Paired Many-to-Many Disjoint Path Covers in Recursive Circulants $G(2^m, 4)$

Sook-Yeon Kim and Jung-Heum Park, *Member, IEEE*

Abstract—A disjoint path cover (DPC for short) of a graph is a set of disjoint paths that cover all the vertices of the graph. A paired many-to-many k-DPC is a DPC composed of k paths between k sources and k sinks, such that each source is joined to a designated sink. We show that recursive circulant $G(2^m, 4)$ with at most f faulty vertices and/or edges being removed has a paired many-to-many k-DPC joining k arbitrary sources and sinks for any f and $k \geq 2$, subject to $f + 2k \leq m + 1$, where $m \geq 5$. The bound $m + 1$ on $f + 2k$ is the best possible.

Index Terms—Fault tolerance, disjoint path covers, interconnection networks, recursive circulants

1 INTRODUCTION

An interconnection network is frequently modeled as a graph in which the vertices and edges represent nodes and links, respectively. One of the key issues in interconnection networks is the detection of disjoint paths, which can be utilized as parallel routes for data communication among nodes. Disjoint paths can also be used to assign multiple jobs having a structure of linear arrays on a parallel computer. Since node and/or link failure is inevitable in a large network, fault tolerance is an important measure of network performance.

A *disjoint path cover* (DPC for short) of a graph is a set of disjoint paths that cover all the vertices of the graph. The disjoint path cover problem finds applications in many areas such as software testing, database design, and code optimization [1], [13], and is concerned with applications where full utilization of network nodes is important [18]. A k-DPC composed of k paths is classified into three types: one-to-one, one-to-many, and many-to-many. The one-to-one k-DPC joins a source to a sink, while the one-to-many joins a source to k sinks. The many-to-many k-DPC, in which k sources s_1, s_2, …, s_k are joined to k sinks t_1, t_2, …, t_k, is subclassified into paired and unpaired types. In the *paired* type, each source s_i should be joined to a specific sink t_i. In the *unpaired* type, a source s_i can be freely matched to a sink $t_{\sigma(i)}$ under an arbitrary permutation σ on $\{1, 2, \cdots, k\}$. The paired type, which has the strongest constraints among the DPC types, will be mainly discussed in this paper.

- *S.-Y. Kim is with the Department of Computer and Web Information Engineering, Hankyong National University, Korea. E-mail: sookyeon@hknu.ac.kr*
- *J.-H. Park is with the School of Computer Science and Information Engineering, The Catholic University of Korea, Korea. E-mail: j.h.park@catholic.ac.kr*

The ability of a graph to possess a DPC is often expressed by "disjoint path coverability." A graph G is called *f-fault paired* (resp. *unpaired*) *many-to-many k-disjoint path coverable* if $|V(G)| \geq f + 2k$ and G has a paired (resp. unpaired) k-DPC joining an arbitrary set S of k sources and a set T of k sinks in $G \setminus F$ for any fault set F with $|F| \leq f$, subject to $S \cap T = \emptyset$. Other types of f-fault k-disjoint path coverability are defined in a similar manner [8].

Given S and T in a graph G, it is NP-complete to determine if there exists a one-to-one, one-to-many, or many-to-many k-DPC joining S and T for any fixed $k \geq 1$ [18], [19]. DPC problems have been studied for graphs such as hypercubes [2], [3], [5], recursive circulants [8], [18], [19], and hypercube-like graphs [9], [15], [16], [19]. In addition, they are closely related to Hamiltonicity [18], [19], panconnectivity and pancyclicity [6], [7], [10], [20]. The necessary conditions for a graph G to be f-fault k-disjoint path coverable have been established in terms of the connectivity $\kappa(G)$ and the minimum degree $\delta(G)$, as follows.

Lemma 1: (a) If a graph G is f-fault one-to-one k-disjoint path coverable, then $f + k \leq \kappa(G)$ [8]. (b) If a graph G with $|V(G)| \geq f+k+2$ is f-fault one-to-many k-disjoint path coverable, then $f + k \leq \kappa(G) - 1$ [11]. (c) If a graph G with $|V(G)| \geq f + 2k + 1$ is f-fault unpaired k-disjoint path coverable, then $f+k \leq \delta(G)-1$ [19]. (d) If a graph G is f-fault paired k-disjoint path coverable, then $f + 2k \leq \kappa(G) + 1$ [18].

The *recursive circulant* $G(N, d)$, $d \geq 2$, proposed in [17], is a graph whose vertex set is $\{v_0, v_1, \cdots, v_{N-1}\}$ and edge set is $\{(v_i, v_j) : i+d^k \equiv j \pmod N\}$ for some k, $0 \leq k \leq \lceil \log_d N \rceil - 1\}$, which can also be defined as a circulant graph with N vertices and jumps of powers of d, $d^0, d^1, \cdots, d^{\lceil \log_d N \rceil - 1}$. There are extensive reports on recursive circulants in the literature because of the attractive properties like short diameter, node symmetry, maximum connectivity, and tree embedding [12], [17]. Thus, graph-theoretic properties of recursive circulants have been also studied, such as maximum induced subgraph [22], independent spanning tree [21], recognition [4], and chromatic number [14].

In general, recursive circulant $G(2^m, 4)$ can be constructed from four copies of $G(2^{m-2}, 4)$ not from two copies of $G(2^{m-1}, 4)$. This paper deals with a class of graphs, called *recursive-circulant-like graphs* (*RCL graphs* for short), in order to take advantage of simple recursive structure that a higher dimensional graph is made of two lower dimensional graphs in the class. RCL graphs include $G(2^m, 4)$ as a subclass.

Every m-dimensional RCL graph, whose degree and connectivity are both m, is known to be (a) f-fault *one-to-one* k-disjoint path coverable for any f and $k \geq 2$ with $f + k \leq m$ [8], (b) f-fault *one-to-many* k-disjoint path coverable for any f and $k \geq 2$ with $f + k \leq m - 1$ [8], (c) f-fault *unpaired* k-disjoint path coverable for any f and $k \geq 2$ with $f + k \leq m - 1$, where $m \geq 5$ [8], and (d) f-fault *paired* k-disjoint path coverable for any f and $k \geq 2$ with $f + 2k \leq m$, where $m \geq 4$ [19]. The bound m on $f + k$ for the one-to-one type and the bound $m - 1$ on $f + k$ for both the one-to-many and the unpaired types are optimal because of Lemma 1, (a) through (c).

The bound on $f + 2k$ for the paired type was first suggested to be $m - 1$ [18] and then improved to be m [19]. However, there is still a gap between the improved bound m [19] and the necessary condition $m+1$ in Lemma 1(d). In this study, we further sharpen the bound on $f + 2k$ for the paired type to achieve the optimal bound $m + 1$. In precise words, we show that every m-dimensional RCL graph is f-fault paired k-disjoint path coverable for any f and $k \geq 2$ with $f + 2k \leq m + 1$, where $m \geq 5$.

The rest of this paper is organized as follows. In Section 2, we define RCL graphs and discuss their properties. In Section 3, we present our main contribution as a theorem and prove it. Finally, we present concluding remarks in Section 4.

2 RECURSIVE STRUCTURES

The recursive circulant $G(2^m, 4)$ has a recursive structure, as follows.

Property 1: Given $G(2^m, 4)$ with $m \geq 2$, let there be a vertex subset $V_i = \{v_j : j \equiv i \pmod 4\}$ for each $i = 0, 1, 2,$ and 3. Then, the subgraph G_i induced by V_i is isomorphic to $G(2^{m-2}, 4)$ [17].

Fig. 1 shows $G(32, 4)$ composed of four copies of $G(8, 4)$. A superclass of $G(2^m, 4)$ called *recursive-circulant-like graphs* or *RCL graphs* has been defined in [8] as follows. Here, K_2 is a complete graph of two vertices, and C_4 is a cycle of four vertices.

Definition 1: A graph that belongs to RCL_m is called an *m-dimensional RCL graph*, where

- $RCL_3 = \{G(8, 4)\}$;
- $RCL_4 = \{G(16, 4), G(8, 4) \times K_2\}$;

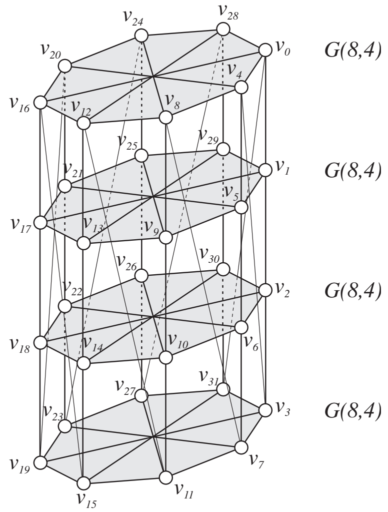


Fig. 1. Recursive structure of $G(32, 4)$

- $RCL_m = \{G(2^m, 4), G(2^{m-1}, 4) \times K_2, G(2^{m-2}, 4) \times C_4\}$ for $m \geq 5$.

Every m-dimensional RCL graph is made of 2^m vertices of degree m. In addition, the graphs are nonbipartite. Given two graphs H_0 and H_1 with a bijection ϕ from $V(H_0)$ to $V(H_1)$, we denote by $H_0 \oplus_\phi H_1$ a graph whose vertex set is $V(H_0) \cup V(H_1)$ and edge set is $E(H_0) \cup E(H_1) \cup \{(v, \phi(v)) : v \in V(H_0)\}$. Here, H_0 and H_1 are called the *components* of $H_0 \oplus_\phi H_1$. To simplify the notation, we often omit the bijection ϕ from $\oplus_\phi$ when it is clear in the context. The following shows the recursive structure of RCL graphs.

Property 2: Every m-dimensional RCL graph with $m \geq 6$ has four subcomponents G_0, G_1, G_2, and G_3 such that for any $i = 0, 1, 2,$ or 3, (a) G_i is isomorphic to an $(m-2)$-dimensional RCL graph, and (b) the subgraph induced by $V(G_i) \cup V(G_{(i+1) \bmod 4})$ is isomorphic to a graph product $G_i \times K_2$, which is indeed an $(m-1)$-dimensional RCL graph [8].

If we let H_0 and H_1 be the subgraphs induced by $V(G_i) \cup V(G_{(i+1) \bmod 4})$ and $V(G_{(i+2) \bmod 4}) \cup V(G_{(i+3) \bmod 4})$, respectively, for any $i = 0, 1, 2,$ or 3, then Property 2 leads to the following.

Property 3: Every m-dimensional RCL graph with $m \geq 6$ can be represented as $H_0 \oplus H_1$, where each component H_i is isomorphic to an $(m-1)$-dimensional RCL graph for $i = 0$ or 1 [8].

We conclude this section with the fault Hamiltonicity of RCL graphs that will be frequently referred to. A Hamiltonian path joining a pair of vertices can be viewed as any type of 1-DPC. A graph G is said to be *f-fault Hamiltonian-connected* (resp. *Hamiltonian*) if any pair of vertices is joined by a Hamiltonian path (resp. there exists a Hamiltonian cycle) in $G \backslash F$ for any fault set F with $|F| \leq f$. A graph G is f-fault Hamiltonian-connected if and only if it is f-fault 1-disjoint path coverable.

Property 4: Every m-dimensional RCL graph is

$(m-3)$-fault Hamiltonian-connected and is $(m-2)$-fault Hamiltonian [8].

3 Paired Many-to-Many DPCs

In this section, we present our main theorem and its proof.

Theorem 1: Every m-dimensional RCL graph, $m \geq 5$, is f-fault paired k-disjoint path coverable for any $f \geq 0$ and $k \geq 2$, subject to $f + 2k \leq m + 1$.

Theorem 1 is proved by induction on m. In the base case $m = 5$, the following lemma is obtained from a computer program that exhaustively searches for DPCs and finds them.

Lemma 2: Every 5-dimensional RCL graph is f-fault paired k-disjoint path coverable for any $f \geq 0$ and $k \geq 2$, subject to $f + 2k \leq 6$.

A k-DPC whose type is not specified is supposed to be a paired many-to-many k-DPC in this section. From now on, we recursively construct an f-fault k-DPC in an m-dimensional RCL graph for $m \geq 6$. The m-dimensional RCL graph is isomorphic to $H_0 \oplus H_1$ for some H_0 and H_1 in RCL_{m-1} by Property 3. Our induction hypothesis is that both H_0 and H_1 are f-fault k-disjoint path coverable for any $f \geq 0$ and $k \geq 2$, subject to $f + 2k \leq m$. In addition, a fact from Property 4 useful for our construction is that both H_0 and H_1 are $(m-4)$-fault 1-disjoint path coverable and $(m-3)$-fault Hamiltonian.

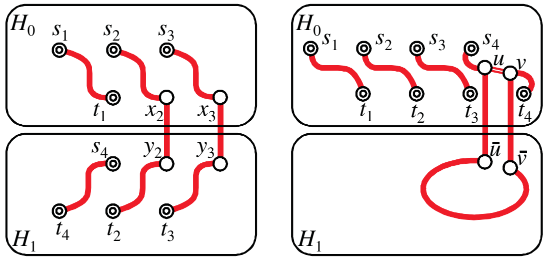
For a virtual edge fault set F', a k-DPC of $H_0 \oplus H_1 \setminus (F \cup F')$ is also a k-DPC of $H_0 \oplus H_1 \setminus F$. Thus, by treating arbitrary $m - 2k - |F| + 1$ fault-free edges as virtually faulty, we assume that

$$|F| = f \text{ and } f + 2k = m + 1.$$

Given a set $S = \{s_1, s_2, \ldots, s_k\}$ of k sources and a set $T = \{t_1, t_2, \ldots, t_k\}$ of k sinks in a graph G with a fault set F, a k-DPC joining S and T in $G \backslash F$ is denoted by k-DPC$[\{(s_1,t_1),(s_2,t_2),\ldots,(s_k,t_k)\}|G,F]$. Then, 1-DPC$[\{(v,w)\}|G,F]$ is a Hamiltonian path joining two vertices v and w in a graph $G \backslash F$. F_0 and F_1 denote the fault sets in H_0 and H_1, respectively. F_2 denotes the set of faulty edges between H_0 and H_1. Let $f_0 = |F_0|$, $f_1 = |F_1|$, and $f_2 = |F_2|$. Then, $F = F_0 \cup F_1 \cup F_2$ and $f = f_0 + f_1 + f_2$. We also denote the number of source-sink pairs in H_i by k_i where $i = 0, 1$, and the number of source-sink pairs between H_0 and H_1 by k_2. Then, $k = k_0 + k_1 + k_2$. We assume without loss of generality (wlog) that

$$k_0 \geq k_1, \text{ and if } k_0 = k_1, f_0 \geq f_1.$$

We let $I_0 = \{1, 2, \ldots, k_0\}$, $I_2 = \{k_0 + 1, \ldots, k_0 + k_2\}$, and $I_1 = \{k_0 + k_2 + 1, \ldots, k\}$. We assume that $\{s_j, t_j : j \in I_0\} \cup \{s_j : j \in I_2\} \subseteq V(H_0)$ and $\{s_j, t_j : j \in I_1\} \cup \{t_j : j \in I_2\} \subseteq V(H_1)$. For a vertex v in a component H_i, we denote by $\bar{v}$ the vertex adjacent to v in the other component H_{1-i}, for $i = 0, 1$. The sources and sinks are generally called



(a) $k_1 + k_2 \geq 1$ (b) $k_1 + k_2 = 0$

Fig. 2. Illustration of Procedure A

terminals. A vertex v is called *free* if it is neither a fault nor a terminal. An edge (v, w) is called *free* if both v and w are free with $(v, w) \notin F$. We will construct k-DPC$[\{(s_1,t_1),(s_2,t_2),\ldots,(s_k,t_k)\}|H_0 \oplus H_1, F]$ for any set F, S, and T such that $|F| = f$, $|S| = |T| = k \geq 2$, and $f + 2k = m + 1$. There are four cases, which will be dealt with in Subsections 3.1 through 3.4.

3.1 When $k_1 \geq 1$ or $f_0 \leq f - 1$

In this easy case, we employ a basic procedure presented in [19], except in three special cases. The exceptional cases will be considered later in Lemmas 4, 5, and 6.

Procedure A($H_0 \oplus H_1, S, T, F$) // See Fig. 2.

1: Select k_2 free edges (x_j, y_j) for $j \in I_2$, with $x_j \in V(H_0)$ and $y_j \in V(H_1)$.

2: Find $(k_0 + k_2)$-DPC$[\{(s_j,t_j) : j \in I_0\} \cup \{(s_j,x_j) : j \in I_2\}|H_0, F_0]$.

3: Case when $k_1 + k_2 \geq 1$

- a: Find (k_1+k_2)-DPC$[\{(s_j,t_j) : j \in I_1\} \cup \{(y_j,t_j) : j \in I_2\}|H_1, F_1]$.
- b: Merge the two DPCs with the k_2 free edges.

4: Case when $k_1 + k_2 = 0$

- a: Select an edge (u, v) on the $(k_0 + k_2)$-DPC of H_0 such that both $(u, \bar{u})$ and $(v, \bar{v})$ are free.
- b: Find 1-DPC$[\{(\bar{u}, \bar{v})\}|H_1, F_1]$.
- c: Merge the two DPCs with the edges $(u, \bar{u})$ and $(v, \bar{v})$. Discard the edge (u, v).

Lemma 3: Procedure A constructs an f-fault k-DPC when $k_1 \geq 1$ or $f_0 \leq f - 1$, except in the following three cases: (a) $k_0 = 1$, $k_1 = 1$, $k_2 = 0$, and $f_0 = m - 3$, (b) $k_0 = 1$, $k_1 = 0$, $k_2 = 1$, and $f_1 = m - 3$, and (c) $k_0 = 2$, $k_1 = 0$, $k_2 = 0$, and $f_1 = m - 3$.

Proof: For Step 1, we claim that at least k_2 free edges exist between H_0 and H_1. There exist 2^{m-1} edges between H_0 and H_1, where some of these edges may not be free because there are f faults and $2k$ terminals. Thus, the number of free edges between H_0 and H_1 should be at least $2^{m-1} - (f + 2k) = 2^{m-1} - (m + 1) > m + 1 > k_2$ for $m \geq 6$. The claim is thus proved. In Step 2, the $(k_0 + k_2)$-DPC of H_0 exists by the induction hypothesis when $k_0 + k_2 \geq 2$, because i) if $k_1 \geq 1$, then $f_0 + 2(k_0 + k_2) \leq f + 2(k - 1) \leq m$, and

ii) if $f_0 \le f-1$, then $f_0+2(k_0+k_2) \le (f-1)+2k = m$. The (k_0+k_2)-DPC of H_0 in Step 2 also exists when $k_0+k_2 = 1$, unless $f_0 = m-3$, because i) $f_0 \le f = (m+1)-2k \le (m+1)-4 \le m-3$ and ii) H_0 is $(m-4)$-fault 1-disjoint path coverable by Property 4. The exceptional case $f_0 = m-3$ with $k_0+k_2 = 1$ implies that (a) $k_0 = 1$, $k_1 = 1$, $k_2 = 0$, and $f_0 = m-3$.

In Step 3a, the (k_1+k_2)-DPC of H_1 exists for reasons similar to those stated in Step 2. When $k_1+k_2 \ge 2$, the (k_1+k_2)-DPC exists since $f_1+2(k_1+k_2) \le m$. When $k_1+k_2 = 1$, the (k_1+k_2)-DPC also exists, unless $f_1 = m-3$. The exceptional case $f_1 = m-3$ with $k_1+k_2 = 1$ implies that (b) $k_0 = 1$, $k_1 = 0$, $k_2 = 1$, and $f_1 = m-3$. For Step 4a, we claim that the edge (u, v) exists. There are at least $|V(H_0)|-f_0-k$ edges on the DPC of H_0 in Step 2, but some of them are not candidates of (u, v) because of the faults in F_1 and F_2. Since each fault prevents at most two edges from being candidates, the number of candidates is at least $|V(H_0)|-f_0-k-2(f_1+f_2) \ge 2^{m-1}-k-2f > 2^{m-1}-2(m+1) \ge 18$ for any $m \ge 6$. The claim is thus proved. In Step 4b, the 1-DPC of H_1 exists by Property 4, unless $f_1 = m-3$. The exceptional case $f_1 = m-3$ with $k_1+k_2 = 0$ implies that (c) $k_0 = 2$, $k_1 = 0$, $k_2 = 0$, and $f_1 = m-3$. □

The three exceptional cases of Lemma 3 are considered in the following Lemmas 4, 5, and 6.

Lemma 4: An f-fault k-DPC exists if $k_0 = 1$, $k_1 = 1$, $k_2 = 0$, and $f_0 = m-3$.

Proof: There is a Hamiltonian cycle C_h in $H_0 \setminus F_0$ from Property 4. Let C_h be (s_1, P_a, t_1, P_b) for some subpaths P_a and P_b. We assume wlog P_a is as long as P_b. Then, $(s_1, P_a, t_1) = (s_1, x, P'_a, y, t_1)$ for some distinct vertices x and y. We have three cases. In the first case $|\{\bar{x}, \bar{y}\} \cap \{s_2, t_2\}| = 0$, it suffices to merge C_h and 2-DPC$[\{(s_2, \bar{x}), (\bar{y}, t_2)\}|H_1, \emptyset]$ with the edges $(x, \bar{x})$ and $(y, \bar{y})$ and discard the edges (s_1, x) and (t_1, y) since H_1 is 0-fault 2-disjoint path coverable by the induction hypothesis. In the second case $|\{\bar{x}, \bar{y}\} \cap \{s_2, t_2\}| = 1$, we assume wlog that $\bar{x} = s_2$. Since H_1 is 1-fault 1-disjoint path coverable by Property 4, we merge C_h and 1-DPC$[\{(\bar{y}, t_2)\}|H_1, \{s_2\}]$ with the edges $(x, \bar{x})$ and $(y, \bar{y})$ and discard the edges (s_1, x) and (t_1, y). In the last case $|\{\bar{x}, \bar{y}\} \cap \{s_2, t_2\}| = 2$, let the subpath (t_1, P_b, s_1) be (t_1, P'_b, z, s_1). The subpath (t_1, P'_b, z) is a singleton (t_1) with $z = t_1$ if t_1 is adjacent to s_1 in C_h. Since H_1 is 2-fault 1-disjoint path coverable by Property 4, we merge C_h and 1-DPC$[\{(\bar{s_1}, \bar{z})\}|H_1, \{s_2, t_2\}]$ with the edges $(x, \bar{x})$, $(y, \bar{y})$, $(s_1, \bar{s_1})$, and $(z, \bar{z})$ and discard (s_1, x), (t_1, y), and (s_1, z). □

Lemma 5: An f-fault k-DPC exists if $k_0 = 1$, $k_1 = 0$, $k_2 = 1$, and $f_1 = m-3$.

Proof: There exists a Hamiltonian cycle C_h in $H_1 \setminus F_1$ by Property 4 such that $C_h = (t_2, x, P_a, y)$ for some distinct vertices x and y. We have three cases. In the first case $\{\bar{x}, \bar{y}\} \not\subseteq \{s_1, t_1, s_2\}$, we assume wlog that $\bar{x}$ is free. Since H_0 is 0-fault 2-disjoint path coverable by the induction hypothesis, it suffices to merge C_h and 2-DPC$[\{(s_1, t_1), (s_2, \bar{x})\}|H_0, \emptyset]$ with the edge $(\bar{x}, x)$ and discard the edge (t_2, x). In the second case $\{\bar{x}, \bar{y}\} = \{s_1, t_1\}$, we find an edge (z, w) in C_h such that $\bar{z}$ and $\bar{w}$ are free. Then, C_h can be represented as $(t_2, \bar{s_1}, P, z, w, P', \bar{t_1})$. Since H_0 is 1-fault 2-disjoint path coverable by the induction hypothesis, it suffices to merge C_h and 2-DPC$[\{(s_2, \bar{w}), (\bar{z}, t_1)\}|H_0, \{s_1\}]$ with the edges $(s_1, \bar{s_1})$, $(z, \bar{z})$, and $(w, \bar{w})$ and discard the edges $(t_2, \bar{s_1})$ and (z, w). In the third case, we assume wlog that $\bar{x} = s_2$. Since H_0 is 1-fault 1-disjoint path coverable by Property 4, it suffices to merge C_h and 1-DPC$[\{(s_1, t_1)\}|H_0, \{s_2\}]$ with the edge $(\bar{x}, x)$ and discard the edge (t_2, x). □

Lemma 6: An f-fault k-DPC exists if $k_0 = 2$, $k_1 = 0$, $k_2 = 0$, and $f_1 = m-3$.

Proof: There exists a Hamiltonian cycle C_h in $H_1 \setminus F_1$ by Property 4. Select an edge (x, y) in C_h such that both $\bar{x}$ and $\bar{y}$ are free. It suffices to merge C_h and 3-DPC$[\{(s_1, t_1), (s_2, \bar{x}), (\bar{y}, t_2)\}|H_0, \emptyset]$ with the edges $(x, \bar{x})$ and $(\bar{y}, y)$ and discard the edge (x, y). The 3-DPC of H_0 exists by the induction hypothesis. □

3.2 When $f_0 = f$ and $k_0 = k$

In this case, Procedure B holds good, except in a special case. The exceptional case will be separately considered in Lemma 8.

Procedure B($H_0 \oplus H_1, S, T, F$**)** // See Fig. 3.

1: Regarding s_1 and t_1 as virtually free vertices, find (k_0-1)-DPC$[\{(s_j, t_j) : j \in I_0 \setminus \{1\}\}|H_0, F_0]$.

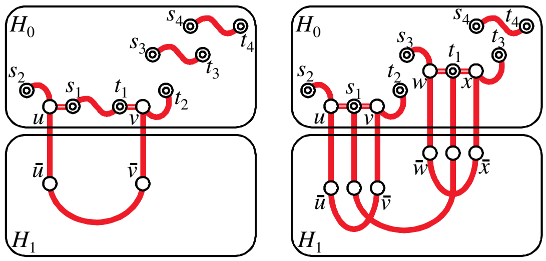
2: Case when both s_1 and t_1 are on the same path of the DPC
   - a: Let the path be $(P_a, u, s_1, P_1, t_1, v, P'_a)$ for some vertices u and v.
   - b: Find 1-DPC$[\{(\bar{u}, \bar{v})\}|H_1, \emptyset]$.
   - c: Merge the two DPCs with the edges $(u, \bar{u})$ and $(v, \bar{v})$. Discard the edges (u, s_1) and (t_1, v).

3: Case when s_1 and t_1 are on distinct paths of the DPC
   - a: Let the paths be (P_a, u, s_1, v, P'_a) and (P_b, w, t_1, x, P'_b) for some vertices u, v, w, and x.
   - b: Find 3-DPC$[\{(\bar{u}, \bar{v}), (\bar{w}, \bar{x}), (\bar{s_1}, \bar{t_1})\}|H_1, \emptyset]$.
   - c: Merge the two DPCs with the edges $(u, \bar{u})$, $(v, \bar{v})$, $(w, \bar{w})$, $(x, \bar{x})$, $(s_1, \bar{s_1})$, and $(t_1, \bar{t_1})$. Discard the edges (u, s_1), (s_1, v), (w, t_1), and (t_1, x).

Lemma 7: Procedure B constructs an f-fault k-DPC when $f_0 = f$ and $k_0 = k$, except when $k_0 = 2$.

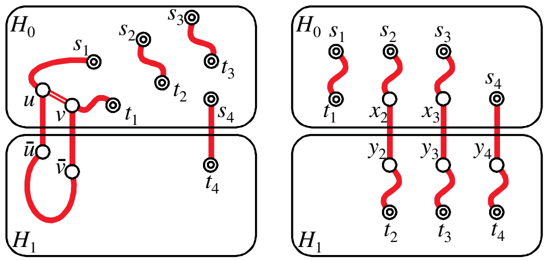
Proof: In Step 1, the (k_0-1)-DPC exists when $k_0-1 \ge 2$ by the induction hypothesis since $f_0+2(k_0-1) = f+2(k-1) \le m$. In other words, the DPC in Step 1 exists, unless $k_0 = 2$. In Step 2b, the 1-DPC of H_1 exists by Property 4. In Step 3b, the 3-DPC of H_1 exists by the induction hypothesis. □

Lemma 8: An f-fault k-DPC exists when $f_0 = f$ and $k_0 = k = 2$.



(a) Step 2 (b) Step 3

Fig. 3. Illustration of Procedure B



(a) Step 4 (b) Step 5

Fig. 4. Illustration of Procedure C

Proof: Note that $f_0 = f = m-3$ since $f+2k = m+1$ with $k = 2$. There exists a Hamiltonian cycle C_h in $H_0 \setminus F_0$ by Property 4. From C_h, we can extract four disjoint paths starting from the four terminals. For example, let C_h be $(s_1, P_x, x, s_2, P_y, y, t_1, P_z, z, t_2, P_w, w)$. Then, we can extract four paths (s_1, P_x, x), (s_2, P_y, y), (t_1, P_z, z), and (t_2, P_w, w). The subpath (s_1, P_x, x) is a singleton (s_1) with $s_1 = x$ if s_1 is adjacent to s_2 in C_h. Similarly, each of the subpaths (s_2, P_y, y), (t_1, P_z, z), and (t_2, P_w, w) may be a singleton. Even if the order of the terminals in C_h is different from that in the above mentioned example, we can always extract four disjoint paths. It suffices to merge the four disjoint paths of H_0 and a 2-DPC of H_1 to obtain the final k-DPC of $H_0 \oplus H_1$. For the aforementioned example, we must use 2-DPC$[\{(\bar{x}, \bar{z}), (\bar{y}, \bar{w})\}|H_1, \emptyset]$ with the edges $(x, \bar{x})$, $(y, \bar{y})$, $(z, \bar{z})$, and $(w, \bar{w})$. □

3.3 When $f_0 = f$, $k_1 = 0$, $k_0 \geq 1$, and $k_2 \geq 1$

When $k \geq 3$, we have two basic procedures. Procedure C is used if $k_2 = 1$ or some $\bar{s_a}$, $a \in I_2$, is not a terminal; otherwise, Procedure D is used. The case when $k = 2$ is studied in Lemma 11.

Procedure C$(H_0 \oplus H_1, S, T, F)$ // See Fig. 4.

1: If $k_2 = 1$, let $a = k$; otherwise, select $a \in I_2$ such that $\bar{s_a}$ is not a terminal.
2: Select $k_2 - 1$ free edges (x_j, y_j) for $j \in I_2 \setminus \{a\}$, with $x_j \in V(H_0)$ and $y_j \in V(H_1)$.
3: Regarding s_a as a *virtual* fault, find $(k_0 + k_2 - 1)$-DPC$[\{(s_j, t_j) : j \in I_0\} \cup \{(s_j, x_j) : j \in I_2 \setminus \{a\}\}|H_0, F_0 \cup \{s_a\}]$.
4: Case when $k_2 = 1$ and $\bar{s_a}$ is a terminal
   a: Select an edge (u, v) on the DPC.
   b: Regarding $\bar{s_a}$ as a *virtual* fault, find 1-DPC$[\{(\bar{u}, \bar{v})\}|H_1, \{\bar{s_a}\}]$.
   c: Merge the two DPCs with $(u, \bar{u})$, $(v, \bar{v})$, and $(s_a, \bar{s_a})$. Discard (u, v).
5: Case when $\bar{s_a}$ is not a terminal
   a: Let x_a and y_a be s_a and $\bar{s_a}$, respectively.
   b: Find k_2-DPC$[\{(y_j, t_j) : j \in I_2\}|H_1, \emptyset]$.
   c: Merge the two DPCs with (x_j, y_j), $j \in I_2$.

Lemma 9: Procedure C constructs an f-fault k-DPC when $f_0 = f$, $k_1 = 0$, $k_0 \geq 1$, $k_2 \geq 1$, $k \geq 3$, and $k_2 = 1$ or some $\bar{s_a}$, $a \in I_2$, is not a terminal.

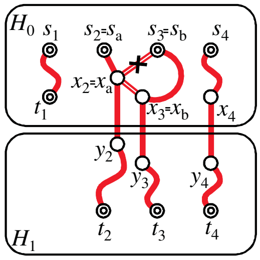


Fig. 5. Illustration of Procedure D

Proof: For Step 2, we can prove the existence of $k_2 - 1$ free edges as done for the proof of Lemma 3. In Step 3, the (k_0+k_2-1)-DPC exists since $k_0+k_2-1 \geq 2$ and $(f_0 + 1) + 2(k_0 + k_2 - 1) = (f + 1) + 2(k - 1) \leq m$. In Step 4a, note that $(u, \bar{u})$ and $(v, \bar{v})$ are free. In Step 4b, the 1-fault 1-DPC exists because of Property 4. In Step 5b, the k_2-DPC exists since $0 + 2k_2 \leq m$ if $k_2 \geq 2$; otherwise, the k_2-DPC exists by Property 4. □

A vertex u is called a *neighbor* of vertex v if u is adjacent to v. In addition, a vertex u is a *free neighbor* of v if u is free and is a neighbor of v with $(u, v) \notin F$.

Procedure D$(H_0 \oplus H_1, S, T, F)$ // See Fig. 5.

1: Select $a \in I_2$ such that the number of free neighbors of s_a is greater than or equal to the number of free neighbors of any other source s_j, $j \in I_2 \setminus \{a\}$.
2: Select $b \in I_2$ such that $a \neq b$.
3: Select a free neighbor x_a of s_a such that x_a is not a free neighbor of s_b.
4: Select $k_2 - 2$ free edges (x_j, y_j), $j \in I_2 \setminus \{a, b\}$, with $x_j \in V(H_0)$, $y_j \in V(H_1)$ such that $x_j \neq x_a$.
5: Regarding s_a as a virtual fault, find $(k_0 + k_2 - 1)$-DPC$[\{(s_j, t_j) : j \in I_0\} \cup \{(s_b, x_a)\} \cup \{(s_j, x_j) : j \in I_2 \setminus \{a, b\}\}|H_0, F_0 \cup \{s_a\}]$.
6: Let x_b be the vertex next to x_a on the DPC such that (s_b, P, x_b, x_a) is a path in the DPC.
7: Let $y_a = \bar{x_a}$ and $y_b = \bar{x_b}$.
8: Find k_2-DPC$[\{(y_j, t_j) : j \in I_2\}|H_1, \emptyset]$.
9: Merge the two DPCs with (s_a, x_a) and (x_j, y_j), $j \in I_2$. Discard (x_a, x_b).

Lemma 10: Procedure D constructs an f-fault k-DPC when $f_0 = f$, $k_1 = 0$, $k_0 \geq 1$, $k_2 \geq 2$, and $\bar{s_j}$ is a terminal for every $j \in I_2$.

Proof: For Step 3, we claim that we can always select the vertex x_a in H_0. The number of neighbors of s_a in H_0 is $m-1$. Some of the neighbors may not be candidates of x_a because they are common neighbors of s_a and s_b. The number of such common neighbors is at most two because any two distinct vertices have at most two common neighbors in every RCL graph. In addition, some of the neighbors of s_a may not be candidates of x_a because of faults or terminals. The number of faults and terminals in H_0 is at most $f_0 + 2k_0 + k_2 = m+1-k_2 \leq m-1$. However, the number of faults and terminals adjacent to s_a in H_0 that are not common neighbors of s_a and s_b is at most $\lfloor (m-1)/2 \rfloor$. Thus, the candidate of x_a is at least $(m-1) - (2 + \lfloor (m-1)/2 \rfloor) \geq 1$ for $m \geq 6$. The claim is thus proved. For Step 4, we can prove the existence of $k_2 - 2$ free edges as done for the proof of Lemma 3. In Step 5, the (k_0+k_2-1)-DPC of H_0 exists since $k_0+k_2-1 \geq 2$ and $(f_0+1)+2(k_0+k_2-1) = f+2k-1 \leq m$. In Step 6, note that $x_b \neq s_b$ since x_a is not a free neighbor of s_b. In Step 8, the k_2-DPC of H_1 exists since $k_2 \geq 2$ and $2k_2 \leq 2(k-1) \leq m$. □

Lemma 11: An f-fault k-DPC exists when $f_0 = f$, $k_1 = 0$, and $k_0 = k_2 = 1$.

Proof: There exists a Hamiltonian cycle C_h in $H_0 \setminus F_0$ by Property 4. C_h can be expressed in one of the following three representations by traversing it in the reverse order if necessary. Let $\{u, v\} = \{s_1, t_1\}$. Recall that a path (u, P_x, x) is a singleton (u) if $u = x$.

Case 1: $C_h = (u, P_x, x, v, P_y, y, s_2, P_z, z)$ with $x, y \neq \bar{t}_2$. We find 1-DPC$[\{(\bar{x}, \bar{y})\}|H_1, \{t_2\}]$ if $\bar{z} = t_2$; otherwise, find 2-DPC$[\{(\bar{x}, \bar{y}), (\bar{z}, t_2)\}|H_1, \emptyset]$. Then, it suffices to merge C_h and the DPC with $(x, \bar{x})$, $(y, \bar{y})$, and $(z, \bar{z})$ and discard (x, v), (y, s_2), and (z, u).

Case 2: $C_h = (u, \bar{t}_2, v, P_y, y, s_2, P_z, z)$. Assume wlog $(P_y, y) = (w, P'_y, y)$, $w \neq v, y$. It suffices to merge C_h and 2-DPC$[\{(\bar{z}, \bar{y}), (\bar{w}, t_2)\}|H_1, \emptyset]$ with $(w, \bar{w})$, $(y, \bar{y})$, and $(z, \bar{z})$ and discard (w, v), (y, s_2), and (z, u).

Case 3: $C_h = (u, v, s_2, P_z, z)$ with $v = \bar{t}_2$. It suffices to merge C_h and 1-DPC$[\{(\bar{z}, t_2)\}|H_1, \emptyset]$ with $(z, \bar{z})$ and discard (z, u) and (v, s_2). □

3.4 When $f_0 = f$ and $k_2 = k$

In this hardest case, all sources are contained in H_0, and all sinks are in H_1. Procedure E holds good if $f > 0$, except in a special case. The exceptional case will be considered in Lemma 13. Procedure F or G is vaild if $f = 0$. All the three procedures in this case use unpaired DPCs. An *unpaired* many-to-many k-DPC joining a set S of sources and a set T of sinks in a graph G with a fault set F is denoted by *unpaired* k-DPC$[S, T|G, F]$. Recall that every m-dimensional RCL graph is f-fault *unpaired* k-disjoint path coverable for any f and $k \geq 2$ with $f + k \leq m-1$, where $m \geq 5$ [8].

Procedure E($H_0 \oplus H_1, S, T, F$**)** // See Fig. 6.

1: Select k free edges (x_j, y_j) for $j \in I_2$, with $x_j \in V(H_0)$ and $y_j \in V(H_1)$.

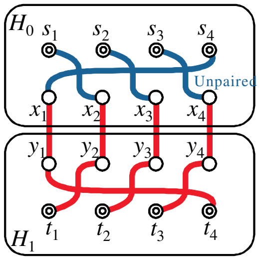


Fig. 6. Illustration of Procedure E

2: Find *unpaired* k-DPC$[\{s_j : j \in I_2\}, \{x_j : j \in I_2\}|H_0, F_0]$. Let $x_{\sigma(j)}$ be the vertex joined to s_j on the DPC for some permutation σ.
3: Find k-DPC$[\{(y_{\sigma(j)}, t_j) : j \in I_2\}|H_1, \emptyset]$.
4: Merge the two DPCs with (x_j, y_j), $j \in I_2$.

Lemma 12: Procedure E constructs an f-fault k-DPC when $f_0 = f > 0$ and $k_2 = k$, except when $k = 2$.

Proof: For Step 1, we can prove the existence of k free edges as the proof of Lemma 3. In Step 2, the unpaired k-DPC exists, except when $k = 2$, since $f_0 + k \leq (f + 2k) - k \leq (m+1) - 3 \leq m-2$. In Step 3, the k-DPC of H_1 exists since $0 + 2k \leq f - 1 + 2k \leq m$. □

Lemma 13: An f-fault k-DPC exists when $f_0 = f > 0$ and $k_2 = k = 2$.

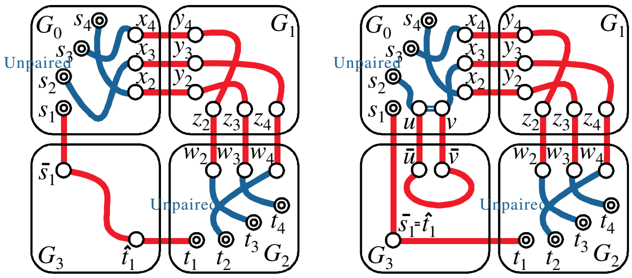
Proof: Since $f_0 = f = m + 1 - 2k = m - 3$, a Hamiltonian cycle C_h exists in $H_0 \setminus F_0$. C_h can be expressed in one of the following four representations by traversing it in the reverse order if necessary.

Case 1: $C_h = (s_1, s_2, P)$. Find an edge (x, y) such that $C_h = (s_1, s_2, P', x, y, P'')$ and $\{\bar{x}, \bar{y}\} \cap \{t_1, t_2\} = \emptyset$. It suffices to merge C_h and 2-DPC$[\{(\bar{x}, t_2), (\bar{y}, t_1)\}|H_1, \emptyset]$ with $(x, \bar{x})$ and $(y, \bar{y})$ and discard (s_1, s_2) and (x, y).

Case 2: $C_h = (s_1, P_x, x, s_2, P_y, y)$ with $\bar{x} \neq t_2$ and $\bar{y} \neq t_1$. When $\bar{x} \neq t_1$ or $\bar{y} \neq t_2$, assuming wlog that $\bar{y} \neq t_2$, we find 1-DPC$[(\bar{y}, t_2)\}|H_1, \{t_1\}]$ or 2-DPC$[\{(\bar{x}, t_1), (\bar{y}, t_2)\}|H_1, \emptyset]$, depending on whether $\bar{x} = t_1$ or not. Then, it suffices to merge C_h and the DPC with $(x, \bar{x})$ and $(y, \bar{y})$ and discard (s_1, y) and (s_2, x). When $\bar{x} = t_1$ and $\bar{y} = t_2$, we claim that there exists an edge (u, v) in C_h such that $(u, \bar{u})$ and $(v, \bar{v})$ are free. The number of edges in the cycle C_h is at least $2^{m-1} - f = 2^{m-1} - (m-3)$. Since the four terminals prevent at most six edges from being candidates of (u, v), the number of candidates is at least $2^{m-1} - (m-3) - 6 \geq 23$ for $m \geq 6$. The claim is thus proved. Then, it suffices to merge C_h and 1-DPC$[\{(\bar{u}, \bar{v})\}|H_1, \{t_1, t_2\}]$ with $(x, \bar{x})$, $(y, \bar{y})$, $(u, \bar{u})$, and $(v, \bar{v})$ and discard (s_1, y), (s_2, x), and (u, v).

Case 3: $C_h = (s_1, \bar{t}_1, P'_x, \bar{t}_2, s_2, P_y, y)$ with nonempty P'_x. Let $P'_x = (w, P_w)$. It suffices to merge C_h and 1-DPC$[\{(\bar{w}, \bar{y})\}|H_1, \{t_1, t_2\}]$ with $(t_1, \bar{t}_1)$, $(t_2, \bar{t}_2)$, $(w, \bar{w})$, and $(y, \bar{y})$ and discard $(\bar{t}_1, w)$, $(s_2, \bar{t}_2)$, and (s_1, y).

Case 4: $C_h = (s_1, \bar{t}_1, \bar{t}_2, s_2, P_y, y)$. We claim that a free vertex u exists in H_0 such that i) $u \neq \bar{t}_1, \bar{t}_2$, and ii) there exists a free edge $(u, \bar{t}_1)$ or $(u, \bar{t}_2)$. The number of ver-



(a) $\bar{s_1} \neq \hat{t_1}$ (b) $\bar{s_1} = \hat{t_1}$

Fig. 7. Illustration of Procedure F

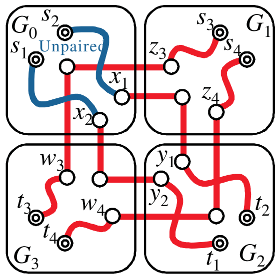


Fig. 8. Illustration of Procedure G

tices in H_0 adjacent to $\bar{t_1}$ or $\bar{t_2}$, excluding themselves, is $2(m-1)-2$. Note that f faults prevent at most f vertices from being candidates of u. Since terminals s_1 and s_2, too, are not candidates of u, the number of candidates is at least $(2(m-1)-2)-f-2 \geq 3$ for $m \geq 6$. The claim is thus proved. Suppose wlog that there exists a free edge $(u, \bar{t_1})$. Then, C_h can be represented as $(s_1, \bar{t_1}, \bar{t_2}, s_2, P', u, v, P'')$ for some vertex v. The vertex v may be identical to s_1. It suffices to merge C_h and 1-DPC$[\{(t_1, \bar{v})\}|H_1, \{t_2\}]$ with $(v, \bar{v})$, $(u, \bar{t_1})$, and $(t_2, \bar{t_2})$ and discard (u, v), $(s_1, \bar{t_1})$, and $(s_2, \bar{t_2})$. □

From now on, we consider the case with no faults, where all sources are contained in H_0, and all sinks are in H_1. From the assumption that $f + 2k = m + 1$, we have $k = (m+1)/2$, where m is an odd number not less than 7. Recall that every m-dimensional RCL graph, $m \geq 6$, has four subcomponents G_0, G_1, G_2, and G_3, where each G_i is isomorphic to an $(m-2)$-dimensional RCL graph by Property 2. In addition, component H_0 can be represented as $G_0 \oplus G_1$ and component H_1 can be $G_2 \oplus G_3$. Our induction hypothesis is added by the assumption that each subcomponent G_i is f-fault paired many-to-many k-disjoint path coverable for any $f \geq 0$ and $k \geq 2$, subject to $f + 2k \leq m - 1$.

In addition, we assume that there exists no source-sink pair whose source exists in G_0 and sink in G_3. Otherwise, our problem is reduced to the cases mentioned in Subsections 3.1, 3.2, and 3.3 since $G_3 \oplus G_0$ and $G_1 \oplus G_2$ can be respectively treated as H_0' and H_1'. For the same reason, we also assume that there exists no pair of a source in G_1 and a sink in G_2. For a vertex v, we denote by $\hat{v}$ the vertex adjacent to v in the other subcomponent of the same component with v. For a vertex v in G_0, vertex $\hat{v}$ is in G_1, and vice versa. For a vertex v in G_2, vertex $\hat{v}$ is in G_3, and vice versa. Let k' be the number of pairs of a source in G_0 and a sink in G_2, and k'' be the number of such pairs in G_1 and G_3. Then, $k' + k'' = k_2 = k$. We assume wlog that $k' \geq k''$. Procedure F is used if $k' = k$; otherwise, Procedure G is.

Procedure F($[G_0 \oplus G_1] \oplus [G_2 \oplus G_3], S, T, F$**)** // See Fig. 7.

1: Select mutually non-adjacent $2(k-1)$ free edges (x_j, y_j) and (z_j, w_j) for $j \in I_2 \setminus \{1\}$, with $x_j \in V(G_0)$, $y_j \in V(G_1)$, $z_j \in V(G_1)$, and $w_j \in V(G_2)$.
2: Regarding s_1 as a virtual fault, find *unpaired* $(k-1)$-DPC$[\{s_j : j \in I_2 \setminus \{1\}\}, \{x_j : j \in I_2 \setminus \{1\}\}|G_0, \{s_1\}]$. Let $x_{\sigma(j)}$ be the vertex joined to s_j on the DPC for some permutation σ.
3: Regarding t_1 as a virtual fault, find *unpaired* $(k-1)$-DPC$[\{w_j : j \in I_2 \setminus \{1\}\}, \{t_j : j \in I_2 \setminus \{1\}\}|G_2, \{t_1\}]$. Let $w_{\tau(j)}$ be the vertex joined to t_j on the DPC for some permutation τ.
4: Find $(k-1)$-DPC$[\{(y_{\sigma(j)}, z_{\tau(j)}) : j \in I_2 \setminus \{1\}\}|G_1, \emptyset]$.
5: Case when $\bar{s_1} \neq \hat{t_1}$
   a: Find 1-DPC$[\{(\bar{s_1}, \hat{t_1})\}|G_3, \emptyset]$.
   b: Merge the four DPCs with $(s_1, \bar{s_1})$, $(t_1, \hat{t_1})$, (x_j, y_j) and (z_j, w_j) for $j \in I_2 \setminus \{1\}$.
6: Case when $\bar{s_1} = \hat{t_1}$
   a: Select any edge (u, v) on the DPC of G_0.
   b: Regarding $\bar{s_1}$ as a virtual fault, find 1-DPC$[\{(\bar{u}, \bar{v})\}|G_3, \{\bar{s_1}\}]$.
   c: Merge the four DPCs with $(s_1, \bar{s_1})$, $(t_1, \hat{t_1})$, $(u, \bar{u})$, $(v, \bar{v})$, (x_j, y_j), and (z_j, w_j) for $j \in I_2 \setminus \{1\}$. Discard (u, v).

Lemma 14: Procedure F constructs an f-fault k-DPC when $f = 0$ and $k' = k_2 = k$.

Proof: Recall that $k = (m+1)/2$ with $m \geq 7$. For Step 1, we claim that the $2(k-1)$ free edges exist. First, we can select $k-1$ free edges between G_0 and G_1 since $2^{m-2} - k = 2^{m-2} - (m+1)/2 \geq (m-1)/2 = k-1$. Second, we can select $k-1$ free edges between G_1 and G_2 since $2^{m-2} - (k-1) - k = 2^{m-2} - m \geq (m-1)/2 = k-1$. The claim is thus proved. In Steps 2 and 3, the unpaired $(k-1)$-DPCs exist since $1 + (k-1) \leq m-3$. In Step 4, the $(k-1)$-DPC exists since $0 + 2(k-1) \leq m-1$. In Steps 5a and 6b, the 1-DPCs obviously exist. □

Let *bend* be a path of length two that passes through three distinct subcomponents. A bend (u, x, v) is said to be *free* if its two edges (u, x) and (x, v) are free. The *center* of a bend (u, x, v) is the vertex x. We partition I_2 into I_2' and I_2'' so that $s_j \in V(G_0), t_j \in V(G_2)$ for $j \in I_2'$ and that $s_j \in V(G_1), t_j \in V(G_3)$ for $j \in I_2''$. Recall that $|I_2'| = k' \geq |I_2''| = k''$. We further partition I_2' into J_1 and J_3 so that $|J_1| = \lceil k'/2 \rceil$ and $|J_3| = \lfloor k'/2 \rfloor$, and partition I_2'' into J_0 and J_2 so that $|J_0| = \lceil k''/2 \rceil$ and $|J_2| = \lfloor k''/2 \rfloor$.

Procedure G($[G_0 \oplus G_1] \oplus [G_2 \oplus G_3], S, T, F$**)** // See Fig. 8.

1: Select mutually disjoint k free bends such that
   a: $\lceil k'/2 \rceil$ bends $(x_j, \hat{x_j}, y_j)$, $j \in J_1$, have a center in G_1, where $x_j \in V(G_0)$ and $y_j \in V(G_2)$,
   b: $\lfloor k'/2 \rfloor$ bends $(x_j, \bar{x_j}, y_j)$, $j \in J_3$, have a center in G_3, where $x_j \in V(G_0)$ and $y_j \in V(G_2)$,
   c: $\lceil k''/2 \rceil$ bends $(z_j, \hat{z_j}, w_j)$, $j \in J_0$, have a center in G_0, where $z_j \in V(G_1)$ and $w_j \in V(G_3)$,
   d: $\lfloor k''/2 \rfloor$ bends $(z_j, \bar{z_j}, w_j)$, $j \in J_2$, have a center in G_2, where $z_j \in V(G_1)$ and $w_j \in V(G_3)$.
2: Find *unpaired* k'-DPC$[\{s_j : j \in I_2'\}, \{x_j : j \in I_2'\} | G_0, \{\hat{z_j} : j \in J_0\}]$. Let $x_{\sigma(j)}$ be the vertex joined to s_j on the DPC for some permutation σ.
3: Find k'-DPC$[\{(y_{\sigma(j)}, t_j) : j \in I_2'\} | G_2, \{\bar{z_j} : j \in J_2\}]$.
4: Find k''-DPC$[\{(s_j, z_j) : j \in I_2''\} | G_1, \{\hat{x_j} : j \in J_1\}]$.
5: Find k''-DPC$[\{(w_j, t_j) : j \in I_2''\} | G_3, \{\bar{x_j} : j \in J_3\}]$.
6: Merge the four DPCs with the k bends.

Lemma 15: Procedure G constructs an f-fault k-DPC when $f = 0$ and $k' < k_2 = k$.

Proof: Recall that $m \geq 7$, $k = (m+1)/2 \geq 4$, and $k' \geq k'' \geq 1$. For Step 1, we can select the bends step by step. For Step 1a, there exist $\lceil k'/2 \rceil$ free bends since i) the number of bends connecting G_0, G_1, and G_2 is 2^{m-2}, ii) the terminals in G_0, G_1, and G_2 can prevent at most k', k'', and k' bends from being candidates, respectively, and iii) $2^{m-2} - (2k' + k'') > 2^{m-2} - 2m > (m+1)/2 = k > \lceil k'/2 \rceil$. For Step 1b, note that the bends in Step 1a can prevent at most $2\lceil k'/2 \rceil (\leq k'+1)$ bends from being candidates. Thus, the number of candidates is at least $2^{m-2} - (2k' + k'') - (k' + 1) > 2^{m-2} - 2m > k > \lfloor k'/2 \rfloor$. Similarly, the numbers of candidates for Steps 1c and 1d are respectively at least $2^{m-2} - (k' + 2k'') - k'$ and $2^{m-2} - (k' + 2k'') - (k' + k'' + 1)$, both of which are greater than $2^{m-2} - 2m > k$. In Step 2, the unpaired k'-DPC of G_0 exists since $\lceil k''/2 \rceil + k' \leq k \leq 2k - 4 = m - 3$. In Step 3, the k'-DPC of G_2 exists since $k' \geq 2$ and $\lfloor k''/2 \rfloor + 2k' \leq 2k - 2 = m - 1$. In Step 4, the k''-DPC of G_1 exists when $k'' \geq 2$ since $\lceil k'/2 \rceil + 2k'' \leq 2k - 2 = m - 1$. Note that the k''-DPC exists even if $k'' = 1$ since $\lceil k'/2 \rceil = \lceil (k-1)/2 \rceil \leq 2k - 6 = m - 5$. In Step 5, the k''-DPC exists since $\lfloor k'/2 \rfloor + 2k'' \leq m - 1$ when $k'' \geq 2$ and since $\lfloor k'/2 \rfloor \leq m - 5$ when $k'' = 1$. □

4 Conclusion

We proved that every m-dimensional RCL graph, $m \geq 5$, is f-fault paired many-to-many k-disjoint path coverable for any $f \geq 0$ and $k \geq 2$, subject to $f + 2k \leq m + 1$. The bound $m + 1$ on $f + 2k$ is optimal. To the best of our knowledge, RCL graphs are the first class of graphs that allows for the optimal construction of paired DPCs except for the classes like complete graphs that obviously allow for it. Moreover, our result completes the fact that RCL graphs are optimally disjoint path coverable in all of the one-to-one, one-to-many, and paired/unpaired many-to-many types.

Acknowledgments

We are very grateful to the anonymous reviewers for their comments and suggestions. This research was partially supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (grant no. 2011-0004419).

References

[1] K. Asdre and S.D. Nikolopoulos, "The 1-fixed-endpoint path cover problem is polynomial on interval graphs," *Algorithmica* **58(3)**, pp. 679-710, 2010.
[2] X.-B. Chen, "Many-to-many disjoint paths in faulty hypercubes," *Inf. Sci.* **179(18)**, pp. 3110-3115, 2009.
[3] T. Dvořák and P. Gregor, "Partitions of faulty hypercubes into paths with prescribed endvertices," *SIAM J. Disc. Math.* **22(4)**, pp. 1448-1461, 2008.
[4] G. Fertin and A. Raspaud, "Recognizing recursive circulant graphs (extended abstract)," *Electr. Notes Disc. Math.* **5**, pp. 112-115, 2000.
[5] P. Gregor and T. Dvořák, "Path partitions of hypercubes," *Inf. Process. Lett.* **108(6)**, pp. 402-406, 2008.
[6] S.-Y. Hsieh and C.-W. Lee, "Conditional edge-fault Hamiltonicity of matching composition networks," *IEEE Trans. Parallel Distrib. Syst.* **20(4)**, pp. 581-592, Apr. 2009.
[7] S.-Y. Hsieh and C.-W. Lee, "Pancyclicity of restricted hypercube-like networks under the conditional fault model," *SIAM J. Disc. Math.* **23(4)**, pp. 2100-2119, Jan. 2010.
[8] S.-Y. Kim, J.-H. Lee, and J.-H. Park "Disjoint path covers in recursive circulants $G(2^m, 4)$ with faulty elements," *Theor. Comp. Sci.* **412(35)**, pp. 4636-4649, 2011.
[9] P.-L. Lai and H.-C. Hsu, "The two-equal-disjoint path cover problem of matching composition network," *Inf. Process. Lett.* **107(1)**, pp. 18-23, 2008.
[10] C.-W. Lee and S.-Y. Hsieh, "Pancyclicity of matching composition networks under the conditional fault model," *IEEE Trans. Computers* **61(2)**, pp. 278-283, Feb. 2012.
[11] J.-H. Lee and J.-H. Park, "General-demand disjoint path covers in a graph with faulty elements," *Intern. J. Comput. Math.* **89(5)**, pp. 606-617, Mar. 2012.
[12] H.-S. Lim, J.-H. Park, and K.-Y. Chwa, "Embedding trees into recursive circulants," *Disc. Appl. Math.* **69**, pp. 83-99, 1996.
[13] S.C. Ntafos and S.L. Hakimi, "On path cover problems in digraphs and applications to program testing, *IEEE Trans. Software Eng.* **5(5)**, pp. 520-529, Sept. 1979.
[14] N. Obradović, J. Peters, G. Ružić, "Minimum chromaticity of circulant graphs," *Disc. Math.* **299(1-3)**, pp. 288-296, 2005.
[15] J.-H. Park, "One-to-many disjoint path covers in a graph with faulty elements," in *Proc. of the International Computing and Combinatorics Conference COCOON 2004*, pp. 392-401, Aug. 2004.
[16] J.-H. Park, "Unpaired many-to-many disjoint path covers in hypercube-like interconnection networks," *Journal of KISS* **33(10)**, pp. 789-796, 2006 (in Korean).
[17] J.-H. Park and K.Y. Chwa, "Recursive circulants and their embeddings among hypercubes," *Theor. Comp. Sci.* **244**, pp. 35-62, 2000.
[18] J.-H. Park, H.-C. Kim, and H.-S. Lim, "Many-to-many disjoint path covers in hypercube-like interconnection networks with faulty elements," *IEEE Trans. Parallel Distrib. Syst.* **17(3)**, pp. 227-240, Mar. 2006.
[19] J.-H. Park, H.-C. Kim, and H.-S. Lim, "Many-to-many disjoint path covers in the presence of faulty elements," *IEEE Trans. Computers* **58(4)**, pp. 528-540, Apr. 2009.
[20] J.-H. Park, H.-S. Lim, and H.-C. Kim, "Panconnectivity and pancyclicity of hypercube-like interconnection networks with faulty elements," *Theor. Comp. Sci.* **377(1-3)**, pp. 170-180, 2007.
[21] J.-S. Yang, J.-M. Chang, S.-M. Tang, and Y.-L. Wang, "On the independent spanning trees of recursive circulant graphs $G(cd^m, d)$ with $d > 2$," *Theor. Comp. Sci.* **410(21-23)**, pp. 2001-2010, 2009.
[22] X. Yang, D.J. Evans, and G.M. Megson, "Maximum induced subgraph of a recursive circulant," *Inf. Process. Lett.* **95(1)**, pp. 293-298, 2005.