
ТОЧНАЯ МАТЕМАТИЧЕСКАЯ ПОСТАНОВКА ДЛЯ ПРОРИСОВКИ ГИПЕРРЕБЕР ДВУХСЛОЙНОГО ГИПЕРГРАФА В СХЕМАХ КООПЕРАЦИИ

Васильев Юрий Михайлович, асп.

Санкт-Петербургский государственный экономический университет, Садовая ул., 21, Санкт-Петербург, Россия, 191023; e-mail: vas_yu_m@mail.ru

Цель: решение задачи минимизации пересечения гиперребер при укладке двухслойного гиперграфа в точной постановке. *Обсуждение:* прорисовка ребер – заключительный этап метода Сугиямы для укладки направленного ациклического иерархического графа. В диаграммах потоков финансовых данных и схемах кооперации контрагентов при выполнении государственных и коммерческих проектов ребра обычно прорисовывают вертикальными или горизонтальными сегментами. При этом сам иерархический граф рассматривается как иерархический гиперграф, который отличается от обычного графа тем, что его ребра соединяют не две вершины различных слоев, а некоторое множество вершин (предков) верхнего слоя с некоторым множеством вершин (потомков), расположенных в нижних слоях. Для прорисовки гиперребер исходный гиперграф представляется в виде множества двухслойных подграфов, состоящих из двух соседних слоев, и задача визуализации гиперребер решается для каждого подграфа отдельно. *Результаты:* дана точная математическая постановка задачи минимизации пересечений гиперребер для представленных требований к прорисовке двухслойного гиперграфа, получены и проанализированы числовые результаты решения сформулированной задачи методом ветвей и границ.

Ключевые слова: направленный ациклический иерархический граф, двухслойный граф, гиперграф, гиперребро, прорисовка ортогональных гиперребер, укладка графа, метод ветвей и границ.

DOI: 10.17308/meps.2017.1/1589

Введение

Наиболее известный метод для размещения направленных ациклических иерархических графов – это метод Сугиямы [12], который предполагает разбиение сложной задачи по укладке графа на несколько последовательных подзадач:

- разбиение множества вершин графа на подмножества (слои), при этом каждое подмножество располагается одно под другим, все вершины-предки расположены выше вершин-потомков;
- определение последовательности вершин на слоях с целью минимизации пересечения ребер;
- поиск координат вершин, учитывая набор эстетических критериев;
- прорисовка ребер.

При прорисовке ребер вершины получают свои финальные координаты, которые нестрого определены назначенным слоем и координатами вершин на предыдущих шагах метода Сугиямы. Существуют различные варианты прорисовки ребер: прямыми линиями, сплайнами, ортогональными сегментами. Виды прорисовки ребер и алгоритмы осуществления выбранной прорисовки разрабатываются под специфичные требования к укладке графа [1-4], например:

- способы изображения вершин графа;
- способы визуализации точек соединения (входа/выхода) ребер с вершинами;
- способы визуализации наложения/пересечения ребер;
- наличие длинных ребер в графе.

Ребра в диаграммах потоков данных и схемах кооперации обычно прорисовывают вертикальными или горизонтальными сегментами [7], а иерархический граф рассматривается как иерархический гиперграф, который отличается от обычного графа тем, что его ребра соединяют не две вершины различных слоев, а некоторое множество вершин (источников) верхнего слоя с некоторым множеством вершин (стоков), расположенных в нижних слоях. Такие «гиперребра», по сути, являются объединением нескольких обычных ребер, см. рис. 1, на котором изображен обычный граф (слева) и соответствующий ему гиперграф с двумя гиперребрами (справа).

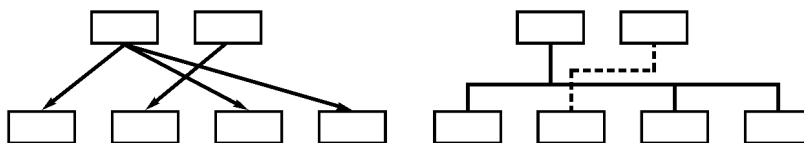


Рис. 1. Иерархический граф (слева) и соответствующий ему гиперграф с гиперребрами (справа).

В математических терминах слойный гиперграф $H = (V, E_H, \lambda)$ – это граф, где каждой вершине $u \in V$ оператор λ ставит в соответствие натуральное число (номер слоя) $\lambda(u)$, $1 \leq \lambda(u) \leq k$, а E_H – множество гиперребер, при этом каждое гиперребро $e = (S, T)$ инцидентно множеству вершин-предков $S \subset V$ и множеству вершин-потомков $T \subset V$. Если S состоит только из одной вершины, то такое гиперребро называется «одноисточным» [6]. На этапе прорисовки гиперребер для каждой вершины – слой-

ного гиперграфа известны координаты $(x(u), y(u))$, где $u \in V$. В данной статье рассматривается случай, когда гиперребра соединяют только вершины соседних слоев.

Разобьем исходный гиперграф на множество подграфов, каждый из которых состоит из двух его соседних слоев, тогда прорисовку гиперребер рационально проводить на каждом двухслойном подграфе отдельно и последовательно сверху вниз. Основная цель состоит в минимизации числа пересечений гиперребер каждого подграфа за счет взаимного расположения их горизонтальных сегментов.

2. Требования к прорисовке

В статье приняты следующие требования к прорисовке гиперребер двухслойного гиперграфа, учитывающие особенности применения иерархических гиперграфов для визуализации финансовых потоков [13]:

1. вершины представлены в виде прямоугольников фиксированного размера шириной 2ρ и высотой ρ ;

2. каждое гиперребро инцидентно одной вершине-источнику и смежным вершинам нижнего слоя;

3. каждое гиперребро состоит из трех сегментов:

- первый сегмент sV_1 представляет собой вертикальный отрезок, выходящий из вершины-источника;
- второй сегмент sH представляет собой горизонтальный отрезок, проходящий на одной ординате с ординатой нижнего конца сегмента sV_1 и имеющий с ним общую точку;
- третий сегмент sV_2 представляет собой набор вертикальных отрезков, начинающихся на сегменте sH и входящих в вершины-стоки.

Следует заметить, что в частном случае, когда вершина-источник находится точно над единственной вершиной-стоком, сегмент sH превращается в точку.

4. наложение (наличие бесконечного числа общих точек) двух гиперребер возможно только для сегментов sV_2 . Во всех остальных случаях наложение горизонтальных и вертикальных сегментов недопустимо. Например, на рис. 2 (сверху) изображено недопустимое наложение первого вертикального сегмента «сплошного» гиперребра и второго вертикального сегмента «пунктирного» гиперребра. Избежать таких наложений можно либо изменением порядка горизонтальных сегментов гиперребер, либо изменением абсцисс вершин нижнего слоя. На рис. 2 (снизу) приведена корректная укладка того же гиперграфа, полученная только за счет изменения порядка горизонтальных сегментов гиперребер;

5. если два гиперребра имеют общую вершину-сток, и при этом для одного гиперребра она является крайней левой среди всех инцидентных этому ребру вершин-стоков, а для второго гиперребра – крайне правой, то горизонтальные сегменты гиперребер могут иметь одинаковую ординату. В

этом случае прорисовка может быть такой, как изображено на рис. 3 (справа). Однако он является равноправным варианту, показанному на рис. 3 (слева), выбор одного из этих вариантов зависит от прорисовки остальных гиперребер;

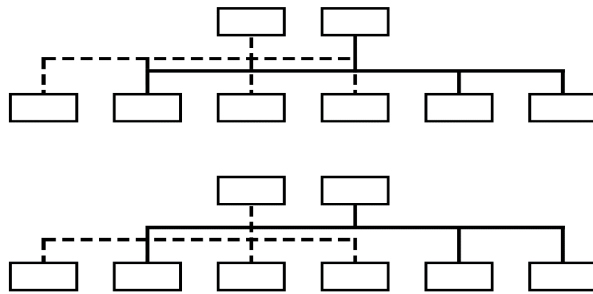


Рис. 2. Недопустимое наложение первого вертикального сегмента «сплошного» гиперребра и второго вертикального сегмента «пунктирного» гиперребра (сверху); корректная укладка гиперграфа, полученная изменением порядка горизонтальных сегментов гиперребер (снизу)

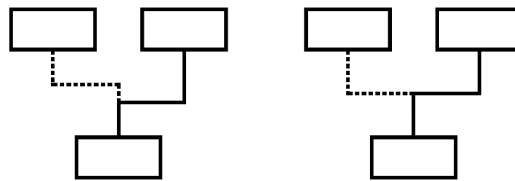


Рис. 3. Гиперграф без пересечения горизонтальных сегментов (слева) и с пересечением горизонтальных сегментов (справа)

6. сегмент sV_1 каждого гиперребра выходит строго из середины вершины-источника;

7. все вертикальные отрезки сегмента sV_2 каждого гиперребра входят строго в середину каждой инцидентной ему вершины-стока.

Введем термин «горизонталь», под которым будем понимать одно из допустимых положений горизонтальных сегментов гиперребер двухслойного гиперграфа. Если гиперграф содержит t гиперребер, то горизонтальный сегмент каждого гиперребра может быть расположен на одной из t горизонталей, пронумерованных от 1 до t сверху вниз.

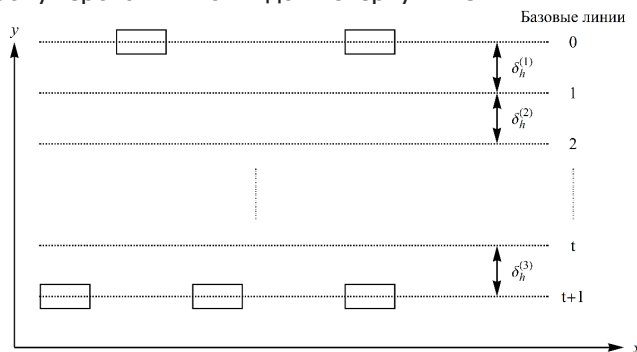


Рис. 4. Шаблон прорисовки гиперребер для двухслойного гиперграфа

Шаблон для прорисовки гиперребер для двухслойного гиперграфа в случае $t \geq 2$ приведен на рис. 4, при этом на базовой линии 0 расположены центры вершин верхнего слоя, а на базовой линии $t + 1$ – центры вершин нижнего слоя. Примем, что минимальное расстояние между соседними вершинами одного слоя равно δ_w . Будем также использовать три вида расстояний для прорисовки гиперребер гиперграфа:

- $\delta_h^{(1)}$ – расстояние между базовой линией верхнего слоя вершин (проходящей через их центры) и первой горизонталью;
- $\delta_h^{(2)}$ – расстояние между соседними горизонталями;
- $\delta_h^{(3)}$ – расстояние между последней горизонталью и базовой линией нижнего слоя вершин (проходящей через их центры) двухслойного гиперграфа.

Пересечением двух гиперребер назовем наличие общей точки горизонтального сегмента первого гиперребра с одним из вертикальных сегментов второго. Число пересечений гиперребер, которое, очевидно, зависит от взаимного расположения их горизонтальных сегментов, будет равно количеству таких общих точек. Введем в рассмотрение три типа пересечений гиперребер:

- пересечение первого типа – пересекаются первый вертикальный сегмент одного гиперребра и горизонтальный сегмент второго гиперребра;
- пересечение второго типа – пересекаются второй вертикальный сегмент одного гиперребра (один или несколько его вертикальных отрезков) и горизонтальный сегмент второго гиперребра;
- пересечение третьего типа (неустранимое пересечение) – пересечение второго типа, которое возникает независимо от взаимного расположения горизонтальных сегментов двух гиперребер в том случае, когда у них существует общая (инцидентная им обоим) вершина-сток.

На рис. 5 (сверху) изображен гиперграф с вариантом прорисовки гиперребер, приводящим к одному пересечению первого типа, одному пересечению второго типа и одному неустранимому пересечению. Выделенная серым цветом вершина инцидентна обоим гиперребрам. На рис. 5 (снизу) показан тот же гиперграф с одним неустранимым пересечением гиперребер. Уменьшение числа пересечений достигнуто перестановкой горизонтальных сегментов двух гиперребер, при этом неустранимые пересечения учитываться при подсчете пересечений в дальнейшем не будут.

Отметим, что ни один используемый на практике в настоящее время алгоритм [2-9] не подходит полностью под сформулированные требования по визуализации гиперребер.

Рассмотрим задачу прорисовки гиперребер двухслойного гиперграфа с целью минимизации числа их пересечений за счет взаимного расположения их горизонтальных сегментов.

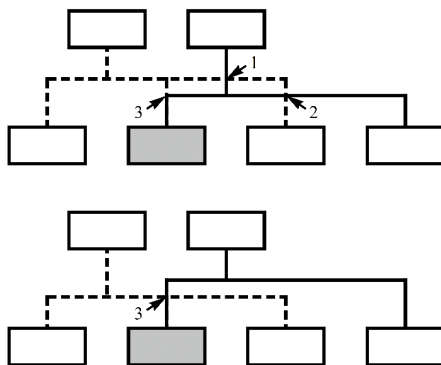


Рис. 5. Пересечения гиперребер различных типов

3. Алгоритм прорисовки гиперребер

Для удовлетворения всех требований, описанных в предыдущем разделе, был разработан алгоритм, включающий в себя следующие этапы:

1. предварительная обработка графа; на этом этапе производится подсчет пересечений второго типа для всех пар гиперребер с учетом взаимного расположения;

2. решение оптимизационной задачи в точной постановке; целью этапа является минимизация пересечений гиперребер и отсутствие недопустимых наложений. В результате должна быть получена укладка графа, где каждому гиперребру сопоставляется уникальная горизонталь;

3. постобработка укладки графа.

4. Постановка оптимизационной задачи

Дан двухслойный гиперграф $H_2 = (V_1, V_2, E_{H_2})$, где $V_1 = \{u_1, \dots, u_{N_1}\}$ и $V_2 = \{v_1, \dots, v_{N_2}\}$ – множества вершин верхнего и нижнего слоя соответственно, а E_{H_2} – множество одноисточных гиперребер. Для каждой вершины $v \in V_1 \cup V_2$ двухслойного гиперграфа известны координаты $(x(v), y(v))$.

Каждое гиперребро $e_n = (u_n, T_n) \in E_{H_2}$ характеризуется списком $R_n = \{r_{n,1}, \dots, r_{n,|V_2|}\}$, элементы которого определены следующим образом:

$$r_{n,j} = \begin{cases} 1, & \text{если } v_j \in T_n \\ 0, & \text{если } x(v_j) \notin \text{int} \\ -1, & \text{если } v_j \notin T_n \text{ и } x(v_j) \in \text{int} \end{cases}, \quad j = 1, \dots, |V_2|,$$

где $\text{int} = \left[\min_{v \in T_n} \{x(v), x(u_n)\}, \max_{v \in T_n} \{x(v), x(u_n)\} \right]$ – это интервал на оси абсцисс, соответствующий горизонтальному сегменту гиперребра e_n .

Списки R_n и R_m позволяют определить число пересечений второго типа c_{e_n, e_m} и c_{e_m, e_n} между гиперребрами $e_n = (u_n, T_n)$ и $e_m = (u_m, T_m)$ в зависимости от взаимного расположения их горизонтальных сегментов. Для каждой пары гиперребер $e_n = (u_n, T_n)$ и $e_m = (u_m, T_m)$ составляется матрица

$$M_{n,m} = \begin{pmatrix} r_{n,1} & \dots & r_{n,|V_2|} \\ r_{m,1} & \dots & r_{m,|V_2|} \end{pmatrix}.$$

В случае, если горизонтальный сегмент гиперребра из вершины-источника u_η выше горизонтального сегмента гиперребра из вершины-источника u_μ , число пересечений второго типа c_{e_n, e_m} равно количеству столбцов вида $\begin{pmatrix} 1 \\ -1 \end{pmatrix}$ в матрице $M_{n,m}$. Действительно, такой столбец означает, что горизонтальный сегмент гиперребра e_m проходит над вершиной нижнего слоя, соответствующей столбцу, но не связан с ней сегментом sV_2 , в то время как горизонтальный сегмент гиперребра e_n , расположенный выше, связан с этой вершиной нижнего слоя вертикальным сегментом, а подобная прорисовка характеризуется пересечением второго типа.

Для расчета величины c_{e_m, e_n} (горизонтальный сегмент гиперребра вершины-источника e_m выше горизонтального сегмента гиперребра вершины-источника e_n), очевидно, необходимо в той же матрице $M_{n,m}$ подсчитать количество столбцов вида $\begin{pmatrix} -1 \\ 1 \end{pmatrix}$.

Оптимальная укладка двухслойного гиперграфа минимизирует число пересечений первого и второго типа для гиперребер, не допуская наложений, изображенных на рис. 2 (сверху). Если тем не менее избежать недопустимых наложений за счет взаимного расположения горизонтальных сегментов гиперребер не удастся, то тогда сдвиг одной или нескольких вершин гиперграфа по горизонтали приведет к нужному результату. Сдвиг вершины $v \in V_1 \cup V_2$ осуществляется на определенное целое число «единичных сдвигов» длины Δ , при этом если число отрицательное, то сдвиг выполняется влево, а если положительное – то вправо. Суммарное количество единичных сдвигов всех вершин также необходимо минимизировать.

При сформулированных ограничениях задача минимизации числа пересечений ортогональных гиперребер и числа сдвигов вершин гиперграфа может быть поставлена и решена как задача целочисленного программирования.

Введем в рассмотрение 8 видов целочисленных переменных Z' , Z'' , AZ' , AZ'' , HH , $CT1$, $CT2$ и BV .

Переменные Z'_i и Z''_j характеризуют количество единичных сдвигов вершин верхнего $u_i \in V_1$ и нижнего слоя $v_j \in V_2$ соответственно, а их знаки определяют направления сдвигов. Тогда новые (вследствие проведенных сдвигов) абсциссы вершин $u_i \in V_1$ и $v_j \in V_2$ будут равны

$$\begin{aligned} x'(u_i) &= x(u_i) + \Delta \times Z'_i, \\ x'(v_j) &= x(v_j) + \Delta \times Z''_j. \end{aligned}$$

Переменные AZ'_i и AZ''_j – это количество единичных сдвигов вершин u_i и v_j , соответственно, т.е. $AZ'_i = |Z'_i|$, а $AZ''_j = |Z''_j|$.

Для каждой пары гиперребер (e_n, e_m) , где $e_n = (u_\eta, T_\eta)$, $e_m = (u_\mu, T_\mu)$ и $x(u_\eta) < x(u_\mu)$ переменные HH , $CT1$ и $CT2$ определены следующим образом:

$$HH_{n,m} = \begin{cases} 0, & \text{если горизонтальный сегмент } e_m \text{ выше, чем } e_n, \\ 1, & \text{в противном случае.} \end{cases}$$

$$CT1_{n,m} = \begin{cases} 0, & \text{если нет пересечения первого типа,} \\ 1, & \text{в противном случае.} \end{cases}$$

$$CT2_{n,m} = \begin{cases} c_{e_n, e_m}, & \text{если горизонтальный сегмент } e_n \text{ выше, чем } e_m, \\ c_{e_m, e_n}, & \text{в противном случае.} \end{cases}$$

Переменные $BV_{\eta,j}$ – это булевы фиктивные переменные, которые вводятся для каждого гиперребра $e_n = (u_\eta, T_\eta)$ и множества вершин $\{v_j | v_j \in V_2 \setminus T_\eta\}$, т.е. вершин нижнего слоя, не смежных вершине u_η .

Выбор целевой функции задачи целочисленного программирования обусловлен целями:

- минимизация пересечений первого и второго типа;
- минимизация сдвигов вершин.

Обозначенным целям удовлетворяет следующий вид целевой функции:

$$L = w_z \sum_{i=1}^{|V_1|} AZ_i + w_z \sum_{j=1}^{|V_2|} AZ_j + \sum_{n=1}^{|E_{H_2}|-1} \sum_{m=n+1}^{|E_{H_2}|} (w_{c_1} \times CT1_{n,m} + w_{c_2} \times CT2_{n,m}) \rightarrow \min \quad (1)$$

Как упоминалось выше, сдвиг вершин осуществляется, только если невозможно избежать запрещенных наложений за счет перестановок горизонтальных сегментов гиперребер. Следовательно, значение весового коэффициента w_z можно принять равным

$$w_z = w_{c_1} + w_{c_2} \times \max_{e_n, e_m \in E_{H_2}} c_{e_n, e_m} + 1.$$

Система ограничений задачи целочисленного программирования имеет вид:

$$CT1_{n,m} \geq \frac{x(u_\eta) + \Delta Z'_\eta - (x(v_j) + \Delta Z'_j(v_j))}{K} - K HH_{n,m}, \\ e_n = (u_\eta, T_\eta) \in E_{H_2}, \quad e_m = (u_\mu, T_\mu) \in E_{H_2}, \quad (2)$$

$$x(u_\eta) < x(u_\mu), \quad v_j \in T_\mu, \quad x(v_j) = \min_{v \in T_\mu} x(v) \\ CT1_{n,m} \geq \frac{x(v_j) + \Delta Z'_j(v_j) - (x(u_\mu) + \Delta Z'_\mu)}{K} - K(1 - HH_{n,m}), \\ e_n = (u_\eta, T_\eta) \in E_{H_2}, \quad e_m = (u_\mu, T_\mu) \in E_{H_2}, \quad (3)$$

$$x(u_\eta) < x(u_\mu), \quad v_j \in T_\eta, \quad x(v_j) = \max_{v \in T_\eta} x(v) \\ CT2_{n,m} \geq c_{e_n, e_m} - K(1 - HH_{n,m}), \\ e_n = (u_\eta, T_\eta) \in E_{H_2}, \quad e_m = (u_\mu, T_\mu) \in E_{H_2}, \quad x(u_\eta) < x(u_\mu) \quad (4)$$

$$CT2_{n,m} \geq c_{e_m, e_n} - K HH_{n,m}, \\ e_n = (u_\eta, T_\eta) \in E_{H_2}, \quad e_m = (u_\mu, T_\mu) \in E_{H_2}, \quad x(u_\eta) < x(u_\mu) \quad (5)$$

$$x(v_j) + \Delta Z'_j(v_j) \leq x(u_\eta) + \Delta Z'_\eta + K BV_{\eta,j} + K HH_{n,m} - \varepsilon, \\ e_n = (u_\eta, T_\eta) \in E_{H_2}, \quad e_m = (u_\mu, T_\mu) \in E_{H_2}, \quad (6)$$

$$\begin{aligned}
& x(u_\eta) < x(u_\mu), \quad v_j \in T_\mu \\
& x(v_j) + \Delta Z_j^-(v_j) \leq x(u_\mu) + \Delta Z_\mu^+ + KBV_{\mu,j} + K(1 - HH_{n,m}) - \varepsilon, \\
& e_n = (u_\eta, T_\eta) \in E_{H_2}, \quad e_m = (u_\mu, T_\mu) \in E_{H_2},
\end{aligned} \tag{7}$$

$$\begin{aligned}
& x(u_\eta) < x(u_\mu), \quad v_j \in T_\eta \\
& x(u_\eta) + \Delta Z_\eta^- \leq x(v_j) + \Delta Z_j^-(v_j) + K(1 - BV_{\eta,j}) + KHH_{n,m} - \varepsilon, \\
& e_n = (u_\eta, T_\eta) \in E_{H_2}, \quad e_m = (u_\mu, T_\mu) \in E_{H_2},
\end{aligned} \tag{8}$$

$$\begin{aligned}
& x(u_\eta) < x(u_\mu), \quad v_j \in T_\mu \\
& x(u_\mu) + \Delta Z_\mu^- \leq x(v_j) + \Delta Z_j^-(v_j) + K(1 - BV_{\mu,j}) + K(1 - HH_{n,m}) - \varepsilon, \\
& e_n = (u_\eta, T_\eta) \in E_{H_2}, \quad e_m = (u_\mu, T_\mu) \in E_{H_2},
\end{aligned} \tag{9}$$

$$\begin{aligned}
& x(u_\eta) < x(u_\mu), \quad v_j \in T_\eta \\
& AZ_i^+ \geq Z_i^+, AZ_i^- \geq -Z_i^-, \quad u_i \in V_1
\end{aligned} \tag{10}$$

$$AZ_j^- \geq Z_j^-, AZ_j^+ \geq -Z_j^-, \quad v_j \in V_2 \tag{11}$$

$$x(u_{i+1}) + \Delta Z_{i+1}^- - (x(u_i) + \Delta Z_i^+) \geq \delta_w, \quad i = 1, \dots, |V_1| - 1 \tag{12}$$

$$x(v_{j+1}) + \Delta Z_{j+1}^+ - (x(v_j) + \Delta Z_j^-) \geq \delta_w, \quad j = 1, \dots, |V_2| - 1 \tag{13}$$

$$\begin{aligned}
& 0 \leq HH_{l,m} - HH_{l,n} + HH_{n,m} \leq 1, \\
& e_l = (u_\lambda, T_\lambda) \in E_{H_2}, \quad e_m = (u_\mu, T_\mu) \in E_{H_2}, \quad e_n = (u_\eta, T_\eta) \in E_{H_2},
\end{aligned} \tag{14}$$

$$x(u_\lambda) < x(u_\mu) < x(u_\eta)$$

Условия (2) – (3) для каждой пары гиперребер $e_n = (u_\eta, T_\eta) \in E_{H_2}$ и $e_m = (u_\mu, T_\mu) \in E_{H_2}$, таких что $x(u_\eta) < x(u_\mu)$, определяют наличие пересечений первого типа в зависимости от значения переменной $HH_{n,m}$ и от текущей абсциссы крайней левой (правой) вершины нижнего слоя для гиперребра, чей горизонтальный сегмент расположен выше.

Ограничения (4) – (5) определяют количество пересечений второго типа между каждой парой гиперребер гиперграфа, которое зависит от взаимного расположения горизонтальных сегментов двух гиперребер, т.е. от значения переменной $HH_{n,m}$.

Ограничения (6) – (9) для каждой пары гиперребер обеспечивают отсутствие недопустимых наложений за счет выполнения следующего условия: в случае, если горизонтальный сегмент гиперребра $e_m = (u_\mu, T_\mu)$ выше горизонтального сегмента гиперребра $e_n = (u_\eta, T_\eta)$, то ни одна из вершин-стоков вершины T_μ не должна лежать строго под вершиной u_η .

Отметим, что во всех формулах $K \gg 0$ – это произвольно выбранное большое положительное число, а $\varepsilon \ll 1$ – малая положительная величина.

Ограничения (10) – (11) необходимы для линеаризации условий $AZ_i^+ = |Z_i^+|$ и $AZ_j^- = |Z_j^-|$, которые определяют для каждой вершины $u_i \in V_1$ и $v_j \in V_2$ количество единичных сдвигов.

Условия (12) – (13) гарантируют соблюдение требования о том, что минимальное расстояние между соседними вершинами одного слоя равно δ_w .

Ограничение (14) обеспечивает выполнение условия транзитивности

для ординат горизонтальных сегментов любых 3 гиперребер гиперграфа.

Числовое решение задачи (1) – (14) дает возможность определить горизонтальные сдвиги вершин. Если при этом для гиперребра $e_n = (u_n, T_n)$ произошло изменение относительного расположения ее вершины-источника и крайней левой (правой) вершины-стока (например, вершина-источник была левее крайней левой вершины-стока $x(u_n) < x(v_j)$, а после сдвигов оказалась правее), то величина c_{e_n, e_m} для любого $m = 1, \dots, |E_{H_2}|$ и $n \neq m$ может изменить свое значение. В этом случае абсциссам вершин присваиваются новые значения, выполняется предварительная обработка измененных исходных данных и повторно решается оптимизационная задача (1) – (14). Если выбрать $w_z = w_{c_1} + w_{c_2} \times \max_{e_n, e_m \in E_{H_2}} c_{e_n, e_m} + 1$, то при повторном решении оптимизационной задачи абсциссы вершин гиперграфа не изменятся, поскольку существует как минимум одна перестановка горизонтальных сегментов без недопустимых наложений, которая была получена в результате решения оптимизационной задачи, давшей сдвиги вершин.

Наконец, для того чтобы определить, на каких горизонталях располагаются горизонтальные сегменты гиперребер, необходимо их всех отсортировать в соответствии со значениями переменных $HH_{n,m}$.

5. Числовые расчеты

Метод ветвей и границ [14] использован для решения задачи. При генерировании всех исходных данных, а также для формирования матрицы ограничений и вектора коэффициентов целевой функции применена компьютерная математическая среда Mathematica 11 [15], поиск числового решения выполнялся в оптимизаторе Gurobi Optimizer 7.0 [16].

На рис. 6 представлена укладка двухслойного гиперграфа, полученная в результате решения оптимизационной задачи (1) – (14). Были выбраны следующие значения параметров прорисовки:

$$\delta_h^{(1)} = \delta_h^{(2)} = \delta_h^{(3)} = \rho, \quad \delta_w = 3\rho.$$

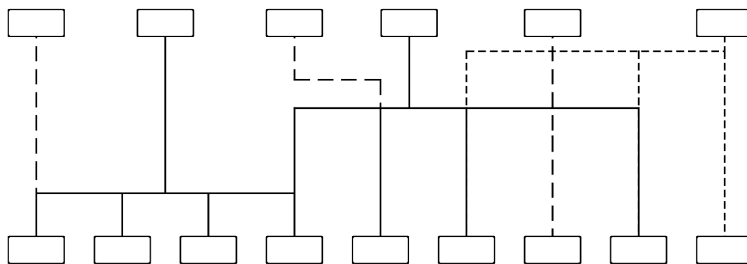


Рис. 6. Укладка двухслойного гиперграфа

Приведенный рисунок показывает, что требуется дополнительная обработка (постобработка) полученных решений, которая позволит сократить число горизонталей, т.е. вертикальный размер укладки гиперграфа за счет подъема горизонтальных сегментов гиперребер при сохранении оптимального числа пересечений и без нарушения всех требований к прорисовке.

6. Заключение

В статье сформулирована точная математическая постановка задачи минимизации пересечений гиперребер для представленных требований к прорисовке двухслойного гиперграфа. Числовое решение получено по методу ветвей и границ, расчеты выполнены с использованием компьютерных систем Mathematica 11 и Gurobi Optimizer 7.0, проанализированы результаты расчетов и определены направления дальнейших исследований.

Точная постановка (1) – (14) может быть использована для прорисовки – слойного гиперграфа, для чего необходимо последовательно решать оптимизационную задачу для каждой пары соседних слоев при фиксированном верхнем слое. Пример прорисовки многослойного гиперграфа, полученной таким способом, представлен на рис. 7.

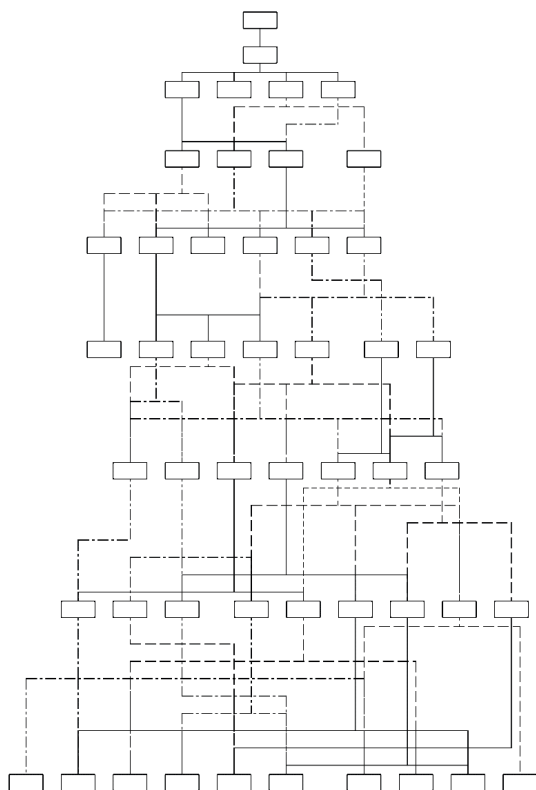


Рис. 7. Укладка многослойного гиперграфа

Список источников

1. Carstens J.J. Node and Label Placement in a Layered Layout Algorithm // *Master Thesis*, Christian-Albrechts-Universität zu Kiel, 2012.
2. Eichelberger H. Aesthetics and Automatic Layout of UML Class Diagrams. *PHD Thesis*, Bayerischen Julius-Maximilians-Universität Würzburg, 2005.
3. Eschbach T., Gunther W., Becker B. Orthogonal hypergraph drawing for improved visibility // *Journal of Graph Algorithms and Applications*, vol. 10, no. 2, 2006, pp. 141–157.
4. Sun. J. Automatic Orthogonal Graph Layout // *Hamburg University of Technology*, 2007.
5. Sander G. A fast heuristic for hierarchical Manhattan layout // *Graph Drawing*,

- vol. 1027 of the series Lecture Notes in Computer Science, 2005, pp. 447-458.
6. Sander G. Layout of directed hypergraphs with orthogonal hyperedges // *Graph Drawing*, vol. 2912 of the series Lecture Notes in Computer Science, 2003, pp. 381-386.
7. Schulze C.D. Optimizing Automatic Layout for Data Flow Diagrams // *Diploma Thesis*, Christian-Albrechts-Universität zu Kiel, 2011.
8. Schulze C.D., Spönemann M., R. von Hanxleden. Drawing Layered Graphs with Port Constraints // *Journal of Visual Languages & Computing*, 2013, pp. 1-34.
9. Siebenhaller M. Orthogonal Graph Drawing with Constraints: Algorithms and Applications // *PHD Thesis*, Eberhard-Karls-Universität, 2009.
10. Spönemann M. On the Automatic Layout of Data Flow Diagrams // *Diploma Thesis*, Christian-Albrechts-Universität zu Kiel, 2009.
11. Spönemann M., Schulze C.D., Rüegg U., R. von Hanxleden. Counting Crossings for Layered Hypergraphs // *Diagrammatic Representation and Inference*, vol. 8578 of the series Lecture Notes in Computer Science, 2014, pp. 9-15.
12. Sugiyama K., Tagawa S., Toda M. Methods for visual understanding of hierarchical systems // *IEEE Transactions on Systems, Man, and Cybernetics*, vol. SMC-11, no. 2, 1981, pp. 109-125.
13. Васильев Ю.М., Фридман Г.М. Визуализация кооперативных схем: гибридный эвристический алгоритм для минимизации количества пересечений ребер при укладке графа // *Известия Санкт-Петербургского государственного экономического университета*, 2017, no. 1.
14. Мину М. *Математическое программирование. Теория и алгоритмы*. Москва, издательство «Наука», 1990.
15. Доступно: <http://www.gurobi.com> (дата обращения: 12.10.16).
16. Доступно: <http://www.wolfram.com> (дата обращения: 12.10.16).

EXACT MATHEMATICAL FORMULATION FOR HYPEREDGES PORTRAYAL OF TWO-LAYER HYPERGRAPH IN COOPERTION SCHEME

Vasiliev Yuriy Mihailovich, Postgraduate

Saint-Petersburg State University of Economics, Sadovaya str., 21, St. Petersburg, Russia, 191023; e-mail: vas_yu_m@mail.ru

Purpose: the article contains a problem decision of minimization the crossings of hyperedges two-layer hypergraph in the exact statement. *Discussion:* hyperedges portrayer is the final stage of the Sugiyama method for drawing directed acyclic hierarchical graph. Professionals usually draw edges as horizontal or vertical segments for financial data flow diagrams and schemes of counterparty cooperation. Specialists use such designations at performance state and commercial projects. The author consider hierarchical graph as a hierarchical hypergraph. It differs from normal graph in that its edges connect a set of top layer vertex with some set of bottom layer vertex. The writer represents an initial hypergraph as a set of two-layer subgraphs for drawing hyperedges. They consist from two adjacent layers. The author solves visualization problem of hyperedges separately for each subgraph. *Results:* the writer suggests exact mathematical formulation of the crossings hyperedges minimization for drawing two-layer graph and analyzes numerical results with a help branch and bound method.

Keywords: directed acyclic hierarchical graph, two-layer graph, hypergraph, hyperedge, orthogonal hyperedge portrayal, drawing graph, branch and bound method.

References

1. Carstens J.J. Node and Label Placement in a Layered Layout Algorithm. *Master Thesis*, Christian-Albrechts-Universität zu Kiel, 2012.
2. Eichelberger H. Aesthetics and Automatic Layout of UML Class Diagrams. *PHD Thesis*, Bayerischen Julius-Maximilians-Universität Würzburg, 2005.
3. Eschbach T., Gunther W., Becker B. Orthogonal hypergraph drawing for improved visibility. *Journal of Graph Algorithms and Applications*, vol. 10, no. 2, 2006, pp. 141-157.
4. Sun J. Automatic Orthogonal Graph Lay-
out. *Hamburg University of Technology*, 2007.
5. Sander G. A fast heuristic for hierarchical Manhattan layout. *Graph Drawing*, vol. 1027 of the series Lecture Notes in Computer Science, 2005, pp. 447-458.
6. Sander G. Layout of directed hypergraphs with orthogonal hyperedges. *Graph Drawing*, vol. 2912 of the series Lecture Notes in Computer Science, 2003, pp. 381-386.
7. Schulze C.D. Optimizing Automatic Layout for Data Flow Diagrams. *Diploma Thesis*, Christian-Albrechts-Universität zu Kiel, 2011.

8. Schulze C.D., Spönemann M., R. von Hanxleden. Drawing Layered Graphs with Port Constraints. *Journal of Visual Languages & Computing*, 2013, pp. 1-34.
9. Siebenhaller M. Orthogonal Graph Drawing with Constraints: Algorithms and Applications. *PHD Thesis*, Eberhard-Karls-Universität, 2009.
10. Spönemann M. On the Automatic Layout of Data Flow Diagrams. *Diploma Thesis*, Christian-Albrechts-Universität zu Kiel, 2009.
11. Spönemann M., Schulze C.D., Rüegg U., R. von Hanxleden. Counting Crossings for Layered Hypergraphs. *Diagrammatic Representation and Inference*, vol. 8578 of the series Lecture Notes in Computer Science, 2014, pp. 9-15.
12. Sugiyama K., Tagawa S., Toda M. Methods for visual understanding of hierarchical systems. *IEEE Transactions on Systems, Man, and Cybernetics*, vol. SMC-11, no. 2, 1981, pp. 109-125.
13. Vasil'ev Iu.M., Fridman G.M. Vizualizatsiia kooperativnykh skhem: gibridnyi evristicheskii algoritm dlia minimizatsii kolichestva peresechenii reber pri ukladke grafa. *Izvestiia Sankt-Peterburgskogo gosudarstvennogo ekonomicheskogo universiteta*, 2017, no. 1. (In Russ.)
14. Minu M. *Matematicheskoe programirovanie. Teoriia i algoritmy*. Moscow, izdatel'stvo «Nauka», 1990. (In Russ.)
15. Available at: <http://www.gurobi.com> (accessed: 12.10.16). (In Russ.)
16. Available at: <http://www.wolfram.com> (accessed: 12.10.16). (In Russ.)