A Fast Grid Layout Algorithm for Biological Pathways

Kaname Kojima  Masao Nagasaki  Satoru Miyano
kaname@ims.u-tokyo.ac.jp  masao@ims.u-tokyo.ac.jp  miyano@ims.u-tokyo.ac.jp
Human Genome Center, Institute of Medical Science, University of Tokyo, 4-6-1 Shirokanedai, Minato-ku, Tokyo 108-8639, Japan

Keywords: grid layout, graph layout, biological pathway, time complexity, search space

1 Introduction

Clearly visualized biological pathways provide a great help in understanding biological systems. However, manual drawing of large-scale biological pathways is a time consuming task.

So far, various layout algorithms have been designed for biological pathways; above all grid layout algorithms [1, 2, 3] succeeded in generating layouts suitable for biological pathways taking into account several aesthetic criteria and complicated positional constraints.

In grid layout algorithms, vertices constituting the graph are mapped to grid points which satisfy their biological localization information, and are arranged to minimize a cost function defined over all possible mappings to penalize edge-edge crossings, vertex-edge crossings and distance between vertices.

Since finding the layout with the minimal cost is NP-hard, grid layout algorithms employ the following heuristic search to find a locally optimal layout: a layout is repeatedly improved by applying the changing reducing the cost the most, considering all the cost differences induced by moves of each vertex to each vacant point of the grid, and the algorithm finishes when no more changing reduces the cost.

The extant algorithms optimize the above computation in such a way that all the cost differences at the previous step are stored and used for the computation of the current step.

In spite of the above efficient computation, those algorithms are still slow for practical use, so that more sophisticated calculation technique is desired. Thus, first we propose a new calculation method that calculates all the cost differences more efficiently than naive way without using the cost differences of the previous step. Such a method can be used to initialize efficiently the $\Delta$ matrix, step which was, so far, requiring a lots of time, in a naive way. In addition, this calculation method can also be applied to update steps using $\Delta$ matrix and hence succeeds in reducing the time complexity of grid layout algorithm.

2 Method

Given a graph $G = (V,E)$ with nodes $V$ and edges $E$, a grid of $h$ rows and $w$ columns, a layout $L = (V,E,U,P)$ of $G$ consists of the underlying graph $G$, $w \cdot h$ grid points $U$ and a function $P : V \rightarrow U$ such that $P(v_\alpha) \neq P(v_\beta)$ for any two distinct nodes $v_\alpha, v_\beta \in V$. This definition does not allow overlaps between nodes in the layout. We define the following functions:

- $\text{Cross}_{e_i,e_j}(L)$: a binary function that returns 1 if an edge $e_i$ crosses with an edge $e_j$ and 0 otherwise.
- $\text{Cross}_{v_i,e_j}(L)$: a binary function that returns 1 if an edge $e_j$ crosses with a node $v_i$ and 0 otherwise.
• Distance\(_{v_i,v_j}(L)\): a function that returns \(w_{v_i,v_j} \cdot md(v_i,v_j)\), where \(w_{v_i,v_j}\) is the weight to the couple of nodes \(v_i\) and \(v_j\), and \(md(v_i,v_j)\) is the Manhattan distance between \(v_i\) and \(v_j\).

By using the above functions, the layout cost \(C(L)\) of \(L\) is defined as follows:

\[
C(L) = W_{ee} \sum_{e_i,e_j \in E} \text{Cross}_{e_i,e_j}(L) + W_{ne} \sum_{v_k \in V, e_l \in E} \text{Cross}_{v_k,e_l}(L) + W_{dc} \sum_{v_m,v_n \in V} \text{Distance}_{v_m,v_n}(L),
\]

(1)

where \(W_{ee}\), \(W_{ne}\), and \(W_{dc}\) are called respectively edge-edge crossing weight, node-edge crossing weight, and distance cost weight and used for adjusting the effect of each factors.

At each step, the algorithm calculates costs of all layouts that can be generated by moving one of all vacant points. The layout with the minimum cost is selected as a starting layout for the next step. After convergence, the algorithm outputs a locally optimal layout. Although calculating all possible adjacent layouts requires high time complexity in a naïve way, the previous approach introduced \(\Delta\) matrix that stores each possible cost difference at the previous step and succeeded in reducing the time complexity at each step from \(O(w \cdot h \cdot (|V|^2 + |E|^2))\) to \(O(|V|^2 + |E|^2 + w \cdot h \cdot |E|v) \cdot (|V| + |E|))\), where \(v_j\) is the node moved at the previous step.

Since the first step of the algorithm cannot utilize data in \(\Delta\) matrix, extant algorithms calculate the first step in naïve way. Thus, we devise a new calculation technique that encode the cost functions to optimize the calculation. By using this technique we proposed a new algorithm that calculates all the cost differences \(O(\max(w,h))\) times as fast as naïve way without using the cost differences of the previous step.

The following proposition briefly shows the process and time complexity of the calculation of the cost differences.

**Proposition 1** We can encode \(\text{Cross}_{e_i,e_j}(L_v)\), \(\text{Cross}_{e_i,e_j}(L_v)\), \(\text{Distance}_{v_k,e_l}(L_v)\) over \(L_v\) in \(O(\min(w,h))\), where \(L_v\) is the set of layouts obtained by moving vertex \(v\) and \(e\) is considered to be connected to \(v\). The decoding process requires \(O(w \cdot h)\) time. The above processes do not require the cost differences at the previous step.

Since from Proposition 1 the cost induce by the movement of vertex \(v\) can be calculated \(O(\min(w,h) \cdot (|V| + |E|)(\deg(v) + 1) + w \cdot h \cdot |V|)\) time, where \(\deg(v)\) is the degree of \(v\), we see that without using the \(\Delta\) matrix all the cost differences can be calculated \(O(\min(w,h) \cdot (|V|^2 + |E|^2) + w \cdot h \cdot |V|)\) time. Although this calculation method requires more time complexity than the method using \(\Delta\) matrix, this calculation method can also be applied to the calculation part that is not covered with \(\Delta\) matrix and hence reduces the time complexity of grid layout algorithm from \(O(w \cdot h \cdot |E|v) \cdot (|V| + |E|))\) to \(O(\min(w,h) \cdot |E|v) \cdot (|V| + |E|) + w \cdot h \cdot |V|)\). We summarize this result as the following proposition.

**Proposition 2** The calculation of \(\Delta\) matrix requires \(O(\min(w,h) \cdot |E|v) \cdot (|V| + |E|) + w \cdot h \cdot |V|)\) time at each step.

**References**

