

1 Algorithms

Kant and He [KH] were the first to design algorithms that determine a regular edge labeling.

Fusy [Fusy2006, Fusy2009] recently developed a different algorithm computing a specific regular edge labeling using a method shrinking a sweepcycle while coloring the outside in accordance with a regular edge labeling.¹

All algorithms in this section will have the same core (based on [Fusy2006]). Consisting of shrinking a sweepcycle by so-called *valid* paths². But will differ in which valid paths they choose (if there are multiple).

We will start this section with some notation and preliminaries in Subsection 1.1. Then we will state the core algorithm and show that it always computes a regular edge labeling in Subsection 1.2. Afterwards we show in Subsections 1.3, 1.4, 1.5. How one can adapt the choice of the valid paths to obtain regular edge labellings with certain properties.

FiXme: what properties

1.1 Notation and Preliminaries

Definition (Interior path). We call a path P an internal path of a cycle C if all its edges are in the interior of C and it connects two distinct vertices of C

FiXme: is interior defined somewhere?

We will use a script \mathcal{C} to indicate the current sweep cycle. We will repeatedly only consider the path $\mathcal{C} \setminus \{S\}$. In that case we will always order it from W to E.

We will let \mathcal{P} denote a interior path. Given such a path of k vertices we will index it's nodes by p_1, \dots, p_k in such a way that p_1 is closer to W then p_k is (and thus that p_k is closer to E then p_1 is).

Then p_1 and p_k indicate the two unique vertices of the walk that are also part of the cycle. We will then let $\mathcal{C}|_{\mathcal{P}}$ denote the part of $\mathcal{C} \setminus \{S\}$ that is between p_1 and p_k (including). $\mathcal{C}_{\mathcal{P}}$ will denote the cycle we get when we paste $\mathcal{C}|_{\mathcal{P}}$ and \mathcal{P} .

1.2 Core

The algorithm will always maintain the following three invariants

Invariants 1

- (I1) The cycle \mathcal{C} contains the two edges SW and SE.
- (I2) $\mathcal{C} \setminus \{S\}$ has no chords in it's interior
- (I3) All inner edges of T outside of \mathcal{C} are colored and oriented in such that the innnervetex condition holds.

FiXme: def chord and chord in interior (or just simplify to has no chords

A cycle satisfying these three invariants will have the same general shape as in figure 1. We note that the cycle has at least 4 vertices because otherwise a separating triangle is created.

It is also nice to note that the union of the cycle and it's interior form a triangulation of the n -gon since it is a induced subgraph of a triangulation of the 4-gon.

¹The specific regular edge labelling Fusy obtained was the minimal element of the lattice of regular edge labellings.

²In Fusy's work these are called *eligible paths*

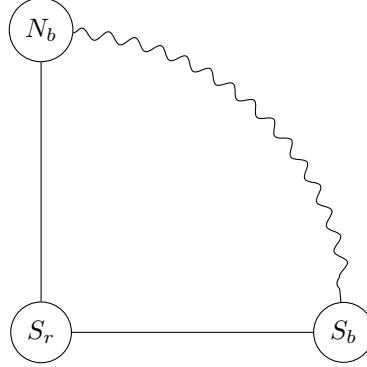


Figure 1: An example of a cycle \mathcal{C} satisfying the invariants

1.2.1 Eligible paths

Definition (valid path). We call an internal path \mathcal{P} from w_1 to w_k valid if

- (E1) Neither w_1 or w_k is S
- (E2) The paths \mathcal{P} and $\mathcal{C}|_{\mathcal{P}}$ both are more than 1 edge ³
- (E3) Each edge in the interior to $\mathcal{C}_{\mathcal{P}}$ connects a vertex of $\mathcal{P} \setminus \{w_1, w_k\}$ and $\mathcal{C}|_{\mathcal{W}} \setminus \{w_1, w_k\}$. In particular $\mathcal{C}_{\mathcal{P}}$ is a non-separating cycle.
- (E4) The cycle \mathcal{C}' obtained by replacing $\mathcal{C}|_{\mathcal{P}}$ by \mathcal{P} in \mathcal{C} has no chord that doesn't involve S.

Remark 2. “Shrinking” the cycle with an eligible path will keep all the invariants true.

We will show the following proposition.

Theorem 3 (Existence of a eligible path). *When the algorithm’s invariant (1 (I1) - 1 (I3)) are satisfied and the cycle \mathcal{C} is separating then there exist a eligible internal path.*

Proof. We will first show that there always exists an internal path \mathcal{P} . We will then show that a internal path can be found that satisfies conditions (E1)–(E4).

In the proof we will often use that a

Let us first note that if the cycle \mathcal{C} is separating (i.e has a non-empty interior), there is at least one interior vertex v . Since the triangulation of a n -gon is 2-connected there are two ways to go from v to (say) S_r . Hence there is an internal path \mathcal{P}_0 .

If this path does not satisfy (E1) we can use the following construction. The other vertex where \mathcal{P}_0 intersects \mathcal{C} is not S_r . Let us call this vertex x and it’s neighbour on the path y . The vertex x might be N_b or S_b but can’t be both, hence it has at least one neighbour z on the cycle that is not S_r . Because the triangulation of a n -gon is internally maximally planar we have that yz is an edge. Now xyz is an internal path satisfying (E1). See also figure 2, here we

³i.e. both have an interior vertex

FiXme: This is very hard to prove, mostly the part that we can always find a path satisfying E4. See page 11. Prrof this from red algo.

made a choice on which side of y the vertex z lies, but this choice can be made without losing generality.

Hence we have now constructed, or already had, a path that satisfies (E1). Let us for the remainder of the proof denote this path by \mathcal{P}_1 .

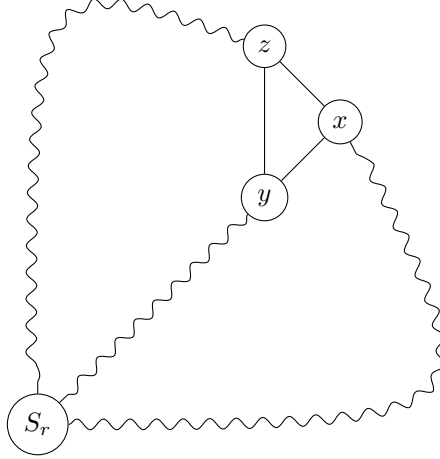


Figure 2: Constructing a path satisfying (E1)

There is a path that also satisfies (E2) If \mathcal{P}_1 satisfies (E2) we set $\mathcal{P}_2 = \mathcal{P}_1$ otherwise we will create a path that satisfies (E1) and (E2). If the path \mathcal{P}_1 does not satisfy (E2)⁴ then there are two possibilities a) \mathcal{P}_1 does not have interior vertices and/or b) $[v, v']$ does not have interior vertices. If a) would be true the existence of P_0 would contradict Invariant 1 (I2). Hence the only problem can be that b) occurs.

If $v = N_b$ and $v' = S_b$ we have found a separating triangle given by $S_r N_b S_b$ ⁵ in original graph. Hence at least one of v or v' is not N_b or S_b . If we call this vertex x its neighbour on the path y and it's neighbour outside $[v, v']$ z . We see that by the interior of \mathcal{C} being maximally planar yz must be an edge. If we now adapt \mathcal{P}_1 by replacing yx by yz we have made $[v, v']$ one vertex longer and hence created a path satisfying (E2). In figure 3 we show this procedure in two cases. Executing this procedure does not change that S_r is not one of the endpoints of the path. Hence we have now created a path \mathcal{P}_2 that satisfies (E1) and (E2).

There is a path that also satisfies (E3) If \mathcal{P}_2 satisfies (E3), we take $\mathcal{P}_3 = \mathcal{P}_2$. Otherwise we will remedy the defect. We separate five different cases of offending edges. All of the five cases will be easy to remedy giving a path \mathcal{P}'_2 still satisfying (E1) and (E2) such that $\mathcal{C}_{\mathcal{P}'_2}$ is strictly contained in $\mathcal{C}_{\mathcal{P}_2}$

- a) edges from $[v, v'] \setminus v, v'$ to $[v, v'] \setminus v, v'$
- b) edges from $\mathcal{P} \setminus v, v'$ to $\mathcal{P} \setminus v, v'$

⁴which will be the case if the above construction has been used

⁵this is the cycle \mathcal{C} which is separating

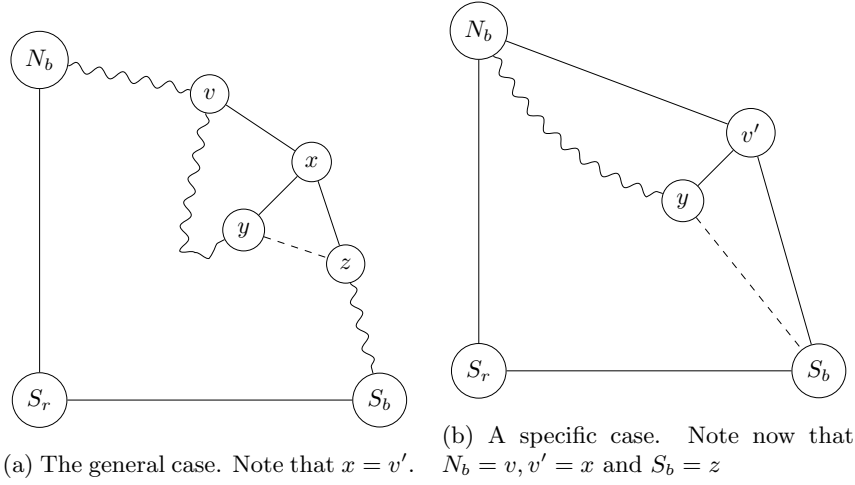


Figure 3: Creating a path satisfying (E2). The dotted line is the edge we take in the new path \mathcal{P}_2

- c) edges incident to v or v and some other vertex on $\mathcal{C}_{\mathcal{P}_2}$
- d) edges from $[v, v']$ to some internal vertex
- e) edges from $\mathcal{P} \setminus v, v'$ to some internal vertex

The existence of an edge as in a) is forbidden by Invariant 1 (I2). If b) occurs we can simply shortcut our original path \mathcal{P}_2 with this edge. If c) occurs this edge can't go to another vertex in $[v, v']$ since that would offend Invariant 1 (I2). Hence they go to a vertex in \mathcal{P}_2 and we can shortcut the path as in b).

If d) occurs we simply make a new path and if e) occurs we take a slightly adapted interior path. See figures

Since all of the moves shrink $\mathcal{C}_{\mathcal{P}_2}$ while keeping (E1) and (E2) intact and we can't infinitely shrink this means at a certain point no more moves are available. Since every offending edges allows a move this means that there are no more offending edges. Hence this version of \mathcal{P}'_2 satisfies (E3). For the final step of the proof we take $\mathcal{P}_3 = \mathcal{P}'_2$.

There is a path that also satisfies (E4) Suppose that \mathcal{P}_3 does not satisfy (E4). Then we can just take the would be interior edge and take this for a nwe path. This is again a finite procedure reducing the sum of $|\mathcal{P}_3| - |[v, v']|$. In the end we have a path satisfying (E1) - (E4). □

1.3 Minimum distributive lattice element

We get this when we take the “leftmost” eligible path.

1.4 Horizontal one-sides

Writers note: we should define the border of a face of a bipolar orientation somewhere. **Writers note:** and what we mean with *cycle border* and *face*

FiXme: write section, potentially just refer to Fusy2006

border **Writers note:** as well as the notation of $\mathcal{F}_{\mathcal{P}}$ the face of a path

As an exercise one could try to addapt Fusy's algorithm to generate horizontally one-sided layouts directly, without doing flips in the distributive lattice. It turns out that this is not that difficult.

Since the horizontal segments correspond to face in the blue bipolar orientation we want that one of the two borders of the face has a length of at most two. Since every eligible path we take splits off one face in the blue bipolar orientation it is easy to control this property.

Theorem 4. *In the update of the algorithm there is always an eligible path \mathcal{P} available such that either \mathcal{P} or $[v, v']$ is of length 2.*

In order to proof this theorem we will first show the following lemma.

Lemma 5. *If \mathcal{P} is an eligible path giving raise to a face $\mathcal{F}_{\mathcal{P}}$ of which both border have length at least 3. Then there exist an eligible path \mathcal{P}' such that the pathborder and cycleborder of its face $\mathcal{F}_{\mathcal{P}'}$ are both at least 1 shorter than those of $\mathcal{F}_{\mathcal{P}}$.*

Proof. In this proof we will frequently use property (E3) of a Eligible path, we won't mention it every time we use it.

We denote the source by s and the sink by t . We also assign names a, b and x, y to the first two vertices on both borders, see Figure 4a. Since every interior face of G is a triangle ax is an edge. Now we distinguis two cases, either ay is an edge (case 1) or bx is an edge (case 2). Tey can't both be an edge at the same time due to planarity, neither can it happen that both of them are not an edge since then the face containing the path $baxy$ is at least of degree 4.

In the first case a may be connected to more vertices on the pathborder, however there is a last one, say z . And this vertex is then also connected to b , otherwise it would not be the last one. Now we can provide an shorter eligible path \mathcal{P}' we start at a go to z and from there we follow the old path \mathcal{P} to t . See figure 4b. It is easy to see that all four properties of an eligible path hold for \mathcal{P}' .

In the second case x may be connected to more vertices along the cycle border, however there is a last one, say c . And this vertex is then also connected to y , otherwise it would not be the last one. Now we can provide an shorter eligible path $\mathcal{P}' = scz$. See figure 4c. It is straightforward to see that all four properties of an eligible path hold for \mathcal{P}' . \square

Proof of Theorem 4. By Theorem 3 we know there is a eligible path \mathcal{P} . If one of the borders of $\mathcal{F}_{\mathcal{P}}$ is of length 2 or less we are done. If this path gives raise to a face $\mathcal{F}_{\mathcal{P}}$ with both borders are both of length at least 3 we can repeatedly apply Lemma 5 until at least one of the borders is of length at most 2. \square

If we in every update of the algorithm take the paths from Theorem 4 we end up with the correct faces in the blue bipolar orientation and hence a horizontally one sided rectangular dual.

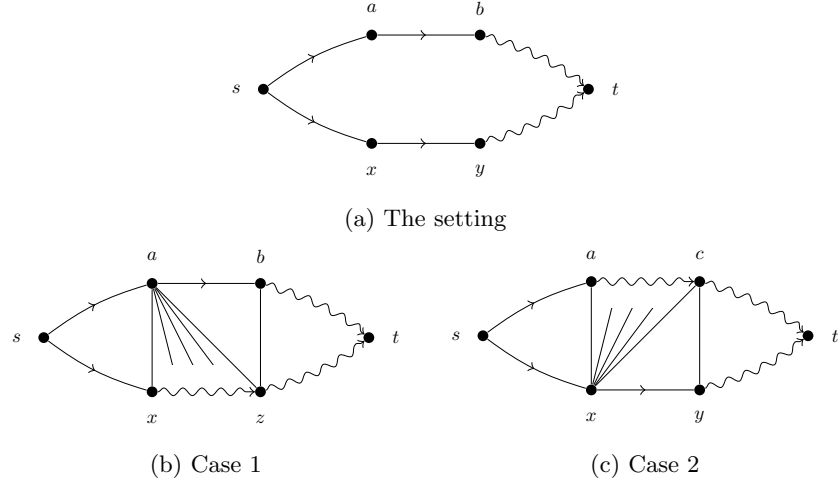


Figure 4

1.5 Vertical one-sided

We can also adapt Fusy's algorithm to generate a vertically one-sided dual. We do this by picking the eligible path with the leftmost starting point and letting it run for as long as possible. In the final oriented regular edge labelling we want to prevent red faces that have 3 or more edges on both borders.

If a red face has

Lemma 6. *A face F with at least 3 edges on each side contains a Z*

But a z can only be the result of a sequence of eligible paths that does not satisfy the requirements.