待ち行列モデルによる タイムドリブン型シミュレーション

飯山尚可　(指導教員：吉田裕亮)

1　はじめに

待ち行列とは何かを待つことにより生じる,「到着時間」と「到着人数」,「サービス開始時間」,「サービス終了時間」,「サービス人数」などの関係を数学的に把握するための理論である. 銀行の ATM や遊園地など私たちの身近に多く存在し, 計算機の内部でも待ち行列が使われている.

どのようなサービス方法がよいかは待ち行列理論の大きな課題の 1 つである. サービス方法の評価に関しては, 平均待ち時間やその分散が小さいほど良いシステムといえる. 本研究では, 待ち行列理論を用いた銀行の ATM のモデルに関して数値シミュレーションを行い, 公平性の高い待ち行列を検証する.

2　待ち行列のシミュレーション

2.1　指数分布とポアソン分布

通常の待ち行列では, まったく客が来ないときと, 一度にたくさんの客が到着するときがある. 待ち行列のシミュレーションを行うには, 客の到着時間間隔と, サービス時間を指数分布に従う乱数で発生させるか, 単位時間当たりの客の到着人数と単位時間当たりのサービス人数をポアソン分布に従った乱数によって決定させることが考えられる. このように, ポアソン分布と指数分布は表裏一体であり, 指数分布に従った時間間隔で客が到着すると, 単位時間当たりの到着数はポアソン分布になる.

2.2　M/M/1 モデル

M/M/1 というのはケンドールの記法で表記された待ち行列のモデルで, 到着時間間隔, サービス時間がそれぞれ指数分布に従い, 窓口が 1 つであるモデルのことである. なお, 新しく到着した客や要求は待ち行列の一番後ろに並び, 窓口やサーバは待ち行列の先頭から順に客や要求を処理するものとする.

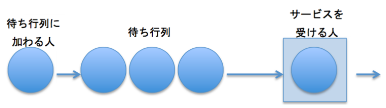

図 1: M/M/1 モデル

2.3　M/M/s モデル

窓口を s 個用意した場合, M/M/s モデルとなる. 新しく到着した客や要求は待ち行列の一番後ろに並び, 任意の処理を終えた窓口やサーバは待ち行列の先頭から順に客や要求を処理するものとする.

2.4　タイムドリブン型

待ち行列のシミュレーション手法としては, イベント・ドリブン型とタイム・ドリブン型がある. 本研究では, タイム・ドリブン型を用いてシミュレーションを行う. タイム・ドリブン型シミュレーションでは, 単位時間当たりのイベント発生率, およびイベント終了率を用いる. ここで, イベントとは, 客の到着, 処理の開始, サービスの終了の 3 つである. 本研究では単位時間当たりの客の到着確率と, 客の仕事量を設定することでイベントの発生と終了を表現した. それぞれの値は, 表 1 の通りである. ある小さな確率で毎時独立に客の到着を設定することで, ポアソン到着を実現する. これは二項分布 $Bin(n,p)$ の p が小さい場合, n が十分大きくなると $\lambda = np$ のポアソン分布で近似されることを応用している.

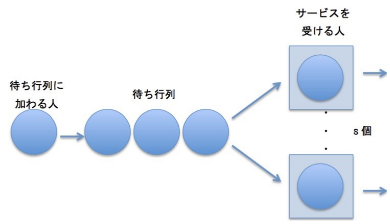

図 2: M/M/s モデル

客	到着確率	仕事量
客 1	*arrive*1	平均 60 秒
客 2	*arrive*2	平均 180 秒

表 1: 客の到着確率と仕事量

2.5　タイム・ドリブン型シミュレーションの応用

ATM での公平性の高い待ち行列を, タイム・ドリブン型シミュレーションにより検証したい. ある ATM コーナーには 18 台の ATM がある. 日常的に我々が目にするように, 仕事量には関係なく先着順でサービスを受けることが原則のルールとなっている. 到着した客は待ち行列の最後尾に並ぶ. 実際のは仕事量によって客を分けることは難しいと考えられるが, シミュレーションでは待ち行列に並ぶ時点で仕分けることが可能だと考える. それぞれ前の客の処理が終わり, ATM が空いたら待ち行列の先頭にいた客が ATM の前に案内され, サービス開始となる.

2.6　シミュレーション手順

現在の時間を $jikoku$, 総シミュレーション時間を $maxtime$ とする.

(1) 初期値の設定 $maxtime = 36000$, $jikoku = 0$

(2) 客の到着 乱数を生成し, その値が到着確率以下なら客が到着する. 客が来たら $waitinglist$ に追加.

(3) 処理の開始 ATM が空いたら $waitinglist$ から先頭の客を ATM に入れ処理開始.

(4) 処理の終了 処理時間が 0 になったら処理を終了.

(5) $jikoku \neq maxtime$ なら, $jikoku$ を増やして 2. へ.

3　提案モデル

3.1　モデル 1

日常的に ATM コーナーで見るように,ATM を分けずに客 1, 客 2 を混在して並ばせる.

3.2　モデル 2

ATM を客 1 用, 客 2 用に分けて並ばせる.

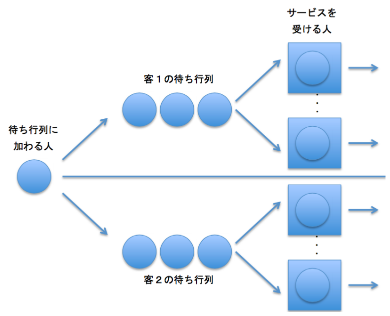

図 3: 提案モデル

4　実験概要

4.1　実験 1

モデル 1 およびモデル 2 において総 ATM 数 18 台で 36000 秒分のシミュレーションを 100 回行う. 時間内に処理が完了した客 1 と客 2 のそれぞれ平均待ち時間を測定し, その 100 回分の平均を求める. モデル 1 では, 客 1 と客 2 の持ってくる単位時間あたり仕事量と ATM の処理できる単位時間あたりの仕事量が釣り合うように $arrive1$ と $arrive2$ を設定する. モデル 2 では, 客 1 の持ってくる単位時間あたり仕事量と客 1 用の ATM の処理できる単位時間あたりの仕事量, 客 2 の持ってくる単位時間あたり仕事量と客 2 用の ATM の処理できる単位時間あたりの仕事量を, 釣り合うように $arrive1$ と $arrive2$ を設定する.

4.2　実験 2

台数による影響も考えられるため, モデル 1 およびモデル 2 において, 実験 1 の ATM の台数を 18 台から 36 台に変更して行った.

4.3　実験 3

実験 1, 実験 2 では ATM の容量いっぱいの客が到着しており, 客 1 が ATM の容量より少ない場合を観測するため, モデル 1 およびモデル 2 において, 客 1 の到着確率を 実験 1 の半分に設定しました. モデル 1 およびモデル 2 において, 客 1 の到着確率を実験 1 の半分に設定する. 実験 3 では $arrive1 = 150$, $arrive2 = 260$, 客 1 用 ATM5 台, 客 2 用 ATM13 台と設定した.

5　実験結果

実験 1 および実験 2 において, 客の到着率比が 1 以上場合は客 2 の方が待ち時間が長く, 1 未満の場合は客 1 の方が待ち時間が長いという結果であり, 総台数による台数の比率の影響は小さいと考えられる.

全体としては, 実験 1 より客が多く, ATM の総台数が多い実験 2 のほうが客 1 も客 2 も平均待ち時間が短いという結果になった.

また, 実験 3 においてモデル 1 よりモデル 2 のほうが客 1 は待ち時間が短くなるという現象が見られた.

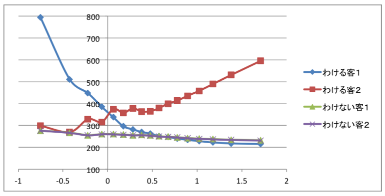

図 4: 実験 1 平均待ち時間

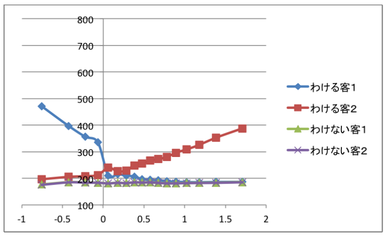

図 5: 実験 2 平均待ち時間

客	モデル 1	モデル 2
客 1 平均待ち時間	12.605	2.84306
客 2 平均待ち時間	12.6266	297.607

表 2: 実験 3 平均待ち時間

6　考察

今回のように客 1 と客 2 の総仕事量によって, ATM の台数を割り当てる方法は仕事量の少ない客の人数が仕事量の多い客の人数より多い場合には有効だった.

仕事量の少ない客の人数が仕事量の多い客の人数より多い場合には仕事量の少ない客用の ATM を設置することで, 仕事量に対する待ち時間が公平になる.

実験 3 の結果より, 客が混在するモデル 1 より ATM を分けるモデル 2 のほうが客 1 の待ち時間は大幅に減った. このことより, モデル 1 の場合は仕事量の少ない客が使うべき ATM を仕事量の多い客が使ってしまってることで, 待ち時間が長くなってしまっていると考えられる.

これは, ATM に限ったことではなく, スーパーのレジや役所の窓口などの行列に応用することができる.

7　今後の課題

ATM の台数や客の到着確率を更に変化させる. 客の持ってくる仕事量を変化させる. 一定時間以上の待ち時間になったら列を抜ける、一定人数以上は並ばないなどの現実的な条件を付加する.

8　参考文献

[1] 田村みのり, 待ち行列モデルのタイム・ドリブン型シミュレーションの応用, お茶の水女子大学大学理学部情報科学科卒業論文,2017.