

# Large Crossing Angles in Circular Layouts

Quan Nguyen<sup>1</sup>, Peter Eades<sup>1</sup>, Seok-Hee Hong<sup>1</sup>, and Weidong Huang<sup>2</sup>

<sup>1</sup> School of Information Technologies, University of Sydney  
{qnguyen,peter,shhong}@it.usyd.edu.au

<sup>2</sup> CSIRO ICT Centre, Australia  
tony.huang@csiro.au

## 1 Introduction

Recent empirical research has shown that increasing the angle of crossings reduces the effect of crossings and improves human readability [5]. In this paper, we introduce a post-processing algorithm, namely MAXCIR, that aims to increase crossing angles of circular layouts by using Quadratic Programming. Experimental results indicate that our method significantly increases crossing angles compared to the traditional equal-spacing algorithm, and that the running time is fairly negligible.

## 2 Algorithm

The post-processing approach MAXCIR in this paper aims to increase the crossing angles after a circular ordering of the vertices is given, such as by a crossing reduction algorithm [3]. With a fixed ordering of the vertices, all of the crossing pairs  $(e, e')$  can be pre-determined in linear time. Let  $\Omega$  denote the set of pairs of crossing edges. We aim to minimise

$$F = \sum_{(e, e') \in \Omega} \left( \alpha_{(e, e')} - \frac{\pi}{2} \right)^2, \quad (1)$$

where  $\alpha_{(e, e')}$  denotes the angle at which edges  $e$  and  $e'$  cross.

The circular layout is a function  $\theta : V \rightarrow [0 .. 2\pi)$  that associates an angle  $\theta(u)$  with each vertex  $u$ . Suppose that the ordering of the vertices around the circle is  $(u_0, u_1, \dots, u_{n-1})$ . We denote  $\theta(u_i)$  by  $\theta_i$ .

For every edge  $e$  of  $G$ , let  $e_m$  and  $e_M$  denote the end vertices such that  $\theta(e_m) < \theta(e_M)$ . For a pair of crossing edges  $e$  and  $e'$ , their crossing angle is given by  $\alpha_{(e, e')} = \frac{1}{2} (\theta(e_M) - \theta(e'_M) + \theta(e_m) - \theta(e'_m))$  where  $\theta(e'_m) < \theta(e_m) < \theta(e'_M) < \theta(e_M)$ .

In practice an angular gap  $g$  between vertices needs to be preserved, e.g., for avoiding overlapping node labels. The circular ordering and the gap lead to the following constraints:

$$0 \leq \theta_0; \quad \theta_i + g < \theta_{i+1}, \forall i = 0..n-2; \quad \theta_{n-1} + g < 2\pi. \quad (2)$$

Minimizing  $F$  in equation (1) subject to the constraints (2) defines a quadratic program.

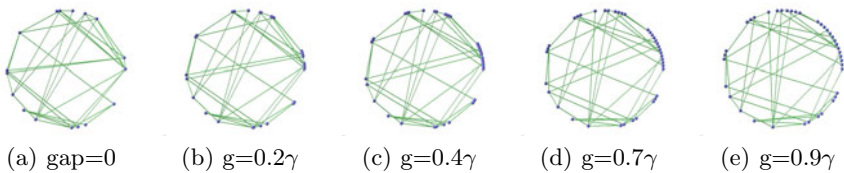
### 3 Experiments

We compare our new MAXCIR layout algorithm with EQCIR — a circular layout that places the nodes equally on the circle. We use the Rome data set [2] and use Baur and Brandes’s algorithm [3] to generate a vertex ordering. The experiments were conducted on a 3.00GHz Pentium IV CPU and 1GB RAM Solaris/SPARC with cplex solver v10.0.1 [1].

We ran MAXCIR with gap values  $g=0$ ,  $0.2\gamma$ ,  $0.4\gamma$ ,  $0.7\gamma$  and  $0.9\gamma$ , where  $\gamma = \pi/|V|$ . We obtained the *average crossing angle* produced by MAXCIR and EQCIR, over all five gap values. A significant improvement of 61.4% was obtained; the average is slightly reduced when the default gap value is increased. The *average execution time* of MAXCIR varies between 6ms and 13ms, and slightly reduces when the gap value increases.

We measured the angular resolution of MAXCIR layout results. We adapted the measurement used in [4] to circular layouts by defining the optimal angle at a vertex  $v$  as  $180/\text{deg}(v)$  instead of  $360/\text{deg}(v)$ . The *average angular resolution* produced by MAXCIR is 49.18 degrees compared to 55.50 degrees produced by EQCIR. When the gap value increases the average angular resolution slightly decreases. A smaller difference implies a better angular resolution as the angles between pairs of coincident edges are closer to optimal.

Figure 1 shows circular layouts of a Rome graph instance produced by MAXCIR with five gap values. For small gap values, the crossing angles are largely optimised, yet several nodes are placed too close to each other or overlap one another. For larger gap values, the crossing angles and angular resolution are slightly improved, but the nodes are better distinguished.



**Fig. 1.** Applying MAXCIR with different gap values on Rome graph 10005\_39

### 4 Conclusion

We have presented a post-processing method based on quadratic programming for circular layouts. The method significantly improves the crossing angles of a circular layout with negligible overheads.

Our future work will incorporate angular resolution into MAXCIR model. Maximising angular resolution is particularly interesting since crossing angle competes with angular resolution when two adjacent edges cross the same edge.

## References

1. cplex (2010), <http://www.ilog.com/products/cplex/>
2. The Rome graphs (2010), <http://www.graphdrawing.org/data/index.html>
3. Baur, M., Brandes, U.: Crossing reduction in circular layouts. In: Hromkovič, J., Nagl, M., Westfechtel, B. (eds.) WG 2004. LNCS, vol. 3353, pp. 332–343. Springer, Heidelberg (2004)
4. Finkel, B., Tamassia, R.: Curvilinear graph drawing using the force-directed method. In: Pach, J. (ed.) GD 2004. LNCS, vol. 3383, pp. 448–453. Springer, Heidelberg (2005)
5. Huang, W., Hong, S.H., Eades, P.: Effects of crossing angles. In: IEEE Pacific Visualization Symposium, pp. 41–46 (2008)