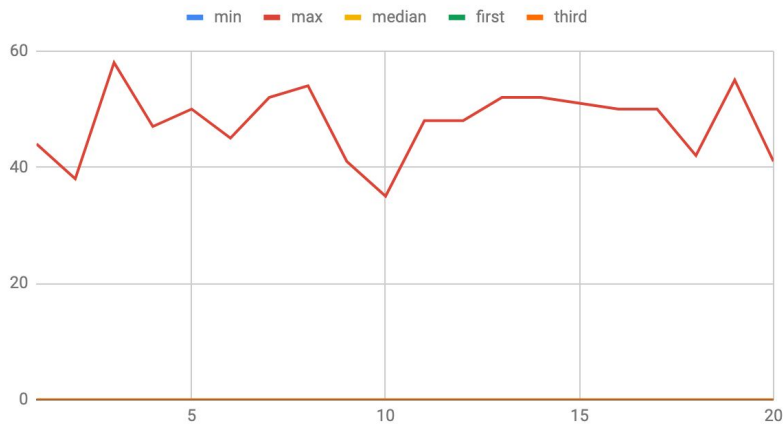


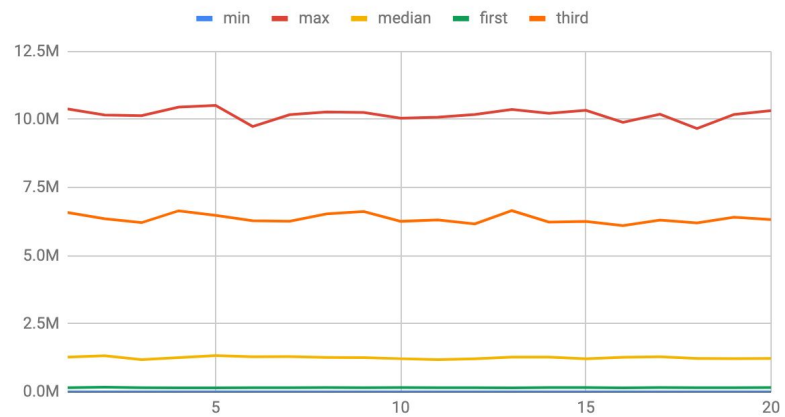
Rob Schwartz
Project 2 PDF
Data Mining

Graphs:

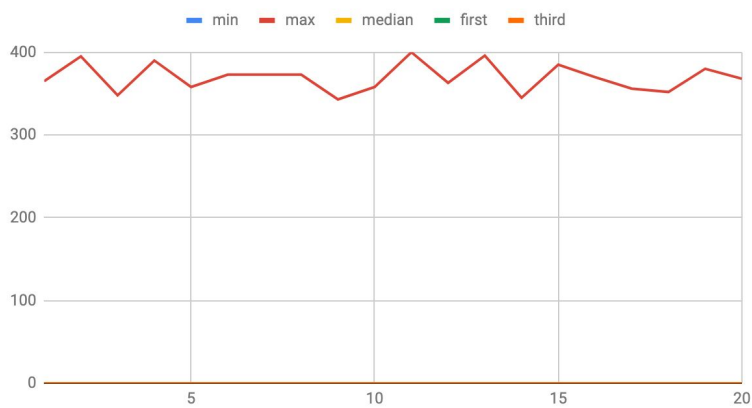
20 Runs of Base M=5000



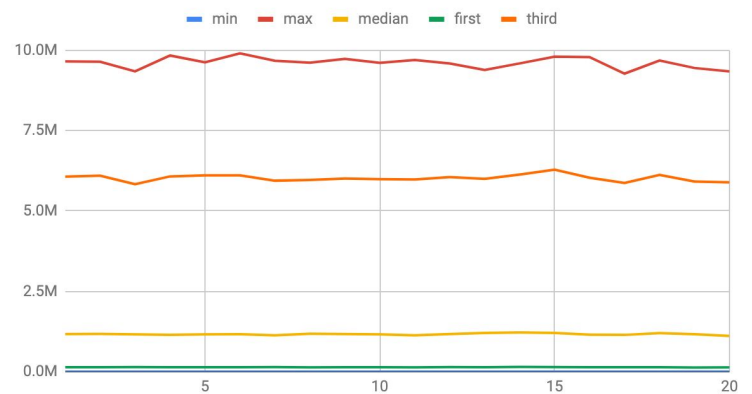
20 Runs of Improved M=5000



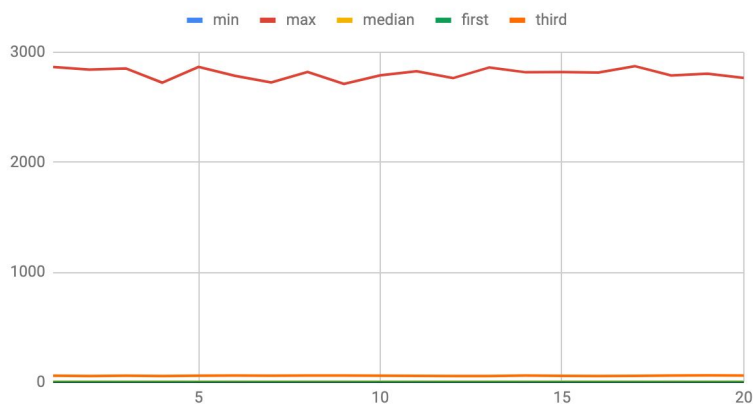
20 Runs of Base M=10000



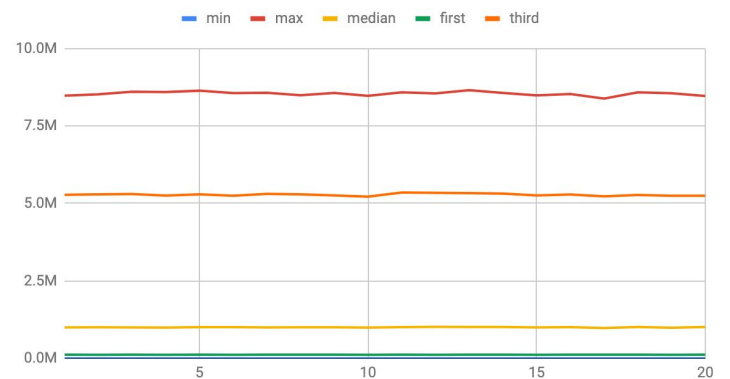
20 Runs of Improved M=10000



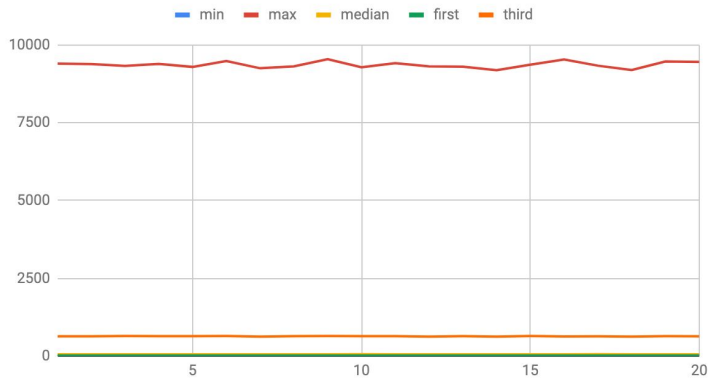
20 Runs of Base M=20000



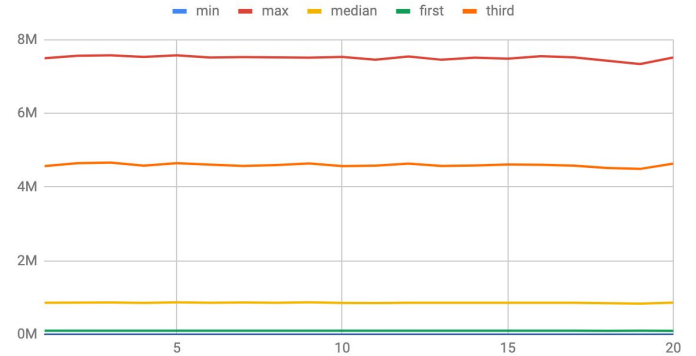
20 Runs of Improved M=20000



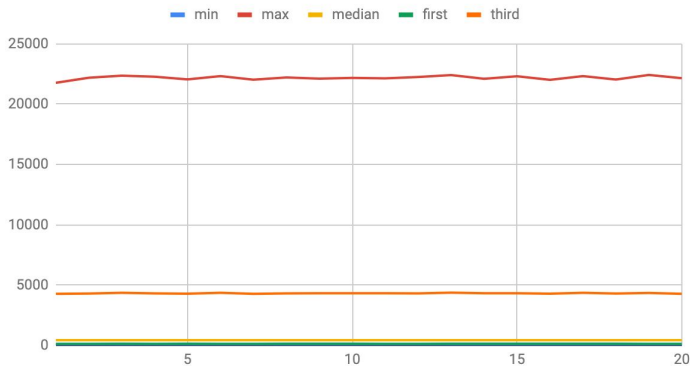
20 Runs of Base M=30000



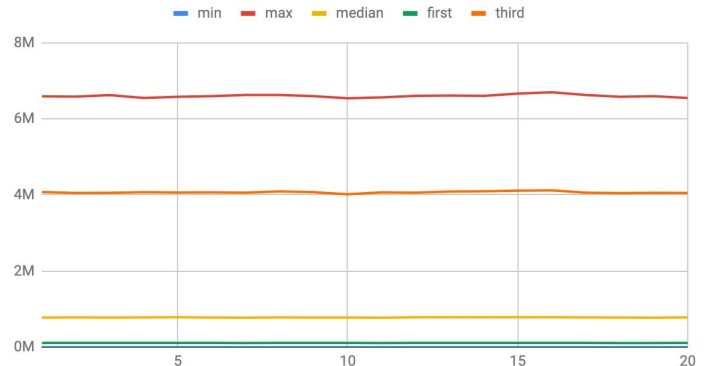
20 Runs of Improved M=30000



20 Runs of Base M=40000



20 Runs of Improved M=40000



Summary:

I'll first look at the results from my implementation of Trièst-Base. The minimum, as it ought to be, is zero for each chart, and the maximum is in line with what I would expect, given how the algorithm operates. When Trièst-Base reaches its memory capacity, the number of predicted triangles tends to plummet, as we remove an edge each time we add one. This means that more memory allows the algorithm to reach greater triangle predictions before the predicted number decreases. For this reason, each successive memory increase is accompanied by an increase in maximum predicted triangles. The same process causes each of the median and first and third quartiles to increase as M increases (though it is difficult to see them on the charts).

Trièst-Improved operates a bit differently. Whereas the predicted triangles can decrease in Trièst-Base, they cannot for Trièst-Improved. Rather, we frequently increase the number of predicted triangles. Whenever we *might* add an edge that would increase the number of triangles, the number of predicted triangles increases. This explains why the values for the Trièst-Improved charts are so much greater than the values for Trièst-Base. However, the

predicted triangles using Trièst-Improved decrease as M increases, which is opposite to the behavior of Trièst-Base. This reversal, however, can be easily explained. With Trièst-Improved, we increase the number of predicted triangles by using an equation - g - which is simply $(t-1)(t-2)/m(m-1)$. Naturally, as M increases, g will decrease, so we will increase the number of predicted triangles by smaller and smaller amounts.

There are two main differences between these two algorithms. They calculate the number of predicted triangles differently, and they calculate replacement of triangles at different times. With Trièst-Base, we complete the edge-replacement process before we calculate the change in triangles. With Trièst-Improved, we calculate the number of triangles while the edge is on the stream - not beforehand. And when we call for the number of triangles, with Trièst-Base, we have to use a function of D and M , while we can just return the D we've been incrementing constantly with Trièst-Improved.