



HAL
open science

Testing Balanced Splitting Cycles in Complete Triangulations

Vincent Despré, Michaël Rao, Stéphan Thomassé

► **To cite this version:**

Vincent Despré, Michaël Rao, Stéphan Thomassé. Testing Balanced Splitting Cycles in Complete Triangulations. Canadian Conference on Computational Geometry (CCCG 2020), Aug 2020, Saskatchewan/Online, Canada. <10.4230/LIPIcs>. <hal-03059811>

HAL Id: hal-03059811

<https://hal.science/hal-03059811v1>

Submitted on 13 Dec 2020

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



HAL Authorization

1 Testing Balanced Splitting Cycles in Complete 2 Triangulations

3 **Vincent Despré**

4 LORIA, INRIA Nancy
5 vincent.despre@inria.fr

6 **Michaël Rao**

7 LIP, CNRS - ENS de Lyon
8 michael.rao@ens-lyon.fr

9 **Stéphan Thomassé**

10 LIP, ENS de Lyon
11 stephan.thomasse@ens-lyon.fr

12 — Abstract —

13 Let T be a triangulation of an orientable surface Σ of genus g . A cycle C of T is splitting if it cuts
14 Σ into two noncontractible parts Σ_1 and Σ_2 , with respective genus $0 < g_1 \leq g_2$. The splitting
15 cycle C is called balanced if $g_1 \geq g_2 - 1$. The complexity of computing a balanced splitting cycle
16 in a given triangulation is open, but seems difficult even for complete triangulations. Our main
17 result in this paper is to show that one can rule out in polynomial time the existence of a balanced
18 splitting cycle when the triangulation is ε -far to have one. Implementing this algorithm, we show
19 that large Ringel and Youngs triangulations (for instance on 22.363 vertices) have no balanced
20 splitting cycle, the only limitation being the size of the input rather than the computation time.

21 **2012 ACM Subject Classification** Theory of computation \rightarrow Randomness, geometry and dis-
22 crete structures

23 **Keywords and phrases** Computational topology, embedded graphs, randomized algorithm

24 **Digital Object Identifier** 10.4230/LIPIcs...

25 **Funding** The first author was partially supported by the grant ANR-17-CE40-0033 of the French
26 National Research Agency ANR (project SoS)

27 **1** Introduction

28 A splitting cycle on a surface Σ of genus at least 2 is a simple cycle (without self-crossing)
29 that allows to cut Σ into two parts non-homeomorphic to disks. In a continuous setting,
30 such a curve always exists and is easy to find. In the discrete setting, Σ is defined by a
31 combinatorial map M which is a graph embedded on Σ such that each face of the graph
32 is an open disk. In this case, a splitting cycle is a simple cycle (a cycle with no repeated
33 vertex) that separates Σ into two parts non-homeomorphic to disks. It is no more true that
34 every surface of genus at least 2 has a splitting cycle and it is NP-complete to decide if a
35 given M admits a splitting cycle [3, 2]. However, splitting cycles can be found when M has
36 some additional properties. For instance, simple triangulations (i.e. without loops, cycle of
37 length 2) are believed to have splitting cycles:

38 **► Conjecture 1 (Barnette (1982) [14, p. 166]).** Every simple triangulation of a surface of
39 genus at least 2 has a splitting cycle.



© Vincent Despré, Michaël Rao and Stéphan Thomassé;
licensed under Creative Commons License CC-BY
Leibniz International Proceedings in Informatics

LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

40 This conjecture is known to be true only in the case of the double torus [9]. It is
 41 formulated for triangulations but has been also investigated for combinatorial maps with
 42 minimum face-width (the minimum number of faces crossed by a non-contractible curve).
 43 It is easy to build a combinatorial map of face-width 2 without splitting cycles. Zha and
 44 Zhao [19] conjectured that a face-width of 3 is sufficient to obtain a splitting cycle and proved
 45 that 6 is actually enough. Triangulations are a particular case of this second conjecture since
 46 any simple triangulation has face-width at least 3.

47 Recall that a triangulation is called *irreducible* if none of its edges can be contracted
 48 without violating the condition of simplicity. It is easy to see that if T has a splitting cycle
 49 and is obtained by contracting an edge from some T' then T' also has a splitting cycle.
 50 Thus, it is sufficient to consider irreducible triangulations. Observe also that irreducible
 51 triangulations have face-width exactly 3. The number of irreducible triangulations of a
 52 given genus being finite [1, 15, 10], it is theoretically possible to check the conjecture for
 53 fixed genus. Sulanke gave an algorithm to compute the set of irreducible triangulations of
 54 a fixed genus [17] and used it to prove the conjecture for genus 2 with a computer assisted
 55 approach [18]. Unfortunately, the number of irreducible triangulations with respect to the
 56 genus grows too fast to hope for a brute force proof, even for genus 3.

57 In this paper we consider only orientable surfaces Σ of genus g . Therefore, a splitting
 58 cycle C cuts Σ into two parts of respective genus g_1, g_2 , where $g_1 \leq g_2$. We call g_1 the *type*
 59 of C , and C is called *balanced* if $g_1 \geq g_2 - 1$ (if such a cycle exists for T , we also say that T
 60 is *balanced*).

61 It was independently conjectured by Zha and Zhao [19] and Mohar and Thomassen [14,
 62 p. 167] that a triangulation (or a combinatorial map of face-width at least 3) have all the
 63 possible types of splitting cycles. However, Despré and Lazarus [4] disproved this by showing
 64 that some triangulations of complete graphs do not have all the possible types of splitting
 65 cycles. More precisely they could certify that some triangulation of K_{19} or K_{43} are not
 66 balanced. However, the algorithm they use could no rule out the existence of balanced large
 67 complete triangulations which still could be "smoother" than small ones and allow all types
 68 of splitting cycles. The key-result of this paper is first to show that existence of balanced
 69 cycle in a complete triangulation T of K_n can be property-tested, and then to provide an
 70 efficient implementation of this algorithm to test large Ringel-Youngs triangulations.

71 Observe that every splitting cycle C of a complete triangulation T of K_n partitions the
 72 edges into three classes (R, L, C) , where C are the edges of the cycle, R the edges to the right
 73 of C , and L the one to the left. Moreover, in the cyclic order σ_v induced by T around the
 74 edges incident to each vertex v , the order of the types of edges is (R, C, L, C) . In particular,
 75 we never have the cyclic pattern R, L, R, L . This allows a relaxation of the notion of splitting
 76 cycle. Precisely, for every $\varepsilon > 0$, an ε -cycle of T is a partition of the edges into three classes
 77 (R, L, U) such that:

- 78 ■ No vertex v have the cyclic pattern R, L, R, L in σ_v .
- 79 ■ All but εn of the vertices v of T are typical, i.e. every cyclic interval of σ_v of length εn
 80 contains an edge R or an edge L .

81 We say that an ε -cycle (R', L', U) approximates a splitting cycle (R, L, C) if $R' \subseteq R$ and
 82 $L' \subseteq L$ (here $U \subseteq C$ and stands for unknown). Our main result is the following:

83 ► **Theorem 2.** *There is a randomized algorithm running in time $f(\varepsilon)\text{poly}(|T|)$ which takes*
 84 *as input a complete triangulation T and returns w.h.p. a set X of ε -cycles such that every*
 85 *splitting cycle of T is approximated by some element of X . Moreover, the size of X only*
 86 *depends on ε .*

87 Note that if T has a balanced cycle C , then the previous algorithm will find w.h.p. a
 88 balanced ε -cycle (in a sense to be defined later). Let us say that T is ε -far to be balanced
 89 if it does not have a balanced ε -cycle. We have the following corollary:

90 **► Theorem 3.** *There is a randomized algorithm running in time $f(\varepsilon)\text{poly}(|T|)$ which takes*
 91 *as input a complete triangulation T which is either balanced or ε -far to be balanced and*
 92 *returns w.h.p. either a balanced ε -cycle, or a certificate that no balanced cycle exists.*

93 The previous algorithms are based on sampling a good set of vertices and can indeed be
 94 derandomized. However, even in the randomized version, the size of the family X is too large
 95 to allow any practical use. Luckily, when restricted to finding a set X approximating every
 96 balanced splitting cycle (hence cutting branches leading to unbalanced cycles), it turns out
 97 that a mix of random sampling and greedy choices can be implemented in a more efficient
 98 way. We could use this implementation in order to rule out the existence of balanced cycles
 99 in large Ringel and Youngs triangulations.

100 The fact that all splitting cycles can be ε -approximated by a bounded set Σ is non-
 101 intuitive if we think of the continuous setting. Indeed, the number of homotopy classes
 102 corresponding to balanced splitting cycles is infinite on the surface of genus g . By fixing a
 103 natural constant curvature metric on the underlying surface, it is known that the number of
 104 homotopy classes corresponding to splitting cycles that can be realized with length at most
 105 L is asymptotically L^{6g-6} [13]. In the discrete setting, we cannot reach an infinite number
 106 of homotopy classes since we only have a finite number of simple cycles. However, it would
 107 have been natural to expect a $K(g)$ (and thus n) dependency for the size of X .

108 The problem of constructing triangulations of complete graphs is a very classical one,
 109 raised by Heawood in 1890 [8]. The original aim was to find an optimal proper coloring
 110 of a graph embedded on a surface of genus $g > 0$. Apart from the case of the sphere (or
 111 the plane) and the Klein bottle, the Euler formula already gives the exact upper bound of
 112 $\gamma(g) = \lfloor \frac{7+\sqrt{1+48g}}{2} \rfloor$ colors. Hence, to prove the tightness of the bound, it was necessary to
 113 produce a graph of genus g with chromatic number $\gamma(g)$. This has been achieved by Ringel
 114 and Youngs [16, 7] using complete graphs. The embeddings they provided are minimal in
 115 the sense that each complete graph cannot be embedded on a smaller genus surface and
 116 some of them are triangulations. Actually, there are many different triangulations of a
 117 given complete graph [12, 11, 6, 5]. For the experiments in this paper we will focus on the
 118 triangulations given by Ringel and Youngs for $n \equiv 7[12]$.

119 The major difficulty here is that the size of the sample which gives the certificate is
 120 too large to allow computation based on a one-step guess. We instead adopt a randomized
 121 greedy strategy in order to iteratively construct the sample. The algorithm is described
 122 in details in Section 5. This algorithm is extremely efficient and allow to address huge
 123 triangulations. Actually, it may be used as soon as the size of the triangulation can be
 124 stored on the computer. It has been implemented independently by Vincent Despré and
 125 Michaël Rao and they were able to reach very huge complete triangulations.

126 **► Theorem 4.** *The complete triangulation with 22.363 vertices (and 250.040.703 edges)*
 127 *given by Ringel and Youngs has no balanced splitting cycle.*

128 The implementations details along with the different results are developed in Section 6.
 129 Our algorithm is a new tool to deal with splitting cycles and may be useful in a larger
 130 spectrum. Indeed, when it fails to prove that the input triangulation has no balanced
 131 splitting cycles, it gives hints to find possible ones since it outputs balanced ε -cycles which
 132 can be the seed of some new investigation. This is probably the most appealing open question

133 left by the paper: Given a balanced ε -cycle, how to decide if it can be extended or not into
 134 a balanced (or near balanced) cycle. If one could design an efficient algorithm in order to
 135 find balanced splitting cycles, it would lead to efficient *divide and conquer* algorithms on
 136 complete triangulations.

137 We first describe the background and notations in Section 2 and give some technical
 138 results about the structure of splitting cycles in Section 4.

139 2 Notations and Background

140 Combinatorial surfaces

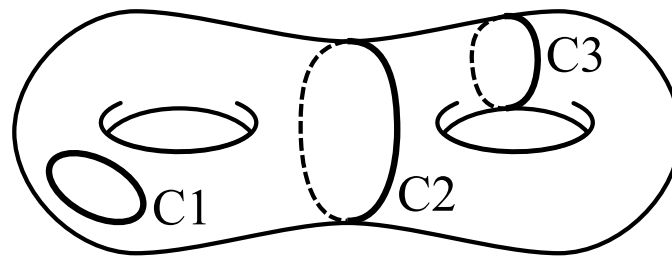
141 As usual in computational topology, we model a surface by a cellular embedding of a *simple*
 142 graph G (without loops or multiple edges) in a compact topological surface Σ . Such a
 143 cellular embedding can be encoded by a *combinatorial surface* composed of the graph G
 144 itself together with a rotation system [14] that records for every vertex of the graph the
 145 clockwise order of the incident edges. The *facial walks* are obtained from the rotation
 146 system by the face traversal procedure as described in [14, p.93]. We denote by n , e and f the
 147 numbers of vertices, edges and faces of the combinatorial surface. The genus g of Σ is linked
 148 to the embedding via a very strong topological property that we call Euler characteristic:
 149 $\chi(\Sigma) = n - e + f = 2 - 2g$. A triangulation is a particular kind of combinatorial map whose
 150 all faces are triangles. The combinatorial maps that we consider in this paper consists of
 151 triangulations of complete graphs where a complete graph is a graph containing all the
 152 possible pairs of vertices as edges. Such a triangulation does not trivially exists. It requires
 153 that $n \equiv 0, 3, 4$ or 7 [12] and even in this case the constructions are not straightforward. We
 154 consider a triangulation T_n for theoretical construction but the experiments are only done
 155 for the triangulations given by Ringel and Youngs [16] for $n = 12s + 7$. To summary, we have
 156 that the rotation scheme around the vertex v_i is a cyclic permutations σ_i of $\{1, \dots, n\} \setminus i$,
 157 such that: for every triangle ijk , if k is the successor of j in σ_i , then i is the successor of k
 158 in σ_j .

159 Data-structure

160 To be able to correctly analyze the complexity of our algorithm, it is necessary to describe a
 161 bit the data-structure we use: the half-edge data-structure. It consists in coding T_n by a set
 162 of half-edges each having an handle to the opposite half-edge (represented by an involution
 163 α_0) and to the next half-edge in the local σ_i (we can think of it as a global permutation σ
 164 whose cycles are the σ_i). At this point, we can notice that the size of the map is actually
 165 $2e \cdot \langle \text{size of an half-edge} \rangle = O(e)$. An edge is an orbit of the action of α_0 on the set of
 166 half-edges and can be stored as one element in the orbit. Similarly, the orbits of σ are the
 167 vertices, it is again sufficient to store one half-edge for each vertex. We need to store on
 168 each vertex a "reverse" dictionary Rev_i that associate to every vertex v_j for $j \neq i$ its position
 169 around v_i (each vertex is associated to a unique half-edge around v_i). The *Revs* dictionaries
 170 are not a general feature in the half-edge data-structure but is required by our algorithm.
 171 Finally, the faces can be construct by alternatively applying α_0 and σ and storing a half-edge
 172 for each corresponding orbit. Here, computing the faces is mainly useful to check that T_n
 173 is a correct triangulation. The construction of the map is considered as a precomputation
 174 and is clearly done using $O(e)$ operations.

175 **Combinatorial curves**

176 Consider a combinatorial surface with its graph G . A *cycle* C is a closed walk in G without
 177 repeated vertex. C may have different topological types. To understand this we need to
 178 define a (free) homotopy. Two closed continuous curves on Σ $\alpha, \beta : \mathbb{R}/\mathbb{Z} \rightarrow \Sigma$ are *homotopic*,
 179 if there exists a continuous map $h : [0, 1] \times \mathbb{R}/\mathbb{Z}$ such that $h(0, t) = \alpha(t)$ and $h(1, t) = \beta(t)$ for
 180 all $t \in \mathbb{R}/\mathbb{Z}$. Intuitively it means that two curves are homotopic if one can be continuously
 181 deformed into the other. We say that C is *contractible* if it is homotopic to a point and
 182 *separating* if its removal leaves two connected components on Σ . C may have three different
 183 homotopy types which are: contractible and separating, non-contractible and non-separating
 184 and non-contractible and separating (see Figure 1). If C is of this last type then we called
 185 it a *splitting cycle*. We can refine the notion of homotopy types for splitting cycle. Indeed,
 186 the genera g_1 and $g_2 \geq g_1$ of the two connected components defined by a splitting cycle are
 187 additive in the sense that $g_1 + g_2 = g$ (this is a direct consequence of the Euler characteristic
 188 which becomes $\chi(\Sigma) = 2 - 2g - b$ if Σ has b boundaries). We define the *type* of a splitting
 189 cycle as the genus g_1 . For instance, a splitting cycle of type 1 cuts Σ into a torus with one
 190 boundary and a surface of genus $g - 1$ with one boundary. We say that C is *balanced* if it
 191 has type $\lfloor \frac{g}{2} \rfloor$.



■ **Figure 1** C_1 is contractible, C_2 is a splitting cycle and C_3 is non-separating.

192 **Approximation of cycles**

193 As said before, a splitting cycle C of T_n induces an edge coloring of the edges of K_n into
 194 colors (L, R, C) such that C is the cycle and no R and L edges are cyclically adjacent in σ_v
 195 for all v . We will now see the splitting cycle C as the partition (L, R, C) . In particular, when
 196 C is balanced, this translates a sparse object (the splitting cycle C) into a dense object (the
 197 edge coloring (L, R, C)) since both R and L have quadratic size. This allows approximation
 198 of L and R by sampling. We now say that (L', R', U) *approximates* (L, R, C) if $R' \subseteq R$ and
 199 $L' \subseteq L$. The partition (L', R', U) is an ε -cycle if:

- 200 ■ For every vertex v_i , the cyclic order σ_i does not contain four cyclically ordered edges
- 201 R', L', R', L' .
- 202 ■ All but εn of the vertices v of T are *typical*, i.e. every cyclic interval of σ_i of length at
- 203 least εn contains an edge of R' or an edge of L' .

204 **3 Efficiently approximating cycles**

205 Our goal is to prove Theorem 2, which shows that one can efficiently find a set X of ε -cycles
 206 approximating all splitting cycles.

207 ► **Theorem 5.** *For every orientable triangulation T_n of K_n and every $\varepsilon > 0$, there is a set*
 208 *X of size $f(\varepsilon)$ consisting of ε -cycles such that every splitting cycle of T_n is approximated by*
 209 *some element of X .*

210 **Proof.** Pick some large constant $c > 4/\varepsilon^2$. We implicitly assume here that n is much larger
 211 than ε and c , otherwise X simply exists by enumeration. Pick uniformly at random a sample
 212 S of vertices of T_n of size c . For each $v_i \in S$, divide the cyclic order σ_i into c cyclic intervals
 213 I_1, \dots, I_c of approximately the same length (i.e. size $\lfloor (n-1)/c \rfloor$ or $\lceil (n-1)/c \rceil$). We now
 214 construct our ε -cycles (R, L, U) . We first decide for each $v_i \in S$ an R, L, U (right, left,
 215 unknown) coloring of the intervals I_j in such a way that two (possibly identical) intervals
 216 are U and these two U intervals separates the R intervals and the L intervals. Note that
 217 when the U intervals are identical or adjacent, the remaining intervals are all colored R or
 218 all colored L . The total number of such choices for a given $v_i \in S$ is $c^2 + c$. And we then
 219 have $(c^2 + c)^c$ possible ways of coloring the edges adjacent to S according to this local rule.
 220 Among these coloring, some of them are inconsistent in the sense that they give both colors
 221 R and L at the two endpoints of some edge between two elements of S . We reject these
 222 colorings. It can also happen that an edge receives both colors U and R (or U and L) in
 223 which case the edge keeps the color different from U . We then color U all edges which were
 224 not incident to vertices of S . We reject all colorings which contain the forbidden pattern
 225 (R, L, R, L) in some σ_i . The set of surviving (R, L, U) colorings is denoted by X_S , and this
 226 is our candidate for X . Note that the size of X_S only depends on c and hence on ε , and
 227 that the total number of U edges incident to points of S is at most $c.2n/c$.

228 The key-observation is that every splitting cycle C of T_n is approximated by some element
 229 of X_S . Indeed, for each vertex $v_i \in S$ one can define the two U intervals of σ_i as these
 230 containing an edge of C , and the R and L intervals are the one which are entirely R or L
 231 according to cycle C . So to reach our conclusion, we just have to show that every element
 232 of X_S is an ε -cycle.

233 We claim that this happens if we are lucky enough with our sampling S . Let us say that
 234 a vertex v_i is *good* if S is well distributed in σ_i . More precisely if for every cyclic interval
 235 of σ_i of size at least εn , the number of elements of S is at least $\varepsilon c/2$. Observe that the
 236 probability that a vertex is good tends to 1, when ε is fixed and c goes to infinity. By
 237 Markov, we can fix c large enough such that with high probability, our sampling S will be
 238 such that all vertices save an arbitrarily small proportion are good. We now claim that in
 239 this case, all (R, L, U) partitions of X_S are ε -cycles.

240 Assume for contradiction that this is not the case. Then there are more than εn non
 241 typical vertices v_i for which σ_i contains an interval I_{j_i} of size at least εn with no $R \cup L$
 242 edge. Since we can neglect these vertices v_i which are either in S or non good vertices, each
 243 of these intervals I_{j_i} contains $\varepsilon c/2$ vertices of S , and none of them have created an $R \cup L$
 244 edge with v_i . So the total number of U edges incident to vertices of S is at least $\varepsilon n. \varepsilon c/2$,
 245 which is contradicting the fact that there are at most $c.2n/c$ of them since $c > 4/\varepsilon^2$. ◀

246 This concludes the proof of Theorem 2, the algorithm simply returning X_S for some
 247 large enough sample S . The main drawback of this approach is the size of the sampling,
 248 which makes it very difficult to implement for some practical use. Since our goal is to look
 249 for balanced splitting cycles, we will only focus on ε -cycles which can be approximations of
 250 balanced cycles. Let us denote by $tr(n)$ the minimum size of R (or equivalently of L) in a
 251 balanced cycle (R, L, C) of an orientable triangulation of K_n . Note that $tr(n) = n^2/4 - O(n)$,
 252 but a more precise value will be given later when we will discuss the implementation. Thus
 253 if some ε -cycle (R', L', U) approximates (R, L, C) , it must have potentially at least $tr(n)$

254 many R' or L' edges. Let us properly define this. The *right-potential* $r(v_i)$ of some vertex
255 v_i is defined as:

- 256 ■ When v_i is incident to some edges of R' and L' , $r(v_i)$ is the size of the longest cyclic
257 interval of σ_i with a point in R' and no point in L' , minus 2.
- 258 ■ When v_i is only incident to edges of R' , we have $r(v_i) = n - 1$.
- 259 ■ When v_i is only incident to edges of L' , $r(v_i)$ is the size of the longest cyclic interval of
260 σ_i with no point in L' , minus 2.

261 The same definition applies for left potential $l(v_i)$. The *right-potential* $r(R', L', U)$ is
262 the sum of the right potential of all the vertices (same for left-potential $l(R', L', U)$). Note
263 that $r(R', L', U) \geq 2|R|$ and $l(R', L', U) \geq 2|L|$ when (R', L', U) approximates (R, L, C)
264 (the factor 2 in the inequality stands for the fact that we are doubly counting edges in the
265 potential). Let us then say that an ε -cycle (R', L', U) is *unbalanced* if $r(R', L', U) < 2tr(n)$
266 or $l(R', L', U) < 2tr(n)$ (otherwise it is *balanced*). A triangulation T_n is *ε -far to be balanced*
267 if it has no balanced ε -cycle.

268 **Proof of Theorem 3.** Now let us prove that we can efficiently separate triangulations which
269 are either balanced or ε -far to be balanced. For this, we compute a set X_S of ε -cycles which
270 approximates all splitting cycles of T_n . Note that if T_n admits a balanced cycle (R, L, C) ,
271 then it is approximated by some ε -cycle (R', L', U) in X_S which hence must be balanced
272 and thus a certificate of separation. Now if T_n does not admit a balanced cycle (R, L, C) ,
273 we compute a set X_S coming w.h.p. from a lucky sample S . The key point is that we can
274 indeed check if S is a good sample or not, just by checking if it is well-distributed in nearly
275 all σ_i . Hence the set X_S probably approximate all splitting cycles of T_n , and if we satisfy
276 the separation hypothesis of Theorem 3, none of the ε -cycles are balanced. Therefore X_S is
277 a certificate of the fact that T_n has no balanced splitting cycle. ◀

278 The nice feature of this property-testing algorithm is that if we try to check if a given
279 T_n has a balanced cycle, we may be lucky and get a NO-certificate. This is basically what
280 happens so far for all Ringel and Youngs triangulations on which the algorithm terminates.
281 However, in the present form, the size of X_S is way too large to be implemented, and we
282 will use a mix of random sampling and greedy choices for S . Also the fact that we divide
283 σ_i into c intervals is convenient for the proof but not for the algorithm, which will only cut
284 into 3 parts.

285 Another exciting direction of research is when we get a set X_S of ε -cycles, some of
286 which being balanced. There is possibly a way to investigate if a given balanced ε -cycle can
287 be completed into a balanced (or near balanced) cycle. For instance, if some σ_i contains
288 the pattern (R, U, R, L) , then the U edge can be turned into an R edge (possibly creating
289 forbidden patterns leading to reduction of X_S). These closure operations (together with a
290 (L, U, L, R) rule) can greatly densify our candidate ε -cycle making it easier to complete or
291 not into a splitting cycle.

292 **4 Properties of Splitting Cycles of Complete Triangulations**

293 We begin by fixing some specific notations. We need to split the neighborhood of the vertices
294 into parts. Mainly, if v_i is a vertex we denote by $(ev_0, ev_1, ev_2)_i$ a partition of the vertices
295 (or equivalently of the half-edges) around v_i such that the edges of each ev_i is consecutive
296 with respect to σ_i . We call a *local configuration* a couple $(i, c)_v$ where i corresponds to the
297 part ev_i and c is a color and a *configuration* a list of local configurations.

XX:8 Testing Balanced Splitting Cycles in Complete Triangulations

298 ► **Lemma 6.** *Let v be a vertex of T_n , (ev_0, ev_1, ev_2) be any partition of the edges in the*
 299 *neighborhood of v and (L, R, C) be a splitting cycle of T_n . At least one of the ev_i is entirely*
 300 *colored L or R .*

301 **Proof.** C may reach at most two of ev_0, ev_1 and ev_2 . It implies that one of the ev_i has to
 302 be colored entirely L or R for any splitting cycle. ◀

303 We thus obtain 6 different configurations for v . The following lemma is a direct consequence
 304 of the previous one.

305 ► **Lemma 7.** *Let (v_0, \dots, v_{k-1}) be a list of vertices of T_n and $(ev_0, ev_1, ev_2)_j$ be a fixed*
 306 *partition of the edges around v_j , for all $0 \leq j < k$. Then, there is a configuration*
 307 *$((i_0, c_0)_{v_0}, \dots, (i_{k-1}, c_{k-1})_{v_{k-1}})$ realized by each splitting cycle (L, R, C) .*

308 Let us now consider the particular properties of balanced splitting cycles of complete
 309 triangulations.

► **Lemma 8.** *Let $\mathcal{C} = (L, R, C)$ be a balanced splitting cycle of T_n . Then,*

$$|C| \geq \left\lceil \frac{5 + \sqrt{2n^2 - 14n + 25}}{2} \right\rceil$$

$$tr(n) = \min(|L|, |R|) \geq \left\lceil \frac{n^2 - 7n + 8 + 4\sqrt{2n^2 - 14n + 25}}{4} \right\rceil$$

310 **Proof.** Since we consider complete graphs, it is not possible that there exists two vertices
 311 colored entirely R for one and L for the other one. Hence, after cutting along C , there is
 312 a map with one boundary and no interior vertex of genus at least $\lfloor \frac{g}{2} \rfloor$. Let $k = |C|$ and
 313 T' be the map without interior vertices obtained after cutting along C . T' has genus at
 314 least $\lfloor \frac{g}{2} \rfloor$ and so $\chi(T') \leq 2 - 2 \lfloor \frac{g}{2} \rfloor - 1 \leq 2 - (g - 1) - 1 = 2 - g$. M' has k vertices,
 315 $e \leq \frac{k(k-1)}{2}$ edges and f faces. The double counting of the number of edges gives $3f = 2e - k$
 316 because all the edges are on exactly 2 faces except the k on the boundary. So $\chi(T') =$
 317 $k - e + 2\frac{e}{3} - \frac{k}{3} = \frac{2k - e}{3} \geq \frac{4k - k(k-1)}{6} = \frac{5k - k^2}{6}$. By putting together the two inequalities we
 318 obtain: $2 - g \geq \frac{5k - k^2}{6}$ leading to $k^2 - 5k + 6 - 6g \geq 0$. $\Delta = 25 - 4(6 - 6g) = 1 + 24g$ and
 319 so $k = |C| \geq \frac{5 + \sqrt{1 + 24g}}{2} = \frac{5 + \sqrt{1 + 2(n-3)(n-4)}}{2} = \frac{5 + \sqrt{2n^2 - 14n + 25}}{2}$.

320 Let us look back at the Euler formula for T' . We have, $\chi(T') = \frac{2k - e}{3} \leq 2 - g$. It implies
 321 that $e \geq 2k + 3g - 6 \geq 5 + \sqrt{2n^2 - 14n + 25} + \frac{3(n-3)(n-4)}{12} - 6 = \frac{(n-3)(n-4) + 4\sqrt{2n^2 - 14n + 25} - 4}{4} =$
 322 $\frac{n^2 - 7n + 8 + 4\sqrt{2n^2 - 14n + 25}}{4}$. ◀

323 It is interesting to notice that $\frac{e}{\min(|L|, |R|)} = \frac{1}{2} - O(\frac{1}{n})$ for balanced splitting cycles in
 324 complete triangulations and thus $tr(n) = \frac{n^2}{4} - O(n)$.

5 Algorithm

Sketch

325 We first describe the sketch of the algorithm. We suppose that a balanced splitting (L, R, C)
 326 exists and we want to obtain a contradiction. We choose at random a set of k vertices
 327 (v_0, \dots, v_{k-1}) of T_n and $(ev_0, ev_1, ev_2)_j$ a balanced partition of the edges around v_j , for all
 328 $0 \leq j < k$. By Lemma 7 we have a configuration $((i_0, c_0)_{v_0}, \dots, (i_{k-1}, c_{k-1})_{v_{k-1}})$ realized by

331 C . Up to a natural symmetry we can assume that $c_0 = L$. Thus, we have $3 \cdot 6^{k-1}$ possible
 332 configurations. We need to show that every configuration is not admissible. We look at
 333 the other vertices of the graph, we consider the colors induced by a given configuration and
 334 we have two tools to show that the configuration is not correct. First, if we can find an
 335 alternated sequence of edges labeled (L, R, L, R) around a vertex, then this vertex violates
 336 the conditions of (L, R, C) . We can also look the biggest number of edges colored R that
 337 each vertex can admits and use $tr(n)$ given by Lemma 8 to reject the configuration.

338 We want to explore the tree of all the possible configurations. We design this tree such
 339 that the layer i corresponds to the choice of the local configuration for the vertex v_i . A
 340 first approach is to take k big enough to reach a contradiction for all leaves of the research
 341 tree. This is not reasonable because of the growth of the size of the tree so we decide to
 342 check all the nodes in the tree where an internal node corresponds to a partial configuration.
 343 If this partial configuration already gives a contre-exemple then all the subtrees from the
 344 corresponding node can be discarded. In addition, we don't need to use the same v_k on all
 345 the nodes of a given layer. It means that we construct a tree of configurations starting from
 346 the root which is the empty configuration and we avoid getting deeper in the tree as soon
 347 as the can prove that the corresponding configurations is not correct.

348 Algorithm

349 **INPUT:** A complete triangulation.

- 350 ■ Let C be an empty vector of configurations. We initialize $RandV$ with a random ver-
 351 tex v_i and a random partition of the neighborhood of v_i into three consecutive parts
 352 $(ev_0, ev_1, ev_2)_i$. We put the configuration $(v_i, (ev_0, ev_1, ev_2)_i, 0, L)$ in C .
- 353 ■ We add a list L_j on each vector v_j that stores the position of the vertices already colored.
 354 At this stage, it means that for all $v_j \in ev_0$ we call $Rev_j(i)$ to know the position of v_i
 355 around v_j and we put $(Rev_j(i), L)$ in L_j . Notice that the L_i s must be sorted during the
 356 algorithm.
- 357 ■ While C is non-empty we do:
 - 358 1. We test if C is valid. This implies two tests:
 - 359 ■ We look at all the L_i s to see if there is no cyclic subsequence of the form (L, R, L, R) .
 - 360 ■ We sum the biggest interval that can be colored L (resp. R) in all the L_i s and we
 361 compare the result to the one of Lemma 8.
 - 362 2. If one of the test fails we update C in the following way:
 - 363 ■ If the last element of C is of the form $(\dots, 2, R)$ then we discard it and we update
 364 C again.
 - 365 ■ Else we consider the next configuration using the order: $(0, L), (1, L), (2, L), (0, R), (1, R)$
 366 and $(2, R)$.
 We update the L_i s to make it coherent with the new configuration and the go back
 367 to step 1.
 - 368 3. We compute a new random vertex v_i not already used by C with a partition of its
 369 neighborhood and we add $(v_i, (ev_0, ev_1, ev_2)_i, 0, L)$ at the end of C . We then update
 370 the L_i s and go back to 1.
 371

372 Analysis of the algorithm

373 ► **Theorem 9.** *If the algorithm terminates then the input triangulation does not have a*
 374 *balanced splitting cycle.*

XX:10 Testing Balanced Splitting Cycles in Complete Triangulations

375 **Proof.** If the algorithm terminates then C has described a research tree \mathcal{T} rooted at the
376 empty configuration. All the leaves of \mathcal{T} corresponds to configurations that are incoherent
377 (with the existence of a splitting cycle) in step 1. Now, if all the sons of a given node are
378 incoherent, it implies that the configuration of the node is incoherent. So, by induction, all
379 configurations in \mathcal{T} are incoherent and this includes its root. If the empty set is incoherent
380 with the existence of a balanced splitting cycle it implies that no such cycle may exist. ◀

381 ▶ **Theorem 10.** *The algorithm describe above requires $O(t \cdot d \cdot n) = O(t \cdot d \cdot \sqrt{e})$ operations
382 where t is the size of the research tree \mathcal{T} and d its depth.*

383 **Proof.** Each node of \mathcal{T} corresponds to one turn in the While. Step 1 requires to read all
384 the lists L_i . There are n such lists and their size is bounded by the size of C which is less
385 than the depth of \mathcal{T} . It implies that this step requires $O(d \cdot n)$ operations. Step 2 and 3
386 may require an insertion or a deletion in one third of the L_i which is clearly done in $O(d \cdot n)$
387 operations. Since we consider t configurations, we obtain a total of $O(t \cdot d \cdot n)$ operations. ◀

388 Optimizations

389 When we reach some depth in \mathcal{T} it becomes interesting to choose smartly the next local
390 configuration. Indeed, if the new vertex v_i already has two half-edges colored L pointing to
391 vertices v_0 and v_1 , then it should be interesting to consider a local configuration $(ev_0, ev_1)_i$
392 such that ev_0 and ev_1 are delimited by v_0 and v_1 . Indeed, one of this two sets have to be
393 entirely colored L in a (L, R, C) splitting cycle. We obtain only two local configurations to
394 check $(0, L)$ and $(1, L)$ instead of 6. To be sure that the ev_0 and ev_1 both contain enough
395 edges, we only use this setting when we find two half-edges of the same color separated by
396 at least a fixed distance $p \cdot e$ around v_i . After some testing, we decided to set p to 0.35, this
397 parameter may be changed but should stay in an interval $[0.3, 0.45]$ to be useful. In addition,
398 some minor optimizations can be made, we can check the (L, R, L, R) conditions while we
399 update the L_i s for instance.

400 The algorithm is highly parallelizable since different subtrees can use uncorrelated ver-
401 tices. The parallelization works as follows: we first set a value d_0 as the initial depth in the
402 tree of research and we choose d_0 fixed random vertices, then we set a list of tasks corre-
403 sponding to the $3 \cdot 6^{d-1}$ leaves of the initial tree. Now a master thread send a configuration
404 corresponding to a leaf to every other threads as soon as they achieved their previous task.
405 To prove that a configuration is impossible, a thread may need to construct its own subtree,
406 thus we decide to give to each thread a different copy of the data-structure. To reach bigger
407 triangulations, it may be useful to use a unique copy on each node but this will require to
408 put it *read-only* and so extract the L_i form the data-structure which may represent a loss
409 of performance.

410 6 Implementation details and experimental results

411 The implementation has been realized in C++ using OPENMPI for parallelization and
412 can be downloaded at <http://vdespre.free.fr/Splitting.tar.gz>. The test had been
413 launched on the cluster Grid'5000¹. Let m be the number of threads for given experiment.
414 The choice of d_0 can be optimized for each case so we precise what we used in each case.

¹ Experiments presented in this paper were carried out using the Grid'5000 testbed, supported by a scientific interest group hosted by Inria and including CNRS, RENATER and several Universities as well as other organizations (see <https://www.grid5000.fr>).

415 We first give results to show the efficiency of the algorithm. Notice that the limit is set
 416 by the RAM on each node and so the number of threads is set to not break the memory
 417 limit. The time column shows the average on 10 tries.

s	n	e	time (s.)	CPU time	m	nodes	d_0	t
833	10 003	50 025 003	425	21h15m	180	45	6	2 000 000
1863	22 363	250 040 703	2990	37h22m	45	45	5	1 700 000

419 It is interesting to notice that the time of the tests highly depends on the exact value of
 420 n . It means that the size of the research tree is not smooth with respect to n . It is pretty
 421 surprising and we have no hint of the reason by now. The following experiments have been
 422 done using 720 threads on 45 nodes.

s	n	time (s.)	σ (s.)	d_0	t
100	1207	18	1	7	1 800 000
101	1219	62	15	7	2 100 000
102	1231	945	224	9	41 000 000
103	1243	970	178	9	42 000 000
104	1255	17	1	7	1 800 000
105	1267	fails in 7200		10	
106	1279	35	8	7	1 900 000
107	1291	42	4	7	1 900 000
108	1303	220	45	7	8 200 000
109	1315	17	1	7	1 800 000
110	1327	18	1	7	1 800 000

424 7 Conclusion

425 The structure of the splitting cycles in triangulations of complete graphs remains quite
 426 mysterious. Even for the case of Ringel and Youngs embeddings restricted to $n = 12s + 7$,
 427 we do not understand what exactly happens. Our new experimental results give some
 428 informations on the absence of balanced splittings. In this specific case, we can imagine to
 429 make tests on bigger triangulations by storing the embedding using $O(n)$ memory. This can
 430 be done using the extreme symmetry of the embeddings but is not likely to be generalized.

431 We can also want to explore other triangulations of complete graphs. A very simple
 432 question remains open on this subject:

433 ► **Question 11.** Is there an unbounded sequence of triangulations of complete graphs admit-
 434 ting balanced splitting cycles?

435 The question is of intrinsic interest and it is difficult to have an intuition about it. The
 436 constructions of triangulations of complete graphs are pretty intricate and it is not clear
 437 if one can be modified to ensure the existence of a balanced splitting. In addition, we
 438 always look for an easy proof that some triangulation does not have a splitting cycle. We
 439 think that Theorem 5 is the kind of idea that can lead to such a proof. However, it is
 440 not clear how much the properties of a specific embedding must be used. In case that
 441 there exists huge triangulations of complete graphs with balanced splittings, embeddings
 442 become critical. If not, we can imagine to prove the non-existence of balanced splitting in
 443 complete triangulations without considering a specific embedding which is very convenient,
 444 in particular for probabilistic arguments.

445 — References —

- 446 **1** David W. Barnette and Allan L. Edelson. All orientable 2-manifolds have finitely many
447 minimal triangulations. *Israel Journal of Mathematics*, 62(1):90–98, 1988.
- 448 **2** Sergio Cabello, Éric Colin de Verdière, and Francis Lazarus. Finding cycles with topological
449 properties in embedded graphs. *SIAM J. Discrete Math.*, 25(4):1600–1614, 2011.
- 450 **3** Erin W Chambers, Éric Colin De Verdière, Jeff Erickson, Francis Lazarus, and Kim Whit-
451 tlesey. Splitting (complicated) surfaces is hard. In *Proceedings of the twenty-second annual
452 symposium on Computational geometry*, pages 421–429. ACM, 2006.
- 453 **4** Vincent Despré and Francis Lazarus. Some triangulated surfaces without balanced splitting.
454 *Graphs and Combinatorics*, 32(6):2339–2353, 2016.
- 455 **5** Mark N. Ellingham and Chris Stephens. Triangular embeddings of complete graphs (neigh-
456 borly maps) with 12 and 13 vertices. *Journal of Combinatorial Designs*, 13(5):336–344,
457 2005.
- 458 **6** Mike J Grannell and Martin Knor. A lower bound for the number of orientable trian-
459 gular embeddings of some complete graphs. *Journal of Combinatorial Theory, Series B*,
460 100(2):216–225, 2010.
- 461 **7** Jonathan L. Gross and Thomas W. Tucker. *Topological graph theory*. Dover, reprint 2001
462 from Wiley edition, 1987.
- 463 **8** P.J. Heawood. Map-color theorem. *Quart. J. Math. Oxford Ser. 24*, 1890.
- 464 **9** Dana L. Jennings. *Separating Cycles in Triangulations of the Double Torus*. PhD thesis,
465 Vanderbilt University, 2003.
- 466 **10** Gwenaël Joret and David R. Wood. Irreducible triangulations are small. *Journal of Com-
467 binatorial Theory, Series B*, 100(5):446–455, 2010.
- 468 **11** Vladimir P Korzhik and Heinz-Jurgen Voss. On the number of nonisomorphic orientable
469 regular embeddings of complete graphs. *Journal of Combinatorial Theory, Series B*,
470 81(1):58–76, 2001.
- 471 **12** Serge Lawrencenko, Seiya Negami, and Arthur T. White. Three nonisomorphic triangu-
472 lations of an orientable surface with the same complete graph. *Discrete Mathematics*,
473 135(1):367–369, 1994.
- 474 **13** Maryam Mirzakhani. Growth of the number of simple closed geodesies on hyperbolic
475 surfaces. *Annals of Mathematics*, pages 97–125, 2008.
- 476 **14** Bojan Mohar and Carsten Thomassen. *Graphs on Surfaces*. Studies in the Mathematical
477 Sciences. Johns Hopkins University Press, 2001.
- 478 **15** Atsuhiro Nakamoto and Katsuhiko Ota. Note on irreducible triangulations of surfaces.
479 *Journal of Graph Theory*, 20(2):227–233, 1995.
- 480 **16** Gerhard Ringel. *Map color theorem*, volume 209. Springer, 1974.
- 481 **17** Thom Sulanke. Generating irreducible triangulations of surfaces. arXiv:math/0606687,
482 2006.
- 483 **18** Thom Sulanke. Irreducible triangulations of low genus surfaces. arXiv:math/0606690, 2006.
- 484 **19** Xiaoya Zha and Yue Zhao. On non-null separating circuits in embedded graphs. *Contem-
485 porary Mathematics*, 147:349–349, 1993.