МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
«ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ
УНИВЕРСИТЕТ»

РЕШЕНИЕ КИНЕМАТИЧЕСКОЙ ЗАДАЧИ ОРИЕНТАЦИИ ТВЕРДОГО ТЕЛА В ПРОСТРАНСТВЕ ДЛЯ ПОСТРОЕНИЯ СИСТЕМЫ ИНЕРЦИАЛЬНОЙ НАВИГАЦИИ

Часть 2

Учебно-методическое пособие

Воронеж
Издательский дом ВГУ
2018

Содержание

что впоследствии дает ему возможность уже не случайно выбирать стратегию поведения, а учитывать опыт предыдущего взаимодействия со средой. Данный метод был предложен как способ оптимизации Марковских процессов принятия решений. Главным преимуществом данного метода является его способность выбирать между немедленной «наградой» (положительным откликом среды на действие агента) и отложенной «наградой». На каждом промежутке времени агент отмечает вектор состояния x_t, а затем выбирает и выполняет действие u_t. При переходе к следующему шагу x_{t+1}, агент получает подкрепление $r(x_t, u_t)$. Цель обучения – найти такую последовательность действий, которая максимизирует сумму будущих подкреплений, таким образом, приводя к финишу по кратчайшему пути.

Целью агента является нахождение такой политики управления, при которой максимизируется ожидаемая сумма наград. Функцией ценности выступает прогнозируемое значение суммы наград при перемещении из любого состояния

$$V(x_t) = E\{\sum_{k=0}^{\infty} \gamma \cdot r_{t+k}\}, \quad (1)$$

где r_i – награда, полученная при переходе системы из состояния x_t в состояние x_{t+1}, а γ – дисконт-фактор ($0 \leq \gamma \leq 1$). Таким образом, $V(x_t)$ представляет собой, так называемую, дисконтированную сумму награды, которую получит агент с момента времени t. Эта сумма зависит от последовательности выбираемых действий, которая определяется политикой управления. В результате работы алгоритма нужно найти такую политику управления, при которой для каждого состояния функция $V(x_t)$ получает максимальное значение. Алгоритм Q-Learning непосредственно не использует функцию ценности, вместо нее используется Q-функция. В Q-функции учитывается состояние и действие агента. Выражение для обновления Q-функции имеет вид

$$Q(x_t, u_t) = r(x_t, u_t) + \gamma \cdot V(x_{t+1}), \quad (2)$$

где u_t–действие, выбранное в момент времени t из множества всех возможных действий U. Так как целью является получить максимум суммарной награды, то $V(x_{t+1})$ заменяется на $\max_{u \in U} Q(x_{t+1}, u)$ и в результате получается следующие выражение

$$Q(x_t, u_t) = r(x_t, u_t) + \gamma \cdot \max_{u \in U} Q(x_{t+1}, u). \quad (3)$$

Оценки Q-значений хранятся в 2-х мерной таблице, входами которой являются состояние и действие. При табличном представлении Q-функций и Марковской среде имеется доказательство сходимости алгоритма Q-Learning.

Параметр γ может изменяться в пределах от 0 до 1, он обеспечивает сходимость суммы. Если параметр γ близок к 0, то агент будет стараться учитывать только немедленные «награды», а если к 1 – то агент будет рас-

сматривать будущие «награды» с большим весом, желая отложить вознаграждение.

Основываясь на (3), опишем алгоритм Q-обучения:

1. Установим параметр γ и положительные отклики среды в матрице R;
2. Инициализируем нулями матрицу Q;
3. Для каждого шага:
    a. Установить случайное начальное состояние;
    b. Выполнять, пока цель не будет достигнута:
        i. Выбрать одно из всех возможных действий для данного состояния;
        ii. Выполнить предполагаемое действие, рассмотреть возможные переходы на следующее состояние;
        iii. Посчитать максимальное значение Q для этого состояния, основываясь на всех возможных действиях;
        iv. Вычислить (3);
        v. Установить следующее состояние как текущее.

Рассмотрим пример применения данного алгоритма. В данной задаче необходимо найти кратчайший путь из любого начального положения (на рисунке 2.1 выбрано положение «2») до конечного положения «5».

2. Применение Q-Learning для поиска пути из лабиринта. Пример.

Рассмотрим типовой пример о поиске пути [9], демонстрирующий концепцию метода Q-обучения. В примере описывается агент, который использует обучение «без учителя», чтобы получать знания о заранее неизвестной окружающей его среде.

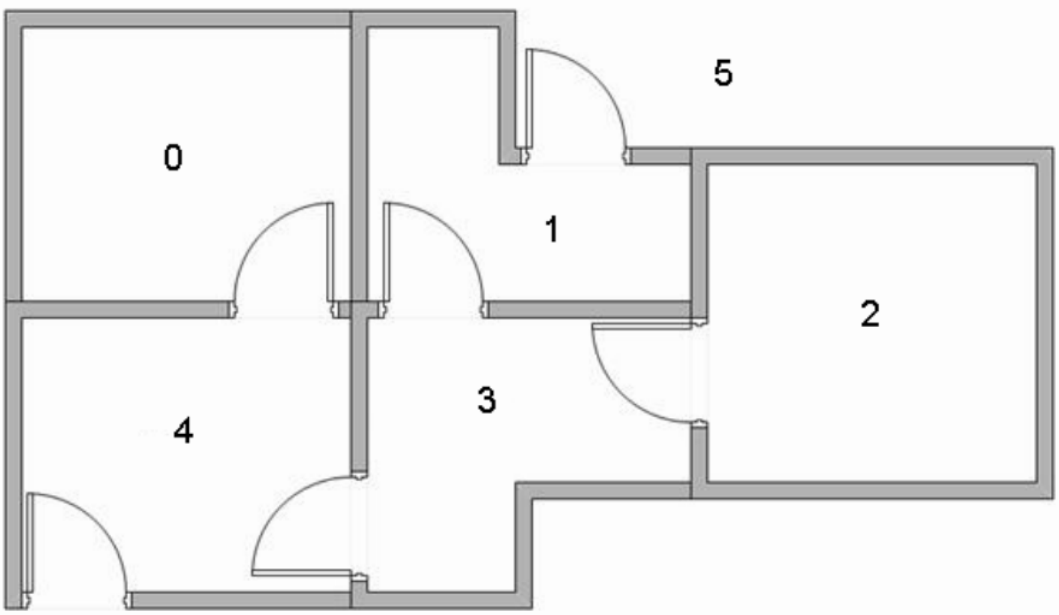

Рисунок 1. *Здание из пяти комнат*

Предположим, что в здании есть 5 комнат, соединенных дверьми, как показано на рисунке 1. Пронумеруем каждую комнату цифрами от 0 до 4. Территория снаружи здания будем считать как одну большую комнату (5). При этом двери комнат 1 и 4 ведут в здание из комнаты 5 (снаружи).

Можно представить расположение комнат здания в виде графа, рисунок 2. Каждую комнату изобразим узлом, а каждую дверь - как ребро.

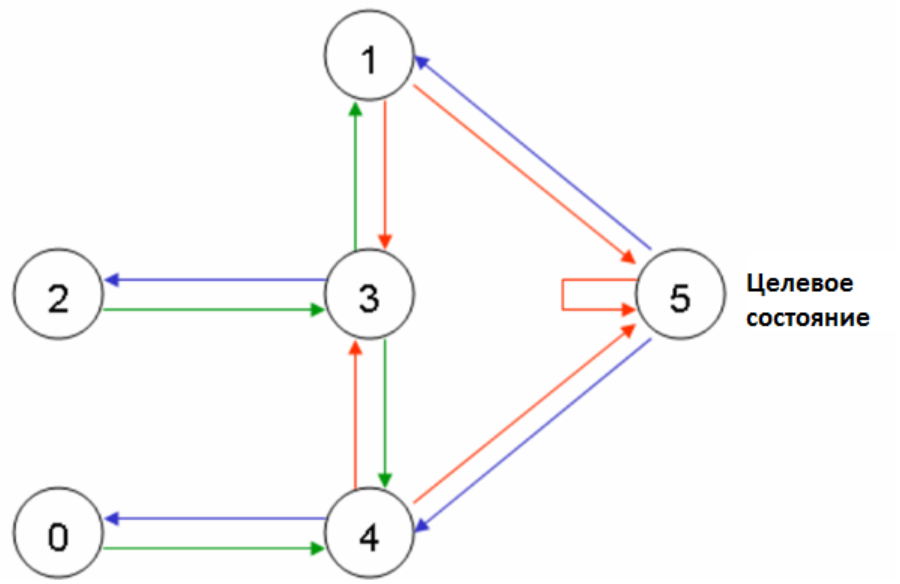

Рисунок 2. *Расположение комнат здания в виде графа*

Агент помещается в случайную комнату. Задача агента выйти за пределы здания (это будет наша целевая комната). Другими словами, целевая комната имеет номер 5. Чтобы установить эту комнату в качестве цели для агента, мы присвоим каждой двери (т.е. связи между узлами) некоторое значение вознаграждения. Двери, которые непосредственно ведут к цели, получают величину награды равную 100 единицам. Другие двери, не связанные напрямую с целевой комнатой, получают нулевую текущую награду. Поскольку двери двусторонние (дверь 0-4 ведет из комнаты 0 в 4, и наоборот из комнаты 4 в комнату 0), для каждой комнаты назначены две стрелки. Каждая стрелка содержит значение текущего вознаграждения, как показано на рисунке 3. Заметим, что переход из комнаты 5 в комнату 5 имеет величину награды 100, как и другие прямые связи с комнатой цели.

При использовании алгоритма Q-Learning цель состоит в том, чтобы агент достиг целевого состояния с самой высокой наградой и остался в этом состоянии. Этот тип цели называется «захватывающая цель».

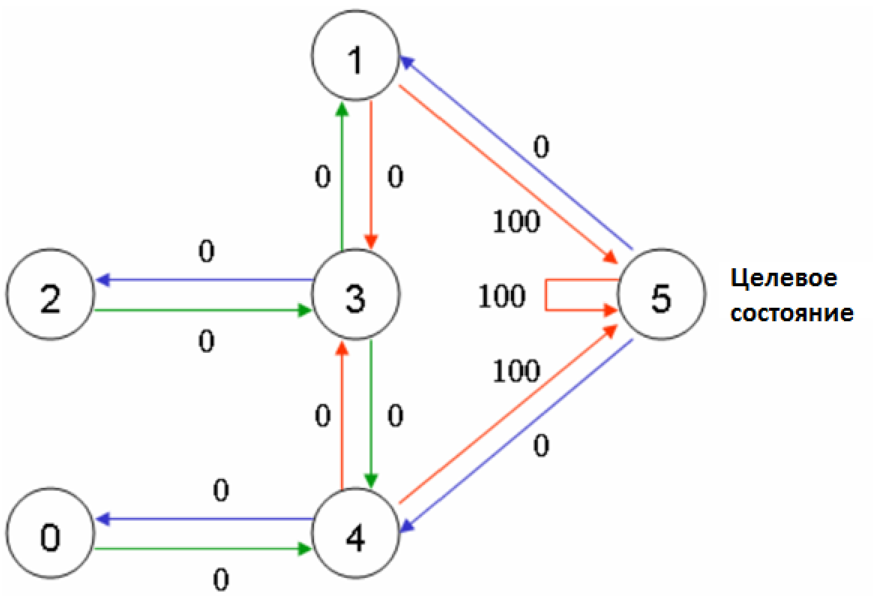

Рисунок 3. *Значение текущего вознаграждения*

Агента можно представить как виртуального робота, который учиться на опыте, возникающем в результате некоторых действий робота. Агент может переходить из одной комнаты в другую, но он не имеет информации об окружающей среде и не знает, какая последовательность дверей ведет наружу.

Предположим, что мы хотим смоделировать эвакуацию агента из заданной комнаты в здании наружу. Пусть агент находится в комнате 2. Его цель научиться выходить из здания или в комнату номер 5, рисунок 4.

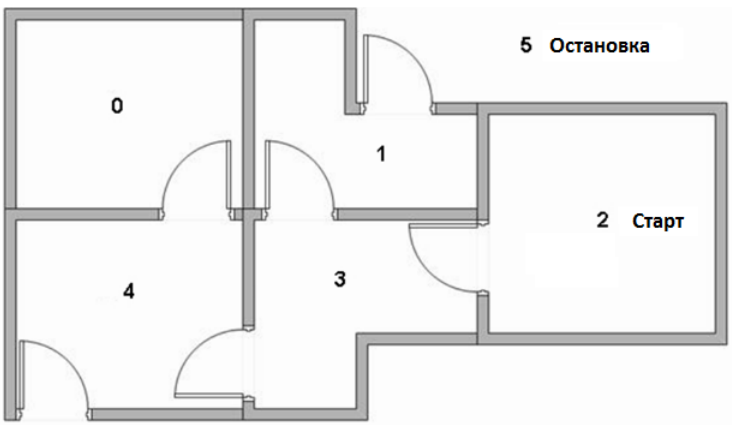

Рисунок 4. Эвакуацию агента из заданной комнаты в здании наружу

Терминология метода Q-Learning включает в себя термины состояние и действие. Будем называть каждую из комнат, в том числе и наружную – состоянием.   А перемещение агента из одной комнаты в другую будет считаться действием. На графе состояние изображено как узел, а действие представлено стрелками, рисунок 5.

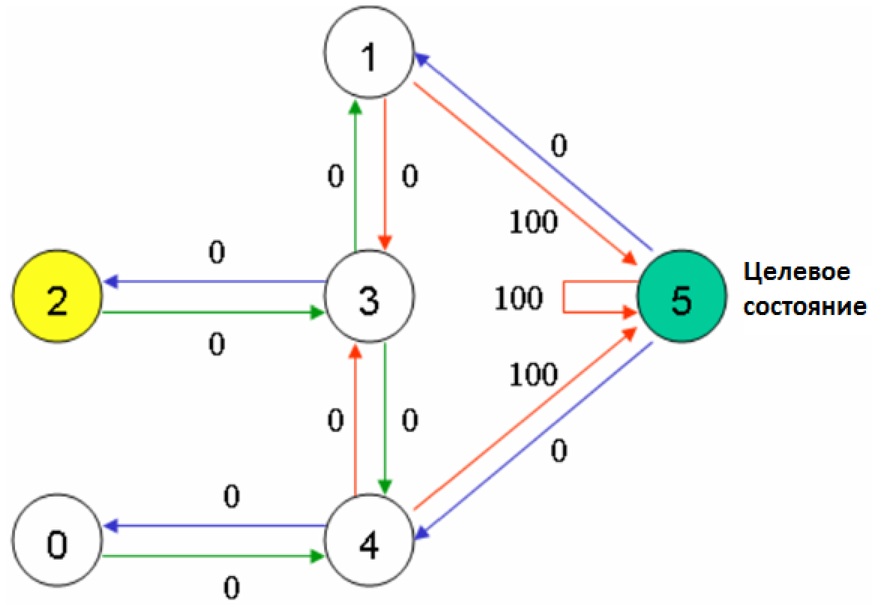

Рисунок 5. *Граф состояние - узел*

Предположим, что агент находится в состоянии 2. Из состояния 2 он может перейти в состояние 3. При этом из состояния 2, агент не может напрямую перейти в состояние 1, так как отсутствует прямая дверь, соединяющая комнаты 1 и 2 (поэтому, связующих стрелок на графе нет). Из состояния 3 он может перейти в состояние 1 или 4, а также вернуться обратно в 2 (посмотрите на все стрелки состояния 3). Если агент находится в состоянии 4, то существует три возможных действия – перейти в состояние 0, 5 или 3. Если агент находится в состоянии 1, он может перейти в состояния 5 или 3. Из состояния 0 он может перейти только в состояние 4.

Можно поместить диаграмму состояний и текущие значения вознаграждения в следующую таблицу вознаграждений, которая будет обозначена как матрица R, таблица 1.

Таблица 1

Действие

$$\textbf{Состояние} \quad \begin{matrix} 0 & 1 & 2 & 3 & 4 & 5 \end{matrix}$$

$$R = \begin{matrix} 0 \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{matrix} \begin{bmatrix} -1 & -1 & -1 & -1 & 0 & -1 \\ -1 & -1 & -1 & 0 & -1 & 100 \\ -1 & -1 & -1 & 0 & -1 & -1 \\ -1 & 0 & 0 & -1 & 0 & -1 \\ 0 & -1 & -1 & 0 & -1 & 100 \\ -1 & 0 & -1 & -1 & 0 & 100 \end{bmatrix}$$

Величиной равной -1 в таблице представляются не допустимые действия (отсутствие связи между узлами). Например, из состояния 0 нельзя перейти в состояние 1.