

待ち行列モデルのシミュレーションとその応用

齋藤菜摘 (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。

時分割サービスと呼ばれる方法と割り込み処理で、見かけ上複数人が同時に一台の計算機を利用することを可能にしている。

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、平均待ち時間の平均やその分散が小さいほど良いシステムといえる。

2 待ち行列のシミュレーション

2.1 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。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とがある。イベントドリブン型とは、イベントの発生をきっかけとして処理を始めるものであり、タイムドリブン型とは時間を少しずつ進めながらシミュレーションを行っていくものである。本研究では、イベントドリブン型を用いてシミュレーションを行う。ここでのイベントは、客の到着、サービスの開始、サービスの終了の3つである。



図 1: M/M/1 モデル

2.2 指数分布とポアソン分布

通常の待ち行列を考えると、まったく客が来ないとき、いっきに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、客の到着時間の間隔と、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。

2.3 シミュレーション手順

シミュレーションではタイムドリブン型を用いる、すなわち毎時刻の状態の推移の時間発展を追跡する。

本研究では店に来る客が持っているタスクの総量と店に設置した窓口が処理することができるタスクの総量を平均で一致させることとする。

各客が持ってくるタスク量はポアソン分布で決めるにより、処理に要する時間が指数分布であることを表現する。

処理率、平均待ち時間、平均待ち標準偏差の結果を出力し、その平均を毎回 30 回実行し、その平均を結果とする。

ここで処理率とは、到着した客数のうち、その客が持ってきたタスク量をシミュレーション時間内に全て処理することができた客数の割合として定義する。

また、平均待ち時間と平均待ち標準偏差の値も持ってきたタスク量を総シミュレーション時間内に全て処理することができた客に対しての値である。

各窓が1回の処理あたりにこなせるタスク量を maxmadotask 、シミュレーションで用意する窓の数を madosuu 、客の持ってくるタスク量の平均を taskheikin とする。

なお、窓がこなせるタスク量とは、一回の処理あたりに一人の客に対して処理することができるタスク量を示す。

本研究では、客に割り当てられる taskheikin の値を複数個に設定した時に taskheikin の値が大きい客も小さい客も待ち時間なるべく平等に小さくなることを目的とする。

3 提案モデル

3.1 モデル 1

提案モデル 1 は、 madosuu の値を複数個として設定し、来店した客に与えられた taskheikin の値を全て処理し終わるまで窓口を占有するモデルである。

シミュレーションの流れ

- (1) 到着した客は待ち客用の椅子に座る。
- (2) 窓口が空いたら椅子の先頭に座っている客を窓口に入れて、椅子に座っていた客を1つずらす。
- (3) その客が持っている全てのタスクの処理が終了したら窓口を空け、椅子の先頭の客を空いた窓口に入れる。

3.2 モデル 2

提案モデル 2 は、 madosuu の値を複数個として設定し、客が持ってきたタスク量が窓口が一回の処理でこなすことができるタスク量を超えてしまったら椅子の最後尾に座り直し、持ってきたタスクが処理し終わるまで並び直すモデルである。

シミュレーションの流れ

- (1) 到着した客は待ち客用の椅子に座る。
- (2) 窓口が空いたら椅子の先頭に座っている客を窓口に入れて、椅子に座っていた客を1つずらす。

(3) もし、窓口に入った客が持ってきたタスク量が窓口が処理できる最大のタスク量を超えてしまっていたら、窓が処理できる最大のタスク量だけを処理し、持ってきたタスク量から maxmadotask の値を差し引いた量のタスクを持ったままもう一度椅子の最後尾に座り、空いた窓には椅子の先頭の客を入れる。

4 実験計画

以下の実験は全て客が平均 200 人来るものと想定している。またモデル 1 において maxmadotask の値は 200、モデル 2 において maxmadotask の値は 20 と設定し、実験を行う。

4.1 実験 1

taskheikin の値を 50 に設定し madosuu を増やし、平均処理率の平均、平均待ち時間の平均、平均待ち標準偏差の平均の値にどのような変化をもたらすかを調べる。

4.2 実験 2

madousuu を 5 つに設定し、taskheikin の値を増やし、結果にどのような変化をもたらすかを調べる。

4.3 実験 3

taskheikin50 持ってくる客と taskheikin100 持ってくる客を同じ割合で用意し、その状態で madosuu を増やすと結果にどのような変化をもたらすかを調べる。

5 実験結果

実験 1 では madosuu が増えるとモデル 1 とモデル 2 ともに処理率が減ったが、平均待ち時間の平均と平均待ち標準偏差の平均の値は小さくなった。

窓数5 (taskheikin50)	処理率	待ち時間	標準偏差
モデル1	0.93	57.66	40.59
モデル2	0.89	102.98	64.60
窓数10 (taskheikin50)	処理率	待ち時間	標準偏差
モデル1	0.91	22.49	17.28
モデル2	0.86	46.86	31.43
窓数15 (taskheikin50)	処理率	待ち時間	標準偏差
モデル1	0.89	11.71	9.89
モデル2	0.84	38.10	23.65

図 2: 実験 1

実験 2 では taskheikin が 150 になるとモデル 2 の平均処理率の平均が大幅に低くなった。また、平均待ち時間の平均と平均待ち標準偏差の平均の値は大きくなった。

窓数5 (taskheikin50)	処理率	待ち時間	標準偏差
モデル1	0.93	57.66	40.59
モデル2	0.89	102.98	64.60
窓数5 (taskheikin100)	処理率	待ち時間	標準偏差
モデル1	0.93	129.94	84.01
モデル2	0.90	249.28	147.83
窓数5 (taskheikin150)	処理率	待ち時間	標準偏差
モデル1	0.94	157.11	110.88
モデル2	0.74	327.11	205.87

図 3: 実験 2

実験 3 では処理率に大きな変化は見られなかったが、madosuu を増やすと平均待ち時間の平均と平均待ち標準偏差の平均の値は小さくなった。

窓数5 (taskheikin75)	処理率	待ち時間	標準偏差
モデル1	0.94	75.76	55.73
モデル2	0.89	157.79	93.26
窓数10 (taskheikin75)	処理率	待ち時間	標準偏差
モデル1	0.91	33.41	27.35
モデル2	0.88	52.50	38.85

図 4: 実験 3

6 考察

実験 1 の結果より、madosuu を増やすと、処理率の値は小さくなったが、平均待ち時間の平均と平均待ち標準偏差の平均の値はモデル 1、モデル 2 共に値が小さくなっていることから客の満足度は高くなることがわかった。また、モデル 2 よりもモデル 1 の方が効率よくタスクを処理することができるとわかった。

実験 2 の結果より、taskheikin の値を増やすと、平均待ち時間の平均と平均待ち標準偏差の平均の値が大きくなっていることから taskheikin の数が大きければ大きいほど客の待ち時間は増え、効率が悪くなってしまうことがわかった。また、この場合においてもモデル 2 よりもモデル 1 の方が効率よく処理できることがわかった。

予想ではモデル 1 よりもモデル 2 の方が効率が良いとしていたが、実験 3 の結果より、taskheikin の値を一樣にしても 2 種類を混ぜても変わらずにモデル 2 よりもモデル 1 の方が効率が良い。

これは、あらゆる待ち行列に応用することが可能と思われる。

7 今後の課題

本研究では客の満足度を優先し、全ての客の待ち時間が平等に少なくなる方法、即ち分散が小さくなるための方法を調べるためのシミュレーションを実行したが、店側の利益、大口の客または小口の客を優先した場合の処理率や待ち時間や分散の変化、持ってきたタスク量ごとの窓口を用意したりするなど、様々な実験が必要である。

参考文献

- (1) 高橋菜穂, 待ち行列モデルによる時分割サービスのシミュレーション, お茶の水女子大学大学院理学部情報科学科卒業論文, 2015.
- (2) 数理情報科学シリーズ 22 「待ち行列の数理とその応用 (改訂版)」 宮沢政清 (著)