Minimizing Wiggles in Storyline Visualizations

Theresa Fröschl and Martin Nöllenburg

Algorithms and Complexity Group, TU Wien, Vienna, Austria

A storyline visualization is a two-dimensional drawing of a special kind of time-varying hypergraph $H(t)$, where the x-axis represents time and the vertices (also called *characters*) are x-monotone curves. At each point in time t, the vertices form a permutation such that groups of adjacent characters in $H(t)$ occupy consecutive vertical positions to indicate a *meeting* at time t, see Fig. 1. Each character can only be part of at most one meeting at each point in time. This kind of visualization has been introduced for illustrating movie narratives [8], but is also more generally used in information visualization [6, 11].

Several aesthetic optimization criteria have been proposed [6, 11], including minimization of crossing, line wiggles, and white-space gaps. While crossing minimization has been studied from an algorithmic point of view in recent years [4, 5, 7], minimizing line wiggles, as another important quality criterion, which is similar to bend minimization in node-link diagrams [9, 10], has not been investigated on its own. We note that the problem of minimizing corners or moves in permutation diagrams [2, 3] is related to wiggle minimization, yet does not include the temporal aspects of storylines with meetings over time and their induced character ordering constraints. We present the first integer linear programming (ILP) model for exact wiggle minimization in storyline visualizations without an initial permutation. We can include crossing minimization into a weighted multicriteria ILP model and show examples of a first case study.

ILP formulation. A storyline visualization can be encoded as an $m \times p$ matrix with columns for the p time points, where meetings start or end, and $m > n$ rows for the slots used by the n characters of $H(t)$, where m is chosen large enough to allow for blank lines between different meetings at the same time points. The position of character i at time point t is expressed as a binary variable $x^t_{i,j}$ that is set to 1 if and only if i uses slot j at time point t. No two characters can use the same slot at the same time point ($\sum_{i=1}^{n} x^t_{i,j} \leq 1$). With this information about the position of the characters over time it is possible to determine the line wiggles of a character i by comparing the position of i for two successive time points t and $t+1$. If the position changes, a wiggle is detected. The absolute value of the difference of the occupied slots at both time points yields the *height* of the wiggle, which is identified with the variable z^t_i. Using this height as the weight of a wiggle we get the following ILP model with the total wiggle height as objective

$$\text{minimize} \sum_{i=1}^{n} \sum_{t=1}^{p-1} z^t_i$$

$$\text{subject to } \sum_{j=1}^{m} j \cdot (x_{i,j}^t - x_{i,j}^{t+1}) \leq z_i^t, \quad \sum_{j=1}^{m} j \cdot (x_{i,j}^{t+1} - x_{i,j}^t) \leq z_i^t \quad \text{for all } z_i^t.$$

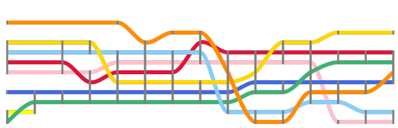
In addition we need to define constraints for correctly representing the character meetings and, for better visual distinction, keeping a blank line between any neighboring characters that do not meet. For a meeting e with k members between time points t_0 and t_1 we define integer variables $j_{\min}^{e,t}$ and $j_{\max}^{e,t}$ for the minimum and maximum slots for e at time points t with $t_0 \leq t \leq t_1$. The difference of these two slots must be exactly $j_{\max}^{e,t} - j_{\min}^{e,t} = k - 1$. By comparing the variables $j_{\min}^{e,t}$ and $j_{\max}^{e',t}$ for distinct meetings e and e' at the same time point t it is possible to define constraints that require blank lines between e and e'.

Finally, by using the position variables of any two characters a and b a binary comparison variable $y_{a,b}^t$ for this pair of characters can be created which takes value 1 if and only if a is placed above b at time point t by the constraints

$$m \cdot y_{a,b}^t \geq \sum_{j=1}^{m} j \cdot x_{b,j}^t - \sum_{j=1}^{m} j \cdot x_{a,j}^t, \qquad y_{a,b}^t + y_{b,a}^t = 1.$$

A crossing between the characters a and b at time point t can be determined if the equation $y_{a,b}^t + y_{a,b}^{t+1} = 1$ is satisfied. If there is no crossing it evaluates to 0 or 2. With this a secondary objective function for crossing minimization can be added to the ILP, similar to the crossing minimization of Gronemann et al. [4].

Implementation. The ILP model is implemented using Gurobi [1]. Figure 1 illustrates results of a snippet of the movie Inception. Figure 1a shows the result of pure wiggle minimization with a total wiggle height of 55 (and 26 crossings) found after 33.37min, while Figure 1b shows the result of minimizing wiggles and crossings in a weighted multi-objective way with weight 1 for wiggles and weight 3 for crossings; it has a total wiggle height of 59 and 20 crossings and was found after 16.65min. The multi-objective seems to produce more appealing results, yet there is still a need for improvements, especially for larger instances. We finally note that the ILP can be modified to minimize the number of wiggles or the maximum wiggle height instead of the total wiggle height.



(a) wiggle minimization (b) wiggle and crossing minimization

Fig. 1: Example snippets of storyline visualizations for the movie Inception; meetings are indicated by vertical lines

References

1. Gurobi optimizer 7.5. http://www.gurobi.com (2017), accessed: 3 August 2017
2. Bereg, S., Holroyd, A.E., Nachmanson, L., Pupyrev, S.: Drawing permutations with few corners. In: Wismath, S., Wolff, A. (eds.) Graph Drawing (GD'13). LNCS, vol. 8242, pp. 484–495. Springer (2013)
3. Bereg, S., Holroyd, A.E., Nachmanson, L., Pupyrev, S.: Representing permutations with few moves. arXiv preprint arXiv:1508.03674 (2015)
4. Gronemann, M., Jünger, M., Liers, F., Mambelli, F.: Crossing minimization in storyline visualization. In: Hu, Y., Nöllenburg, M. (eds.) Graph Drawing and Network Visualization (GD'16). LNCS, vol. 9801, pp. 367–381. Springer (2016)
5. Kostitsyna, I., Nöllenburg, M., Polishchuk, V., Schulz, A., Strash, D.: On minimizing crossings in storyline visualizations. In: Di Giacomo, E., Lubiw, A. (eds.) Graph Drawing and Network Visualization (GD'15). LNCS, vol. 9411, pp. 192–198. Springer International Publishing (2015), http://arxiv.org/abs/1509.00442
6. Liu, S., Wu, Y., Wei, E., Liu, M., Liu, Y.: Storyflow: Tracking the evolution of stories. IEEE Transactions on Visualization and Computer Graphics 19(12), 2436–2445 (2013)
7. Markfelder, P., Ravsky, A., Suri, S., Wolff, A.: Block crossings in storyline visualizations. In: Hu, Y., Nöllenburg, M. (eds.) Graph Drawing and Network Visualization (GD'16). vol. 9801, pp. 382–398. Springer (2016)
8. Munroe, R.: Xkcd# 657: Movie narrative charts (2009)
9. Purchase, H.: Which aesthetic has the greatest effect on human understanding? In: Di Battista, G. (ed.) Graph Drawing (GD'97). LNCS, vol. 1353, pp. 248–261. Springer (1997)
10. Purchase, H.C., Cohen, R.F., James, M.: Validating graph drawing aesthetics. In: Brandenburg, F.J. (ed.) Graph Drawing (GD'95). LNCS, vol. 1027, pp. 435–446. Springer (1996)
11. Tanahashi, Y., Ma, K.L.: Design considerations for optimizing storyline visualizations. IEEE Transactions on Visualization and Computer Graphics 18(12), 2679–2688 (2012)