УДК 519.17

ПОСТРОЕНИЕ ПОКРЫВАЮЩЕГО ДЕРЕВА НАИМЕНЬШЕГО ВЕСА В ТЕОРИИ ГРАФОВ

Гречаников А. А., студ., Скоков М. А., студ., Коваленко А. В., ст. преп.

*Витебский государственный технологический университет,
г. Витебск, Республика Беларусь*

Реферат. *Данная работа посвящена исследованию покрывающих деревьев в теории графов. Описаны алгоритмы Прима и Краскала построения покрывающих деревьев минимального веса. В статье разработана математическая модель практической реализации этих алгоритмов. На основании этой модели созданы компьютерные программы, позволяющие определить наикратчайший маршрут в дереве, с учётом веса дуг графа.*

Ключевые слова: алгоритм Краскала, алгоритм Прима, вершина, вес, граф, дерево, ориентированное дерево, покрывающее дерево, связанный граф, суграф, ребро, цикл.

Пусть на плоскости задано некоторое конечное множество точек $\boldsymbol{V=\{v_1,v_2,...,v_n\}}$, которые назовём вершинами и линиями $\boldsymbol{R=\{r_1,r_2,...,r_m\}}$, соединяющими некоторые из вершин. Зададим граф $\boldsymbol{G=\langle V,R\rangle}$, как бинарное отношение множества $\boldsymbol{V}$ на множество $\boldsymbol{R}$, то есть $\boldsymbol{f:V\to R}$. Линия, соединяющая какие-либо две вершины графа, называется ребром или дугой графа. Точки пересечения некоторых дуг графа могут не являться вершинами графа. Если на графе указывается направление дуг, то граф является ориентируемым, в противном случае – неориентируемым. Граф $\boldsymbol{G}$ является связанным, если между любыми его двумя вершинами существует маршрут. Если в маршруте нет повторяющихся рёбер, то маршрут представляет собой цепь. Если при движении по рёбрам от начальной вершины $\boldsymbol{v}$ попадаем в эту же вершину $\boldsymbol{v}$, то полученный маршрут образует цикл.

Определим дерево, как конечный, неориентированный граф, который не имеет ни одного цикла.

Покрывающее дерево графа $\boldsymbol{G=\langle V,R\rangle}$ представляет собой связанный суграф $\boldsymbol{H_1=\langle V,R_H\rangle}$, где $\boldsymbol{R_H\subseteq R}$, который не имеет циклов. Любое покрывающее дерево, которое имеет $\boldsymbol{n}$ вершин, содержит $\boldsymbol{n-1}$ ребро.

В теории графов покрывающие деревья широко используются при разработке различных алгоритмов на графах. Они находят применение в различных прикладных задачах, таких как сетевая и графовая теория, теория алгоритмов, логистика. При этом не рассматривается задача о построении логической модели, описывающей алгоритм исследования деревьев. В связи с этим возникла задача построения математической модели, которая даёт возможность определить путь в заданном графе, при прохождении по которому издержки были бы минимальными.

Для построения математической модели, минимального по ценности покрывающего дерева, наиболее удобно использовать алгоритмы Прима и Краскала, которые позволяют построить покрывающие деревья наименьшего веса. Для решения поставленной задачи разработаны компьютерные программы, каждого из алгоритмов, которые позволяют определять маршрут по дугам графа минимальной ценности. Программы, которые выполняют определение минимального цикла в заданном взвешенном графе, выполнены на языке программирования C#.

Рассмотрим связанный граф $\boldsymbol{G}$. Имеется несколько алгоритмов для построения покрывающего дерева. Одним из алгоритмов является алгоритм «поиска в глубину». Однако этот алгоритм позволяет получить невзвешенное дерево, каждое ребро которого имеет постоянный вес, в общем случае вес дуги принимают за единицу. При изучении реального технологического процесса, для которого построен граф, надо учитывать ценность ребра.

Построим для заданного графа G_1, задающий некоторый технологический процесс, взвешенное покрывающее дерево наименьшего веса, то есть каждое ребро графа имеет свой вес. Тогда вес покрывающего дерева будет равен сумме весов всех его составляющих рёбер.

Предположим, что известна матрица смежности Sm связанного взвешенного графа G_1, которая изображена на рисунке 1.

Матрица смежности

```
0, 8, 0, 0, 0, 7, 13
8, 0, 2, 0, 0, 0, 11
0, 2, 0, 20, 0, 0, 3
0, 0, 20, 0, 5, 0, 4
0, 0, 0, 5, 0, 6, 15
7, 0, 0, 0, 6, 0, 14
13, 11, 3, 4, 15, 14, 0
```

Рисунок 1 – Матрица смежности Sm графа G_1

В результате имеем наглядное представление заданного графа G_1 с выделенным взвешенным покрывающим деревом наименьшего веса равным 27 (рис. 2).

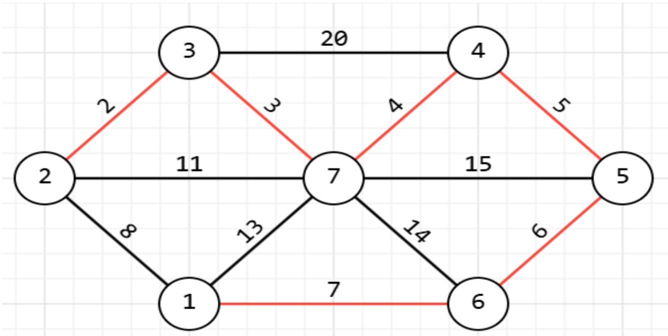

Рисунок 2 – Связанный граф G_1

Рассмотрим алгоритм Прима построения покрывающего дерева наименьшего веса.

1. Положить $R_H = \varnothing$, $V_T = \{v_i\}$, где v_i и – произвольная вершина графа.
2. Найти ребро $\{v_i; v_j\}$ наименьшего веса, которое соединяет вершину $v_i \in V_H$ с вершиной $v_j \in V \setminus V_H$. Если такой вершины нет, то прекращаем вычисления, так как связанный граф не имеет покрывающего дерева.
3. Положить $V_H := V_H \cup \{v_j\}$, $R_H := R_H \cup \{v_i, v_j\}$.
4. Положить $V_H = V$, то прекратить вычисления, так как покрывающее дерево построено, его рёбра содержат множество R_H.

В алгоритме Прима предполагается, что задан связанный граф, представленный матрицей смежности Sm, в которой каждый элемент определяет вес ребра. В результате применения алгоритма Прима построен граф наименьшего покрывающего дерева, который выводится программой в виде выделенных линий с указанием наименьшего веса.

Для наглядной демонстрации работы алгоритма Прима разработан программный продукт, который позволяет выполнять следующие операции:

1) сформировать матрицу смежности для взвешенного графа;
2) наглядно изобразить заданный граф и минимальное покрывающее дерево;
3) начертить рёбра минимального покрывающего дерева поверх исходного графа;
4) вычислить наименьший вес покрывающего дерева в графе.

Результат работы компьютерной программы «Алгоритм Прима» для взвешенного графа G_1 с заданной матрицей смежности Sm (рис.1) приведён на рисунке 3.

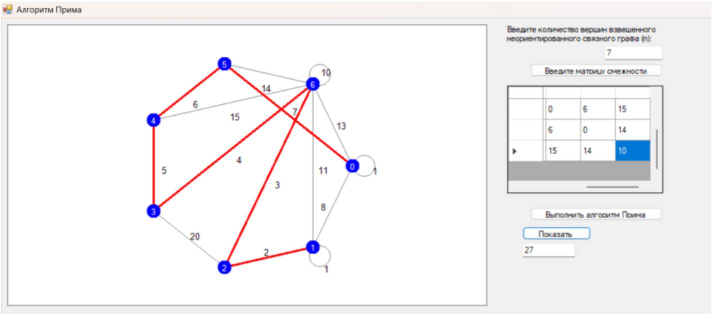

Рисунок 3 – Алгоритм Прима

Для полноты системного анализа рассмотрим ещё один алгоритм построения наименьшего покрывающего дерева, такой как алгоритм Краскала. В алгоритме Краскала будем обозначать $v = w$, если вершины $v, w \in V$ принадлежат одной и той же компоненте связанности графа G.

1. Сортировать рёбра заданного графа в порядке возрастания их весов. Результатом сортировки является перечень упорядоченных рёбер L.
2. Положить $R_H = \varnothing$. Положить также, что каждая вершина графа G образует одну отдельную компоненту связности, то есть $v \neq w$ для всех $v, w \in V$.
3. Если $L = \varnothing$, то закончить вычисления – заданный граф не имеет покрывающего дерева. В противном случае выбрать первое ребро $\{v; w\}$ из перечня L. Положить $L := L \setminus \{v; w\}$.
4. Если $v \neq w$, то положить $R_H := R_H \cup \{v; w\}$, $v := w$, $k := k + 1$.
5. При выполнении неравенства $k < n - 1$ переходим к третьему пункту. Если неравенство не выполняется, то это означает, что искомое покрывающее дерево построено.

Для наглядной демонстрации работы алгоритма Краскала разработан программный продукт, выполняющий задачи, сопоставимые с задачами алгоритма Прима.

Итоговый результат работы приложения «Алгоритм Краскала» для взвешенного графа G_1 с сгенерированной матрицей смежности Sm (рис.1) отображён на рисунке 4.

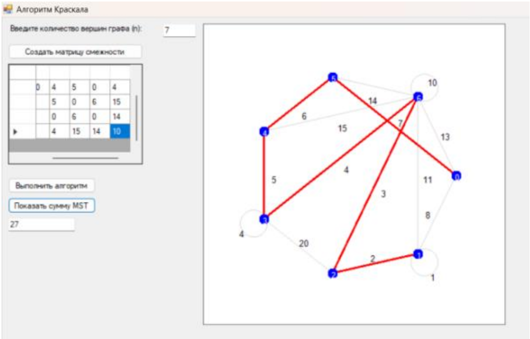

Рисунок 4 – Алгоритм Краскала

Вывод: в статье созданы логические модели для алгоритма Прима и алгоритма Краскала, определения взвешенного покрывающего дерева наименьшего веса в связанном графе. Используя построенные модели, разработаны приложения на языке C#, которые рассчитывают минимальный вес покрывающего дерева для произвольной матрицы смежности любого графа.

Список использованных источников

1. Флегонтов, А. В. Моделирование задач принятия решений при нечётких исходных данных / А. В. Флегонтов, В. Б. Вилков, А. К. Черных. – 2-е изд., стер. – Санкт-Петербург : Лань, 2023. – 332 с.

УДК 338.5

ОПРЕДЕЛЕНИЕ ОПТИМАЛЬНЫХ СТРАТЕГИЙ ЦЕНООБРАЗОВАНИЯ В УСЛОВИЯХ КОНКУРЕНЦИИ

Ковзова А. В., студ., Густова И. А., студ., Никонова Т. В., к.ф.-м.н., доц.

Витебский государственный технологический университет,
г. Витебск, Республика Беларусь

Реферат. *В статье исследуется некооперативные взаимодействия фирм в условиях дуополии. Анализ сопровождается примерами решения задач с использованием таких моделей как модель Курно (одновременный выбор объема продаж) и Штакельберга (последовательный выбор объема продаж).*

Ключевые слова: некооперативное взаимодействие, функция реакции, стратегическая переменная, выпуск конкурента, максимизация прибыли, предельные издержки, равновесие.

Некооперативное взаимодействие компаний на рынке дуополии остается одной из ключевых тем в экономической теории. Для анализа выделим две модели с одинаковой стратегической переменной (объем продаж): Курно и Штакельберга, выбор которых зависит от последовательности принятия решений (одновременность или последовательность).

Модель Курно – одна из первых и наиболее известных моделей дуополии. В ней фирмы одновременно и независимо определяют объем производства, предполагая, что объем конкурента неизменен [1].

Рассмотрим применение данной модели на конкретном примере. На рынке представлены две спортивные компании: компания *A* и компания *B*, которые производят спортивные костюмы. Объем производства каждой компании обозначается Q_A и Q_B. Общий объем производства продукции определяется по формуле (1):

$$Q = Q_A + Q_B \tag{1}$$

Рыночная цена спортивных костюмов зависит от общего объема производства и описывается функцией (2):

$$P(Q) = 100 - Q \tag{2}$$

Издержки компании *A* составляют $C_A(Q_A) = 10Q_A$, а компании *B* – $C_B(Q_B) = 10Q_B$.

Для начала, нам нужно найти функции реакции каждой фирмы. Функция реакции показывает оптимальный объем производства фирмы *A* в зависимости от объема производства фирмы *B*, и наоборот.

Прибыль *A* рассчитывается по формуле (3) [2]:

$$\pi_A = P(Q) \times Q_A - C_A(Q_A) = (100 - Q_A - Q_B) \times Q_A - 10Q_A \tag{3}$$

Далее максимизируем прибыль *A*. Найдем производную формулы (3) и приравняем к нулю:

$$\frac{d\pi_A}{dQ_A} = 100 - 2Q_A - Q_B - 10 = 0,$$
$$Q_A = 45 - 0.5Q_B. \tag{4}$$

Полученное уравнение (4) является функцией реакции компании *A*.

Аналогично посчитаем прибыль фирмы *B* и максимизируем её. Таким образом, получим функцию реакции фирмы *B* (6)

$$\pi_B = P(Q) \times Q_B - C_B(Q_B) = (100 - Q_A - Q_B) \times Q_B - 10Q_B, \tag{5}$$
$$Q_B = 45 - 0.5Q_A. \tag{6}$$