УДК 519.1

Метод динамического программирования и жадный алгоритм на примере поиска оптимальных путей в сети

А. М. Ревякин[1], *И. В. Бардушкина*[1], *А. Н. Исаченко*[2]

Аннотация. Идея жадности лежит в основе как метода динамического программирования (ДП), так и жадного алгоритма. На примере поиска оптимальных путей в сети продемонстрировано различие этих методов. Жадный алгоритм успешно применяется для задач с матроидной структурой, а метод ДП для задач календарного сетевого планирования.

Ключевые слова: сеть, метод динамического программирования, жадный алгоритм, остов минимального или максимального веса.

Dynamic programming method and greedy algorithm on the example of finding optimal paths in a network

A.M. Revyakin, I.V. Bardushkina, A. N. Isachenko

Abstract. The idea of greed underlies both the dynamic programming (DP) method and the greedy algorithm. An example of finding optimal paths in a network demonstrates the difference between these methods. The greedy algorithm is successfully used for problems with a matroid structure, and the DP method for problems of scheduling network planning.

Keywords: network, dynamic programming method, greedy algorithm, minimum or maximum weight spanning tree.

Рассматривается задача поиска оптимальных структур в сети двумя методами: жадным алгоритмом и методом динамического программирования.

Принцип оптимальности динамического программирования [1]: «Каково бы ни было состояние S системы в результате какого-либо числа шагов, на ближайшем шаге нужно выбирать управление так, чтобы оно в совокупности с оптимальным управлением на всех последующих шагах приводило к оптимальному выигрышу на всех оставшихся шагах, включая данный». Другими словами, процесс ДП разворачивается от конца к началу. Сначала делаются различные предположения о том, чем кончился предпоследний шаг, и для

[1]**Ревякин Александр Михайлович,** кандидат физ.-мат. наук, доцент, доцент института физики и прикладной математики, Национальный исследовательский университет «Московский институт электронной техники» (arevyakin@mail.ru)

[1]**Бардушкина Ирина Вячеславовна**, канд. физ.-мат. наук, доц., доцент института физики и прикладной математики, ФГАОУ ВО «Национальный исследовательский университет «Московский институт электронной техники», e-mail: i_v_bars@mail.ru

[2]**Исаченко Александр Николаевич**, кандидат физ.-мат. наук, доцент, доцент кафедры информационных систем управления, Белорусский государственный университет (isachen@bsu.by)

каждого из них выбирается управление на последнем. Затем делаются предположения о том, чем кончился предпоследний шаг, и выбирается управление на втором от конца шаге так, что оно в уже выбранном управлении на последующем шаге обеспечивало наилучший эффект на двух последующих шагах и т.д.

Жадный алгоритм [2-4,6] – метод решения оптимизационных задач, основанный на том, что процесс принятия решения можно разбить на элементарные шаги, на каждом из которых принимается отдельное решение. Решение, принимаемое на каждом шаге, должно быть оптимальным только на текущем шаге и должно приниматься без учёта предыдущих или последующих решений. Другими словами, на каждом шаге делается локально наилучший выбор в надежде, что итоговое решение будет оптимальным.

На рис. 1 дана сеть с односторонними дорогами, не содержащая ориентированных циклов (веса на дугах – длина дороги или стоимость проезда по ней). Легко видеть, что данная сеть разбита на слои, а вершины легко пронумеровать (рис.2).

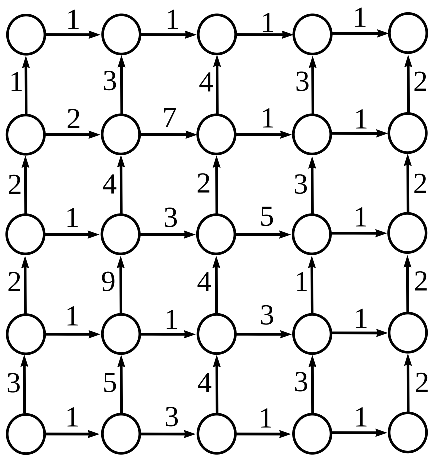

Рис. 1.

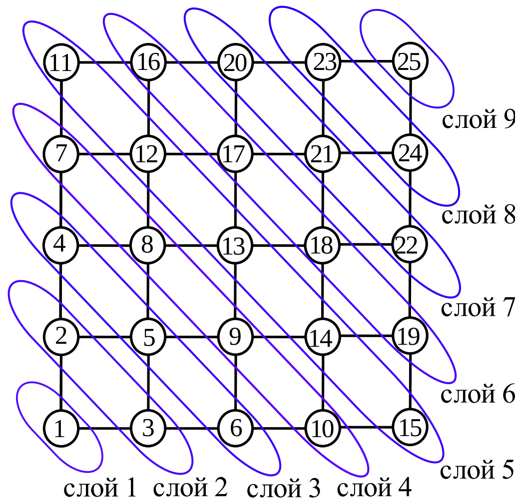

Рис. 2.

Для сетей, разбитых на слои, легко найти кратчайший и самый длинный пути между начальной вершиной 1 и конечной вершиной с номером 25 путем последовательной расстановки пометок, начиная с последней вершины (под номером 25) [6]. Заметим, что в рассматриваемой сети существует 70 различных путей из вершины 1 в вершину 25. Каждому такому пути соответствует последовательность длины 8, состоящая из 0 и 1 (0, если двигаемся вправо и 1, если вверх). Очевидно, каждый путь из вершины 1 в 25 должен содержать ровно 4 дороги вправо и столько же – вверх. На рис. 3 приведен найденный кратчайший путь (его длина 12), а на рис. 4 – самый длинный путь (его длина равна 32). Оба пути найдены методом динамического программирования путем расстановки пометок (см. рис. 3 и 4). Метки размещены внутри вершин.

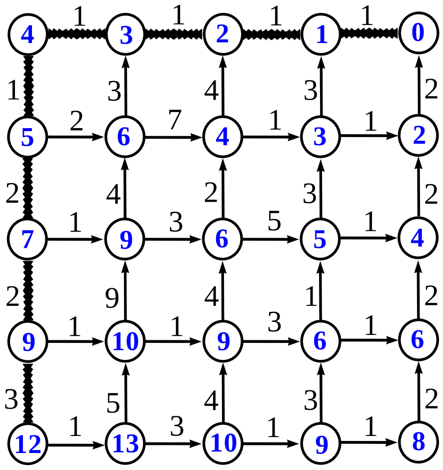

Рис. 3.

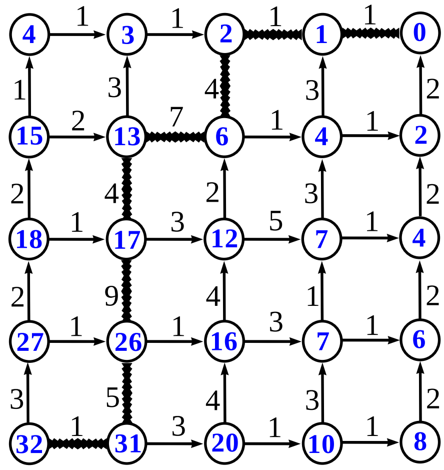

Рис. 4.

Задачу нахождения самого длинного пути между вершинами в сети часто называют «***задачей таксиста***», а задачу нахождения кратчайшего пути – «***задачей пассажира***».

Заметим, что если «жадный таксист» на каждом перекрестке будет выбирать дорогу наибольшей длины, то часто выбранный им маршрут окажется не самым длинным. Например, так это произойдет в нашем случае. На рис. 5 изображен маршрут жадного таксиста (его длина равна 22). Длина самого длинного маршрута из 1 в 25, как было найдено ранее, равна 32.

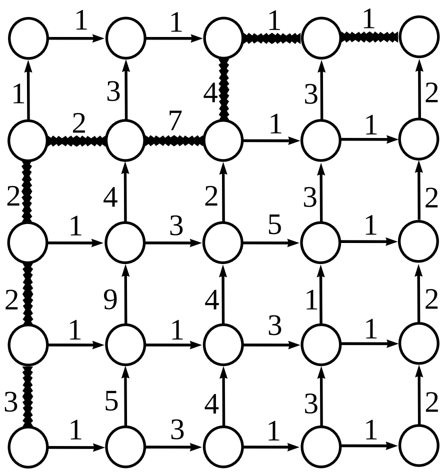

Рис. 5.

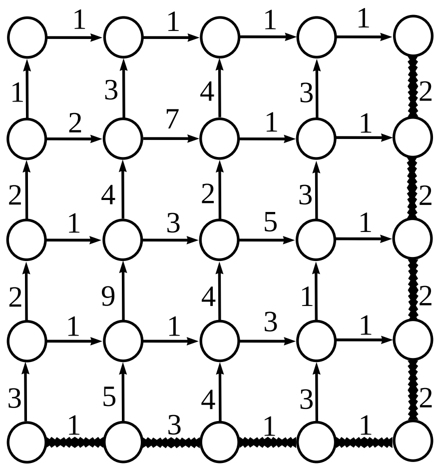

Рис. 6.

Аналогичная ситуация может возникнуть и с «экономным» (жадным в обратном смысле) пассажиром, если на каждом перекрестке выбирать дорогу наименьшей длины (рис. 6). Длина маршрута окажется равной 14, хотя существует путь длины 12.

Для сети, представленной на рис.1, приведём различные остовные деревья: минимального веса (кратчайший с весом 33) (рис.7); максимального веса (самый тяжёлый с весом 71) (рис.8) и остов кратчайших расстояний из вершины 1 во все остальные вершины с весом 41 (рис.9). Остовы на рис.7 и 8 получены жадным алгоритмом.

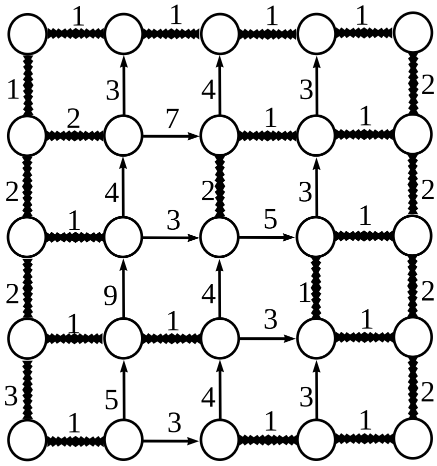

Рис.7.

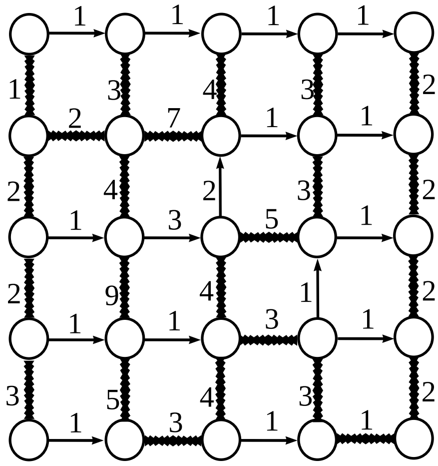

Рис.8.

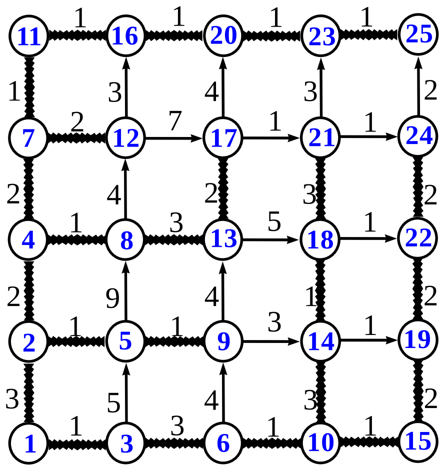

Рис.9.

Остов кратчайших расстояний из вершины 1 во все остальные вершины (рис.9) получен с помощью алгоритма Дейкстры. Кратчайший путь из 1 в 24 равен 12, из 1 в 25 – 12, из 1 в 21 – 12, а из 1 в 17 – 11. Заметим, что остов кратчайших расстояний из вершины 1 не совпадает с остовом минимального веса.

Самый известный из жадных алгоритмов на графах – это *задача поиска остова минимального веса* в произвольной сети (алгоритм Краскала) [5-8].

Жадный шаг алгоритма Краскала: берется ребро с минимальным весом и добавляется в остовное дерево, если при этом не создает там цикла. На рис.10 выбраны ребра с весами 1, 2 и 3. Далее ребро с весом 4 добавить нельзя, так как образуется цикл. Поэтому берется ребро с весом 5, после чего остов минимального веса найден.

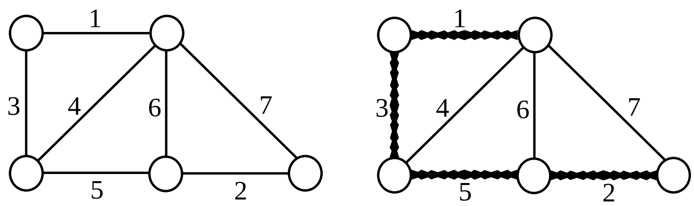

Рис.10.

Обратим внимание на интересное свойство ациклических подграфов. Если в одном ациклическом подграфе меньше ребер, чем в другом, то в большем всегда можно найти ребро, чтобы добавить в меньший, сохранив его ацикличность (рис.11). Иногда даже не одно (рис.12).

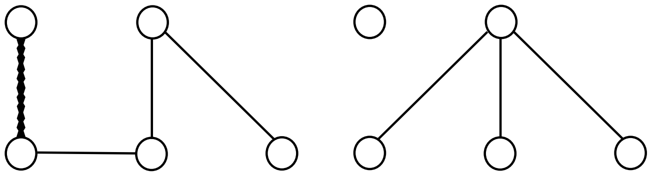

Рис.11.

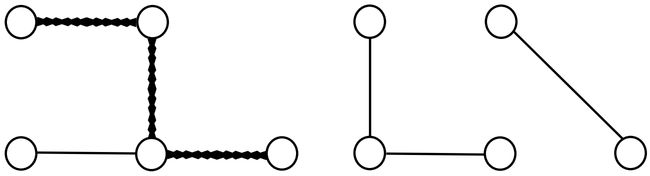

Рис.12.

Иногда может возникнуть искушение использовать жадный подход везде, где только это возможно, но на некоторых задачах это неприемлемо. Следующий пример это подтверждает.

Вор пробрался на склад, в котором хранятся три вещи весом 10 кг, 20 кг и 30 кг стоимостью 60, 100 и 120 у.е. соответственно. Вор максимально может унести 50 кг. Требуется максимизировать прибыль вора. Если поступать здесь жадно и выбирать самую ценную вещь (т.е., 6 у.е. за кг первой штуки, 5 у.е. за кг второй и 4 у.е. за кг третьей), то вор должен взять первую вещь, потом останется место для второй вещи, однако оптимальное решение составляет вторая и третья вещь.

Наличие матроидной структуры [2-4,6-8] служит гарантом того, что найденный жадным алгоритмом ответ является оптимальным. Использование идеи жадности с конца при решении задач календарного сетевого планирования (или графов, разбитых на слои) также приводит к оптимальным решениям.

Литература

1. Беллман Р. Динамическое программирование. М.: ИЛ, 1960.

2. Edmonds J. Matroids and the greedy algorithm. Math. Programming, 1971, v. 1, № 2. – P. 127– 136.

3. Recski A. Matroid theory and its applications in electric network theory and in statics. – Budapest: Akad. Kiado, 1989.

4. Revyakin A.M. Matroids. // J. Math. Sci., 2002, V.108, № 1. – С. 71 – 130.

5. Ревякин А.М., Бардушкина И.В. Математические методы моделирования в экономике. Учебное пособие. – М.: МИЭТ, 2013.

6. Исаченко А.Н., Ревякин А.М. Матроиды в математическом моделировании экономических систем // Экономические и социально-гуманитарные исследования. 2015. № 1 (5). С. 13–18.

7. Корте Б., Фиген Й. Комбинаторная оптимизация. Теория и алгоритмы. – М.: МЦНМО, 2015.

8. Ревякин А.М., Исаченко А.Н. Матроиды и их некоторые приложения // Материалы XIII Международного семинара «Дискретная математика и ее приложения», имени академика О.Б. Лупанова (Москва, МГУ, 17–22 июня 2019 г.). М.: Изд-во механико-математического факультета МГУ, 2019. С. 37 –49.