

Центры графов

Введение

Перед тем как непосредственно перейти к понятию центра графа, рассмотрим ряд задач, имеющих естественную графовую интерпретацию.

1. В небольшом городе строится пожарная часть. Каким образом выбрать место для строительства, чтобы время, за которое пожарники доедут в самый отдаленный от части район города, было минимальным?
2. Как создать в городе сеть ресторанов, чтобы из любой точки города можно было приехать в один из таких ресторанов менее чем за десять минут? (Ответом на данную задачу будет минимальное число ресторанов и места их расположения.)
3. Как наиболее оптимально разместить 5 служб доставки пиццы на дом, чтобы максимальное время пути курьера до любого дома было минимально.

Заметим, что все эти задачи сводятся к поиску некоторой точки (или в общем случае точек) в графе, удовлетворяющей некоторому критерию, при этом вершины соответствуют районам города, а ребра — дорогам. Подобные задачи часто встречаются в практической деятельности при размещении центров обслуживания. Цель таких задач — разместить некоторый(ые) объект(ы) так, чтобы минимизировать максимальное из расстояний (или время проезда) от этого объекта до какого-то района. По очевидным причинам этот класс задач принято называть **минимаксными задачами размещения**.

Договоренности

Граф G задается двумя множествами: множеством вершин V и множеством ребер E .

Каждому ребру соответствуют некоторое неотрицательное число, которое имеет смысл длины этого ребра (или времени проезда по нему). Расстояние от вершины x_i до вершины x_j обозначается $d(x_i, x_j) = d_{ij}$ и равняется *минимальной* из сумм длин ребер всех путей между этими вершинами. Будем считать, что длина ребра обладает аддитивностью. Это означает, что если поделить ребро некоторой точкой, то сумма длин получившихся отрезков равна длине ребра.

Каждой вершине x_i сопоставим некоторое положительное число v_i , которое имеет смысл важности или приоритета. Тогда **взвешенным расстоянием** между двумя вершинами x_i и x_j будем называть число $l_{ij} = l(x_i, x_j) := v_j d(x_i, x_j) = v_j d_{ij}$. Заметим, что взвешенное расстояние в общем случае не симметрично даже для неориентированного графа, т. е. $l_{ij} \neq l_{ji}$. Будем говорить, что точка y_2 достижима из точки y_1 не более чем за λ , если $l(y_1, y_2) \leq \lambda$.

Центр графа

Определения ([2], [3])

Эксцентриситет $e(x_i)$ вершины в связном графе G определяется как $\max\{d(x_i, x_j)\}$, $x_j \in V$.

Радиусом графа $r(G)$ называется наименьший из эксцентриситетов вершин. Вершина x_i называется **центральной вершиной** графа, если $e(x_i) = r(G)$. **Центр** графа — это множество центральных вершин. Вершина с наибольшим эксцентриситетом называется **периферийной** вершиной.

На рис. 1 представлены графы, у которых показан эксцентриситет каждой вершины.

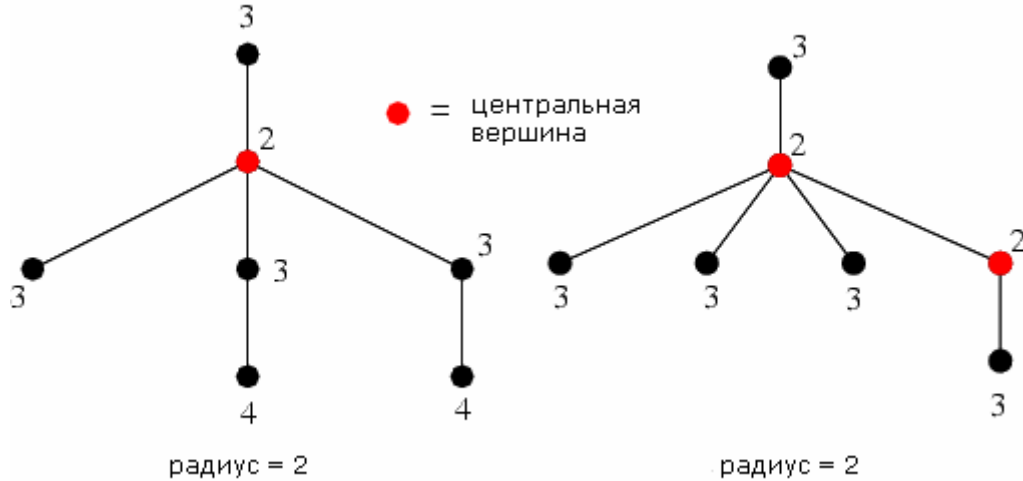


Рис. 1. Эксцентриситеты вершин графа

На рис. 2 приведен пример графов с одной и двумя центральными вершинами.

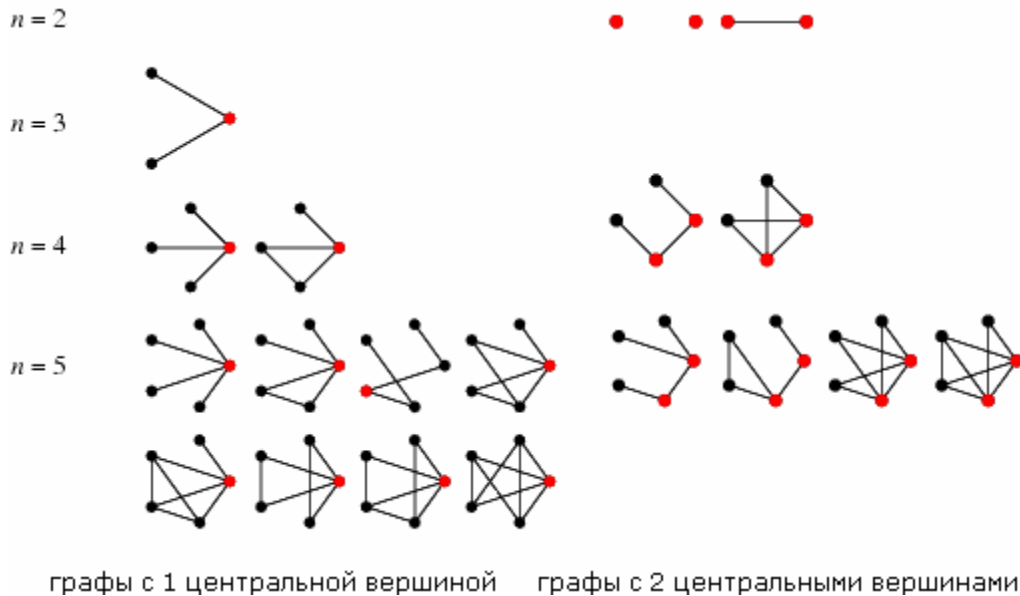


Рис. 2. Центр графа.

Теорема: Каждое дерево имеет центр, состоящий из одной вершины или из двух смежных вершин.

Доказательство: Утверждение очевидно для деревьев с одной и двумя вершинами. Покажем, что у любого другого дерева T те же центральные вершины, что и у дерева T' , полученного из T удалением всех его висячих вершин. Расстояние от данной вершины дерева u до любой другой вершины v достигает наибольшего значения, когда v – висячая вершина. Таким образом, эксцентриситет каждой вершины дерева T' точно на единицу меньше эксцентриситета этой же вершины в дереве T , следовательно, центры этих деревьев совпадают. Продолжим процесс удаления и получим требуемое. На рис. 3 и рис. 4 приведены деревья с одной и двумя центральными вершинами соответственно. ■

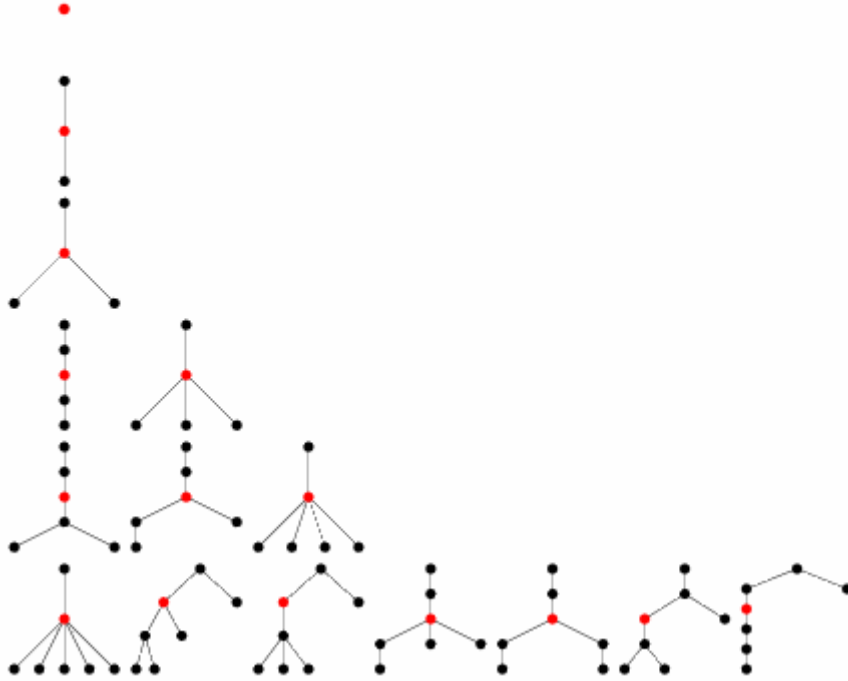


Рис. 3. Деревья с одной центральной вершиной

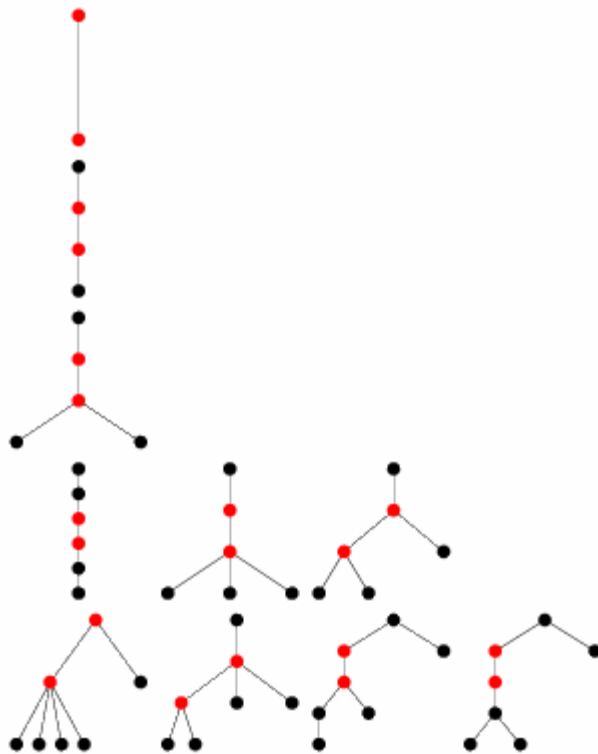


Рис. 4. Деревья с двумя центральными вершинами

Разделения

Замечание: в дальнейшем всегда будем использовать только взвешенное расстояние, дополнительно это не оговаривая.

Введем два множества:

© Андрей Клебанов, 2006

© <http://rain.ifmo.ru/cat/>

$$R_\lambda^o(x_i) := \{x_j \mid l_{ij} \leq \lambda, x_j \in V\},$$

$$R_\lambda^t(x_i) := \{x_j \mid l_{ji} \leq \lambda, x_j \in V\}.$$

$R_\lambda^o(x_i)$ — это множество всех вершин, расстояние от x_i до которых не больше λ , а $R_\lambda^t(x_i)$ — множество всех вершин, расстояние от которых до x_i не больше λ . Очевидно, что существуют такие λ , что эти множества не пусты (конечно, если в графе больше одной вершины).

Для каждой вершины определим два числа:

$$s_o(x_i) := \max(l_{ij}), x_j \in V,$$

$$s_t(x_i) := \max(l_{ji}), x_j \in V.$$

$s_o(x_i)$ и $s_t(x_i)$ называют соответственно **внешним** и **внутренним разделением** для вершины x_i .

Пусть λ_o — наименьшее значение λ такое, что для некоторой вершины x_i

$$R_\lambda^o(x_i) = V,$$

то есть длина пути от x_i до любой вершины графа не превосходит λ_o . Тогда $s_o(x_i) = \lambda_o$ (непосредственно следует из определений $R_\lambda^o(x_i)$ и $s_o(x_i)$).

Аналогично, λ_t — наименьшее значение λ такое, что для некоторой вершины x_i

$$R_\lambda^t(x_i) = V,$$

то есть длина пути от любой вершины графа до x_i не превосходит λ_t . Тогда $s_t(x_i) = \lambda_t$.

Очевидно, что у графа G числа внешнего и внутреннего разделения любой вершины конечны только тогда, когда граф сильно связный, то есть когда каждая вершина достижима из всякой другой вершины.

Определения ([1])

Вершина x_o^* такая, что

$$s_o(x_o^*) = \min[s_o(x_i)], x_i \in V,$$

называется **внешним центром** графа G . Аналогично, вершина x_t^* такая, что

$$s_t(x_t^*) = \min[s_t(x_i)], x_i \in V,$$

называется **внутренним центром**.

Значения $s_o(x_o^*)$ и $s_t(x_t^*)$ называются соответственно **внешним** и **внутренним радиусами** графа G и обозначаются ρ_o и ρ_t .

Так же допустимо определить **внешне-внутреннее разделение**:

$$s_{ot}(x_i) = \max\{v_j [d(x_i, x_j) + d(x_j, x_i)]\}, x_j \in V,$$

и **внешне-внутренний центр** как вершину, на которой достигается минимум внешне-внутреннего разделения.

Можно интерпретировать внешний центр как пожарный участок (минимально время на дорогу от участка к месту пожара), внутренний — как бомбоубежище (минимально время на дорогу туда), внешне-внутренний — как службу скорой помощи (минимально время на дорогу туда и обратно).

Алгоритм поиска

Приведем алгоритм поиска внешнего центра.

1. Построим матрицу $A_{n \times n}$ (n — мощность множества V), где $a_{ij} = d_{ij}$, то есть матрицу кратчайших путей. Для ее построения можно воспользоваться алгоритмом Флойда-Уоршелла или Дейкстры. Далее умножим каждый j -ый столбец матрицы A на v_j .

2. Внешнее разделение для вершины с номером i — это максимальное значение в i -ой строчке матрицы A . Подсчитаем максимум в каждой строчке. Таким образом, получим массив длины n , где i -ый элемент — внешнее разделение i -ой вершины.
3. Найдем наименьший элемент в этом массиве. Это будет наименьшее из внешних разделений. Вершина с таким разделением и есть внешний центр. В том случае, когда этих вершин несколько, все они — внешние центры.

Алгоритм поиска внутреннего центра отличается лишь на первом шаге: здесь j -ая строка матрицы A на умножается v_j .

Визуализатор данного алгоритма можно посмотреть в [5].

Абсолютный центр графа

Определения

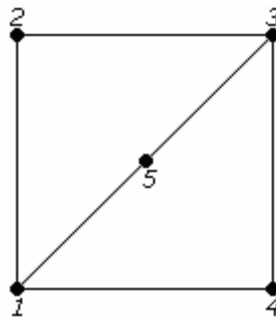
Если предположить, что центр обслуживания может находиться не только в каком-то районе, то есть в вершине, а вообще в любой точке графа, то нужно немного модифицировать данные выше определения. Для начала распространим понятия разделений на точки ребер графа. При этом в самих определениях фактически ничего не изменится. К примеру, так будет выглядеть определение внешнего разделения для произвольной точки графа:

$$s_o(y) := \max[v_d(y, x_i)], x_i \in V, y \in G.$$

Используя это расширенное определения разделений, определим абсолютный внешний и внутренний центр. **Абсолютный внешний центр** — это такая точка $y_o^* \in G$, у которой внешнее разделение минимально. Аналогично, **абсолютный внутренний центр** — это такая точка $y_t^* \in G$, у которой внутреннее разделение минимально.

Абсолютные радиусы определяются как значения разделений соответствующих абсолютных центров и обозначаются r_o и r_t .

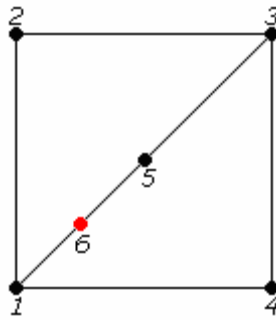
Пример:



Весы всех ребер и вершин равны 1, центр — каждая вершина. Матрица расстояний графа имеет вид

0	1	2	1	1
1	0	1	2	2
2	1	0	1	1
1	2	1	0	2
1	2	1	2	0

При добавлении красной точки в середину ребра 1 - 5, она становится абсолютным центром.



Для нее матрица расстояний имеет вид

0.5	1.5	1.5	1.5	0.5	0
-----	-----	-----	-----	-----	---

Таким образом, точка 6 более «центральна», чем любая из вершин графа G. ■

Алгоритм поиска

Отметим, что приведенный выше алгоритм поиска центра здесь использовать невозможно, так как количество «потенциальных абсолютных центров» бесконечно, а, следовательно, невозможно посчитать разделение каждой такой точки.

В ориентированном графе абсолютные центры не могут лежать на ориентированном ребре. Поэтому будем рассматривать неориентированный граф (в случае ориентированного графа можно выбросить из рассмотрения все направленные ребра, и тогда задача сведется к поиску абсолютного центра в неориентированном графе). В этом случае абсолютные внешний и внутренний центры совпадают. Поэтому нет смысла уточнять, какой именно центр мы ищем.

Рассмотрим два принципиально разных алгоритма.

Метод Хакими (S. L. Hakimi, 1964)

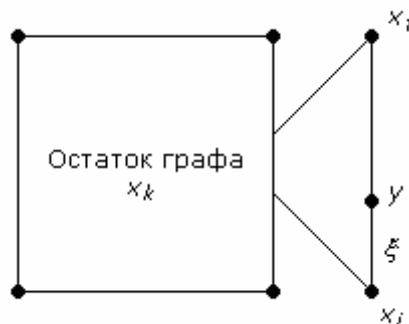
Этот алгоритм позволяет найти точное расположение абсолютного центра и значение его разделения (радиус). Он состоит всего из двух шагов:

1. Для каждого ребра найти точку с наименьшим разделением.
2. Из всех этих точек выбрать точку с наименьшим разделением – она и будет абсолютным центром.

Первый шаг метода осуществляется следующим образом.

Возьмем ребро x_{ij} . Для любой точки $y \in x_{ij}$ верно, что

$$s(y) = \max[v_k d(y, x_k)] = \max[v_k \min\{d(y, x_i) + d(x_i, x_k), d(y, x_j) + d(x_j, x_k)\}] = \max[v_k \min\{\xi + d(x_i, x_k), c_{ij} + d(x_j, x_k) - \xi\}], x_k \in V.$$



Для фиксированной вершины $x_k \in V$ и всякого значения ξ ($0 \leq \xi \leq c_{ij}$) можно найти наименьшие значения выражений, заключенных в квадратные скобки. Для этого выпишем отдельно два указанных выражения

$$T_k = v_k (\xi + d(x_i, x_k)),$$

$$T'_k = v_k (c_{ij} + d(x_j, x_k) - \xi)$$

и рассмотрим их, как функции от ξ . Далее построим нижнюю огибающую, для соответствующих им прямых линий. В действительности эта огибающая представляет собой ломаную линию, состоящую из двух «нижних» лучей (от точки пересечения) рассматриваемых прямых линий. Повторим эту процедуру для всех вершин и получим верхнюю огибающую для семейства нижних огибающих. Построенная огибающая может иметь несколько минимумов, наименьший из которых является абсолютным локальным центром, отвечающим данному ребру.

Описанный метод Хакими можно значительно оптимизировать. Идея состоит в том, чтобы заранее оценить значение абсолютного радиуса и не рассматривать те ребра, на которых, исходя из этих оценок, абсолютного центра быть не может.

Предположим, что абсолютный центр графа лежит на ребре e_{ij} , тогда значение абсолютного локального радиуса не меньше, чем

$$p_{ij} = \max[v_k \min\{d(x_i, x_k), d(x_j, x_k)\}], x_k \in V.$$

Следовательно,

$$P = \min(p_{ij}), e_{ij} \in E$$

– обоснованная нижняя оценка для абсолютного радиуса.

Пусть теперь абсолютный центр лежит в середине ребра e_{ij} , тогда абсолютный радиус

$$r = p_{ij} + v_k^* c_{ij} / 2,$$

где v_k^* – вес той вершины, в которой достигается $\max[v_k \min\{d(x_i, x_k), d(x_j, x_k)\}]$ (то есть той, которая определяет значение p_{ij}). Исходя из этого,

$$H = \min[p_{ij} + v_k^* d(x_i, x_j) / 2]$$

– обоснованная верхняя оценка для абсолютного радиуса.

Таким образом, всякое ребро $e_{ij} \in E$, для которого $p_{ij} \geq H$, можно не рассматривать. Визуализатор оптимизированного метода Хакими можно посмотреть в [6].

Итерационный алгоритм

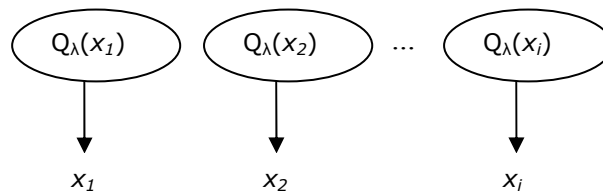
Его суть состоит в следующем.

1. Выбираем некоторое $\delta > 0$. Пусть λ изначально равна λ_0 .

$$\lambda_0 = \max[v_i v_j / (v_i + v_j) d(x_i, x_j)]$$

2. Найдем пересечение всех таких областей

$$Q_\lambda(x_i) = \{y \in G: l(y, x_i) \leq \lambda\}.$$



3. Если это пересечение непустое, то центр находится в нем. Если же оно пустое, то увеличим λ на δ и вернемся к шагу 2.

Кратные центры (p -центры)

Определения

В том случае, когда нам нужно разместить несколько объектов (как в задачах номер 2 и 3, приведенных в начале статьи), то можно рассматривать некоторое обобщение центра графа: не одну точку, а некоторое множество, состоящее из p точек, называемое кратным центром (p -центром). Пусть X_p – подмножество (содержащее p вершин) множества вершин графа G . Тогда определим наикратчайшее расстояние между вершинами множества X_p и вершиной x_i и наоборот:

$$d(X_p, x_i) = \min[d(x_j, x_i)], x_j \in X_p,$$

$$d(x_i, X_p) = \min[d(x_i, x_j)], x_j \in X_p.$$

Исходя из этого, дадим следующие определения: **внешним разделением множества вершин X_p** назовем

$$s_o(X_p) = \max[v_j d(X_p, x_i)], x_i \in V,$$

аналогично, **внутренним разделением множества вершин X_p** –

$$s_t(X_p) = \max[v_j d(x_i, X_p)], x_i \in V.$$

Множество вершин X_{po}^* такое, что

$$s_o(X_{po}^*) = \min[s_o(X_p)], X_p \subseteq V,$$

называется **внешним p -центром**, аналогично определяется **внутренний p -центр**.

Алгоритм поиска

Отметим, что здесь можно было бы воспользоваться алгоритмом, который перебирает все множества из p вершин и сравнивает их. Но такой алгоритм потребовал бы от нас $p(n-p)C_n^p$ операций при условии, что матрица расстояний уже построена. Подобное решение потребует очень много времени и больших вычислительных мощностей, поэтому здесь лучше использовать итерационный алгоритм. Он будет приведен далее для более общего случая абсолютных p -центров.

Абсолютные p -центры

Определения

Аналогично тому, как мы построили определение абсолютных центров, определим абсолютные p -центры. Пусть $X_p \subset G$, то есть элементы множества X_p могут лежать в любой точке графа: и на ребре, и в вершине.

Тогда наикратчайшее расстояние между вершинами множества X_p и вершиной x_i и наоборот:

$$d(X_p, x_i) = \min[d(y_j, x_i)], y_j \in X_p,$$

$$d(x_i, X_p) = \min[d(x_i, y_j)], y_j \in X_p$$

Отсюда, **внешнее разделение множества точек X_p** :

$$s_o(X_p) = \max[d(X_p, x_i)], x_i \in V,$$

аналогично, **внутреннее разделение множества точек X_p** :

$$s_t(X_p) = \max[d(x_i, X_p)], x_i \in V.$$

Множество точек X_{po}^* такое, что

$$s_o(X_{po}^*) = \min[s_o(X_p)], X_p \subseteq G,$$

называется **абсолютным внешним p -центром**, аналогично определяется **абсолютный внутренний p -центр**.

Алгоритм поиска

Решение задачи о поиске абсолютного p -центра намного сложнее, чем задача о поиске абсолютного центра. Метод Хакими, приведенный выше, невозможно обобщить на случай абсолютных p -центров. Рассмотрим быстро сходящийся итерационный алгоритм, обладающий следующими двумя преимуществами:

1. Процесс можно закончить сразу же, как только достигнута необходимая «точность» в расположении центров.
2. Метод легко видоизменить таким образом, чтобы можно было находить решения, близкие к оптимальному, и, следовательно, проводить анализ устойчивости решения.

Как и ранее, будем рассматривать неориентированный граф. Идея алгоритма состоит в построении специальных областей $\Phi_\lambda(V_k)$, где $V_k \subset V$ — множество из k вершин, таких, что любая вершина $v_i \in V_k$ достижима из любой точки этой области путем, длина которого не превосходит λ , а если $v_i \notin V_k$, то ни для какой точки $\Phi_\lambda(V_k)$ это не выполняется (другими словами, области не пересекаются).

Чтобы построить эти области, сначала строят множества $Q_\lambda(x_i) = \{y \in G : l(y, x_i) \leq \lambda\}$, то есть множество всех точек, из которых x_i достижима не более чем за λ .

Тогда $\Phi_\lambda(\emptyset) = \{y \in G\} \setminus (\cup_i Q_\lambda(x_i))$ — область, из которой ни одна вершина не доступна путем, не превосходящим λ . Здесь второй член исключает области графа, из которых можно достигнуть хотя бы одну вершину x_i .

$\Phi_\lambda(V_k)$ можно выразить через $Q_\lambda(x_i)$ следующим образом:

$$\Phi_\lambda(V_k) = (\cap Q_\lambda(x_i)) \setminus ((\cap Q_\lambda(x_i)) \cap (\cup Q_\lambda(x_j))),$$

где $x_i \in V_k$, а $x_j \in V \setminus V_k$. Здесь второй член исключает области, из которых достижимы необходимые вершины и еще хотя бы одна из оставшихся вершин графа.

Перейдем непосредственно к алгоритму.

1. Выберем некоторое $\delta > 0$. Положим $\lambda = \delta$.
2. Построим множества $Q_\lambda(x_i)$ для всех x_i .
3. При помощи $Q_\lambda(x_i)$ построим области $\Phi_\lambda(V_k)$.
4. Построим двудольный граф $G' = (V' \cup V, E')$, где V' — множество вершин, каждая из которых соответствует некоторой области $\Phi_\lambda(V_k)$, а E' — множество ребер, такое, что ребро между вершиной-областью и вершиной x_i существует тогда и только тогда, когда x_i может быть достигнута из этой области.
5. Найдем наименьшее доминирующее множество графа G' .

Для графа доминирующее множество вершин (называемое также внешне устойчивым множеством [3]) есть множество вершин $S \subseteq V$, выбранное так, что для каждой вершины $x_i \notin S$, существует ребро, исходящее из некоторой вершины множества S в x_i .

Доминирующее множество минимально, если нет другого доминирующего множества, содержащегося в нем.

Алгоритм нахождения наименьшего доминирующего множества описан в [1], [3] и [7].

6. Если число областей в этом множестве больше, чем p , то увеличим λ на δ и вернемся к шагу 2; в противном случае надо остановиться. Области этого множества образуют абсолютный p -центр графа G , а λ является **абсолютным p -радиусом**.

Остановимся подробнее на деталях реализации.

Предположим, что λ зафиксировано. Таким образом, мы получим задачу, где для заданного «критического» расстояния требуется найти наименьшее число центров и такое их размещение, что все вершины графа лежали в пределах этого критического расстояния (по крайней мере, каждая вершина — от ближайшего к ней центра).

Любое ребро графа либо достижимо целиком, либо частично, либо совсем не достижимо из вершины x_i . Если достижима только часть ребра (от какого-либо конца ребра до некоторой «предельной» точки), то над предельной точкой ставится метка. Эти метки содержат всю

необходимую информацию для описания множества $Q_\lambda(x_i)$. Таким образом, $Q_\lambda(x_i)$ состоит из точек всех ребер (или частей ребер), принадлежащих кратчайшим маршрутам между метками и вершиной x_i .

После размещения всех меток (для всех вершин) каждое ребро будет разделено на ряд участков, каждый из которых характеризуется теми вершинами, которые из него достижимы. Каждый участок может быть описан вектором из нулей и единиц длины $|V|$, где единица на i -ом месте означает достижимость вершины x_i из этого участка. Следовательно, совокупность всех участков с одинаковыми векторами образует область Φ_λ . Такое представление выгодно с вычислительной точки зрения, хотя оно и не содержит никакой информации о месте расположения области на графе.

На шаге 4 алгоритма требуется построить двудольный граф G' . Приводимая ниже теорема позволяет уменьшить его размеры, исключая те области, которые не влияют на получаемое оптимальное решение (но если существует несколько оптимальных решений, то при такой процедуре некоторые из них можно потерять).

Теорема: При заданной величине λ для получения некоторого минимального доминирующего множества графа G' можно предварительно исключить из V' вершины, соответствующие тем векторам, над которыми доминируют другие. Мы говорим, что $(SI)_1$ доминирует над $(SI)_2$, если $(SI)_1 \otimes (SI)_2 = (SI)_2$, где \otimes означает булево произведение. ■

Литература

1. Кристофидес Н. Теория графов. Алгоритмический подход. — М.: Мир, 1978.
2. Харари Ф. Теория графов. — М.: Мир, 1973.
3. Берж К. Теория графов и ее применения. — М.: Иностранная литература, 1962.

Статьи и визуализаторы на сайте CAT

4. Смаль А. Размещение центров. — 2004.
5. Алешкин А. Центры и кратные центры графа. — 2002.
6. Добровольский В. Метод Хакими. — 2001.
7. Горюнов А., Коломейцева Т. Покрытия. — 2004.