3	250.2	301.4
	16		102.3		149.8		197.7	247.1	296.7
	20		102.8		149.0		197.3	247.4	297.6

（単位：秒）

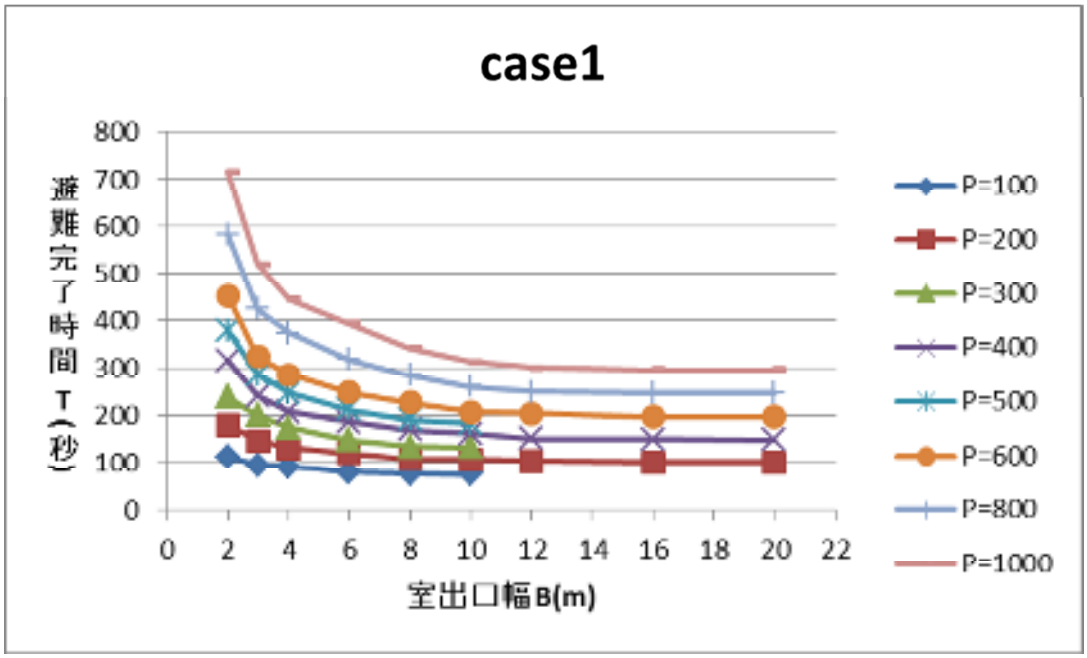


図 18　case1 のシミュレーション結果

はじめ，図 10 のように避難者はランダムに配置され，それぞれが避難を開始する．避難開始から少し経った後，避難を完了するものが出てくるが，図 11 のように，室出口では滞留がすでに発生している．そして，その後出口付近でも滞留が発生し（図 12），この出口付近の滞留は部屋の人間が全て廊下へ避難できた後にまで続く．避難者全員が部屋から避難できたあと，出口付近の滞留は次第に解消され，そのまま全員の避難が完了する．

図 18 からは，B の値が B=12m を超えると，避難完了時間 T の値はほぼ一定となることがわかる．また，B=12m より小さい範囲では，B の値が小さくなればなるほど急激に避難完了時間が長くなる．

3.2　case2（部屋数：4，室出口位置：端）の結果考察

case2 でのシミュレーションの様子と，室出口を通過した人数と避難が完了した人の人数の時系列グラフ，パラメータ P，B を変えて行ったシミュレーション結果を以下に記す．図 19 は，P=600 人，B=4m の場合，図 20 は，P=600 人，B=20m の場合のシミュレーションの途中の様子であり，時系列グラフは，P=200 人，B=12m の場合である．

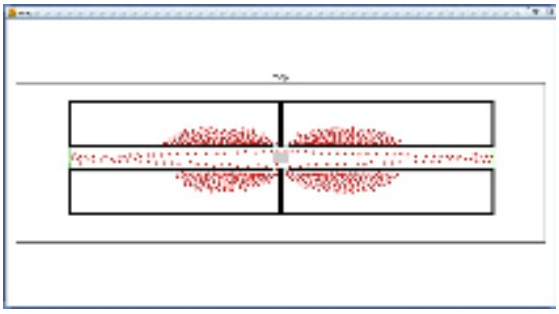


図 19　case2 のシミュレーションの様子
（P=600 人，B=4m の場合）

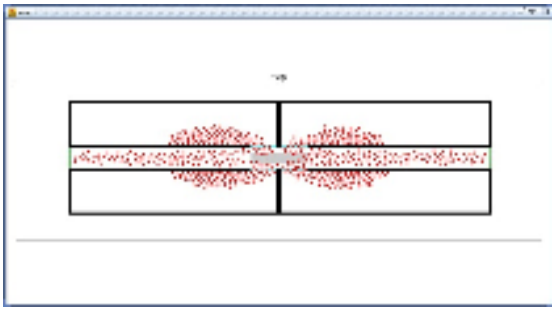


図 20　case2 のシミュレーションの様子
（P=600 人，B=20m の場合）

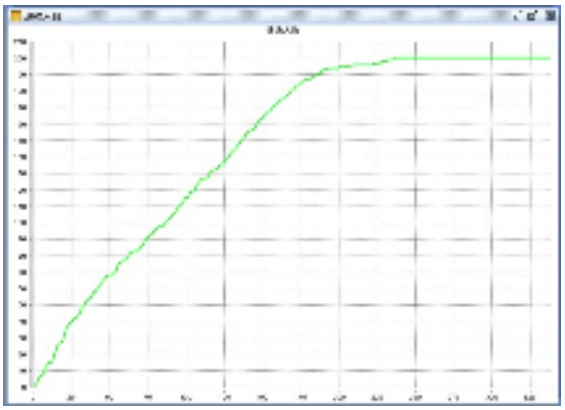


図 21　case2 の時系列グラフ
（室出口通過人数）

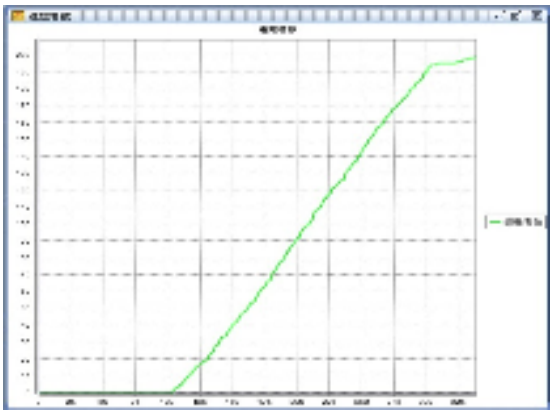


図 22　case2 の時系列グラフ
（避難完了人数）

表 3　case2 でのシミュレーション結果

case2						
		避難者数の合計 P(人)				
		P=200	P=400	P=600	P=800	P=1000
室出口幅合計 B(m)	4	123.4	187.6	265.9	338.5	411.4
	6	108.9	168.5	228.6	291.0	350.6
	8	102.8	157.5	210.4	264.3	318.3
	12	100.4	148.6	195.9	244.0	293.7
	16	98.2	144.7	192.3	243.0	291.5
	20	96.4	143.2	193.0	241.8	291.1

（単位：秒）

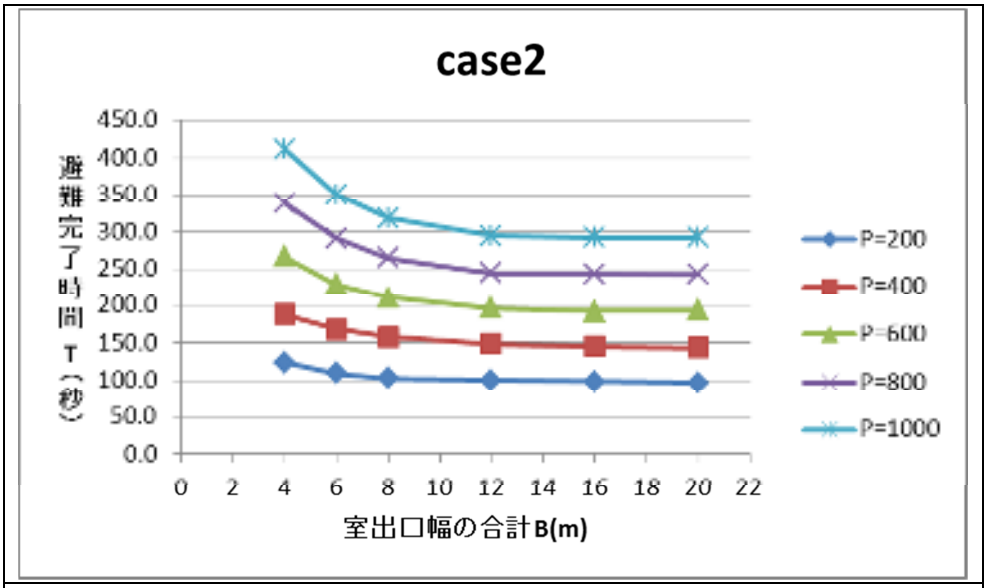


図 23　case2 のシミュレーション結果

図 19 を見ると，室出口幅が狭いと部屋内で滞留が発生していることがわかり，図 20 を見ると室出口幅が広いと廊下で滞留が発生することがわかる．図 23 からは，B≧12 の範囲で，避難完了時間 T の値がほぼ一定となっていることがわかる．

3.3　case3（部屋数：4，室出口位置：case2 と case4 の中間）の結果考察

case3 でのシミュレーションの様子と，室出口を通過した人数と避難が完了した人の人数の時系列グラフ，パラメータ P，B を変えて行ったシミュレーション結果を以下に記す．図 24 は，P=600 人，B=4m の場合，図 25 は，P=600 人，B=20m の場合のシミュレーションの途中の様子であり，時系列グラフは，P=200 人，B=12m の場合である．

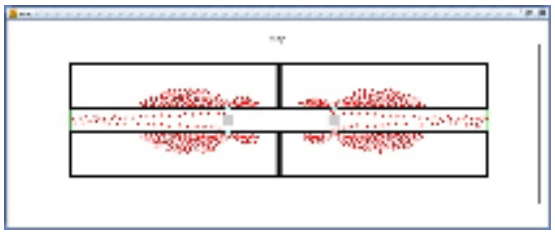


図 24　case3 のシミュレーションの様子
（P=600 人，B=4m の場合）

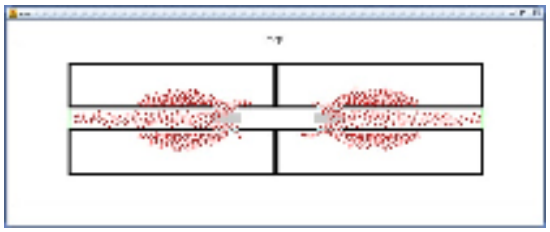


図 25　case3 のシミュレーションの様子
（P=600 人，B=20m の場合）

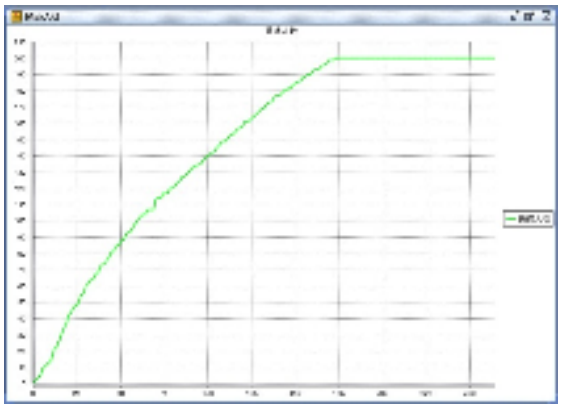


図 26　case3 の時系列グラフ
（室出口通過人数）

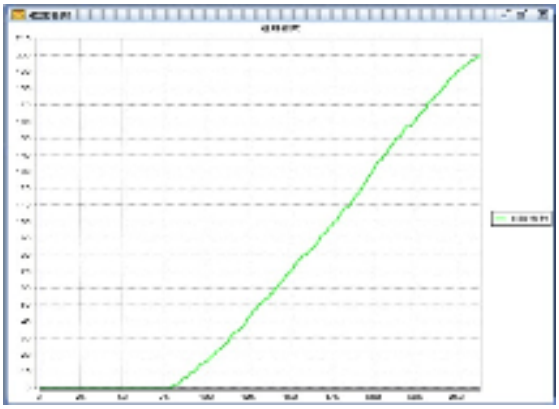


図 27　case3 の時系列グラフ
（避難完了人数）

表 4　case3 でのシミュレーション結果

case3						
		避難者数の合計 P(人)				
		P=200	P=400	P=600	P=800	P=1000
室出口幅合計 B(m)	4	109.1	180.6	256.5	327.1	403.2
	6	96.7	157.5	221.3	284.8	342.3
	8	92.2	145.7	202.6	247.6	310.9
	12	89.0	136.8	186.0	237.2	287.7
	16	87.5	135.8	184.7	233.2	283.0
	20	87.1	135.3	184.1	232.1	283.3

（単位：秒）

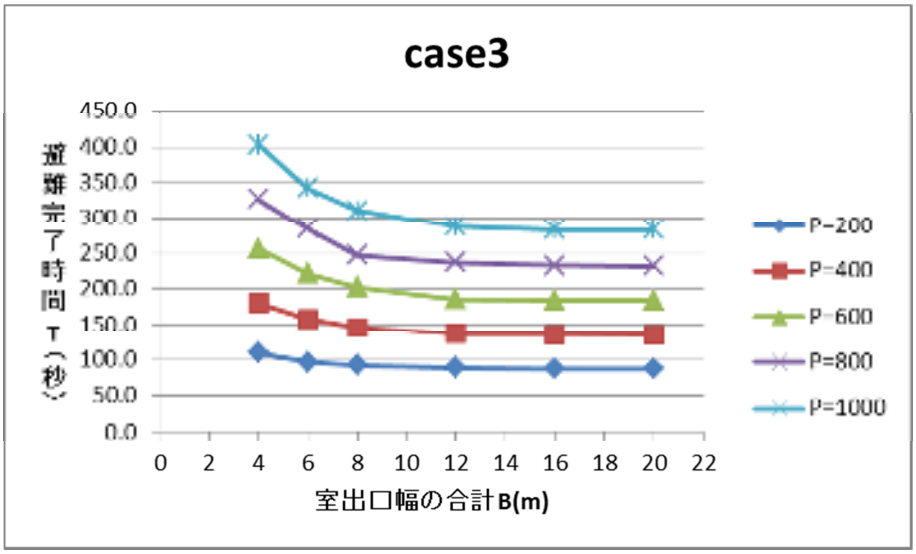


図 28　case3 のシミュレーション結果

case2 と同様，室出口幅が狭いと部屋内で滞留が発生し，室出口幅が広いと廊下で滞留が発生することがわかり，B≧12 の範囲では，避難完了時間 T の値はほぼ一定となっていることがわかる．

3.4　case4（部屋数：4，室出口位置：中央）の結果考察

case4 でのシミュレーションの様子と，室出口を通過した人数と避難が完了した人の人数の時系列グラフ，パラメータ P，B を変えて行ったシミュレーション結果を以下に記す．図 29 は，P=600 人，B=4m の場合，図 30 は，P=600 人，B=20m の場合のシミュレーションの途中の様子であり，時系列グラフは，P=200 人，B=12m の場合である．

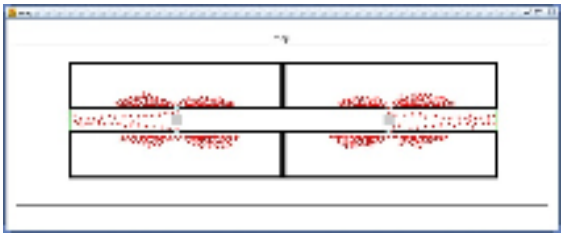


図 29　case4 のシミュレーションの様子
（P=600 人，B=4m の場合）

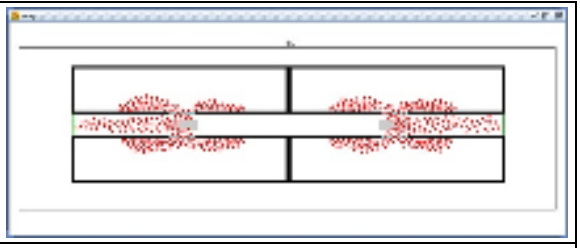


図 30　case4 のシミュレーションの様子
（P=600 人，B=20m の場合）

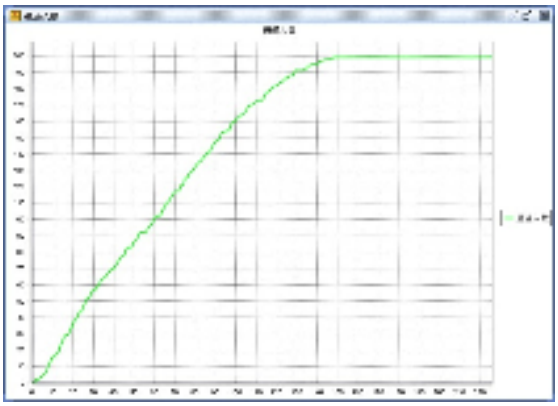


図 31　case4 の時系列グラフ
（室出口通過人数）

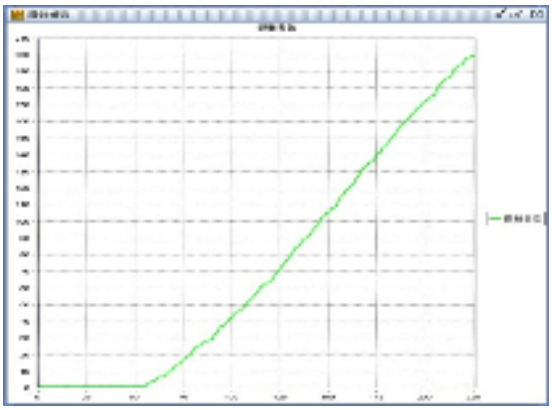


図 32　case4 の時系列グラフ
（避難完了人数）

表 5　case4 でのシミュレーション結果

case4						
		避難者数の合計 P(人)				
		P=200	P=400	P=600	P=800	P=1000
室出口幅合計 B(m)	4	97.4	169.8	239.3	307.1	378.0
	6	88.9	143.8	204.1	264.0	330.8
	8	79.3	126.2	176.8	235.0	277.8
	12	77.4	126.0	176.7	228.6	275.1
	16	75.1	124.3	173.3	222.9	275.0
	20	75.0	123.6	172.7	222.7	271.9

（単位：秒）

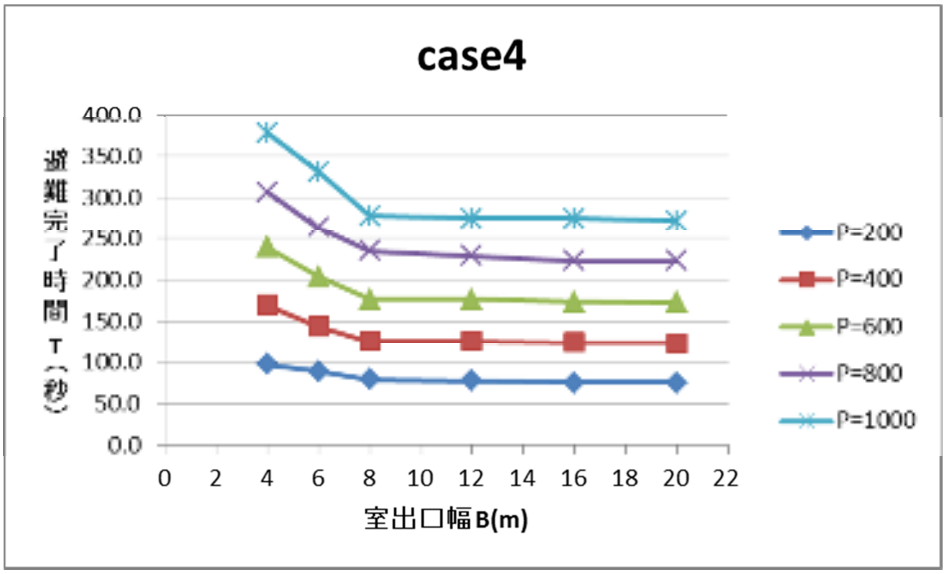


図 33　case4 のシミュレーション結果

case2 と同様，室出口幅が狭いと部屋内で滞留が発生し，室出口幅が広いと廊下で滞留が発生することがわかり，B≧12 の範囲では，避難完了時間 T の値はほぼ一定となっていることがわかる．

4 章　避難時間簡易予測式の提案

4.1　パラメータ毎のシミュレーション結果の考察

ｉ）　パラメータ P，B についての分析

前章で得られた結果をもとに，「室出口幅の合計」・「室出口の位置」・「総避難者数」のパラメータ毎に分析する．避難者数の合計 P を横軸に，室出口幅の合計 B の値で整理したグラフを図 34 に示す．また，それぞれの 1 次近似式をグラフ内に示す．

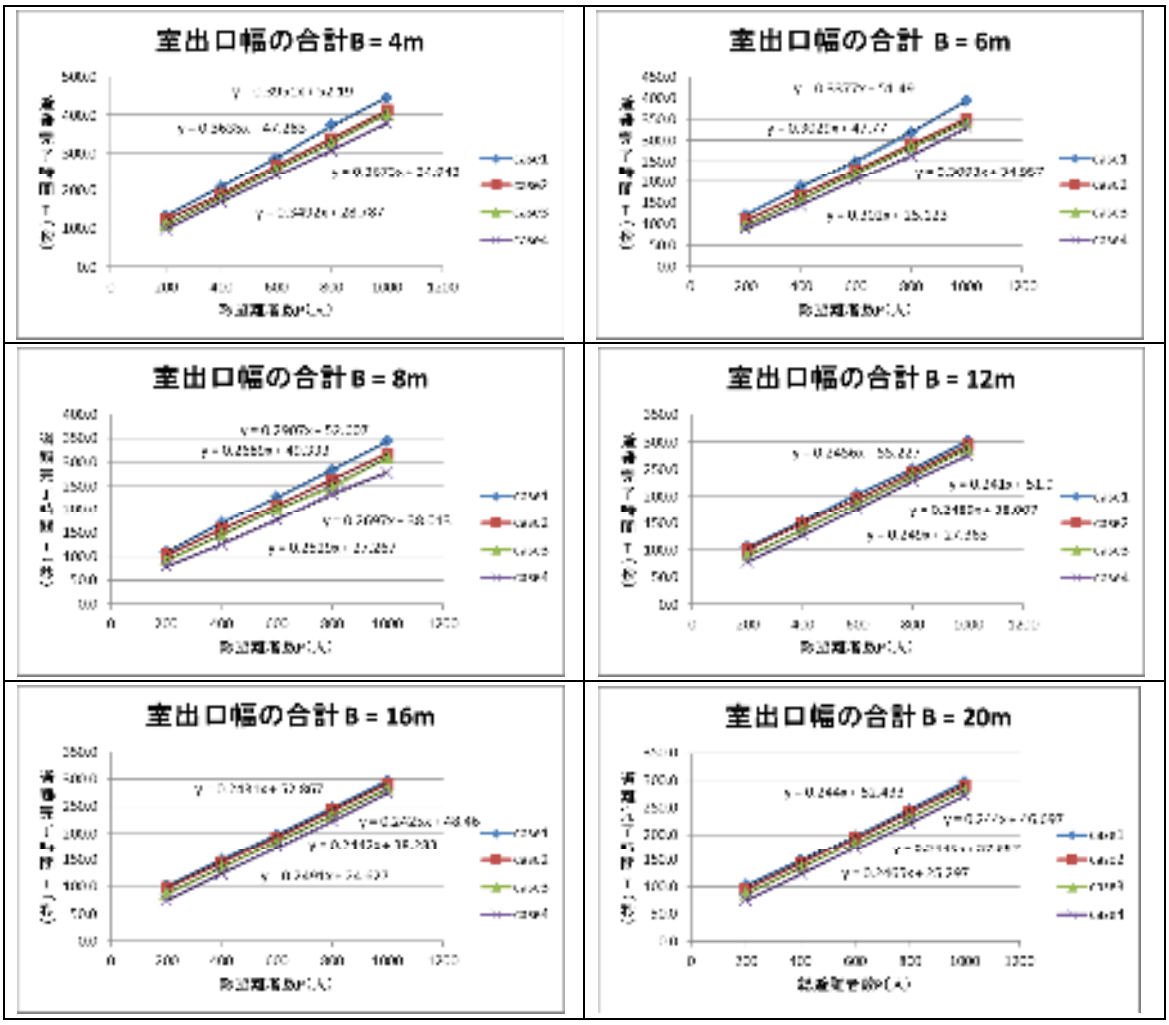


図 34　分析データ（パラメータ：P，B）

ii） パラメータ P，b についての分析

ｉ)と同様に分析を行い，避難者数の合計 P を横軸に，1 部屋あたりの室出口幅 b の値で整理したグラフを図 35 に示す．また，それぞれの 1 次近似式をグラフ内に示す．

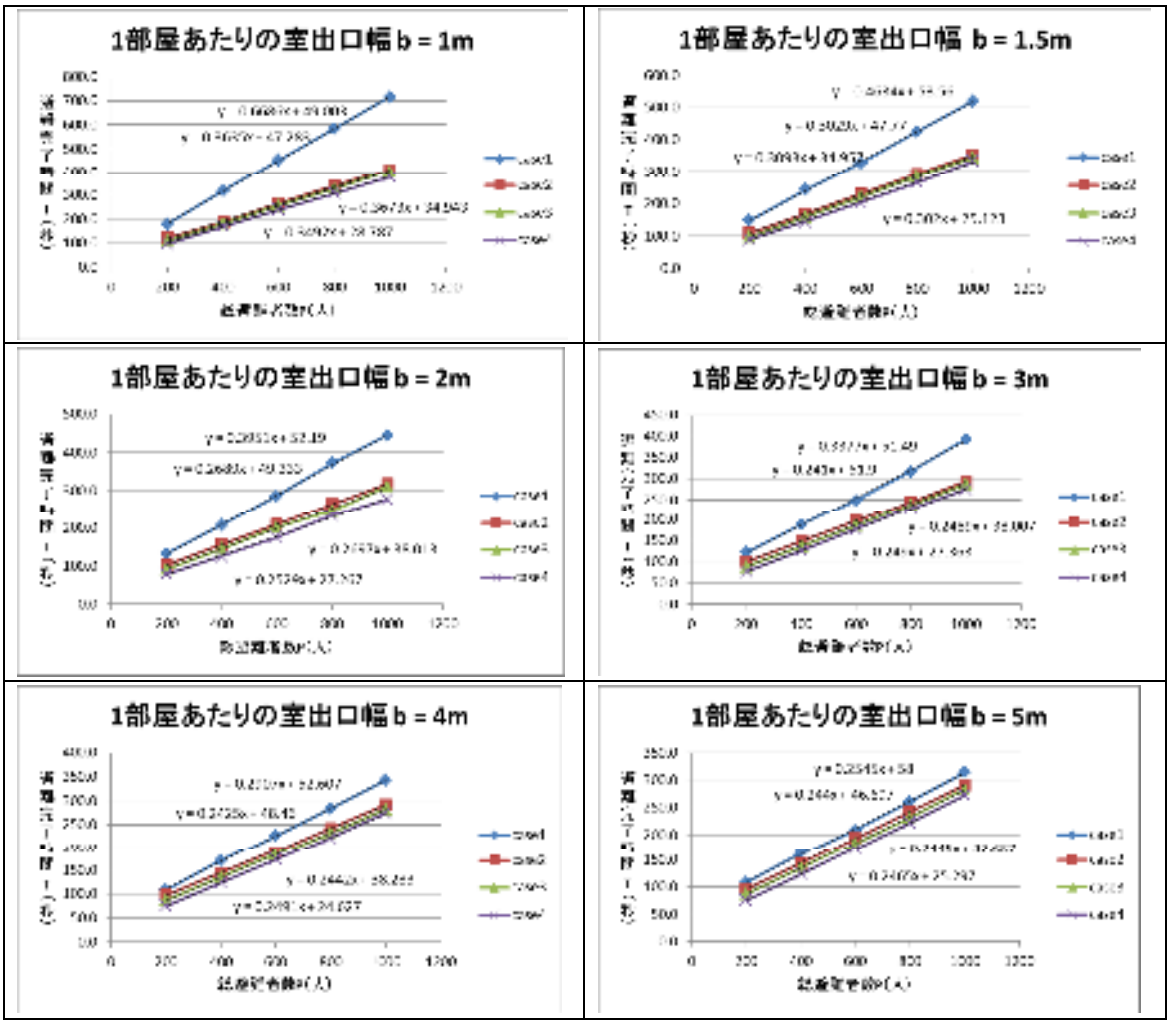


図 35　分析データ（パラメータ：P，b）

iii） パラメータ p，b についての分析

ｉ)と同様に分析を行い，1 部屋あたりの避難者数 p を横軸に，1 部屋あたりの室出口幅 b の値で整理したグラフを図 36 に示す．また，それぞれの 1 次近似式をグラフ内に示す．

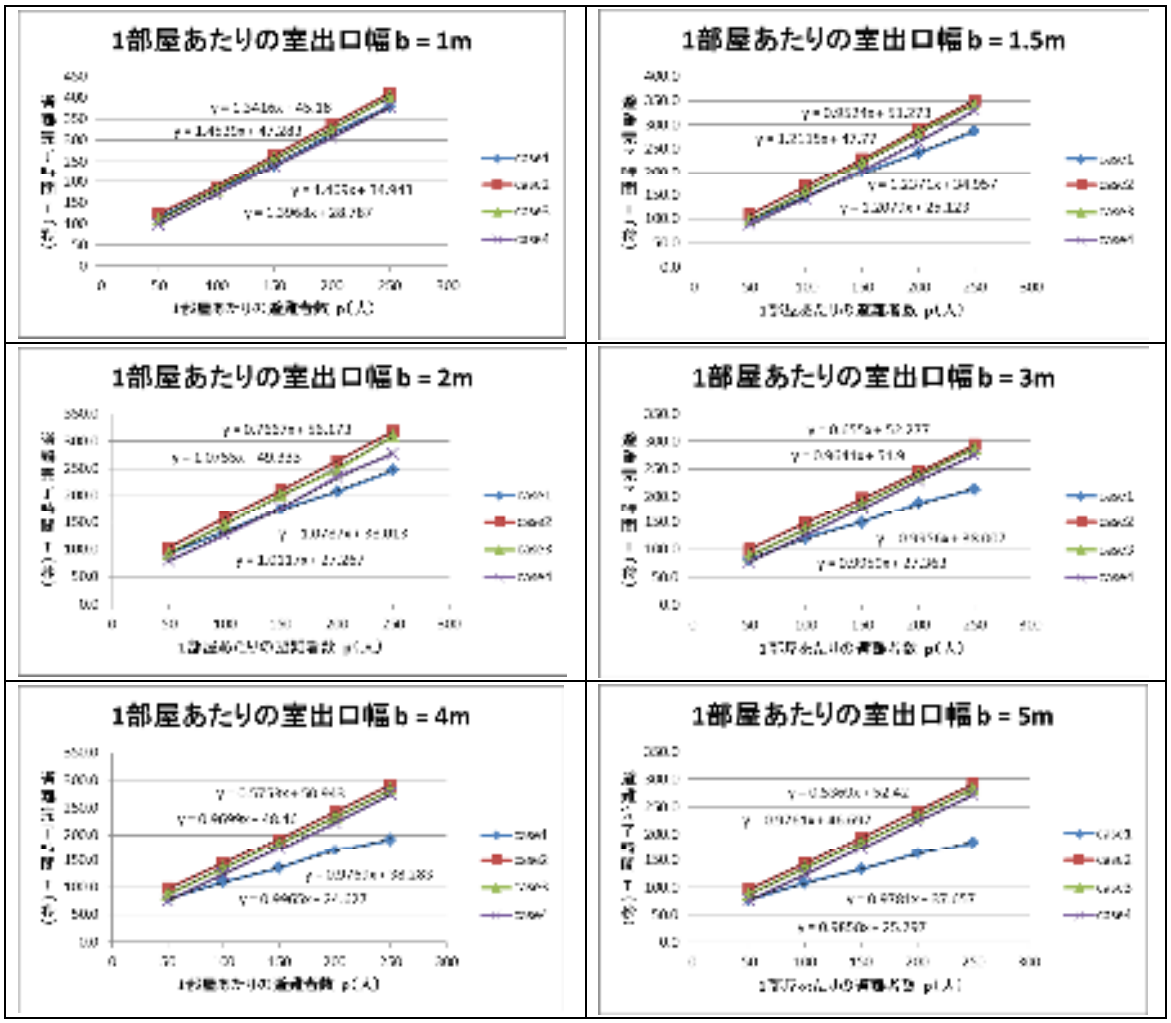


図 36　分析データ（パラメータ：p，b）

4.1.1　室出口幅

前章の考察より，室出口幅の合計 B の値が B=12m でグラフの形が大幅に変わることから，B≦12 の範囲と，B≧12 の範囲で簡易予測式が異なってくると推測できるため，各々分析する必要がある．

B≧12，または，b≧3 の範囲では，各モデルパターンの中で避難完了時間 T の値はほぼ一定となっていたことから，T を求める簡易式の中にパラメータ B は含まれないと推測される．B≦12，または，b≦3 の範囲では，B の値によって大幅に T の値が大幅に変わってくるため，T を求める簡易式は B の関数であると推測される．

4.1.2　室出口の位置

case2，case3，case4 の違いは室出口の位置のみであるが，前章のシミュレーション結果から，case2，case3，case4 の各モデルパターンにおいて避難完了時間 T の値は違っていることがわかる．そこで，

t_i(秒)：出口から最も距離がある位置から出口までの距離を標準速度(1.2m/s)で除した値とし，このt_iを「室出口の位置」を表すパラメータとして考える．t_iは各モデルにおいて値を求めることができ，case1，case2 の場合，t_1=t_2=67.7 秒，case3 の場合，t_3=51.4 秒，case4 の場合，t_4=35.3 秒である．

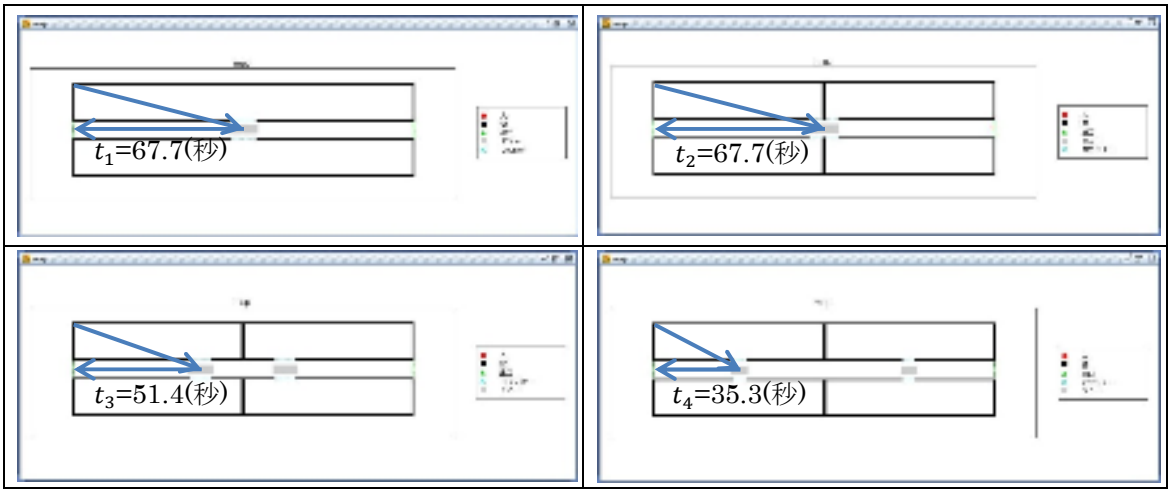


図 37　t_iについての説明

図～図の中のそれぞれの 1 次近似式を見ると，近似式の切片の値が case1 で一番大きく，case4 で一番小さいことがわかり，このことから，1 次近似式の切片の値はt_iに依存してくるのではないかと推測できる．

4.1.3　避難者数

図 34～図 36 のグラフを見てもわかるように，避難者数の合計 P，1 部屋あたりの避難者数 p を横軸にとったどのグラフでも避難完了時間 T を 1 次の近似式で表すことができる．

ここで，T を求める式を，

$$\mathrm{T} = \begin{cases} \alpha_1 \times P + \beta_1 \\ \alpha_2 \times p + \beta_2 \end{cases} \tag{1}$$

とおく．

4.2　簡易予測式の構築

ｉ）　パラメータ P，B による考察

(1)式の P の係数α_1，切片β_1を求めるために，図 34 で得た 1 次近似式の横軸 P の係数と切片だけを取り出し，グラフにしたものを下の図 38，図 39 に示し，さらに，切片をt_iで除したものを図 40 に示す.

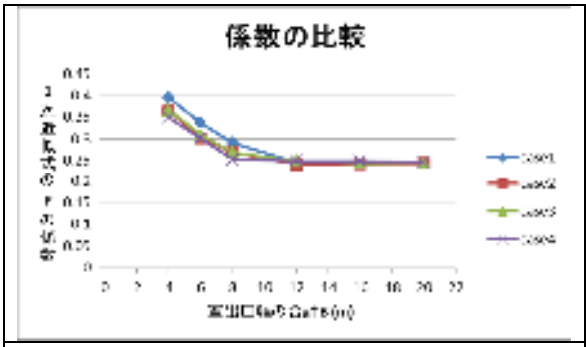


図 38　係数の比較（パラメータ：P，B）

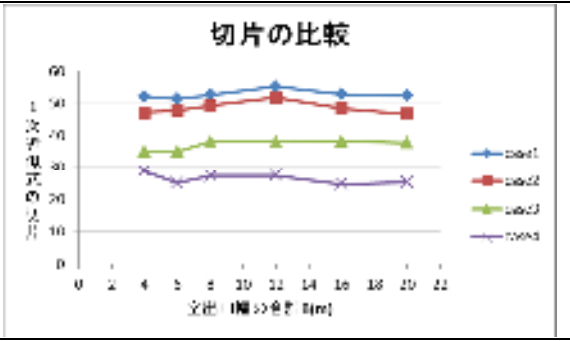


図 39　切片の比較（パラメータ：P，B）

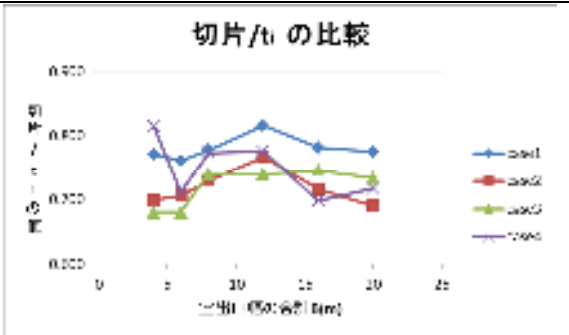


図 40　切片$/t_i$の比較（パラメータ：P，B）

・α_1を求める

B≦12 の範囲での図 38 のグラフの case1～case4 の平均値を算出する．そして，グラフの形状から

$$\alpha_1 = \mathrm{x} \times (12 - B)^y + z \tag{2}$$

と仮定し，最小二乗法を用いて，x，y，z の値を導き出し，α_1を求める．算出した平均値のデータから得たグラフと最小二乗法を用いて得られた近似式のグラフを下の図 41，図 42 に示す.

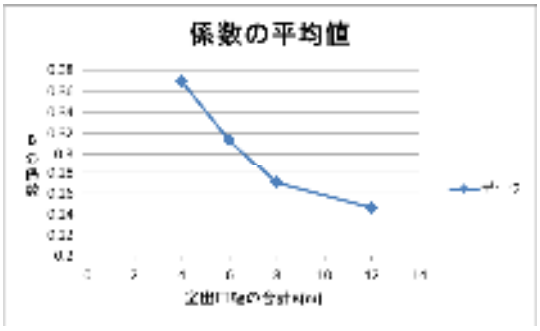


図 41　データより得られたグラフ

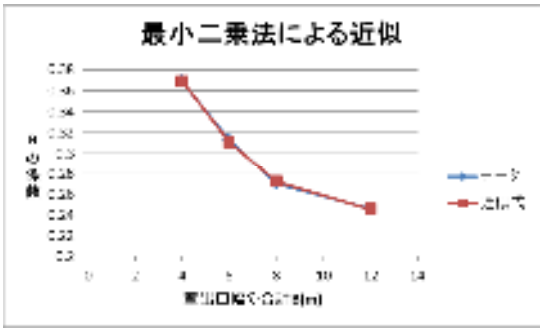


図 42　最小二乗法により得られたグラフ

最小二乗法を用いた結果，

$$\alpha_1 = 0.0013 \times (12 - B)^{2.21} + 0.25 \quad (B \leq 12) \tag{3}$$

という式が得られた．また，図 38 より，B ≥ 12の範囲では，αの値は一定値である 0.25 に収束しているため，

$$\alpha_1 = 0.25 \quad (\mathrm{B} \geq 12) \tag{4}$$

という式が得られる．

・$\boldsymbol{\beta_1}$を求める

図 47，図 48 を見ると，どちらのグラフでも切片をt_iで除した値が 0.7~0.8 の範囲でおさまっていることがわかる．このことから，

$$\beta_1 = 0.8 \times t_i \tag{5}$$

という式が得られる．t_iの係数を 0.8 にしたのは，避難時間の予測式は大きい値で見積もっていたほうが，安全性を考慮できるというところに因る．

以上の(1)式～(5)式より，(3)式～(5)式を(1)式に代入することで．避難完了時間 T の簡易予測式は，

$$\mathrm{T} = \begin{cases} \{0.0013 \times (12 - B)^{2.21} + 0.25\} \times P + 0.8 \times t_i & (B \leq 12) \\ 0.25 \times P + 0.8 \times t_i & (B \geq 12) \end{cases} \tag{6}$$

と提案できる．

ii） パラメータ P，b による考察

(1)式の P の係数α_1，切片β_1を求めるために，図 35 で得た 1 次近似式の横軸 P の係数と切片だけを取り出し，グラフにしたものを下の図 43，図 44 に示し，さらに，切片をt_iで除したものを図 45 に示す.

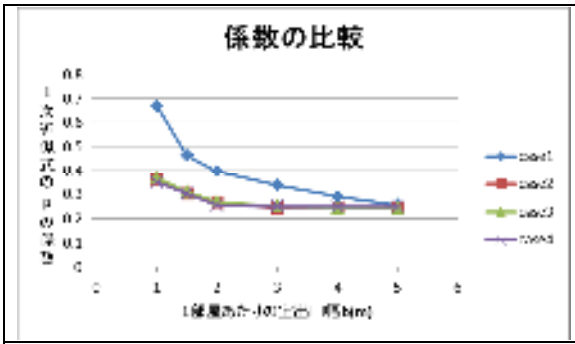


図 43　係数の比較（パラメータ：P，b）

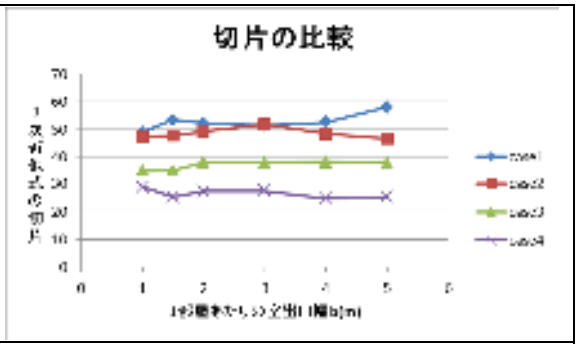


図 44　切片の比較（パラメータ：P，b）

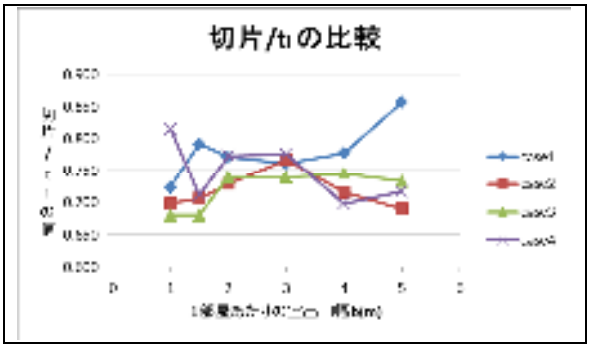


図 45　切片/t_iの比較（パラメータ：P，b）

切片β_1に関しては，図 45 を見てもわかるように，切片をt_iで除した値が 0.7~0.8 の範囲でおさまっていることがわかる．しかし，避難者数の合計 P の係数α_1について，case1 の場合は，次第に一定の値に収束するという性質が似てはいるが，他の case2~case4 の場合との差が無視できないものとなっている.

このことは，case1 の場合について，他の case2~case4 の場合と比べて 1 部屋あたりの避難者数が多い（p が 2 倍）ことによるものであり，一定の値に収束するのは，1 部屋あたりの室出口幅 b の値が大きくなるとどの場合でも滞留が発生しなくなるからだと推測できる.

以上のことから，避難者数の合計 P と 1 部屋あたりの室出口幅 b での分析は適切ではないと考えられる．そこで，次に 1 部屋あたりの避難者数 p と 1 部屋あたりの室出口幅 b での分析を行う.

iii） パラメータ p，b による考察

(1)式の P の係数α_2，切片β_2を求めるために，図 36 で得た 1 次近似式の横軸 P の係数と切片だけを取り出し，グラフにしたものを下の図 46，図 47 に示し，さらに，切片をt_iで除したものを図 48 に示す.

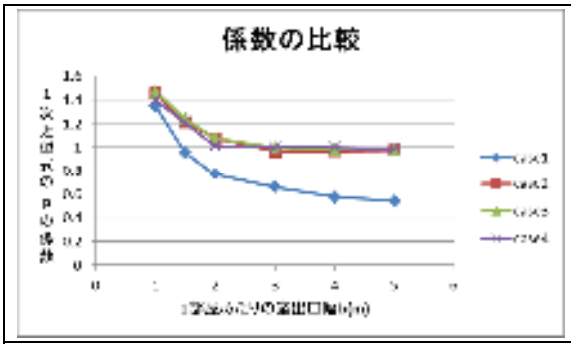


図 46　係数の比較（パラメータ：p，b）

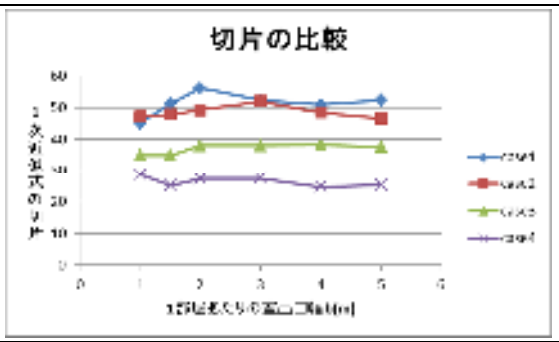


図 47　切片の比較（パラメータ：p，b）

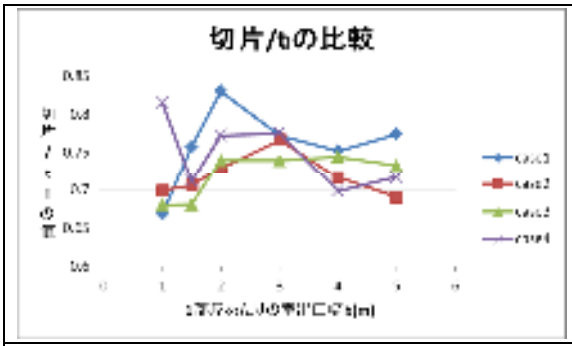


図 48　切片/t_iの比較（パラメータ：p，b）

切片β_2に関して，ⅱ)のパラメータ P, b での分析同様, 図 48 を見てもわかるように, b=1, b=2 でわずかながらはずれてはいるものの，切片をt_iで除した値が 0.7~0.8 の範囲でおさまっていることがわかる．しかし，1 部屋あたりの避難者数 p の係数α_2について，case1 の場合は，ⅰ)やⅱ)の場合とは違い，1 部屋あたりの室出口幅 b の値が大きくなっても一定の値に収束せず，他の case2~case4 の場合との差はどんどん拡がっている.

このことは，case1 の場合について，他の case2~case4 と比べて避難者数の合計が少ない（P が半分）ことによるものであり，b の値が大きくなるほど case1 の場合と case2~case4 の場合との差が大きくなるのは，両者の避難者数の合計 P の値の差が大きくなるからであると推測できる.

以上のことから，1 部屋あたりの避難者数 p と 1 部屋あたりの室出口幅 b での分析は適切ではないと考えられる.

4.3　定性的な分析によるアプローチ

部屋内で滞留が発生した場合，滞留が解消されるまで部屋から廊下への流出人数は一定になり，その流出人数は室出口幅に依存すると推測される．そこで，

・p(b)（人/秒）：室出口幅 b(m)の場合の単位時間あたりの流出人数

とし，この p(b)を用いて避難完了時間 T を求める式を提案する．

避難が開始してから，避難が完了するまでは，

① 避難開始位置から室出口へ向かう　　③ 滞留が解消する

② 滞留が発生する　　④ 室出口から出口へと避難する

という流れであり①から②で費やす時間と③から④で費やす時間の合計は，t_iで表すことができる．ここで，滞留が発生してから解消されるまでの時間 t は，

$$\mathrm{t} = \frac{p}{p(b)} \tag{7}$$

と表すことができ，避難完了時間 T′の式は，

$$\mathrm{T}' = \frac{p}{p(b)} + t_i \tag{8}$$

である．

次に，case1 のモデルを使って p(b)を求める．室出口の通過人数を表した時系列グラフをエクセルで出力し，曲線の最初と最後を除いた値域（直線部分）での 1 次近似式の係数を，1 部屋あたりの室出口幅 b をパラメータとしてデータをとる．さらに，得られたデータのグラフから最小二乗法を用いて近似することで p(b)の式とする．

データより得られたグラフと最小二乗法を用いて得られたグラフを図 49，図 50 に示す

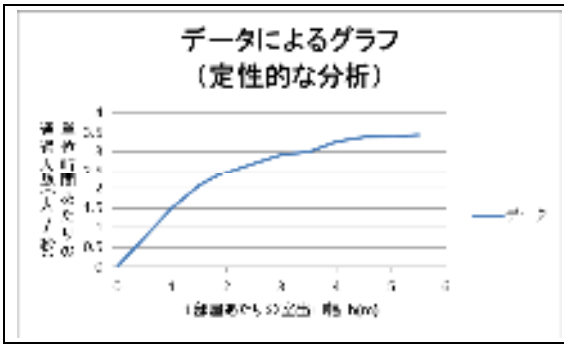


図 49　データより得られたグラフ

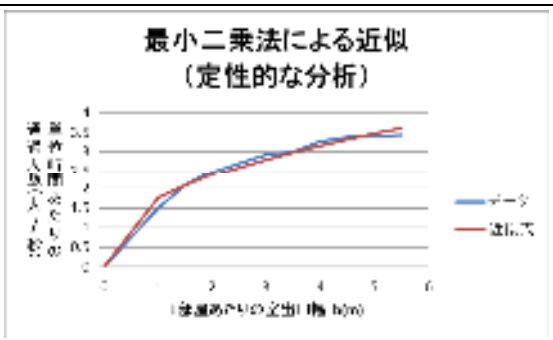


図 50　最小二乗法により得られたグラフ

最小二乗法を用いた結果，

$$\mathrm{p(b)} = 1.76 \times b^{0.42} \tag{9}$$

という式が得られた．さらに，(9)式を(8)式に代入すると，避難完了時間 T′の式は，

$$\mathrm{T}' = \frac{p}{1.76 \times b^{0.42}} + t_i \tag{10}$$

と提案できる．そこで，定性的な分析から得られた(10)式とシミュレーションデータから得られた(6)式を比較し，グラフに示したものが図 51，図 52 である．両図とも case4 での結果であり，図 51 は p=50 人，図 52 は p=250 人の場合である．

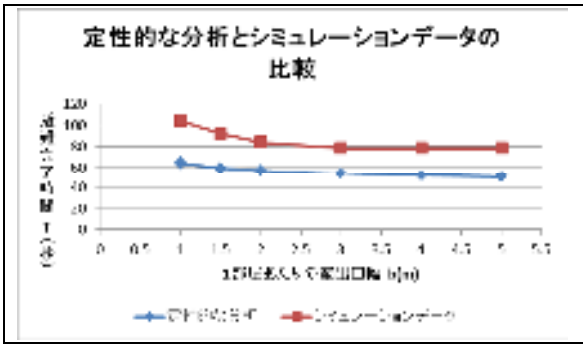


図 51　シミュレーションによる結果と定性的な分析による比較（p=50 人）

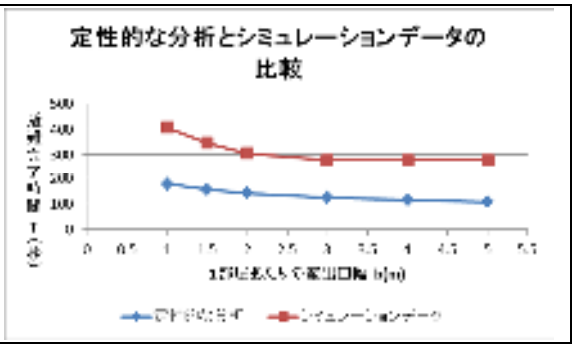


図 52　シミュレーションによる結果と定性的な分析による比較（p=250 人）

図 51，図 52 を見ると，どちらの場合でも，定性的な分析によって得られた T′の値よりもシミュレーションデータから得られた T の値のほうが大きいことがわかる．このことは，定性的な分析では，避難開始から避難完了までを標準速度（1.2m/s）のみで考えているからだと推測できる．また，避難者数が少ないほど T と T′の差が小さいことからも，避難者数が少ないと滞留の規模が小さくなり，滞留による避難速度の減少の影響が小さいと考えられるため，避難完了時間の差が生じる原因が避難速度によるものだと推測できる．

5 章　結論

本研究での成果は以下の通りである.

1) 構造計画研究所が開発したソフトウェアである artisoc academic 3.0 を用いて，前方の滞留によって速度を減少させながら，人や壁を避けて進む避難行動のルールを構築し，仮想的なフロアモデルを作成した.
2) 作成したモデルを用いて, パラメータ毎のシミュレーションを行い, 分析を行った.
3) シミュレーションから得られたデータと分析をもとに, パラメータ毎の考察を行い, 避難時間を予測できる簡易式を提案した.
4) 定性的な分析を行い，シミュレーションから得られた簡易式と比較し，考察した.

本研究のフロアモデルは，各部屋の避難者数と室出口幅が同じであり，その条件のもとでしか提案した簡易予測式が適用できない．そのため，提案した簡易予測式を実存する建物に適用するのは現段階では現実的ではなく，条件を複雑にしたモデルでの分析・検証が今後の課題として残されている.

謝辞

本研究を進めていくにあたり，ご多忙の中，丁寧なご指導とご鞭撻を賜りました吉富信太准教授や向坊恭介助教授に厚く感謝の意を表します．研究に関することだけではなく，公私にわたって様々な助言を頂き，また，プライベートに関する事情も考慮して頂き，本当にありがとうございました．

そして，共に研究を進めてきた構造デザイン研究室のメンバーには多くのご協力を頂きました．忙しい時期にも関わらず，研究以外の仕事でも積極的に取り組んでくれた B4・M1 の後輩の方々，研究のことだけなく様々な相談にのってくれた M2 の方々，本当にありがとうございました．

最後に，この論文を完成するにあたり，様々な方々にご協力頂きました．ここに記して謝意を表します．

2012 年 3 月

倉田　佳幸

参考文献

1) 今村文彦，鈴木介，谷口将彦：津波避難数値シミュレーション法の開発と北海道奥尻島青苗地区への適用，自然災害科学　vol.20, No.2, pp.183-195, 2001-08-31
2) 鈴木介，今村文彦：住民意識・行動を考慮した津波避難シミュレーションモデル，自然災害科学　vol.23, No.4, pp.521-538, 2005-02-28
3) 金用哲，高井伸雄：集中豪雨を想定した地域避難計画に関する研究　―避難シミュレーションの構築によるソフト対策の提案を中心に―，学術講演梗概．F-1，都市計画，建築経済・住宅問題　2009, pp.467-468, 2009-07-20
4) 円谷信一，西野智研，田中哮義：自家用車利用避難のモデル化　都市災害を想定した避難シミュレーションシステムの開発，学術講演梗概集．A-2，防火，海洋，情報システム技術　2010, pp.257-258, 2010-07-20
5) 高瀬大樹，円満隆平，佐野友紀，渡辺仁史：歩行者動線シミュレーション，日本建築学会技術報告集　No.3，pp.263-267, 1996-12
6) 尾崎匠，長名優子：マルチエージェントを用いた避難シミュレーションによるオフィスレイアウト案の評価，全国大会後縁論文集 vol.71, No.2, pp.2-173, pp.2-174,2009-03-10
7) 城田拓耶，堤和敏：マルチエージェントによる避難計画を踏まえた教室配置の検証～芝浦工業大学大宮キャンパス新 2 号館対象～
8) 渡辺公次郎，近藤光男：津波防災まちづくり計画支援のための津波避難シミュレーションモデルの開発　日本建築学会計画系論文集 vol.74, No.637, pp.627-634, 2009-03
9) 安福健祐：地図データベースを用いた広域避難シミュレーションシステムの開発　学術講演梗概集, A-2, 防災, 海洋, 情報システム技術 2009, pp.515-516, 2009-07-20
10) 玉川奈都子，大畑大志郎，高井伸雄，鏡味洋史：マルチエージェントシミュレーションによる釧路市中心部の津波からの避難シミュレーション　日本建築学会北海道支部研究報告集　No.79, pp.175-178, 2006-07-01
11) 山影進：人工社会構築指南―artisoc によるマルチエージェント・シミュレーション入門―　書籍工房早山
12) 日本建築センター：新・建築防災計画指針―建築物の防火・避難計画の解説書―

付録

付１　Universe のルールプログラム

・case1

```
Univ_Init{
Dim i As Integer
Dim j As Integer
Dim k As Integer
Dim WX1 As Double
Dim WY1 As Double
Dim WX2 As Double
Dim WY2 As Double
Universe.counter1 = 0

//部屋に人を配置
For k = 0 To Universe.ninzu1 - 1
  WX1 = Rnd() * 159 + 21
  WY1 = Rnd() * 15 + 37
  CreatePeople(WX1, WY1)
Next k

For k = 0 To Universe.ninzu1 - 1
  WX2 = Rnd() * 159 + 21
  WY2 = Rnd() * 15 + 11
  CreatePeople(WX2, WY2)
Next k

//出口エージェントの設置
For k = 25 To 35
  BuildExit(20, k)
  BuildExit(181, k)
Next k

//廊下の壁を設置
```

```
For k = 21 To 100 - Universe.door / 2
  BuildWall(k, 27)
  BuildWall(k, 36)
Next k
For k = 101 + Universe.door / 2 To 180
  BuildWall(k, 27)
  BuildWall(k, 36)
Next k

//ダミーの配置
For i = 100 - Universe.door / 2 To 101 + Universe.door / 2
  For k = 30 To 33
    BuildDummy(i, k)
  Next k
Next i

//カウンターの配置
For k = 101 - Universe.door / 2 To 100 + Universe.door / 2
  BuildCounter1(k, 27)
  BuildCounter1(k, 36)
Next k
}

Univ_Step_Begin{

}

//時系列での室出口の通過人数を出力
Univ_Step_End{
OpenFileCSV("output.txt", 2, 3)
WriteFileCSV(2, Universe.counter1, False)
WriteFileCSV(2, GetCountStep(), True)
CloseFileCSV(2)

//室出口の通過人数を 1step 毎にカウント
Dim one As Agt
```

```
Dim people As Agtset
MakeAgtsetSpace(people, Universe.map)
For each one in people
  If one.switch5 == True Then
    Universe.counter1 = Universe.counter1 + 1
    one.switch5 = False
  End if
Next one

//避難者がいなくなったら避難完了
If CountAgt(Universe.map.hito) == 0 Then
  ExitSimulation()
End if
}

Univ_Finish{
PrintLn(GetCountStep() & " steps"&"で避難完了")
}

Sub CreatePeople(XX As Double, YY As Double){
Dim obj As Agt
obj=CreateAgt(Universe.map.hito)
obj.X=XX
obj.Y=YY
}

Sub BuildWall(XX As Double, YY As Double){
Dim obj1 As Agt
obj1 = CreateAgt(Universe.map.wall)
obj1.X = XX
obj1.Y = YY
}

Sub BuildExit(XX As Double, YY As Double){
Dim obj2 As Agt
obj2 = CreateAgt(Universe.map.exit)
```

```
obj2.X = XX
obj2.Y = YY
}

Sub BuildDummy(XX As Double, YY As Double){
Dim obj3 As Agt
obj3=CreateAgt(Universe.map.dummy)
obj3.X=XX
obj3.Y=YY
}

Sub BuildCounter1(XX As Double, YY As Double){
Dim obj4 As Agt
obj4 = CreateAgt(Universe.map.counter1)
obj4.X = XX
obj4.Y = YY
}
```

・case2～case4

```
Univ_Init{
Dim i As Integer
Dim j As Integer
Dim k As Integer
Dim WX1 As Double
Dim WY1 As Double
Dim WX2 As Double
Dim WY2 As Double
Dim WX3 As Double
Dim WY3 As Double
Dim WX4 As Double
Dim WY4 As Double
Universe.counter1 = 0

//部屋に人を配置
For k = 0 To Universe.ninzu1 - 1
```

```
  WX1 = Rnd() * 78 + 21
  WY1 = Rnd() * 15 + 37
  CreatePeople(WX1, WY1)
Next k

For k = 0 To Universe.ninzu1 - 1
  WX2 = Rnd() * 78 + 102
  WY2 = Rnd() * 15 + 37
  CreatePeople(WX2, WY2)
Next k

For k = 0 To Universe.ninzu1 - 1
  WX3 = Rnd() * 78 + 21
  WY3 = Rnd() * 15 + 11
  CreatePeople(WX3, WY3)
Next k

For k = 0 To Universe.ninzu1 - 1
  WX4 = Rnd() * 78 + 102
  WY4 = Rnd() * 15 + 11
  CreatePeople(WX4, WY4)
Next k

//出口エージェントの設置
For k = 25 To 35
  BuildExit(20, k)
  BuildExit(181, k)
Next k

//廊下の壁を設置（各 case で異なる）
For j = 11 To 27
  BuildWall(100, j)
  BuildWall(101, j)
Next j

For j = 36 To 52
```

```
  BuildWall(100, j)
  BuildWall(101, j)
Next j

For i = 21 To 99 - Universe.door
  BuildWall(i, 27)
  BuildWall(i, 36)
Next i

For i = 102 + Universe.door To 180
  BuildWall(i, 27)
  BuildWall(i, 36)
Next i

//ダミーの配置（各 case で異なる）
For i = 100 - Universe.door To 101 + Universe.door
  For k = 30 To 33
    BuildDummy(i, k)
  Next k
Next i

//カウンターの配置（各 case で異なる）
For i = 100 - Universe.door To 99
  BuildCounter1(i, 27)
  BuildCounter1(i, 36)
Next i

For i = 102 To 101 + Universe.door
  BuildCounter1(i, 27)
  BuildCounter1(i, 36)
Next i
}

Univ_Step_Begin{

}
```

```
//時系列での室出口の通過人数を出力
Univ_Step_End{
OpenFileCSV("output.txt", 2, 3)
WriteFileCSV(2, GetCountStep(), False)
WriteFileCSV(2, Universe.counter1, True)
CloseFileCSV(2)

//室出口の通過人数を 1step 毎にカウントする
Dim one As Agt
Dim people As Agtset
MakeAgtsetSpace(people, Universe.map)
For each one in people
  If one.switch5 == True Then
    Universe.counter1 = Universe.counter1 + 1
    one.switch5 = False
  End if
Next one

//避難者がいなくなったら避難完了
If CountAgt(Universe.map.hito) < 0 Then
  ExitSimulation()
End if
}

Univ_Finish{
PrintLn(GetCountStep() & " steps"&"で避難完了")
}

Sub CreatePeople(XX As Double, YY As Double){
Dim obj As Agt
obj=CreateAgt(Universe.map.hito)
obj.X=XX
obj.Y=YY
}
```

```
Sub BuildWall(XX As Double, YY As Double){
Dim obj1 As Agt
obj1 = CreateAgt(Universe.map.wall)
obj1.X = XX
obj1.Y = YY
}

Sub BuildExit(XX As Double, YY As Double){
Dim obj2 As Agt
obj2 = CreateAgt(Universe.map.exit)
obj2.X = XX
obj2.Y = YY
}

Sub BuildDummy(XX As Double, YY As Double){
Dim obj3 As Agt
obj3=CreateAgt(Universe.map.dummy)
obj3.X=XX
obj3.Y=YY
}

Sub BuildCounter1(XX As Double, YY As Double){
Dim obj5 As Agt
obj5 = CreateAgt(Universe.map.counter1)
obj5.X = XX
obj5.Y = YY
}
```

付 2　各エージェントのルールプログラム

・case1

(1) hito エージェント

```
Agt_Init{
My.Direction = Rnd() * 360
My.speed = 0.8
My.switch1 = False
My.switch2 = False
My.switch3 = False
My.switch4 = False
My.switch5 = False
}

Agt_Step{
Dim i As Integer
Dim deg As Double

If My.Y > 36 Then
  My.switch1 =True
End if

My.speed = 0.8

//まずダミーへ向かう
If My.switch1 == True Then
  My.Direction = GetDirection(My.X, My.Y, 100.5, 31, Universe.map)
Else
  My.Direction = GetDirection(My.X, My.Y, 100.5, 31, Universe.map)
End if

//出口に向かう
If My.switch3 == True Then
  If My.X <= 100.5 Then
```

```
  My.Direction = GetDirection(My.X, My.Y, 20, Rnd() * 7 + 28, Universe.map)
 Else
  My.Direction = GetDirection(My.X, My.Y, 181, Rnd() * 7 + 28, Universe.map)
 End if
End if

AvoidWallAndPeople()
}

Function WatchAhead(direc As Double, dist As Double, range As Double, type As Agttype) As Integer{
Dim neighbor As Agtset
Dim NoB As Integer
Turn(direc)
Forward(dist)
MakeOneAgtsetAroundOwn(neighbor, range, type, False)
NoB = CountAgtset(neighbor)
Forward(-dist)
Turn(-direc)
Return(NoB)
}

//人を避けるルール
Function AvoidWallAndPeople() As Integer{
Dim deg As Double
Dim destin As Double
Dim x As Double
Dim r1 As Double
Dim r2 As Double
x = 1.2      //混雑による速度減少の前方範囲
r1 = 3.0      //混雑による速度減少の前方人数
r2 = 6.0      //混雑による速度減少の前方人数
deg=0.0
destin=0.0
//まず壁の無い方向を向く
If WatchAhead(0, 1, 1, Universe.map.wall) > 0 Then
```

```
  deg = (2 * Round(Rnd()) - 1) * 90
  Turn(deg)
  If WatchAhead(0, 1, 1, Universe.map.wall) > 0 Then
    Turn(-deg * 2)
  End if
End if
//前方に人がいるか
If WatchAhead(0, 1, 1, Universe.map.hito) > 0 Then
  deg = (2 * Round(Rnd()) - 1) * 45
  Turn(deg)
  If WatchAhead(0, 1, 1, Universe.map.hito) > 0 Then
    Turn(-deg * 2)
    If WatchAhead(0, 1, 1, Universe.map.hito) > 0 Then
      Turn(-deg / 2)
      Forward(0.05)
    Else
      If WatchAhead(0, 1, 1, Universe.map.wall) > 0 Then
        Turn(-deg / 2)
        Forward(0.05)
      Else
       If WatchAhead(0, x, x, Universe.map.hito) >= 0 Then
        Forward(My.speed)
       Elseif WatchAhead(0, x, x, Universe.map.hito) > r1 Then
        My.speed = My.speed * 0.7
        Forward(My.speed)
       Elseif WatchAhead(0, x, x, Universe.map.hito) > r2 Then
        My.speed = My.speed * 0.5
        Forward(My.speed)
       Else
        My.speed = My.speed * 0.2
        Forward(My.speed)
       End if
      End if
    End if
  Else
    If WatchAhead(0, x, x, Universe.map.wall) > 0 Then
```

```
  Turn(-deg / 2)
  If WatchAhead(0, x, x, Universe.map.hito) > 0 Then
    Turn(-deg / 2)
    Forward(0.05)
  Else
    If WatchAhead(0, x, x, Universe.map.wall) > 0 Then
      Turn(-deg / 2)
      Forward(0.05)
    Else
      If WatchAhead(0, x, x, Universe.map.hito) >= 0 Then
        Forward(My.speed)
      Elseif WatchAhead(0, x, x, Universe.map.hito) > r1 Then
        My.speed = My.speed * 0.7
        Forward(My.speed)
      Elseif WatchAhead(0, x, x, Universe.map.hito) > r2 Then
        My.speed = My.speed * 0.5
        Forward(My.speed)
      Else
        My.speed = My.speed * 0.2
        Forward(My.speed)
      End if
    End if
  End if
Else
   If WatchAhead(0, x, x, Universe.map.hito) == 0 Then
    Forward(My.speed)
   Elseif WatchAhead(0, x, x, Universe.map.hito) > r1 Then
    My.speed = My.speed * 0.7
    Forward(My.speed)
   Elseif WatchAhead(0, x, x, Universe.map.hito) > r2 Then
    My.speed = My.speed * 0.5
    Forward(My.speed)
   Else
    My.speed = My.speed * 0.2
    Forward(My.speed)
   End if
```

```
      End if
   End if
Else
        If WatchAhead(0, x, x, Universe.map.hito) >= 0 Then
          Forward(My.speed)
        Elseif WatchAhead(0, x, x, Universe.map.hito) > r1 Then
          My.speed = My.speed * 0.7
          Forward(My.speed)
        Elseif WatchAhead(0, x, x, Universe.map.hito) > r2 Then
          My.speed = My.speed * 0.5
          Forward(My.speed)
        Else
          My.speed = My.speed * 0.2
          Forward(My.speed)
        End if
End if
Return(destin)
}
```

(2) wall エージェント

```
Agt_Init{

}

Agt_Step{

}
```

(3) exit エージェント

```
Agt_Init{

}

Agt_Step{
```

```
Dim surround As Agtset
Dim escaped As Agt
MakeOneAgtsetAroundOwn(surround, 1, Universe.map.hito, False)
If CountAgtset(surround) Then
  escaped = GetAgt(surround, int(Rnd() * CountAgtset(surround)))
  DelAgt(escaped)
End if
}
```

(4) dummy エージェント

```
Agt_Init{

}

Agt_Step{
Dim surround As Agtset
Dim passed As Agt
If Rnd() > 0.2 Then
   MakeoneAgtsetAroundOwn(surround,2,Universe.map.hito,False)
   If CountAgtSet(surround) > 0 Then
      passed = GetAgt(Surround,int(Rnd()*CountAgtset(surround)))
      passed.switch3 = True
   End if
End if
}
```

(5) adjust エージェント

```
Agt_Init{

}

Agt_Step{
Dim surround As Agtset
Dim catch As Agt
```

```
MakeOneAgtsetAroundOwn(surround, 1, Universe.map.hito, False)
If CountAgtset(surround) > 0 Then
  catch = GetAgt(surround, Int(Rnd() * CountAgtset(surround)))
  TerminateAgt(catch)
End if
}
```

(6) counter エージェント

```
Agt_Init{

}

Agt_Step{
Dim passed As Agt
Dim surround1 As Agtset
MakeOneAgtsetAroundOwn(surround1, 1, Universe.map.hito, False)
If CountAgtset(surround1) > 0 Then
  For each passed in surround1
    If passed.switch4 == False Then
      passed.switch4 = True
      passed.switch5 = True
    End if
  Next passed
End if
}
```

・case2～case4

(1) hito エージェント

```
Agt_Init{
My.Direction = Rnd() * 360
My.speed = 0.8
My.switch1 = False
My.switch2 = False
```

```
My.switch3 = False
My.switch4 = False
My.switch5 = False
}

Agt_Step{
Dim i As Integer
Dim deg As Double

If My.Y > 36 Then
  My.switch1 =True
End if

My.speed = 0.8

//まずダミーへ向かう（各 case で異なる）
If My.X < 100 Then
  My.Direction = GetDirection(My.X, My.Y, 81.5, 31.5, Universe.map)
Else
  My.Direction = GetDirection(My.X, My.Y, 120.5, 31.5, Universe.map)
End if

//出口へ向かう
If My.switch3 == True Then
  If My.X <= 100 Then
    My.Direction = GetDirection(My.X, My.Y, 20, Rnd() * 7 + 28, Universe.map)
  Else
    My.Direction = GetDirection(My.X, My.Y, 181, Rnd() * 7 + 28, Universe.map)
  End if
End if

AvoidWallAndPeople()
}

Function WatchAhead(direc As Double, dist As Double, range As Double, type As Agttype) As Integer{
```

```
Dim neighbor As Agtset
Dim NoB As Integer
Turn(direc)
Forward(dist)
MakeOneAgtsetAroundOwn(neighbor, range, type, False)
NoB = CountAgtset(neighbor)
Forward(-dist)
Turn(-direc)
Return(NoB)
}

//人を避けるルール
Function AvoidWallAndPeople() As Integer{
Dim deg As Double
Dim destin As Double
Dim x As Double
Dim r1 As Double
Dim r2 As Double
x = 1.2      //混雑による速度減少の前方範囲
r1 = 3.0      //混雑による速度減少の前方人数
r2 = 6.0      //混雑による速度減少の前方人数
deg=0.0
destin=0.0
//まず壁の無い方向を向く
If WatchAhead(0, 1, 1, Universe.map.wall) > 0 Then
   deg = (2 * Round(Rnd()) - 1) * 90
   Turn(deg)
   If WatchAhead(0, 1, 1, Universe.map.wall) > 0 Then
      Turn(-deg * 2)
   End if
End if
//前方に人がいるか
If WatchAhead(0, 1, 1, Universe.map.hito) > 0 Then
   deg = (2 * Round(Rnd()) - 1) * 45
   Turn(deg)
   If WatchAhead(0, 1, 1, Universe.map.hito) > 0 Then
```

```
  Turn(-deg * 2)
  If WatchAhead(0, 1, 1, Universe.map.hito) > 0 Then
    Turn(-deg / 2)
    Forward(0.05)
  Else
    If WatchAhead(0, 1, 1, Universe.map.wall) > 0 Then
      Turn(-deg / 2)
      Forward(0.05)
    Else
     If WatchAhead(0, x, x, Universe.map.hito) == 0 Then
      Forward(My.speed)
     Elseif WatchAhead(0, x, x, Universe.map.hito) > r1 Then
      My.speed = My.speed * 0.7
      Forward(My.speed)
     Elseif WatchAhead(0, x, x, Universe.map.hito) > r2 Then
      My.speed = My.speed * 0.5
      Forward(My.speed)
     Else
      My.speed = My.speed * 0.2
      Forward(My.speed)
     End if
    End if
  End if
Else
  If WatchAhead(0, x, x, Universe.map.wall) > 0 Then
    Turn(-deg / 2)
    If WatchAhead(0, x, x, Universe.map.hito) > 0 Then
      Turn(-deg / 2)
      Forward(0.05)
    Else
      If WatchAhead(0, x, x, Universe.map.wall) > 0 Then
        Turn(-deg / 2)
        Forward(0.05)
      Else
        If WatchAhead(0, x, x, Universe.map.hito) == 0 Then
         Forward(My.speed)
```

```
            Elseif WatchAhead(0, x, x, Universe.map.hito) > r1 Then
              My.speed = My.speed * 0.7
              Forward(My.speed)
            Elseif WatchAhead(0, x, x, Universe.map.hito) > r2 Then
              My.speed = My.speed * 0.5
              Forward(My.speed)
            Else
              My.speed = My.speed * 0.2
              Forward(My.speed)
            End if
          End if
        End if
      Else
          If WatchAhead(0, x, x, Universe.map.hito) == 0 Then
            Forward(My.speed)
          Elseif WatchAhead(0, x, x, Universe.map.hito) > r1 Then
            My.speed = My.speed * 0.7
            Forward(My.speed)
          Elseif WatchAhead(0, x, x, Universe.map.hito) > r2 Then
            My.speed = My.speed * 0.5
            Forward(My.speed)
          Else
            My.speed = My.speed * 0.2
            Forward(My.speed)
          End if
      End if
    End if
  Else
          If WatchAhead(0, x, x, Universe.map.hito) == 0 Then
            Forward(My.speed)
          Elseif WatchAhead(0, x, x, Universe.map.hito) > r1 Then
            My.speed = My.speed * 0.7
            Forward(My.speed)
          Elseif WatchAhead(0, x, x, Universe.map.hito) > r2 Then
            My.speed = My.speed * 0.5
            Forward(My.speed)
```

```
        Else
          My.speed = My.speed * 0.2
          Forward(My.speed)
        End if
End if
Return(destin)
}
```

(2) wall エージェント

```
Agt_Init{

}

Agt_Step{

}
```

(3) exit エージェント

```
Agt_Init{

}

Agt_Step{
Dim surround As Agtset
Dim escaped As Agt

MakeOneAgtsetAroundOwn(surround, 1, Universe.map.hito, False)
If CountAgtset(surround) Then
  escaped = GetAgt(surround, int(Rnd() * CountAgtset(surround)))
  DelAgt(escaped)
End if
}
```

(4) dummy エージェント

```
Agt_Init{

}

Agt_Step{
Dim surround As Agtset
Dim passed As Agt
If Rnd() > 0.2 Then
   MakeoneAgtsetAroundOwn(surround,2,Universe.map.hito,False)
   If CountAgtSet(surround) > 0 Then
      passed = GetAgt(Surround,int(Rnd()*CountAgtset(surround)))
      passed.switch3 = True
   End if
End if
}
```

(5) adjust エージェント

```
Agt_Init{

}

Agt_Step{
Dim surround As Agtset
Dim catch As Agt
MakeOneAgtsetAroundOwn(surround, 1, Universe.map.hito, False)
If CountAgtset(surround) > 0 Then
  catch = GetAgt(surround, Int(Rnd() * CountAgtset(surround)))
  TerminateAgt(catch)
End if
}
```

(6) counter エージェント

```
Agt_Init{
```

```
}

Agt_Step{
Dim passed As Agt
Dim surround1 As Agtset
MakeOneAgtsetAroundOwn(surround1, 1, Universe.map.hito, False)
If CountAgtset(surround1) > 0 Then
  For each passed in surround1
    If passed.switch4 == False Then
      passed.switch4 = True
      passed.switch5 = True
    End if
  Next passed
End if
}
```