Paired Many-to-Many Disjoint Path Covers in Faulty Hypercubes ☆

Shinhaeng Jo[a], Jung-Heum Park[b], Kyung-Yong Chwa[a,*]

[a] *Department of Computer Science, KAIST, Daejeon 305-701, Korea*
[b] *School of Computer Science and Information Engineering, The Catholic University of Korea, Bucheon 420-743, Korea*

Abstract

A paired many-to-many *k-disjoint path cover* (*k-DPC* for short) of a graph is a set of k disjoint paths joining k distinct source-sink pairs that cover all the vertices of the graph. Extending the notion of DPC, we define a paired many-to-many *bipartite k-DPC* of a bipartite graph G to be a set of k disjoint paths joining k distinct source-sink pairs that altogether cover the same number of vertices as the maximum number of vertices covered when the source-sink pairs are given in the complete bipartite, spanning supergraph of G. We show that every m-dimensional hypercube, Q_m, under the condition that f or less faulty elements (vertices and/or edges) are removed, has a paired many-to-many bipartite k-DPC joining any k distinct source-sink pairs for any f and $k \geq 1$ subject to $f + 2k \leq m$. This implies that Q_m with $m-2$ or less faulty elements is strongly Hamiltonian-laceable.

Keywords: Disjoint path cover, hypercube, fault-tolerance, strongly Hamiltonian-laceability, graph theory.

1. Introduction

Finding node-disjoint paths is one of the most important issues in various interconnection networks, which is concerned with routing among nodes and embedding of linear arrays. Node-disjoint paths can be used as parallel paths to avoid congestion and provide fault-tolerance. Also, each of the node-disjoint paths can be utilized in its own pipeline computation. Interconnection networks are usually modeled as graphs, in which vertices and edges respectively correspond to nodes and links. In the rest of this paper, we use standard terminology in graphs (See [1]).

☆This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (grant number 2010-0010626).

*Corresponding author
Email addresses: josh@jupiter.kaist.ac.kr (Shinhaeng Jo), j.h.park@catholic.ac.kr (Jung-Heum Park), kychwa@jupiter.kaist.ac.kr (Kyung-Yong Chwa)

Let $G = (V, E)$ be a simple graph. Paths are disjoint if they share no vertices. Let $S = \{s_1, s_2, \ldots, s_k\}$ be the set of k *sources* and $T = \{t_1, t_2, \ldots, t_k\}$ be the set of k *sinks* such that $S, T \subset V$ and $S \cap T = \emptyset$. Many-to-many k-disjoint paths joining S and T are k disjoint paths $P_1, P_2, \ldots, P_k$ such that each P_i runs between s_i and $t_{\phi(i)}$, where ϕ is a bijection on $\{1, 2, \ldots, k\}$. They are called *paired* if $\phi(i) = i$ for every i. Otherwise, they are called *unpaired*. Generally, sources and sinks are called *terminals*.

A many-to-many *k-disjoint path cover* (k-DPC for short) joining S and T in G is a set of many-to-many k-disjoint paths joining S and T that cover all the vertices of G. A graph G is called *many-to-many k-disjoint path coverable* if $|V| \geq 2k$ and there exists a many-to-many k-DPC joining S and T for any pairwise disjoint S and T. For other kinds of DPC, readers are referred to [18, 19]. The k-DPC problem, originated from the community of interconnection networks, is concerned with applications where the full utilization of nodes is important [18]. Every paired many-to-many k-disjoint path coverable graph is Hamiltonian-connected for any $k \geq 1$ [18], i.e., every pair of vertices is joined by a Hamiltonian path. The existence of Hamiltonian paths and cycles is of crucial importance in parallel computing, since they are used in many distributed algorithms and they admit paths of various lengths.

However, no bipartite graph except for the complete graph on two vertices is Hamiltonian-connected. This stems from the nature of bipartite graphs that vertices of different balances appear alternatively in a path. For a bipartite graph $G = (V, E)$ with the bipartition $V = V^b \cup V^w$, where the vertices of V^b are referred to as black and the vertices of V^w as white, let $\beta(u)$, the *balance* of a vertex u, be -1 if u is black; 1 if u is white. We further define the balance of an edge as zero and the balance of a vertex pair (s, t) as $\beta((s, t)) = (\beta(s) + \beta(t))/2$. To describe Hamiltonian properties of bipartite graphs, the concept of strongly Hamiltonian-laceability was introduced, as found in [11, 17]. A bipartite graph with a fault set F is said to be *strongly Hamiltonian-laceable* if every fault-free vertex pair (s, t) is joined by a path of $G \setminus F$ that contains $|V \setminus F| - |\beta(V \setminus F) - \beta((s, t))|$ vertices. Here, $G \setminus F$ is the resultant graph by removing all the faulty elements of F from G, and $\beta(X) = \sum_{x \in X} \beta(x)$ for a set X of graph elements (vertices and edges) and vertex pairs.

As a consequence, no bipartite graph is paired many-to-many k-disjoint path coverable for any fixed $k \geq 1$ either, with the unique exception of the complete graph on two vertices for $k = 1$. A question regarding the upper bound on the number of vertices that can be covered by paired many-to-many k-disjoint paths in bipartite graphs then naturally arises. The tight upper bound can be established in terms of V, F, and K as follows, where K denotes the set of source-sink pairs, i.e., $K = \{(s_i, t_i) : 1 \leq i \leq k\}$. Many-to-many k-disjoint paths joining S and T altogether may cover at most $|V \setminus F| - |\beta(V \setminus F) - \beta(K)|$ vertices (as shown in Lemma 1 of the next section). Hereafter, let $\beta(V, F, K)$ or simply $\beta(G)$ denote $|\beta(V \setminus F) - \beta(K)|$. This motivates us to define a *many-to-many bipartite DPC* (*BiDPC* for short) as a set of many-to-many disjoint paths that pass through the same number of vertices as the upper bound.

Definition 1. *Given a set of k sources $S = \{s_1, s_2, \ldots, s_k\}$ and a set of k sinks $T = \{t_1, t_2, \ldots, t_k\}$ in $G \setminus F$ such that $S \cap T = \emptyset$, a* paired many-to-many bipartite k-DPC *joining S and T is a set of fault-free disjoint paths P_i joining s_i and t_i, for $1 \le i \le k$, that cover $|V \setminus F| - \beta(G)$ vertices of $G \setminus F$.*

Definition 2. *A bipartite graph G is* f-fault paired many-to-many k-bicoverable *if $|V| \ge f + 2k$ and there exists a paired many-to-many bipartite k-DPC joining S and T in $G \setminus F$ for any F, S, and T such that $|F| \le f$, $|S| = |T| = k \ge 1$, and $S \cap T = \emptyset$.*

A bipartite graph is paired many-to-many 1-bicoverable, by definition, if and only if it is strongly Hamiltonian-laceable. The unpaired many-to-many bipartite k-DPC and f-fault unpaired many-to-many k-bicoverable graph can be defined analogously. Notice that a many-to-many k-DPC becomes a many-to-many bipartite k-DPC, and the converse holds true if and only if $\beta(G) = 0$.

The Hypercube is one of the most popular interconnection networks possessing many attractive properties such as regularity, symmetry, small diameter, etc. The m-dimensional hypercube Q_m is a bipartite graph with 2^m vertices. It was shown by Gregor and Dvořák [7] that Q_m has a paired k-DPC if $2k - e < m$ and $\beta(G) = 0$, where e is the number of source-sink pairs that form edges of Q_m. Let f^v denote the number of faulty vertices and let f^e denote the number of faulty edges. In the presence of faulty vertices, Dvořák and Gregor [6] showed that $Q_m \setminus F$ has a paired k-DPC when $f^e = 0$, $3f^v + 2k \le m - 3$, and $\beta(G) = 0$. Chen [4] proved that, in the presence of faulty edges, $Q_m \setminus F$ has a paired k-DPC if $f^v = 0$, $f^e + 2k < m$, and $\beta((s_i, t_i)) = 0$ for every $1 \le i \le k$. Unpaired many-to-many disjoint paths were studied in [3].

The problem of embedding long paths and cycles in faulty hypercubes has attracted much attention in the literature. For path embedding, $Q_m \setminus F$ is strongly Hamiltonian-laceable if $f^v = 0$ and $f^e \le m - 2$ [20], and $Q_m \setminus F$ has a path joining a pair of vertices s and t that covers at least $2^m - 2f^v - |\beta((s,t))|$ vertices when $f^e = 0$ and $f^v \le m-2$ [10]. For cycle embedding, $Q_m \setminus F$ contains a cycle of length at least $2^m - 2f^v$ if $f^e \le m - 4$ and $f^v + f^e \le m - 1$ [24], or if $f^e = 0$ and $f^v \le 2m - 4$ [9]. These problems have been also studied in [8, 21, 23] under the so-called conditional fault model. For more discussion on the Hamiltonian paths/cycles in hypercubes, refer to, for example, [12, 13].

In this paper, we prove that Q_m is f-fault paired many-to-many k-bicoverable for any f and $k \ge 1$ subject to $f + 2k \le m$. This is a generalization of previous works on the paired DPC problem on faulty hypercubes [4, 6] in that hybrid faults are tolerated, the bound $f + 2k \le m$ is expanded, and the case when $\beta(G) \ne 0$ is also taken into account. Furthermore, our result for $k = 1$ is equivalent to Q_m being $(m-2)$-fault strongly Hamiltonian-laceable, which implies that $Q_m \setminus F$ contains a cycle of length $2^m - 2\max\{f^b, f^w\}$ when $f \le m-2$, where f^b and f^w respectively are the numbers of black and white faulty vertices. The strongly Hamiltonian-laceability is an improvement of the aforementioned results in [10, 20]. The length $2^m - 2\max\{f^b, f^w\}$ of a cycle is the longest in the true sense, and is also greater than the length $2^m - 2f^v = 2^m - 2(f^b + f^w)$ of [9, 24] when $f^b, f^w \ge 1$.

This paper is organized as follows: Section 2 gives preliminaries. In Section 3, we present some basic construction methods for bipartite disjoint path cover. These methods are applied to give a constructive proof of our main theorem in Sections 4 and 5. Section 6 presents the conclusion.

2. Preliminary

We begin with the upper bound, mentioned in the previous section, on the number of vertices that can be covered by paired many-to-many k-disjoint paths. Let $G = (V, E)$ be a bipartite graph with the bipartition $V^b \cup V^w$. An *s-t path* denotes a path joining two vertices s and t.

Lemma 1. *Let $\mathcal{P}$ be a set of paired many-to-many k disjoint paths joining $S = \{s_1, s_2, \ldots, s_k\}$ and $T = \{t_1, t_2, \ldots, t_k\}$ in $G \setminus F$ such that $S \cap T = \emptyset$. Then, (a) $\mathcal{P}$ covers at most $|V^b \setminus F| + \beta(K)$ white vertices and at most $|V^b \setminus F|$ black vertices when $\beta(V \setminus F) \geq \beta(K)$, (b) $\mathcal{P}$ covers at most $|V^w \setminus F|$ white vertices and at most $|V^w \setminus F| - \beta(K)$ black vertices when $\beta(V \setminus F) \leq \beta(K)$, and (c) $\mathcal{P}$ covers at most $|V \setminus F| - \beta(G)$ vertices in total, where $\beta(G) = |\beta(V \setminus F) - \beta(K)|$.*

Proof. If each s_i-t_i path in $\mathcal{P}$ covers n_i^w white vertices and n_i^b black vertices, then $n_i^w = n_i^b + \beta((s_i, t_i))$. Summing up the equalities over all $1 \leq i \leq k$, it follows $\sum_i n_i^w = \sum_i n_i^b + \beta(K)$. Plugging this into $\sum_i n_i^w \leq |V^w \setminus F|$ and $\sum_i n_i^b \leq |V^b \setminus F|$ results in the following:

$$\sum n_i^w \leq \min\{|V^w \setminus F|, |V^b \setminus F| + \beta(K)\}$$
$$\sum n_i^b \leq \min\{|V^b \setminus F|, |V^w \setminus F| - \beta(K)\}$$

Since $\beta(V \setminus F) = |V^w \setminus F| - |V^b \setminus F|$, the two inequalities above lead to (a) and (b). If $\beta(V \setminus F) \geq \beta(K)$, then, by (a) of this lemma,

$$\begin{aligned}\sum n_i^w + \sum n_i^b &\leq (|V^b \setminus F| + \beta(K)) + |V^b \setminus F| \\ &= (|V^b \setminus F| + \beta(K)) + (|V^w \setminus F| - \beta(V \setminus F)) \\ &= |V \setminus F| - (\beta(V \setminus F) - \beta(K)) = |V \setminus F| - \beta(G).\end{aligned}$$

If $\beta(V \setminus F) \leq \beta(K)$, then, by (b) of this lemma,

$$\begin{aligned}\sum n_i^w + \sum n_i^b &\leq |V^w \setminus F| + (|V^w \setminus F| - \beta(K)) \\ &= (|V^b \setminus F| + \beta(V \setminus F)) + (|V^w \setminus F| - \beta(K)) \\ &= |V \setminus F| + (\beta(V \setminus F) - \beta(K)) = |V \setminus F| - \beta(G).\end{aligned}$$

Thus, the statement (c) was also proved. □

Corollary 1. *If $\mathcal{P}$ of Lemma 1 is a paired many-to-many bipartite k-DPC, then (a) $\mathcal{P}$ covers exactly $|V^b \setminus F| + \beta(K)$ white vertices and $|V^b \setminus F|$ black vertices when $\beta(V \setminus F) > \beta(K)$, (b) $\mathcal{P}$ covers exactly $|V^w \setminus F|$ white vertices and $|V^w \setminus F| - \beta(K)$ black vertices when $\beta(V \setminus F) < \beta(K)$, and (c) $\mathcal{P}$ covers exactly $|V^w \setminus F|$ white vertices and $|V^b \setminus F|$ black vertices when $\beta(V \setminus F) = \beta(K)$.*

According to Corollary 1, the fault-free vertices that are not covered by $\mathcal{P}$ are all white when $\beta(V \setminus F) > \beta(K)$, and are all black when $\beta(V \setminus F) < \beta(K)$. Notice that $\beta(G)$ is the very number of vertices that are not covered by $\mathcal{P}$. Therefore, $\mathcal{P}$ becomes a paired many-to-many k-DPC if and only if $\beta(G) = 0$.

Each vertex of an m-dimensional hypercube Q_m is represented by a binary string in $\{0,1\}^m$, and two vertices u and v are joined by an edge $uv \in E$ if they differ in exactly one bit position. Hereafter, we use $G = (V, E)$ with the bipartition $V^b \cup V^w$ to denote Q_m. Let $G_p = (V_p, E_p)$ be the subgraph of G induced by V_p, where V_p is the set of binary strings (of length m) prefixed with p. We denote by $l(p)$ the length of a binary string p. Then, G_p is isomorphic to $Q_{m-l(p)}$. When p is equal to the empty string ϵ, we usually omit the reference to p for consistency. In addition, we let $V_p^b = V_p \cap V^b$ and $V_p^w = V_p \cap V^w$. If $l(p) < m$, then $V_p = V_{p0} \cup V_{p1}$ and $E_p = E_{p0} \cup E_{p1} \cup E_{p2}$, where E_{p2} is the set of edges between G_{p0} and G_{p1}, i.e., $E_{p2} = \{uv \in E_p : u \in V_{p0}, v \in V_{p1}\}$. E_{p2} is further decomposed into $E_{p2}^{b,w}$ and $E_{p2}^{w,b}$, where $E_{p2}^{b,w}$ is defined as $\{uv \in E_{p2} : u \in V_{p0}^b, v \in V_{p1}^w\}$ and $E_{p2}^{w,b}$ is defined similarly.

In this paper, a *unit* refers to a vertex, an edge, or an ordered pair of vertices. We represent an edge joining u and v as uv, and an ordered pair of u and v as (u, v). For a set of units X, we denote by X_p the set of units in X that are contained in G_p, i.e., $X_p = X \cap (V_p \cup E_p \cup (V_p \times V_p))$. If $l(p) < m$, then $X_p = X_{p0} \cup X_{p1} \cup X_{p2}$, where X_{p2} is defined as $X_p \setminus (X_{p0} \cup X_{p1})$. Also, the units can be classified according to their balances. For a set of units X, we use X^b, X^w, and X^o to denote $\{x \in X : \beta(x) = -1\}$, $\{x \in X : \beta(x) = 1\}$, and $\{x \in X : \beta(x) = 0\}$, respectively. That is, X^b is the subset of black vertices and pairs of black vertices, X^w is the subset of white vertices and pairs of white vertices, and X^o is the subset of edges and pairs of different colored vertices, Thus, $X = X^b \cup X^w \cup X^o$. Moreover, we let $X_p^c = X_p \cap X^c$ and $X_{p2}^c = X_{p2} \cap X^c$, where $c \in \{b, w, o\}$.

For the fault set F and the set of source-sink pairs K, we follow the same notation of X_p, X^c, and X_p^c described above, since F and K are also sets of units. For example, F_0 is the set of faulty elements in G_0, F^b is the set of black faulty vertices, F^o is the set of faulty edges, and $F_2 = F \cap E_2$. Thus, $F = F_0 \cup F_1 \cup F_2$ and $F = F^b \cup F^w \cup F^o$. Also, we have $K = K_0 \cup K_1 \cup K_2$ and $K_2 = K_2^w \cup K_2^b \cup K_2^o$. Here, K_2^o is the set of vertex pairs (s, t) such that $(s, t) \in (V_0 \times V_1) \cup (V_1 \times V_0)$ and $\beta((s, t)) = 0$. To distinguish which vertex of $(s, t) \in K_2^o$ is black (or white), we further define $K_2^{b,w} = K_2^o \cap ((V_0^b \times V_1^w) \cup (V_1^w \times V_0^b))$. $K_2^{w,b}$ is defined similarly, thus $K_2^o = K_2^{b,w} \cup K_2^{w,b}$. To represent the cardinalities of these sets, we use lower case letters such as $f = |F|$, $f^b = |F^b|$, $f_0^w = |F_0^w|$, $k = |K|$, $k_2 = |K_2|$, and $k_2^{b,w} = |K_2^{b,w}|$.

For a set of units X, let $\mathcal{U}(X) = (X \cap (V \cup E)) \cup \{x, y : (x, y) \in X\}$. We say two sets of units C and C' are *disjoint* if $\mathcal{U}(C)$ and $\mathcal{U}(C')$ are disjoint. A vertex v is *free* with respect to X if $v \notin \mathcal{U}(X)$, and an edge uv is *free* with respect to X if $u, v, uv \notin \mathcal{U}(X)$. A vertex or an edge is simply said to be *free* if it is free with respect to $S \cup T \cup F$.

A set of ordered pairs of vertices $C = \{(u_j, v_j) : 1 \le j \le l\}$ is called a u_1-v_l

chain if its members can be ordered to form a sequence $(u_{i_1}, v_{i_1}), (u_{i_2}, v_{i_2}), \ldots,$ (u_{i_l}, v_{i_l}) such that $v_{i_j} u_{i_{j+1}}$ forms an edge for every $1 \leq j < l$; such edges $v_{i_j} u_{i_{j+1}}$ are called *linking edges*. If there exists a u_j-v_j path for every j, then a u_1-v_l path can be constructed by merging the u_j-v_j paths with the $l-1$ linking edges. A set C of vertices and/or ordered pairs of vertices is also called a u_1-v_l *chain* if C' is a u_1-v_l chain, where $C' = \{(u, v) : (u, v) \in C\} \cup \{(w, w) : w \in C\}$. A u_1-v_l path can be obtained if there exists a u_j-v_j path for every j such that $u_j \neq v_j$.

A u_1-v_l chain is *closed* if $u_1 v_l$ forms an edge. Especially, we regard that $u_1 v_l$ is also a linking edge of a closed u_1-v_l chain. A u_1-v_l chain C is *free* if every vertex of $\mathcal{U}(C)$ (except u_1 and v_l) is free and all of its linking edges are also free. A free chain C is *simple* if its members share no vertex, i.e., $\{u_i, v_i\} \cap \{u_j, v_j\} = \emptyset$ for any $(u_i, v_i), (u_j, v_j) \in C$; $\{u_i, v_i\} \cap \{w\} = \emptyset$ for any $(u_i, v_i), w \in C$; and $w \neq w'$ for any $w, w' \in C$. We write $C \succ (u_1, v_l)$ if C is a free and simple u_1-v_l chain. If $C = \{(u_1, v_1), (u_2, v_2), \ldots, (u_l, v_l)\}$ is a simple u_1-v_l chain such that $v_j u_{j+1}$ is a linking edge for every $1 \leq j < l$, then

$$\beta(C) = \sum_{j=1}^{l} \beta((u_j, v_j)) = \frac{\beta(u_1)}{2} + \frac{\beta(v_l)}{2} + \sum_{j=1}^{l-1} \beta((v_j, u_{j+1})) = \beta((u_1, v_l)). \quad (1)$$

We list some works from the literature on paired DPC and strongly Hamiltonian-laceability of the hypercube Q_m in the following. They will be utilized for our construction of paired many-to-many bipartite DPCs in Q_m.

Lemma 2. *[2, 5, 15] Q_m without faulty elements has a paired 2-DPC joining S and T if $m \geq 3$ and $\beta((s_i, t_i)) = 0$ for each $i \in \{1, 2\}$, or if $m \geq 4$ and $\beta(K) = 0$.*

Lemma 3. *[7] Q_m without faulty elements has a paired k-DPC joining S and T if $\beta(K) = 0$ and $2k - e < m$, where e is the number of source-sink pairs that form edges of Q_m.*

Lemma 4. *Q_m with fault set F is strongly Hamiltonian-laceable if $f^v = 0$ and $f^e \leq m - 2$ [20], or if $f^v \leq 1$ and $f^v + f^e \leq m - 2$ [14].*

3. Construction of Paired Many-to-Many Bipartite DPC in Q_m

We present basic approaches for constructing a paired many-to-many bipartite DPC in an m-dimensional hypercube Q_m. We denote by (f'-fault k'-)BiDPC$[K'|G', F']$ a paired many-to-many bipartite k'-DPC of a bipartite graph G' with the fault set F' joining a set of source-sink pairs K', where $f' = |F'|$ and $k' = |K'|$. Recall that G is an equitable bipartite graph, i.e., $|V^b| = |V^w| = 2^{m-1}$. It follows $\beta(V \setminus F) = -\beta(F)$ and $\beta(G) = |-\beta(F) - \beta(K)|$. Without loss of generality, we assume $\beta(F) + \beta(K) \geq 0$; otherwise, it suffices to recolor all black vertices in white and vice versa. Thus,

$$\beta(G) = \beta(F) + \beta(K) \geq 0.$$

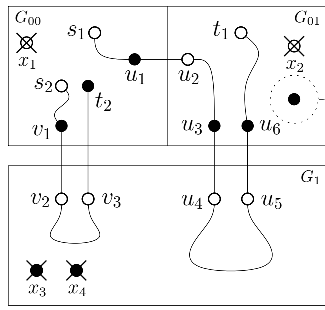


Figure 1: Divide-and-conquer approach.

Given a set of faulty elements F, a set of sources $S = \{s_1, s_2, \ldots, s_k\}$, and a set of sinks $T = \{t_1, t_2, \ldots, t_k\}$ in G, we construct f-fault k-BiDPC$[K|G, F]$. From Corollary 1, the BiDPC covers all of the white fault-free vertices and leaves $\beta(G)$ black fault-free vertices not covered. We state our main theorem as follows, which will be proved hereafter in this paper.

Theorem 1. *An m-dimensional hypercube Q_m, $m \geq 2$, is f-fault paired many-to-many k-bicoverable for any f and $k \geq 1$ subject to $f + 2k \leq m$.*

The proof will proceed with induction on m. The proof for $m = 2$ is trivial, and the proof for $m = 3$ is due to Lemma 4. Let $m \geq 4$. We assume, as the induction hypothesis, that $Q_{m'}$ with $2 \leq m' < m$ is f-fault k-bicoverable for any f and $k \geq 1$ with $f + 2k \leq m'$.

Divide-and-conquer will be a natural approach to construct a BiDPC of G. That is, we divide G into subcubes, find a BiDPC in every subcube, and merge the BiDPCs of subcubes into a BiDPC of G. For example, suppose that we are given $F = \{x_1, x_2, x_3, x_4\}$ and $K = \{(s_1, t_1), (s_2, t_2)\}$, as shown in Figure 1, where $\beta(G) = 1$ and G is divided into subcubes G_{00}, G_{01}, and G_1. If we select inter-subcube edges u_1u_2, u_3u_4, and u_5u_6 as linking edges, we then obtain an s_1-t_1 simple chain $C[1] = \{(s_1, u_1), (u_2, u_3), (u_4, u_5), (u_6, t_1)\}$. Similarly, selecting inter-subcube edges v_1v_2 and v_3t_2 results in an s_2-t_2 simple chain $C[2] = \{(s_2, v_1), (v_2, v_3), t_2\}$. The two chains are disjoint. Let $R = C[1] \cup C[2]$. Then, we have $R_{00} = \{(s_1, u_1), (s_2, v_1), t_2\}$, $R_{01} = \{(u_2, u_3), (u_6, t_1)\}$, and $R_1 = \{(u_4, u_5), (v_2, v_3)\}$. If there exist BiDPC$[R_{00} \setminus \{t_2\}|G_{00}, F_{00} \cup \{t_2\}]$, BiDPC$[R_{01}|G_{01}, F_{01}]$, and BiDPC$[R_1|G_1, F_1]$, then we can construct a BiDPC of G, namely $\mathcal{P}$, by merging the three BiDPCs of subcubes with the selected linking edges. Notice that only one fault-free vertex, a member of V_{01}^b, is not covered by $\mathcal{P}$.

The idea described above is formalized as a lemma. For a set of binary strings $\mathbb{P}$, we say a unit set X is $\mathbb{P}$-*separated* if every element of X is contained in G_p for some $p \in \mathbb{P}$, i.e., $X = \bigcup_{p \in \mathbb{P}} X_p$. We denote $\mathcal{F}(X) = X \cap (V \cup E)$ and $\mathcal{K}(X) = X \cap (V \times V)$, so $X = \mathcal{F}(X) \cup \mathcal{K}(X)$.

Lemma 5 (Merging Lemma). *Let $\mathbb{P}$ be a set of binary strings such that $\{V_p : p \in \mathbb{P}\}$ is a partition of V. Suppose that there exist k chains $C[i]$, $1 \le i \le k$, such that*

(a) *$C[i] \succ (s_i, t_i)$ for each i, and $C[i]$ and $C[j]$ are disjoint for each $i \ne j$,*
(b) *R is $\mathbb{P}$-separated, where $R = \bigcup_{i=1}^{k} C[i]$,*
(c) *for each $p \in \mathbb{P}$, $0 \le \beta(F_p \cup R_p) \le \beta(G)$,*
(d) *for each $p \in \mathbb{P}$, $\mathcal{K}(R_p) \ne \emptyset$, and*
(e) *for each $p \in \mathbb{P}$, there exists BiDPC[$\mathcal{K}(R_p)|G_p, F_p \cup \mathcal{F}(R_p)$].*

Then, there exists k-BiDPC[$K|G, F$].

Proof. Let $\mathcal{P}_p$=BiDPC[$\mathcal{K}(R_p)|G_p, F_p \cup \mathcal{F}(R_p)$] for $p \in \mathbb{P}$, and let $\mathcal{P} = \bigcup_{p \in \mathbb{P}} \mathcal{P}_p$. By (b), $\mathcal{P}$ is a set of disjoint paths joining each element of $\mathcal{K}(R)$. The existence of $\mathcal{P}$ is guaranteed by (e). Hence, by (a), we can find fault-free paired many-to-many disjoint paths joining S and T by merging vertices of $\mathcal{F}(R)$ and paths in $\mathcal{P}$ using linking edges. We claim that it is a desired BiDPC. It will suffice to show that $\mathcal{F}(R)$ and $\mathcal{P}$ together cover all the fault-free vertices except $\beta(G)$ black ones. By (c) and (d), $\mathcal{F}(R_p)$ and $\mathcal{P}_p$ together cover all the free vertices of V_p except $\beta(F_p \cup R_p)$ black ones in V_p (see G_{01} of Fig. 1). From (1), we obtain $\beta(R) = \sum_{i=1}^{k} \beta((s_i, t_i)) = \beta(K)$. Thus, the number of free vertices that are not covered by $\mathcal{P}$ is:

$$\sum_{p \in \mathbb{P}} \beta(F_p \cup R_p) = \beta(F) + \beta(R) = \beta(F) + \beta(K) = \beta(G).$$

Therefore, we have the claim. □

We are to find chains joining source-sink pairs satisfying the Merging Lemma, Lemma 5. Hereafter in this section and in the next section, we consider a partition of G into two subcubes G_0 and G_1 that are isomorphic to Q_{m-1}. That is, we consider $\mathbb{P} = \{0, 1\}$. For a vertex u, let $N(u)$ denote the set of neighbors of u and let $\bar{u}$ denote the unique member of $N(u) \cap V_{1-p}$, where $p \in \{0, 1\}$ is such that $u \in V_p$. We say that $\bar{u}$ is u's *mate*. For a binary string p, we define the following two functions.

$$h_p^w(K, F) = \beta(F_{p0} \cup K_{p0}) - (k_{p2}^b + k_{p2}^{b,w} + \beta(F_p \cup K_p))$$
$$h_p^b(K, F) = -\beta(F_{p0} \cup K_{p0}) - (k_{p2}^w + k_{p2}^{w,b})$$

We omit the subscription if $p = \epsilon$. We will abbreviate $h^w(K, F)$ and $h^b(K, F)$ respectively as $h^w(G)$ and $h^b(G)$ for simplicity. From $\beta(G) = \beta(F_0 \cup K_0) + \beta(F_1 \cup K_1) + k_2^w - k_2^b$, we obtain the following representations.

$$h^w(G) = \beta(F_0 \cup K_0) - (k_2^b + k_2^{b,w} + \beta(G)) \quad (2)$$
$$= -\beta(F_1 \cup K_1) - (k_2^w + k_2^{b,w}) \quad (3)$$
$$h^b(G) = -\beta(F_0 \cup K_0) - (k_2^w + k_2^{w,b}) \quad (4)$$
$$= \beta(F_1 \cup K_1) - (k_2^b + k_2^{w,b}) - \beta(G)$$

Note that $h^w(G) > 0$ and $h^b(G) > 0$ cannot occur simultaneously since $h^w(G) + h^b(G) = -k_2 - \beta(G) \le 0$. We claim that we can assume $h^b(G) \le 0$ without loss of generality. If $h^b(G) > 0$ ($h^w(G) \le 0$), then we consider an automorphism of G that exchanges G_0 and G_1, i.e., a mapping of every vertex to its mate while preserving its color. (Due to the assumption of $\beta(G) \ge 0$, it is necessary to keep the color of every vertex.) Let $\bar{K}$ and $\bar{F}$ be images of K and F under the mapping, respectively. Then, the problem of finding k-BiDPC$[K|G, F]$ is equivalent to the problem of finding k-BiDPC$[\bar{K}|G, \bar{F}]$. Hence, it will suffice to show that $h^b(\bar{K}, \bar{F}) \le 0$. From these definitions, we obtain

$$\begin{aligned} h^b(\bar{K}, \bar{F}) &= -\beta(\bar{F_0} \cup \bar{K_0}) - (|\bar{K}_2^w| + |\bar{K}_2^{w,b}|) \\ &= -\beta(F_1 \cup K_1) - (k_2^w + k_2^{b,w}) = h^w(G). \end{aligned} \tag{5}$$

Since $h^w(G) \le 0$, the claim is proved. Similarly, we obtain

$$\begin{aligned} h^w(\bar{K}, \bar{F}) &= \beta(\bar{F_0} \cup \bar{K_0}) - (|\bar{K}_2^b| + |\bar{K}_2^{b,w}|) - \beta(G) \\ &= \beta(F_1 \cup K_1) - (k_2^b + k_2^{w,b}) - \beta(G) = h^b(G). \end{aligned} \tag{6}$$

Throughout this paper, we assume

$$h^b(G) \le 0.$$

Then, there are two cases depending on whether $h^w(G) \le 0$ or not.

3.1. Constructions for $h^w(G) \le 0$ ($h^b(G) \le 0$)

Hereafter in this paper, we assume $K_2 = \{(s_i, t_i) : 1 \le i \le k_2\}$, so $K_0 \cup K_1 = \{(s_i, t_i) : k_2 < i \le k\}$. Furthermore, we assume $s_i \in V_0$ for every $(s_i, t_i) \in K_2$; otherwise, it suffices to switch the roles of s_i and t_i.

To prove Theorem 1, we distinguish four subcases. The first two are handled in this section, and the remaining two will be handled in Section 4.

1. For each $p \in \{0, 1\}$, $k_p + k_2 \ge 1$ and $f_p + 2(k_p + k_2) \le m - 1$.
2. For some $p \in \{0, 1\}$, $k_p = k_2 = 0$, $f_{1-p} + 2k_{1-p} \le m - 1$, and $f_p \le m - 3$.
3. For some $p \in \{0, 1\}$, $k_p + k_2 \ge 1$ and $f_p + 2(k_p + k_2) = m$.
4. For some $p \in \{0, 1\}$, $k_p = k_2 = 0$, $f_{1-p} + 2k_{1-p} \le m - 1$, and $f_p = m - 2$.

Case 1. For each $p \in \{0, 1\}$, $k_p + k_2 \ge 1$ and $f_p + 2(k_p + k_2) \le m - 1$.

We build a $\{0, 1\}$-separated simple s_i-t_i chain $\{(s_i, u_i), (\bar{u}_i, t_i)\}$ for each $(s_i, t_i) \in K_2$, where $u_i\bar{u}_i$ is a free edge in E_2.

Procedure BiDPC-A(G, K, F) (See Fig. 2a.)
/* Build chains $C[i] = \{(s_i, u_i), (\bar{u}_i, t_i)\}$ for $1 \le i \le k_2$ and $C[j] = \{(s_j, t_j)\}$ for $j > k_2$. $R = \bigcup_{i=1}^{k} C[i]$. */

1. Select k_2 free edges $u_i\bar{u}_i$ for $1 \le i \le k_2$ such that $u_i \in V_0$ and
   (a) Case $-h^w(G) - \beta(G) \le 0$: $u_i\bar{u}_i \in E_2^{b,w}$ for each $1 \le i \le k_2$.

(b) Case $-h^w(G)-\beta(G) > 0$: $-h^b(G)$ ones are from $E_2^{b,w}$ and $-h^w(G)-\beta(G)$ ones are from $E_2^{w,b}$.

2. Find BiDPC$[\mathcal{K}(R_0)|G_0, F_0]$ and BiDPC$[\mathcal{K}(R_1)|G_1, F_1]$, and merge them using the linking edges $u_i\bar{u_i}$ for $1 \le i \le k_2$.

Lemma 6. *Suppose $h^b(G) \le 0$, $f + 2k \le m$, and $m \ge 4$. Then,* (a) *there exist k_2 free edges in $E_2^{b,w}$ if $-h^w(G) - \beta(G) \le 0$ and* (b) *there exist $-h^b(G)$ free edges in $E_2^{b,w}$ and $-h^w(G) - \beta(G)$ free edges in $E_2^{w,b}$ if $-h^w(G) - \beta(G) > 0$.*

Proof. Let n_p^c be the number of faulty vertices and terminals in V_p^c, where $c \in \{b, w\}$. For example, $n_0^b = f_0^b + 2k_0^b + k_0^o + k_2^b + k_2^{b,w}$ and $n_1^w = f_1^w + 2k_1^w + k_1^o + k_2^w + k_2^{b,w}$.

To prove (a), it suffices to show $f_2 + n_0^b + n_1^w + k_2 \le 2^{m-2}$. Notice that the number of non-free edges in $E_2^{b,w}$ is at most $f_2 + n_0^b + n_1^w$ and that 2^{m-2} is the cardinality of $E_2^{b,w}$. Since $n_0^b = (f_0^b + k_0^b + k_2^b + k_2^{b,w}) + (k_0^b + k_0^o)$ and $-h^w(G) - \beta(G) = (f_0^b + k_0^b + k_2^b + k_2^{b,w}) - (f_0^w + k_0^w)$, the inequality $-h^w(G) - \beta(G) \le 0$ can be rewritten as $f_0^b + k_0^b + k_2^b + k_2^{b,w} \le f_0^w + k_0^w$. Hence,

$$n_0^b = (f_0^b + k_0^b + k_2^b + k_2^{b,w}) + (k_0^b + k_0^o) \le (f_0^w + k_0^w) + (k_0^b + k_0^o) \le f_0 + k_0.$$

Since $n_1^w \le f_1 + 2k_1 + k_2$ and $f + 2k \le m \le 2^{m-2}$, it follows

$$f_2 + n_0^b + n_1^w + k_2 \le f_2 + (f_0 + k_0) + (f_1 + 2k_1 + k_2) + k_2 \le f + 2k \le m \le 2^{m-2}.$$

To prove (b), we show the following:

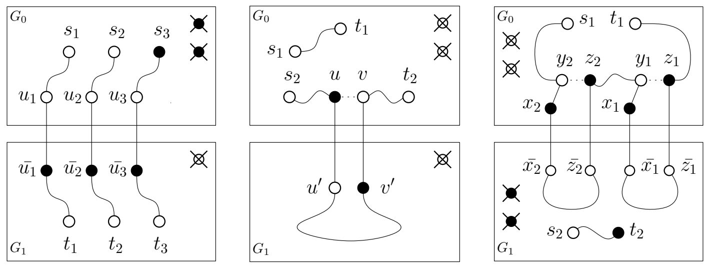
$$f_2 + n_0^b + n_1^w - h^b(G) \le f + 2k \le m \le 2^{m-2}. \tag{7}$$

$$f_2 + n_0^w + n_1^b - h^w(G) - \beta(G) \le f + 2k \le m \le 2^{m-2}. \tag{8}$$

Notice that $f_2 + n_0^b + n_0^w + n_1^b + n_1^w \le f + 2k$. From (4) and (2), we obtain $-h^b(G) = f_0^w + k_0^w + k_2^w + k_2^{w,b} - f_0^b - k_0^b \le n_0^w$ and $-h^w(G) - \beta(G) = f_0^b + k_0^b + k_2^b + k_2^{b,w} - f_0^w - k_0^w \le n_0^b$. By plugging them into (7) and (8), we deduce (b). □

Lemma 7. *Suppose $f + 2k \le m$, $m \ge 4$, $h^b(G) \le 0$, and $h^w(G) \le 0$. Also, suppose $k_p + k_2 \ge 1$ and $f_p + 2(k_p + k_2) \le m - 1$ for any $p \in \{0, 1\}$. Then, Procedure BiDPC-A constructs BiDPC$[K|G, F]$.*

Proof. It will suffice to show that every condition of the Merging Lemma is satisfied. Conditions (a) and (b) are trivial from the construction. Notice that they imply $\beta(F_1 \cup R_1) = \beta(G) - \beta(F_0 \cup R_0)$. Thus, to prove (c), it suffices to show $0 \le \beta(F_0 \cup R_0) \le \beta(G)$. We observe $\beta(F_0 \cup R_0) = \beta(F_0 \cup K_0) + \alpha$, where $\alpha = (k_2^w + k_2^{w,b} - k_2^b - k_2^{b,w})/2 + \beta(U)/2$, where $U = \{u_i : 1 \le i \le k_2\}$. We distinguish two cases. Suppose for the first case that $-h^w(G) - \beta(G) \le 0$. Since $h^w(G) \le 0$, it follows $-\beta(G) \le h^w(G) \le 0$. From $\beta(F_0 \cup R_0) = \beta(F_0 \cup K_0) - (k_2^b + k_2^{b,w}) = h^w(G) + \beta(G)$, we conclude that $0 \le \beta(F_0 \cup R_0) =$



(a) BiDPC-A (b) BiDPC-B (c) BiDPC-C

Figure 2: Illustrations of procedures in Section 3.

$h^w(G) + \beta(G) \leq \beta(G)$. Suppose for the second case that $-h^w(G) - \beta(G) > 0$. Because $\beta(U) = -2\beta(F_0 \cup K_0) + (-k_2^w - k_2^{w,b} + k_2^b + k_2^{b,w})$, it follows $\alpha = -\beta(F_0 \cup K_0)$. Hence, $\beta(F_0 \cup R_0) = 0$. Therefore, we have condition (c). Since $|\mathcal{K}(R_0)| = k_0 + k_2 \geq 1$ and $|\mathcal{K}(R_1)| = k_1 + k_2 \geq 1$, we have condition (d). Finally, since $f_0 + 2(k_0 + k_2) \leq m - 1$ and $f_1 + 2(k_1 + k_2) \leq m - 1$, there exists BiDPC$[\mathcal{K}(R_p)|G_p, F_p]$ for each $p \in \{0,1\}$ by the induction hypothesis. Condition (e) is satisfied. This finishes the proof. □

Case 2. For some $p \in \{0,1\}$, $k_p = k_2 = 0$, $f_{1-p} + 2k_{1-p} \leq m-1$, and $f_p \leq m-3$.

Let us consider a more general situation for future use in Section 5. Let $\mathbb{P}'$ be a set of binary strings such that $\{V_q : q \in \mathbb{P}'\}$ is a partition of V. Suppose that K is $\mathbb{P}'$-separated. Let $p, q \in \mathbb{P}'$ be such that every vertex of V_q has a neighbor in V_p, $f_q + 2k_q \leq m - l(q)$, and $f_p + 2k_p \leq m - l(p) - 2$. Suppose $k_q \geq 1$ and $k_p \geq 0$. Notice that we have the case condition if $\mathbb{P}' = \{0,1\}$, $p \in \{0,1\}$, $q = 1 - p$, and $k_p = 0$.

We utilize a closed chain $C = \{(u,v),(v',u')\}$, where $u, v \in V_q$, $u' \in N(u) \cap V_p$, and $v' \in N(v) \cap V_p$. Let $(s,t) \in K_q$. We merge C into an s-t chain in the form of $\{(s,u),(u',v'),(v,t)\}$ so that the s-t path will eventually cover vertices of G_p. Let $R_q = (K_q \setminus (s,t)) \cup \{(s,u),(v,t)\}$. Then, we need to find f'-fault k'-BiDPC$[\mathcal{K}(R_q)|G_q, F_q \cup \mathcal{F}(R_q)]$, where it is possible that $f' + 2k' > m - l(q)$. The following Procedure BiDPC-B builds such a BiDPC using an f_q-fault k_q-BiDPC.

Procedure BiDPC-B(G, K, F, q, p) (See Fig. 2b.)
/* Merge a closed chain $C = \{(u,v),(v',u')\}$ into an s-t chain in the form of $\{(s,u),(u',v'),(v,t)\}$, where $(s,t) \in K_q$, $u' \in N(u) \cap V_p$, and $v' \in N(v) \cap V_p$. */

1. Find $\mathcal{P}_q$=f_q-fault k_q-BiDPC$[K_q|G_q, F_q]$.
2. Find an edge uv on $\mathcal{P}_q$ such that $uu', vv' \notin F$ and both u' and v' are free, where $u' \in N(u) \cap V_p$ and $v' \in N(v) \cap V_p$. Let $P = (s, P_u, u, v, P_v, t)$ be the path of $\mathcal{P}_q$ that covers uv, where $(s,t) \in K_q$.
3. Find $\mathcal{P}_p$=f_p-fault (k_p+1)-BiDPC$[K_p \cup \{(u',v')\}|G_p, F_p]$.
4. Merge $\mathcal{P}_p$ and $\mathcal{P}_q \setminus P$, and two paths (s, P_u, u) and (v, P_v, t) using linking edges uu' and vv'.

Lemma 8. *Suppose $f + 2k \leq m$, $m \geq 4$, $h^b(G) \leq 0$, and $h^w(G) \leq 0$. Also, suppose $k = k_{1-p}$, $f_{1-p} + 2k_{1-p} \leq m - 1$, and $f_p \leq m - 3$ for $p \in \{0, 1\}$. Then, applying BiDPC-B$(G, K, F, 1 - p, p)$ constructs BiDPC$[K|G, F]$.*

Proof. Let $q = 1 - p$. Since $f_q + 2k_q \leq m - 1$, there exists $\mathcal{P}_q$ by the induction hypothesis. We claim the existence of an edge uv at Step 2. From $k_2 = 0$ and $h^b(G), h^w(G) \leq 0$, we obtain $0 \leq \beta(F_p \cup K_p), \beta(F_q \cup K_q) \leq \beta(G)$. There are $2^{m-1} - f_q^w - f_q^b - \beta(F_q \cup K_q) - k_q$ candidates for uv, i.e., edges in $\mathcal{P}_q$. Since each element of $F_2 \cup F_p^w \cup F_p^b$ can block at most two candidates, there are at most $2(f_2 + f_p^w + f_p^b)$ blocked candidates. Since $f_q^w + f_q^b \leq f_q$, $\beta(F_q \cup K_q) \leq f_q + k_q$, $f_p^w + f_p^b \leq f_p$, and $f + k \leq m - 1$, it follows

$$\begin{aligned}
&2^{m-1} - f_q^w - f_q^b - \beta(F_q \cup K_q) - k_q - 2(f_2 + f_p^w + f_p^b) \\
&\geq 2^{m-1} - 2(f_q + k_q + f_2 + f_p) \qquad (9)\\
&= 2^{m-1} - 2(f + k) \geq 2^{m-1} - 2(m - 1) \geq 2.
\end{aligned}$$

Therefore, we have the claim.

Now, we show that the path set built at Step 4 is a desired BiDPC. Let $C[i] = \{(s_i, t_i)\}$ if $(s, t) \neq (s_i, t_i)$. Let $C[i] = \{(s_i, u), (u', v'), (v, t_i)\}$ if $(s, t) = (s_i, t_i)$. Let $R = \bigcup_{i=1}^{k} C[i]$. The Merging Lemma's conditions (a), (b), and (d) are obvious. We recall that $0 \leq \beta(F_q \cup K_q), \beta(F_p \cup K_p) \leq \beta(G)$. From $\beta((u, v)) = 0$, we obtain $\beta(F_p \cup R_p) = \beta(F_p \cup K_p)$ and $\beta(F_q \cup R_q) = \beta(F_q \cup K_q)$. Hence, we have condition (c). Since $\mathcal{P}_q$ covers $2^{m-1} - f_q^w - f_q^b - \beta(F_q \cup K_q)$ vertices and $\beta(K_q) = \beta(R_q)$, two paths (s, P_u, u) and (v, P_v, t) and paths in $\mathcal{P}_q \backslash P$ together form BiDPC$[\mathcal{K}(R_q)|G_q, F_q \cup \mathcal{F}(R_q)]$. Since $f_p + 2(k_p + 1) \leq m - 1$, there exists $\mathcal{P}_p$ by the induction hypothesis. Condition (e) is satisfied. The proof is finished. □

3.2. Constructions for $h^w(G) > 0$ ($h^b(G) \leq 0$)

We sketch our approach. First, we build chains by following the construction of Step 1 of Procedure BiDPC-A. Then, we obtain a set of chains R such that $\beta(F_0 \cup R_0) = \beta(F_0 \cup K_0) - (k_2^b + k_2^{b,w}) = h^w(G) + \beta(G)$ and $\beta(F_1 \cup R_1) = -h^w(G)$. To meet condition (c) of the Merging Lemma, we add $h^w(G)$ black units to G_0 and the same number of white units to G_1 in the form of closed chains $\{(x_i, z_i), (\bar{z}_i, \bar{x}_i)\}$, where $x_i, z_i \in V_0^b$ and $1 \leq i \leq h^w(G)$. For example, if there exists a vertex pair $(s, t) \in \mathcal{K}(R_0)$, such closed chains can be joined to an s-t chain in the form of $\{(s, x_1), (\bar{x_1}, \bar{z_1}), (z_1, x_2), \ldots, (z_{h^w(G)}, t)\}$. This approach requires an f_0-fault $(k_0 + k_2 + h^w(G))$-BiDPC in G_0, and the following Procedure BiDPC-C builds it using an f_0-fault (or $(f_0 + h^w(G))$-fault) $(k_0 + k_2)$-BiDPC.

Procedure BiDPC-C(G, K, F) (See Fig. 2c.)
/* Build chains $C[i] = \{(s_i, u_i), (\bar{u}_i, t_i)\}$ for $1 \leq i \leq k_2$, $C[j] = \{(s_j, t_j)\}$ for $j > k_2$, then merge them with closed chains $\{(x_i, z_i), (\bar{z}_i, \bar{x}_i)\}$ for $1 \leq i \leq h^w(G)$. $R = \bigcup_{i=1}^{k} C[i]$. */

1. Select k_2 free edges $u_i\bar{u}_i$ for $1 \leq i \leq k_2$, where $u_i \in V_0^b$. Let $C[i] = \{(s_i, u_i), (\bar{u}_i, t_i)\}$ for $1 \leq i \leq k_2$ and let $C[j] = \{(s_j, t_j)\}$ for $j > k_2$.

2. Case $f_1 = f_1^b$ and $f_2 + k_1 + k_2^w + k_2^{b,w} = 0$: Find $\mathcal{P}$=f_0-fault $(k_0 + k_2)$-BiDPC$[R_0|G_0, F_0]$. Let $X = \{x_i : 1 \le i \le h^w(G)\}$ be a set of $h^w(G)$ free vertices of V_0^b that are not covered by $\mathcal{P}$.
3. Case $f_1 > f_1^b$ or $f_2 + k_1 + k_2^w + k_2^{b,w} \ge 1$: Select $h^w(G)$ free edges $x_i\bar{x_i}$, $1 \le i \le h^w(G)$, such that $x_i \in V_0^b$ and let $X = \{x_i : 1 \le i \le h^w(G)\}$. Then, find $\mathcal{P}$=$(f_0 + h^w(G))$-fault $(k_0 + k_2)$-BiDPC$[R_0|G_0, F_0 \cup X]$.
4. For each $x_i \in X$, choose a vertex $y_i \in N(x_i)$ that is covered by $\mathcal{P}$ such that (i) $y_i \ne y_j$ if $i \ne j$, (ii) y_i is not a sink, (iii) x_iy_i is fault-free, (iv) $z_i \notin \{u_j : 1 \le j \le k_2\}$, (v) $z_i\bar{z_i}$ is fault-free, and (vi) $\bar{z_i}$ is free, where z_i is the vertex next to y_i in the direction of the sink in the path of $\mathcal{P}$ that covers y_i; in s_i-u_i paths, u_is are regarded as sinks.
5. For each $P \in \mathcal{P}$, if P contains some y_is, say $P = (s, P_{j_0}, y_{j_1}, z_{j_1}, P_{j_1}, y_{j_2}, z_{j_2}, \ldots, z_{j_n}, P_{j_n}, t)$, then replace (s, t) with (s, x_{j_1}), $(\bar{x_{j_1}}, \bar{z_{j_1}})$, (z_{j_1}, x_{j_2}), …, (z_{j_n}, t) at some $C[j]$ such that $(s, t) \in C[j]$; and replace P with the paths $(s, P_{j_0}, y_{j_1}, x_{j_1})$, $(z_{j_1}, P_{j_1}, y_{j_2}, x_{j_2})$, …, (z_{j_n}, P_{j_n}, t) in $\mathcal{P}$.
6. Find $\mathcal{P}' = f_1$-fault $(k_1 + k_2 + h^w(G))$-BiDPC$[R_1|G_1, F_1]$.
7. Merge $\mathcal{P}$ and $\mathcal{P}'$ by using linking edges $x_i\bar{x_i}$, $z_i\bar{z_i}$, and $u_j\bar{u_j}$, where $1 \le i \le h^w(G)$ and $1 \le j \le k_2$.

Lemma 9. *Suppose $f + 2k \le m$, $m \ge 4$, $h^b(G) \le 0$, and $h^w(G) > 0$. Also, suppose $k_0 + k_2 \ge 1$ and $2h^w(G) \le f_0 + f_2 + 2k_0 - 1$. Then, Procedure BiDPC-C constructs BiDPC$[K|G, F]$.*

Proof. For Steps 1 through 3, we need to show the existence of $k_2 + h^w(G)$ free edges in $E_2^{b,w}$. Let n_p^c be the number of faulty vertices and terminals in V_p^c, where $c \in \{b, w\}$. Then, our goal is to show $(n_0^b + n_1^w + f_2) + (k_2 + h^w(G)) \le 2^{m-2}$. Since $(n_0^b + n_1^w + f_2) + n_0^w + n_1^b \le f + 2k$, it suffices to show $k_2 + h^w(G) \le n_0^w + n_1^b$. Adding k_2 to both sides of (2) gives $k_2 + h^w(G) = f_0^w + k_0^w + k_2^w + k_2^{w,b} - \beta(G) - (f_0^b + k_0^b)$. We are done since $f_0^w + k_0^w + k_2^w + k_2^{w,b} \le n_0^w$.

Next, we show the existence of $\mathcal{P}$ at Steps 2 and 3. Suppose that Step 2 is taken. We are to show $f_0 + 2(k_0 + k_2) \le m - 1$. Suppose to the contrary that $f_0 + 2(k_0 + k_2) = m$. Then, $f_1 + f_2 + k_1 = 0$. Then, by (3), we obtain $h^w(G) = -k_2^w - k_2^{b,w} \le 0$, a contradiction. Suppose instead that Step 3 is taken. We are to show $f_0 + h^w(G) + 2(k_0 + k_2) \le m - 1$. From (3), $f + 2k \le m$, and $f_1 > f_1^b$ or $f_2 + k_1 + k_2^w + k_2^{b,w} \ge 1$, we obtain

$$\begin{aligned} &f_0 + 2(k_0 + k_2) + h^w(G) \le m - (f_1 + f_2) - 2k_1 + h^w(G) \\ &= m - (f_1 - f_1^b) - f_2 - (2k_1 - k_1^b) - (f_1^w + k_1^w + k_2^w + k_2^{b,w}) \le m - 1. \end{aligned}$$

We claim that we can choose y_is at Step 4. Since y_i is a neighbor of x_i in G_0, there are $m-1$ candidates for each y_i. Each graph element in the following may block at most one candidate of y_i: (i) $i-1$ vertices in $N(x_i) \cap V_0$ chosen as y_j for $1 \le j < i$, (ii) k_0 sink vertices in G_0, (iii) f_0^o faulty edges, (iv) k_2 vertices $u_1, u_2, \ldots, u_{k_2}$, (v) f_2 faulty edges between V_0 and V_1, (vi) n_1^w white faulty vertices and terminals in G_1, and (vii) f_0^w white faulty vertices in G_0. Notice that $i-1 \leq h^w(G)-1$ for (i) and that vertices u_j, $1 \leq j \leq k_2$, do not overlap to any y_i for (ii). They together block at most $(h^w(G)-1)+k_0+f_0^o+k_2+f_2+n_1^w+f_0^w$ candidates. Since $h^w(G) = -f_1^w - k_1^w - (k_2^w + k_2^{b,w}) + f_1^b + k_1^b$ by (3) and $n_1^w = f_1^w + k_1^w + (k_1 - k_1^b) + k_2^w + k_2^{b,w}$, we obtain

$$\begin{aligned}&(h^w(G)-1)+k_0+f_0^o+k_2+f_2+n_1^w+f_0^w \\ &= f_1^b + f_0^o + f_0^w + f_2 + k_0 + k_1 + k_2 - 1 \leq f + k - 1 < m - 1.\end{aligned}$$

Therefore, we have the claim.

It remains to check the conditions of the Merging Lemma. Conditions (a) and (b) are obvious from the construction. Since $\beta(\{(s, x_{j_1}), (z_{j_1}, x_{j_2}), \ldots, (z_{j_n}, t)\}) = \beta((s,t)) - n$, it follows $\beta(F_0 \cup R_0) = \beta(G)$ and $\beta(F_1 \cup R_1) = 0$. Since $k_0 + k_2 \geq 1$ and R_1 contains a vertex pair $(\bar{x_1}, \bar{z_1})$, we have $\mathcal{K}(R_0) \neq \emptyset$ and $\mathcal{K}(R_1) \neq \emptyset$. Conditions (c) and (d) are verified. Since Step 5 inserted $h^w(G)$ free vertices of V_0^b to $\mathcal{P}$, at this point, $\mathcal{P}$ contains all the free vertices of V_0 except $\beta(G)$ black ones. It still holds that $\mathcal{P}$ is a set of paths joining elements of $\mathcal{K}(R_0)$. Notice that $\mathcal{K}(R_0) = R_0$. Hence, $\mathcal{P}$ is BiDPC$[\mathcal{K}(R_0)|G_0, F_0 \cup \mathcal{F}(R_0)]$. Since $2h^w(G) \leq f_0 + f_2 + 2k_0 - 1$, it follows $f_1 + 2(k_1 + k_2 + h^w(G)) \leq m - (f_0 + f_2 + 2k_0) + 2h^w(G) \leq m - 1$. Thus, there exists $\mathcal{P}'$ by the induction hypothesis. Condition (e) is satisfied. This finishes the proof. □

The following Procedure BiDPC-C1 considers the case when $k_1 + k_2 \geq 1$ and $2h^w(G) \leq f_1 + f_2 + 2k_1 - 1$. Its construction is symmetric to that of Procedure BiDPC-C, and therefore its correctness proof is symmetric to that of Lemma 9, too. Therefore, we omit the proof of Lemma 10 below.

Procedure BiDPC-C1(G, K, F)
/* Steps 1, 4, 5, and 7 are identical to the corresponding steps of Procedure BiDPC-C. */

2. Case $f_0 = f_0^w$ and $f_2 + k_0 + k_2^b + k_2^{b,w} = 0$: Find $\mathcal{P}$=f_1-fault $(k_1 + k_2)$-BiDPC$[R_1|G_1, F_1]$. Let $X = \{x_i : 1 \leq i \leq h^w(G)\}$ be the set of $h^w(G)$ free vertices of V_1^w that are not covered by $\mathcal{P}$.
3. Case $f_0 > f_0^w$ or $f_2 + k_0 + k_2^b + k_2^{b,w} \geq 1$: Select $h^w(G)$ free edges $x_i\bar{x_i}$, $1 \leq i \leq h^w(G)$, such that $x_i \in V_1^w$ and let $X = \{x_i : 1 \leq i \leq h^w(G)\}$. Then, find $\mathcal{P}$=$(f_1 + h^w(G))$-fault $(k_1 + k_2)$-BiDPC$[R_1|G_1, F_1 \cup X]$.
6. Find $\mathcal{P}' = f_0$-fault $(k_0 + k_2 + h^w(G))$-BiDPC$[R_0|G_0, F_0]$.

Lemma 10. *Suppose $f + 2k \leq m$, $h^b(G) \leq 0$, $h^w(G) > 0$, and $m \geq 4$. Also, suppose $k_1 + k_2 \geq 1$ and $2h^w(G) \leq f_1 + f_2 + 2k_1 - 1$. Then, Procedure BiDPC-C1 constructs BiDPC[$K|G, F$].*

Lemmas 9 and 10 above leave open the case when $k_p + k_2 = 0$ or $2h^w(G) > f_p + f_2 + 2k_p - 1$ for each $p \in \{0, 1\}$. This case will be considered in Section 5.

We close this section with a lemma which is useful to check conditions $2h^w(G) \leq f_p + f_2 + 2k_p - 1$ of Lemmas 9 and 10, where $p \in \{0, 1\}$.

Lemma 11. (a) $f_0^w \le 3f_0^b+f_0^o+f_2+4k_0^b+2k_0^o+2k_2^b+2k_2^{b,w}+2\beta(G)-1$ *if and only if* $2h^w(G) \le f_0+f_2+2k_0-1$. (b) $f_1^b \le 3f_1^w+f_1^o+f_2+4k_1^w+2k_1^o+2k_2^w+2k_2^{b,w}-1$ *if and only if* $2h^w(G) \le f_1+f_2+2k_1-1$.

Proof. Let us show part (a). We recall that $\beta(F_0 \cup K_0) = f_0^w + k_0^w - f_0^b - k_0^b$. Substituting (2) into $2h^w(G) \le f_0 + f_2 + 2k_0 - 1$ gives

$$2(f_0^w + k_0^w - (f_0^b + k_0^b) - (k_2^b + k_2^{b,w} + \beta(G))) \le f_0 + f_2 + 2k_0 - 1.$$

By rearranging it, we obtain (a). Similarly, we substitute (3) into $2h^w(G) \le f_1 + f_2 + 2k_1 - 1$ to obtain (b). □

4. Proof of Theorem 1 for $f^b \le 1$

We begin by considering a special case that the total number of white faulty vertices and white terminals is at most one, i.e., $f^w + k^o + 2k^w \le 1$.

Lemma 12. *Suppose* $\beta(G) \ge 0$*,* $f + 2k \le m$*, and* $f^w + k^o + 2k^w \le 1$*. Then, there exists BiDPC[*$K|G, F$*].*

Proof. Suppose $f^w + k^o + 2k^w = 0$. Then, we have $\beta(K) < 0$ and $\beta(F) \le 0$, which contradict $\beta(G) \ge 0$. Now, suppose $f^w = 0$ and $k^o + 2k^w = 1$. Since $\beta(G) = (f^w + k^w) - (f^b + k^b) \ge 0$, the only possibility is that $\beta(G) = 0$, $f^w = f^b = k^w = k^b = 0$, $k = k^o = 1$, and $f = f^o$. Hence, Lemma 4 applies. Suppose instead that $f^w = 1$ and $k^o + 2k^w = 0$. Since $\beta(G) \ge 0$, the only possibility is that $f^w = 1$, $f^b = 0$, $k = k^b = 1$, and $\beta(G) = 0$. Hence, Lemma 4 applies. □

Hereafter in this section, due to Lemma 12 above, we assume $f^w+k^o+2k^w \ge 2$. In what follows, we need the following assumptions.

(A1) Each of G_0 and G_1 contains at least one white faulty vertex or white terminal. That is, $f_0^w+k_0^w+k_0^o+k_2^w+k_2^{w,b} \ge 1$ and $f_1^w+k_1^w+k_1^o+k_2^w+k_2^{b,w} \ge 1$.

(A2) If $k_2^w = 0$, then $k^w = 0$.

(A3) If $k_2 = 1$, and $k_0 = 0$ or $k_1 = 0$, then $s_1 \ne \bar{t_1}$, where $K_2 = \{(s_1, t_1)\}$.

We do not lose any generality by our assumptions. There exist two white faulty vertices and/or terminals w_1 and w_2; we choose $(w_1, w_2) \in K^w$ whenever possible. It is obvious that w_1 and w_2 differ in at least two bits, say the ith bit and the jth bit. Assuming either $i = 1$ or $j = 1$ establishes (A1) and (A2). Suppose that (A3) is violated when $i = 1$ and when $j = 1$. Then, there exist two source-sink pairs (s, t) and (s', t') such that, without loss of generality, s and s' are white, t and t' are black, s and t differ in the ith bit only, and s' and t' differ in the jth bit only. If s and s' differ in the ith bit and the jth bit only, then $t = t'$, a contradiction. Therefore, there exists an l such that $1 \le l \le m$, $l \notin \{i, j\}$, and s and s' differ in the lth bit. Consequently, by assuming $l = 1$

and assigning s and s' respectively as w_1 and w_2, we obtain $k_0, k_1 \geq 1$. In this way, all assumptions are established.

Without loss of generality, we can assume $h^b(G) \leq 0$. Suppose otherwise. Then, since $h^w(G) > 0$ and $h^b(G) > 0$ do not happen simultaneously, it follows $h^b(G) > 0$ and $h^w(G) \leq 0$. By exchanging G_0 and G_1, as described in Section 3, we obtain $h^b(G) \leq 0$ and $h^w(G) > 0$ (see (5) and (6)). Recall that this exchange does not invalidate $\beta(G) \geq 0$. Also, assumptions (A1), (A2), and (A3) still hold true. With the aid of assumption (A1), we can easily handle the case when $h^w(G) > 0$.

Lemma 13. *Suppose $f + 2k \leq m$, $m \geq 4$, $f^b \leq 1$, $h^b(G) \leq 0$, and $h^w(G) > 0$. Then, Procedure BiDPC-C1 constructs BiDPC[$K|G, F$].*

Proof. We are to show $k_1 + k_2 \geq 1$ and $2h^w(G) \leq f_1 + f_2 + 2k_1 - 1$. To show $k_1 + k_2 \geq 1$, suppose to the contrary that $k_1 + k_2 = 0$. Since G_1 has no white terminal, it follows $f_1^w \geq 1$ by (A1). Since $h^w(G) = f_1^b - f_1^w > 0$ by (3), it follows $f_1^b \geq 2$, which contradicts $f^b \leq 1$. It remains to show $2h^w(G) \leq f_1 + f_2 + 2k_1 - 1$. From (A1), we obtain $2f_1^w + 2k_1^w + 2k_1^o + 2k_2^w + 2k_2^{b,w} - 1 \geq 1 \geq f_1^b$. By Lemma 11(b), we obtain $2h^w(G) \leq f_1 + f_2 + 2k_1 - 1$. □

The rest of this section covers the case when $h^w(G) \leq 0$. There are four cases discussed in Section 3.1. Since Cases 1 and 2 have already been considered, it remains to consider Cases 3 and 4.

Case 3. For some $p \in \{0, 1\}$, $k_p + k_2 \geq 1$ and $f_p + 2(k_p + k_2) = m$.

We assume $k_1 + k_2 \geq 1$ and $f_1 + 2(k_1 + k_2) = m$ ($f = f_1$ and $k = k_1 + k_2$). Otherwise, we exchange G_0 and G_1. We recall that this exchange does not invalidate $\beta(G) \geq 0$ and $h^b(G), h^w(G) \leq 0$. From (A1) and $f_0 = 0$, we obtain $k_2 \geq 1$.

We follow the construction of Procedure BiDPC-A with some modifications. Basically, we build simple chains $\{(s_i, u_i), (\bar{u_i}, t_i)\}$ for every $(s_i, t_i) \in K_2$, where $u_i \in V_0$. Although G_1 may not be f-fault k-bicoverable, we can use $(f+1)$-fault $(k-1)$-bicoverability or $(f-1)$-fault k-bicoverability of G_1 instead. We use k_2-bicoverability of G_0 if $2k_2 < m$, and Lemma 2 or 3 if $2k_2 = m$. In order to use Lemma 3, we need $\beta(\{(s_i, u_i) : 1 \leq i \leq k_2\}) = 0$ and two s_iu_is forming edges.

Case 3.1. $k \geq 2$ and there exists an $(s_i, t_i) \in K_2$ such that $\bar{t_i}$ is free.

Without loss of generality, we can assume that $\bar{t_1}$ is free. Recall that we are assuming $(s_i, t_i) \in K_2$ and $s_i \in V_0$ for $1 \leq i \leq k_2$. It will be useful that $\{(s_1, \bar{t_1}), t_1\} \succ (s_1, t_1)$. The detailed construction is given in the following Procedure BiDPC-D.

Procedure BiDPC-D(G, K, F) (See Fig. 3a.)
/* Build chains $C[1] = \{(s_1, \bar{t_1}), t_1\}$, $C[i] = \{(s_i, u_i), (\bar{u_i}, t_i)\}$ for $2 \leq i \leq k_2$, and $C[j] = \{(s_j, t_j)\}$ for $j > k_2$. $R = \bigcup_{i=1}^{k} C[i]$. */

1. If $(s_1, t_1) \in K_2^o$ and there exists $(s_i, t_i) \in K_2^b \cup K_2^w$ such that $\bar{t_i}$ is free, then exchange (s_1, t_1) and (s_i, t_i). If $k_2 \geq 2$, we assume that $\beta(s_2) = -\beta(s_1)$ or $\beta(s_i) = \beta(s_1)$ for every $2 \leq i \leq k_2$.

2. Select $k_2 - 1$ free edges $u_i\bar{u_i}$, $2 \le i \le k_2$, such that $u_i \in V_0$ and
   (a) Case $\beta(s_1) = \beta(t_1)$: $\beta(s_i) = -\beta(u_i)$ and if $k_2 \ge 3$, s_2u_2 and s_3u_3 form edges.
   (b) Case $\beta(s_1) = -\beta(t_1)$, $k_2 \ge 2$, and $\beta(s_2) = -\beta(s_1)$: $\beta(s_2) = \beta(u_2)$, $\beta(s_j) = -\beta(u_j)$ for $j \ge 3$, and if $k_2 \ge 4$, s_3u_3 and s_4u_4 form edges.
3. Find $\mathcal{P}_0$=k_2-BiDPC$[\mathcal{K}(R_0)|G_0, \emptyset]$ and $\mathcal{P}_1$=$(f_1 + 1)$-fault $(k_1 + k_2 - 1)$-BiDPC$[\mathcal{K}(R_1)|G_1, F_1 \cup \{t_1\}]$. Merge $\mathcal{P}_0$ and $\mathcal{P}_1$ with the linking edges.

Lemma 14. *Suppose $f + 2k \le m$, $f_0 + f_2 + k_0 = 0$, and $k \ge 2$. Also, suppose that there exists $(s_1, t_1) \in K_2$ such that $\bar{t_1}$ is free. Then Procedure BiDPC-D constructs BiDPC[$K|G, F$] if one of the following is satisfied:*

- *$\beta(s_1) = \beta(t_1)$.*
- *$\beta(s_1) = -\beta(t_1)$, there exists $(s_2, t_2) \in K_2$ such that $\beta(s_2) = -\beta(s_1)$. In addition, $f_1 + k_1 \ge 1$ or $k_2 \ge 4$.*
- *$\beta(s_1) = -\beta(t_1)$, $k_1 \ge 1$, $k_2 = 1$, and $\beta(G) \ge 1$.*

Proof. We claim the existence of free edges at Step 2. We need at most $k_2^w + k_2^{w,b}$ free edges of $E_2^{b,w}$ and at most $k_2^b + k_2^{b,w}$ free edges of $E_2^{w,b}$. Since $f_0 = k_0 = 0$, it follows $k_2^w + k_2^{w,b} = -h^b(G)$ and $k_2^b + k_2^{b,w} = -h^w(G) - \beta(G)$. By Lemma 6, we have the claim. We further claim that we can sequentially choose u_2 and u_3 at Step 2(a). There are $m - 1$ candidates for each of u_2 and u_3 and at most $m - 2$ of them are blocked by faulty vertices and terminals, as s_2 (resp. s_3) blocks no candidate of u_2 (resp. u_3) and because s_1 and t_1 together block at most one candidate of u_2 and of u_3. Although u_2 and u_3 should be distinct, either s_2 or u_2 blocks at most one candidate for u_3 since s_2 and u_2 have different colors. Therefore, there remain at least one candidate of u_3 after we choose u_2. The claim is proved. Similarly, we can pick up $k_2 - 1$ free edges satisfying the conditions of Step 2(b).

To show that the result of Step 3 is a desired BiDPC, we check conditions of the Merging Lemma. Let us consider the first part, $\beta(s_1) = \beta(t_1)$. From the construction, conditions (a), (b), and (d) are obvious. Similar to the proof of Lemma 7, condition (c) follows from $\beta(F_0 \cup R_0) = 0$ and conditions (a) and (b). To verify condition (e), since $F_0 = \mathcal{F}(R_0) = \emptyset$ and $\mathcal{F}(R_1) = \{t_1\}$, it will suffice to show the existence of $\mathcal{P}_0$ and $\mathcal{P}_1$. Since $(f_1 + 1) + 2(k_1 + k_2 - 1) < m$, there exists $\mathcal{P}_1$ by the induction hypothesis. The existence of $\mathcal{P}_0$ is guaranteed by the induction hypothesis if $f_1 + k_1 \ge 1$, by Lemma 3 if $f_1 + k_1 = 0$ and $k_2 \ge 3$, and by Lemma 2 if $f_1 + k_1 = 0$ and $k_2 = 2$. The proof for the second part is similar. (The existence of $\mathcal{P}_0$ is from the induction hypothesis if $f_1 + k_1 \ge 1$, and from Lemma 3 if $f_1 + k_1 = 0$ and $k_2 \ge 4$.) Let us consider the third part. Conditions (a), (b), and (d) are obvious. By (A1), s_1 is white, so t_1 is black and $\beta((s_1, \bar{t_1})) = 1$. Since $\beta(G) \ge 1$, it follows $0 \le \beta(R_0) = 1 \le \beta(G)$ and $0 \le \beta(F_1 \cup R_1) = \beta(G) - 1 \le \beta(G)$. Thus, condition (c) is verified. There exist $\mathcal{P}_0$ and $\mathcal{P}_1$ by the induction hypothesis. Hence, we have condition (e). This finishes the proof. □

By the selection of (s_1, t_1) in Step 1 of Procedure BiDPC-D, hereafter in this section, we assume that $\bar{t_i}$ is not free for each $(s_i, t_i) \in K_2^b \cup K_2^w$. Lemma 14 above

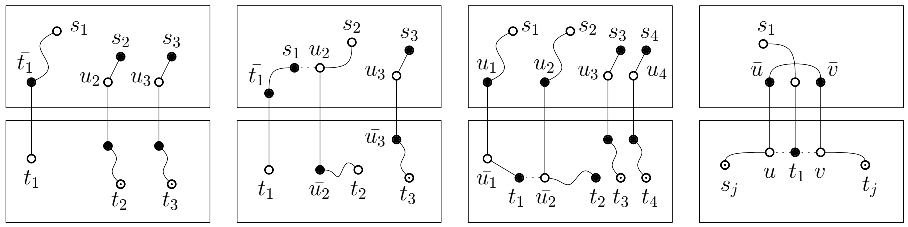


Figure 3: Illustrations of the constructions in Section 4.

leaves open the following two cases: (i) $f = 0$, $k = k_2 \in \{2, 3\}$, $\beta(s_1) = -\beta(t_1)$, and there exists $(s_2, t_2) \in K_2$ such that $\beta(s_2) = -\beta(s_1)$, and (ii) $\beta(s_1) = -\beta(t_1)$ and there exists no $(s, t) \in K_2$ such that $\beta(s) = -\beta(s_1)$. Let us consider case (i). Suppose $k_2 = 2$. If $\beta(G) = 0$, then Lemma 2 applies. Let $\beta(G) \geq 1$. Then, the only possibility is that $k_2^o = k_2^w = 1$ and $\beta(G) = 1$, i.e., $(s_2, t_2) \in K_2^w$. Since $\bar{t_2}$ is not free, it follows that s_1 is a black vertex, so t_1 is a white vertex. This implies that $\bar{s_2}$ is free. Thus, by exchanging G_0 and G_1, we come to the case where there exists a source-sink pair $(s, t) \in K_2^w$ such that $\bar{t}$ is free. It is the case in which the first part of Lemma 14 is applicable. The following Lemma 15 considers when $k_2 = 3$ ($m = 6$).

Lemma 15. *Suppose $f = 0$, $k = k_2 = 3$, and $m = 6$. Also, suppose that there exist $(s_1, t_1), (s_2, t_2) \in K_2$ such that $\bar{t_1}$ is free, $\beta(s_1) = -\beta(t_1)$, and $\beta(s_2) = -\beta(s_1)$. Then, there exists BiDPC$[K|G, F]$.*

Proof. We build chains $C[1] = \{(s_1, \bar{t_1}), t_1\}$, $C[2] = \{(s_2, u_2), (\bar{u_2}, t_2)\}$, and $C[3] = \{(s_3, u_3), (\bar{u_3}, t_3)\}$. First, select a free edge $u_3\bar{u_3}$ such that $u_3 \in V_0$ and s_3u_3 forms an edge. Then, find $\mathcal{P}_0$=1-BiDPC$[\{(\bar{t_1}, s_2)\}|G_0, \{s_3, u_3\} \cup F']$, where $F' = \{s_1\bar{t_3}\} \cap E$. Let $(\bar{t_1}, P_1, s_1, u_2, P_2, s_2)$ be its unique path. Finally, merge the three paths $(\bar{t_1}, P_1, s_1)$, (u_2, P_2, s_2), and (s_3, u_3) and $\mathcal{P}_1$=BiDPC $[\mathcal{K}(R_1)|G_1, \mathcal{F}(R_1)]$ using the linking edges, where $R = \bigcup_{i=1}^{3} C[i]$ (see Fig. 3b).

The Merging Lemma's conditions (a), (b), (c), and (d) can be verified with ease. There exists $\mathcal{P}_0$ by the induction hypothesis. Since $\beta((\bar{t_1}, s_2)) = 0$ and $\beta((s_3, u_3)) = 0$, $\mathcal{P}_0$ is a $\bar{t_1}$-s_2 Hamiltonian path of $G_0 \setminus \{s_3, u_3\}$. Hence, three paths $(\bar{t_1}, P_1, s_1)$, (u_2, P_2, s_2), and (s_3, u_3) form BiDPC$[\mathcal{K}(R_0)|G_0, F_0 \cup \mathcal{F}(R_0)]$. By the induction hypothesis, there exists $\mathcal{P}_1$, which is a 1-fault 2-BiDPC if $\bar{u_2} \neq t_2$ and is a 2-fault 1-BiDPC otherwise. Hence, condition (e) is satisfied. As a result, we have a desired BiDPC. □

Now, we consider the case (ii), where $\beta(s_1) = -\beta(t_1)$ and there exists no $(s, t) \in K_2$ such that $\beta(s) = -\beta(s_1)$. We distinguish two subcases: $k_2 \geq 2$ and $k_2 = 1$.

Suppose first that $k_2 \geq 2$. Since all the terminals in G_0 share the same color white by (A1), we have $k_2^{b,w} = k_2^b = 0$. If $k_2^w \geq 1$, by the choice of (s_1, t_1) in Step 1 of Procedure BiDPC-D, we are done. Therefore, $k_2 = k_2^{w,b}$. We observe that an edge $u\bar{u}$ is always free if $\bar{u}$ is a free vertex such that $\bar{u} \in N(t_1) \cap V_1$. From (A1) and $k_2 = k_2^{w,b}$, we obtain $f_1 + k_1 \geq 1$. The following Procedure BiDPC-E and Lemma 16 below handle this case; we will use the procedure again in a later case.

Procedure BiDPC-E(G, K, F) (See Fig. 3c.)
/* Builds chains $C[i] = \{(s_i, u_i), (\bar{u_i}, t_i)\}$ for $1 \leq i \leq k_2$, and $C[j] = \{(s_j, t_j)\}$ for $j > k_2$. $R = \bigcup_{i=1}^{k} C[i]$. */

1. Select a free edge $u_1\bar{u_1}$ such that $\bar{u_1}t_1$ is a fault-free edge in G_1.
2. Select $k_2 - 2$ free edges $u_i\bar{u_i}$ for $3 \leq i \leq k_2$ such that $u_i \in V_0$ and $\beta(u_i) = -\beta(s_i)$.
3. Find $\mathcal{P}_1$=$(f_1 + 1)$-fault $(k-1)$-BiDPC$[K_1 \cup \{(\bar{u_i}, t_i) : 3 \leq i \leq k_2\} \cup \{(t_1, t_2)\} | G_1, F_1 \cup \{\bar{u_1}\}]$. Let $P = (t_1, \bar{u_2}, P_u, t_2)$ be $\mathcal{P}_1$'s t_1-t_2 path.
4. Find $\mathcal{P}_0$=k_2-BiDPC$[\mathcal{K}(R_0) | G_0, \emptyset]$. Merge $\mathcal{P}_0$, $\mathcal{P}_1 \backslash P$, and two paths $(\bar{u_1}, t_1)$ and $(\bar{u_2}, P_u, t_2)$ using the linking edges.

Lemma 16. *Suppose $f + 2k \leq m$, $f_0 + f_2 + k_0 = 0$, $k_2 \geq 2$, and $f_1 + k_1 \geq 1$. Suppose that there exist $(s_1, t_1), (s_2, t_2) \in K_2$ such that $\beta(s_1) = -\beta(t_1)$, $\beta(s_2) = \beta(s_1)$, and $\beta(t_2) = \beta(t_1)$. Also, suppose that an edge $u\bar{u}$ is free if $\bar{u} \in N(t_1) \cap V_1$ is free. Then, Procedure BiDPC-E constructs BiDPC$[K|G, F]$.*

Proof. The existence of free edges in Step 1 and 2 can be shown in a way similar to the proof of Lemma 14. The Merging Lemma's conditions (a), (b), (c), and (d) can be verified with ease. Notice that $\beta(R_0) = 0$ and $\beta(F_1 \cup R_1) = \beta(G)$. Let us verify condition (e). Since $(f_1 + 1) + 2(k-1) \leq m - 1$, there exists $\mathcal{P}_1$ by the induction hypothesis. Since $\beta(K_1 \cup \{(\bar{u_i}, t_i) : 3 \leq i \leq k_2\} \cup \{(t_1, t_2), \bar{u_1}\} \cup F_1) = \beta(G)$, $\mathcal{P}_1$ contains $l = 2^{m-1} - (f^b + f^w) - \beta(G) - 1$ vertices. Let us consider a set of paths consists of all the paths in $\mathcal{P}_1 \backslash P$ and two paths $(\bar{u_1}, t_1)$ and $(\bar{u_2}, P_u, t_2)$. It contains $l+1$ vertices, and it is obviously a paired many-to-many disjoint paths joining vertex pairs of $\mathcal{K}(R_1)$. Hence, the path set is BiDPC$[\mathcal{K}(R_1) | G_1, F_1]$. There exists $\mathcal{P}_0$ by the induction hypothesis. Thus, (e) is verified. This finishes the proof. □

Suppose instead that $k_2 = 1$. Since G_0 contains no faulty vertex or terminal other than s_1, s_1 is white and $\bar{t_1}$ is free by (A1) and (A3). By Lemma 14, there remains only one case to be considered: $k_1 \geq 1$, $k_2 = k_2^{w,b} = 1$, and $\beta(G) = 0$. If $f_1 = 0$ and $k_1 = 1$, then Lemma 2 applies. Hence, it only remains to consider when $f_1 + 2k_1 \geq 3$. The following Procedure BiDPC-F and Lemma 17 consider this case.

Procedure BiDPC-F(G, K, F) (See Fig. 3d.)
/* Builds chains $C[1] = \{(s_1, \bar{t_1}), t_1\}$, $C[j] = \{(s_j, u), (\bar{u}, \bar{v}), (v, t_j)\}$ for some $j \geq 2$, and $C[i] = \{(s_i, t_i)\}$ for $2 \leq i \neq j \leq k$. $R = \bigcup_{i=1}^{k} C[i]$. */

1. Find $\mathcal{P}_1$=f_1-fault $(k-1)$-BiDPC$[K_1 | G_1, F_1]$. Let $P = (s_j, P_s, u, t_1, v, P_t, t_j)$ be the $\mathcal{P}_1$'s path containing t_1.

2. Merge $\mathcal{P}_1 \setminus P$, paths (s_j, P_s, u), (t_1), (v, P_t, t_j), linking edges $u\bar{u}$, $v\bar{v}$, and $\bar{t_1}t_1$, and $\mathcal{P}_0$=2-BiDPC$[\{(\bar{u}, \bar{v}), (s_1, \bar{t_1})\}|G_0, \emptyset]$.

Lemma 17. *Suppose $f + 2k \leq m$, $f_0 + f_2 + k_0 = 0$, $k \geq 2$, $k_2 = 1$, and $f_1 + 2k_1 \geq 3$. Also, suppose that there exists $(s_1, t_1) \in K_2^{w,b}$ such that $\bar{t_1} \neq s_1$. Then, Procedure BiDPC-F constructs BiDPC$[K|G, F]$.*

Proof. There exists $\mathcal{P}_1$ by the induction hypothesis. Notice that $f_1 + 2(k-1) < m$. We claim that $\mathcal{P}_1$ covers t_1. Notice that $\beta((s_1, t_1)) = 0$. Since $\beta(G) = 0$ and $\beta(F \cup K \setminus (s_1, t_1)) = \beta(F \cup K)$, the path set $\mathcal{P}_1$ is a disjoint path cover of $G_1 \setminus F_1$. Hence, we have the claim. Notice that $\mathcal{P}_1 \setminus P$ and the three paths (s_j, P_s, u), (t_1), and (v, P_t, t_j) form a disjoint path cover of $G_1 \setminus F_1$. Since $m \geq 5$, the induction hypothesis guarantees the existence of $\mathcal{P}_0$. Verifying the conditions of the Merging Lemma can be easily done using these facts. □

Case 3.2. $k \geq 2$ and $\bar{t_i}$ is a terminal for any $(s_i, t_i) \in K_2$.

We claim $k_2 \geq 2$. Suppose to the contrary that $k_2 = 1$ and $\bar{t_1}$ is not free. The only possibility is that $\bar{t_1} = s_1$, which contradicts (A3). Hence, we have the claim. We observe that for any t_i such that $t_i \in V_1^w$ and $(s_i, t_i) \in K_2$, $\bar{t_i}$ is a terminal. Hence, $k_2^w + k_2^{b,w} \leq k_2^b + k_2^{b,w}$. A symmetric reasoning reveals that $k_2^b + k_2^{w,b} \leq k_2^w + k_2^{w,b}$. Therefore, $k_2^b = k_2^w$.

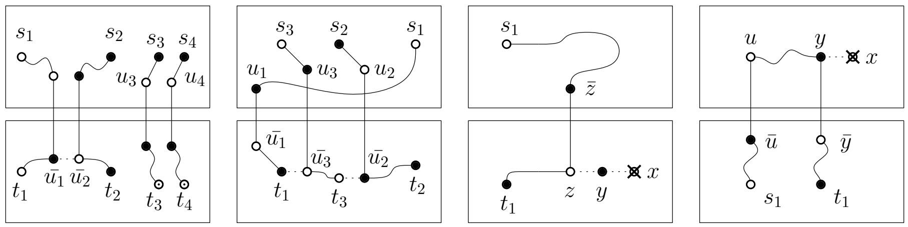
Suppose $f_1 + k_1 = 0$ and $k_2 = 2$. Since $k_2^b = k_2^w$, it follows $k_2^{w,b} = k_2^{b,w} = 1$ or $k_2^w = k_2^b = 1$ by (A1). Therefore, Lemma 2 applies. In what follows, we assume that $f_1 + k_1 \geq 1$ or $k_2 \geq 3$.

Suppose $k_2^{b,w} \geq 2$ or $k_2^{w,b} \geq 2$. Then, we rearrange source-sink pairs in K_2 so that $(s_1, t_1), (s_2, t_2) \in K_2^{w,b}$ or $(s_1, t_1), (s_2, t_2) \in K_2^{b,w}$. Notice that for any $(s, t) \in K_2$, an edge $u\bar{u}$ is free if $\bar{u}$ is free, where $\bar{u} \in N(t) \cap V_1$. Hence, if $f_1 + k_1 \geq 1$, Procedure BiDPC-E constructs a desired BiDPC. Let us consider a modified version of Procedure BiDPC-E that selects u_i from $N(s_i) \cap V_0$ for $3 \leq i \leq k_2$ at Step 2. This modified version can be used to construct a desired BiDPC when $f_1 + k_1 = 0$ and $k_2 \geq 3$.

Lemma 18. *Suppose $f + 2k \leq m$, $f_0 + f_2 + k_0 = 0$, $f_1 + k_1 = 0$, and $k_2 \geq 3$. Suppose that there exist $(s_1, t_1), (s_2, t_2) \in K_2$ such that $\beta(s_1) = -\beta(t_1)$, $\beta(s_2) = \beta(s_1)$, and $\beta(t_2) = \beta(t_1)$. Also, suppose that $u\bar{u}$ is free if $\bar{u} \in N(t_1) \cap V_1$ is free. Then, the modified Procedure BiDPC-E constructs BiDPC$[K|G, F]$.*

Proof. The proof is similar to that of Lemma 16. However, the existence of $\mathcal{P}_0$ relies on Lemma 3 instead of the induction hypothesis. Notice that if $k_2 = 3$, s_3u_3 and either s_1u_1 or s_2u_2 form two edges in G_0. □

Suppose instead that $k_2^{b,w} \leq 1$ and $k_2^{w,b} \leq 1$. We claim that we have $k_2^{b,w}, k_2^{w,b} \geq 1$ or $k_2^b, k_2^w \geq 1$. If $k_2^{b,w} = k_2^{w,b} = 1$, we are done. If $k_2^{b,w} + k_2^{w,b} \leq 1$, then $k_2^w = k_2^b \geq 1$ follows from $k_2 \geq 2$. Thus, we have the claim. By the claim, we can assume without loss of generality that there exist $(s_1, t_1), (s_2, t_2) \in K_2$ such that $\beta(s_1) = -\beta(s_2)$ and $\beta(t_1) = -\beta(t_2)$. The following Procedure BiDPC-G and Lemma 19 consider the case when $f_1 + k_1 \geq 1$ or $k_2 \geq 4$. The proof for the case when $f_1 + k_1 = 0$ and $k_2 = 3$ is given in Lemma 20 below.



(a) BiDPC-G (b) Lemma 20 (c) BiDPC-H (d) BiDPC-I

Figure 4: Illustrations of the constructions in Section 4.

Procedure BiDPC-G(G, K, F) (See Fig. 4a.)
/* Build chains $C[i] = \{(s_i, u_i), (\bar{u_i}, t_i)\}$ for $1 \leq i \leq k_2$ and $C[j] = \{(s_j, t_j)\}$ for $j > k_2$. $R = \bigcup_{i=1}^{k} C[i]$. */

1. Select $k_2 - 2$ free edges $u_i\bar{u_i}$, $3 \leq i \leq k_2$, such that $u_i \in V_0$ and $\beta(s_i) = -\beta(u_i)$. Let s_3u_3 and s_4u_4 form edges if $k_2 \geq 4$.
2. Find $\mathcal{P}_1 = (k-1)$-BiDPC$[K_1 \cup \{(\bar{u_i}, t_i) : 3 \leq i \leq k_2\} \cup \{(t_1, t_2)\} | G_1, F_1 \cup F']$, where $F' = \{t_1t_2\} \cap E$. Let $P = (t_1, P_1, \bar{u_1}, \bar{u_2}, P_2, t_2)$ be its t_1-t_2 path.
3. Find k_2-BiDPC$[\mathcal{K}(R_0)|G_0, \emptyset]$ and merge it with $\mathcal{P}_1 \setminus P$ and two paths $(t_1, P_1, \bar{u_1})$ and $(\bar{u_2}, P_2, t_2)$ using the linking edges.

Lemma 19. *Suppose $f + 2k \leq m$, $f_0 + f_2 + k_0 = 0$, $k_2 \geq 2$, and $f_1 + k_1 \geq 1$ or $k_2 \geq 4$. Also, suppose that $\bar{t_i}$ is not free for each $(s_i, t_i) \in K_2$ and there exist $(s_1, t_1), (s_2, t_2) \in K_2$ such that $\beta(s_1) = -\beta(s_2)$ and $\beta(t_1) = -\beta(t_2)$. Then, Procedure BiDPC-G construct BiDPC$[K|G, F]$.*

Proof. The proof is similar to that of Lemma 16. Notice that, by the selection of F', the length of the path P is greater than one. Since P is a path of odd length, P contains at least four vertices. Hence, there exist an edge $\bar{u_1}\bar{u_2}$ on P such that both $u_1\bar{u_1}$ and $u_2\bar{u_2}$ are free. □

Lemma 20. *Suppose $f = 0$, $k = k_2 = 3$, and $m \geq 6$. Also, suppose that $\bar{t_i}$ is not free for each $(s_i, t_i) \in K_2$. Then, there exists BiDPC$[K|G, F]$.*

Proof. It suffices to consider that $s_i \notin N(t_i)$ for $1 \leq i \leq 3$, otherwise Lemma 3 applies. Since $k_2^w = k_2^b$, it follows $k_2^b = k_2^w = k_2^o = 1$ or $k_2^o = 3$. However, as will be shown later, $k_2^o = 3$ does not happen. Hence, without loss of generality, we can assume $(s_1, t_1) \in K_2^o$. Also, we can rearrange source-sink pairs so that $\bar{t_1} = s_3$. Then, it follows $\bar{t_2} = s_1$ and $\bar{t_3} = s_2$. (See Fig. 4b.) We claim $\beta(t_1) = \beta(t_2)$. Suppose $k_2^b = k_2^w = k_2^o = 1$. From $(s_1, t_1) \in K_2^o$, it follows $\beta(s_2) = \beta(t_2)$ and $\beta(s_3) = \beta(t_3)$. From this and two conditions $\bar{t_1} = s_3$ and $\bar{t_3} = s_2$, we obtain $\beta(t_1) = -\beta(s_3) = -\beta(t_3) = \beta(s_2) = \beta(t_2)$. Suppose $k_2^o = 3$. Then, we obtain $\beta(t_1) = -\beta(s_3) = \beta(t_3) = -\beta(s_2) = \beta(t_2)$. From this, it follows $k_2 = k_2^{b,w}$ or $k_2 = k_2^{w,b}$, which violates (A1). Thus, we have the claim. We use chains $C[i] = \{(s_i, u_i), (\bar{u_i}, t_i)\}$ for $1 \leq i \leq 3$. There exists a free vertex $\bar{u_1} \in N(t_1) \cap V_1$.

Let $F' = \{t_1t_3, t_3t_2\} \cap E$. There exists $\mathcal{P}_1$=1-BiDPC$[\{(t_1, t_2)\}|G_1, \{\bar{u_1}\} \cup F']$ by the induction hypothesis. Let $P = (t_1, \bar{u_3}, P_3, t_3, \bar{u_2}, P_2, t_2)$ be $\mathcal{P}_1$'s unique path. Since $\beta(t_1) = \beta(t_2)$, the path (u_1, P) is a Hamiltonian path of G_1, so three paths $(\bar{u_1}, t_1)$, $(\bar{u_3}, P_3, t_3)$, $(\bar{u_2}, P_2, t_2)$ form BiDPC$[\mathcal{K}(R_1)|G_1, \emptyset]$, where $R = \bigcup_{i=1}^{3} C[i]$. By selection of F', it is guaranteed that $\bar{u_2}$ and $\bar{u_3}$ are free and their mates are free. Since s_3u_3 and s_2u_2 form two edges in G_0, there exists 3-BiDPC$[\mathcal{K}(R_0)|G_0, \emptyset]$ by Lemma 3. Since $F_i \cup \mathcal{F}(R_i) = \emptyset$ for $i \in \{0, 1\}$, the condition (e) of the Merging Lemma is verified, and other conditions can be verified easily. □

Case 3.3. $k = 1$.

We recall that $f = f_1$. By (A1), we have $k = k_2 = 1$ and s_1 is white. Due to Lemma 4, we need to consider only the case where $f_1^b + f_1^w \geq 2$. For a faulty vertex $x \in F_1$, we utilize a t_1-x path in G_1. Details are given in the following Procedure BiDPC-H.

Procedure BiDPC-H(G, K, F) (See Fig. 4c.)
/* Builds a chain $C[1] = \{(s_1, \bar{y}), (y, t_1)\}$ or $\{(s_1, \bar{z}), (z, t_1)\}$. $R = C[1]$. */

1. Select a faulty vertex $x \in F_1$.
2. Find 1-BiDPC$[\{(t_1, x)\}|G_1, F_1 \setminus x]$. Let $P = (t_1, P_z, z, y, x)$ be its unique path.
3. If $x \in F_1^b$, then merge (t_1, P_z, z, y) and an s_1-$\bar{y}$ Hamiltonian path of G_0 using $y\bar{y}$. If $x \in F_1^w$, merge (t_1, P_z, z) and an s_1-$\bar{z}$ Hamiltonian path of G_0 using $z\bar{z}$.

Lemma 21. *Suppose $f + 2k \leq m$, $m \geq 4$, $f_0 + f_2 + k_0 = 0$, $f_1^b + f_1^w \geq 1$, and $k = k_2 = 1$. Then, Procedure BiDPC-H constructs BiDPC[$K|G, F$].*

Proof. Notice that P contains $l = 2^{m-1} - (f^b + f^w - 1) - \beta(\{(t_1, x)\} \cup (F \setminus x))$ vertices. Suppose first that $x \in F_1^b$. Then, we have $\beta(x) = -1$, so $\beta(F \setminus x) = \beta(F) + 1$. If $\beta(t_1) = -1$, then $l = 2^{m-1} - (f^b + f^w) - \beta(F) + 1$. Since $\beta(F) = \beta(G)$, the path (t_1, P_z, z, y) contains $2^{m-1} - (f^b + f^w) - \beta(G)$ vertices. Therefore, we can merge the path with an s_1-$\bar{y}$ Hamiltonian path of G_0 in order to obtain a desired BiDPC. If $\beta(t_1) = 1$, then $l = 2^{m-1} - (f^b + f^w) - \beta(F)$. Since $\beta(F) = \beta(G) - 1$, the path (t_1, P_z, z, y) contains $2^{m-1} - (f^b + f^w) - \beta(G)$ vertices. Again, we can merge the path with an s_1-$\bar{y}$ Hamiltonian path of G_0 to obtain a desired BiDPC. The proof for $x \in F_1^w$ is similar, and therefore is omitted here. □

Case 4. For some $p \in \{0, 1\}$, $k_p = k_2 = 0$, $f_{1-p} + 2k_{1-p} \leq m - 1$, and $f_p = m - 2$.

Without loss of generality, we can assume that $k_0 = k_2 = 0$, $f_1 + 2k_1 \leq m - 1$, and $f_0 = m - 2$. Thus, we have $k = k_1 = 1$ and $f = f_0 = m - 2$. By (A1) and (A2), we must have $f_0^w \geq 1$ and $k = k_1^o$. Without loss of generality, we can assume that s_1 is white and t_1 is black. We use $(m - 3)$-fault 1-bicoverability of G_0.

Procedure BiDPC-I(G, K, F) (See Fig. 4d.)
/* Builds a chain $C[1] = \{(s_1, \bar{u}), (u, y), (\bar{y}, t_1)\}$ if $\bar{y} \neq s_1$ and $\{s_1, (\bar{s_1}, u), (\bar{u}, t_1)\}$ otherwise. */

1. Select a free edge $u\bar{u}$ such that $u \in V_0^w$.
2. Find $(m-3)$-fault 1-BiDPC$[\{(u,x)\}|G_0, F_0 \setminus x]$ for an arbitrary $x \in F_0^w$. Let $P = (u, P_y, y, x)$ be its unique path.
3. Merge a path (u, P_y, y), linking edges $u\bar{u}$ and $y\bar{y}$, and the following:
   (a) 2-BiDPC$[\{(s_1, \bar{u}), (\bar{y}, t_1)\}|G_1, \emptyset]$ if $\bar{y} \neq s_1$.
   (b) 1-BiDPC$[\{(t_1, \bar{u})\}|G_1, \{s_1\}]$ and the path (s_1) if $\bar{y} = s_1$.

Lemma 22. *Suppose $f + 2k \leq m$, $m \geq 4$, $k = k_1^o = 1$, $f = f_0 = m-2$, and $f_0^w \geq 1$. Also, suppose that s_1 is white and t_1 is black. Then, Procedure BiDPC-I constructs BiDPC[$K|G,F$].*

Proof. The existence of $u\bar{u}$ at Step 1 is obvious. At Step 2, the existence of the 1-BiDPC is guaranteed by the induction hypothesis. Let us show that existence of BiDPCs at Step 3. Notice that $F_1 = \emptyset$. When $\bar{y} \neq s_1$, since $\beta((s_1, \bar{u})) = \beta((\bar{y}, t_1)) = 0$, Lemma 2 applies. When $\bar{y} = s_1$, since $\beta((t_1, \bar{u})) = -1$ and $\beta(s_1) = 1$, Lemma 4 applies. The proof of the correctness of our construction is similar to that of Lemma 21. The path (u, P_y, y) contains $2^{m-1} - (f^b + f^w) - \beta(G)$ vertices. Notice that $\beta((F \setminus x) \cup \{(u,x)\}) = \beta(F) = \beta(G)$. Thus, merging the path (u, P_y, y) and the 2-BiDPC of Step 3(a) results in BiDPC$[K|G,F]$. Similarly, merging the path (u, P_y, y), the 1-BiDPC of Step 3(b), and a path (s_1) results in BiDPC$[K|G,F]$. □

5. Proof of Theorem 1 for $f^b \geq 2$

Contrary to the previous section, the case when $h^w(G) \leq 0$ can be handled easily. So the major portion of this section is devoted to the case when $h^w(G) > 0$. When $h^w(G) > 0$ and Procedures BiDPC-C and BiDPC-C1 are not applicable, we solve a relaxed problem where the solution may contain additional closed chains (that are not joining source-sink pairs) and the condition (d) of the Merging Lemma is ignored. Then, we postprocess the relaxed solution in order to obtain chains satisfying every condition of the Merging Lemma.

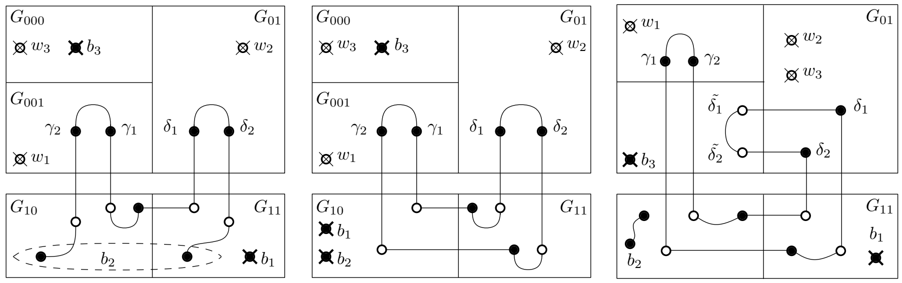
By the same way as we got (A1), with no loss of generality, we can assume that $f_0^b \geq 1$ and $f_1^b \geq 1$. We can further assume that $h^b(G) \leq 0$.

Lemma 23. *Suppose $f + 2k \leq m$, $m \geq 4$, $f_0^b \geq 1$, $f_1^b \geq 1$, and $h^b(G) \leq 0$. Then, Procedure BiDPC-A, BiDPC-B, BiDPC-C, or BiDPC-C1 can construct BiDPC[$K|G,F$] unless*

$$f_0^w \geq 3,\ f_0^b \geq 1,\ f_1^b \geq 1,\ h^w(G) \geq 2,\ \text{and } m \geq 8. \tag{10}$$

Proof. From $f + 2k \leq m$ and $f_0, f_1 \geq 1$, we obtain $f_p + 2(k_p + k_2) \leq m - 1$ for each $p \in \{0, 1\}$. Therefore, Lemma 7 or 8 applies if $h^w(G) \leq 0$.

We claim that if $h^w(G) > 0$, then Lemma 9 or 10 applies unless $h^w(G) \geq 2$ and $f_0^w \geq 3$. Suppose $h^w(G) = 1$. From (2), we obtain $h^w(G) = 1 \leq f_0^w + k_0^w - f_0^b$. Since $f_0^b \geq 1$, it follows $3 \leq f_0^w + k_0^w + f_0^b$ and $2h^w(G) = 2 \leq f_0 + f_2 + 2k_0 - 1$. Hence, if $k_0 + k_2 \geq 1$, Lemma 9 applies. If $k_0 + k_2 = 0$, we have $k_1 = k \geq 1$. Since $f_1^b \geq 1$, it follows $f_1 + f_2 + 2k_1 - 1 \geq 2 = 2h^w(G)$. Hence, Lemma 10



(a) First canonical form (b) First canonical form (2) (c) Second canonical form

Figure 5: Illustrations of constructions for the 6-tuple.

applies. Suppose instead that $f_0^w \leq 2$. Then, $f_0^w \leq 3f_0^b + 2\beta(G) - 1$. From Lemma 11(a), we obtain $2h^w(G) \leq f_0 + f_2 + 2k_0 - 1$. If $k_0 + k_2 \geq 1$, then Lemma 9 applies. If $k_0 + k_2 = 0$, then $h^w(G) = f_0^w - f_0^b - \beta(G) \leq 1$, so we are done. Therefore, we have the claim.

Let $f_0^w \geq 3$, $h^w(G) \geq 2$, and $m \leq 7$. Then, the only possibility is $m = 7$, $f_0^w = 3$, $f_0^b = f_1^b = 1$, $h^w(G) = 2$, and $k = k_1^b = 1$. It is the unique subcase of $m = 7$ where the aforementioned lemmas are not applicable. To overcome this problem, we flip color of every vertex so that $f^b = 3$, $f^w = 2$, and $k = k^w = 1$, and then partition G into two subcubes G_0' and G_1' each of which contains at least one (new) black faulty vertex. This causes no problem since (i) $\beta(G)$ still remains zero, and (ii) if $h^b(G) > 0$ at this point, it suffices to exchange G_0' and G_1'. Then, one of the lemmas should be applicable, i.e., Lemma 7 or 8 applies when $h^w(G) \leq 0$ and Lemma 9 or 10 applies when $h^w(G) > 0$ ($f_0^w \leq f^w = 2$). The proof is finished. □

Hereafter, we assume (10). There exist three white faulty vertices w_1, w_2, $w_3 \in F_0^w$ and two black faulty vertices $b_1 \in F_1^b$ and $b_3 \in F_0^b$. From (3) and $h^w(G) \geq 2$, it follows $f_1^b + k_1^b \geq 2$. So, there exists $b_2 \in F_1^b \cup K_1^b$ other than b_1. The set $\{w_1, w_2, w_3, b_1, b_2, b_3\}$ will be called the *6-tuple*. We further assume the following two conditions.

(B1) Either $w_3, b_3 \in V_{000}$, $w_1 \in V_{001}$, and $w_2 \in V_{01}$ (see Fig. 5a)
or $w_1 \in V_{000}$, $b_3 \in V_{001}$, $w_2, w_3 \in V_{01}$, $f_{00}^w = 1$, and $f_{01}^b = 0$ (see Fig. 5c).

(B2) If $\{b_1, b_2\} \subset F_p \cup K_p$ for some $p \in \{10, 11\}$, then $F_1^b \cup K_1^b = F_p^b \cup K_p^b$.

We show that we do not lose any generality by our assumptions. It is safe to assume that $w_1, b_3 \in V_{00}$ and $w_2 \in V_{01}$. We distinguish two cases according to whether $w_3 \in V_{00}$ or not. Suppose for the first case that $w_3 \in V_{00}$. Without loss of generality, we can assume that $w_3, b_3 \in V_{000}$ and $w_1 \in V_{001}$. We will call this arrangement the *first canonical form*. Suppose for the second case that $w_3 \in V_{01}$. Without loss of generality, we can assume that $w_1 \in V_{000}$ and $b_3 \in V_{001}$. We will call this arrangement the *second canonical form*. If we have the second canonical form and $f_{00}^w \geq 2$, then there exists $w_4 \in F_{00}^w$ such that $w_4 \neq w_1$. By renaming w_4 as w_3 (and vise versa), we can obtain the first canonical form. We have a similar argument when $f^b_{01} \geq 1$. In this way, (B1) is satisfied. For (B2), we avoid choosing b_2 from $F^b_p \cup K^b_p$ if possible, where $p \in \{10, 11\}$ is such that $b_1 \in V_p$.

Now, we define our relaxed problem. For a set of units X, let $\text{size}(X) = |\mathcal{F}(X)| + 2|\mathcal{K}(X)|$. Let us define the notion of *regular refinement*. For a set of units $X = \{x_1, x_2, \ldots, x_n\}$ with $\beta(X) \geq 0$, and a set of binary strings $\mathbb{Q}$, we say a set of units Y is a $\mathbb{Q}$-*regular refinement* of X if Y can be partitioned into $\mathcal{F}(X)$ and pairwise disjoint free and simple chains $Y[1], Y[2], \ldots, Y[n']$ for some $n' \geq n$ such that

- (a′) $Y[i] \succ x_i$ if $x_i \in \mathcal{K}(X)$, $Y[i] = \emptyset$ if $x_i \in \mathcal{F}(X)$, and $Y[i]$ is a closed chain if $i > n$;
- (b′) $\bigcup_{i=1}^{n'} Y[i]$ is $\mathbb{Q}$-separated;
- (c′) for each $p \in \mathbb{Q}$, $0 \leq \beta(Y_p) \leq \beta(X)$;
- (e′) for each $p \in \mathbb{Q}$, $\text{size}(Y_p) \leq \text{size}(X)$; and
- (f′) if $\mathcal{K}(X) \neq \emptyset$, then for each $p \in \mathbb{Q}$, either $\bigcup_{i=1}^{n} \mathcal{K}(Y[i]_p) \neq \emptyset$ or $\text{size}(Y_p) \leq \text{size}(X) - |\mathcal{K}(X)|$.

We observe similarities between the conditions of a regular refinement and those of the Merging Lemma. Here, condition (d′) is intentionally omitted. For a set of units X, we say X's $\mathbb{Q}$-regular refinement Y is *strong* if conditions (e′) and (f′) hold true even if the term $\text{size}(X)$ is replaced with $\text{size}(X) - l(p)$. Our first goal is to find a strong regular refinement of $F \cup K$.

5.1. Phase 1: Building regular refinements

Let us sketch the construction of regular refinements. Hereafter, $\mathbb{P}$ denotes the set $\{000, 001, 01, 10, 11\}$. We will partition $F \cup K$ into cells of non-negative balances, then find $\mathbb{P}$-regular refinements of all the cells. To find regular refinements, we use divide-and-conquer approaches. First, we proceed with subset-by-subset. Notice that $Y[1] \cup Y[2]$ is a $\mathbb{P}$-regular refinement of $X[1] \cup X[2]$ for disjoint unit sets $X[1]$ and $X[2]$ and their respective $\mathbb{P}$-regular refinements $Y[1]$ and $Y[2]$ that are disjoint. Second, we proceed with subcube-by-subcube. For example, if a unit set X has a $\{0, 1\}$-regular refinement Y, where Y_0 and Y_1 respectively have a $\{00, 01\}$-regular refinement Z contained in G_0 and a $\{10, 11\}$-regular refinement Z' contained in G_1, then $Z \cup Z'$ is a $\{00, 01, 10, 11\}$-regular refinement of X.

We begin by partitioning $F \cup K$ into cells with non-negative balances. From (2) and (3), we can observe that $F \cup K$ can be partitioned as follows.

- Type-1: $\beta(G)$ singletons each containing a white unit in G_0.
- Type-2: $f^o + k^o - k^o_2$ singletons each containing a unit of balance zero.
- Type-3: $f^b_0 + k^b_0 - 1$ sets $\{w, b\}$, where w and b respectively are a white and a black unit in G_0, and $f^w_1 + k^w_1$ sets $\{w', b'\}$, where w' and b' respectively are a white and a black unit in G_1.
- Type-4: $h^w(G) - 2$ sets $\{w, b\}$, where $w \in F^w_0 \cup K^w_0$ and $b \in F^b_1 \cup K^b_1$.

Type-5: The 6-tuple $\{w_1, w_2, w_3, b_1, b_2, b_3\}$.
Type-6: k_2^b sets $\{(s,t), w\}$, where $(s,t) \in K_2^b$ and $w \in F_0^w \cup K_0^w$.
Type-7: k_2^w sets $\{(s,t), b\}$, where $(s,t) \in K_2^w$ and $b \in F_1^b \cup K_1^b$.
Type-8: $k_2^{b,w}$ sets $\{(s,t), w, b\}$, where $(s,t) \in K_2^{b,w}$, $w \in F_0^w \cup K_0^w$, and $b \in F_1^b \cup K_1^b$.
Type-9: $k_2^{w,b}$ singletons $\{(s,t)\}$, where $(s,t) \in K_2^{w,b}$.

We will construct a strong $\mathbb{P}$-regular refinement of $F \cup K$ as follows. Initially, we let $Z = F \cup K$. For each cell of the partition, say X, we find its $\mathbb{P}$-regular refinement, say Y. Especially for the 6-tuple X, we find X's strong $\mathbb{P}$-regular refinement, say Y^*. Whenever a cell X is refined to its $\mathbb{P}$-regular refinement Y, we update $Z = (Z \setminus X) \cup Y$. Therefore, after every cell is refined, Z becomes a strong $\mathbb{P}$-regular refinement of $F \cup K$.

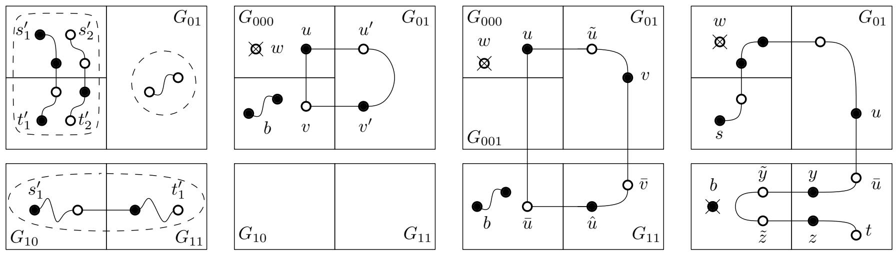
We present five procedures. Procedures Regular-A and B consider sets of types-1, 2, and 3. Procedures Regular-C and E respectively consider sets of type-4; and types-6, 7, 8, and 9. Procedure Regular-D constructs the set Y^* that is a strong $\mathbb{P}$-regular refinement of the 6-tuple.

Before we proceed to the detailed construction, let us give some notation. For a vertex u, $\tilde{u}$ denotes the u's neighbor such that u and $\tilde{u}$ differ in the second bit; $\hat{u}$ denotes the mate of $\tilde{u}$ (see Fig. 6c for an illustration). A *depth-1 free corner* is a vertex u such that $u\bar{u}$, $\bar{u}\hat{u}$, $\hat{u}\tilde{u}$, and $\tilde{u}u$ are all free with respect to Z. A *depth-2 free corner* is a vertex u such that $u\dot{u}$, $\dot{u}\tilde{\dot{u}}$, $\tilde{\dot{u}}\tilde{u}$, and $\tilde{u}u$ are all free with respect to Z, where $\dot{u}$ is the u's neighbor such that u and $\dot{u}$ differ in the third bit and $\tilde{\dot{u}}$ is the common neighbor of $\tilde{u}$ and $\dot{u}$ other than u. We claim that there exist enough number of free corners and edges that are free with respect to Z for our construction. The existence proof is deferred to Lemma 24.

A type-3 set $X = \{w, b\}$, where w is a white unit and b is a black unit, is *bad* if each of G_{p0} and G_{p1} contains either of w or b for some $p \in \{0, 1, 00\}$; otherwise it is *good*. Given a type-1 or 2, or a good type-3 set X contained in G_p (or possibly a set of two vertex pairs with balances zero that are contained in either G_0 or G_1), Procedure Regular-A returns X's $\mathbb{P}$-regular refinement contained in G_p. It is assumed that the binary string p is a prefix of an element of $\mathbb{P}$ such that X is contained in G_p. We enumerate elements of $\mathcal{K}(X_{p2})$ as $(s'_1, t'_1), (s'_2, t'_2), \ldots, (s'_{k'}, t'_{k'})$, where k' denotes $|\mathcal{K}(X_{p2})|$. We assume that $s'_i \in V_{p0}$ for each $(s'_i, t'_i) \in \mathcal{K}(X_{p2})$.

Procedure Regular-A(G, Z, X, p) (See Fig. 6a)

0. If $X = \emptyset$, $X \subset F^o$, or $p \in \mathbb{P}$, then return X.
1. Select k' edges $u_i v_i$ that are free with respect to Z for $1 \le i \le k'$ such that $u_i \in V_{p0}$, $v_i \in V_{p1}$, and
   (a) $-h_p^b(\mathcal{K}(X), \mathcal{F}(X))$ ones are from $E_{p2}^{b,w}$ and $-h_p^w(\mathcal{K}(X), \mathcal{F}(X)) - \beta(X)$ ones are from $E_{p2}^{w,b}$.
   (b) $u_i \in V_{00r}$ if $p = 0$, where $r \in \{0, 1\}$ is such that $s'_i \in V_{00(1-r)}$.
2. Put $Y = (X \setminus \mathcal{K}(X_{p2})) \cup \{(s'_i, u_i), (v_i, t'_i) : 1 \le i \le k'\}$.
3. Put Y'=Regular-A$(G, Z \cup Y, Y_{p0}, p0)$ and Y''=Regular-A$(G, Z \cup Y, Y_{p1}, p1)$. Return $Y' \cup Y''$.



(a) Regular-A (b) Regular-B (c) Regular-C (d) Regular-E

Figure 6: Illustrations for $\mathbb{P}$-regular refinements.

The Regular-A procedure works as follows. It computes a $\{p0, p1\}$-regular refinement of X at Steps 1 and 2, and then recursively refines it to a $\mathbb{P}$-regular one at Step 3. At Step 1, the selection of free edges (with respect to Z) is very similar to that of free edges in Step 1(b) of Procedure BiDPC-A. From the construction and the proof of Lemma 7, we can deduce that Y is a $\{p0, p1\}$-regular refinement of X. It is obvious that Y is contained in G_p. For Step 0, notice that if X is an empty set, a type-2 set consists of one faulty edge, or a set contained in G_p for some $p \in \mathbb{P}$, then it is a $\mathbb{P}$-regular refinement of itself. We claim that, when Step 3 is reached, each of Y_{p0} and Y_{p1} is (i) an empty set; (ii) a type-1, 2, or a good type-3 set; or (iii) a set of two vertex pairs with balances zero. The case (iii) may happen when X is a good type-3 set such that $X \subset K_{02}$ or $X \subset K_{12}$. Notice that each of Y_{p0} and Y_{p1} contains at most two units and $0 \le \beta(Y_{p0}), \beta(Y_{p1}) \le 1$. Hence, it suffices to show that none of them is a bad type-3 set. Since Y_{01}, Y_{11}, and Y_{10} cannot be bad by definition, it suffices to check Y_{p0} when $p = 0$. If $k' \ge 1$, then (s_1', u_1) is a vertex pair between V_{000} and V_{001}, so Y_{00} cannot be bad. When $k' = 0$, we can check that Y_{00} is bad only if X is bad, too. Thus, the claim is proved. Hence, Y' and Y'' are $\mathbb{P}$-regular refinements of Y_{p0} and Y_{p1}, respectively. Notice that our usage of $Z \cup Y$ in recursive calls guarantees that chains in $Y' \cup Y''$ are free and disjoint. Also, notice that Y' and Y'' are contained in G_{p0} and G_{p1}, respectively. Therefore, $Y' \cup Y''$ is a $\mathbb{P}$-regular refinement of X contained in G_p.

The Procedure Regular-A above will be utilized as a subroutine for other procedures. We observe that for any type-1 set X, its regular refinement Y satisfies $\beta(Y_0) = 1$ and $\beta(Y_{10}) = \beta(Y_{11}) = 0$. Since cells of other types are of balance zero, we will eventually have $\beta(Z_{10}) = \beta(Z_{11}) = 0$.

Given a bad type-3 set X, Procedure Regular-B returns its $\mathbb{P}$-regular refinement Y by adding a closed chain. It is assumed that $X = \{w, b\}$, and $p \in \{0, 1, 00\}$ is such that each of G_{p0} and G_{p1} contains either w or b.

Procedure Regular-B(G, Z, X, p) (See Fig. 6b)

1. Case $h_p^w(\mathcal{K}(X), \mathcal{F}(X)) = 1$: Let u be a depth-$l(p)$ free corner contained in V_{p0}^b. If $p = 0$ and w is contained in G_r, where $r \in \{000, 001\}$, then u should be chosen from V_r^b.

2. Case $h_p^b(\mathcal{K}(X), \mathcal{F}(X)) = 1$: Let u be a depth-$l(p)$ free corner contained in V_{p0}^w. If $p = 0$ and b is contained in G_r, where $r \in \{000, 001\}$, then u should be chosen from V_r^w.
3. Put $Y = X \cup \{u, v, (v', u')\}$, where v, v', and u' respectively are the unique members of $N(u) \cap V_{p1}$, $N(v) \cap V_{p'(1-q)}$, and $N(u) \cap V_{p'(1-q)}$, where p' denotes the prefix of p of length $l(p) - 1$ and q denotes p's last bit.
4. (a) Case $p = 00$: Return Y.
   (b) Case $p = 0$: Put Y'=Regular-A$(G, Z \cup Y, Y_1, 1)$ and Y''=Regular-A$(G, Z \cup Y, Y_{00}, 00)$. Return $Y_{01} \cup Y' \cup Y''$.
   (c) Case $p = 1$: Put Y'=Regular-A$(G, Z \cup Y, Y_0, 0)$. Return $Y_1 \cup Y'$.

The Regular-B procedure works as follows. Steps 1, 2, and 3 add a closed chain $\{u, v, (v', u')\}$. When Step 4 is reached, it is easily seen that Y is a $\{p0, p1, p'(1-q)\}$-regular refinement of X. Notice that Y_{p0} and Y_{p1} are good type-3 sets and $Y_{p'(1-q)}$ is a type-2 set. Step 4 further refines Y into a $\mathbb{P}$-regular one.

The Procedure Regular-C returns a $\mathbb{P}$-regular refinement of a type-4 cell X by adding a closed chain that contributes one black unit to G_0 and one white unit to G_1. We assume that $X = \{w, b\}$, $w \in F_0^w \cup K_0^w$, and $b \in F_1^b \cup K_1^b$. Without loss of generality, we can assume that if $w \in K_{02}$ and $w = (s, t)$, then $s \in V_{00}$ and that if $b \in K_{12}$ and $b = (s', t')$, then $s' \in V_{11}$.

Procedure Regular-C(G, Z, X) (See Fig. 6c)

1. Select two depth-1 free corners $u \in V_{00}^b$ and $v \in V_{01}^b$ such that $u \in V_{000}^b$ if w is contained in G_{000}, and $u \in V_{001}^b$ otherwise.
2. (a) Case w is in G_{00}: Put $Y = \{(v, \tilde{u}), u, w\}$.
   (b) Case w is in G_{01}: Put $Y = \{w, v, (\tilde{v}, u)\}$.
   (c) Case $w \in K_{02}$: Put $Y = \{(s, u), (v, t)\}$, where $w = (s, t)$.
3. (a) Case b is in G_{10}: Put $Y = Y \cup \{b, \bar{u}, (\hat{u}, \bar{v})\}$.
   (b) Case b is in G_{11}: Put $Y = Y \cup \{(\bar{u}, \hat{v}), \bar{v}, b\}$.
   (c) Case $b \in K_{12}$ and $w \notin K_{02}$: Put $Y = Y \cup \{(s', \bar{v}), (\bar{u}, t')\}$, where $b = (s', t')$.
   (d) Case $b \in K_{12}$ and $w \in K_{02}$: Put $Y = Y \cup \{b, (\bar{u}, \bar{v})\}$.
4. Put Y'=Regular-A$(G, Z \cup Y, Y_{00}, 00)$. Put Y''=Regular-A$(G, Z \cup Y, Y_1, 1)$. Return $Y_{01} \cup Y' \cup Y''$.

The selection of u at Step 1 prevents Y_{00} from being a bad type-3 set. Notice that Steps 2(a), 2(b), 3(a), and 3(b) construct a v-u chain in G_0 and a $\bar{u}$-$\bar{v}$ chain in G_1. When Step 4 is reached, Y is a $\{00, 01, 1\}$-regular or $\{00, 01, 10, 11\}$-regular refinement of X. Step 4 further refines it to a $\mathbb{P}$-regular one.

The Procedure Regular-D returns a set Y^*, a strong $\mathbb{P}$-regular refinement of the 6-tuple X, using two chains each of which contributes one black unit to G_0 and one white unit to G_1. We assume that if b_2, an element of X, is a member of K_{12}, then $s \in V_{11}$ and $t \in V_{10}$, where $b_2 = (s, t)$. If $b_2 \in F_1^b \cup K_{12}^b$, then the refinement Y^* will have a single chain, which is an s-t chain if $b_2 \in K_{12}^b$ (see Fig. 5a) and a closed chain if $b_2 \in F_1^b$ (see Fig. 5b). If $b_2 \in K_{10}^b$ or $b_2 \in K_{11}^b$, then Y^* will have an s-t chain and a closed chain (see Fig. 5c).

Procedure Regular-D(G, Z, X) (See Fig. 5)

1. Select four depth-1 free corners γ_1, γ_2, δ_1, and δ_2 such that $\gamma_1, \gamma_2 \in V^b_{001}$ and $\delta_1, \delta_2 \in V^b_{01}$ for the first canonical form and $\gamma_1, \gamma_2 \in V^b_{000}$ and $\delta_1, \delta_2 \in V^b_{011}$ for the second canonical form.
2. Put C=Regular-C$(G, Z, \{w_1, b_1\})$ and C'=Regular-C$(G, Z, \{w_2, b_2\})$, where we use γ_1 and δ_1 for C, and γ_2 and δ_2 for C' instead of selecting u and v, respectively.
3. Put $Y^* = C_1 \cup C'_1 \cup D$, where $D = \{(\gamma_1, \gamma_2), (\delta_2, \delta_1)\}$ if the 6-tuple is in the first canonical form; $D = \{(\gamma_1, \gamma_2), \delta_2, (\tilde{\delta_2}, \tilde{\delta_1}), \delta_1\}$ otherwise.
4. Put $Y^* = (Y^* \setminus \mathcal{K}(C_1)) \cup \{(y, x) : (x, y) \in \mathcal{K}(C_1)\}$.
5. Return Y^*.

The Regular-D procedure works as follows. Steps 1 and 2 construct two chains C and C'. Step 3 merges the two chains into a set Y^*. At this point, Y^*_0 contains a γ_1-γ_2 chain and a δ_2-δ_1 chain, and Y^*_1 contains a $\bar{\gamma_1}$-$\bar{\delta_1}$ chain. At Step 4, in order to merge these three chains into a δ_2-γ_2 chain, we convert the $\bar{\gamma_1}$-$\bar{\delta_1}$ chain into a $\bar{\delta_1}$-$\bar{\gamma_1}$ chain by taking the reverse of ordered pairs. Now, the δ_2-γ_2 chain and a $\bar{\gamma_2}$-$\bar{\delta_2}$ chain in Y^*_1, if exists, collectively form a closed chain or we obtained an s-t chain if $b_2 = (s, t)$ and $b_2 \in K_{12}$.

It is straightforward to check that Y^* is a string $\mathbb{P}$-regular refinement of X. For example, let us think of a case when we have the first canonical form, $b_1 \in F^b_{11}$, and $b_2 \in K^b_{12}$ (see Fig. 5a). Let $b_2 = (s, t)$, where $s \in V^b_{11}$ and $t \in V^b_{10}$. The resulted set Y^* is partitioned into $Y^*_{000} = \{w_3, b_3\}$, $Y^*_{001} = \{w_1, (\gamma_1, \gamma_2)\}$, $Y^*_{01} = \{w_2, (\delta_2, \delta_1)\}$, $Y^*_{10} = \{(\bar{\gamma_2}, t), (\hat{\delta_1}, \bar{\gamma_1})\}$, and $Y^*_{11} = \{b_1, \bar{\delta_1}, (s, \bar{\delta_2})\}$. Conditions (a$'$), (b$'$), and (c$'$) can be checked easily. From size$(Y^*_{000}) = 2 < 7 - l(000)$, size$(Y^*_{001}) = 3 < 7 - l(001)$, size$(Y^*_{01}) = 3 < 7 - l(01)$, size$(Y^*_{10}) = 4 < 7 - l(10)$, and size$(Y^*_{11}) = 4 < 7 - l(11)$, we conclude that Y^* is a strong $\mathbb{P}$-regular refinement of X. Other cases can be checked similarly.

In Phase 2, it will be useful that Y^* has a chain that contains vertex pairs in both G_{10} and G_{11} unless both b_1 and b_2 are contained in either G_{10} or G_{11} (see Fig. 5). By (B2), it occurs only when all the black units in G_1 are contained in either G_{10} or G_{11}.

Given a type-6, 7, 8, or 9 set X, the following Procedure Regular-E returns a set Y that is a $\mathbb{P}$-regular refinement of X.

Procedure Regular-E(G, Z, X) (See Fig. 6d)

1. Select an edge $u\bar{u}$ that is free with respect to Z, such that $u \in V^b_{0r}$ and
   (a) Case $(s, t) \notin K^w_2$: $r \in \{0, 1\}$ is such that $s \in V_{0(1-r)}$.
   (b) Case $(s, t) \in K^w_2$: $r \in \{0, 1\}$ is such that $t \in V_{1(1-r)}$.
2. (a) Case $(s, t) \in K^{b,w}_2$, $(\bar{u}, t)$ is in G_{1p}, and b is in $G_{1(1-p)}$ for some $p \in \{0, 1\}$: Select two edges $y\tilde{y}$ and $z\tilde{z}$ that are free with respect to Z, such that $y, z \in V^b_{1p}$. Put $Y = \{b, w, (s, u), (\bar{u}, y), (\tilde{y}, \tilde{z}), (z, t)\}$.
   (b) Case otherwise: Put $Y = X_0 \cup X_1 \cup \{(s, u), (\bar{u}, t)\}$.
3. Put Y'=Regular-A$(G, Z \cup Y, Y_0, 0)$ and Y''=Regular-A$(G, Z \cup Y, Y_1, 1)$. Return $Y' \cup Y''$.

Step 1 prevents Y_0 and Y_1 from being bad with the unique exception that is handled in Step 2(a). Step 2(a) constructs a set Y that is a $\{0, 10, 11\}$-regular refinement of X such that Y_0, Y_{10}, and Y_{11} are not bad. Step 2(b) builds a set Y that is a $\{0, 1\}$-regular refinement of X such that neither Y_0 nor Y_1 is a bad type-3 set. Then, at Step 3, Y is refined to a $\mathbb{P}$-regular one.

Now, it is the turn to show that these constructions are possible.

Lemma 24. *There exist enough number of edges that are free with respect to Z and free corners for Procedures Regular-A, B, C, D, and E.*

Proof. It suffices to show that there remain free corners after all the cells are refined, which implies the abundance of inter-subcube edges that are free with respect to Z. We have $Z = \bigcup_{p\in\mathbb{P}} Z_p \cup F'$, where F' is the set of faulty edges contained in no G_p for $p \in \mathbb{P}$. By condition (e$'$), for each $p \in \mathbb{P}$, $\mathrm{size}(Z_p) \leq \mathrm{size}((F \cup K) \setminus F') - l(p) = \mathrm{size}(F \cup K) - |F'| - l(p)$. Hence, $\mathrm{size}(Z) = |\mathcal{U}(Z)| \leq 5 \cdot \mathrm{size}(F \cup K) - 5|F'| - (3 + 3 + 2 + 2 + 2) + |F'| \leq 5 \cdot \mathrm{size}(F \cup K) - 12$. Let us count depth-1 free corners in V^b_{000}. There are 2^{m-4} candidates, i.e., vertices in V^b_{000}. Faulty graph elements and terminals together block at most $\mathrm{size}(F \cup K)$ $(= |\mathcal{U}(F \cup K)| = f + 2k)$ candidates. We observe that for each $u \in \mathcal{U}(Z) \setminus \mathcal{U}(F \cup K)$, where u must be a nonterminal vertex, there exists a linking edge uv for some $v \in \{\bar{u}, \tilde{u}, \dot{u}\}$, which implies $v \in \mathcal{U}(Z)$. Since u and v together can block at most one candidate, vertices in $\mathcal{U}(Z) \setminus \mathcal{U}(F \cup K)$ additionally block at most $|\mathcal{U}(Z) \setminus \mathcal{U}(F \cup K)|/2 = 2 \cdot \mathrm{size}(F \cup K) - 6 = 2(f + 2k) - 6$ candidates. If $m \geq 9$, then $2^{m-4} - (f + 2k) - (2(f + 2k) - 6) \geq 2^{m-4} - 3m + 6 \geq 11$, so we are done. If $m = 8$, then $2^{m-4} - 3m + 6 \geq -2$. However, this bound is not tight because colors and locations of vertices are not considered. For example, let us think of the case where we have the first canonical form. It is regarded that two faulty vertices w_3 and w_1 block two candidates and γ_1, γ_2, δ_1, δ_2, and their four neighbors block four candidates. However, we can observe that they do not block any candidates (see Fig. 5a). Therefore, there remain at least four non-blocked candidates. By the same manner, we can show that there remain at least two free corners of other kinds. The proof is finished. □

Remark 1. *Concerned with the strong $\mathbb{P}$-regular refinement Z of $F \cup K$ constructed in Phase 1, it is worth noting that $\mathrm{size}(Z_p) \leq \mathrm{size}(F \cup K) - l(p) \leq m - l(p)$ for every $p \in \mathbb{P}$, and $\mathrm{size}(Z_q) \leq \mathrm{size}(F \cup K) - l(q) - |K| \leq m - l(q) - |K|$ for each $q \in \mathbb{P}$ such that no s_i-t_i chain for $1 \leq i \leq k$ has a vertex pair contained in Z_q.*

Remark 2. *Furthermore, every linking edge of Z between G_0 and G_1 joins a vertex in V^b_0 and V^w_1. In addition, for every closed chain C, there exist $p, q \in \mathbb{P}$ (depending on C) such that $Z_p \cap \mathcal{K}(C) \neq \emptyset$ and $Z_q \cap \mathcal{K}(C) \neq \emptyset$.*

5.2. Phase 2: Postprocessing

Now, we have a set Z that is a strong $\mathbb{P}$-regular refinement of $F \cup K$. We will remove the additional closed chains and let chains joining the original source-sink pairs collectively have vertex pairs in every subcube.

We denote by $C[i]$, where $1 \le i \le n$ for some $n \ge k$, a chain constructed in Phase 1. We assume that $C[i]$ is an s_i-t_i chain for $1 \le i \le k$ and is an additional closed chain for $i > k$. We denote $R = \bigcup_{i=1}^{k} C[i]$ and $L = \bigcup_{i=k+1}^{n} C[i]$, so $Z = F \cup R \cup L$. Let $\Phi_p = m - l(p) - \mathrm{size}(Z_p)$. By condition (f′), if $\mathcal{K}(R_p) = \emptyset$, then $\Phi_p \ge k$ (see Remark 1).

Although the condition $\Phi_p \ge 0$ is necessary to apply the induction hypothesis to G_p, it is stronger than what we actually need. We know that $\Phi_p \ge 0$ and $\mathcal{K}(Z_p) \ne \emptyset$ imply the existence of $\mathcal{P}_p$=BiDPC[$\mathcal{K}(Z_p)|G_p, \mathcal{F}(Z_p)$], but the converse is not always true. Let us say that the set Z is *fine* if (i) it satisfies conditions (a′), (b′), and (c′) of $\mathbb{P}$-regular refinement with respect to $F \cup K$, (ii) the existence of $\mathcal{P}_p$ is guaranteed for each $p \in \mathbb{P}$ such that $\mathcal{K}(Z_p) \ne \emptyset$, and (iii) $\mathrm{size}(Z_p) \le \mathrm{size}(F \cup K) - l(p) - |K|$ for each $p \in \mathbb{P}$ such that $\mathcal{K}(Z_p) = \emptyset$. If Z is a strong $\mathbb{P}$-regular refinement of $F \cup K$, then Z is fine by definition. In order to obtain a fine set Z with fewer additional closed chains, we use four operations: *CycleMerge*, *Stretch*, *Propagate*, and *Join*.

We use an undirected graph H for analysis, where the vertex set $V(H) = \mathbb{P}$ and pq is an edge of H if there exist some $C[i]$, $1 \le i \le n$, that passes through both G_p and G_q via vertex pairs, i.e., $\mathcal{K}(C[i]_p) \ne \emptyset$ and $\mathcal{K}(C[i]_q) \ne \emptyset$.

We remove additional closed chains by merging them into chains joining source-sink pairs. The following Operation CycleMerge repeatedly merges a closed chain into some other chain, which is possibly an additional closed chain.

Operation CycleMerge(p)

1. Repeat until no more update is possible.
   (a) Choose two chains $C[i]$ and $C[j]$ such that $1 \le i \le n$, $j > k$, $\mathcal{K}(C[i]_p) \ne \emptyset$, and $\mathcal{K}(C[j]_p) \ne \emptyset$. Let $(u, v) \in \mathcal{K}(C[i]_p)$ and $(x, y) \in \mathcal{K}(C[j]_p)$.
   (b) Update $C[i] = (C[i] \setminus (u, v)) \cup (C[j] \setminus (x, y)) \cup \{(u, y), (x, v)\}$.
   (c) Put $C[j] = \emptyset$.

The CycleMerge operation decreases the number of additional closed chains by one at every iteration of the main loop. After CycleMerge(p) is applied, if $\mathcal{K}(R_p) \ne \emptyset$ (resp. $\mathcal{K}(R_p) = \emptyset$), then there remains no (resp. at most one) additional closed chain $C[j]$ having vertex pairs in G_p. Provided $\Phi_p \ge 0$, applying CycleMerge(p) preserves the strong $\mathbb{P}$-regular refinement property and the fine set property as shown below. Also, the connected components of H remain unchanged.

Lemma 25. *Suppose that $p \in \mathbb{P}$, $\Phi_p \ge 0$, and Z is a strong $\mathbb{P}$-regular refinement of $F \cup K$. If CycleMerge(p) is applied, then (a) Z remains a strong $\mathbb{P}$-regular refinement of $F \cup K$ and (b) the connected components of H remain unchanged.*

Proof. To prove part (a), we show that the conditions (a′) through (f′) still hold true. Let us represent $C[i]$ and $C[j]$ respectively as $\{C1, (u, v), C2\}$ and $\{C3, (x, y), C4\}$, where $C1$ through $C4$ are sequences of vertices and vertex pairs. The exchange of Step 1(b) inserts $C[j]$ into $C[i]$ in the form of $\{C1, (u, y), C4, C3, (x, v), C2\}$. It is obvious that if $1 \le i \le k$, then $C[i]$ remains an s_i-t_i chain and that if $i > k$, then $C[i]$ remains a closed chain that does not join any source-sink pair. Also, it can be easily seen that $C[i]$ remains free, simple, and disjoint to other chains. Notice that the set of linking edges of $C[j]$ is merged into that of $C[i]$. Since two vertex pairs (u, y) and (x, v) are contained in G_p, it is obvious that $R \cup L$ is still $\mathbb{P}$-separated. Notice that for each $p \in \mathbb{P}$, the values of size(Z_p) and $\beta(Z_p)$ are preserved. The proof of part (a) is finished. The part (b) is obvious since no edge of H is removed and a new edge qr can be added only if there exist edges qp and pr in H. □

Lemma 26. *Suppose that $p \in \mathbb{P}$, $\Phi_p \geq 0$, and Z is a fine set. If CycleMerge (p) is applied, then (a) Z remains a fine set and (b) the connected components of H remain unchanged.*

Proof. The proof is similar to that of Lemma 25. Notice that if $q \in \mathbb{P}$ and $q \neq p$, then Z_q remains unchanged. Hence, if the existence of BiDPC$[\mathcal{K}(Z_q)|G_q, \mathcal{F}(Z_q)]$ was guaranteed before applying CycleMerge(p), then it is still guaranteed. For G_p, the induction hypothesis applies since $\Phi_p \geq 0$. □

As the first step of Phase 2, we apply CycleMerge(p) for every $p \in \mathbb{P}$. By Lemma 25, Z remains a strong $\mathbb{P}$-regular refinement of $F \cup K$. At this point, if we have $\mathcal{K}(R_p) \neq \emptyset$ for every $p \in \mathbb{P}$, then there remains no additional closed chains. That is, $L = \emptyset$ and $F \cup R$ is fine. Therefore, we can obtain BiDPC$[K|G, F]$ by finding BiDPC$[\mathcal{K}(R_p)|G_p, F_p \cup \mathcal{F}(R_p)]$ for every $p \in \mathbb{P}$ and merging them using the linking edges.

Let us sketch our construction in view of the graph H. It is assumed that CycleMerge(p) is applied for each $p \in \mathbb{P}$. Let $p, q \in \mathbb{P}$ be such that pq is an edge in H. Suppose $\mathcal{K}(R_p) \neq \emptyset$. Then, $\mathcal{K}(L_p) = \emptyset$. There exists a chain $C[i]$ such that $\mathcal{K}(C[i]_p) \neq \emptyset$ and $\mathcal{K}(C[i]_q) \neq \emptyset$, where $1 \leq i \leq n$. Since $\mathcal{K}(L_p) = \emptyset$, it follows $1 \leq i \leq k$. Thus, $\mathcal{K}(R_q) \neq \emptyset$. In this way, we deduce that if p and q are connected in H, then either $\mathcal{K}(R_p), \mathcal{K}(R_q) \neq \emptyset$ or $\mathcal{K}(R_p) = \mathcal{K}(R_q) = \emptyset$. Therefore, we are done if H becomes connected or there exists some $p \in \mathbb{P}$ such that $\mathcal{K}(R_p) \neq \emptyset$ for each connected component of H.

Let us describe Operation Stretch. A subchain $\{x, (y, z)\}$ is (p, q)-*stretchable* if $p, q \in \mathbb{P}$; $x \in V_p$; $y, z \in V_q$; $\Phi_p \geq 1$; $\Phi_q \geq 0$; and $\{x, (y, z)\} \subset C[i]$ for some $1 \leq i \leq n$, where xy is a linking edge. Given a (p, q)-stretchable subchain $\{x, (y, z)\}$, the following operation 'stretches' the vertex x into a vertex pair (x, x'), where $x' \in V_p$.

Operation Stretch$(x, (y, z))$

1. Select $x' \in V_p$ such that $\beta(x') = \beta(x)$ and $x'y'$ is free with respect to Z, where $y' \in N(x') \cap V_q$.
2. Update $C[i] = (C[i] \setminus \{x, (y, z)\}) \cup \{(x, x'), (y', z)\}$.
3. Apply CycleMerge(p).

Lemma 27. *Suppose that Z is fine. Let $\{x, (y, z)\}$ be a (p, q)-stretchable subchain. Let P and Q be the connected components of H containing p and q, respectively. If Stretch$(x, (y, z))$ is applied, then (a) Z remains a fine set and (b) $P \cup Q$ becomes a connected component of H.*

Proof. The proof for part (a) is similar to that in Lemma 25. At Step 2, the condition (f′) may be invalidated in G_p. Notice that we have $\Phi_p \geq 0$ since Φ_p is decreased by one, and we obtained $\mathcal{K}(Z_p) \neq \emptyset$. Thus, the existence of BiDPC$[\mathcal{K}(Z_p)|G, \mathcal{F}(Z_p)]$ is guaranteed by the induction hypothesis. The existence of the edge $x'y'$ is due to the proof of Lemma 24. The part (b) is obvious. □

The following Operation Propagate uses the Procedure BiDPC-B.

Operation Propagate(q, p)

1. Identical to Steps 1 and 2 of BiDPC-B$(G, \mathcal{K}(Z), \mathcal{F}(Z), q, p)$.
2. Update $C[i] = (C[i] \setminus (s,t)) \cup \{(s,u), (u',v'), (v,t)\}$, where $1 \leq i \leq n$ is such that $(s,t) \in C[i]$.
3. Apply CycleMerge(p).

Lemma 28. *Suppose that Z is fine, $\Phi_q \geq -5$, $\Phi_p \geq 2$, and every vertex of V_q has a neighbor in V_p, where $p, q \in \mathbb{P}$. Let P and Q be the connected components of H containing p and q, respectively. If Propagate(q,p) is applied, then (a) Z remains a fine set and (b) $P \cup Q$ becomes a connected component of H.*

Proof. At Step 1 of Procedure BiDPC-B, there exists $\mathcal{P}_q$ since Z is fine. We claim the existence of the edge uv required in the Procedure BiDPC-B. In the proof of Lemma 8, (9) is a lower bound of the number of nonblocked candidates for uv. Notice that $f^o \leq m-7$. By substituting 2^{m-1}, $f_q + k_q$, f_2, and f_p respectively by $2^{m-l(q)}$, $m - l(q) + 5$, $m-7$, and $m - l(p) - 2$ at (9), we obtain $2^{m-l(q)} - 6m + 2l(p) + 2l(q) + 8$. Since $p, q \in \mathbb{P}$, it follows $2^{m-l(q)} - 6m + 2l(p) + 2l(q) + 8 \geq 2^{m-3} - 6m + 18 \geq 2$. The claim is proved. Part (a) is obvious from the construction and the proof of Lemma 8. Notice that $\Phi_p \geq 0$ since Φ_p is decreased by two. Part (b) is obvious. □

We remark that there exists a simple construction when $k \geq 2$. In this case, Propagate operation can be repeatedly applied to make the graph H connected. Since applying Propagate(q,p) decreases Φ_q by two, the condition $\Phi_q \geq -5$ indicates that we can apply Propagate(q,p) for a fixed q at least three times, where $p, q \in \mathbb{P}$ and p is not fixed. For example, suppose that $\mathcal{K}(R_{000}) \neq \emptyset$, and $\mathcal{K}(R_p) = \emptyset$ for each $p \in \mathbb{P} \setminus 000$. Let us apply Propagate$(000, 001)$, Propagate$(000, 01)$, and Propagate$(000, 10)$. Notice that we do not need to find BiDPC$[\mathcal{K}(Z_{000})|G_{000}, \mathcal{F}(Z_{000})]$ each time and that the third application is possible since the first two do not make $\Phi_{000} < -4$. Finally, applying Propagate$(01, 11)$ makes H connected, and it is finished.

A subchain $\{x, \alpha\}$ is (p,q)-*joinable* if all of the following are satisfied: (i) $\{x, \alpha\} \subset C[j]$ for some $j > k$, and α is a vertex y or a vertex pair (y,z), where xy is a linking edge; (ii) $p, q \in \mathbb{P}$, $x \in V_p$, $y \in V_q$ (or $y, z \in V_q$), and $\Phi_p \geq 0$; (iii) $\Phi_q \geq 1$ if $\alpha = y$ and $\Phi_q \geq 0$ if $\alpha = (y,z)$; (iv) $\beta(Z_p) = 0$, $k = 1$, $\beta(s_1) = \beta(t_1) = -\beta(x)$, $C[1] = \{(s_1, t_1)\}$, and $\mathcal{K}(Z_p) = \{(s_1, t_1)\}$; and (v) the edge uu' is free with respect to Z if $u \in N(t_1) \cap V_p$ is free with respect to Z, where $u' \in N(u) \cap V_q$.

Given a (p, q)-joinable subchain $\{x, \alpha\}$, the following operation joins the closed chain $C[j]$ that contains $\{x, \alpha\}$ into the chain $C[1]$. This operation is designed to cover special cases where Stretch and Propagate operations are not suitable.

Operation Join(x, α)

1. Find $\mathcal{P}$=BiDPC$[\{(s_1, x)\}|G_p, \mathcal{F}(Z_p) \setminus x]$. Let $P = (s_1, P_u, u, t_1, P_x, x)$ be its s_1-x path.
2. Case $\alpha = (y, z)$: Put $C[1] = \{(s_1, u), (u', z)\} \cup (C[j] \setminus \{x, (y, z)\}) \cup \{(x, t_1)\}$, where $u' \in N(u) \cap V_q$. Put $C[j] = \emptyset$.
3. Case $\alpha = y$: Put $C[1] = \{(s_1, u), (u', y)\} \cup (C[j] \setminus \{x, y\}) \cup \{(x, t_1)\}$, where $u' \in N(u) \cap V_q$. Put $C[j] = \emptyset$.
4. Apply CycleMerge(q).

Lemma 29. *Suppose that Z is fine. Let $\{x, \alpha\}$ be a (p, q)-joinable subchain. Let P and Q be connected components of H containing p and q, respectively. If Join(x, α) is applied, then (a) Z remains a fine set and (b) $P \cup Q$ or one of its superset becomes a connected component of H.*

Proof. The Join operation works as follows. The existence of $\mathcal{P}$ is guaranteed by the induction hypothesis. Since $\beta(Z_p) = 0$ and $\beta((s_1, t_1)) + \beta(x) = \beta((s_1, x))$, P is a Hamiltonian path of $G_p \setminus (\mathcal{F}(Z_p) \setminus x)$. Hence, the path P has the form of $(s_1, P_u, u, t_1, P_x, x)$. Since s_1 and t_1 have the same color, it follows $u \neq s_1$. At this point, u, u', and uu' are free with respect to Z. Notice that the chain $C[j]$ with the subchain $\{x, (y, z)\}$ (resp. $\{x, y\}$) removed is a z'-x' chain (resp. a y'-x' chain), namely D, where $z' \in N(z)$ (resp. $y' \in N(y)$) and $x' \in N(x)$. Steps 2 and 3 merge $C[j]$ into the s_1-t_1 chain in the form of $\{(s_1, u), (u', z), D, (x, t_1)\}$ (resp. $\{(s_1, u), (u', y), D, (x, t_1)\}$). Now, the s_1-t_1 chain $C[1]$ passes through G_p and G_q via vertex pairs. For the proof of part (a), conditions (a$'$) and (b$'$) are obvious from the arguments above. Since $\beta(\{(s_1, u), (x, t_1)\}) = \beta(\{(s_1, t_1), x\})$ and $\beta(u') = \beta(y)$, condition (c$'$) follows. The two paths (s_1, P_u, u) and (t_1, P_x, x) form BiDPC$[\mathcal{K}(Z_p)|G_p, \mathcal{F}(Z_p)]$, and the existence of BiDPC$[\mathcal{K}(Z_q)|G_q, \mathcal{F}(Z_q)]$ is guaranteed by the induction hypothesis. Notice that Z_r is not changed in any $r \in \mathbb{P} \setminus \{p, q\}$. Therefore, part (a) is proved. Let us consider part (b). If α is a vertex pair, $P \cup Q$ becomes a connected component of H. However, when α is a vertex, $P \cup Q \cup \{r \in \mathbb{P} : \mathcal{K}(C[j]_r) \neq \emptyset\}$ becomes a connected component of H. □

Now, we will apply these operations to construct BiDPC$[K|G, F]$.

Lemma 30. *Suppose that $f + 2k \leq m$ and* (10) *is true. Then, there exists BiDPC$[K|G, F]$.*

Proof. Let Z be the strong $\mathbb{P}$-regular refinement of $F \cup K$ constructed in Phase 1. Apply CycleMerge(p) for every $p \in \mathbb{P}$. If $\mathcal{K}(R_p) \neq \emptyset$ for each $p \in \mathbb{P}$, then $L = \emptyset$ and $F \cup R$ is a strong $\mathbb{P}$-regular refinement of $F \cup K$. Therefore, we can obtain a desired BiDPC by finding BiDPC$[\mathcal{K}(R_p)|G_p, F_p \cup \mathcal{F}(R_p)]$ for every $p \in \mathbb{P}$

and merging them using the linking edges. Thus, in what follows, we assume that H has at least two connected components: one contains $p \in \mathbb{P}$ such that $\mathcal{K}(R_p) \neq \emptyset$ and the other does not. We will show that Stretch, Propagate, and Join operations can be applied to make H connected or every connected component of H contains some $p \in \mathbb{P}$ such that $\mathcal{K}(R_p) \neq \emptyset$, where p varies by the connected component. We distinguish two cases according to which canonical form is being used.

Case 1. The 6-tuple is in the first canonical form.

By Procedure Regular-D, H has a connected component that is a superset of $\{001, 01, 1(1-i)\}$, where $i \in \{0,1\}$ (see Fig.s 5a and 5b). Also, $\mathcal{K}(R_{000}) = \emptyset$ only if $\Phi_{000} \geq 2$. This is because $\text{size}(Y^*_{000}) = 2 \leq \text{size}(X) - l(000) - |\mathcal{K}(X)| - 2$ for the strong $\mathbb{P}$-regular refinement Y^* of the 6-tuple X and $\text{size}(Y'_{000}) \leq \text{size}(X') - |\mathcal{K}(X')|$ for the $\mathbb{P}$-regular refinement Y' of any cell X' other than the 6-tuple. Similarly, we have $\mathcal{K}(R_{01}) = \emptyset$ only if $\Phi_{01} \geq 2$. We distinguish three subcases according to the partition of $\mathbb{P}$ that is determined by H's connected components.

Case 1.1. There are two connected components $\{001, 01, 10, 11\}$ and $\{000\}$, or $\{001,\ 01,\ 1(1-i)\}$ and $\{000,\ 1i\}$.

Suppose $\mathcal{K}(R_{000}) \neq \emptyset$. Then, we have $\mathcal{K}(R_{01}) = \emptyset$, so $\Phi_{01} \geq 2$. By Lemma 28, applying Propagate(000, 01) suffices. Otherwise, we have $\mathcal{K}(R_{000}) = \emptyset$, $\mathcal{K}(R_{001}) \neq \emptyset$, and $\Phi_{000} \geq 2$. Therefore, applying Propagate$(001, 000)$ suffices.

Case 1.2. There are two connected components $\{000, 001, 01, 1(1-i)\}$ and $\{1i\}$.

Since 10 and 11 are disconnected in H, due to the construction of Procedure Regular-D, we deduce $F^b_1 \cup K^b_1 = F^b_{1i} \cup K^b_{1i}$. There exists a subchain $\{x, (\tilde{x}, z)\}$ in $R \cup L$ such that $x \in V^w_{1i}$ and $z \in V_{1(1-i)}$, where $x\tilde{x}$ is a linking edge. (For example, when $i = 0$, Y^* contains $\{\bar{\gamma_2}, (\hat{\gamma_2}, \bar{\delta_2})\}$ (see Fig. 5b), where $\bar{\gamma_2}$, $\hat{\gamma_2}$, and $\bar{\delta_2}$ respectively correspond to x, $\tilde{x}$, and z.) The existence of such a subchain is preserved under the CycleMerge operation. Note that this property holds without regard to the canonical form being used.

Suppose $\mathcal{K}(R_{1i}) = \emptyset$. Then, $\Phi_{1i} \geq 1$. Hence, $\{x, (\tilde{x}, z)\}$ is $(1i, 1(1-i))$-stretchable, so Lemma 27 applies. Suppose instead that $\mathcal{K}(R_{1i}) \neq \emptyset$. Then, $\mathcal{K}(R_{1(1-i)}) = \emptyset$ and $\Phi_{1(1-i)} \geq k$. If $\Phi_{1i} \geq 1$, we apply Stretch$(x, (\tilde{x}, z))$. If $\Phi_{1(1-i)} \geq 2$, we apply Propagate$(1i, 1(1-i))$. Lemmas 27 and 28 guarantee their correctness. Let $\Phi_{1i} = 0$ and $\Phi_{1(1-i)} = 1$. We claim that $\{x, (\tilde{x}, z)\}$ is $(1i, 1(1-i))$-joinable. By Lemma 29, provided the claim is true, applying Join$(x, (\tilde{x}, z))$ suffices. Notice that our construction does not change Z_0.

Let us prove the claim. We use the subchain $\{x, (\tilde{x}, z)\}$. That is, $\alpha = (\tilde{x}, z)$. The conditions (i) through (iii) of (p,q)-joinability, where $p = 1i$ and $q = 1(1-i)$ are obvious by the arguments above. Recall that we have $\beta(Z_{10}) = \beta(Z_{11}) = 0$ at the end of Phase 1. Since CycleMerge operation preserves $\beta(Z_{10})$ and $\beta(Z_{11})$, we still have $\beta(Z_{1i}) = 0$. Supposing $k \geq 2$ leads to $\Phi_{1(1-i)} \geq 2$, which contradicts $\Phi_{1(1-i)} = 1$. Thus, $k = k_{1i} = 1$. Suppose to the contrary that there exists a source-sink pair $(s,t) \in K^w_{1i} \cup K^o_{1i}$. If $(s,t) \in K^w$, then there exists a type-3 good cell $\{(s,t), b\}$ that is refined in Phase 1, where $b \in F^b_{1i} \cup K^b_{1i}$; recall that $F^b_{1i} \cup K^b_{1i} = F^b_1 \cup K^b_1$. We observe that such a type-3 set contributes at least three to $\Phi_{1(1-i)}$, which is a contradiction. In a similar way, we can show that supposing $(s,t) \in K^o$ leads to a contradiction. Hence, we have $\beta(s_1) = \beta(t_1) = -\beta(x)$. From $(s_1,t_1) \in K^b_{1i}$, we observe that $C[1] = \{(s_1,t_1)\}$ when Phase 1 is terminated. Suppose to the contrary that we had $\mathcal{K}(L_{1i}) \neq \emptyset$ when Phase 2 is reached. It implies that, by Remark 2, $1i$ and some $p \in \mathbb{P}\backslash 1i$ are connected in H, which is a contradiction. Hence, we conclude that $\mathcal{K}(L_{1i}) = \emptyset$, $C[1] = \{(s_1,t_1)\}$, and $\mathcal{K}(Z_{1i}) = \{(s_1,t_1)\}$. The condition (iv) is verified. Suppose, to the contrary, that a vertex $u \in N(t_1) \cap V_{1i}$ is free with respect to Z and $u\tilde{u}$ is not. Then, u is a white vertex. Since each member of F^o_{12} contributes one to $\Phi_{1(1-i)}$, $u\tilde{u}$ must be fault-free. There is no possibility of $\tilde{u}$ being a faulty vertex or a terminal since $F^b_1 = F^b_{1i}$ and $k = k_{1i} = 1$. The remaining possibility is that $\tilde{u} \in \mathcal{U}(Z) \setminus (F \cup S \cup T)$. Since $\tilde{u}$ is a black vertex in G_1, it always appears in the form of a linking edge $u\tilde{u}$ in the construction of Phase 1, by Remark 2. Recall that the set of linking edges is preserved under CycleMerge operations. Hence, u is not free with respect to Z, which is a contradiction. The condition (v) is verified. Therefore, we have the claim.

Case 1.3. There are three connected components $\{001, 01, 1(1-i)\}$, $\{1i\}$, and $\{000\}$.

We first handle G_0, then apply the construction of Case 1.2 to G_1. Suppose $\mathcal{K}(R_{000}) \neq \emptyset$ and $\mathcal{K}(R_{01}) \neq \emptyset$. Then, we have (i) $\mathcal{K}(R_p) \neq \emptyset$ for any $p \in \{000, 001, 01, 1(1-i)\}$. Suppose otherwise. If $\mathcal{K}(R_{000}) = \emptyset$, i.e., $\Phi_{000} \geq 2$, then we apply Propagate(001, 000). If $\mathcal{K}(R_{000}) \neq \emptyset$, i.e., $\mathcal{K}(R_{01}) = \emptyset$ and $\Phi_{01} \geq 2$, then we apply Propagate(000, 01). By doing so, we obtain that (ii) H contains two connected components $\{001, 01, 1(1-i), 000\}$ and $\{1i\}$, which is the same as Case 1.2. For applying the construction of Case 1.2, there is essentially no difference between (i) and (ii). It is finished if $\mathcal{K}(R_p) \neq \emptyset$ for each $p \in \mathbb{P}$; otherwise, applying the construction of Case 1.2 establishes this condition. Since the construction of Case 1.2 does not change Z_0, the constructions in G_0 and G_1 can be done independently.

Case 2. The 6-tuple is in the second canonical form.

By Procedure Regular-D, H has a connected component that is a superset of $\{000, 001, 1(1-i)\}$ for some $i \in \{0,1\}$. There are four subcases.

Case 2.1. There are two connected components $\{000, 001, 10, 11\}$ and $\{01\}$.

There exist subchains $\{\delta_2, (\tilde{\delta_2}, z)\}$ and $\{\delta_1, \alpha\}$ in $R \cup L$, where $z \in V_{001}$ and α is either a vertex $\tilde{\delta_1}$ or a vertex pair $(\tilde{\delta_1}, z')$, where $z' \in V_{11}$ (see Fig. 5c). They stem from Y^* of the 6-tuple. Suppose $\mathcal{K}(R_{01}) = \emptyset$. Then, $\Phi_{01} \geq 1$, so $\{\delta_2, (\tilde{\delta_2}, z)\}$ is (01, 001)-stretchable. Hence, Lemma 27 applies. Suppose instead that $\mathcal{K}(R_{01}) \neq \emptyset$. If $\Phi_{01} \geq 1$, we apply Stretch$(\delta_2, (\tilde{\delta_2}, z))$. If $\Phi_{11} \geq 2$, we apply Propagate(01, 11). Lemmas 27 and 28 guarantee their correctness. It remains to consider the case where $\Phi_{01} = 0$ and $\Phi_{11} = 1$. In this case, applying Join(δ_1, α) suffices. We can show that $\{\delta_1, \alpha\}$ is (01, 11)-joinable with arguments similar to the ones in Case 1.2. Notice that there was no type-1 cell in Phase 1 since a type-1 cell consists of a faulty vertex (resp. a source-sink pair) contributes 1 (resp. 2) to Φ_{11}. From this and $\Phi_{11} = 1$, we obtain $\beta(Z_{01}) = 0$. Also, from the case condition and $k = k_{01}$, we can deduce that for each white vertex $v \in \mathcal{U}(Z) \setminus (F \cup S \cup T)$ in V_{11}, there always exists a linking edge $v\bar{v}$.

Case 2.2. There are two connected components $\{000, 001, 01, 1(1-i)\}$ and

$\{1i\}$.

The proof for this case is identical to that of Case 1.2.

Case 2.3. There are two connected components $\{000, 001, 1(1-i)\}$ and $\{1i, 01\}$.

By the construction of Procedure Regular-E, there is no possibility of being $k_2 \geq 1$. Hence, $k_2 = 0$ and there exists an additional closed chain that contains vertex pairs in G_{1i} and in G_{01}. Such a chain exists only if Procedure Regular-C is invoked with a type-4 cell $\{w, b\}$, where w is a white unit in G_{00} and b is a black unit in $G_{1(1-i)}$. By (B1), w is a source-sink pair. Therefore, it suffices to consider when $\mathcal{K}(R_{01}) = \emptyset$ and $\Phi_{01} \geq 1$. Since there exists a $(01, 001)$-stretchable subchain $\{\delta_2, (\tilde{\delta_2}, z)\}$, where $z \in V_{001}$, applying Stretch$(\delta_2, (\tilde{\delta_2}, z))$ suffices.

Case 2.4. There are three connected components $\{000, 001, 1(1-i)\}$, $\{1i\}$, and $\{01\}$.

If $\Phi_{1i} \geq 1$, then there exists a $(1i, 1(1-i))$-stretchable subchain $\{x, (\tilde{x}, z)\}$ for some $x \in V_{1i}^w$ and $z \in V_{1(1-i)}$. If $\Phi_{01} \geq 1$, then there exists a $(01, 001)$-stretchable subchain $\{\delta_2, (\tilde{\delta_2}, z')\}$ for some $z' \in V_{001}$. Notice that $\Phi_{1i} = 0$ only if $k = k_{1i}$ and that $\Phi_{01} = 0$ only if $k = k_{01}$. Thus, both $\Phi_{1i} = 0$ and $\Phi_{01} = 0$ cannot occur simultaneously. If $\Phi_{1i} \geq 1$ and $\Phi_{01} \geq 1$, applying Stretch$(x, (\tilde{x}, z))$ and Stretch$(\delta_2, (\tilde{\delta_2}, z'))$ suffices. Suppose $\Phi_{1i} = 0$ and $\Phi_{01} \geq 1$. Then, $k = k_{1i}$. Proceeding with the construction of Case 1.2, we apply Propagate$(1i, 1(1-i))$ if $\Phi_{1(1-i)} \geq 2$ and we apply Join$(x, (\tilde{x}, z))$ if $\Phi_{1i} = 0$ and $\Phi_{1(1-i)} = 1$. If $\mathcal{K}(R_p) \neq \emptyset$ is obtained for every $p \in \mathbb{P}$, then it is finished. If not, the only possibility is that H has two connected components $\{000, 001, 10, 11\}$ and $\{01\}$. It still holds $\Phi_{01} \geq 1$ since Z_0 remains unchanged. Hence, there exists a $(01, 001)$-stretchable subchain $\{\delta_2, (\tilde{\delta_2}, z')\}$, so we apply Stretch$(\delta_2, (\tilde{\delta_2}, z'))$. Now, suppose $\Phi_{1i} \geq 1$ and $\Phi_{01} = 0$. Then, $k = k_{01}$. Let $\beta(Z_{01}) = 0$. Then, there exists a (01, 11)-joinable subchain $\{\delta_1, \alpha\}$, where $\alpha = \bar{\delta_1}$ or $\alpha = (\bar{\delta_1}, z'')$ for some $z'' \in V_{11}$. The existence proof is similar to the proof in Case 1.2, and therefore omitted here. Sequently, we apply Join(δ_2, α). When $i = 1$, the Join operation makes H connected since the subchain $\{\delta_1, \alpha\}$ is a subset of a closed chain (stems from Y^*) that passes G_{000}, G_{001}, and G_{01} via vertex pairs. When $i = 0$, there exists a $(10, 11)$-stretchable chain $\{x, (\tilde{x}, z')\}$ that stems from Y^*. Hence, Lemma 27 applies. Let $\beta(Z_{01}) \geq 1$. Notice that $\beta(Z_{01}) > 0$ implies the existence of type-1 cells in Phase 1. From this and $k = k_{01}$, we obtain $\Phi_{10}, \Phi_{11} \geq 2$. So applying Propagate(01, 11) and Propagate(11, 10) suffices. This finishes the proof. □

6. Conclusion

We proved that the hypercube Q_m with at most f faulty graph elements removed has a paired many-to-many bipartite k-disjoint path cover for any $k \geq 1$ subject to $f + 2k \leq m$. We presented a constructive proof using two strategies: a traditional divide-and-conquer approach using two Q_{m-1}s that span Q_m and a relaxation using three Q_{m-2}s and two Q_{m-3}s that span Q_m. We believe that this relaxation technique may apply for proving similar problems.

References

[1] J.A. Bondy and U.S.R. Murty, *Graph Theory*, 2nd printing, Springer, 2008.

[2] R. Caha and V. Coubek, "Spanning multi-paths in hypercubes," *Discrete Math.* **307** (2007) 2053–2066.

[3] X.-B. Chen, "Many-to-many disjoint paths in faulty hypercubes," *Information Sciences* **179** (2009) 3110–3115.

[4] X.-B. Chen, "Paired many-to-many disjoint path covers of hypercubes with faulty edges," *Inf. Process. Lett.* **112** (2012) 61–66.

[5] T. Dvořák, "Hamiltonian cycles with prescribed edges in hypercubes," *SIAM Journal on Discrete Mathematics* **19** (2005) 135–144.

[6] T. Dvořák and P. Gregor, "Partitions of faulty hypercubes into paths with prescribed endvertices," *SIAM Journal on Discrete Mathematics* **22(4)** (2008) 1448–1461.

[7] P. Gregor and T. Dvořák, "Path partitions of hypercubes," *Inf. Process. Lett.* **108(6)** (2008) 402–406.

[8] J. Fink and P. Gregor, "Long paths and cycles in hypercubes with faulty vertices," *Information Sciences* **179** (2009) 3634–3644.

[9] J.-S. Fu, "Fault-tolerant cycle embedding in the hypercube," *Parallel Computing* **29** (2003) 821-832.

[10] J.-S. Fu, "Longest fault-free paths in hypercubes with vertex faults," *Information Sciences* **176** (2006) 759–771.

[11] S.-Y. Hsieh, G.-H. Chen, and C.-W. Ho, "Hamiltonian-laceability of star graphs," *Networks* **36** (2000) 225–232.

[12] S.-Y. Hsieh and P.-Y. Yu, "Fault-free mutually independent Hamiltonian cycles in hypercubes with faulty edges," *Journal of Combinatorial Optimization* **13(2)** (2007) 153–162.

[13] S.-Y. Hsieh and Y.-F. Weng, "Fault-tolerant embedding of pairwise independent Hamiltonian paths on a faulty hypercube with edge faults," *Theory of Computing Systems* **45(2)** (2009) 407–425.

[14] S. Jo, J.-H. Park, and K.Y. Chwa, "Paired 2-disjoint path covers and strongly Hamiltonian laceability of bipartite hypercube-like graphs," *Information Sciences* **242** (2013) 103–112.

[15] C.-D. Park and K.Y. Chwa, "Hamiltonian properties on the class of hypercube-like networks," *Inf. Process. Lett.* **91(1)** (2004) 11–17.

[16] 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*, 2004, pp. 392–401.

[17] J.-H. Park and H.-C. Kim, "Longest paths and cycles in faulty star graphs," *Journal of Parallel and Distributed Computing* **64(11)** (2004) 1286–1296.

[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 and Distrib. Syst.* **17(3)** (2006) 227–240.

[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. Comput.* **58(4)** (2009) 528–540.

[20] C.-H. Tsai, J.J.M. Tan, T. Liang, and L.-H. Hsu, "Fault-tolerant hamiltonian laceability of hypercubes," *Inf. Process. Lett.* **83(6)** (2002) 301–306.

[21] C.-H. Tsai, "Linear array and ring embeddings in conditional faulty hypercubes," *Theor. Comput. Sci.* **314(3)** (2004) 431–443.

[22] C.-H. Tsai, J.J.M. Tan, and L.-H. Hsu, "The super-connected property of recursive circulant graphs," *Inf. Process. Lett.* **91(6)** (2004) 293–298.

[23] C.-H. Tsai, "Embedding various even cycles in the hypercube with mixed link and node failures," *Applied Mathematics Letters* **21(8)** (2008) 855–860.

[24] Y.-C. Tseng, "Embedding a ring in a hypercube with both faulty links and faulty nodes," *Inf. Process. Lett.* **59** (1996) 217–222.