

Level-Crossing 法による GI/G/1 バケーション待ち行列モデルの解析

小林 香・片山 勁

(工学部電子情報工学科)

日常生活で頻繁に発生する、有限個の資源を複数の利用者で使用するという状況では、利用者は誰も自分が資源を利用できるまでの待ち時間を出来るだけ短くしたいという要求がある。このような状況を定量的に解析するのに、有効な手段として待ち行列理論が挙げられる。以前に解析したもっとも基本的な GI/G/1 待ち行列システムでは、システム内利用者数がゼロで待機状態にある処理装置は、新たな利用者が到着した時点でサービスを開始し、利用者を待たせることはない。しかし、処理装置が利用者へのサービスの他にも作業を持つ待ち行列システムでは、到着した利用者により「待ち」が発生する。

本稿では、直感的に理解可能な Level-Crossing 法を用いて、「バケーション」と総称される副次作業付きの GI/G/1 待ち行列モデルにおける待ち時間分布を導出し、その留意点をまとめる。

キーワード : Level-Crossing 法, GI/G/1 バケーション待ち行列モデル, 残余仕事量, 待ち時間分布

1 GI/G/1 待ち行列モデル及び Level-Crossing 法によるその解析

Kendall の記号を用いて GI/G/1 と表記される待ち行列システム [1] は、その処理時間分布が一般分布 $H(t)$ に従う単一の処理装置 (サーバ) と、処理装置が他の利用者により使用中である時、自分の処理の順番が廻ってくるまで利用者が待つことのできる無限長の待ち行列 (バッファ) から構成される。利用者のシステムへの到着間隔が独立でかつ同一の一般分布に従う到着過程であることが要請されるが、その分布関数としては何ら具体的な関数形が与えられる必要はない。まずは、この基本的な待ち行列システムを例にとり、Level-Crossing 法による解析方法を示す。

今、到着順 (FIFO) に処理される GI/G/1 待ち行列モデルに対する、利用者の適当な到着状況を仮定する。利用者がシステムに到着する毎に、その利用者へのサービス時間に相当する仕事量をシステムに持ち込むことになる。システム内に存在する全仕事量 W_t の時系列の一例を描画すると図 1 のようになる。このような図をサンプルパスと称する。観測開始時刻 $t = 0$ でのシステム内利用者数 0 を初期

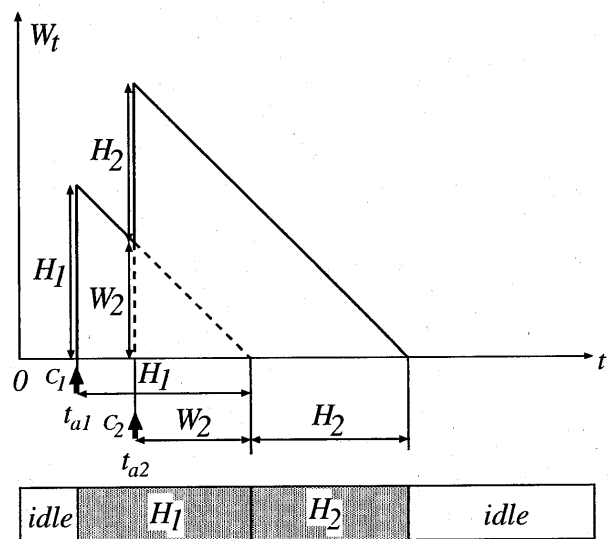


図 1: sample path

状態とし、1 番目、2 番目に到着した利用者 (c_1, c_2) の到着時刻を各々 t_{a1}, t_{a2} とする。 c_1 の到着時点ではシステム内利用者数 0 のため、 c_1 は待つことなく処理を受ける (待ち時間 $W_1 = 0$)。 t_{a1} 時点でのシステムの仕事量 W_t は c_1 の持ち込んだ仕事量 H_1 だけ増加する。時間の経過とともに

処理装置が処理した分、仕事量 W_t は減少する。 c_1 の処理中に到着する c_2 は時刻 ta_2 に H_2 の仕事量をシステムにもたらし、自分に対する処理が開始されるまでに W_2 待つことになる。つまり、ある利用者の到着時点におけるシステムの全仕事量 W_t が、その利用者が処理を受けるまでに必要な待ち時間であることが分かる。また図1の下部に示すように、GI/G/1 待ち行列システムにおいては、処理装置の取りうる状態は、利用者に対するサービス中(図中ハッチ部分)か待機中(図中「idle」部分)のどちらかであることがわかる。

ここで、システムが安定に存在し、利用者の待ち時間が発散しないために、処理装置の稼働状況すなわちシステムの利用率 $\rho = \lambda E(H) < 1$ であることが必要である。

図1のサンプルパスにおいて、 $W_t = x (x > 0)$ となる直線を設定する。観測時間 $(0, T)$ においてシステムの仕事量 W_t がこのレベル x の直線を、上から下に横切る回数を $N_{down}(T)$ 、下から上に横切る回数を $N_{up}(T)$ と表す。 $N_{down}(T)$ と $N_{up}(T)$ は高々1しか異なることはなく、十分長い間観測することでその時間的な平均は等しいものとなる。

$$\lim_{T \rightarrow \infty} \frac{N_{down}(T)\Delta t}{T} = \lim_{T \rightarrow \infty} \frac{N_{up}(T)\Delta t}{T} \quad (1)$$

エルゴード性により $\lim_{T \rightarrow \infty} \{N_{down}(T)\Delta t/T\}$ 及び $\lim_{T \rightarrow \infty} \{N_{up}(T)\Delta t/T\}$ は全体の観測時間 T に対して、システム内仕事量が x から $x + \Delta x$ の間にある時間的な割合を示している。この式(1)がLevel-Crossing法で重要な、レベル x における平衡方程式で、対象とする待ち行列モデルに合わせて式(1)の右辺と左辺の解析的表現を得る作業が必要となる。

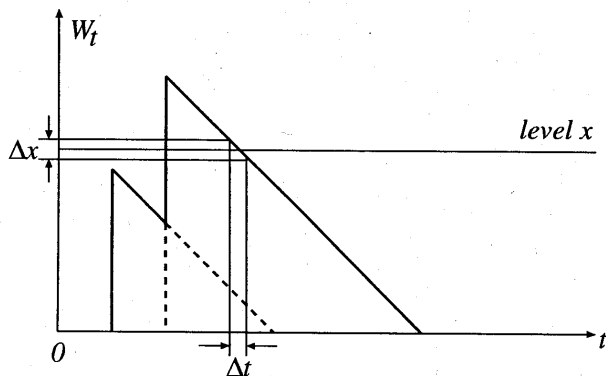


図 2: downward crossing case

観測時間 $(0, T)$ において、処理装置が利用者进行处理することでシステムの仕事量 W_t がこのレベル x の直線を、上から下に横切る場合、図2に示すように、横切る微小時間

Δt の間に仕事量は Δx 変化する。定義より、残余仕事量に対応する待ち時間 W_t の確率密度関数 $w_t(x)$ は $w_t(x)\Delta x \equiv Pr[x < W_t \leq x + \Delta x]$ であるから、

$$\lim_{T \rightarrow \infty} \frac{N_{down}(T)\Delta t}{T} = w_t(x)\Delta x \quad (2)$$

を得る。

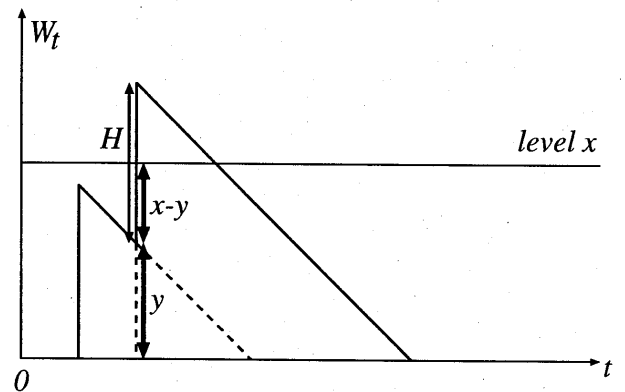


図 3: upward crossing case

次に観測時間 $(0, T)$ においてシステムの仕事量 W_t がこのレベル x の直線を、下から上に横切る場合を考える。GI/G/1 システムにおいて想定されるケースは、新たな利用者のシステム到着時の残余仕事量(レベル x より下)と、その利用者がシステムにもたらし仕事量の和がレベル x を越える場合である。つまりある利用者の到着時の残余仕事量を $y (y \leq x)$ とすると、この利用者のもたらし仕事量である処理時間 H が $(x - y)$ より大きいと、残余仕事量のパスはレベル x を横切ることになる。また、新規に利用者が到着してその処理時間に対応する仕事量をシステムにもたらしことを既に述べたが、処理時間がゼロとなる利用者の到着はありえない(そのような利用者は到着していないのと同じ扱いになる)ことから、条件 $x > 0$ を考慮する必要がある。確率 $Pr[H > (x - y)]$ は処理時間分布 $H(x)$ の補分布 $\bar{H}(x) = 1 - H(x)$ を用いて以下のように表現される。

$$Pr[H > (x - y)] = \bar{H}(x - y) \quad (3)$$

利用者の到着時点での待ち時間 W_a が y と $y + \Delta y$ の間にある確率は、確率密度関数 $w(y)$ を用いて、

$$w(y)\Delta y = Pr[y < W_a \leq y + \Delta y] = dW(y) \quad (4)$$

となる。従って観測時間 $(0, T)$ において単位時間当たり平均 λ の利用者が到着すると、レベル x を下から上に横切る回

数は

$$N_{up}(T) = \lambda T \int_0^x \bar{H}(x-y) dW(y) \quad (5)$$

であり、変形すると以下ようになる。

$$\lim_{T \rightarrow \infty} \frac{N_{up}(T)}{T} = \lambda \int_0^x \bar{H}(x-y) dW(y), (x > 0) \quad (6)$$

得られた $\lim_{T \rightarrow \infty} \frac{N_{down}(T)}{T}$ 及び $\lim_{T \rightarrow \infty} \frac{N_{up}(T)}{T}$ を式 (1) に代入する。この時、処理装置が処理を行うことによるシステムの仕事量 W_t の変化率は $\Delta x / \Delta t = 1$ であることから、

$$\lim_{T \rightarrow \infty} \frac{N_{down}(T)}{T} = w_t(x) \quad (7)$$

と置換できる。ゆえに以下のような関係式を得る。

$$w_t(x) = \lambda \int_0^x \bar{H}(x-y) dW(y) \quad (x > 0) \quad (8)$$

この式 (8) を解くために、確率分布 $W_t(x)$ の LST $W_t^*(s)$ を定義する。

$$\begin{aligned} W_t^*(s) &\equiv \int_{0-}^{\infty} e^{-sx} dW_t(x) \quad (x \geq 0) \\ W_t^*(s) - W_t(0) &= \int_{0+}^{\infty} e^{-sx} dW_t(x) \quad (x > 0) \end{aligned} \quad (9)$$

この $W_t(0)$ は待ち時間がゼロ、つまり利用者の到着時にはシステムが待機状態にあり、待ち合わせを受けることなく直ちに処理が開始される確率である。式 (9) に式 (8) を代入し展開する。補分布のラプラス変換については [3] 附録を参照されたし。

$$\begin{aligned} W_t^*(s) - W_t(0) &= \int_{0+}^{\infty} e^{-sx} \left[\lambda \int_0^x \bar{H}(x-y) dW(y) \right] dx \\ &= \lambda \cdot \mathcal{L}[\bar{H}(x)] \cdot W^*(s) \\ &= \lambda \cdot \left(\frac{1 - H^*(s)}{s} \right) \cdot W^*(s) \end{aligned} \quad (10)$$

整理して式 (11) を得る。

$$W_t^*(s) - \lambda \cdot \left(\frac{1 - H^*(s)}{s} \right) \cdot W^*(s) = W_t(0) \quad (11)$$

式 (11) において $s \rightarrow 0$ の極限を取り、 $W^*(s)|_{s=0} = 1$,

$$W_t^*(s)|_{s=0} = 1 \text{ から得られる } W_t(0)$$

$$W_t(0) = 1 - \lambda E(H) = 1 - \rho \quad (12)$$

を式 (10) に代入して、GI/G/1 待ち行列システムの待ち時間の表式を得る。

$$W_t^*(s) - \lambda \cdot \left(\frac{1 - H^*(s)}{s} \right) \cdot W^*(s) = 1 - \rho \quad (13)$$

式 (13) を変形して式 (14) を得る。

$$W_t^*(s) = (1 - \rho) + \rho \left(\frac{1 - H^*(s)}{s E(H)} \right) \cdot W^*(s) \quad (14)$$

式 (14) の右辺第一項はシステム内利用者数ゼロに伴う処理装置の待機状態、第二項は利用者が到着した時点のシステム内の残余仕事量が現時点においてどの程度減少しているかを示している。図 4 に示すように、式 (14) の左辺は任意時刻でのシステム内残余仕事量 $W_t^*(s)$ になる。

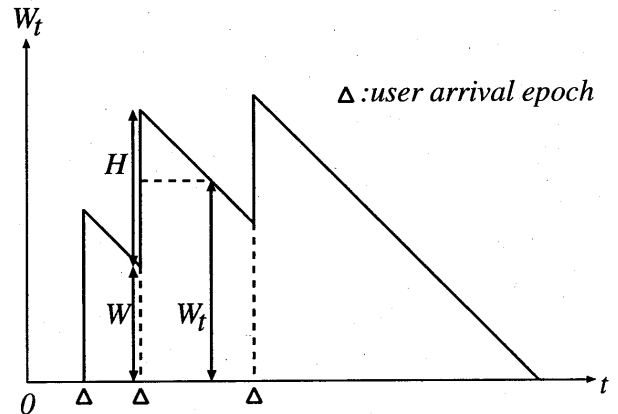


図 4: virtual waiting time and waiting time

これまで想定してきた、利用者の到着過程が一般分布である GI/G/1 システムにおいては、任意時点における待ち時間分布 $W_t(x)$ 、利用者の到着時における待ち時間分布 $W(x)$ 、利用者がシステムを退去する時点での待ち時間分布は必ずしも一致するものではない。しかし、利用者の到着過程がポアソン過程である M/G/1 システムにおいては、PASTA(Poisson Arrivals See Time Averages)[4] の性質により、これら三者が一致することが保証されている。この性質を利用するために、以後想定するシステムを M/G/1 とする。 $W_t^*(s) = W^*(s)$ を式 (14) に代入して、式 (15) の待

ち時間の分布関数の LST $W^*(s)$ を得る.

$$W^*(s) = \frac{s(1-\rho)}{s-\lambda+\lambda H^*(s)} \quad (15)$$

さらに待ち時間の確率密度関数に関する以下の式を得る.

$$w(x) = \lambda \int_0^x \bar{H}(x-y) dW(y) \quad (x > 0) \quad (16)$$

母関数を用いる解析方法では, LST を n 回微分する数学的操作で待ち時間の n 次積率を得ることができるが, その確率密度関数の形を具体的に得ることはできない. しかし, Level-Crossing 法では式 (16) のように表現ができるのである.

2 GI/G/1 バケーション待ち行列モデル

第 1 章で示した基本的な GI/G/1 待ち行列システムにおいては, システム内利用者数が 1 以上の場合処理装置は「サービス中」であり, システム内利用者数がゼロの場合は「待機中」である. また, 待機状態にある処理装置は, 新規の利用者の到着後直ちにサービスを開始する.

しかし, 待機状態にある処理装置の利用効率を高めることを目的として, 利用者に対するサービス以外の副次的な作業を処理装置にさせることを考えると, システム内利用者数が 1 以上なのに処理装置はサービス中ではない第 3 の状態が発生する. 例えば, システム内利用者数がゼロになった時点で処理装置がバッチ的に副次的作業を行うケースを考える. 1つのバッチ処理を行っているときに新規の利用者が到着しても, そのバッチ処理が終了するまではその利用者に対するサービスを開始することができない. この第 3 の状態を「バケーション」と総称する.

本章では, 次の 3 つの GI/G/1 バケーション待ち行列モデルを対象とし, Level-Crossing 法による解析上, 基本的な GI/G/1 待ち行列モデルとの各々の相違点を明確にしつつ, 待ち時間分布を導出する.

1. マルチプル・バケーションモデル
2. シングル・バケーションモデル
3. セットアップモデル

これらの待ち行列モデルにおいて, 系内利用者数に着目して確率母関数から待ち時間分布を得る方法については Takagi [2] が簡潔にまとめている.

2.1 GI/G/1 マルチプル・バケーションモデル

本節では GI/G/1 マルチプル・バケーションモデルを扱う. これは, システム内利用者数が 0 になった時点で処理装置がバッチ的に副次的作業を行うケースに相当する. バッチ処理が 1 セット終了した時点で新規の利用者の到着がない場合は次のセットのバッチ処理を開始し, バッチ処理中に新規の利用者の到着があれば利用者を待たせ, バッチ処理終了後利用者へのサービスを開始することになる. このモデルでのサンプルパス及び処理装置の取りうる状態を図 5 に示す. バケーション時間 V は一般分布 $V(t)$ に従い, そ

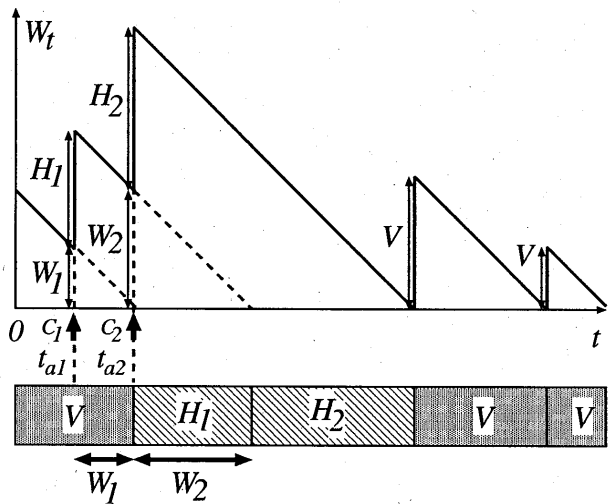


図 5: sample path of GI/G/1 multipul vacation model

の LST 及び n 次積率を $V^*(s)$, $E(V^n)$ とする. 処理装置がバケーション状態にあることを図 5 下部の黒のハッチ部分で示す. 処理装置がバケーション中に到着した利用者 c_1 の待ち時間 $W_1 \neq 0$ であることが GI/G/1 モデルとの違いである.

図 1 に示す GI/G/1 モデルのサンプルパスと図 5 に示す GI/G/1 マルチプル・バケーションモデルのサンプルパスを比較すると, レベル x に対してサンプルパスが下向きに横切る頻度の計算方法には変わりがなく, 式 (2) と同様に残余仕事量 W_t の密度関数を用いて以下のように表される.

$$\lim_{T \rightarrow \infty} \frac{N_{down}(T)\Delta t}{T} = w_t(x)\Delta x \quad (17)$$

レベル x に対してサンプルパスが上向きに横切る頻度を計算する際には, 利用者の到着によるシステムへの仕事量の持ち込みの他に, 処理装置がバケーション状態に移行し

た時点で、システムに持ち込まれるバケーション時間 V に相当する仕事を考慮しなくてはならない。このバケーションを仮想的に利用者（以後、仮想利用者と呼ぶ）と見なし、その平均到着間隔を $1/\lambda_v$ とする。図1に示す $GI/G/1$ モデルに対して、仮想利用者による仕事量 V によりサンプルパスがレベル x を上向きに横切る回数が新たに加わるため、式(5)の右辺にこの効果を組み入れる。

$$\lim_{T \rightarrow \infty} \frac{N_{up}(T)\Delta t}{T} = \lambda \int_0^x \bar{H}(x-y)dW(y) + \lambda_v \bar{V}(x) \quad (x > 0) \quad (18)$$

式(17), (18)を式(1)に代入し、レベル x に関する平衡方程式を得る。

$$w_t(x) = \lambda \int_0^x \bar{H}(x-y)dW(y) + \lambda_v \bar{V}(x) \quad (x > 0) \quad (19)$$

第1章で述べたように、確率分布 $W_t(x)$ のLST $W_t^*(s)$ を求め、

$$W_t^*(s) - W_t(0) = \lambda \cdot \left(\frac{1 - H^*(s)}{s} \right) \cdot W^*(s) + \lambda_v \cdot \frac{1 - V^*(s)}{s} \quad (20)$$

$s \rightarrow 0$ の極限操作により

$$1 - W_t(0) = \lambda E(H) + \lambda_v E(V) \quad (21)$$

の表式を得る。ここから以下のように $W_t^*(s)$ を得る。

$$W_t^*(s) = (1 - \rho) + \rho \cdot \left(\frac{1 - H^*(s)}{sE(H)} \right) \cdot W^*(s) + \lambda_v E(V) \cdot \left(\frac{1 - V^*(s)}{sE(V)} - 1 \right) \quad (22)$$

$$= \left(1 - \rho - \lambda_v E(V) \right) + \lambda_v E(V) \cdot \frac{1 - V^*(s)}{sE(V)} + \rho \cdot \left(\frac{1 - H^*(s)}{sE(H)} \right) \cdot W^*(s) \quad (23)$$

図5に示される通り $GI/G/1$ マルチプル・バケーションモデルでは、処理装置が待機状態になることはなく、したがって到着した利用者の待ち時間がゼロとなる確率 $W_t(0)$ はゼロであることから、仮想利用者の平均到着率 $\lambda_v = (1 - \lambda E(H))/E(V)$ が導かれる。

利用者の到着過程がポアソン過程である $M/G/1$ マルチプル・バケーションモデルでは、PASTA から $W_t^*(s) = W^*(s)$ であることを用いて、 $W^*(s)$ を得ることができる。

$$W^*(s) = \frac{s(1 - \rho)}{s - \lambda + \lambda H^*(s)} \cdot \frac{1 - V^*(s)}{sE(V)} \quad (24)$$

式(24)の右辺の積項前半部は式(15)の $M/G/1$ モデルの $W^*(s)$ と同一であり、したがってバケーションの効果は同積項の後半部分に集約されている。

2.2 $GI/G/1$ シングル・バケーションモデル

$GI/G/1$ シングル・バケーションモデルは、システム内利用者数が0になった時点で処理装置が終了処理を行うケースに相当する。2.1との違いは、システム内利用者数が0になった時に1回だけ処理装置がバケーション状態となり、それが終了すると待機状態になる点である。このモデルでのサンプルパス及び処理装置の取りうる状態を図6に示す。バ

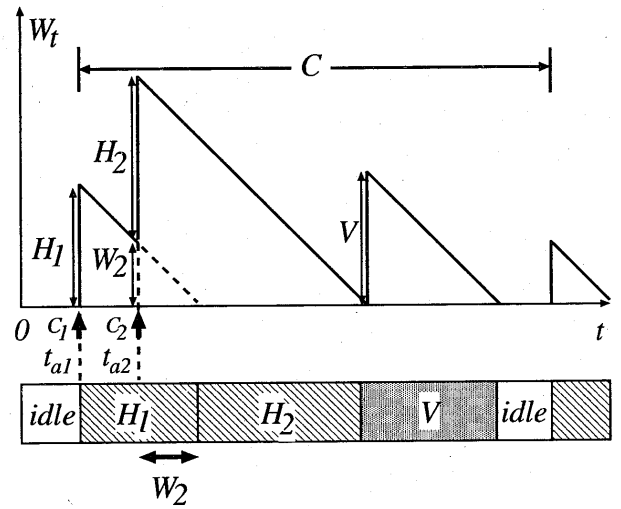


図6: sample path of $GI/G/1$ single vacation model

ケーション時間 V は一般分布 $V(t)$ に従い、そのLST及び n 次積率を $V^*(s)$, $E(V^n)$ とする。処理装置がバケーション状態にあることを図6下部の黒のハッチ部分で示す。

$GI/G/1$ シングル・バケーションモデルでのレベル x に関する平衡方程式は $GI/G/1$ マルチプル・バケーションモデルでの式(19)と同じく、

$$w_t(x) = \lambda \int_0^x \bar{H}(x-y)dW(y) + \lambda_v \bar{V}(x) \quad (x > 0) \quad (25)$$

と与えられる。また、得られる $W_t^*(s)$ は式(23)と同じく

$$W_t^*(s) = \left(1 - \rho - \lambda_v E(V) \right) + \lambda_v E(V) \cdot \frac{1 - V^*(s)}{sE(V)} + \rho \cdot \left(\frac{1 - H^*(s)}{sE(H)} \right) \cdot W^*(s) \quad (26)$$

である。 $GI/G/1$ マルチプル・バケーションモデルとの違

いは、仮想利用者の平均到着間隔 $1/\lambda_v$ と、利用者及び仮想利用者がシステム内に存在しない時に処理装置が待機状態を取ること、つまり $W_t(0) \neq 0$ であることの二点にある。したがって、式 (25) を解いて $W_t^*(s)$ を得るには、 λ_v と到着した利用者の待ち時間がゼロになる確率 $W_t(0)$ を求める必要がある。

図 6 に示すように、処理装置が取る一連の状態変化 (待機状態→サービス状態→バケーション状態) の周期 C の平均を $E(C)$ とおく。この 1 周期中に 1 回のバケーション状態が含まれるので、仮想利用者の平均到着間隔に対応する。

$$\frac{1}{\lambda_v} = E(C) \quad (27)$$

処理装置が待機状態にあるときにシステムに到着した利用者の待ち時間はゼロになる。この利用者の待ち時間がゼロである確率 $W_t(0)$ は、サービス開始から始まる 1 周期の中で $(1/E(C))$ 、利用者に対するサービスが終了 (サービス状態→バケーション状態へ移行: $1/\lambda$) し、仮想利用者に対するサービス (バケーション) が終了 (バケーション状態→待機状態へ移行) した時点でシステム内の利用者数がゼロである確率 (q_0 とする) として計算できる。

$$W_t(0) = \frac{1}{E(C)} \times \frac{1}{\lambda} \times q_0 \quad (28)$$

また、平均周期 $E(C)$ は平均サービス時間、平均バケーション時間及び平均待機時間の和であることから、次の式を得る。

$$\begin{aligned} E(C) &= \frac{\rho}{1-\rho} \left(E(V) + \frac{q_0}{\lambda} \right) + E(V) + \frac{q_0}{\lambda} \\ &= \frac{1}{1-\rho} \left(E(V) + \frac{q_0}{\lambda} \right) \end{aligned} \quad (29)$$

式 (27), (29) より $E(C)$ を消去する。あとは、バケーション終了時点でシステム内の利用者数がゼロである確率 q_0 が得られれば、 λ_v および $W_t(0)$ を解析的に得ることができる。

$$\lambda_v = \frac{\lambda(1-\rho)}{q_0 + \lambda E(V)}, \quad W_t(0) = \frac{q_0(1-\rho)}{q_0 + \lambda E(V)} \quad (30)$$

利用者の到着過程が一般分布 (GI) である場合は、ここまでの数式表現となるが、具体的に待ち時間分布 $W(t)$ の LST $W^*(s)$ を得るために、2.1 節と同様に利用者の到着過程を平均到着率 λ のポアソン過程とする。バケーション時間が $V = x$ という条件下で、システムに到着する利用者数

$N = k$ である確率 $P(N = k|V = x)$ は

$$P(N = k|V = x) = \frac{(\lambda x)^k}{k!} e^{-\lambda x}, \quad (31)$$

また、バケーション時間が $V = x$ である確率 $P(V = x)$ は

$$P(V = x) = P(x < V \leq x + dx) = dV(x) \quad (32)$$

となる。ゆえにバケーション時間が $V = x$ かつシステムに到着する利用者数 $N = k$ である確率 $P(N = k, V = x)$ は、式 (31) と (32) の積から得る。これを、 $x = 0 \sim \infty$ 区間で積分し、

$$P(N = k) = \int_0^\infty \frac{(\lambda x)^k}{k!} e^{-\lambda x} dV(x) \quad (33)$$

求める q_0 はバケーション時間 V 間に 1 人も利用者が到着しない確率なので、式 (33) において $k = 0$ を代入し、

$$\begin{aligned} q_0 &= P(N = 0) \\ &= \int_0^\infty e^{-\lambda x} dV(x) \\ &= V^*(\lambda) \end{aligned} \quad (34)$$

を得る。

既に導入した、ポアソン到着過程による $W_t^*(s) = W^*(s)$ (PASTA)、および λ_v , $W_t(0)$ を式 (20) に代入することで、 $M/G/1$ シングル・バケーションモデルの待ち時間分布の LST $W^*(s)$ を得る。

$$W^*(s) = \frac{s(1-\rho)}{s-\lambda + \lambda H^*(s)} \cdot \frac{\lambda - \lambda V^*(s) + sV^*(\lambda)}{sV^*(\lambda) + s\lambda E(V)} \quad (35)$$

式 (35) の右辺の積項前半部は式 (15) の $M/G/1$ モデルの $W^*(s)$ と同一であり、したがってバケーションの効果は同積項の後半部分に集約されている。

2.3 GI/G/1 セットアップモデル

GI/G/1 セットアップモデルは、一度待機状態になった処理装置がサービスを開始するまでに起動処理を必要とするケースに相当する。GI/G/1 モデルであれば、処理装置が待機状態の時に到着した利用者は直ちにサービスを受けられるが、本モデルでは、起動時間 S だけ待つことになる。このモデルでのサンプルパス及び処理装置の取りうる状態を図 7 に示す。起動時間 S は一般分布 $S(t)$ に従い、その密度関数、LST 及び n 次積率を各々 $s(t)$, $S^*(s)$, $E(S^n)$ とする。処理装置が起動処理状態にあることを図 7 下部の黒の

ハッチ部分で示す。システム内利用者数がゼロではないのに処理装置が利用者にサービスを行わない時間帯(バケーション)があるという点では、2.1, 2.2節で述べたバケーションモデルと変わりはない。

図6に示すGI/G/1シングル・バケーションモデルのサンプルパスと図7に示すGI/G/1セットアップモデルのサンプルパスを比較すると、レベル x に対してサンプルパスが上向きに横切る頻度を計算する際には、利用者の到着によるシステムへの仕事量の持ち込みの他に、処理装置がバケーション状態に移行した時点で、システムに持ち込まれる起動時間 S に相当する仕事量を考慮することに違いはない。同様に起動準備時間に入る平均頻度を λ_s とし、レベル x に対する平衡方程式を立てる。

GI/G/1セットアップモデルにおいて、サンプルパスがレベル x を上向きに横切る頻度を求める際に留意する点は、仮想利用者による仕事量を表す変数が V から S に変化したことである。

$$\lim_{T \rightarrow \infty} \frac{N_{up}(T)\Delta t}{T} = \lambda \int_0^x \bar{H}(x-y) dW(y) + \lambda_s \bar{S}(x) \quad (x > 0) \quad (36)$$

逆に、レベル x に対してサンプルパスが下向きに横切る頻度の計算において、残余仕事量 W_t の密度関数はサンプルパスが下向きに横切る頻度のほかに、仮想利用者による残余起動時間の密度関数を考慮する必要がある。ここで p_0 をシステム内利用者数がゼロである確率とする。

$$P(x < W_t \leq x + dx) = \lim_{T \rightarrow \infty} \frac{N_{down}(T)\Delta t}{T} + p_0 P(x < S \leq x + dx) - p_0 \delta(x) \quad (37)$$

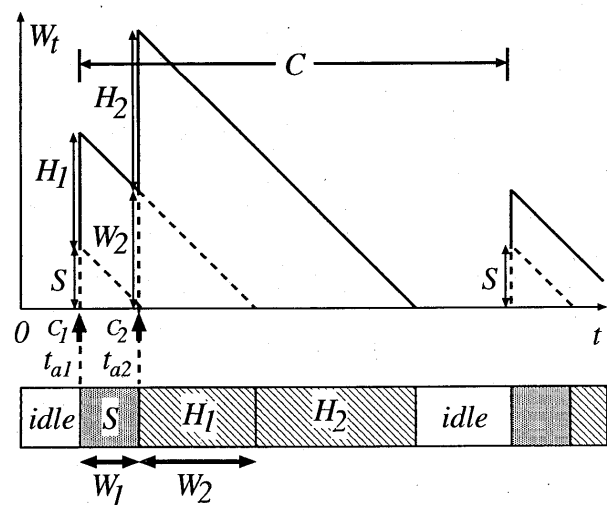


図7: sample path of GI/G/1 setup time model

右辺第二項の $P(x < S \leq x + dx)$ は仮想利用者の到着時点でのみ定義されている確率であるのに対して、左辺 $P(x < W_t \leq x + dx)$ は任意時点で定義されている残余仕事量に関する確率となっている。 $P(x < S \leq x + dx)$ を任意時点で定義される確率に補正するために第三項 $p_0 \delta(x)$ を加えている(図8参照)。

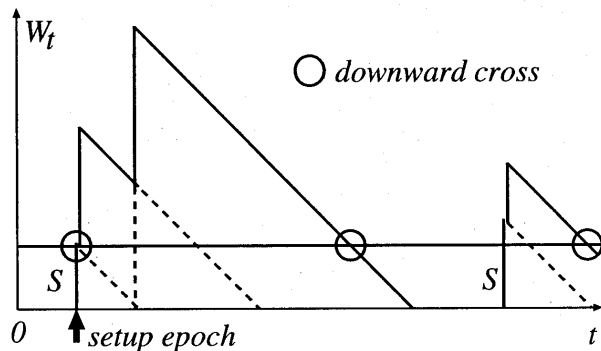


図8: downward cross at level x

密度関数 $w_t(x)$, $s(x)$ および $\Delta x/\Delta t = 1$ を用いて整理して、

$$w_t(x) = \lim_{T \rightarrow \infty} \frac{N_{down}(T)}{T} + p_0 s(x) - p_0 \delta(x) \quad (38)$$

を得る。式(36), (38)を平衡方程式(式(1))に代入し、整理して得た式(ただし、 $x > 0$)

$$w_t(x) - p_0 s(x) + p_0 \delta(x) = \lambda \int_0^x \bar{H}(x-y) dW(y) + \lambda_s \bar{S}(x) \quad (39)$$

を変換し、LSTを求める。

$$\begin{aligned} & (W_t^*(s) - W_t(0)) - p_0 (S^*(s) - S(0)) + p_0 \\ &= \lambda \cdot \left(\frac{1 - H^*(s)}{s} \right) \cdot W^*(s) + \lambda_s \cdot \frac{1 - S^*(s)}{s} \quad (40) \end{aligned}$$

処理装置に起動処理があるために本モデルでは $W_t(0) = p_0$ である。また $S(0) = 0$ を仮定して、式(40)を整理する。その $s \rightarrow 0$ の極限では

$$1 - p_0 = \lambda_s E(S) + \rho \quad (41)$$

を変換し、LSTを求める。

$$\begin{aligned} & W_t^*(s) - p_0 S^*(s) \\ &= \lambda \cdot \left(\frac{1 - H^*(s)}{s} \right) \cdot W^*(s) + \lambda_s \cdot \frac{1 - S^*(s)}{s} \quad (42) \end{aligned}$$

処理装置が取る一連の状態変化 (待機状態→サービス状態→バケーション状態) の周期 C の平均を $E(C)$ とおく. この 1 周期中に 1 回の起動処理状態が含まれるので, 仮想利用者の平均到着間隔に対応する.

$$\frac{1}{\lambda_s} = E(C) \quad (43)$$

具体的に待ち時間分布の LST $W^*(s)$ を得るために, 2.1 節と同様に利用者の到着過程を平均到着率 λ のポアソン過程とする. このとき到着間隔分布が指数分布となり, 到着過程の無記憶性より

$$p_0 = \frac{1/\lambda}{E(C)} \quad (44)$$

を得る. これは待機状態をとる時間の 1 周期に占める割合である. λ_s と p_0 を式 (41) に代入し $E(C)$ を求め, λ_s と p_0 を既知の値で書き直す.

$$\begin{aligned} p_0 &= \frac{1-\rho}{1+\lambda E(S)} \\ \lambda_s &= \frac{\lambda(1-\rho)}{1+\lambda E(S)} \\ E(C) &= \frac{1+\lambda E(S)}{\lambda(1-\rho)} \end{aligned} \quad (45)$$

先に, 利用者の到着過程をポアソン過程としたので, PASTA から $W_t^*(s) = W^*(s)$ となり, λ_s と p_0 から

$$W^*(s) = \frac{s(1-\rho)}{s-\lambda+\lambda H^*(s)} \cdot \frac{S^*(s) + \frac{1-S^*(s)}{s}\lambda}{1+\lambda E(S)} \quad (46)$$

以上のように $M/G/1$ セットアップモデルの待ち時間分布の LST $W^*(s)$ を得る. 式 (46) の右辺の積項前半部は式 (15) の $M/G/1$ モデルの $W^*(s)$ と同一であり, したがって起動処理 (バケーション) の効果は同積項の後半部分に集約されている.

3 まとめ

第 1 章で述べた様に, 待ち行列を含むシステムの常套的な解析法として, ある利用者の処理終了直後における系内利用者数に着目する隠れマルコフ連鎖法や, 系内利用者数の他に残余処理時間 (あるいはサービス経過時間) を状態変数として取り入れる補助変数法が一般的である. しかし, これらの方法は, 待ち時間という時間に関する表現を得るために, それに直接関与することのない利用者の待ち数あるいは系内利用者数に関する, 物理的に解釈が困難である母

関数を用いており, 計算を進めれば解を得られるものの直感的な理解が難しいという問題点を持つ.

それに対して Level-Crossing 法を導入する場合, 連続量であるシステム内残余仕事量 (残余処理時間) W_t に関する平衡方程式を得ることで, 定常状態における待ち時間分布の LST $W^*(s)$ を容易に得ることができる. この際に必要な数学的手法は単純な微分及び積分であり, 数学的困難な点はほとんど見られない.

Level-Crossing 法では, 直接的に系内利用者数に関する情報を得ることは出来ないが, ほぼ最短の手間で待ち時間に関する量を得ることができるということは特筆すべきことである. また, 式 (16) を解析的あるいは数値的に計算することで, 待ち時間の確率密度関数を得ることができ, 平均待ち時間のみならず待ち時間の揺らぎを直接評価できる. 実時間性を重要視するシステムにおいては, 待ち時間の揺らぎを定量的に評価することで, システムの性能を数値的に表現することが可能となる.

Level-Crossing 法の骨子は, 任意時点での残余仕事量 W_t が, 利用者の到着時点でその利用者のもたらす仕事量つまり処理時間だけ増加し, 残余仕事量がゼロでない限り処理装置により処理された仕事量だけ減少するサンプルパスにおける平衡方程式である. 仕事量つまり処理時間という時間量で評価するということは, 利用者数という離散量でしか測ることが出来なかった場合と比べると, 精密に評価できるという利点があることになる. $M/G/1$ 待ち行列モデル [3], 今回の $M/G/1$ バケーション待ち行列モデルへの適用を通じて, Level-Crossing 法を適用する際の留意点を以下にまとめる.

1. 処理装置がバケーションを取ることによる仕事量の増加は仮想利用者の到着と見なすことで, Level-Crossing 法を適用することが可能である. ただし, この時の仮想利用者の平均到着間隔は何らかの手段で与える必要があり, このときに待ち行列システムにおける知識が別途必要になる.
2. $M/G/1$ マルチプル・バケーションモデルと $M/G/1$ シングル・バケーションモデルとでは, 得られる平衡方程式が同一である. モデルによる違いを平衡方程式に繰り入れるには, 待ち行列システムの知識が不可欠である. また系内利用者数が 0 ではないのに処理できない「バケーション」状態の時の利用者の到着過程が明確にするために, 早くからポアソン到着過程の導入が必要となる.
3. 仕事量の増加分が「利用者」の到着時点でのみ評価されるということは, 利用者のシステム到着後にシステム内状態に依存して処理時間が変化するようなシステ

ム ([5],[6]) は, そのままの形では Level-Crossing 法で取り扱うことができないということである. 例えば, 利用者の処理終了時点で, 確率的に次の利用者の処理状況を決定するベルヌーイ式の場合は, 利用者の処理終了時点の残余仕事量の変化が, 利用者の到着時点で発生するようにサンプルパスを修正することで対応することができる. またトラヒック規制手段の一つとして, システム到着からある規定時間を越えた利用者をシステムから離脱させる待ち時間制御方式を取入れたシステムや到着順 (FIFO) でない処理規律のシステムも, そのままでは Level-Crossing 法で扱うことができないケースになる. その際には, Level-Crossing 法で扱うことの出来るよう, サンプルパスの修正及び等価モデルへ落す作業が必要となる.

今後, 様々な待ち行列システムにおいて Level-Crossing 法の適用を継続的に検討し, 他に配慮の必要な事項があるかを調査を進める.

参考文献

- [1] 藤木 正也, 雁部 颯一 (1980): 通信トラヒック理論, 丸善.
- [2] H. Takagi(1991): *Queueing Analysis, Vol.1 Vacation and Priority Systems, Part 1*, North-Holland.
- [3] 小林 香, 片山 勁 (2002): Level Crossing 法による GI/G/1 待ち行列モデルの解析, 富山県立大学紀要, 第12巻, 47-53.
- [4] 秋丸春夫, 河島幸之助 (1990): 情報通信トラヒック-基礎と応用-, オーム社.
- [5] B. T. Doshi (1992): *Level-crossing analysis of queues, Queueing and Related Models*, edited by U. Narayan Bhat and I. V. Basawa, 3-33.
- [6] T. Katayama (1999): A note on sojourn time analysis of a two-stage queueing system, *Stochastic Models*, 15, 298-313.

An Analysis of $GI/G/1$ Queueing Model with Vacation by the Level-Crossing Method

Kaori KOBAYASHI and Tsuyoshi KATAYAMA

Department of Electronics and Informatics, Faculty of Engineering

Summary

A situation is frequently happened in our daily life that a user commonly shares some limited resources with other users or waits until he can use these resources; e.g. at the checkout counter of supermarket. As one of these queueing systems, a $GI/G/1$ queueing model with vacations (denoted by Kendall's notation; GI specifies the arrival process is a renewal arrival process, G specifies the service time distribution is general distribution, and 1 denotes a single server) is a basic queueing system with non-exhaustive service, and has a fundamental and an important role in evaluation of quality of service (QoS) of queueing systems. To analyze a $M/G/1$ (a special case of the $GI/G/1$ queueing model) with vacations, the method of the imbedded Markov chain is commonly used for finding the distribution of the number of users (or clients, customers, messages, e.t.c.) in the system, though an obtained generating function of the distribution of the number of users is expressed in mathematical form and is not easily understandable.

In this paper, taking the Level-Crossing method, we have obtained the probability density function of waiting time in the $GI/G/1$ queueing model with vacations more intuitively than using the generating function of the distribution of the number of users, and discuss this method's validity.

keywords: the level-crossing method, $GI/G/1$ queueing model, virtual waiting time, distribution function of the waiting time