EXA FINALES

1. **18 DICIEMBRES**
   a. p1 ejercito español tema 1
   b. p2 LP
   c. p3 game thery invasion normandia
2. **17 DICIEMBRES**
   a. p1 modelado del juego raro ese de la tabla e ir viendo los conjuntos y solo aplicarles una operacion
   b. p2
   c. p3
3. **16 DICIEMBRES**
   a. **p1 modelado de las** centrales de los bomberos
   b. p2
   c. **p3 Decision theory y game theory,** el de penelope investigadora y beca**,** hay q plantear arbol y pure strategy
4. **15 DICIEMBRES**
   a. **p1 modelado** cuatro alumnos que se van a cenar por navidad Y GOAL PROGRAMING
   b. p2
   c. **p3 Non linnear model**
5. **14 DICIEMBRES**
   a. **p1 modelado** -->el del chaval q se prepara el ironman ->Y GOAL PROGRAMING y muy bien plantedo
   b. p2
   c. p3
6. **13DICIEMBRES**
7. **12 DICIEMBRES**

****

1. **18 JUNIOS**
   a. **p1 Modelado** de los influencer y -->en el apartado 2 has tenido que definir doblemente como activar la binaria nueva que te habias crado, tanto para cuando es 1 como cuando es cero
   b. p2 LP
   c. p3 Non linear programmin.-->teorema de ballesta, necesarias condiciones, sistema para sacar ptos
2. **17 JUNIOS**
   a. LP
   b. Modelado -->3 profes que dan un curso online
   c. MIP
   d. NOP-->ver en funcion del valor si es el hessiano estrictamente positivo o no
3. **16 JUNIOS**
   a. p1 --> Modelado la maquina expendedora y GOAL programing
   b. p2
   c. p3
4. **15 JUNIOS**
   a. **p1** -->modelado el de los pack de productos y COMPROMISE PROGRAMING

****

PROBLEMAS CLASE

1. MODELADO
   a. La ultima parte de Modelado
   b. Los problemas de clase
2. LP -->
   a. El problema de degenerate que habia q sacar la ec de la recta
3. Non Linear y Mixed
4. Decision Theory
5. Game theory
6. Simulation
7. Que Theory

INTERS -->

18 y 17

EXAMENES PRUEBA 2 -->

18 y 17

1. **MODELADO**
    1. Definir como activar una nueva binaria doblemente
    2. cuidado con la algo media, q hay q dividirlo todo en un sumatorio
    3. si a esta en el equipo 1 b tiene q estar en el 2-->Xa,1 =Xb,2 Xa,2 =Xb,1
2. **LP**
    1. la funcion objetivo no se pinta | ACTIVE = BINDING , que es lo opuesto a relaxed | en stnd form todo con <=
    2. Las excess var se meten con un negativo
    3. el numero de constrain es el num de elementos que tiene que teener Xb
3. **NLP**
    1. Para ver si las necesary KKT coniditons son sufficiente, hay q ver si el problema es convex,viendo una por una si las constrains son convex
        a. si al menos una de las constrains no es convex, las KKT coditions son solo necesarias y no suficientes
    2. **Si la region es bounded, el global optimum debe existir por el teorema de Weristras,aunque alguna de las constrains no sea convex y por tanto las necesarias conditions no sean suficientes**
    3. Una vez q tienes los putos ves su objective fucntion para ver cual es el mejor
    4. las landas pueden valer lo q sea mientras que sean mayores que ceros, es decir positivas
    5. cuidado q las contrains tienes q ponerlas como <= todas, ojo
4. **MIP**
    1. si a medida que bajas por las ramas el valor objetivo aumetna es un problema de minimizacion
5. **Decision Theory**
    1. los cuadrados en el arblo son decisiones y los circulos son cosas q pasan creo
    2. Expected value or laplace criterion
6. **Game Theory**
    1. Dominanted strategy
        a. Eliminas en filas la mas grande y en col eliminas las mas peque
    2. Te pueden aplicar aquí el **dual problem** porque tu oponente su problema es el dual al tuyo
        a. Si calculas su Ya, siendo Y la var dual y la a que elija la opcion a y te da cero, evidentemente no la va a coger
        b. Si quieres plantear las ec del jugador de las filas, tienes q plantear las constrains por columnas, y si eliges el de las columnas las constrains las coges por filas
        c. el problema es que la constrain te queda siempre en funcion de v y te tienes q quitar esa v dividiento y eso ya da problemas
    3. Te pueden mezclar con **perfect info**
        a. coges los valores que te han dado el sistema de ecuaciones del mixed strategy y lo multiplicas por el valor mayor
    4. **Si es un zero sum game -->**en filas te quedas con el mas grande y en col con el mas peque(por tanto eliminas las filas mas peque y las columnas mas grande)-->**EN FILAS MAX Y EN COLUMNAS MINIMIZAS**
7. **Queue Theory**

DUDAS

MODELADO

a. D 18 en el problema de modelao en el aparatado c no hace falta meter 2 constrains mas???????
b. I15 como lo ha modelado???
c. I 18, problema del reality sow, ese Y+ y Y-????????

Linear Programing

a. NOV 19 pregunta 7, cual es la otra correcta?????????????
b. LP en la optimal la dual y la primal no coinciden excepto en QT por el equilibrio no????
    i. **La funcion objetivo si el problema es feasible vale lo mismo tanto la dual como la primal, los que no eran lo mismo pero estaban realicionados por una formula era la sol Xb y Yb**
c. Binding es lo contrario a relaxed no???????????? RRRRRRRRRRRRRR
d. D18 problema 2, en el c2. creo que el dual tienen 3 var basicas porque tambine incluye a las var artificailes de la dual
e. QT problemas de clase el de la matriz de sum zero game, ver en el d de donde sale esa v

Queue T

a. QT,problema 5, porque Sigma^2 es cero?-->
    i. **Porque dice que el modelo tiene una media casi exacta,por lo que no tiene varianza**
b. Que formula usa el Lq en el M/G/1

Decision T

a. NOV 18 en la 1 del test, el del wald criterium, se hace con porcentajes?????????
b. NOV18A del test de decision theory como se hace el el 4,el del oportunity cost
c. NOV 18 a en el test del 5 no seria 80-83
    i. **Correcto, pero como estas en costes y no en profit que te salga -3 siginfica que te cuesta menos, por tanoto en realidad te estas ahorrando 3 euros, sales ganando**

Game T

a. GT, ene el problema 1 el de los mobiles como se hace
b. NOV 18 A el 8, a3 no domina a nada no??????????????????????????
c. D16 p3 apartado 4, porque en col te quedas con el mayor y en filas con el menor??????

Simulation

a. D18 P4 TEST soluciones

Mixed Integer

a. cuando podas una rama?????????MMMMMMMMMMMMMMMMMMM

Non Linear

a. NOV 19 pregunta 14 porque es positivo si el deeterminante da negativo?????
    i. **esque el det no daba -2, sino que daba 0 y eso es semidefinida positiva**
b. con las kkt, el pto en q hesiano lo sustituyes, en el de la OF o en el lagrangiano MMMMMMM
c. Diferencias entre cond suficientes y necesarias
d. J17, porque en el b4 del problema 3 quieres el hesiano del lagrangiano?????????????

****

LP

- [x] Exa 2 de LP
- [ ] hacer los problemas de clase de LP

DT Y GT

- [x] repasar los problemas de dt y gt de nov 18
- [x] acabar problemas QT

SIMULATION

- [x] hacer el test de clase

MODELADO

- [ ] Problemas clase modelado
- [x] Problema modelado J15
- [ ] acabar I17

- [ ] repasar el modelado de D18

EXAMENES

- [ ] hacer J14

TEMA 1 Logical Preposition Integer(num natural) VS continuous(cualtivativa) VS binary

1. 1.Sets | parameters | variables | objetive function | constrains | entender enun | ver todas constrains | modelar
2. **Constrains** Pueden salir del enunciado directamente o relacionadas con activacion de var binaria-->*petrolero*
   1. A veces es para todo j y dentro del sumatorio solo cambia en i-->*production of 3 products*
   2. meter una constrain que sea que solo puedes elegir una celda de la matriz si se diese el caso
   3. una que conecte dos variables -->*staf selection, y juntarla con q no pueda tener valor max* | Yi >=hmin *Xi
   4. **Problemas de nodos** -->*la sequia-->las piscinas-->conectar 2 pipelines-->petrolero*
      i. demand= los que llegan-los q se van - excedentes | la falta de algo no pude superar a la demanda
      ii. plantear la constrain del abastecimeinto de los nodos | y de capacidad max del caudal
   5. **Grupos de personas** -->*cristmas dinner, inter 16,basketball player*
   6. relacionadas con la variables, por ejemplo que no superen un valor maximo(la max demanda)
   7. que no se puedan dar dos var binarias a la vez -->bases de datos lo utlimo
   8. a veces hay que poner 1 -binarya -->*swiming pools,el bodeguero2*
   9. **Dobles implicacion(viceversa)** -->*swimming pools,inter 17.1 b, inter 17.2 section b,inter 16 constrain 6*
      i. aplicas a cada par el not a -->b y obtienes 2, y luego con el otro par,al final tienes 4 constrains
      ii. metodo 2, haces sumatorio x-->sigma=1 /// sigma =1 -->sumatorio y
   10. que algo sea n veces mas grande | 3*x<y san pedro | TV digital sumatorio x <1/2*sumatorio y
   11. si un binaria te dice que en un num concreto se cambia de la var x a la y--->X<max*(1-z)-->*juntas apar b2*
   12. No te olvides de poner al final las natural constrains | Detectar el cambio de algo
   13. expresar lo del porcentaje de algo las separas, | (1-porcente) -->*la sequiade mo*
   14. Plantear la demanda | la disponibilidad de algo | la capacidad maxima de algo
   15. si x no puede estar en el mismo equipo q y o no pueden estar juntos | Ypepe +Ymanolo<=1
   16. **Constrains muy evidentes** 1 coche por pais-->1 uni por alumno-->1 forma de ensamblar una maquina
3. **Modelado**
   1. comprueba al final que el modelo que has puesto(la inecuacion) tiene sentido
   2. el viceversa en el enunciado siginfica que la flecha es de dos sentidos -->*inter 17.2 ,inter 17.1*
   3. Si la sigma es cero tienes dos opciones
      i. En la formula en vez de poner (1-sigma) pones sigma
      ii. Cambiar sentido inecuacion y volver la sigma 1 | A->B == notB --->not A inter 16,cena navidad
   4. la epsilon esa es un valor minusculo que ademas hay q dejarlo con epsilon sin asingarle valor,en bin vale 1
      i. Si todas las vari son ints, epsilon tambine vale 1 | M + epsilon = M
   5. **Disjunciones -->** A->B == B o not(A) = not A ó B --> junio 15 | **se pone sigma =1-->not(A)**
      i. En realidad ves como defines tu sigma y en funcion de eso haces q valga 1 o 0 pero siempre delante
      ii. que se den dos mayores y iguales a la vez sin sumatorios-->*bases de datos ,inter 17.2,inter 17.2*
      iii. sumatorios >a -->sumatorio >3 *inter 16* | sumatorio = 2-->x>1 bases de datos constrein ultima
      iv. si es para todo j debes meter la j a la delta que metas | obtienes 2 constrains de 1
   6. **las binarias** las tenemos a la izquierda para desactivar o activar la cond de la derecha
      i. en binarias auqnue no tienen mucho sentido se pone menor o igual q cero ,nunca pones igual a cero
      ii. convertir un "=" en un mayor o menor porque cuadra-->*bakery ($\sum x$ =2 -->z>1,se hace como siempre)*
      iii. la m de y1+y2+y3 >x1+x2+x3 es -3, sustituyes y por 0 y las x por 1-->*ultima const. cristmas diner*
   7. **valor absoluto**--> *inter 16*
   8. si tienes dos sumatiors unidos con la flecha tienes que inventarte una delta intermedia y modelar dos
      i. el del bakery apartado c | inter 18 reality show ultima constrain
4. **Parameter y variable :**si en un problema hay lack of algo mete tambien y excess de eso por si acaso()
   1. su diferenica es q uno sabes a priori su valor y otro no(la variable).La vari se pueden suamr, pero no multi
   2. una variable binra que te dice si algo va delante o detrás de algo -->task in 2 machines
5. **Objective function :** no puedes poner para todo j en la objective function, ya q solo puedes tener un valor
   1. siempre hay q poner los indices en el sumatorio abajo | no se pueden multiplicar binarias
   2. $\sum_{i,j} Xi + Yij = \sum_{i,j} Xi + \sum_{i,j} Yij$ | W (max delay), y poner que min W, y luego en cons W>Vj,siendo Vj delay of j
   3. Cuidado que cuando te meten mas constrains en el apartado b puede cambiar!!!!!!!!!!!

1. **MULTICRITERIA MODEL -->en los dos tienes la w, que te dice lo q te interesa cada uno**
    1. **Goal programing --->**el q tiene formula pequeña-->tienes desviacion positiva y negativa
    2. **Compromise programing-->**el q su formula son dos fracciones sumandas y minimizadas
2. **EJERCICIOS**
    1. "los q estan en rojo estan hechos en clase, lo fosforito son constrin y lo verde son q tipo de problema es"
    2. production of 3 product no tiene tipo
    3. charter flight no tiene tipo
        i. la clave era incluir en los parametros grupos de aviones, en vez de hacerlo de avion en avion,tendras Pij, q sera la capidad total de usar j aviones de tipo i
        ii. tienes una constrain que te obliga a solo elegir un numero de aviones de un tipo concreto,logico
    4. paper roll
        i. en la constrain 4 hay un tipo 6 no binario | cada parte de un i puede salir de distintios rollos j maestros
        ii. Xi,j numero de rolles de tipo i asignados al master roll j | apartado d???
    5. bakery contrain tipo6 en c -->
    6. basketball player problema de gente en la constrain 5 hay un tipo 5 | player 8 o player 9 -->X9+X8 =1
    7. job shop : constrain 3 de tipo 1. Se podia hacer fusionando dos sets
    8. car renting constrain evidente de 1 coche por pais | detect car change in a contry tipo 6
    9. connection 2 pipelines tener 3 set del mismo vector i,j,k | problema de nodos
    10. swimming pools problema de nodos poner i,j para el mismo vec | constrain
    11. sequia problema de nodos
    12. aerogeneradores problema de nodos
    13. task in 2 machines tipo 1 binario y tipo 6 binario
        i. min el max delay.Tenias una var con 3 parametros.Te creabas j y k referidos al mismo vec
        ii. Var donde se iba acumulando el tiempo
    14. 1. exams in 7 days m = -p | ord(j) para referirse q es la posicion
    15. database no te dice el enunciado el tiempo de consulta pero debes meterlo como un parametro
    16. christmas dinner --> problema de gente pasar de binario = 0 a binario =1 cambiando el otro lado
        i. tenias una mega condicion con muchas or y muchas opciones
            1) te inventabas var auxialres a,b,c,d… y lo ponias todo relacionado
    17. juntas problema de tablas | inventario medio es inventario al principio del mes + el del mes siguiente entre 2
    18. TV Digital--> problema de tablas que es lo de la logica de indicador de fabricacion???
        i. El porcentaje que te queda es (1-ni) | sin modelado
    19. el bodeguero problema de tablas el porcentaje | sin modelado
        i. los beneficios los plantea raros | planteas como var el num de listros de uva destinados al prod i(no lo dice)
    20. el petrolero problema de tablas | constrain tipo 4 en la c3
        i. no es lo mismo el crudo transportado que el crudo vendido
        ii. como el petrolero puede partir con menos petroleo que el max, el total vendido es igual al que sale de Dubai
    21. san pedro problema de tablas
        i. si tu tienes que hacer 3 veces mas x que y--> 3*y <= X | 1 gracia = -1 malevolo
        ii. tanto las acciones malas como las buenas habia que meterlas en el mismo set
        iii. uff la funcion objetivo del apartado 2 para lo de ir al purgatorio

Tabla resumen

Proposition	Modeling
$\delta = 1 \rightarrow \sum_j a_j x_j \leq b$	$\sum_j a_j x_j \leq b + M(1-\delta)$
$\sum_j a_j x_j \leq b \rightarrow \delta = 1$	$\sum_j a_j x_j \geq b + \varepsilon + (m-\varepsilon)\delta$
$\delta = 1 \rightarrow \sum_j a_j x_j \geq b$	$\sum_j a_j x_j \geq b + m(1-\delta)$
$\sum_j a_j x_j \geq b \rightarrow \delta = 1$	$\sum_j a_j x_j \leq b - \varepsilon + (M+\varepsilon)\delta$
$\delta = 1 \rightarrow \sum_j a_j x_j = b$	$\sum_j a_j x_j \geq b + m(1-\delta)$ $\sum_j a_j x_j \leq b + M(1-\delta)$
$\sum_j a_j x_j = b \rightarrow \delta = 1$	$\sum_j a_j x_j \geq b + \varepsilon + (m-\varepsilon)\delta'$ $\sum_j a_j x_j \leq b - \varepsilon + (M+\varepsilon)\delta''$ $\delta' + \delta'' - 1 \leq \delta$

Proposition	Modeling
$\delta = 1 \leftrightarrow \sum_j a_j x_j \leq b$	$\sum_j a_j x_j \leq b + M(1-\delta)$ $\sum_j a_j x_j \geq b + \varepsilon + (m-\varepsilon)\delta$
$\delta = 1 \leftrightarrow \sum_j a_j x_j \geq b$	$\sum_j a_j x_j \geq b + m(1-\delta)$ $\sum_j a_j x_j \leq b - \varepsilon + (M+\varepsilon)\delta$
$\delta = 1 \leftrightarrow \sum_j a_j x_j = b$	$\sum_j a_j x_j \geq b + m(1-\delta)$ $\sum_j a_j x_j \leq b + M(1-\delta)$ $\sum_j a_j x_j \geq b + \varepsilon + (m-\varepsilon)\delta'$ $\sum_j a_j x_j \leq b - \varepsilon + (M+\varepsilon)\delta''$ $\delta' + \delta'' - 1 \leq \delta$

TEMA 2 Simplex method Proceso iterativo

- **xb=B^-1*b** = b gorro; z= $c_B^T x_B$ | $\hat{c}_N^T = c_N^T - c_B^T B^{-1} N$ | los c son los coef q multiplican a la x de la OF
- calcular los reduced cost cnT | -->si todos son 0 la sol es optima.Si no eliges la que es negativa
- Calcu Y =(B^-1*N)(te quedas con la col que corresponde con la var EVB q has elegido) | luego lo del **min(b gorro(ó aveces Xb)/y) de los positivos**, la mas peque es la LBV .Pero coges la Xb q tienes en ese momento
- Pueses coger la col de EVB de N, multiplicarlo por B^-1 y eso es lo que usas para dividir entre Xb
  - Un un coceinte es 0/5 te quedas con ese Si un cociente te da 5/0 no lo eligas evidentemente,tampoco si es -5/0.Coges solo los positivos **SI TODOS SON (-) EL PROBLEMA ES UNBOUDED**
- Meter la LBV en Xn y esa en Xb | actualizar B poniendo en la col de la LBV la aj | Actualizas la N poniendo en la col q has movido de la B

1. **Two phase metodo para minimiar var artificiales.** Queremos min las var artificiales.
2. **Estándar form-->**lo max EN en min /<= transform en = metiendo una h /pones como posit las var negativas
3. **Tipos de var:**
   - **excess e** cuando tienes => | **slack h,** cuando tienes <= | **EBV** = entering basic var | **LBV** =leaving basic var
   - **Free var:** que puede ser negativa o positvia | **Basic** => var q puede coger valor 0 | **non basic** => set to 0
4. **Matrices**
   1. **La Y** -->cada fila de Y son los elementos de Xb y cada columna son los Xn
      a. Cuando calculas la LVB la columna que necesitas de la Y es la de Xn que corresponde con la EVB, y luego lo divides entre tu actual Xb(si es la primera es la c)
   2. **La B** La B por columnas tiene a Xb y por filas tiene las constrains | **La N** por columnas tiene a Xn
   3. **La X non basic**-->si te introducen una nueva var se mete aquí y no en Xb
   4. **Reduced cost** -->son los Cn con gorro
      a. un cero,el resto negativos y estas maximizando significa que vas a elegir la del cero como EVB
      b. si tienes 2 (-) coge el primero empezando por la izquierda
      c. Si todos son (+)y estas min : sol optima es unica-->si tienes un 0=0 puedes tener multiple optimal sol
      d. OJO que pueden no coincidir en tamaño con el num de var duales,casi nunca.Si la dual esta asociada con una bascia en el prima su valor es cero
   5. **en c** van los coeficientes de la funcion objetivo,tmbn se le llama gradiente de z
   6. **en el Cn t** aparecen los elementos de Xn
   7. **UNA SOL INFEASIBLE ES AQUELLA QUE VIOLA ALGUNA DE LAS CONSTRAINS O NO ESTAN DENTRO DE LA REGION FEASIBLE**
5. **Tipos de constrains** -->x>y>0 -->x-y -->**BINDING == ACTIVE Es binding la constrain q corta a la sol**
   1. **binding/satisfied or meat con equality/active**-->Si valorandola en el optimal point el valor izquierdo es igual al derecho-->si la var slack o excess asociada a esa constrain es cero, es decir es Xn
   2. **Una constrain relaxed**-->si la e o h var asociada a esa constrain,distinta de 0(si esa e o h estan en las XB)
6. **Tipos de soluciones de Xb -->TIENES TANTAS Xb como CONSTRAINS TENGAS!!!!!!!!!!!!!!!!!!!!!!!!!!**
   1. **Problem not bouded or unbunded** Si encuentras otra optimal solution el problema,es cuando aplicas el proceso iterativo y te da todo negativo y no puedes continuar
   2. Si tiene algo que es cero --> degenerate,pero lo miras en Xb
   3. **is unique** -->si todos sus reduced cost son negativos | **multiple o degenerate** --> algun reduced cost
   4. si tiene alguna (-)->not feasible basic sol(solo esa) | si todas son + es feasible,hay q comprobar reduc.cost
   5. **Is degenarte** --> si uno de los elementos de la feasible basci solution es 0 y 3 o mas constrains cortan en el mismo pto. Es super degen. si sol. es todo 0 **degenerate = MULTIPLE OPTIMAL**
7. **Si estas minimizando** tu EVB sera la mas negativa del conjunto de reduced cost.La que eligen min es la LVB
   1. Llegas a la optima cuando los reduced cost son todos positivos | Ente un cero y un neg coges el negativo
8. **En maximizando**Llegas a la optima cuando reduced cost son todos negativos | Tu EVB es la positiva de Cn o la cero
9. **Poner como una expresion todas las infinitas sol**-->problema 22 con alfa*Sol1+(1-alfa)*Sol2
   1. En la Sol1 y 2 como no pueden estar las var slack o excces,solo x1,x2 y x3,las del problema

1. **Resolverlo graficamente**
   a. Pintas rectas de las constrain pero igualadas a su valor y luego dices si a un lado u otro si es mayor/menor
   b. Pintas la recta de la fun objetivo q pase por el pto de corte e igualada un valor inventado, y su gradiente =c
   c. Cuanta mas negativa la pendiente mas inclinada/empinada
   d. Ves los ptos q podrian ser la sol, ves su valor de z y ves cual es el optimo
   e. la funcion objetivo no se pinta | hay que ver el gradiente, q es el vector c(los coef de la OF)

2. **Entender la tabla rara**
   En la primera fila sabes los reduced cost y los q tienen un cero son las basic(q ademas coinciden con la primera col de la fila 2 y 3)
   De abajo sacas la matriz B creo y la ultima col es la Xb

Transformation into Standard Form

- Objective function: $\max z \rightarrow \min -z$
- Constraints: first, check the sign of the independent term b_j
  - If $b_j<0$ → Change the sign of the whole constraint
- Constraints ≤: A slack variable h_j is introduced:
  $\sum_i a_{ij} \cdot x_i \le b_j \quad \sum_i a_{ij} \cdot x_i + h_j = b_j \quad b_j \ge 0, h_j \ge 0$
- Constraints ≥: An excess variable e_j is introduced:
  $\sum_i a_{ij} \cdot x_i \ge b_j \quad \sum_i a_{ij} \cdot x_i - e_j = b_j \quad b_j \ge 0, e_j \ge 0$
- Negative variables: $x_j \le 0 \rightarrow x_j = -y_j \rightarrow y_j \ge 0$
- Negative bounded variables: $L_j \le x_j \le 0 \rightarrow x_j = -y_j \rightarrow 0 \le y_j \le -L_j$
- Free variables: x_j free $\rightarrow x_j = x_j^+ - x_j^- \quad x_j^+, x_j^- \ge 0$

3.

min		max
Variable		Constraint
≥0	⟺	≤
≤0	⟺	≥
Free	⟺	=
Constraint		Variable
≥	⟺	≥0
≤	⟺	≤0
=	⟺	Free

PRIMAL (m x n)	DUAL (n x m)
$\max_x z = c^T x$ $Ax \le b$ $x \ge 0$	$\min_y y_0 = b^T y$ $A^T y \ge c$ $y \ge 0$
$\max_{x_j} z = \sum_{j=1}^{n} c_j x_j$ $\sum_{j=1}^{n} a_{ij} x_j \le b_i \quad i = 1,\dots,m$ $x_j \ge 0 \quad j = 1,\dots,n$	$\min_{y_i} y_0 = \sum_{i=1}^{m} b_i y_i$ $\sum_{i=1}^{m} a_{ij} y_i \ge c_j \quad j = 1,\dots,n$ $y_i \ge 0 \quad i = 1,\dots,m$

4. **Duality**
   a. **La dual var asociada a la constrain x es el valor del reduced cost de la surplus/excces var de la cons x**
   b. Si una constrain (no es active o no es binding) en la optimal solution el reduced cost de la var slack asociada a esa constrain es cero y por tanto la dual variable asociada vale cero
   c. c traspuesta * x = b transpose *y | **una constrain BINDING == ACTIVE**
   d. **degenerate = MULTIPLE OPTIMAL** | Por cada constrain tienes una var dual
   e. **Valor de las dual var**
      a. si esa var asociada a una slack/e variables es basic, su valor es cero
      b. si esa var asociada a una slack/excess variable es non basic su valor es lo que valga en los reduced cost
   f.

Si una constrain en el primal no es binding	en el dual esa var asociada vale cero
Si una constrain en el prima es binding	en el dual esa var asociada tiene que valer distinto de 0
• If solution has a h or e var > 0(estan en XB) in constrain	the dual variable asociada a esa slack =0
• If an original variable of the primal (not artificial) in the optimal solution has a value > 0, es XB	constraint of the dual has h/e var = 0, es decir que estan en Xn

   g. Si el primal es unbonded el otro es infeasible.Pero si primal es infeasible el otro puede ser nfeasible o unbound
   h. **• If a POS(primal optimum solution) has a h or e var = 0,esas variables son non Basic es active(es igual a cero)**, the dual variable to this constraint may have a value ≠ 0 (either + or - depending on the type of constraint and OF –maximization or minimization–) in the optimal solution of the dual problem.
   i. lo de aumentar en 1 unidad --> se hace por duality, viendo su var dua

TEMA 3 y 4 MIP y NOP

Mixed Integer Programming ARBOL

- Una sol es infeasible si los valores a los q ha llegado incumplen alguna constrain
- Hay q ver el enunciado que var te dice que deben ser num enteroso porque puede ser que no sea todas
- OJO cuando separas en la rama q tienes en un lado x<=5 y en el otro x>=6
- Se va alternando, primero se divide en funcion de x(cuando haces x<5 o x>6), luego en y,luego otra vez en x….
- Cuandos tienes 2 nodos, y uno de ellos tiene una mejor solucion q el otro cortas el otro y te quedas con el mejor
- Hay veces que puedes llegar a un nodo enfeasible, q es aquel q no cumple las constrains o alguna
- Por fuerza bruta si tardas 1 ms en calcular una sol,no tardas mucho en calcularas todas,unos 60 ms
- Una vez q llegas a todas la var que te dicen q sean integer sean asi, paras y ves entre las finale cual tiene mejor
- si a medida que bajas por las ramas el valor objetivo aumetna es un problema de minimizacion

TEMA 4 Non linnear programming

- **MATRICES el hesiano =[fxx fxy; fyx fyy]**
  - **Semidefinida positiva/negativa** -->si el det de todos su menores son mayores o iguales q cero(tmbn sus autovalores)
  - **Positiva/negativa**-->todos los det de sus menores son mayores q cero(positivos)/menores que cero
- **HAY 2 TIPOS PROBLEMAS,CON CONSTRAINS(las kkt conditions y sistema) Y SIN CONSTRAINS**
  - **sin constrain** Si gradiente = 0 y el hessiano no es ni semidefinite posit. ni neg. no es ni un local min ni max
- **LAGRANGIANO**
  - **Establish KKT necesary condition -->tienes tantas landas como contrains tenga el problema**
    - Tienes q plantear lagrangiano multiplicadon cada constrain por una landa y ir viedno cada una de las landas =0 o 1 -->**tambien se llaman las Complementary Slackness Condition**
    - L(var1, var2, landa1,landa2,…) = OF + landa1*constrain1+landa2*constrain2 +…
      - Derivada de L con respecto a var n =0 | Deriva. L con respecto a landa n =0 | de ahí sacas mas cond
      - Todos los landas deben ser positivos-->no puede haber landas negativos-->vas hayando soluciones y landas asociadas a esas soluciones
  - Empiezas eligiendo una landa.El caso 1 sera landa1=0 y el otro landa1=1.El resto son solo con landas
  - cuidado q las contrains tienes q ponerlas como <= todas, ojo
  - Para ver si las necesary KKT coniditons son sufficiente, hay q ver si el problema es convex,viendo una por una si las constrains son convex Y LA OF,pero las constrains originales del problema
  - las landas pueden valer lo q sea mientras que sean mayores que ceros, es decir positivas
  - Una vez q tienes los putos ves su objective fucntion para ver cual es el mejor
  - si al menos una de las constrains no es convex, las KKT coditions son solo necesarias y no suficientes
- **EL PUNTO** el OF y las constrain deben ser concave o convex en ese pto , hay q comprobarlo con el hesiano
  - El putno es feasible si satisface constrains y condiciones KKT | Hay q ver si es un min o un stasionary point
- **TEOREMA DE WERISTRA**
  - **Si la region es bounded, el global optimum debe existir por el teorema de Weristras,aunque alguna de las constrains no sea convex y por tanto las necesarias conditions no sean suficientes,hay max y min**
- **Tipos de minimos**
  - **Local minimum**-->es minimo pero hay muchos como el(linea horizontal) | si gradiente de f en pto es 0
  - **Strict local minium**-->es un minimo q esta solo
    - si el gradiente de f en ese pto es cero y el hessiano en ese punto es una positive definite
  - **stric local global minimum** -->hay mas minimos pero el esta solo y es el mayor minimo
    - si f es convex y difeneciable en un pto y f en ese pto es un local min tenemos un global min
  - **strict global minimum**-->el unico minimo

Convex Function	Strictly Convex Function	Concave Function	Strictly Concave Function
Positive semidefinite Hessian	Positive definite Hessian matrix	Negative semidefinite Hessian	Negative definite Hessian matrix
Positive curvature (2nd derivative)		Negative curvature (2nd derivative)	

Local minimum + convexity of the entire region	Local maximum + concavity of the entire region
Global Minimum	Global Maximum

Positive semidefinite in the point (negative)	Positive definite in the point (negative)
Local minimum (maximum)	Strict local minimum (maximum)

TEMA 5 Y 6 Decision/Game th

Decision theory

1. **Type of criteria**
   a. **Known probabilities-->Decision under risk**
      i. Expected value or laplace criterion
      ii. using de mode
      iii. mean scenario
   b. **Unknown probabilities-->decision under uncertain**
      i. Wald criterion --> te quedas con el mayor de los peores
      ii. Optimistic criterion-->
      iii. Hurwicz criterion
      iv. Savage criterion or opportunity costs
         1) Penalty matriz
2. **Arboles**
   a. **puedes talar ramas q no tengan probabilidad!!!!!!!!!!!!!!!**
   b. Cuidado que si estas en costes el mejor es el mas bajo y en profit el mejor es el mas alto
3. **Perfect information**
   a. EVPI =expected profit with perfect info- expected profit with uncertain = EPPI -EPU
   b. EPPI for each state consider the best decision.Then calculate its expected value(average weighted by probabili)
      i. se calcula sumando los productos de cada max de cada col por la probabilida
   c. Al final tienes que ver si el valor de EVPI es mayor que lo q te cuesta tener perfect info, porque si es mas bajo no te renta

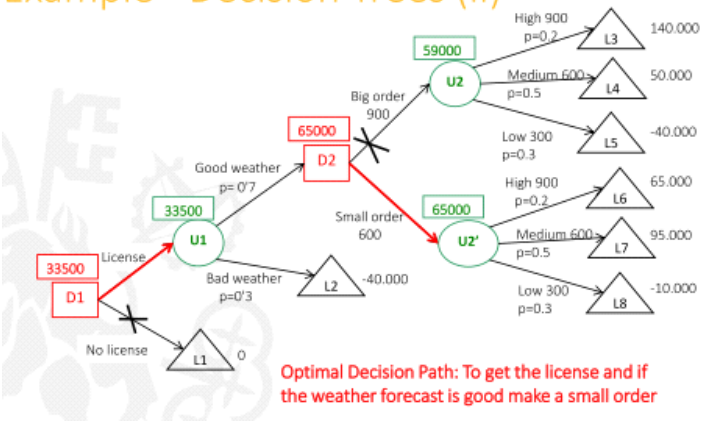
Game theory

- **dominated strategy**
  - Eliminas en filas la mas grande y en col eliminas las mas peque
- **pure strategy**
  Despues de eliminar las estrategies que son dominates aplicas el pesimistic criterion,quedandote con el mejor de la filas y el menor de las col y ver si coincide,si no no tiene pure strategy
- **mixed strategy** -->
  - lo calculas cuanod no hay pure strategy, tienes que plantear ecuaciones y luego aplicar cosas **del mixed strategy**
  - En filas te quedas con el mas alto y en columnas el mejor es el mas bajo
  - Te pueden aplicar aquí el **dual problem** porque tu oponente su problema es el dual al tuyo
    - Si calculas su Ya, siendo Y la var dual y la a que elija la opcion a y te da cero, evidentemente no la va a coger
    i. Si quieres plantear las ec del jugador de las filas, tienes q plantear las constrains por columnas, y si eliges el de las columnas las constrains las coges por filas
    ii. el problema es que la constrain te queda siempre en funcion de v y te tienes q quitar esa v dividiento y eso ya da problemas

  - Te pueden aplicar aquí el dual problem porque tu oponente su problema es el dual al tuyo
    i. Si es un zero sum game -->en filas te quedas con el mas grande y en col con el mas peque(por tanto eliminas las filas mas peque y las columnas mas grande)-->EN FILAS MAX Y EN COLUMNAS MINIMIZAS

Decision tree

Example - Decision Trees (ii)



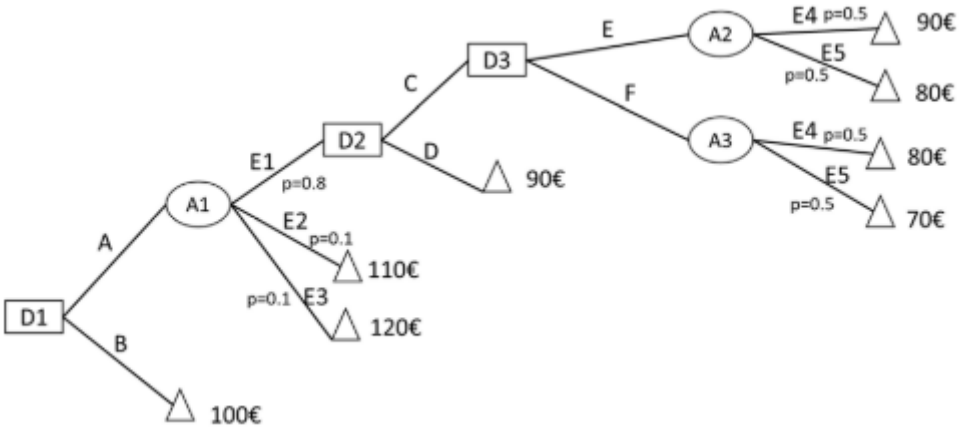
Optimal Decision Path: To get the license and if the weather forecast is good make a small order

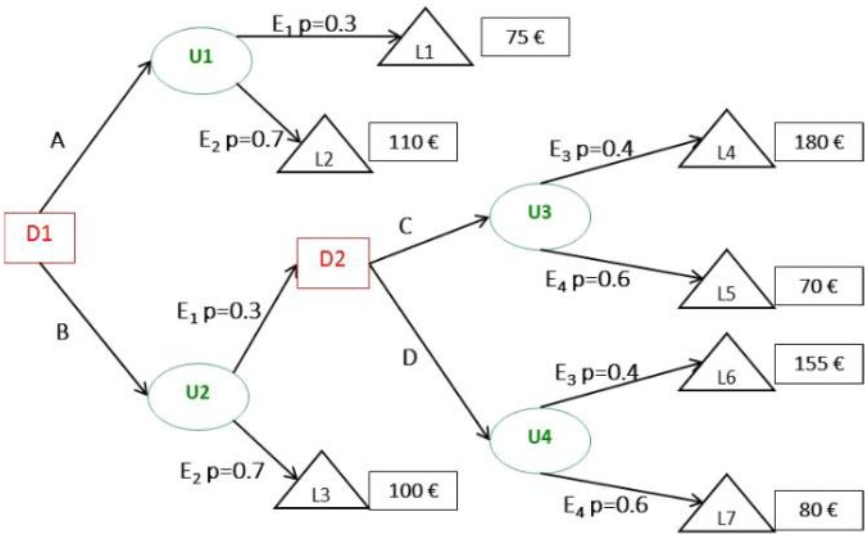
Example - Decision Trees (iv)

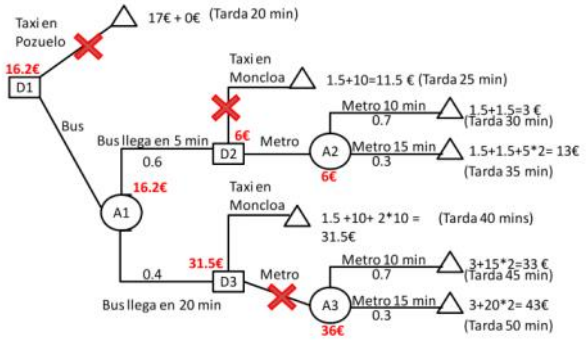
- Expected Value of Perfect Information

Weather Forecast	Demand	Scenario Probability	Optimal Decision	Profit	Weighted Profit
Good Weather	High	0.7×0.3	License and Big Order	140,000	29,400
Good Weather	Medium	0.7×0.5	License and Small Order	95,000	33,250
Good Weather	Low	0.7×0.2	No license	0	0
Bad Weather	-	0.3	No license	0	0
Total		1			62,650

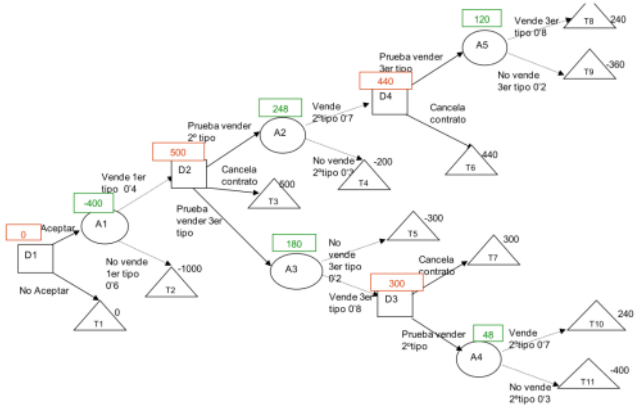
EVPI = EPPI – EPU = 62,650 – 33,500 = 29,150 €







Escenarios		Probabilidad	Decisión óptima	Coste
Bus 5'	Metro 10'	0.6*0.7=0.42	Bus + Metro	3€
Bus 5'	Metro 15'	0.6*0.3=0.18	Bus + Taxi (desde Moncloa)	11.5€
Bus 20'	Metro 10'	0.4*0.7=0.28	Taxi (desde Pozuelo)	17€
Bus 20'	Metro 15'	0.4*0.3=0.12	Taxi (desde Pozuelo)	17€



Infor.	Benef.	Probabilidad	Decisión	Benef. esperado
1	0	0.6	No contratar	0
2	500	0.4·(1-0.7) (1-0.8)	Vender tipo 1 y cancelar contrato	12
3	500	0.4·0.7·0.8	Vender tipo 1 y cancelar contrato	112
4	500	0.4·0.7·(1-0.8)	Vender tipo 1 y cancelar contrato	28
5	500	0.4·(1-0.7) 0.8	Vender tipo 1 y cancelar contrato	48
total	2000	1	1	200

TEMA 8 SIMULATION

1. GRAFICAS

a. Si dos histogramas ves que tienen como una cola que va bajadno hacia la derecha es debido a la variabilidad-->FALSO

2. INFINITE HORIZON

a. Si es un año es infinito

b. WARM UP PERIOD

i. Es un metodo visual
ii. Su desventaja es que requiere mucha data storage y un aumento en el tiempo computacional para obtener la misma info usable que si tu no usases el warm up
iii. cuando el clock de sumulacion es igual a la longitud del warm up period todas las estadsitivcas de ese perodo son borradas,pero las entitys no son eliminidas,y el run length de la simulacion debe de ser 10 veces mayor que el warm up period
iv. **El stady state** -->cuando la grafica se hace estable, una linea rectas
v. Cuando tienes varias graficas en la misma, te quedas con el warm up period mayor
vi. Las condicione iniciales no se tienen en cuenta, porque estan por detrás del warmp up
vii. Sirve para mitigar el "inicialization bias problem"

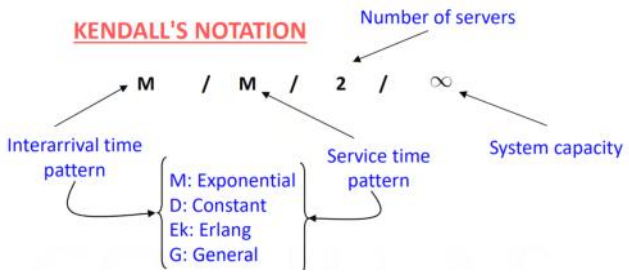
3. FINITE HORIZON

a. Si es un horario concreto todos los dias con varias replicaciones de un dia de longitud te vale

TEMA 9 Queing theory

Elements of wating time

Population:	Queue:	Priority rules:	Service Center :	Client actions:



λ Mean arrival rate (average number of customers arriving per unit of time)
μ Mean service rate (average number of customers that can be served per u
t Time
c Number of servers in the system
k System capacity
$1/\lambda$ Mean time between consecutive arrivals
$1/\mu$ Mean service time
ρ Utilization ratio ($\rho = \lambda/c\mu$)

n	system state, clients in the System (waiting or being served)	
P_n	probability of n customers in the System at any given time	
L	average number of customers in the System	$L = E[n]$
n_q	queue length, number of customers waiting in line	
L_q	average number of clients waiting in line	$L_q = E[n_q]$
t	total time spent in the System, including service time	
W	average total time spent in the System	$W = E[t]$
t_q	waiting time in the queue	
W_q	average waiting time in the queue	$W_q = E[t_q]$

- **Performance of waiting line**
  - The steady state of a Queueing System is reached when the number of customers probability distribution is the same over the time
- "q" es el numero de linea del queue
- "n" es el numeor del customer
- si el arraival time es poisson el interarraival time es exponential
- "idle" es ocioso

Poisson Process

- **QT assumes that arrival process follows a Poisson process**
- **6 propiedades**

Birth death process

1. **"Most of queueing models emulate customer arrivals like births and customer departures like deaths"**
2. **4 caracterisitcas**
3. **Diagrama de transicion-->matriz de transicion**
4. **GENERAL PERFORMANCE MEASURES-->las formulitas**

Standard Models

Infinite customer population

1. **M/M/1:** Single server with Exponential interarrival and service times For example: a unique entrance parking
2. **M/M/1/k:** Same as previous with a system capacity For example: a truck scale
3. **M/G/1:** Single server with a general service time distribution For example: a car washing machine
4. **M/M/c:** multiserver For example: a petrol station.-->no sigue el birht dead process
5. **M/M/c/k:** multiserver with finite system capacity For example: a call center with a limited waiting line

Finite customer population-->tmbn tiene un stady state

1. **M/M/1:** For example: the fax with a reduced office staff
2. **M/M/c:** For example: maintenance personnel for finite number of machine

Formulas Standard Models
Infinite customer population

1. **M/M/1:**
   a. $$C_n = \rho^n \Rightarrow p_n = \rho^n p_0$$
   $$\sum_{n=0}^{\infty} p_n = p_0 \sum_{n=0}^{\infty} \rho^n = p_0 \frac{1}{1-\rho} = 1 \Rightarrow p_0 = 1-\rho$$
   $$p_n = (1-\rho)\rho^n \qquad \rho = \frac{\lambda}{\mu}$$

   $$L = \sum_{n=0}^{\infty} n\, p_n = \frac{\rho}{1-\rho} = \frac{\lambda}{\mu-\lambda}$$ Average number of customers in the System

   $$L_q = \sum_{n=2}^{\infty} (n-1)\, p_n = \frac{\rho^2}{1-\rho} = \rho L$$ Average number of customers waiting in line

   $$W = \frac{L}{\lambda} = \frac{1}{\mu(1-\rho)}$$ Average total time spent in the System

   $$W_q = W - \frac{1}{\mu} = \frac{\rho}{\mu(1-\rho)}$$ Average waiting time in line

   $$\bar{c} = L - L_q = \rho = 1 - p_0$$ Average server utilization

2. **M/M/1/k:** Same as previous with a system capacity For example: a truck scale

   $$\lambda_n = \begin{cases} \lambda & n < k \\ 0 & n \geq k \end{cases} \qquad \mu_n = \mu \quad \forall n \qquad \rho = \frac{\lambda}{\mu} \qquad p_n = \begin{cases} \rho^n p_0 & n \leq k \\ 0 & n > k \end{cases}$$

   ↳ Steady state probabilities:

   a. $$(\rho \neq 1) \quad p_0 = \frac{1-\rho}{1-\rho^{k+1}}, \quad p_n = \begin{cases} \rho^n p_0 & n \leq k \\ 0 & n > k \end{cases}$$
   $$(\rho = 1) \quad p_n = \frac{1}{k+1} \quad n = 0,1,\ldots,k$$

   ↳ Customer **effective** entrance rate: $$\lambda_{EF} = \lambda(1-p_k)$$

   ↳ Customer **loss** rate: $$\lambda_{LOSS} = \lambda p_k$$

   $$L = \sum_{n=1}^{k} n\rho^n p_0$$ Average number of customers in the System

   $$L_q = L - (1-p_0)$$ Average number of customers waiting in line

   $$W = \frac{L}{\lambda_{EF}}$$ Average total time spent in the System

   $$W_q = W - \frac{1}{\mu}$$ Average waiting time in line

3. **M/G/1:** Single server with a general service time distribution For example: a car washing machine

   and variance are:

   $$E[S] = \frac{1}{\mu} \qquad V[S] = \sigma^2$$

   a. **Pollaczek-Khintchine Formula:**

   $$L = \rho + \frac{\rho^2 + \lambda^2\sigma^2}{2(1-\rho)} \qquad \rho = \frac{\lambda}{\mu}$$

Formulas

1. **M/M/c:** multiserver For example: a petrol station.-->no sigue el birht dead process

$$\rho = \frac{\lambda}{c\mu}$$

ɔabilities after setting all derivatives to zero :

$$p_0 = \frac{1}{\frac{(c\rho)^c}{c!(1-\rho)} + \sum_{n=0}^{c-1}\frac{(c\rho)^n}{n!}}$$

$$p_n = \begin{cases} \frac{1}{n!}\left(\frac{\lambda}{\mu}\right)^n p_0 = \frac{c\rho}{n} p_{n-1} & 1 \le n \le c \\ \frac{1}{c!\,c^{n-c}}\left(\frac{\lambda}{\mu}\right)^n p_0 = \rho p_{n-1} & n \ge c \end{cases}$$

$$L = \sum_{n=0}^{\infty} n p_n = \frac{(c\rho)^c \rho}{c!(1-\rho)^2} p_0 + c\rho$$

$$L_q = L - c\rho = \frac{(c\rho)^c \rho}{c!(1-\rho)^2} p_0$$

M/M/c/k: multiserver with finite system capacity For example: a call center with a limited waiting line

$$\lambda_n = \begin{cases} \lambda & n < k \\ 0 & n \ge k \end{cases} \qquad \mu_n = \begin{cases} n\mu & n < c \\ c\mu & n \ge c \end{cases}$$

$$\rho = \frac{\lambda}{c\mu}$$

$$(\rho = 1) \quad L_q = \frac{(c\rho)^c (k-c)(k-c+1)}{2c!} p_0$$

Steady state probabilities:

$$(\rho \ne 1) \quad p_n = \begin{cases} \frac{(\lambda/\mu)^n}{n!} p_0 & 1 \le n \le c \\ \frac{(\lambda/\mu)^n}{c!\,c^{n-c}} p_0 & c \le n \le k \end{cases}$$

$$\sum_{n=0}^{k} p_n = 1 \Rightarrow p_0$$

$$(\rho \ne 1) \quad L_q = p_0 \frac{(c\rho)^c \rho}{c!(1-\rho)^2}\left[1 - \rho^{k-c+1} - (k-c+1)(1-\rho)\rho^{k-c}\right]$$

Customer **effective** entrance rate:

$$\lambda_{EF} = \lambda(1 - p_k)$$

Customer **loss** rate:

$$\lambda_{LOSS} = \lambda p_k$$

Average service rate:

$$\bar{\mu} = \sum_{n=1}^{c} n\mu p_n + \sum_{n=c+1}^{k} c\mu p_n$$

Finite customer population (CLOSE SYSTEM)

1. **M/M/1:** For example: the fax with a reduced office staff

$$p_0 = \left(1 + \sum_{n=1}^{m} \frac{m!\rho^n}{(m-n)!}\right)^{-1}$$

$$\rho = \frac{\lambda}{\mu}$$

Probability of not having customer

$$p_n = \frac{m!}{(m-n)!}\rho^n p_0 = (m-n+1)\rho p_{n-1} \quad 0 < n \le m$$
$$p_n = 0 \quad n > m$$

Steady State Probabili

2. $$L = \sum_{n=1}^{m} n\, p_n = m - \frac{1-p_0}{\rho}$$

$$L_q = \sum_{n=2}^{m} (n-1)\, p_n = m - \frac{1+\rho}{\rho}(1-p_0)$$

$$\lambda_{EF} = (m - L)\lambda$$

$$W_q = \frac{L_q}{(m-L)\lambda} = \frac{1}{\mu}\left[\frac{m}{1-p_0} - \frac{1+\rho}{\rho}\right]$$

$$W = \frac{L}{\lambda_{EF}}$$

3. **M/M/c:** For example: maintenance personnel for finite number of machine

$$\lambda_n = \begin{cases} (m-n)\lambda & 0 \le n \le m \\ 0 & n > m \end{cases}$$

$$\mu_n = \begin{cases} n\mu & 0 \le n \le c \\ c\mu & c \le n \le m \\ 0 & n > m \end{cases}$$

$$p_n = \begin{cases} \binom{m}{n}\left(\frac{\lambda}{\mu}\right)^n p_0 & 1 \le n \le c \\ \binom{m}{n}\frac{n!(\lambda/\mu)^n}{c!\,c^{n-c}} p_0 & c \le n \le m \end{cases} \Rightarrow \sum_{n=0}^{m} p_n = 1 \Rightarrow p_0 \Rightarrow \begin{cases} L \\ L_q \\ W \\ W_q \end{cases}$$

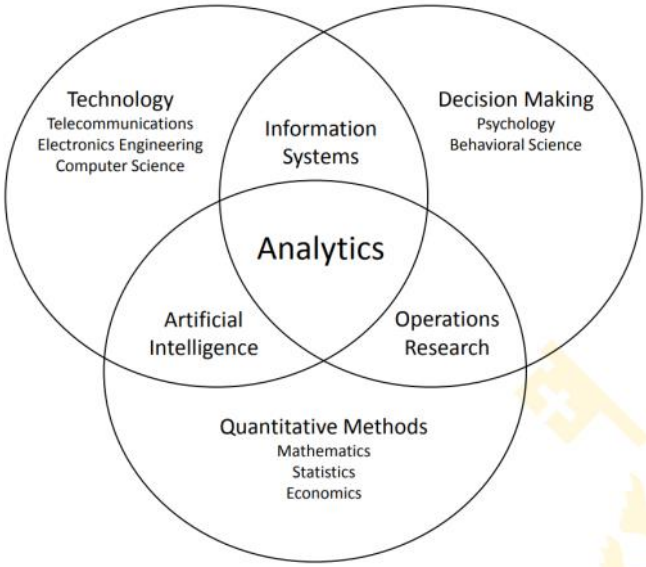
TEMA 1 Introduction to Operations Research

Introducction

Operation Research-Definition

• The science of better	• Decision support models	• Advanced analytical methods

• Application of **advanced scientific analytical methods** in improving the effectiveness in operations, decisions and management of a company.
  • Design and improvement in operations and decisions
  • Problem solving and support in management, planning or forecasting functions
  • Provide knowledge and decision support
• **Tasks**:
  • Collect and analyze data.
  • Develop and test mathematical models
  • Interpret the information
  • Propose solutions and recommendations
  • Help to implement improvement actions
• **Results**:
  • Computer appli, systems, services or products.



Introduccion to Analitical techniques

• We will present mathematical tools that can be used for supporting decisions in different frameworks

• Operations management	• Financial management

  • Marketing management
• How these techniq can be applied to business decisions
  • Inventory management and control

Staff planning	Fuel logistics	Maintenance scheduling	• Forecasting electricity demand, prices, wind

• Short and medium term operations	• Market bidding	• Market equilibrium

Optimization

What is?

• To find the value of the variables that make the objective function optimal while satisfying the set of constraints.
• Components of an optimization problem:
  • Objective function: Quantitative performance function (fitness) of a system we want to maximize or minimize
  • Variables: Decisions that influence the objective function
  • Constraints: Set of relations that variables are force to satisfy

Optimization is Everywhere

Shortest Path Problem

Particular cases

• Multiple objective functions: **Multicriteria optimization**
  • Different optimum for each objective function
  • Multiobjective optimization
  • Satisfying methods
• **There is no objective function**
  • (Non) Linear system of equations
  • Find a feasible solution
• There are no constraints: **Unconstrained optimization**
  • Determine the minimum of a function

MODEL AND MODELING

Model

- Accurate representation of a reality
- Decision support tool
- May involve a multidisciplinary team
- Balance between a detailed representation and the skill to obtain a solution
- Modeler: specifies and develops de model
- Expert: knows the real problem

2 main risk

Exhaustive model, quasi-real. It may be impossible to find an algorithm for solving the problem

- Simple model to use an available algorithm. It may obtain solutions for a non existent problem
- Model must be a compromise between those pathological extremes

Modeling

- Science
  - Analysis and detection of relations among data
  - Hypotheses and approximations
  - Specific solution algorithms
  - Model solution
- Art
  - Vision and interpretation of the reality
  - Style in modeling and documentation
  - Elegance and simplicity in development
  - Creative use of tools

Mdeling benefits

- Organization of available information
- Structures the understanding of the problem
- Internalizes the organizational structure of the company
- Dialog between modeler and expert. Allows to share hypotheses between modeler and expert
- Gives an analysis tool
- Shows the improvement path by taking these decisions

Stages in Developing a Model

1. Problem Identification
2. Mathematical specification and problem formulation
3. Resolution
4. Verification, validation and refinement
5. Result analysis and interpretation
6. Implementation, documentation and maintenance

Problem Identification

- Collection of relevant information
- Problem definition in vague terms
- Interpretation and translation to accurate terms
- Data are crucial, use to be the bottleneck
- Essential stage for the decisions being useful

MULTICRITERIA RESUME

Methods

- Continuous decisions • Multiobjective optimization • Compromise programming • Satisfying methods

Basic Concepts

• Attribute: observable value (measurable) of an alternative, independent of the decision maker
• Objective: direction to improve an attribute (max o min if numerical; otherwise, preferences)
• Target: an acceptable level of achievement for an attribute • Goal: combination of an attribute with its target
• Criterion: relevant attributes, objectives or goals to a decision problem

Solution Concept

• Efficiency or Pareto optimality criterion: One feasible solution is efficient or Pareto optimal if no other feasible solution can yield an improvement in one attribute without causing a degradation in at least another attribute.
• Dominated or non efficient alternative: there is another one with better attributes

Multiobjective Optimization

• Pay-off matrix: optimum value for an objective (without considering the others), and the value of the other objectives for this solution. Shows the level of conflict among the different objectives.

• Trade-offs: degradation of one objective to improve another in one unit. Efficient set slopes=la pendiente

$$\left.\begin{array}{l}\max \quad z=\left(z_1(x), z_2(x), \ldots, z_p(x)\right) \\ x \in F\end{array}\right\} \quad \begin{array}{ll} z_i(x) & : \text{Mathematical expression of attribute } i \\ x & : \text{decisional variables vector} \\ F & : \text{constraints defining feasible solutions}\end{array}$$

$$\left.\begin{array}{ll}\max & \sum_{i=1}^{p} \lambda_i \cdot z_i(x) \\ s.t.: & \\ x \in F & \lambda \geq 0\end{array}\right\} P(\lambda)$$

• Theorem: if $\lambda i \geq 0$ para todo i then any optimal solution of P()● is efficient
• Converse of the theorem only true under some assumptions
• normalize criteria

Compromise Programming

Ideal Point: optimum values of each objective subject to problem constraints.
• Optimum element or best compromise solution: Efficient solution closest to the ideal point (Zeleny's axiom of choice, 1976)
• Degree of closeness between attribute i and its anchor value, normalised

$$\min_{x \in F} L_\pi = \left[\sum_{i=1}^{p} w_i^\pi \left(\frac{z_i^* - z_i(x)}{z_i^* - z_{*i}}\right)^\pi\right]^{\frac{1}{\pi}}$$

Weights w_i : importance of discrepancy of criteria (subjective)
π=1 : Linear addition of weighted distances (linear)
π→∞ : Tchebychev distance, minimising maximum distance (li
Compromise set: varying π. Usually (L_1, L_∞)

Goal Programming= *"Satisfaying logic:Decision making behaviour where, instead of attempting to optimize system performance, a level of aspiration is set either subjectively or heuristically and no further effort is expended to exceed that level of performance"*

- Attribute: mathematical expression $z_i(x)$
- Target or level of aspiration: acceptable level of achievement $\hat{z}_i$
- Goal: $z_i(x) \geq \hat{z}_i$
- With deviation variables: $z_i(x) + n_i - p_i = \hat{z}_i \quad n_i, p_i \geq 0$

n_i : negative deviation $\quad p_i$: positive deviation

$$\begin{array}{ll}\min \sum_{i=1}^{p} n_i & \\ s.t.: & \\ x \in F & \\ z_i(x) + n_i - p_i = \hat{z}_i & \forall i \\ n_i, p_i \geq 0 & \forall i\end{array}$$

- Deviation variables to be minimized:
  - If a goal is "at least" a value $z_i(x) \geq \hat{z}_i$ the negative deviation n_i is minimize
  - If a goal is "at most" a value $z_i(x) \leq \hat{z}_i$ the positive deviation p_i is minimize

TEMA 4 NON LINEAR PROGRAMMING PROBLEMS

Introduction

Type of NLP Problems

1. **Optimization WITHOUT constraints**
2. **Optimization WITH constraints**
3. **Quadratic Programming**
4. **Convex Programming**
    a. f(x) is convex (concave if we are maximizing) and is convex,
5. **Separable Programming**
6. **Geometric Programming**

Preliminary Concepts

1. **Gradient and Hessian**
2. **Taylor series expansion**
3. **Positive definite matrix** -->si todos sus autovalores son positivos
4. **Positibe semidefinite matrix**
5. **Negative definite matrix**
6. **Convexity**
7. **Concavity**
8. **Teorema de Weierstrass**

Optimality Conditions for Non-Linear Unconstrained Optimization

Local minimum, global minimum

Optimality Conditions for Non-Linear Constrained Optimization

Langrangian

Methods for Unconstrained Optimization

LINEAR PROGRAMMING pg 5, 17,27,28,30

Graphical Solution	Graphical Sensitivities	Simplex Method	Simplex Methodology	Duality	Sensitivity Analysis

• Graphical Solution

• Graphical Sensitivities

- Feasible region :region que queda dentro de los limites | El gradiente | Craftsman :artesano | Las curvas de nivel
- LP :linear problem | The feasible region of a linear programming problem is a polyhedron
- Vertex :vertice | If an optimal solution of a LP problem exists, it can be found in a vertex

• Simplex Method

1. **Geometry of linear programming**
2. **Standard Form**
3. **Transformation into Standard Form**
   - | | la h=0 es el limite, la constrain
   -
   - **free variable**
4. **Slack/Excess Variables**
5. **Feasible and Infeasible Vertices**
6. **Types of Solution**
   - **feasible**
   - **infeasible**
   - **basic feasible** :Has m* variables than can take a non-zero value, called basic variables, while the other variables are equal to zero. The feasible vertices of the feasible region are basic feasible solutions.
   - **optimal solution** : Basic feasible solution (feasible vertex) with best objective function value
7. **Fundamental Theorem of Linear Programming**

• Simplex Methodology

- | **non basi**c variables set to zero | **Xb** |
  | **basic variables** can take a non zero value | **Xn** |

1.

• Duality

• Sensitivity Analysis

VOCABULARY

- Unbounded solution : si te quedas sin region, si no tienes constrains te da una unbounded solution, que esta mal
- Bounded : obligado a
- Standar form
- types of solution
- Basic solution

TEMA 3 Mixed Integer Linear Modeling:

SEVERAL CHARACTERISTIC PROBLEM

Transportation Problem Formulation

Sets

i origins {1 ... m}

j destinations {1 ... n}

Parameters

a_i product supply in origin i

b_j product demand in destination j

c_{ij} unitary cost from i to j

Variables

x_{ij} units of product transported from i to j

Objective function

$$\min \sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij} \cdot x_{ij}$$

Constraints

1. Offer available in each origin i

$$\sum_{j=1}^{n} x_{ij} = a_i \quad \forall i = 1,\dots,m$$

2. Demand in each destination j

$$\sum_{i=1}^{m} x_{ij} = b_j \quad \forall j = 1,\dots,n$$

3. Variables constraints

$$x_{ij} \geq 0 \quad \forall i, j$$

Hypothesis: supply equals demand of the product:

$$\sum_{i=1}^{m} a_i = \sum_{j=1}^{n} b_j$$

If $\sum_{i=1}^{m} a_i > \sum_{j=1}^{n} b_j$ add an universal sink with zero cost

If $\sum_{i=1}^{m} a_i < \sum_{j=1}^{n} b_j$ add an universal source with very high cost

Transshipment Problem

Determine in a network of n nodes the cheapest routes to carry product units from their origins to their destinations through intermediate transshipment locations

• Each origin generates units	• Each destination consumes units
• Each transshipment neither generates nor consumes units	• transportation unit cost from i to j in this direction

Sets

i, j nodes {1 ... n}

Parameters

b_i product supply/demand in node i

c_{ij} unitary cost from i to j

Variables

x_{ij} units of product transported from i to j

Objective function

$$\min \sum_{i=1}^{n}\sum_{j=1}^{n} c_{ij} \cdot x_{ij}$$

• Hypothesis: supply equals demand $\sum_{i=1}^{n} b_i = 0$

Constraints

1. Balance (or flow conservation) in each node i

$$\sum_{j=1}^{n} x_{ij} - \sum_{j=1}^{n} x_{ji} = b_i \quad \forall i = 1,\dots,n$$

2. Variables constraints

$$x_{ij} \geq 0 \quad \forall i, j$$

Transshipment Problem

Sets

i tasks {1 ... n}

j people {1 ... n}

Parameters

c_{ij} cost of doint task i by person j

Variables

x_{ij} whether task i is done by person j or not

Objective function

$$\min \sum_{i=1}^{n}\sum_{j=1}^{n} c_{ij} \cdot x_{ij}$$

x_{ij} are binary variables but it is not necessary to declare them as binary

Constraints

1. Each task i is done by 1 person

$$\sum_{j=1}^{n} x_{ij} = 1 \quad \forall i = 1,\dots,n$$

2. Each person j does 1 task

$$\sum_{i=1}^{m} x_{ij} = 1 \quad \forall j = 1,\dots,n$$

3. Variables constraints

$$x_{ij} \geq 0 \quad \forall i, j$$

Task Assignment Problem

Task Assignment Problem

• Min the total cost of doing n tasks by n people, knowing that each person does 1 task and each task is done by 1 person.

• It is a particular case of the transportation problem.	• cost of doing task i by person j

Knapsack Problem

Max the total value of selecting a set of projects without exceeding the available budget, among n projects

. •c cost of project j	•v value of project j	b available budget

Sets

j projects {1 ... n}

Parameters

c_j cost of project j

v_j value of project j

b available budget

Variables

x_j whether project j is selected or not

Objective function

$$\max \sum_{j=1}^{n} v_j \cdot x_j$$

Constraints

1. Limit on the available budget

$$\sum_{j=1}^{n} c_j \cdot x_j \leq b$$

2. Variables constraints

$$x_j \in \{0,1\} \quad \forall j$$

Knapsack Problem

Sets

i characteristics {1 ... m}

j sets of characteristics {1 ... n}

Parameters

c_j cost of set j of characteristics

a_{ij} Membership matrix (0/1). 1 if characteristic i belongs to set j

Variables

x_j whether set j of characteristics is selected or not

Objective function

$$\min \sum_{j=1}^{n} c_j \cdot x_j$$

Constraints

1. Each characteristic i has to be selected at least once

$$\sum_{j=1}^{n} a_{ij} \cdot x_j \geq 1 \quad \forall i$$

2. Variables constraints

$$x_j \in \{0,1\} \quad \forall j$$

Set Covering Problem

Set Covering Problem

• m characteristics (flights) and n sets of characteristics (sequences of flights)
• Every set j of characteristics (sequences of flights) has a cost
• If a set of characteristics is selected, all the characteristics (flights) of this set must be done.
• Minimize the total cost of the selected sets of characteristics in such a way that all characteristics are

covered at least once.

Set Packing Problem

• m projects and n sets of projects • Every set j of projects has a benefit • Maximize the total benefit of the selected sets of projects without doing one project more than once.

Sets

i projects {1 … m}

j sets of projects {1 … n}

Parameters

b_j benefit of set j of projects

a_{ij} Membership matrix (0/1).

1 if project i belongs to set j

Variables

x_j whether set j of projects is selected or not

Objective function

$$\max \quad \sum_{j=1}^{n} b_j \cdot x_j$$

Constraints

1. Each project i cannot be selected more than once

$$\sum_{j=1}^{n} a_{ij} \cdot x_j \leq 1 \quad \forall i$$

2. Variables constraints

$$x_j \in \{0,1\} \quad \forall j$$

LOGICAL PROPOSITION

1. Disjunctive preposition
2. Fulfill at least k o N constraints
3. Selecting from N values
4. Fixed cost problem
5. Simple preposition
6. Proposition of $\leq$ Constraint (1)
7. Proposition of $\leq$ Constraint (2)
8. Proposition of $\geq$ Constraint (1)
9. Proposition of $\geq$ Constraint (2)
10. Proposition of = Constraint (1)
11. Proposition of = Constraint (2)
12. Double preposition
13. Preoposition Modeling table
14. Double preposition Modeling table
15. Basket team Problem

Game theory

Clasificaction

• **Number of players**	• Two person game	• N-person game
• **Information exchange among player**	• Cooperative	• No coperative
• **Time evolution**	• Static	• Dynamic
• **Number of strategies**	• Finite	• Infinite (continuous)

Two person zero-sum game

Que uno gane supone que el otro pierda.No son copertivos
Vamos eliminando estrategias q no le benefician a cada uno de los players hasta q solo te quede 1 celda

GAMS resumen y errores

1. Cuidado con los puntos y comas
2. Si no pones lo verde te sale un error muy feo | * si pinchas sobre el error te lleva a la linea de solve y no a esta
3. Para cambiar el SOLVER | file-->options-->solvers-->BARON MINLP
4. Si quieres meter un valor concreto de un set | X(i,j,'MC')
5.

```
$TITLE CharterFlight
OPTIONS optcr =0;
$onsymxref
sets
i  Type of plane /A, B, C/
j  Number of planes that are used    /1, 2, 3, 4, 5/

scalar
d Transportationdemand /372/

parameter
cap(i) capacity of airplain of type i
/
A 80
B 68
C 55
/

p(i,j) Transport capacity (in passengers) with j
number of planes of type i;
p(i,j) = ord(j)*cap(i)

table
c(i,j) Cost using j number of planes of type i
        1      2      3      4      5
A       17     26     36     46     56
B       15     23     30     40     51
C       14     21     27     32     37

variable
OFV objective function [$]
X(i,j) Indicator if j number of planes of type i are
used

binary variable   X
```

```
equations

EQ_OF           objective function equation
EQ_PASS         number of passenger to transport
EQ_PLANES(i)    exact number of planes of type i
EQ_NEW          new equations
;

EQ_OF           ..  OFV =E= sum[(i,j),c(i,j) * X(i,j)];

EQ_PASS         ..  sum[(i,j),p(i,j)*X(i,j)] =G= d;

EQ_PLANES(i)    ..  sum[j, X(i,j)]=L= 1;

EQ_NEW .. sum[j,X('B',j)] =G= sum[j,X('A',j)];

model CharterFlight /EQ_OF,EQ_PASS,EQ_PLANES / ;
model CharterFlight2 /all/;

parameter
empty_seat num of empty seats in the plenws selected ;


solve CharterFlight minimizing OFV using MIP ;
empty_seat = sum[(i,j),p(i,j)*X.L(i,j)] -d ;
display X.L
display empty_seat

solve CharterFlight2 minimizing OFV using MIP ;
empty_seat = sum[(i,j),p(i,j)*X.L(i,j)] -d;
display X.L
display empty_seat
```

GAMS

• Lines with * in the first column are comments	• No distinction between uppercase and lowercase letters

• Parenthesis (), square brackets [], or braces {} can be used indistinctly to distinguish levels
• Language reserve words appear in bold • Sentences end with a semicolon ; (that can be suppressed when the following word is a reserve word)
• The ordering of entities is chosen so that no symbol is referred to before it is defined
• Names given to entities must start with a letter and can be followed by up to nine more letters or digits
• Multiword names like "New York" are not allowed, so hyphens can be inserted "New-York

SETS

sets i nodes in the network /T1,T2,D1*D3,P1,P2/ t(i) nodes that correspond to tanks /T1,T2/ d(i) nodes that correspond to deposits /D1*D3/ p(i) nodes that correspond to pools /P1,P2/	alias(i,j) sets k(i,j) existing connections between nodes of the network /T1.D1, D2.D1, D3.D1, P1.D1, P2.D3, D3.T2/; k(i,j)$k(j,i) = YES; D1*D3--> same as D1,D2,D3 alias(i,j) --> i and j can be used indistinctly

DATA ENTRY BY LIST

scalar	Scalar b budget available [€] /300/ ;

parameters

parameters cd(d) Daily domestic consumption of deposit d [m3] / D1 150 D2 100 D3 400 /;	parameters cd(d) Daily domestic consumption of deposit d [m3] /D1 150, D2 100, D3 400/ ;	parameter c(i,j) Transport cost from node i to node j [c€ per m3] / T1.D1 8 T2.D3 7 P1.D1 5 /;

DATA ENTRY BY TABLE

sets i /MAD, BCN/ j /A1, A2/ k /A, B, C/	table Capacity(i,j,k) maximum capacity A B C MAD.A1 1 0 3 MAD.A2 2 1 2	table Capacity(i,j,k) maximum capacity A1.A A1.B A1.C A2.A A2.B MAD 1 0 3 2 1 BCN 2 1 2 2 4

DATA ENTRY BY DIRECT ASSIGNEMET

Parameter c(i,j) transport cost [€ per m3] ; c(i,j) = f*d(i,j)/100 ;	c('T1','D1') = 0.08 ;

$ Operator in Assignments, Summations, Equations

x(i)$(z(i) > 0) = 1; *if(Zi>0) then x(i)=1*	x(i) = 1$(z(i) > 0);	$(Value > 0) $(Number1 <> Number2)

RELATIONAL OPERATOR

• Lt <	• gt >	• eq =	• ne <>	• le <=	• ge >=	• Not	• And	• Or	• xor

Summation (and product) notation:

SUM[j, X(i,j)]	SUM[(i,j), c(i,j)*X(i,j)]	PROD[j, X(i,j)]

VARIABLES

type: free (by default) -∞ to +∞ positive 0 to ∞ negative -∞ to 0 binary 0 or 1 Integer 0 to 100	• Suffixes • Var_name.lo lower bound • Var_name.up upper bound • Var_name.l current value (initial value before solving, optimal after) • Var_name.m marginal value (reduced cost) • Var_name.fx fixes a variable to a constant

VARIABLES EXAMPLE IN THE SWIMMING POOL

variables OF objective function X(i,j) daily flow between node i and node j [m3] U(d) lack of water for domestic use in the deposit d [m3] W(d) Lack of water for agricultural use in deposit d [m3] Y(p) Whether we close or not the swimming pool p Z Whether we close both pools	positive variables X,U,W; binary variables Y,Z; *upper bound of variables U.up(d) = cd(d); W.up(d) = ca(d)

EQUATION

equations EQ_OF objective function EQ_BALANCE(d) balance in each deposit d ;	EQ_OF .. OF =E= SUM[(i,j)$k(i,j) , c(i,j)*X(i,j)] + pd*SUM[d, U(d)] + pa*SUM[d, W(d)] + SUM[p, pp(p)*(1 - Y(p))] + q*(1 - Z) ; EQ_BALANCE(d) .. SUM[j$(k(j,d)), X(j,d) - X(d,j)] =E= cd(d) + ca(d) - U(d) - W(d) ;

• The name of the equation being defined

• The domain • Domain restriction condition (optional)		• The symbol '..'	• Left-hand-side expression
• Relational operator: =l=, =e=, or =g=	• Right-hand-side expression		

MODEL AND SOLVE STATEMENT

model pools /EQ_OF, EQ_BALANCE, EQ_TRANS_POOL, EQ_TRANS_TANK, EQ_CLOSE_2POOLS_1, EQ_CLOSE_2POOLS_2/ solve pools using mip minimizing OF; display X.L, Y.L, U.L, W.L, Z.L;	

TYPES OF PROBLEMS AND SOLVER

• LP (linear program): BDMLP, CLP
• MIP (mixed integer linear program): CPLEX, CBC, Gurobi, XPRESS
• RMIP (relaxed mixed integer linear program): BDMLP, CLP
• NLP (non linear program): CONOPT, MINOS, SNOPT, PATHNLP, LGO, MOSEK
• MINLP (mixed integer non linear program): DICOPT, SBB, BARON, OQNLP
• GAMS solvers (docs: https://www.gams.com/latest/docs/S_MAIN.html)

FUNCTIONS

- Elementary: +, -, *, /,** or power(x,n)
- Ord, card: Ordinal and cardinal of a set
  - z=sum((i,j)$(ord(i) ne ord(j)), x(i,j))
- With indexes: sum, prod, smax, smin
- Other functions: abs, arctan, sin, cos, ceil, floor, exp, log, log10, max, min,
- mod, round, sign, sqr, sqrt, trunc, normal, uniform

DYNAMIC SETS

Subset of static sets whose content may change by assignments

```
Sets
m months    /m1*m12/
mp(m) even months;
mp(m)$[MOD(ord(m),2) = 0] = YES;
display mp;
mp('m3') = yes; display mp;
mp(m)$(ord(m)=3) = NO; display mp;
```

ARENA

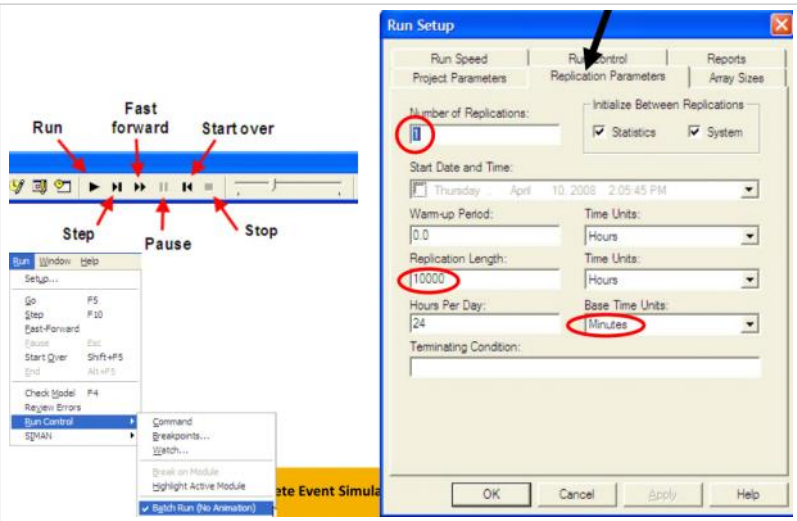
1. **Pasos**
    1. Metes los entities
    2. **Una vez que has simualdo y visto los resultados, tienes que pulsar STOP**
2. **Basic Proces Modules :**
    - **Create**--> el que va al principio -->The way to introduce entities into a model
        - En el pones la expresion del enunciado que viene vinculada a la entity(los customer)
    - **dispose-->** el que va al final -->the entities that come into this module leave the model
    - **process-->** el q va en medio -->Set of activities experienced by an entity-->es un rectangulo
        - Ahí añades(add) el resource que tines(la farmacia), y ademas ahí dices el num de farmacias y la expresion que sige
    - **read/Write-->**
    - **record -->**This module "records" info each time an entity passes through it -->

• We will compute the average time in the System	es como un rectangulo con una esquina doblada

    - **assign** -->where values of variables/attributes are assigned-->es un octogono

The order of assignments is very important!	Ojo que cuando asignas las probabilidades son las acumuladas

    a. **entity** -->Elements that flow through the system **(customers)**
    b. **resource** -->An element demanded by entities with finite capacity-->**pharmacist**
        i. **en su menu de abajo es donde se cambia el numero de pharmacist´-**
    c. **variable -->**Quantities that are properties of the System that may change
        i. se le pueden poner columnas | empiezan por v
    d. **atributes -->**It is a named property or characteristic of an entity (dimensions, color, reference,)
        i. empieza por my |
3. **Ejecutar con RUN**

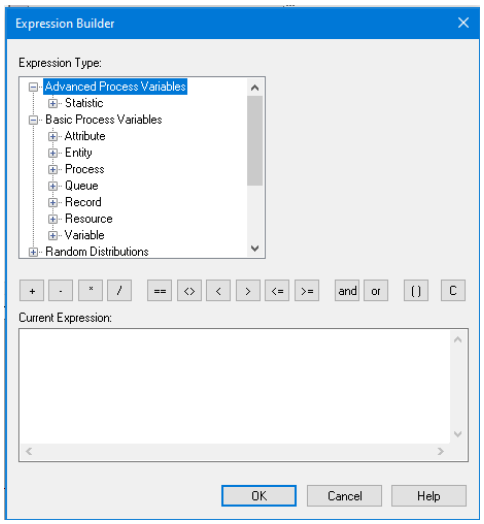


fast forward para que ser acabe cuanto antes

El numer of replication es el que a veces hay q cambiar

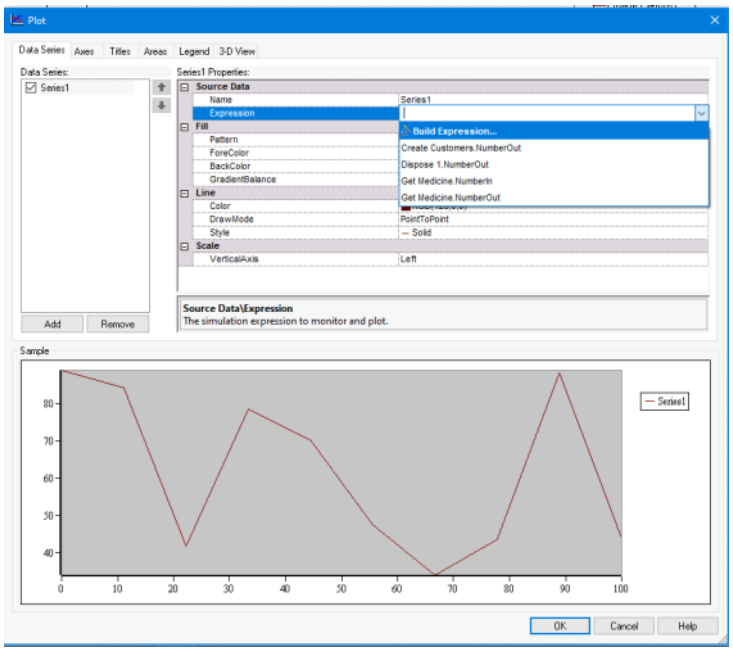
Cuidado con el warm up periode

4. **Build expresion**



Boton derecho en new value cuando metes un nuevo assigments o cuando plotesas

1. **Analizar resultados**
   a. Average waiting time-->esta en el apartado de queue
2. **Plotear**



Ahi va build expresion, te vas luego a queue y a current numer in Queue

3. **Vocabulario rario**
   a. TNOW: Current simulation time. TNOW records the simulation clock time as the model progresses.
   b. TFIN: Final simulation time. TFIN is the ending time scheduled for the replication; it is a real-valued quantity.
   c. DISC: Discrete cumulative distribution function. DISC(cumulative probability 1, value 1, cumulative prob.1, value 2, …, c.p.N, value N).
4. **Escribir/leer de ficheros**
   a. module helps to debug the model (analyzing arrival and departure times) in the Advanced Process Panel