不完備情報下での Min-Max チェックポイント配置

尾崎達也 [†], 土肥正 (01307065)[†], 岡村寛之 (01013754)[†], 海生直人 (01105445)[‡]
[†] 広島大学大学院工学研究科情報工学専攻, [‡] 広島修道大学経済情報学部経営情報学科

1. はじめに

大規模データベースなどのファイル系システムにおいてシステム障害が発生した場合，その影響により非常に大きな経済的，社会的損失を被ることは周知の通りである．システム障害によるデータの損失の影響を最小限に抑えるために，予め定められたチェックポイントごとに主記憶装置上の情報を安定した二次記憶媒体に保存することが行われる．本稿では，システム障害発生時間分布が未知であるという不完備情報下において，Min-Max チェックポイント配置法を提案し，提案手法の有効性を定量的に検証することを目的とする．

2. 逐次的 CP 配置問題

ファイル系システムが時刻 $t=0$ で動作を開始し，時刻列 $\{t_1, t_2, \cdots, t_n, \cdots\}$ においてチェックポイント (CP) が配置されるものとする．簡単のため，各 CP では主記憶装置上のファイルの内容は安定な二次記憶媒体に瞬時に保存されるものと仮定する．システム障害発生時間は絶対連続で非減少の確率分布関数 $F(t)$ に従い，その平均を $1/\mu_1$ (>0) とする．CP を配置するためには固定コストが必要とされ，1 回当たりの CP 配置コストを c_0 (>0) とする．一方，システム障害が発生した場合，損失したデータの回復動作に要するコストはシステム障害発生時間に依存した関数 $L(\cdot)$ によって表され，2 階微分可能であるとする．

このとき，総期待稼動コストを最小にする CP 列 $\mathbf{t} = \{t_1, t_2, t_3 \cdots\}$ を求める問題は以下のように定式化される．

$$\min_{\mathbf{t}} : C(\mathbf{t}, F) = \sum_{n=0}^{N} \int_{t_n}^{t_n+1} g_n(t_n, t) dF(t). \tag{1}$$

ここで，$g_n(x, y) = c_0(n+1) + L(y-x)$，$N = \min\{n \geq 0 : T > t_{n+1}\}$ および $0 = t_0 < t_1 < \cdots < t_N < t_{N+1} = T$ である．文献 [1] では N $(T) \to \infty$ の無限計画期間において，総期待稼動コストを最小にする CP 列を求める問題を考察している．点検モデルとの類似性から，$F(t)$ が指数分布に従うならば最適 CP 列は等間隔，すなわち $t_2 - t_1 = t_3 - t_2 = \cdots = t_{n+1} - t_n = \cdots$ となる．一般に，確率分布関数 $F(t)$ が PF_2 (Polya Frequency Function of Order 2) であるならば，最適 CP 列 $\mathbf{t}$ は非増加列になる．

回復動作に要するコストが動作時間の線形関数である，すなわち，動作時間 x に対して $L(x) = a_0 x + b_0$ (a_0, b_0 (>0) は既知の定数）のような affine 形式が仮定されるならば，式 (1) に対する一階の最適性条件は

$$t_{n+1} - t_n = \frac{F(t_n) - F(t_{n-1})}{f(t_n)} - \frac{c_0}{a_0},\ n = 1, 2, 3, \cdots \tag{2}$$

となる．故に，総期待稼動コスト $C(\mathbf{t}, F)$ を最小にする CP 列 $\mathbf{t}^*$ を求めるためには，任意の初期値 t_1 を設定し，式 (2) を満たす CP 列 $\mathbf{t}$ を求める問題に帰着される．しかしながら，このアルゴリズムは初期値 t_1 の設定に試行錯誤を含み，極めて不安定であることが容易に類推され，必ずしも実利用上有効な CP 列生成アルゴリズムとは言えない．

上述の問題点を克服するために，Kaio and Osaki [2] は隣合う CP 列 $(t_{i-1}, t_i]$ $(i = 1, 2, \cdots)$ 間でシステム障害が発生する確率が一定と見なせるといった仮定の下で，近似的に最適 CP 列を求めるアルゴリズムを提案している．さらに，Fukumoto *et al.* [3], Ling *et al.* [4] は，単位時間当たりに配置する CP 頻度を連続関数によって近似し，変分原理に基づいて近似的に最適 CP 列を求めるアルゴリズムを開発している．

3. Min-Max CP 配置アルゴリズム

ここでは，システム障害発生時間分布 $F(t)$ のクラス（例えば, PF_2) は既知であるが，具体的な分布のパラメータは未知であるという状況を想定する．このとき，最も安全側に立脚した保全方策は，システム障害発生確率が最大となるような状況において CP 列を配置することであろう．以下では，不完備情報下で近似的に最適 CP 列を求めるための 3 種類の方法を紹介する．

3.1 アルゴリズム 0

$\mathcal{F}$ を全ての障害発生時間分布 F の集合とすれば，障害が最も頻繁に起こる状況における総期待稼動コストは $\max_{F \in \mathcal{F}} C(\mathbf{t}, F)$ によって表すことができる．

補題 3.1:

$$\max_{F \in \mathcal{F}} C(\mathbf{t}, F) = \max_{n=0,1,\cdots,N} g_n(t_n, t_{n+1}). \tag{3}$$

これより，問題は

$$\min_{\mathbf{t}} \max_{F \in \mathcal{F}} C(\mathbf{t}, F) \tag{4}$$

を満たす Min-Max CP 列 $\mathbf{t}$ を求めることである．

補題 3.2: $L(t_n) > nc_0$ $(n = 0, 1, 2, \cdots)$ かつ $\sum_{n=0}^{N} t_n = T$（すなわち，$\sum_{n=0}^{N} L^{-1}(c_0 n) < T$）を満たす最大の CP 配置数を N^* とすれば，最適 CP 列 $\mathbf{t}^*$ は次のような関係を満足する．

$$g_0(0, t_1) = g_1(t_1, t_2) = \cdots = g_{N^*}(t_{N^*}, T). \tag{5}$$

定理 3.3: $L(x) = a_0 x + b_0$ の仮定の下で，最適 CP 列は

$$t_n^* = n\left[\frac{T}{N^*+1} + \frac{c_0}{2a_0}(N^* - n + 1)\right] \quad (n = 0, 1, \cdots) \quad (6)$$

によって与えられる．ここで N^* は

$$N(N+1) < \frac{2a_0 T}{c_0} \quad (7)$$

を満たす最大の N である．

3.2 近似アルゴリズム 1

Kaio and Osaki [2] の方法に基づいて近似的に Min-Max CP 列を求めることを考える．以前にシステム障害が発生してないという条件下において，隣合う CP 列 $(t_{i-1}, t_i]$ $(i = 1, 2, \cdots)$ 間でシステム障害が発生する確率が一定であり，その確率は $p \in [0, 1]$ によって与えられるものとする．すなわち，

$$\frac{F(t_i) - F(t_{i-1})}{1 - F(t_{i-1})} = p \quad (8)$$

が成立すると仮定する．これより，CP 列上でのシステム障害発生時間分布は $F(t_i) = 1 - (1-p)^i$ によって近似され，$t_i = F^{-1}[1 - (1-p)^i]$ の関係が成り立つ．この仮定は，比較的微小な隣合う CP 列 $(t_{i-1}, t_i]$ $(i = 1, 2, \cdots)$ の存在を仮定すれば正当化され，そのときの近似総期待稼動コストは

$$\begin{aligned} C(\mathbf{t}, F) &\approx C(p) \\ &= \sum_{n=1}^{N} \int_{t_{n-1}}^{t_n} \Big[c_0(n+1) + L\Big(t - F^{-1}(1 \\ &\quad -(1-p)^{n-1})\Big)\Big] dF(t) \end{aligned} \quad (9)$$

によって与えられる．具体的な CP 列生成アルゴリズムとして，まず最初に総期待稼動コストを最小にする確率 p^* を求めた後に，式 (8) から逐次的に CP 列 $\mathbf{t}^*$ を更新する．

次に，$\min_{\mathbf{t}} \max_F C(\mathbf{t}, F)$ を満たす Min-Max CP 列 $\mathbf{t}$ を考える [1]．システム障害発生時間分布 $F(t)$ が IFR であると仮定すると，一般の IFR 分布に対する上限は

$$F(t) \le F_{up}(t) = \begin{cases} 1 - \mathrm{e}^{-t/\lambda_r^{1/r}} & (t < \mu_r^{1/r}) \\ 1 & (t \ge \mu_r^{1/r}) \end{cases} \quad (10)$$

となる．ここで，$\mu_r = \int_0^\infty t^r dF(t)$ は $F(t)$ の r 次モーメントであり，$\lambda_r = \mu_r / \Gamma(r+1)$ である（$\Gamma(\cdot)$ は標準ガンマ関数）．

3.3 近似アルゴリズム 2

Fukumoto *et al.* [3], Ling *et al.* [4] による近似アルゴリズムを適用した Min-Max CP アルゴリズムについて述べる [1]．単位時間当たりに配置する CP 頻度を CP 濃度と呼び，連続関数 $n(t)$ によって近似する．ここで，$1/n(t)$ は時刻 t における CP 間の時間間隔を表し，総期待稼動コストは

$$\begin{aligned} C(\mathbf{t}, F) &\approx C(n(t), F(t)) \\ &= c_0 \int_0^T n(t)\overline{F}(t) dt + \int_0^T L(1/2n(t)) dF(t) \end{aligned} \quad (11)$$

によって表現される．ここで，$\overline{F}(\cdot) = 1 - F(\cdot)$ である．上述の総期待稼動コストの近似表現において，CP 配置問題は総期待稼動コストを最小にする関数 $n(t)$ を求める変分問題に帰着される．

補題 3.4: 式 (11) の近似総期待稼動コストを最小にする CP 濃度は

$$n^*(t) = \sqrt{\frac{L'(1/n(t)) f(t)}{2c_0 \overline{F}(t)}} \quad (12)$$

によって与えられ，そのときの最適 CP 列は

$$n = \int_0^{t_n} n^*(t) dt, \; n = 1, 2, \cdots \quad (13)$$

を満たす $\mathbf{t}^* = \{t_1^*, t_2^*, \cdots\}$ によって与えられる．

さらに，$\min_{n(t)} \max_F C(n(t), F)$ を満たす CP 濃度に基づいて決定される CP 列を Min-Max CP 列として定義する．

補題 3.5: Min-Max 演算と Max-Min 演算の順序交換は可能である．すなわち，

$$\min_{n(t)} \max_F C(n(t), F) = \max_F \min_{n(t)} C(n(t), F). \quad (14)$$

定理 3.6: $L(t) = a_0 t + b_0$ の仮定の下で，Min-Max CP 濃度は

$$n^{**}(t) = \sqrt{\frac{a\lambda}{2c_0(1 - \lambda t)}}, \; 0 \le t \le \frac{1}{\lambda} \quad (15)$$

によって与えられ，そのときの近似総期待稼動コストは

$$\min_{n(t)} \max_{F(t)} C(n(t), F(t)) = \sqrt{2c_0 a_0 / \lambda} \quad (16)$$

となる．ここで $\lambda\ (> 0)$ は任意の積分定数である．

注意 3.7: システム障害発生時間分布の r 次モーメントの推定値 $1/\hat{\mu}_r$ $(r = 1, 2, \cdots)$ が与えられたとき，任意定数 λ は

$$\frac{1}{\mu_r} = \frac{\lambda}{2} \int_0^{1/\lambda} \frac{t^r}{\sqrt{1 - \lambda t}} dt \quad (17)$$

の方程式の解として求めることができる．

謝辞: 本研究は文部科学省科学研究費萌芽研究 (15651076)，および平成 15 年度広島修道大学総合研究所研究費の助成の下で行われたものである．

参考文献

[1] 尾崎, 土肥, 岡村, 海生, 不完備情報下での近似的チェックポイント配置, 第 8 回電子情報通信学会アシュアランスシステム研究会技術研究報告, 29–36 (2003).

[2] N. Kaio and S. Osaki (1985), A note on optimum checkpointing policies, *Microelectron. Reliab.*, **25**, 451–453.

[3] S. Fukumoto, N. Kaio and S. Osaki (1992b), Optimal checkpointing strategies using the checkpointing density, *J. Information Processing*, **15**, 87–92.

[4] Y. Ling, J. Mi and X. Lin (2001), A variational calculus approach to optimal checkpoint placement, *IEEE Trans. Computers*, **50** (7), 699-707 (2001).