PRÁCTICA INTELIGENCIA ARTIFICIAL

Fecha: **14/05/2023 -** ENTREGA: **15/05/2023**

GRUPO: **80**

Alumnos: **Alberto García de la Torre / Ana María Ortega Mateo**

ÍNDICE

1. RESUMEN EJECUTIVO

En esta práctica hemos tenido que desarrollar un termostato para optimizar el consumo de energía. El usuario quiere alcanzar una temperatura de 22 grados(la deseada) en una habitación y queremos implementar un termostato que actúe según lo que sea más óptimo comprometiendo el confort del usuario. El termostato irá midiendo la temperatura de la habitación de medio grado a medio grado, entre el intervalo de 16 y 25 grados.

El termostato irá tomando decisiones cada media hora(a modo de referencia) e irá actuando según las distintas probabilidades, dependiendo si se encuentra apagado o encendido. La temperatura de la habitación puede aumentar, permanecer igual o disminuir, en función del estado(temperatura que se encuentre), teniendo en cuenta excepciones de los límites del intervalo(16 grados y 25 grados), donde actúan distinto ya que en 16 no se podrá disminuir y en 25 no se podrá aumentar, debido a su limitación por ser parte de los extremos. Estas son la dos tablas de probabilidades creadas por nosotros en función de las temperaturas y de si hay que apagar o encender el termostato:

OFF	16	16,5	17	17,5	18	18,5	19	19,5	20	20,5	21	21,5	22	22,5	23	23,5	24	24,5	25
16	0,9	0,1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
16,5	0,7	0,2	0,1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
17	0	0,7	0,2	0,1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
17,5	0	0	0,7	0,2	0,1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
18	0	0	0	0,7	0,2	0,1	0	0	0	0	0	0	0	0	0	0	0	0	0
18,5	0	0	0	0	0,7	0,2	0,1	0	0	0	0	0	0	0	0	0	0	0	0
19	0	0	0	0	0	0,7	0,2	0,1	0	0	0	0	0	0	0	0	0	0	0
19,5	0	0	0	0	0	0	0,7	0,2	0,1	0	0	0	0	0	0	0	0	0	0
20	0	0	0	0	0	0	0	0,7	0,2	0,1	0	0	0	0	0	0	0	0	0
20,5	0	0	0	0	0	0	0	0	0,7	0,2	0,1	0	0	0	0	0	0	0	0
21	0	0	0	0	0	0	0	0	0	0,7	0,2	0,1	0	0	0	0	0	0	0
21,5	0	0	0	0	0	0	0	0	0	0	0,7	0,2	0,1	0	0	0	0	0	0
22	0	0	0	0	0	0	0	0	0	0	0	0,7	0,2	0,1	0	0	0	0	0
22,5	0	0	0	0	0	0	0	0	0	0	0	0	0,7	0,2	0,1	0	0	0	0
23	0	0	0	0	0	0	0	0	0	0	0	0	0	0,7	0,2	0,1	0	0	0
23,5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0,7	0,2	0,1	0	0
24	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0,7	0,2	0,1	0
24,5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0,7	0,2	0,1
25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0,7	0,3

ON	16	16,5	17	17,5	18	18,5	19	19,5	20	20,5	21	21,5	22	22,5	23	23,5	24	24,5	25
16	0,3	0,5	0,2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
16,5	0,1	0,2	0,5	0,2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
17	0	0,1	0,2	0,5	0,2	0	0	0	0	0	0	0	0	0	0	0	0	0	0
17,5	0	0	0,1	0,2	0,5	0,2	0	0	0	0	0	0	0	0	0	0	0	0	0
18	0	0	0	0,1	0,2	0,5	0,2	0	0	0	0	0	0	0	0	0	0	0	0
18,5	0	0	0	0	0,1	0,2	0,5	0,2	0	0	0	0	0	0	0	0	0	0	0
19	0	0	0	0	0	0,1	0,2	0,5	0,2	0	0	0	0	0	0	0	0	0	0
19,5	0	0	0	0	0	0	0,1	0,2	0,5	0,2	0	0	0	0	0	0	0	0	0
20	0	0	0	0	0	0	0	0,1	0,2	0,5	0,2	0	0	0	0	0	0	0	0
20,5	0	0	0	0	0	0	0	0	0,1	0,2	0,5	0,2	0	0	0	0	0	0	0
21	0	0	0	0	0	0	0	0	0	0,1	0,2	0,5	0,2	0	0	0	0	0	0
21,5	0	0	0	0	0	0	0	0	0	0	0,1	0,2	0,5	0,2	0	0	0	0	0
22	0	0	0	0	0	0	0	0	0	0	0	0,1	0,2	0,5	0,2	0	0	0	0
22,5	0	0	0	0	0	0	0	0	0	0	0	0	0,1	0,2	0,5	0,2	0	0	0
23	0	0	0	0	0	0	0	0	0	0	0	0	0	0,1	0,2	0,5	0,2	0	0
23,5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0,1	0,2	0,5	0,2	0
24	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0,1	0,2	0,5	0,2
24,5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0,1	0,2	0,7
25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0,1	0,9

El termostato a implementar seguirá la política óptima basada en el Modelo de Decisiones de Markov para lograr saber cual es la mejor acción(on/off) que deberá tomar el usuario. Se pretende que el algoritmo tome la decisión más eficiente para el termostato, reduciendo el consumo de energía, intentando minimizar los costes y buscar la opción más rentable para cada situación.

2. OBJETIVOS

El objetivo de la práctica será alcanzar la temperatura ideal, teniendo en cuenta la política óptima que hay que implementar (algoritmo de Bellman). Vamos a intentar garantizar que el programa sea lo más genérico posible, pero abarcando las necesidades específicas del proyecto. Por ello, hemos tenido en cuenta que cumpliera con los siguientes requisitos:

1. Se deben atribuir unos costes asequibles a la situación expuesta.
2. Identificar las tareas necesarias para llegar al objetivo y poder implementarlo.
3. Hay que simular una ejecución hasta el infinito hasta que converja el valor esperado de cada estado.
4. El código debe permitir analizar distintos escenarios fácilmente, con el objetivo de comprobar si sigue un funcionamiento lógico.
5. Debe simular de la mejor manera posible el funcionamiento real de un termostato, dentro de las condiciones de la práctica. Intentando analizar los resultados como si de un caso real se tratase y teniendo en cuenta aspectos como el consumo de la energía basados en casos existentes de la actualidad.
6. Los cálculos a realizar deben tener relación con el contenido de la asignatura, implementando y mostrando todo lo aprendido sobre los modelos de Markov y todo lo que este tema abarca.

3. DESCRIPCIÓN FORMAL DEL MODELO MDP

Los procesos de Markov son un tipo de proceso estocástico, lo que significa que la probabilidad de transición de un estado a otro solo depende del estado actual, y no de los estados previos o de los procesos de tiempo, donde el tiempo a tener en cuenta se considera continuo(abarca un rango continuo de valores en lugar de específicos). Los resultados del futuro son independientes del pasado, tal y como indica la propiedad de Markov:

$$P\left[X_{t+1} = j \middle/ X_t = i \right]$$

El estado actual Xt contiene toda la información relevante de los estados pasados e influye en el estado siguiente. Un modelo de Markov está compuesto de los siguientes elementos:

1. **Conjunto de estados(S)**: Representa el conjunto de situaciones en los que puede encontrarse la situación dada. Cada estado sirve como descripción de una situación en un momento determinado.
2. **Conjunto de acciones(A):** El conjunto de acciones que el agente puede tomar en cada estado. Cada acción determina una transición entre estados.
3. **Función de transición(P):** Función que define la probabilidad de transición. Suele utilizar el estado actual, la acción tomada y el estado resultante.
4. **Costes(C):** El coste específico para tomar una acción.

En nuestra práctica, estos elementos los hemos definido como:

1. **Conjunto de estados(S)**: Los estados serán las distintas temperaturas posibles en la habitación. Las temperaturas posibles por lo tanto serían entre 16 y 25 grados, con incrementos de medio grado.
   S={16, 16.5, 17, 17.5, 18, 18.5, 19, 19.5, 20, 20.5,21, 21.5, 22, 22.5, 23, 23.5, 24, 24.5, 25}
   Un total de 19 estados.
2. **Conjunto de acciones(A):** Las acciones posibles son "on y off", el usuario puede o bien apagar el termostato o bien encenderlo.
3. **Función de transición(P):** En nuestro caso se basan en las probabilidades establecidas en el enunciado. La función de transición serían las distintas tablas que hemos creado teniendo en cuenta la acción tomada y las distintas transiciones entre temperaturas(tablas que se encuentran en resumen ejecutivo).
4. **Costes(C):** Los costes van en función a la acción tomada por lo que hemos decidido poner un coste de en on y un coste de en off(explicación más detallada en el apartado 4 del modelo de costes).

La política óptima de Markov es una estrategia de toma de decisiones que maximiza el valor esperado de la recompensa acumulada. En nuestro caso sería minimizar los costes de las acciones posibles de tal forma poder sacar las acciones más óptimas de cada estado. La política óptima define qué acción tomar en cada estado para lograr el mejor resultado posible. En nuestro caso también se le es asignado a las acciones una distribución de probabilidad para cada estado, permitiendo combinar la probabilidad con la política óptima.

Para resolver un Modelo de Markov utilizando la política óptima hay que simular una ejecución hasta el infinito que converja el valor esperado de cada estado con la acción más óptima. Las ecuaciones de Bellman nos servirán para encontrar dicho mejor valor posible. Según el problema, se busca maximizar o minimizar un resultado. Si tenemos recompensas al llegar a ciertos estados, buscaremos el máximo al realizar acciones, mientras que si lo que tenemos es costes por hacer acciones tendremos que minimizar.

$$V_{i+1}(E_{orig}) = \min_{accion}[\,coste(accion) + \sum_{E_{dest}} P_{accion}(E_{dest}|E_{orig}) \cdot V_i(E_{dest})]$$

Se va calculando el valor de V hasta que converge para cada estado y se escoge el valor mínimo entre los resultados dependiendo de las acciones tomadas. Dicho valor esperado es utilizado para encontrar la política óptima, es decir, la mejor acción probabilísticamente calculada para el estado E. Por ello en nuestro código hemos programado la ecuación de Bellman hasta que los resultados convergen y devuelva una lista con la acción más óptima para cada temperatura.

4. ANÁLISIS DETALLADO DEL MODELO DE COSTES

El termostato es un aparato que calcula si es mejor activar o desactivar el sistema de calefacción, teniendo en cuenta la temperatura a la que se desea tener la casa. Por ello, el factor más importante que hemos tenido en cuenta para definir los costes es el consumo energético en función del precio de la luz.

Al observar el precio medio de la calefacción, nos encontramos con que, en la tarifa de la luz, cada KWh cuesta alrededor de 20 céntimos. Esto significa que si se mantiene activo un radiador, el cual consume 2 KWh, cada hora cuesta 40 céntimos.

El valor del termostato apagado será 1, ya que sigue teniendo un consumo energético aunque la calefacción no esté encendida. Esto se debe a que, a pesar de no activar la calefacción, sigue tomando medidas periódicamente de los niveles de humedad, CO_2 y temperatura externa.

Como mantener el termostato activado, para un radiador, y durante media hora cuesta alrededor de 20 céntimos, el valor del termostato encendido será 20, lo que significa que para nosotros, el consumo energético cuando se enciende la calefacción es 20 veces mayor que cuando no está encendida.

De todas forma, también utilizaremos otros valores más exagerados (como aumentar el coste del termostato encendido a 300) para ver cómo varía la política óptima según las distintas situaciones posibles, ya que el termostato puede estar conectado a, por ejemplo, el sistema de calefacción entero de la casa y, por tanto, tener un consumo mucho mayor que el que vamos a utilizar inicialmente.

En definitiva, teniendo en cuenta todos los datos analizados hemos decidido poner un coste de 1 cuando la calefacción está apagada y un coste de 20 si la calefacción está encendida.

5. POLÍTICA ÓPTIMA

Nuestros costes están asociados al consumo de energía, por lo que nuestra política óptima es aquella que minimiza el consumo de energía intentando alcanzar la temperatura deseada por el usuario(la ideal, 22 grados). Utilizando los costes que hemos analizado en el apartado anterior, hemos simulado la política óptima para tales costes.

```
['16', 16.5, 17.0, 17.5, 18.0, 18.5, 19.0, 19.5, 20.0, 20.5, 21.0, 21.5, 22.0, 22.5, 23.0, 23.5, 24.0, 24.5, 25.0]
['off', 'on', 'on', 'on', 'on', 'on', 'on', 'on', 'on', 'on', 'on', 'on', 0, 'off', 'off', 'off', 'off', 'off', 'off']
```

Se puede comprobar que hay un resultado para cada estado ya que la política óptima devuelve la acción más óptima en función del estado en el que se encuentre. Las decisiones a tomar son o bien apagado, o bien encendido(on/off). La temperatura 22, al ser considerada la temperatura ideal (es decir, un estado sumidero/absorbente), no obtiene como óptima ninguno de estos dos valores, ya que cuando el usuario llega a su objetivo, como dice en la práctica, el termostato dejará de funcionar. Esto implica que las probabilidades de encendido o apagado en el estado 22 como los costes no tienen efecto si la temperatura de la habitación ha alcanzado dicho valor. Ninguna acción sale de dicho estado y por tanto nunca es origen.

Cuando la temperatura es inferior a la ideal, se encenderá, mientras que si supera a la ideal, se apagará, siguiendo un argumento lógico y con respecto a los costes establecidos. Las temperaturas excepto las excepciones tienen las mismas probabilidades en ambas tablas, para subir, bajar o mantenerse, por lo que las temperaturas por debajo mantendrán el mismo comportamiento y las de por encima igual.

El valor cuando hay una temperatura de 16 grados se debe a que en este caso, cuando la calefacción está apagada, tiene una probabilidad del 90% de mantenerse igual y una probabilidad del 10% de aumentar. Por tanto, se podría decir que en el momento que el coste de encenderlo es más de 10 veces mayor que de apagarlo, siempre será más eficiente tenerlo apagado, puesto que tan sólo puede mantenerse o subir (pero no bajar). Al fin y al cabo, es más rentable permanecer en off en este estado ya que el coste es menor y acabaría subiendo, que tener que encenderlo aunque haya más probabilidad de subir pero al ser el coste mayor es menos óptimo.

No es nuestro caso, ya que los costes que hemos establecido pretenden ser realistas(encender siempre será más caro que apagar). Si el coste de apagar llegase a ser mayor que el de encender, en los casos límites de 24.5 y 25 podría ser más eficiente encenderlo porque las probabilidades cambian a diferencia de las demás y hay que tener en cuenta que nuestro termostato depende de estas probabilidades.

Estos son los resultados que nos ha dado al aplicar la política óptima cuando converge al infinito y así obtener la acción más rentable para minimizar los costes:

```
[287.6, 278.0, 255.1, 230.4, 205.4, 180.4, 155.4, 130.4, 105.4, 80.6, 55.2, 32.2, 0, 1.6, 3.2, 4.7, 6.3999999999999995, 8.0, 9.399999999999999]
```

6. FASES DEL PROYECTO

Para poder sacar conclusiones y poder analizar correctamente el problema planteado hemos tenido que implementar en código la ecuaciones de bellman para cada estado, ya que la ecuación de Bellman permite calcular los valores óptimos de los estados en un proceso de decisiones secuenciales. Estos son los pasos que seguimos para obtener la política óptima:

1. Crear las tablas de transición para cada acción.
2. Pasarlas a python y meterlas en matrices.
3. Implementar la ecuación de Bellman.
4. Analizar los resultados.
5. Recrear el funcionamiento del termostato.

- ### DISEÑO

Nuestro código tiene dos partes esenciales, la simulación del termostato y la política óptica basada en las ecuaciones de Bellman. Queremos hacer especial hincapié en cómo hemos programado Bellman ya que es la función principal de lo que nos pide esta práctica y gracias a ella hemos calculado la PDM.

```python
def politica_optima(c_off, c_on):
    temp_final = 12
    actual = []
    siguiente = []
    optima = []
    coste = min(c_off, c_on)
    for i in range(19):
        actual.append(0)
        siguiente.append(coste)
        optima.append("")
        if i == temp_final:
            siguiente[i] = 0
```

```
            actual[i] = 0
            optima[i] = 0
    cont = 0
    #Calcular bellman
    while actual != siguiente:
        for i in range(19):
            actual[i] = siguiente[i]
        for indice in range(19):
            valor_off = 0
            valor_on = 0
            if indice != temp_final:
                for valor in range(19):
                    valor_off +=
round(actual[valor]*tabla_off[indice+1][valor+1],1)
                    valor_on +=
round(actual[valor]*tabla_on[indice+1][valor+1],1)
                valor_off += c_off
                valor_on += c_on
                siguiente[indice] = min(valor_off, valor_on)

                if siguiente[indice] == valor_off:
                    optima[indice] = "off"
                elif siguiente[indice] == valor_on:
                    optima[indice] = "on"
    return optima
```

Esta función crea tres listas, siguiente guardará los valores que dan en cada iteración mientras que la actual guarda la iteración anterior. La lista óptima guardará finalmente los valores de la acción más favorable por cada estado. Para la iteración cero todos los valores de V son igual a cero porque ninguna acción sale de dicho estado y por tanto nunca es origen. Para la iteración 1 los valores de V son iguales al coste mínimo de las dos acciones por eso inicializamos actual todo a ceros y siguiente todo al mínimo coste(apagado en nuestro caso).

Hasta que ambas listas no sean exactamente iguales, es decir, llegue una iteración que sea igual a la anterior, ya haya convergido, no saldrá del bucle. Igualamos la lista actual a la siguiente para que se guarden los valores en actual. Para calcular la V de los 19 estados, inicializamos todo a cero para restaurar las listas en cada iteración. Vamos recorriendo las matrices de probabilidades, multiplicando por la V de cada estado y almacenando el sumatorio de off y de on en valor_on y valor_off. Finalmente suma a este sumatorio el valor de cada coste y escoge el mínimo de los dos resultados y va guardando los nuevos resultados en el siguiente. Para saber qué acción es más óptima por cada iteración si coincide el valor de siguiente con el valor de off o siguiente con el valor de on, con el que coincida será la acción que minimiza el coste, por lo que obtenemos la acción más óptima por cada vuelta.

Vamos haciendo la ecuación de Bellman por partes:

1. sumatorio = $\sum Edest\ Paccion\ (Edest/Eorig) \cdot Vi\ (Edest)$
2. accion = coste $accion$ + sumatorio
3. min$accion$[$accion$(on), $accion$(off)

IMPLEMENTACIÓN

Por lo que el código que hemos tenido que implementar primero lee los datos de los archivos on_tabla.csv y off_tabla.csv, archivos que contienen las tablas de transición de cada estado de la acción correspondiente. Se lee el fichero de manera que las convertimos en matrices que se guardan en tabla_on y tabla_on para poder hacer uso de ellas y utilizarlas en la ecuación de bellman. Creamos una función que se llama política óptima que implementa el algoritmo de Bellman, en la cual no se meterá en el estado 22 y le hemos puesto directamente el valor de 0, ya que como anteriormente hemos explicado es un estado sumidero. Esta función devolverá una lista con la acción más óptima para cada estado. Hace n iteraciones, hasta que los resultados vayan convergiendo y de lo mismo la iteración i a la iteración i-1, en la que parará.

Para simular el comportamiento del termostato, hemos decidido generar un número aleatorio entre el 16 y 25, para que empiece en un estado cualquiera y no haya una temperatura definida. Se llama a la función de política óptima con los costes de apagado y encendido, en dicho orden. Si la política óptima indica que la calefacción debe estar apagada, se recorre la fila correspondiente en la tabla de probabilidad de apagado y actuará de manera aleatoria según las probabilidades(lo mismo si está encendida). Se irá actualizando la temperatura de la habitación por cada iteración del bucle, simulando de tal forma el comportamiento de un termostato. Una vez que alcanza la temperatura deseada por el usuario parará, cuando la habitación esté a 22 grados.

PRUEBAS

En un primer caso, el código con el que comenzamos a realizar la práctica no tenía ningún coste especificado, es decir, pusimos ambos valores a 1 simplemente para ir realizando el código. Al no tener ningún coste especificado en un primer momento estos fueron los resultados de las acciones más favorables que nos salieron como resultado.

```
['on', 'on', 'on', 'on', 'on', 'on', 'on', 'on', 'on', 'on', 'on', 'on', 0, 'off', 'off', 'off', 'off', 'off', 'off']
```

Lo cual nos hizo pensar que nuestro algoritmo de Bellman estaba bien planteado ya que temperaturas por debajo de 22 necesitan que la calefacción esté encendida y las que están por encima de la deseada necesitan que esté apagada.

A continuación hemos ido probando distintas combinaciones de costes para ver cómo variaba la política óptima. Al realizarlo, hemos dejado un valor estático mientras que incrementamos el otro. Como en el caso real hemos dado mayor coste al apagado que al encendido hemos decidido dejar estático el off e ir cambiando los datos de on. El primer caso que utilizamos de prueba fue poner coste(1,10), en este caso ya hay una variación en el que en el primer on, estado 16, es más rentable dejarlo apagado, ya que habrá algún momento en el que suba que dejarlo encendido y asumir el coste que conlleva hacer la acción de on.

```
['off', 'on', 'on', 'on', 'on', 'on', 'on', 'on', 'on', 'on', 'on', 'on', 0, 'off', 'off', 'off', 'off', 'off', 'off']
```

La siguiente prueba que hicimos es ver cuando cambiaba el siguiente on por off, ya que a medida que fuésemos aumentando los costes de on, las temperaturas más bajas irían cambiando su política óptima ya que si el coste es muy grande hay un punto en el que deja de ser rentable independientemente de si se llega al estado ideal con mayor rapidez. Por ello, pusimos los valores de costes(1, 68), con este valor el estado 16.5 cambia.

```
['off', 'off', 'on', 'on', 'on', 'on', 'on', 'on', 'on', 'on', 'on', 'on', 0, 'off', 'off', 'off', 'off', 'off', 'off']
```

```
0 :
17.0
1 :
17.5
2 :
18.5
3 :
19.5
4 :
20.0
5 :
20.5
6 :
21.0
7 :
21.5
8 :
22.0
```

Al realizar esta prueba y que confirmaba nuestra teoría de que al cambiar las temperaturas más bajas por la acción de off tardaría más en llegar al estado ideal, decidimos hacer una comprobación. Esta sería la simulación si dejásemos los costes en 1 y 1, empezando en el estado 16. Se puede ver en la imagen que ha llevado ocho iteraciones. Para no solo hacerlo una sola vez decidimos similar unas cinco veces. Los resultados fueron 8 iteraciones, 10 iteraciones, 22 iteraciones, 10 iteraciones y 18 iteraciones, una media de alrededor de 14 iteraciones. Lo que consideramos un número reducido de iteraciones ya que del 16 al 22 hay 13 estados entre medias y sólo hace una iteración más. Hay que tener en cuenta que en on se puede saltar al estado próximo o al siguiente(dos saltos de entre estados), por lo que la probabilidad de avanzar es también mayor.

Decidimos hacer la prueba de iteraciones para los costes de 1 y 68 empezando en el estado 16:

Como se puede ver en esta prueba, y afirmando nuestra observación al poner un mayor coste en on, los estados de abajo van cambiando su política óptima y tardan más en llegar al objetivo. Aún así, es de mayor rentabilidad dejarlo encendido y habrá algún momento en que llegue a asumir los costes. En este caso también lo hemos realizado cinco veces y nos ha dado 74 iteraciones, 68 iteraciones, 234 iteraciones, 81 iteraciones y 43 iteraciones. Como podemos ver varían mucho las pruebas entre sí, la media en este caso es de unas 100 iteraciones, siete veces mayor que el resultado anterior.

```
0 :
16.0
1 :
16.0
2 :
16.0
3 :
16.0
4 :
16.5
5 :
16.0
6 :
16.0
7 :
16.5
8 :
16.0
9 :
16.0
10 :
16.0
```

```
63 :
16.0
64 :
16.0
65 :
16.0
66 :
16.5
67 :
17.0
68 :
18.0
69 :
19.0
70 :
19.5
71 :
20.0
72 :
20.5
73 :
21.5
74 :
22.0
```

Como última prueba dejamos el coste de off en 1 y el coste de on en 300000, los costes de los estados de temperaturas más bajas cambian pero no todos:

```
['off', 'off', 'off', 'off', 'off', 'off', 'on', 'on', 'on', 'on', 'on', 'on', 0, 'off', 'off', 'off', 'off', 'off', 'off']
```

```
269308 :
18.5
269309 :
19.0
269310 :
19.0
269311 :
19.5
269312 :
20.0
269313 :
21.0
269314 :
21.0
269315 :
21.5
269316 :
21.5
269317 :
22.0
```

Esto es debido a que los estados más cercanos al estado de absorción necesitaran un coste mayor para que les compense que la calefacción esté apagada ya que tardan mucho más en alcanzar este estado, que asumir el coste de la acción de encendido y llegar al estado deseado. El tiempo que tardan en llegar al estado 22 hace que sea más óptimo la opción de encendido, porque aunque el coste sea 1, la probabilidad de subir si está apagado es de 0.1 , lo que es una probabilidad demasiado baja y de bajar es un 0.7 y las probabilidades de on de subir es de 0.5 al estado siguiente y 0.2 al estado siguiente del siguiente por lo que les compensa por probabilidad subir y asumir el coste ya que la probabilidad está a su favor, más que asumir n iteraciones hasta que haya un 0.1 de subir. Llega un punto que el coste de la acción on tendría que ser muy alto para que todas las temperaturas de por debajo de 22 les compense más realizar la acción de apagado.

Lo mismo sucedería con las temperaturas más altas de la deseada pero manteniendo estático el on y modificando el apagado a costes elevados(no sería nuestro caso).

Para nuestro caso en concreto la simulación del termostato de on y off, al poner valores asequibles en los costes y basados en el consumo de energía de la actualidad, solo compensa dejar apagado en el estado 16.

```
16.0
1 :
16.0
2 :
16.0
3 :
16.0
4 :
16.0
5 :
16.0
6 :
16.0
7 :
16.0
8 :
16.0
9 :
16.0
10 :
16.0
```

```
24 :
16.0
25 :
16.5
26 :
17.0
27 :
17.5
28 :
18.0
29 :
18.5
30 :
19.5
31 :
20.0
32 :
20.5
33 :
21.0
34 :
22.0
```

Una vez que sale del estado 16 ya solo tiende a ir aumentando hacia arriba por las probabilidades de on.

En definitiva, los estados que están por debajo como hemos podido ver les compensa más dejar la calefacción encendida y los que están por encima apagada. Si vamos modificando los costes se podrán ver variaciones de estos resultados porque dependiendo del coste al que le atribuye una acción igual compensa más asumir menor coste y aunque las probabilidades no estén a favor del estado al que quieres alcanzar llega un momento que se llega al estado deseado. Pero a medida que te vas acercando al estado deseado las probabilidades que están a tu favor te harán llegar más rápido al objetivo mientras que si la probabilidad es muy baja independientemente del coste, sale más óptimo mayor coste en menor tiempo que menor coste a una probabilidad muy baja de alcanzar el objetivo.

Por último para comprobar que nuestra política óptima es acertada hemos cambiado las probabilidades de 16 en la tabla off. Con los siguientes valores:

`16;0.7;0.1;0;0;0;0;0;0;0.2;0;0;0;0;0;0;0;0;0`

La política óptima que nos sale es la siguiente(con costes de 1 y 20):
['off', 'off', 'off', 'off', 'off', 'off', 'off', 'off', 'on', 'on', 'on', 'on', 0, 'off', 'off', 'off', 'off', 'off', 'off']
Esto implica que como en el resto de los estados en apagado la probabilidad de ir hacia abajo es más probable y en 16 se encuentra la posibilidad de ir al estado 20, es mejor asumir menor coste para que haya mayor probabilidad de salto. En el estado 20 ya es más óptimo el encendido para llegar al estado deseado.

7. PRESUPUESTO

En el caso de que alguien hubiera contratado nuestro servicio y debido al estudio y al código que hemos tenido que implementar pondremos un presupuesto de 550 euros.

Nos hemos basado en que si se gasta 40 céntimos la hora en condiciones normales, nos sale 9,6 euros al día, suponiendo que lo va a utilizar en un plazo de 10 años y teniendo en cuenta que le reduciremos alrededor de 3 euros(supuesto caso), si sigue las acciones correctas, ahorrará un total de 10. 950 euros(3*365*10). Si nosotros cobramos un 5% del total que se va a ahorrar, quedaría en un total de 547,5 que lo redondeamos a 550 por el análisis profundo.

8. CONCLUSIONES

La realización de la práctica ha sido de gran ayuda para ver el comportamiento del algoritmo de Bellman en casos con más de 5 estados (número de estados de ejercicios en clase). Hemos notado que, conforme va aumentando el número de estados, la cantidad de operaciones que hay que realizar van aumentando de forma vertiginosa, y con los 19 estados que se utilizan en la práctica, sería imposible realizar el cálculo a papel. También nos ha servido para entender mejor la combinación que hay entre las probabilidades, los costes y las acciones entre estados y hacer un análisis en profundidad de los resultados dados.

Además, una conclusión independiente que hemos sacado es que al observar la forma en que iba cambiando la política óptima en función de los valores que se le asignaba, hemos visto que en nuestro caso consideramos que sigue el comportamiento de una función logarítmica.

Para comprobarlo, hemos fijado el coste de encender la calefacción a 1, para ver cuáles son los valores más pequeños que se le pueden asignar al coste de apagar la calefacción para que se produzca un cambio en la política óptima y ver si nuestra observación es acertada, y son los siguientes:
coste_off < 7:

```
['on', 'on', 'on', 'on', 'on', 'on', 'on', 'on', 'on', 'on', 'on', 'on', 0, 'off', 'off', 'off', 'off', 'off', 'off']
```

7 <= coste_off < 55:

```
['on', 'on', 'on', 'on', 'on', 'on', 'on', 'on', 'on', 'on', 'on', 'on', 0, 'off', 'off', 'off', 'off', 'off', 'on']
```

55 <= coste_off < 405:

```
['on', 'on', 'on', 'on', 'on', 'on', 'on', 'on', 'on', 'on', 'on', 'on', 0, 'off', 'off', 'off', 'off', 'on', 'on']
```

405 <= coste_off < 2951:

```
['on', 'on', 'on', 'on', 'on', 'on', 'on', 'on', 'on', 'on', 'on', 'on', 0, 'off', 'off', 'off', 'on', 'on', 'on']
```

2951 <= coste_off < 21741:

```
['on', 'on', 'on', 'on', 'on', 'on', 'on', 'on', 'on', 'on', 'on', 'on', 0, 'off', 'off', 'on', 'on', 'on', 'on']
```

21471 <= coste_off <≈ 160252:

```
['on', 'on', 'on', 'on', 'on', 'on', 'on', 'on', 'on', 'on', 'on', 'on', 0, 'off', 'on', 'on', 'on', 'on', 'on']
```

Nótese que, en cada caso, aparece un ‘on’ extra por encima de la temperatura ideal (representada con 0).

Echando un vistazo rápido a los valores que hay que asignar al coste de apagar la calefacción para que se produzca un cambio en la política óptima (7, 55, 405, 2951, 21741), vemos que siguen un crecimiento exponencial. Más concretamente, cada uno de los números se obtiene multiplicando su anterior por, aproximadamente, 7,37 (en especial los números más grandes, ya que al tener más dígitos, también tienen un mayor grado de precisión). Y ya que tiene un crecimiento continuo, esto significa que lo podemos expresar en forma de función.

En conclusión, varía siguiendo la siguiente función:

$$f(x) = log_{7,37}(x)$$

Siendo el entero de f(x) el número de ‘on’ que se encuentran por encima de la temperatura ideal.