

Optimal Staff Allocation based on Multi-agent Simulation

Haruki NISHIMURA*, Kenjiro ODA*, Hirohide HAGA**

(Received April 18, 2014)

The one of the most important aims of management simulation is to optimize the profit. These simulation don't consider the human factor, such as fatigue of staffs and dissatisfaction of the customers. This is partly because describing such kind of human factor is virtually impossible and even if we could describe, it is difficult to solve the equation which describes the system to be simulated. To solve this problem, this article proposes the adoption of MAS (Multi Agent Simulation) technique. MAS is the method of simulation with distributed nature. MAS contains two dominant components: agent and environment.

Our current target is the management simulation of a restaurant. In any restaurant, there are essentially two different agents; one is a staff agent and the other is a customer agent. Therefore we designed three different kinds of agents for the simulation of restaurant; first one is a kitchen agent (KA), the second agent is a hall agent (HA), and the third agent is a consumer agent (CA)

We have simulated the profit of restaurant when we change the number of staffs. This simulation describes the optimal solution of the number of staffs.

Key words: management simulation, Multi-agent simulation, optimization

キーワード: 経営シミュレーション, マルチエージェントシミュレーション, 最適化

マルチエージェントシミュレーションを用いた店員の最適配置の求解

西村栄毅, 織田憲二郎, 芳賀博英

1. はじめに

本論文は、飲食店内で店員が最も効率良く動けるための配置の仕方を、マルチエージェントシミュレーション(Multi-Agent-Simulation : MAS)により求めることを目的とする。店員の最適配置とは、店舗の大きさに基づく人員数や店舗の席の配置による人員の効率的な配置場所のことである。

近年、数多くの企業で自社の経営方針を定める手法としてシミュレーションが用いられている¹⁾。企業が用いるシミュレーションには様々なものがあり、どのシミュレーションも自社の営業利益を上げることが目的としている。しかし、企業がこれまで用いてきたシミュレーション技法は人間的

なファクタ、例えば、疲労や怠慢といった人の生物学的な特徴を踏まえたシミュレーション結果が少ないという課題が存在する。また、それらのファクタをシミュレーションモデルに導入しようとしても、それらのファクタを支配する法則を、数式で表現することは困難であり、仮に表現できたとしても、大幅な簡略化や非現実的な設定を導入せざるを得ないことも多く、シミュレーション結果の信頼性は、必ずしも高くない。これは、従来のシミュレーション技法が、系の全体を記述する体系(支配方程式)を基にしているために、そのような体系を構築できない、あるいは構築のために大幅な簡略化等が必要な体系のシミュレーション

*Graduate School of Science and Engineering, Doshisha University, Kyoto

Telephone:+81-774-65-6979, E-mail:hnishimura@ishss10.doshisha.ac.jp, koda@ishss10.doshisha.ac.jp

**Department of science and Engineering, Doshisha University, Kyoto

Telephone:+81-774-65-6978, E-mail:hhaga@mail.doshisha.ac.jp

が困難であることを意味している。

そこで、本研究では MAS^{2,3)}という技法を用いることで、上記の問題点を考慮したシミュレーションを行う手法を提案する。MAS ではエージェントと呼ばれる自律的に行動する存在が、環境と呼ばれる空間内で自律行動を行い、他のエージェントや環境と相互作用を行い、系全体に現象を創発する。MAS はその創発される現象を観察するシミュレーションである。この MAS の特長として、エージェントを集中的に管理する機構が不要で、エージェントに組み込まれたルールやパラメータに基づいて行動するという点、そしてマイクロな相互作用からマクロな状態や秩序が創発していくという現象をシミュレーションできるという点などがある。つまり、MAS は、従来存在したシミュレーション手法が、中央集権的な集中型のシミュレーションであったのに対して、個々のエージェントが自律的に自らの行動を決定する、という点から、分散的なシミュレーション技法であると考えることができる。

本論文では、飲食店を対象としたシミュレーションを行う。具体的には、店員というエージェントのパラメータに、疲労具合や接客の早さといった人間的な要素を付加させる。そして、飲食店内での店員の接客というマイクロな箇所から、店の評判とそれに伴う来客数の変化というマクロな状態を創発させる。この創発した状態の来客数や客の不満の数、そして、最終的には売上結果を分析することで、店舗に対する最も適した店員の配置を考えることが本研究の目的である。

2. シミュレーションの概要

まず、空間として店舗というものを作成する。空間には時間や来客数というパラメータを持たせることで、時間経過や人の店舗への出入り頻度を統括させ、現実世界に近い空間を作成する。そして、作成した空間に“店員”と“客”という2種のエージェントを配置する。店員エージェントは接客を、客エージェントは店員への注文を行うルールを設ける。店員のエージェントには速さと疲

労というパラメータを、客には品を待つことのできる限界の時間というパラメータを付加して、シミュレーションを始める。この2種のエージェントは、それぞれのパラメータを変化させながら、商品の提供・注文という行動を繰り返し続ける。これによって、飲食店の店内の動き、そしてそれに伴う、飲食店側の利益や店員の効率などの項目に対してシミュレーションを行う。構築したモデルを用いて、異なった条件下、例えば、店員数の増減や店員の行動範囲の制限などの条件を変化させて、シミュレーションを繰り返す。多くの可能な条件を変化させたシミュレーションの結果を比較考察し、店側の利益が最大になったり、店員の最も効率よく動ける配置を決定する。

3. シミュレーション環境の構築

3.1. シミュレーション空間の作成

空間はエージェントが相互作用する場のことである。本シミュレーションでは飲食店の店内空間をイメージした正方形の空間を作成した、この空間の大きさは、店内を動き回るエージェントの大きさを1とした場合、50の大きさとなっている。通常、MASでは平坦トラス型の空間を用いてシミュレーションを行う。しかし、本シミュレーションでは飲食店内という閉鎖空間での店員の動きを観察するために空間をトラスで表すことは行わずにシミュレーションを行った。

3.2. 空間のパラメータ

店舗に付随させるパラメータとして、本シミュレーションでは“時間”と“来客確率”の2つのパラメータを作成した (Fig. 1)。“時間”は現実世界での時間経過と同じものである。例えば、時間のパラメータが9となっていた場合、その空間内は現実世界で言う9時であるという意味である。今回のシミュレーションでは、営業時間内の店員の動きをシミュレーションするものであるので、営業時間として設定した9時から24時までの時間を1日とし、24時になれば翌日の9時から再びシミュレーションを続ける設定となっている。

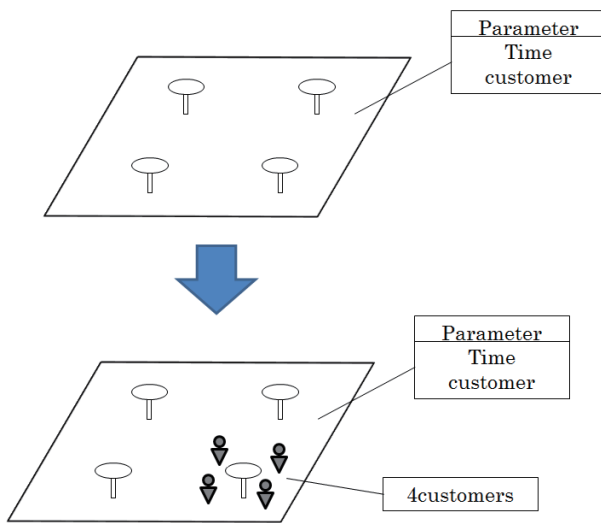


Fig. 1. Simulation environment.

本シミュレーションでは、空間内に等間隔に並べられた25カ所のテーブルをイメージした範囲に客エージェントを生成することで客の来店を表現する。客エージェント一つを一つの客集団とし、客エージェント一つ一つに人数というパラメータを作成し、集団の人数を決定した。

“来客確率”というパラメータは、客の来店する確率を計算するためのパラメータである。本研究では、より現実に近いシミュレーションを実現するために、次の来客数変化の表現の実現を目指した。

- ① 時間帯による繁盛期と閑散期
- ② 客の不満度によるリピーター量と新規客の変化

以上を実現するために、来客数を以下のように実装した。

- (I) 時間ごとの1分あたりの客の平均来客数と、来店検知頻度を Table 1 と Table 2 のように設定して、繁盛期と閑散期を表現した。
- (II) 不満度の蓄積により来客の減少を、 X_t を時刻 t での1分あたりの平均来客数とすると、

$$X_t = X_{t-1} - \text{“不満度”} \cdots (1)$$

で実現した。

Table 1. The number of counting of every time

time	average of customer's number
9:00～11:00	0.16
11:00～13:00	0.5
13:00～18:00	0.16
18:00～21:00	0.5
21:00～24:00	0.16

Table 2. The average number of customer

time	detective span(minute)
9:00～11:00	10
11:00～13:00	5
13:00～18:00	10
18:00～21:00	3
21:00～24:00	10

(I)の検知頻度とは来客数の計算を行う回数の頻度のことである。平均来客数が0と計算され無人のままの席には次の検知までは、無人のままとなる。そこで、検知回数を増やすことで無人席が有人席となる可能性を増やし、繁盛期・閑散期における、店内の満席・空席状態を表現した。

また、時間帯による来客数に変化をつけるために、1分あたりの平均来客数はポアソン分布⁴⁾を用いて定めた。ポアソン分布とは、ある一定区間の中で偶然に起こる事象の数の分布である。式(1)から1分あたりの平均来客数“ X ”を計算し、“ X ”を基に発生させたポアソン乱数から得られた乱数値“ Y ”の値を切り捨てた値“ Z ”をとり、“ Z ”を1分あたりの平均来客数とした。この確率計算を、ある一定時間おきに空間内の全ての無人テーブルで行い、整数値が0となった場合は計算を行ったテーブルには客が来なかった、1以上のときは客が来たとして顧客の来店の有無を表現した。こ

のときの整数値“ Z ”がテーブルに存在する顧客の数となる。

3.3. エージェントの作成

エージェントとは作成した空間内で自律行動をする要素である。本シミュレーションにおいて作成したエージェントは店員エージェントと客エージェントであるが、店員エージェントはさらに、“ホールエージェント”、“キッチンエージェント”に分かれる。この3つのエージェントがそれぞれ異なった自律ルールを持つ。

3.3.1. ホールスタッフエージェント

ホールスタッフエージェント(以下ホールエージェント)は、店舗内を動き回り、客エージェントへの対応を行うエージェント(Fig. 2)である。ルールとしては、客エージェントが存在しない場合は、店内をランダムに歩き回り、客エージェントが注文を行うと近寄るとなっている。

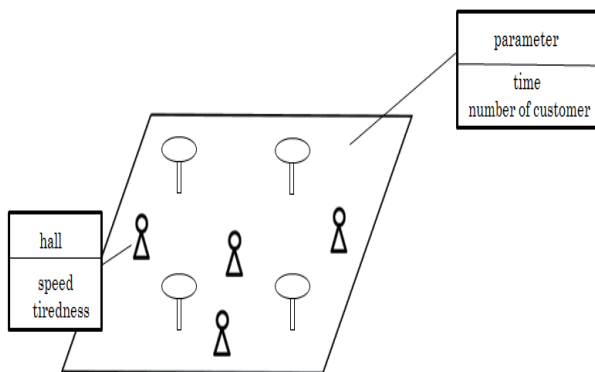


Fig. 2. The hall agent.

ホールエージェントは“疲労”と“速さ”という2つのパラメータを持つ。“疲労”とは、その名の通り、ホールエージェントの疲労の度合いを表し、客への対応をしたときに溜まる。また、接客時の疲労の溜まり方は、団体の人数が多いほど大きくなる。このことを表現するため、疲労値は接客を行う団体内の人数1人につき0.001ずつ溜まっていくとした。

“速さ”はホールエージェントの移動速度のことである。店舗を動き回るホールエージェントの速さは“速さ”のパラメータに起因し、値が小さいほど移動速度が遅い。疲労は速さに影響を与え、ホールエージェントの速さの値から疲労値を引いた値をホールエージェントの速さとした。時刻 t の時のホールエージェントの速さを“ V_t ”，1分あたりの疲労度を“ W ”とすると、実際の速さは下記の式となる。

$$V_t = V_{t-1} - W \quad \dots(2)$$

本シミュレーションでは、1日の作業をするホールエージェントは全く同一人物であると仮定し、ホールエージェントの疲労は営業中には回復しないとした。しかし、1日が経過し、翌日となった場合は疲労の値はリセットさせた。

3.3.2. キッチンスタッフエージェント

キッチンスタッフエージェント(以下キッチンエージェント)は客の注文を調理するエージェントである。キッチンエージェントの役割は、注文品を調理するだけであるので、店舗という空間内を移動することはないとし、店舗空間内には存在しないものとした。イメージとしては店舗とは異なった空間に存在し、調理を続けるだけの存在(Fig. 3)である。

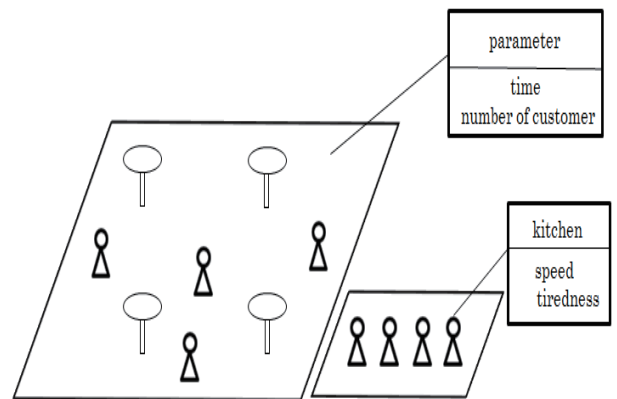


Fig. 3. The kitchen agent.

キッチンエージェントにはパラメータとしてホールエージェントと同様に“速さ”と“疲労”を持たせた。

“疲労”の溜まり方はホールエージェントとは異なっており，調理数一つずつにつき 0.001 ずつ増加させている．よって，疲労値の式は，式(3)となる．

$$\text{疲労値} = (\text{調理数}) \times 0.001 \cdots (3)$$

“速さ”はキッチンエージェントの調理速度である．キッチンエージェントの生成時に全てのキッチンエージェントの速さのパラメータ値を“1”と設定し，疲労値を加算させた値を現在の速さとする．そして，この速さの値を注文された品の調理に必要な時間にかけることで，疲労による調理速度の低下を表現している．よって，調理速度の式は，

$$\text{調理速度} = \text{完成に必要な基本の調理時間} \times (1 + \text{疲労値}) \cdots (4)$$

となる．例えば，注文品“B”の完成に必要な基本の調理時間は“4”分だとする．このとき，調理スタッフの疲労値が 0.5 であったとする．すると，そのときの調理スタッフの速さは基本の速さ 1 に 0.5 を加算した 1.5 になる．この値を調理時間 4 分に乗算すると 6 分となり，B の完成には 6 分がかかったということになる．つまり，注文品“B”は理想的には 4 分の時間で調理できるが，疲労が原因で 2 分の遅延が発生したということである．

キッチンエージェントの調理は 1 人につき 1 品ずつ調理すると仮定した．キッチンエージェントの数より注文の数が多い場合は注文された順番にストックさせていき，ストックされた注文は調理が終わったキッチンエージェントが順次調理を行っていくとした．注文が 1 つのときに，手の空いたキッチンエージェントが 2 人以上いた場合は調理者をランダムで決定するものとした．

なお，本シミュレーションでは，調理を終えた段階で客に品が届いたものとし，品の配膳動作はないものとしてシミュレーションを行った．

3.3.3. 客エージェント

客エージェントは，ホールエージェントに注文を行うエージェントである．客エージェントは生成されたのちに注文行動を行い，ホールエージェントが声をかける，つまりホールエージェントが近寄るまではホールエージェントの呼び出し状態として待つ．本シミュレーションにおいて，客の座る位置は客自身が決めるものとして，客の座席は空いている席にランダムに配置するという方法をとった．ホールエージェントが近寄ったあとは注文が終わったものとみなし，再び待機する．客エージェントには注文数，限界時間，そして不満度という 3 つのパラメータを持たせている (Fig. 4)．

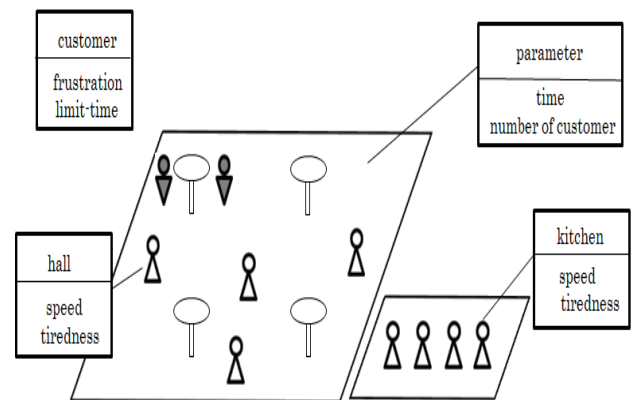


Fig. 4. The customer agent.

“注文数”とは客エージェントの注文した品数のことである．注文数は客エージェントごとにランダムで求め，客エージェントごとに 1 品から 6 品までの間の注文数をとるものとした．よって，先に求めた来客数と本節で求めた注文数とをかけた「来客数×注文数」の値がこの客が注文する品の合計数となる．

客エージェントが注文する品はシミュレーションの簡略化のために A, B, C のいずれかのみとし、客エージェントが注文した合計数をそれぞれの品にランダムに振り分けることで、各品の注文数を求めた。A, B, C それぞれの完成までにかかる時間を A は 7 分, B は 4 分そして C は 2 分と設定した。よって、それぞれのキッチンエージェントの調理速度が同じ場合、調理にかかる時間は、例えば A ならば 7 分で一律完成となる。また、それぞれの品の値段は A が 1000 円, B が 600 円, C が 300 円と設定し、この 3 つの売れた数により店舗の売り上げを計算する。具体例をあげて説明する。客の注文数が計 D 品であったとする。そして、A が a 品, B が b 品, C が c 品とそれぞれ振り分けられたとき、その客エージェントに対する売上は式(5)となる。

$$\text{客 1 団体あたり売上} = a \times 1000 + b \times 600 + c \times 300 \quad \dots(5)$$

“限界時間”とは、客エージェントが注文状態になってから品が完成するまでに耐えられる待ち時間を表すパラメータである。限界時間の値は客が生成されたと同時に 20～30 分までの間でランダムに設定した。客エージェントが注文状態になってから品が完成するまでは、

- (A).ホールエージェントが注文をとるために接近する
(B).キッチンエージェントが調理する

という 2 つの動作が行われなければならない。この (A), (B) の動作を終えるまでに時間がかかりすぎてしまい、注文した客の限界時間を超えたとする。すると、客エージェントからスピードに関して不満がでたものとみなし、不満度という客エージェントのパラメータを上昇させる。

“不満度”とは、注文した品の提供スピードに関する客の評価指標パラメータである。上記のよ

うに、店員側の提供スピードが客エージェントの限界時間を超えてしまったときに不満度が上昇する。不満度の計算法は、初期状態を 0 とし、限界時間を超えると品が届くまで毎分 0.001 ずつ加算させていく。

3.4. ホールエージェントの行動

ホールエージェントが客エージェントと接するためには、ホールエージェントがホールの中を移動する必要がある。この時、どのようにして移動するかによって、客エージェントとの接触の様子が分かってくる。本シミュレーションでは、次の 2 つの移動の仕方を設定した。

(1) 最近傍法

最近傍法はホールエージェントをランダムに歩かせ、注文が入ったときに最も客エージェントに近いホールエージェントが向かうという方法である。最近傍法のルールは、

1. ホールエージェントは店舗空間内のどこにでも移動できる、
2. 注文がないときはランダムに歩き回る、
3. 客が入り、客が注文状態になると各店員と注文をした客との距離を測定する、
4. 最も距離の近いホールエージェントが客の元へ向かい注文を取る、

という手順となる(Fig. 5).

そして、ホールスタッフの人数が異なった状態でシミュレーションを行い最近傍法におけるホールエージェントの最適人数を考察する。

(2) 4 分割法

4 分割法は店舗を 4 分割し、分割した各エリアそれぞれにホールエージェントを 1 人ずつ配置していく方法である(Fig. 6).

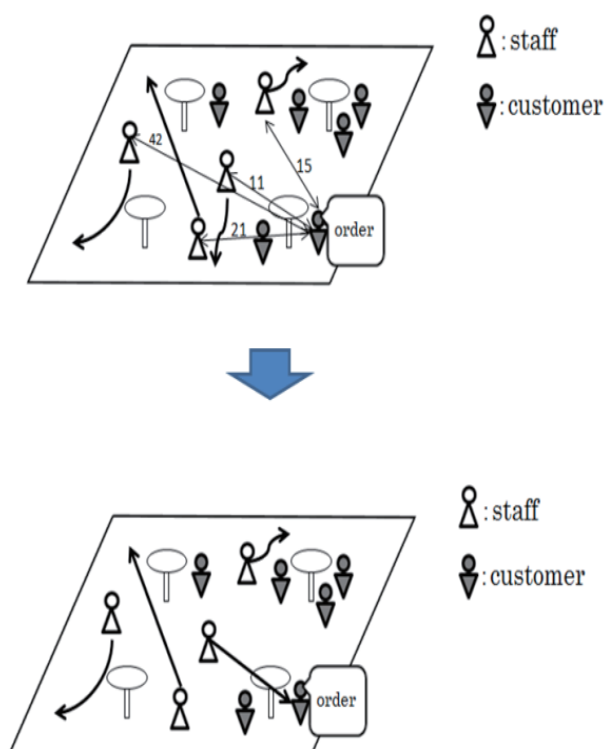


Fig. 5. Nearest neighbor method.

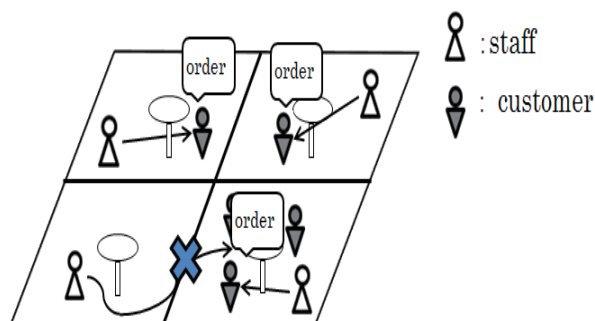


Fig. 6. 4blocks separation method.

4分割法のルールは下記となっている.

1. 中心点で店舗を4つに区切る
2. 各エリアに1人ずつホールエージェントを配置
3. 各エリア内の注文を全て担当のホールエージェントが行う
4. 注文がないときはランダムに歩き回る
5. 他のエリアにホールエージェントは移動不可

本論文では, 最近傍法と4分割法の比較を行い, どちらの方式が優れた方法であるかも検討する.

3.5. 評価指標

本シミュレーションにおいて, 店員の最適配置を求めるうえでの指標としては,

- (1) 店員全体の疲労が小さい
- (2) シミュレーション終了後の店員の疲労の差が小さい
- (3) 店の利益が大きい

という3つの事柄に注目することにした. 最終的には, 利益が最も高い状態が店員の最適配置であると考え, 考察を行っていく.

4. 結果

本節ではシミュレーションの結果について述べる. シミュレーションは(株)構造計画研究所から発売されている, シミュレーションソフトウェア artisoc バージョン 2.6 を用いて行った.

4.1. 最近傍法

キッチンスタッフの数を4人に固定した状態でホールスタッフの人数を変更した. ホールスタッフの人数は3人・4人・10人・15人の4パターンで行った. シミュレーションが1回終了した段階で人数別の売上, 疲労の平均値・最大値・最小値・最大最少差, そして客の不満度を求めた. この試行を100回行い, それぞれの値はこの平均値で求めた. 得られた値を Table 3, 4, 5 に記載する.

Table 3. Total sales

number of staff	average(Yen)	Max(Yen)	Min(Yen)
3	211490	238500	180700
4	226541	256500	198900
10	261310	278200	231700
15	268138	302500	246900

Table 4. Fatigue of hall-agent

number of staff	average	Max	Min	diffirence of fatigue
3	0.11684	0.12850	0.10596	0.02254
4	0.09434	0.10726	0.08135	0.02591
10	0.04344	0.06098	0.02826	0.03272
15	0.03002	0.04098	0.01532	0.03166

Table 5. Discontent of customer

number of staff	average	Max	Min
3	3.183	5.101	2.387
4	2.092	3.099	1.235
10	0.798	1.46	0.395
15	0.557	1.151	0.233

4.2. 4 分割法

最近傍法と同様にして、人数別の売上 (Table 6)、疲労の平均値・最大値・最小値・最大最少差 (Table 7)、そして客の不満度 (Table 8) を求めた。4 分割法は分割した各エリアに 1 人ずつの店員を配置していくので、ホールが 4 人のときのデータのみ求めた。

Table 6. Total sales

number of staff	average(Yen)	Max(Yen)	Min(Yen)
4	192996	226600	118800

Table 7. Fatigue of hall agent

number of staff	average	Max	Min	diffirence of fatigue
4	0.08669	0.10104	0.06611	0.03493

Table 8. Discontent of customer

number of staff	average	Max	Min
4	4.140	10.091	2.583

5. 考察

5.1. 最近傍法における店員の最適人数

人数別の優劣を検証するために、平均売上から人件費を引いた純利益を算出して比較を行う。人件費は各ホールエージェントに対して時給 800 円であるとし、営業時間 15 時間と合計人数をかけた値であるとした、つまり、

$$\text{純利益} = \text{平均売上} - (800 \times 15 \times \text{人数}) \cdots (6)$$

という計算で純利益を求めた。求めた純利益を Table 9 に記載する。本シミュレーションでは、シミュレーションの単純化のため、店舗の営業に必要な費用は人件費のみであると仮定して計算を行った。

Table 9. Net profit

number of staff	3	4	10	15
Net profit	175490	178541	141310	88138

Table 9 より、純利益はホールの数が 3 人もしくは 4 人のときが高く、10 人 15 人では利益が小さいことが分かる。また、Table 4, 5 より店員の疲労値・客の不満度ともに人数が 4 人のときが小さいことが分かる。

つまり、4 人の場合は、店員側から見れば疲労が小さくなり、客側の不満も溜まらないということになる。

以上のことから、本シミュレーション空間、つまり、正方形の空間かつ座席数が計 25 個の空間においては、ホールの人数が 4 人のときが店員の最適人数であると考えられる。

5.2. 最近傍法と4分割法の比較

ホールスタッフが4人のときの最近傍法と4分割法との比較を行う。

Table 3~8より、4分割法は売上・疲労値・不満度の全ての箇所で最近傍法よりも悪い結果となっている。この理由として、以下の2つが考えられる。

- (1) 客の座る位置が1つのエリアに固まる場合があったことを考慮していなかった。

本シミュレーションでは、客の位置は客自身が決めるとして、それを表現するために客の席位置をランダムで配置させた。そのため、1日を通して分割したうちの1つのエリアに客が来続ける状況が発生していた。4分割法では他のエリアのホールは他のエリアで接客をしないというルールを設けたので、忙しいエリア内の1人のホールがすべての接客を行い続ける現象が起きた (Fig. 7)。結果として、疲労の差が大きくなり売上も低下したのだと考えられる。

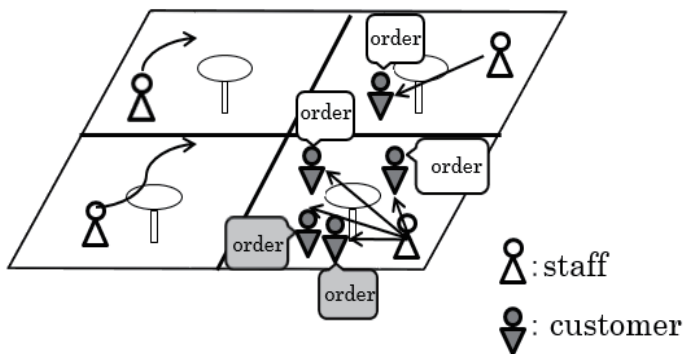


Fig. 7. Simulation in case that many customers get together one area.

- (2) ホールスタッフの疲労の溜まる要因として“移動距離”を考慮していなかった。

4分割法の長所として、店員とテーブルとの距離が縮まるので店員の移動距離が少ないことがあげられる。しかし、本シミュレーションでは移動による店員の疲労の蓄積は設定しておらず、この長所が上手く活かされていなかったと考えられる。

以上のことから、現段階では、最近傍法に有利な状況での比較となっており、優劣の判断材料としては必ずしも十分であるとは考えられない。

5.3. 今後の展望

今後は、考察を元に、4分割法で客の座る位置を店側が指定する方式でシミュレーションを行うつもりである。4分割法の欠点として、客が1か所に固まる場合に弱いという結果が得られた。よって、客がある部分に固まらないように、各エリア内に入る客の数を均等にさせる。また、移動によるホールスタッフの疲労の蓄積を新たに追加し、4分割法の長所が発揮される状態で最近傍法とどちらが優勢なのかを検証する。

また、パラメータの追加や調整を行い、より現実に近いシミュレーションを行っていく。検証の結果、キッチンの疲労は人数を増やせば減少させることができた。しかし、現実世界では人数を増やしすぎると人件費が嵩むことになる。このように、実際は管理費や維持費、人件費といった様々な要素が加わり、これらを計算することで営業利益が求まる。しかし、今回は売上と、人件費を差し引いた利益のみ求めた。従って、パラメータとして新たな要素を加えていき、より現実世界の店に近い形態でシミュレーションを行っていく。

6. 結言

本研究では、MASという手法を用いることにより飲食店の店員の配置の優劣を検証した。これにより、テーブル25席の正方形の店舗ではホールスタッフ数を4人にしたときの利益が多いことが分かった。また、顧客の着席場所がランダムでは、4分割法にて客が一つのエリアに固まる現象が発生し、移動距離と疲労との間に関係性を持たせなければ、4分割法の長所を表現できないという考察も得ることができた。

今回のシミュレーション結果だけでは、既存のシミュレーション結果に対抗しうる判断材料にはなっていないはずである。パラメータやエージェントの動作もまだ改良の余地があり、配置例も2

種のみしか検証していないからである。今後は、パラメータの付加・調整とエージェントの動きを現実に近いものにしていくことや、実験を元に得られた考察が正しいかどうかを MAS により明らかにしていくことを課題としたい。

参考文献

- 1) 山崎暢也, “大学と企業におけるシミュレーションの開発や活用の実態調査”, http://www.mizuho-ir.co.jp/publication/giho/pdf/001_11.pdf#search='%E5%A4%A7%E5%AD%A6%E3%81%A8%E4%BC%81%E6%A5%AD%E3%81%AB%E3%81%8A%E3%81%91%E3%82%8B%E3%82%B7%E3%83%9F%E3%83%A5%E3%83%AC%E3%83%BC%E3%82%B7%E3%83%A7%E3%83%B3', 2013.
- 2) 山影進, 人工社会構築指南, (書籍工房早山, 東京, 2007), P. 1-446.
- 3) 兼田敏之, 構造計画研究所創造工学部, 名古屋大学兼田研究所, artisoc で始める歩行者エージェントシミュレーション, (書籍工房早山, 東京, 2010), P. 1-187 .
- 4) 小暮仁, “ポアソン分布と指数分布”, <http://www.kogures.com/hitoshi/webtext/stat-poisson-bunpu/>, 2013.