Journal of Graph Algorithms and Applications
http://jgaa.info/ vol. 27, no. 4, pp. 281–327 (2023)
DOI: 10.7155/jgaa.00623

1-Complex s, t Hamiltonian Paths: Structure and Reconfiguration in Rectangular Grids

Rahnuma Islam Nishat[1] *Venkatesh Srinivasan*[2] *Sue Whitesides*[2]

[1]Department of Computer Science, Mathematics, Physics and Statistics,
University of British Columbia, Kelowna, BC, Canada
[2]Department of Computer Science, University of Victoria, Victoria, BC, Canada.

Submitted:	July 2022	Reviewed:	November 2022	Revised:	February 2023
Reviewed:	April 2023	Revised:	May 2023	Accepted:	May 2023
Final:	May 2023	Published:	May 2023	Article type:	Regular paper

Communicated by: M.S. Rahman, P. Mutzel, Slamin

Abstract. We give a complete structure theorem for 1-complex s, t Hamiltonian paths in rectangular grid graphs. We use the structure theorem to design an algorithm to reconfigure one such path into any other in linear time, making a linear number of *switch* operations in grid cells.

1 Introduction

Let $\mathbb{G}$ be an $m \times n$ *rectangular grid graph*, which is an induced, embedded subgraph of the infinite integer grid and has m rows and n columns in an $(m-1) \times (n-1)$ rectangle $R_{\mathbb{G}}$. Let s and t be the top left and bottom right corners of $R_{\mathbb{G}}$.

Definition 1 (1-complex path) *A* 1-complex path *P is an s, t Hamiltonian path (s and t are the endpoints of the path) of $\mathbb{G}$, where each vertex of $\mathbb{G}$ can be connected to a vertex on one of the sides $\mathcal{E}, \mathcal{W}, \mathcal{N}, \mathcal{S}$ of $\mathcal{R}_{\mathbb{G}}$ by a straight line segment consisting of grid edges on the path, as in Fig. 1(a). Since we regard $\mathbb{G}$ as an embedded graph any Hamiltonian path can only travel along grid edges.*

Special Issue on the 16th Int. Conference and Workshops on Algorithms and Computation, WALCOM 2022

A preliminary version of this paper appeared in WALCOM 2022 [35]. Some figures used here are taken from our previous work on simple s, t Hamiltonian paths that appeared in IWOCA 2021 [34].

E-mail addresses: rahnuma.nishat@ubc.ca (Rahnuma Islam Nishat) srinivas@uvic.ca (Venkatesh Srinivasan) sue@uvic.ca (Sue Whitesides)

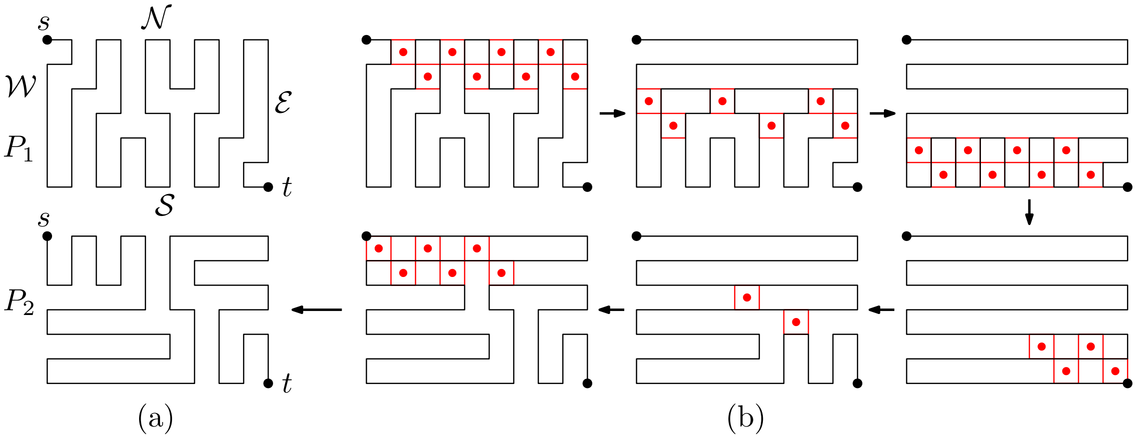


Figure 1: (a) top and bottom: Two 1-complex s,t Hamiltonian paths P_1 and P_2. The sides of $R_{\mathbb{G}}$ are $\mathcal{N}, \mathcal{S}, \mathcal{W}$, and $\mathcal{E}$. (b) clockwise from upper left: A sequence of *switches* (shown by red dots) taking P_1 to P_2.

The term "1-complex" comes from k-*complex paths*, which are discussed in Section 7. In this paper, we give algorithms to reconfigure 1-complex paths using *switches* in 1×1 grid *cells*.

Suppose $\mathbb{G}$ has a *cover* of its vertices consisting of one s,t path (not necessarily Hamiltonian) together with 0 or more cycles, all disjoint from each other. (For example, initially $\mathbb{G}$ is covered by an s,t Hamiltonian path and no cycles.) If a grid cell has two parallel grid edges that belong to such a cover and two parallel grid edges that do not, we call it a *switchable cell* with respect to the cover; a *switch* exchanges the edges in that switchable cell that belong to the cover for the two edges not in the cover (see Figs. 1(b) and 24). A switch may change the edges of the cover incident to a particular vertex but not the degree of that vertex (each vertex has degree 2 except for s and t, which have degree 1). Thus a switch produces a new cover comprising one s,t path and 0 or more cycles, all disjoint from one another, and it may be viewed as an operation on the family of such covers. Whether a particular grid cell may be switched depends on the cover.

The question we ask is this: given any two 1-complex paths of $\mathbb{G}$, can one of them be reconfigured to the other with only $O(|\mathbb{G}|)$ switches in grid cells, and if so, can the sequence of switches to be performed be computed efficiently? When the sequence of switches exists, then the intermediate results remain in the family of covers having one s,t path and 0 or more cycles, all disjoint. See Fig. 2(a) for an example where one switch creates a cover consisting of an s,t path and a cycle; Fig. 2(b) shows the next switch operation, which joins the cycle and the path to give an s,t Hamiltonian path.

As shown in Fig. 3, a "cross-separator" of $P_{s,t}$ (i.e., a subpath η_i joining vertices on opposite sides of $R_{\mathbb{G}}$) may have many forms, depending on whether and where this subpath has bends. In previous work [34] we introduced a special case of 1-complex s,t Hamiltonian paths we called "simple". By definition, a simple s,t path has only "straight" cross-separators, so none of the subpaths in Fig. 3 that contain "bent" cross-separators can occur in simple paths.

A key contribution of this work is a complete characterization of 1-complex s,t Hamiltonian paths in rectangular grid graphs, building on our work in [34] which considered only the very special case of simple paths. Using our characterization (structure theorem) for 1-complex paths,

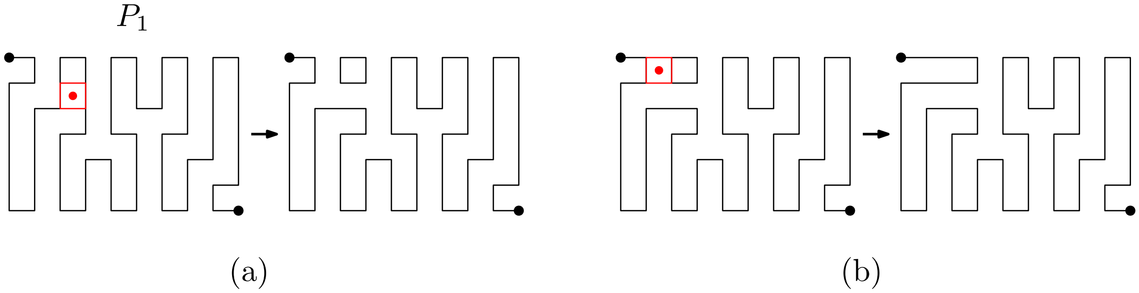


Figure 2: The first two switches in Figure 1(b).

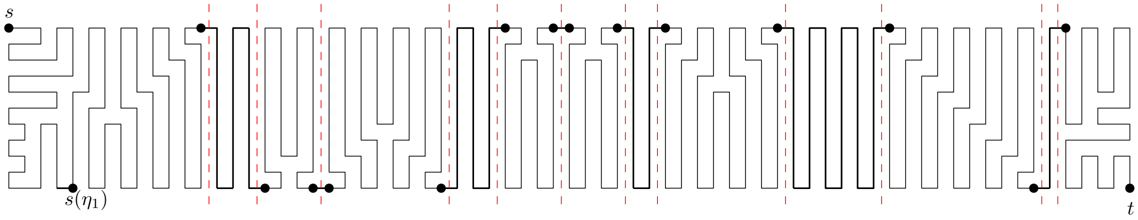


Figure 3: The structure of the s, t path $P_{s,t}$ is used to break $\mathcal{R}_{\mathbb{G}}$ into sub-rectangles.

we achieve a linear-time algorithm for reconfiguration of 1-complex s, t Hamiltonian paths $P_{s,t}$. Roughly, our algorithm uses the structure of a $P_{s,t}$ to define smaller sub-rectangles within $R_{\mathbb{G}}$. Path $P_{s,t}$ determines s', t' Hamiltonian paths $p'_{s',t'}$ within each sub-rectangle. We define the sub-rectangles so that our structure theorem as well as our new reconfiguration tools (Section 4) apply to each path $p'_{s',t'}$. See Fig. 3 and Section 5.

1.1 Related work

Reconfiguration of Hamiltonian cycles and Hamiltonian paths has attracted attention in recent years. Takaoka [47] has shown that for some unembedded graph classes, deciding whether there is a sequence of "switch" operations between two given Hamiltonian cycles is a PSPACE-complete problem. An example of that author's switch operation is shown in Fig. 4. The same switch operation was used by Lignos [30]; he showed that that it can be decided in linear time whether a Hamiltonian cycle can be reconfigured to another in a graph of maximum degree 5.

Nishat and Whitesides studied reconfiguration of Hamiltonian cycles of "bend complexity 1" in grid graphs without holes [36, 37]; an application of their work in the context of 3D printing was investigated by Bedel et al. [5], where they use reconfiguration to optimize the "toolpath" of 3D printers under orientation objectives. Nishat defined 1-complex Hamiltonian paths in her doctoral dissertation [33]. We studied the structure and reconfiguration of "simple" s, t Hamiltonian paths, which constitute a subclass of 1-complex s, t Hamiltonian paths, in [34]. To the best of our knowledge, this work initiates the study of characterization of Hamiltonian paths in embedded graphs. In this paper, we use the characterization to give reconfiguration algorithms; we also give a foundation for further research in reconfiguring Hamiltonian paths and cycles.

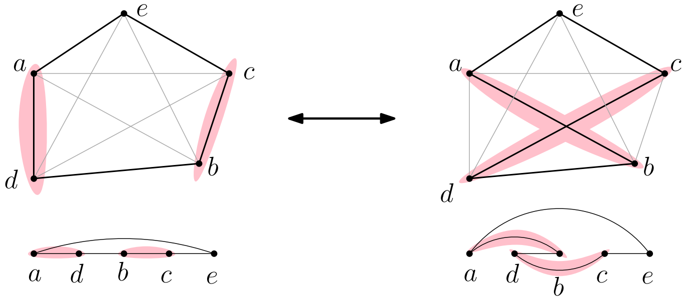


Figure 4: A switch operation on a Hamiltonian cycle of K_5: the edges (a, d) and (b, c) are replaced with edges (a, b) and (c, d), where the two latter edges are not on the original Hamiltonian cycle. This produces a new Hamiltonian cycle for K_5.

Although reconfiguration of Hamiltonian paths and cycles is a recent topic in the literature, the complexity and combinatorial aspects of the problem of finding Hamiltonian paths and cycles have been studied for decades. Itai et al. initiated the study of Hamiltonian paths and cycles in grid graphs [22]; related problems have been extensively studied over the years [52, 48, 49, 13, 12, 8, 28, 10, 23, 24, 41, 15, 46]. Hamiltonian paths and cycles in triangular and hexagonal grids, as well as dimensions higher than 2, have also been explored [42, 18, 21, 4]. Researchers have studied the structural properties of Hamiltonian cycles and paths in grid graphs for the purpose of counting and enumerating such paths and cycles [27, 26, 25, 1, 29, 43, 11].

Our work was initially inspired by reconfiguration of plane triangulations using 'flips' [7]. Călinescu et al. studied reconfiguration of vertex-sets in (infinite) embedded grid graphs [9]. Many graph problems have been studied for graphs embedded on the plane, including grid graphs, in computational geometry (e.g., Chapter 10 in [16], Chapter 15 in [44]) and in graph drawing [40, 39]. The 'sliding token' puzzle is an example of reconfiguration of objects on the plane [2, 20, 51, 19]. For more information on reconfiguration problems, we refer the reader to the survey paper by Nishimura [38].

We note that studying and understanding the structure of an entity has been crucial in obtaining enumeration or algorithmic results in many research areas. For example, in graph drawing, Schnyder [45] showed that any maximal planar graph can be decomposed into three "Schnyder trees", and gave a linear-time algorithm to obtain a planar straight-line embedding of such a graph using the Schnyder trees. Understanding structure of an embedded Hamiltonian cycle in a grid graph have led to describing the Hamiltonian cycle as a sequence of integers [29], which in turn have helped to give generating functions for such cycles [11]. Nishat [32] has used the structure of a "crease pattern" of a map to be folded to give an algorithm that recognizes whether a given ordering of the unit squares of the map is feasible or not.

1.2 Our contributions

In this paper, we advance knowledge on structure and reconfiguration of Hamiltonian paths, motivated by the many applications of both Hamiltonian cycles and Hamiltonian paths in grid graphs (see e.g. [31], [17], [6], [14], [5], [3], [50]). We aim not only to study the problem, but also to lay out the foundations for future research by making the following contributions:

1. a complete structure for 1-complex s, t Hamiltonian paths (Section 3);

2. two powerful new reconfiguration tools that find cells to switch and then sequence the switches to create straight path segments (Section 4); and

3. an algorithm to reconfigure any 1-complex Hamiltonian path to any other such path in $O(mn)$ time, making $O(mn)$ switches in 1×1 grid *cells*, where mn is the number of vertices of the grid graph $\mathbb{G}$, and therefore, the number of vertices of the input and target 1-complex paths. It serves as a measure of the input size (Sections 5 and 6). The time required for each switch is constant.

2 Preliminaries

We introduce terminology used throughout the paper, and state some properties of 1-complex s, t Hamiltonian paths in grid graphs.

Fig. 5 demonstrates the basic terminology related to grid graph $\mathbb{G}$. A vertex of $\mathbb{G}$ with coordinates (x, y) is denoted by $v_{x,y}$, where $0 \leq x \leq n-1$ and $0 \leq y \leq m-1$. The top left corner vertex s of $\mathbb{G}$ is $(0, 0)$; the positive x-direction is rightward and the positive y-direction is downward.

Assumption. $\mathbb{G}$ is an $m \times n$ grid graph, where $m, n \geq 4$, and α and β are the bottom left and top right corner vertices of $\mathbb{G}$. Without loss of generality, *we assume the input 1-complex path $P_{s,t}$ visits α before β* as shown in Fig. 6, as otherwise, we can assume that the positive x-direction is downward and the positive y-direction is rightward, thus interchanging the x and y coordinates. The target 1-complex path for the reconfiguration as well as intermediate configurations may visit β before α.

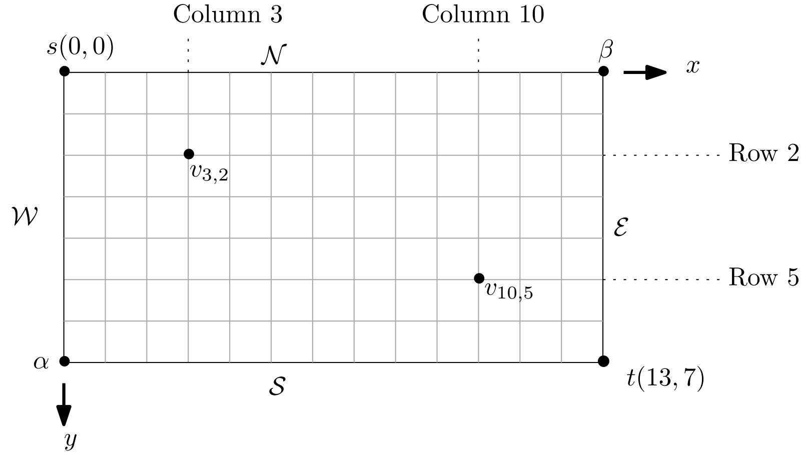


Figure 5: An 8×14 rectangular grid graph.

Column x of $\mathbb{G}$ is the shortest path of $\mathbb{G}$ between $v_{x,0}$ and $v_{x,m-1}$, and *Row y* is the shortest path between $v_{0,y}$ and $v_{n-1,y}$. We call Columns 0 and $n-1$ the *west* ($\mathcal{W}$) and *east* ($\mathcal{E}$) boundaries of $\mathbb{G}$ (i.e., sides of $R_{\mathbb{G}}$), respectively, and Rows 0 and $m-1$ the *north* ($\mathcal{N}$) and *south* ($\mathcal{S}$) boundaries (sides). An *internal vertex* of $\mathbb{G}$ is a vertex that is not on any boundary of $\mathbb{G}$.

Throughout this paper, a *1-complex path* P means a 1-complex s, t Hamiltonian path on $\mathbb{G}$; P visits each vertex of $\mathbb{G}$ exactly once and uses only edges in $\mathbb{G}$. We denote by $P_{u,v}$ the directed subpath of P from vertex u to vertex v. Straight subpaths, i.e., subpaths without bends, of P are called *segments*, denoted $seg[u, v]$, where u and v are the segment endpoints. An edge (u, v) of $\mathbb{G}$ that lies on P may be regarded as a segment of P and denoted $seg[u, v]$.

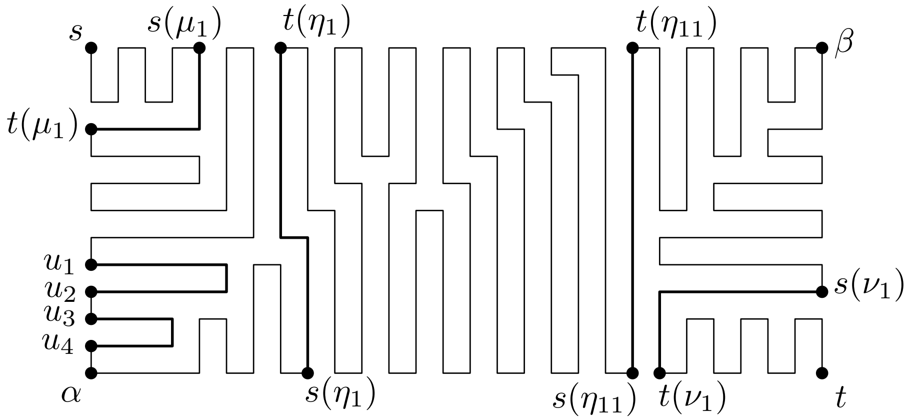


Figure 6: A 1-complex s, t path $P_{s,t}$ in a 13×28 grid graph.

Definition 2 (Cookie, cross-separator and corner separator) *Every internal vertex v of $\mathbb{G}$ lies on a* maximal internal subpath *of P, namely the subpath through internal vertices of $\mathbb{G}$ that has endpoints v_s and v_t, where v_s is the first boundary vertex met when travelling along P from v towards s, and v_t is the first boundary vertex met when travelling along P from v towards t. Such a maximal internal subpath is called a* cookie *if v_s and v_t lie on the same boundary, a* corner separator *if they lie on adjacent boundaries (e.g., $\mathcal{W}$–$\mathcal{N}$, $\mathcal{E}$–$\mathcal{S}$ etc.), and a* cross-separator *if they lie on opposite boundaries (i.e., either $\mathcal{E}$–$\mathcal{W}$ or $\mathcal{N}$–$\mathcal{S}$). Therefore, every maximal internal subpath of P is either a cookie, or a corner separator or a cross-separator.*

In the 1-complex path in Fig. 6, the maximal internal subpaths P_{u_1,u_2} and P_{u_3,u_4} are two cookies from the $\mathcal{W}$ boundary, the subpaths $P_{s(\mu_1),t(\mu_1)}$ and $P_{s(\nu_1),t(\nu_1)}$ are two corner separators, and the subpaths $P_{s(\eta_1),t(\eta_1)}$ and $P_{s(\eta_{11}),t(\eta_{11})}$ are cross-separators.

A cross-separator can be *straight* ($P_{s(\eta_{11}),t(\eta_{11})}$ in Fig. 6), or it can have two adjacent bends ($P_{s(\eta_1),t(\eta_1)}$ in Fig. 6); but a cross-separator cannot have an odd number of bends as its endpoints are on opposite boundaries. The possibility of a cross-separator with two adjacent bends can arise because P is 1-complex, so each bend can be connected by a straight segment to a boundary. Since the bends are adjacent vertices of $\mathbb{G}$, all the other internal vertices of the cross-separator are connected to a boundary by a segment of P as well. Indeed, if a cross separator has two bends they must be adjacent in $\mathbb{G}$ as otherwise, the vertices between them could not reach a boundary of $\mathbb{G}$ by a straight segment of P. We call a cross-separator with two adjacent bends a *bent* cross-separator.

A cookie c is one of four types, $\mathcal{N}$, $\mathcal{S}$, $\mathcal{E}$, and $\mathcal{W}$, depending on the boundary where c has its *base*, i.e., the grid edge that connects the endpoints of c. See Figure 6. A cookie consists of three segments of P; the common length of the two parallel segments of c measures the *size* sz of c. The third segment of c is composed of a single edge, thus, c covers an $sz \times 1$ rectangular region on the plane.

We say a corner separator *cuts off* a corner of $\mathbb{G}$. In Fig. 6, $P_{s(\mu_1),t(\mu_1)}$ cuts off the corner s of $\mathbb{G}$, and $P_{s(\nu_1),t(\nu_1)}$ cuts off the corner t of $\mathbb{G}$.

From now on, we assume that a 1-complex path $P_{s,t}$ has $j \geq 0$ corner separators cutting off s, and $k \geq 1$ cross-separators, and $\ell \geq 0$ corner separators cutting off t. Traveling along $P_{s,t}$, we denote the i-th corner separator cutting off s by μ_i, where $0 \leq i \leq j$. We denote its internal bend by $b(\mu_i)$, and its endpoints by $s(\mu_i)$ and $t(\mu_i)$, where $s(\mu_i)$ is the first endpoint met along $P_{s,t}$.

Similarly, we denote the i-th corner separator cutting off t by ν_i; endpoint $s(\nu_i)$ is met along $P_{s,t}$ before $t(\nu_i)$, with internal bend at $b(\nu_i)$, where $0 \leq i \leq \ell$. A corner separator that has one of its endpoints connected to s or t by a segment of P is called a *corner cookie*. The path in Fig. 6 has $j = 2$ corner separators cutting off s, and $k = 11$ cross-separators, and $\ell = 1$ corner separator cutting off t. Note that a corner separator has exactly one bend.

Path P can have at most two corner cookies, one at either end of the path.

Similar to the corner separators, we denote the i-th cross-separator met along $P_{s,t}$ by η_i where $0 \leq i \leq k$. We denote its endpoints by $s(\eta_i)$ and $t(\eta_i)$, where $s(\eta_i)$ is the first endpoint met. If η_i is bent, then $b(\eta_i)$ denotes the first bend met. In total, we have j corner separators μ_i cutting off s, and k cross-separators η_i, and ℓ corner separators ν_i cutting off t. The path in Fig. 6 has $j = 2$ corner separators cutting off s, and $k = 11$ cross-separators, and $\ell = 1$ corner separator cutting off t.

Definition 3 (Runs of Cookies) *A* run of cookies *is a subpath of P consisting of cookies of the same type, spaced one unit apart and joined by the single boundary edges between them, possibly extended at either end by an edge joining a cookie endpoint to an adjacent boundary vertex.*

A run of cookies is denoted $Run[u, v]$, where u and v belong to the same boundary and delimit the range of boundary vertices covered; $Run[u, v]$ may consist of a single boundary edge (u, v). The subpath P_{u_1,u_4} in Fig. 6 is a run of two $\mathcal{W}$ cookies and is denoted by $Run[u_1, u_4]$.

To describe the path structure, we define three types of runs, depending on the cookie sizes along the run: the sizes may remain the same, or be non-increasing (denoted $Run^{\geq}[u, v]$) or non-decreasing ($Run^{\leq}[u, v]$). Runs are assumed to have cookies of the same size unless specified otherwise.

Definition 4 (Canonical Paths) *A* canonical path $\mathbb{P}$ *is a 1-complex path with no bends at internal vertices. If m is odd, $\mathbb{P}$ can be of type $\mathcal{W}$-$\mathcal{E}$, filling rows of $\mathbb{G}$ one by one; if n is odd, $\mathbb{P}$ can be of type $\mathcal{N}$-$\mathcal{S}$, filling columns one by one. Fig. 7 shows example of canonical paths.There are no other types.*

We now observe some properties of path $P_{s,t}$ that we use in the next sections.

Lemma 1 *Let $P_{s,t}$ be a 1-complex Hamiltonian path from s to t, and let $P_{s,v}$ be an initial subpath of $P_{s,t}$. Then removal of $P_{s,v}$ from $\mathbb{G}$ leaves exactly one non-empty component of $\mathbb{G} \setminus P_{s,v}$.*

Proof: Suppose for a contradiction that $\mathbb{G} \setminus P_{s,v}$ has two or more components. Then $P_{s,v}$ leaves at least two vertices uncovered. Hence vertex t cannot be equal to v, nor can t lie on $P_{s,v}$. Let $\mathbb{G}^t$ be the component of $\mathbb{G} \setminus P_{s,v}$ containing t, and let w be a vertex of a component of $\mathbb{G} \setminus P_{s,v}$ that is distinct from $\mathbb{G}^t$. Then the subpath $P_{w,t}$ of $P_{s,t}$ crosses the initial subpath $P_{s,v}$, contradicting that $P_{s,t}$ is 1-complex and by definition, non-crossing. Thus the statement of the lemma holds. □

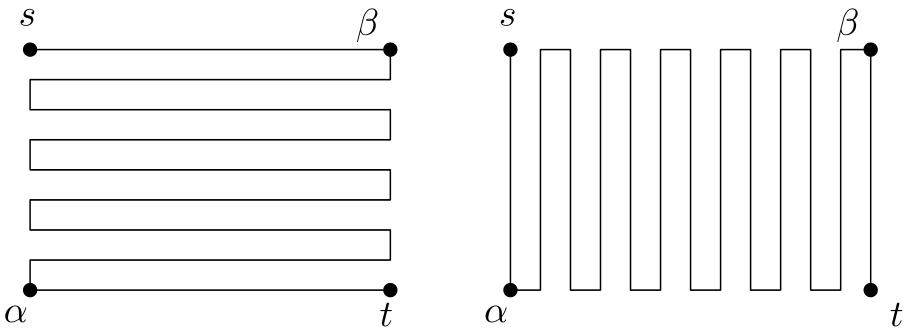


Figure 7: Left: an $\mathcal{W}$-$\mathcal{E}$ canonical path. Right: an $\mathcal{N}$-$\mathcal{S}$ canonical path.

The lemma above finds immediate application in the proof of the next lemma, which gives some basic properties of $P_{s,t}$ used in later sections. An observation similar to this lemma in the more restricted setting of "simple" s, t paths was stated without proof in our conference paper [34].

Lemma 2 (Property Lemma) *Let $P_{s,t}$ be a 1-complex Hamiltonian path of $\mathbb{G}$ that visits α before β. Then $P_{s,t}$ satisfies the following properties:*

i) $P_{s,t}$ visits α before any other vertex of $\mathcal{S}$; vertex β is the last vertex of $\mathcal{N}$ that $P_{s,t}$ visits;

ii) the subpath of $P_{s,t}$ from s to α, i.e., $P_{s,\alpha}$, covers all vertices of $\mathcal{W}$; the subpath of $P_{s,t}$ from β to t, i.e., $P_{\beta,t}$, covers all the vertices of $\mathcal{E}$;

iii) no corner separators cut off α or β;

iv) any corner separators cutting off s occur in the subpath $P_{s,\alpha}$ of $P_{s,t}$; any corner separators cutting off t occur in the subpath $P_{\beta,t}$ of $P_{s,t}$; and all cross-separators occur in the subpath $P_{\alpha,\beta}$ of $P_{s,t}$;

v) the number k of cross-separators η_i is odd. The first and last cross-separators η_1 and η_k travel from $\mathcal{S}$ to $\mathcal{N}$;

vi) a cross-separator that travels from $\mathcal{S}$ to $\mathcal{N}$ is immediately preceded on $P_{s,t}$ by an edge on $\mathcal{S}$ that lies on the west side of the cross-separator and is immediately succeeded on $P_{s,t}$ by an edge on $\mathcal{N}$ that lies on the east side of the cross-separator; a cross-separator that travels from $\mathcal{N}$ to $\mathcal{S}$ is immediately preceded on $P_{s,t}$ by an edge of $\mathcal{N}$ that lies on the west side of the separator and is immediately succeeded on $P_{s,t}$ by an edge of $\mathcal{S}$ that lies on the east side of the separator;

vii) when $1 < i \leq k$, each separator η_i lies east of its predecessor η_{i-1}.

Proof:

i) Suppose $P_{s,t}$ visits vertex v on $\mathcal{S}$ before α. Then α and t belong to distinct components of $\mathbb{G} \setminus P_{s,v}$, contradicting Lemma 1. Thus $P_{s,t}$ reaches α before any other vertex of $\mathcal{S}$. Now consider the 1-complex Hamiltonian path $P_{t,s}$. By the same reasoning, it visits β before any other vertex of $\mathcal{N}$. Hence $P_{s,t}$ visits all other vertices of $\mathcal{N}$ before it reaches β.

ii) The statement holds as otherwise, the path $P_{s,\alpha}$ would contradict Lemma 1. By considering the initial path $P_{t,\beta}$ of the reverse path $P_{t,s}$, it follows that the final subpath $P_{\beta,t}$ of $P_{s,t}$ covers all the vertices of $\mathcal{E}$.

iii) Suppose for a contradiction that $P_{s,t}$ contains a subpath $P_{u,v}$ that is a corner separator cutting off α or β, where u occurs before v on $P_{s,t}$. Then the initial subpath $P_{s,v}$ contradicts Lemma 1. (See Fig. 8(a) and (b).)

iv) Any corner separator μ_i cutting off s has one endpoint v_W on $\mathcal{W}$ and one endpoint v_N on $\mathcal{N}$. By part ii) above, v_W lies in $P_{s,\alpha}$. We claim that the other endpoint v_N of μ_i also lies on $P_{s,\alpha}$. If v_N occurs before v_W on $P_{s,t}$, the claim is true. If v_N occurs after v_W, then $P_{s,t}$ contains horizontal $seg[v_W, b(\mu_i)]$ and vertical segment $seg[b(\mu_i), v_N]$, neither of which contains α. Therefore the claim is true in this case as well.

By property i) above, all starting endpoints of cross-separators that travel from $\mathcal{S}$ to $\mathcal{N}$ occur after α on $P_{s,t}$, and their terminal endpoints on $\mathcal{N}$ occur before β on $P_{s,t}$. Therefore the cross-separators that travel from $\mathcal{S}$ to $\mathcal{N}$ occur on $P_{s,t}$ between α and β. By considering the reverse path $P_{t,s}$ and using similar reasoning, it follows that all cross-separators that travel from $\mathcal{N}$ to $\mathcal{S}$ occur on $P_{s,t}$ between α and β.

v) By assumption, $P_{s,t}$ reaches α before β. After $P_{s,t}$ reaches α, it must return to $\mathcal{N}$ in order to reach β. Since $P_{s,\alpha}$ covers $\mathcal{W}$ and $P_{\beta,t}$ covers $\mathcal{E}$, path $P_{\alpha,\beta}$ must contain a cross-separator to return to $\mathcal{N}$. Thus the first cross-separator η_1 travels from $\mathcal{S}$ to $\mathcal{N}$. After η_1, path $P_{\alpha,\beta}$ may possibly make round trips from $\mathcal{N}$ to $\mathcal{S}$ and back to $\mathcal{N}$. To reach β, the last cross-separator η_k must travel from $\mathcal{S}$ to $\mathcal{N}$. Therefore, $k \geq 1$ is odd. Recall that the η_i are indexed by the order in which they are met along $P_{s,t}$. If η_{i+1} were west of η_i, then $P_{s,t}$ would cross η_i to reach t, i.e., $P_{s,t(\eta_1)}$ contradicts Lemma 1.

vi) Let η_i be a cross-separator that travels from $\mathcal{S}$ to $\mathcal{N}$ such that the start vertex $s(\eta_i)$ is immediately preceded on $P_{s,t}$ by vertex v, where v lies one unit east of $s(\eta_i)$ on $\mathcal{S}$. Then the initial path $P_{s,v}$ of $P_{s,t}$ contradicts Lemma 1. The remainder of the statement of this item follows by similar reasoning.

vii) If some η_i lay west of its predecessor η_{i-1} then the subpath $P_{s,t(\eta_{i-1})}$ would contradict Lemma 1 as t and η_i would lie in separate components of $\mathbb{G} \setminus P_{s,t(\eta_{i-1})}$.

This completes the proof. □

3 Structure of 1-Complex Paths

Let $P_{s,t}$ be a 1-complex Hamiltonian path for a rectangular grid graph $\mathbb{G}$ that visits α before β, where η_1 and η_k are the first and last cross-separators of $P_{s,t}$. We regard P in its directed form $P_{s,t}$ as composed of an *initial* subpath $P_{s,s(\eta_1)}$, followed by a *middle* subpath $P_{s(\eta_1),t(\eta_k)}$, and then a *final* subpath $P_{t(\eta_k),t}$. By reversing the edge directions, the final subpath of $P_{s,t}$ can be viewed as the initial subpath of the Hamiltonian path $P_{t,s}$ from t to s. Then the grid $\mathbb{G}$ can be rotated by π to place t in the upper left corner and s in the lower right corner. Thus, apart from changes in notation, the structural possibilities for the final and initial subpaths of $P_{s,t}$ are the same, and hence we do not discuss the final subpath in any further detail. We describe the structure of 1-complex paths by specifying the structure of their initial subpaths in Section 3.1 and their middle subpaths in Section 3.2.

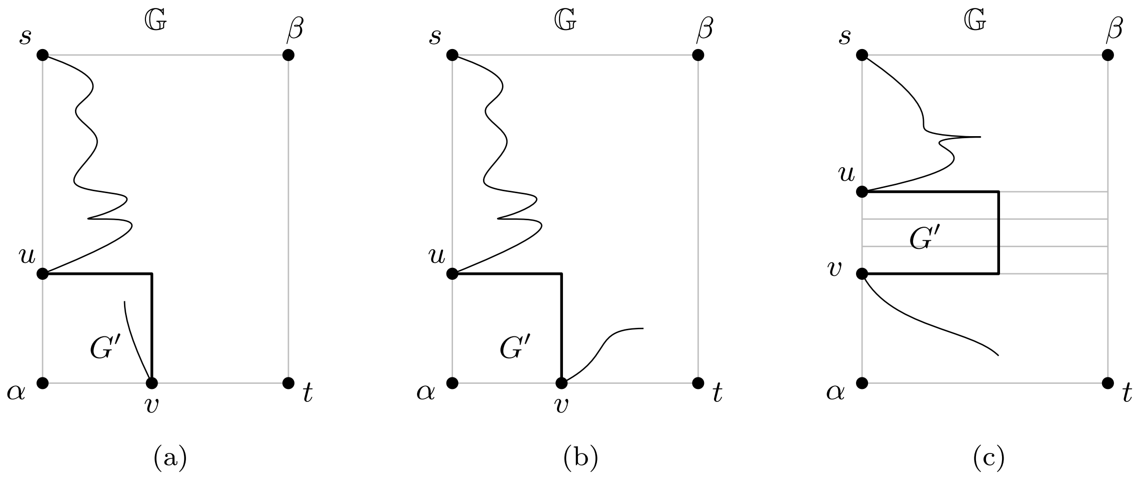


Figure 8: A path $p_{u,v}$ of $\mathbb{G}$ that cannot belong to $P_{s,t}$: (a) under the assumption that the vertex after v in $P_{s,t}$ is inside G', and (b) when it is outside G'. (c) A cookie must have endpoints adjacent on a boundary as otherwise, there are uncovered vertices inside the cookie or $P_{s,t}$ crosses itself.

3.1 Initial subpath

The structure of the initial subpath depends on the form of the first cross-separator η_1 of $P_{s,t}$. By Lemma 2(v) and our assumption that $P_{s,t}$ visits α before β, vertex $s(\eta_1)$ lies on $\mathcal{S}$ and vertex $t(\eta_1)$ lies on $\mathcal{N}$. As η_1 may be bent or straight, we identify five possible forms for it (in Fig. 9, see forms A, B, D, F, and G). Note that the labels assigned to the cross-separator forms are arbitrary and are not intended to suggest any alphabetical order.

The first cross-separator η_1 can be straight (form B). If η_1 is not straight, then it may have a bend one unit above $\mathcal{S}$. At such a bend, η_1 may turn left, towards $\mathcal{W}$ (form G), but it cannot turn right, towards $\mathcal{E}$ as in that case, it would not be possible for the path to reach the vertex on $\mathcal{S}$ in the column containing $t(\eta_1)$. Similarly, if η_1 has a bend one unit below $\mathcal{N}$, then this bend must turn towards $\mathcal{W}$ (form D). If the bends of η_1 lie more than one unit from both $\mathcal{S}$ and $\mathcal{N}$, then η_1 may turn right (form A) or left (form F).

When η_1 contains vertices in Column 1, we that η_1 is "near" $\mathcal{W}$. In that case, the initial path $P_{s,s(\eta_1)}$ is determined, as shown in Fig. 10. When

We discuss and give the structure of the initial subpath for each of these five forms in the next three subsections: 3.1.1 (considers B or G, which cover the internal vertices in the column of $t(\eta_1)$), 3.1.2 (considers D, which covers the internal vertices in the column of $s(\eta_1)$), and 3.1.3 (considers A or F, which do not cover all the internal vertices of any column).

3.1.1 Separator η_1 has form B or G.

In the lemmas below, vertex w is the vertex on $\mathcal{N}$ just one column west of $t(\eta_1)$, and a' is the vertex on $\mathcal{S}$ in the same column as w, as shown in Figs. 11(c) and (d). The vertex $v_{0,2}$ is denoted by a.

Lemma 3 (η_1 has form B or G) *(a) If $t(\eta_1)$ is in Column 1 right next to s on the $\mathcal{N}$ boundary, then the initial subpath consists of two boundary segments seg$[s, \alpha]$ on the $\mathcal{W}$ boundary and*

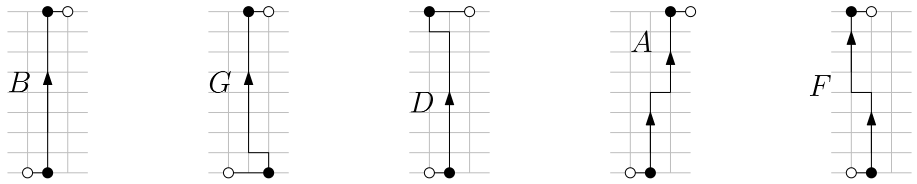


Figure 9: Five forms for η_1. Segments of separators that has length at least two contains arrows.

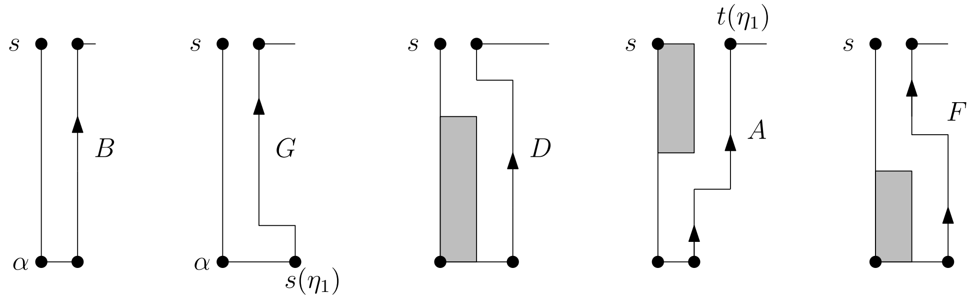


Figure 10: The structure of the initial subpath when η_1 is near $\mathcal{W}$; the grey box denotes a run of unit size $\mathcal{W}$ cookies. Segments of separators that has length at least two contains arrows.

$seg[\alpha, s(\eta_1)]$ on the $\mathcal{S}$ boundary; see Fig. 11(a) and (b).

(b) *If $j = 0$ and $x(t(\eta_1)) > 1$, $P_{s,s(\eta_1)}$ must have a corner cookie containing s and w. This corner cookie is connected by a boundary edge to vertex a on $\mathcal{W}$. The path $P_{a,a'}$ from a to a' takes one of the following four forms: $seg[a, \alpha]$ $Run[\alpha, a']$; or $Run[a, \alpha]$ $seg[\alpha, a']$; or $Run^{\geq}[a, \alpha]$ $seg[\alpha, u]$ $Run^{\leq}[u, a']$, where u is at least two units from α on $\mathcal{S}$; or $Run^{\geq}[a, u']$, $seg[u', \alpha]$ $Run^{\leq}[\alpha, a']$, where u' is at least two units from α on $\mathcal{W}$. The remainder of path $P_{s,s(\eta_1)}$ is the segment $seg[a', s(\eta_1)]$. See Fig. 11(c) for an example when η_1 has form B.*

(c) *If $j \geq 1$, then $P_{s,t(\mu_j)} = Run[s, s(\mu_1)]$ μ_1 $Run[t(\mu_1), s(\mu_2)]$ μ_2 $\ldots$ μ_i $Run[t(\mu_i), s(\mu_{i+1})]$ μ_{i+1} $\ldots$ μ_{j-1} $Run[t(\mu_{j-1}), s(\mu_j)]$ μ_j. Since μ_j ends at $t(\mu_j)$ on $\mathcal{W}$, vertex $s(\mu_1)$ must lie on $\mathcal{N}$ for j odd and on $\mathcal{W}$ for $j \geq 1$ even. Path $P_{t(\mu_j),s(\eta_1)}$ either has the structure of $P_{a,s(\eta_1)}$ in (b), or, as shown in Fig. 11(a)–(b), the path $P_{t(\mu_j),s(\eta_1)}$ consists of edge $(t(\mu_j), \alpha)$ followed by $seg[\alpha, s(\eta_1)]$. See Fig. 11(d) for an example when η_1 has form G.*

Proof:

(a) If $t(\eta_1)$ is in Column 1, then $s(\eta_1)$ must be in Column 1 for form B, and in Column 2 for form G. In either case, the internal vertices in Column 1 are covered by η_1. Therefore, there is no room for any cookies or corner separators in the initial subpath $P_{s,s(\eta_1)}$.

(b) By Lemma 2(ii), $P_{s,\alpha}$ must visit all the $\mathcal{W}$ vertices. Therefore, there cannot be any $\mathcal{N}$ cookies as there is no corner separator to move to the $\mathcal{W}$ boundary from the $\mathcal{N}$ boundary. Then w must be a tip of a corner cookie from the $\mathcal{W}$ boundary. The rest follows from Lemma 2(ii) and (v).

(c) For $j \geq 1$, there must be j endpoints of μ_i on $\mathcal{W}$ and j on $\mathcal{N}$. The endpoint $t(\mu_j)$ must lie on $\mathcal{W}$, not $\mathcal{N}$, and $t(\mu_j)$ must be the corner separator endpoint nearest to α on $\mathcal{W}$. The endpoints

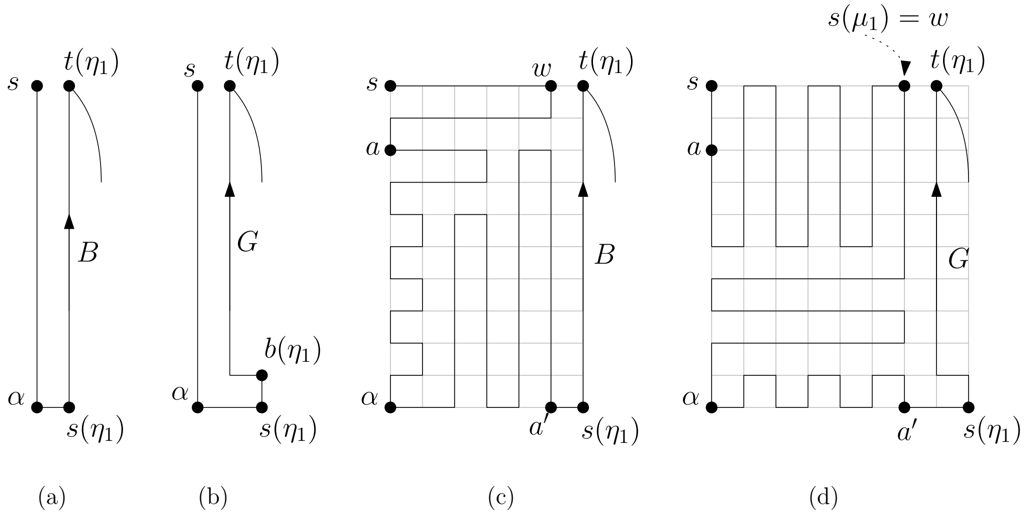


Figure 11: (a) Cross-separator η_1 has form B and is near $\mathcal{W}$, and (b) η_1 has form G and is near $\mathcal{W}$; the initial subpath does not have any cookies as η_1 is near $\mathcal{W}$. (c) Separator η_1 has form B and is far from $\mathcal{W}$, (d) η_1 has form G and is far from $\mathcal{W}$. Segments of separators that has length at least two contains arrows.

of the μ_i's must alternate on $\mathcal{W}$ as shown in Fig. 12(a) for $j \geq 1$ even, and in (d) for j odd. Thus for j odd, the top endpoint on $\mathcal{W}$ is $t(\mu_1)$, so $s(\mu_1)$ is on $\mathcal{N}$; for $j \geq 1$ even, $s(\mu_1)$ is on $\mathcal{W}$. In either case, $P_{s,s(\mu_1)} = Run[s, s(\mu_1)]$ as shown in Fig. 12(b), (c).

Path $P_{s(\mu_1),t(\mu_j)}$ makes round trips (see Fig. 13) between $\mathcal{W}$ and $\mathcal{N}$ via the μ_i. These trips have the form μ_i $Run[t(\mu_i), s(\mu_{i+1})]$ μ_{i+1} and alternate leaving from $\mathcal{N}$ or from $\mathcal{W}$, as the return leg of one trip is the outgoing leg of the next. Furthermore, $t(\mu_{i+1})$ must be adjacent to $s(\mu_i)$; Otherwise, $P_{s,t}$ cannot be Hamiltonian. Between $t(\mu_i)$ and $s(\mu_{i+1})$, the path must have the form $Run[t(\mu_i), s(\mu_{i+1})]$, which may be just a single edge $(t(\mu_i), s(\mu_{i+1}))$.

The structure of $P_{t(\mu_j),s(\eta_1)}$ can be established in a way similar to (b).

This completes the proof. □

3.1.2 Separator η_1 has form D.

If η_1 has form D, then $P_{s,s(\eta_1)}$ cannot contain any corner separator cutting off s, as otherwise, P cannot visit all the internal vertices in the column of $t(\eta_1)$. Thus $j = 0$ in this case. Since there is no room for a corner separator, only a corner cookie, the structure of $P_{a,s(\eta_1)}$ must be the same as in Lemma 3(b).

The following lemma is straightforward.

Lemma 4 (η_1 has form D) *(a) If $x(t(\eta_1)) = 1$, $P_{s,s(\eta_1)} = seg[s,a]$ $Run[a,\alpha]$ $seg[\alpha, s(\eta_1)]$, where the $\mathcal{W}$ cookies have unit size. See Fig. 14(a).*

(b) Otherwise, $P_{s,s(\eta_1)}$ consists of a corner cookie containing s and w, followed by an edge on $\mathcal{W}$ connecting to $P_{a,s(\eta_1)}$, which has the structure given in Lemma 3(b). See Fig. 14(b).

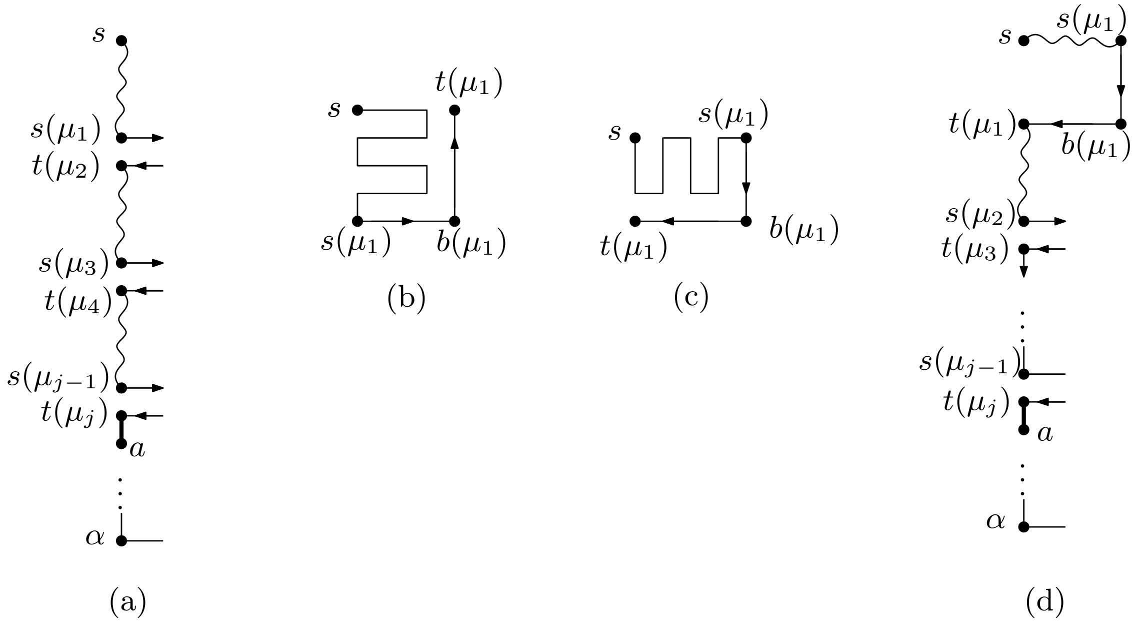


Figure 12: [34] (a) Order of endpoints of μ on $\mathcal{W}$ for even $j > 0$; (b) form of $P_{s,s(\mu_1)}$ for even $j > 0$; (c) form of $P_{s,s(\mu_1)}$ for j odd; and (d) order of endpoints of μ on $\mathcal{W}$ for j odd.

3.1.3 Separator η_1 has form A or F.

We define two rectangular regions of grid graph $\mathbb{G}$ covered by the initial subpath, and use them in designing our algorithm in Sections 5 and 6. Let w' denote the vertex on the $\mathcal{W}$ boundary in Row $y(b(\eta_1))$ for form F, and in Row $y(b(\eta_1)) - 1$ for form A as shown in Figs. 15(c)–(d). Recall that $b(\eta_1)$ is the first bend met on η_1 from the start $s(\eta_1)$ to the end $t(\eta_1)$. Let w'' be the vertex on the $\mathcal{W}$ boundary one row below w'. We denote by $\mathbb{R}_s$ the rectangular region of $\mathbb{G}$ that is delimited by Columns 0 and $x(w)$ and Rows 0 and $y(w')$; the rectangular region delimited by Columns 0 and $x(a')$ and Rows $y(w'')$ and $m - 1$ is denoted $\mathbb{R}_\alpha$.

Lemma 5 (**η_1 has form A or F**) *(a) If $t(\eta_1)$ is in Column 1, η_1 has form F, and the initial subpath is $(\mathbb{R}_s = seg[s, w'])seg[w', w''](\mathbb{R}_\alpha = Run[w'', \alpha],\ seg[\alpha, a'])\ seg[a', s(\eta_1)]$. If $s(\eta_1)$ is in Column 1, η_1 has form A, and the initial subpath is $(\mathbb{R}_s = Run[s, w'])seg[w', w''](\mathbb{R}_\alpha = seg[w'', \alpha], seg[\alpha, a'])\ seg[a', s(\eta_1)]$. In both cases, the run contains unit size $\mathcal{W}$ cookies. See Fig. 10.*

(b) Otherwise, if $j = 0$, then $P_{s,w'}$ contains a run of $\mathcal{W}$ cookies $Run[s, w']$ of size $x(w)$. If $j \geq 1$, then $P_{s,w'} = Run[s, s(\mu_1)]\ \mu_1\ Run[t(\mu_1), s(\mu_2)]\ \mu_2\ \ldots\ \mu_i\ Run[t(\mu_i), s(\mu_{i+1})]\ \mu_{i+1}\ \ldots\ \mu_{j-1}\ Run[t(\mu_{j-1}), s(\mu_j)]\ \mu_j$, followed by $Run[t(\mu_j), w']$ if w' and $t(\mu_j)$ do not coincide. $P_{w'',s(\eta_1)}$ has the structure of $P_{a,s(\eta_1)}$ in Lemma 3(b).

Proof:

(a) It is straightforward to see that when one of the endpoints of η_1 is in Column 1, the other

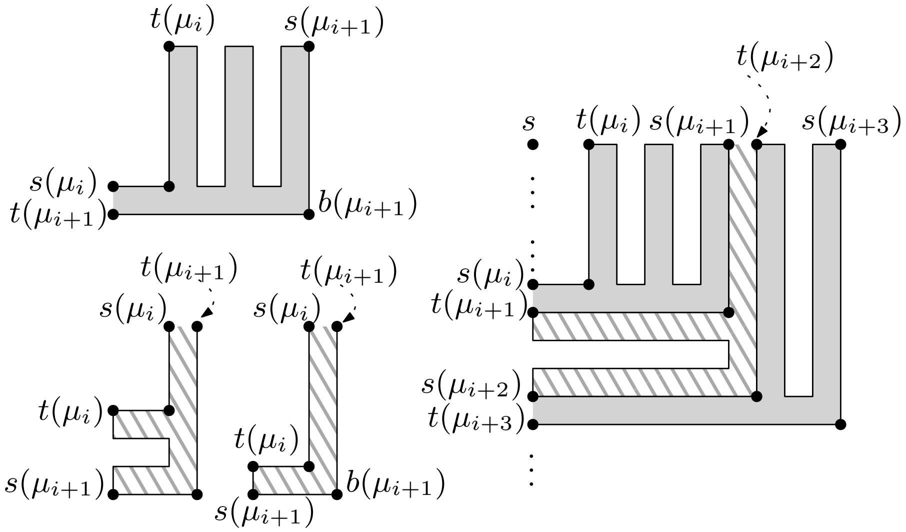


Figure 13: [34] Round trips $P_{s(\mu_i),t(\mu_{i+1})}$ start on $\mathcal{W}$ (solid gray) or on $\mathcal{N}$ (gray striped).

endpoint must be in Column 2. Therefore, the initial subpath can only contain one set of $\mathcal{W}$ cookies of unit size, and no $\mathcal{S}$ cookies or corner separators.

(b) First consider the case when $j = 0$. Then any internal path through vertices of $\mathbb{G}$ to the left of η_1 must be a cookie. If η_1 has form A, then the vertices in column $x(w)$ must be covered by a $\mathcal{W}$ corner cookie and then by $\mathcal{W}$ cookies above the lower segment of η_1. Now assume that η_1 has form F, and that there is an $\mathcal{S}$ cookie covering some vertices of Column $x(w)$ on or above Row $y(b(\eta_1))$. Then the internal vertices of Column $x(w)+1$ that lie in rows below Row $y(b(\eta_1))$ cannot be covered by any cookies and P fails to be Hamiltonian. Therefore, internal vertices of column $x(w)$ on or above Row $y(b(\eta_1))$ can only lie on $\mathcal{N}$ or $\mathcal{W}$ cookies or on a $\mathcal{W}$ corner cookie. Since j is 0, no $\mathcal{N}$ cookies are possible. Now consider $j \geq 1$. The structure of the subpath $P_{s,t(\mu_j)}$ can be established as in Lemma 3. The remaining vertices of $\mathbb{R}_s$, if any, must be covered by $\mathcal{W}$ cookies of size $x(w)$ by the logic above.

The proof for the structure of $\mathbb{R}_\alpha$ is similar to the proof of Lemma 3.

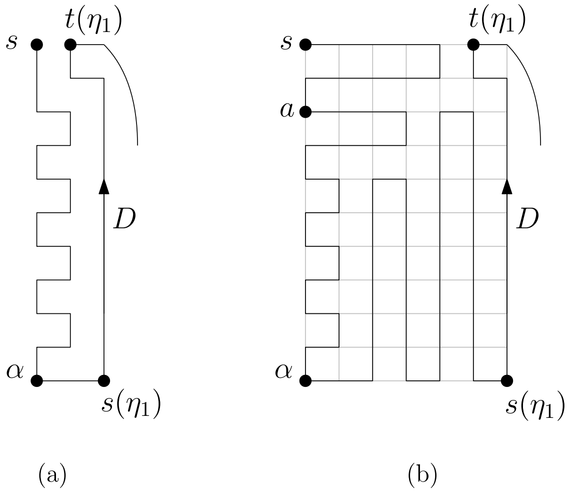
This completes the proof. □

Using the above lemmas, we define some terminology that we use in the algorithms in Sections 5 and 6. We say $\mathbb{R}_s$ is $\mathcal{W}$ *compatible* if it contains an even number of rows, and $\mathbb{R}_\alpha$ is $\mathcal{W}$ *compatible* if it contains an odd number of rows. When a region is $\mathcal{W}$ *compatible*, the subpath occupying this region (i.e., $\mathbb{R}_s$ and $\mathbb{R}_\alpha$) can be composed of only $\mathcal{W}$ cookies and boundary edges.

Fig. 16 illustrates the above terminology with some examples.

Lemma 6 *When η_1 has form A or F:*

(a) $\mathbb{R}_s$ is $\mathcal{W}$ compatible iff any $\mathcal{N}$ cookie in $\mathbb{R}_s$ has even size; $\mathbb{R}_\alpha$ is $\mathcal{W}$ compatible iff any $\mathcal{S}$ cookie in $\mathbb{R}_\alpha$ has even size.



(a) (b)

Figure 14: (a) Cross-separator η_1 has form D and is near $\mathcal{W}$; the initial subpath does not have any cookies. (b) Separator η_1 has form D and is far from $\mathcal{W}$. In both cases, there is no room for corner separators. Segments of separators that has length at least two contains arrows.

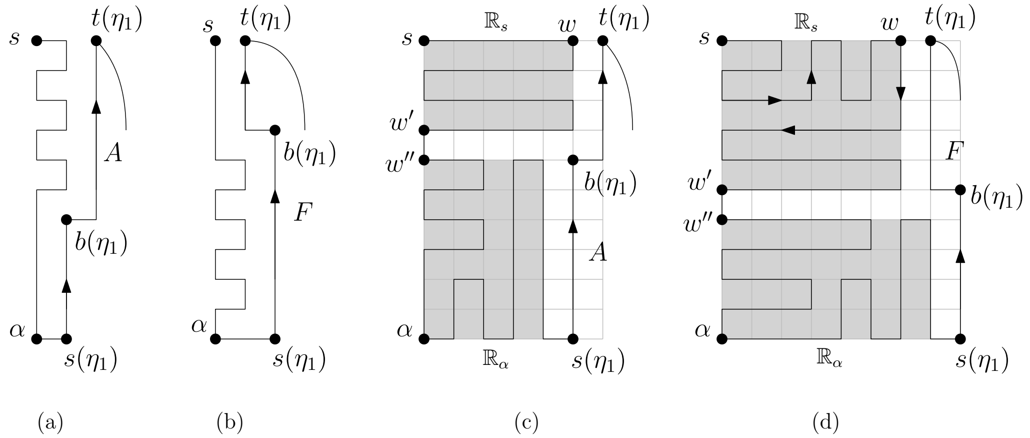
(b) At least one of $\mathbb{R}_s$ *and* $\mathbb{R}_\alpha$ *must be* $\mathcal{W}$ *compatible.*

Proof:

(a) We first prove that all the $\mathcal{N}$ cookies in the initial subpath must have the same parity in size, i.e., either they all have even sizes or all have odd sizes. From Lemma 5, two runs of $\mathcal{N}$ cookies must be separated by an even number of corner separators and together with sets of $\mathcal{W}$ cookies. Since each $\mathcal{W}$ cookie occupies two rows, the difference between the sizes of the $\mathcal{N}$ cookies must be even. Therefore, if the first run of $\mathcal{N}$ cookies has odd sized cookies, all the following $\mathcal{N}$ cookies have odd size; the same holds for even sizes.

First assume that $\mathbb{R}_s$ is $\mathcal{W}$ compatible. Then the total number of rows in $\mathbb{R}_s$ is even by definition of $\mathcal{W}$ compatibility. If $j = 0$, by Lemma 5, $\mathbb{R}_s$ contains a run of $\mathcal{W}$ cookies. If $j > 0$ is even, then all the vertices of Column 1 in $\mathbb{R}_s$ are covered by $\mathcal{W}$ cookies including a $\mathcal{W}$ corner cookie, and even number of corner separators. Then any $\mathcal{N}$ cookie between μ_i and μ_{i+1}, where i is odd, must cover an odd number of rows and thus has even size. For the case $j > 0$ odd, vertices of Column 1 in R_s must be covered by an $\mathcal{N}$ cookie, an odd number of corner separators, and some $\mathcal{W}$ cookies. Since there are an odd number of separators occupying an odd number of rows, the first corner $\mathcal{N}$ cookie must occupy an odd number of rows and must have even size. Therefore, all the $\mathcal{N}$ cookies in the initial subpath have even size.

We now assume that the $\mathcal{N}$ cookies have even sizes in the initial subpath. If $j \geq 0$ is even, then Column 1 is covered by $\mathcal{W}$ cookies, and if $j > 0$ by an even number of corner separators. Therefore, the number of rows in $\mathbb{R}_s$ must be even. If $j > 0$ is odd, then Column 1 is covered in some order by a combination of an even sized $\mathcal{N}$ cookie, an odd number of separators, and possibly some $\mathcal{W}$ cookies. Therefore, the total number of rows in $\mathbb{R}_s$ must be even, and $\mathbb{R}_s$ is $\mathcal{W}$ compatible. The claim for $\mathbb{R}_\alpha$ can be proved similarly.



(a) (b) (c) (d)

Figure 15: (a) Separator η_1 is near $\mathcal{W}$ and has form A of η_1; (b) η_1 is near $\mathcal{W}$ and has form F. (a) Separator η_1 is far from $\mathcal{W}$ and has form A of η_1; (b) η_1 is far from $\mathcal{W}$ and has form F. Rectangular regions are shown in grey. Segments of separators have arrows when they have length at least two.

(b) If m is even, then either one or both $\mathbb{R}_s$ and $\mathbb{R}_\alpha$ contain an even number of rows, or else both of them contain an odd number of rows. Therefore, one of them must be $\mathcal{W}$ compatible. If m is odd then one of them has an even number of rows and the other has an odd number of rows. If $\mathbb{R}_s$ contains an even number of rows then both $\mathbb{R}_s$ and $\mathbb{R}_\alpha$ are $\mathcal{W}$ compatible.

We now assume for a contradiction that m is odd and $\mathbb{R}_\alpha$ contains an even number of rows, and hence is not $\mathcal{W}$ compatible; consequently, $\mathbb{R}_s$ contains an odd number of rows. By Lemma 5, subpath $P_{a,a'}$ is a Hamiltonian path between two diagonally opposite corners of a rectangular grid (i.e., $\mathbb{R}_\alpha$). Since $\mathbb{R}_\alpha$ has an even number of rows, $\mathbb{R}_\alpha$ must have an odd number of columns [11]. Since the number of columns in $\mathbb{R}_s$ is one less or one more than in $\mathbb{R}_\alpha$, $\mathbb{R}_s$ must have an even number of columns. But the subpath $P_{s,w'}$ is a Hamiltonian path between the top-left and bottom-left corners of the rectangular grid $\mathbb{R}_s$, which cannot exist when $\mathbb{R}_s$ has an even number of columns and an odd number of rows [11]. Therefore, $\mathbb{R}_s$ cannot have an odd number of rows when m is odd, a contradiction. Thus, $\mathbb{R}_\alpha$ must have an odd number of rows, and so is $\mathcal{W}$ compatible, when m is odd.

This completes the proof. □

3.1.4 Summary of forms for the initial subpath

The following theorem follows from Lemmas 3–5, and summarizes the structure of the initial subpath for all the forms of η_1.

Theorem 1 (Structure of Initial Subpath) *(a) If η_1 is near $\mathcal{W}$, $P_{s,\alpha}$ is either a segment seg$[a, \alpha]$ (η_1 has form B or G), or composed of a segment and a run of unit $\mathcal{W}$ cookies (η_1 has form D, F or A). The subpath $P_{\alpha,s(\eta_1)}$ is always a segment on the $\mathcal{S}$ boundary. See Fig. 17.*

(b) If η_1 is far from $\mathcal{W}$ and $j = 0$, $P_{s,s(\eta_1)}$ must have a corner cookie containing s and w followed by an edge on the $\mathcal{W}$ boundary connecting the base of the corner cookie to vertex a. The path

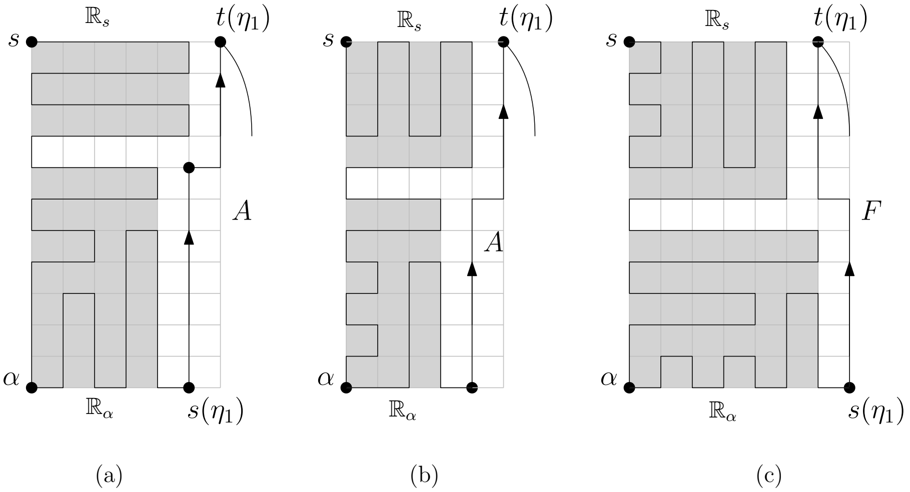


Figure 16: (a) Separator η_1 has form A and $\mathbb{R}_s$ is $\mathcal{W}$ compatible but $\mathbb{R}_\alpha$ is not $\mathcal{W}$ compatible. (b) $\mathbb{R}_s$ is not $\mathcal{W}$ compatible but $\mathbb{R}_\alpha$ is $\mathcal{W}$ compatible. (c) Separator η_1 has form F and $\mathbb{R}_s$ is $\mathcal{W}$ compatible but $\mathbb{R}_\alpha$ is not $\mathcal{W}$ compatible. Segments of separators have arrows when they have length at least two.

$P_{a,a'}$ from a to a' takes one of the following four forms: $seg[a,\alpha]$ $Run[\alpha,a']$; or $Run[a,\alpha]$ $seg[\alpha,a']$; or $Run^{\geq}[a,\alpha]$ $seg[\alpha,u]$ $Run^{\leq}[u,a']$, where u is at least two units from α on $\mathcal{S}$; or $Run^{\geq}[a,u']$, $seg[u',\alpha]$ $Run^{\leq}[\alpha,a']$, where u' is at least two units from α on $\mathcal{W}$. The remainder of path $P_{s,s(\eta_1)}$ is the segment $seg[a',s(\eta_1)]$. See Fig. 18.

(c) If $j \geq 1$ (η_1 must be far from $\mathcal{W}$ and has form B, G, F or A), then $P_{s,t(\mu_j)} = Run[s,s(\mu_1)]$ μ_1 $Run[t(\mu_1),s(\mu_2)]$ μ_2 $\ldots$ μ_i $Run[t(\mu_i),s(\mu_{i+1})]$ μ_{i+1} $\ldots$ μ_{j-1} $Run[t(\mu_{j-1}),s(\mu_j)]$ μ_j. Path $P_{t(\mu_j),s(\eta_1)}$ either has the structure of $P_{a,s(\eta_1)}$ in (b), or, consists of edge $(t(\mu_j),\alpha)$ followed by $seg[\alpha,s(\eta_1)]$. See Fig. 19.

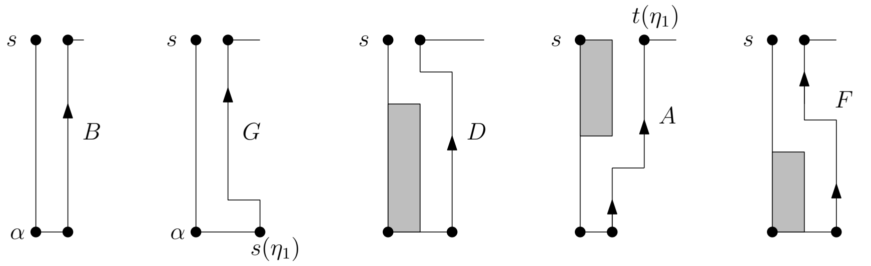


Figure 17: The structure of the initial subpath when η_1 is near $\mathcal{W}$; the grey box denotes a run of unit size $\mathcal{W}$ cookies. Segments of separators have arrows when they have length at least two.

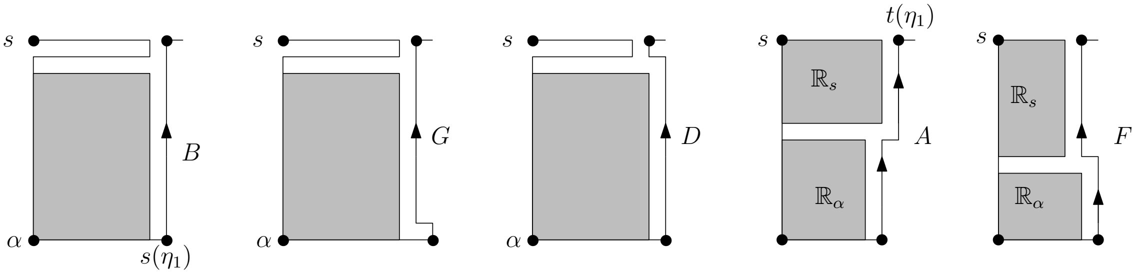


Figure 18: The initial subpath when η_1 is far from $\mathcal{W}$ and $j = 0$; there are no corner separator cutting off s. The grey boxes are explained in Theorem 1(b). Segments of separators have arrows when they have length at least two.

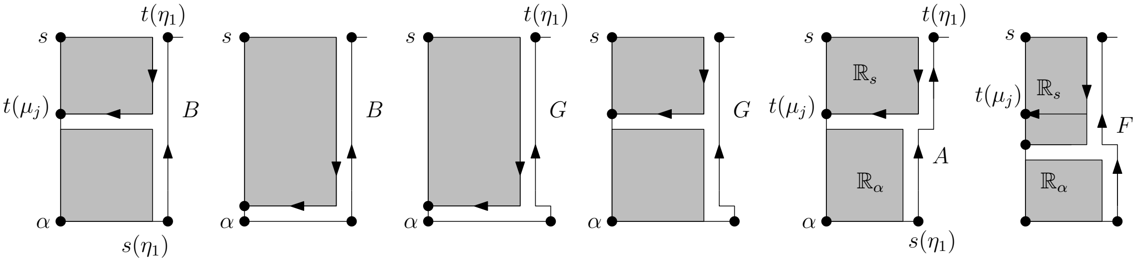


Figure 19: The initial subpath when η_1 is far from $\mathcal{W}$ and $j \geq 1$; there is at least one corner separator cutting off s. The grey boxes are explained in Theorem 1(c). Segments of separators have arrows when they have length at least two.

3.2 Middle subpath

The middle subpath $P_{s(\eta_1),t(\eta_k)}$ travels from the start vertex $s(\eta_1)$ of the first cross-separator η_1 to the terminal vertex of the last cross-separator η_k and thus by definition contains all the cross-separators η_i, $1 \leq i \leq k$. For an example of a middle subpath, see Fig. 6, which shows a 1-complex path with a middle subpath that passes through 11 cross-separators as it travels from its start vertex $s(\eta_1)$ to its terminal vertex $t(\eta_{11})$.

In this subsection, we determine all the possible structures for the middle subpath of a 1-complex s, t Hamiltonian path $P_{s,t}$. Since the number k of cross-separators is not fixed, we describe these structures by giving a table in Theorem 2 at the end of this section; this table can be used to generate the possibilities for the middle subpath, based on any given form for the first cross-separator η_1.

Recall that by Lemma 2(v), k is odd. When $k = 1$, the middle subpath is comprised entirely of η_1, i.e., $P_{s(\eta_1),t(\eta_1)} = \eta_1$. The possible forms for the first cross-separator η_1 have been given already in Subsection 3.1 and shown in Fig. 9. We used these forms in Subsection 3.1 to determine the boundary of the region that the initial path $P_{s,s(\eta_1)}$ has to cover. We will later introduce additinal forms for $k > 1$.

The rest of this subsection is devoted to the case $k > 1$, i.e., k is odd and at least 3. We first give a brief overview of our approach and then give a key lemma, Lemma 7. One part of this lemma motivates our notation for forms for η_i, $i > 1$. Once the notation is in place, we them provide a further guide to the remainder of the subsection.

Our approach to handling $k \geq 3$ cross-separators is as follows. The middle subpath begins with η_1, which travels from $\mathcal{S}$ to $\mathcal{N}$, and then makes one or more roundtrips that each travel from $\mathcal{N}$ to $\mathcal{S}$ and back to $\mathcal{N}$; see Figures 20 and 21. We refer to such roundtrips as $\mathcal{NSN}$ roundtrips. Similarly, the middle subpath may make $\mathcal{SNS}$ roundtrips. Roundtrips of either type are comprised of two consecutive cross-separators η_i and η_{i+1} together with the part of the middle subpath that joins them. Our goal is to provide a table that, based on the form of η_i for i odd, $i \leq k-2$, lists all the possibilities for the $\mathcal{NSN}$ roundtrip that follows η_i. Hence by stepping through the odd values of i, $1 \leq i \leq k-2$, and repeatedly using the possibilities for the next $\mathcal{NSN}$ roundtrip, we obtain all the possible structures for the middle subpaths when $k \geq 3$.

The next lemma provides key properties of roundtrips of either type, $\mathcal{NSN}$ or $\mathcal{SNS}$. Both types will be used later. This is because the first cross-separator η_{i+1} of an $\mathcal{NSN}$ roundtrip containing η_{i+1} and η_{i+2} is in fact the return portion of the preceding $\mathcal{SNS}$ roundtrip that contains η_i and η_{i+1}, i.e., η_{i+1} is shared by the two roundtrips. Thus the properties of roundtrips of either type are essential. In the lemma below, index i may be even or odd. Recall that odd-indexed η_i travel from $\mathcal{S}$ to $\mathcal{N}$ and even-indexed η_i travel from $\mathcal{N}$ to $\mathcal{S}$.

Lemma 7 (constraints on roundtrips η_i, η_{i+1}) *(a) When η_i is bent, then its terminal vertex $t(\eta_i)$ and its start vertex $s(\eta_i)$ lie in adjacent columns: $x(t(\eta_i)) = x(s(\eta_i)) \pm 1$. Otherwise, when η_i is straight, the vertices lie in the same column: $x(t(\eta_i)) = x(s(\eta_i))$.*

(b) The subpath of P from $s(\eta_i)$ to $t(\eta_{i+1})$ covers all the grid vertices that lie in $\mathbb{G}$ between η_i and η_{i+1}. In particular, the terminal vertex $t(\eta_{i+1})$ of η_{i+1} lies one unit east of the start vertex $s(\eta_i)$ of η_i: $x(t(\eta_{i+1})) = 1 + x(s(\eta_i))$.

(c) The terminal vertex $t(\eta_i)$ of η_i and the start vertex $s(\eta_{i+1})$ of η_{i+1} are 1, 2, or 3 units apart.

(d) When start vertex of η_{i+1} lies 3 units east of $t(\eta_i)$, the subpath of P between them contains exactly one cookie: $P_{t(\eta_i),s(\eta_{i+1})}$ is comprised of one cookie and its adjoining boundary edge on each side of the cookie. When η_{i+1} has bends in a row one unit from $\mathcal{N}$ or $\mathcal{S}$, then the cookie has size 0 and becomes an edge on the boundary.

Proof:

(a) The two bends of a cross-separator are adjacent in $\mathbb{G}$, as otherwise, any vertex between them would not reach a boundary by a segment of P. Since P is 1-complex, the bends themselves connect directly to opposite boundaries by segments of P. This implies statement (a), as the second part of the statement is clear.

(b) The path $P_{s(\eta_i),t(\eta_{i+1})}$ forms a round trip between opposite boundaries. By Lemma 1, removal of the path $P_{s,t(\eta_{i+1}}$ from s to $t(\eta_{i+1})$ leaves only one connected component of $\mathbb{G}$. Since this component contains t, there can be no grid vertices between η_i and η_{i+1} left uncovered by the subpath from s to $t(\eta_{i+1})$. In particular, there can be no uncovered vertex on the boundary between η_i and η_{i+1}. Since η_{i+1} lies east of η_i by Lemma 2 vii), it follows that the terminal vertex $t(\eta_{i+1})$ of η_{i+1} lies one unit east of $s(\eta_i)$.

(c) Statement (c) follows immediately from statement (a).

(d) The subpath $P_{t(\eta_i),s(\eta_{i+1})}$ of $P_{s,t}$ between the terminal vertex of η_i and the start vertex of η_{i+1} belongs to the roundtrip subpath $P_{s(\eta_i),t(\eta_{i+1})}$, which by statement (b) covers all grid vertices lying in $\mathbb{G}$ strictly between η_i and η_{i+1}. Hence $P_{t(\eta_i),s(\eta_{i+1})}$ must cover all grid vertices strictly between η_i and η_{i+1}. Since cross-separators are indexed in the order they are met along $P_{s,t}$, subpath $P_{t(\eta_i),s(\eta_{i+1})}$ cannot contain any cross- separator. Clearly no corner separators, no west cookies, and no east cookies can contain a vertex between $t(\eta_i)$ and $s(\eta_{i+1})$ on the boundary, nor can any cookie based on the opposite boundary. Therefore the two boundary vertices between $t(\eta_i)$ and $s(\eta_{i+1})$ must be covered by a cookie based on that boundary. The statement now follows.

This completes the proof. □

We now introduce our notation for the possible forms of cross-separators when $k \geq 1$. (Refer to Figs. 20 and 21, explained in more detail below.) Recall that since P is 1-complex, each internal grid point of $\mathbb{G}$ is connected to a boundary by a straight line segment consisting of one or more grid edges belonging to P. Hence cross-separators are either straight, or bent towards $\mathcal{E}$ or $\mathcal{W}$ by one unit from their start vertices $s(\eta_i)$. The bends of a cross-separator must be adjacent on the grid, as otherwise, the vertices between them could not reach any boundary along a straight segment of P.

The first five forms for i odd have been previously introduced as possibilities for η_1; they are also possibilities for i odd but greater than one. In addition, there are two more possibilities for $i > 1$ and odd. We discuss these further below.

As shown in Figs. 20 and 21, we distinguish between a cross-separator that bends in a row one unit from the $\mathcal{N}$ or $\mathcal{S}$ boundary and one that bends in the same direction, but at least two units from either boundary (e.g., we distinguish among forms G, D, and F). This is useful because forms with bends one unit from a boundary must be preceded or followed by two boundary edges on P, as shown by the white dot in the figures. Thus even though such a cross-separator bends, it nevertheless assures coverage on an entire column of grid vertices.

In the figures of this subsection, an arrow on a segment indicates that the segment must have length at least 2; segments without arrows are grid edges of length 1 that lie on P.

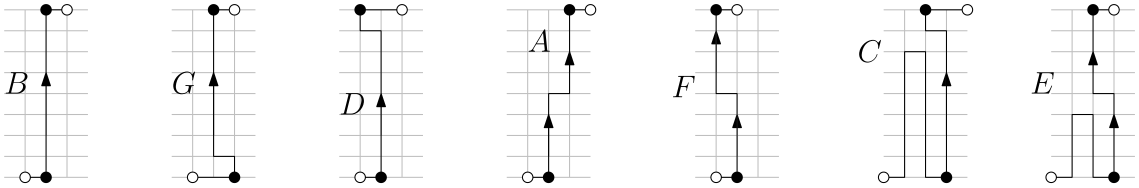


Figure 20: Seven forms for η_i, when i is odd. Forms C and E apply only when $t(\eta_{i-1})$ is 3 units west of $s(\eta_i)$, as these forms by definition include the $\mathcal{S}$ cookie required by Lemma 7(d). Segments of separators have arrows when they have length at least two.

By Condition (d) of Lemma 7, it can occur that after η_1, some bent separator η_i, $i > 1$, must have a preceding cookie. To describe this situation, we introduce two special cases E and C (see Fig. 20) of forms D and F, respectively. Forms E and C describe the scenario that an $\mathcal{S}$ cookie must occur on $P_{s,t}$ between $t(\eta_{i-1})$ and $s(\eta_i)$, where this cookie is joined to $s(\eta_i)$ by a boundary edge on $P_{s,t}$. These forms can apply only to those η_i such that $i > 1$ is odd. They facilitate checking that all grid vertices between η_1 and η_k are covered. It may occur that in the initial path $P_{s,s(\eta_1)}$, an $\mathcal{S}$ cookie is joined by an edge on $\mathcal{S}$ to the start vertex $s(\eta_1)$ of η_1; such a cookie has no effect on the middle subpath $P_{s(\eta_1),t(\eta_k)}$. As forms E and C are not needed to characterize the initial subpath, they are only introduced now for the middle subpath.

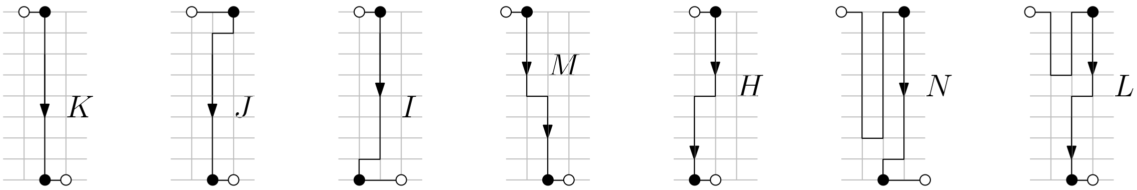


Figure 21: Seven forms for η_i when i is even. The forms for i even and for i odd are mirror images with respect to the horizontal midline of $R_{\mathbb{G}}$, e.g., K and B correspond, as do J and G, etc. Segments of separators have arrows when they have length at least two.

Form A is the only form for η_i where i is odd such that η_i bends one unit towards the east to reach its terminal point: $x(t(\eta_i)) = x(s(\eta_i))$. This is because the bends of such a cross-separator cannot occur in the row one unit above $\mathcal{S}$ or one unit below $\mathcal{N}$; in the former case, the vertex on $\mathcal{S}$ one unit east of $s(\eta_i)$ cannot be covered by P, and in the latter case, the vertex on $\mathcal{N}$ one unit west of $t(\eta_i)$ cannot be covered by P.

Fig. 21 enumerates all the forms for cross-separators η_i, where $i > 1$ is even. Each of these forms is the mirror image, with respect to the horizontal mid-line of $R_{\mathbb{G}}$, of a corresponding form for i odd.

Our approach for the remainder of the subsection is as follows. First, we use Lemma 7 (roundtrip constraints) to prove Lemma 8 (SNS roundtrip forms) below. This lemma lists all the possible $\mathcal{SNS}$ roundtrips, i.e., roundtrips containing η_i and η_{i+1} for odd i. Then we exploit the fact that cross-separator η_{i+1} is the first cross-separator of the $\mathcal{NSN}$ roundtrip that immediately follows η_i for i odd ($i \leq k-2$). Given the form of η_i, the possibilities for the next $\mathcal{NSN}$ roundtrip can easily be determined by taking the mirror images, with respect to the horizontal midline of $R_{\mathbb{G}}$, of the list of possibilities for the preceding roundtrip that also travels along η_{i+1}. Because the notation must be suitably adjusted and for ease of reference, we state the "mirror image" of Lemma 8 as a separate lemma, Lemma 9 (NSN roundtrip forms).

Finally, we put the lists of possibilities for the first roundtrip, which contains η_i and η_{i+1} together with the second roundtrip, which contains η_{i+1} and η_{i+2}, to obtain the list of possibilities for the NSN roundtrips that come just after each of the forms of η_i, $i \leq k-2$ and odd. This yields the theorem that specifies the middle subpath structure.

Now we state and prove the lemma that lists the possibilities for the $\mathcal{SNS}$ roundtrips. The proof makes repeated use of Lemma 7, especially the fact that the start and end vertices of a roundtrip must be adjacent grid points on the same boundary and the fact that the start and end vertices of a cross-separator lie in the same or adjacent columns of $\mathbb{G}$. These facts, combined with other items in Lemma 7 and Lemma 2, enable a straight-forward case analysis.

Lemma 8 ($\mathcal{SNS}$ roundtrip forms) *Let η_i be a cross-separator of a 1-complex path P where η_i is directed $\mathcal{S}$ to $\mathcal{N}$ and i is odd and less than k. Then based on the form of η_i, the SNS roundtrip from $s(\eta_i)$ to $t(\eta_{i+1})$ can only take the following forms (see Figures 20 and 21 for the forms):*

(a) η_i bends towards $\mathcal{W}$ one unit above $\mathcal{S}$ (form G): GN where N bends one unit above $\mathcal{S}$, and GM where M bends two units above $\mathcal{S}$;

(b) η_i bends towards $\mathcal{W}$ two or more units from both $\mathcal{S}$ and $\mathcal{N}$ (forms E and F): EL and FL where the bends of E or F and the bends of L are in the same row, and EM and FM where the bends of M lie in the row one unit above the row of bends of E or F;

(c) η_i *bends towards* $\mathcal{W}$ *one unit below* $\mathcal{N}$ *(forms C and D):* CJ, CK, DJ, DK*;*

(d) η_i *is straight (form B):* BJ, BK*;*

(e) η_i *bends towards* $\mathcal{E}$ *(form A):* AH *and* AI*, where the bends of* A *are exactly 2 units above* $\mathcal{S}$*.*

Proof:

(a) By Lemma 7 part (b), the terminal vertex $t(\eta_{i+1})$ lies one unit east of $s(\eta_i)$, and by part (c), the location of the start vertex $s(\eta_{i+1})$ must lie one, two, or three units east of $t(\eta_i)$.

If $s(\eta_{i+1})$ lies one unit east of $t(\eta_i)$, then η_{i+1} must bend towards $\mathcal{W}$ to reach $t(\eta_{i+1})$. The row of its bends must lie one unit above the row of the bends of η_i, so that as required by Lemma 7, the roundtrip does not leave any uncovered grid vertices between η_i and η_{i+1}. Thus the bends of η_{i+1} are at least two units above $\mathcal{S}$; furthermore, they cannot lie one unit below $\mathcal{N}$ because in that case, the grid vertex on $\mathcal{N}$ above the second bend of η_{i+1} could not be covered by the middle path. Note that by definition of form M, its bends are at least two units from both $\mathcal{S}$ and $\mathcal{N}$. Hence the form GM is possible.

If $s(\eta_{i+1})$ lies two units east of $t(\eta_i)$, then η_{i+1} is straight and has form K. By Lemma 2 part (vi), the two grid edges on $\mathcal{N}$ between $t(\eta_i)$ and $s(\eta_{i+1})$ must lie on the roundtrip path, which contradicts the requirement of Lemma 7 part (b) because the interior grid vertices in the column of $s(\eta_i)$ above the first bend of η_i would not be covered by the roundtrip. Thus GK is not a possible roundtrip.

Finally, if $s(\eta_{i+1})$ lies three units east of $t(\eta_i)$, then η_{i+1} must bend towards $\mathcal{W}$ to reach $t(\eta_{i+1})$. By Lemma 7 part (d), there must be a single $\mathcal{N}$ cookie between $t(\eta_i)$ and $s(\eta_{i+1})$. In order to satisfy the vertex coverage requirement of Lemma 7 part (b), the end of this cookie must lie one unit above the row containing the bends of η_i. To avoid collision with the cookie, η_{i+1} must have its bends in the row one unit above $\mathcal{S}$, just like η_i. Note that by definition, the bends of N are one unit above $\mathcal{S}$. Thus GN is a possible form for the roundtrip containing η_i and η_{i+1}.

(b) The analysis is the same as for item (a). However the labels of the forms are different because their bends occur at least two units above $\mathcal{S}$.

(c) Vertex $t(\eta_{i+1})$ must lie one unit east of $s(\eta_i)$ by Lemma 7(b). Hence by part (c) of that Lemma, $s(\eta_{i+1})$ could only lie 1, 2, or 3 units east of $t(\eta_i)$. If $s(\eta_{i+1})$ were one unit east of $t(\eta_i)$, then C would have to bend eastward to reach $t(\eta_{i+1})$, and its first bend would intersect η_i, so CM and DM cannot be SNS roundtrips.

If $s(\eta_{i+1})$ were 2 units east of $t(\eta_i)$, then $s(\eta_{i+1})$ would lie in the same column as $t(\eta_{i+1})$. This possibility gives rise to form K, so CK and DK are possible SNS roundtrips as they leave no vertices uncovered between the two cross-separators.

If $s(\eta_{i+1})$ were 3 units east of $t(\eta_i)$, then by Lemma 7( ), there would have to be an $\mathcal{N}$ cookie between them, joined to them by their edges on $\mathcal{N}$. The cookie and adjoining edges would have to lie on P by part (d) of the lemma. However, a bend in this cookie would intersect η_i unless the cookie had size 0 and degenerated to a boundary edge. The latter possibility gives rise to an $\mathcal{SNS}$ roundtrip of the form CJ or DJ because the path $P_{s(\eta_i),t(\eta_{i+1})}$ leaves no vertices uncovered between η_i and η_{i+1}.

(d) Vertex $t(\eta_{i+1})$ lies in the column one unit east of η_i. Hence $s(\eta_{i+1})$ can only lie in this column, in which case η_{i+1} is straight and has form K, or in the next column to the east. In this case, η_{i+1} must bend to the west in order to reach $t(\eta_{i+1})$. To ensure coverage of all internal vertices between η_i and η_{i+1} as required by Lemma 7 part(b), in particular the internal vertices in the column of $t(\eta_{i+1})$ above the bend of η_{i+1} in that column, the bend must lie one unit below $\mathcal{N}$, so η_{i+1} has form J. By Lemma 2 part vi), the grid edges on $\mathcal{N}$ between $t(\eta_i)$ and $s(\eta_{i+1})$ belong to the path P, so the vertex between those two vertices is covered by the roundtrip. Hence BK and BJ are possible SNS roundtrips.

(e) The bends in form A cannot lie in the row next to $\mathcal{S}$ or $\mathcal{N}$, as otherwise, by Lemma 2 (vi), path P would not be able to cover the grid vertex on the $\mathcal{S}$ boundary or the $\mathcal{N}$ boundary that is adjacent to the first or second bend on A, respectively. Thus the vertical segments of A each have length at least 2. By Lemma 7(b), $t(\eta_{i+1})$ must lie one unit east of $s(\eta_i)$. By part (c) of the same lemma, $s(\eta_{i+1})$ must lie one unit west of $t(\eta_i)$, or at $t(\eta_i)$, or one unit east of $t(\eta_i)$. Only the last possibility avoids self-intersection. Hence η_{i+1} bends to the west to reach $t(\eta_{i+1})$. If the bends of η_{i+1} occur at least two units above $\mathcal{S}$, then the roundtrip has form AH, and if the bends are exactly 2 units above $\mathcal{S}$, then the roundtrip has form AI. Both the roundtrips are possible as they leave no vertices between the cross-separators uncovered.

This completes the proof. □

The next lemma "mirrors" the previous lemma with respect to the horizontal midline of $R_{\mathbb{G}}$.

Lemma 9 ($\mathcal{NSN}$ roundtrip forms) *Let η_i be a cross-separator of a 1-complex path P where η_i is directed $\mathcal{N}$ to $\mathcal{S}$ and i is even. Then based on the form of η_i, the NSN roundtrip from $s(\eta_i)$ to $t(\eta_{i+1})$ can only take the following forms (see Figures 20 and 21 for the forms):*

(a) η_i bends towards $\mathcal{W}$ one unit below $\mathcal{N}$ (form J): JC where C bends one unit below $\mathcal{N}$, and JA where A bends two units below $\mathcal{N}$;

(b) η_i bends towards $\mathcal{W}$ two or more units from both $\mathcal{N}$ and $\mathcal{S}$ (forms L and H): LE and HE where the bends of L or H and the bends of E are in the same row, and LA and HA where the bends of A lie in the row one unit below the row of bends of L or H;

(c) η_i bends towards $\mathcal{W}$ one unit one unit above $\mathcal{S}$ (forms N and I): NG, NB, IG, IB;

(d) η_i is straight (form K): KG, KB;

(e) η_i bends towards $\mathcal{E}$ (form M): MF, where the bends of M are one unit below $\mathcal{N}$.

Proof: The proof is the same as the proof of Lemma 8 after a change of notation as follows.

- Interchange the forms (refer to Figures 20 and 21 for the forms): $B \longleftrightarrow K$, $G \longleftrightarrow J$, $D \longleftrightarrow I$, $A \longleftrightarrow M$, $F \longleftrightarrow H$, $C \longleftrightarrow N$, and $E \longleftrightarrow L$.
- Interchange the horizontal boundaries $\mathcal{N} \longleftrightarrow \mathcal{S}$.
- Interchange the terms *above* and *below.*

This completes the proof. □

Based on the previous two lemmas, we now can give the possible forms this situation: a cross-separator η_i, where i is odd and $k \geq 3$, followed by a $\mathcal{NSN}$ roundtrip containing η_{i+1} and η_{i+2}. In

Theorem 2 below, we present the possible forms for this triple of cross-separators as a table that lists the ways to append a $\mathcal{NSN}$ roundtrip to a middle subpath currently ending with a given form for η_i, i odd. The forms for η_i are listed in groups that admit the same possibilities for the following roundtrip.

The content of the table is illustrated in Fig. 22.

An illustration of the application of the table is given in Figure 23. This figure shows the generation of a middle subpath with 5 cross-separators in a grid with 6 rows. The figure gives a layered graph whose first layer lists all the possibilities for the form of η_1. The next layer uses the table to list all the possibilities for η_{i+1}, where edges from layer 1 to layer 2 show the allowed successors. Each layer is labelled below with its corresponding η_i.

Each layer is represented as a grey box, where one side of the box lists forms in a suitable order with respect to the incoming edges from the previous layer, and the other side of the box permutes the list of forms to a good order with respect to the outgoing edges to the next layer. The permutations are chosen to reduce edge crossings in the diagram. Edges internal to any box are drawn between identical forms.

Any middle subpath with 5 cross-separators in a grid with 6 rows corresponds to a path in the layered diagram from the first to the fifth layer. Each pair of adjacent layers represents a roundtrip that satisfies the roundtrip lemmas (Lemma 8 and Lemma 9) of this section. The upper part of Fig. 23 highlights with colors three example paths in the layered graph. Their corresponding paths in the grid are shown in the same color in the lower part of the figure.

Theorem 2 (Structure of Middle Subpath) *Let η_i be a cross-separator of a* 1*-complex path P, where η_i is directed $\mathcal{S}$ to $\mathcal{N}$ and i is odd, $1 \leq i \leq k-2$. i.e., $k \geq 3$ and η_i is followed by at least one $\mathcal{NSN}$ roundtrip. Then based on the form of η_i, the $\mathcal{NSN}$ roundtrip from $s(\eta_{i+1})$ to $t(\eta_{i+2})$ must take one of the forms listed in the table below. See Figures 20 and 21 for the forms.*

	Form of η_i	*$\mathcal{NSN}$ roundtrip of η_{i+1} and η_{i+2}*
Group I	A	HA, HE, IB, *or* IG
Group II	B, D *or* C	JC, JA, KB, *or* KG
Group III	F *or* E	LA, LE, MD, *or* MF
Group IV	G	MD, MF, NB, *or* NG

Proof: The proof follows immediately from Lemma 8 and Lemma 9. For each group, the first form in the list of $\mathcal{NSN}$ roundtrips (shown as pairs in the second column) is obtained from Lemma 8, based on the form or forms for η_i for each group. The second form in each pair in the list of $\mathcal{NSN}$ roundtrips is obtained from Lemma 9, based on the first form of the pair. □

4 Zip Procedure

In this section, we define a *zip* procedure Z (zip for short) that applies a sequence of switches to cells that appear on two sides of a line (row or column) of $\mathbb{G}$; see Fig. 25(b) and Fig. 26. The cells must be *switchable*, i.e., each such grid cell has two parallel grid edges that belong to a cycle-path cover of $\mathbb{G}$ and two parallel grid edges that do not. We call the line (row or column of $\mathbb{G}$) on which the switchable cells are incident a *zipline*, and denote it by $l_z^{q_1,q_2}$ (the superscript may be omitted for short), directed from vertex q_1 to vertex q_2 where q_1 and q_2 are not corners of $R_{\mathbb{G}}$.

We begin by observing some properties of the switch operation. As mentioned in Section 1, a *cycle-path cover* $\mathbb{P}$ of $\mathbb{G}$ is a set of cycles and paths that collectively cover all the vertices of $\mathbb{G}$;

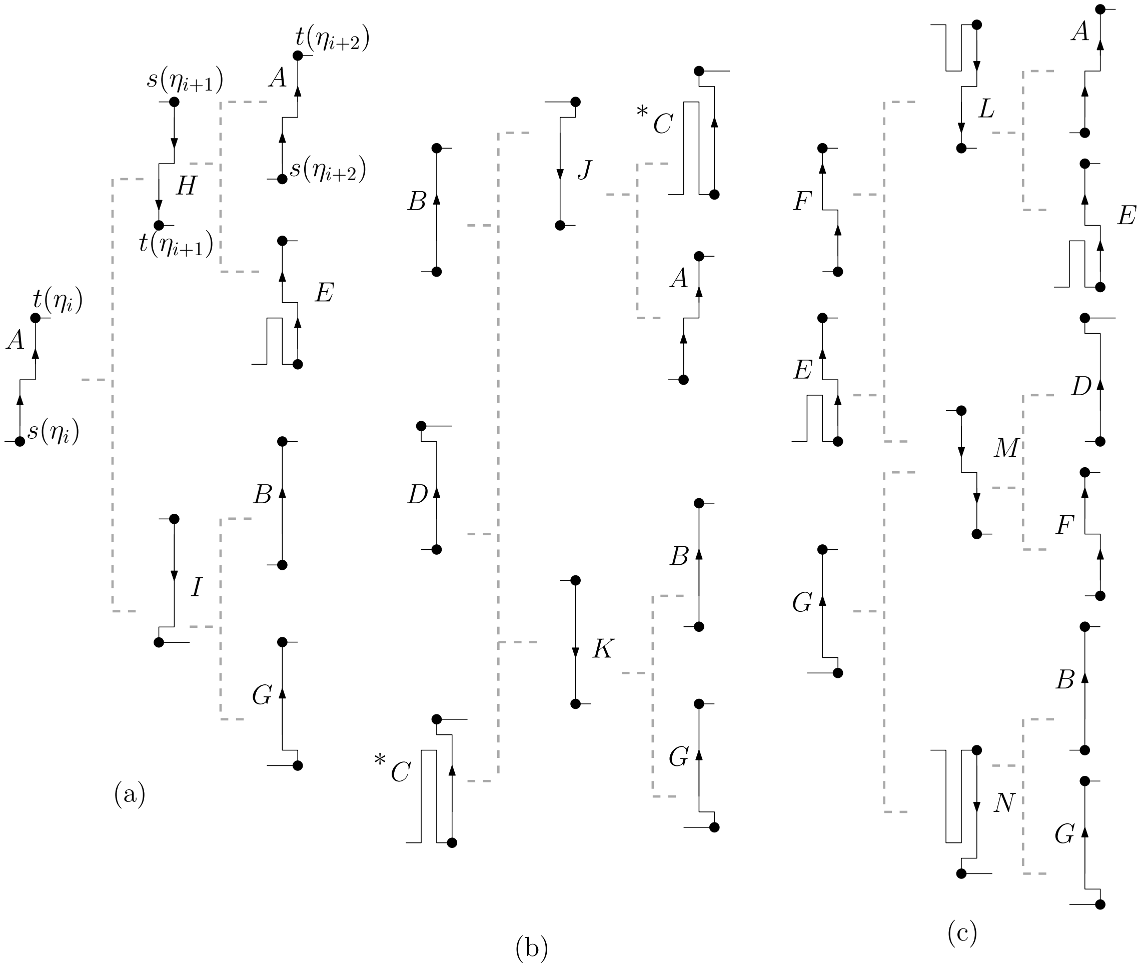


Figure 22: The adjacency constraints on η_i and the following roundtrip containing η_{i+1} and η_{i+2}, where i is odd and $1 \leq i \leq k-2$: (a) η_i in Group I; (b) η_i in Group II (η_1 cannot have form C); and (c) η_i in Group III (η_1 cannot have form E) and η_i in Group IV.

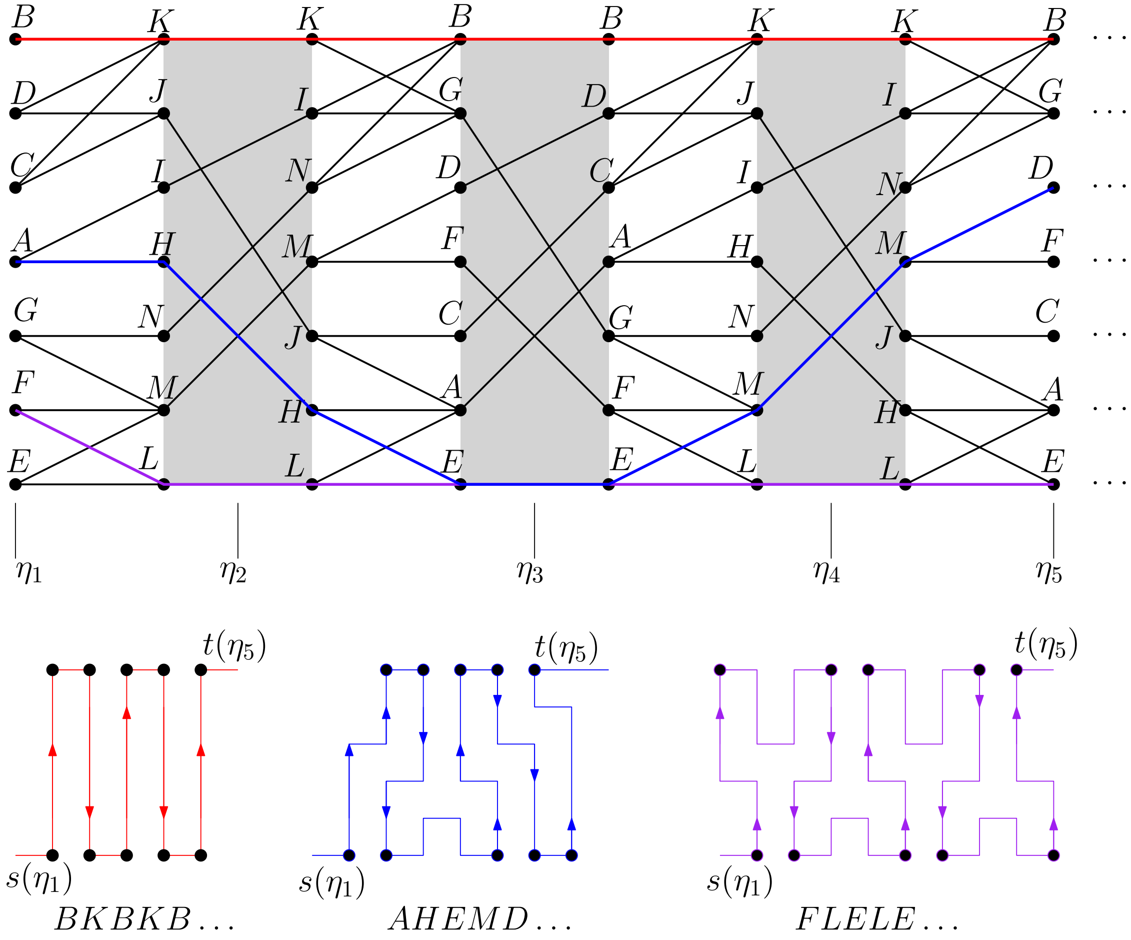


Figure 23: *Above:* graphical representation of the adjacency constraints of the forms of the first five separators η_1 to η_5. *Below:* three examples showing the actual structure of the subpath of P from $s(\eta_1)$ to $t(\eta_5)$ for the red, blue and purple paths above. Segments of separators have arrows when they have length at least two.

a switch operation applied to a cycle-path cover always produces a cycle-path cover. To see this note that a switch may change the edges of the cover incident to a vertex but not the degree of the vertex (degree 2 for each vertex except s and t, which remain degree 1). See Fig. 24(b)–(c) for an example. The following observation shows that a switch operation on a cover when chosen carefully can give a Hamiltonian path.

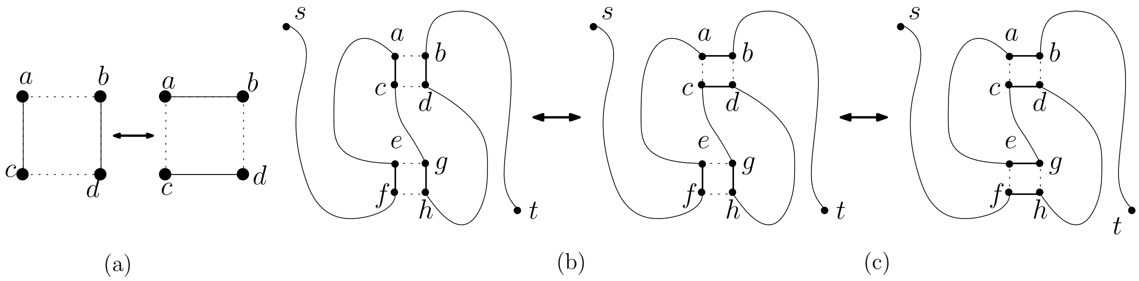


Figure 24: (a) A *switch* in a cell f of an embedded grid graph. (b) A switch in a switchable cell a, b, d, c of an *embedded* s, t Hamiltonian path P in cycle-path cover $\mathbb{P} = \{P\}$ yields new cover $\mathbb{P}' = \{P', C\}$. (c) Switching the cell e, f, h, g joins P' and C into another s, t Hamiltonian path.

Observation 1 *A switch operation in a grid cell that is switchable for s, t Hamiltonian path P of $\mathbb{G}$ gives a cycle-path cover of $\mathbb{G}$: $\mathbb{P} = \{C, P'\}$ where C is a cycle and P' is a path with ends s and t. A switch in a switchable cell of $\mathbb{P}$, where each of C and P' covers one edge of the cell, gives an s, t Hamiltonian path.*

Proof: Assume that the vertices on the switchable cell f' are a, c, d, b in counterclockwise order, and the edges (a, c) and (b, d) are on P as shown in Fig. 24. Without loss of generality, we assume that the vertices appear on P in the same order a, c, d, b. If neither of s and t is on f', then $P = P_{s,a}, (a, c), P_{c,d}, (d, b), P_{b,t}$. Removing edges (a, c) and (d, b) gives us a cycle-path cover of $\mathbb{G}$ consisting of three paths $P_{s,a}$, $P_{c,d}$ and $P_{b,t}$. Adding the edges (a, b) and (c, d) gives a cycle-path cover of $\mathbb{G}$ consisting of a single path $P' = P_{s,a}, (a, b), P_{b,t}$ and a cycle C containing the edge (c, d) on it. See Fig. 24(b).

Now assume that $s = a$; the case when $t = d$ is similar. When, $s = a$, the given s, t Hamiltonian path is $P = (s, c), P_{c,d}, (d, b), P_{b,t}$. Removing edges $(s = a, c)$ and (d, b) gives a cycle-path cover of $G - s$ consisting of two paths $P_{c,d}$, and $P_{b,t}$. Adding edges (s, b) and (c, d) gives a cycle-path cover of $\mathbb{G}$ consisting of a single path from $(s, b), P_{b,t}$ and a single cycle containing edge (c, d).

We now prove that a switch in a switchable cell f'' of $\mathbb{P}$, where each of the s, t path P' and the cycle C of $\mathbb{P}$ contribute one edge on f'', gives an s, t Hamiltonian path. Let the vertices on f'' be e, g, h, f in counterclockwise order as shown in Fig. 24(c). Without loss of generality, assume that the edges (f, e) and (g, h) are on P' and C, respectively.

First assume that none of the vertices s and t are on f''. If P' encounters f before e, then $P' = P_{s,f}, (f, e), P_{e,t}$, where edge (a, b) is either on subpath $P_{s,f}$ or on $P_{e,t}$. Removing the two edges (f, e) and (g, h) gives a cycle-path cover of $\mathbb{G}$ consisting of three paths $P_{s,f}$, $P_{e,t}$ and $P_{g,h}$. Adding the edges (e, g) and (f, h) connects the three subpaths to the s, t Hamiltonian path $P_{s,f}, (f, h), P_{h,g}, (g, e), P_{e,t}$ of $\mathbb{G}$.

Now assume that $s = a$. The case when $t = d$ is similar. Without loss of generality assume that $P' = (s = f, e), P_{e,t}$. Removing the two edges $(s = f, e)$ and (g, h) gives a cycle-path cover of

$G - s$ consisting of two paths $P_{e,t}$ and $P_{h,g}$. Adding the edges $(s = f, h)$ and (e, g) connects the two subpaths and the vertex $s = f$ into the s, t Hamiltonian path $(a = f, h), P_{h,g}, (g, e), P_{e,t}$ of $\mathbb{G}$. □

The *zone* R_Z of a zip is a rectangle determined by the zipline $l_z^{q_1 q_2}$ and the two adjacent and parallel grid lines $l_a = [a_1, a_2]$ and $l_b = [b_1, b_2]$, where a_1 and b_1 are adjacent to q_1 on a boundary of $\mathbb{G}$ and a_2 and b_2 are adjacent to q_2 on the opposite boundary. Thus the corners of R_Z are a_1, a_2, b_1, b_2. We call the subgraph of $\mathbb{G}$ induced by l_a and l_z the *main track tr* of the zip, and the subgraph with sides l_z and l_b the *side track* tr'.

To describe the structure of a Hamiltonian path P inside the zone R_Z of a zip, we define the notion of a *local cookie*. See Figs. 25 and 26.

Definition 5 (local cookie) Let f' be a switchable cell for P in the side track tr' of the zone R_Z of a zipline l_z, such that the two sides (a, d) and (b, c) of f' that belong to P are perpendicular to l_z and l_b, and the other two sides (a, b) and (d, c) lie in l_z and l_b, respectively. Let C be a cycle of grid edges in tr such that (a, b) belongs to C and is the only edge of C not in P. Then the edges (a, d) and (b, c) of f' together with the edges of C except for (a, b) determine a subpath of P called a *local cookie* with base (d, c) on l_b. Depending on its shape, a local cookie has one of four types: I, T, q_1-facing and q_2-facing. For example, C and f_2' in Fig. 26 make a type T local cookie.

The goal of a zip procedure is to create a new s, t Hamiltonian path P' that contains all of l_a and l_z as two segments in P' and joins them with a boundary edge of tr in P'. These two segments and the boundary edge that connects them can be viewed as a round trip from one boundary to the opposite boundary and back. This goal motivates the following definition.

Definition 6 *The main track tr is* locally covered *by P provided (i) P contains edge* (a_1, q_1) *or else contains a regular cookie whose base is the edge* (a_1, q_1) *– either way, P contains the edge* (q_1, b_1)*, and (ii) P covers the remaining vertices of tr with local cookies with base in* l_b *(e.g., Fig. 26 (Left)).*

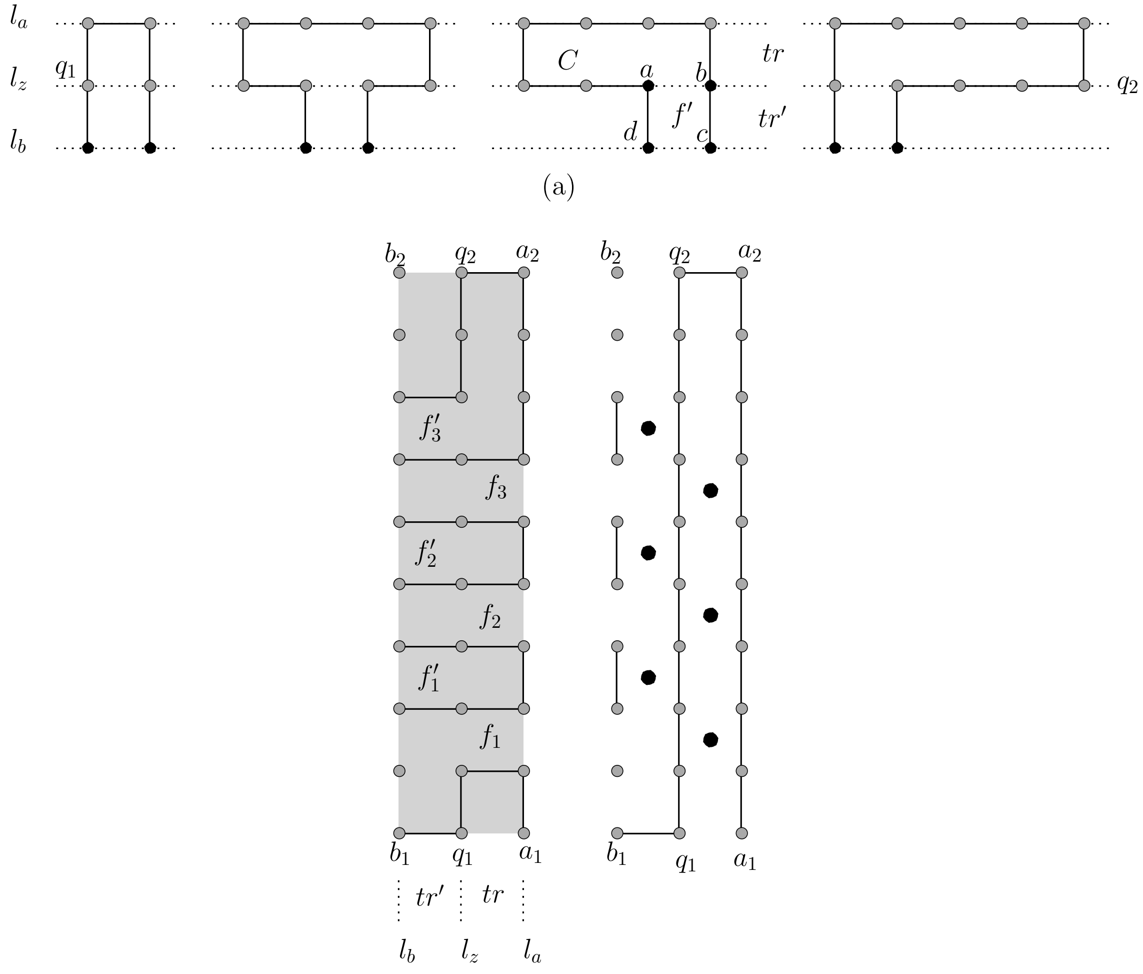
Note that here, either a_1 has degree 1 on path P and lies at a corner of $R_{\mathbb{G}}$, or else a_1 is incident to a boundary edge that belongs to P but lies outside tr.

Definition 7 *The zone* R_Z *of a zip Z is zippable provided the main track tr is locally covered by P (See Fig. 26 (Left)).*

Observation 2 (Special switchable cells) *(a) By Definition 5, each grid cell* f' *of* tr' *that contains the base of a local cookie of* R_Z *is switchable. Switching any such* f' *creates a cycle-path cover* $\mathbb{P} = \{P', C\}$ *of* $\mathbb{G}$ *where cycle C lies in tr. (b) Each grid cell f in the main track tr of a zippable zone* R_Z *that does not lie inside a local cookie is switchable for P.*

We now define two special sets of switchable cells S_{tr} in tr and $S_{tr'}$ in tr' for a zippable zone R_Z and tell how we index the cells.

The set S_{tr} consists of the following cells of tr: any cell that has one side in each of two distinct local cookies; any cell that has $a_1 q_1$ as a side where (a_1, q_1) belongs to P, and has its parallel side in a local cookie; and any cell that has as one side the end of a cookie lying in tr with base a_1, q_1, and has for a parallel side an edge in a local cookie. We index the cells of S_{tr} $f_1 \ldots$ in their order of occurrence from the q_1 end to the q_2 end of tr. We define the set $S_{tr'}$ to be all the cells in tr' that have a side on l_b that is the base of a local cookie. We index the cells of $S_{tr'}$ $f_1' \ldots$ in order of their occurrence in tr'. We note that $|S_{tr}| = |S_{tr'}|$. We index each local cookie according to the cell f_i' it encloses.



(a)

(b)

Figure 25: (a) Local cookie shapes I, T, q_1-facing, and q_2-facing. (b) A vertical zip procedure. Zip Z turns l_a and l_z into segments. *Left*: R_Z (in grey) before zip Z. Track tr is covered locally. The cells in S_{tr} and $S_{tr'}$ are labeled f_i and f'_i. *Right*: The big dots mark the switched cells after the zip. Not all details of l_b are shown in either the left or right part.

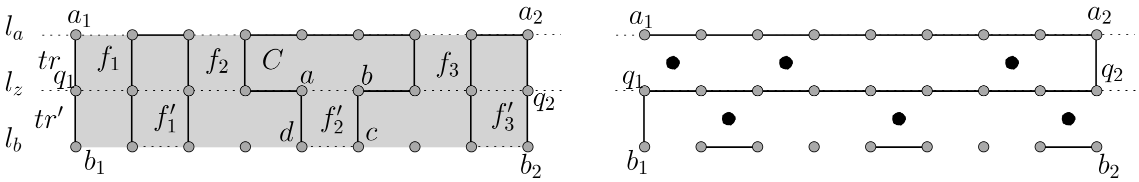


Figure 26: A horizontal zip procedure: zip Z turns l_a and l_z into segments. *Left*: R_Z (in grey) before zip Z. Dotted edges do not belong to P. Track tr is covered locally. The cells in S_{tr} and $S_{tr'}$ are labeled f_i and f'_i. *Right*: The big dots mark the switched cells after the zip. Not all details of l_b are shown in either the left or right part.

Definition 8 (Zip for zippable R_Z) *Let $l_z^{q_1,q_2}$ be a zipline of $\mathbb{G}$ whose zone R_Z is zippable, and let S_{tr} and $S_{tr'}$ be the special sets of switchable cells in the two tracks tr and tr' of R_Z. The* zip *operation $Z = zip(\mathbb{G}, P, l_z^{q_1,q_2}, tr)$ applies switch to all the cells of S_{tr} and $S_{tr'}$ in the following order: $f_1, f'_1, f_2, f'_2, \ldots$.*

Note that the zip procedure is only defined for a zippable zone R_Z. Proofs of correctness of our algorithms will show that the zips are done in zippable zones.

Observation 3 *Let P' be the s,t path resulting from a zip Z on an s,t path P of $\mathbb{G}$, where the zone R_Z is zippable. Then, the following hold: (1) Path P' is Hamiltonian and differs from P only in the cells of S_{tr} and $S_{tr'}$; (2) P' contains segments $seg[a_1, a_2]$ on l_a and $seg[q_2, q_1]$ on l_z and the boundary edge (a_2, q_2) joining their end points a_2 and q_2; (3) The boundary edge (q_1, b_1) is the only edge of P' that joins $seg[q_2, q_1]$ to l_b; and (4) P' can be obtained from P in $O(\max\{m, n\})$ switches.*

In the next section, we will use path structure (Section 3) to show that doing a zip leaves the next pair of tracks zippable. When this condition holds, we will be able to apply a sequence of zips advancing the zipline by 2 units each time. We refer to such a sequence of zips as a *sweep*.

Definition 9 (Sweep) *A sweep procedure $S = Sweep(\mathbb{G}, P, l_z^{q_1,q_2}, p, r, d)$ applies a sequence of zips with zipline $l_z^{q_1,q_2}$ starting from row/column p and ending on row/column r, where the zipline is moved 2 units in direction d after each zip except the last one. The direction d of the sweep is either left or down. For all the zips in a sweep procedure, the main track tr is on the right of l_z if $d = left$, and it is above l_z if $d = down$.*

Fig. 27 shows an example of sweep down from Row $p = 1$ to $r = m - 2$. Note that after each zip the main track is covered by two straight line segments joined by an edge on the $\mathcal{E}$ boundary.

5 Reconfiguring 1-Complex Paths to Canonical Forms

We give an algorithm called 1ComplexToCanonical to reconfigure any 1-complex s,t path P to a canonical path (see Definition 4 in Section 2). In Section 6, we will use this algorithm to design another algorithm to reconfigure between any two 1-complex paths.

We first define some terminology in order to describe the algorithm. A *sub-rectangle* $\mathbb{G}_i$ is an $m \times n_i$, $n_i \leq n$, subgrid of $\mathbb{G}$ such that the subpath P_i of P covering the vertices of $\mathbb{G}_i$ is a

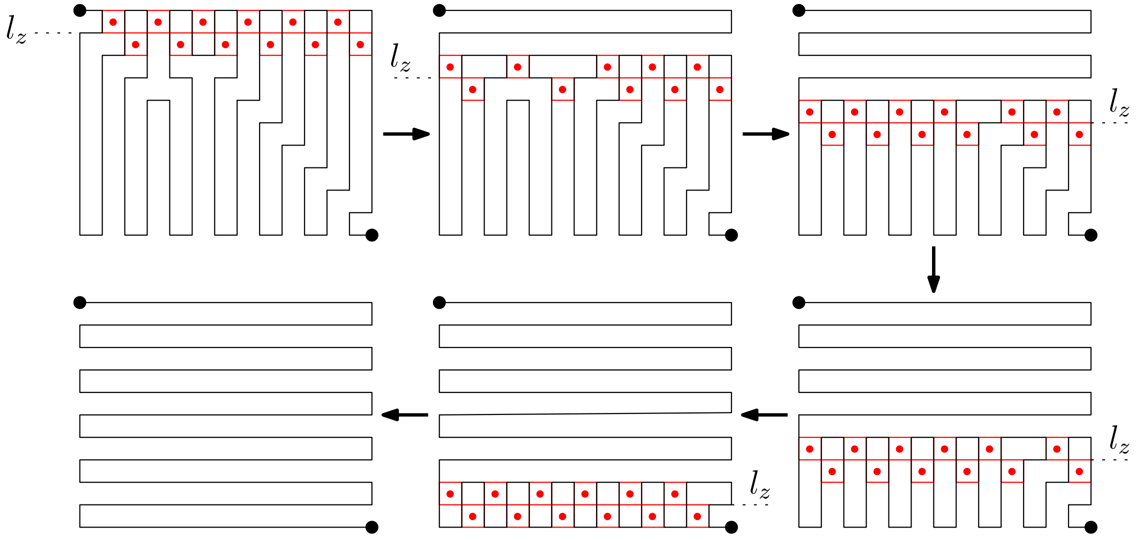


Figure 27: An example of sweep down procedure.

Hamiltonian path between two corner vertices of $\mathbb{G}_i$; we call P_i a *sub-rectangular path*. We use *sub-rectangle* and *sub-rectangular path* interchangeably for the rest of the section.

We now give a brief overview of the algorithm. It works in three steps.

1. First we break $\mathbb{G}$ into sub-rectangles using the structure of P (by Theorems 1 and 2), so that each sub-rectangle $\mathbb{G}_i$ contains an s', t' Hamiltonian path $P'_{s',t'}$ that can be reconfigured to a canonical path of $\mathbb{G}_i$ using at most two sweeps in $\mathbb{G}_i$. See Section 5.1.

2. We then reconfigure each sub-rectangle into a canonical form. Note that each sub-rectangle is incident to at least two boundaries that are opposite to each other. See Section 5.2.

3. Finally, we merge all the canonical sub-rectangular paths into an s, t Hamiltonian path of $\mathbb{G}$; and using at most one sweep in $\mathbb{G}$, we reconfigure it to a canonical path of $\mathbb{G}$. See Section 5.3.

5.1 Breaking P into (extended) sub-rectangles

We break P into sub-rectangles $\mathbb{G}_h$, $1 \leq h \leq Q$, by removing the following edges: all straight separators, and the edges on $\mathcal{N}$ and $\mathcal{S}$ preceding and following them; the edge between Columns x and $x+1$ on $\mathcal{N}$ or $\mathcal{S}$, when the internal vertices of Column x are completely covered by a separator of form D or I, respectively, or the internal vertices of Column $x+1$ are completely covered by a separator of form J or G. In the path in Fig. 3, removing the bold black edges will break the path into the $Q = 8$ sub-rectangles in Fig. 30(a). We call $\mathbb{G}_1$ and $\mathbb{G}_Q$ the *terminal* sub-rectangles, and the others the *middle* sub-rectangles.

We now define some terminology for the sub-rectangles. We will then observe some properties of the middle and terminal sub-rectangles which we use in our algorithm in the next section. Let s_h and t_h be the starting and ending points of the sub-rectangular path P_h of $\mathbb{G}_h$. If s_h is on $\mathcal{S}$, we flip $\mathbb{G}_h$ along $\mathcal{S}$ when $h < Q$. In case of $\mathbb{G}_Q$, we rotate it by π about its center. If t_h is on $\mathcal{N}$, we add a column to the east of $\mathbb{G}_h$, $1 \leq h \leq Q$, to create the *extended sub-rectangle* $\mathbb{G}'_h$, then connect t_h to the bottom-right corner t'_h of $\mathbb{G}'_h$ through the new edges to get an s_h, t'_h Hamiltonian path of $\mathbb{G}'_h$. From now on, we use $\mathbb{G}_h$ to denote the final sub-rectangle obtained after the optional flipping and/or extending steps. We observe some properties of the sub-rectangles.

Observation 4 *$\mathbb{G}_h$, $1 \leq x \leq Q$, does not contain cross-separators of form I, G, or J; if $\mathbb{G}_h$ has a separator of form D, then it must be the last cross-separator.*

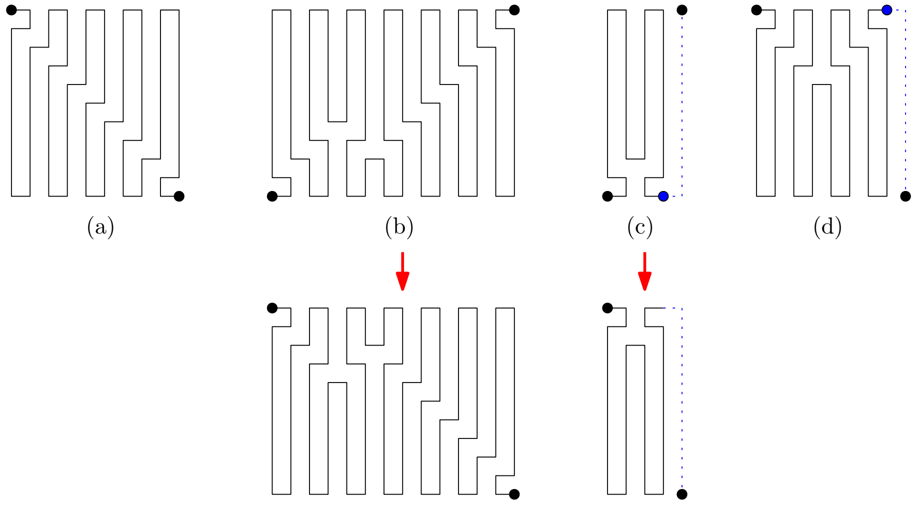


Figure 28: (a)–(b) The endpoints of the subpaths are diagonally opposite corners of the rectangle; (c) both endpoints are on the $\mathcal{S}$ boundary, (d) both endpoints are on the $\mathcal{N}$ boundary; blue dotted lines show the dummy edges added to the subpaths; (e) and (f) show the rectangles in (b) and (c), respectively, after flipping them along the $\mathcal{S}$ boundary.

We now observe some properties of the middle and terminal sub-rectangles, then describe Algorithm ReconfigSubRect, to reconfigure a sub-rectangle to a canonical form.

5.1.1 Middle sub-rectangles

As shown in Fig. 30, each middle sub-rectangle $\mathbb{G}_h$, $1 < h < Q$, must have a corner $\mathcal{W}$ cookie c of unit size. If an $\mathcal{S}$ cookie follows c, then it must be followed by a separator of form D and then the dummy $\mathcal{E}$ boundary. Otherwise, c is followed by a separator of form F, which is the start of the middle subpath of P_h.

Lemma 10 *Let P_i be a middle sub-rectangular path. Then any $\mathcal{S}$ cookie of P_i has the same parity as m and any $\mathcal{N}$ cookie has even size.*

Proof: We assume that P_i has at least one $\mathcal{S}$ or $\mathcal{N}$ cookie, as the claim trivially holds in the other case.

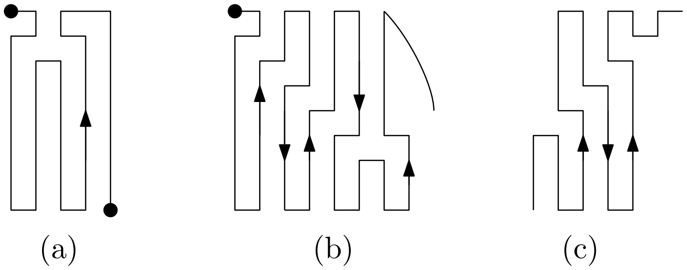


Figure 29: (a) Only form of P_i where the initial subpath has an $\mathcal{S}$ cookie; (b) otherwise, the first cookie of P_i must be of type S. (c) There are odd number of separators between an $\mathcal{S}$ and an $\mathcal{N}$ cookie.

If there is an $\mathcal{S}$ cookie in the initial subpath, then P_i must have the form shown in Fig. 29(a). The claim holds in this case as the size of the only $\mathcal{S}$ cookie is $m-2$ and there are no $\mathcal{N}$ cookie.

Now assume that there are no $\mathcal{S}$ cookie in the initial subpath (Fig. 29(b)). Then the first cookie c_1 must be an $\mathcal{S}$ cookie. Suppose for a contradiction that c_1 is an $\mathcal{N}$ cookie. Then it must be preceded by a separator of form F, which can be preceded by form M. In this way, if we trace back, we will end up at a separator of form G, which is not possible according to Observation 4; or a separator of form $\mathcal{E}$ with an $\mathcal{S}$ cookie, contradicting our claim that the first cookie had type N.

The cookie c_1 must be preceded by round trips of AH, following the unit $\mathcal{W}$ cookie. Then each such round trip will bend on an even row on the way up and then on an odd row on the way down; and the size of c_1 would be $m-x$, where x is even. Therefore, the number of rows occupied by c_1 must have the same parity as m. If there is a cookie after c_1, then it must be an $\mathcal{N}$ cookie, since c_1 must be followed by separator of form F ($\mathcal{E}$ if we consider c_1 and the separator together), and P_i does not have a separator of form J by Observation 4. Since there are an odd number of separator between c_1 and the $\mathcal{N}$ cookie, each bending on a different row above c_1, the $\mathcal{N}$ cookie occupies $x-x'$ rows, where x is even and x' is odd. Therefore, the size of the $\mathcal{N}$ cookie must be even. In this way the cookies must alternate between the $\mathcal{S}$ and $\mathcal{N}$ boundaries, since P_i does not have separators of forms G and J by Observation 4, and the number of rows occupied by any $\mathcal{S}$ cookie must have the same parity as m and the size of all the $\mathcal{N}$ cookies will be even. □

5.1.2 Terminal sub-rectangles

We limit our discussion of terminal sub-rectangles to $\mathbb{G}_1$ in this section, since $\mathbb{G}_Q$ has similar structure. If η_1 of the 1-complex path P of $\mathbb{G}$ has form B or G, then $\mathbb{G}_1$ contains only the initial subpath of P. Otherwise, $\mathbb{G}_1$ contains η_1 of P as the first cross-separator of P_1, where η_1 must have either form D, A or F. If η_1 has form D, then it must also be the last separator, and there are no corner separators or $\mathcal{N}$ cookies in $\mathbb{G}_1$ by Lemma 4. We now assume that η_1 has either form A or F and thus can have cookies in the middle subpath. We show that the size of those cookies depends on the $\mathcal{W}$ compatibility of $\mathbb{R}_s$ and $\mathbb{R}_\alpha$.

Lemma 11 *Let η_1 of $\mathbb{G}_1$ have form A or F. (a) If $\mathbb{R}_s$ is $\mathcal{W}$ compatible, then any $\mathcal{N}$ cookie in the middle subpath of $\mathbb{G}_1$ must have even size, and the size of all $\mathcal{S}$ cookies have opposite parity as m. (b) Otherwise, any $\mathcal{S}$ cookie in the middle subpath must have even size, and the sizes of the $\mathcal{N}$ cookies will have the opposite parity of m.*

Proof: By definition of $\mathcal{W}$ compatibility, if $\mathbb{R}_s$ is $\mathcal{W}$ compatible then it must contain an even number of rows. Otherwise, if $\mathbb{R}_s$ is not $\mathcal{W}$ compatible, then $\mathbb{R}_\alpha$ must be $\mathcal{W}$ compatible by Lemma 6(b), and $\mathbb{R}_\alpha$ must contain an odd number of rows.

(a) We first assume that η_1 has form A. Since $\mathbb{R}_s$ is $\mathcal{W}$ compatible, η_1 must bend on an even row. As in the proof of Lemma 10, the first cookie in the middle subpath must be an $\mathcal{S}$ cookie, and there are round trips of separators of form AH, bending on even rows on the way up and on odd rows on the way down to the $\mathcal{S}$ boundary. Therefore, the number of rows occupied by the $\mathcal{S}$ cookie must have the same parity as m. Since $\mathcal{S}$ and $\mathcal{N}$ cookies must alternate and there are odd number of separators between two consecutive $\mathcal{S}$ and $\mathcal{N}$ cookies, the size of the $\mathcal{N}$ cookies must be even.

We now assume that η_1 has form F, which must bend on an odd row. Now in this case, the first cookie in the middle subpath must be a $\mathcal{N}$ cookie, preceded by $\mathcal{N}$-$\mathcal{S}$-$\mathcal{N}$ round trips of separators of forms MF, bending on an even row on the way down and bending on an odd row on the way up. Therefore, the first $\mathcal{N}$ cookie covers $x - x'$ rows, where x is the number of rows covered by $\mathbb{R}_s$ and x' is the number of separator preceding the $\mathcal{N}$ cookie. Since x is even, and x' is odd, the $\mathcal{N}$ cookie covers an odd number of rows making its size even. Similar to the case when η_1 has form A, we can prove that $\mathcal{N}$ and $\mathcal{S}$ cookies must alternate in the middle subpath and the $\mathcal{S}$ cookies will occupy rows of the same parity as m.

(b) We first assume that η_1 has form A. Since $\mathbb{R}_s$ is not $\mathcal{W}$ compatible, η_1 must bend on an odd row. Following the same reasoning as in (a), the number of rows any $\mathcal{S}$ cookie will occupy has the opposite parity as m, and any $\mathcal{N}$ cookies in the middle subpath must have odd size. We can prove the same claim when η_1 has form F.

Now, $\mathbb{R}_s$ must be $\mathcal{W}$ compatible if m is odd, by Lemma 6. Therefore, m must be even in this case, and any $\mathcal{S}$ cookie in the middle subpath must occupy an odd number of rows; hence, they will have even size. We can prove in a similar way as in (a) that the $\mathcal{N}$ cookies will have the same parity as m. □

5.2 Reconfiguring sub-rectangles

We now describe Algorithm ReconfigSubRect. If $\mathbb{G}_h$ is a middle sub-rectangle, we apply a *SweepDown* procedure, first placing the zipline on Row 1, and moving down two rows after each zip until we reach the $\mathcal{S}$ boundary. If m is odd, we get an $\mathcal{E} - \mathcal{W}$ canonical path of $\mathbb{G}_h$ at this point. Otherwise, we will end up with unit size $\mathcal{S}$ cookies after the sweep; therefore, we *SweepLeft* to grow the $\mathcal{S}$ cookies all the way to the $\mathcal{N}$ boundary and obtain an $\mathcal{N} - \mathcal{S}$ canonical path of $\mathbb{G}_h$.

If $\mathbb{G}_1$ contains only the initial subpath of P, then we apply a *SweepLeft*, and then a *SweepDown* if we end up with unit size $\mathcal{W}$ cookies in Column 1 after the first sweep. Otherwise, depending on the $\mathcal{W}$ compatibility of η_1, we either *SweepDown* or *SweepUp* (that is, *SweepDown* after rotation), and then we *SweepLeft* if we have unit size $\mathcal{S}$ or $\mathcal{N}$ cookies, respectively, after the first sweep. We now give an algorithm to reconfigure any sub-rectangle of $\mathbb{G}$ to a canonical form based on the above property.

Algorithm *SubRectToCanonical*$(\mathbb{G}_i, P_i, m, n_i)$

Input: A sub-rectangle $\mathbb{G}_i$ of $\mathbb{G}$, the sub-rectangular path P_i of $\mathbb{G}_i$, number of rows m and number of columns n_i of $\mathbb{G}_i$.

Output: A canonical Hamiltonian path of $\mathbb{G}_i$

```
if $\mathbb{G}_i$ is a terminal sub-rectangle
        if $\mathbb{G}_i$ does not have any cross-separator of $P$
            then $P = Sweep(\mathbb{G}_i, P_i, l_z^{m-1,0}, n_i - 2, 1, left)$
```

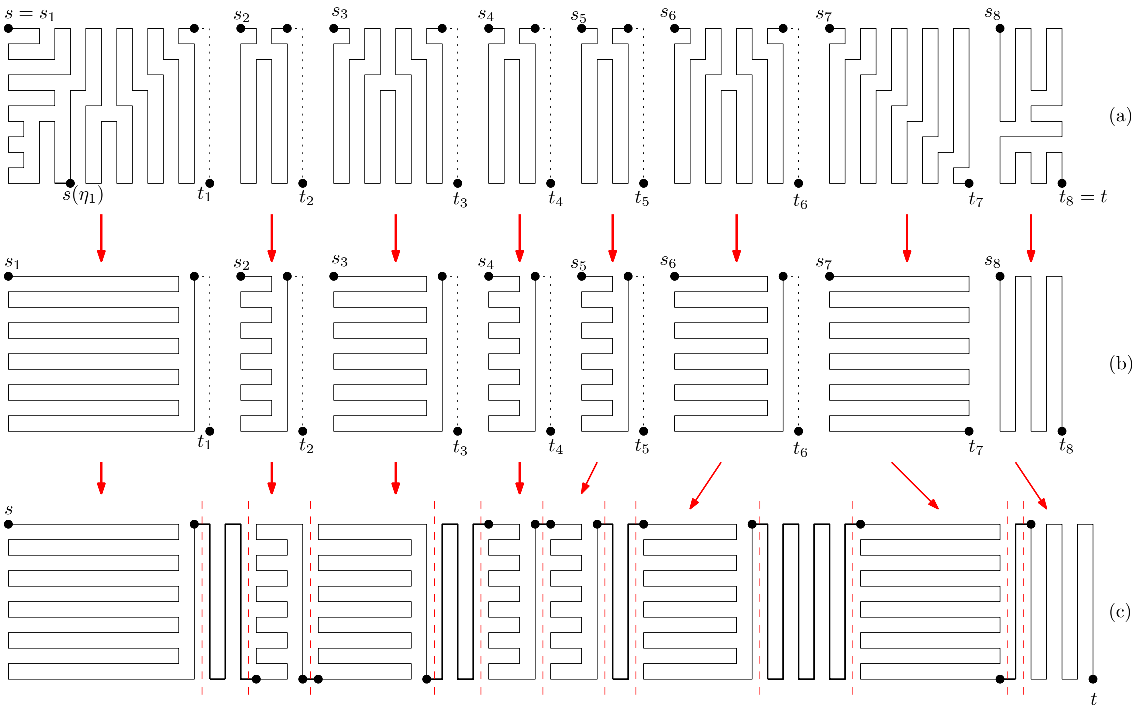


Figure 30: (a) Extended sub-rectangles of the path in Fig. 3; (b) the canonical forms for the sub-rectangles; (c) the canonical sub-rectangles merged.

```
                  if $n_i$ is even
                     then $P = Sweep(\mathbb{G}_i, P_i, l_z^{0,n_i-1}, 1, m-2, down)$
            else  if $\mathbb{R}_s$ is not $\mathcal{W}$ compatible
                     then Rotate $\mathbb{G}_i$ by 180° cw
                  $P = Sweep(\mathbb{G}_i, P_i, l_z^{0,n_i-1}, 1, m-2, down)$
                  if $m$ is even
                     then $P = Sweep(\mathbb{G}_i, P_i, l_z^{m-1,0}, n_i-2, 1, left)$
   else
         $P = Sweep(\mathbb{G}_i, P_i, l_z^{0,n_i-1}, 1, m-2, down)$
         if $m$ is even
            then $P = Sweep(\mathbb{G}_i, P_i, l_z^{m-1,0}, n_i-2, 1, left)$
return $P$
```

Lemma 12 *For each middle sub-rectangle $\mathbb{G}_h$, $1 < h < Q$, the zone of each zip in Algorithm* SubRectToCanonical *is zippable.*

Proof: Zips in Sweep Down: By Lemma 10, all the $\mathcal{N}$ cookies have even size, so the top of an $\mathcal{N}$ cookie will always be in an even row except Row 0; and all the $\mathcal{S}$ cookies have the same parity as m, so the top of an $\mathcal{S}$ cookie must also be on an even row except Row 0. Since, the line l_b is for the first zip procedure is on Row 2, and moves two rows after every zip, the top of any $\mathcal{N}$ cookie must be on the line l_b for some zip. Similarly, l_a is Row 0 for the first zip, and moves two rows down after every zip, so any $\mathcal{S}$ cookie must have its top on the line l_a for some zip.

In the first zip procedure Z_1, the main track tr_0^h contains a corner $\mathcal{W}$ cookie with base is the edge $(s = a_1, q_1)$, and the remaining vertices of the track is covered by local coolies based in l_b, where each cookie is formed by either a $\mathcal{SNS}$ round trip of the cross-separators; or a cross-separator and one side of an $\mathcal{N}$ cookie; and the last local cookie is formed by the last cross-separator and the edged on the $\mathcal{E}$ boundary. Since each cross-separator except the last one has form F, M, A, or H by Observation 4, all the local cookies will have I shape; if the last cross-separator has form D, then the last local cookie will be q_1-facing. Therefore, by Definition 6, the main track tr_0^h is locally covered from q_1 on the $\mathcal{W}$ boundary to q_2 on the $\mathcal{E}$ boundary. Hence, by Definition 7, the zone of the first zip is zippable. Set S_{tr} will have cells from the following list: a cell between the corner $\mathcal{W}$ cookie and the first cross-separator, cells covered by $\mathcal{NSN}$ round trips of cross-separators, cells inside an $\mathcal{N}$ cookie. The set $S_{tr'}$ contains cells from the following list: cells covered by $\mathcal{SNS}$ round Strips of cross-separators, cells with one side on a $\mathcal{SN}$ cross-separator and the other on an $\mathcal{N}$ cookie, cells with one side on an $\mathcal{N}$ cookie and the other side on an $\mathcal{NS}$ cross-separator, cell incident to the last cross-separator and $\mathcal{E}$ boundary.

After Z_1 is applied, by Observation 3, Rows 0 and 1 will be two segments of the s, t Hamiltonian path returned by Z_1, connected by the edge of track tr_0^h on the $\mathcal{E}$ boundary. Let Z_2 be the second zip procedure. Then the a_1, a_2 subpath in Z_2 does not go above line l_a.

In Z_2, the main track contains the edge (a_1, q_1). The local cookies will be formed in a similar way as in the first zip procedure; however, there may be other shapes of local cookies as well as I. We can have q_1-facing or q_2-facing when we reach a bend of a cross-separator, and T shaped local cookies if we reach the top of an $\mathcal{N}$ cookie, between two cross-separators. The sets S_{tr} and $S_{tr'}$ for Z_2 may contain the types of cells that were available for the same sets for Z_1; additionally S_{tr} may contains cells between a cross-separator and an $\mathcal{S}$ cookie, and $S_{tr'}$ may contain cells that are inside an $\mathcal{S}$ cookie. Therefore, the zone of Z_2 is zippable by Definitions 6 and 7. After Z_2 is applied, by Observation 3, Rows 2 and 3 will be segments on the returned s, t Hamiltonian path.

All the remaining zip procedures follow the same format as Z_2; we can prove in a similar way that the zones of each of those zip procedures is zippable, and after the zip procedure, lines l_a and l_z will be two segments of the s, t Hamiltonian path returned by the zip.

Zips in Sweep Left: If we sweep left, all the rows 0 to $m-3$ must be segments on the Hamiltonian paths, and Row $m-2$ must be covered by unit $\mathcal{S}$ cookies. Therefore, the zone of each zip procedure during the sweep will have similar zone: a cookie in the main track with base on edge (a_1, q_1), then I shaped local cookies formed by two segments of the s, t Hamiltonian path. Therefore, the zone of each such zip procedure is zippable. The set S_{tr} contains the cell between the $\mathcal{S}$ cookie and line on Row $m-3$, and cells between lines x and $x+1$, where $x < m-4$ is odd; set $S_{tr'}$ contains cells between lines x' and $x'+1$, where $x < m-3$ is even. □

Lemma 13 *For each terminal sub-rectangle $\mathbb{G}_1$ and $\mathbb{G}_Q$, the zone of each zip in Algorithm* SubRectToCanonical *is zippable.*

Proof: $\mathbb{G}_i$ has some cross-separators of P: In this case, the algorithm progresses in a way similar to Algorithm MiddleSubRectToCanonical.

First assume that $\mathbb{R}_s$ is $\mathcal{W}$ compatible. By Lemma 11, all the $\mathcal{N}$ cookies in the middle subpath have even size; by Lemma 6 all the $\mathcal{N}$ cookies in the initial subpath has even size. Therefore, the top of each $\mathcal{N}$ cookie will be in an even row except Row 0.

Since $\mathbb{R}_s$ has an even number of rows, and the size of $\mathcal{S}$ cookies in the initial subpath must have opposite parities as the number of rows in $\mathbb{R}_\alpha$ by Lemma 6(a), size of the $\mathcal{S}$ cookies have the opposite parity of m. By Lemma 11, the $\mathcal{S}$ cookies in the middle subpath have opposite parities of m. So the top of an $\mathcal{S}$ cookie must also be on an even row except Row 0. Since, the line l_b is for the first zip procedure is on Row 2, and moves two rows after every zip, the top of any $\mathcal{N}$ cookie must be on the line l_b for some zip. Similarly, l_a is Row 0 for the first zip, and moves two rows down after every zip, so any $\mathcal{S}$ cookie must have its top on the line l_a for some zip.

In each zip procedure, the edge $(s = a_1, q_1)$ in the main track is either the base of a $\mathcal{W}$ cookie (when the number of corner separators $j \geq 0$ is even), or an edge on the $\mathcal{W}$ boundary (when j is odd). The rest of the main track contains round trips of corner separators and $\mathcal{N}$ cookies by Lemma 5, then $\mathcal{SNS}$ round trips of cross-separators, and cookies nested between cross-separators forming local cookies based in l_b. Therefore, by Definition 6, the main track is locally covered from q_1 on the $\mathcal{W}$ boundary to q_2 on the $\mathcal{E}$ boundary. Hence, by Definition 7, the zone of each zip is zippable. Set S_{tr} will have cells from the following list: cell between the $\mathcal{W}$ cookie and the first cross-separator (when $j = 0$), or between the $\mathcal{W}$ cookie and the first corner separator; cells covered by $\mathcal{NSN}$ round trips of cross-separators, or $\mathcal{NWN}$ round trips of corner separators; cells inside an $\mathcal{N}$ cookie; and cell between μ_j and η_1. The set $S_{tr'}$ contains cells from the following list: cells covered by $\mathcal{SNS}$ round Strips of cross-separators, or $\mathcal{WNW}$ round trip of corner separators; cells with one side on a cross/corner separator and the other on an $\mathcal{N}$ cookie; cell incident to the last cross-separator and $\mathcal{E}$ boundary. After each zip is applied, by Observation 3, the two rows of the main track will be segments on the returned s, t Hamiltonian path.

If m is even, we apply an optional *SweepLeft*, in a similar way as in reconfiguration of a middle sub-rectangle to a canonical form.

We now assume that $\mathbb{R}_s$ is not $\mathcal{W}$ compatible. Then by Lemma 6, $\mathbb{R}_\alpha$ must be $\mathcal{W}$ compatible; and m must be even. Then by Lemma 6 all the $\mathcal{S}$ cookies in the initial subpath have even size; by Lemma 6, the size of the $\mathcal{N}$ cookies will be odd; by Lemma 11, the $\mathcal{S}$ cookies in the middle subpath must have even size and the $\mathcal{N}$ cookies will have odd size (the opposite parity of m).

After rotating, the $\mathcal{S}$ and $\mathcal{N}$ cookies in the initial subpath become $\mathcal{N}$ and cookies in the final subpath, respectively; the $\mathcal{W}$ cookies become $\mathcal{E}$ cookies; and the $\mathcal{N}$ and $\mathcal{S}$ cookies in the middle subpath exchange their cookie types. Therefore, all the $\mathcal{N}$ cookies have even size, and the $\mathcal{S}$ cookies have odd size. We can prove in a similar way as in the case of $\mathbb{R}_s$ being $\mathcal{W}$ compatible that the zones of all the zips are zippable.

$\mathbb{G}_i$ **has no cross-separator of** P**:** Then we first sweep to the left, where the first zipline is placed on Column $n-2$. If we follow the zipline from q_1 on the $\mathcal{S}$ boundary to q_2 on the $\mathcal{N}$ boundary, the main track either contains the edge (a_1, q_1) on the $\mathcal{S}$ boundary or an $\mathcal{S}$ cookie based on that edge, followed by zero or more $\mathcal{W}$ cookies; and in case of $j > 0$, the vertical segment $seg[s(\mu_j), b(\mu_j)]$ of u_j. If $j > 0$, then the zipline will contain either the segment $seg[b(\mu_{j-1}), s(\mu_{j-1})]$ or part of an $\mathcal{N}$ cookie preceding μ_j. Therefore, the main track contains I shaped local cookies formed by $\mathcal{W}$ cookies; and the last cookie may be q_2 facing if $j > 0$. All the local cookies have base on l_b. By Definition 6 and 7, the zone of the first zip Z_1 is zippable. Let P_1 be the s, t Hamiltonian path obtained by applying the first zip procedure. We now show that P_1 is 1-complex. All the $\mathcal{W}$ cookies are shortened by 2 units by Z_1. If an $\mathcal{N}$ cookie precedes μ_j, then the vertical segment of μ_j is shifted to the west segment of the $\mathcal{N}$ cookie, and the number j remains the same. In case μ_{j-1} was incident to the zipline, both μ_j and μ_{j-1} are dissolved and a $\mathcal{W}$ cookie is created by the horizontal segments of those two corner separators. Therefore, in P_1 all the cookies in the initial subpath is 1-complex, and thus, P_1 is 1-complex.

Moving l_z two columns westward, we apply the next zip on P_1 and obtain another 1-complex path. In case there are even number of columns in the sub-rectangle we have Column 1 covered by unit size $\mathcal{W}$ cookies at the end of the *SweepLeft.* We apply a *SweepDown* to grow the $\mathcal{W}$ cookies which is very similar to the optional *SweepLeft* in the previous case.

□

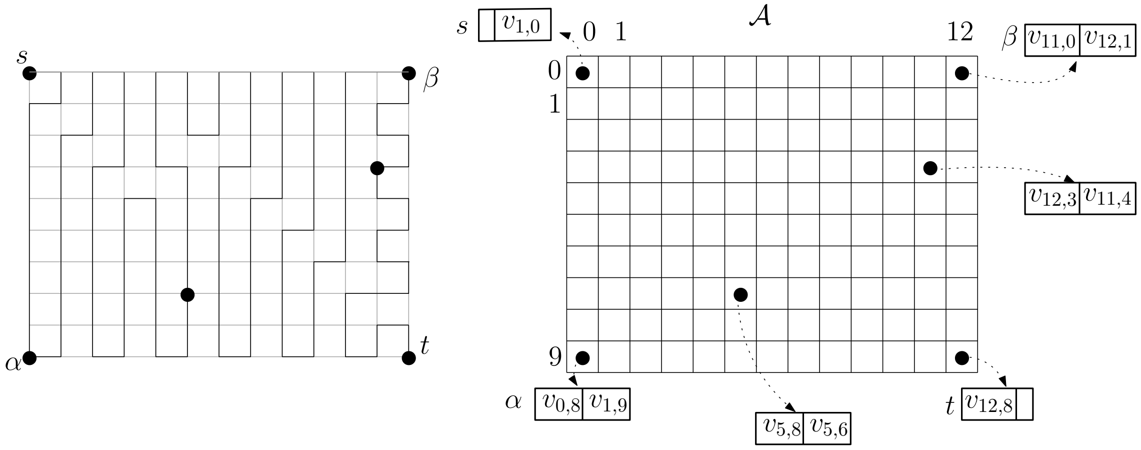
Theorem 3 *Algorithm* ReconfigSubRect *reconfigures a sub-rectangle* $\mathbb{G}_h$ *to a canonical form in* $O(|\mathbb{G}_h|)$ *switch operations.*

Proof: It follows from Lemmas 12 and 13, that the algorithm returns a canonical form. At most two sweeps are required, one downward and one leftward.

Let $m_h = m$ and n_h be the number of rows and columns, respectively, of $\mathbb{G}_h$. The downward sweep applies zips on a horizontal zipline, where the zipline is placed on the odd numbered rows only. Therefore, at most $m_h/2$ zips are performed, where each zip consists of at most $O(n_h)$ switch operations. Therefore, the downward sweep requires $O(|\mathbb{G}_h|)$ switch operations. In a similar way, the leftward sweep performs $n_h/2$ zips each of which performs $O(m_h)$ switches. Therefore, $O(|\mathbb{G}_h|)$ switch operations are required by the two sweeps in total. To show that the sweeps take $O(\mathbb{G}_h)$ time, we now show that each switch operation can be performed in $O(1)$ time using the following data structure.

We create an $m \times n$ two-dimensional array $\mathcal{A}_{\mathbb{G}}$ whose entries correspond to the vertices of $\mathbb{G}$. For each vertex $v_{i,j}$ of G, the entry $\mathcal{A}[i][j]$ stores the indices of the two neighbors of $v_{i,j}$ on P, the first one is the neighbor on the path to s and the second one is on the path to t from $v_{i,j}$. (See Fig. 31 for an example, where the dots indicate selected $v_{i,j}$'s in $\mathbb{G}$ and their corresponding entries in $\mathcal{A}$.) To check whether a cell c is switchable, we just check the entries in $\mathcal{A}$ of the four vertices on c; each of the vertices will have a neighbor on the cell and one not on the cell c. If cell c is switchable with respect to the 1-complex cycle, performing a switch in c requires changing the entries in $\mathcal{A}$ corresponding to the four vertices of c only. Therefore, determining switchability of a cell of $\mathbb{G}$ and performing a switch operation take $O(1)$ time each.

□



(a) (b)

Figure 31: (a) A 1-complex path P in a 10×13 grid graph. (b) The 10×13 2D array $\mathcal{A}$ storing P; entries corresponding to the vertices shown with dots are given.

5.3 Merging the canonical sub-rectangles

For each $\mathbb{G}_h$, if the $\mathcal{E}$ boundary is a dummy column, and m is odd, we apply one vertical zip with the zipline on Column $n_i - 2$ of $\mathbb{G}_h$ such that both Columns $n - 2$ and $n - 1$ become path segments after the zip (Fig. 30(b)). We remove the dummy edges; flip the sub-rectangle back, if it was flipped before; then add all the straight separators and edges on the $\mathcal{N}$ and $\mathcal{S}$ boundaries that were removed, to get an s, t Hamiltonian path P' of $\mathbb{G}$. If P' is not a canonical path, it must have "comb" shaped subpaths connected by straight separators as shown in Fig. 30(c). We then apply one more *SweepDown* to obtain an $\mathcal{E} - \mathcal{W}$ canonical path of $\mathbb{G}$.

```
Algorithm 1ComplexToCanonical($\mathbb{G}, P, m, n$)
Input: An $m \times n$ grid graph $\mathbb{G}$, and an 1 complex $s, t$ Hamiltonian path $P$ of $\mathbb{G}$, number of rows
    $m$ and number of columns $n$ of $\mathbb{G}$.
Output: A canonical Hamiltonian path of $\mathbb{G}$
Break $\mathbb{G}$ into $Q$ sub-rectangles $\mathbb{G}_1, \mathbb{G}_2, \ldots, \mathbb{G}_Q$
Let $P_i$ and $n_i$, where $1 \leq i \leq Q$, be the sub-rectangular path and number of columns of $\mathbb{G}_i$.
    The number of rows for all $\mathbb{G}_i$ is $m$.
for $i = 1$ to $Q$
        $P_i = SubRectToCanonical(\mathbb{G}_i, P_i, m, n_i)$
        if $\mathbb{G}_i$ has a dummy $\mathcal{E}$ boundary
           then $P_i = Sweep(\mathbb{G}_i, P_i, l_z^{m-1,0}, n_i - 2, n_i - 2, left)$
                Remove the dummy edges
        Flip back $\mathbb{G}_i$ along the $\mathcal{S}$ boundary if it was flipped before
Merge the sub-rectangles into $P'$
if $m$ is odd
   then $P' = Sweep(\mathbb{G}, P', l_z^{0,n-1}, 1, m - 2, down)$
return $P'$
```

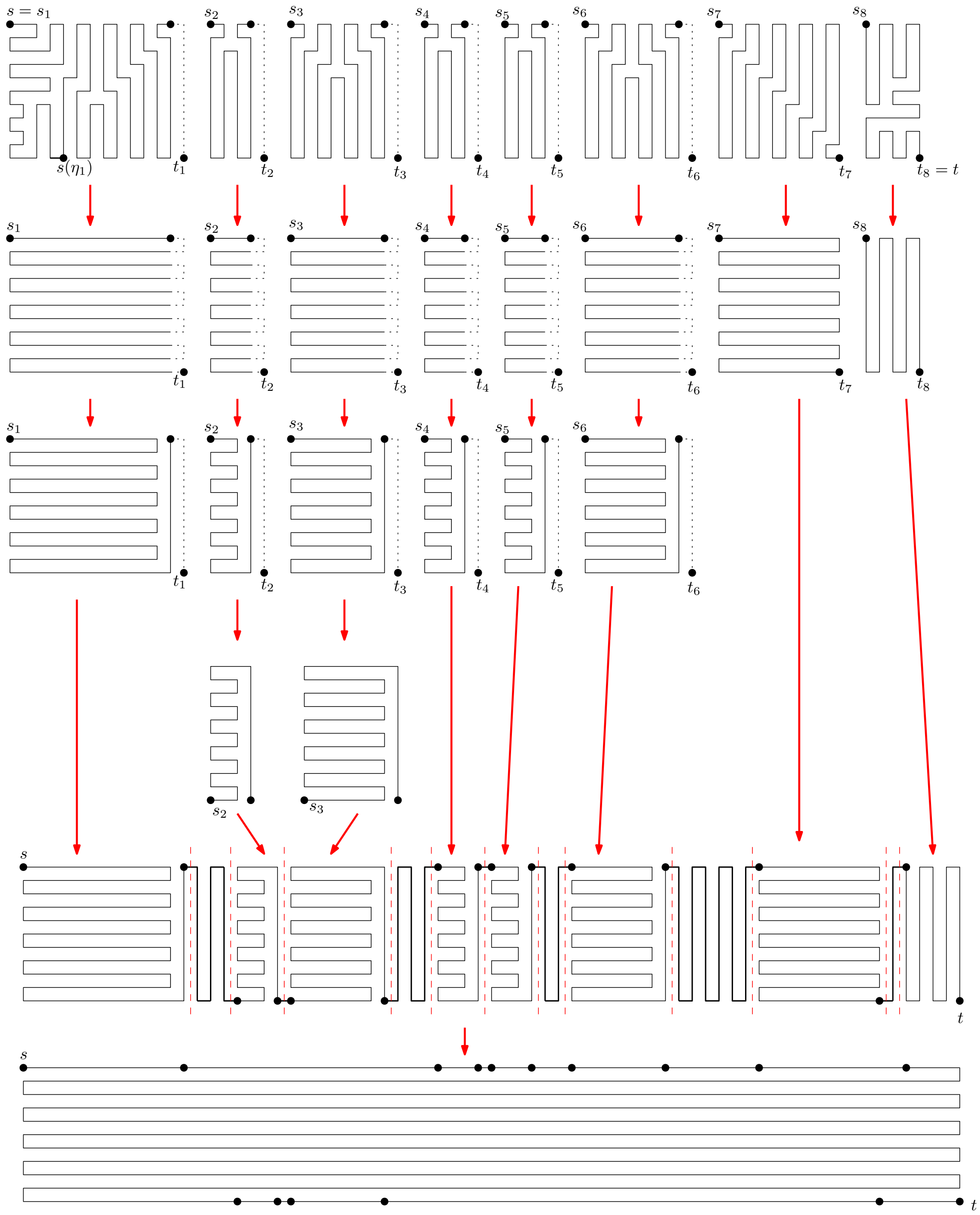


Figure 32: Reconfiguring the 8 sub-rectangles of the path in Fig. 3: $\mathbb{G}_2$ and $\mathbb{G}_3$ are flipped, and $\mathbb{G}_2$–$\mathbb{G}_6$ have been extended using dummy edges while creating the sub-rectangles. When merging the canonical forms of the sub-rectangular paths, we remove dummy edges from $\mathbb{G}_2$–$\mathbb{G}_6$, and then flip $\mathbb{G}_2$ and $\mathbb{G}_3$. In this way we find a 'comb' structure, and one final sweep gives a canonical path in the original grid $\mathbb{G}$.

Theorem 4 *Algorithm* 1ComplexToCanonical *reconfigures a* 1*-complex* s, t *Hamiltonian path to a canonical Hamiltonian path of* $\mathbb{G}$ *in* $O(|\mathbb{G}|)$ *time using* $O(|\mathbb{G}|)$ *switch operations.*

Proof: The correctness and the number of required switch operations follow from Theorem 3. Applying each switch operation takes $O(1)$ time. Therefore, applying all the switch operations will take $O(|\mathbb{G}|)$ time.

We break P into subpaths using the following algorithm. We assign s as the starting endpoint s_1 of the first subpath. If η_1 of P, occupying Column i of G, has either form B or G, then we take the vertex of Column $i-1$ on the $\mathcal{S}$ boundary to be the other endpoint t_1 of the first subpath. Otherwise, we go through the list of separators sequentially to find the first cross-separator η of form D or I, occupying Column i; then we assign the vertex on the $\mathcal{N}$ (form D) or $\mathcal{S}$ (form I) boundary on Column i to be t_1. We then find the next cross-separator η' of form G or J, occupying Column i', assign the the vertex on the $\mathcal{N}$ (form J) or $\mathcal{S}$ (form G) boundary on Column i as s_2, the starting of the next subpath; find the first cross-separator of form D or I that comes after that in order to find t_2. In this way, we keep assigning endpoints of subpaths. When passing the last cross-separator η_k of P in the list, we have two cases to consider: if we have already assigned the starting point s_Q of a subpath (the last one), and are looking for the ending point, we simply assign $t = t_Q$; otherwise, η_k must have form D or B, occupying Column i'', and we assign the vertex of Column $i''+1$ on the $\mathcal{N}$ boundary as s_Q and $t = t_Q$. Since the algorithm make one pass through the list of separators, which has size less than n, the breaking up process takes $O(n)$ time.

After assigning all the endpoints of the subpaths, we can add a dummy $\mathcal{E}$ boundary and rotate the sub-rectangles in $O(|\mathbb{G}|)$ time.

Reconfiguring any sub-rectangle $\mathbb{G}_i$, $1 \leq i \leq Q$, takes $O(|\mathbb{G}_i|)$ time by Theorem 3. It is straight forward to see that the merging of the sub-rectangles and the final sweep will take $O(|\mathbb{G}|)$ time each. Therefore, the total time required by the algorithm is $O(n) + \sum_{i=1}^{Q} O(|\mathbb{G}_i|) + O(|\mathbb{G}|) + O(|\mathbb{G}|) = O(|\mathbb{G}|)$. □

6 Reconfiguring between 1-Complex Paths

In this section, we give an algorithm called 1ComplexTo1Complex to reconfigure between any two 1-complex s, t Hamiltonian paths P_1 and P_2 in $O(|\mathbb{G}|)$ time (see Definition 4 in Section 2). Our strategy is to use two *canonical Hamiltonian paths* $\mathbb{P}_1$ and $\mathbb{P}_2$ as intermediate paths, where the two canonical paths may or may not be the same based on the parity of m and n. Recall from Definition 4 and Fig. 7 that a canonical path is a 1-complex path with no bends at internal vertices.

The reconfiguration sequence is as follows: (a) P_1 to $\mathbb{P}_1$, (b) $\mathbb{P}_1$ to $\mathbb{P}_2$ if they are different, and finally (c) $\mathbb{P}_2$ to P_2. Algorithm 1ComplexToCanonical suffices for Steps (a) and (c), since reconfiguring $\mathbb{P}_2$ to P_2 is similar to reversing the steps of the reconfiguration of P_2 to $\mathbb{P}_2$.

To check whether $\mathbb{P}_1$ and $\mathbb{P}_2$ are different canonical paths, we check the first edge on each path. If the edge is on the $\mathcal{W}$ boundary then the path is an $\mathcal{N}$-$\mathcal{S}$ canonical path; otherwise, it is an $\mathcal{W}$-$\mathcal{E}$ canonical path. If $\mathbb{P}_1$ and $\mathbb{P}_2$ are the different canonical paths, then we call Algorithm CanonicalToCanonical to reconfigure one canonical Hamiltonian path to the other.

We now describe Algorithm CanonicalToCanonical. If $\mathbb{P}_1$ is $\mathcal{N}$-$\mathcal{S}$ and $\mathbb{P}_2$ is $\mathcal{W}$-$\mathcal{E}$, we apply a *SweepDown* procedure on $\mathbb{P}_1$ starting from Row 1 and ending on Row $m-2$, as shown in Fig. 33.

In the remaining case, when $\mathbb{P}_1$ is $\mathcal{W}$-$\mathcal{E}$ and $\mathbb{P}_2$ is $\mathcal{N}$-$\mathcal{S}$, we apply *SweepLeft* on $\mathbb{P}_1$ with the zipline sweeping from Column 1 to Column $n-2$. To conclude,

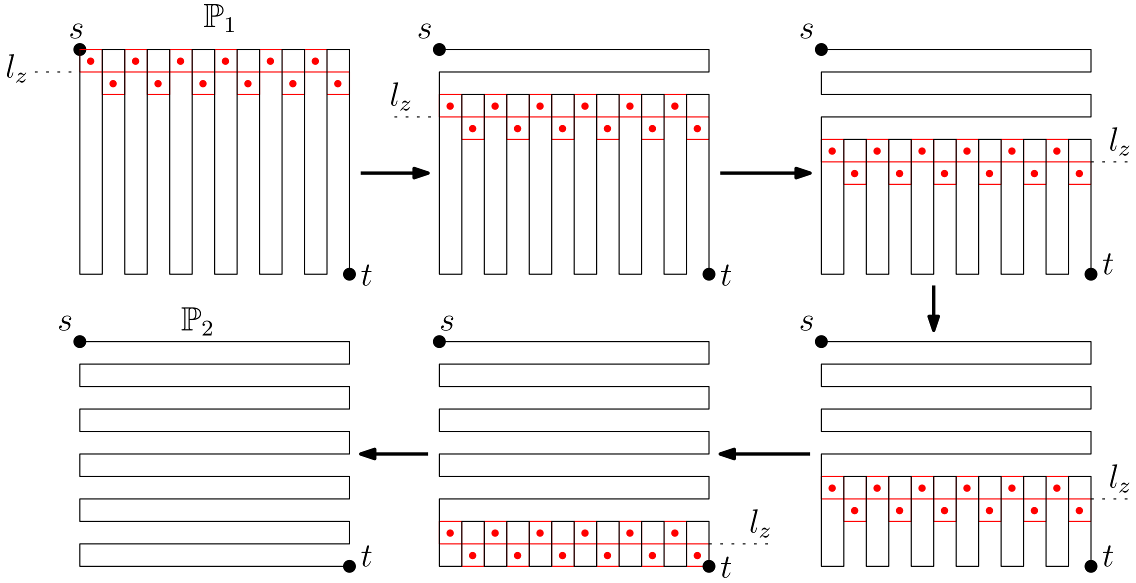


Figure 33: SweepDown procedure to reconfigure $\mathcal{N}$–$\mathcal{S}$ canonical path $\mathbb{P}_1$ to $\mathcal{E}$–$\mathcal{W}$ canonical path $\mathbb{P}_2$.

Theorem 5 *Suppose* $\mathbb{G}$ *has two distinct canonical paths* $\mathbb{P}_1$ *and* $\mathbb{P}_2$*. Then Algorithm* CanonicalToCanonical *reconfigures* $\mathbb{P}_1$ *to* $\mathbb{P}_2$ *in* $O(|\mathbb{G}|)$ *time using* $O(|\mathbb{G}|)$ *switches.*

Proof: Each of *Sweep Down* and *Sweep Left* procedures requires $O(|\mathbb{G}|)$ switch operations. Since applying each switch operation takes $O(1)$ time, applying all the switch operations will take $O(|\mathbb{G}|)$ time. □

We observe that reconfiguring between two distinct canonical Hamiltonian paths requires at least $|P|/2$ switch operations as each switch operation switches only two path edges. This gives a worst case lower bound on the number of switches required in reconfiguration between two 1-complex s, t paths of $\mathbb{G}$.

Theorem 6 *Reconfiguring between two distinct canonical paths* $\mathbb{P}_1$ *and* $\mathbb{P}_2$ *requires* $\Omega(|\mathbb{G}|)$ *switch operations.*

We summarize our main algorithmic result, based on Theorems 4 and 5.

Theorem 7 *[**Main algorithmic result.**] Let* P_1 *and* P_2 *be two* 1*-complex* s, t *Hamiltonian paths of a grid graph* $\mathbb{G}$*. Then Algorithm* 1Complex To1Complex *reconfigures* P_1 *to* P_2 *in* $O(|\mathbb{G}|)$ *time, using* $O(|\mathbb{G}|)$ *switches in total.*

Proof: Each of Steps (a) and (c) requires $O(|\mathbb{G}|)$ time by Theorem 4; Step (b) requires $O(|\mathbb{G}|)$ time by Theorem 5. □

7 Conclusion

We established the structure of any 1-complex s, t Hamiltonian path in $\mathbb{G}$. We gave an $O(|\mathbb{G}|)$-time algorithm to reconfigure between any two given 1-complex paths of $\mathbb{G}$ using *switches* in grid cells, where $|\mathbb{G}|$ denotes the *size* of the grid graph $\mathbb{G}$ in terms of the total number of vertices mn of $\mathbb{G}$. It would be interesting to find an algorithm that keeps the s, t Hamiltonian paths in the intermediate steps 1-complex.

Generalization: k-complex paths. A *k-complex path P* is an s, t Hamiltonian path of $\mathbb{G}$ where each vertex of $\mathbb{G}$ is connected to a vertex on the boundary rectangle $\mathcal{R}_{\mathbb{G}}$ by at most k straight line segments on P. We leave the following questions open.

- Is the Hamiltonian path graph for k-complex paths of $\mathbb{G}$ connected for switch operation? What is the diameter of that graph?
- For general grid graphs, deciding whether a Hamiltonian path exist between two given vertices s and t is NP-complete. What is the complexity of deciding whether there is a reconfiguration sequence between two given s, t Hamiltonian paths in such a grid graph?

The reconfiguration problem remains open for grid graphs with arbitrary boundary, and in d-dimension, $d \geq 3$.

We conclude with a problem suggested by one of our anonymous reviewers. Let $p \geq 1$ be the maximum length of path segment that connects two internal bends of an s, t Hamiltonian path. For 1-complex paths, $p = 1$. Can we reconfigure s, t Hamiltonian paths with parameter $p > 1$ using switch operations?

Acknowledgments

We thank the anonymous reviewers for their useful feedback and suggestions, which helped us tremendously. Much of the work by the first author was done while she was a postdoctoral fellow at the Toronto Metropolitan University, Toronto, Canada.

References

[1] F. Afrati. The Hamilton circuit problem on grids. *RAIRO - Theoretical Informatics and Applications - Informatique Theorique et Applications*, 28(6):567–582, 1994. URL: http://www.numdam.org/item/ITA_1994__28_6_567_0/.

[2] H. A. Akitaya, E. D. Demaine, M. Korman, I. Kostitsyna, I. Parada, W. Sonke, B. Speckmann, R. Uehara, and J. Wulms. Compacting squares: Input-sensitive in-place reconfiguration of sliding squares. In A. Czumaj and Q. Xin, editors, *18th Scandinavian Symposium and Workshops on Algorithm Theory, SWAT 2022, June 27-29, 2022, Tórshavn, Faroe Islands*, volume 227 of *LIPIcs*, pages 4:1–4:19. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2022. doi:10.4230/LIPIcs.SWAT.2022.4.

[3] E. M. Arkin, M. A. Bender, E. D. Demaine, S. P. Fekete, J. S. B. Mitchell, and S. Sethia. Optimal covering tours with turn costs. *SIAM J Comput.*, 35(3):531–566, 2005. doi:10.1137/S0097539703434267.

[4] E. M. Arkin, S. P. Fekete, K. Islam, H. Meijer, J. S. Mitchell, Y. N. Rodríguez, V. Polishchuk, D. Rappaport, and H. Xiao. Not being (super)thin or solid is hard: A study of grid Hamiltonicity. *Computational Geometry*, 42(6-7):582–605, 2009. doi:10.1016/j.comgeo.2008.11.004.

[5] A. Bedel, Y. Coudert-Osmont, J. Martínez, R. I. Nishat, S. Whitesides, and S. Lefebvre. Closed space-filling curves with controlled orientation for 3d printing. In *Computer Graphics Forum*, volume 41, pages 473–492. Wiley Online Library, 2022. doi:10.1111/cgf.14488.

[6] O. Bodroža Pantić, B. Pantić, I. Pantić, and M. Bodroža Solarov. Enumeration of Hamiltonian cycles in some grid graphs. *MATCH Communications in Mathematical and in Computer Chemistry*, 70:181–204, 2013.

[7] P. Bose. Flips. In W. Didimo and M. Patrignani, editors, *Graph Drawing - 20th Int. Symp., GD 2012*, volume 7704 of *Lecture Notes in Computer Science*, page 1. Springer, 2012. doi:10.1007/978-3-642-36763-2_1.

[8] M. Bousquet-Mélou, A. J. Guttmann, and I. Jensen. Self-avoiding walks crossing a square. *Journal of Physics A: Mathematical and General*, 38(42):9159, 2005. doi:10.1088/0305-4470/38/42/001.

[9] G. Călinescu, A. Dumitrescu, and J. Pach. Reconfigurations in graphs and grids. *SIAM J. Discret. Math.*, 22(1):124–138, 2008. doi:10.1137/060652063.

[10] S. D. Chen, H. Shen, and R. Topor. An efficient algorithm for constructing hamiltonian paths in meshes. *Parallel Computing*, 28(9):1293–1305, 2002. doi:10.1016/S0167-8191(02)00135-7.

[11] K. L. Collins and L. B. Krompart. The number of Hamiltonian paths in a rectangular grid. *Discrete Mathematics*, 169(1–3):29–38, 1997. doi:10.1016/0012-365X(95)00330-Y.

[12] A. R. Conway, I. G. Enting, and A. J. Guttmann. Algebraic techniques for enumerating self-avoiding walks on the square lattice. *J Phys A Math Gen*, 26(7):1519, 1993. doi:10.1088/0305-4470/26/7/012.

[13] H. Everett. Hamiltonian paths in nonrectangular grid graphs. Master's thesis, University of Saskatchewan, Canada, 1986.

[14] M. Fellows, P. Giannopoulos, C. Knauer, C. Paul, F. A. Rosamond, S. Whitesides, and N. Yu. Milling a graph with turn costs: A parameterized complexity perspective. In *WG 2010*, volume 6410 of *LNCS*, pages 123–134. Springer, 2010. doi:10.1007/978-3-642-16926-7_13.

[15] F. Göbel. On the number of hamilton cycles in product graphs. *Technische Hogeschool Twente*, 1979.

[16] J. E. Goodman and J. O'Rourke, editors. *Handbook of Discrete and Computational Geometry, Second Edition*. Chapman and Hall/CRC, 2004. doi:10.1201/9781420035315.

[17] A. Gorbenko, V. Popov, and A. Sheka. Localization on discrete grid graphs. In *CICA 2011*, pages 971–978, Dordrecht, 2012. Springer Netherlands. doi:10.1007/978-94-007-1839-5_105.

[18] V. S. Gordon, Y. L. Orlovich, and F. Werner. Hamiltonian properties of triangular grid graphs. *Discrete Mathematics*, 308(24):6166–6188, 2008. doi:10.1016/j.disc.2007.11.040.

[19] R. A. Hearn and E. D. Demaine. Pspace-completeness of sliding-block puzzles and other problems through the nondeterministic constraint logic model of computation. *Theor. Comput. Sci.*, 343(1-2):72–96, oct 2005. doi:10.1016/j.tcs.2005.05.008.

[20] D. A. Hoang, A. Khorramian, and R. Uehara. Shortest reconfiguration sequence for sliding tokens on spiders. In P. Heggernes, editor, *Algorithms and Complexity - 11th International Conference, CIAC 2019, Rome, Italy, May 27-29, 2019, Proceedings*, volume 11485 of *Lecture Notes in Computer Science*, pages 262–273. Springer, 2019. doi:10.1007/978-3-030-17402-6_22.

[21] K. Islam, H. Meijer, Y. N. Rodríguez, D. Rappaport, and H. Xiao. Hamilton circuits in hexagonal grid graphs. In *Proceedings of the 19th Ann. Canadian Conf. on Computational Geometry, CCCG 2007, August 20-22, 2007, Carleton University, Ottawa, Canada*, pages 85–88, 2007. URL: http://cccg.ca/proceedings/2007/04a2.pdf.

[22] A. Itai, C. H. Papadimitriou, and J. L. Szwarcfiter. Hamilton paths in grid graphs. *SIAM J Comput.*, 11(4):676–686, 1982. doi:10.1137/0211056.

[23] H. Iwashita, Y. Nakazawa, J. Kawahara, T. Uno, and S. ichi Minato. Efficient computation of the number of paths in a grid graph with minimal perfect hash functions. Technical Report TCS-TR-A-13-64, Division of Computer Science, Hokkaido University, April 2013.

[24] J. L. Jacobsen. Exact enumeration of Hamiltonian circuits, walks and chains in two and three dimensions. *J. of Phys. A: Math. Theor.*, 40:14667–14678, 2007. doi:10.1088/1751-8113/40/49/003.

[25] F. Keshavarz-Kohjerdi and A. Bagheri. Hamiltonian paths in L-shaped grid graphs. *Theoretical Computer Science*, 621:37–56, 2016. doi:10.1016/j.tcs.2016.01.024.

[26] F. Keshavarz-Kohjerdi and A. Bagheri. Finding hamiltonian cycles of truncated rectangular grid graphs in linear time. *Appl. Math. Comput.*, 436:127513, 2023. doi:10.1016/j.amc.2022.127513.

[27] F. Keshavarz-Kohjerdi and R. Hung. Finding hamiltonian and longest (s, t)-paths of c-shaped supergrid graphs in linear time. *Algorithms*, 15(2):61, 2022. doi:10.3390/a15020061.

[28] D. E. Knuth. *The Art of Computer Programming, Volume 4, Fascicle 1.* Addison-Wesley Professional: Boston, MA, USA, 2009.

[29] Y. H. H. Kwong and D. G. Rogers. A matrix method for counting hamiltonian cycles on grid graphs. *European Journal of Combinatorics*, 15(3):277—-283, 1994. doi:10.1006/eujc.1994.1031.

[30] I. Lignos. *Reconfigurations of Combinatorial Problems: Graph Colouring and Hamiltonian Cycle*. PhD thesis, Durham University, 2017.

[31] P. Muller, J.-Y. Hascoet, and P. Mognol. Toolpaths for additive manufacturing of functionally graded materials (FGM) parts. *Rapid Prototyping Journal*, 20(6):511–522, 2014. doi:10.1108/RPJ-01-2013-0011.

[32] R. I. Nishat. Map folding. Master's thesis, University of Victoria, Canada, 2013.

[33] R. I. Nishat. *Reconfiguration of Hamiltonian Cycles and Paths in Grid Graphs.* PhD thesis, University of Victoria, Canada, 2020.

[34] R. I. Nishat, V. Srinivasan, and S. Whitesides. Reconfiguring simple s,t Hamiltonian paths in rectangular grid graphs. In *IWOCA 2021*, volume 12757 of *LNCS*, pages 501–515, 2021. doi:10.1007/978-3-030-79987-8_35.

[35] R. I. Nishat, V. Srinivasan, and S. Whitesides. 1-complex s, t hamiltonian paths: Structure and reconfiguration in rectangular grids. In *WALCOM 2022*, volume 13174 of *LNCS*, pages 59–70, 2022. doi:10.1007/978-3-030-96731-4_6.

[36] R. I. Nishat and S. Whitesides. Bend complexity and Hamiltonian cycles in grid graphs. In *COCOON 2017*, volume 10392 of *LNCS*, pages 445–456, 2017. doi:10.1007/978-3-319-62389-4_37.

[37] R. I. Nishat and S. Whitesides. Reconfiguring Hamiltonian cycles in l-shaped grid graphs. In *WG 2019*, volume 11789 of *LNCS*, pages 325–337, 2019. doi:10.1007/978-3-030-30786-8_25.

[38] N. Nishimura. Introduction to reconfiguration. *Algorithms*, 11(4):52, 2018. doi:10.3390/a11040052.

[39] T. Nishizeki and M. S. Rahman. *Planar Graph Drawing*, volume 12 of *Lecture Notes Series on Computing*. World Scientific, 2004. doi:10.1142/5648.

[40] T. Nishizeki and M. S. Rahman. Rectangular drawing algorithms. In R. Tamassia, editor, *Handbook on Graph Drawing and Visualization*, pages 317–348. Chapman and Hall/CRC, 2013.

[41] V. Pettersson. Enumerating Hamiltonian cycles. *The Electronic Journal of Combinatorics*, 21(4):P4.7, 2014. doi:10.37236/4510.

[42] J. R. Reay and T. Zamfirescu. Hamiltonian cycles in T-graphs. *Discrete & Computational Geometry*, 24(2-3):497–502, 2000. doi:10.1007/s004540010051.

[43] F. Ruskey and J. Sawada. Bent hamilton cycles in d-dimensional grid graphs. *the electronic journal of combinatorics*, 10(1):R1, 2003. doi:10.37236/1694.

[44] J. Sack and J. Urrutia, editors. *Handbook of Computational Geometry.* North Holland / Elsevier, 2000. doi:10.1016/b978-0-444-82537-7.x5000-1.

[45] W. Schnyder. Embedding planar graphs on the grid. In D. S. Johnson, editor, *Proceedings of the First Annual ACM-SIAM Symposium on Discrete Algorithms, 22-24 January 1990, San Francisco, California, USA*, pages 138–148. SIAM, 1990.

[46] R. E. Stoyan and V. Strehl. Enumeration of hamiltonian circuits in rectangular grids. *Journal of Combinatorial Mathematics and Combinatorial Computing*, 1996.

[47] A. Takaoka. Complexity of Hamiltonian cycle reconfiguration. *Algorithms*, 11(9):140, 2018. doi:10.3390/a11090140.

[48] C. Umans and W. Lenhart. Hamiltonian cycles in solid grid graphs. In *38th Ann. Symp. on Foundations of Computer Science, FOCS '97*, pages 496–505, 1997. doi:10.1109/SFCS.1997.646138.

[49] C. M. Umans. An Algorithm for Finding Hamiltonian Cycles in Grid Graphs Without Holes. Bachelor of Arts with honors thesis, Williams College, Williamstown, Massachusetts, USA, 1996.

[50] S. Winter. Modeling costs of turns in route planning. *Geoinformatica*, 6(4):345–361, Dec. 2002. doi:10.1023/A:1020853410145.

[51] T. Yamada and R. Uehara. Shortest reconfiguration of sliding tokens on subclasses of interval graphs. *Theor. Comput. Sci.*, 863:53–68, 2021. doi:10.1016/j.tcs.2021.02.019.

[52] C. Zamfirescu and T. Zamfirescu. Hamiltonian properties of grid graphs. *SIAM J. Discrete Math.*, 5(4):564–570, 1992. doi:10.1137/0405046.