Паньгин Андрей Александрович,
Паньгин Александр Викторович

ЗАДАЧИ НА ПЕРЕПРАВУ. МОДЕЛИРОВАНИЕ. АЛГОРИТМЫ

Аннотация

Рассматривается широкий класс задач на переправу с учетом различных ограничений на связи между персонажами. С помощью графов представлены алгоритмы решения, реализованные программно на алгоритмическом языке Pascal и в электронных таблицах Excel.

Ключевые слова: задачи на переправу, поиск в ширину на графе, моделирование в MS Excel.

ПОСТАНОВКА ЗАДАЧИ

Задачи данного типа обычно используются для развития алгоритмического стиля мышления школьников (в игровой форме). В коллекции цифровых образовательных ресурсов (ЦОР) [1] в состав информационного источника сложной структуры (ИИСС) «Математика на компьютерах» входит имитационная модель переправы с конструктором задач и заданием различных ограничений на связи между персонажами (рис. 1).

Эти ограничения могут быть следующего вида:

– задание грузоподъемности или вместимости лодки;

– присутствие лодочника;

– отдельный персонаж «конфликтует» с другим персонажем или объектом («коза-капуста»);

– отдельный персонаж «конфликтует» с другим персонажем без присутствия (или при присутствии) третьего типа персонажа («муж–жена», «рыцарь–оруженосец»);

– количество одного типа персонажей больше количества другого типа персонажей («разбойники-купцы»).

Примеры задач на переправу, представленные в сборнике [2], демонстрируют отдельные логические приемы для ограниченного количества персонажей.

В этой статье сделан подход к формализации данного типа задач и программного получения решения с учетом снятия этих ограничений.

В качестве модели переправы рассматривается граф состояний, а за решение задачи принимается кратчайший путь на этом графе из начального состояния, где все персонажи располагаются на одном берегу, в конечное состояние, где все персонажи находятся на противоположном берегу.

ОПРЕДЕЛЕНИЯ И ОБОЗНАЧЕНИЯ

Персонаж – любой субъект действия или предмет, который может участвовать в переправе с одного берега на другой: например, волк, коза или капуста. Отдельным персонажем, всегда присутствующим в любой задаче на переправу, является лодка. В некоторых задачах подразумевается, что лодка управляется лодочником. Это означает, что персонаж «лодка» может переплывать реку самостоятельно, без участия других персонажей. Несколько персонажей могут быть однотипными. В этом случае одному наименованию сопоставляется некоторое количество персонажей данного типа, например, два солдата.

Рис. 1

N – количество различных типов персонажей. В задаче про волка, козу и капусту $N = 4$ (лодка – еще один тип персонажа, который всегда присутствует в количестве 1 шт.).

C_i – количество персонажей i-го типа.

Левый берег – берег реки, на котором находятся все персонажи до начала переправы.

Правый берег – противоположный берег, куда необходимо переправить всех персонажей.

Состояние – набор $V = (A_1, A_2, ..., A_N)$, где A_i – количество персонажей i-го типа, находящихся в указанный момент на левом берегу. Такой набор V однозначно определяет расположение всех персонажей в данный момент переправы. Противоположным состоянием назовем набор $V' = (B_1, B_2, ..., B_N)$, где B_i – количество персонажей i-го типа на правом берегу. Если $V = (A_1, A_2, ..., A_N)$, то $V' = (C_1 - A_1, C_2 - A_2, ..., C_N - A_N)$ – это очевидно, кроме того, $(V')' = V$.

Недопустимое состояние – состояние, противоречащее условиям задачи, например то, в котором волк и коза находятся на одном берегу, а лодка – на противоположном. Очевидно, если V – недопустимое состояние, то V' является также недопустимым.

M – количество всех состояний переправы в данной задаче, как допустимых, так и недопустимых.

Ход – однократное перемещение лодки от одного берега к другому, пустой либо с группой персонажей, помещающихся в лодке. Ход изменяет состояние переправы.

ГРАФ СОСТОЯНИЙ

Количество всех возможных состояний переправы M равно числу всех возможных наборов $(A_1, A_2, ..., A_N)$, где A_i – целое число в диапазоне от 0 до C_i. Значит, $M = (C_1 + 1) \cdot (C_2 + 1) \cdot ... \cdot (C_N + 1)$. Таким образом, состояния можно пронумеровать от 0 до $M - 1$. Ниже приведен канонический способ нумерации, при котором состоянию $V_K = (A_1, A_2, ..., A_N)$ сопоставляется номер K, равный значению числа $A_N ... A_2 A_1$ в позиционной системе счисления с переменным основанием: от $(C_1 + 1)$ для младшего разряда до $(C_N + 1)$ для старшего разряда.

$V_0 = (0, 0, ..., 0)$
$V_1 = (1, 0, ..., 0)$
$V_{C1} = (C_1, 0, ..., 0)$
$V_{C1+1} = (0, 1, ..., 0)$
...
$V_{C1+C2} = (0, C_2, ..., 0)$
$V_{C1+C2+1} = (1, C_2, ..., 0)$
...
$V_{M-1} = (C_1, C_2, ..., C_N)$

Начальным состоянием назовем V_{M-1}, в котором все персонажи находятся на левом берегу, а конечным состоянием – V_0, в котором все персонажи переправлены на правый берег.

Построим ориентированный граф $G(V, E)$, где G – Graph (граф), V – Vertices (множество вершин), E – Edges (множество дуг). Этот граф содержит M вершин, отождествляемых с состояниями V_0, ..., V_{M-1}, в котором дуга из V_i в V_j существует, только если из состояния V_i можно попасть в состояние V_j за один ход. Задача о переправе сводится к поиску кратчайшего пути на таком графе из вершины V_{M-1} в вершину V_0. На рис. 2 показан пример графа для классической задачи про волка, козу и капусту.

Чтобы удовлетворять условиям задачи, кратчайший путь из V_{M-1} в V_0 не должен проходить через недопустимые состояния, поэтому необходимо исключить из рассмотрения все такие вершины. Для этого достаточно исключить из графа все дуги, входящие в вершины недопустимых состояний. На рис. 2 такие дуги обозначены розовым цветом.

РЕШЕНИЕ ЗАДАЧИ ПЕРЕПРАВЫ СРЕДСТВАМИ MS EXCEL

Способы решения различных задач на графах с помощью Excel рассматривается как замена программирования стандартных алгоритмов на графах [3].

Рассмотрим решение оптимизационной задачи на орграфе, изображенном на рис. 2, с использованием встроенных средств MS Excel, а именно надстройки «Поиск решения».

По условию задачи необходимо найти (кратчайший) путь между состояниями-вершинами 15 и 0. Исключим из графа (для упрощения задачи) вершины с недопустимыми состояниями, а также дуги, входящие в них. В общем случае вершины недопустимых состояний рассматриваются как изолированные, то есть в них не должны входить/выходить дуги. Для результирующего графа, состоящего из $n = 10$ вершин (нумерацию состояний оставим прежней) построим матрицу смежности $S = \{s_{ij}\}$ порядка n.

$$s_{ij} = \begin{cases} 1, \text{ если вершины } i \text{ и } j \text{ соединены дугой (ребром);} \\ 0, \text{ в противном случае} \end{cases}$$

Очевидно, что матрица S – симметричная.

Введём дополнительно матрицу с целочисленными переменными $X = \{x_{ij}\}$, где $1 \le i, j \le n$, и $x_{ij} = 1$, если кратчайший путь содержит переход из вершины i в j; $x_{ij} = 0$ в противном случае.

Целевая функция имеет следующий вид:

$$\sum_{i=1}^{n}\sum_{j=1}^{n} s_{ij}x_{ij} \to \min.$$

В ней вычисляется количество переходов между вершинами в кратчайшем пути.

Первая пара ограничений задаёт условия для начальной вершины пути $V_{\text{нач.}}$ В искомом пути в эту вершину не должно быть входа, но должен быть один выход:

$$\sum_{i=1}^{n} s_{i\,\text{нач}} x_{i\,\text{нач}} = 0,$$

$$\sum_{j=1}^{n} s_{\text{нач}\,j} x_{\text{нач}\,j} = 1.$$

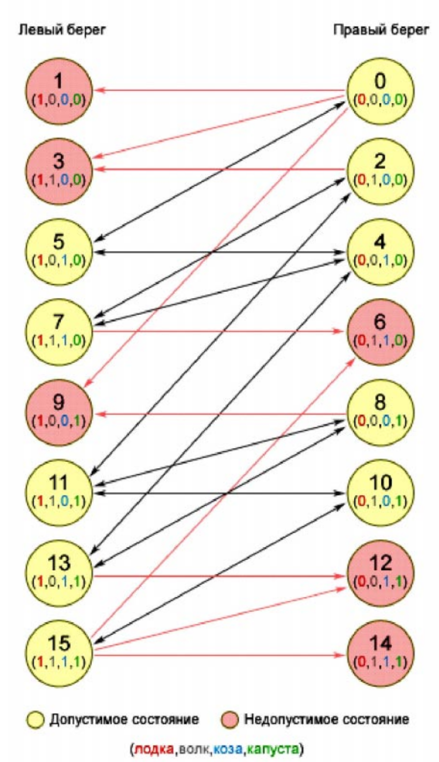

Рис. 2. Граф состояний переправы

Вторая пара ограничений задаёт условия для конечной вершины пути $V_{\text{кон}}$. В неё должен быть один вход, но не должно быть выхода:

$$\sum_{i=1}^{n} s_{i\,\text{кон}} x_{i\,\text{кон}} = 1\,,$$

$$\sum_{j=1}^{n} s_{\text{кон}\,j} x_{\text{кон}\,j} = 0\,.$$

Для всех остальных вершин V_k (кроме $V_{\text{нач}}$ и $V_{\text{кон}}$) устанавливаются ограничения, задающие равенство количества входов и выходов в каждую из них в искомом кратчайшем пути:

$$\sum_{i=1}^{n} s_{ik} x_{ik} - \sum_{j=1}^{n} s_{kj} x_{kj} = 0\,.$$

Для каждой вершины количество входов не должно быть более одного:

$$\sum_{i=1}^{n} s_{ik} x_{ik} \le 1, \quad k = 1..n\,.$$

Для каждой вершины количество выходов не должно превышать одного:

$$\sum_{j=1}^{n} s_{kj} x_{kj} \le 1, \quad k = 1..n\,.$$

Граничные условия устанавливают в качестве области допустимых значений искомых переменных значения 0 или 1:

$$0 \le x_{ij} \le 1, \text{ где } 1 \le i,\ j \le n.$$

Подготовим данные для решения задачи на рабочем листе Excel 2007 в соответствии с рис. 3 для матрицы смежности S. Замечание: изначально матрица X – нулевая, ячейки Q3:Z12 – пустые.

Ячейки C3:L12 содержат матрицу смежности S вершин графа, а ячейки Q3:Z12 – искомую матрицу X.

В ячейки M3:M12 вписываются формулы для вычисления суммы произведений соответствующих строк матрицы смежности и матрицы переменных:
СУММПРОИЗВ(C3:L3;Q3:Z3) ...
СУММПРОИЗВ(C12:L12;Q12:Z12).
В ячейки C13:L13 вписываются формулы для вычисления суммы произведений соответствующих столбцов этих же матриц: СУММПРОИЗВ(C3:C12;Q3:Q12) ...
СУММПРОИЗВ(L3:L12;Z3:Z12).

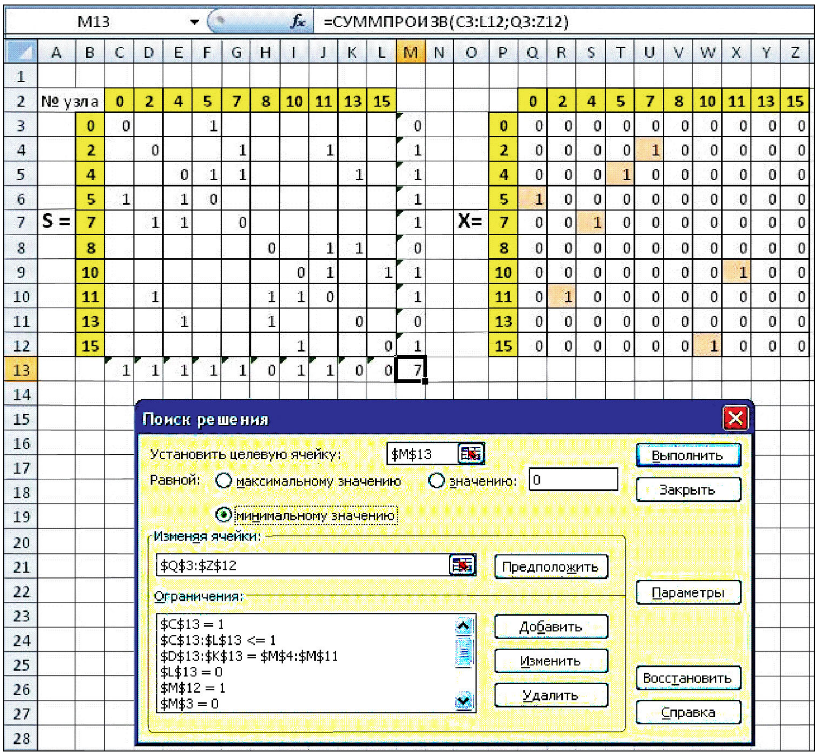

Рис. 3. Моделирование задачи переправы в MS Excel

В ячейку M13 (на рисунке выделена) введём формулу для вычисления целевой функции: СУММПРОИЗВ(C3:L12;Q3:Z12).

Откроем окно «Поиск решения» (пункт меню «Данные/Поиск решения») и установим следующие значения.

Целевая ячейка – $M$13 равна минимальному значению.

Изменяемые ячейки – $Q$3:$Z$12.

Ограничения:

$Q$3:$Z$12 = целое;
$Q$3:$Z$12 >= 0;
$Q$3:$Z$12 <= 1;
$L$13 = 0;
$M$12 = 1;
$C$13 = 1;
$M$3 = 0;
$M$4:$M$11 = $D$13:$K$13;
$M$3:$M$12 <=1;
$M$3:$M$12 >= 0;
$C$13:$L$13 <=1;
$C$13:$L$13 >= 0;

Параметры – линейная модель.

После ввода всех параметров нажатием кнопки «Выполнить» ищем решение.

Возможны 2 результата:

1. Кратчайший путь существует, и решение найдено. Матрица неизвестных X преобразуется в соответствии с найденным решением (рис. 2). Мы видим, что оптимальный путь за 7 ходов проходит через вершины: 15 (начальное положение) $\rightarrow$ 10 (перевезли козу) $\rightarrow$ 11 (лодка вернулась) $\rightarrow$ 2 (перевезли капусту) $\rightarrow$ 7 (лодка с козой вернулись) $\rightarrow$ 4 (перевезли волка) $\rightarrow$ 5 (лодка вернулась) $\rightarrow$ 0 (перевезли козу).

2. Кратчайший путь не существует и не найден. В таком случае система выдаёт сообщение «Поиск не может найти подходящего решения».

Для выделения цветом значений найденного решения для ячеек таблицы неизвестных используется пункт меню «Условное форматирование/Создать правило...».

Готовая модель и решение содержится в файле Pereprava_xls.zip Приложения.

Выбор Excel для решения поставленной задачи несет скорее демонстративный пример использования готовых инструментальных средств. С увеличением числа персонажей очень быстро растет число возможных состояний, и подготовка данных становится затруднительной. Следует привлекать эвристические методы для упрощения задачи (например, сведение к простой частной подзадаче) или использовать программные алгоритмы задания исходных данных и поиска решений, как рассмотрено ниже.

ПРОГРАММНЫЙ АЛГОРИТМ ПОИСКА

В основе алгоритма решения задачи на переправу лежит метод поиска в ширину (BFS, Breadth-first search), также известный под названием «волна на графе».

Алгоритм поиска в ширину заключается в следующем. На первом шаге рассматриваются все дуги, исходящие из V_{M-1}. Смежные с V_{M-1} вершины помечаются как достижимые за один ход – вершины первого ранга. На втором шаге поочередно рассматриваются вершины первого ранга и дуги, исходящие из них. Все вершины, в которые входят эти дуги, помечаются как достижимые за два хода, если они не были помечены ранее. Это будут вершины второго ранга. Действие повторяется до тех пор, пока среди помеченных вершин не окажется V_0. Ранг вершины V_0 и будет являться длиной кратчайшего пути, то есть минимальным количеством ходов, за которое можно осуществить переправу. Если же на очередном шаге алгоритма ни одна новая вершина не будет помечена, то пути из V_{M-1} в V_0 не существует, и задача решения не имеет.

В приведенном выше примере первый ранг получит вершина V_{10} как достижимая из начальной вершины V_{15} за один ход. V_{11} станет вершиной второго ранга, благодаря наличию дуги из V_{10} в V_{11}. Смежные с V_{11} вершины V_2 и V_8 станут вершинами третьего ранга. Следующими будут помечены вершины четвертого ранга V_7 в V_{13}, далее V_4 – пятого ранга, V_5 – шестого и, наконец, V_0 – седьмого.

После того как вершинам вплоть до V_0 сопоставлены ранги, можно восстановить кратчайший путь с конца, обойдя последовательно вершины уменьшающего ранга по дугам в обратном порядке. Так, в нашем примере мы найдем один из двух существующих кратчайших путей:

$V_{15} \to V_{10} \to V_{11} \to V_2 \to V_7 \to V_4 \to V_5 \to V_0$

или

$V_{15} \to V_{10} \to V_{11} \to V_8 \to V_{13} \to V_4 \to V_5 \to V_0$

Теперь останется лишь расшифровать порядковые номера состояний обратно в наборы, характеризующие количество персонажей на обоих берегах, и задача решена.

Программа – решатель задач на переправу (RiverSolver.exe) и примеры заданий (файлы *.txt) приведены в Приложении.

Возможны и другие алгоритмы решения задачи переправы (например, с использованием динамического программирования), хотя они могут приводить к курьезным ситуациям, когда «правильное» решение содержит вариант переправы, в котором одновременно полволка и полкапусты [4].

Пример.

Данная задача на переправу использовалась в Японии в качестве теста для определения интеллекта при приеме на работу.

Требуется переправить на другой берег семью (отца, мать, двух дочерей и двух сыновей), а также полицейского с преступником, соблюдая следующие правила:

На плоту могут находиться не более двух человек одновременно. Отец не может остаться с любой из дочерей без присутствия их матери. Мать не может остаться ни с одним из сыновей без присутствия их отца. Преступник не может оставаться ни с каким членом семейства, если рядом нет полицейского. Только отец, мать и полицейский могут управлять плотом.

Попробуйте найти решение самостоятельно. Флеш-анимация задачи приведена в [5]. Чтобы начать игру, следует нажать на синий кружок справа. Для перемещения плота щелкните мышкой на красный шарик на жерди. Чтобы переместить людей на плот или с него, просто щелкните на выбранном персонаже.

Структура файла задания следующая (family.tsk формируется автоматически при сохранении условия задачи)

```
# список персонажей
отец
сын 2
мать
дочь 2
полицейский
преступник
# кто может плыть в лодке
отец
отец *
мать
мать *
полицейский
полицейский *
# список ограничений
отец x дочь : мать
мать x сын : отец
преступник x отец : полицейский
преступник x  сын : полицейский
преступник x мать : полицейский
преступник x дочь : полицейский
```

Нуждается в пояснении только блок данных «кто может плыть в лодке». Одна звездочка в строке обозначает одного представителя любого типа персонажа (две звездочки – два представителя любых типов персонажей). Возможно задание имени конкретного типа персонажа.

В блоке «список ограничений» перестановка персонажей и порядка строк ограничений не важна.

Информационные блоки исходного файла данных ограничиваются строкой с символом # или пустой строкой.

В результате работы программы RiverSolver получим ответ на экране дисплея (рис. 4).

ЗАКЛЮЧЕНИЕ

Построена модель задачи переправы и дано ее решение с использованием стандартного пакета приложений и алгоритмов. Формализация задачи позволяет провести поиск оптимального решения при задании сложных комбинаций ограничений для большого количества персонажей в условии задачи.

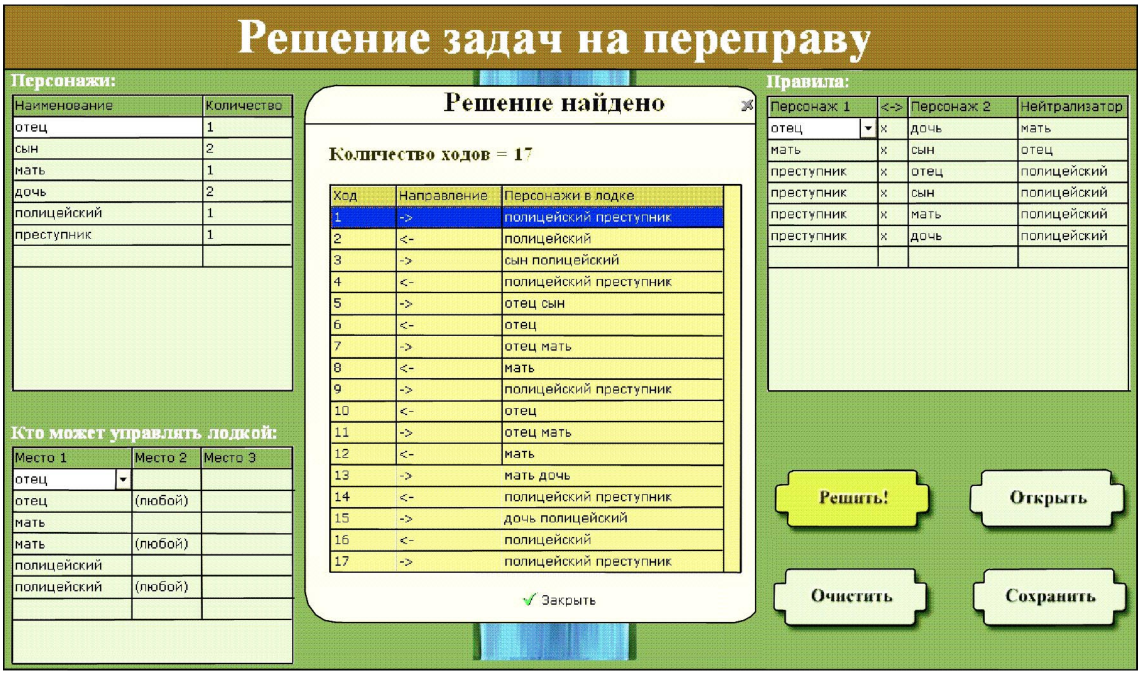

Рис. 4. Компьютерное решение задачи переправы

Литература

1. http://school-collection.edu.ru/ – Единая коллекция Цифровых Образовательных Ресурсов.
2. *Л.Л. Босова, А.Ю. Босова, Ю.Г. Коломенская.* Занимательные задачи по информатике. М.: Бином. Лаборатория знаний, 2005.
3. http://zhurnal.ape.relarn.ru/articles/2006/155.pdf
4. http://www.zib.de/Publications/Reports/SC-95-27.pdf
5. http://dagobah.biz/flash/riverIQGame.swf

Abstract

A range of river crossing puzzles with various types of constraints is considered. An algorithm based on a graph model is proposed. A method of solving the problem using Excel is described.

Keywords: river crossing puzzles, breadth-first search, Excel modelling.

Паньгин Андрей Александрович,* *ведущий инженер-программист фирмы Sun Microsystems,
andrei.pangin@sun.com.

Паньгин Александр Викторович, инженер Центра Информационных Технологий (ЦИТ) г. Сосновый Бор,
pang@sbor.net.