Discussiones Mathematicae
Graph Theory 36 (2016) 743–758
doi:10.7151/dmgt.1888

ON PATH-PAIRABILITY IN THE CARTESIAN PRODUCT OF GRAPHS

Gábor Mészáros

Alfred Renyi Institute of Mathematics
Central European University

e-mail: meszaros.gabor@renyi.mta.hu

Abstract

We study the inheritance of path-pairability in the Cartesian product of graphs and prove additive and multiplicative inheritance patterns of path-pairability, depending on the number of vertices in the Cartesian product. We present path-pairable graph families that improve the known upper bound on the minimal maximum degree of a path-pairable graph. Further results and open questions about path-pairability are also presented.

Keywords: path-pairable graphs, Cartesian product of graphs.

2010 Mathematics Subject Classification: 05C38, 05C85.

1. Introduction

We discuss graph theoretic concepts emerging from a practical networking problem introduced by Csaba, Faudree, Gyárfás, Lehel, and Shelp [3, 4, 6, 8]. A graph G on at least $2k$ vertices is called *k-path-pairable* if, for any pair of disjoint sets of pairwise different vertices $X = \{x_1, \dots, x_k\}$ and $Y = \{y_1, \dots, y_k\}$ of G, there exist k edge-disjoint x_iy_i-paths joining the vertices. The *path-parability number* $\mathrm{pp}(G)$ of a graph G is the largest positive integer k, for which G is k-path-pairable. A graph on exactly $2k$ vertices is simply called *path-pairable*, if it is k-path-pairable. The vertices of the set $X \cup Y$ are often referred to as *terminals* while the pairs (x_i, y_i) of terminals are simply called *pairs* or *partners*.

Path-pairability is closely related to linkedness and weak-linkedness properties of graphs. A graph G is *k-linked* (*weakly k-linked*) if, for every pair of k-element sets, $X = \{x_1, \dots, x_k\}$ and $Y = \{y_1, \dots, y_k\}$, there exist internally vertex-disjoint (edge-disjoint) paths $P_1, \dots, P_k$, such that each P_i is an x_iy_i-path. Observe that k-linked graphs are weakly-k-linked and weakly-k-linked graphs are k-path-pairable. Note also that both linkedness and weak-linkedness require sufficiently high connectivity or edge-connectivity; a k-linked graph is $(2k-1)$-connected and a weakly-k-linked graph is k-edge-connected. On the other hand, path-pairability does not imply similar lower bounds on connectivity or edge-connectivity: there exist k-path-pairable graphs for arbitrary value of k that are connected but are neither 2-connected nor 2-edge-connected. The star graph $K_{1,2k-1}$ is one of the most illustrative examples of the mentioned graph family. To date the only known connectivity-related condition that must hold for every k-path-pairable graphs is the following *cut-condition*.

Definition 1 (Cut-condition)**.** In a graph G let $V(G)$ and $E(G)$ denote the set of vertices and edges, respectively. For a subset $S \subset V(G)$ let $d(S)$ denote the number of edges in $E(G)$ with exactly one endvertex in S. A graph G satisfies the k-cut-condition if, for every $S \subset V(G)$ whith $|S| \leq k$, $d(S) \leq |S|$ holds. A graph G on $2n$ vertices satisfies the cut-condition, if for every $S \subset V(G)$, $|S| \leq n$, $d(S) \leq |S|$ holds.

If G is k-path-pairable, then it satisfies the k-cut condition. Indeed, if there exist $S \subset V(G)$ that violates the condition, terminals placed at every vertex of S with their pairs in $G \backslash S$ cannot be joined without using an edge between the two sets at least twice. Note that the cut condition states a necessary but not sufficient condition for path-pairability. We present counterexamples in the additional remarks. To date there are no known conditions that imply k-path-pairability but do not imply weak-k-linkedness.

One of the main open questions concerning path-pairability of graphs is the minimal possible value of the maximum degree $\Delta(G)$ of a path-pairable graph G. Faudree, Gyárfás, and Lehel [7] gave examples of k-path-pairable graphs with maximum degree $\Delta = 3$ for arbitrary values of k. In contrast, the same authors proved in [8] that the maximum degree has to grow together with the number of vertices in path-pairable graphs. They in fact showed that a path-pairable graph with maximum degree Δ has at most $2\Delta^{\Delta}$ vertices. The result places a lower bound of $O(\frac{\log n}{\log \log n})$ on the maximum degree of a path-pairable graph on n vertices. This bound is conjectured to be asymptotically sharp, though examples of path-pairable graphs with maximum degree of the right order of magnitude have yet to be explored. The best known constructions are due to Kubicka, Kubicki and Lehel [9] as well as Mészáros [11] and have maximum degree of order of magnitude $O(\sqrt{n})$. The construction in [9] is obtained by taking the Cartesian product of two complete graphs. That motivated the author of this present paper to study path-pairability in the Cartesian product in more details.

The *Cartesian product* of graphs G and H is the graph $G \square H$ with vertices $V(G \square H) = V(G) \times V(H)$, and $(x;u)(y;v)$ is an edge, if $x = y$ and $uv \in E(H)$ or $xy \in E(G)$ and $u = v$. The Cartesian product of graphs has been extensively studied in the past decades. It gave rise to important classes of graphs;

for example, the n-dimensional grid can be considered as the Cartesian product of lower dimensional grids. Hypercubes are well known members of this family with similar recursive structures: the Cartesian product of m-dimensional and n-dimensional hypercubes is an $(m+n)$-dimensional one. The study of graph products leads to various deep structural problems such as invariance and inheritance of graph parameters. Chiue and Shieh [2] proved that the Cartesian product of a k-connected and an l-connected graph is $(k+l)$-connected. Similar result for edge connectivity was proved by Xu and Yang [12]. Inheritance of linkedness has been investigated by Mészáros [10], who proved that the Cartesian product of an a-linked graph G and a b-linked graph H is $(a+b-1)$-linked, given that the graphs are sufficiently large in terms of a and b.

This paper has two main objectives. We prove an inheritance theorem of path-pairability (Theorem 2) that is similar to the inheritance of linkedness presented in [10]. We also prove an extension of Theorem 2, which states that, given sufficient space in the product graph, reasonably higher path-pairability can be achieved (Theorem 4). We mention that neither linkedness, nor weak-linkedness share the latter property. Both results are sharp up to constant factors.

Theorem 2 and 4 concern themselves with path-pairability of the product graph $G\square H$, where path-pairability in the factors G and H is conveniently small compared to $|V(G)|$ and $|V(H)|$. Our other objective is the examination toward the other extremity, when $\mathrm{pp}(G)$ and $\mathrm{pp}(H)$ are as large as possible, that is, both G and H are path-pairable. The ultimate goal would be to find sufficient conditions that guarantee path-pairability of the product graph and offer valuable tools to generate new path-pairable graph families. Note that the Cartesian product of two path-pairable graphs is not necessarily path-pairable. A counterexample is presented in Proposition 3.

Kubicka, Kubicki, and Lehel [9] investigated path-pairability of complete grid graphs, that is, the Cartesian product of complete graphs, and proved that the two-dimensional complete grid $K_a\square K_b$ on $n = a\cdot b$ vertices is path-pairable. Our objective is to show that the Cartesian product of the complete bipartite graph $K_{m,m}$ with itself is path-pairable for sufficiently large even values of m (Theorem 8). The examined path-pairable product has $n = 4m^2$ vertices and maximum degree $\Delta = 2m = \sqrt{n}$, which improves the previously discussed upper bound ($\approx 2\sqrt{n}$) on $\Delta(G)$. It also presents a new infinite family of path-pairable graphs, as well as gives examples of non-complete path-pairable graphs whose Cartesian product is path-pairable as well.

We follow the notation of [1]. For the sake of completeness, we recall definitions of the mainly used concepts. A G-layer G_x $(x \in V(H))$ of the Cartesian product $G\square H$ is the subgraph induced by the set of vertices $\{(u;x) : u \in V(G)\}$. An H-layer is defined analogously. We call edges of $G\square H$ lying in G-layers horizontal while edges lying in H-layers are called vertical. Unless it is misleading, we also use the notation $G_z = G_x$ and $H_z = H_y$ for layers corresponding to $z = (x; y) \in V(G\Box H)$. We also refer the reader to [1] for further details on product graphs. For a comprehensive survey of results concerning path-pairability, we refer to [5] and [8].

2. Theorem 2 and Theorem 4

Our overall pairing approach in the actual proofs as well as in the later presented techniques is building the required pairing paths in several phases. When building a path between terminals u and v, we often choose a third vertex w and construct a path uv as the union of appropriate paths uw and wv. Obviously, once an path uw is found in a graph G, finishing the pairing of u and v is equivalent to pairing w and v in G', where G' is obtained from G by deleting the egdes of the path uw. Given a triple (u, v, w) with terminal vertices u, v and a path uw, we call the operation of deleting the edges of the path and assigning w as the new pair of v a *reduction.* Throughout our proofs we accomplish the pairing of the terminals by numerous reductions. We call w the *representant* of u. Assigning a representant w to terminal u of a pair (u, v) translates as applying reduction for the triple (u, v, w) with an appropriate path uw. Representants of representants (and so on) are defined recursively.

Theorem 2. *If G is an a-path-pairable graph with $|V(G)| \geq 8a$ and H is a b-path-pairable graph with $|V(H)| \geq 8b$, then $G\Box H$ is $(a+b)$-path-pairable.*

Proof. Let M denote the set of $2(a+b)$ terminals in $G\Box H$. We may assume that $a \geq b$. We first prove the theorem in the "base" case, when no G-layer contains terminals belonging to $(a+1)$ or more pairs. This assumption implies that no G-layer contains more than $2a$ terminals. Our strategy is different for the following two cases.

Case A. We join terminals lying on the same G-layer by finding a direct path in that layer (without the use of reduction).

Case B. If terminals u and v form a pair and do not share a G-layer, we choose representants u' and v', respectively, such that $u' \in G_u$ and $v' \in G_v$. We define paths uu' and vv' and apply reductions on the triples (u, u', v) and (v, v', u). Apparently, the required path uv will be formed as the union of appropriate paths uu', $u'v'$, and $v'v$.

We describe the above steps in more details as follows. Take a G-layer G_x $(x \in H)$ with terminals $u_1, \dots, u_t$ $(1 \leq t \leq 2a)$. For a terminal u of G_x without a pair on G_x, we choose a representant $u' \in G_x$. We assign representants, for every choice of u (and/or v), such that

(i) different terminals get different representants,
(ii) layer $H_{u'}$ (or $H_{v'}$) contains no terminal,
(iii) if u and v are terminals forming a pair but not sharing a G-layer, then $H_{u'} = H_{v'}$, and conversely, if $H_{u'} = H_{v'}$, then terminals u and v form a pair.

Since $|V(G)| \geq 8a$ and G contains at most $4a$ terminals, there are at least $4a$ H-layers at our disposal that contain no terminal. The number of (u', v') pairs of representants is at most $2a$, hence we can greedily assign H-layers to them. After the reductions, every G-layer contains at most a pairs of terminals or terminal-representant pairs. Using the fact the G-layers are a-path-pairable, we can assign edge-disjoint paths joining the terminals of case A and the terminal-representant pairs (u, u') or (v, v') for case B within every one of the G-layers. Having done that, the appropriate (u', v') representants can get paired within their H-layers, given that H-layers are connected and none of their edges have been used so far. That completes the proof of the base case.

Now we turn to the examination of the general case. As $4(a+1) > 2(a+b)$, at most three G-layers contain $(a+1)$ terminals of different pairs. Our goal is to reduce our problem to the base case by redistributing the terminals among the G-layers. It will be done by assigning representants for each terminal within its original H-layer. Observe that, as the solution of the base case contains a horizontal shift, the combination of the initial redistribution, and the solution of the base case will use no vertical or horizontal edges more than once. For the redistribution of the terminals we follow a case-by-case analysis.

Case 1. Assume first that G_x is the only G-layer that contains at least $a+1$ terminals belonging to different pairs, say $u_1, \ldots, u_{a+t}$ for $1 \leq t \leq b$. Then there are at most $a + 2b - t$ terminals outside of G_x. One of the layers different from G_x contains at most $a - t$ terminals, since otherwise the graph $G\Box(H - x)$ would contain at least $(8b-1)(a-t+1) > (a+2b-t)$ terminals, a contradiction. Take a G-layer G_y with at most $a - t$ terminals. Our plan is to choose t terminals $\{\hat{u}_1, \ldots, \hat{u}_t\} \subset \{u_1, \ldots, u_{a+t}\}$ and assign them representants in G_y; the representant of $\hat{u}$ shall be $\hat{u}' = H_{\hat{u}} \cap G_y$. We choose an arbitrary $\hat{u}\hat{u}'$-path within $H_{\hat{u}}$ for the reduction. Note that if for some terminal $\hat{u}$ its partner $\hat{v}$ lies in G_x as well, we apply the same operation on $\hat{u}$.

The bottleneck of the described operation is the appropriate choice of the terminals $\{\hat{u_1}, \ldots, \hat{u_t}\}$. Note that we do not wish to assign a representant to a vertex that already contains a terminal. In other words, the vertex of the assigned representant $\hat{u_i}'$ should not contain any terminal. The terminals initially in G_y prohibit the assignment of representants for at most $a - t$ of the terminals (singleton or paired) of G_x, that is, at least $(a+t) - (a-t) = 2t$ terminals can get representants, while we only needed t. Note also that the total number of pairs in G_y is at most $(a-t) + t = a$ after the redistributing step, as prescribed in the base case. We can now apply the solution of the base case on a new set of terminals, where representants take the place of their initial terminals.

Case 2. If two G-layers contain at least $a+1$ terminals of different pairs, the remaining terminals occupy at most $2b-2$ G-layers, that is, there exists at least $6b$ G-layers that are free of terminals. If $b=1$, both layers contain exactly $a+1$ terminals of different pairs. One can arbitrarily pick a pair, shift them vertically into a G-layer completing our task just as in Case 1. If $b \geq 2$, then for a terminal u we can arbitrarily define a representant u' in H_u, such that

(a) $G_{u'}$ contains no terminal and contains at most a representants at the end of the procedure,
(b) uu' pairs are joined within $H_u = H_{u'}$ by edge-disjoint paths.

Indeed, to satisfy the first condition, observe that we have at most $2a+2b$ terminals that we distribute among $6b$ empty G-layers without any particular constraint (remember, here a terminal and its pair do not have to get representants assigned to the same G-layer), thus a balanced distribution with at most $\lceil \frac{2a+2b}{6b} \rceil \leq a$ terminals can be chosen. The second condition can be guaranteed by 2-path-pairability, as we assign at most 2 representants within an H-layer.

Case 3. The case with three overloaded layers works similarly to the previous one. Observe first that in the examined case $3(a+1) \leq 2a+2b$, hence $b \geq \frac{a+3}{2} \geq 2$. Remember that $a \geq b$, thus $a \geq \frac{a+3}{2} \Rightarrow a \geq 3$, which yields $b \geq 3$ as well. We have at least $a+6b$ empty G-layers at disposal, each of them expected to have at most $\lceil \frac{2a+2b}{6b} \rceil \leq a$ representants on average. Since $b \geq 3$ and we have at most three paths to find within every H-layer, the examination of the case and so the proof is complete. ■

Before proving Theorem 4 we give an upper bound on the additive inheritance. It verifies that the result of Theorem 2 is sharp up to a constant factor.

Proposition 3. *The Cartesian product $K_{1,a} \square K_{1,b}$ is at most $\lceil \frac{a+b}{2} \rceil$-path-pairable.*

Proof. We may assume that $a \geq b$. If $a=b=1$, the statement is straightforward. If $a \geq b \geq 2$, the product graphs has a unique vertex of degree $a+b$; let us denote it by z_{a+b}. If $a \geq 2$ and $b=1$, then the graph has exactly two vertices of degree $a+b$; choose one of them arbitrarily and denote it by z_{a+b}.

Let C and R denote the sets of vertices of degree two in an arbitrary column and in an arbitrary row not containing z_{a+b} and let y denote the intersection $C \cap R$. Moreover, let x be an additional vertex (not in $C \cup R$) of degree two. Finally, we introduce vertices z_{a+1} and z_{b+1} as the intersections of R and C with the $K_{1,b}$- and $K_{1,a}$-layers corresponding to z_{a+b}, respectively.

Our plan is to define a pairing of terminals in $T = C \cup R \cup \{x\}$. In case T contains an odd number of vertices, we add an arbitrary vertex of $G \backslash T$ to T to fix its parity. We place terminals such that vertices x and y as well as z_{a+1} and z_{b+1} form pairs; the remaining pairs of terminals in T can be placed arbitrarily.

Observe that paths that join the pairs (x, y) and (z_{a+1}, z_{b+1}) have to use either the edge between z_{a+1} and z_{a+b} or between z_{b+1} and z_{a+b} (the use of any other edge starting at the above vertices would cause collision with another terminal), hence the pairing cannot be achieved. ∎

Theorem 4. *If G is an a-path-pairable graph and H is a b-path-pairable graph and $|V(G)|, |V(H)| \geq 4s$, $s < \frac{(a+1)(b+1)}{2}$, then $G\square H$ is s-path-pairable.*

Proof. We use the same techniques as in the proof of Theorem 2. Again, we may assume $b \leq a$. If no G-layer contains more than a terminals of different pairs, we can join the pairs that share a G-layer, and assign representants to the terminals having their pairs on a different G-layer, just as we did in the base case of the previous proof. The representants can be chosen, such that

(i) their H-layers contain no terminal,
(ii) representants of a pair of terminals are located on the same H-layer, and
(iii) every H-layer contains at most b pairs of representants.

The initial terminals occupy at most $2s$ H-layers. We need an additional empty H-layer for every one of the s pairs that is guaranteed by the condition $|V(G)| \geq 4s$. Pairing of the representants can be carried out within the H-layers. Note that we are far from an optimal solution, as an H-layer is capable of joining up to b pairs of terminals, hence similar theorem with a stronger condition on the number of vertices can be proved. For the sake of convenience and clarity, we stick to the weaker variant and proceed by investigating the general case.

If the layers $G_{x_1}, \ldots, G_{x_t}$ contain more than a terminals of different pairs, then observe first that $t \leq b$, else $G\square H$ would consist of at least $(a+1)(b+1)$ terminals, contradicting $s < \frac{(a+1)(b+1)}{2}$. It means that in every H-layer that contains a terminal u, we can assign a representant u' and — using that H is b-path-pairable, and so is every H-layer in $G\square H$ — define edge disjoint paths uu' for every u. We can distribute the representants among the initially empty horizontal layers equally, such that none of them contain more than a representants. Indeed, we have at least $2s$ empty G-layers at our disposal and have to redistribute $2s$ terminals in total. Having done this, we can join the representants as described in the above base case. ∎

Corollary 5. *If G is an a-path-pairable graph and H is a b-path-pairable graph and $|V(G)|, |V(H)| \geq \frac{(a+1)(b+1)}{2} - 1$, then $G\square H$ is $\left(\frac{(a+1)(b+1)}{2} - 1\right)$-path-pairable.*

Proof. Corollary 5 follows trivially from Theorem 4. ∎

We show that the bound presented in Corollary 5 is also sharp up to a constant factor. That is, the order of magnitude in the inheritance of path-pairability cannot be expanded more than indicated in Theorem 4 by simply providing an abundance of space in the product graph. To prove our claim, we first make the following observation: if G_0 is a subgraph of a graph G and H_0 is a subgraph of a graph H, such that they violate the cut-condition, that is, $d(G_0) < |V(G_0)|$ and $d(H_0) < |V(H_0)|$, then the subgraph $G_0 \square H_0$ of the product graph $G \square H$ does not necessarily violate the same condition. In order to generate violating product sets, stronger assumptions are needed.

Proposition 6. *Let G_0 be a subgraph of G and H_0 be a subgraph of H such that $2 \cdot d(G_0) < |V(G_0)|$ and $2 \cdot d(H_0) < |V(H_0)|$. Then $d(G_0 \square H_0) < |V(G_0 \square H_0)|$, that is, $G_0 \square H_0$ violates the cut-condition.*

Proof. Clearly $|V(G_0 \square H_0)| = |V(G_0)| \cdot |V(H_0)|$, while $d(G_0 \square H_0) = |V(G_0)| \cdot d(H_0)| + |V(H_0)| \cdot d(G_0) < \frac{|V(G_0)| \cdot |V(H_0)|}{2} + \frac{|V(G_0)| \cdot |V(H_0)|}{2} = |V(G_0)| \cdot |V(H_0)|$. ■

Let $n = k \cdot m$ and define $G(k, m)$ as follows: $V(G) = \{x_{i,j} : 0 \leq i \leq k-1, 0 \leq j \leq m-1\}$ where $x_{i,j}$ and $x_{i',j'}$ are connected, if $(i - i') \in \{-1, 0, 1\}$. In other words, we take a path on k vertices, replace every vertex by a complete graph K_m and every edge of the initial path by the edge set of a complete bipartite graph $K_{m,m}$ between the two cliques. We use the notation $S_i = \{x_{i,j} \in V(G) : 0 \leq j \leq m-1\}$ and refer to the set as the ith class of G.

Proposition 7. *$G(k, m)$ is m^2-path-pairable if $k \geq 2m$.*

Proof. Given a distribution of m^2 pairs of vertices, we can carry out pairing by starting at one end of the path, greedily joining terminals to vertices of the consecutive class, and finishing the joining of terminals within the classes. For a terminal u, we will assign several $u', u'', \ldots$ representants in the consecutive classes until we finally pair one with the appropriate v partner. We start by pairing terminals that lie in the same class by direct edges of the cliques. From now on we may assume that, for every pair (u, v), one of the terminals is closer to the left end of the path, hence it will be encountered earlier than its partner in our left-to-right sweeping algorithm. Being at class S_i, the consecutive class S_{i+1} contains at most m terminals. If some of them have appropriate representants in S_i, they can be joined by direct edges (here we are using that path-pairability prohibits repeated terminal assignment of a vertex). Then the remaining terminals of S_i can be assigned a new representant in S_{i+1}, maintaining the condition that a vertex $x \in S_{i+1}$ hosts at most m terminals and representants that have not been paired. Having visited at most t^2 terminals, this condition can be easily maintained. Having reached $t^2 + a$ terminals, we must have encountered at least a pairs, that is, the number of still unmatched terminals is at most $t^2 - a$, thus our above reasoning works just as well as before. ■

Now let $G = G(k,a)$ and $H = G(k,b)$ be such that $a \geq b \geq 2$ and $k \geq 4a^2+1$. Moreover, let G_0 be a subgraph of G and H_0 be a subgraph of H formed by $2a^2+1$ and $2b^2+1$ consecutive classes, respectively, starting at the left end of the blown-up paths. The subgraphs G_0 and H_0 satisfy the conditions of Proposition 6 thus $G\Box H$ is not $(2a^2+1)(2b^2+1)$-path-pairable, regardless of the initial number of vertices of G and H. That justifies our claim.

3. Theorem 8

Theorem 8. *The product graph $K_{m,m}\Box K_{m,m}$ is path-pairable for even values of m if $m \geq 80$.*

Proof. Let us denote the two partite sets of the complete bipartite graph $K_{m,m}$ by A_1 and A_2. We introduce further notation for certain subsets of the vertices in the product graph $G = K_{m,m}\Box K_{m,m}$ as follows: $A_{11} = A_1 \times A_1$, $A_{12} = A_1 \times A_2$, $A_{21} = A_2 \times A_1$, and $A_{22} = A_2 \times A_2$. We will refer to these sets as *classes* of G. We set a cyclic order of the four classes clockwise, that is, $A_{11}, A_{12}, A_{22}, A_{21}$. References to next class and previous class are translated in accordance with that given cyclic order. We label the m^2 elements of each class by $(u;v)$, where $u, v \in \{1,\ldots,m\}$. We will join our terminals by applying reduction on them several times. A vertex is said to *host k* terminals or representants, if there are k of them assigned to that particular vertex at a certain moment of the pairing. Note that out of k hosted terminals or representants at most one of them can be an actual terminal.

Given a pairing of the vertices, we carry out the joining of the terminals in three phases named: *swarming*, *line-up*, and *final match.* For a pair of terminals of G, we first assign to one of them a representant that lies in the class of its pair (swarming), then assign representants to the representants that lie in the same row/column of the next class (line-up). Finally, we join the representants with a common neighbor of the next class (final match).

Swarming. In this phase, we choose a representant of one terminal of each pair in the class of its partner. If a pair has both its vertices within one class, then we do not apply the swarming phase on them. We follow a case-by-case analysis:

(i) If terminal $(u_x;u_y)_{11}$ belongs to A_{11} while its partner $(v_x;v_y)_{12}$ lies in class A_{12}, we apply reduction on the triple $((ux;uy)_{11}, (v_x;v_y)_{12}, (u_x+1;u_y)_{12})$ with the 2-path $(u_x;u_y)_{11}(u_x+1;u_y)_{12}$. Addition of the coordinates is calculated modulo m.

(ii) If terminal $(u_x;u_y)_{11}$ belongs to A_{11} while its partner $(v_x;v_y)_{21}$ lies in class A_{21}, we apply reduction on the triple $((u_x;u_y)_{11}, (v_x;v_y)_{21}, (u_x;u_y+1)_{21})$ with the 2-path $(u_x;u_y)_{11}(u_x;u_y+1)_{21}$.

(iii) If terminal $(u_x; u_y)_{11}$ belongs to A_{11} while its partner $(v_x; v_y)_{22}$ lies in class A_{22}, we apply reduction on the triple $((u_x; u_y)_{11}, (v_x; v_y)_{22}, (u_x+1; u_y+2)_{22})$ with the 3-path $(u_x; u_y)_{11}(u_x+1; u_y)_{12}(u_x+1; u_y+2)_{22}$.

Terminals belonging to other classes will be assigned representants by the same rules, increasing the appropriate coordinate by 1 and 2, respectively. In case (iii), reduction is applied via a path that leads from the terminal to its representant clockwise.

One can easily verify that the above arrangement of paths assures that, if $m \geq 5$, no edge is being utilized twice during the swarming phase. We now choose the terminal for which a representant is to be assigned for each pair, such that at the end of the swarming phase every class hosts exactly $\frac{m^2}{2}$ pairs. Starting with an arbitrary selection, we can assume without loss of generality that A_{11} hosts the most pairs, and that at least one terminal $x \in A_{11}$ shares a class with the representant of its partner y from a class hosting less than $\frac{m^2}{2}$ pairs. Assigning a representant to x instead of y balances the distribution of the pairs. Repetition of the previous step leads to an equal distribution.

We define G' with $V(G') = V(G)$ and a new edge set $E(G')$ by deleting those edges from $E(G)$ we used in the swarming phase. Observe that every vertex of G hosts at most 4 terminals or representants and got at most 6 of its initial edges deleted, that is, the minimal degree of G' is at least $m - 6$. We continue the linking in G'.

Line-up. For every initial pair of terminals we now have a terminal-representant pair lying in the same class. As described in the reduction, we consider these pairs as an initial pairing of terminals in G'. To every pair of terminals $G(u, v)$ we assign representants u', v' lying in the next class, such that u' and v' share a column of the next class if uu' and vv' are horizontal edges and they share a row if the mentioned edges are vertical. For every pair, there are at least $m - 12$ available columns/rows in the next class. Our intention is to pair up the pairs with the rows/columns, such that every one of them will contain $\frac{m}{2}$ pairs. We recall a straightforward corollary of Hall's Matching Theorem.

Lemma 9. *A bipartite graph $G = (A, B, E)$ with partite sets of order n whose minimum degree is at least $\frac{n}{2}$ contains a perfect matching.*

We define the following bipartite graph $G = (A, B, E)$ as follows: represent each pair of terminals hosted in A_{11} by a vertex in A, while each column of A_{12} is represented by $\frac{m}{2}$ independent vertices in B. Certainly, $|A| = |B| = \frac{m^2}{2}$. We connect two vertices of A and B by an edge if both terminals of the corresponding pair have horizontal edges to the corresponding column of A_{12}. Notice that the graph G has minimum degree at least $\frac{m^2}{2} - 6m$, hence, by Lemma 9, it contains a perfect matching for $m \geq 24$.

Observe that if two pairs of terminals sharing a vertex of a class C are distributed to the same H-layer of the next class C', we will not be able to use reduction at both terminals using the same edge leading from that particular vertex to the next class. We want to guarantee a matching between the pairs and the layers of C' without such a collision. Recall that each vertex of C hosts at most 4 terminals, hence each pair of terminals has at most 6 additional pairs to collide with. Consider a perfect matching for which the number of above collisions is minimal. Let (u_1, v_1) and (u_2, v_2) be colliding pairs of terminals that are assigned representants at layer L of C'. We want to find a pair (u, v) with representants on a layer $L' \neq L$ such that

(i) (u_1, v_1) can be assigned representants on L' (instead of L) during the line-up without causing further collision,

(ii) (u, v) can be assigned representants on L (instead of L') during the line-up without causing further collision.

The pair (u_1, v_1) can be initially assigned representants on $m - 12$ layers of C', at most 6 of which might contain representants that initially shared vertex with (u_1, v_1) in C. In order to avoid further collisions we exclude these layers, leaving us at least $m - 18$ choices of L'. We also want to exclude layers that already host representants of the vertex of x or y, yielding at most 6 additional excluded layers. That is, we have at least $m - 24$ choices of L' and so $(m - 24) \cdot \frac{m}{2}$ choices for (u, v).

We want to choose (u, v) such that it initially did not share vertex in C with any terminal currently hosted in L and that u and v still can be given representants (having withdrawn from L') in L (that is, the corresponding edges have not yet been used). For the first constraint, recall that L contains $\frac{m}{2}$ pairs, every one of which shares vertex with at most 6 additional terminals. There are at most $3m$ additional terminals that initially cannot be assigned representants in L, because the appropriate edges had already been used during the first phase.

Now assume that the appropriate edge that would connect u or v to L has already been used. This can either occur if another terminal was given a representant in L during the line-up, or if the edges were used during the swarming phase. The first condition means that (u, v) collides with the other pair of terminals that was given representants in L, hence (u, v) is one of the above listed $3m$ pairs. In the remaining case, the missing edge is one of those at most $6 \cdot \frac{m}{2} = 3m$ edges the complete layer L used up during the swarming. The mentioned edges have at most $3m$ endpoints in C and at most $4 \cdot 3m = 12m$ pairs of terminals corresponding to them.

Overall, it means that if $(m - 24) \cdot \frac{m}{2} > 15m$ (that is, $m > 54$), one can find an appropriate pair (u, v). Swapping the positions of (u, v) and (x, y), we can reduce the number of collisions, contradicting our initial assumption. We repeat the same procedure for the remaining three classes. It can be easily verified that no edge is used more than once. We define G'' by the deletion of the used edges the same way we obtained G'. We proceed in G'' to the final match.

Final match. For a row/column filled with $\frac{m}{2}$ pairs of terminals, we assign to every pair a vertex of the appropriate row/column of the next class that is adjacent to both terminals. Note that during the first two phases, each vertex has used at most 10 of its edges. We use Lemma 6 to find the appropriate assignment. Let A form the set, in which every pair of terminals of a certain row/column is represented by a vertex. The set B is formed by any $\frac{m}{2}$ vertices of the appropriate column/row of the next class. We connect vertices by edges, if both terminals of the pair are adjacent to the appropriate vertex in the next class. Our bipartite graph has two classes on $\frac{m}{2}$ vertices and minimum degree $\frac{m}{2} - 20$. If $m \geq 80$, the required matching is provided by Lemma 6. That completes the proof. ■

Corollary 10. *There exists a path-pairable graph G on n vertices with $\Delta(G) = \sqrt{n}$ for infinitely many values of n.*

4. Additional Remarks and Open Questions

4.1. The cut-condition is not sufficient

We first prove that the k-cut condition does not imply k-path-pairability by defining the graph $\tilde{G}$ in the following way: consider the disjoint union of the star graph $K_{1,k}$ and the complete graph K_N on $N \geq 2k$ vertices. Join each vertex of degree one to an arbitrary vertex of K_N by an edge, such that different vertices of $K_{1,k}$ get joined to different vertices of K_N.

Proposition 11. *The graph $\tilde{G}$ satisfies the k-cut-condition, yet it is not k-path pairable.*

Proof. Place $k + 1$ terminals in $K_{1,k}$ such that the pair of the terminal in the center of the star graph is placed in K_N. It is easy to see that any path starting in the center of the star severs its neighbor, using both of its edges. On the other hand, any $S \subset V(G)$ trivially satisfies the cut-condition, if it contains a vertex of K_N. In the remaining case we may assume $S \subset K_{1,k}$, in which case the verification of the cut-condition is also straightforward. ■

Appropriate fine-tuning of the construction provides examples of graphs that are not path-pairable while they satisfy the cut-condition. We define graph $\hat{G}$ as follows: take the disjoint union of $K_{1,k}$ and K_{k-1} and join the two graphs by a matching (avoiding the center of the star) of size $k-1$. Join the remaining vertex of degree one to any vertex of K_{k-1}.

Proposition 12. *The graph $\hat{G}$ satisfies the cut-condition for $k \geq 6$, yet it is not path-pairable.*

Proof. Just as in case of $\tilde{G}$, the set of degree-two vertices joined to the center of the star make it impossible for $\hat{G}$ to channel k edge-disjoint paths, thus $\hat{G}$ is not path-pairable. Assume now that the subset $S \subset G$ of at most k vertices violates the condition. We proceed by a case-by-case analysis.

Case 1. If $K_{k-1} \subset S$, then S must contain an additional vertex, that is, $|S| = k$. It is easy to see that adding neither the center nor any end of the star graph to the vertex set of K_{k-1} violates the condition.

Case 2. If $|S \cap K_{k-1}| = k-2$, then $d(S) \geq k-2$ because of the edges leaving S within K_{k-1}. Also, at least $k-4$ of them have neighbors in $K_{1,k}$ not belonging to S, that is, $d(S) \geq k-2+k-4 \geq k$. Since $|S| \leq k$, it cannot violate the cut-condition.

Case 3. If $1 \leq |S \cap K_{k-1}| \leq k-3$, then $d(S) \geq 2k-6 \geq k$ only by considering the edges leaving S within K_{k-1}.

Case 4. If $S \subset K_{1,k}$, then S must contain the center of the star else it trivially satisfies the condition. Observe that each non-central vertex of $K_{1,k}$ has an edge leaving S toward K_{k-1} and so does at least one edge of the star. It completes the proof. ■

It has been known for some time that not only linkedness and weak-linkedness do force high connectivity and edge-connectivity of the graph, but that sufficiently large connectivity and edge-connectivity can imply high linkedness and weak-linkedness, respectively. It would be interesting to see if similar result can be proved about the relation of the cut-conditions and path-pairability.

4.2. Path-pairability of hypercubes and grids

As discussed previously, path-pairable graphs on n vertices have a certain lower bound on the minimal value of the maximum degree Δ that is approximately $\frac{\log n}{\log\log n}$. On the other hand, the smallest achieved maximum degree provided by Theorem 8 has the order of magnitude of $\sqrt{n}$, still leaving plenty of room for improvements on both sides. One particularly interesting and promising path-pairable candidate is the d-dimensional hypercube Q_d on $n = 2^d$ vertices with $\Delta(Q_cd) = d = \log n$. Although it is known that Q_d is not path-pairable for even values of d [4], the question is open for odd dimensional hypercubes if $d \geq 5$ (Q_1 and Q_3 are both path-pairable).

Conjecture 13 [3]**.** *The $(2k+1)$-dimensional hypercube Q_{2k+1} is path-pairable for all $k \in \mathbb{N}$.*

The question regarding the path-pairability number of larger n dimensional affine and projective grids, that is, the Cartesian product of d paths or d cycles has not been answered either. It can be derived rather easily from Theorem 4 that sufficiently large d-dimensional projective grids are $O(2^d)$-path-pairable. Similar result concerning affine grids can be obtained. On the other hand, it can be proved that a d-dimensional projective grid is at most $O((2d)^{2d})$-path-pairable, regardless the number of its vertices if the grid is large enough in every dimension.

Proposition 14. *Let $G = C_{m_1} \square C_{m_2} \square \cdots \square C_{m_d}$, where C_{m_i} denotes the cycle of length m_i and $m_i \geq 2d+1$, $i = 1, 2, \ldots, d$. Then* $\mathrm{pp}(G) \leq (2d)^{d-1} \cdot (2d+1)$.

Proof. We may assume $|G| \geq 2 \cdot (2d)^{d-1} \cdot (2d+1)$, else the statement is trivial. Consider now the d-dimensional subgrid $G_0 = C_{2d} \square \cdots C_{2d} \square C_{2d+1}$. It is easy to see that G_0 violates the cut-condition as $V(G_0) = (2d)^{d-1} \cdot (2d+1) > 2 \cdot ((d-1)(2d)^{d-2}(2d+1) + (2d)^{d-1}) = d(G_0)$. This shows that G is less than $(2d)^{d-1} \cdot (2d+1)$-path-pairable. ■

The presented bounds are still far apart and leave plenty of room for improvements.

Problem 15. Determine the values of $\mathrm{pp}(P_{m_1} \square P_{m_2} \square \cdots \square P_{m_d})$ and $\mathrm{pp}(C_{m_1} \square C_{m_2} \square \cdots \square C_{m_d})$ (P_{m_i} denotes the path on m_i vertices).

4.3. Path-pairable products

This paper only deals with a special type of products of complete bipartite graphs. With a detailed and cumbersome analysis of our presented techniques, we believe one can extend the result of Theorem 8.

Conjecture 16. *The product graph $K_{a,b} \square K_{c,d}$ is path-pairable if $\frac{\max(a,b,c,d)}{\min(a,b,c,d)} < 2$ and a, b, c, d are large enough in terms of $\frac{\max(a,b,c,d)}{\min(a,b,c,d)}$.*

We have proved in Proposition 3 that path-pairability of the factors G and H does not imply that the product graph $G \square H$ is path-pairable. We believe that, with a somewhat more detailed analysis, one can actually prove that the bound of Proposition 3 is sharp.

Conjecture 17. *Verify that* $\mathrm{pp}(K_{1,a} \square K_{1,b}) = \lceil \frac{a+b}{2} \rceil$.

Annoyingly enough, we do not know whether path-pairability of at least one of the multiplicands is necessary at all for path-pairability of the product graph. We believe that the described condition is not necessary, but cannot verify it by means of a counterexample, hence we state it as a conjecture as well.

Conjecture 18. *There exist non-path-pairable graphs G and H such that $G \square H$ is path-pairable.*

We close up by highlighting that path-pairability of $K_{a,b}\Box K_{c,d}$ in the general case is still subject to further investigation, as well as proposing another intriguing open question motivated by [9].

Problem 19. For which values of $a, b, c, d \in \mathbb{N}$ ($a \leq b$, $c \leq d$) is the product graph $K_{a,b}\Box K_{c,d}$ path-pairable?

Problem 20. What are the necessary and sufficient conditions for a graph G to guarantee that $G\Box K_n$ is path-pairable if n is large enough?

Acknowledgment

The author wishes to thank Professor Ervin Győri for his helpful suggestions, interest, and guidance. The author would also like to thank the anonymous referees for their valuable comments and corrections. Research was supported by the Balassi Institute, the Fulbright Commission, and the Rosztoczy Foundation.

References

[1] W. Imrich and S. Klavžar, Product Graphs: Structure and Recognition (J. Wiley & Sons, New York, 2000).

[2] W.-S. Chiue and B.-S. Shieh, *On connectivity of the Cartesian product of two graphs*, Appl. Math. Comput. **102** (1999) 129–137. doi:10.1016/S0096-3003(98)10041-3

[3] L. Csaba, R.J. Faudree, A. Gyárfás, J. Lehel and R.H. Schelp, *Networks communicating for each pairing of terminals*, Networks **22** (1992) 615–626. doi:10.1002/net.3230220702

[4] R.J. Faudree, *Properties in pairable graphs*, New Zealand J. Math. **21** (1992) 91–106.

[5] R.J. Faudree, *Some strong variations of connectivity*, in: Combinatorics, Paul Erdős is Eighty, Bolyai Soc. Math. Stud. **1** (1993) 125–144.

[6] R.J. Faudree, A. Gyárfás and J. Lehel, *Minimal path pairable graphs*, Congr. Numer. **88** (1992) 111–128.

[7] R.J. Faudree, A. Gyárfás and J. Lehel, *Three-regular path pairable graphs*, Graphs Combin. **8** (1992) 45–52. doi:10.1007/BF01271707

[8] R.J. Faudree, A. Gyárfás and J. Lehel, *Path-pairable graphs*, J. Combin. Math. Combin. Comput. **29** (1999) 145–157.

[9] E. Kubicka, G. Kubicki and J. Lehel, *Path-pairable property for complete grids*, Combin. Graph Theory Algorithms **1** (1999) 577–586.

[10] G. Mészáros, *On linkedness in the Cartesian product of graphs*, Period. Math. Hungar., to appear. doi:10.1007/s10998-016-0113-8

[11] G. Mészáros, *Note on the diameter of path-pairable graphs*, Discrete Math. **337** (2014) 83–86.
doi:10.1016/j.disc.2014.08.011

[12] J. Xu and C. Yang, *Connectivity of Cartesian product graphs*, Discrete Math. **306** (2006) 159–165.
doi:10.1016/j.disc.2005.11.010

Received 29 September 2014
Revised 4 November 2015
Accepted 10 November 2015