

## 研究速報

ハイパキューブの耐故障経路選択算法の耐リンク故障への拡張

金子 敬一<sup>†</sup>(正員) 伊藤 秀男<sup>†</sup>(正員)

Extension of a Fault-Tolerant Routing Algorithm for Hypercube Systems to Tolerate Link Faults

Keiichi KANEKO<sup>†</sup> and Hideo ITO<sup>†</sup>, Members

<sup>†</sup> 千葉大学工学部情報画像工学科, 千葉市

Faculty of Engineering, Chiba University, Inage-ku, Chiba-shi, 263-8522 Japan

あらまし 従来, 故障ノードをもつハイパキューブに対する効率的な動的経路選択算法 FR が提案されていた. 本研究では, 故障リンクをもつ場合にも適用できるように, これを拡張した算法を提案する. また, シミュレーションにより, その有効性を確認する.

キーワード ハイパキューブ, 経路選択算法, 耐故障性, リンク故障, 安全ノード

### 1. まえがき

故障ノードをもつハイパキューブ [4] において, 各ノードが隣接ノードの状態を分類して保持することによる効率的な動的経路選択法 FR が提案されている [3]. 本研究では, 故障リンクをもつハイパキューブにも適用できるように, これを拡張した算法 FRx を提案する. また, 無作為標本抽出によるシミュレーション実験を行い, この算法の性能を評価し報告する.

FRx 算法は, FR 算法と同様に, 目的ノードまでのハミング距離の情報を利用している. 同様の手法を用いる算法としては, Chiu と Chen [1] によるものと Wu [5] によるものがある. しかしながら, これらの算法は, 可能な距離のすべてについて各ノードを分類しなくてはならないという欠点がある. 一方 FR 算法や FRx 算法は, 各ノードを分類する際にどのハミング距離までの情報を利用するかをパラメータで指定できるという特徴をもち, 今回実験を行った範囲では, 最短の距離 3 まで分類を行えば十分な結果を得られることがわかった.

以下では,  $d$  次元ハイパキューブを構成する  $2^d$  個のノードに 0 から  $2^d - 1$  までの相異なるラベルを付け, このラベルを用いてノードを指定することとする. ハイパキューブ中の故障ノードの集合を  $F$  で表す. 更に, ノード  $n$  の隣接ノード集合を  $N(n)$ , 隣接ノードのうち目的ノード  $t$  へ近づくことのできるものの集合を  $D(n, t)$  と表すこととする. また, 二つのノード  $n_1, n_2$  のハミング距離を  $H(n_1, n_2)$  によって表す.

## 2. FR 算法

FR 算法は, Chiu と Wu による経路選択算法 route [2] に目的ノードまでの距離情報による分類を付加したものである. このため, まず route 算法について説明し, 次に FR 算法について説明する.

### 2.1 route 算法とノードの分類

算法 route では, まずハイパキューブ内の非故障ノード集合  $\bar{F}$  を安全ノード集合  $S$  と危険ノード集合  $U$  に分類して, 更に  $U$  を常危険ノード集合  $\bar{U}$  と強危険ノード集合  $\tilde{U}$  に分類する.

[定義 1] (安全ノードと危険ノード) 非故障ノード  $n$  に対して  $|N(n) \cap F| \geq 2$  又は  $|N(n) \cap (F \cup U)| \geq 3$  ならば  $n \in U$  であり, さもなくば  $n \in S$  である.

[定義 2] (強危険ノードと常危険ノード) 危険ノード  $n$  に対して  $N(n) \cap S = \emptyset$  ならば  $n \in \tilde{U}$  であり, さもなくば  $n \in \bar{U}$  である.

またハイパキューブの性質として以下を定義する.

[定義 3] (全危険ネットワーク)  $S = \emptyset$  ならば, そのハイパキューブは全危険であるという.

このときハイパキューブの性質から, 出発ノード  $s$ , 目的ノード  $t$  に対して,  $s \in S$  又は  $t \in S$  ならば, 長さ  $H(s, t)$  なる最短経路でメッセージ通信可能となり;  $t \in U$  であつても  $s \in \bar{U}$  ならば, 長さがただか  $H(s, t) + 2$  の経路で通信可能であり; ハイパキューブが全危険でなければ, 長さがただか  $H(s, t) + 4$  の経路で通信可能であることがいえる [2].

$d$  次元ハイパキューブが全危険でないとは仮定すると以上のことからハイパキューブの各ノードは隣接ノードと情報を交換して, 隣接ノードを安全ノード, 常危険ノード, 強危険ノード, 故障ノードのいずれかに分

```

procedure route(c, t)
begin
  h := H(c, t); N := N(c); D := D(c, t);
  if h = 0 then
    deliver the message to c and exit
  else if  $\exists n \in D \cap S$  then next := n
  (* a comment line for future replacement *)
  else if  $\exists n \in D \cap \bar{U}$  then next := n
  else if  $\exists n \in D \cap \tilde{U}$  and  $(c \in \tilde{U}$  or  $h \leq 2)$  then
    next := n
  else if  $\exists n \in (N - D) \cap S$  then next := n
  else if  $\exists n \in (N - D) \cap \bar{U}$  then next := n
  else error('unable to deliver the message');
  route(next, t)
end

```

図1 Chiu と Wu による経路選択算法 route  
Fig. 1 Routing algorithm route by Chiu and Wu.

類しておくことで、効率的な経路選択を実現することができる。このことに基づく Chiu と Wu の route 算法を図 1 に示す。

## 2.2 FR 算法とノードの分類

FR 算法 [3] は、メッセージ配送時に経由すべきノードを隣接ノードの集合から選択する際に、目的ノードまでの距離の情報も利用することができる点に着目している。このため、あるノードが、そのノードから特定の距離にある非故障ノードすべてにハミング距離で到達可能であるか否かという性質を定義する。

[定義 4] (距離に関する全到達可能性) ある非故障ノード  $n$  からハミング距離  $h$  のすべての非故障ノードに対し、長さ  $h$  の経路で到達可能ならば、ノード  $n$  は距離  $h$  に関して全到達可能であるという。

ハイパキューブ内の距離  $h$  に関して全到達可能なノードの集合を  $R_h$  とする。しかしながら、 $R_h$  を求めたり、 $R_h$  に基づいて直接経路選択を行うことは難しいので、より扱いやすい  $R_h$  の部分集合  $S_h$  を定義し、距離  $h$  に関する安全ノード集合と呼ぶ。

[定義 5] (距離に関する安全ノード)  $d$  次元ハイパキューブの全非故障ノード  $n$  に対して、 $n \in S_1$ 。また、 $|N(n) \cap S_{h-1}| \geq d-h+1$  ならば  $n \in S_h$  とする。

このとき  $S_h$  と  $R_h$  に関して、次の定理が成立する。

[定理 1] 任意の距離  $h$  に対して  $S_h \subset R_h$ 。

一般にノード  $n$  にとって、 $N(n) \cap S_h$  は隣接ノードとの情報交換だけで決定できるため簡単に得られる。定理 1 より  $S_h \subset R_h$  なので、メッセージを経由しても安全なノードの集合として、 $S_h$  を経路選択に利用できる。前処理として、 $S_1, \dots, S_k$  (ただし  $3 \leq k < d$ ) を求めておけば、図 1 のコメント部分 (\*...\*) を `else if  $h \leq k+1$  and  $\exists n \in D \cap S_{h-1}$  then  $nxt := n$`  と変更して新たな経路選択法 FR を得る。

算法 FR では、次の定理 2 が示すように、算法 route よりも安全に経由できるノードが増えている。

[定理 2] 任意の距離  $h$  に対して  $S \subset S_h$ 。

## 3. FRx 算法

以下、本章では、故障リンク集合を  $L$ 、 $L$  と接続する非故障ノードを擬似故障ノードと呼び、その集合を  $P$  とする。また、 $N'(s) = \{n \mid H(s, n) = 1, (s, n) \notin L\}$ 、 $D'(s, t) = \{n \mid H(n, t) = H(s, t) - 1, n \in N'(s)\}$  とする。ただし  $N'(s)$  の定義の右辺に含まれる条件  $(s, n) \notin L$  において、 $(s, n)$  はノード  $s$  及び  $n$  を端点とするリンクを意味するものとする。すなわち  $N'(s)$  はノード  $s$  と非故障リンクで隣接するノード集合を意

味する。

### 3.1 route 算法の拡張

FR 算法をリンク故障に対応できるように拡張するために、まず route 算法の拡張を行う。

Chiu と Wu よりノードの分類を以下のように拡張する [2]。ここでは、まずハイパキューブ内の非故障ノードの集合  $\bar{F}$  を安全ノード集合  $S'$  と危険ノード集合  $U'$  に分類し、更に  $U'$  を常危険ノード集合  $\bar{U}'$  と強危険ノード集合  $\tilde{U}'$  に分類する。

擬似故障ノードを経由するとメッセージの送信に失敗する可能性がある。このためノードの分類時には、擬似故障ノードを故障ノードとみなして扱う。擬似故障ノード自身は、非故障ノードであることに注意。すなわち、擬似故障ノードは、以下、定義 6 によって危険ノードに分類され、定義 7 によって、更に強危険または常危険ノードへと分類されて経路選択に使用される。

[定義 6] (安全ノードと危険ノード) 非故障ノード  $n$  に対して、 $n \in P$  または  $|N(n) \cap (F \cup P)| \geq 2$ 、あるいは  $|N(n) \cap (F \cup P \cup U')| \geq 3$  ならば、 $n \in U'$  であり、さもなければ  $n \in S'$  である。

[定義 7] (強危険ノードと常危険ノード) 危険ノード  $n$  に対して  $N(n) \cap S' = \emptyset$  ならば  $n \in \tilde{U}'$  であり、さもなければ  $n \in \bar{U}'$  である。

このときハイパキューブの性質から、出発ノード  $s$ 、目的ノード  $t$  のいずれかが安全であれば長さ  $H(s, t)$  の経路で送信可能となる。また、 $S' \neq \emptyset$  ならば最悪でも  $H(s, t) + 4$  の経路で送信可能である。拡張された route 算法 routex を図 2 に示す。

```

procedure routex(c, t)
begin
  h := H(c, t); N := N'(c); D := D'(c, t);
  if h = 0 then
    deliver the message to c and exit
  else if D = {t} then nxt := t
  else if  $\exists n \in D \cap S'$  then nxt := n
  else if  $\exists n \in D \cap \bar{F} - P$  and  $h \leq 2$  then nxt := n
  (* a comment line for future replacement *)
  else if  $\exists n \in D \cap \bar{U}'$  then nxt := n
  else if  $\exists n \in D \cap \tilde{U}'$  and  $c \in \tilde{U}'$  then nxt := n
  else if  $\exists n \in (N - D) \cap S'$  then nxt := n
  else if  $\exists n \in (N - D) \cap \bar{U}'$  then nxt := n
  else error('unable to deliver the message');
  routex(nxt, t)
end

```

図 2 拡張経路選択算法 routex

Fig. 2 Extended routing algorithm routex.

3.2 FR 算法の拡張

リンク故障に対応するために距離に関する安全ノードの定義も以下のように拡張する。

[定義 8] (距離に関する安全ノード)  $d$  次元ハイパキューブにおいて、すべての非故障かつ非擬似故障ノードに対して、 $n \in S'_1$  . 更に  $|N'(n) \cap S'_{h-1}| \geq d-h+1$  ならば  $n \in S'_h$  とする。

再び  $S'_h, R_h, S'$  に関して次の 2 定理が成立する。

[定理 3] 任意の距離  $h$  に対して  $S'_h \subset R_h$  .

(証明)  $h$  に関する数学的帰納法による。  $h = 1$  のときは明らか。今、 $h < k$  について  $S'_h \subset R_h$  と仮定する。このとき  $n \in S'_k$  とすると、定義 8 より  $|N'(n) \cap S'_{k-1}| \geq d-k+1$  . すなわち、 $|N(n) - S'_{k-1}| \leq k-1$  . ここで、 $H(n, n') = k$  なる任意の非故障ノード  $n'$  に対して  $D(n, n')$  を考えると  $|D(n, n')| = k$  から  $D'(n, n') \cap S'_{k-1} \neq \emptyset$  . 仮定より  $S'_{k-1} \subset R_{k-1}$  . よって、そのノードを経由すれば長さ  $k$  の経路で  $n'$  へと到達できる。したがって  $n \in R_k$  . 以上から  $S'_k \subset R_k$  となり、任意の  $h$  に対して、 $S'_h \subset R_h$  となる。 □

[定理 4] 任意の距離  $h$  に対して  $S' \subset S'_h$  .

(証明)  $h$  に関する数学的帰納法による。  $S' \subset S'_1$  は明らか。  $n \in S'$  ならば  $|N(n) \cap (F \cup P)| \leq 1$  なので定義 8 より  $n \in S'_2$  . よって  $S' \subset S'_2$  . 今、 $h < k$  について  $S' \subset S'_h$  と仮定する。このとき  $n \in S'$  とすると、 $|N(n) \cap (F \cup P \cup U')| \leq 2$  より  $|N(n) \cap S'_h| \geq d-2$  . 仮定より  $S' \subset S'_{k-1}$  . したがって、 $|N(n) \cap S'_{k-1}| \geq d-2$  . 今、 $k \geq 3$  より、 $d-2 \geq d-k+1$  . よって  $|N(n) \cap S'_{k-1}| \geq d-k+1$  から、定義 8 より  $n \in S'_k$  . したがって  $S' \subset S'_k$  . □

route から FR を得た場合と同様に、 $S'_h \subset R_h$  が成立するので  $S'_h$  を新たに安全なノード集合とすれば経路選択に利用でき、 $S' \subset S'_h$  なので routex よりも良い結果を得ることができる。前処理として  $S'_1, \dots, S'_k$  を求めておけば、図 2 の (\*...\*) を else if ( $h \leq k+1$  and  $\exists n \in N \cap S'_{h-1}$ ) then  $next := n$  と変更して算法 FRx を得る。

4. シミュレーション実験

4.1 実験方法

算法 FRx の能力を検証するため、以下の手順を繰り返して、ネットワークが全危険にならないような故障ノード及び故障リンクの配置と目的ノードの位置の組合せに対して、シミュレーション実験を行う。

(1)  $d$  次元ハイパキューブにおいて、その対称性

から出発ノードを常にノード 0 に固定。

(2) システム内に、出発ノードを除いて  $f$  個の故障ノードと  $l$  個の故障リンクを設定。

(3) 各ノードを故障、安全、常危険、強危険に分類。また、 $S'_h (1 \leq h \leq k)$  を求める。この際、ハイパキューブが全危険になるものは以下の対象から除外。

(4) 0 以外の非故障ノードを目的ノードとする。

(5) routex と FRx を呼出して、それぞれについて、最短経路でメッセージを送信できずに迂回してしまう場合を数える。

[定義 9] (改善率) routex 算法が不必要に迂回した場合のうち、FRx 算法が最短経路で送信できた割合を改善率とする。

各々の改善率の推定値が、99%の確率で、誤差 0.01 未満になるまで、シミュレーションを反復実行する。

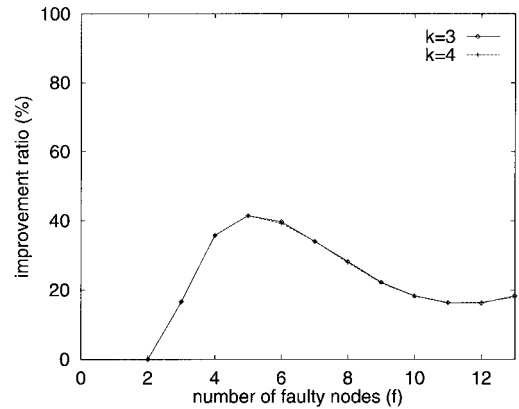


図 3 5次元ハイパキューブ(1リンク故障)の改善率  
Fig. 3 Improvement ratio ( $d = 5, l = 1$ ).

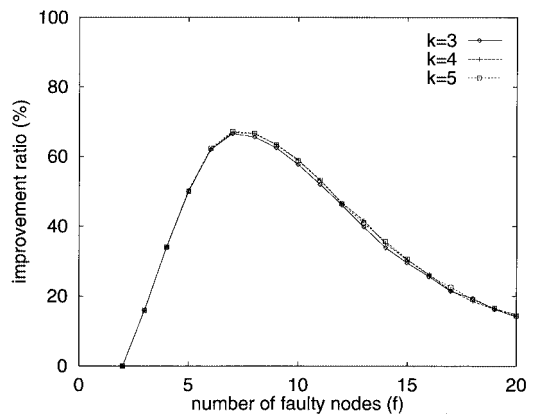


図 4 6次元ハイパキューブ(1リンク故障)の改善率  
Fig. 4 Improvement ratio ( $d = 6, l = 1$ ).

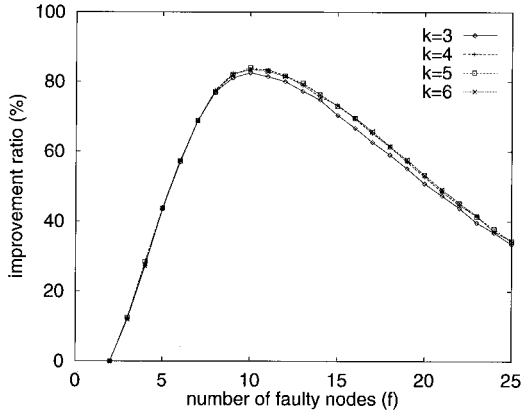


図5 7次元ハイパキューブ(1リンク故障)の改善率  
Fig.5 Improvement ratio ( $d = 7, l = 1$ ).

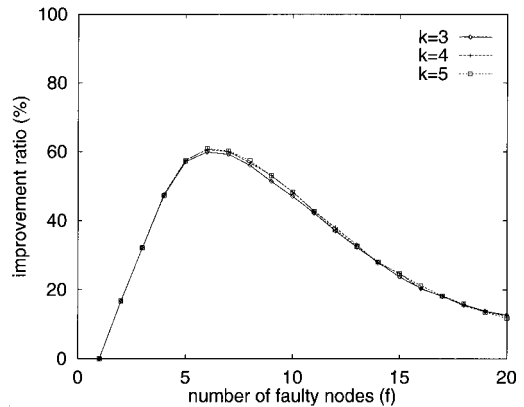


図7 6次元ハイパキューブ(2リンク故障)の改善率  
Fig.7 Improvement ratio ( $d = 6, l = 2$ ).

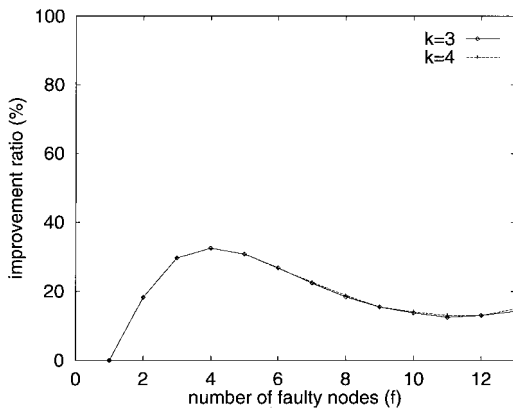


図6 5次元ハイパキューブ(2リンク故障)の改善率  
Fig.6 Improvement ratio ( $d = 5, l = 2$ ).

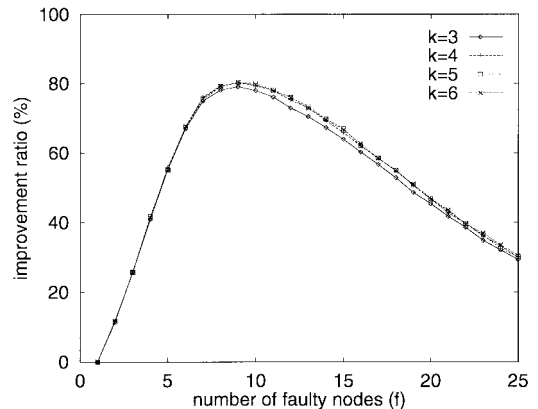


図8 7次元ハイパキューブ(2リンク故障)の改善率  
Fig.8 Improvement ratio ( $d = 7, l = 2$ ).

#### 4.2 実験結果

5, 6, 7次元ハイパキューブにおいて, 1又は2個のリンク故障を設定して, 実験を行った結果を図3から図8に示す.

改善率は最高で30%から80%に達しており, 算法FRxの有効性を示している. また, どのハミング距離まで距離に関する安全ノードを求めるといったパラメータ  $k$  の値にはほとんど依存せず, 計算量が最小となる  $k = 3$  としても十分な結果を得ることがわかる.

#### 5. むすび

本研究では, 故障ノードをもつハイパキューブにおける耐故障経路選択算法FRを拡張して, 故障リンクをもつハイパキューブにおいても利用できるような算法FRxを提案した. 更に, シミュレーション実験を行

い, この算法の性能評価を行った結果, 実験の範囲では, 性能はパラメータ  $k$  の値にほとんど依存せず, 距離3までについてのみ分類を行えば十分な結果を得られることがわかった. 今後の課題としては, システム運用時に起きる故障に対して, ノード状態の効率的な再計算方法を開発する必要がある.

#### 文献

- [1] G.M. Chiu and K.S. Chen, "Use of routing capability for fault-tolerant routing in hypercube multicomputers," IEEE Trans. Computers, vol.46, no.8, pp.953-958, Aug. 1997.
- [2] G.M. Chiu and S.P. Wu, "A fault-tolerant routing strategy in hypercube multicomputers," IEEE Trans. Computers, vol.45, no.2, pp.143-155, Feb. 1996.
- [3] 金子敬一, 伊藤秀男, "全到達可能性によるハイパキューブの耐故障経路選択算法," 信学論(D-I), vol.J81-D-I,

- no.8, pp.1024-1030, Aug. 1998.
- [4] C.L. Seitz, "The cosmic cube," Commun. ACM, vol.28, no.7, pp.22-33, July 1985.
- [5] J. Wu, "Adaptive fault-tolerant routing in cube-based multicomputers using safety vectors," IEEE Trans. Parallel and Distributed Systems, vol.9, no.4, pp.321-334, April 1998.
- (平成 10 年 8 月 31 日受付, 10 月 12 日再受付)
-