Two algorithms extending a perfect matching of the hypercube into a Hamiltonian cycle

Jiří Fink *

Department of Theoretical Computer Science and Mathematical Logic

Faculty of Mathematics and Physics

Charles University in Prague

Abstract

In paper [7] we proved Kreweras' conjecture [16] asserting that every perfect matching of the n-dimensional hypercube can be extended into a Hamiltonian cycle. In this paper, we present two algorithms to find a Hamiltonian cycle extending a given perfect matching. The first algorithm finds whole Hamiltonian cycle in time linear in the number of vertices whose existence was asked by Knuth [15]. The second algorithm prints edges as they occur on the Hamiltonian cycle and the first k edges are found in time polynomial in both n and k.

1 Introduction

A set of edges $P \subseteq E$ of a graph $G = (V, E)$ is a *matching* if every vertex of G is incident with at most one edge of P. If a vertex v of G is incident with an edge of P, we say that v is *covered* by P and $V(P)$ denotes the set of vertices covered by P. A matching P is *perfect* if every vertex of G is covered by P. For graph $G = (V, E)$, we denote by $V(G)$ and $E(G)$ the set of vertices and edges of G, respectively.

The *n-dimensional hypercube* Q_n is a graph whose vertex set consists of all subsets of $[n] = \{1, \dots, n\}$, with two vertices $u, v \in [n]$ being adjacent whenever $|u \triangle v| = 1$ where $n \geq 1$ and $u \triangle v$ denotes the symmetric difference of u and v. Elements of $[n]$ are called *coordinates*. There are several appealing problems related to hypercubes. Probably the most prominent of them was the notorious Middle Levels Conjecture: Despite the attention it has attracted, it took over three decades until, in a recent breakthrough, Mütze answered it affirmatively [17] and with Nummenpalo [18] presented an algorithm generating a Hamiltonian cycle in the Middle Levels Graph in constant time per vertex.

It is well known that Q_n is Hamiltonian for every $n \geq 2$. This statement can be traced back to 1872 [13]. Since then the research on Hamiltonian cycles in hypercubes satisfying certain additional properties has received considerable attention. An interested reader can find more details about this topic in the survey of Savage [20], e.g. Dvořák [3] showed that any set of at most $2n-3$ edges of Q_n ($n \geq 2$) that induces vertex-disjoint paths is contained in a Hamiltonian cycle.

Kreweras [16] conjectured that every perfect matching in the n-dimensional hypercube with $n \geq 2$ extends to a Hamiltonian cycle which was proved in [7]. Paper [7] actually proved a stronger statement where $K(G)$ is the complete graph on vertices of a graph G.

Theorem 1.1 ([7]). *For every perfect matching P of $K(Q_n)$ for $n \geq 2$ there exists a perfect matching R of Q_n such that $P \cup R$ forms a Hamiltonian cycle of $K(Q_n)$.*

Let us call edges of $E(Q_n)$ *short* while edges of $E(K(Q_n)) \setminus E(Q_n)$ *long*. The idea of using long edges has been successfully applied in many other papers, see e.g. [8, 9, 11, 6, 2, 10, 12, 1]. Although

*Supported by the Czech Science Foundation grant GA14-10799S. E-mail: fink@ktiml.mff.cuni.cz

various modifications of the proof [7] of Kreweras' conjecture have been presented [8, 9, 11], all of them are based on long edges. Section 3 presents an alternative proof of Kreweras' conjecture which does not need to use long edges and this method is used in both algorithms for extending perfect matching of Q_n into a Hamiltonian cycle of Q_n. Furthermore, this alternative proof may be useful to solve another long-standing open problem formulated in 1993 by Ruskey and Savage [19]: Does every matching (not only perfect) in a hypercube Q_n extend to a Hamiltonian cycle of Q_n? Considering Theorem 1.1, one may ask whether every matching P of $K(Q_n)$ extends to a Hamiltonian cycle of $K(Q_n)$ using short edges. Clearly, the matching P must be balanced, i.e. the number of vertices $u \in V(P)$ with $|u|$ even is equal to the number of vertices $u \in V(P)$ with $|u|$ odd. However, this condition is not sufficient since Dvořák and Fink [4] found a balanced matching of $K(Q_n)$ which cannot be extended into a Hamiltonian cycle of $K(Q_n)$ using short edges. So, extending a perfect matching of Q_n into a Hamiltonian cycle without using long edges may be useful to solve Ruskey and Savage conjecture.

Knuth [15] asked whether a Hamiltonian cycle extending a given perfect matching of Q_n can be found in linear time. The answer to this question may depend on the computational model. In this paper, we consider the Random Access Machine (word-RAM), i.e. one vertex of the n-dimensional hypercube can be stored in one word of memory (e.g. using the characteristic vector of the subset of $[n]$) and bitwise operations can be computed by a single instruction, so e.g., we can compute the symmetric difference $u \triangle v$ of two vertices u and u of $V(Q_n)$ and find an arbitrary coordinate in which two vertices differ in constant time.

This computational model is important in the first (off-line) algorithm which finds a Hamiltonian cycle extending a given perfect matching on Q_n in linear time. The size of the input and the output is $\Theta(2^n)$ and our algorithm requires $\Theta(2^n)$ arithmetic and bitwise operations on $\mathcal{O}(n)$-bit words. In this algorithm, the input matching P is given as an array A of length 2^n such that $uA[u]$ is an edge of P for every vertex u. Note that $A[A[u]] = u$.

The off-line algorithm does not generate edges of a Hamiltonian cycle extending a given perfect matching in the order in which these edges lie on the cycle. Although it is possible to store all generated edges and then list them in the appropriate order, this is not an efficient way to find only first few edges. We are motivated by efficient algorithms generating all objects in a particular combinatorial class such as permutations, subsets, combinations, partitions, trees, strings etc. in particular order (see e.g. [20, 5, 14, 18]). So, our goal is developing an algorithm which for a given perfect matching P of Q_n iteratively finds a perfect matching R of Q_n extending P into a Hamiltonian cycle of Q_n so that edges of R are generated in the order of Hamiltonian cycle $P \cup R$ and first k edges are found in time which is polynomial in both n and k for every $k = 1, \dots, 2^n$. Since the size of the given perfect matching P is exponential in n, we assume that P is given by an oracle which for a given vertex u of Q_n returns the vertex u^P.

Our second algorithm presented in this paper is even more powerful. It is an on-line algorithm solving the following problem.

Problem 1.2. *Design an on-line algorithm for extending a perfect matching of Q_n into a Hamiltonian cycle as follows.*

Initial input: *An oracle of a perfect matching P of Q_n*
On-line input: *A vertex u of Q_n*
On-line output: *An edge uv of Q_n incident with the vertex u*
Requirement: *The union $P \cup R$ is a Hamiltonian cycle of Q_n where R is the set of edges returned by the on-line algorithm iteratively called for every vertex of Q_n.*

That is, the on-line algorithm obtains the oracle in the beginning. Then iteratively, the on-line algorithm obtains a vertex u of Q_n and returns an incident edge uv of Q_n. When the online algorithm is called for every vertex of Q_n, then the set of all returned edges forms a perfect matching of Q_n extending the perfect matching given by the oracle into a Hamiltonian cycle of Q_n. Using such an on-line algorithm, we can start in an arbitrary vertex u_1 and iteratively for every $k = 1, \dots, 2^{n-1}$ call this on-line algorithm to find an edge $u_k v_k$ of R and then call the oracle to find the edge $v_k u_{k+1}$ of P (see Function `find_hamiltonian_cycle` in Algorithm 5.1). We prove that the running time of the k-th iteration of the on-line algorithm is $\mathcal{O}(k \cdot \mathsf{poly}(n))$, so first k edges of the Hamiltonian cycle extending a given perfect matching can be found in $\mathcal{O}(k^2 \cdot \mathsf{poly}(n))$ time.

2 Preliminaries

In a multigraph, two vertices can be connected by more than one edge. We say that a multigraph G is connected if G contains a path between each pair of vertices. A component of a multigraph is a maximal connected subgraph. Every multigraph can be split into components and two vertices belong to the same component if and only if there exists a path between these vertices. A direct consequence is the following lemma.

Lemma 2.1. *A multigraph is connected if and only if the multigraph contain a vertex u such that for every vertex v there exists a path between u and v. Furthermore, a multigraph is connected if and only if there exists two vertices u and v connected by a path such that for every w there exist a path between u and w or between v and w.*

A connected multigraph (V, E) on at least two vertices is 2-edge-connected if and only if $(V, E \setminus \{e\})$ is connected for every $e \in E$. A 2-edge-connected component of a multigraph is a maximal 2-edge-connected subgraph. Every multigraph can be split into 2-edge-connected components and edges joining different 2-edge-connected components.

Lemma 2.2. *Two vertices of a multigraph belong to the same component if and only if there exists two edge-disjoint paths between these vertices. Furthermore, a multigraph on at least two vertices is 2-edge-connected if and only if there exists two edge-disjoint paths between every pair of vertices.*

Lemma 2.3. *Let G be a connected multigraph and u_1, u_2, u_3, u_4 be vertices of G. Then G contains two edge-disjoint paths P_1 and P_2 such that the set of all endvertices of P_1 and P_2 is $\{u_1, u_2, u_3, u_4\}$ and these paths can be found in time $\mathcal{O}(|E(G)|)$.*

Proof. Without lost of generality, we assume that G is a tree since we can restrict P_1 and P_2 to contain edges of a given spanning tree of G. Let P_1 be a path in G between u_1 and u_2 and P_2 be a path in G between u_3 and u_4. If P_1 and P_2 have a common edge, then the graph on edges $E(P_1) \triangle E(P_2)$ has no cycle and only vertices u_1, u_2, u_3 and u_4 have odd degrees, so $E(P_1) \triangle E(P_2)$ forms another two edge-disjoint paths with endvertices u_1, u_2, u_3 and u_4. This proof is constructive and it can be implemented in linear time. □

Given sets $A \subseteq B \subsetneq [n]$ where n is a positive integer, a subcube of Q_n determined by the pair (A, B) is the subgraph of Q_n induced by the set of vertices u of Q_n with $u \cap B = A$. Let $\mathrm{coord}(Q) = [n] \setminus B$. Clearly, such a subcube Q is isomorphic to the hypercube of dimension $\dim(Q) = |\,\mathrm{coord}(Q)| \geq 1$. We can split the subcube Q by a coordinate $d \in \mathrm{coord}(Q)$ into two subcubes $Q^{d,0}$ and $Q^{d,1}$ determined by pairs $(A, B \cup \{d\})$ and $(A \cup \{d\}, B \cup \{d\})$, respectively. For a vertex u of Q, let $Q_{d,u}$ be the subcube $Q^{d,0}$ or $Q^{d,1}$ containing u and let the other subcube be denoted by $Q'_{d,u}$. Since all $Q^{d,0}, Q^{d,1}, Q_{d,u}$ and $Q'_{d,u}$ are subcubes, we can split them by other coordinate $d' \in [n] \setminus (B \cup \{d\})$ and we can applied notations denoting subcubes, so e.g. $(Q^{d,0})^{d',1}$ is the subcube determined by the pair $(A \cup \{d'\}, B \cup \{d, d'\})$.

Let $\mathcal{C}$ be a non-empty set of vertex-disjoint subcubes of Q_n. Let $V(\mathcal{C})$ and $E(\mathcal{C})$ be the set of all vertices and edges of all subcubes of $\mathcal{C}$, respectively, and let $G(\mathcal{C})$ be the graph $(V(\mathcal{C}), E(\mathcal{C}))$ and let $K(\mathcal{C})$ be the complete graph on vertices $V(\mathcal{C})$. Note that $G(\mathcal{C})$ contains only short edges and furthermore only edges having both endvertices in the same subcube while $K(\mathcal{C})$ contains also long edges and also edges between different subcubes. An interconnection graph of $\mathcal{C}$ and a set of edges $Z \subseteq E(K(\mathcal{C}))$ is a multigraph $I(\mathcal{C}, Z)$ where every subcube of $\mathcal{C}$ is represented by a single vertex and two vertices of $I(\mathcal{C}, Z)$ are connected by as many edges as there are edges of Z between corresponding subcubes. In this paper, $\mathcal{C}$ always denotes a non-empty set of vertex-disjoint subcubes of Q_n.

Algorithm 2.1: Algorihm finding four vertices of S according to Lemma 2.5

1 **Function** `find_four_vertices`(Q, S)
2 **while** *there exists a coordinate* d *and* $i \in \{0,1\}$ *such that* $|S \cap V(Q^{d,i})| \leq 1$ **do**
3 $Q := Q^{d,1-i}$
4 Find a coordinate d and $i \in \{0,1\}$ with minimal $|S \cap V(Q^{d,i})|$
5 Choose two vertices u_1 and u_2 of $S \cap V(Q^{d,i})$
6 Choose a coordinate $d' \in u_1 \triangle u_2$
7 Find u_3 a vertex of $S \cap V((Q^{d,1-i})^{d',0})$ and u_4 a vertex of $S \cap V((Q^{d,1-i})^{d',1})$
8 **return** (u_1, u_2, u_3, u_4)

Theorem 2.4 (Gregor [11])**.** *For every perfect matching P of $K(\mathcal{C})$ there exists a perfect matching R of $G(\mathcal{C})$ such that $P \cup R$ is a Hamiltonian cycle of $K(\mathcal{C})$ if and only if $I(\mathcal{C}, P)$ is connected.*

In Section 3 we present an alternative proof of Theorem 2.4 which does not rely on long edges of $K(Q_n)$. For a vertex u covered by a matching P we denote u^P a vertex such that $uu^P \in P$.

For a set of vertices S of a subcube Q of $\mathcal{C}$ let $\mathcal{D}(S)$ be the set of coordinates that split Q into two subcubes so that each subcube contains at least one vertex of S. Formally, $\mathcal{D}(S) = \bigcup_{u,v \in S} u \triangle v$.

Lemma 2.5. *Let S be a set of vertices of a subcube Q. Then $|\mathcal{D}(S)| \geq |S| - 1$, or Q can be split by two coordinates d and d' into four subcubes $(Q^{d,0})^{d',0}$, $(Q^{d,0})^{d',1}$, $(Q^{d,1})^{d',0}$ and $(Q^{d,1})^{d',1}$ such that each of them contains at least one vertex of S. Furthermore if $|\mathcal{D}(S)| < |S| - 1$, then these coordinates d and d' and a vertex of S from every subcube $(Q^{d,0})^{d',0}$, $(Q^{d,0})^{d',1}$, $(Q^{d,1})^{d',0}$ and $(Q^{d,1})^{d',1}$ can be fount in time $\mathcal{O}(|S| \cdot \mathsf{poly}(n))$.*

Proof. Clearly, this lemma holds if $|S| \leq 3$ or $\dim(Q) \leq 2$. For the sake of contradiction, let us consider a counterexample with the smallest possible dimension of Q. Let k be the smallest number such that there exists a coordinate d and $i \in \{0,1\}$ of Q such that $Q^{d,i}$ contains k vertices of S. Note that $k \geq 2$; for otherwise, the subcube $Q^{d,1-i}$ and the set of vertices S' form a smaller counterexample since $\mathcal{D}(S') \subseteq \mathcal{D}(S) \setminus \{d\}$ where $S' = S \cap V(Q^{d,1-i})$. Hence, there exists a coordinate d' which splits $Q^{d,i}$ so that both subcubes $(Q^{d,i})^{d',0}$ and $(Q^{d,i})^{d',1}$ contains at least one vertex of S. Since vertices of S form a counterexample to the lemma, $(Q^{d,1-i})^{d',0}$ or $(Q^{d,1-i})^{d',1}$ contains no vertex of S. Hence, if we split Q by the coordinate d', then $Q^{d',0}$ or $Q^{d',1}$ contains less than k vertices of S which is a contradiction to the minimality of k.

If $|\mathcal{D}(S)| < |S| - 1$, then Function `find_four_vertices` (Algorithm 2.1) presents an algorithm finding vertices u_1, u_2, u_3 and u_4 of S from subcubes $(Q^{d,0})^{d',0}$, $(Q^{d,0})^{d',1}$, $(Q^{d,1})^{d',0}$ and $(Q^{d,1})^{d',1}$ as required by this lemma. Since the while-loop on line 2 halves the subcube Q in every iteration and it terminantes with a subcube Q of dimension at least two, this while-loop iterates at most n times. A single iteration of the while-loop can be implemented so that whole set S is processes $2\dim(Q)$-times to enumerate the number of vertices of S in every $Q^{d,i}$. Note that the minimality of $|S \cap V(Q^{d,i})|$ ensures that both $S \cap V((Q^{d,1-i})^{d',0})$ and $S \cap V((Q^{d,1-i})^{d',1})$ are non-empty. The remaining lines of Function `find_four_vertices` are trivial. Hence, the complexity of Function `find_four_vertices` is $\mathcal{O}(|S| \cdot \mathsf{poly}(n))$. ☐

We say that an edge uv of $K(Q_n)$ crosses a coordinate d if $d \in u \triangle v$.

3 Extending perfect matching without using long edges

In this section, we present an alternative proof of Kreweras' conjecture which does not use long edges of $K(Q_n)$. Methods presented in this sections are used in algorithms described in the following sections. The idea of our proof is based on Theorem 2.4, however, we present another proof. Gregor [11] used an induction on the number of subcubes in $\mathcal{C}$ and for the base case $|\mathcal{C}| = 1$ he applied [7]. In our approach, subcubes of $\mathcal{C}$ of dimension at least two are iteratively split into subcubes. Once dimensions of all subcubes of $\mathcal{C}$ are one, edges of all subcubes form a perfect matching R which extends a given perfect matching P into a Hamiltonian cycle of Q_n. As Theorem 2.4 states, we have to split subcubes of $\mathcal{C}$ so that $I(\mathcal{C}, P)$ remains connected. The splitting is described in the following two lemmas which hold in both cases where P is a perfect matching of $K(Q_n)$ and P is a perfect matching of Q_n.

Lemma 3.1. *Let P be a perfect matching of $K(\mathcal{C})$ and uv be an edge of P such that u and v are vertices of the same subcube Q of $\mathcal{C}$ of dimension at least two. Let d be an arbitrary coordinate of $u \triangle v$ which splits Q into $Q_{d,u}$ and $Q_{d,v} = Q'_{d,u}$. Let $\mathcal{C}' = \mathcal{C} \setminus \{Q\} \cup \{Q_{d,u}, Q_{d,v}\}$. If $I(\mathcal{C}, P)$ is connected, then $I(\mathcal{C}', P)$ is connected.*

Proof. Subcubes $Q_{d,u}$ and $Q_{d,v}$ belong to the same components of $I(\mathcal{C}', P)$ because of the edge uv. Furthermore, paths in $I(\mathcal{C}, P)$ terminating at Q remain paths in $I(\mathcal{C}', P)$ terminating at $Q_{d,u}$ or $Q_{d,v}$. By Lemma 2.1, the graph $I(\mathcal{C}', P)$ remains connected if $I(\mathcal{C}, P)$ is connected. □

Lemma 3.2. *Let P be a perfect matching of $K(\mathcal{C})$ and uu^P be an edge of P such that u is a vertex of a subcube Q of $\mathcal{C}$ and u^P is a vertex of another subcube Q' of $\mathcal{C}$. Then there exists another edge $vv^P \in P$ such that v is a vertex of Q and v^P is a vertex of a subcube that belongs in the same component of $I(\mathcal{C} \setminus \{Q\}, P)$ as Q'. Moreover, assume that $\dim(Q) \geq 2$ and let d be an arbitrary coordinate of $u \triangle v$ and let $\mathcal{C}' = \mathcal{C} \setminus \{Q\} \cup \left\{Q_{d,u}, Q'_{d,u}\right\}$. If $I(\mathcal{C}, P)$ is connected, then $I(\mathcal{C}', P)$ is connected.*

Proof. First, we prove the existence of the edge vv^P. Let B be the component of $I(\mathcal{C}, P) \setminus \{Q\}$ containing Q'. Let F be the subset of edges of P having exactly one endvertex in B. Since every subcube of $\mathcal{C}$ contains an even number of vertices of Q_n, the total number of vertices of Q_n contained in all subcubes of the component B is also even. Furthermore, P is a perfect matching and $P \cap E(B)$ covers an even number of vertices which implies that $|F|$ is even. Since F contains uu^P, the size of F is at least two, and so F contains another edge vv^P. Since B is a component of $I(\mathcal{C}, P) \setminus \{Q\}$, every edge of F has one endvertex in Q, so v is the endvertex in Q.

Next, we prove the connectivity of $I(\mathcal{C}', P)$ assuming $I(\mathcal{C}, P)$ is connected. Edges uu^P and vv^P and the component B guarantee that $Q_{u,d}$ and $Q_{v,d}$ belong into the same component of $I(\mathcal{C}', P)$. Furthermore, paths in $I(\mathcal{C}, P)$ terminating at Q remain paths in $I(\mathcal{C}', P)$ terminating at $Q_{d,u}$ or $Q_{d,v}$. By Lemma 2.1, the graph $I(\mathcal{C}', P)$ remains connected if $I(\mathcal{C}, P)$ is connected. □

In order to find a Hamiltonian cycle of Q_n extending a given perfect matching P of Q_n where $n \geq 2$, we can start by setting $\mathcal{C} = \{Q_n\}$. Then iteratively choose a subcube Q of $\mathcal{C}$ of dimension at least two and a vertex u of Q, and split Q into two subcubes using Lemma 3.1 or Lemma 3.2 depending on whether u^P belongs to Q. At the end, all subcubes of $\mathcal{C}$ has dimension one and the following lemma applies.

Lemma 3.3. *Assume that $\mathcal{C}$ contains at least two subcubes and $\mathcal{C}$ contains subcubes of dimension one only. Let P be a perfect matching of $K(\mathcal{C})$ and let R be the set of all edges of all subcubes of $G(\mathcal{C})$. If $I(\mathcal{C}, P)$ is connected, then $P \cup R$ is a Hamiltonian cycle of $K(\mathcal{C})$.*

Proof. Since $\mathcal{C}$ is a set of vertex disjoint subcubes of Q_n of dimension one, every vertex of $\mathcal{C}$ is covered by R exactly once and so R is a perfect matching of $G(\mathcal{C})$. Every vertex of $\mathcal{C}$ has degree two in $P \cup R$, so $P \cup R$ is a union of disjoint cycles. The graph $I(\mathcal{C}, P)$ can be obtained from $(V(\mathcal{C}), P \cup R)$ by contracting every subcube of $\mathcal{C}$ into a single vertex. Therefore, we have a bijection between components of $I(\mathcal{C}, P)$ and components of $(V(\mathcal{C}), P \cup R)$. Since $I(\mathcal{C}, P)$ is connected, $(V(\mathcal{C}), P \cup R)$ is connected and $P \cup R$ forms a single Hamiltonian cycle. □

Lemma 3.3 completes the proof of Theorem 2.4. Note that Lemmas 3.1, 3.2 and 3.3 do not use long edges of Q_n, if a given perfect matching P contains only short edges.

A slightly modified approach is used in both algorithms presented in the following sections. The basic difference is that if $\mathcal{C}$ contains a subcube Q of dimension 1 on vertices u and v, then we can add the edge uv to R and remove Q from $\mathcal{C}$ without waiting until all subcubes of $\mathcal{C}$ have dimension one. However, we have to modify edges of P and use long edges in this case. Note that the following lemma requires $|V(\mathcal{C})| \geq 6$ only for formal purposes to prevent the possibility that $K(\mathcal{C}')$ contains at most two vertices where Hamiltonian cycles may not be well defined.

Lemma 3.4. *Assume that $V(\mathcal{C})$ contains at least six vertices and $I(\mathcal{C}, P)$ is connected. Let P be a perfect matching of $K(Q_n)$ and Q be a subcube of $\mathcal{C}$ of dimension one on vertices u and v. Let $P' = P \setminus \{uu^P, vv^P\} \cup \{u^P v^P\}$ and $\mathcal{C}' = \mathcal{C} \setminus \{Q\}$. Then P' is a perfect matching of $K(\mathcal{C}')$ and $I(\mathcal{C}', P')$ is also connected. Moreover, if R' is a perfect matching of $G(\mathcal{C}')$ extending P' to a Hamiltonian cycle of $K(\mathcal{C}')$, then $R = R' \cup \{uv\}$ is a perfect matching of $G(\mathcal{C})$ extending P to a Hamiltonian cycle of $K(\mathcal{C})$.*

Proof. The edge uv is not an edge of P, otherwise Q is an isolated vertex of $I(\mathcal{C}, P)$. Hence, P' is a perfect matching of $G(\mathcal{C}')$. Using the same parity argument as in Lemma 3.2, vertices u^P and v^P belong to the same component of $I(\mathcal{C}, P) \setminus \{Q\}$, so $I(\mathcal{C}, P) \setminus \{Q\}$ is connected if $I(\mathcal{C}, P)$ is connected. The graph $I(\mathcal{C}', P')$ is also connected, because the graph $I(\mathcal{C}', P')$ differ from $I(\mathcal{C}, P) \setminus \{Q\}$ at most by adding one new edge $u^P v^P$. The Hamiltonian cycle $P \cup R$ of $K(\mathcal{C})$ is obtained from the Hamiltonian cycle $P' \cup R'$ of $K(\mathcal{C}')$ by replacing the edge $u^P v^P$ by the path on vertices u^P, u, v and v^P. □

Lemmas 3.1, 3.2 and 3.4 give another proof of Kreweras' conjecture. Note that we can repeatedly apply Lemma 3.4 until $|V(\mathcal{C})| \geq 6$. Once $\mathcal{C}$ consists of two subcubes of dimension one, P consists of two edges between subcubes of $\mathcal{C}$, so the resulting perfect matching R consists of both edges of subcubes of $\mathcal{C}$ and $P \cup R$ is a Hamiltonian cycle on four vertices of $V(\mathcal{C})$.

4 Off-line algorithm

In this section, we describe Algorithm 4.1 which for a given perfect matching P of $K(Q_n)$ finds a perfect matching R of Q_n such that $P \cup R$ is a Hamiltonian cycle of $K(Q_n)$ in linear time. Algorithm 4.1 uses a recursive function `extend` and three global arrays: a given perfect matching P, an extending perfect matching R, and an auxiliary array $A[u]$ for every vertex u of Q_n so that $A[u]$ is another vertex of Q_n. The array A is the matching obtained from P by updates described in Lemma 3.4. After the initialization, R is empty and $A[u] = u^P$ for every $u \in V(Q_n)$. To simplify the notation, A denotes both an array and a matching so that $A[u] = u^A$. In order to analyze the connectivity of the interconnection graph, Algorithm 4.1 also provides in comments the set of subcubes $\mathcal{C}$ consisting of vertices which have not been covered by R yet and a matching $Z \subseteq A$ such that $I(\mathcal{C}, Z)$ is connected, so Z is used to prove that $I(\mathcal{C}, A)$ is connected. Clearly, the initialization is $\mathcal{C} = \{Q_n\}$ and $Z = \emptyset$, so $Z \subseteq A$ and $I(\mathcal{C}, Z)$ is connected.

The recursive function `extend(Q, u)` splits Q to two subcubes using Lemmas 3.1 or 3.2 and recursively calls `extend` on both subcubes. Once the recursion reaches a subcube of dimension one, a function `join` is called to apply Lemma 3.4. Hence, $\mathcal{C}$ contains for every dimension d except one at most one subcube of dimension d, and for the exceptional dimension $\mathcal{C}$ contains at most two subcubes and this exceptional dimension is the smallest dimension of all subcubes of $\mathcal{C}$. The following lemma proves that graph $I(\mathcal{C}, Z)$ always forms a tree and we define that the root of that tree is the subcube of $\mathcal{C}$ with the largest dimension.

Lemma 4.1. *Algorithms 4.1 satisfies the following statements.*

(1) The graph $I(\mathcal{C}, Z)$ always forms a tree and the dimension of every subcube of $I(\mathcal{C}, Z)$ is at most the dimension of its parent and $Z \subseteq A$.

(2) Whenever the recursive function `extend(Q,u)` is called, Q is a subcube of $\mathcal{C}$ of the smallest dimension and Q is a leaf of $I(\mathcal{C}, Z)$ and

(3) u is a vertex of Q and

(4) if $|\mathcal{C}| \geq 2$, then $uu^A \in Z$.

Algorithm 4.1: Off-line algorithm

Input: Perfect matching P of hypercube $K(Q_n)$

9 $R := \emptyset$
10 **for** *each* uv *in* P **do**
11     $A[u] := v$
12     $A[v] := u$
    `//` $\mathcal{C} := \{Q\}$ `and` $Z := \emptyset$
13 Let u be an arbitrary vertex of Q_n
14 `extend(`Q_n`,` u`)`

Output: Perfect matching R of Q_n

15 **Function** `join(`u`,`v`)`
    `// Lemma 3.4`
16   Insert the edge uv to R
17     $A[A[u]] := A[v]$
18     $A[A[v]] := A[u]$

19 **Function** `extend(`Q`,` u`)`
20   Choose $v \in V(Q)$ arbitrarily (different from u)
21   **if** *the dimension of* Q *is 1* **then**
        `// Lemma 3.4:` $\mathcal{C} := \mathcal{C} \setminus \{Q\}$ `and` $Z := Z \setminus \left\{uu^A\right\}$
22   `join(`u, v`)`
23   **else**
24   **if** $A[v] \notin V(Q)$ **then**
            `// Split by Lemma 3.2`
25   Choose $d \in v \triangle u$ arbitrarily
26   **else**
            `// Split by Lemma 3.1`
27   Choose $d \in v \triangle A[v]$ arbitrarily
28   **if** $v \in V(Q_{d,u})$ **then**
29                 $v := A[v]$
    `// Split` Q`:` $\mathcal{C} := \mathcal{C} \setminus \{Q\} \cup \left\{Q'_{d,u}, Q_{d,u}\right\}$ `and` $Z := Z \cup \left\{vv^A\right\}$
30   `extend(`$Q'_{d,u}$`,` v`)`
31   `extend(`$Q_{d,u}$`,` u`)`

Proof. After the initialization, $I(\mathcal{C}, Z)$ is an isolated vertex, so (1) is satisfied and the initial call `extend(`Q_n`,` u`)` satisfies all conditions. When `extend(`Q`,` u`)` is applied to a subcube Q of dimension one (see line 21 of Algorithm 4.1), then Q is a leaf of $I(\mathcal{C}, Z)$ by (2) and Q with uu^A is removed from $I(\mathcal{C}, Z)$, so (1) is satisfied.

When `extend(`Q`,` u`)` is called with a subcube Q of dimension at least two (line 23), Algorithm 4.1 chooses a vertex v of Q different from u. If $v^A \notin V(Q)$ (line 24), then we split Q by an arbitrary coordinate $d \in u \triangle v$ using Lemma 3.2. Since Q is a leaf of $I(\mathcal{C}, Z)$, the subcube $Q_{d,u}$ replaces Q in $I(\mathcal{C}', Z')$ and $Q'_{d,u}$ is a new leaf of $I(\mathcal{C}', Z')$ connected by the edge vv^A, so (1) is satisfied for $I(\mathcal{C}', Z')$ where $\mathcal{C}' = \mathcal{C} \setminus \{Q\} \cup \left\{Q'_{d,u}, Q_{d,u}\right\}$ and $Z' = Z \cup \left\{vv^A\right\}$. Furthermore, when `extend(`$Q'_{d,u}$`,` v`)` is called, (2), (3), (4) are satisfied. During the evaluation of `extend(`$Q'_{d,u}$`,` v`)`, the subcube $Q_{d,v}$ is removed from $I(\mathcal{C}', Z')$ while the rest of the graph $I(\mathcal{C}', Z')$ is untouched. Hence, $I(\mathcal{C}'', Z'') = I(\mathcal{C}', Z') \setminus \{Q_{d,v}\}$, where $\mathcal{C}''$ and Z'' are the corresponding sets $\mathcal{C}$ and Z when `extend(`$Q'_{d,u}$`,` v`)` is evaluated. So, $Q_{d,u}$ is a leaf in $I(\mathcal{C}'', Z'')$, and (2), (3), (4) are satisfied when `extend(`$Q_{d,u}$`,` u`)` is called.

If $v^A \in V(Q)$ (line 26), then we split Q by an arbitrary coordinate $d \in v \triangle v^A$ using Lemma 3.1. Clearly, either $v^A \in V(Q_{d,u})$ or $v \in V(Q_{d,u})$ and we set $v := v^A$ in the latter case. So, $Q_{d,u}$ replaces Q in $I(\mathcal{C}', Z')$ and $Q'_{d,u}$ is a leaf of $I(\mathcal{C}', Z')$ and the edge uu^A connects $Q_{d,u}$ and $Q'_{d,u}$. All condition are satisfied to call `extend(`$Q'_{d,u}$`, `v`)` which removes $Q'_{d,u}$ from $\mathcal{C}'$ so $Q_{d,u}$ is a leaf when `extend(`$Q_{d,u}$`, `u`)` is called. □

Theorem 4.2. *Algorithm 4.1 finds a perfect matching R of Q_n for a given perfect matching P of $K(Q_n)$ such that $P \cup R$ is a Hamiltonian cycle of $K(Q_n)$. The time complexity is linear in the number of vertices of the hypercube Q_n, assuming a single vertex can be stored in a single word of a RAM machine.*

Proof. Lemma 4.1 proves that $I(\mathcal{C}, Z)$ is a tree and $Z \subseteq A$, so $I(\mathcal{C}, A)$ is always connected. The function `join` fulfills Lemma 3.4. When $\mathcal{C}$ contains only one subcube Q of dimension 1, then Function `extend` splits Q into $Q^{d,0}$ and $Q^{d,1}$ so that both edges of A contain one endvertex in $Q^{d,0}$ and the other endvertex in $Q^{d,1}$ and Function `join` adds the only edge of $Q^{d,0}$ and the only edge of $Q^{d,1}$ into R. Hence, $P \cup R$ is a Hamiltonian cycle of $K(Q_n)$. Time complexity follows from the fact that the function `extend` is called $(2^n - 1)$-times and the function `join` is called 2^{n-1}-times and a single evaluation of both functions (without recursion) requires $\Theta(1)$ operations. □

5 On-line algorithm

In this section, we present an on-line algorithm for Problem 1.2; see Algorithm 5.1. The basic idea of the algorithm is simple (see Function `online`): for a given vertex u, the on-line algorithm repeatedly splits the subcube $Q \in \mathcal{C}$ containing u using a variant of Lemmas 3.1 and 3.2 until the dimension of Q is one. Then the only edge uv of Q is the resulting uv edge required by Problem 1.2.

Similarly as in the off-line algorithm, we denote by A the perfect matching of $K(\mathcal{C})$ that has to be extended to a Hamiltonian cycle of $K(\mathcal{C})$, so A is initialized by P and then modified by a variant of Lemma 3.4. However, the initialization $A := P$ would take exponential time in n, so the on-line algorithm uses a data structure (e.g. a trie) which for a vertex u of $V(\mathcal{C})$ stores the incident edge uv of A only if $u^P \neq u^A$. Hence, this trie is empty in the beginning and so it can be initialized in constant time. In order to keep the notation consistent, we use the matching notation u^A, although the algorithm has to first search the trie to find the matching edge uu^A, and then call the oracle if the edge uu^A is not found in the trie (see Function `A`). Furthermore, changing the matching A means an appropriate modification of the trie (see Function `online`).

Again, we use a set of edges $Z \subseteq A$ to guarantee the connectivity of the interconnection graph as required in Theorem 2.4. However, the connectivity of $I(\mathcal{C}, Z)$ is insufficient in this section, so we keep $I(\mathcal{C}, Z)$ 2-edge-connected. Recall that $I(\mathcal{C}, A)$ is a multigraph which preserves the multiplicity of edges between subcubes which is important in this section. The interested reader can observe that $I(\mathcal{C}, A)$ is connected if and only if $I(\mathcal{C}, A)$ is 2-edge-connected, so e.g. Theorem 2.4 holds if the connectivity of $I(\mathcal{C}, P)$ is replaced by the 2-edge-connectivity of $I(\mathcal{C}, P)$. Lemma 5.1 is a modification of Lemma 3.4 for 2-edge-connected multigraphs.

Lemma 5.1. *Assume that $V(\mathcal{C})$ contains at least six vertices. Let A be a perfect matching of $K(\mathcal{C})$ and let $Z \subseteq A$ and let Q be a subcube of $\mathcal{C}$ of dimension one on vertices u and v. Assume that $I(\mathcal{C}, Z)$ is 2-edge-connected. Let $A' = A \setminus \{uu^A, vv^A\} \cup \{u^A v^A\}$ and $Z' = Z \setminus \{uu^A, vv^A\} \cup \{u^A v^A\}$ and $\mathcal{C}' = \mathcal{C} \setminus \{Q\}$. Then $Z' \subseteq A'$ and A' is a perfect matching of $K(\mathcal{C}')$ and $I(\mathcal{C}', Z')$ is 2-edge-connected. Furthermore, if R' is a perfect matching of $G(\mathcal{C}')$ extending A' to a Hamiltonian cycle of $K(\mathcal{C}')$, then $R = R' \cup \{uv\}$ is a perfect matching of $G(\mathcal{C})$ extending A to a Hamiltonian cycle of $K(\mathcal{C})$.*

Proof. It suffices to prove that $I(\mathcal{C}', Z')$ is 2-edge-connected since the remaining statements of this lemma follow from Lemma 3.4. From the 2-edge-connectivity of $I(\mathcal{C}, Z)$ it follows that $uu^A, vv^A \in Z$. Every path in $I(\mathcal{C}, Z)$ with endvertices different from Q is a path in $I(\mathcal{C}', Z')$ or it can be

Algorithm 5.1: On-line algorithm (part 1 of 3)

```
Function find_hamiltonian_cycle(oracle)
    Initialize an empty trie
    Z := ∅
    R := ∅
    C := {Q_n}
    Choose a starting vertex u
    for i := 1 to 2^{n-1} do
        v := online(u)
        w := oracle(v)
        Output: uv, vw
        u := w
Function online (u)
    if u is covered by R then
        return u^R
    else
        while the subcube Q containing u has dimension at least two do
            split(Q)
        // Lemma 5.1
        Let v be the other vertex of Q
        u^A := A(u)
        v^A := A(v)
        Z := Z \ {uu^A, vv^A} ∪ {u^A v^A}
        C := C \ {Q}
        R := R ∪ {uv}
        Insert the key u with the value v^A into the trie
        Insert the key v with the value u^A into the trie
        return v
Function A (vertex u)
    if u is stored in the trie then
        return find the vertex u^A in the trie
    else
        return call the oracle to find u^P
```

shortened to a path in $I(\mathcal{C}', Z')$ using the edge $u^A v^A$. Hence, $I(\mathcal{C}', Z')$ is 2-edge-connected by Lemma 2.2. □

Now, we show how to split a subcube of $\mathcal{C}$ and preserve 2-edge-connectivity of the interconnection graph.

Lemma 5.2. *Let $Q \in \mathcal{C}$ be a subcube of dimension at least 2 and Z be a matching of $K(\mathcal{C})$. Let Q be split into $Q^{d,0}$ and $Q^{d,1}$ by a coordinate d and let $F \subseteq E(K(\mathcal{C}'))$ be a set of edges forming two edge-disjoint paths between $Q^{d,0}$ and $Q^{d,1}$ in $I(\mathcal{C}', F)$ where $\mathcal{C}' = \mathcal{C} \setminus \{Q\} \cup \{Q^{d,0}, Q^{d,1}\}$. If $I(\mathcal{C}, Z)$ is 2-edge-connected, then also $I(\mathcal{C}', Z')$ is 2-edge-connected where $Z' = Z \cup F$.*

Proof. For a contradiction, we assume that $I(\mathcal{C}', Z')$ is not 2-edge-connected. Since $Q^{d,0}$ and $Q^{d,1}$ are connected by two edge-disjoint paths in $I(\mathcal{C}', Z')$, Lemma 2.2 implies that $Q^{d,0}$ and $Q^{d,1}$ belong to the same 2-edge-connected component of $I(\mathcal{C}', Z')$, say B. Let B' be another 2-edge-connected component of $I(\mathcal{C}', Z')$. Since $I(\mathcal{C}, Z)$ is 2-edge-connected, there are two edge-disjoint paths P_1 and P_2, each joining a vertex of B with a vertex of B' and assume that P_1 and P_2 have the shortest

Algorithm 5.2: On-line algorithm (part 2 of 3)

62 **Function** `find_Si_and_S'` (Q)
  // Find sets $S_1, \dots, S_b$ and S' satisfying (1) or (3) in Lemma 5.4.
63 Let $B_1, \dots, B_b$ be the set of all components of $I(\mathcal{C}, Z) \setminus \{Q\}$
64 **for** $i := 1$ **to** b **do**
65   Let S_i be the set of all vertices v of Q such that $vv^A \in Z$ and v^A is a vertex of a subcube of the component B_i
66 Let S' be the set of edges of Z having both endvertices in Q
67 **while** $2|S'| + \sum_{i=1}^{b} |S_i| < 2\dim(Q) + 2$ **do**
68   Choose a vertex v of Q which is not contained in any $S_1, \dots, S_b$ and also v is not covered by S'
69   $v^A :=$ `A`(v)
70   **if** *v^A is a vertex of Q* **then**
71   Insert vv^A into S'
72   **else**
73   **for** $i := 1$ **to** b **do**
74   **if** *v^A is a vertex of a subcube of B_i* **then**
75   Insert v into S_i
76 **return** $(S_1, \dots, S_b, S')$

possible length. Then both P_1 and P_2 contain at most one vertex of B. If Q is an endvertex of P_1 or P_2, we replace Q by $Q^{d,0}$ or $Q^{d,1}$ to obtain two edge-disjoint paths between B and B' which contradicts the assumption that B and B' are distinct 2-edge-connected components. □

The following lemma presents a constructive proof to split a given subcube and this approach is summarized in Functions `find_four_vertices` and `split`.

Lemma 5.3. *Let A be a perfect matching of $K(\mathcal{C})$ and $Z \subseteq A$ and $Q \in \mathcal{C}$ be a subcube of dimension at least two and $I(\mathcal{C}, Z)$ be 2-edge-connected. Then Q can be split into $Q^{d,0}$ and $Q^{d,1}$ by a coordinate d and there exists a set of edges Z' such that $Z \subseteq Z' \subseteq A$ and $|Z'| \leq |Z| + 4$ and $I(\mathcal{C}', Z')$ is 2-edge-connected where $\mathcal{C}' = \mathcal{C} \setminus \{Q\} \cup \{Q^{d,0}, Q^{d,1}\}$. Moreover, both the splitting coordinate d and the set Z' can be found in time $\mathcal{O}((|\mathcal{C}| + |Z|) \cdot \mathsf{poly}(n))$.*

Proof. Observe that the statement holds if $\dim(Q) = 2$ so we assume that $\dim(Q) \geq 3$. Let $B_1, \dots, B_b$ be the set of all components of $I(\mathcal{C}, Z) \setminus \{Q\}$. Let S_i be the set of all vertices u of Q such that $uu^A \in Z$ and u^A is a vertex of a subcube of the component B_i (line 65). Since $I(\mathcal{C}, Z)$ is 2-edge-connected, Lemma 2.2 implies that $|S_i| \geq 2$. Furthermore, let S' be the set of edges of Z having both endvertices in Q (line 66). Note that Function `find_four_vertices` enlarges sets $B_1, \dots, B_b$ and S', so the above definitions are only initial values and these sets always satisfy assumptions of the following lemma.

Lemma 5.4. *Let S_i be a subset of vertices of Q and S' be a subset of $E(K(Q)) \cap A$ where $i = 1, \dots, b$. Assume that u^A is a vertex of B_i for every vertex u of S_i. Then, at least one of the following conditions is satisfied.*

(1) There exists a component B_i with $|\mathcal{D}(S_i)| < |S_i| - 1$.

(2) There exists a vertex v of Q which is not contained in any of $S_1, \dots, S_b$ and also v is not covered by S' and $2|S'| + \sum_{i=1}^{b} |S_i| < 2\dim(Q) + 2$.

(3) It holds that $\sum_{uv \in S'} |u \triangle v| + \sum_{i=1}^{b} |\mathcal{D}(S_i)| \geq \dim(Q) + 1$.

Algorithm 5.3: On-line algorithm (part 3 of 3)

77 **Function** `split` (Q)
78 $(S_1, \dots, S_b, S') :=$ `find_Si_and_S'`(Q)
79 **if** *there exists a component* B_i *with* $|\mathcal{D}(S_i)| < |S_i| - 1$ **then**
80 $(u_1, u_2, u_3, u_4) :=$ `find_four_vertices`(Q, S_i)
81 $Z := Z \cup \{u_1\mathtt{A}(u_1), u_2\mathtt{A}(u_2), u_3\mathtt{A}(u_3), u_4\mathtt{A}(u_4)\}$
82 Let P_1 and P_2 be two edge-disjoint paths P_1 and P_2 of the graph $(V(\mathcal{C}), Z)$ with endvertices u_1, u_2, u_3 and u_4 by Lemma 2.3.
83 Let $d \in \mathrm{coord}(Q)$ such that both P_1 and P_2 are path between $Q^{d,0}$ and $Q^{d,1}$
84 **else**
85 Initialize an array of integers h of length n by 0
86 **for** $uv \in S'$ **do**
87 **for** $d \in u \triangle v$ **do**
88 $h[d] := h[d] + 1$
89 **for** $i := 1$ **to** b **do**
90 **for** $d \in \mathcal{D}(S_i)$ **do**
91 $h[d] := h[d] + 1$
92 Let d be a coordinate such that $h[d] \geq 2$
93 paths $:= 0$
94 **for** $uv \in S'$ **do**
95 **if** $d \in u \triangle v$ *and paths* < 2 **then**
96 $Z := Z \cup \{uv\}$
97 paths $:=$ paths $+ 1$
98 **for** $i := 1$ **to** b **do**
99 **if** $d \in \mathcal{D}(S_i)$ *and paths* < 2 **then**
100 Find a vertex of u of $S_i \cap V(Q^{d,0})$ and a vertex v of $S_i \cap V(Q^{d,1})$
101 $Z := Z \cup \{u\mathtt{A}(u), v\mathtt{A}(v)\}$
102 paths $:=$ paths $+ 1$
103 $\mathcal{C} := \mathcal{C} \setminus \{Q\} \cup \{Q^{d,0}, Q^{d,1}\}$

Proof. We assume that (1) does not hold. Then

$$\sum_{uv \in S'} |u \triangle v| + \sum_{i=1}^{b} |\mathcal{D}(S_i)| \geq |S'| + \sum_{i=1}^{b} (|S_i| - 1) \geq |S'| + \sum_{i=1}^{b} \frac{|S_i|}{2}$$

since $|S_i| \geq 2$ and $|u \triangle v| \geq 1$. If (3) also does not hold, then $2|S'| + \sum_{i=1}^{b} |S_i| < 2\dim(Q) + 2 \leq 2^{\dim(Q)}$ where the second inequality follows from the assumption that $\dim(Q) \geq 3$. Since every edge of S' covers two vertices and the total number of vertices of Q is $2^{\dim(Q)}$, there exists a vertex of Q not counted in $2|S'| + \sum_{i=1}^{b} |S_i|$ which implies that (2) holds. $\square$

Now, we continue in the proof of Lemma 5.3. When (2) is satisfied (line 67), there exists a vertex v of Q which is not contained in any of $S_1, \dots, S_b$ and also v is not covered by S'. Function `find_four_vertices` iteratively inserts v into S_i if v^A is a vertex of B_i for some i, or it inserts vv^A into S' if v^A is a vertex of Q. After at most $2\dim(Q) + 2$ iterations, (2) fails. Note that assumptions of Lemma 5.4 are still satisfied, so (1) or (3) hold.

If (1) holds (line 79), then let B_i be a component with $|\mathcal{D}(S_i)| < |S_i| - 1$. By Lemma 2.5 Q can be split by two coordinates d' and d'' into four subcubes so that each of them contains at least one vertex of S_i, so let u_1, u_2, u_3, u_4 be the four vertices of S_i in the four subcubes on Q. By Lemma 2.3, there exist two edge-disjoint paths in B_i with endvertices $u_1^A, u_2^A, u_3^A, u_4^A$ and these paths can be extended by edges $u_1u_1^A, u_2u_2^A, u_3u_3^A, u_4u_4^A$ into two edge-disjoint paths P_1 and P_2 with endvertices u_1, u_2, u_3, u_4. For at least one coordinate $d \in \{d', d''\}$ it holds that P_1 and P_2 form two edge-disjoint paths between $Q^{d,0}$ and $Q^{d,1}$ in $I(\mathcal{C}', Z')$ where $Z' = Z \cup \{u_1u_1^A, u_2u_2^A, u_3u_3^A, u_4u_4^A\}$ and $\mathcal{C}' = \mathcal{C} \setminus \{Q\} \cup \{Q^{d,0}, Q^{d,1}\}$. By Lemma 5.2, the graph $I(\mathcal{C}', Z')$ is 2-edge-connected.

Every edge $uv \in S'$ contributes to the sum $\sum_{uv\in S'} |u\triangle v| + \sum_{i=1}^{b} |\mathcal{D}(S_i)|$ of (3) by $|u\triangle v|$ coordinates (line 88) and every component S_i contributes by $|\mathcal{D}(S_i)|$ coordinates (line 91). If (3) holds (line 84), then there exists a coordinate d which contributes to the sum $\sum_{uv\in S'} |u\triangle v| + \sum_{i=1}^{b} |\mathcal{D}(S_i)|$ by at least 2 (line 92), that is

- S' contains two edges crossing the coordinate d, or
- S' contains an edge crossing the coordinate d and there exists a component B_i and a path in $I(\mathcal{C}', Z')$ between $Q^{d,0}$ and $Q^{d,1}$ passing through B_i, or
- there exists two components B_i and B_j and there exists two edge-disjoint paths P_1 and P_2 in $I(\mathcal{C}', Z')$ between $Q^{d,0}$ and $Q^{d,1}$ passing through B_i and B_j, respectively,

where Z' is obtained from Z by inserting at most four appropriate edges of A (lines 96 and 101). In all cases, the graph $I(\mathcal{C}', Z')$ is 2-edge-connected by Lemma 5.2.

All components can be found in $\mathcal{O}((|\mathcal{C}| + |Z|) \cdot poly(n))$. Then it suffices to analyze at most $2\dim(Q) + 2$ vertices of Q to ensure that (2) does not hold and this is done in time $\mathsf{poly}(n)$. Next, we determine whether (3) holds in time $\mathsf{poly}(n)$ and finally apply (3) or (1) to split Q in time $\mathcal{O}((|\mathcal{C}| + |Z|) \cdot poly(n))$. □

One iteration of the on-line algorithm splits some subcube at most n-times, so $|\mathcal{C}| \leq kn$ and $|Z| \leq 4kn$ after k iterations. Time complexity of k-th iteration is $\mathcal{O}(k \cdot poly(n))$ which implies the following theorem.

Theorem 5.5. *Described algorithm finds a Hamiltonian cycle of Q_n extending a given perfect matching as Problem 1.2 requires and the complexity of finding first k edges is $\mathcal{O}(k^2 \cdot poly(n))$.*

6 Conclusion

One downside of the on-line algorithm is that the running time of one iteration is increasing after every iteration. It may be interesting whether there exists an algorithm for Problem 1.2 with complexity polynomial only in n. Next, it would be natural to generalize these off-line and on-line algorithms for finding a Hamiltonian cycle of Q_n extending a given general (not only perfect) matching Q_n, if such Hamiltonian cycle exists. However, it is still open whether every matching of Q_n can be extended to a Hamiltonian cycle [19].

Acknowledgements. I am very grateful to Petr Gregor and Tomáš Dvořák for fruitful discussions on this topic.

References

[1] A. Alahmadi, R.E.L. Aldred, A. Alkenani, R. Hijazi, P. Solé, and C. Thomassen. Extending a perfect matching to a hamiltonian cycle. *Discrete Mathematics & Theoretical Computer Science*, 17, 2015.

[2] X.B. Chen. Unpaired many-to-many vertex-disjoint path covers of a class of bipartite graphs. *Information Processing Letters*, 110:203–205, 2010.

[3] T. Dvořák. Hamiltonian cycles with prescribed edges in hypercubes. *SIAM J. Discret. Math.*, 19(1):135–144, 2005.

[4] T. Dvořák and J. Fink. Gray codes extending quadratic matchings. Submitted, 2017.

[5] G. Ehrlich. Loopless algorithms for generating permutations, combinations, and other combinatorial configurations. *Journal of the ACM (JACM)*, 20(3):500–513, 1973.

[6] T. Feder and C. Subi. Matchings and cycle decompositions in hypercubes. Unpublished, http://theory.stanford.edu/~tomas/circmat.ps.

[7] J. Fink. Perfect matchings extend to Hamilton cycles in hypercubes. *J. Comb. Theory, Ser. B*, 97(6):1074–1076, 2007.

[8] J. Fink. Connectivity of matching graph of hypercube. *SIAM J. Discrete Math*, 23:1100–1109, 2009.

[9] J. Fink. Matching graphs of hypercubes and complete bipartite graphs. *European Journal of Combinatorics*, 30(7):1624–1629, 2009.

[10] D.G. Fon-Der-Flaass. Extending pairings to Hamiltonian cycles. *Siberian Electronic Mathematical Reports*, 7:115–118, 2010.

[11] P. Gregor. Perfect matchings extending on subcubes to Hamiltonian cycles of hypercubes. *Discrete Mathematics*, 309(6):1711–1713, 2009.

[12] P. Gregor, T. Novotný, and R. Škrekovski. Extending perfect matchings to gray codes with prescribed ends. Submitted, 2017.

[13] L. Gros. *Théorie du Baguenodier*. Aimé Vingtrinier, Lyon, 1872.

[14] D. E. Knuth. *The Art of Computer Programming, Volume 4, Fascicles 0-4*. Addison-Wesley Professional, 2009.

[15] Donald E. Knuth. Personal communication, 2008.

[16] G. Kreweras. Matchings and Hamiltonian cycles on hypercubes. *Bull. Inst. Combin. Appl.*, 16:87–91, 1996.

[17] T. Mütze. Proof of the middle levels conjecture. *Proceedings of the London Mathematical Society*, 112(4):677–713, 2016.

[18] T. Mütze and J. Nummenpalo. A constant-time algorithm for middle levels gray codes. In *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 2238–2253, 2017.

[19] F. Ruskey and C.D. Savage. Hamilton Cycles that Extend Transposition Matchings in Cayley Graphs of S_n. *SIAM Journal on Discrete Mathematics*, 6(1):152–166, 1993.

[20] C. Savage. A survey of combinatorial Gray codes. *SIAM Review*, 39(4):605–629, 1997.