論文・事例研究

待ち行列シミュレータを用いた区役所窓口における最適な職員数の提案

田澤　浩二，吉住　宗朔，平野　豪一，片山　翔太，中田　和秀

1. はじめに

IoT 社会の進展とともに，社会のさまざまな場面でデータの収集・蓄積が行われるようになり，この収集された“ビッグデータ”の利活用が課題となっている．これは民間企業に限った話ではなく，行政機関においても同様である．板橋区役所では，2014 年 10 月より，13 カ所の窓口で番号札の発券や来庁者の呼出，業務に関する受付ログデータを統一的に記録し，データの収集・蓄積が可能となったが，そのデータを有効に活用するには至っていなかった．

本論文では，経営科学系研究部会連合協議会主催，平成 27 年度データ解析コンペティションにおいて提供された受付ログデータをもとに，窓口業務を分析し，業務効率化，サービス向上を視野に入れた改善提案を行う．主に使用した受付ログデータは 2015 年 10 月から 11 月までの 2 カ月間に取得された新しい記録形式[1]のものである．窓口名と窓口ごとの来庁者数は表 1 にまとめた．分析の手法として，まず板橋区役所の現状を反映させた待ち行列モデルの提案と，そのモデルを実装した待ち行列シミュレータの作成を行う．このシミュレータによって，板橋区役所庁舎内における窓口の状況が再現できる．また，ポアソン回帰によって月，曜日，天候などの情報から時間ごとの来庁者数の予測を行う．これらを組み合わせることにより，来庁者の待ち時間分布や区役所職員の窓口業務の状況を高精度に予測することが可能となる．さらにこれらの予測に基づいて区役所窓口における最適な職員数の提案を行う．

たざわ　こうじ，かたやま　しょうた，なかた　かずひで
東京工業大学工学院経営工学系
〒 152–8552 東京都目黒区大岡山 2–12–1
よしずみ　しゅうさく，ひらの　ごういち
東京工業大学大学院社会理工学研究科経営工学専攻
〒 152–8552 東京都目黒区大岡山 2–12–1
受付 16.7.25　採択 16.11.9

表 1　窓口名と延べ来庁者数

窓口 ID	窓口名	来庁者数（人）
W001	戸籍係	2438
W002	住民台帳係	435
W003	住民異動係	10819
W004	証明係	16481
W005	保険給付係	2538
W006	国保収納係	2875
W007	資格賦課係	5115
W008	国民年金係	2118
W009	子どもの手当医療係	2695
W010	後期高齢医療制度課	1818
W011	介護保険課	2228
W012	板橋高齢者相談係	300
W013	長寿社会推進課	292

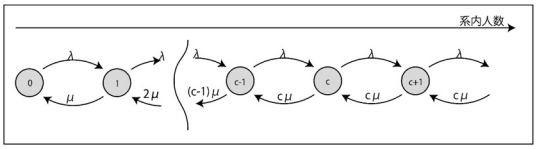

図 1　M/M/c モデルにおける状態推移図

2. 待ち行列モデル

本節では，板橋区役所の窓口で生じる待ち行列をできるだけ再現するモデル化とデータ加工について説明する．

2.1 複数サーバからなる待ち行列モデル

複数サーバからなる待ち行列の基本モデルとして，ポアソン到着，サービス時間が指数分布，サーバ数 c を仮定した M/M/c モデルが知られている [1]．ポアソン到着のパラメータを λ，各サーバのサービス時間が従う指数分布のパラメータを μ としたときの M/M/c モデルは図 1 のような状態推移図で表すことができる．

[1] 時刻が記録されるタイミングが変更されている．詳しくは 2.3 節を参照．

Copyright ⓒ by ORSJ. Unauthorized reproduction of this article is prohibited.

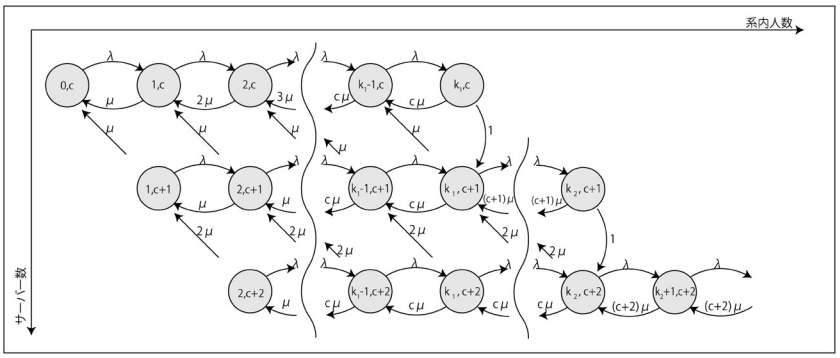

図 2　サーバ数が変化する待ち行列モデルにおける状態推移図

図において，ノードの番号 i は系内人数が i 人であることを意味する．また，枝に付されている重みはある状態から別の状態へ推移する確率分布のパラメータを表している．すなわち，ノード i からノード $i+1$ へ推移する確率分布はパラメータ λ のポアソン到着に従う．ノード i からノード $i-1$ へ推移する確率分布は，$1 \leq i \leq c$ のときにはそれぞれの状態におけるサーバ数 i に比例するパラメータ $i\mu$ の指数分布に従い，$c+1 \leq i$ のときにはパラメータ $c\mu$ の指数分布に従う．M/M/c モデルでは，それぞれの状態の存在確率の定常分布を求めることで，待ち時間分布や平均行列長などを解析的に求めることができる．

しかし，社会のさまざまな場面で生じる待ち行列の多くの場合と同様に，板橋区役所の待ち行列は M/M/c モデルで表現できるほど単純ではない．以下では板橋区役所の現状を精緻にモデル化するのに必要なサーバ数の変化，処理中断，窓口連携の三要素について説明する．

2.2　サーバ数が変化する待ち行列モデル

板橋区役所の各窓口におけるサーバ数を知るため，各時刻において来庁者の対応をしている端末数を調べた．すると，板橋区役所では，通常時と待ち行列の長さが一定値を超えた混雑時では，サーバ数を変化させていることがわかった．これは，通常時は窓口対応以外の仕事をしている職員が，窓口の混雑時に応援に呼ばれるためである．すなわち，普段は利用していないサーバを開放することで，混雑に対応している．このように，サーバ数が変化する場合，サーバ数が c で一定であると仮定している上記の待ち行列モデルを適用することができない．

このようにサーバ数が変化する場合の待ち行列モデルとしては，M/M/k/setup モデルが知られている [2, 3]．M/M/k/setup モデルは，データサーバにおけるサーバ管理の状況をモデル化しており，ジョブの待ち状況に応じてサーバが増減する．ジョブが来たときに，起動していないサーバがあればこれをセットアップし，ジョブの対応にあてるが，すべてのサーバが起動しているときには，ジョブは待ち行列に入る．サーバでのジョブの対応が終わったときに待っているジョブがなければ，このサーバの電源を落とすことで，消費電力を抑える．系内からジョブがなくなると，すべてのサーバの電源が落ち，新たにジョブが到着してからサーバの起動が始まることになる．また，待ち行列中のジョブ数がしきい値を超えてから初めてサーバが起動するモデルも提案されている [4, 5]．

本研究では，M/M/k/setup モデルをもとに新しい待ち行列モデルを提案する．具体的には，系内に客がいないときも含めて，通常時のサーバ数は c で固定されているが，待ち行列の長さが一定値を超えたとき，サーバ数が増加する待ち行列モデルである．客の到着がパラメータ λ のポアソン到着，各サーバにおけるサービス時間がパラメータ μ の指数分布とした場合における状態推移図を図 2 に示す．各ノードの番号 (i, j) は系内人数が i 人，サーバ数が j であることを意味する．系内人数がしきい値 k_1 $(k_1 \geq c)$ を超えると，パラメータ α の指数分布に従ってサーバ数が $c+1$ に増加する．さらに，しきい値 k_2 $(k_2 \geq k_1)$ を超えると，サーバ数が $c+2$ に増加する．以降同様にそれぞれのサーバ数ごとに定められたしきい値を超えるごとにサーバ数の上限が増加し，最大 $c_{\max}$ まで増加する（図 2 では $c_{\max} = c+2$）．サーバ数の減少は，増加分のサーバでのサービスが終了し，かつそのとき，待ち行列の長さがサーバ数増加のしきい値を下回っていたときに発生する．なお，増加分でないサーバでサービスが終了してもサーバ数は減少しない．このため，たとえば状態 $(k_1, c+2)$ から状態 $(k_1-1, c+2)$ への推移は，通常

 Copyright © by ORSJ. Unauthorized reproduction of this article is prohibited.

時のサーバ c カ所のいずれかでサービスが終了する場合に起きるので，指数分布のパラメータは $c\mu$ となっている．また，状態 $(k_1-1, c+1)$ への推移は増加分のサーバ 2 カ所のいずれかでサービスが終了する場合に起きるので，指数分布のパラメータは 2μ となっている．なお，本研究においては，基礎分析の結果から常に $k_i = k_{i-1}+1$ と設定した．提案した待ち行列モデルの妥当性は，3.3 節でシミュレーションによる待ち時間の頻度分布の再現性をもとに検討する．

2.3 “処理中断”について

板橋区役所の窓口においては，場合により業務が“処理中断”されることがある．処理中断とは，来庁者が窓口に呼び出され，一段階目のサービスを受けた後，区役所職員側での処理に時間がかかってしまう場合に，来庁者に一度待合に戻ってもらうことをいう．この後に再び呼び出され二段階目のサービスを受けることになる．処理中断をしている間，職員側の処理に窓口端末を必要としない場合，この端末をほかの職員が使うことで，待っているほかの来庁者を呼び出して待ち時間の短縮を図ることもある．この場合，手続き中の来庁者数が，そのとき使用可能な端末数よりも多くなることがある．

ただし，窓口ログデータには一段階目の完了時刻と二段階目の呼出時刻が記録されていない．このため，それぞれの来庁者について処理中断中の時間も含めた総サービス時間を知ることはできるが，一段階目のサービス時間と二段階目のサービス時間を正しく知ることができないという問題がある．その解決策として，来庁者が手続きを正常に終了していないデータ（未完了データ）と，2015 年 9 月までのデータ（旧データ）を組み合わせることで，処理中断前後のサービス時間に相当する経験分布を得ることができた．以下で詳しく述べる．

未完了データには，誤った窓口で発券してしまった受付番号についての発券時刻と取消時刻などを記録した“取消”データや，ある窓口を訪れた後に次の窓口を訪れる予約をしたものの実際には次の窓口に現れなかった人の受付番号についての発券時刻などを記録した“連携中”データなどが含まれている．それらのデータのうち，今回は“中断中”データに着目する．中断中データは，処理中断を必要とする手続きにおいて，一段階目の手続きが完了したものの二段階目の呼出に応じなかった人についてのデータである．このデータには一段階目の手続きの呼出時刻や完了時刻が記録されている．すなわち，一段階目のサービス時間の経験分布を得ることができる．

対して旧データでは，一段階目の呼出時刻の代わりに二段階目の呼出時刻が記録されている．これは，データ上，一段階目の呼出時刻が二段階目の呼出時刻により上書きされているからである．このデータからは一段階目の呼出時刻から二段階目の呼出時刻までの処理中断中の時間も含めた総サービス時間を知ることはできないが，二段階目の呼出時刻から完了時刻が記録されているため，二段階目のサービス時間の経験分布を得ることができる．

一般にデータ分析を行う際には，前処理としてデータに関する知識をもとにデータクリーニングを適切に行うことが重要であるとされている [6, 7]．データクリーニングに関する既存研究の多くは，欠損値を含んだデータを除去する，もしくは欠損値のないデータを用いて欠損値の補完を行っている．これに対して本研究では，記録形式が異なるデータや手続きが正常に終了していないデータなど，通常であれば分析から除外してしまうような，互いに欠損のあるデータ同士を組み合わせることで有益な情報を抽出した．数値実験の結果は 3.3 節で述べるが，この工夫により待ち行列モデルの精度を大幅に向上させることが可能となった．

2.4 “窓口連携”について

板橋区役所では，複数の窓口を訪問する来庁者の待ち時間低減のために，“窓口連携”システムを導入している．従来，複数の窓口を訪問する来庁者は，1 カ所目の窓口での手続きが終了した後，2 カ所目の窓口の待ち行列に新たに並び直す必要があった．しかし窓口連携システムでは，それぞれの来庁者は，1 カ所目の窓口でその後訪問する窓口を申告することで，以降の窓口では 1 カ所目の窓口の待ち行列に並び出した時間を基準として 2 カ所目の待ち行列に割り込むことができる．これにより，2 カ所目以降の窓口での待ち時間を低減することができるようになっている．

3. 待ち行列シミュレータと精度検証

前節で説明したモデルは処理中断や窓口連携などを考慮する必要があるため，待ち時間分布などを解析的に求めることは困難である．そこで，提案モデルを再現する待ち行列シミュレータを作成し，分析を行った．なおシミュレータは，エージェント・ベース・シミュレーションを Python を用いて実装している．

3.1 待ち行列シミュレータの概要

待ち行列シミュレータは，設定項目として，

・それぞれの来庁者の到着間隔

Copyright © by ORSJ. Unauthorized reproduction of this article is prohibited.

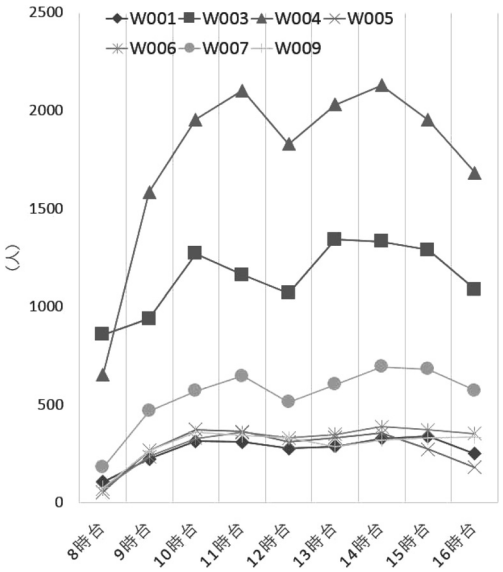

図 3　時間帯ごとの来庁者数

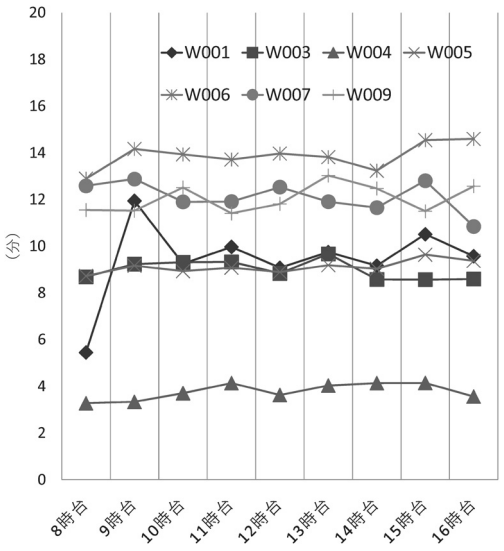

図 4　時間帯ごとの平均サービス時間

・それぞれの来庁者の訪れる窓口（組み合わせ）
・各窓口のサービス時間の分布
・通常時のサーバ数 c
・混雑時の最大サーバ $c_{\max}$
・サーバを増やすしきい値 k

を必要とする．これらを設定し実行をすると，それぞれの来庁者が待ち行列に並び，窓口でサービスを受ける動きをシミュレーションして，待ち時間を測定する．また，各サーバの稼働時間，すなわち区役所職員の来庁者対応時間も測定できる．シミュレータ上の来庁者は，入力された来庁時刻に来庁し最初に訪れる窓口の待ち行列に並ぶ．自分の順番がくると，その窓口のサービス時間の分布により生成されるサービス時間だけサービスを受ける．次に訪問する予定の窓口があれば，最初の窓口の来庁時刻をもとに，次の窓口の待ち行列に割り込み，次に訪問する予定の窓口がなければ，そのまま退庁する．

3.2　設定項目

シミュレータの設定項目は以下の考えに基づいて決定した．

来庁者数が 2,500 人以上の窓口（7 カ所）における時間帯ごとの来庁者数のグラフを図 3，平均サービス時間のグラフを図 4 に表している．なお，サービス時間 100 分以上の来庁者については異常値として除去した．このグラフから，来庁者の到着間隔（時間当たりの来庁者数）は時間帯に強く依存すること，各窓口のサービス時間の分布は時間帯に依存しないことが確認できる．ここで，図 4 において W001 の 8 時台のサービス時間がほかの時間帯と比べて大きく異なっているのは，来庁者数が少なかったために，観測値の偏りが平均値に大きく影響を与えてしまったためである．またほかの基礎分析の結果，ある来庁者の訪れる窓口の組み合わせは時間帯に依存しないこと，各窓口におけるサーバ数は通常時／混雑時の別にかかわらず曜日や時間帯に依存しないこともわかった．

こうした基礎分析の結果から，それぞれの来庁者の到着間隔や訪問窓口には実際のデータをそのまま用いながら，各窓口のサービス時間としては，それぞれの窓口を訪れた来庁者のサービス時間を経験分布としてランダムに用いることとした．また，各窓口における通常時のサーバ数 c，混雑時の最大サーバ $c_{\max}$，サーバを増やすしきい値 k も基礎分析により表 2 のとおり決定した．サーバ数が増加する場合については，系内人数がしきい値に到達してから 1 分後にサーバが増加するように設定した．

3.3　精度検証の結果

提案した待ち行列モデルの精度を検証するため，サーバ数の変化や処理中断を考慮していないモデル（単純モデル）のシミュレーション結果，サーバ数の変化や処理中断を考慮した提案モデルのシミュレーション結果，実測値の三つの比較を行った．

図 5 は来庁者の平均待ち時間に関して，単純モデルの結果，提案モデルの結果，実測値を窓口ごと棒グラフ

 Copyright © by ORSJ. Unauthorized reproduction of this article is prohibited.

表 2　各窓口におけるパラメータの設定値

窓口	c	$c_{\max}$	k
W001	3	6	6
W002	1	3	1
W003	8	15	8
W004	6	7	11
W005	4	5	4
W006	4	8	7
W007	4	9	19
W008	4	5	5
W009	4	5	4
W010	3	6	4
W011	3	6	4
W012	1	3	1
W013	1	2	2

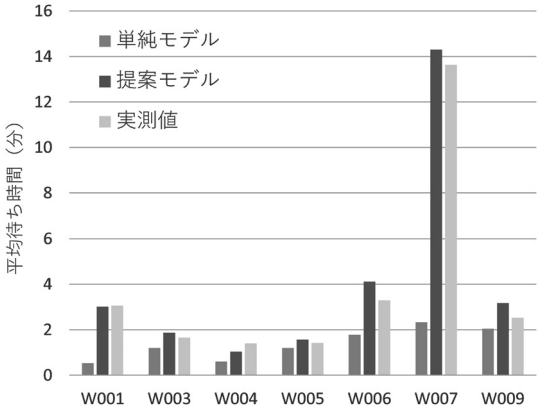

図 5　シミュレーションと実際の平均待ち時間

で示したものである．いずれの窓口においても提案モデルを実装したシミュレーションの結果が実際の平均待ち時間に近いことがわかる．特に，単純モデルの精度が悪かった W001 や W007 などでは提案モデルの有効性が顕著である．これは，W007 ではしきい値 k が大きいため，c と $c_{\max}$ の乖離が待ち行列の状態に大きな影響を与えたからだと考えられる．また，W001 に関しては，特に処理中断が多い窓口であり，未完了データと旧データを組み合わせることが有効であった．これらの結果は，単純な待ち行列モデルを採用するのでなく，待ち行列が生じる現場の状況に合わせて待ち行列モデルを構築することの必要性を示している．次に，図 6–8 は，それぞれ W001, W004, W007 の来庁者の待ち時間の頻度分布を表したものである．いずれも，提案モデルを適用したシミュレーションにより得られた待ち時間分布が，実際の待ち時間分布とよく適合していることがわかる．グラフは省略したが，ほかの窓

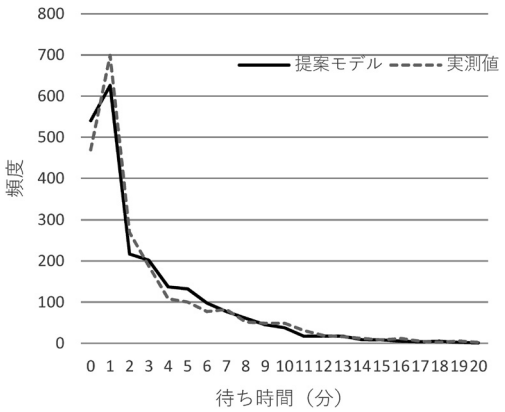

図 6　W001 におけるシミュレーションと実際の待ち時間の頻度分布

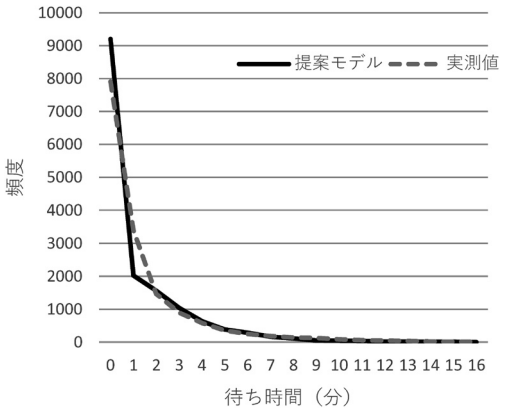

図 7　W004 におけるシミュレーションと実際の待ち時間の頻度分布

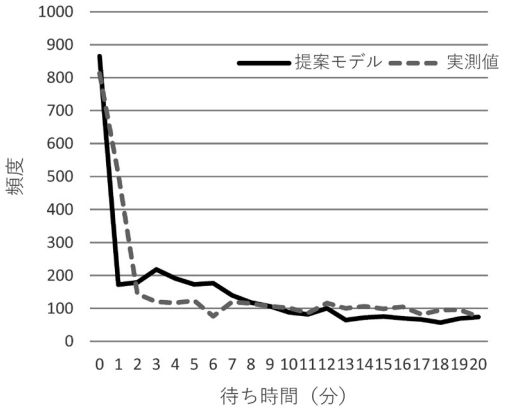

図 8　W007 におけるシミュレーションと実際の待ち時間の頻度分布

Copyright © by ORSJ. Unauthorized reproduction of this article is prohibited.

表 3　W004 におけるポアソン回帰の回帰係数

	係数	標準誤差	z 値	P 値
3 月	0.104	0.030	3.47	0.001
4 月	0.010	0.032	0.31	0.760
5 月	0.009	0.033	0.26	0.793
6 月	0.225	0.031	7.25	0.000
7 月	0.118	0.032	3.71	0.000
8 月	−0.031	0.031	−0.97	0.333
9 月	0.136	0.032	4.25	0.000
10 月	0.037	0.032	1.16	0.247
11 月	0.000	—	—	—
月曜日	0.696	0.035	19.66	0.000
火曜日	0.587	0.035	16.90	0.000
水曜日	0.542	0.035	15.40	0.000
木曜日	0.519	0.035	14.99	0.000
金曜日	0.570	0.035	16.39	0.000
土曜日	0.000	—	—	—
9 時台	0.000	—	—	—
10 時台	0.203	0.029	6.89	0.000
11 時台	0.250	0.030	8.39	0.000
12 時台	0.085	0.030	2.85	0.004
13 時台	0.173	0.030	5.78	0.000
14 時台	0.205	0.030	6.90	0.000
15 時台	0.160	0.030	5.36	0.000
16 時台	0.048	0.030	1.48	0.139
晴れ	0.236	0.060	3.92	0.000
曇り	0.188	0.062	3.02	0.003
雨	0.016	0.064	0.25	0.806
豪雨	0.000	—	—	—
切片	2.938	0.072	40.63	0.000

口についても同様に実際の待ち時間分布を再現できていることが確認できた．以上より，提案モデルは板橋区役所で生じる待ち行列を非常に精度高くシミュレートできたといえる．

4.　来庁者数予測の導入

本節では，ポアソン回帰を用いて，窓口ごとの来庁者数の予測を行う．この予測をもとに待ち行列シミュレータの設定項目である到着データの生成を行う．なお，来庁者数の記録に関しては，2015 年 9 月までの旧データと 10 月以降の新データに違いはないので，本節では二つのデータを同様に扱う．

4.1　ポアソン回帰による来庁者数予測

基礎分析の結果，来庁者数はポアソン分布に従ってばらつくことがわかった．また，窓口ごとの来庁者数は，月・曜日・時間帯・天候といった状況に依存することが予想される．そこで，上記尺度に関するダミー変数を説明変数として，来庁者数を目的変数とするポアソン回帰を行った．その際，月・曜日・時間帯・天候のそれぞれについて，11 月，土曜日，9 時台，豪雨のときを基準としている．

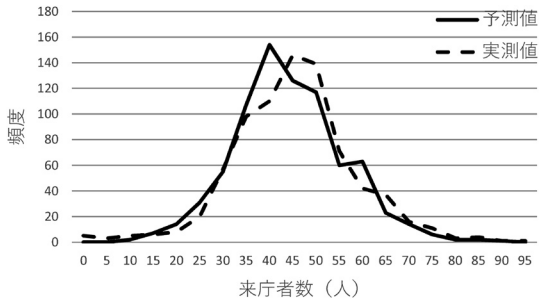

図 9　予測された来庁者数と実際の分布

一例として，W004 における回帰により得られた回帰係数を表 3 に示す．休日の前後の曜日やよい天候のときに係数が大きくなっており，直感とも一致する．

ポアソン回帰によって，来庁者数の分布に相当するポアソン分布を得ることができる．1 時間ごとの予測来庁者数を一つ得るために，回帰式によって計算できる値をパラメータとするポアソン分布に従って乱数を発生させた．この予測来庁者数と実際の時間帯ごとの 100 日分の来庁者数の分布を図 9 にて比較する．二つの分布はよく一致しており，来庁者数を精度よく予測できていると考えられる．

4.2　待ち行列シミュレータへの導入

予測来庁者数が一つに定まれば，3 節で説明した待ち行列シミュレータに対する来庁者の到着データを生成することができる．それ以外の設定項目は 3.2 節で説明したものを使うとシミュレータの実行が可能となる．実際は予測来庁者数をポアソン分布で表現しているため，この分布から 100 個の乱数を発生させ，そこから 100 回分の到着データを生成する．この到着データを用いてシミュレーションを 100 回実行し，その平均を取ることで来庁者の待ち時間や区役所職員の窓口業務の状況を予測する．

5.　区役所窓口における最適な職員数の提案

本節では，実際にシミュレータを用いて通常時の窓口職員数の最適化を行う．なお，この窓口職員数は，本研究で考えているサーバ数変化モデルにおける通常時のサーバ数に対応している．

5.1　最適な窓口職員数

最適な窓口職員数を考えるうえで，シミュレーションにより計算できる三つの指標，来庁者の待ち時間，応援の職員が窓口対応を行っている時間（応援時間），窓

 Copyright © by ORSJ. Unauthorized reproduction of this article is prohibited.

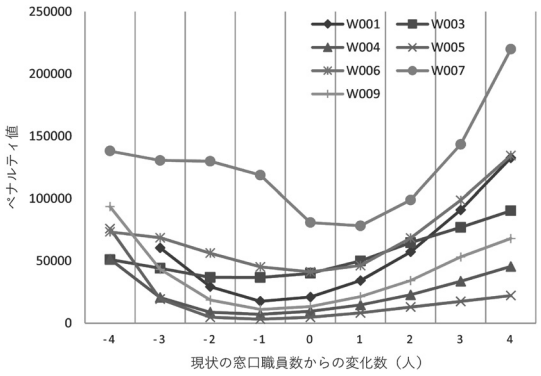

図 10　10 月水曜日雨の場合の窓口職員数の変化数ごとのペナルティ値

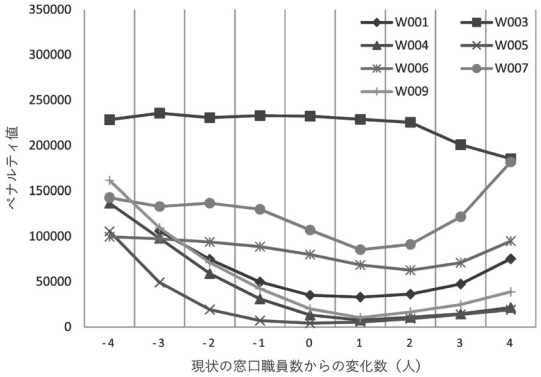

図 11　3 月月曜日晴れの場合の窓口職員数の変化数ごとのペナルティ値

口職員が余ってしまっている時間（空き時間）を考慮する．窓口職員数と各指標の関係を考えると，窓口職員数が多く設定されているときには，待ち時間や応援時間が低く抑えられるものの空き時間が発生しやすくなる．一方，窓口職員数が少なく設定されているときには，空き時間は低く抑えられるが待ち時間や応援時間が長くなってしまう．このように，上記 3 指標はすべてを同時に最小化することはできない．多目的最適化の方法としては，パレート最適な解を列挙するなどいくつかの方法が提案されているが [8]，本研究では重み和をペナルティとして捉え，その最小化を行う．それによりペナルティが最小になるという意味で最善な窓口職員数を一つ提示することができる．なお，窓口職員数として想定されるパターンは多くないため，そのすべての場合において 3 指標を列挙して，現場の人に選んでもらうことも一案である．

重み和の最小化にあたっては，重みの決定が問題となる．重みは実務家の要請により決定すればよいが，本研究で用いた重みについては 5.3 節で述べる．

なお，本研究では待ち時間・応援時間・空き時間の三つの指標をもとに最適化を行ったが，シミュレータによって計算できる別の指標，たとえば最長待ち時間や応援の回数などをもとに最適化することも可能である．

表 4　各窓口のシミュレーション結果

窓口	現在の人数	最適人数	累計サービス時間
W001	3	2	982.9
W002	1	1	97.5
W003	8	7	1707.7
W004	6	5	1132.2
W005	4	3	416.6
W006	4	4	1034.1
W007	4	5	1551.4
W008	4	4	980.0
W009	4	3	773.8
W010	3	2	396.3
W011	3	2	501.4
W012	1	1	55.9
W013	1	1	19.6

5.2　改善策の提案

ここでは，二つの状況下における最適な職員数を提案する．

まず，来庁者数が少ないと予想できる 10 月・水曜日・雨の日に注目する．来庁者が多い 7 窓口での，窓口職員数の変化数に対するペナルティ値を図 10 に示した．多くの窓口において窓口職員数を現状よりも減らしたときに重み和が最小となっていることがわかる．たとえば W001 については，窓口職員数を 1 人減らした場合，ペナルティ値を窓口職員数を現状どおりとする場合から 16%減らすことができる．

次に，来庁者数が多いと予想できる 3 月・月曜日・晴れの日に注目する．来庁者が多い 7 窓口での，窓口職員数の変化数に対するペナルティ値を図 11 に示した．先ほどとは対照的に，窓口職員数を現状よりも増やしたときに重み和が最小となっていることがわかる．先ほどと同様に W001 について見ると，窓口職員数を 1 人増やすことで，ペナルティ値を現状から 6%減らすことができる．

また，10 月・水曜日・雨の日の各窓口における現在の職員数，最適な窓口職員数，現在の累計サービス時間（来庁者数に平均サービス時間を乗じた数値）を表 4 に示す．これを見ると職員数の最適化によって，職員数が累計サービス時間に比例する方向に増減していることがわかる．

Copyright © by ORSJ. Unauthorized reproduction of this article is prohibited.

表 5　窓口ごとに決定した重み

窓口	待ち時間	応援時間	空き時間
W001	0	0.669	0.331
W002	0	0.446	0.554
W003	0.854	0.045	0.101
W004	0.714	0	0.286
W005	0	0.972	0.028
W006	0.712	0	0.288
W007	0.255	0.041	0.704
W008	0	0.884	0.116
W009	0	0.870	0.130
W010	0	0.940	0.060
W011	0	0.890	0.110
W012	0	0.462	0.538
W013	0	0.909	0.091

5.3　重みの決定

本研究では待ち時間・応援時間・空き時間の窓口ごとの重みを以下のように決定した．まず，現在の板橋区役所における窓口職員数が，平均的な来庁者数に対しては最適となっていると仮定した．この仮定のもとに，窓口職員数が ±0 のときのペナルティ値が最小となるように重みを決定している．このためにまず，平均的な来庁者数の到着データを用いてシミュレーションを行い，窓口職員数が −4〜+4 のときの待ち時間・応援時間・空き時間をそれぞれ計算する．この窓口職員数が i 人の時の指標をそれぞれ t_i^1, t_i^2, t_i^3 $(i = -4, -3, \ldots, 4)$ とする．またそれぞれの指標についての重みを w_1, w_2, w_3 とする．このとき，w_1, w_2, w_3, p_i $(i = -4, -3, \ldots, 4)$ を変数とする次の最適化問題を解くことで，現在の窓口職員数（$i = 0$ のとき）がロバストに最適となるような重みを決定できる．なお，この最適化問題は線形計画問題に変形できるため解くことは容易である．

$$
\begin{aligned}
\max \min_{i=0,1,2,3} & \{p_{i+1} - p_i,\ p_{-i-1} - p_{-i}\} \\
\text{s.t.}\ \ & p_i = w_1 t_i^1 + w_2 t_i^2 + w_3 t_i^3 \\
& \qquad\qquad \text{for } i = -4, -3, \ldots, 4, \\
& w_1 + w_2 + w_3 = 1, \\
& w_1, w_2, w_3 \geq 0.
\end{aligned}
$$

得られた窓口ごとの重みを表 5 に示す．多くの窓口で，待ち時間に対する重みが 0 になっているが，応援時間の最小化は同時に待ち時間を最小化することにも繋がるため，間接的に待ち時間の最小化も考慮された重み付けになっているといえる．なお，待ち時間に対する重みが 0 より大きな値を取っているのは W003, W004, W006, W007 の四つの窓口である．これらのうち W003, W004, W007 は来庁者数がほかの窓口に比べ多く（表 1），W006 は平均サービス時間が最も長い窓口である（図 4）ことが見て取れる．また空き時間の重みが大きい W002 と W012 は来庁者数が非常に少ない窓口であり（表 1），そのため窓口に常駐する職員を減らし，来庁者が訪れたときにだけ職員が窓口に出てきて対応をしているからだと考えられる．

6.　おわりに

本論文では，系内人数の増加に従ってサーバ数が増える待ち行列モデルを導入し，処理中断や窓口連携も考慮することにより，板橋区役所で生じる待ち行列を再現できるシミュレータを実装することに成功した．また，月，曜日，天候などの情報から 1 時間ごとの来庁者数の予測を行った．その結果，来庁者の待ち時間分布や区役所職員の窓口業務の状況を高精度に予測することが可能となった．それらを踏まえ，板橋区役所に対する改善策として，来庁者の待ち時間，窓口職員の空き時間，窓口業務の応援時間に基づいた最適な窓口職員数の提案を行った．ほかにも，本シミュレータを用いることによる来庁者に対しての待ち時間情報の提供や，来庁者数予測を用いることによる現在板橋区役所 HP にて提供されている混雑予想カレンダーの精緻化などの実務面への応用が考えられる．

謝辞　データを提供していただきました，データ解析コンペティション関係者の皆様方に感謝いたします．

参考文献

[1] 牧本直樹，『待ち行列アルゴリズム—行列解析アルゴリズム—』，朝倉書店，2001.
[2] A. Gandhi, S. Doroubi, M. Harchol-Balter and A. Scheller-Wolf, “Exact analysis of the M/M/k/setup class of Markov chains via recursive renewal reward,” *ACM SIGMETRICS Performance Evaluation Review*, **41**, pp. 153–166, 2013.
[3] T. Phung-Duc, “Exact solutions for M/M/c/setup queues,” *Telecommunication Systems*, pp. 1–16, 2014.
[4] I. Mitrani, “Service center trade-offs between customer impatience and power consumption,” *Performance Evaluation*, **68**, pp. 1222–1231, 2011.
[5] I. Mitrani, “Managing performance and power consumption in server farm,” *Annals of Operations Research*, **202**, pp. 121–134, 2013.
[6] 慶應 SFC データ分析教育グループ編，『データ分析入門（第 4 版）』，慶應義塾大学出版会，2002.
[7] 寺野隆雄，“データマイニングの展望，”計測と制御，**41**, pp. 315–318, 2002.
[8] 穴井宏和，『数理最適化の実践ガイド』，講談社，2013.

 Copyright © by ORSJ. Unauthorized reproduction of this article is prohibited.