Gestión de Cadenas de Suministros IND430.

Tarea Computacional N°2

Revenue Management & Scheduling

Instrucciones:

I- Resolver los ejercicios de Localización y Layout que se detallan, empleando herramientas de programación matemática (Gurobi, Jump, MIP Solver, AMPL u otras, no se permite Excel). Se deberá adjuntar en un notebook el planteamiento del modelo, código de la implementación y la solución al problema planteado, según corresponda, dando respuesta a cada punto, de forma clara y ordenada. Revise que el notebook debe correr para ser corregido por el equipo docente.

II- La tarea debe ser realizada en grupos de 3 ó 4 alumnos, todos de la misma sección.

III- Método de entrega: subir archivo al enlace de webcursos.

Restricciones:

I- Esta estrictamente prohibido cualquier intercambio de información o ayuda entre personas de distintos grupos, así como cualquier instancia de resolución global o ayuda de personas externas al grupo. Cualquier sospecha relativa a infracciones al Código de Honor serán reportadas a las autoridades competentes.

II- En caso de emplear códigos, la autoría debe ser propia e inédita. No se aceptarán códigos de terceros.

III- Toda referencia y uso de código de internet debe ser documentado de acuerdo con la norma APA.

IV- Los datos disponibles para realizar la tarea están en https://shorturl.at/E9NdO :

Problema 1 [10 ptos]. Revenue Management, regla de Littlewood.

a- [5 ptos] Considere un hotel con 100 habitaciones. El hotel ofrece una tarifa alta de $150, con una demanda que sigue una distribución normal con media de 40 y varianza de 10. También ofrece una tarifa baja de $70. Obtenga la cantidad promedio a reservar para tarifa alta y simule N=2000 veces la cantidad demandada y compute lo siguiente: la cantidad vendida de cada tarifa, el ingreso total, y las cantidades de over- y under-stock ("stock que sobró" y "stock que faltó") de la tarifa alta. Reporte el promedio de cada una de estas cantidades sobre las N repeticiones

b- [5 ptos] Considere un hotel con 100 habitaciones. El hotel ofrece una tarifa alta de $150, con una demanda que sigue una distribución Poisson con media de 40, y también ofrece una tarifa baja de $70. Repita el experimento de la parte a., pero utilizando la distribución Poisson para la demanda alta. Obtenga la cantidad a reservar para la demanda alta, luego simule la demanda N=2000 veces, y reporte el ingreso promedio, las cantidades promedio vendidas de cada tarifa, y para la tarifa alta, el over- y under-stock promedio.

Problema 2 [10 ptos]. Revenue Management, heurística EMSR-b.

Usted trabaja en el departamento de Revenue Management de una emprea ferroviaria y su objetivo es optimizar el uso de los asientos disponibles en trenes con diferentes tarifas. En este contexto, se le pide que programe una solución que permita calcular cuántos asientos se deben reservar para cada tipo de tarifa con el fin de maximizar los ingresos.

Se le proporciona un tren con una capacidad total de 120 asientos y cuatro tipos de tarifas: D, E, F, G, cuyos precios son $50, $80, $40 y $70 respectivamente.

a- [3 pts] Programe una función llamada EMSR-B que reciba como input:
Un entero *C* que representa la capacidad total de asientos (en este caso, 120). Un arreglo *q* con los precios de las tarifas en orden descendente. Un arreglo *mu* con las demandas esperadas de cada tarifa. Un arreglo *var* con las varianzas correspondientes a cada tarifa (de tal manera que el producto j con tarifa *q[j]* tiene demanda distribuida Normal(*mu[j]*, *var[j]*)). La función debe devolver un arreglo *R* con la cantidad de asientos que se deben reservar para cada tarifa. Para cada tarifa *j*, el número de asientos reservados *R[j]* se debe calcular teniendo en cuenta que la demanda para dicha tarifa sigue una distribución normal, y que maximizar los ingresos implica priorizar las tarifas más caras.

b- [7 pts] En los archivos adjuntos *TrainData.csv* y *TestData.csv* se encuentran datos históricos y de prueba sobre la demanda de los últimos 500 periodos para las cuatro tarifas. Utilice el archivo *TrainData.csv* para ajustar una distribución normal a cada tipo de demanda (tarifas D, E, F, G) y aplique su función EMSR-b para determinar la cantidad de asientos que deberían reservarse para cada tarifa. Luego, evalúe su decisión utilizando los datos de demanda real contenidos en *TestData.csv* y compare si la cantidad de asientos reservada para cada tarifa es adecuada con respecto a la demanda observada.

Problema 3 [10 ptos]. Revenue Management, Overbooking.

Considere un vuelo de Santiago, Chile, a Río de Janeiro con 200 asientos disponibles. El precio promedio de un boleto es de $475. Se estima que el número de pasajeros que realizan una reserva pero no se presentan sigue una distribución normal, con una media de 30 y una desviación estándar de 15. Con el fin de maximizar los ingresos, se decide aceptar un número adicional de reservas.

a. [5 ptos] Determine cuántas reservas adicionales se deberían aceptar, teniendo en cuenta que el costo de compensar a los pasajeros sobrevendidos es de $800 por pasajero.

b. [5 ptos] Si se permiten 220 reservas, calcule la probabilidad de que se necesite compensar a los pasajeros que no se presenten.

Problema 4 [15 ptos]. Problema *Flow Shop Scheduling*.

Usted es un gerente de producción encargado de optimizar el flujo de trabajo de una instalación de fabricación. Su objetivo es programar una serie de trabajos en múltiples máquinas para lograr la operación más eficiente posible.

Datos de Trabajos y Máquinas:

Tiene **10 trabajos** que deben procesarse en el siguiente orden en **3 máquinas**. Los tiempos de procesamiento para cada trabajo en cada máquina se describen en la siguiente tabla:

Trabajo/Máquina	Máquina 1	Máquina 2	Máquina 3
T1	5	3	9
T2	6	1	6
T3	8	3	8
T4	8	2	7
T5	2	4	6
T6	7	2	9
T7	10	2	8
T8	5	3	4
T9	1	4	6
T10	1	2	4

a) [3 ptos] Formule un modelo de optimización lineal con variables enteras para minimizar el tiempo total de finalización de todos los trabajos. Defina claramente las variables de decisión, la función objetivo y las restricciones. Considere M máquinas y N trabajos, donde cada trabajo debe procesarse en todas las máquinas en el orden secuencial: primero en la máquina 1, luego en la máquina 2, y así sucesivamente. El tiempo de procesamiento para el trabajo i en la máquina j está dado por p_{ij}.

b) [7 ptos] Implemente su modelo computacionalmente y resuelva una instancia con 10 trabajos en 3 máquinas. Presente su solución en una tabla, donde para cada trabajo i (filas) y cada máquina j (columnas) se muestre el tiempo de inicio y finalización del procesamiento de cada trabajo en cada máquina.

c) [5 ptos] Resuelva el problema del apartado anterior utilizando el algoritmo de Johnson. Explique la metodología de solución.

Problema 5 [15 ptos]. Problema *Job Shop Scheduling*.

Usted tiene 8 trabajos (T1-T8) que deben procesarse en 5 máquinas diferentes. Cada trabajo debe seguir una secuencia específica a través de las máquinas, y los tiempos de procesamiento varían para cada una de ellas.

La tabla a continuación muestra la secuencia en la que cada trabajo debe pasar por las máquinas y el tiempo que tarda en cada una. La primera columna indica la secuencia de la máquina, y la segunda columna indica el tiempo de procesamiento para ese trabajo en esa máquina.

T1	3	50	1	90	4	30	5	60	2	40
T2	2	65	3	35	5	75	1	50	4	45
T3	5	80	2	60	4	25	1	70	3	55
T4	1	55	4	40	3	45	2	65	5	30
T5	4	45	3	85	5	60	2	35	1	90
T6	3	40	5	70	1	85	4	35	2	55
T7	1	65	3	40	5	50	2	75	4	30
T8	2	55	5	45	3	80	4	60	1	65

A) [10 ptos] Debe resolver el problema de Scheduling, minimizando el makespan o tiempo de entrega del último trabajo. Debe adjuntar el código que usó, la secuencia de trabajos óptima y el makespan del procedimiento. Cada trabajo debe seguir su secuencia en las máquinas indicada en la tabla. Por ejemplo, el trabajo T1 primero debe procesarse en la máquina M3 y toma 50 unidades de tiempo (UT), luego pasa a la máquina M1 y toma 90 UT, luego a la máquina M4 con 30 UT, a la máquina M5 con 60 UT, y finalmente máquina M2 y toma 40 UT.

B) [5 ptos] Implemente una función que grafique una carta Gantt donde salgan todas las máquinas (eje y) y para cada máquina se muestre visualmente a través del tiempo (eje x) la ejecución de cada tarea. Se debe distinguir claramente la ubicación en el tiempo de cada tarea.