ELSEVIER

Discrete Applied Mathematics 74 (1997) 101–121

DISCRETE APPLIED MATHEMATICS

Long cycles and long paths in the Kronecker product of a cycle and a tree

Pranava K. Jha[a,*], Naveen Agnihotri[b], Rajesh Kumar[c]

[a] *Department of Computer Engineering, Delhi Institute of Technology, Delhi, Kashmere Gate, Delhi 110 006, India*

[b] *Department of Biological and Agricultural Engineering, University of Georgia, Athens, GA 30605, USA*

[c] *Centre for Development of Telematics, 39 Main Pusa Road, New Delhi 110 005, India*

Abstract

Let $C_m \times T$ denote the Kronecker product of a cycle C_m and a tree T. If m is odd, then $C_m \times T$ is connected, otherwise this graph consists of two isomorphic components. This paper presents a scheme which constructs a long cycle in each component of $C_m \times T$. If T satisfies certain degree constraints, then the cycle thus traced is shown to be a dominating set, and in some cases, a vertex cover of that component. The procedure builds on (i) results on longest cycles in $C_m \times P_n$, and (ii) a path factor of T. Additional results include characterizations for the existence of a Hamiltonian cycle and for that of a Hamiltonian path in $C_m \times T$.

Keywords: Kronecker product; Cycle; Tree; Path factor; Long cycle; Long path

1. Introduction

Let $C_m \times T$ denote the *Kronecker product* of a *cycle* C_m and a *tree* T. Principal result of this paper consists of a procedure which constructs a long cycle in $C_m \times T$. If T satisfies certain degree constraints, then the cycle thus traced is shown to be a dominating set (and in some cases, a vertex cover). The scheme builds on (i) a previous work by one of the authors [9] with respect to $\times$-product of a cycle and a path, and (ii) a *path factor* of T. Additional results include characterizations for the existence of a *Hamiltonian cycle* and for that of a *Hamiltonian path* in $C_m \times T$.

Batagelj and Pisanski [1] earlier presented a characterization for the existence of a Hamiltonian cycle in the Cartesian product of a cycle and a tree. For the analogous problem with respect to the strong product, Bermond et al. [2] reported certain sufficient conditions.

* Corresponding author. E-mail: pkj@dit.ernet.in.

0166-218X/97/$17.00 © 1997 Elsevier Science B.V. All rights reserved
PII S0166-218X(96)00022-4

By a graph is meant a finite, simple and undirected graph. Unless indicated otherwise, graphs are also connected and have at least two vertices. For graphs $G = (V,E)$ and $H = (W,F)$, the Kronecker product (or $\times$-*product*) of G and H is denoted by $G \times H$ and is defined as follows: $V(G \times H) = V \times W$ and $E(G \times H) = \{\{(u,x),(v,y)\} \mid \{u,v\} \in E \text{ and } \{x,y\} \in F\}$. This product is commutative and associative in an obvious way. Further, it is distributive with respect to edge-disjoint union of graphs. Among various associative products studied by Imrich and Izbicki [8], the Kronecker product has proved to be one of the most important. Several applications have been listed by Jha et al. [10].

For $m \geqslant 3$ and $n \geqslant 1$, let C_m and P_n, respectively, denote a cycle on m vertices and a path on n vertices, where $V(C_k) = V(P_k) = \{0,\ldots, k-1\}$, and where adjacencies are defined in the natural way. If k is even (resp. odd), then P_k is said to be an *even path* (resp. *odd path*). A tree is a connected, acyclic graph. Note that if T is a tree on n vertices, then $|V(C_m \times T)| = m \cdot n$ and $|E(C_m \times T)| = 2 \cdot m \cdot (n-1)$.

If a graph G is obtainable from a graph H by a sequence of edge subdivisions, then G is said to be *homeomorphic from* H. A vertex subset S of G is called a *dominating set* if every vertex of G not in S is adjacent to some element of S. Further, S is called a *vertex cover* if every edge of G has at least one end vertex in S.

For a graph G, let $c(G)$ and $l(G)$, respectively, denote the length of a longest cycle and the length of a longest path in G. The general problem of determining any of these two invariants is NP-hard, and remains so even if the graph is known to be bipartite [5]. The following definition is relevant to our study.

Definition 1 (Jung et al. [12]). A connected graph is said to be almost Hamiltonian if it is biconnected and it contains a cycle which is a vertex cover. □

For isomorphic graphs G and H, we write $G \cong H$. If G is a graph and v is a vertex of G, then $deg_G(v)$ denotes the degree of v while $\Delta(G)$ denotes the largest degree of G. If $S \subseteq V(G)$, then $\langle S \rangle$ denotes the subgraph induced by S. The distance between two vertices u, v of a graph G is denoted by $dist_G(u,v)$. For a tree T, a vertex of degree one is called an *endpoint* while a vertex adjacent to an endpoint is called a *support vertex*. T is said to be 1-*contractable* to a path if either T itself is a path or T minus its endpoints is a path.

The following theorem states certain relevant characteristics of $C_m \times T$.

Theorem 1.1. 1. *$C_m \times T$ is a bipartite graph* [7].

2. *$C_m \times T$ is connected iff m is odd* [15].
3. *If m is even, then $C_m \times T$ consists of two isomorphic components* [10].
4. *Each component of $C_m \times T$ is biconnected but not triconnected.* □

Whether $C_m \times T$ is planar/outerplanar depends mainly on the structure of T. In particular, if $m = 4$ or T is 1-contractable to a path, then $C_m \times T$ is planar. On the other hand, $C_m \times T$ is outerplanar iff $T \cong K_2$. Characterizations for planarity and

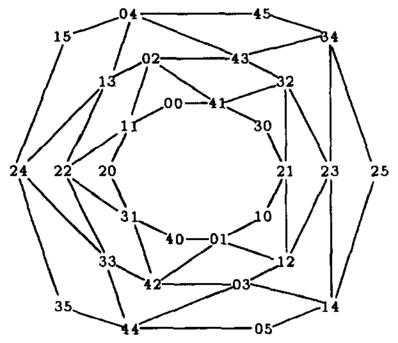

Fig. 1. Graph $C_5 \times P_6$.

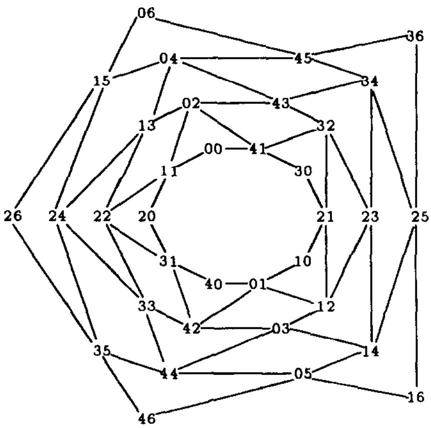

Fig. 2. Graph $C_5 \times P_7$.

outerplanarity of $\times$-product graphs have, respectively, been reported by Farzan and Waller [4], and Jha and Slutzki [11]. For any undefined terms, see Harary [6].

Graphs $C_5 \times P_6$ and $C_5 \times P_7$ appear in Figs. 1 and 2, respectively. For the sake of clarity, vertex (i,j) has been shown as ij.

The following observations about the structure of the graph $C_m \times P_n$ are instructive [9]. First suppose that m is odd and n is even, and consider the partition of $V(C_m \times P_n)$ into the following subsets: $\{0,\dots,m-1\} \times \{2i, 2i+1\}$, $0 \leqslant i \leqslant (n/2)-1$. Each of these vertex subsets induces a cycle of length $2m$. Based on this fact, the graph $C_m \times P_n$ may be viewed as containing $n/2$ "concentric cycles", each of length $2m$. (See Fig. 1.) If m and n are both odd, then $V(C_m \times P_n)$ may be partitioned into the folllwing subsets: $\{0,\dots,m-1\} \times \{2i, 2i+1\}$, $0 \leqslant i \leqslant (n-3)/2$, and $\{0,\dots,m-1\} \times \{n-1\}$. In this case, the first $(n-1)/2$ subsets correspond to as many "concentric cycles", each of length $2m$ while vertices of the subset $\{0,\dots,m-1\} \times \{n-1\}$ (which constitutes an independent set) are "attached" to the "rest of the graph" as part of a cycle of length $2m$. (See Fig. 2.) Next suppose that m is even so that $C_m \times P_n$ consists of two isomorphic components. In this case, there is an analogous partition of the vertex set of each component into certain "concentric cycles," each of length m. The following result is relevant.

Lemma 1.2 (Jha et al. [10]). *If m is even, $m/2$ is odd, and G is a bipartite graph, then each component of $C_m \times G$ is isomorphic to $C_{m/2} \times G$.* $\square$

An obvious upper bound on $c(G)$ and $l(G)$ of a bipartite graph G appears below.

Lemma 1.3. *If $G = (V_0 \cup V_1, E)$ is a biparite graph and $|V_0| \leqslant |V_1|$, then each of $c(G)$ and $l(G)$ is at most $2 \cdot |V_0|$.* $\square$

The following definition will be useful in the sequel.

Definition 2. If T is a tree, then sub (T) is the tree obtained from T as follows: For each support vertex x of T, if the number k_x of endpoints adjacent to x is greater than one, then remove any $k_x - 1$ endpoints adjacent to x. $\square$

For a tree T, *sub*(T) is a subtree of T, and is obtainable by suitably "trimming" the "periphery" of T. It is clear that for a fixed tree T, *sub*(T) is unique up to isomorphism. If T is a tree in which every support vertex has exactly one endpoint adjacent to it, then *sub*$(T) = T$. In general, *sub*(*sub*$(T)) =$ *sub*(T). Further, *sub*$(K_{1,r}) = K_2$, and if T is a tree of diameter three, then *sub*$(T) = P_4$. This operation is useful in our study. In particular, it turns out that contribution to the length of a longest cycle in $C_m \times T$ due to several endpoints "bunched" at one support vertex x is no greater than that due to a single endpoint "hanging" from x.

Lemma 1.4. (1) $c(C_m \times T) = c(C_m \times sub(T))$.
(2) $l(C_m \times sub(T)) \leqslant l(C_m \times T) \leqslant 1 + l(C_m \times sub(T))$.

Proof. Let T be a tree, and let v be a support vertex of T. Suppose that $x_1, \ldots, x_k$ are the endpoints adjacent to v. For a vertex i of C_m, the vertices $(i,x_1), \ldots, (i,x_k)$ of the graph $C_m \times T$ are such that at most one of them may be included in a cycle of length greater than four. This is because each of $(i,x_1), \ldots, (i,x_k)$ is of degree two, and has the same set of neighbors, viz, $\{(i-1,v),(i+1,v)\}$. Thus, from the viewpoint of tracing a longest cycle in $C_m \times T$, it suffices to retain exactly one vertex from among $(i,x_1), \ldots, (i,x_k)$. Note further that the graph obtained from $C_m \times T$ by retaining exactly one vertex from among $(i,x_1), \ldots, (i,x_k)$ corresponding to every support vertex v of T is isomorphic to $C_m \times sub(T)$.

For (2), let P be a longest path in $C_m \times T$, and let (i,x_j) be a vertex of $C_m \times T$, where x_j is an endpoint of T. If at most one terminal vertex of P is of the form (i,x_j), then P must be of length $l(C_m \times sub(T))$. On the other hand, if both terminal vertices of P are of the form (i,x_j), then P will be of length at most $1 + l(C_m \times sub(T))$. This follows by an argument similar to that above. $\square$

It follows that $c(C_m \times T)$ is sensitive to the form of T. (See also Lemma 1.5.)

Definition 3. Let G and H be graphs. A subgraph H' of H is said to participate in a cycle C of $G \times H$ if at least one vertex of C is of the form (u,x), where $x \in V(H')$, otherwise H' is said not to participate in C. □

Lemma 1.5. *Let T be a tree.*

1. *Let m be odd $\geqslant 3$. If v is a vertex of T whose neighbors are $x_1,\ldots,x_k$, where $k > m$, then at most m of the vertices $x_1,\ldots,x_k$ may participate in any cycle of $C_m \times T$.*
2. *Let m be even $\geqslant 4$. If v is a vertex of T whose neighbors are $x_1,\ldots,x_k$, where $k > m/2$, then at most $m/2$ of the vertices $x_1,\ldots,x_k$ may participate in any cycle of a connected component of $C_m \times T$.*

Proof. Let m be odd $\geqslant 3$, and let T, v, and $x_1,\ldots,x_k$ be as stated in (1), where $k > m$. For $1 \leqslant i \leqslant k$, let T_{x_i} be the largest subtree which includes x_i and excludes v. It is easy to see that x_i participates in a cycle of $C_m \times T$ if and only if T_{x_i} does. The subgraph of T induced by $\{v,x_1,\ldots,x_k\}$ is isomorphic to $K_{1,k}$ where v is the "center" of the star. Clearly, $C_m \times K_{1,k}$ is an induced subgraph of $C_m \times T$.

In order for a subtree T_{x_i} to participate in a cycle of $C_m \times T$, exactly two edges of the following form must appear on that cycle: $\{(r,v),(r',x_i)\}$ and $\{(s,v),(s',x_i)\}$, where $\{r,r'\}$ and $\{s,s'\}$ are edges of C_m. Consequently, if p subtrees from among $T_{x_1},\ldots,T_{x_k}$ participate in a cycle of $C_m \times T$, then $2p$ edges of the foregoing form must appear on that cycle. Since each such edge includes a vertex of the form (r,v) and there are exactly m such vertices, it follows that out of $T_{x_1},\ldots,T_{x_k}$, a maximum of m subtrees may participate in any cycle of $C_m \times T$.

For the case when m is even, recall Lemma 1.2 and the remarks preceding it. □

By Lemma 1.5, $c(C_m \times T) = \max\{c(C_m \times T_1),\ldots,c(C_m \times T_r)\}$, where $\{T_1,\ldots,T_r\}$ is the set of subtrees of T such that $\Delta(T_i) \leqslant m$ if m is odd, and $\Delta(T_i) \leqslant m/2$ if m is even, $1 \leqslant i \leqslant r$.

Section 2 states results on edge decompositions of $C_m \times P_n$ into long cycles and long paths, respectively. In particular, exact values are presented for $c(C_m \times P_n)$ and $l(C_m \times P_n)$. Section 3 deals with a path factor of a tree. The purpose is to prepare ground for the development of a scheme for a long cycle in $C_m \times T$, where T satisfies certain degree constraints. The scheme itself appears in Section 4 and constitutes our main result. Section 5 consists of certain concluding remarks.

2. Preliminary results

Important results of this section include (i) characterizations for the existence of a Hamiltonian cycle and for that of a Hamiltonian path in $C_m \times T$, and (ii) edge decompositions of each component of $C_m \times P_n$ into long cycles and long paths.

Theorem 2.1. *Let T be a tree having n vertices and p endpoints.*

1. *$C_m \times T$ is Hamiltonian iff m is odd and $T \cong K_2$.*
2. *If m is odd and $n \geqslant 3$, then $c(C_m \times T) \leqslant mn - p$, if $mn - p$ is even, and $c(C_m \times T) \leqslant mn - p - 1$, otherwise.*
3. *If m is even and $n \geqslant 3$, then $c(C_m \times T) \leqslant (mn/2) - p$, if $(mn/2) - p$ is even, and $c(C_m \times T) \leqslant (mn/2) - p - 1$ otherwise.*

Proof. Let m, n, T be as stated. First note that (i) if m is odd, then $C_m \times K_2 \cong C_{2m}$, and (ii) if m is even, then $C_m \times T$ is disconnected. Next, let m be odd, and let $1, \dots, p$ be the endpoints of T, where $n \geqslant 3$. Consider the following vertex subsets of $C_m \times T$: $V_j = \{(i,j) \mid 0 \leqslant i \leqslant m-1\}$, $1 \leqslant j \leqslant p$. Clearly, $V_1, \dots, V_p$ are mutually disjoint and every vertex in each of these sets is of degree two. Further, for each V_j, there is exactly one simple cycle in $C_m \times T$, which includes all m vertices of V_j and that cycle is of length $2m$. (See Figs. 1 and 2, where $m = 5$.) Thus, any simple cycle in $C_m \times T$ which is of length greater than $2m$ must exclude at least one vertex from each V_j, $1 \leqslant j \leqslant p$. Consequently, $c(C_m \times T) \leqslant mn - p$. If $mn - p$ is odd, then $c(C_m \times T) \leqslant mn - p - 1$. This is because $C_m \times T$ is a bipartite graph. Statements (1) and (2) follow. For (3), recall Lemma 1.2 and the remarks preceding it. □

We next show that the result of Theorem 2.1 is sharp.

Theorem 2.2. *Let $m, n \geqslant 3$.*

1. *If m is odd and n is even, then $c(C_m \times P_n) = mn - 2$, and $C_m \times P_n$ is edge decomposable into two cycles, one of which is longest.*
2. *If m and n are both odd, then $c(C_m \times P_n) = m(n-1)$, and $C_m \times P_n$ is edge decomposable into two longest cycles.*
3. *If m and n are both even, then $c(C_m \times P_n) = (mn/2) - 2$, and each component of $C_m \times P_n$ is edge decomposable into two cycles, one of which is longest.*
4. *If m is even and n is odd, then $c(C_m \times P_n) = m(n-1)/2$, and each component of $C_m \times P_n$ is edge decomposable into two longest cycles.*

Proof. By Lemma 1.3, it follows that (i) if m and n are both odd, then $c(C_m \times P_n) \leqslant m(n-1)$, and (ii) if m is even and n is odd, then $c(C_m \times P_n) \leqslant m(n-1)/2$. Everything else appears in [9]. □

Proof of Theorem 2.2 (1–2) is illustrated by cycle decompositions of $C_5 \times P_6$ and $C_5 \times P_7$, which, respectively, appear in Figs. 3 and 4. In each case, the first cycle is a longest cycle of that graph, and it may be viewed as a sequence $\sigma = (i_0, j_0), \dots, (i_{k-1}, j_{k-1})$ of vertices, where $(i_0, j_0) = (0,0)$, $(i_{k-1}, j_{k-1}) = (m-1, 1)$, and $k = mn - 2$ (resp. $k = mn - m$) if n is even (resp. odd). Because of symmetry, it is easy to see that the sequence $(i_0 + a, j_0), \dots, (i_{k-1} + a, j_{k-1})$ will also correspond to a cycle of the same size, where $1 \leqslant a \leqslant m - 1$. To this end, Figs. 5 and 6, respectively, consist of alternative cycle decompositions of $C_5 \times P_6$ and $C_5 \times P_7$, which are based on this

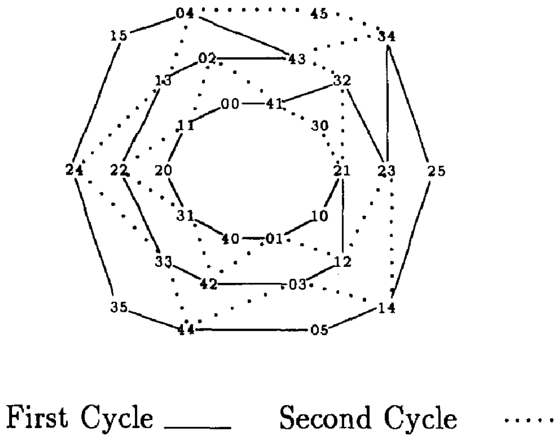

Fig. 3. Cycle decomposition of $C_5 \times P_6$.

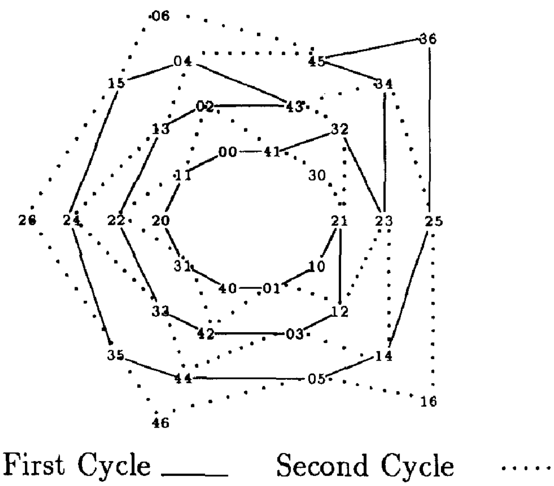

Fig. 4. Cycle decomposition of $C_5 \times P_7$.

observation. Again, the first cycle is a longest cycle and corresponds to the sequence $\sigma' = (i_0 + m - 1, j_0), \ldots, (i_{k-1} + m - 1, j_{k-1})$.

If m is odd $\geqslant 3$, and $n \geqslant 1$, then $c(C_m \times P_n)$ satisfies the following recurrence:

$$c(C_m \times P_n) = \begin{cases} 0 & \text{if } n = 1, \\ 2m & \text{if } n = 2, 3, \\ c(C_m \times P_{n-1}) + 2m - 2 & \text{if } n \text{ is even} \geqslant 4, \\ c(C_m \times P_{n-1}) + 2 & \text{if } n \text{ is odd} \geqslant 5. \end{cases}$$

Section 4 makes use of this recurrence. For the case when m is even, a recurrence for $c(C_m \times P_n)$ is obtainable from the foregoing by replacing "$2m$" by "m" on the right

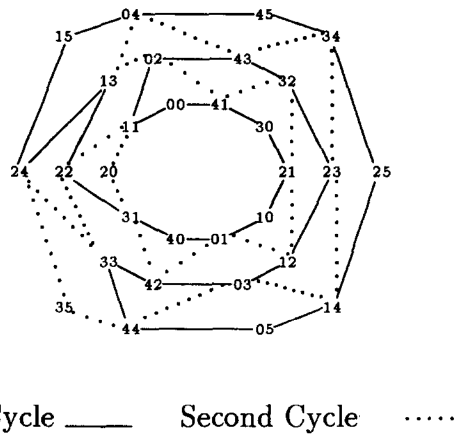

Fig. 5. Alternative cycle decomposition of $C_5 \times P_6$.

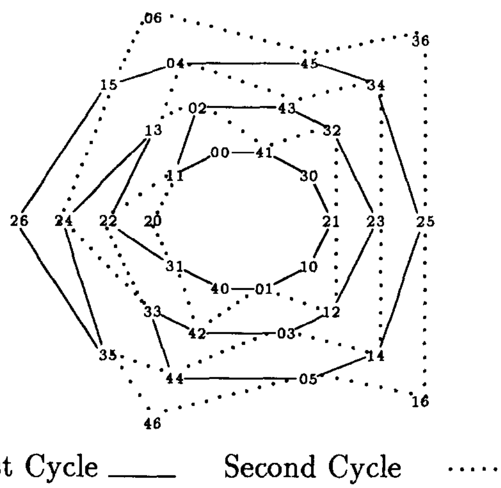

Fig. 6. Alternative cycle decomposition of $C_5 \times P_7$.

side. By Lemma 1.4 and Theorem 2.2, if T is a tree such that $sub(T)$ is a path, then we have an exact value for $c(C_m \times T)$.

The next result deals with decomposition of $C_m \times P_n$ into long paths.

Theorem 2.3. *Let* $m, n \geqslant 3$.

1. *If* m *is odd and* n *is even, then* $l(C_m \times P_n) = mn - 1$, *and* $C_m \times P_n$ *is edge decomposable into two paths, one of which is a Hamiltonian path.*

2. *If* m *and* n *are both odd, then* $l(C_m \times P_n) = m(n-1)$, *and* $C_m \times P_n$ *is edge decomposable into two longest paths.*

3. *If* m *and* n *are both even, then* $l(C_m \times P_n) = (mn/2) - 1$, *and each component of* $C_m \times P_n$ *is edge decomposable into two paths, one of which is a Hamiltonain path of that component.*

4. *If m is even and n is odd, then $l(C_m \times P_n) = m(n-1)/2$, and each component of $C_m \times P_n$ is edge decomposable into two longest paths.*

Proof. By Lemma 1.3, it follows that (i) if m and n are both odd, then $l(C_m \times P_n) \leqslant m(n-1)$, and (ii) if m is even and n is odd, then $l(C_m \times P_n) \leqslant m(n-1)/2$. Everything else appears in [9]. □

The following recurrence is analogous to that for $c(C_m \times P_n)$ presented earlier:

$$l(C_m \times P_n) = \begin{cases} 0 & \text{if } n = 1 \\ 2m-1 & \text{if } n = 2 \\ l(C_m \times P_{n-1}) + 1 & \text{if } n \text{ is odd} \geqslant 3 \\ l(C_m \times P_{n-1}) + 2m - 1 & \text{if } n \text{ is even} \geqslant 4. \end{cases}$$

For the case when m is even, a recurrence for $l(C_m \times P_n)$ is obtainable from the foregoing by replacing "$2m$" by "m" on the right side. The next result is analogous to Theorem 2.1.

Theorem 2.4. *Let T be a tree having n vertices and p endpoints.*

1. *$C_m \times T$ contains a Hamiltonian path iff m is odd and T is an even path.*
2. *If m is odd and T is not a path, then $l(C_m \times T) \leqslant mn - p + 1$.*
3. *If m is even and T is not a path, then $l(C_m \times T) \leqslant (mn/2) - p + 1$.*

Proof. First note that (i) if m is even, then $C_m \times T$ is disconnected, and (ii) $C_m \times T$ contains a Hamiltonian path precisely when $l(C_m \times T) = mn - 1$. By Theorem 2.3, (i) if m is odd and n is even, then $l(C_m \times P_n) = mn - 1$, and (ii) if m and n are both odd, then $l(C_m \times P_n) = m(n-1)$. Next observe that if m is odd and T is not a path (in which case, $p \geqslant 3$), then $l(C_m \times T) \leqslant mn - p + 1$. This follows by an argument as in the proof of Theorem 2.1(2). By a similar reasoning, if m is even and T is not a path, then $l(C_m \times T) \leqslant (mn/2) - p + 1$. □

3. Path factor of a tree

By a path factor of a graph is meant a spanning subgraph which has at least one edge, and each of whose connected components is a path (possibly K_1).

Definition 4. Let T be a tree having p endpoints. T is said to admit of a path factor consisting of a sequence $P_{n_1}, \ldots, P_{n_{p-1}}$ of paths if

1. $P_{n_1}, \ldots, P_{n_{p-1}}$ constitute a vertex decomposition of T,
2. P_{n_1} is such that both of its terminal vertices are endpoints of T, and
3. for $2 \leqslant i \leqslant p-1$, P_{n_i} is a path (possibly K_1) exactly one terminal vertex of which is an endpoint of T. □

The foregoing statement is easily seen to be well-defined. Path factor of a tree need not be unique, but every factor (based on our definition) will consist of exactly $p-1$ paths, where p is the number of endpoints in the tree. This topic has been an object of study for long; see [13] and [14]. However, there is no unanimity on the definition. In any event, every formulation may be viewed as a solution to the problem of moving along edges in order to visit each vertex exactly once.

The present discussion is going to be useful in the next section, where we view a tree as a collection of vertex-disjoint paths and employ results of the previous section to obtain a long cycle in each component of $C_m \times T$. The following procedure is relevant.

procedure PathFactor(**input** T: tree; **output** S: sequence of oriented paths);
(* This procedure builds a sequence of oriented paths forming a path factor of T *)
begin
  let $P_{n_1} = a_0 - a_1 - \cdots - a_{n_1-1}$ be a path of T, where $deg_T(a_0) = deg_T(a_{n_1-1}) = 1$;
  $S := \{\boldsymbol{P}_{n_1}\}$, where $\boldsymbol{P}_{n_1}$ is the oriented path $a_0 \to a_1 \to \cdots \to a_{n_1-1}$;
  $T' := P_{n_1}$; $i := 1$;
  while $T' \neq T$ **do begin**
    let u be a vertex of T' such that $deg_{T'}(u) = 2$ and $deg_T(u) = r \geqslant 3$;
    let $P_{n_{i+1}}, \ldots, P_{n_{i+r-2}}$ be vertex–disjoint paths from $(T \setminus T')$ such that each P_{n_k} is of the form $b_0 - \cdots - b_{n_k-1}$, where $\{u, b_0\} \in E(T)$ and $deg_T(b_{n_k-1}) = 1$;
    let $\boldsymbol{P}_{n_{i+1}}, \ldots, \boldsymbol{P}_{n_{i+r-2}}$ be the corresponding oriented paths, that is, $\boldsymbol{P}_{n_k}$ is of the form $b_0 \to \cdots \to b_{n_k-1}$, where $deg_T(b_{n_k-1}) = 1$;
    $S := S \cup \{\boldsymbol{P}_{n_{i+1}}, \ldots, \boldsymbol{P}_{n_{i+r-2}}\}$;
    $T' := \langle T' \cup P_{n_{i+1}} \cup \cdots \cup P_{n_{i+r-2}} \rangle$;
    $i := i + r - 2$
  end; (* **while** *)
end; (* PathFactor *)

Clearly, procedure PathFactor runs in linear time and builds a path factor of T in the sense of Definition 4. A tree T and a path factor of T appear in Fig. 7. Note that each path is being oriented towards that terminal vertex which is an endpoint of T. Purpose is to evolve a deterministic algorithm.

Procedure PathFactor may easily be changed so that the (first) path P_{n_1} in the resulting collection is a longest path of the tree.

4. Main result

We first present a procedure LCycle which constructs a long cycle of $C_m \times T$, where m is odd and T is homeomorphic from $K_{1,r}$, $3 \leqslant r \leqslant m$. The cycle thus constructed is

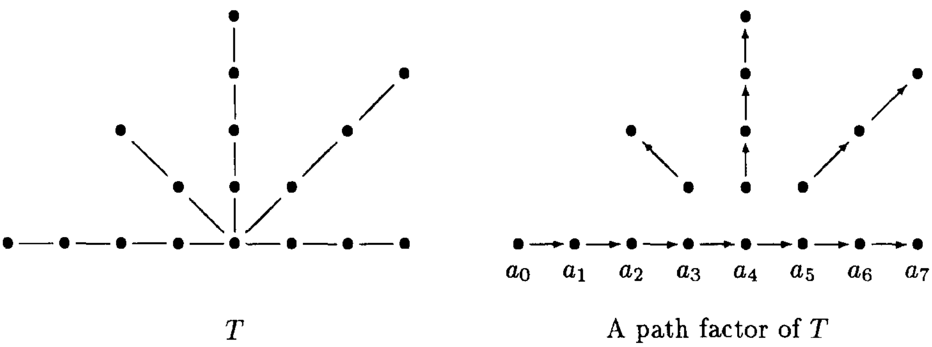

Fig. 7. A tree T and a path factor of T.

shown to be a vertex cover and of length

$$\left(\sum_{i=1}^{r-1} c(C_m \times P_{n_i})\right) - 2 \cdot (s-1),$$

where $\{P_{n_1}, \ldots, P_{n_{r-1}}\}$ is a path factor of T and s is the number of paths in the factor which are different from K_1. We will subsequently present a general scheme which builds on procedure LCycle.

The scheme may intuitively be described as follows. Assuming that T is homeomorphic from $K_{1,r}$ (whence T has r endpoints), let $\boldsymbol{P}_{n_1}, \ldots, \boldsymbol{P}_{n_{r-1}}$ be a sequence of oriented paths of T obtainable by means of procedure PathFactor. First let C be the (longest) cycle of $C_m \times P_{n_1}$ which is similar to the first cycle from Fig. 3/Fig. 4. For $k \geqslant 2$, suppose that P_{n_k} is different from K_1. The algorithm constructs a longest cycle C' of $C_m \times P_{n_k}$, and then builds a cycle D from C and C' by appropriately removing two consecutive edges from each of C and C', and by introducing two new edges between the resulting "horseshoes." D is subsequently reassigned to C, and if $k \leqslant r-2$, then the procedure continues with the next iteration.

procedure LCycle(**input** C_m: odd cycle, T: tree; **output** C: cycle);
(* T is homeomorphic from $K_{1,r}$, $3 \leqslant r \leqslant m$ *)
(* The procedure constructs a long cycle C of $C_m \times T$ *)
begin
(1) let $\boldsymbol{P}_{n_1}, \ldots, \boldsymbol{P}_{n_{r-1}}$ be a sequence of oriented paths of T obtainable
by means of procedure PathFactor;
(* $P_{n_1}, \ldots, P_{n_{r-1}}$ are the corresponding unoriented paths *)
(* T has r endpoints, hence there are $r-1$ paths in the factor *)
(2) let $\boldsymbol{P}_{n_1} = a_0 \to a_1 \to \cdots \to a_{n_1-1}$; (* $deg_T(a_0) = deg_T(a_{n_1-1}) = 1$, $n_1 \geqslant 3$ *)
(3) let C be the cycle of $C_m \times P_{n_1}$ prescribed below:
(3a) **if** $n_1 = 3$, **then** C is the cycle induced by $\{0, \ldots, m-1\} \times \{a_0, a_1\}$;
(3b) **if** $n_1 \geqslant 4$, **then** C is the cycle similar to the first cycle from Fig. 3/Fig. 4;
(* If n_1 is even, then use the first cycle of Fig. 3 as the template,

otherwise use the first cycle of Fig. 4 *)
(* A vertex of the form (s,t) in Fig. 3/Fig. 4 will appear as (s,a_t) here *)
(* Note that C is a longest cycle of $C_m \times P_{n_1}$ *)
(4) let a_i be the vertex of T such that $deg_T(a_i) = r$; (* $1 \leqslant i \leqslant n_1 - 2$ *)
(* a_i must appear on P_{n_1}, and is actually the center of T *)
(* For $2 \leqslant k \leqslant r-1$, $\boldsymbol{P}_{n_k} = b_0 \to \cdots \to b_{n_k-1}$ is such that $\{a_i, b_0\} \in E(T)$ *)
(5a) **if** a_i is such that i is even, **then** $C :=$ CYC–A$(C, a_i, a_{i+1}, \boldsymbol{P}_{n_2}, \ldots, \boldsymbol{P}_{n_{r-1}})$;
(5b) **if** a_i is such that i is odd, **then** $C :=$ CYC–B$(C, a_i, a_{i-1}, \boldsymbol{P}_{n_2}, \ldots, \boldsymbol{P}_{n_{r-1}})$;
end; (* LCycle *)
(***)
function
CYC–A(C: cycle; a_i, a_{i+1}: tree vertex; $\boldsymbol{P}_{j_1}, \ldots, \boldsymbol{P}_{j_s}$: oriented path): cycle;
(* Global variables: m, T *) (* i is even *)
begin
(1) $p := 0$;
$q := m - 1$;
(2) **for** $k := 1$ **to** s **do**
(3) **if** $j_k \geqslant 2$ **then begin**
(4) let $\boldsymbol{P}_{j_k} = b_0 \to \cdots \to b_{j_k-1}$;
(* $j_k \geqslant 2$, $deg_T(b_0) = 2$, $deg_T(b_{j_k-1}) = 1$ and $\{a_i, b_0\} \in E(T)$ *)
(5) let C' be the cycle of $C_m \times P_{j_k}$ specified below:
(5a) **if** $j_k = 2, 3$, **then** C' is the cycle induced by $\{0, \ldots, m-1\} \times \{b_0, b_1\}$;
(5b) **if** $j_k \geqslant 4$, **then** C' is the cycle similar to the first cycle from Fig. 5/Fig. 6;
(* A vertex of the form (c,d) in Fig. 5/Fig. 6 will appear as (c, b_d) here *)
(6) construct a cycle D from C and C' as follows:
(6a) drop the edges (p, a_i) — $(p+1, a_{i+1})$ — $(p+2, a_i)$ from C;
(6b) drop the edges (q, b_0) — $(q+1, b_1)$ — $(q+2, b_0)$ from C';
(6c) introduce the edges (p, a_i) — (q, b_0) and $(p+2, a_i)$ — $(q+2, b_0)$;
(* Vertices $(p+1, a_{i+1})$ and $(q+1, b_1)$ are being bypassed by D *)
(* $|D| = |C| + |C'| - 2$, where $|C|$ denotes the length of C *)
(7) $C := D$;
(8) $p := p+2$; $q := q+2$; (* addition modulo m *)
end; (* **if** $j_k \geqslant 2$ *)
(9) CYC–A:=C
end; (* CYC–A *)
(***)
function
CYC–B(C: cycle; a_i, a_{i-1}: tree vertex; $\boldsymbol{P}_{j_1}, \ldots, \boldsymbol{P}_{j_s}$: oriented path): cycle;
(* Global variables: m, T *) (* i is odd *)
begin
(1) **if** $i \equiv 1 \pmod 4$, **then** $p := 1$ **else** $p := m - 1$;
$q := 0$;
(2) **for** $k := 1$ **to** s **do**

(3) **if** $j_k \geqslant 2$ **then begin**
(4) let $\boldsymbol{P}_{j_k} = b_0 \to \cdots \to b_{j_k-1}$;
(* $j_k \geqslant 2$, $deg_T(b_0) = 2$, $deg_T(b_{j_k-1}) = 1$ and $\{a_i, b_0\} \in E(T)$ *)
(5) let C' be the cycle of $C_m \times P_{j_k}$ specified below:
(5a **if** $j_k = 2, 3$, **then** C' is the cycle induced by $\{0, \dots, m-1\} \times \{b_0, b_1\}$;
(5b) **if** $j_k \geqslant 4$, **then** C' is the cycle similar to the first cycle from Fig. 3/Fig. 4;
(* A vertex of the form (c, d) in Fig. 3/Fig. 4 will appear as (c, b_d) here *)
(6) construct a cycle D from C and C' as follows:
(6a) drop the edges (p, a_i) — $(p+1, a_{i-1})$ — $(p+2, a_i)$ from C;
(6b) drop the edges (q, b_0) — $(q+1, b_1)$ — $(q+2, b_0)$ from C';
(6c) introduce the edges (p, a_i) — (q, b_0) and $(p+2, a_i)$ — $(q+2, b_0)$;
(* Vertices $(p+1, a_{i-1})$ and $(q+1, b_1)$ are being bypassed by D *)
(* $|D| = |C| + |C'| - 2$, where $|C|$ denotes the length of C *)
(7) $C := D$;
(8) $p := p+2$; $q := q+2$; (* addition modulo m *)
end; (* **if** $j_k \geqslant 2$ *)
(9) CYC–B:=C
end; (* CYC–B *)

The following technical lemma is based on Theorem 2.2, and will be useful in the proof of correctness of procedure LCycle.

Lemma 4.1. *Let $m \geqslant 3$, $n \geqslant 4$, where m is odd, and let C (resp. D) be the longest cycle of $C_m \times P_n$ illustrated by the first cycle appearing in Fig. 3/Fig. 4 (resp. Fig. 5/Fig. 6).*

(a) *Cycle C contains $\lfloor n/2 \rfloor$ vertex-disjoint paths, each isomorphic to P_{2m-2}, induced by the sets $V_0, \dots, V_{\lfloor n/2 \rfloor - 1}$, where*

$$V_k = \begin{cases} (\{0, \dots, m-1\} \times \{2k, 2k+1\}) \setminus \{(m-2, 2k), (m-1, 2k+1)\}, & k \text{ even} \\ (\{0, \dots, m-1\} \times \{2k, 2k+1\}) \setminus \{(m-2, 2k), (m-3, 2k+1)\}, & k \text{ odd.} \end{cases}$$

(b) *Cycle D contains $\lfloor n/2 \rfloor$ vertex-disjoint paths, each isomorphic to P_{2m-2}, induced by the sets $W_0, \dots, W_{\lfloor n/2 \rfloor - 1}$, where*

$$W_k = \begin{cases} (\{0, \dots, m-1\} \times \{2k, 2k+1\}) \setminus \{(m-3, 2k), (m-2, 2k+1)\}, & k \text{ even} \\ (\{0, \dots, m-1\} \times \{2k, 2k+1\}) \setminus \{(m-3, 2k), (m-4, 2k+1)\}, & k \text{ odd.} \end{cases}$$

For odd $m \geqslant 3$, the cycle in the graph $C_m \times P_2 / C_m \times P_3$ corresponding to $\{0, \dots, m-1\} \times \{0, 1\}$ also contains a path on $2m-2$ vertices as indicated above.

Proposition 4.2. *Procedure LCycle correctly constructs a cycle C of $C_m \times T$, where m is odd and T is a tree which is homeomorphic from $K_{1,r}$, $1 \leqslant r \leqslant m$.*

Proof. Termination being obvious, we establish correctness. Step (1) invokes procedure PathFactor and obtains a sequence $\boldsymbol{P}_{n_1}, \dots, \boldsymbol{P}_{n_{r-1}}$ of oriented paths corresponding to a path factor of T. Path P_{n_1} (mentioned at Step 2) is different from all else in that it has at least three vertices (including the center of T) and each of its terminal vertices is an endpoint of T. By results of Section 2, statements at Step (3) are sound. C is a longest cycle of $C_m \times P_{n_1}$. Its form is depicted by the first cycle of Fig. 3/Fig. 4. T has exactly one vertex of degree r. Step (4) lets a_i be that vertex. It is clear that a_i must appear on $\boldsymbol{P}_{n_1}$, $1 \leqslant i \leqslant n_1 - 2$.

By Lemma 4.1(a), the following path ($\cong P_{2m-3}$) appears as a part of cycle C:

$$(0,a_i)-(1,a_{i+1})-(2,a_i)-\cdots-(m-1,a_i)-(0,a_{i+1})-(1,a_i)-\cdots-(m-4,a_i), \quad \text{if } i \text{ is even,}$$

$$(1,a_i)-(2,a_{i-1})-(3,a_i)-\cdots-(0,a_i)-(1,a_{i+1})-(2,a_i)-\cdots-(m-3,a_i), \quad \text{if } i \equiv 1 \pmod 4,$$

$$(m-1,a_i)-(0,a_{i-1})-(1,a_i)-\cdots-(m-1,a_{i-1})-(0,a_i)-(1,a_{i-1})-\cdots-(m-5,a_i), \quad \text{if } i \equiv 3 \pmod 4.$$

For the purpose of this discussion, the oriented paths $\boldsymbol{P}_{j_1},\ldots,\boldsymbol{P}_{j_s}$ appearing in the body of function CYC–A/CYC–B may, respectively, be referred to as oriented paths $\boldsymbol{P}_{n_2},\ldots,\boldsymbol{P}_{n_{r-1}}$, which appear in the calling procedure.

Depending on the parity of index i of a_i, exactly one of the functions CYC–A and CYC–B is invoked at Step (5). First suppose that i is even so that control transfers to CYC–A.

Step (1) of CYC–A initializes two integer variables p and q. The "for" loop, which runs from Step (2)–(8), consists of $r-2$ iterations corresponding to the paths $P_{n_2},\ldots,P_{n_{r-1}}$. At each iteration, a longest cycle of $C_m \times P_{n_k}$ is appropriately coupled with the existing cycle C to yield a longer cycle. If $n_k = 1$, then Steps (4)–(8) are skipped, since $C_m \times P_1$ (which is an edgeless graph) cannot contribute anything. Assume that $n_k \geqslant 2$. Step (5) constructs a longest cycle C' of $C_m \times P_{n_k}$. If $n_k \geqslant 4$, then C' is similar to the first cycle from Fig. 5/Fig. 6. This is in contrast to the form of the cycle at Step (3) of the calling procedure. By Lemma 4.1(b), the following path ($\cong P_{2m-3}$) appears as a part of cycle C':

$$(m-1,b_0)-(0,b_1)-(1,b_0)-\cdots-(m-1,b_1)-(0,b_0)-(1,b_1)-\cdots-(m-5,b_0).$$

We now examine Step (6) of CYC–A. The reader may verify that at each iteration of the "for" loop, the following holds: (i) the segment $(p,a_i)-(p+1,a_{i+1})-(p+2,a_i)$ appears in cycle C, (ii) the segment $(q,b_0)-(q+1,b_1)-(q+2,b_0)$ appears in cycle C', (iii) $\{p,q\} \in E(C_m)$, and (iv) the edges $(p,a_i)-(q,b_0)$ and $(p+2,a_i)-(q+2,b_0)$ appear in $C_m \times T$ but not in C or C'. Consequently, Step (6) is sound. The cycle D thus constructed is reassigned to C at Step (7). Subsequently, each of the two integer variables p and q is incremented by two. This is done to ensure correct couplings of cycles C and C' across two different iterations.

Coupling of cycles is done along a path (having $2m-3$ vertices) of the cycle constructed at Step (3) of the calling procedure. It is easy to check that at most $m-2$ such couplings are possible. This is the reason why $\Delta(T) \leqslant m$. Fig. 8 is a schematic diagram for the case when $m = 5$ and T is homeomorphic from $K_{1,5}$.

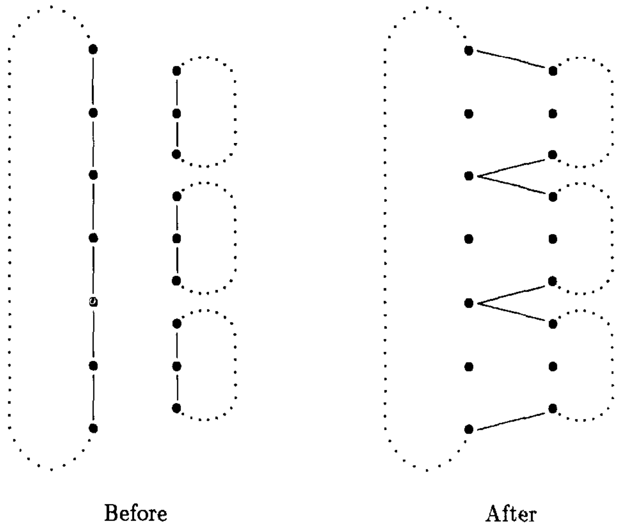

Fig. 8. Cycle construction based on procedure LCycle.

If a_i is such that i is odd, then control reaches function CYC–B, which is analogous to CYC–A. The only important difference between the two functions is this: While the cycle at Step (5b) of CYC–A is after the first cycle from Fig. 5/Fig. 6, that at Step (5b) of CYC–B is after the first cycle from Fig. 3/Fig. 4. Details are omitted. □

A useful remark on procedure LCycle, and functions CYC–A and CYC–B is in order. By systematically replacing "Fig. 3/Fig. 4" by "Fig. 5/Fig. 6" and vice versa, and by suitably changing initial values of the integer variables p and q at Step (1) of CYC–A/CYC–B, we can have a scheme which does exactly the same job. The resulting procedure LCycle′, and functions CYC–A′ and CYC–B′ appear below.

procedure LCycle′(**input** C_m: odd cycle, T: tree; **output** C: cycle);
(* T is homeomorphic from $K_{1,r}$, $3 \leqslant r \leqslant m$ *)
(* The procedure constructs a long cycle C of $C_m \times T$ *)
begin
⋮
(3b) **if** $n_1 \geqslant 4$, **then** C is the cycle similar to the first cycle from Fig. 5/Fig. 6;
⋮
(5a) **if** a_i is such that i is even, **then** $C :=$ CYC–A′$(C, a_i, a_{i+1}, \boldsymbol{P}_{n_2}, \ldots, \boldsymbol{P}_{n_{r-1}})$;
(5b) **if** a_i is such that i is odd, **then** $C :=$ CYC–B′$(C, a_i, a_{i-1}, \boldsymbol{P}_{n_2}, \ldots, \boldsymbol{P}_{n_{r-1}})$;
(* All missing steps are identical to the corresponding steps of procedure LCycle *)
end; (* LCycle′ *)
(***)
function

CYC–A′(C: cycle; a_i, a_{i+1}: tree vertex; $\boldsymbol{P}_{j_1}, \ldots, \boldsymbol{P}_{j_s}$: oriented path): cycle;
(* Global variables: m, T *) (* i is even *)
begin
(1) $p := m - 1$;
$q := 0$;
⋮
(5b) **if** $j_k \geqslant 4$, **then** C' is the cycle similar to the first cycle from Fig. 3/Fig. 4;
⋮
(9) CYC–A′:=C
(* All missing steps are identical to the corresponding steps of function CYC–A *)
end; (* CYC–A′ *)
(***)
function
CYC–B′(C: cycle; a_i, a_{i-1}: tree vertex; $\boldsymbol{P}_{j_1}, \ldots, \boldsymbol{P}_{j_s}$: oriented path): cycle;
(* Global variables: m, T *) (* i is odd *)
begin
(1) **if** $i \equiv 1 \pmod 4$, **then** $p := 0$ **else** $p := m - 2$;
$q := m - 1$;
⋮
(5b) **if** $j_k \geqslant 4$, **then** C' is the cycle similar to the first cycle from Fig. 5/Fig. 6;
⋮
(9) CYC–B′:=C
(* All missing steps are identical to the corresponding steps of function CYC–B *)
end; (* CYC–B′ *)

The length of the cycle constructed by procedure LCycle/LCycle′ is easily seen to be $(\sum_{i=1}^{r-1} c(C_m \times P_{n_i})) - 2 \cdot (s-1)$, where $\{P_{n_1}, \ldots, P_{n_{r-1}}\}$ is a path factor of T and s is the number of paths in the factor which are different from K_1. Clearly, cycle length is dependent on path factor of T, which is far from unique. In any event, the cycle traced is a vertex cover, since the vertices missed are mutually nonadjacent.

The following is a relevant remark with respect to procedure LCycle/LCycle′. Let a_i be the center of the tree T, as stated at Step (4). The vertices $(0, a_i), \ldots, (m-1, a_i)$ of $C_m \times T$ are all of (maximum) degree $2r$ while remaining vertices are of degree two or four. The reader may verify that all vertices of maximum degree appear on the cycle constructed by procedure LCycle/LCycle′. See Broersma et al. [3] for useful characteristics of a cycle which includes vertices of maximum degree.

Timing analysis of procedure LCycle/LCycle′ is in order. Important work consists of obtaining oriented paths $\boldsymbol{P}_{n_1}, \ldots, \boldsymbol{P}_{n_{r-1}}$ corresponding to a path factor of T, and constructing a longest cycle of each of $C_m \times P_{n_1}, \ldots, C_m \times P_{n_{r-1}}$. As stated in Section 3, path factor is obtainable in time linear in the size of T. Further, a longest cycle of $C_m \times P_{n_i}$ is constructible in time proportional to $m \cdot n_i$. (Precise sequences appear in [9] with respect to a longest cycle in $C_m \times P_k$.) Thus, the running time is bounded by a constant multiple of $m \cdot (n_1 + \ldots + n_{r-1}) = m \cdot n$, where n is the number of vertices in T. In other words, the procedure runs in time linear in the size of $C_m \times T$.

We are now ready to present a general scheme which builds on procedures LCycle and LCycle′, and constructs a long cycle in $C_m \times T$. T is viewed as a graph consisting of subgraphs, each of which is homeomorphic from some $K_{1,r}$ where $2 \leqslant r \leqslant m$.

Definition 5. Let T be a tree having p endpoints, which admits of a factor into a sequence $\boldsymbol{P}_{n_1}, \ldots, \boldsymbol{P}_{n_{p-1}}$ of oriented paths (obtainable by means of procedure PathFactor) such that the following conditions are satisfied:

1. If u, v are vertices of T of degree $\geqslant 3$ and u, v appear on distinct oriented paths, then $dist_T(u, v) \geqslant 3$, and
2. If $\boldsymbol{P}_{n_i} = b_0 \to \cdots \to b_{n_i - 1}$, then for $0 \leqslant j \leqslant \lfloor n_i/2 \rfloor - 1$, at most one of b_{2j} and b_{2j+1} is of degree $\geqslant 3$ in T. □

If T is a tree such that any two distinct vertices of T which are of degree $\geqslant 3$ are at a distance of at least three, then T necessarily satisfies Definition 5.

procedure LongCycle(**input** C_m: odd cycle, T: tree; **output** C: cycle);
(* This procedure constructs a long cycle C of $C_m \times T$ where $\Delta(T) \leqslant m$ and
  T satisfies Definition 5 *)
begin
(1) let $P_{n_1} = d_0 — \cdots — d_{n_1-1}$ be a path in T, where
  $deg_T(d_0) = deg_T(d_{n_1-1}) = 1$, $n_1 \geqslant 3$;
(1a) let $\boldsymbol{P}_{n_1} = d_0 \to \cdots \to d_{n_1-1}$ be the oriented path corresponding to P_{n_1};
(2) let C be the cycle of $C_m \times P_{n_1}$ prescribed below:
(2a) **if** $n_1 = 3$, **then** C is the cycle induced by $\{0, \ldots, m-1\} \times \{d_0, d_1\}$;
(2b) **if** $n_1 \geqslant 4$, **then** C is the cycle similar to the first cycle from Fig. 3/Fig. 4;
  (* A vertex of the form (s, t) in Fig. 3/Fig. 4 will appear as (s, d_t) here *)
  (* Note that C is a longest cycle of $C_m \times P_{n_1}$ *)
(3) let $\boldsymbol{P}_{n_1}$ be colored blue;
(4) $T' := P_{n_1}$; $S := \{\boldsymbol{P}_{n_1}\}$;
  (* T' is a subtree of T which grows at every iteration of the algorithm *)
  (* S is a set of vertex-disjoint oriented paths of T' *)
  (* At end, T' will equal T while S will correspond to a vertex
    decomposition of T *)
(5) **while** $T' \neq T$ **do begin**
(6)   let u be a vertex of T' such that $deg_{T'}(u) = 2$ and $deg_T(u) = r \geqslant 3$;
(7)   let $\boldsymbol{P}_{n_j} = a_0 \to \cdots \to a_{n_j-1}$ be the oriented path which contains u;
    (* The oriented path $\boldsymbol{P}_{n_j}$ must appear in the set S *)
(7a)   let $u = a_i$, $1 \leqslant i \leqslant n_j - 2$;

(8) let $\boldsymbol{P}_{k_1}, \ldots, \boldsymbol{P}_{k_{r-2}}$ be the vertex-disjoint oriented paths from $(T \backslash T')$ such that each $\boldsymbol{P}_{k_t}$ is of the form $b_0 \rightarrow \cdots \rightarrow b_{k_t-1}$, where $\{a_i, b_0\} \in E(T)$ and $deg_T(b_{k_t-1}) = 1$;

(9) systematically couple a longest cycle of each of $C_m \times P_{k_1}, \ldots, C_m \times P_{k_{r-2}}$ to the cycle C as follows:

(9a) **if** $\boldsymbol{P}_{n_j}$ is colored blue and i is even, **then** $C :=$ CYC–A$(C, a_i, a_{i+1}, \boldsymbol{P}_{k_1}, \ldots, \boldsymbol{P}_{k_{r-2}})$;

(9b) **if** $\boldsymbol{P}_{n_j}$ is colored blue and i is odd, **then** $C :=$ CYC–B$(C, a_i, a_{i-1}, \boldsymbol{P}_{k_1}, \ldots, \boldsymbol{P}_{k_{r-2}})$;

(9c) **if** $\boldsymbol{P}_{n_j}$ is colored black and i is even, **then** $C :=$ CYC–A$'(C, a_i, a_{i+1}, \boldsymbol{P}_{k_1}, \ldots, \boldsymbol{P}_{k_{r-2}})$;

(9d) **if** $\boldsymbol{P}_{n_j}$ is colored black and i is odd, **then** $C :=$ CYC–B$'(C, a_i, a_{i-1}, \boldsymbol{P}_{k_1}, \ldots, \boldsymbol{P}_{k_{r-2}})$;

(10) color each of $\boldsymbol{P}_{k_1}, \ldots, \boldsymbol{P}_{k_{r-2}}$ as follows:

(10a) **if** i is even, **then** the color assigned is different from that of $\boldsymbol{P}_{n_j}$;
(* There are just two colors: blue and black *)

(10b) **if** i is odd, **then** the color assigned is same as that of $\boldsymbol{P}_{n_j}$;
(* All of $\boldsymbol{P}_{k_1}, \ldots, \boldsymbol{P}_{k_{r-2}}$ are colored alike *)

(11) $T' := \langle T' \cup P_{k_1} \cup \cdots \cup P_{k_{r-2}} \rangle$; $S := S \cup \{\boldsymbol{P}_{k_1}, \ldots, \boldsymbol{P}_{k_{r-2}}\}$

end; (* **while** *)

end; (* LongCycle *)

Proposition 4.3. *Procedure LongCycle correctly constructs a cycle C of $C_m \times T$, where m is odd and T is a tree such that $\Delta(T) \leqslant m$ and T satisfies Definition 5.*

Proof. We present an inductive argument for correctness of the algorithm. Important book-keeping consists of maintaining a subtree T' of T and a set S of vertex-disjoint oriented paths of T'. Each oriented path that appears in S is colored either blue or black. Both T' and S grow at every iteration of the algorithm. Inductive assertion consists of the following:

- C is a cycle of $C_m \times T'$.
- S consists of oriented paths which constitute a vertex decomposition of T'.
- Each oriented path $a_0 \rightarrow \cdots \rightarrow a_{r-1}$ in S is such that $a_0 — \cdots — a_{r-1}$ is a path in T, where $deg_T(a_{r-1}) = 1$.
- Suppose that $\boldsymbol{P}_r = a_0 \rightarrow \cdots \rightarrow a_i \rightarrow \cdots \rightarrow a_{r-1}$ is an oriented path in the set S such that $deg_{T'}(a_i) = 2$ and $deg_T(a_i) \geqslant 3$. If $\boldsymbol{P}_r$ is colored blue (resp. black), then cycle C contains a path P on $2m-2$ vertices similar to that mentioned in Lemma 4.1(a) (resp. Lemma 4.1(b)), where vertices on P are of the form $(j, a_i)/(j, a_{i+1})$, if i is even and $(j, a_{i-1})/(j, a_i)$, if i is odd.

Step (1) of the algorithm lets P_{n_1} be the path in a factor of T, where each terminal vertex of P_{n_1} is an endpoint of T. Let $\boldsymbol{P}_{n_1}$ be the corresponding oriented path. At Step (2), a longest cycle C of $C_m \times P_{n_1}$ is constructed, and at the next step, $\boldsymbol{P}_{n_1}$ is colored blue. Step (4) consists of initializations to (sub)tree T' and set S. It is easy to see that induction basis holds.

Let us examine a typical iteration of the "while" loop. Step (6) selects a vertex u of T' such that $deg_{T'}(u) = 2$ and $deg_T(u) = r \geqslant 3$. It is clear that such a vertex exists in T'. Step (7) lets $\boldsymbol{P}_{n_j} = a_0 \rightarrow \cdots \rightarrow a_{n_j-1}$ be the (unique) oriented path that appears in S and contains u, where $n_j \geqslant 3$. Suppose that $u = a_i$, $1 \leqslant i \leqslant n_j - 2$. Vertex a_i has r neighbors in T. Two of them, viz a_{i-1} and a_{i+1}, appear on $\boldsymbol{P}_{n_j}$, and hence in T'. Each of the remaining $r-2$ neighbors of a_i appears as the first vertex on an oriented path $\boldsymbol{P}_{k_t}$, $1 \leqslant t \leqslant r-2$. This is what is stated at Step (8).

Most of the work is done at Step (9). Depending on the color of $\boldsymbol{P}_{n_j}$ and parity of i, exactly one of (9a)–(9d) is executed, and a longest cycle of each of $C_m \times P_{k_1}, \ldots, C_m \times P_{k_{r-2}}$ is coupled with the existing cycle C leading to a longer cycle, which is again being called C. Soundness of this step follows from induction hypothesis and from correctness of procedure LCycle/LCycle′ presented earlier.

At Step (10), the oriented paths $\boldsymbol{P}_{k_1}, \ldots, \boldsymbol{P}_{k_{r-2}}$ are colored alike. If i is even (resp. odd), then the color assigned is different from (resp. same as) that of $\boldsymbol{P}_{n_j}$. The purpose of coloring an oriented path is to remember whether function CYC–A/CYC–B or function CYC–A′/CYC–B′ was used in conjunction with that path. At Step (11), (sub)tree T' and set S are appropriately updated.

What remains to be shown is that there is no interference between couplings of cycles at Step (9) across two different iterations of the algorithm. Here is where the two conditions on the structure of T, stated in Definition 5, come to the fore. Note that a total of $r-2$ cycles are coupled to the existing cycle at Step (9), where $3 \leqslant r \leqslant m$. Let a_i be the vertex of degree r of the oriented path $\boldsymbol{P}_{n_j}$, as stated at Step (7). If i is even (resp. odd), then vertices a_i, a_{i+1} (resp. a_i, a_{i-1}) of this oriented path participate in the coupling during this iteration. Condition (2) of Definition 5 ensures that these two vertices are not relevant during any other iteration.

Further, if an oriented path $\boldsymbol{P}_{k_t} = b_0 \to \cdots \to b_{k_t - 1}$ is one of the $r-2$ paths mentioned at Step (9), then the vertices of $\boldsymbol{P}_{k_t}$ which take part in the coupling at that point are b_0 and b_1. By condition (1) of Definition 5, both b_0 and b_1 are of degree at most two, and it is straightforward to see that these two vertices will not participate in any other coupling of cycles. For example, if b_2 (or b_3) is of degree $\geqslant 3$, then the vertices of $\boldsymbol{P}_{k_t}$ participating during that iteration will be b_2 and b_3.

To conclude, note that the cycle C, (sub)tree T' and set S obtained at the end of the "while" loop conform to the inductive assertion. □

It is straightforward to see that the cycle of $C_m \times T$ constructed by procedure Long-Cycle is of length

$$\left(\sum_{i=1}^{p-1} c(C_m \times P_{n_i}) \right) - 2 \cdot (s-1),$$

where p is the number of endpoints in T, $\{P_{n_1}, \ldots, P_{n_{p-1}}\}$ is a path factor of T and s is the number of paths in the factor which are different from K_1. Further, the cycle traced is a dominating set, since every missed vertex is adjacent to some vertex on the cycle. In fact, in many cases, the cycle will be a vertex cover. The cycle thus constructed has the additional characteristic that it includes all vertices of maximum degree. Note also that cycle length is dependent on the path factor.

As stated in Section 3, procedure PathFactor could be slightly modified so that it would always include a longest path in the collection it builds. In this process, diameter of T will appear in the lower bound on $c(C_m \times T)$.

Timing analysis of procedure LongCycle is similar to that of LCycle/LCycle′ presented earlier. The amount of work is governed by construction of a longest cycle of each of $C_m \times P_{n_1}, \dots, C_m \times P_{n_{p-1}}$, where $\{P_{n_1}, \dots, P_{n_{p-1}}\}$ is a path factor of the tree. Since a longest cycle of $C_m \times P_{n_i}$ is obtainable in time proportional to $m \cdot n_i$ (see [9]), it follows that the running time of the procedure is $O(mn)$, where $n = n_1 + \dots + n_{p-1} =$ number of vertices in T. In other words, the algorithm runs in time linear in the size of $C_m \times T$. This is the best that a sequential algorithm can accomplish with respect to a nontrivial graph problem.

By Theorem 1.1(3), if m is even, then $C_m \times T$ consists of two isomorphic components. We have the following result.

Theorem 4.4. *Let m be even $\geqslant 4$. If T is a tree such that $\Delta(T) \leqslant m/2$ and T admits of a path factor $\{P_{n_1}, \dots, P_{n_{p-1}}\}$ in the sense of Definition 5, then each component of $C_m \times T$ contains a cycle whose length is $(\sum_{i=1}^{p-1} c(C_m \times P_{n_i})) - 2 \cdot (s-1)$, where p is the number of endpoints in T and s is the number of paths in the factor which are different from K_1.* □

A proof of Theorem 4.4 consists of appropriately adapting procedures LCycle, LCycle′ and LongCycle. Details are being omitted.

At this point, suppose that T is a tree which does not conform to Definition 5. In order to trace a long cycle in $C_m \times T$, we may first "trim" T to obtain a subtree T' which is as large as possible and which satisfies that definition. Procedure LongCycle may subsequently act on C_m and T' and return a long cycle C of $C_m \times T'$. Clearly, C will be a cycle in $C_m \times T$ as well.

5. Concluding remarks

Tracing a longest cycle or a longest path in a graph is one of the classical combinatorial problems, with potential applications. This paper addresses this question with respect to $C_m \times T$, which is a bipartite graph.

It is demonstrated in Section 1 that the length of a longest cycle/longest path in $C_m \times T$ critically depends on the structure of T. Section 2 recapitulates and illustrates important results from [9] on edge decompositions of $C_m \times P_n$ into long cycles and long paths, respectively. Further, characterizations are established for the existence of a Hamiltonian cycle and for that of a Hamiltonian path in $C_m \times T$.

Towards developing an algorithm for a long cycle in $C_m \times T$, we view a tree having p endpoints as a collection of $p-1$ paths, which constitute a vertex decomposition of T. The resulting path factor apppears in Section 3.

Main result of this paper consists of a scheme LongCycle in Section 4, which traces a long cycle in $C_m \times T$, where m is odd and T is a tree that satisfies certain degree constraints. The algorithm carefully employs results of the preceding sections. The scheme thus obtained is easily adaptable for the case when m is even.

Certain questions arise: (1) Several results on $c(C_m \times T)$ are stated as inequalities; are there examples where equality fails?, (2) Is it possible to devise an improvement upon procedure LongCycle so that the algorithm traces a long cycle of $C_m \times T$, where (a) conditions on the input tree T, as stated in Definition 5, are weakened, and (b) the cycle thus traced is still a vertex cover/dominating set?

Acknowledgement

We are grateful to the referee whose detailed commentary on the earlier draft led to a substantial improvement in the presentation of the paper.

References

[1] V. Batagelj and T. Pisanski, Hamiltonian circuits in the Cartesian product of a tree and a cycle, Discrete Math. 38 (1982) 311–312.

[2] J.-C. Bermond, A. Germa and M.C. Heydemann, Hamiltonian cycles in strong products of graphs, Canad. Math. Bull. 22 (1979) 305–309.

[3] H.J. Broersma, J. van den Heuvel, H.A. Jung and H.J. Veldman, Cycles containing all vertices of maximum degree, J. Graph Theory 17 (1993) 373–385.

[4] M. Farzan and D. Waller, Kronecker products and local joins of graphs, Canad. J. Math. 29 (1977) 255–269.

[5] M.R. Garey and D.S. Johnson, Computers and Intractability: A Guide to the Theory of NP-Completeness (W.H. Freeman and Co, San Francisco, CA, 1979).

[6] F. Harary, Graph Theory (Addison-Wesley, Reading, MA, 1969).

[7] S.T. Hedetniemi, Homomorphisms of graphs and automata, Tech. Report 03105-44-T, Univ. of Michigan, Ann Arbor, MI, 1966.

[8] W. Imrich and H. Izbicki, Associative products of graphs, Montas Math. 80 (1975) 277–281.

[9] P.K. Jha, Decompositions of the Kronecker product of a cycle and a path into long cycles and long paths, Indian J. Pure Appl. Math. 23 (1992) 585–602.

[10] P.K. Jha, S. Klavžar and B. Zmazek, Isomorphic components of Kronecker product of bipartite graphs, submitted for publication, 1994.

[11] P.K. Jha and G. Slutzki, A note on outerplanarity of product graphs, Zastos Math. 21 (1993) 537–544.

[12] H.A Jung, S. Kyaw and B. Wei, Almost-Hamiltonian graphs, in: R. Bodendiek, ed., Contemporary Methods in Graph Theory (Bibliographisches Institut, Mannheim, 1990) 409–427.

[13] P.J Slater, Path coverings of the vertices of a tree, Discrete Math. 25 (1979) 65–74.

[14] Z. Skupien and W. Zygmunt, Generating all maximum path-factors of a tree, in: Colloq. Math. Soc Janos Bolyai 42, Algebra, Combinatorics and Logic in Computer Science (North-Holland, Amsterdam, 1986) 711–717.

[15] P.M. Weichsel, The Kronecker product of graphs, Proc. Am. Math. Soc. 13 (1962) 47–52.