

УДК624.953.014.2.004.15

МОДЕЛИРОВАНИЕ КОМБИНАТОРНЫХ ЗАДАЧ НА ГРАФАХ В ТЕРМИНАХ ЗАДАЧИ МАТЕМАТИЧЕСКОГО ПРОГРАММИРОВАНИЯ

СЕМЕНЕЦ С. Н.^{1*}, к. т. н, доц.,НАСОНОВА С. С.^{2*}, к. т. н, доц.

^{1*} Кафедра прикладной математики, Государственное высшее учебное заведение «Приднепровская государственная академия строительства и архитектуры», ул. Чернышевского, 24-а, 49600, Днепропетровск, Украина, тел. +38(056)756-34-10, e-mail: semenets.serg@list.ru, ORCID ID: 0000-0003-0477-8795

^{2*} Кафедра высшей математики, Государственное высшее учебное заведение «Украинский государственный химико-технологический университет», пр. Гагарина, 8, 49005, Днепропетровск, Украина, тел. +38 (0562) 47-24-64, e-mail: ms.nasonova@list.ru, ORCID ID: 0000-0002-0920-7417

Аннотация. Постановка проблемы. Различают три основных подхода к решению комбинаторных задач на графах. Первый предусматривает разработку соответствующих алгоритмов на основе методов теории графов. Второй и третий подходы основаны на приведении исходной задачи, сформулированной в терминах графов, к задаче оптимизации. При этом для решения полученной задачи оптимизации в рамках второго подхода разрабатываются соответствующие алгоритмы оптимизации, а в рамках третьего – используются прикладные компьютерные технологии, инструментальная среда которых адаптирована к решению задач оптимизации. Для компьютерной реализации первого и второго подходов требуется специальное программное обеспечение, разработка которого доступна далеко не каждому пользователю. Кроме того, в случае уточнения постановки той или иной типовой задачи на графах (например, путем введения дополнительных ограничений) обычно оказывается необходимой модификация известных алгоритмов ее решения и соответствующая ревизия программного обеспечения. Все это в значительной мере затрудняет на практике проведение численных экспериментов на графовых моделях. Наиболее удобным для широкого круга пользователей является третий подход к решению комбинаторных задач на графах, поскольку он не требует для своей компьютерной реализации разработки специальных алгоритмов и соответствующего программного обеспечения. Однако вопросы согласования получаемой модели оптимизации и возможностей применяемой компьютерной технологии требуют дальнейшей научной и практической проработки. В данной статье рассматривается один из возможных способов реализации третьего подхода к решению комбинаторных задач на графах, предусматривающий приведение исходной графовой модели к задаче математического программирования и последующее решение последней в инструментальной среде надстройки MS Excel «Поиск решения». **Цель статьи** – показать результативность и достаточную общность такого способа решения комбинаторных задач на графах. **Выводы.** Полученные в статье результаты показывают, что многие комбинаторные задачи, сформулированные в терминах графов, могут быть достаточно легко пересформулированы в виде задачи математического программирования. Получаемая при этом оптимизационная модель, как правило, оказывается линейной относительно неизвестных. Для численной реализации таких моделей хорошо приспособлена надстройка MS Excel «Поиск решения», что делает табличный процессор Excel эффективной компьютерной технологией решения комбинаторных задач на графах даже в случае их достаточно большой размерности.

Ключевые слова: граф, модель, оптимизация, комбинаторные задачи на графах, задача математического программирования.

МОДЕЛЮВАННЯ КОМБІНАТОРНИХ ЗАДАЧ НА ГРАФАХ У ТЕРМІНАХ ЗАДАЧІ МАТЕМАТИЧНОГО ПРОГРАМУВАННЯ

СЕМЕНЕЦЬ С. М.^{1*}, к. т. н, доц.,НАСОНОВА С. С.^{2*}, к. т. н, доц.

^{1*} Кафедра прикладної математики, Державний вищий навчальний заклад «Придніпровська державна академія будівництва та архітектури», вул. Чернишевського, 24-а, 49600, Дніпропетровськ, Україна, тел. +38 (056) 756-34-10, e-mail: semenets.serg@list.ru, ORCID ID: 0000-0003-0477-8795

^{2*} Кафедра вищої математики, Державний вищий навчальний заклад «Український державний хіміко-технологічний університет», пр. Гагаріна, 8, 49005, Дніпропетровськ, Україна, тел. +38 (0562) 47-24-64, e-mail: ms.nasonova@list.ru, ORCID ID: 0000-0002-0920-7417

Анотація. Постановка проблеми. Розрізняють три основні підходи до вирішення комбінаторних задач на графах. Перший передбачає розроблення відповідних алгоритмів на основі методів теорії графів. Другий і третій підходи засновані на приведенні вихідної задачі, сформульованої в термінах графів, до задачі оптимізації. При цьому для розв'язання отриманої задачі оптимізації у рамках другого підходу розробляються відповідні алгоритми оптимізації, а у рамках третього застосовуються прикладні комп'ютерні технології, інструментальне середовище яких адаптоване до розв'язання таких задач. Для комп'ютерної реалізації першого і другого підходів потрібне спеціальне програмне забезпечення, розроблення якого доступне далеко не кожному користувачеві. Крім того, у

разі уточнення постановки тієї або іншої типової задачі на графах (наприклад, шляхом уведення додаткових обмежень) зазвичай виявляється необхідною модифікація відомих алгоритмів її розв'язання і відповідна ревізія програмного забезпечення. Усе це значною мірою утрудняє на практиці проведення числових експериментів на графових моделях. Найбільш зручним для широкого кола користувачів є третій підхід до розв'язання комбінаторних задач на графах, оскільки він не вимагає для своєї комп'ютерної реалізації розроблення спеціальних алгоритмів і відповідного програмного забезпечення. Проте питання узгодження отриманої моделі оптимізації і можливостей вживаної комп'ютерної технології вимагають подальшого наукового і практичного опрацювання. У даній статті розглядається один із можливих способів реалізації третього підходу до розв'язання комбінаторних задач на графах, що передбачає приведення вихідної графової моделі до задачі математичного програмування і подальше розв'язання останньої в інструментальному середовищі надбудови MS Excel «Пошук рішення». **Мета статті** – показати результативність і достатню спільність такого способу розв'язання комбінаторних задач на графах. **Висновки.** Отримані в статті результати показують, що багато комбінаторних задач, сформульованих у термінах графів, можуть бути досить легко переформульовані у вигляді задачі математичного програмування. Отримана при цьому оптимізаційна модель, як правило, виявляється лінійною щодо невідомих. Для числової реалізації таких моделей добре адаптована надбудова Excel «Пошук рішення», що робить табличний процесор Excel ефективною комп'ютерною технологією розв'язання комбінаторних задач на графах навіть у разі їх досить великої розмірності.

Ключові слова: граф, модель, оптимізація, комбінаторні задачі на графах, задача математичного програмування

DESIGN OF COMBINATORICS TASKS ON COLUMNS IN TERMS OF TASK MATHEMATICAL PROGRAMMING

SEMENETS S. N.^{1*}, *Cand. Sc. (Tech), Ass. Prof.*,

NASONOVA S. S.^{2*}, *Cand. Sc. (Tech), assoc. Prof.*

^{1*} Department of the applied mathematics, State Higher Education Establishment «Pridneprov's'ka State Academy of Civil Engineering and Architecture», 24-A, Chernishevskogo str., Dnipropetrovsk 49600, Ukraine, тел. +38 (056) 756-34-10, e-mail: semenets.serg@list.ru, ORCID ID: 0000-0003-0477-8795

^{2*} Department of higher mathematics, State Higher Education Establishment «Ukrainian State chemical - technological university», Gagarin str., 8, Dnipropetrovsk 49005, Ukraine, tel. +38(0562) 47-24-64, e-mail: ms.nasonova@list.ru, ORCID ID: 0000-0002-0920-7417

Summary. Raising of problem. To the decision of combinatorics tasks on columns distinguish three basic approaches. The first envisages development of corresponding algorithms on the basis of methods of theory of the graphs. Second and third approaches are based on bringing the initial task over, set forth in terms of counts, to the task of optimization. Thus for the decision of the got task of optimization within the framework of the second approach the proper algorithms of optimization are developed, and within the framework of the third – the applied computer technologies the instrumental environment of which is adapted to the decision of tasks of optimization are used. For computer realization of the first and second approaches the special software development of which is accessible to far not every user is required. In addition, in the case of clarification of raising of one or another model task on columns (for example, by introduction of additional limitations) usually modification of the known algorithms of its decision and proper revision of software appears necessary. All this makes it very difficult in practice, the implementation of numerical experiments on a graph model. The most convenient for a wide range of users is a third approach to the solution of combinatorial problems on graphs, since it does not require for its computer implementation, the development of specific algorithms and software. However, issues of harmonization and optimization of the resulting model the ability to apply computer technology require further research and practical study. In this article one of possible methods of realization of the third going is examined near the decision of combinatorics tasks on columns, envisaging bringing an initial count model over to the task of the mathematical programming and subsequent decision of the last in an instrumental environment building on of Excel "Solver". **Purpose of the article** – to show effectiveness and sufficient community of such method of decision of combinatorics tasks on columns. **Conclusion.** The results got in the article show that many combinatorics tasks set forth in terms of counts, it is undifficult reformulate as a task of the mathematical programming. The optimization model got here, as a rule, appears linear relatively unknown. For numeral realization of such models building on of MS Excel is well adjusted "Solver", that does the tabular processor of Excel effective computer technology of decision of combinatorics tasks on columns even in case of their large enough dimension.

Keywords: graph, model, optimization, combinatorial problems on graphs, mathematical programming problem

Постановка проблеми. Комбінаторними задачами на графах називаються задачі пересчета и перечисления, сформулированные в терминах графов. При этом граф вы-

ступает в роли модели исследуемого объекта, а решение конкретной задачи сводится к поиску некоторого подграфа, пути, цикла, множества вершин или ребер, обладающих тре-

буемыми свойствами. Обычно эти требования задаются в виде некоторого критерия выбора наилучшего в определенном смысле решения из множества возможных. Тем самым, комбинаторные задачи на графах по своей сути – это задачи принятия решений. В настоящее время различные графовые модели широко используются для формализации многих прикладных задач как технического, так и экономического характера [1-4], а разработка эффективных методов численной реализации таких моделей представляет теоретический и практический интерес [2; 6].

Различают три основных подхода к решению комбинаторных задач на графах. Первый предусматривает разработку соответствующих алгоритмов на основе методов теории графов [4-8]. Вторым и третьим подходами основаны на приведении исходной задачи, сформулированной в терминах графов, к задаче оптимизации [9; 11]. При этом для решения полученной задачи оптимизации в рамках второго подхода разрабатываются соответствующие алгоритмы оптимизации, а в рамках третьего – используются прикладные компьютерные технологии, инструментальная среда которых адаптирована к решению таких задач.

Для компьютерной реализации первого и второго подходов требуется специальное программное обеспечение, разработка которого доступна далеко не каждому пользователю. Кроме того, в случае уточнения постановки той или иной типовой задачи на графах (например, путем введения дополнительных ограничений) обычно оказывается необходимой модификация известных алгоритмов ее решения и соответствующая ревизия программного обеспечения. Все это в значительной мере затрудняет проведение на практике численных экспериментов на графовых моделях.

Наиболее удобным для широкого круга пользователей является третий подход к решению комбинаторных задач на графах, поскольку он не требует для своей реализации разработки специальных алгоритмов и соответствующего программного обеспечения. Однако вопросы согласования получаемой модели оптимизации и возможностей применяемой компьютерной технологии для ее ре-

ализации требуют дальнейшей научной и практической проработки.

В данной статье рассматривается один из возможных способов реализации третьего подхода к решению комбинаторных задач на графах, предусматривающий приведение исходной графовой модели к задаче математического программирования и последующее решение последней в инструментальной среде надстройки MS Excel «Поиск решения». Цель статьи – показать результативность и достаточную общность такого способа решения комбинаторных задач на графах.

Основная часть. В данной статье рассматриваются четыре классические графовые модели, которые приводятся к задаче математического программирования, а для численного решения последней используется MS Excel.

1. Задача о наименьшем доминирующем множестве вершин

Задача о наименьшем доминирующем множестве вершин графа формулируется следующим образом. Пусть $G = \langle V, E \rangle$ – некоторый граф, который задан множеством своих вершин V и матрицей смежности S . Требуется найти доминирующее множество вершин графа, имеющее наименьшее число элементов. (Доминирующим множеством вершин U графа $G = \langle V, E \rangle$ называется такое подмножество множества вершин V , что для каждой вершины v_i , не входящей в U , существует ребро, соединяющее хотя бы одну вершину множества U с вершиной v_i). Если трактовать вершины графа как охраняемые объекты, а элементы матрицы смежности s_{ij} рассматривать как заданные бинарные параметры, равные 1, если объект v_j виден (может контролироваться) из объекта v_i , и равные 0 – в противном случае, то получим известную задачу о часовых, в которой требуется расставить часовых по охраняемым объектам так, чтобы все объекты были взяты под контроль при минимальном количестве часовых.

Пусть n – число вершин графа G ; $x_i (i = 1, n)$ – бинарная переменная, равная 1,

если вершина v_i входит в минимальное доминирующее множество, и равная 0 – в противном случае. Тогда задачу о наименьшем доминирующем множестве вершин графа можно записать в виде следующей оптимизационной модели:

$$\begin{aligned} \sum_{i=1}^n x_i &\rightarrow \min \\ \sum_{i=1}^n x_i s_{ij} &\geq 1, j = \overline{1, n}; \\ x_i &\in \{0, 1\}, i = \overline{1, n}. \end{aligned}$$

Данная модель – это задача линейного целочисленного программирования. Целевой функцией определяется число вершин, включенных в доминирующее множество. Первая группа из n ограничений – это условия того, что каждая вершина, не входящая в доминирующее множество, должна иметь общее ребро хотя бы с одной вершиной из доминирующего множества. Вторая группа из n ограничений – это условия двоичности переменных x_i .

В качестве примера найдем наименьшее доминирующее множество вершин графа G_1 , показанного на рисунке 1. Модель оптимизации при этом содержит 7 неизвестных, а ее численная реализация в инструментальной среде MS Excel дала следующие результаты: $x_1 = 0$; $x_2 = 1$; $x_3 = 0$; $x_4 = 0$; $x_5 = 0$; $x_6 = 0$; $x_7 = 0$. Это значит, что наименьшее доминирующее множество вершин графа G_1 состоит из вершин 2 и 6, т. е. $U_{\min} = \{2, 6\}$.

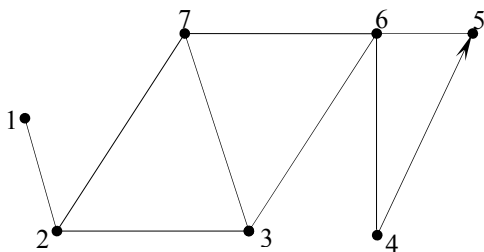


Рис.1. Граф G_1

Следует отметить, что полученное оптимальное решение не является единственным, поскольку граф G_1 имеет еще одну пару вершин $\{2, 4\}$, также образующую доминирующее множество.

2. Задача о кратчайшем пути

Задача о кратчайшем пути на графе формулируется следующим образом. Пусть $G = \langle V, E \rangle$ – связный граф, который задан множеством своих вершин V и матрицей смежности S . Известна весовая матрица графа P , элементы которой p_{ij} задают вес (обычно трактуемый как длина) соответствующих ребер. Даны две произвольные вершины $v_i, v_q \in V$. Требуется найти путь наименьшего веса (кратчайший путь) из v_i в v_q .

Сформулируем задачу о кратчайшем пути как задачу математического программирования. Пусть $n = |V|$ – число вершин графа; x_{ij} – бинарная переменная, равная 1, если ребро $(i, j) \in E$ принадлежит оптимальному маршруту из v_i в v_q , и равную 0 – в противном случае. Тогда, учитывая то, что все маршруты из v_i в v_q , содержащие циклы, заведомо не являются оптимальными, задачу о кратчайшем пути можно записать в виде следующей оптимизационной модели:

$$\begin{aligned} \sum_{i=1}^n \sum_{j=1}^n x_{ij} p_{ij} s_{ij} &\rightarrow \min \\ \sum_{i=1}^n x_{ii} s_{ii} &= 0; \sum_{j=1}^n x_{ij} s_{ij} = 1; \\ \sum_{i=1}^n x_{iq} s_{iq} &= 1; \sum_{j=1}^n x_{qj} s_{qj} = 0; \\ \sum_{i=1}^n x_{ik} s_{ik} &= \sum_{j=1}^n x_{kj} s_{kj}, k = \overline{1, n}, k \neq i, k \neq q; \\ \sum_{i=1}^n x_{ik} s_{ik} + \sum_{j=1}^n x_{kj} s_{kj} &\leq 2, k = \overline{1, n}, k \neq i, k \neq q; \\ x_{ij} &\in \{0, 1\}, i, j \in \overline{1, n}. \end{aligned}$$

Данная модель относится к классу задач линейного целочисленного программирования. Целевой функцией задается вес маршрута, связывающего вершины v_i и v_q . Первая пара ограничений – это условия того, что степень входа начальной вершины должна быть равна 0, а степень выхода равна 1. Вторая пара ограничений – это условия того, что степень входа конечной вершины должна быть равна 1, а степень выхода равна 0. Следующие две группы ограничений – это балансо-

вые условия для промежуточных вершин. Смысл их состоит в том, что для каждой промежуточной вершины, через которую проходит маршрут, должно выполняться условие равенства входящих и выходящих ребер и при этом количество инцидентных ребер не может быть больше 2 (в любую промежуточную вершину можно зайти и выйти из нее только 1 раз). Последняя группа ограничений задает условия двоичности переменных x_{ij} . Решением сформулированной задачи являются оптимальные значения неизвестных x_{ij} , которые определяют ребра графа, составляющие кратчайший путь из v_i в v_q , и, соответственно, вес этого маршрута.

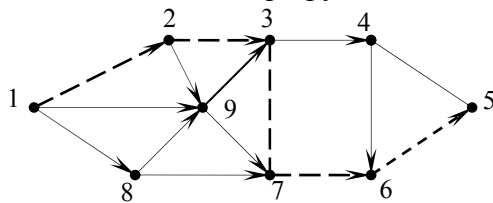


Рис. 2. Граф G_2

Таблица 1

Весовая матрица графа G_2

	1	2	3	4	5	6	7	8	9
1		10						5	1
2			15						5
3				40			10		
4					20	5			
5				20		25			
6					25				
7			10			10			
8							20		15
9			25				30		

В качестве примера рассмотрим задачу выбора кратчайшего пути из вершины 1 в вершину 5 на смешанном графе G_2 , изображенном на рисунке 2. Весовая матрица, задающая длину ребер, показана в таблице 1.

Задача выбора кратчайшего пути на графе G_2 имеет 16 неизвестных. Ее решение в среде MS Excel привело к следующим результатам: $x_{12} = 1$; $x_{18} = 0$; $x_{19} = 0$; $x_{23} = 1$; $x_{29} = 0$; $x_{87} = 0$; $x_{89} = 0$; $x_{93} = 0$; $x_{97} = 0$; $x_{37} = 1$; $x_{73} = 0$; $x_{34} = 0$; $x_{46} = 0$; $x_{45} = 0$; $x_{76} = 1$; $x_{65} = 1$.

Таким образом, получили такой кратчайший маршрут, связывающий вершины 1 и 5: 1-2-3-7-6-5 (см. рис. 2). Его протяженность равна 70.

3. Задача о кратчайшем гамильтоновом цикле

Задача о кратчайшем гамильтоновом цикле на графе формулируется следующим образом. Пусть $G = \langle V, E \rangle$ – простой связный граф, имеющий, по крайней мере, два гамильтоновых контура, задан множеством своих вершин V и матрицей смежности S . Известна весовая матрица графа P , элементы которой p_{ij} задают вес соответствующих ребер. Требуется найти гамильтонов цикл наименьшего веса (кратчайший гамильтонов цикл). Если интерпретировать вершины как населенные пункты, ребра – как дороги, их соединяющие, а вес p_{ij} – как длину соответствующей дороги, то получим известную задачу коммивояжера, в которой бродячему торговцу требуется обойти несколько городов и вернуться обратно, посещая каждый город только 1 раз, и при этом затратить минимальное время на переходы.

Сформулируем задачу о кратчайшем гамильтоновом цикле как задачу математического программирования. Пусть $n = |V|$ – число вершин графа; $I = \overline{1, n}$; J – некоторое подмножество множества I ($J \subset I$); x_{ij} – бинарная переменная, равная 1, если ребро (i, j) из E принадлежит оптимальному гамильтонову контуру, и равную 0 – в противном случае. Тогда задачу о кратчайшем гамильтоновом цикле можно записать в виде следующей оптимизационной модели:

$$\sum_{i=1}^n \sum_{j=1}^n x_{ij} p_{ij} s_{ij} \rightarrow \min$$

$$\sum_{j=1}^n x_{ij} s_{ij} = 1, i \in I;$$

$$\sum_{j=1}^n x_{ji} s_{ji} = 1, i \in I;$$

$$\sum_{i \in I} \sum_{j \in I \setminus J} x_{ij} s_{ij} \geq 1, 2 \leq |J| \leq n - 2, J \subset I;$$

$$x_{ij} \in \{0, 1\}, i, j \in I.$$

Таблиця 2

Весовая матрица графа G_3

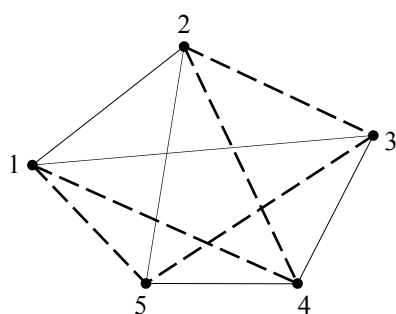
	1	2	3	4	5
1		30	35	10	15
2	30		10	15	40
3	35	10		35	10
4	10	15	35		20
5	15	40	10	20	

4. Задача о максимальном потоке в сети

Транспортной системой называется транспортная инфраструктура, включающая в себя транспортные узлы (терминалы) и транспортные коммуникации (коридоры), предназначенная для транспортировки некоторого продукта (продукт здесь понимается в широком смысле, это может быть, например, нефть, газ, лес, поезда, вагоны, пароходы, самолеты и т. д.). Транспортные коммуникации служат для перемещения продукта, а транспортные узлы – для перераспределения (грузо)потока продукта. Сеть (или транспортная сеть) – это связный ориентированный взвешенный граф. Сети (их вершины обычно называют узлами) традиционно используются для моделирования транспортных систем. При этом ребрам ставятся в соответствие транспортные коммуникации, а вершинам (узлам) – транспортные узлы. Истоком в сети называется узел, степень входа которого равна 0. Исток рассматривается как источник, из которого выходит продукт. Стоком в сети называется узел, степень выхода которого равна 0. Сток рассматривается как приемник, в который приходит продукт. Пропускной способностью ребра (i, j) называется максимальное количество продукта p_{ij} , которое может пропустить ребро за единицу времени. Аналогично, пропускной способностью сети называется максимальное количество продукта, которое может пропустить сеть за единицу времени. Поток в сети – это функция, которая каждому ребру сети (i, j) ставит в соответствие неотрицательное число $x_{ij} \leq p_{ij}$, равное фактическому количеству продукта, проходящего через ребро за единицу времени. Величина потока в сети оценивается количеством продукта, входящего в сток за единицу времени. Поток называется максимальным, если его величина равна пропускной способ-

Данная модель – это задача линейного целочисленного программирования. Целевой функции задается вес гамильтонова контура. Первая группа из n ограничений – это условия того, что степень входа каждой вершины должна быть равна 1 (в любую вершину можно войти только 1 раз). Вторая группа из n ограничений – это условия того, что степень выхода каждой вершины должна быть равна 1 (из любой вершины можно выйти только 1 раз). Следующая группа ограничений – это условия, исключающие возможное появление неполных циклов. Последняя группа ограничений задает условия двоичности переменных x_{ij} . Решением сформулированной задачи являются оптимальные значения неизвестных x_{ij} , которые определяют ребра графа, образующие гамильтонов цикл наименьшего веса.

В качестве примера рассмотрим задачу выбора кратчайшего гамильтонова цикла на полном графе G_3 , изображенном на рисунке 3. Весовая матрица, задающая длину ребер, показана в таблице 2.

Рис. 3. Граф G_3

Задача выбора кратчайшего гамильтонова цикла на графе G_3 имеет 20 неизвестных. Ее решение в среде MS Excel привело к следующим результатам: $x_{12} = 0$; $x_{13} = 0$; $x_{14} = 0$; $x_{15} = 1$; $x_{21} = 0$; $x_{23} = 0$; $x_{24} = 1$; $x_{25} = 0$; $x_{31} = 0$; $x_{32} = 1$; $x_{34} = 0$; $x_{35} = 0$; $x_{41} = 1$; $x_{42} = 0$; $x_{43} = 0$; $x_{45} = 0$; $x_{51} = 0$; $x_{52} = 0$; $x_{53} = 1$; $x_{54} = 0$.

Таким образом, получили такой кратчайший гамильтонов цикл: 1-5-3-2-4-1 (см. рис. 3). Его протяженность равна 60.

ности сети.

Задача о максимальном потоке в сети формулируется следующим образом. Пусть $G = \langle V, E \rangle$ – сеть, которая определена множеством своих узлов V , матрицей смежности S и весовой матрицей P , элементы которой p_{ij} задают пропускную способность соответствующих ребер. Заданы исток и сток сети, соответственно узлы v_i, v_q . Требуется найти максимальный поток в сети.

Сформулируем задачу о максимальном потоке в сети как задачу математического программирования. Пусть $n = |V|$ – число узлов сети; $x_{ij} (0 \leq x_{ij} \leq p_{ij})$ – переменная, равная фактическому количеству продукта, проходящего через ребро $(i, j) \in E$ за единицу времени. Тогда задачу о максимальном потоке можно записать в виде следующей оптимизационной модели:

$$\sum_{i=1}^n x_{iq} s_{iq} \rightarrow \min$$

$$\sum_{j=1}^n x_{ij} s_{ij} = \sum_{i=1}^n x_{iq} s_{iq};$$

$$\sum_{i=1}^n x_{ik} s_{ik} = \sum_{j=1}^n x_{kj} s_{kj}, k = \overline{1, n}, k \neq t, k \neq q;$$

$$0 \leq x_{ij} \leq p_{ij}, i, j \in \overline{1, n}.$$

Данная модель – это классическая задача линейного программирования. Целевая функция определяет величину потока в сети. Первое ограничение – это условия того, что количество продукта, выходящего из истока, должно быть равно количеству продукта, заходящему в сток (рассматривается идеальная сеть без потерь).

Следующая группа из $n - 2$ ограничений – это балансовые условия для промежуточных узлов. Смысл их состоит в том, что в каждой промежуточный узел должно войти и выйти из него одинаковое количество продукта.

Последняя группа ограничений – это граничные условия значений неизвестных x_{ij} . Решением сформулированной задачи являются оптимальные значения неизвестных x_{ij} , которые определяют количество продукта,

проходящего через соответствующие ребра, а также величину максимального потока в сети. В качестве примера найдем максимальный поток сети, показанной на рисунке 4. Пропускная способность ребер задана весовой матрицей (табл. 3.).

Таблица 3

Весовая матрица сети G_4

	1	2	3	4	5	6	7	8	9	10
1		20	20							
2				30						
3					20					
4						15	30			
5						5		10		
6									10	
7										20
8										30
9							20	20		
10										

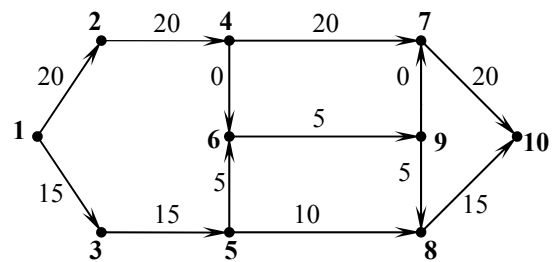


Рис. 4. Сеть G_4

Задача о максимальном потоке на графе G_4 имеет 13 неизвестных. Ее решение в среде MS Excel привело к следующим результатам: $x_{12} = 20$; $x_{13} = 15$; $x_{24} = 20$; $x_{35} = 15$; $x_{47} = 20$; $x_{46} = 0$; $x_{58} = 10$; $x_{56} = 5$; $x_{69} = 5$; $x_{79} = 0$; $x_{710} = 20$; $x_{88} = 5$; $x_{810} = 15$. Таким образом, пропускная способность сети G_5 равна $x_{710} + x_{810} = 35$. Соответствующее этой величине пропускной способности распределение продукта по ребрам показано на рисунке 4.

Выводы. Полученные в статье результаты показывают, что многие комбинаторные задачи, сформулированные в терминах графов, могут быть достаточно легко переформулированы в виде задачи математического программирования. Получаемая при этом оптимизационная модель, как правило, оказывается линейной относительно неизвестных. Для численной реализации таких моделей хо-

рошо приспособлена надстройка MS Excel технологией решения комбинаторных задач «Поиск решения», что делает табличный процессор Excel эффективной компьютерной на графах даже в случае их достаточно большой размерности.

ВИКОРИСТАНІ ДЖЕРЕЛА

1. Современная логистика / Джеймс С. Джонсон, Дональд Ф. Вуд, Дэниел Л. Вордлоу, Поль Р. Мэрфи-мл. ; [пер. с англ. А. И. Мороза, С. Г. Тригуб]. – 7-е изд. – Москва : Вильямс, 2002. – 615 с.
2. Ловас Л. Прикладные задачи теории графов. Теория паросочетаний в математике, физике, химии / Л. Ловас, М. Пламмер ; пер. с англ. Г. П. Гаврилова [и др.] ; под ред. Г. П. Гаврилова. – Москва : Мир, 1998. – 653 с. : ил.
3. Касьянов В. Н. Графы в программировании: обработка, визуализация и применение / В. Н. Касьянов, В. А. Евстигнеев. – Санкт-Петербург : БХВ-Петербург, 2003. – 1104 с. – (Научное издание).
4. Костюкова Н. И. Графы и их применение. Комбинаторные алгоритмы для программистов / Н. И. Костюкова. – Москва, 2007. – 310 с. – (Основы информационных технологий).
5. Майника Э. Алгоритмы оптимизации на сетях и графах : пер. с англ. / Э. Майника. – Москва : Мир, 1981. – 323 с.
6. Алгоритмы: построение и анализ : пер. с англ. / Т. Кормен, Ч. Лейзерсон, Р. Ривест, К. Штайн. – Москва : Вильямс, 2006. – 1296 с.
7. Зыков А. А. Основы теории графов / А. А. Зыков. – Москва : Вузовская книга, 2004. – 664 с.
8. Горбатов В. А. Фундаментальные основы дискретной математики. Информационная математика / В. А. Горбатов. – Москва : Наука : Физматлит, 2000. – 544 с.
9. Берж К. Теория графов и ее применение / К. Берж ; пер. с фр. и послесл. А. А. Зыкова ; под ред. И. А. Вайнштейна. – Москва : Иностран. лит., 1962. – 319 с.
10. Кофман А. Введение в прикладную комбинаторику / Кофман А. ; пер. с фр. В. П. Мякишева, В. Е. Тараканова ; под ред. Б. А. Севастьянова. – Москва : Наука, 1975. – 479 с.
11. Ахо А. Построение и анализ вычислительных алгоритмов : пер. с англ. / А. Ахо, Дж. Хопкрофт, Дж. Ульман ; пер. с англ. А. О. Слисенко ; под ред. Ю. В. Матиясевича. – Москва : Мир, 1979. – 536 с.
12. Басакер Р. Конечные графы и сети : пер. с англ. / Р. Басакер, Т. Саати. – Москва : Наука, 1974. – 368 с.

REFERENCES

1. Johnson J.C., Wood D.F., Vordlou D.L. and Murphy-ml P.R. *Sovremennaya logistika* [Modern Logistics]. Moscow: Vil'yams, 2002, 615 p. (in Russian).
2. Lovas L. and Plammer M. *Prikladnye zadachi teorii grafov. Teoriya parosochetaniy v matematike, fizike, khimii* [Applied tasks of graph theory. Matchings theory in mathematics, physics, chemistry]. Moscow: Mir, 1998, 653 p. (in Russian).
3. Kas'yanov V.N. and Evstigneev V.A. *Grafy v programmirovanii: obrabotka, vizualizatsiya i primeneniye* [Graphs in programming: processing, visualization and application]. Sankt-Peterburg: BKHV Peterburg, 2003, 1104 p. (in Russian).
4. Kostyukova N.I. *Grafy i ikh primeneniye. Kombinatornye algoritmy dlya programmistov* [Graphs and their application. Combinatorial algorithms for programmers]. Moscow, 2007, 310 p. (in Russian).
5. Majnika E. *Algoritmy optimizatsii na setyakh i grafakh* [Optimization algorithms on networks and graphs]. Moscow: Mir, 1981, 323 p. (in Russian).
6. Kormen T., Lejzerson Ch., Rivest R. and Shtain K. *Algoritmy: postroenie i analiz* [The algorithms for working with graphs]. Moscow: Vil'yams, 2006, 1296 p. (in Russian).
7. Zykov A.A. *Osnovy teorii grafov* [Fundamentals of graph theory]. Moskva: Vuzovskaya kniga, 2004, 664 p. (in Russian).
8. Gorbатов V.A. *Fundamental'nye osnovy diskretnoy matematiki. Informatsionnaya matematika* [Fundamentals of discrete mathematics. Information Mathematics]. Moscow: Nauka, Fizmatlit, 2000, 544 p. (in Russian).
9. Berzh K. *Teoriya grafov i ee primeneniye* [Graph theory and its applications]. Moscow: Inostran. lit., 1962, 319 p. (in Russian).
10. Kofman A. *Vvedenie v prikladnuyu kombinatoriku* [Introduction to applied combinatorics]. Moscow: Nauka, 1975, 479 p. (in Russian).
11. Akho A., Khopkroft D. and Ul'man D. *Postroenie i analiz vychislitel'nykh algoritmov* [Construction and analysis of computational algorithms]. Moscow: Mir, 1979, 536 p. (in Russian).
12. Basaker R. and Saati T. *Konechnye grafy i seti* [Finishing graphs and networks]. Moscow: Nauka, 1974, 368 p. (in Russian).

Рецензент: д-р т. н., проф. Н. М. Ершова

Надійшла до редколегії: 29.08.2015 р. Прийнята до друку: 18.10.2015 р.