1 Introduction

Performing an efficient exploration in unknown environments is a fundamental problem in several areas of mathematics. We consider the problem of graph exploration, sometimes called graph traversal, in which a mobile agent has to visit all n nodes of an arbitrary graph G. The mobile agent does not know the structure of the entire graph, it must use some local information and a starting point in order to eventually visit every vertex of the graph. It was shown in [21] that a mobile agent performing a random walk of length $O(n^3)$ will visit every vertex with high probability. One important aspect of any deterministic solution is the amount of information available, which includes both how much memory the agent has, and what information is stored locally on the graph. If the agent has at most constant memory, it is often modelled as a finite state automata. The minimum local information that every graph has is that every vertex knows which edges it is incident to. In a graph with no additional local information available, limiting the agent to only constant memory is not a trivial restriction. A single mobile agent cannot explore all graphs, in fact even a finite group of finite automata can not cooperatively explore all cubic planar graphs [19]. This model was extended by Cook and Rackoff to a more powerful tool called the Jumping Automaton for Graphs, or JAG, where any automaton in the team can jump to the location of any other. However, even this model is not powerful enough to explore all graphs, see [9].

If we are allowed to store some extra information locally at each vertex of the graph, the results become more interesting. For example, geometric graphs allow each vertex a unique identifier, which includes its geometric position with respect to an embedding of the graph on a surface. It turns out that planar geometric graphs can be explored with only a single finite state automata with constant memory, see [5] and [11]. These results were extended to a larger class of graphs called quasi-planar graphs in [8].

Another method of storing local information is to use port numbers, which were first proposed in [14]. While leaving vertices indistinguishible, each vertex assigns an ordering to all the edges incident to it. This is sometimes called a local orientation of the edges, and is usually represented by integers between 1 and the degree of the vertex. It was shown in [14] that a single agent can use these port numbers to explore any graph in linear time, and furthermore this exploration will traverse at most $10n$ edges. Periodic graph exploration requires that an algorithm visits every vertex infinitely many times in a periodic manner, where the period is the maximum number of edge traversals performed between two consecutive visits of a generic vertex, denoted by $pi(n)$. The problem of periodic graph traversal is concerned with finding upper and lower bounds of $pi(n)$. In [18], this upper bound of $\pi \leq 10n$ was improved to $\pi(n) \leq 4n$, and then again in [17] to $\pi(n) < 3.75n - 2$. Here, we relax the definition of port numbers to allow integers so that they may be larger than the degree of the vertex, and obtain a stronger upper bound. This problem has the trivial lower bound of $pi(n) \geq 2n - 2$ as shown in [].

2 Definitions and Notation

A finite simple graph G is an ordered pair $G = (V, E)$, where V is a finite set of vertices, and E is a finite set of edges such that each edge is a two element subset of V. We refer to finite simple graphs as just graphs. By convention, we use n for the number of vertices, and m for the number of edges of the graph. For an edge $e = \{u, v\}$ we simply write $e = uv$, and say that e is incident to both u and v. The *degree* of a vertex v, denoted $d(v)$, is the number of edges incident to v. A *uv-walk* in a graph G is an ordered sequence of vertices $u = v_1, v_2, \ldots, v_{n-1}, v_k = v$, such that where $v_i v_{i+1}$ is an edge of G for all $1 \leq i \leq k - 1$. The vertices u and v are the *terminal* vertices of a uv-walk, and if they are understood, we simply refer to it as a walk. The *length* of a walk is the number of edges $v_i v_{i+1}$ in the sequence, and are referred to as edges of the walk. A *path* is a walk that does not allow the repetition of vertices. A *cycle* C of G is a uv-walk such that $u = v$. A cycle in G that contains every vertex of G exactly once is called a *Hamiltonian cycle*. A *Hamiltonian walk* is a cycle in G that contains every vertex of G at least once.

The *distance* between any two vertices u and v is the length of a shortest uv-path. The *square* of a graph G, denoted G^2, is constructed by starting from G by adding an edge between each pair of vertices at distance two. Clearly, G is a subgraph of G^2. A *mixed* graph G is an ordered triple $G = (V, E, A)$, such that (V, E) is a graph, and A is a finite set of arcs, where each arc is an ordered pair of vertices. A mixed graph is called simple if (V, E) is a simple graph, and for each arc (u, v), $uv \notin E$. The *underlying* graph is obtained by replacing every arc of a mixed graph by an edge with the same endpoints. A *directed* graph is a mixed graph where $E = \emptyset$. For the arc (u, v), we say that u is the tail (vertex) of the arc, and v is the head (vertex) of the arc. We say that (u, v) is an arc from u to v. A *directed uv-path* P is an ordered sequence of vertices $u = v_1, v_2, \ldots, v_{n-1}, v_n = v$, without repetition, such that where $(v_i v_{i+1})$ is an arc for all $1 \leq i \leq n - 1$. If uv is an edge of the graph $G = (V, E)$, we assign an orientation to uv by replacing G with $(V, E \backslash \{uv\}, \{(u, v)\})$, which is now a mixed graph. Similarly we can assign an orientation to any edge in a mixed graph. If the orientation of an arc is not relevant, we sometimes refer to an arc (u, v) as the edge uv of the underlying graph.

3 Relay Vertices and Wedges

Let G^2 be the square of a graph G. An edge e of G^2 is called a *real* edge if e is an edge of G, otherwise e is a *virtual* edge. A path in G^2 that contains only real edges

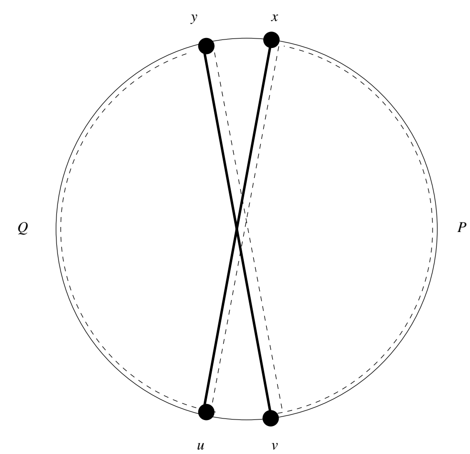

Figure 1: The Hamiltonian cycle H represented by a circle, and similarly H' by a dotted line.

is a *real path*, and similarly a path containing only virtual edges is a *virtual path.*

It was shown by Fleischner in [] that the square of any two connected graph is Hamiltonian, so we will suppose that G is two connected. Suppose that H is a Hamiltonian cycle of G^2 with the minimal number of virtual edges. For our purposes, we will place the vertices of H on a circle in the order they appear on H. Moreover, we will assume a counterclockwise orientation of H, and assign the corresponding orientation to each edge of H in G^2, so that they are now also refered to as arcs, so G^2 is a mixed graph. These arcs can be referred to as either arcs of G^2 or arcs of H, and notice that each vertex is incident with exactly two arcs.

Observation 3.1. *If (u, v) is any virtual arc, then there does not exist an arc (x, y) such that ux and vy are real edges.*

Proof. Suppose there does exist such an arc (x, y) of H such that ux and vy are real edges. Let P be the vx-path in H, and let Q be the yu-path in H. We can now construct a Hamiltonian cycle H' of G^2 by following P, then the edge xu, then Q, and finally the edge yv. See figure 1. Obviously H' has at least one fewer virtual edge than H has, a contradiction. □

For each virtual arc (u, v) we assign a real path of length 2 joining u and v which will be used by our labelling of ports. This assigned path is called the *relay path* of the arc (u, v). The fact that a relay path always exists follows from the definition of G^2. The unique internal vertex of a relay path is called the *relay vertex* of the arc. Note that more than one such real path may exist for a virtual arc. The relay paths can be chosen arbitrarily, it is only important that they are fixed. The real edges of a relay path may or may not be arcs, a real edge can be in more than one relay paths for different virtual arcs, and a vertex may be a relay vertex for more than one virtual arc.

If P is a maximal virtual path in H, containing at least one arc, where all virtual arcs share a common relay vertex w, then the subgraph W of G^2 consisting of P, the vertex w, and all real edges incident with both w and some vertex of P is called a *wedge*, see Figure 2 for an example.

Furthermore, we call P the *support* of the wedge, similarly an arc in P is a *support arc* of the wedge, and vertices of P are called *support vertices* of the wedge. The common relay vertex w is referred to as the *relay vertex of the wedge*. A real edge in W is called a *rib* of W. If P is a directed uv-path; we call u the *tail support vertex* of the wedge, and v the *head support vertex* of the wedge; these are both also referred to as *external vertices* of the wedge. Internal vertices of P are called *internal support vertices*, or simply *internal vertices*, of the wedge. The rib uw is the *left external rib* of W, and the rib wv the *right external rib* of the W; they are both referred to as *external ribs* of W. Real edges of W that are not external ribs are *internal ribs*. The *length* of a wedge W is the number of ribs of the wedge, denoted by $\ell(W)$. See Figure 2 for an example of a wedge containing five ribs and four support arcs.

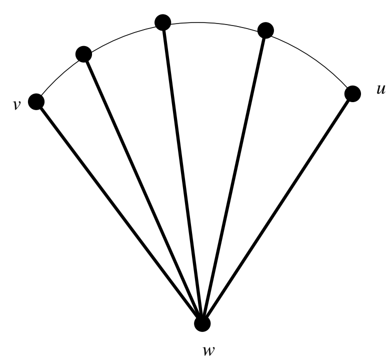

Figure 2: A wedge with four support arcs. The ribs are real edges of G^2 and are represented in this figure by thicker lines. The support path consists of virtual edges of G^2 and are represented by regular lines.

If the right external rib of a wedge W is also an arc, then W is a *right tied* wedge; similarly if the left external rib is an arc then W is a *left tied* wedge. Clearly, internal ribs of a wedge cannot be arcs; if any edge e is both a rib of W and an arc of G^2, it must be an external rib of W. Note that it is possible that a wedge is both right tied and left tied, in which case it contains all the vertices of G. A *tied wedge* is either right tied or left tied. A wedge that is not tied is called a *free wedge*. The following lemma characterizes how two wedges can interact.

Lemma 3.2. *Suppose W_1 and W_2 are two distinct wedges. Then*
(**a**) *There is no arc e that is a support arc of both W_1 and W_2.*

(**b**) *If W_1 and W_2 have the same relay vertex, they have no ribs in common.*
(**c**) *If e is an edge that is common to both W_1 and W_2, then e is the only edge they have in common. Furthermore, e is either the right external rib of both wedges, or the left external rib of both wedges.*

Proof. Let W_1 and W_2 be two distinct wedges, with relay vertices w_1, w_2 and supports P_1,P_2, respectively. To show (a), suppose to the contrary that e is a support arc of both W_1 and W_2. Now w_1 is the relay vertex for e, since e is a support arc for W_1. But e is also a support arc of W_2, so w_2 is the relay vertex for e. This forces $w_1 = w_2$. Since we supposed that $W_1 \neq W_2$, it must be that $P_1 \neq P_2$. Since e belongs to both P_1 and P_2, it must be the case that either one of the paths is a subpath of the other, contradicting the maximality of the former; or P_1 and P_2 overlap, in which case neither one is maximal, again a contradiction. So the wedges have no support arcs in common.

Next, we prove (b). Suppose that $w_1 = w_2$. Suppose a real edge e is a rib of both W_1 and W_2. If e is an internal rib of either wedge, say W_1, then there is at least one support arc of W_1 contained in W_2, which contradicts (a). Otherwise e is an external rib of both wedges. Suppose without loss of generality that e is the right external rib of W_1. If e is also the right external rib of W_2, then the wedges share at least one support arc, which contradicts (a). If e is the left external rib of W_2, then the union of P_1 and P_2 is a longer path than both P_1 and P_2 alone, and all arcs in the union are virtual and share the same relay vertex. This contradicts the choice of P_1 and P_2, hence the wedges have no ribs in common, so (b) is proven.

For (c), suppose that e is an edge common to both wedges. From (a), e is not a support arc, so it must be a rib of both wedges. Also, from (b) we know that $w_1 \neq w_2$. Since e is a rib of both W_1 and W_2, this forces $e = w_1w_2$. So w_1 is also a support vertex of W_2, and w_2 is also a support vertex of W_1; see Figure 3.

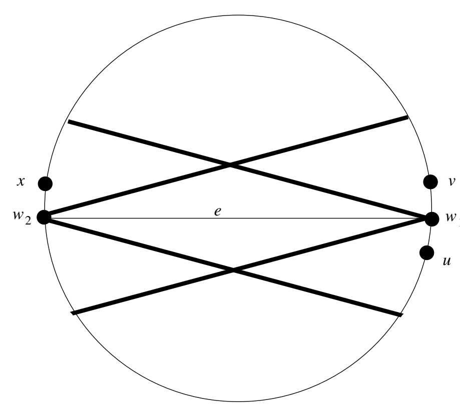

Figure 3: The edge e is a rib of both wedges.

Next we will show that w_1 is not an internal vertex of W_2, and w_2 is not an internal vertex of W_1. Suppose without loss of generality that w_1 is an internal vertex of W_2. Let $f = (u, w_1)$ and $g = (w_1, v)$, which are the two support arcs of W_2 incident with w_1; we know there are two such arcs since w_1 is internal. There is at least one arc of W_1 incident with w_2, call it h. Suppose that $h = (x, w_2)$. The edge xw_1 is a real edge since it is a rib of W_1, and vw_2 is a rib of W_2, which contradicts Observation 3.1. If $h = (w_2, x)$ we get a similar contradiction. Since all cases lead to contradiction, w_1 is an external vertex of W_2, and similarly w_2 is an external vertex of W_1. This forces e to be an external rib of both wedges.

Suppose without loss of generality that e is the left external rib of W_1 and the right external rib of W_2, see Figure 4. Let us suppose that w_1 is incident to a support arc (w_1, x) of the wedge W_2, and w_2 is incident to a support arc (y, w_2) of the wedge W_1. So yw_1 and xw_2 are real edges since they are ribs of their respective wedges. This condtradicts Observation 3.1, so (c) is proved. □

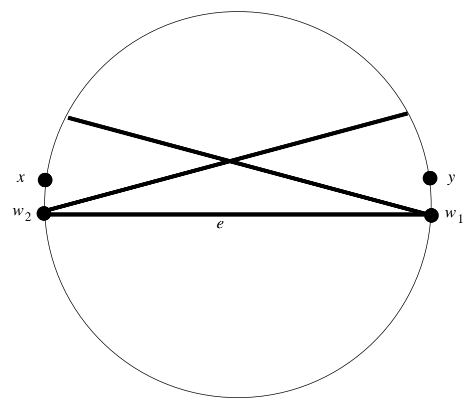

Figure 4: The edge e is an external rib of both wedges.

If a vertex v is the relay vertex of k wedges, we order these wedges at v based on their occurrence along H. The *list of ordered wedges* $W_1, W_2, \ldots, W_k = \{W_i\}_{i=1}^k$ *at* v is the ordered list of all wedges with the common relay vertex v such that for W_i and W_j with $i < j$, the support path of W_i occurs before the support path of W_j when following H starting from v; see Figure 5. This list may be denoted by $\{W_i\}_i$ if k is understood. If e is a real edge incident with v, we say that e is an *isolated edge* of v if e is not a rib of any wedge in the list of ordered wedges at v. When the list of ordered wedges is understood, we write ℓ_i for $\ell(W_i)$.

For each vertex v of G^2, we specify two edges incident with v in the following definition.

Definition 3.3. *Let* (x, v) *and* (v, y) *be the two arcs incident with* v*. The* incoming edge *of* v *is either the arc* (x, v)*, if* (x, v) *is a real arc, or the real edge* wv*, where* w *is the relay vertex of* (x, v)*. Similarly, the* outgoing edge *of* v *is either the arc* (v, y)*,*

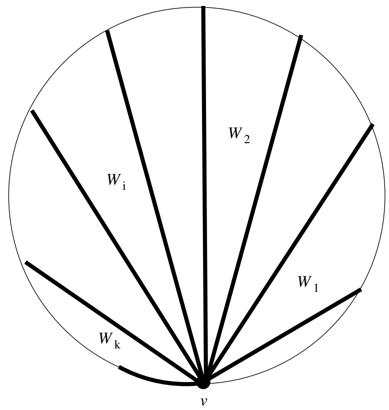

Figure 5: The list $W_1, W_2, \ldots, W_k$ is the list of ordered wedges at v.

if (v, y) is a real arc, or the real edge wv, where w is the relay vertex of (v, y). If the incoming edge of v is equal to its outgoing edge, this is called the backtrack edge *of v.*

Note that the incoming and outgoing edges of a vertex are always edges of G, and hence real edges of G^2. By definition, each vertex has exactly one incoming edge, one outgoing edge, and at most one backtrack edge. Since the relay vertices of all virtual arcs are fixed, the choice of incoming, outgoing, and backtrack edges are fixed as well for every vertex of G. In the next lemma, we characterize when a given real edge is a backtrack edge for its incident vertex.

Lemma 3.4. *For any vertex v and real edge e incident to v, e is a backtrack edge of v if and only if either e is the internal rib of some wedge with relay vertex $w \neq v$; or e is both an arc of H and an external rib of some tied wedge with relay vertex $w \neq v$, see Figure 6.*

Proof. Let (x, v) and (v, y) be the the two arcs of H incident with v. Let e be an edge incident to v. Suppose that e is an internal rib of some wedge W with relay vertex $w \neq v$. By the definition of a rib, $e = wv$ and v is a support vertex of W. Since (x, v) and (v, y) are both virtual arcs incident to v and v is a support vertex of W, this means both (x, v) and (v, y) are support arcs of W. Let w' be the relay vertex of (x, v), where w' is the relay vertex of the wedge W'. This means that (x, v) is a support arc of both W and W'. By Lemma 3.2, $W = W'$ and $w = w'$. So w is the relay vertex of the arc (x, v), and similarly w is the relay vertex of (y, v). Since (x, v) is a virtual arc with relay vertex w, the incoming edge of v is e. Similarly, e is the outgoing edge of v. Therefore e is the backtrack edge of v.

Suppose that e is both an arc of H and an external rib of some tied wedge W with relay vertex $w \neq v$. Since e is a rib of W, and $w \neq v$, then v is a support vertex

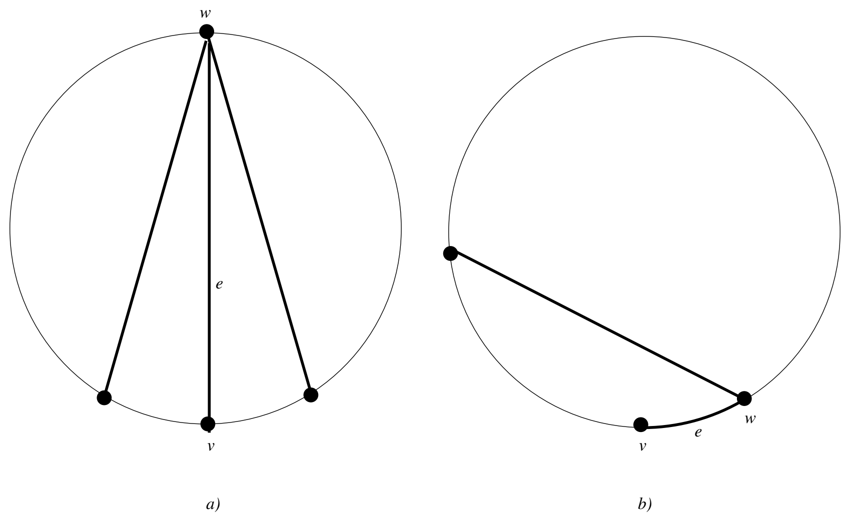

Figure 6: The two possible configurations for when e is a backtrack edge of v.
a) The edge e is an internal rib of W with relay vertex $w \neq v$.
b) The edge e is the right external rib of W with relay vertex $w \neq v$.

of W and $e = wv$. Let (x, v) and (v, y) be the arcs of H incident to v. Then, since e is an arc of H incident to v, either $e = (x, v)$ or $e = (v, y)$. Suppose without loss of generality that $e = (x, v)$. Since e is a real arc, e is the incoming edge of v. Also, since e is a rib of W, and v is a support vertex of W, (v, y) is a support arc of W and thus a virtual arc. By Lemma 3.2, w is the support vertex of (v, y). Now wv is the outgoing edge of v. Since $e = wv$, e is both the incoming edge and the outgoing edge of v, so e is a backtrack edge of v.

Now suppose that e is a backtrack edge of v. So e is a real edge that is both the incoming and outgoing edge of v. Let (x, v) and (v, y) be the arcs of H incident to v. Since H has at least three vertices, $(x, v) \neq (v, y)$. Suppose first that e is not an arc of H. Since e is the incoming edge of v and $e \neq (x, v)$, $e = vw_1$ where w_1 is the relay vertex of (x, v). Since e is the outgoing edge of v and $e \neq (v, y)$, $e = vw_2$ where w_2 is the relay vertex of (x, v). Thus $w_1 = w_2$. By Lemma 3.2, (x, v) and (v, y) are support arcs of some wedge W with relay vertex $w = w_1 = w_2$. Since e is a real edge incident to both v and w, e is a rib of W. Since there are two support arcs of W incident with v, e is an internal rib of W. Now suppose e is an arc of H. Suppose without loss of generality that $e = (x, v)$. Since e is the incoming edge of v, e is a real arc. If we suppose that (v, y) is a real edge, then (v, y) is the outgoing edge of v, a contradiction since $e \neq (v, y)$. Let w be the relay vertex of the virtal arc (v, y). Since e is the outgoing edge of v, $e = vw$. Since $e = (x, v)$, this forces $w = x$. Let W be the wedge with relay vertex w. So (v, y) is a support arc of W, and e is a rib of W. Since e is also an arc, e is an external rib of W, and W is a tied wedge. $\square$

In the following we determine how, in some cases, incoming and outgoing edges of some vertex v must look like. It is not difficult to observe that if v is not a relay vertex or a support vertex of any wedge, then the incoming and outgoing edges of v are real arcs of H which enter and leave v, respectively. Similarly, if v is an internal support vertex, then its incoming and outgoing edges are identical and constitute the backtrack edge of v. Moreover, if v is an external support vertex of exactly one wedge, then one of the two will be the external rib incident to v, and the other will be the real arc of H incident to v. If v is the external support vertex of two distinct wedges, then the incoming and outgoing edges will be the two external ribs of these wedges incident to v. The following lemma will determine how these edges occur in the case when v is identical to some relay vertex w.

Lemma 3.5. *Let W be a wedge with relay vertex w. Let e be the incoming edge of w, and f be the outgoing edge of w, and $e \neq f$. If e is a rib of W, then e is the right external rib of W. If f is a rib of W, then f is the left external rib of W.*

Proof. Let e be a rib of W, say $e = vw$ where v is a support vertex of W. Suppose first that e is an arc, so e is an external rib of the tied wedge W. Since e is the incoming edge of v, and e is an arc, we must have $e = (v, w)$. So by definition, W is a right tied wedge, and e is the right external rib of W. Now suppose that e is not an arc. Let (x, w) be an arc of H incident with w. Since $e \neq (x, w)$, and e is the incoming edge of w, v is the relay vertex of the arc (x, w). So vx is a real edge. Let (v, y) be an arc of H incident with v. Since e is a real edge, by Observation 3.1 there is no real edge joining y and w. Therefore (v, y) is not a support arc of W, and hence e is the right external rib of W. The argument for f is symmetric.

Suppose that f is an arc, so f is an external rib of the tied wedge W. Since e is the incoming edge of v, and e is an arc, we must have $f = (w, v)$. So by definition, W is a left tied wedge, and f is the left external rib of W. Now suppose that f is not an arc of H. Let (w, x) be an arc of H incident with w. Since $f \neq (w, x)$, and f is the incoming edge of w, v is the relay vertex of the arc (w, x). So vx is a real edge. Let (y, v) be an arc of H incident with v. Since f is a real edge, by Observation 3.1 there is no real edge joining y and w. Therefore (y, v) is not a support arc of W, and hence f is the left external rib of W.

□

The next section describes how to assign port numbers to the graph, and how the robot will traverse the graph using these port numbers.

4 Port Numbers

Let $e = uv$ be an edge of G. In this section, we assign two numbers (port numbers) to e, one port at each end vertex. The two port numbers for a given edge may be different. We say that a vertex v *sees* port number p, or p is *seen* at v, if there is some edge e incident to v such that the port of e at v is p. Otherwise v does not see the port number p, and we say that p is *unseen* at v. During during each step of the traversal, the robot will always visit exactly one current vertex. To *leave* a vertex u on port number p, the will traverse the edge $e = uv$ such that e at u has the port number p. The robot moves to the vertex v and this vertex becomes the current vertex of the next step. We also say that the robot *entered* v on the port q if q is the port number of e at v. If the current vertex u and the edge $e = uv$ are understood, we simply say that the robot will leave on p, or enter on q.

All port numbers for every vertex v are chosen uniquely from the set $\{1, 2, 3, \ldots, 2d(v)\}$. When comparing two port numbers, we use the natural ordering of integers.

We now describe how to assign an interval of port numbers to several edges of a wedge. Given any wedge W of width k with relay vertex w and support P, where P vertices $u = u_1, u_2, u_3, \ldots, u_k = v$ in order, we will assign the interval $[p, q]$ by assigning integers between p and q to ribs of W at w. This is only possible if the length of the interval equals the length of the wedge, meaning $k = q - p + 1$.

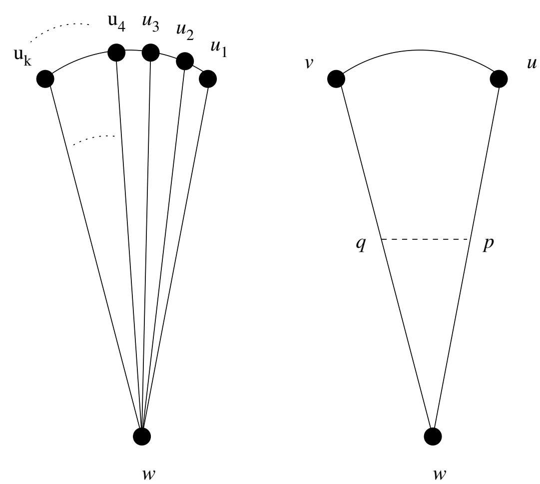

Figure 7: The first image is a wedge showing a few internal ribs; the second image shows port numbering notation.

We assign $[p, q]$ to the wedge W for $p < q \leq 2d(w)$, by assigning port number $p + i - 1$ to the edge wu_i at w, for $1 \leq i \leq k$. This is depicted in our figures with labels p and q on the external edges of W joined by a dotted line, see figure 7. We assign $[p, 1]$ to W for $p > 1$ by assignung port number $p + i - 1$ to the edge wu_i at w for $1 \leq i < k$, and then port number 1 to wu_k at w. We assign $[p, q] \cup \{r\}$ for $r < p < q < 2d(w)$ and $k = q - p + 2$ by assigning port number $p + i - 1$ to wu_i at w, and then port number r to wu_k at w. Furthermore, we assign an interval of labels $[p, q]$ to a series of wedges $W_1, W_2, \ldots, W_j$ by breaking $[p, q]$ into j disjoint subintervals, and assigning $[p, p+\ell_1-1]$ to W_1, $[p+\ell_1, p+\ell_1+\ell_2-1]$ to W_2, and so on. Recall that $\ell(W_i)=\ell_i$. Notice that assigning port numbers to a wedge in this way only assigns port numbers at the relay vertex of the wedge, not at any of the support vertices.

Next we describe the traversal rules that the robot will follow, after port numbers have been assigned. The robot will be programmed such that Rule 0 is only applied if v is the initial vertex, otherwise it will apply one of the other three rules. Then the applicable rule instructs the robot on how to move to a new current vertex in the next step.

- **Rule 0**: If v sees port 1, leave on 2. If 1 is unseen at v and $2d(v)$ is seen at v, leave on $2d(v)$. If both 1 and $2d(v)$ are unseen at v, leave on $2d(v)-1$.
- **Rule 1**: If v was entered on p for $1 \le p \le 2d(v)-2$, and p is not the largest port number seen at v, leave v on port q such that q the smallest port number larger than p that is seen at v. If p is the largest port number seen at v, leave on 1.
- **Rule 2**: If v was entered on $2d(v)-1$ and v sees $2d(v)$, leave v on $2d(v)$. If v was entered on $2d(v)-1$ and $2d(v)$ is unseen at v, leave v on port number $q < 2d(v)$ such that $q-1$ is the largest unseen port number at v.
- **Rule 3**: If v was entered on $2d(v)$ and v sees port 1, then leave on 1. If port 1 is unseen, leave v on $2d(v)$.

The assignment of the port numbers will be accomplished by applying the procedure destailed below to each vertex of G.

---

procedure Port-Numbering(vertex v)

Let a be the incoming edge of v
Let b be the outgoing edge of v
Let k be the number such that v is a relay vertex of k distinct wedges
if $k > 0$ **then**
    Let $\{W_i\}_{i=1}^k$ be the list of ordered wedges at v
end if
if $a = b$ **then**
    Assign $2d(v)$ to a at v
    if $k > 0$ **then**
        Assign $[2, \sum_{i=1}^k \ell_i + 1]$ to $\{W_i\}_{i=1}^k$ { *Figures 19, 25, 31* }
    end if

end if
if $a \neq b$ and both a and b are isolated edges of v **then**
  Assign 1 to a at v
  Assign 2 to b at v { *Figures 9, 14, 20, 26* }
  if $k > 0$ **then**
    Assign $[3, \sum_{i=1}^{k} \ell_i + 2]$ to $\{W_i\}_{i=1}^{k}$
  end if
end if
if $a \neq b$ and both a and b are not isolated edges of v **then**
  if W_1 is both right tied and left tied **then**
    Assign $[1, \ell_1]$ to W_1 { *Figure 13* }
  else
    Pick j such that a is a rib of W_j
    Pick j' such that b is a rib of $W_{j'}$
    if $j = j'$ **then**
      if W_j is either left tied or right tied **then**
        Assign $[2d(v) - \ell_j, 2d(v) - 1]$ to W_j { *Figures 18, 24* }
      end if
      if W_j is a free wedge **then**
        Assign $[2d(v) - \ell_j + 1, 2d(v) - 1] \cup \{2d(v) - \ell_j\}$ to W_j { *Figure 30* }
      end if
      if $k \geq 2$ **then**
        Assign $[2, \sum_{i \neq j} \ell_i + 1]$ to $\{W_i\}_{i \neq j}$
      end if
    end if
    if $j \neq j'$ **then**
      Assign $[2, \ell_{j'} + 1]$ to $W_{j'}$ { *Figures 12, 17,* **??**, *29* }
      Assign $[2d(v) - \ell_j + 2, 1]$ to W_j
      if $k \geq 3$ **then**
        Assign $[\ell_{j'} + 2, \sum_{i \neq j, i \neq j'} \ell_i + 1]$ to $\{W_i\}_{i \neq j, i \neq j'}$
      end if
    end if
  end if
end if
if $a \neq b$, a is an isolated edge of v but b is not **then**
  Assign 1 to a at v
  Pick j such that b is a rib of W_j
  Assign $[2, \ell_j + 1]$ to W_j { *Figures 11, 16, 22, 28* }
  if $k \geq 2$ **then**
    Assign $[\ell_j + 2, \sum_{i \neq j} \ell_i + 1]$ to $\{W_i\}_{i \neq j}$
  end if

53: **end if**
54: **if** $a \neq b$, b is an isolated edge of v but a is not **then**
55:     Assign 2 to b at v
56:     Pick j such that a is a rib of W_j
57:     Assign $[2d(v) - \ell_j + 2, 1]$ to W_j { *Figures 10, 15, 21, 27* }
58:     **if** $k \geq 2$ **then**
59:         Assign $[\ell_j + 2, \sum_{i \neq j} \ell_i + 1]$ to $\{W_i\}_{i \neq j}$
60:     **end if**
61: **end if**

By the construction of this procedure, and from our interval notation, it is clear that assigned ports will be unique at each vertex. It is apparent that the Port-Numbering procedure assigns ports to a vertex in one of five subroutines (starting at lines 7, 13, 20, 46, and 54) based on structural conditions of G^2 and the chosen Hamiltonian cycle. We will be referring to these subroutines by line number on which they begin. It is clear that no vertex will enter more than one of these five subroutines. By the construction of these subroutines, and from our interval notation, it is clear that assigned ports will be unique at each vertex. The only subroutine that could potentially assign two different port numbers to the same place is at line 7. But this cannot happen, since Lemma 3.4 guarantees that any backtrack edge of a vertex v will also be an isolated edge of v.

Based on the Port-Numbering procedure, the following two lemmas show that all the traversal rules are well defined.

Lemma 4.1. *Every vertex v sees at least one of the ports 1, $2d(v) - 1$ or $2d(v)$. If v sees port 1, v also sees port 2.*

Proof. Let v be any vertex, a be the incoming edge of v, and b be the outgoing edge of v. If $a = b$, then the subroutine at line 7 assigned port numbers to v, and v sees port $2d(v)$ by line 8. Moreover, port 1 is not assigned to v in this subroutine. Otherwise $a \neq b$. Suppose that a is an isolated edge of v. If b is also an isolated edge of v, then the subroutine at line 13 assigned ports to v. So v sees port 1 by line 14 and port 2 by line 15. If b is not an isolated edge of v, then the subroutine on line 46 assigned ports to v. So v sees port 1 by line 47 and port 2 to 49.

Now suppose that a is not an isolated edge of v. If b is an isolated edge of v, then the subroutine from line 54 assigned ports to v. So v sees port 1 by line 57 and port 2 by line 55. If b is not an isolated edge of v, then the subroutine on line 20 assigned ports to v. Suppose that v is the relay vertex of k distinct wedges. Since there is at least one edge of v that is not isolated, $k \geq 1$. Let $W_1, W_2, \ldots, W_k$ be the list of ordered wedges at v. If there is a wedge with relay vertex v that is both right tied and left tied, then this wedge is W_1 and $k = 1$. By line 22 v sees both ports 1 and 2. Otherwise suppose there is no such wedge. Since a is not an isolated edge of v, it is a rib. Suppose a is a rib of W_j. Similarly suppose b is a rib of $W_{j'}$, where $1 \leq j, j' \leq k$. If $j = j'$, then either by line 28 or 31 (depending on whether or not W_j is a tied wedge), port $2d - 1$ is seen at v. Moreover, port 1 is not assigned to v on these two lines. If $j \neq j'$, then v sees port 1 by line 39 and port 2 by line 38. This concludes all cases, and thus the proof. □

This ensures that Rule 0 and Rule 1 are well defined. Notice that Rule 3 is well defined wherever it is applicable by definition. The following lemma shows that Rule 2 is well defined.

Lemma 4.2. *At every vertex v, there is a port less than $2d(v)$ that is unseen at v.*

Proof. Let v be any vertex of G. Since G is two connected, $d(v) \geq 2$ and therefore $2d(v) \geq 4$. Recall that only edges of G are assigned port numbers, so v will see at most $d(v)$ ports. Let S be the set of port numbers seen at v. Since $2d(v) - 1 \geq 3$, we have $2d(v) - 1 > d(v)$, so by the pigeonhole principle there is at least one port p in $[1, 2d(v) - 1]$ that is not in the set S. Therefore $p < 2d(v)$ and p is unseen at v. □

We know now that the four traversal rules are well defined, so we will assume in proofs that whenever a rule is used, the corresponding ports are seen. Next we define an invariant which will guarantee, under certain conditions, that the robot will start periodically traversing G. Let H^* be the graph constructed by starting from H, then for every virtual arc (u, v) of H with relay vertex w, add the real edges (with the new orientation) (u, w) and (w, v) to H^*, and delete the virtual arc (u, v) from H^*. Naturally, H^* induces a (directed) Hamiltonian walk of G by following H, where each virtual arc is traversed via the corresponding relay path. For convenience, and to futher distinguish from arcs of H, we refer to arcs of H^* as simply edges of H^*. It is clear that the underlying graph of H^* contains only real edges of G^2, so H^* can be treated as subgraph of G. We will abuse notation and identify the directed graph H^* with this Hamiltonian walk of G, denote both by H^*. In this context, we can say that an (undirected) edge and an arc (directed edge) are equal if their end vertices are equal, so we identify edges of H^* with their corresponding edges in G. If we refer to the port number of an edge of H^*, this of course means the appropriate port number in G, since port numbers are only assigned in G and not in H^*. This means, for example, that both (v, u) and (u, v) will always have the same port number at v, even though v only sees this port number once in G. It is also important to notice that if $e = uv$ is in the relay path of some virtual arc of H, it is possible (but not neccesarily the case) that both (u, v) and (v, u) are in H^*.

From the definition of H^*, we can make several simple observations. If $e = uv$ is the incoming edge of v, and vw is the outgoing edge of v, then (u, v) and (v, w) are in H^*. If W is a wedge, and wx is any internal rib of W, then both (w, x) and (x, w)

are in H^*. If uw is the left external rib of the wedge W, and vw is the right external rib of W, then (u, w) and (w, v) are in H^*.

Naturally, H^* induces a (directed) Hamiltonian walk of G by following H, where each virtual arc of H is traversed via the corresponding relay path in G. This gives rise to the natural successor function defined on edges of H^*; the *successor* of an edge of e in H^* is the edge following e on H^*, denoted by e^+. To leave a vertex v on an edge of H^*, there is a unique edge of H^* for which v is its tail. Clearly if $e = (u, v)$, then $e^+ = (v, w)$ for some vertex w. With this in mind, we define an invariant which will guarantee, under certain conditions, that the robot will start periodically traversing G.

The Invariant: *If the robot enters a new current vertex on an edge e in H^*, then the robot will leave on the edge e^+ in H^*.*

It is not difficult to see that the invariant guarantees a periodic traversal once a robot uses an edge of H^* to enter a vertex. Before we prove that the invariant always holds, we first prove that starting from any initial vertex, the robot always leaves the vertex on an edge of H^*.

Lemma 4.3. *Let v be the initial vertex. Then the robot will leave v on an edge $e = (v, u)$ in H^*.*

Proof. Suppose that the robot starts at v, and leaves v on the edge $e = (v, u)$. It is sufficient to show that either e is the outgoing edge of v, or e is the right external rib or an internal rib of some wedge. Let a be the incoming edge of v, b be the outgoing edge of v, and suppose that v is the relay vertex of k distinct wedges. Let p be the port of e at v. The only rule that applies to v is Rule 0 (since v is the initial vertex). There are three cases: p is either 2, $2d(v) - 1$, or $2d(v)$.

Case 1: Suppose first that $p = 2$. Since G is two connected, $d(v) \geq 2$, so $2d(v)-1 \geq 3 > 2$, and by Rule 0, port 1 is seen at v. If $a = b$, then the subroutine on line 7 was used to assign ports to v. Since $2d(v) \neq 1$, port 1 is not assigned by either line 10 or 8. This is a contradiction, so $a \neq b$.

Suppose that b is an isolated edge of v. So one of the subroutines on line 13 or 54 was used at v. Both assign port 2 to b at v. So $e = b$, which is an edge of H^*.

Now suppose b is not an isolated edge of v. Suppose further that a is an isolated edge of v. So the subroutine at line 46 was executed at v. Since b is not an isolated edge of v, there exists a wedge W with relay vertex v such that b is a rib of W. By Lemma 3.2, b is the left external rib of W. The line 49 assigned port 2 the left external rib of W. So $e = b$, which is an edge of H^*. Now suppose a is not an isolated edge of v, so the subroutine at line 20 was executed at v. Since a and b are not isolated edges of v, this implies there is at least one wedge with relay vertex v, so $k > 0$. Let $W_1, W_2, \ldots, W_k$ be the list of ordered wedges at v. Suppose that one of these wedges, say W_j, is both right tied and left tied. This implies that W_j contains all the vertices of G, we see that $k = j = 1$. Line 22 assigned the interval $[1, \ell_1]$ to

W_1. Since $\ell_1 \geq 2$, the edge e is either an internal rib or the right external rib of W_1. Thus e is an edge of H^*. Otherwise there is no wedge that is both right tied and left tied. Since a and b are not isolated edges of v, a is a rib of W_j and b is a rib of $W_{j'}$, for some $1 \leq j, j' \leq k$. By Lemma 3.2, a is the right external rib of W_j, and b is the left external rib of $W_{j'}$. Suppose that $j \neq j'$, so line 38 assigned port 2 to the b, and $e = b$ which is an edge of H^*. Now suppose $j = j'$. In this case, we show that port 1 is not assigned at v, which is a contradiction. Indeed, no edge of W_j is assigned port 1 at v (by lines 28 and 31) because the smallest such port number is $2d(v) - \ell_j$. And since $2 \leq \ell_j \leq d(v)$, we have $2d(v) - \ell_j \geq d(v) > 1$. Moreover, edges of all remaining wedges with relay vertex v are assigned ports no smaller than 2 by line 34. This is a contradiction.

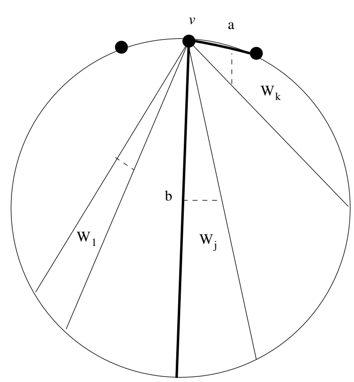

Figure 8: The incoming edge of v is an isolated edge; the outgoing edge of v is an arc and an external rib of a right tied wedge

Case 2: Suppose next that $p = 2d(v) - 1$. Since Rule 0 was used, and $2d(v) - 1 > 2$, ports 1 and $2d(v)$ are both unseen at v. If $a = b$, we obtain a contradiction since port $2d(v)$ is seen at v because of line 8. So $a \neq b$.

Suppose b is an isolated edge of v. If we suppose further that a is also an isolated edge, then the subroutine at line 13 assigned ports to v. But now port 1 was assigned to v on line 14, contradicting our assumption. Suppose now that a is not an isolated edge of v, so the subroutine at line 54 assigned ports to v. But now port 1 was assigned to v on line 57, a contradiction.

Now suppose that b is not an isolated edge of v. If a is an isolated edge of v, then the subroutine at line 46 assigned port 1 to v, a contradiction. So a is also not an isolated edge of v, and that the subroutine at line 20 assigned ports to v. Moreover, $k > 0$. If there is a wedge W with relay vertex v that is both left tied and right tied, then by line 22, v would see port 1, a contradiction. Thus there is no such wedge. Let $W_1, W_2, \ldots, W_k$ be the list of ordered wedges at v. Since a and b are not isolated edges of v, a is a rib of W_j and b is a rib of $W_{j'}$, for some $1 \leq j, j' \leq k$. By Lemma 3.2, a is the right external rib of W_j, and b is the left external rib of $W_{j'}$. Suppose

$j = j'$. If W_j is either left tied or right tied, then by line 28, e is the right external rib of W_j and thus an edge of H^*. If W_j is a free wedge, then by line 31, and since $p = 2d(v) - 1$, e is an internal rib of W_j and thus an edge of H^*. If $j \neq j'$, then by line 39, port 1 is seen at v, a contradiction.
Case 3: Finally, suppose $p = 2d(v)$. So port 1 is unseen at v by Rule 0. Much of the arguments here are similar to case 2, since we again obtain a contradiction if port 1 was assigned at v. If $a = b$, then port $2d(v)$ was assigned to b in the subroutine at line 7, so $e = b$ which is an edge of H^*. So now $a \neq b$. All of the following subcases lead to a contradiction.

Suppose that b is an isolated edge of v. If a is also an isolated edge of v, the subroutine at line 13 assigned ports to v. But now port 1 was assigned to v on line 14, contradicting our assumption. Suppose now that a is not an isolated edge of v, so the subroutine on line 54 was used at v. Now port 1 was assigned to v on line 57, contradicting our assumption.

Now suppose that b is not an isolated edge of v. If a is an isolated edge of v, the the subroutine on line 46 was executed at v. This is a contradiction since port 1 is assigned to v at line 47. Now suppose a is not an isolated edge of v. So $k > 0$ and the subroutine at line 20 was used at v. Suppose that there is a wedge W with relay vertex v that is both right tied and left tied. Then by line 22, port 1 is seen at v, a contradiction. So there is no such wedge. Let $W_1, W_2, \ldots, W_k$ be the list of ordered wedges at v. Let a be a rib of W_j and b a rib of $W_{j'}$, for some $1 \leq j, j' \leq k$. By Lemma 3.2, a is the right external rib of W_j, and b is the left external rib of $W_{j'}$. Suppose $j = j'$. All cases are handled the same; we can see by lines 28, 31, and 34 that the maximum port number assigned to v is $2d(v) - 1$. This follows since $\sum_i \ell_i + 1 \leq d(v) + 1 < 2d(v)$. This is a contradiction, since $p = 2d(v)$. So now suppose $j \neq j'$. By 39, port 1 is seen at v a contradiction.

Thus in all cases, the robot will leave v on an edge e of H^*.

□

Now we know that the robot enters the second vertex along an edge in H^*, so we next need to show that the invariant holds for any vertex. To this end, we first, we prove one supporting lemma that will be used in the proof of the invariant.

Lemma 4.4. *If the robot enters vertex v on the edge $e = (u, v)$, and e is the right external rib of some wedge W with relay vertex v, then e is the incoming edge of v.*

Proof. Recall that since e is an edge of H^*, there are three cases: either e is also a real arc of H, or v is the relay vertex for some virtual arc (u, x) of H, or u is the relay vertex for some virtual arc (x, v) of H.

Let P be the support path of W. First, suppose that e is a real arc of H. Then by Definition 3.3, e is the incoming edge of v. Next suppose u is the relay vertex for some virtual arc (x, v) of H. This again implies by Definition 3.3 that e is the incoming edge of v. Finally, suppose that v is the relay vertex for some virtual arc (u, x) of H. So $P + (u, x)$ is a virtual path, which contradicts the maximality of P. □

The following lemma proves that the invariant holds.

Lemma 4.5. *Let v be the current vertex of the robot, and suppose the robot enters a v along the edge e of H^*. Then the robot will leave v on e^+.*

Proof. Suppose $e = (u, v)$ has port number p at v. Let a be the incoming edge of v, b the outgoing edge of v, and v be the relay vertex of k wedges.

Suppose first that $a = b$, then the subroutine on line 7 was executed to assign port numbers to v. If $p = 2d(v)$, then $e = a = e^+$ by line 8 (except that e^+ has the opposite direction of e in H^* of course). Port 1 is unseen at v since $2d(v) \neq 1$ and by line 10. Now by Rule 3, the robot will leave v on e^+. So now $p < 2d(v)$. Since e is neither the incoming nor outgoing edge of v, e is a rib of some wedge with relay vertex v, (by the definition of H^*). Since this implies $k > 0$ we let $W_1, W_2, \ldots, W_k$ be the list of ordered wedges at v. So e is a rib of some wedge in this list, say W_j. By line 10, $2 \leq p \leq \sum_i \ell_i + 1$. Since $e \neq a$ we know by Lemma 4.4 that e is not the right external rib of W_j. This implies $p \neq \sum_i \ell_i + 1$, and that there is a support arc (u, x) of W_j for some vertex x such that $e^+ = (v, x)$. Clearly, the edge vx was assigned port $p+1$ at v. Since $p < \sum_i \ell_i + 1 \leq d(v) \leq 2d(v) - 1$, we have $p \leq 2d(v) - 2$, so by Rule 1 the robot will leave v on e^+. From now on suppose $a \neq b$.

Suppose that b is an isolated edge of v. So either the subroutine on line 13 or line 54. If $p = 2$, then $e = b$ by either line 15 or 55; we will show that this leads to a contradiction. To this end, first recall that since e is an edge of H^*, there are three cases: either e is a real arc of H, v is the relay vertex for some virtual arc (u, x) of H, or u is the relay vertex for some virtual arc (x, v) of H, or e is a real arc of H. In these cases we call e a a left relay edge, a right relay edge, or a real edge, repectively. If e is a left relay edge then e is not an isolated edge of v. This is a contradiction since $e = b$ and e is an isolated edge of v. If e is a right relay edge then e is the incoming edge of v. This is a contradiction since $e = b \neq a$. Therefore, $p \neq 2$. Finally suppose e is a real edge, so e is a real arc of H (and e must follow the same counterclockwise orientation as the arcs of H). But since b has the orientation (v, u) in H, and $e = (u, v)$, it follows that uv is the only edge in G. This implies $a = b$, a contradiction. Therefore, $p \neq 2$.

Now suppose a is an isolated edge, so the subroutine on line 13 assigned ports to v. If $p = 1$, then $e = a$ by line 14, which implies $e^+ = b$. Since b has port 2 at v by line 15, the robot will leave v on e^+ by Rule 1. Now let $p > 2$. Since a and b have ports 1 and 2 at v, respectively, $e \neq a$ and $e \neq b$. This implies that e is a rib of some wedge with relay vertex v, so $k > 0$. Let $W_1, W_2, \ldots, W_k$ be the list of ordered wedges at v. So e is a rib of some wedge in this list, say W_t. Since e is not the outgoing edge of v, by Lemma 4.4 e is an internal or the left external rib of W_t. So $e^+ = (v, x)$ where (u, x) is a support arc of W_t. Since e (in particular) is not the right external rib of W_k, we know by by line 17 that, $p < \sum_i \ell_i + 2$ and e^+ has port $p+1$ at v. Since v has at least two distinct isolated edges, $\sum_i \ell_i \leq d(v) - 2$, so Since $p < \sum_i \ell_i + 2 \leq d(v) + 2 - 2 = d(v) \leq 2d(v) - 2$, the robot will leave v on e^+ by Rule 1.

Now suppose that a is not an isolated edge of v. Since b is an isolated edge of v (by our above supposition) the subroutine on line 54 was used at v. Since a is not an isolated edge of v, $k > 0$, so let $W_1, W_2, \ldots, W_k$ be the list of ordered wedges at v. By Lemma 3.5, a is the right external rib of some wedge in this list, say W_j. And by line 57, a has port 1 at v. So if $p = 1$, then $e = a$ by line 57, which implies $e^+ = b$. Since e^+ has port 2 at v, by Rule 1 the robot will leave v on e^+. Let $p > 2$. Now, since a and b have ports 1 and 2 at v, respectively, $e \neq a$ and $e \neq b$. This implies e is a rib of some wedge with relay vertex v, say W_t. If $j = t$, then by line 57 $2d(v) - \ell_j \leq p \leq 2d(v)$. Furthermore, $e^+ = (v, x)$ is a rib of W_j such that (u, x) is a support arc of W_j. If $p = 2d(v)$, then e^+ has port 1 at v, so by Rule 3, the robot will leave v on e^+. If $p = 2d(v) - 1$, then e^+ has port $p + 1 = 2d(v)$ at v, and the robot will leave v on e^+ by Rule 2. If $2d(v) - \ell_j \leq p \leq 2d(v) - 2$, then e^+ has port $p+1$ and the robot will leave v on e^+ by Rule 1. Now suppose $t \neq j$. Since $e \neq a$, we know by Lemma 4.4 that e is either an internal rib or the right external rib of W_t. So $\ell_j + 2 \leq p < \sum_{i \neq j} \ell_i + 1$ by line 59, and $e^+ = (v, x)$ such that (u, x) is a support arc of W_t. This implies e^+ has port $p+1$ at v. Since $p < \sum_i \ell_i + 1 \leq d(v) + 1 \leq 2d(v) - 1$, it follows that $p \leq 2d(v) - 2$, so the robot will leave v on e^+ by Rule 1.

Next, we suppose that b is not an isolated edge of v, which implies $k > 0$. Let $W_1, W_2, \ldots, W_k$ be the list of ordered wedges at v. Suppose further that a is an isolated edge of v, so the subroutine at line 46 assigned port numbers to v. Since b is not an isolated edge of v, b is a rib of some wedge with relay vertex v, say W_j. By Lemma 3.5, b is the left external rib of W_j. If $p = 1$, then $e = a$, which implies $e^+ = b$. By line 49, e^+ has port 2 at v, so by Rule 1, the robot will leave v on e^+. Otherwise $p > 1$, so $e \neq a$ and e is not an isolated edge of v. Hence e is a rib of some wedge with relay vertex v, say W_t. Since $e \neq a$, e is either the left external or an internal rib of W_t by Lemma 4.4. In either case, $e^+ = (v, x)$ such that (u, x) is a support arc of W_t. This implies e^+ has port $p + 1$ at v. By either line 49 or 51 (depending on whether or not $t = j$), $2 \leq p \leq \sum_{i \neq j} \ell_i + 1$. Since e is (in particular) not the right external rib of W_k, $p < \sum_i \ell_i + 1 \leq d(v) + 1 \leq 2d(v) - 1$. So by Rule 1 the robot will leave v on e^+.

Finally, suppose that a is not an isolated edge of v, which implies (since we supposed b is not an isolated edge of v above) that the subroutine on line 20 was used at v. Since both a and b are not isolated edges of v, a is a rib of W_j and b is a rib of $W_{j'}$ for some $1 \leq j, j' \leq k$. By Lemma 3.5, a is the right external rib of W_j and b is the right external rib of $W_{j'}$. If there is a wedge with relay vertex v that is both left tied and right tied, it must be W_1 (since it contains all vertices of G). This also implies that $k = j = j' = 1$. By line 22, b has port 1 at v, and a has port ℓ_1 at v. If $p = \ell_1$, then $e = a$, which implies $e^+ = b$. Since W_1 contains all vertices of G, ℓ_1 is the largest port number seen at v. Since we also have $\ell_1 \leq d(v) \leq 2d(v) - 2$, the robot will leave v on e^+ by Rule 1. Otherwise $1 \leq p < \ell_1$, which implies $e^+ = (v, x)$ is a rib of W_1 such that (u, x) is a support arc of W_1. So e^+ has port $p + 1$ at v, and the robot will leave v on e^+ by Rule 1. Now suppose there is no such wedge. Suppose that $j = j'$. If we suppose that e is a rib of some wedge other than W_j, say W_t, then clearly $k \geq 2$. So $2 \leq p \leq \sum_{i \neq j} \ell_i + 1$ by line 34. By Lemma 4.4, since e is not the incoming edge of v, e is not the right external rib of W_t. So $e^+ = (v, x)$ such that (u, x) is a support arc of W_t. This implies e^+ has port $p+1$ at v. So by Rule 1, the robot will leave v on e^+. Now let e be a rib of W_j. If W_j is a left tied or a right tied wedge, then by line 28, b has port $2d(v) - \ell_j$ at v, and a has port $2d(v) - 1$ at v. Suppose $p = 2d(v) - 1$. So $e = a$ and $e^+ = b$. Let $q = 2d(v) - \ell_j$. Clearly $q - 1$ is unseen at v if $k < 2$, so it is the largest unseen port at v. So by Rule 2, the robot will leave v on e^+. Suppose $k \geq 2$, so the port $\sum_{i \neq j} \ell_i + 1$ is the largest port seen at v outside of W_j. But notice that $d(v) < 2d(v) - \ell_j$ holds for any j as long as $k > 1$, so $\sum_{i \neq j} \ell_i + 1 \leq d(v) - 2 + 1 = d(v) - 1 < 2d(v) - \ell_j - 1 = q - 1$. So $q - 1$ is the largest unseen port at v, and $2d(v)$ is unseen at v, thus the robot will leave v on e^+ by Rule 2. Now $p < 2d(v) - 1$. Since e is not the right external rib of W_j by Lemma 4.4, port $p + 1$ was assigned to e^+ which is also a rib of W_j. So the robot will leave v on e^+ by Rule 1. Now suppose W_j is a free wedge. By line 31, a has port $2d(v) - \ell_j$ at v and b has port $2d(v) - \ell_j + 1$ at v. If $p = 2d(v) - \ell_j$, then $e = a$. So $e^+ = b$, which has port $2d(v) - \ell_j + 1 = p + 1$ at v. By Rule 1, and since $p \leq 2d(v) - 2$, the robot leaves v on e^+. Let $p = 2d(v) - 1$ and $q = 2d(v) - \ell_j$. So e is the (specific) internal rib of W_j such that (u, x) is a support arc of W_j and $a = vx$. This means $e^+ = a$, which has port q at v. By the same argument as above (in the case where W_j was not a free wedge), $q - 1$ is the largest unseen port at v, so the robot will leave v on e^+. Now let $2d(v) - \ell_j < p \leq 2d(v) - 2$. Since e is not the right external rib of W_j, and by a similar argument to what we have seen several times above, the robot will leave v on e^+.

Finally, suppose $j \neq j'$. So a has port 1 at v, and b has port 2 at v, by lines 39 and 38 respectively. If e is a rib of some wedge other than W_j and $W_{j'}$, say W_t, then clearly $k \geq 3$. So $\ell_{j'} + 2 \leq p \leq \sum_{i \neq j, i \neq j'} \ell_i + 1$ by line 41. As above, e cannot be the right external rib of W_t, and the robot will leave v on e^+ by the same reasoning. Suppose first that e is a rib of W_j. If $p = 1$ then $e = a$ which implies $e^+ = b$. Since e^+ has port 2 at v, the robot will leave v on e^+ by Rule 1. If $p = 2d(v)$, then e is the (specific) internal rib of W_j such that (u, x) is a support arc of W_j and $a = vx$. So $e^+ = (v, x)$ which has port 1 at v. So by Rule 3 the robot will leave v on e^+. Similarly, we get the same result if $2d(v) - \ell_j + 2 \leq p \leq 2d(v) - 1$. Now, if e is a rib of $W_{j'}$, then $2 \leq p \leq \ell_j + 1$. Since e is not the incoming edge of v, Lemma 4.4 means

e is not the right external rib of $W_{j'}$. By the same argument as above, the robot will leave v on e^+.

$\square$

...

5 Results

...

Theorem 5.1. *Let G be a graph, and H be a Hamiltonian cycle of G^2 with the minimal number of virtual edges. If the ports of G are labelled with the Port-Numbering procedure, then a robot programmed with the four traversal rules will perform a periodic traversal of G with period $\pi(n) \leq 2n - 2$.*

Proof. We know from Lemma 4.5 that the invariant holds. So the robot visits every vertex of G by traversing the Hamiltonian cycle H of G^2. Recall that the period $\pi(n)$ of the traversal is the maximum number of edge traversals between two consecutive visits of any vertex. For any arc $e = (u, v)$ of H, the robot will traverse at most two edges to get from u to v; one if e is real and two if e is virtual. The maximum number of edge traversals between two visits of v will clearly occur when v traverses the entire cycle H after leaving v. Suppose that H has i virtual arcs, and j real arcs, where necessarily $i + j = n$. Thus the number of edges of G traversed is $2i + j$.

Next we claim that H always has at least two real arcs. Suppose to the contrary that H has only virtual arcs, and let $e = (u, v)$ be any virtual arc of H. Let w be the relay vertex for e; by definition $w \neq u$ and $w \neq v$. Construct H' by first removing e and then adding the edges wu and wv, so clearly H' is a Hamiltonian cycle of G. Since H was chosen to have the minimal number of virtual arcs, and H' has two fewer virtual arcs, we have a contradiction.

So to maximize the number of edges traversed $2i + j$, we maximize i by setting $i = n - 2$, and $j = 2$, to obtain $\pi(n) \leq 2(n - 2) + 2 = 2n - 2$.

□

...

6 References

References

[1] S. Albers, M.R. Henzinger. Exploring unknown environments. *SIAM J. Computing* **29** (2000) 1164–1188.

[2] B. Awerbuch, M. Betke, R. Rivest, M. Sing. Piecemeal graph exploration by a mobile robot. *Information and Computation* **152**(2) (1999) 155–172.

[3] M. Bender, D.K. Slonim. The power of team exploration: two robots can learn unlabeled directed graphs. *35th Annual Symposium on Foundations of Computer Science (FOCS).* (1994) 75–85.

[4] M. Bender, A. Fernandez, D. Ron, A. Sahai, S. Vadhan. The power of a pebble: Exploring and mapping directed graphs. *Information and Computation* **176**(1) (2002) 1–21.

[5] P. Bose, P. Morin. An improved algorithm for subdivision traversal without extra storage. *Internat. J. Comput. Geom. Appl.* **12**(4) 297–308 (2002)

[6] A. Blum, P. Raghavan, B. Schieber. Navigating in unfamiliar geometric terrain. *SIAM Journal on Computing*, **26** (1997) 110-137.

[7] L. Budach. Automata and labyrinths. *Mathematische Nachrichten* (1978) 195–282.

[8] E. Chávez, S. Dobrev, E. Kranakis, J. Opatrny, L. Stacho, J. Urrutia. Traversal of a quasi-planar subdivision without using mark bits. *Journal of Interconnection Networks* **5**(4) (2004).

[9] S. Cook, C. Rackoff. Space lower bounds for maze threadability on restricted machines. *SIAM J. Computing* **9**(3)(1980) 636–652.

[10] R. Cohen, P. Fraigniaud, D. Ilcinkas, A. Korman, D. Peleg. Label-guided graph exploration by a finite automaton. *ACM Transactions on Algorithms* **4**(4) (2008) 331–344.

[11] M. de Berg, M. van Kreveld, R. van Oostrum, M. Overmars. Simple traversal of a subdivision without extra storage. *International Journal of Geogrpahic Information Systems*, **11** 349–373, (1997).

[12] X. Deng, T. Kameda, C.H. Papadimitriou. How to learn an unknown environment: The rectilinear case. *Journal of the ACM*, **45** (1998) 215–245.

[13] X. Deng, C.H. Papadimitriou. Exploring an unknown graph. *J. Graph Theory* **32**(3) (1999) 265–297.

[14] S. Dobrev, J. Jansson, K. Sadakane, W.K. Sung. Finding short right-hand-on-the-wall walks in graphs. *Proc. 12th Colloquium on Structural Information and Communication Complexity (SIROCCO)*. Volume LNCS 3499. (2005) 127–139.

[15] R. Fleischer, G. Trippen. Exploring an unknown graph efficiently. *Proc. 13th Annual European Symposium on Algorithms (ESA)*. (2005) 11–22.

[16] P. Fraigniaud, D. Ilcinkas, G. Peer, A. Pelc, D. Peleg. Graph exploration by a finite automaton. *Theoretical Computer Science* **345**(2-3) (2005) 331-344.

[17] L. Gąsieniec, R. Klasing, R. Martin, A. Navarra, X. Zhang. Fast periodic graph exploration with constant memory. *J Computer and System Science*, **74**(5) (2008) 808–822.

[18] D. Ilcinkas. Setting port numbers for fast graph exploration. *Theoretical Computer Science* **401** (2008) 236–242.

[19] H. Rollik, Automaten in planaren graphen. Acta Informatica **13** (1980) 287–298.

[20] N. Roo, S. Hareti, W. Shi, S. Iyengar. Robot navigation in unknown terrains: Introductory survey of length, non-heuristic algorithms. *Techincal Report ORNL/TM12410* Oak Ridge National Lab (1993).

[21] U. Friege. A tight upper bound on the cover time for random walks on graphs. *Random Structures and Algorithms*, **6**(1) 51–54 (1995).

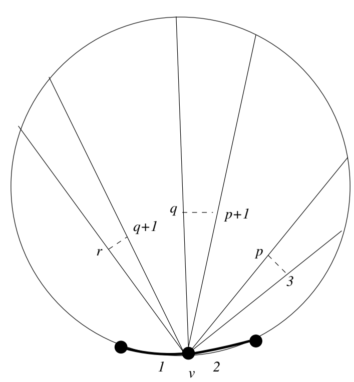

Figure 9: The incoming and outgoing edges of v are isolated real arcs.

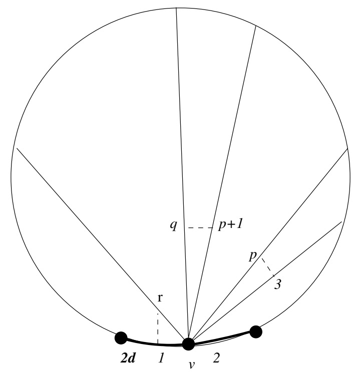

Figure 10: The incoming and outgoing edges of v are arcs, the outgoing edge is isolated.

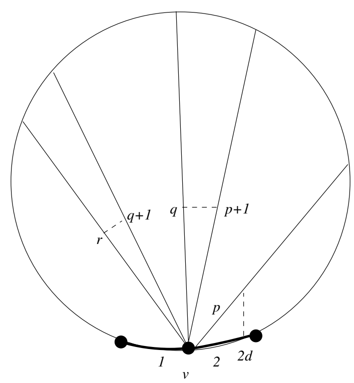

Figure 11: The incoming and outgoing edges of v are arcs, the incoming edge is isolated.

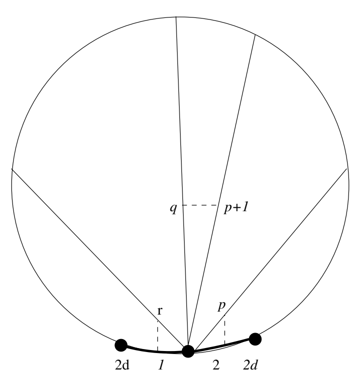

Figure 12: The incoming and outgoing edges of v are arcs, and both are ribs of distinct tied wedges

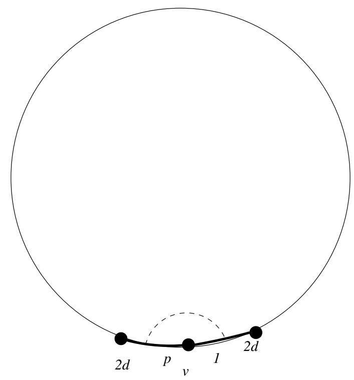

Figure 13: The incoming and outgoing edges of v are arcs, and both are ribs of the same tied wedge

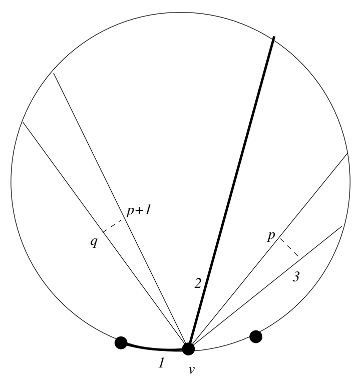

Figure 14: The incoming edge of v is an isolated arc, the outgoing edge of v is an isolated edge

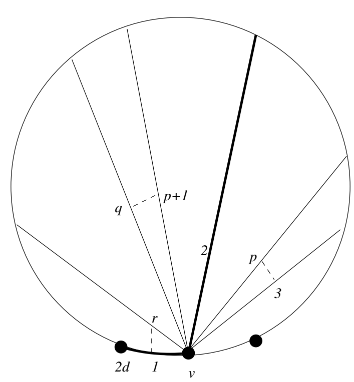

Figure 15: The incoming edge of v is an arc and a rib of a tied wedge, the outgoing edge of v is an isolated edge

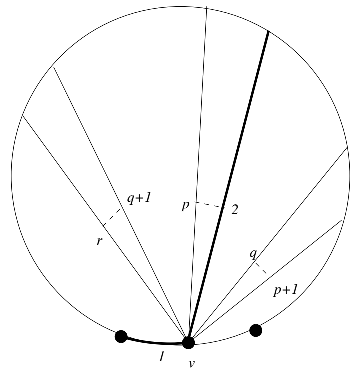

Figure 16: The incoming edge of v is an isolated arc, the outgoing edge of v an external rib of a free wedge

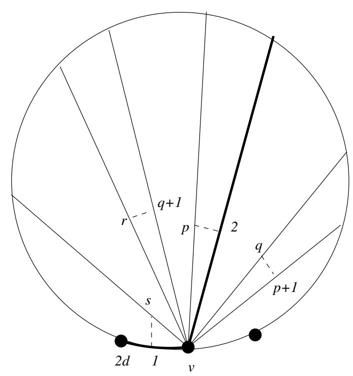

Figure 17: The incoming edge of v is an arc and a rib of a tied wedge, the outgoing edge of v an external rib of a free wedge

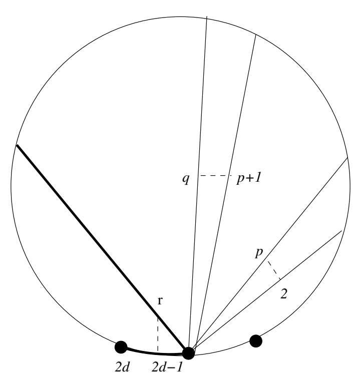

Figure 18: The incoming and outgoing edges of v are both external ribs of the same right tied wedge

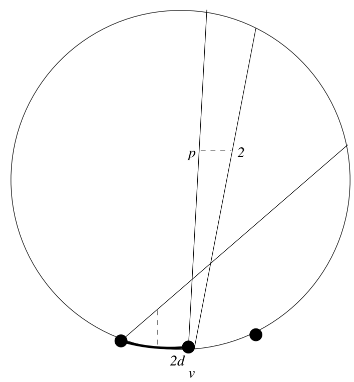

Figure 19: The backtrack edge of v is an arc and an external rib of a left tied wedge

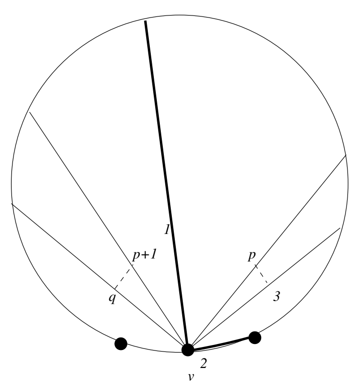

Figure 20: The incoming edge of v is an isolated edge, the outgoing edge of v is an isolated arc

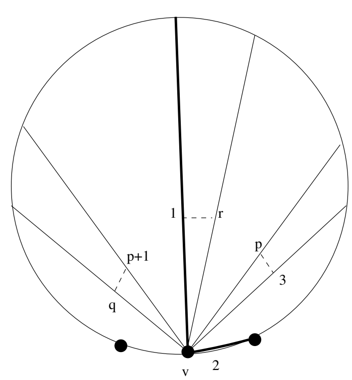

Figure 21: The incoming edge of v is an external rib of a free wedge, the outgoing edge of v is an isolated arc

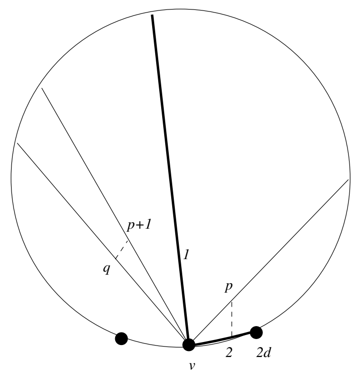

Figure 22: The incoming edge of v is an isolated edge, the outgoing edge of v is an arc and an external rib of a left tied wedge

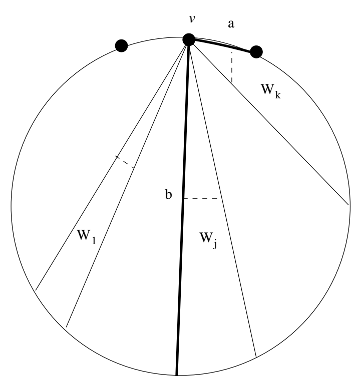

Figure 23: The incoming edge of v is an isolated edge, the outgoing edge of v is an arc and an external rib of a left tied wedge

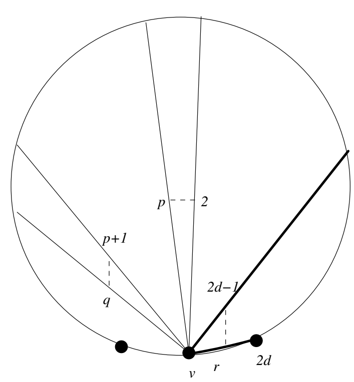

Figure 24: The incoming and outgoing edges of v are both external ribs of the same left tied wedge

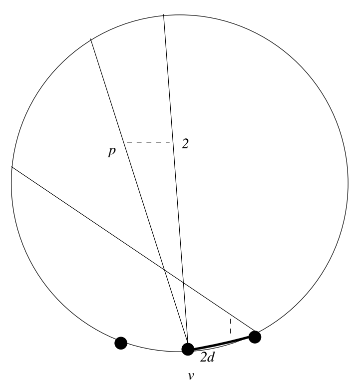

Figure 25: The backtrack edge of v is an arc and an external rib of a right tied wedge

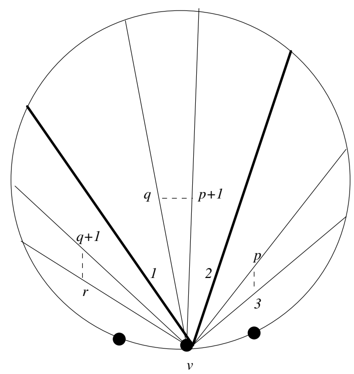

Figure 26: The incoming and outgoing edges of v are both isolated edges

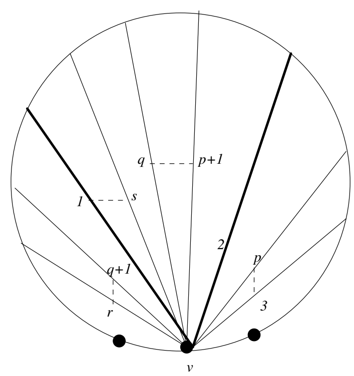

Figure 27: The incoming edge of v is the right external rib of a free wedge, the outgoing edge of v is an isolated edge

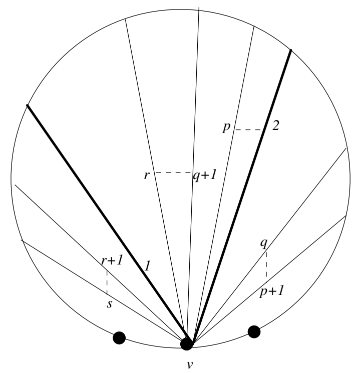

Figure 28: The incoming edge of v is an isolated edge, the outgoing edge of v is the left external rib of a free wedge

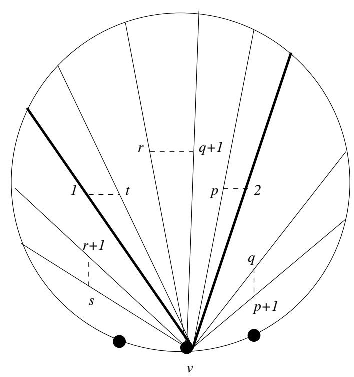

Figure 29: The incoming and outgoing edges of v are both external ribs of distinct free wedges

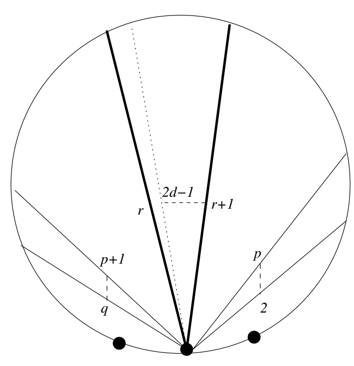

Figure 30: The incoming and outgoing edges of v are both external ribs of the same free wedge

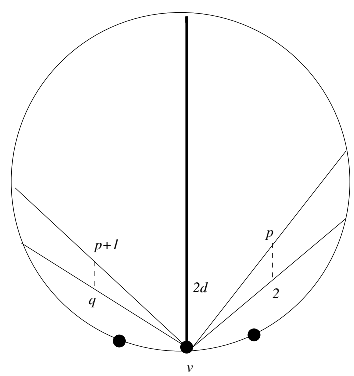

Figure 31: The backtrack edge of v is an internal rib of a wedge