OBSTRUCTIONS FOR 2-MÖBIUS BAND EMBEDDING EXTENSION PROBLEM*

MARTIN JUVAN†‡ AND BOJAN MOHAR†‡

Abstract. Let $K = C \cup e_1 \cup e_2$ be a subgraph of G, consisting of a cycle C and disjoint paths e_1 and e_2, connecting two interlacing pairs of vertices in C. Suppose that K is embedded in the Möbius band in such a way that C lies on its boundary. An algorithm is presented which in linear time extends the embedding of K to an embedding of G, if such an extension is possible, or finds a "nice" obstruction for such embedding extensions. The structure of obtained obstructions is also analysed in details.

Key words. surface embedding, obstruction, Möbius band, algorithm

AMS subject classifications. 05C10, 05C85, 68Q20

1. Introduction. Let K be a subgraph of a graph G. A K-*bridge* (or a K-*component*) in G is a subgraph of G which is either an edge $e \in E(G) \setminus E(K)$ (together with its endpoints) which has both endpoints in K, or it is a connected component of $G - V(K)$ together with all edges (and their endpoints) between this component and K. Each edge of a K-bridge B having an endpoint in K is a *foot* of B. The vertices of $B \cap K$ are the *vertices of attachment* of B. A vertex of K of degree in K different from 2 is a *main vertex* of K. For convenience, if a connected component of K is a cycle, then we choose an arbitrary vertex of it and declare it to be a main vertex of K as well. A *branch* of K is any path (possibly a closed path) in K whose endpoints are main vertices but no internal vertex on this path is a main vertex. If a K-bridge has all vertices of attachment on a single branch of K, it is said to be *local*.

This paper is part of a larger project [JMM, M4] which shows that there is a linear time algorithm to construct embeddings of graphs in an arbitrary (fixed) surface, generalizing the well-known Hopcroft-Tarjan algorithm [HT] for testing planarity in linear time. Our algorithms rely on the theory of bridges: a subgraph K of G is embedded in the surface and then this embedding is either extended to an embedding of G, or an obstruction for such extensions is found. In this paper we solve and analyse a particular case of this problem where the underlying surface is the Möbius band dissected by K into two faces. It is shown that obstructions for extending the embedding of K are either small, or have a very special (millipede) structure. Moreover, finding an embedding extension or such an obstruction requires only linear time (Theorem 5.3).

These results are used and extended in [JM] and [M1]. Related results are also obtained in [M1, M2].

In our algorithms, we consider embeddings of graphs. In case of orientable surfaces, embeddings can be described combinatorially [GT] by specifying a *rotation system*: for each vertex v of the graph G we have cyclic permutation π_v of its neighbors, representing their circular order around v on the surface. Although the Möbius band is non-orientable, such a presentation suffices also in our case since it is enough to specify rotation system in each of the faces of the chosen embedding of K. In order

* This article appeared in *SIAM J. Discrete Math.* 10 (1997) 57–72.

† Supported in part by the Ministry of Science and Technology of Slovenia, Research Project P1–0210–101–94.

‡ Department of Mathematics, University of Ljubljana, Jadranska 19, 1111 Ljubljana, Slovenia

to make a clear presentation of our algorithm, we have decided to use this description only implicitly. Whenever we say that we have an embedding, we mean such a combinatorial description.

Concerning the time complexity of our algorithms, we assume a random-access machine (RAM) model with unit cost for basic operations. This model was introduced by Cook and Reckhow [CR]. More precisely, our model is the *unit-cost* RAM where operations on integers, whose value is $O(n)$, need only constant time (n is the size of the given graph).

2. Parallel computations with constant time overhead. We will need the following simulation of parallelism performed on a unit-cost RAM. At certain steps of our algorithm we will not be able to decide in advance between two possible choices. In such a case we will continue computations simultaneously in both directions. This will enable us to efficiently choose between the two alternatives. During such parallel computations no new parallelism will be introduced.

Denote by $\mathcal{P}_1$ and $\mathcal{P}_2$ both parallel processes. During the parallel computation exactly one of the following three cases will occur:

(i) The process $\mathcal{P}_1$ terminates *successfully*. This means that at the beginning of the parallelism the decision for $\mathcal{P}_1$ would be the right one. In this case, we say that the parallel computation terminates *successfully*. We also stop $\mathcal{P}_2$ (if still active) and restore the memory to the state before starting parallelism, choose the alternative $\mathcal{P}_1$ as the proper one and continue with (non-parallel) computation from this point on.

(ii) If $\mathcal{P}_2$ terminates successfully, then we act as in the previous case, except that we stop $\mathcal{P}_1$ and choose the second alternative as the right one.

(iii) If none of $\mathcal{P}_1, \mathcal{P}_2$ terminates successfully, then the parallel computation is said to terminate *non-successfully*.

If one of the processes fails, we still continue to run the remaining one. If it succeeds, case (i) or (ii) occurs; if also the other process fails, we have case (iii).

In our application of parallelism, the processes $\mathcal{P}_1$ and $\mathcal{P}_2$ will try to extend a partial embedding of a graph in two different ways. If appropriate embedding extension is found by one of them, this process will be termed as successful. Otherwise an obstruction for a particular type of embedding extension problem will be found. In case (iii) the "union" of both obstructions will give rise to a more general obstruction.

We want to ensure that the amount of time spent by both processes is proportional to the work done by either of them. To reach this goal, the actual implementation proceeds as follows. Each parallel process will have only read access to the memory of the main process (*global memory*) and also its own "copy" of this memory (*local memory*). Because of the restrictions on the time spent by the parallel computations we do not copy the data from the global memory to the process' local memory. Otherwise it might happen that the process performs only a small amount of work and then terminates successfully, therefore the amount of work done at this parallel session is small, while copying the whole graph and auxiliary structures to the local memory could take time proportional to the size of the input. To avoid these time consuming operations we propose the following simple memory management for local memory. Each cell in the local memory is either *empty* or *occupied*. If it is empty, this means that its corresponding cell in the global memory would still have the initial contents if the current parallel process would be performed on the global memory. If it is occupied, its new contents is stored in the local memory, so that the global memory remains unchanged. When requiring contents of a cell, the current process first checks in the local memory if the cell is empty of occupied. If it is empty, it reads the contents from the corresponding cell in the global memory. Otherwise it takes data from the local memory. New cell contents is always stored in the process' local memory.

To be able to efficiently delete the contents of parallel process' local memory after the termination of the process (and so prepare it for another parallel session) each parallel process is associated with a list of occupied cells in its local memory. When deleting the contents of the local memory, only these cells need to be considered. (Only the very first "cleaning" is done by the main process in the initialization phase of the algorithm.) Initially, at the start of the parallel process, all cells in the local memory are empty. Moreover, the list of occupied cells is also empty. When, during the computation, an empty cell becomes occupied, the list is updated accordingly.

It is obvious that the above memory management adds only constant time overhead to every operation performed by the parallel process. Moreover, the final "cleaning" of the local memory needs at most time proportional to the amount of work performed by the process.

It can be shown that parallelism can be realized on the standard RAM although we do not have access to the program counter. The time complexity increases by a constant factor (depending on the length of the program) in order to maintain parallelism.

Let us mention at the end that the above method of choosing among alternatives by testing them in parallel could also be (equally efficiently) implemented when the number of alternatives is constant (but possibly greater than two).

3. Obstructions. Let K be a fixed graph embedded in some surface. *Embedding extension problem* asks if for a given graph $G \supseteq K$ it is possible to extend the chosen embedding of K to an embedding of G. A subgraph Ω of $G - E(K)$ is an *obstruction* (for embedding extensions of K to G) if there is no embedding of $K \cup \Omega$ extending the chosen embedding of K. Because of Lemma 4.1 we will be able to assume that all obstruction we will work with contain only entire K-bridges. Moreover, we will be only interested in *minimal* obstructions, i.e., obstructions in which no bridge is redundant. It will turn out that for our particular case of embedding extension problem, minimal obstructions can be precisely characterized. They are either small, i.e. composed of a small number of bridges, or (although arbitrarily large) of a very special form which will be introduced in the sequel.

Let $K = C \cup e_1 \cup e_2$ be a graph homeomorphic to K_4, where C is a cycle and e_1, e_2 are disjoint paths connecting pairs of interlacing vertices in C. Suppose that K is 2-cell embedded in the Möbius band in such a way that C lies on its boundary. Denote by F_1 and F_2 the faces of K under this embedding (cf. Figure 1). We say that K-bridges B and B' *overlap* in a face of K if they cannot be simultaneously embedded in that face.

For the purpose of the following definitions we will assume that all bridges of K in G are small (Lemma 4.1). If this were not the case, the bridges B_i° appearing in the definitions should be replaced by their H-subgraphs (cf. [M2, M3]).

A *thin millipede* in G *based* on e_1 and with *apex* $x \in V(e_2)$ is a subgraph M of $G - E(K)$ which can be expressed as $M = B_1^\circ \cup \cdots \cup B_m^\circ$ ($m \geq 7$) where:

(M1) Each of B_1° and B_m° is a K-bridge in G. Moreover, $B_1^\circ \cup B_2^\circ \cup B_3^\circ$ is uniquely embeddable in $F_1 \cup F_2$. Let F_α be the face containing B_1° under this embedding. Similarly, $B_{m-2}^\circ \cup B_{m-1}^\circ \cup B_m^\circ$ is uniquely embeddable, and let F_β be the face containing B_m°. If m is even, then $\alpha = \beta$. If m is odd, then $\alpha \neq \beta$.

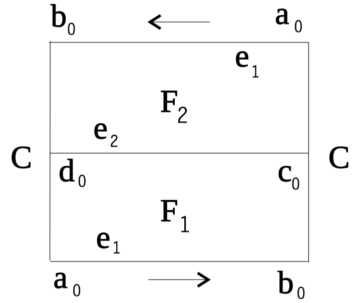

FIG. 1. *Embedding of K in the Möbius band*

(M2) $B_2^\circ, \ldots, B_{m-1}^\circ$ are distinct K-bridges that are attached to e_1 and to x and are not attached to K elsewhere.
(M3) For each $i = 1, 2, \ldots, m-1$, B_i° and B_{i+1}° overlap in F_1 and in F_2.
(M4) For $i > 1$ and $i + 2 \le j < m$, B_i° and B_j° can be simultaneously embedded in F_1 and in F_2. The same holds when $i = 1$ and $3 \le j < m$ for the face F_α. Similarly, B_i° $(1 < i \le m-2)$ and B_m° can be simultaneously embedded in F_β. Additionally, $B_1^\circ \cup B_m^\circ$ can be embedded in $F_\alpha \cup F_\beta$.

It is clear by (M1) and (M3) that a thin millipede M obstructs embedding extensions of K to G.

Our notion of millipedes slightly differs from the concept of millipedes introduced in [M2]. The millipedes in [M2] can be shorter (i.e., $m < 7$ is allowed) and their subgraphs B_i° are allowed to be proper subgraphs of bridges in order that millipedes become minimal obstruction (with respect to the graph inclusion). On the other hand, after eliminating redundant branches in bridges B_i°, we can get from our thin millipedes a millipede in the sense of [M2].

We will also need *skew millipedes* based on e_1. They are defined similarly as thin millipedes. The apex of a thin millipede is replaced by a pair of vertices $x, y \in V(e_2)$ where no K-bridge is attached to e_2 on the (open) segment between x and y. The bridges $B_1^\circ, B_2^\circ, \ldots, B_m^\circ$ satisfy (M1) and (M3), while (M2) and (M4) are replaced by:
(M2′) $B_2^\circ, \ldots, B_{m-1}^\circ$ are distinct K-bridges. If i is even $(1 < i < m)$, then B_i° is attached to e_1 and to x (and not elsewhere). If i is odd $(1 < i < m)$, then B_i° is attached to e_1 and to y (and not elsewhere).
(M4′) For $i > 1$ and $i + 2 \le j < m$, B_i° and B_j° can be simultaneously embedded in F_α if either $i \not\equiv \alpha \pmod 2$, or $j \equiv \alpha \pmod 2$ (or both). They can be simultaneously embedded in $F_{3-\alpha}$ if either $i \equiv \alpha \pmod 2$, or $j \not\equiv \alpha \pmod 2$ (or both). For $3 \le j < m$, $B_1^\circ \cup B_j^\circ$ can be embedded in F_α. For $1 < i \le m-2$, $B_i^\circ \cup B_m^\circ$ can be embedded in F_β. Additionally, $B_1^\circ \cup B_2^\circ \cup B_3^\circ \cup B_{m-2}^\circ \cup B_{m-1}^\circ \cup B_m^\circ$ can be embedded in $F_1 \cup F_2$.

Equivalent definition of a skew millipede is that (M2′) together with the last condition in (M4′) holds and after contracting the (closed) segment on e_2 between x and y, we get a thin millipede.

In referring to a *millipede*, we mean either a thin or a skew millipede. It is clear from the description that millipedes are obstructions for embedding extensions. It follows from (M4) ((M4′), respectively) that they are also minimal (no bridge is redundant).

An obstruction will be called *nice* if it is either composed of a small number of bridges (at most 13), or it is a millipede. Millipedes based on e_2 and with apex

$x \in V(e_1)$ ($\{x, y\} \subseteq V(e_1)$, respectively) are defined analogously. If the numbering of bridges in a millipede is reversed (i.e., $B'_i = B^\circ_{m-i+1}$), then $B'_1, \ldots, B'_m$ also satisfy (M1)–(M4) (or (M1)–(M4′)).

4. 2-Möbius band algorithm. Let G be a connected graph and $K = C \cup e_1 \cup e_2$ a subgraph of G homeomorphic to K_4, where C is a cycle and e_1, e_2 are disjoint paths connecting interlacing pairs a_0, b_0 and c_0, d_0 (respectively) of vertices in C. Suppose that K is embedded in the Möbius band with C on its boundary, and that F_1 and F_2 are the faces of this embedding (cf. Figure 1). The problem of extending the embedding of K to an embedding of G will be referred to as the *2-Möbius band embedding extension problem* [M1].

In this section we will outline a linear time algorithm for the 2-Möbius band embedding extension problem which finds an embedding extension whenever possible. We will show in Section 5 how to extend this algorithm in order to construct a nice obstruction in case when embedding extensions do not exist.

Next result will enable us to replace every K-bridge B in G by a small subgraph $\tilde{B} \subseteq B$ such that the embedding extension problem for the new graph is equivalent to the original one.

If B is a bridge of K in G, denote by $b(B)$ the number of branches of $B \cup K$ that are contained in B. The number $b(B)$ is called the *size* of B.

LEMMA 4.1. [M3] *Let G, K be as above. Every K-bridge B in G contains a subgraph $\tilde{B}$ with size at most 13 such that for an arbitrary set of non-local K-bridges $B_1, ..., B_k$, any embedding of $K \cup \tilde{B}_1 \cup ... \cup \tilde{B}_k$ in the Möbius band with C on the boundary can be extended to an embedding of $K \cup B_1 \cup \cdots \cup B_k$. Moreover, the replacement of all K-bridges B by their subgraphs $\tilde{B}$ can be done in linear time.*

Let $\mathcal{B}$ be the set of K-bridges in G. We assume that no bridge in $\mathcal{B}$ is local on e_1 or on e_2. Denote by $\mathcal{B}_0$ the subset of $\mathcal{B}$ containing exactly those bridges which have no vertex of attachment in $C - e_1 - e_2$. These bridges are candidates to be embedded either in F_1 or in F_2. From now on we will also assume that the replacement of all K-bridges B by their small subgraphs $\tilde{B}$ (Lemma 4.1) has already been made. Moreover, we assume that every bridge can be embedded in at least one of the faces F_1, F_2. Otherwise we get a small obstruction and stop immediately. In particular, if some bridge is attached only to two vertices of K, the above replacement changes it into a branch. Moreover, we will assume that multiple branches between the same vertices of K have been replaced by a single one.

Suppose that $B \in \mathcal{B}_0$. For $\mathrm{y} \in \{a, b\}$, let y_B be the vertex of attachment of B on e_1 as close to y_0 as possible. Define similarly c_B and d_B as "extreme" attachments of B on e_2. Since there are no local bridges, the quantities x_B ($\mathrm{x} \in \{a, b, c, d\}$) are well defined for every $B \in \mathcal{B}_0$. We define $\overline{a} = d$, $\overline{b} = c$, $\overline{c} = b$, $\overline{d} = a$ and $\tilde{a} = c$, $\tilde{b} = d$, $\tilde{c} = a$, and $\tilde{d} = b$. Note that $\overline{\mathrm{x}}_0$ and x_0 are in the same side (left, or right) of F_1 and that $\tilde{x}_0$ and x_0 lie in the opposite corners of F_1.

We will first construct four lists of bridges in $\mathcal{B}_0$. They will be denoted by S_x, where x stands for either of a, b, c, or d. The list S_x corresponds to the (oriented) branch e_1, or e_2 of K containing the vertex x_0 oriented from x_0 towards the other endpoint (e.g., S_c corresponds to e_2 oriented from c_0 towards d_0). Every list S_x will link all bridges from $\mathcal{B}_0$. Their order in S_x will be consistent with the following requirements:

(S1) If x_Q is closer to x_0 than x_R, then the bridge Q precedes R in S_x.
(S2) If $\mathrm{x}_Q = \mathrm{x}_R$ and $\overline{\mathrm{x}}_Q$ is closer to $\overline{\mathrm{x}}_0$ than $\overline{\mathrm{x}}_R$, then Q precedes R in S_x.

(S3) If $\mathrm{x}_Q = \mathrm{x}_R$, $\overline{\mathrm{x}}_Q = \overline{\mathrm{x}}_R$, and Q is attached only to x_Q and $\overline{\mathrm{x}}_Q$ and R has at least 3 vertices of attachment, then Q precedes R in the list S_{x}.

If a pair of bridges from $\mathcal{B}_0$ does not fit any of (S1), (S2), or (S3), then their order in S_{x} is irrelevant. If a set of bridges from $\mathcal{B}_0$ is embedded in F_1, then their order in F_1 from left to right is consistent with S_a and S_d and inverse to their order in S_b or S_c.

Suppose that e_j is the branch containing x_0. Let $v_1, v_2, \ldots, v_k$ be the vertices of e_j in direction from x_0 towards the other end. The list S_{x} is the concatenation of lists S_{x}^s, $s = 1, \ldots, k$, where each S_{x}^s links all bridges $B \in \mathcal{B}_0$ with $\mathrm{x}_B = v_s$ (in order respecting (S2) and (S3)). The lists S_{x}^s are constructed simultaneously as follows:

```
S_x^s := ∅, s = 1,...,k
Label all bridges in B_0.
for all u ∈ V(e_{3-j}) do
  { The vertices u are taken in order as they appear on
    e_{3-j} from x̄_0 towards the other end. }
  for all edges f incident with u do
    if f is a foot of a labeled bridge B then
      if B is attached only to two vertices then
        add B at the end of S_x^s, where s is such that v_s = x_B
        unlabel B
      endif
  endfor
  for all edges f incident with u do
    if f is a foot of a labeled bridge B then
      if B is attached to three or more vertices then
        add B at the end of S_x^s, where s is such that v_s = x_B
        unlabel B
      endif
  endfor
endfor
Link S_x^1,...,S_x^k into S_x.
```

It is easy to realize the traversals in the above algorithm so that the overall time spent by the algorithm is linear. Note that the double traversal of bridges with $\overline{\mathrm{x}}_B = u$ assures that condition (S3) will be fulfilled. Condition (S2) is satisfied at the end since the traversal of the "opposite" branch e_{3-j} is performed in the direction from $\overline{\mathrm{x}}_0$ towards the other end. Clearly, (S1) is guaranteed by the use of sublists S_{x}^s and their appropriate linking at the end.

We are now ready to discuss the main part of the algorithm. Roughly speaking, it is based on the following idea. Suppose that a subset of bridges $\mathcal{B}' \subseteq \mathcal{B}$ is *already embedded* in $F_1 \cup F_2$. Their presence in $F_1 \cup F_2$ blocks some embeddings of the remaining bridges. Some of the bridges thus need to be embedded in F_1, some others can only be embedded in F_2. We say that these bridges are *forced* in F_1 (or F_2, respectively). By adding these blocked bridges to $\mathcal{B}'$, we obtain additional bridges with only one face left for their embeddings. By repeating this procedure, we either get stuck (which proves that no embedding extension exists with the initial $\mathcal{B}'$ embedded as given), or no more bridges are blocked by the chosen embedding of $\mathcal{B}'$. In the latter case, it is clear that the bridges in $\mathcal{B}'$ can be left embedded as they are without obstructing any possible embeddings of the remaining bridges. The procedure described above is called FORCING.

At the very beginning, the bridges from $\mathcal{B} \setminus \mathcal{B}_0$ are uniquely embeddable and they are used as the starting set $\mathcal{B}'$. If FORCING does not embed all of the bridges (and does not get stuck), then the problem is how to re-start. (This cannot be avoided if, for example, $\mathcal{B}_0 = \mathcal{B}$.) This problem will be solved by using parallel computations. We choose a bridge B and start two parallel processes: the first one corresponds to embedding B in F_1, the other process to the case when we embed B in F_2. The details how to perform such parallel computations without increasing the overall time complexity are described in Section 2. Each of the two parallel processes either finds an embedding for a set of bridges which does not interfere with any embedding of the remaining bridges (*successful* termination), or it gets stuck (*non-successful* termination). It has been described in Section 2 how the two parallel processes react if one or the other stops successfully. To ensure linear time complexity we have to choose the starting bridge B appropriately: it must be the initial bridge in one of the lists S_{x}. Of course, these lists are updated during the algorithm by removing the already embedded bridges.

For x being any of a, b, c, or d, we will use three vertices, x, x_1, x_2 on the branch e_j ($j \in \{1,2\}$) containing the vertex x_0. For $u, v \in V(e_j)$, denote by $[u,v)$ the segment of e_j from u to v (including u but not including v), and similarly by $[u,v]$ the closed segment of e_j from u to v (including both u and v). During the algorithm, all bridges attached to $[\mathrm{x}_0, \mathrm{x})$ are already embedded and all remaining bridges attached to $[\mathrm{x}, \mathrm{x}_i)$ ($i = 1, 2$) are blocked in F_{3-i} by already embedded bridges, so they will need to be put in F_i. (In particular, if a bridge that has not yet been embedded is attached to $[\mathrm{x}, \mathrm{x}_1) \cap [\mathrm{x}, \mathrm{x}_2)$, then we are in trouble.) In the algorithm we also use bridges $B_{\mathrm{x},1}$, $B_{\mathrm{x},2}$. They are needed only for efficient construction of obstructions and their use is described in more details in Section 5.

The main part of the algorithm is the following:

```
Determine lists S_x, x ∈ {a,b,c,d}, as explained above.
Determine B_0. Let B' := B \ B_0.
Embed B'.
if no embedding exists then OBSTRUCTION
Determine initial values of x, x_1, x_2, B_{x,1}, B_{x,2} for x ∈ {a,b,c,d}.
FORCING
if not successful then OBSTRUCTION
Initialize auxiliary variables for parallel computations.
while B_0 ≠ ∅ do
  B := the first bridge in S_a
  for every embedding of B in F_1 ∪ F_2 do in parallel
    Determine initial values of x, x_1, x_2, B_{x,1}, B_{x,2} for x ∈ {a,b,c,d}.
    FORCING
  end parallel for
  if not successful then OBSTRUCTION
endwhile
{ If we reach this point, all the bridges have been embedded. }
Return the obtained embedding extension.
```

Procedure OBSTRUCTION reports that no embedding extension exists, and terminates. We will show in Section 5 that by extending this procedure, one can also construct nice obstructions (cf. Section 3) for embedding extensions in linear time. Procedure FORCING is described below:

```
procedure Forcing
{ Some bridges are already embedded. They block some embeddings of
  the remaining bridges. A bridge B ∈ 𝓑0 is blocked exactly when it is
  attached to a segment [x, xi) for some x ∈ {a, b, c, d}, i ∈ {1, 2}.
  In that case, it must be embedded in Fi. }
  while ∃x ∈ {a, b, c, d} such that x ≠ x1 or x ≠ x2 do
    if x ≠ x1 and x ≠ x2 then
      y := min{x1, x2} (closer to x)
      if ∃B ∈ 𝓑0 attached to [x, y) then Stop(not successful)
      x := y
    endif
    if x ≠ x1 then i := 1 else i := 2 endif
    𝓑i := all bridges in 𝓑0 attached to [x, xi)
    Embed 𝓑i in Fi.
    if no embedding exists then Stop(not successful)
    x := xi
    Update a3−i, b3−i, c3−i, d3−i, and Sx.
    𝓑0 := 𝓑0 \ 𝓑i
    Bx,i := extreme bridge in 𝓑i
    Let Bx,i point to Bx,3−i.
    { This will be needed in the construction of obstructions. }
  endwhile
  Return(successful)
end {Forcing}
```

The search for $B \in \mathcal{B}_0$ that is attached to $[\mathrm{x}, y)$ in the above procedure can be easily implemented by advancing through the list S_{x}. Similarly, the embeddability of $\mathcal{B}_i$ in F_i is checked by moving along the list S_{x} and comparing extreme vertices of attachment of bridges with already blocked segments on e_1 and e_2. More precisely, this is achieved as follows. Let $B_1, \ldots, B_t$ be the bridges in $\mathcal{B}_i$ listed in the order as they appear in S_{x}. Suppose that $\mathrm{x}_0 \in V(e_j)$ and denote by y_0 the other endpoint of e_j. Obviously, each bridge B_k $(1 \le k \le t)$ must have an embedding in F_i. Suppose first that $i = 1$. Each B_k must also be attached to e_j (entirely on the segment $[\mathrm{x}, \mathrm{y}_{3-i}]$) and to e_{3-j} (entirely to the segment $[\overline{\mathrm{x}}_{3-i}, \overline{\mathrm{y}}_{3-i}]$; otherwise it overlaps with the already embedded bridges). Moreover, for $k = 1, \ldots, t-1$ the bridge B_{k+1} must be entirely attached to the segment $[\mathrm{y}_{B_k}, \mathrm{y}_0]$ of e_j and to the segment $[\tilde{\mathrm{x}}_{B_k}, \tilde{\mathrm{x}}_0]$ of e_{3-j}; otherwise it overlaps with B_k in F_i. If none of these tests fails, then the bridges in $\mathcal{B}_i$ can be simultaneously embedded in F_i. When $i = 2$, some details in the above tests have to be modified appropriately since the list S_{x} is constructed with respect to embeddings in the face F_1. In particular, $\overline{\mathrm{x}}$ and $\overline{\mathrm{y}}$ have to be replaced by $\tilde{\mathrm{x}}$ and $\tilde{\mathrm{y}}$, respectively, and vice-versa. Moreover, bridges $B \in \mathcal{B}_i$ with the same extreme attachment x_B have to be considered in the order that is opposite to their order in S_{x}. During the above test we also change $\mathcal{B}_0$.

Initial values of a, a_1, a_2 (and similarly for other $\mathrm{x}, \mathrm{x}_1, \mathrm{x}_2$, $\mathrm{x} \in \{b, c, d\}$) are determined at the very beginning as follows. We take $a = a_0$. The vertex a_1 is equal to the vertex of attachment on e_1 closest to b_0 of bridges in $\mathcal{B} \setminus \mathcal{B}_0$ that are attached to the open segment from c_0 to a_0 on C. The corresponding bridge is taken as $B_{a,2}$. If there are no such bridges, then $a_1 = a_0$ (and $B_{a,2}$ is undefined). Similarly, a_2 is the attachment on e_1 closest to b_0 of bridges in $\mathcal{B} \setminus \mathcal{B}_0$ attached to the open segment on C from a_0 to d_0. The corresponding bridge is then $B_{a,1}$.

There is a slight difference in determining the initial values of $\mathrm{x}, \mathrm{x}_1, \mathrm{x}_2$ in the parallel part. The values x remain unchanged. If B (the initial bridge in S_a) is embedded in F_1, then: $\mathrm{x}_1 = \mathrm{x}$ for $\mathrm{x} \in \{a, b, c, d\}$, $a_2 = b_B$, $b_2 = b$, $c_2 = c$, and $d_2 = c_B$. We take $B_{a,1} = B_{d,1} = B$. Other $B_{\mathrm{x},j}$ are undefined.

If B is embedded in F_2, the situation is more complex. In this case the process of determining the initial values of $\mathrm{x}, \mathrm{x}_1, \mathrm{x}_2, B_{\mathrm{x},1}, B_{\mathrm{x},2}$ ($\mathrm{x} \in \{a, b, c, d\}$) will require some additional preprocessing in order to decide between two possible choices:

(a) If all bridges attached to e_1 only at a can be simultaneously embedded in F_2 (together with B), then they can go in F_2 without loss of generality. All other bridges attached to e_2 on $(d_B, c]$ must be in F_1, and after fixing these embeddings, we change c to become the vertex d_B and proceed in the same way as in the above case when B was in F_1. (We set $a_1 = b_B$, $c = c_1 = d_B$, $c_2 = d_R$, $b_2 = a_R$, where R is the "leftmost" bridge among those which were embedded in F_1; if R is not attached to the segment $[d, d_B)$, we take $c_2 = c$; if there are no such bridges, then $c_2 = c$, $b_2 = b$. Also, $B_{a,2} = B$, $B_{b,1} = B_{c,1} = R$, or undefined, other $B_{\mathrm{x},j}$ are always undefined.) If the above bridges cannot be simultaneously embedded in F_1, we terminate non-successfully.

(b) Two bridges B', B'' attached to e_1 only at a overlap in F_2 or such a bridge B' overlaps with B in F_2. Hence, one of B', B'' should be embedded in F_1. Then all bridges attached to e_2 on $[d, d_B)$ must be in F_2. Similarly, all bridges attached to e_2 only at d_B and attached to $(a, b]$ on e_1 will necessarily go into F_2. After fixing these embeddings we let $d = d_B$ and change other values $\mathrm{x}, \mathrm{x}_1, \mathrm{x}_2$ ($\mathrm{x} \in \{a, b, c, d\}$) as described below.

We need to make the decision about (a) or (b) in such a way that the time spent on this is proportional to the number of bridges whose embedding is determined during this process. (Otherwise, we can lose linearity.) This is achieved by traversing the list S_c. Let B' be the current bridge in the traversal. If $b_{B'} \neq a$, then we must embed B' in F_1; if it overlaps with already embedded bridges, call Obstruction. If not, embed B' in F_1 and proceed with the next bridge in the list S_c. Otherwise ($b_{B'} = a_{B'} = a$) we try to embed B' in F_2. If successful, we proceed with the next bridge in the list. If B' overlaps in F_2 with some already embedded bridge, we have (b). If B' overlaps in F_2 with an already embedded bridge $B'' \neq B$, then B'' is unique. Therefore, all other bridges that have been embedded during our traversal, can retain their embeddings without loss of generality. The same is true in the other case when B' overlaps with B. In the first case we set $R = B''$ while in the latter case we take $R = B'$. In both cases we will consider R as a non-embedded bridge in the sequel. Let Q be the last bridge embedded in F_1 during the traversal of S_c which has an attachment on $[c, c_R]$ (possibly undefined). Next we embed in F_2 all bridges attached to e_2 on $[d, d_B)$ and all bridges attached to d_B and to $(a, b]$. (If this is not possible, call Obstruction.) After these changes, the values $\mathrm{x}, \mathrm{x}_1, \mathrm{x}_2$ are determined as follows: a, b remain unchanged, $a_1 = a_2 = a$, $b_1 = a$, $b_2 = a_Q$ (or b if Q is undefined or attached to e_2 only at $(c_R, c]$), $c = c_1 = c_R$, $c_2 = d_Q$ (or c if Q is undefined), $d = d_B$, $d_1 = c_B$, $d_2 = d_B$. Bridges $B_{\mathrm{x},i}$ are defined accordingly.

If none of the above stop cases occurs, we stop when reaching d_B and then we have case (a).

5. 2-Möbius band obstructions. Our algorithm can be extended in a relatively simple way so that in case when no embedding extension exists, it returns a nice obstruction. Procedure Obstruction takes care of this task if we modify it as explained in the sequel.

There are three places where the presence of an obstruction is discovered:

(i) when embedding bridges of $\mathcal{B}'$,

(ii) in procedure FORCING,

(iii) when determining the initial values in the parallel part.

In case (i), we either get a K-bridge $B \in \mathcal{B}'$ that cannot be embedded in any of the faces, or we get two bridges $B_1, B_2 \in \mathcal{B}'$ that are both embeddable only in F_i ($i \in \{1,2\}$) where they overlap. It is clear that this case leads to a small obstruction which can be determined efficiently by applying the results of [M2].

Consider now (ii). In FORCING, there are two obstruction stops. The first possibility is when a bridge $B \in \mathcal{B}$ is attached to $[\mathrm{x}, y)$. This means that possible embedding of B in F_1 is blocked by $B_{\mathrm{x},1}$ and its embedding in F_2 is blocked by $B_{\mathrm{x},2}$. When $B_{\mathrm{x},1}$ was embedded, we remembered which bridge forced it to be in F_1. Similarly for all other embedded bridges. Thus we can reconstruct a chain

$$(B_1, F_{i_1}) \to (B_2, F_{i_2}) \to \ldots \to (B_p, F_{i_p})\,, \tag{1}$$

where the notation $(Q, F) \to (R, F')$ means that Q and R cannot be simultaneously embedded in F (Q being embedded in F forces R being embedded in F'), and where B_1 is one of the initial bridges with fixed embedding, and $(B_p, F_{i_p}) = (B, F_1)$. Let us note that $i_1, \ldots, i_p \in \{1,2\}$ and that any two consecutive i_r, i_{r+1} are distinct. Also, $B_{p-1} = B_{\mathrm{x},1}$. Similarly, we have a chain forcing B to be in F_2:

$$(B'_1, F_{j_1}) \to (B'_2, F_{j_2}) \to \ldots \to (B'_q, F_{j_q})\,, \tag{2}$$

where $(B'_q, F_{j_q}) = (B, F_2)$. It is clear that $(Q, F_i) \to (R, F_{3-i})$ is equivalent to $(R, F_i) \to (Q, F_{3-i})$. Therefore (2) is equivalent to

$$(B'_q, F_{3-j_q}) \to (B'_{q-1}, F_{3-j_{q-1}}) \to \ldots \to (B'_1, F_{3-j_1})\,. \tag{3}$$

Note that $(B'_q, F_{3-j_q}) = (B_p, F_{i_p}) = (B, F_1)$. Now, (1) and (3) can be concatenated and rewritten in the form:

$$(R_1, F_{s_1}) \to (R_2, F_{s_2}) \to \ldots \to (R_r, F_{s_r})\,, \tag{4}$$

where $(R_1, F_{s_1}) = (B_1, F_{i_1})$ and $(R_r, F_{s_r}) = (B'_1, F_{3-j_1})$.

The second stop in FORCING occurs when $\mathcal{B}_i$ cannot be simultaneously embedded in F_i. If $B \in \mathcal{B}_i$ overlaps in F_i with some of the already embedded bridges, we have exactly the same situation as above — we get (4). (As explained, this can be discovered by a simple comparison of the extreme attachments of bridges in $\mathcal{B}_i$ with $a_{3-i}, b_{3-i}, c_{3-i}, d_{3-i}$.)

Next possibility is that a bridge $B \in \mathcal{B}_i$ cannot be embedded in F_i (i.e., its only embedding is in F_{3-i}). Then we have

$$(R_1, F_{s_1}) \to (R_2, F_{s_2}) \to \ldots \to (R_r, F_{s_r})\,, \tag{5}$$

where $(R_r, F_{s_r}) = (B, F_i)$. This chain is not only of the same form as (4) but also obeys the same condition that will be used in producing nice obstructions: R_1 is embeddable only in F_{s_1} and R_r is embeddable only in F_{3-s_r}, the opposite face of F_{s_r}.

Similarly, if two bridges from $\mathcal{B}_i$ overlap in F_i. We easily get a chain of form (4) having the same properties as in the other cases.

If procedure OBSTRUCTION is reached because of unsuccessful termination of the parallel computation, we get two chains of the form (4), one from each parallel process. The first one starts with (B, F_1) and it is discovered in (ii). It satisfies the chain condition (the first bridge uniquely embeddable, the last one assigned to the wrong face) under assumption that B is embeddable only in F_1. The second process gives rise to a similar chain. However, in this case the situation is slightly different. We either get a chain of the form (4) that is obtained in (ii) and starts with (B, F_2), or we get a small obstruction from (iii) which itself gives rise to a chain of the form (4). More precisely, there are two possible calls to OBSTRUCTION in (iii). If there are two bridges R', R'' that overlap in F_2 and are forced in F_2 by B, then

$$(B, F_2) \to (R', F_1) \to (R'', F_2) \to (B, F_1)$$

is the required chain of the form (4). The second possibility is when the set of bridges

$$\mathcal{B}'' = \{R \in \mathcal{B} \mid d_R \in [d, d_B) \text{ or } (d_R = d_B \text{ and } b_R \in (a, b])\}$$

cannot be simultaneously embedded in F_2. In this case, there is also a pair B', $\tilde{B}$ of bridges (where $\tilde{B} = B''$ or B) attached to e_1 only at a and attached to e_2 entirely on $[c, d_B]$ that overlap in (F_1 and) F_2. Suppose first that there are two bridges $R', R'' \in \mathcal{B}''$ that overlap in F_2. Then the bridges B', $\tilde{B}$, R', R'' form a small obstruction for the whole embedding extension problem. The remaining possibility why the bridges from $\mathcal{B}''$ cannot be simultaneously embedded in F_1 is that there is a uniquely embeddable bridge $R' \in \mathcal{B}''$ that has no embedding in F_2. Then B', $\tilde{B}$, and R' form a small obstruction and we are done.

If the chain of the first parallel process starts with (B, F_1) and the chain of the other process starts with (B, F_2), we can concatenate one chain with "inverse" of the other to get a chain of the form:

$$(R_1, F_{s_1}) \to (R_2, F_{s_2}) \to \ldots \to (R_r, F_{s_r})\,. \tag{6}$$

In general, there are three possibilities why the chain of form (6) (or (4)) leads to an obstruction:

(A) As described before: R_1 is embeddable only in F_{s_1} and R_r is embeddable only in F_{3-s_r}. We allow that $R_1 = R_r$.
(B) $(R_1, F_{s_1}) = (R_r, F_{s_r}) = (B, F_1)$ and (B, F_2) appears somewhere in the chain.
(C) R_1 is embeddable only in F_{s_1} and $(B, F_1), (B, F_2)$ both appear somewhere in the chain.

The last case (C) can be transformed into a chain of type (A) as follows. If $(B, F_1) = (R_i, F_{s_i})$, $(B, F_2) = (R_j, F_{s_j})$, $i < j$, then we get:

$$(R_1, F_{s_1}) \to \ldots \to (R_j, F_{s_j}) \to (R_{i-1}, F_{3-s_{i-1}}) \to \ldots \to (R_1, F_{3-s_1})\,.$$

We will show that the obstruction formed by the chain (6) (viewing (4) as case (A) of (6)) can be efficiently transformed either into a small obstruction or into a (thin or skew) millipede. This will be achieved through a series of successive reductions of the chain (6). We will assume that $r \geq 14$. Otherwise we have a small obstruction formed by at most 13 bridges from our chain. If during the following reductions the length of the chain drops below 14, we automatically stop because we have obtained a small obstruction.

We say that bridges R and R' are *parallel* in F_i ($i \in \{1, 2\}$) if they cannot be simultaneously embedded in F_{3-i}, i.e. $(R, F_{3-i}) \to (R', F_i)$.

LEMMA 5.1. *Let bridges R_i and R_{i+2} from (6) be parallel in F_{s_i}. Then in every embedding of $R_i \cup R_{i+1} \cup R_{i+2}$, the bridge R_{i+2} is embedded in F_{s_i}.*

Proof. Assume that there is an embedding of $R_i \cup R_{i+1} \cup R_{i+2}$ such that R_{i+2} is embedded in F_{3-s_i}. Since R_i is parallel with R_{i+2} in F_{s_i}, it is embedded in F_{s_i}. By (6), R_{i+1} is embedded in F_{3-s_i} and R_{i+2} should be embedded in F_{s_i}, a contradiction. □

Similar arguments also show that if R_i and R_{i+2j} are parallel in F_{s_i}, then in every embedding of $R_i \cup \cdots \cup R_{i+2j}$, the bridge R_{i+2j} is embedded in F_{s_i}. In such a case the bridge R_{i+2j} can be regarded as uniquely embeddable under the condition that the final obstruction contains also the bridges $R_i, \ldots, R_{i+2j-1}$. In the sequel, we will need the above claim for $j = 1$ and $j = 2$.

If there is a pair (R_i, F_{s_i}) $(1 < i < r)$ in the chain (6) such that R_i can be embedded only in one face, then we act as follows. We may assume that R_i can be embedded in F_{s_i}, since otherwise we could look at the reversed chain

$$(R_r, F_{3-s_r}) \to \ldots \to (R_2, F_{3-s_2}) \to (R_1, F_{3-s_1}), \qquad (6')$$

where R_i appears in the right face. If the chain is of type (A), then we can shorten the obstruction by taking $(R_i, F_{s_i}) \to \ldots \to (R_r, F_{s_r})$. If chain is of type (B), then we can change it into type (C). Let j $(1 < j < r)$ be an index such that $(R_j, F_{s_j}) = (R_r, F_{3-s_r})$. We take the chain $(R_i, F_{s_i}) \to \ldots \to (R_r, F_{s_r})$, if $i < j$. Similarly if $i > j$ when we take $(R_i, F_{s_i}) \to \ldots \to (R_r, F_{s_r}) \to (R_2, F_{s_2}) \to \ldots \to (R_j, F_{s_j})$. The obtained chain can be further reduced to type (A) as shown previously. It is easy to see how to implement the above tests and reductions in linear time.

From now on we will assume that every bridge participating in the chain (6), except the first and the last one when we have a chain of type (A), has (allowed) embeddings in faces F_1 and F_2. If there is a pair (R, F) which appears twice in the chain of type (A) we leave out pairs between the two appearances. In chains of type (B) this is performed only when the two appearances lie in the same segment of the chain between R_1 and its appearance in the other face. Again, this task can be easily performed in linear time.

Suppose that we have a chain of type (B). Then we perform another checking which will be needed in the proof of Lemma 5.2. Let (R_j, F_{s_j}) be the occurrence of R_1 in the other face. If $(R_{j-3}, F_{s_{j-3}}) \to (R_j, F_{s_j})$ or $(R_j, F_{s_j}) \to (R_{j+3}, F_{s_{j+3}})$, then we can shorten our chain by leaving out the two superfluous pairs. We repeat this change as long as possible. Under every embedding of $R_1 \cup \ldots \cup R_{j-1}$ in $F_1 \cup F_2$, the bridge $R_1 = R_j$ is embedded in F_{s_j}. Therefore we may assume that $j \geq 6$ since otherwise we can transform our chain of type (B) into a chain of type (A) (with at most four additional bridges which guarantee unique embeddability of R_1 in F_{s_j}). In this case we also repeat previous reductions on the new chain. Similarly, we may assume that $j \leq r - 5$. Note that all these changes can be done in linear time.

Next we check if there are pairs of parallel bridges which appear not far apart in the chain. Suppose that we have a chain of type (B) with bridges R_i and R_{i+2} being parallel in F_k. By reversing the chain, if necessary, we may assume that $F_k = F_{s_i}$. There exists an index j, $1 < j < r$, such that $(R_j, F_{s_j}) = (R_1, F_{3-s_1})$. We will regard R_{i+2} as uniquely embeddable in F_{s_i} (Lemma 5.1). We will actually achieve this property at the end by adding bridges R_i and R_{i+1} into the final obstruction. If $i+2 \leq j$, our chain can be shortened and transformed into type (C) by taking $(R_{i+2}, F_{s_{i+2}}) \to \ldots \to (R_r, F_{s_r})$. If $i + 2 > j$, we transform our chain into $(R_{i+2}, F_{s_{i+2}}) \to \ldots \to (R_r, F_{s_r}) = (R_1, F_{s_1}) \to \ldots \to (R_j, F_{s_j})$ which can be viewed as a chain of type

(C). In both cases, our chain of type (C) can be further changed into type (A). If we have a pair of parallel bridges R_i, R_{i+4}, we take the same steps, except that in this case the final obstruction will have to contain not only the bridges R_i, R_{i+1} but also bridges R_{i+2}, R_{i+3}. Obtaining the chain of type (A) we again perform the above reductions (no intermediate bridge uniquely embeddable, no repetitions). Note that this additional work can occur only once — when changing type (B) into type (A).

Let us now explain how to react regarding parallel bridges if we have a chain of type (A). For $j = r, r-1, \ldots, 3$ we check whether R_j is parallel with R_{j-2} and whether R_{j+2} is parallel with R_{j-2} (when $j \leq r-2$). If R_j and R_{j-2} are parallel in F_{s_j}, we shorten the chain by removing the initial part $(R_1, F_{s_1}) \to \ldots \to (R_{j-1}, F_{s_{j-1}})$ and stop. If they are parallel in F_{3-s_j}, then we remove the tail $(R_{j+1}, F_{s_{j+1}}) \to \ldots \to (R_r, F_{s_r})$ and continue with work. Similarly when R_{j-2} and R_{j+2} are parallel.

Let us remark that if R_1 is not embeddable in F_{3-s_1}, then R_1 and R_3 are parallel in $F_{s_1} = F_{s_3}$. Similarly, R_{r-2} and R_r are usually parallel in F_{3-s_r}. It is obvious how to perform the above tasks in linear time. By Lemma 5.1 the chain obtained after this reduction (together with at most $4 + 4 = 8$ additional bridges which guarantee the unique embeddability of the first and the last bridge in the chain of type (A)) still determines an obstruction. By the above remark, each bridge R_i $(1 \leq i \leq r)$ can be embedded in F_1 and in F_2 and no two bridges R_i, R_{i+2} $(1 \leq i \leq r-2)$ or R_i, R_{i+4} $(1 \leq i \leq r-4)$ are parallel in any of the faces.

Let $\mathcal{R}_1 = \{R_{2i-1} \,|\, 1 \leq i \leq \lceil r/2 \rceil\}$ and $\mathcal{R}_2 = \{R_{2i} \,|\, 1 \leq i \leq \lfloor r/2 \rfloor\}$.

LEMMA 5.2. *There exists $j \in \{1, 2\}$ and a vertex $x \in V(e_j)$ such that every bridge from $\mathcal{R}_1$ is attached to e_j only at x. Similarly, there exists $k \in \{1, 2\}$ and a vertex $y \in V(e_k)$ such that every bridge from $\mathcal{R}_2$ is attached to e_k only at y.*

Proof. Since we have decided to stop whenever our obstructing family of bridges contains 13 or fewer members, we have $r \geq 6$. Consider the bridges $R_i, R_{i+2}, R_{i+4} \in \mathcal{R}_1$. Since they are pairwise non-parallel in F_1 and in F_2, they can be simultaneously embedded in any of the faces. Therefore their union cannot contain two disjoint paths connecting branches e_1 and e_2. Note that not all three bridges can be equal to each other. Hence there exists a vertex x in one of the branches, say e_j, such that x is the only vertex of attachment of $R_i \cup R_{i+2} \cup R_{i+4}$ to e_j. Moreover, $R_i \cup R_{i+2} \cup R_{i+4}$ is attached to at least two vertices on the branch e_{3-j}. Similarly, there is a vertex x' in the branch $e_{j'}$ such that $R_{i+2} \cup R_{i+4} \cup R_{i+6}$ is attached to $e_{j'}$ only at x'. If $R_{i+2} \neq R_{i+4}$, then it easily follows that $e_{j'} = e_j$ and $x' = x$. On the other hand, $R_{i+2} = R_{i+4}$ can only happen if our chain is of type (B) and $(R_{i+3}, F_{s_{i+3}}) = (R_1, F_{3-s_1})$. Since in this case $(R_{i+2}, F_{s_{i+2}}) \to (R_{i+3}, F_{s_{i+3}})$ and $(R_{i+3}, F_{s_{i+3}}) \to (R_{i+4}, F_{s_{i+4}}) = (R_{i+2}, F_{s_{i+2}})$, bridges R_{i+2} and R_{i+3} must overlap on e_1 or e_2. If they overlap on e_j, then $(R_i, F_{s_i}) \to (R_{i+3}, F_{s_{i+3}})$ which is not possible because of previous reductions. Therefore R_{i+2} and R_{i+3} overlap on e_{3-j}. Suppose that $x' \neq x$. Then also $e_{j'} = e_{3-j}$. Since R_{i+2} overlaps on $e_{3-j} = e_{j'}$ with R_{i+3} and since R_{i+6} is attached on $e_{j'}$ to the same vertex x' as R_{i+2}, we have $(R_{i+3}, F_{s_{i+3}}) \to (R_{i+6}, F_{s_{i+6}})$. But this is a contradiction, since we have reduced such forcing at previous steps. Consequently, $x' = x$. By increasing i, we easily derive the claimed result.

The proof of the second part is almost identical. □

Additionally, we claim that either x and y lie on the same branch, or there is a small obstruction. For vertices $u, v \in V(e_1)$ we say that u is to the *left* of v (or v is to the *right* of u) if u is closer to a_0 than v. Similarly if $u, v \in V(e_2)$ we say that u is to the *left* of v if it is closer to d_0. Suppose now that $r > 5$ and that $x \in V(e_1)$, $y \in V(e_2)$. We will distinguish between two possibilities:

(i) If there is a bridge $R_i \in \mathcal{R}_1$ which is attached on e_2 to the left and to the right of y, then a small obstruction is obtained as follows. When the chain is of type (A), pairs $(R_1, F_{s_1}) \to (R_2, F_{s_2}) \to (R_i, F_{s_i})$ together with (R_r, F_{s_r}) if r is even, or together with $(R_{r-1}, F_{s_{r-1}}) \to (R_r, F_{s_r})$ if r is odd, form the desired obstruction. If the chain is of type (B), then R_1 is just the branch xy since $R_1 \in \mathcal{R}_1 \cap \mathcal{R}_2$. Let $(R_j, F_{s_j}) = (R_1, F_{3-s_1})$ be the occurrence of R_1 in the other face. Since R_2 overlaps with R_1 in F_{s_1} and R_1, R_2 are attached to e_2 only at y, R_2 is attached to e_1 to the left and to the right of x. Similarly, R_{j+1} is attached to the left and to the right of y on e_2. Then $R_1 \cup R_2 \cup R_{j+1}$ is a small obstruction. The case when there is a bridge $R \in \mathcal{R}_2$ attached on e_1 to the left and to the right of x is similar.

(ii) There is no bridge attached on e_1 to the left and to the right of x and also, there is no bridge attached on e_2 to both sides of y. In this case the chain must be of type (A) since otherwise $R_1 \in \mathcal{R}_1 \cap \mathcal{R}_2$ would be just the branch xy and would not be obstructed by any of the bridges. It is easy to see that under every embedding of $R_1 \cup R_2 \cup R_3 \cup R_4 \cup R_r$, the bridge R_r is embedded in F_{s_r}. Since this is wrong face for R_r, we have an obstruction.

In both cases the obtained small obstruction (together with bridges which assure the unique embeddability of R_1 and R_r) contains at most 13 bridges.

So far we have been able to restrict attachments of the bridges from the chain at one of the branches to at most two vertices. It remains to find a millipede (or a small obstruction) composed of some of these bridges. First we examine the case when $x = y$. By a planarity testing we try to embed $\mathcal{R}_1 \cup \mathcal{R}_2$ in $F_1 \cup F_2$. (Planarity testing can be used because $\mathcal{R}_1 \cup \mathcal{R}_2$ is attached to one of e_1, e_2 just at a point.) If the test fails, there will be a small obstruction composed of three mutually overlapping bridges. Such bridges can be discovered in linear time by a traversal of the corresponding branch e_i ($i \in \{1, 2\}$) since bridges R_j, R_k overlap if and only if the interiors of their attachment intervals on e_i are not disjoint. This fact can also be used to prove that we always get exactly three such bridges. The other case is when $\mathcal{R}_1 \cup \mathcal{R}_2$ admits an embedding in $F_1 \cup F_2$. Then the chain is of type (A). In this case we must also consider the additional bridges that assure the unique embeddability of R_1 and R_r. They either give rise to a small obstruction (together with R_1, R_2, R_{r-1}, R_r), or we get a thin millipede after eliminating possible superfluous additional bridges (cf. Claims 2 and 3 below).

Suppose now that $x \neq y$. Then our chain is of type (A). Note that in this case $\mathcal{R}_1 \cap \mathcal{R}_2 = \emptyset$. Without loss of generality we may assume that $x, y \in V(e_1)$ so that x is to the left of y and that $F_1 = F_{s_1}$. The main idea of the algorithm is to traverse e_2 from left to right and at each step embed those bridges from $\mathcal{R}_1 \cup \mathcal{R}_2$ which are forced in one of the faces by previously embedded bridges.

Bridges forming a millipede will be denoted by $Q_1, Q_2, \ldots$. For $i = 1, 2, \ldots,$ we will denote by l_i and r_i the leftmost and the rightmost vertex of attachment of Q_i on e_2, respectively. Let $Q_1 = R_1$. Since Q_1 has to be embedded in F_1, every bridge from $\mathcal{R}_2$ with vertex of attachment (strictly) to the left of r_1 should go in F_2. Therefore we embed these bridges in F_2. (If they cannot be simultaneously embedded, then we get a small obstruction and stop.) Denote by Q_2 the rightmost (with respect to attachments on e_2) of these bridges. If r_2 lies to the left of r_1 (or $r_2 = r_1$), we can find a small obstruction (for details see case (iii) below). Hence, every bridge from $\mathcal{R}_1$ with vertex of attachment to the left of r_1 is forced in F_1 by Q_2. We may assume that all these bridges can be simultaneously embedded in F_1. Otherwise a small obstruction can be found. Continuing this process we obtain a sequence of bridges $Q_1, Q_2, Q_3, \ldots$ such that for every i, bridge Q_i overlaps on e_2 with Q_{i+1}. There are several possibilities when we terminate this construction. Throughout the discussion of each possibility we will assume that the last embedded bridge in the above sequence is Q_s and that it is embedded in F_1. Note that in this case $Q_1, Q_3, \ldots, Q_s \in \mathcal{R}_1$ and $Q_2, Q_4, \ldots, Q_{s-1} \in \mathcal{R}_2$. Let $\mathcal{B}$ be the set of bridges from $\mathcal{R}_2$ that have an attachment on $[r_{s-1}, r_s)$.

(i) When trying to simultaneously embed in F_2 all bridges from $\mathcal{B}$, we encounter a pair of overlapping bridges Q, Q'. Since Q_s, Q, Q' pairwise overlap on e_2, they form a small obstruction.

(ii) If $R_r \in \mathcal{B}$, then we set $Q_{s+1} = R_r$ and stop.

(iii) Embed in F_2 all bridges from $\mathcal{B}$ and let Q_{s+1} be the rightmost among these bridges. Assume that r_{s+1} is not strictly to the right of r_s. If among the remaining bridges there is no bridge attached to e_2 entirely on the segment $[r_s, c_0]$, then $R_r \in \mathcal{R}_1$. Moreover, since $(R_{r-1}, F_2) \to (R_r, F_1)$ and since R_{r-1} is already embedded, also (Q_{s+1}, F_2) forces (R_r, F_1). Hence Q_s, Q_{s+1} and R_r (together with additional bridges guaranteeing unique embeddability of R_r) form a small obstruction. Otherwise, let R_i be the first bridge from the chain that is attached to e_2 only at $[r_s, c_0]$. By minimality of i and since $(R_{i-1}, F_{s_{i-1}}) \to (R_i, F_{s_i})$, the bridge R_{i-1} must be attached to the left and to the right of r_s (and also $F_{s_{i-1}} = F_1$). Then $R_{i-2} \in \mathcal{R}_2$ must be attached on e_2 entirely to the left of r_s. Since Q_{s+1} is the rightmost among bridges embedded in F_2, Q_{s+1} and R_{i-1} overlap on e_2. Therefore, Q_s, Q_{s+1} and R_{i-1} form a small obstruction.

(iv) Now we have r_{s+1} strictly to the right of r_s. Next we check if there is a non-embedded bridge $Q \in \mathcal{R}_1$ attached to $[r_{s-1}, r_s)$. If it exists, then Q_{s-1}, Q_s, Q_{s+1} and Q form a small obstruction. Otherwise, every bridge attached to the left of r_s has been embedded, another member Q_{s+1} of a possible millipede has been obtained and we can proceed with the next iteration.

If in the above steps a small obstruction has not been encountered, then we have stopped in (ii) and the bridges $Q_1 = R_1, Q_2, \ldots, Q_s, Q_{s+1} = R_r$ taken as $B_2^\circ, \ldots, B_{m-1}^\circ$ ($m = s+3$), respectively, satisfy (M2′), (M3) and (M4′) from the definition of skew millipedes. We will obtain B_1° and B_m° from the additional bridges (which guarantee the unique embeddability of R_1 and R_r, respectively) and either prove that the obtained sequence $B_1^\circ, B_2^\circ, \ldots, B_m^\circ$ satisfies (M1)–(M4′), or obtain a small obstruction from these bridges.

Denote by Q_0 the additional bridges that guarantee the unique embeddability of Q_1. Define similarly Q_{s+2} (the corresponding bridges for Q_{s+1}). Recall that each of Q_0 and Q_{s+2} is composed of one up to at most four bridges. In the following paragraphs we are going to show how to change Q_0 and Q_{s+2} to get a skew millipede. In each claim we will either prove the desired property or a small obstruction will be found.

Claim 0. *$\tilde{Q} = Q_0 \cup Q_1 \cup Q_2 \cup Q_s \cup Q_{s+1} \cup Q_{s+2}$ has an embedding in $F_1 \cup F_2$.* If there is no such embedding, this is a small obstruction, and we are done. Note that every embedding of $\tilde{Q}$ has Q_1 in F_1, Q_2 in F_2. Similarly we know the faces where Q_s and Q_{s+1} are embedded.

Claim 1. *No bridge is attached to a vertex on $(x, y) \subset e_1$.* Suppose there is such a bridge B. If $\tilde{Q} \cup B$ is an obstruction, it contains at most 13 bridges, and we are done. Otherwise, B is attached only to $[x, y]$ and to $[r_1, l_{s+1}] \subseteq e_2$. Since B is not local, it has an attachment z on e_2. For some i, $2 \le i \le s$, $z \in (l_i, r_i)$. It is easy to see that $B \cup Q_{i-1} \cup Q_i \cup Q_{i+1}$ is an obstruction.

CLAIM 2. *Q_0 contains one bridge and l_1 is strictly to the left of l_2.* Consider an embedding of $Q_0 \cup Q_1$ induced by an embedding of $\tilde{Q}$. By definition of Q_0, Q_1 cannot be re-embedded in F_2 under this embedding. Since our embedding is induced by $\tilde{Q}$, there is a bridge $B \subseteq Q_0$ which is attached on (l_1, r_1). If there are more candidates, we take the leftmost one. If B is attached out of e_2 to a vertex different from y, then $B \cup Q_1 \cup Q_2$ has unique embedding in $F_1 \cup F_2$, and we can replace Q_0 by the single bridge B and still retaining the property (M1) (for $B_j^\circ = Q_{j-1}$, $j = 1, 2, 3$). It is also clear that in this case l_1 is to the left of l_2. The remaining case is when B is attached to e_1 only at y. In this case we extend the sequence $Q_1, \ldots, Q_{s+1}$ by adding B at its beginning and changing Q_0 into $Q_0 \setminus B$. Using similar arguments as above, one can prove that every embedding of the new Q_0 forces B to be embedded in F_2. Then we repeat the above reductions starting with CLAIM 0 (with the appropriate change of roles of x, y, F_1, F_2, etc.). Note that this extension occurs at most three times.

CLAIM 3. *Q_{s+2} contains one bridge and r_{s+1} is strictly to the right of r_s.* The proof of this claim is analogous to the proof of the previous claim.

Having all of the above properties, we define $m = s + 3$ and $B_j^\circ = Q_{j-1}$, $j = 1, \ldots, m$. Using above claims and properties of the sequence $Q_1, \ldots, Q_{s+1}$, we see that the bridges B_j° $(1 \le j \le m)$ satisfy conditions (M1)–(M4$'$) from the definition of skew millipedes.

To summarize, we have proved the following result.

THEOREM 5.3. *Let $K = C \cup e_1 \cup e_2$ be a subgraph of a graph G for 2-Möbius band embedding extension problem. Suppose that no K-bridge in G is local on one of the branches e_1, e_2. There is a linear time algorithm that either finds an embedding extension of K to G, or returns an obstruction Ω for embedding extendibility. In the latter case, Ω is either small and contains at most 13 bridges, or it is a millipede based on one of the branches e_1, e_2 and with apex on the other branch.*

Let us recall that large bridges in the original graph have been replaced by small bridges ($b(B) \le 13$). Moreover, when we have a millipede, all bridges except B_1° and B_m° can be replaced by triads ($b(B) = 3$).

REFERENCES

[CR] S. A. COOK AND R. A. RECKHOW, *Time bounded random access machines*, J. Comput. Syst. Sci. 7 (1976), 354–375.

[GT] J. L. GROSS AND T. W. TUCKER, *Topological Graph Theory*, Wiley-Interscience, New York, 1987.

[HT] J. E. HOPCROFT AND R. E. TARJAN, *Efficient planarity testing*, J. ACM 21 (1974), 549–568.

[JM] M. JUVAN AND B. MOHAR, *2-restricted extensions of partial embeddings of graphs*, submitted.

[JMM] M. JUVAN, J. MARINČEK, AND B. MOHAR, *A linear time algorithm for embedding graphs in the torus*, submitted.

[M1] B. MOHAR, *Projective plane and Möbius band obstructions*, submitted.

[M2] ———, *Obstructions for the disk and the cylinder embedding extension problems*, Comb. Probab. Comput. 3 (1994), 375–406.

[M3] ———, *Universal obstructions for embedding extension problems*, submitted.

[M4] ———, *A linear time algorithm for embedding graphs in an arbitrary surface*, submitted.