2024 年度版

応用プログラミング実験

確率プログラミング　第4回

担当：中嶋 一貴

第 4 回：待ち行列シミュレーション

1　本日の目標

第 3 回で学んだ待ち行列シミュレーションでは，到着したイベントが必ず処理される状況を想定していた．これは例えば，バスが来たら必ず乗れるような状況に該当する．一方で，現実には様々な状況が想定されるため，状況に沿った待ち行列シミュレーションを考える必要がある．本実験では，待ち行列モデルを表現する記号であるケンドール記号 A/B/C/K について学び，M/M/S/S で表されるシミュレーションを実装することで，待ち行列の理解を深める．

2　実験の理論概要

2.1　ケンドールの記号

待ち行列のモデルを説明・分類するために使われる標準的な方法がケンドールの記号である．本実験では，4 つの要素から成る待ち行列のモデルを扱う．

- A: 客の到着間隔を表す到着分布.
- B: 窓口に来た客がサービスを受けている時間を表すサービス分布.
- C: サービスを提供するサーバの個数.
- K: システムの許容量.

ケンドールの記号を用いて，この待ち行列を A/B/C/K と表記する．図 1 に A/B/C/K 待ち行列の概要を示す.

2.2　M/M/S/S 待ち行列

ケンドール記号の分布 A,B に用いられる分布には M（Markokian; 指数分布），D（Deteministic; 決定分布），G（General; 一般分布）などの分布がある．例えば，M/M/S/S 待ち行列モデルは以下の要素から成る：

- A=M: 到着時間の間隔の分布が指数分布.
- B=M: サービス分布が指数分布.
- C=S: サーバの個数が S 個.

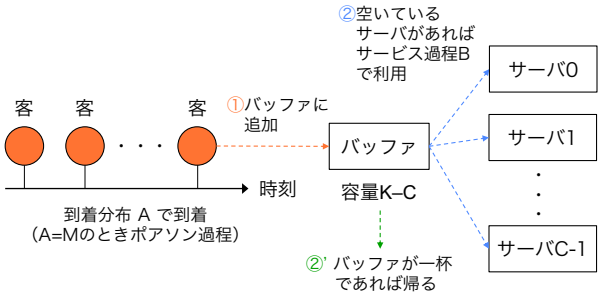

図 1: A/B/C/K 待ち行列.

- K=S: システムの許容量 K はサーバの個数 C とバッファサイズ T で表される．つまり K = C + T である．M/M/S/S では T = 0，つまり待つ場所がない．

例えば，待つスペースがないレストランを M/M/S/S 待ち行列でモデル化することを考えてみる：

- 客の到着時間間隔は平均 $1/\lambda$ の指数分布に従う．
- 客が着席してから離席するまでの時間（＝サービス時間）は平均 $1/\mu$ の指数分布に従う．
- レストランの座席数は S 個．
- 待つスペースはない．したがって，客が到着したときに座席が空いていなければ，その客はすぐに帰る．

待ち行列シミュレーションにおいて，用いるモデルを評価するために様々な指標が用いられる．その 1 つは，訪れた全客数に対してレストランに入れず帰ってしまう客数の割合で定義される “ロス率” である．M/M/S/S 待ち行列のシミュレーションの例を図 2 に示す．

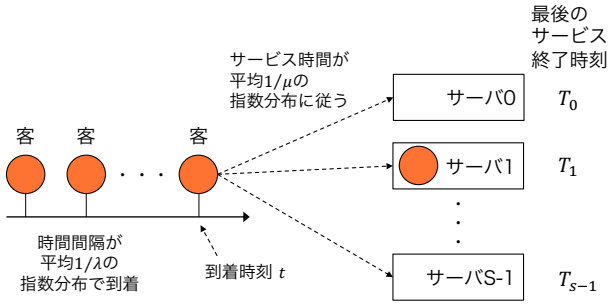

図 2: M/M/S/S 待ち行列．ある客が時刻 t に到着し，その時刻でサーバ 0 と S-1 は空いているがサーバ 1 は他の客が利用中である．サーバ i $(i = 0, \ldots, S-1)$ のサービスが最後に利用された時刻を T_i と書くと，この例では $T_0 < t$, $T_1 > t$, $T_{S-1} < t$ である．

課題 4-1: M/M/S/S シミュレータの作成

M/M/S/S 待ち行列のロス率をシミュレーションするための関数 checkout_counter_sim を実装せよ．関数は，引数に（1）客の到着分布のパラメータ λ（2）サービス分布のパラメータ μ（3）サーバの個数 S（4）客の数 n を受け取り，ロス率を計算するものとする．

レポートには，作成した関数のソースファイル名を言及すること．例えばファイル名が 'xyz.cpp' の場合，'xyz.cpp を参照のこと．' と書けばよい．ファイルが複数ある場合，全て言及すること．なお，考察を書く必要はない．

以下の手順に従って設計するとよい．

1. 課題 2 で設計した関数 'poisson_exp' を活用して，パラメータ λ のポアソン過程に従う n 個の 'Event' オブジェクト（＝訪れた客）のリストを生成する．
2. 3 つの変数を設定する：
   - 't': 客の到着時刻を表す変数．
   - 't_s': i 番目の要素 t_s[i] が i 番目のサーバの利用終了時刻を表す配列．
   - 'num_loss': 到着したが，サービスを受けずに帰った客の数．
3. $i = 0, \ldots, n-1$ について，'t' を i 番目の客が到着した時間に設定し，S 個のサーバの利用終了時刻の最小値 'min_t_s' を配列 't_s' から求める．'min_t_s' と 't' の大小関係に応じて以下の処理を行う．
   (a) 'min_t_s' < 't' であれば，その客は 'min_t_s' に該当するサーバでサービスを受ける．サービス時間は，パラメータ μ の指数分布に従う．これに伴い，配列 't_s' を正しく更新する．
   (b) 'min_t_s' $\geq$ 't' であれば，その客は帰る．つまり，'num_loss' が 1 増える．
4. 訪れた客数 n と帰った客数 'num_loss' からロス率を計算する．

課題 4-2: M/M/1/1 シミュレーション

レジが一台しかなく，先客がいた場合には到着した客が帰るスーパーを考える．関数 checkout_counter_sim を用いて，このスーパーを M/M/1/1 待ち行列でモデル化したときのロス率のシミュレーションを実行せよ．なお，$n = 100,000$ とせよ．

レポートには以下を記載すること．

1. $\lambda = 2.0$ とし，$\mu = 1.5, 1.6, 1.7, 1.8, 1.9, 2.0, 2.1, 2.2, 2.3, 2.4, 2.5$ に対してロス率をプロットした図．図 3 にプロット例を示す．
2. シミュレーションにより得られたロス率と理論値の比較に関する考察．なお，M/M/1/1 待ち行列のロス率の理論値は $\lambda/(\lambda + \mu)$ で与えられる．

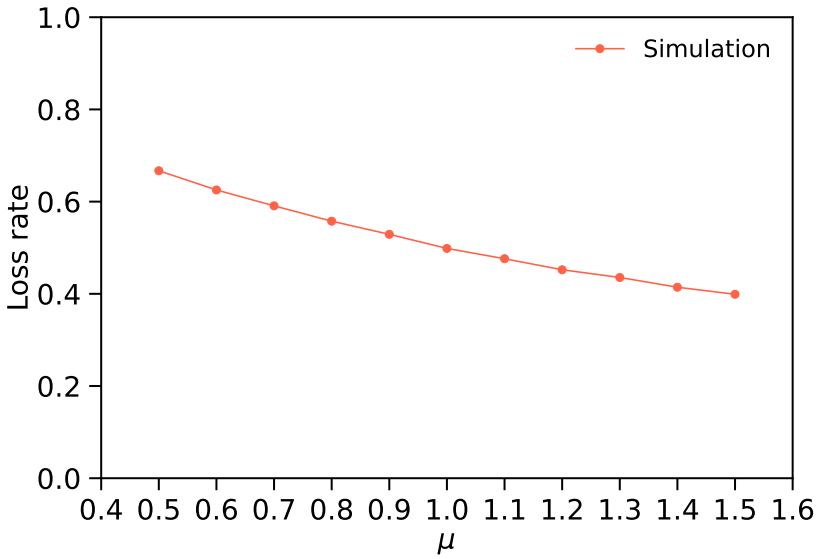

図 3: $\lambda = 1.0$ のときの M/M/1/1 待ち行列のロス率のシミュレーション結果．

課題 4-3: M/M/S/S シミュレーション

レジが S 台あり，先客がいた場合には到着した客が帰るスーパーがある．関数 checkout_counter_sim を用いて，このスーパーを M/M/S/S 待ち行列でモデル化したときのロス率のシミュレーションを S = 1, 2, 3, 4, 5 それぞれに対して実行せよ．なお，$n = 100,000$ とせよ．

レポートには以下を記載すること．

1. S = 1, 2, 3, 4, 5 のそれぞれに対して，$\lambda = 2.0$ とし，$\mu = 0.5, 0.6, 0.7, 0.8, 0.9, 1.0, 1.1, 1.2, 1.3, 1.4, 1.5$ のロス率をプロットした図．S = 1, 2, 3, 4, 5 の結果を 1 枚の図にまとめること．
2. シミュレーションにより得られたロス率の結果の考察．異なる S の間のロス率の結果の考察を含めること．なお，ロス率の理論値との比較の考察は不要である．

課題 4-4: M/M/1/∞ シミュレーション

M/M/S/S 待ち行列では，S 個のサーバ全てが利用中のときに到着した客はすぐ帰るという設定であった．ここでは，S 個のサーバ全てが利用中のときに到着した客は 1 個のサーバが空くまで待機し，空いたサーバのサービスを受けるという M/M/S/∞ 待ち行列を考える．具体的には，M/M/S/∞ 待ち行列は以下の要素から成る．

- A=M: 到着時間の間隔がパラメータ λ の指数分布．
- B=M: サービス時間の分布がパラメータ μ の指数分布．
- C=S: サーバの個数が S 個．
- K=∞: 到着した客はサービスを受けるまで待機する．

M/M/S/S 待ち行列とは異なり，サービスを受けずに帰る客はいないため，システム評価にはロス率を用いず，代わりに '到着した客の平均待ち時間' を用いる．

$\mu > \lambda$ の M/M/1/∞ 待ち行列における '平均待ち時間' をシミュレーションせよ．レポートには，平均待ち時間のシミュレーションの数値結果の図とシミュレーション結果の考察を記載せよ．シミュレーションで用いたパラメータの値（λ や μ など）を明記すること．複数の λ と μ に対して数値結果を取り，結果の報告および考察を行うこと．シミュレーション結果の考察には，数値結果と理論値の比較を含めること．平均待ち時間の理論値はウェブや本などで調べること．

レポート提出方法

以下を kibaco の課題ページから提出すること．

- 課題の内容を記載したレポート．
  - 課題 4-1, 4-2, 4-3, 4-4 の内容は必須である．各課題について，回答が正しくない場合でも，自力でコードの実装・数値結果の出力・結果の検証と考察がされていれば，部分点が付与される．なお，白紙や無回答の場合，その課題の得点は 0 点となる．
  - ファイル名は “apl_pp_xxx_yyy.pdf” とする．xxx には学修番号，yyy には氏名を入力する．例えば学修番号が 21012345 で氏名が Kazuki Nakajima の場合，“apl_pp_21012345_kazuki_nakajima.pdf” とする．漢字やひらがなをファイル名に含めないこと．
- 各課題を解くために使用したソースコード．
  - ソースコードが複数に分かれている場合は，zip ファイルにまとめて提出すること．
  - 各ソースコードが，誰のもので，どの課題に対応しているかわかるようにファイル名を設定すること．例えば，氏名が Kazuki Nakajima で課題 2-1 に対する C++ コードのファイル名は “kadai_2_1_kazuki_nakajima.cpp” とする．

注意事項

- **本課題の提出締め切りは 1 週間後（2024/11/20）の 12:00 である．**
- レポート作成時には，テクニカルライティングの資料を確認し，全体をよく検証してから提出すること．