



**Universidad de Buenos Aires Facultad de Ingeniería Año  
2018 - 1er Cuatrimestre**

## **71.14 Modelos y optimización I** **Tercera Parte**

**Github:** <https://github.com/BlancoSebastianEzequiel/HeuristicaDisney>

**Integrantes:**

- Blanco, Sebastian 98539
- Llauro, Manuel Luis 95736
- Ruiz, Francisco 99429

# **Parte A**

## **Análisis**

Tenemos un problema de tipo mochila y scheduling. Hay que tener en cuenta las actividades y las restricciones que pone cada amigo para que todos disfruten al máximo de su estadía en el parque.

Sabemos que como mínimo deberíamos realizar las actividades que son señaladas como imprescindibles al menos una vez y además priorizar las actividades que obtuvieron mayor puntaje. Hacer al menos tres veces actividades de tipo montaña, y en caso de hacer la actividad Mickey's PhilharMagic, se podrá hacer todas las actividades montaña. En el caso de las actividades de categoría niños no se pueden realizar todas.

Las actividades tienen diferentes horarios durante todo el día, por lo tanto podemos realizarlas más de una vez al día, estas pueden ser vistas como actividades diferentes.

Todas estas actividades se deben realizar en conjunto, como lo desea Emilio, por lo tanto no podemos tener superposición, este es el motivo por el cual consideramos que tenemos también un problema de scheduling donde los amigos están incapacitados de realizar cualquier actividad mientras están en otra.

## **Objetivo completo**

Determinar qué actividades que deben realizarlas para el día que están en Magic Kingdom, para maximizar la suma de los puntajes que los amigos le pusieron a las actividades, cumpliendo los deseos y gustos de cada integrante del grupo.

## **Hipótesis**

- Consideramos que el parque tiene un horario de apertura y de cierre. Se puede ingresar exactamente en el horario de apertura y salir exactamente en el horario de cierre, sin perder minutos.
- Los amigos estarán en el parque al momento de su apertura.
- El horario en el cual el parque permanece abierto es desde el primer tooktime registrado hasta el último
- Cuando cierra el parque, el centro comercial sigue abierto.
- El tiempo de espera se mantiene constante entre cada update-time.
- El wait time es la suma el tiempo de espera y el de realización de la actividad correspondiente.
- Solo se puede comenzar a hacer una actividad en un took time
- El tiempo de finalización de la actividad es el tooktime más el wait time
- Trasladarse de una atracción a otra, no requiere pérdida de tiempo.
- Tomamos ir a comer como una actividad más.
- Se puede ir al restaurante durante toda la jornada.
- Se pueden repetir las actividades en el mismo día.
- La actividad con el mismo nombre y distintos tooktime pueden ser tomadas como actividades diferentes.
- El parque permanecerá abierto frente a cualquier situación climática.

# Heurística

Tengo dos estructuras de datos:

- El primero es una lista donde cada elemento es otra lista contiene qué actividad es, su categoría, su puntaje y si es o no imprescindible. Estas listas están ordenadas según si es imprescindible o no, en caso de empate por imprescindibilidad según el puntaje de mayor a menor. En caso de empate me quedo con el primero que se comparó, es decir, entre dos números iguales me quedo con el primero.  
(activitiesList[i] = [activity, category, points, imprescindible] )
- Luego tengo un diccionario donde la clave es la actividad y el valor es una lista de listas. Cada una de esas listas contiene el tooktime y su wait time. Estas listas de listas están ordenadas de menor a mayor según el wait time. En caso de empate me quedo con el primero que se comparó, es decir, entre dos números iguales me quedo con el primero.

Tengo variables donde se contabiliza la cantidad de veces que hago cada actividad, cada categoría, la cantidad de imprescindibles realizadas. Tengo una bivalente MPM que vale uno si hice la actividad Mickeys PhilharMagic al menos una vez

Tengo una lista donde almaceno las actividades que realizare, en el took time específico.

(madeActivities[ i ] = [activity, wait time, tooktime, category, points, imprescindible])

1. Inicializar la posición actual de la lista de actividades (activitiesList) en cero
2. Mientras el diccionario no este vacío o no haya hecho todos los imprescindibles o la cantidad de veces que hice actividades de categoría montaña sea menor a tres. Si es falso saltar al ítem 4
  - a. Seleccionar el elemento de la posición actual de la lista de actividades (activitiesList), veo que actividad es y busco los took times con su wait time correspondiente para la misma.
  - b. Mientras la lista de took times y wait times no esté vacía, itero, si no, saltar a 3
    - i. Leo el primer par de took time-wait time.
    - ii. Si la cantidad de veces que hice comida es uno y la actividad actual es comida, eliminarla de activitiesList y saltar a 3.
    - iii. Si la cantidad de imprescindibles realizados hasta el momento es menor al total de imprescindibles y la actividad que leo no es imprescindible, saltar a 5.
    - iv. Verificar que el par (took-time, wait-time) actual sea compatible, es decir que no se superponga, con el resto de las actividades que estén dentro de madeActivities, si no hay superposición entonces:
      - Si la categoría de la actividad actual es montaña y me falta una para hacer todas las montañas y no hice Mickeys PhilharMagic entonces:
        - a. Si la actividad Mickeys PhilharMagic fue eliminada para todos los tooktimes, es decir, ya fue descartada como para hacerse, entonces elimino el elemento actual de la lista.
        - b. saltar a 3
      - Si la categoría actual es niños y me falta uno para hacer todas las actividades de categoría niños entonces elimino el elemento actual de la lista y saltar a 3.
      - Guardar en madeActivities la actividad que estoy leyendo con toda su información adicional.
      - Incrementar la cantidad de veces que hice esa actividad y esa categoría
      - Si la actividad es imprescindible, incrementar la cantidad de imprescindibles realizadas.
      - Saltar a 3

- v. Si se superpone, eliminar el elemento actual de la lista y seguir con el siguiente par tooktime-waittime.
3. Si hice todos los imprescindibles y la etiqueta para realizar este ítem es verdadera entonces reordenar activitiesList según el puntaje solamente y pongo una etiqueta en falso tal que este ítem (ítem 3) no se vuelva a ejecutar.
4. Incrementar la posición actual de activitiesList y saltar a 2
5. Si la cantidad de imprescindibles es menor al total, o la cantidad de montañas es menor a tres o la cantidad de niños es igual al total o si no hice Mickeys PhilharMagic y la cantidad de montañas es igual al total, entonces la heurística no encontro una solución.
6. Obtener el puntaje de la lista de actividades hechas (madeActivities) sumando el quinto elemento de cada lista.

### **Casos teoricos donde funciona mal**

En casos donde los imprescindibles tienen wait times muy bajos y puntajes muy altos y las actividades no imprescindibles tienen wait times muy altos pero puntajes muy bajos, ocurriría que nuestra heurística trataría de hacer al menos una vez las imprescindibles, luego intentaría agregar actividades no imprescindibles y termina agregando la que no se superponga, pero que al tardar mucho en realizarse no dejaría meter muchas otras actividades más, es decir, nuestra heurística, una vez agregado al menos una vez las imprescindibles, intenta agregar al menos una vez las no imprescindibles, entonces como en este caso teórico los wait times de las no imprescindibles son muy altos terminaría agregando como actividades a hacer todas las no imprescindibles que pueda y cuando quiera volver a ingresar imprescindibles quizás no pueda.. En ese caso hubiera sido más conveniente seguir metiendo actividades imprescindibles.

### **Casos teoricos donde funciona bien**

Por el contrario en casos donde los imprescindibles tengan puntajes muy bajos y waittimes altos, nuestra heurística intentaría hacer al menos una vez los imprescindibles y luego seguir con los demás. Si los no imprescindibles tuvieran puntajes muy altos y wait times muy bajos, pasaría que metería al menos una vez los no imprescindibles, luego posiblemente no pueda introducir más imprescindibles debido a su alto tiempo de ejecución, y metiera más no imprescindibles. En ese caso cumpliría con la condicion del enunciado de hacer la menos una vez las imprescindibles, pero luego aprovecharía metiendo no imprescindibles que tienen mayor puntaje. De esta manera el puntaje total sería más alto al aprovechar hacer actividades no imprescindibles.

Otro caso donde funcionaria bien es si todos tuvieran wait times muy bajos, todos el mismo wait time, y todos el mismo puntaje porque es posible hacer más actividades, cumpla con las mínimas imprescindibles, pero al todos tener mucho puntaje y el mismo wait time para todo tooktime nuestra heurística trataría a ser la menos una de cada una, y equitativamente hacer muchas de cada una. Tendríamos que haríamos muchas veces todas las actividades de forma pareja con un puntaje altísimo total.

### **Propuesta de mejora de la heurística**

Para mejorar la heurística, una vez que se hicieron los imprescindibles al menos una vez y cumpliendo con las cotas inferiores de montaña:

1. Ordenar para cada actividad sus waittimes, con su información adicional, de menor a mayor, en caso de empate elige aquel que fue primero en la comparación, y seleccionar los primeros de cada lista de esos waittimes ( uno para cada actividad).

2. Dividir el puntaje de la actividad por el waittime seleccionado para esa actividad. Ordenarlos de mayor a menor, en caso de empate elige aquel que fue primero en la comparación.
3. Sacar al mayor y comprobar que sea compatible con aquellas actividades que ya estén en madeActivities. Si lo es agregarlo a madeActivities, sino descartar ese waittime para esa actividad.
4. Agarrar al siguiente el elemento de la lista de waittimes para la actividad que se intentó agregar en el paso previo. Mientras haya waittimes volver a 2.
5. Cuando no haya más waittimes en ninguna lista, devolver la suma de los puntajes de las actividades en madeActivities.

### **Resultados de la heurística**

El tiempo que tardó en ejecutarse la heurística 0.141438961029 segundos donde el puntaje total obtenido es de 139. Si vemos cómo se distribuyen los puntajes, se ubican mayoritariamente sobre los imprescindibles, pero el error de la heurística que se comentó antes y aplicado a este resultado es que no aprovecha hacer más actividades imprescindibles que tiene un alto puntaje, por eso es que nuestro puntaje final no es del todo bueno.

## **Parte B**

1) En este punto Coca Cola nos ofrece montar un stand que ocupa 20 metros en algún lugar del predio. La idea sería permitir al modelo poder decidir como sería más conveniente distribuir el espacio, dado que Coca Cola está dispuesta a abonar \$30000 por ese espacio.

Para averiguar si es conveniente este trato, se le puede agregar al funcional el valor acordado y agregar el la restricción de espacio que ya serán ocupados 20 metros con el stand.

Al comparar los resultados de los modelos, podemos observar que el resultado del óptimo sin el stand es de un valor de \$11.560.000 mientras que el óptimo obtenido incorporando el stand es de \$11.619.300. Por esta razón podemos decir que finalmente es conveniente aceptar el trato con Coca Cola.

2) Si el precio de las entradas generales sube a \$815, el planeamiento obtenido por la corrida del modelo anterior (con el precio a \$800) seguirá siendo el óptimo que se puede obtener con los recursos disponibles. Lo único que cambiará es el funcional que de ser \$11560000 pasará a ser \$11567500, es decir aumentará \$7500 en total, a razón de \$15 más por entrada vendida (el total de entradas generales vendidas son 500).

Esto lo podemos ver porque al calcular el rango de variación del coeficiente de las entradas generales, obtenemos que el  $c_j$  correspondiente tiene que ser menor a \$820 para la solución óptima, como el precio sugerido está dentro del rango recién anunciado concluimos que no pasará nada respecto a la solución anterior.

En la solución mostrada por GLPK, la slack asociada a la restricción de la demanda mínima de entradas generales, tiene un valor marginal de -20, es decir, que si nos obligan a realizar una más de estas perderíamos \$20 por lo tanto, si las venderíamos a \$820 no tendríamos esa pérdida, pero como el precio sugerido es \$815, seguiríamos perdiendo \$5 por cada entrada.

3) Tenemos una disponibilidad total de 800 parlantes para comprar y nosotros utilizamos efectivamente 450 parlantes en nuestro espectáculo, por este motivo nos quedan 350 parlantes sin comprar.

Con la propuesta que nos hace Universal Studios, estamos dispuestos a comprar los 350 parlantes que aún no habíamos adquirido por un total de \$245000 y obtener a cambio \$5000000 por cada 50 parlantes que le damos entonces el total de dinero que recibimos es de \$35000000.

Nos quedaría una ganancia neta de \$34750000 al darle los 350 parlantes que no habíamos utilizado a Universal Studios.

4) Analizando el estado de cada restricción a partir de lo que se puede ver en la columna St tenemos que:

- **Espacio:** tiene en la columna St las siglas NU, este recurso está saturado. Todos los m<sup>2</sup> disponibles están siendo utilizados.
- **Sonido VIP:** tiene en la columna St las siglas NL, la cobertura de los parlantes para las entradas VIP es la justa y necesaria, es decir que no hay parlantes que cubran entradas VIP que no estén hechas, la slack asociada vale 0. Está cumpliendo con la restricción de mínimo, aunque ésta no le convenga al funcional.
- **Sonido GENERAL:** es exactamente igual a la restricción anterior, pero para las entradas generales.
- **Sonido Total:** tiene en la columna St la sigla B, esta variable slack representa la cantidad de parlantes disponibles no utilizados, este recurso no está saturado, tiene un sobrante de 350 unidades.
- **Mínimo de Generales:** tiene las siglas NL, no hay entradas generales más allá de aquellas que son realizadas para el cumplimiento de la recomendación del Departamento de Marketing. Está cumpliendo con la restricción de mínimo, aunque ésta no le convenga al funcional.
- **Mínimo de VIP:** tiene la sigla B, la variable slack asociada a esta restricción representa la cantidad de entradas VIP hechas por sobre la demanda mínima, en este caso toma el valor de 7650 entradas por sobre el mínimo que son 100.

5) Nos ofrecen un predio del mismo tamaño del que ya poseemos, vamos a tener 8000 m<sup>2</sup> adicionales. Ahora en total, tenemos una disponibilidad de espacio de 16000m<sup>2</sup>.

Con el plan anterior, teníamos un funcional de \$11560000 y ahora deberíamos invertir un total de \$1172000 para ganar esos metros cuadrados adicionales. Por lo que a priori no parece ser un buen negocio invertir en este predio, porque ya con el óptimo obtenido tuvimos un beneficio menor a la inversión a realizar.

Para verificar que realmente no es un buen negocio, deberíamos cambiar de tabla viendo el rango de variación del recurso espacio y hasta qué punto sigue siendo óptima esta tabla, y ver que el funcional no varíe en más de \$1172000 con respecto al actual.

6) Nos proponen agregar un sector para verdaderos fans de Star Wars. Se estima que 80 fans entran en un espacio de 10 metros cuadrados, serían 0.125 fans por metro cuadrado y como un parlante le brinda cobertura de sonido a 15 fans, tenemos que 1/15 parlantes le proporciona sonido a un solo fan.

Teniendo el uso de los recursos que me implica crear una sola unidad de este nuevo tipo de entradas para los verdaderos fanáticos de Star Wars, puedo calcular su lucro cesante:

$$\text{LucroCesante} = VM_{\text{parlante}} * USO_{\text{parlante}} + VM_{\text{espacio}} * USO_{\text{espacio}}$$

$$\text{LucroCesante} = 0 * \frac{1}{15} + 1465 * 0.125 = 183.125$$

Pero a este valor falta amortizarle el valor de la compra del parlante, porque si bien hay disponibilidad para comprar parlante ( $VM_{\text{parlante}}$  es 0), cada parlante hay que comprarlo y su costo es de \$700. Entonces por cada entrada para verdaderos fanáticos de Star Wars se gastaría \$46.666 en parlantes, que sumado al lucro cesante, obtenemos que la entrada para verdaderos fanáticos de Star Wars debe tener como mínimo un precio de venta de \$229.792 para que nos resulte conveniente asignarles un lugar en nuestro espectáculo.