

# 低コストで柔軟性を持つインターコネクトトポロジーの研究 An Interconnection Topology with Low-Cost and Flexibility

佐藤 智文\*

Tomofumi Sato

法政大学 情報科学部 コンピュータ科学科

Email: 13k0120@cis.hosei.ac.jp

**Abstract**—This paper presents a new interconnection network topology, called the Generalized-Star Crossed Cube (GSCC( $n, k, m$ )), which is a product graph of the  $m$ -dimensional Crossed Cube and the  $(n, k)$ -Star Graph. GSCC( $n, k, m$ ) has three parameters:  $n, k$ , and  $m$ . So, the network size of GSCC( $n, k, m$ ) is more flexibly than a single graph: Crossed Cube or  $(n, k)$ -Star graph. And the diameter of this graph is smaller than one of GSC( $n, k, m$ ). GSC( $n, k, m$ ) is a product graph of the  $m$ -dimensional Hypercube and  $(n, k)$ -Star graph. The diameter of GSCC( $n, k, m$ ) is shorter than the one of GSC( $n, k, m$ ) because the diameter of hypercube is shorter than the one of Crossed Cube. This paper describes a shortest-path routing algorithm. The algorithm is varified by the execution time.

## 1. 序論

近年、コンピュータを並列につなぎ、巨大なデータを高速に処理することが必要である。特にスーパーコンピュータでは接続する方法で処理速度が大きく変化する。例として、コンピュータ同士を全接続すると高速で処理できるが、リンク数が多くなり費用がかかる。しかしながら、コンピュータをリング型に接続すると費用は抑えられるが、直径が大きくなり処理に時間がかかる。これら両方のバランスを取り、できるだけ費用を抑え、かつ高速に処理できるトポロジーを設計することが求められている。また、既存の研究で提案されたトポロジーの例として Hypercube [1] が挙げられる。これは、ノード数が  $2^m$  で次数と直径が  $m$  のグラフである。しかし、このグラフが設計できるノード数は 2 の累乗数のみで、それ以外は不可能である。この問題の解決策の 1 つとして、グラフと他のグラフとの積グラフを設計する方法がある。これは、あるグラフのノードの中に他のグラフを埋め込む方法である。これを使用することで、2 つのトポロジーの成分両方を所持し、ノード数はそれぞれのグラフが持つノード数の積で、次数と直径は 2 つのグラフの和で表すことができる。このようにノードがどんな数でもグラフを設計できる柔軟性のあるトポロジーを考える。

そのトポロジーの例として、Hypercube と  $(n, k)$ -Star Graph [2] の積グラフである GSC( $m, n, k$ ) (Generalized-Star Cube) [3] がある。これを使用することで、Hypercube  $\cdot$   $(n, k)$ -Star Graph の 1 つだけよりも柔軟性があるグラフが設計できた。しかし、Hypercube であるとノード数は  $2^m$  で直径が  $m$  とノード数が  $\frac{n!}{(n-k)!}$  であり、直径が  $n-1$  である  $(n, k)$ -Star Graph と比較すると大きい。そこで、今回 Hypercube の代わりに Crossed Cube [4] を使う。これは、Hypercube と比較すると次数は  $m$  と変

化しないが、直径が  $\lceil \frac{m+1}{2} \rceil$  となるグラフである。このグラフを  $(n, k)$ -Star Graph との積グラフにすることで、Hypercube が Crossed Cube に変化した分、直径を減らすことができる。本研究では、GSC( $n, k, m$ ) から直径を減らした GSCC( $n, k, m$ ) を提案する。

## 2. 先行研究

### 2.1. Hypercube

Hypercube (HQ( $m$ )) とは、ノード数が  $2^m$  であり、次数と直径が  $m$  のグラフである。アドレスは図 1 のように  $m$  ビットの 2 進数で表す。また、000 の隣接ノードは 001 や 010, 100 の 3 つであり、そのアドレスと 1 ビット異なるものに接続する。

### 2.2. Crossed Cube

Crossed Cube (CQ( $m$ )) とは、ノード数の  $2^m$  と次数の  $m$  は Hypercube と同じであるが、直径が  $\lceil \frac{m+1}{2} \rceil$  で Hypercube のおよそ半分からなるグラフである。これは図 2 のように表すことができる。図 2 との違いの例として、アドレス 110 の隣接ノードは図 1 の HQ(3) では 010, 100, 111 であるが、CQ(3) の隣接ノードは 001, 100, 111 である。また、アドレス 000 から 111 への経路探索として Hypercube の最短経路は 3 であるが、Crossed Cube の最短経路は 2 となり、1 回少ない探索回数で到達することができる。このようにリンクの接続する場所を変えることで直径を小さくすることができる。

また、CQ $_m$  の隣接ノードは、CQ $_{m-1}$  2 つが組み合わせたものであるから、CQ $_{m-1}^{(0)}(u_{m-2}, \dots, u_1, u_0)$  と CQ $_{m-1}^{(1)}(v_{m-2}, \dots, v_1, v_0)$  を定義する。次に、 $R = \{(00, 00), (10, 10), (01, 11), (11, 01)\}$  を定義する。これを満たすものを Crossed Cube のペア関係 (pair related) が成り立つという。これらから、隣接ノードは次の条件からなる。また、経路の詳細は 4 節で示す。

- 1)  $m$  が偶数のとき、 $u_{m-2} = v_{m-2}$
- 2)  $(u_{2j+1}u_{2j}, v_{2j+1}v_{2j}) \in R$  ( $0 \leq j < \lfloor \frac{m-1}{2} \rfloor$ )

### 2.3. $(n, k)$ -Star Graph

$(n, k)$ -Star Graph ( $(n, k)$ -Star) とは、ノード数が  $\frac{n!}{(n-k)!}$  で表すことができるグラフである。次数は  $k$  の値に依存せず常に  $n-1$  である。直径は  $n$  と  $k$  のパラメータで次の 2 つの状態に場合分けされる。(1)  $1 \leq k \leq \lfloor \frac{n}{2} \rfloor$  のときの  $2k-1$  と、( $\lfloor \frac{n}{2} \rfloor + 1 \leq k \leq n-1$ ) のときの  $k + \lfloor \frac{n+1}{2} \rfloor$  である。次数は  $k$  の値では変化しないため、 $n$  と  $k$  の値次第でコストが非常に小さくなる。このグラフは図 3 と図 4 のように表すことができる。また、アドレス番号は 1 から  $n$  の数値を  $k$  個並べたものからなる。

\* Supervisor: Prof. Yamin Li

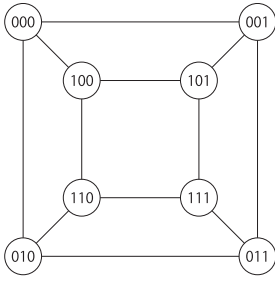


図 1. HQ(3)

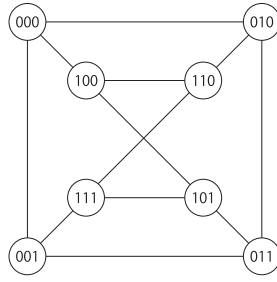


図 2. CQ(3)

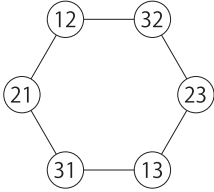


図 3. (3,2)-Star

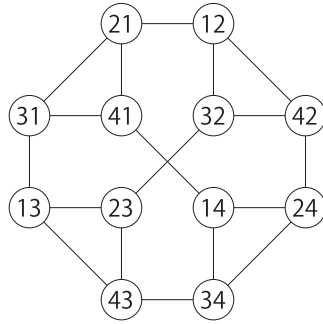


図 4. (4,2)-Star

## 2.4. Generalized-Star Cube

このグラフは Hypercube と  $(n, k)$ -Star Graph の積グラフである。このグラフは Hypercube の  $m$ ,  $(n, k)$ -Star Graph 部分の  $n$  と  $k$  の 3 つの変数を所持している。これら 3 つの変数を変えることで様々なサイズのグラフを設計することができる。次数はこれら 2 つの和  $m + n - 1$  である。直径は  $(n, k)$ -Star Graph と同様に  $k$  の値で 2 つの状態に場合分けされる。図 5 は、GSC(3,2,3) のグラフを示している。

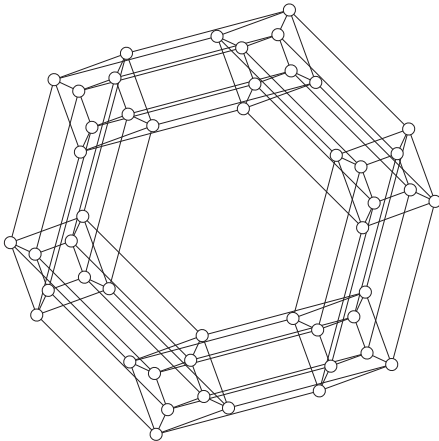


図 5. (3,2)-Star x HQ(3)

## 3. Generalized-Star Crossed Cube

このグラフは、GSC( $n, k, m$ ) の Hypercube 部分を Crossed Cube に変化することで設計することができる。これは次数は変化しないが、Hypercube 部分の直径  $m$  から Crossed Cube が持つ直径  $\lceil \frac{m+1}{2} \rceil$  と小さくすることができる。図 6 は GSCC(3,2,3) のグラフを示している、

図 5 との違いは 6 個の HQ(3) が CQ(3) に変化している点である。

定義 1. GSCC( $n, k, m$ ) のノード数は  $2^m \frac{n!}{(n-k)!}$  である。

証明 1. Crossed Cube のノード数は  $2^m$  である。また、 $(n, k)$ -Star Graph のノード数は  $\frac{n!}{(n-k)!}$  である。GSCC( $n, k, m$ ) はこれらの積グラフであるから、ノード数は 2 つのグラフの積  $2^m \frac{n!}{(n-k)!}$  となる。 ■

定義 2. GSCC( $n, k, m$ ) の直径は  $1 \leq k \leq \lfloor \frac{n}{2} \rfloor$  のとき  $\lceil \frac{m+1}{2} \rceil + 2k - 1$  である。

証明 2. Crossed Cube の直径は  $\lceil \frac{m+1}{2} \rceil$  である。また、 $(n, k)$ -Star Graph の直径は  $(1 \leq k \leq \lfloor \frac{n}{2} \rfloor)$  のとき  $2k - 1$  である。GSCC( $n, k, m$ ) はこれらの積グラフであるから、直径は  $(1 \leq k \leq \lfloor \frac{n}{2} \rfloor)$  のとき 2 つのグラフの和  $\lceil \frac{m+1}{2} \rceil + 2k - 1$  となる。 ■

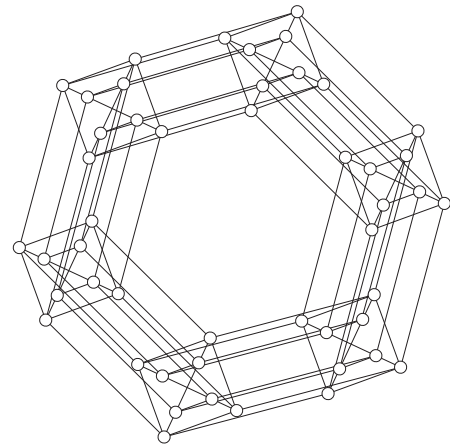


図 6. (3,2)-Star x CQ(3)

## 4. 最短経路問題

GSCC( $n, k, m$ ) の最短経路の探索は、次の方法からなる。

- 1) Crossed Cube 部分の対象ノード  $s$  を宛先ノード  $t$  へ移動する。
- 2)  $(n, k)$ -Star Graph 部分の対象ノード  $u$  を宛先ノード  $v$  へ移動する。

最初に、Crossed Cube 部分の最短経路の条件として  $l$  を  $s$  と  $t$  で異なる最大ビットとする。次に  $i^* = \lfloor \frac{l}{2} \rfloor$  と定義する。これは、Crossed Cube のビットの関係が 2 ビットずつ変化するためである。  $j \geq i^*$  のとき、 $\rho$  と  $s, t$  の関係は式 (1)、式 (2) のようになる。

$$\rho_{i^*}(s, t) = 0 \quad (j \geq i^* + 1) \quad (1)$$

$$\rho_j(s, t) = \begin{cases} 2 & (s_{i^*+1} s_{i^*} = \bar{t}_{i^*+1} \bar{t}_{i^*}) \\ 1 & (\text{Otherwise}) \end{cases} \quad (2)$$

次に、 $j < i^*$  のときを考える。Crossed Cube はビットが変化すると下位ビットも変化する。そのために、下の 3 つ条件のいずれかを満たす  $s$  と  $t$  を距離維持ペア関係 (distance-preserve pair related) が成り立つとして、その式を  $s_{2j+1} s_{2j} \stackrel{d.\sim p.}{\sim} t_{2j+1} t_{2j}$  のように表す。また、 $j < i^*$  のときの  $\rho$  と  $s, t$  の関係は式 (3) のようになる。

- 1)  $(s_{2j+1}s_{2j}, t_{2j+1}t_{2j}) \in \{(01, 01), (11, 11)\}$  かつ  $\sum_{i=j+1}^{\lfloor \frac{m-1}{2} \rfloor} \rho_i(s, t)$  が偶数
- 2)  $(s_{2j+1}s_{2j}, t_{2j+1}t_{2j}) \in \{(01, 11), (11, 01)\}$  かつ  $\sum_{i=j+1}^{\lfloor \frac{m-1}{2} \rfloor} \rho_i(s, t)$  が奇数
- 3)  $(s_{2j+1}s_{2j}, t_{2j+1}t_{2j}) \in \{(00, 00), (10, 10)\}$

また,  $(n, k)$ -Star Graph 部分の隣接ノードは, 先頭のアドレスだけが異なる, あるいは先頭と  $n$  番目にあるアドレスが交換されたものである. 先頭部分が異なる場合, それが宛先ノードにあるときはそれと交換する. 無い場合は, 宛先ノードに存在して対象ノードの先頭が同じであるが他の値が異なる場合, 先頭とその異なる場所の中から最も左にあるものと入れ替える. これらの最短経路アルゴリズムは次の Algorithm 1, Algorithm 2 に示す.

$$\rho_j(s, t) = \begin{cases} 0 & (s_{2j+1}s_{2j} \stackrel{d, \sim p}{\sim} t_{2j+1}t_{2j}) \\ 1 & (\text{Otherwise}) \end{cases} \quad (3)$$

このアルゴリズムの例として, GSCC(7,5,8) の大ききで Crossed Cube 部分のアドレス  $s = 00101110$ ,  $t = 00011001$ ,  $(n, k)$ -Star Graph 部分のアドレス  $u = 73215$ ,  $v = 12345$  とする. まず Crossed Cube 部分を探索する. まず,  $\rho_i(s, j)$  と  $T, Q$  を定義する. 式 (1),(2),(3) から  $\rho_j(s, t) = \{1, 1, 2, 0, 0\}$ , Algorithm 1 から  $Q = \{0, 1, 2\}$ ,  $T = \{0, 1\}$  が導かれる. まず Step1 へ移動する.  $T \neq \phi$  であるから, ONE\_STEP\_ROUTE(0,  $Q$ ) と ONE\_STEP\_ROUTE(1,  $Q$ ) の順で実行する. 1 回目の実行で  $m = 0$  が探索されて,  $u = 00101111$ ,  $Q = \{1, 2\}$  となる. 2 回目で  $m = 2$  が探索されて,  $u = 00101001$ ,  $Q = \{2\}$  となる.  $T = \phi$  であるため Step2 へ移動する.  $Q \neq \phi$  であるから, Step3 へ進む. Step3 では, 条件を満たす  $Q$  の最小値は 2 であるから,  $i = 2$  が選択され, ONE\_STEP\_ROUTE(2,  $Q$ ) が実行され,  $m = 5$  を探索する.  $\rho_2(s, t) = 1$ ,  $Q = 2$  であり, また Step3 の条件を満たすため再度 ONE\_STEP\_ROUTE(2,  $Q$ ) が実行されて,  $m = 4$  を探索する.  $Q = \phi$  となり  $S$  が出力されて探索が終了する. Crossed Cube 部分の探索回数は 4 回で終了する. その探索経路  $S$  は以下に示す.

00101110  $\rightarrow$  00101111  $\rightarrow$  00101001  $\rightarrow$  00001011  $\rightarrow$  00011001

次に,  $(n, k)$ -Star Graph 部分を探索する. そのために, まず対象ノードの先頭の値を見る.  $u_0 = 7$  であるが,  $u_0 \notin v$  であるから,  $u \not\subset v$  である最小値 4 を  $u_0$  へ代入する. 次に,  $u_0 = v_3$  であるから,  $u_0$  と  $u_3$  を入れ替えて  $u = 13245$  となる. 次は,  $u_0 \neq v_0$  であるから,  $u_j \neq v_j$  となる  $j$  の最小値は 1 であるため,  $u_0$  と  $u_1$  を入れ替えて  $u = 23145$  が導かれる.  $u_0 = v_1$  であるから,  $u_0$  と  $u_1$  を入れ替える. 次に  $u_0 = v_2$  であるから  $u_0$  と  $u_2$  を入れ替えて  $u = v$  となる. 対象ノードと宛先ノードが同じになり,  $(n, k)$ -Star Graph 部分の探索は 5 回で終了する.  $(n, k)$ -Star Graph 部分の経路は下に示す. また, この最短経路はこれら 2 つのグラフの和の 9 である.

73215  $\rightarrow$  43215  $\rightarrow$  13245  $\rightarrow$  23145  $\rightarrow$  32145  $\rightarrow$  12345

Crossed Cube は, 最大で  $\lceil \frac{m+1}{2} \rceil$  回で到達するため実行時間は  $O(m)$  となる. また,  $(n, k)$ -Star Graph 部分の実行時間は  $O(k)$  となる. このアルゴリズムの実行時間は平均で 2 つのグラフの和  $O(m+k)$  となる.

---

### Algorithm 1 Routing Algorithm

---

**Require:** CQ( $m$ ) と  $(n, k)$ -Star の対象ノード  $s, u$

**Require:** CQ( $m$ ) と  $(n, k)$ -Star の宛先ノード  $t, v$

**Require:** Crossed Cube 部分の経路  $S$ .

$\rho_i(s, t)$  を定義.

$Q \leftarrow \{j | \rho_j(s, t) \neq 0\}$ ,

$T \leftarrow \{\rho_j(u, v) \neq 0, j < i^* \text{ かつ}$

$\bar{s}_{2j+1}s_{2j} \stackrel{d, \sim p}{\sim} t_{2j+1}t_{2j} \text{ or } s_{2j+1}\bar{s}_{2j} \stackrel{d, \sim p}{\sim} t_{2j+1}t_{2j}\}$   
 /\*\*\*\*\* Step1 \*\*\*\*\*/

**if**  $T = \phi$  **then**

Step2 に進む.

**else**

$j \in T$  を探し, ONE\_STEP\_ROUTE( $i, Q$ ) を実行.

**end if**

/\*\*\*\*\* Step2 \*\*\*\*\*/

**if**  $Q = \phi$  **then**

$S$  を出力.

**else**

Step3 へ進む.

**end if**

/\*\*\*\*\* Step3 \*\*\*\*\*/

**if**  $i \in Q$  を満たす  $\bar{s}_{2i+1}s_{2i} \stackrel{d, \sim p}{\sim} t_{2i+1}t_{2i}$  または  $s_{2i+1}\bar{s}_{2i} \stackrel{d, \sim p}{\sim} t_{2i+1}t_{2i}$  が存在する **then**  
 最小の  $i$  を選ぶ.

**else**

$i \leftarrow \max\{j | j \in Q\}$

**end if**

ONE\_STEP\_ROUTE( $i, Q$ ) を実行して Step2 へ戻る.

**while**  $u \neq v$  **do**

$u$  の最初のアドレスをみる.

**if**  $u_0 \neq v_0$  **then**

$u_0 = v_j$  となる  $j$  ( $j \neq 0$ ) を探す.

$u_0$  と  $u_j$  を入れ替える.

**else**

$u_j \neq v_j$  となる最小の  $j$  を探す.

**if**  $j \neq \text{null}$  **then**

$u_0$  と  $u_j$  を入れ替える.

**else**

$u \not\subset v$  である  $v$  の最小値  $\min$  を探す.

$u_0 \leftarrow \min$

**end if**

**end if**

**end while**

---

### Algorithm 2 ONE\_STEP\_ROUTE( $j, Q$ )

---

**if**  $\rho_j(s, t) = 2$  **then**

$2j$  番目または  $2j + 1$  番目の探索.

$\rho_j(s, t) \leftarrow \rho_j(s, t) - 1$ ;

**else**

**if**  $\bar{s}_{2j+1}s_{2j} \stackrel{d, \sim p}{\sim} t_{2j+1}t_{2j}$  **then**  
 $(2j + 1)$  番目の探索.

**else if**  $s_{2j+1}\bar{s}_{2j} \stackrel{d, \sim p}{\sim} t_{2j+1}t_{2j}$  **then**  
 $2j$  番目の探索.

**end if**

$Q \leftarrow Q - \{j\}$

$\rho_j \leftarrow \rho_j(s, t) - 1$

**end if**

$S \leftarrow S + \{u'\}$

$u \leftarrow u'$

---

表 1. トポロジーの比較

パラメータ	HQ( $m$ )	CQ( $m$ )	$(n, k)$ -Star	GSC( $n, k, m$ )	GSCC( $n, k, m$ )
ノード数	$2^m$	$2^m$	$\frac{n!}{(n-k)!}$	$2^m \frac{n!}{(n-k)!}$	$2^m \frac{n!}{(n-k)!}$
次数	$m$	$m$	$n-1$	$m+n-1$	$m+n-1$
リンク	$m2^{m-1}$	$m2^{m-1}$	$\frac{n!}{(n-k)!} \frac{n-1}{2}$	$2^{m-1} \frac{n!}{(n-k)!} (m+n-1)$	$2^{m-1} \frac{n!}{(n-k)!} (m+n-1)$
直径	$m$	$\lceil \frac{m+1}{2} \rceil$	$2k-1$ (if $1 \leq k \leq \lfloor \frac{n}{2} \rfloor$ ) $k + \lfloor \frac{n-1}{2} \rfloor$ (if $\lfloor \frac{n}{2} \rfloor + 1 \leq k \leq n-1$ )	$m+2k-1$ (if $1 \leq k \leq \lfloor \frac{n}{2} \rfloor$ ) $m+k + \lfloor \frac{n-1}{2} \rfloor$ (if $\lfloor \frac{n}{2} \rfloor + 1 \leq k \leq n-1$ )	$\lceil \frac{m+1}{2} \rceil + 2k-1$ (if $1 \leq k \leq \lfloor \frac{n}{2} \rfloor$ ) $\lceil \frac{m+1}{2} \rceil + k + \lfloor \frac{n-1}{2} \rfloor$ (if $\lfloor \frac{n}{2} \rfloor + 1 \leq k \leq n-1$ )
コスト	$m^2$	$m \times \lceil \frac{m+1}{2} \rceil$	$(n-1)(2k-1)$ (if $1 \leq k \leq \lfloor \frac{n}{2} \rfloor$ ) $(n-1)(k + \lfloor \frac{n-1}{2} \rfloor)$ (if $\lfloor \frac{n}{2} \rfloor + 1 \leq k \leq n-1$ )	$(m+n-1)(m+2k-1)$ (if $1 \leq k \leq \lfloor \frac{n}{2} \rfloor$ ) $(m+n-1)(m+k + \lfloor \frac{n-1}{2} \rfloor)$ (if $\lfloor \frac{n}{2} \rfloor + 1 \leq k \leq n-1$ )	$(m+n-1)(\lceil \frac{m+1}{2} \rceil + 2k-1)$ (if $1 \leq k \leq \lfloor \frac{n}{2} \rfloor$ ) $(m+n-1)(\lceil \frac{m+1}{2} \rceil + k + \lfloor \frac{n-1}{2} \rfloor)$ (if $\lfloor \frac{n}{2} \rfloor + 1 \leq k \leq n-1$ )

## 5. トポロジーの比較

表 1 は、それぞれのトポロジー HQ( $m$ ), CQ( $m$ ),  $(n, k)$ -Star, GSC( $n, k, m$ ) と GSCC( $n, k, m$ ) のノード数, 次数, リンク数, 直径とコスト (コスト = 直径  $\times$  次数) を示している. GSC( $m, n, k$ ) と比較すると, GSCC( $m, n, k$ ) は  $(1 \leq k \leq \lfloor \frac{n}{2} \rfloor)$  のとき  $m+2k-1$  から  $\lceil \frac{m+1}{2} \rceil + 2k-1$  へと減少していることが確認できる. また, コストも  $(1 \leq k \leq \lfloor \frac{n}{2} \rfloor)$  のとき  $(m+n-1)(m+2k-1)$  から  $(m+n-1)(\lceil \frac{m+1}{2} \rceil + 2k-1)$  に減少している.

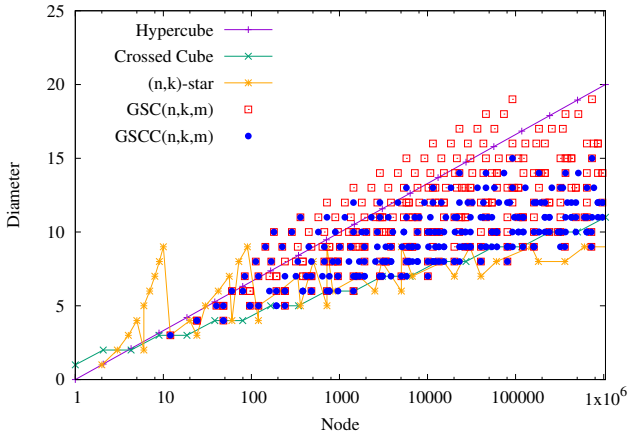


図 7. トポロジーのノード数と直径の比較

図 7, 図 8 はそれぞれのトポロジーの直径とコストをグラフに示している. グラフの点はそのトポロジーを設計することができるノード数を示している.  $(n, k)$ -Star Graph と比較すると GSCC( $n, k, m$ ) よりも直径が小さくなるが, 作ることができるノード数は多い. また, GSC( $n, k, m$ ) と比較すると柔軟性に変化はないが, 直径が減少してコストが抑えられている. 特にノード数が 737,280 個の直径は GSC( $n, k, m$ ) は最大で 19 であるが, GSCC( $n, k, m$ ) は 15 と 4 減少し, コストも 323 から 255 と 68 減少していることが確認できる.

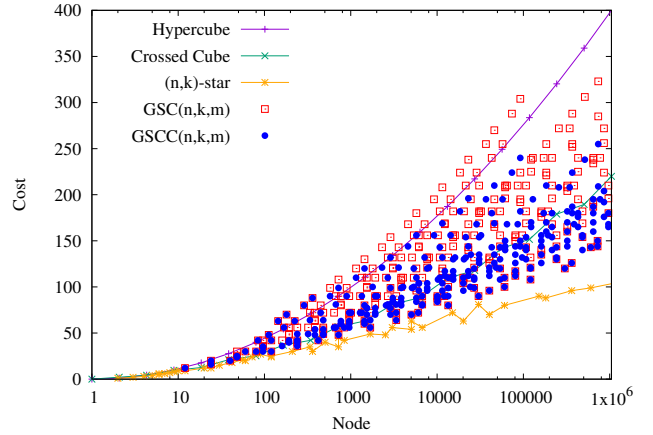


図 8. トポロジーのノード数とコストの比較

## 6. まとめ

本研究では, Generalized-Star Crossed Cube の利点, 最短経路アルゴリズムとその実行時間について述べた. このグラフは柔軟性があり, さらに直径が減少することで, 低コストで設計することができた. このグラフの更なる特徴を発見することが今後の課題となる.

## 参考文献

- [1] Y. Saad and M. H. Schultz, "Topological properties of hypercubes," *IEEE Transactions on Computers*, vol. 37, no. 7, pp. 867–872, Jul 1988.
- [2] W.-K. CHIANG and R.-J. CHEN, "Topological properties of the  $(n, k)$ -star graph," *International Journal of Foundations of Computer Science*, vol. 09, no. 02, pp. 235–248, 1998. [Online]. Available: <http://www.worldscientific.com/doi/abs/10.1142/S0129054198000167>
- [3] D. Arai and Y. Li, "Generalized-star cube: A new class of interconnection topology for massively parallel systems," in *2015 Third International Symposium on Computing and Networking (CANDAR)*, Dec 2015, pp. 68–74.
- [4] C.-P. Chang, T.-Y. Sung, and L.-H. Hsu, "Edge congestion and topological properties of crossed cubes," *IEEE Transactions on Parallel and Distributed Systems*, vol. 11, no. 1, pp. 64–80, Jan 2000.