Informe proyecto

Grupo: 6
Nombres: Diego Villegas,Tomas Donoso, Martin Moraga, Nelson Sepulveda

Descripción del problema

Un fabricante de chips tiene que planificar la producción para los próximos M meses. Se conoce el precio de venta de los chips as´ı como la demanda mensual de estos (la demanda varıa en cada mes y va en aumento (demandaMes1 <demandaMes2 <.. . <demandaMesM). Se prevee una subida mensual de los costos de producción (costoMes1 <costoMes2 <.. . <costoMesM). La fabrica puede generar una producción mensual fija en horario normal y además, en horario extra la producción puede ser incrementada en cierta cantidad de unidades a un costo mensual (diferente para cada mes). Se cuenta con un almacén de capacidad limitada para almacenar el exceso de producción incurriendo en un costo unitario mensual (igual para cada mes). Al final de los M meses la existencia de productos en almacén tiene que ser nula y las demanda de chips siempre debe ser satisfecha.

Además, se ha definido que:

- En el mes 1 la cantidad de chips fabricados tiene que ser a lo mas la quinta parte de la producción total de chips para todo el periodo de planificación (sumatoria de M meses).
- Si decide producir en horario extra entonces, la cantidad de unidades producidas puede ser a lo menos la mitad de la producción mensual en horario normal para el mes en curso.

Función Objetivo: Minimizar el costo total

Modelo Matemático

Parámetros

- d_i: Demanda del mes i.
- c_i: Costo de producción normal en el mes i.
- e_i: Costo de producción en horario extra en el mes i.
- a: Costo de almacenamiento por unidad en el mes.
- p: Producción mensual final en horario normal.

Variables de decisión

- x_i: $\begin{cases} 1, & \text{Se produce en horario extra} \\ 0, & \text{Caso contrario.} \end{cases}$
- y_i: la cantidad de unidades almacenadas al final del mes i.
- h_i: la cantidad de unidades producidas en horario extra en el mes i.

Función Objetivo

$$\text{mín}(Z) = \sum_{i=1}^{m} (c_i * p + e_i * h_i + y_i * a)$$

Restricciones

- Restricción de producción máxima en el mes 1: $p + h_1 \leqq (\sum_{i=1}^{m} p + h_i) * 1/5$
- Restricción de producción en horario extra en relación a la producción normal: $h_i \geq 0,5 * x_i * p$
- Restricción de capacidad de almacenamiento: $y_i \leqq S$, donde S es: capacidad máxima de almacenamiento.
- Restricción de balance de inventario: $y_i = y_{i-1} + p + h_i - d_i$ Para $i > 1$, $y_1 = p + h_1 - d_1$
- Restricción de no almacenamiento al final de los m meses: $y_m = 0$
- Restricción de satisfacción de la demanda: $p + h_i + y_i \geq d_i$
- Restricción de no negatividad: $y_i, h_i \geq 0$

Generación de instancias y Factibilidad

Para la generación de instancias en primer lugar se definieron las 15 instancias a utilizar para así generar los parámetros, definimos una lista M que contiene estos valores, posteriormente se inician los input que piden los limites de cada parámetro definido en el modelo, anteriormente descrito, luego se definen las listas que contendrán los valores de los parámetros en los meses correspondientes. imagen de referencia:

```python
from random import randint

# De la tabla de instancias M, escogemos las siguientes valores, respectivamente.
M = [7,15,25,35,45,115,145,175,185,195,500,650,700,800,900]
# inputs de la demanda del mes
d_inf = int(input("Ingresa el limite inferior de la demanda: ")) # 1
d_sup = int(input("Ingresa el limite superior de la demanda: ")) # 20000
# inputs del costo de produccion en horario normal
c_inf = int(input("Ingresa el limite inferior del costo de produccion en horario normal: ")) # 1
c_sup = int(input("Ingresa el limite superior del costo de produccion en horario normal: ")) # 10000
# inputs del costo de produccion en horario extra
e_inf = int(input("Ingresa el limite inferior del costo de produccion en horario extra: ")) # 10
e_sup = int(input("Ingresa el limite superior del costo de produccion en horario extra: ")) # 10000
# inputs costo de almacenamiento por unidad en el mes (valor fijo).
a_inf = int(input("Ingresa el limite inferior del costo de almacenamiento por unidad en el mes: ")) # 10
a_sup = int(input("Ingresa el limite superior del costo de almacenamiento por unidad en el mes: ")) # 1000
# inputs cantidad de unidades producidas al mes en horario normal (valor fijo).
p_inf = int(input("Ingresa el limite inferior de la cantidad de unidades producidas en horario normal: ")) # 1
p_sup = int(input("Ingresa el limite superior de la cantidad de unidades producidas en horario normal: ")) # 500

# Nota: El limite inferior de la produccion siempre debe ser considerablemente menor al limite superior de de la demanda para que exista factibilidad.

#Listas de listas con las demandas de prueba
d=[]
#Listas de listas con los costos de produccion en horario normal en el mes
c=[]
#Lista de listas con los costos de produccion en horario extra en el mes
e=[]
#Lista costo de almacenamiento por unidad en el mes (valor fijo)
a=[]
#Lista de produccion mensual fija en horario normal
p=[]
```

Antes de continuar cabe recalcar que los limites a escoger son a elección, sin embargo deben respetar ciertas características, en primer lugar el limite inferior de la producción siempre debe ser considerablemente menor al limite superior de la demanda para que exista factibilidad, ya que se debe cumplir que para m meses la sumatoria de producción en horario normal debe ser menor o igual a la sumatoria de los m meses de la demanda(p*m<=sum(d)), para que así se cumpla la restricción de no almacenamiento al final de los m meses. También se debe tener en cuenta los rangos de los limites en los costos de producción en horario normal y en horario extra, ya que como podemos observar las ultimas instancias son de 900 meses, y si el rango en el que oscilan estos datos es menor a 900 el problema seria infactible, por ende es ideal tener parámetros mayores o iguales a 900(Los valores que están comentados a lado de los input son los que posteriormente utilizamos).

Ahora continuando con el desarrollo de la generación de instancias, tomamos el primer caso conjunto de parámetros anteriormente mencionado p*m<=sum(d) en m meses, todo esto cumpliendo la restricción de una demanda alzan te al paso de los meses, sin embargo este ultimo punto lo hablaremos en aleatoriedad. Para generar los parámetros para la producción en horario normal simplemente cojeemos el valor arrojado por el randint y lo repetimos los m meses debido a que la producción en horario normal es fija. Posteriormente se obtienen los valores de la demanda cumpliendo con la restricción del aumento de la demanda, el como se logro esto sera explicado en aleatoriedad, y finalmente con la ayuda de un if se comprueba si se cumple que p*m<=sum(d), si esto se cumple los valores de producción en horario normal y demanda son agregados a sus respectivas listas, sino se vuelve a intentar nuevamente. Tal como se muestra en la imagen:

```
i=0
while(i<15):
  meses=M[i]
  flag=True
  while flag:#Restriccion almacenamiento nulo al terminar el problema es decir p*m <= sum(d), en otras palabras, la produccion en horario normal debe ser m
    i_2=0
    produccion = randint(p_inf,p_sup)
    lista_aux1=[]
    while i_2<meses: # Parametros de produccion horario normal
      lista_aux1.append(produccion)
      i_2+=1
    i_2=0
    aux=0
    sum_d=0
    lista_aux=[]
    while i_2<meses: # Parametros de demanda, cumpliendo el requisito de cada demanda mayor al paso del mes
      if(i_2 == 0):
        superior = meses/d_sup
        valor=randint(d_inf,int(d_sup*superior))
      if(i_2 >= 1):
        superior = (meses-aux)/valor
        valor=randint(valor+1,int(((meses-aux)*superior)+valor))
      lista_aux.append(valor)
      aux-=1
      i_2+=1
      sum_d+=valor
    if(meses*produccion<=sum_d):# Comprobacion de la restriccion anteriormente mencionada
      flag=False # Se cumple la restriccion, sino se vuelve a intentar.
  p.append(lista_aux1)
  d.append(lista_aux)
```

Luego para los costos de producción en horario normal se utilizo el mismo algoritmo utilizado en la obtención de parámetros de demanda, ya que ambos debían cumplir la misma restricción de aumento a medida que pasaban los meses, dicho algoritmo sera explicado en la sección de aleatoriedad.

Para los parámetros de almacenamiento al igual que para la capacidad de producción, se cojee el valor obtenido del radint y se repite los m meses, finalmente para los parámetros de costo de producción en horario extra, la restricción es que los valores deben ser diferentes para cada mes, por ende con la ayuda de 'flag = valor in lista_aux', se comprueba si el valor existe en la lista, si este existe se vuelve a iterar, si este no existe se agrega el valor a la lista y se continua, ambos parámetros anteriormente mencionados se reflejan en la siguiente imagen:

```
almacenamiento = randint(a_inf,a_sup)
lista_aux=[]
i_2=0
while i_2<meses: #Parametros costo de almacenamiento fijo
  lista_aux.append(almacenamiento)
  i_2+=1
a.append(lista_aux)

i_2=0
lista_aux=[]
while i_2<meses: #Parametros costo de produccion en horario extra, comprobacion de restriccion de parametros diferentes.
  flag=True
  while flag:
    valor = randint(e_inf,e_sup)
    if(i_2==0):
      lista_aux.append(valor)
      flag=False
    if(i_2>=1):
      flag = valor in lista_aux # se comprueba si se repite el costo en algun lugar de la lista, es decir en algun mes.
      if(not(flag)):
         lista_aux.append(valor)
  i_2+=1
e.append(lista_aux)
```

Aleatoriedad

Para la generación de los parámetros se utilizo la función randint para obtener valores aleatorios de forma directa con los limites definidos en todos los parámetros, excepto para la demanda y para los costos de producción en horario normal, para lograr que estos valores fueran aumentando a medida que pasan los meses, se realizo lo siguiente, para el primer mes se calcula un nuevo limite superior, en donde calculamos el porcentaje de aumento máximo en relación al limite superior, es decir 'superior = meses/c_sup', que representa ese porcentaje, el valor máximo que puede alcanzar el parámetro. por ejemplo: si tenemos que m es igual a 7 y queremos calcular el costo de producción en horario normal para un rango [1,10], superior seria igual a 0.3, ese dato se multiplica, al limite superior obteniendo 3, esto significa que el valor de este parámetro en el mes 1, oscila entre [1,3], ya que si fuera superior a 3 el problema es infactible. Continuando con este ejemplo la siguiente iteración se realiza el mismo podrecimiento sin embargo ya no son 7 meses sino que son 6 meses, y la cantidad de datos escogibles se reduce, ya que el valor del limite inferior debe aumentar,ya que no se pueden escoger valores menores,esto quiere decir que si en la primera iteración obtenemos el valor 2, en la segunda iteración el rango escogible seria [valor+1,((7-1)*(7-1/valor)+valor)] y así sucesivamente, tal como se muestra en la siguiente imagen para la obtención de parámetros de los costos de producción en horario normal.

```
i_2=0
aux=0
lista_aux=[]
while i_2<meses: #Parametros costo de produccion en horario normal en el mes.
  if(i_2 == 0):
    superior = meses/c_sup
    valor=randint(c_inf,int(c_sup*superior))
  if(i_2>=0):
    superior = (meses-aux)/valor
    valor=randint(valor+1,int(((meses-aux)*superior)+valor))
  lista_aux.append(valor)
  aux-=1
  i_2+=1
c.append(lista_aux)
```

Link a GoogleDrive

Click Aqui: Generador de instancias.