

# ワードプロセッサにおける 4R-7 最適オートセーブ時期の決定法

三道弘明  
流通科学大学

河合 一  
鳥取大学

## 1. はじめに

近年のワードプロセッサはその低価格化により、自身の急速な普及を実現し、業務から個人的趣味に至るまで利用されている。このようなワードプロセッサを用いて作成した文書ファイルは、通常フロッピーディスクに保存される。しかしながら、次に述べるような理由により、苦勞の末に作成した文書を消失してしまうことも少なくない。

- 1) 文書を保存することを忘れて、ワードプロセッサのスイッチを切る。
- 2) 作成中の文書の上に、誤って他の文書を読み込んでしまう。

なお、1)による文書の消失を防止することを目的として、バッテリーによるバックアップ機能を備えていることが多いが、それもバッテリーが切れていないことを前提としたものである。従って、ワードプロセッサ自身にオートセーブ機能を実現し、文書作成中に随時保存していくことで、上述のような文書ファイル消失による損失を少なくすることも必要である。このようなオートセーブ機能は、ソフト的に容易に実現可能である。しかし、そのセーブをいつ実施すればよいかという問題は依然として残されている。

本研究では、以上のような観点から、オートセーブ時期を決定するための定式化を行うとともに、最適オートセーブ時期について考察する。

## 2. 期待費用

ここでは、ワードプロセッサのCPUの記憶容量を1とし、その1/Nを消費した時点で保存することを考える。このとき、Nの値を決定することが問題である。なお、ここでは次のように仮定する。

- 1) 文書を消失することは極めて稀なことであり、CPUメモリを消費し尽くすまで荷文書を消失する回数は平均λのポアソン分布に従う。
- 2) 文書の消失は即座に検出される。
- 3) 保存の最中に文書を消失することはなく、保存に要する時間は無視できるくらいに小さい。

以上のような方策のもとでは、プロセスの振舞は、以下に述べる事象のうちいずれか一つが生起した時点を再生点とする再生過程として捉えることができる<sup>(1)</sup>。

- i) 記憶容量の1/Nを消費し、それまでの文書の保存が完了する。
- ii) 記憶容量の1/Nを消費し終わるまでに文書を消失したため、文書フロッピーディスクから、前回のセーブ時点までに作成した部分の読み込みが完了する。

これらのことから、十分に長時間を考えた場合の単位時間当り期待費用C(N)は次式のようになる。

$$C(N) = \frac{B(N)}{A(N)}, \quad N = 1, 2, \dots \quad (1)$$

ここに、A(N)、B(N)はそれぞれ1サイクルにおける期待時間と期待費用を表す。なお、1

A Method for Determination of an Optimal Auto-Saving Time for a Word Processor  
Hiroaki SANDOH<sup>1</sup>, Hajime KAWAI<sup>2</sup>

<sup>1</sup> University of Marketing & Distribution Sciences, <sup>2</sup> Tottori University

サイクルとは、連続する再生点間を意味する。このとき、仮定 i), ii) より次式が得られる。

$$A(N) = \int_0^{1/N} \bar{F}(x) dx \quad (2)$$

$$C(N) = \int_0^{1/N} c_1 x dF(x) + c_2 \bar{F}(1/N) \quad (3)$$

ここに、

$$F(x) = 1 - \exp(-\lambda x) \quad (4)$$

$$\bar{F}(x) = 1 - F(x) \quad (5)$$

であり、 $c_1$  は記憶容量 1 に相当する文書を消失した場合の損失費用を表し、 $c_2$  はオートセーブ 1 回当りに要する固定費用を表す。

式 (2), (3) を計算し、式 (1) に代入した結果は次のとおりである。

$$C(N) = \frac{c_1(e^{\lambda/N} - \lambda/N - 1) + c_2 \lambda}{e^{\lambda/N}} \quad (6)$$

### 3. 最適オートセーブ時期

式 (6) の単位時間当り期待費用を最小にするという意味での最適なオートセーブ時期に関する解析結果をまとめると、以下のようになる。

$$3.1 \quad c_2/c_1 \geq e^{-\lambda/\lambda} + 1$$

上の不等式が成立する場合、最適解は  $N^*=1$  となり、文書が CPU メモリ一杯になった時点で保存すればよいこととなる。なお、この場合には、CPU メモリを消費し尽くしたことにより、いずれにせよ、それまでの文書を 1 つのファイルとして一端保存しないと文書作成作業を継続できない。

$$3.2 \quad c_2/c_1 < e^{-\lambda/\lambda} + 1$$

この場合には、 $C(N)$  を最小にする  $N^*$  が唯一存在することが証明できる。その最適解を陽に示すことは困難であるが、これは 1 次元の探索問題であり、しかも  $N$  が自然数をとることから、数値的に容易に求めることができる。

### 4. 数値例

表 1 に  $(c_1, c_2) = (1000, 1)$  とし、ポアソン分布の平均  $\lambda$  が変化した場合の最適な  $N$  の値を示す。

表 1 最適オートセーブ時期

$\lambda$	$N^*$	$\lambda$	$N^*$	$\lambda$	$N^*$
0.001	1	0.01	2	0.1	7
0.002	1	0.02	3	0.2	10
0.003	1	0.03	4	0.3	12
0.004	2	0.04	4	0.4	14
0.005	2	0.05	5	0.5	16
0.006	2	0.06	5	0.6	17
0.007	2	0.07	6	0.7	19
0.008	2	0.08	6	0.8	20
0.009	2	0.09	7	0.9	21
				1.0	22

### 参考文献

- [1] S.M. Ross, "Applied Probability Models with Optimization Applications", Holden-Day, San Francisco, California (1970).