МИНОБРНАУКИ РОССИИ

САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ

ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)

Кафедра Информационных систем

ПРАКТИЧЕСКАЯ РАБОТА

по дисциплине «Теория принятия решений»

Тема: Применение методов линейного и динамического программирования для решения практических задач (по вариантам). Вариант: 39

Студент гр. 0372	_______________	Масленников К.М.
Преподаватель	_______________	Степуленок Д.О.

Санкт-Петербург

2023

Содержание

Задача №3

3.1. Условие задачи

В целях проведения научного исследования планируется размещение автономного беспроводного сенсора с возможностью подзарядки (от солнечной батареи).

Функционирование сенсора можно описать с помощью следующей модели с дискретным временем. В каждый временной слот сенсор может либо осуществить передачу данных, либо отложить ее. Решение принимается в зависимости от состояния среды передачи данных (уровень помех и пр.) и имеющейся у сенсора энергии. Передача сообщения требует одну единицу энергии, при этом, при благоприятном состоянии среды передачи данных сообщение доходит до получателя с вероятностью 0.99, а при неблагоприятном состоянии – с вероятностью 0.55. Неопределенность, связанная с подзарядкой посредством солнечных батарей, раскрывается следующим образом. В каждый временной слот вероятность получения единицы энергии равняется 0.2. Аккумулятор устройства рассчитан на 20 единиц энергии.

Среда передачи данных может находиться в двух состояниях: благоприятном и неблагоприятном. Вероятность перехода из неблагоприятного состояния в благоприятное (за один временной слот) – 0.25, вероятность того, что канал останется в благоприятном состоянии – 0.7.

Необходимо:

1. Определить математическое ожидание количества переданных пакетов за 5000 временных слотов при реализации оптимальной стратегии передачи.

2. Сравнить оптимальные стратегии на 1000-м и 5000-м временных слотах. Есть ли отличия и если есть, то в чем они заключаются?

3. Построить реализацию случайного процесса, порождаемого оптимальной стратегией.

4. Определить влияние вероятности получения единицы энергии на математическое ожидание количества переданных пакетов.

3.2. Формализация задачи

Постановка задачи требует наилучшей стратегии выбора действия в каждый временной слот с целью максимизации получаемого вознаграждения. Из условия следует, что процесс передачи является многошаговым. Следовательно, рассматриваемая задача относится к классу задач динамического программирования. В терминах задачи применяются дискретное время, конечное число состояний, переходные вероятности между состояниями и структура вознаграждений, поэтому задача может быть решена при помощи Марковского процесса принятия решений.

Перейдем к формализации задачи.

Параметры:

$E = \{0, 1, \dots ,20\}$ – число единиц энергии в аккумуляторе.

$S = \{1, 2\}$ – состояние среды передачи данных. 1 – неблагоприятное, 2 – благоприятное.

$q = 0.2$ – вероятность получения единицы энергии в каждый временной слот.

p_S , $p_1 = 0.55$, $p_2 = 0.99$ – вероятность успешной передачи данных при состоянии среды передачи данных S.

$\lambda_0 = 0.25$ – вероятность перехода среды передачи данных из неблагоприятного состояния в благоприятное состояние.

$\lambda_1 = 0.7$ – вероятность того, что канал останется в прежнем состоянии. (E, S) – состояние системы.

$A = \{T,D\}$ – множество действий. T – передать данные, D – отложить передачу данных.

N – количество рассматриваемых временных слотов.

Обратимся к рассмотрению задачи с точки зрения терминов динамического программирования. При анализе условия задачи необходимо выделить три составляющие, обязательные при решении задачи методом динамического программирования: выигрыш, управление и состояние.

В задаче N точек принятия решений.

Выигрыш соответствует математическому ожиданию количества переданных пакетов данных.

Управление – решение о выборе действия A. Состояние – пара параметров (E, S).

Выигрыш в временной слот i в зависимости от состояния $(E_i\ ,\ S_i\ )$ и управления A_i:

$$R((E_i\ ,S_i), A_i)\ =\ \begin{cases} p_S\ , \text{если } A_i\ =\ T \\ 0, \text{если } A_i\ =\ D \end{cases}$$

Опишем изменения состояния в системе под влиянием управления. Если выбрано действие T, то происходит уменьшение E на единицу. Однако аккумулятор может получить энергию с вероятностью q. Когда канал в состоянии S, состояние канала в следующий слот времени может быть 2 с вероятностью λ_s и 1 с вероятностью $(1-\lambda_s)$. Если выбрано действие D, логика рассуждения сохраняется, но необходимо учесть, что E не меняется. Обозначим $V_n(E,\ S)$ оптимальный ожидаемый выигрыш, полученный на временных слотах от n до N включительно при условии, что система находится в начале временного слота n в состоянии $(E,\ S)$. Тогда основное функциональное уравнение динамического программирования (уравнение Беллмана) для данной задачи будет иметь вид:

$$V_n(E,S)\ =\ max\{V_T(E,S), V_D(E,S)\},$$

где $V_T(E,S)$ – выигрыш, полученный при выборе действия T в состоянии (E,S). $V_D(E, S)$ – выигрыш, полученный при выборе действия D в состоянии (E, S).

На основании рассуждений об изменениях состояния под влиянием управления рассмотрим подробнее каждый переход.

Если выбрано действие T, то ожидаемый выигрыш составляет p_S. Тогда $V_T(E,S)$ имеет вид:

$$V_T(E,S) = pS + q\lambda_s V_{n+1}(E,2) + q(1 - \lambda_s)V_{n+1}(E,1) + (1 - q)\lambda_s V_{n+1}(E - 1,2) + (1 - q)(1 - \lambda_s)V_{n+1}(E - 1,1).$$

Если выбрано действие D, ожидаемый выигрыш равен 0. $V_D(E, S)$ имеет вид:

$$V_D(E,S) = q\lambda_s V_{n+1}(E + 1,2) + q(1 - \lambda_s)V_{n+1}(E + 1,1) + (1 - q)\lambda_s V_{n+1}(E,2) + (1 - q)(1 - \lambda_s)V_{n+1}(E,1).$$

Математическая запись задачи завершена.

3.3 Решение задачи

Программа разработана в Octave. Код программы сопровождается комментариями, поясняющими ход выполнения работы. Перейдем к рассмотрению исходного кода, представленного в приложении В. Вначале происходит очистка консоли и окна виджета и удаление переменных:

```
clc; clear all; clf;
```

Вводится вероятность получения аккумулятором единицы энергии в каждый временной слот:

```
q = 0.2;
```

Указывается количество слотов времени.

```
timeSlotsAmount = 5000;
```

Определяется объём аккумулятора в единицах энергии.

```
maxVolume = 20;
```

Задается вероятность смены состояния канала. Первое значение – вероятность перехода из $S = 1$ в $S = 2$, второе – переход из $S = 2$ в $S = 2$:

```
lambda = [0.2 0.7];
```

Вводятся вероятности успешной доставки пакета данных при $S = 1$ и $S = 2$.

```
p = [0.55 0.99];
```

Моделируется выигрыш на последнем шаге:

```
nextV = zeros(maxVolume, 2);
nextV(2:end, 1) = p(1);
nextV(2:end, 2) = p(2);
```

Далее в цикле происходит итеративное вычисление выигрыша в каждый временной слот. При циклическом вычислении используется функция maxMatrix, исходный код которой представлен в приложении В. Необходимость написания данной функции обусловлена тем, что требуется фиксировать оптимальное управление. Также применяются функции Vd и Vt для вычисления выигрыша, полученного при выборе действия T и действия D. После окончания вычисления математических ожиданий на экран выводятся результаты.

```
Старт работы программы
Вычисление математических ожиданий...
Математическое ожидание количества переданных пакетов за
5000 временных слотов при реализации оптимальной стратегии:
(ниже представлен вид таблицы, которой будут выведены значения
1 - передать, 2 - отложить)
        |S = 1|S = 2
-------- ----- -----
E = 0   |.....|.....
-------- ----- -----
E = 1   |.....|.....
-------- ----- -----
E = ... |.....|.....
-------- ----- -----
E = 20  |.....|.....

    989.40    989.40
    990.39    990.39
    991.38    991.38
    992.37    992.37
    993.36    993.36
    994.35    994.35
    995.34    995.34
    996.33    996.33
    997.32    997.32
    998.30    998.31
    999.29    999.29
   1000.27   1000.28
   1001.25   1001.27
   1002.23   1002.25
   1003.19   1003.23
   1004.13   1004.20
   1005.03   1005.16
   1005.87   1006.09
   1006.60   1006.99
   1007.15   1007.81
```

Проанализируем полученные данные. Из результатов следует, что математическое ожидание при одном и том же состоянии аккумулятора, но разных состояниях среды, приблизительно равно. Однако, чем больше количество единиц энергии в аккумуляторе, тем выше математическое ожидание количества переданных пакетов, что свидетельствует о том, что автономный сенсор напрямую зависит от состояния своего аккумулятора.

Далее происходит сравнение оптимальных стратегий на 1000-м и 5000-м временном слоте:

tmpr = nVi_1 == nVi_2;

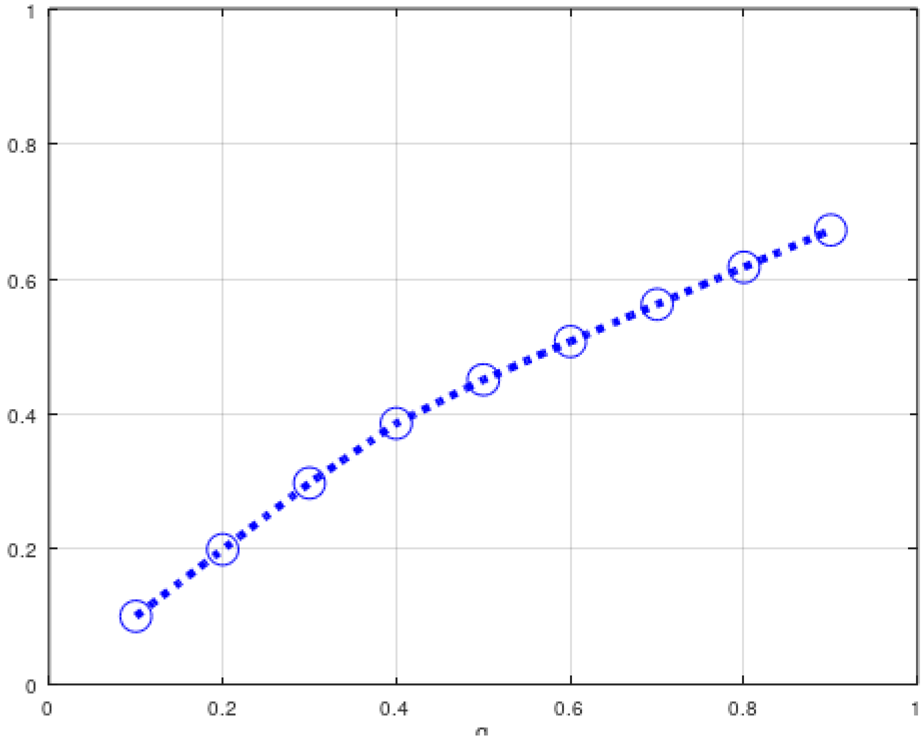
После этого производится вывод результатов анализа на экран.

```
Оптимальные стратегии на 1000-м и 5000-м временных
слотах идентичны (1 - передать, 2 - отложить):
   2   2
   2   1
   2   1
   2   1
   2   1
   2   1
   2   1
   2   1
   2   1
   2   1
   2   1
   2   1
   2   1
   2   1
   2   1
   2   1
   2   1
   2   1
   2   1
   1   1
```

Рассмотрим подробнее полученные результаты. Хорошо заметны две крайние точки – состояние, при котором аккумулятор полностью разряжен, тогда следует неизбежно отложить передачу, и состояние, когда аккумулятор полностью заряжен, тогда стоит произвести передачу. Во всех остальных случаях передача производится только при благоприятном состоянии среды, так как в такой ситуации вероятность того, что состояние останется благоприятным, выше.

Определяются информация для вывода графика – указывается вектор вероятностей q = (0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9), для каждого значения вычисляется суммарный выигрыш и пропускная способность, вычисленная как отношение среднего значения математического ожидания к числу количеству временных слотов.

Следующим шагом производятся интерполяция для вывода линейного графика, а не точечного, и непосредственно формирование графика. График изображен на рисунке.



По графику можно произвести анализ того, как влияет вероятность получения единицы энергии на математическое ожидание количества переданных пакетов. По оси абсцисс откладывается вероятность, а по оси ординат – пропускная способность. Из данных графика следует вывод, что с повышением вероятности получения единицы энергии повышается и математическое ожидание количества переданных пакетов, что обусловлено зависимостью рассматриваемого автономного беспроводного сенсора от состояния его аккумулятора.

Таким образом, на все поставленные в задаче вопросы даны ответы, снабженные вычислениями, задача решена успешно.

ЗАКЛЮЧЕНИЕ

В результате выполнения работы была решена динамическая оптимизационная задача с помощью среды GNU Octave. Были выполнены все необходимые пункты, прописанные в условии.