Paper Chase Assignment

For this assignment, I'd like you to get into the mindset of a reviewer, and to practice the skills of hunting down references to find clarifications and explanations. I also want you to start looking around for topics for your end-of-semester project.

- Find a research paper on computational topology; let's call this Paper 1. Paper 1 can be something related to your own research that you'd like to examine more closely, or you can dive into a relevant conference/journal and pick something you find interesting. I've included a list of potential starting points at the end of this handout, but only as suggestions; you're welcome to use papers outside this list.

- Now imagine that you were assigned Paper 1 as a reviewer. The first part of any paper review is a succinct summary (at most two short paragraphs) of the major contributions of the paper. **Write that summary.** Your summary should be objective; keep your *opinions* about the paper limited. (An actual paper review would also include your opinions about the paper after the summary, but they will not be a part of this assignment.) Your summary should also be easy for a program committee member / editor to read. Remember that even if they have more *general* expertise, they know less *about the paper* than you do!

- **Identify and explain the major topological and computational components of the paper**, using up to one full page. Again, your summary should be objective and easy for an inexpert editor / program committee member to read. (Imagine you six months ago.)

- Skim the rest of the paper to get a general idea of what is going on, and then read as much detail as you can. **Describe where you got lost**, using up to one full page. (This is *not* normally part of a review.) Include enough context that your point of confusion is clear to someone who is not familiar with the paper. If you can't figure out the major contributions of the paper, start over with a different Paper 1. If you never get lost, start over with a different Paper 1.

- Identify another paper that would help enhance your understanding of Paper 1; let's call this Paper 2. This could be a paper that Paper 1 cites, a paper that cites Paper 1, or just a result of your mad googling skills. Read Paper 2, first skimming to get a general idea of its content, and then trying to read as much detail as you can. **Briefly summarize paper 2, describe its connection to where you got lost in paper 1, and describe where you got lost in Paper 2.** Use up to one full page.

- Repeat the previous step one more time. Identify yet another paper that would help your understanding of Paper 2; let's call this Paper 3. Read Paper 3. **Briefly summarize Paper 3, describe its connection to where you got lost in previous papers, and describe where you got lost in Paper 3.** Use up to one full page.

Altogether, your writeup should be roughly 3–5 pages long. Don't forget to properly cite the papers you read (and any other papers that you reference in your writeup). Please submit your writeup on Gradescope by **September 21, 2020**. The link to enroll on Gradescope will be available soon.

Choosing Papers

You're welcome to choose any three related papers on any aspect of computational topology, subject to the following restrictions:

- Don't review papers that you already know well. I want you to deliberately go outside your comfort zone. I do *not* expect you to understand your papers in complete detail.

- In particular, don't review any papers written by either you or your advisor; you have an obvious conflict of interest. (But you are welcome to use those papers to find *other* papers to read and review.)

- Similarly, don't review any of *my* papers; I have an obvious conflict of interest! (Again, you are welcome to use my papers to find *other* papers to read and review.)

I've included a list of possible starting points in the references. I've tried to choose papers that I think are relevant to this class, that describe interesting results, that are relatively accessible, and that don't look like each other. This list is skewed toward, but not limited to, papers directly related to topics that we will cover in class; please don't be offended if your favorite papers, authors, venues, or topics are missing. If you don't otherwise know where to start, I recommend looking at the abstracts and figures in these papers for something interesting. You are *not* required to choose your papers from this list.

Once you've chosen an initial paper to start with, there are several recommended strategies for finding additional papers to read, not only for this assignment or your project, but in your own research.

- **By citation**: Go backward in time: Look at the papers that your favorite papers cite. Or go forward in time: Look at the papers that cite your favorite papers, using an indexing service like Google Scholar or Scopus. These are often the most fruitful strategies, and the ones I recommend for this assignment, but they're not the only strategies you should use in general. Yes, of course, *you* always cite every possible piece of related work in *your* papers, but most authors don't.

- **By keyword**: Look for more papers (or lecture notes, or slides, or whatever) that use *similar* key terminology as your favorite papers. Try likely synonyms, even if you think they mean something completely different; different research communities use different words for the same concepts, and the same words for different concepts.

- **By venue:** Look at other papers in the venues (conference, journal, workshop, seminar series, arXiv category, Github . . .) that published your favorite papers.

- **By author:** Look at other papers (or lecture notes, or slides, or whatvever) produced by the authors of your favorite papers, as well as their coauthors, students, and advisors.

- **By recommendation:** Ask people who are likely to know something about your favorite topics to suggest papers you might not otherwise think of.

- **By chance:** Keep your eyes peeled for anything vaguely reminiscent of your favorite papers: figures in other papers, Numberphile videos, old Martin Gardner columns, whiteboard scribbles, bathroom graffiti, dreams, Jack Stauber lyrics, children's toys, ... Use them to launch a Google Images/Scholar search. No, I'm not joking.

Grading Rubric

	Excellent	Good	Needs work	Unacceptable
Paper 1 summary	Clear summary of main contributions, and easy to follow for someone with general background in computatioal topology	Contributions are discussed, but some confusion or lack of clarity about main points or technical material	Summary includes some contributions, but does not rank major contributions or contains major errors	Completely unclear what major contributions are
Paper 1 tools	Clear discussion of major topological and computational tools and techniques	Discussion is unclear, or focuses only on topology or only on algorithms	Mention of topology and/or algorithms, but vague and/or clearly indicates misunderstanding	Misunderstood or omitted entirely
Paper 1 confusion	Clear description of where you got lost in Paper 1 and why, written for someone not familiar with the paper	Some discussion, but not clear what the problem was	No context, or discussion of paper without clear description of where you got lost	Omitted entirely
Paper 2	Brief summary of Paper 2; connection to where you got lost in Paper 1; clear description of where you got lost in Paper 2	Some discussion, but not at excellent level	More serious inaccuracies or issues in description	Major issues; for example, not clear what paper 2 was about
Paper 3	Brief summary of Paper 3; connection to where you got lost in Papers 1 and 2; clear description of where you got lost in Paper 3	Some discussion, but not at excellent level	More serious inaccuracies or issues in description	Major issues; for example, not clear what paper 3 was about

References

[1] Zachary Abel, Erik D. Demaine, Martin L. Demaine, Hiro Ito, Jack Snoeyink, and Ryuhei Uehara. Bumpy pyramid folding. *Comput. Geom. Theory Appl.* 75:22–31, 2018.

[2] Amariah Becker, Eli Fox-Epstein, Philip N. Klein, and David Meierfrankenfeld. Engineering an approximation scheme for traveling salesman in planar graphs. *Proc. 16th Int. Symp. Exper. Algorithms*, 8:1–8:17, 2017. Leibniz Int. Proc. Informatics 75, Schloss Dagstuhl–Leibniz-Zentrum für Informatik.

[3] Glencora Borradaile, Erin Wolf Chambers, Kyle Fox, and Amir Nayyeri. Minimum cycle and homology bases of surface-embedded graphs. *J. Comput. Geom.* 8(2):58–79, 2017.

[4] Glencora Borradaile, Philip N. Klein, Shay Mozes, Yahav Nussbaum, and Christian Wulff-Nilsen. Multiple-source multiple-sink maximum flow in directed planar graphs in near-linear time. *SIAM J. Comput.* 46(4):1280–1303, 2017.

[5] Luca Castelli Aleardi and Olivier Devillers. Array-based compact data structures for triangulations: Practical solutions with theoretical guarantees. *J. Comput. Geom.* 9(1):247–289, 2018.

[6] Erin Wolf Chambers and Yusu Wang. Measuring similarity between curves on 2-manifolds via homotopy area. *J. Comput. Geom.* 10(1):96–126, 2019.

[7] Hsien-Chih Chang, Pawel Gawrychowski, Shay Mozes, and Oren Weimann. Near-optimal distance emulator for planar graphs. *Proc. 26th Ann. Europ. Symp. Algorithms*, 16:1–16:17, 2018. Leibniz Int. Proc. Informatics 112, Schloss Dagstuhl–Leibniz-Zentrum für Informatik.

[8] David Cohen-Steiner, André Lieutier, and Julien Vuillamy. Lexicographic optimal homologous chains and applications to point cloud triangulations. *Proc. 36th Int. Symp. Comput. Geom.*, 32:1–32:17, 2020. Leibniz Int. Proc. Informatics 164, Schloss Dagstuhl–Leibniz-Zentrum für Informatik.

[9] Christian A. Duncan, David Eppstein, Michael T. Goodrich, Stephen G. Kobourov, Maarten Löffler, and Martin Nöllenburg. Planar and poly-arc Lombardi drawings. *J. Comput. Geom.* 9(1):328–355, 2018.

[10] David Eppstein and Bruce Reed. Finding maximal sets of laminar 3-separators in planar graphs in linear time. *Proc. 30th Ann. ACM-SIAM Symp. Discrete Algorithms*, 589–605, 2019.

[11] Parker Evans, Brittany Terese Fasy, and Carola Wenk. Combinatorial properties of self-overlapping curves and interior boundaries. *Proc. 36th Int. Symp. Comput. Geom.*, 41:1–41:17, 2020. Leibniz Int. Proc. Informatics 164, Schloss Dagstuhl–Leibniz-Zentrum für Informatik.

[12] Stefan Felsner and Günter Rote. On primal-dual circle representations. *Proc. 2nd Symp. Simplicity in Algorithms*, 8:1–8:18, 2018. OpenAccess Ser. Inform. 69, Schloss Dagstuhl–Leibniz-Zentrum für Informatik.

[13] Leo Ferres, José Fuentes-Sepúlveda, Travis Gagie, Meng He, and Gonzalo Navarro. Fast and compact planar embeddings. *Comput. Geom. Theory Appl.* 89:101630, 2020.

[14] Arnold Filtser. A face cover perspective to ℓ_1 embeddings of planar graphs. *Proc. 31st Ann. ACM-SIAM Symp. Discrete Algorithms*, 1945–1954, 2020.

[15] Xavier Goaoc, Pavel Paták, Zuzana Patáková, Martin Tancer, and Uli Wagner. Shellability is NP-complete. *J. ACM* 66(3):21:1–21:18, 2019.

[16] Adam Karczmarz and Piotr Sankowski. Min-cost flow in unit-capacity planar graphs. *Proc. 27th Ann. Europ. Symp. Algorithms*, 66:1–66:17, 2019. Leibniz Int. Proc. Informatics 144, Schloss Dagstuhl–Leibniz-Zentrum für Informatik.

[17] Ken-ichi Kawarabayashi and Anastasios Sidiropoulos. Polylogarithmic approximation for Euler genus on bounded degree graphs. *Proc. 51st Ann. ACM Symp. Theory Comput.*, 164–175, 2019.

[18] Linda Kleist, Boris Klemz, Anna Lubiw, Lena Schlipf, Frank Staals, and Darren Strash. Convexity-increasing morphs of planar graphs. *Comput. Geom. Theory Appl.* 84:69–88, 2019.

[19] Anna Lubiw, Daniela Maftuleac, and Megan Owen. Shortest paths and convex hulls in 2D complexes with non-positive curvature. *Comput. Geom. Theory Appl.* 89:101626, 2020.

[20] Dániel Marx, Marcin Pilipczuk, and Michal Pilipczuk. On subexponential parameterized algorithms for Steiner tree and directed subset TSP on planar graphs. *Proc. 59th Ann. IEEE Symp. Foundations Comput. Sci.*, 474–484, 2018.

[21] Shay Mozes, Yahav Nussbaum, and Oren Weimann. Faster shortest paths in dense distance graphs, with applications. *Theor. Comput. Sci.* 711:11–35, 2018.

[22] Benjamin Raichel and C. Seshadhri. Avoiding the global sort: A faster contour tree algorithm. *Discrete Comput. Geom.* 58(4):946–985, 2017. Erratum in https://personal.utdallas.edu/~bar150630/paint.pdf.

[23] Jonathan Rollin, Lena Schlipf, and André Schulz. Recognizing planar Laman graphs. *Proc. 27th Ann. Europ. Symp. Algorithms*, 79:1–79:12, 2019. Leibniz Int. Proc. Informatics 144, Schloss Dagstuhl–Leibniz-Zentrum für Informatik.

[24] Nicholas Sharp, Yousuf Soliman, and Keenan Crane. Navigating intrinsic triangulations. *ACM Trans. Graph.* 38(4):55:1–55:16, 2019.

[25] Yipu Wang. Maximum integer flows in directed planar graphs with vertex capacities and multiple sources and sinks. *Proc. 30th Ann. ACM-SIAM Symp. Discrete Algorithms*, 554–568, 2019.