

Ejercicios Programación Matemática. Parte I

Programación Matemática. Curso 2010-2011

Leyenda Rodríguez, María

Índice

1. Ejercicio 1	3
1.1. $f(x, y) = -x^2 - y^2 + 2x + 2xy - y$	3
1.2. $f(x, y) = x^4 - y^2$	4
1.3. $f(x, y) = x^2 + 2xy + 2 * y^2 - 10x - 10y$	5
2. Ejercicio 2	6
2.1. $f(x, y) = \ln(x + y)$ con $(x, y) \in \mathbb{R}_{++}^2$	6
2.2. $f(x, y) = ax + by + c$ con $(x, y) \in \mathbb{R}^2$ $a, b, c \in \mathbb{R}$	7
3. Ejercicio 3	8
4. Ejercicio 4	9
5. Ejercicio 5	11
5.1. Comenzar con la solución inicial $(x_1, x_2) = (\frac{1}{4}, \frac{1}{4})$ para realizar tres iteraciones del algoritmo de Frank-Wolfe	11
5.2. Utilizar las condiciones de Kuhn-Tucker para verificar si la solución obtenida en el apartado anterior es óptima	16
6. Ejercicio 6	19
7. Ejercicio 7	25
7.1. Resolverlo gráficamente	25
7.2. Comprobar la respuesta utilizando la teoría de Kuhn-Tucker	26
7.3. Encontrar la función dual $I(\lambda, \mu)$ y probar que es cóncava	27
8. Ejercicio 8	29
9. Ejercicio 9	32
10. Ejercicio 10	36
10.1. Representar gráficamente la región factible X.	36
10.2. Representar el espacio de objetivos Z(X).	37
10.3. Determinar gráficamente los puntos eficientes por medio de los contornos de los objetivos.	39
10.4. Comprobar si se verifica el recíproco del segundo teorema de Yu and Zeleni.	39
11. Ejercicio 11	40
12. Ejercicio 12	44
12.1. Establecer y resolver una sucesión de programas lineales para calcular una solución lexicográfica del modelo considerando los objetivos en el orden dado.	44

12.2. Establecer y resolver un programa lineal con suma ponderada de objetivos que pondere al primero el doble que al segundo.	46
12.3. Supóngase que se establece en el coste una meta de 30.000 y en la puntuación de 13.000. Plantear un problema de programación por metas que minimice la suma no ponderada de no cumplimiento de las metas.	47
12.4. Establecer y resolver una sucesión de programas lineales para calcular una solución de programación por metas lexicográfica tomando los objetivos en el orden dado. .	48
12.5. Formular una función objetivo alternativa del programa por metas de la parte c) que efectúe la optimización por metas lexicográfica de la parte d) en un único paso.	50

1. Ejercicio 1

Estudiar la concavidad y la convexidad de las siguientes funciones:

Definición

$\emptyset \neq S \subset R^n$ se denomina un **conjunto convexo** si el segmento que une dos puntos cualesquiera de S también pertenece a S :

$$x_1, x_2 \in S, \lambda \in [0, 1] \Rightarrow \lambda x_1 + (1 - \lambda)x_2 \in S \quad (1)$$

Sea $f : S \rightarrow R$ con S un conjunto convexo. Se dice que es una función convexa (J. L. W. Jensen, 1905) sobre S si:

$$x_1, x_2 \in S, \lambda \in [0, 1] \Rightarrow f(\lambda x_1 + (1 - \lambda)x_2) \leq \lambda f(x_1) + (1 - \lambda)f(x_2) \quad (2)$$

i. e., el valor de f en los puntos del segmento $\lambda x_1 + (1 - \lambda)x_2$ es menor o igual que la altura de la cuerda que une los puntos $(x_1, f(x_1))$ y $(x_2, f(x_2))$.

Resultados

- Si $\lambda \in (0, 1)$ y la desigualdad anterior se satisface estrictamente, entonces f es una función estrictamente convexa.
- Si f es una función convexa entonces $-f$ es una función cóncava.

TEOREMA

Sea $f : S \rightarrow R$ una función dos veces diferenciable sobre un conjunto no vacío, abierto y convexo $S \subset R^n$. Entonces f es convexa si y sólo si la Hessiana es semidefinida positiva en cada punto de S . Si la Hessiana es definida positiva en cada punto de S , entonces la función f es estrictamente convexa. Pero si f es estrictamente convexa, entonces la Hessiana es semidefinida positiva en cada punto de S .

TEOREMA (Teorema de Sylvester)

Para la matriz simétrica $n \times n$ Q , definimos Δ_k como el determinante de la submatriz $k \times k$ de Q de la esquina superior izquierda, para $k = 1, \dots, n$ y lo denominamos k -ésimo menor principal de Q . Entonces, Q es definida positiva si y sólo si $\Delta_k > 0$ para $k = 1, \dots, n$ y definida negativa si y sólo si $(-1)^k \Delta_k > 0$ para $k = 1, \dots, n$. En otro caso, Q es indefinida.

TEOREMA

La matriz simétrica Q es definida positiva (negativa) si y sólo si todos los autovalores de Q son positivos (negativos) y es semidefinida positiva (negativa) si y sólo si todos los autovalores de Q son no negativos (no positivos). La matriz Q es indefinida si y sólo si Q tiene al menos un autovalor positivo y al menos un autovalor negativo.

1.1. $f(x, y) = -x^2 - y^2 + 2x + 2xy - y$

SOLUCIÓN

Si esta función fuese convexa, su matriz Hessiana sería semidefinida positiva, es decir, si todos sus autovalores son no negativos. Lo que no es cierto ya que sus autovalores son -4 y 0. Luego f no es convexa.

Implementación en maxima

```
(%i1) load(vect)$
(%i2) f:-x^2-y^2 + 2*x + 2*x*y -y;
          2          2
(%o2)      - y  + 2 x y - y - x  + 2 x
(%i3) H:hessian(f,[x,y]);
          [ - 2    2 ]
(%o3)      [         ]
          [  2    - 2 ]
(%i4) determinant(H);
(%o4)      0
(%i5) eigenvalues(H);
(%o5)      [[- 4, 0], [1, 1]]
```

A continuación estudiamos si esta función es cóncava o lo que es equivalente si -f es convexa. Denotamos $g(x,y)=-f(x,y)$ y estudiamos la matriz Hessiana de esta nueva función. Obtenemos que sus autovalores son no negativos 0 y 4. Por tanto -f es convexa, luego f es cóncava.

Implementación en maxima

```
(%i6) g:x^2+y^2 - 2*x - 2*x*y +y;
          2          2
(%o7)      y  - 2 x y + y + x  - 2 x
(%i8) H1:hessian(g,[x,y]);
          [  2    - 2 ]
(%o8 )    [         ]
          [ - 2    2 ]
(%i9) determinant(H1);
(%o9)      0
(%i10) eigenvalues(H1);
(%o10)      [[0, 4], [1, 1]]
```

1.2. $f(x, y) = x^4 - y^2$

SOLUCIÓN

Esta función no es convexa, pues su matriz Hessiana no tiene sus autovalores bo negativos. Además como un atovalor es negativo.-2, y otro positivo, $12x^2$, podemos afirmar que la matriz Hessiana es indefinida. Por tanto, esta función no es ni cóncava ni convexa.

Implementación en maxima

```
(%i1) f:x^4-y^2;
(%o1)

$$x^4 - y^2$$

(%i2) H:hessian(f,[x,y]);
(%o2)

$$\begin{bmatrix} 4x^3 & 0 \\ 0 & -2y \end{bmatrix}$$

(%i3) determinant(H);
(%o3)

$$-2y^2$$

(%i4) eigenvalues(H);
(%o4)

$$\begin{bmatrix} 12x^3 & 0 \\ 0 & -2y \end{bmatrix}, [1, 1]$$

```

1.3. $f(x, y) = x^2 + 2xy + 2y^2 - 10x - 10y$

SOLUCIÓN

Esta función es estrictamente convexa pues todos los autovalores de su matriz Hessiana tiene sus autovalores mayores que cero($0.763932 > 0, 5.236068 > 0$). Por tanto, f es una función convexa.

Implementación en maxima

```
(%i1) f:x^2+2*x*y+2*y^2-10*x-10*y;
(%o1)

$$x^2 + 2xy + 2y^2 - 10x - 10y$$

(%i2) H:hessian(f,[x,y]);
(%o2)

$$\begin{bmatrix} 2 & 2 \\ 2 & 4 \end{bmatrix}$$

(%i3) determinant(H);
(%o3)

$$-4$$

(%i4) eigenvalues(H);
(%o4)

$$[[3 - \sqrt{5}, \sqrt{5} + 3], [1, 1]]$$

```

2. Ejercicio 2

Estudiar si las siguientes funciones son cuasiconvexas o cuasicóncavas:

DEFINICIÓN

Sea S un subconjunto no vacío y convexo de R^n y sea $f : S \rightarrow R$.

- Se dice que f es cuasiconvexa (B. de Finetti, 1954) si para todo $x, y \in S$ se verifica: $f(\lambda x + (1 - \lambda)y) \leq \max\{f(x), f(y)\} \forall \lambda(0; 1)$ f es cuasicóncava si $-f$ es cuasiconvexa.
- Se dice que f es estrictamente cuasiconvexa si para todo $x, y \in S$ con $x \neq y$ se verifica: $f(\lambda x + (1 - \lambda)y) < \max\{f(x), f(y)\} \forall \lambda(0; 1)$ f es estrictamente cuasicóncava si $-f$ es estrictamente cuasiconvexa.

PROPOSICIÓN

Sea S un subconjunto abierto, no vacío y convexo de $\mathbb{R}^n$ y f una función definida de S en $\mathbb{R}$ y diferenciable en S . Se verifica que:

- La función f es cuasiconvexa en S si y sólo si para cualesquiera $x, y \in S$ tales que $f(y) \leq f(x)$ se tiene que $\nabla f(x)(y - x) \leq 0$
- La función f es cuasiconvexa en S si y sólo si para cualesquiera $x, y \in S$ tales que $f(y) \leq f(x)$ se tiene que $\nabla f(x)(y - x) < 0$ (o equivalentemente, si $\nabla f(x)(y - x) \geq 0$, en particular cuando $\nabla f(x) = 0$, se tiene que $f(y) > f(x)$) entonces la función f es estrictamente cuasiconvexa en S .

2.1. $f(x, y) = \ln(x + y)$ con $(x, y) \in \mathbb{R}_{++}^2$

SOLUCIÓN

Al calcular los autovalores de la matriz de la matriz Hessiana, obtenemos que sus autovalores son no negativos, pues $\frac{-2}{(x+y)^2} < 0$. Por tanto la matriz es semidefinida negativa en $\mathbb{R}_{++}^2$ luego f es cóncava en $\mathbb{R}_{++}^2$ y en particular f es cuasicóncava.

Implementación en máxima

```
%i1) f:log(x+y);
(%o1)                                     log(y + x)
(%i2) H:hessian(f,[x,y]);
[      1      1      ]
[ - ---- - ---- ]
[      2      2      ]
[ (y + x) (y + x) ]
(%o2) [      ]
[      1      1      ]
[ - ---- - ---- ]
[      2      2      ]
```

```

                                [ (y + x)      (y + x) ]
(%i3) determinant(H);
                                0
(%o3)
(%i4) eigenvalues(H);
                                2
(%o4) [[- -----, 0], [1, 1]]
                                2          2
                                y  + 2 x y + x

```

2.2. $f(x, y) = ax + by + c$ **con** $(x, y) \in \mathbb{R}^2$ $a, b, c \in \mathbb{R}$

SOLUCIÓN

A la hora de estudiar esta función tendremos en cuenta la proposición anterior; es decir, para estudiar si esta función es cuasiconvexa comprobaremos que para cualesquiera $(x_1, y_1), (x_2, y_2) \in \mathbb{R}^2$ tales que $f(x_2, y_2) \leq f(x_1, y_1)$ se tiene que $\nabla f(x_1, y_1)((x_2, y_2) - (x_1, y_1)) \leq 0$.

Luego si

$$f(x_2, y_2) \leq f(x_1, y_1) \Rightarrow ax_2 + by_2 + c \leq ax_1 + by_1 + c \Rightarrow ax_2 - ax_1 + by_2 - by_1 \leq 0 \quad (3)$$

Por otra parte tenemos que

$$\nabla f(x_1, y_1)((x_2, y_2) - (x_1, y_1))^t = (a, b)(x_2 - x_1, y_2 - y_1) = a(x_2 - x_1) + b(y_2 - y_1) = ax_2 - ax_1 + by_2 - by_1 \quad (4)$$

Por tanto, tenemos que $\nabla f(x_1, y_1)((x_2, y_2) - (x_1, y_1)) = ax_2 - ax_1 + by_2 - by_1$ por (2) y por (1) sabemos que

$$\nabla f(x_1, y_1)((x_2, y_2) - (x_1, y_1)) = ax_2 - ax_1 + by_2 - by_1 \leq 0$$

3. Ejercicio 3

Resolver graficamente el siguiente problema de programación no lineal. Optimizar

$$f(x_1, x_2) = (x_1 - 2)^2 + (x_2 - 2)^2 \quad (5)$$

s.a.

$$2x_1 + 3x_2 \geq 6 \quad (6)$$

$$3x_1 + 2x_2 \leq 12 \quad (7)$$

SOLUCIÓN

A continuación representamos (Figura 1) tres curvas de nivel de la función objetivo (5); representamos en rojo la curva de nivel $f(x_1, x_2) = 1$, en naranja la curva de nivel $f(x_1, x_2) = 2$ y en violeta la curva de nivel $f(x_1, x_2) = 3$. A continuación representamos en verde la restricción dada por (6), en azul la restricción dada por (7) y en negro los ejes de coordenadas. A la vista del gráfico, observamos que el mínimo se alcanza cuando $(x_1, x_2) = (0, 0)$ por tanto $f(0,0)=0$

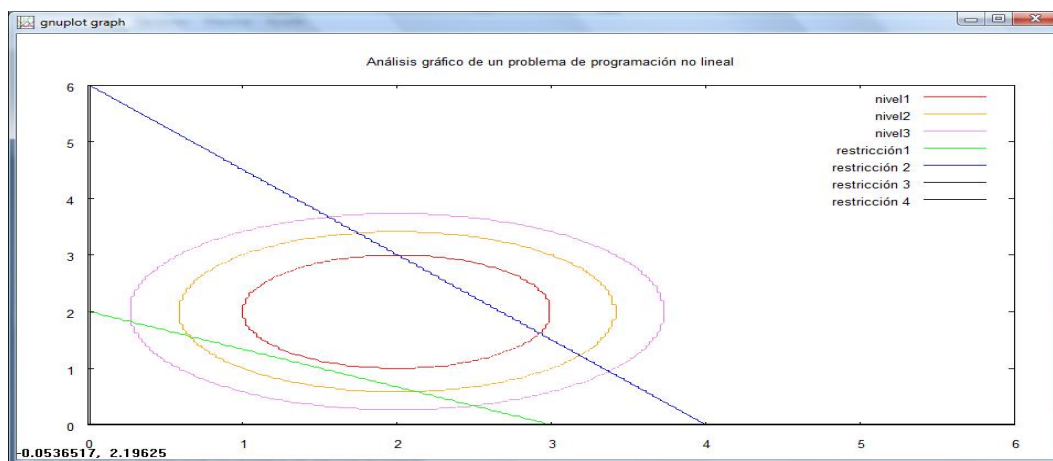


Figura 1: SOLUCIÓN gráfica

Implementación en máxima

```
(%i1)load(draw)$
(%i2) draw2d(key="nivel1",color=red,implicit((x-2)^2+(y-2)^2=1,x,0,6,y,0,6), key="nivel2",
key="restricción 2",color=blue,implicit(3*x+2*y-12=0,x,0,6,y,0,6),key="restricción 3",col
```

4. Ejercicio 4

Un monopolista vende dos productos 1 y 2 cuyas funciones de demanda vienen definidas por las ecuaciones: $0,1p_1 - 1,2 + 0,2x_1 = 0$ $10p_2 - 320 + 40x_2 = 0$ en donde p_1 ; p_2 y x_1 ; x_2 son, respectivamente, los precios y cantidades de los productos 1 y 2. Su función de costes es:

$$C(x_1; x_2) = x_1^2 + 2x_1x_2 + x_2^2$$

Se pide encontrar el precio y el output de cada bien que maximiza el beneficio empleando la función Lagrangiana comprobando que se verifica la condición de cualificación de restricciones.

Implementación en máxima En primer lugar, observamos que tal como está planteado el problema observamos al resolver el problema mediante **xmaxima** que el problema no tiene solución pues necesitamos cantidades de producto negativas.

```
(%i1) L:x^2+2*x*y+y^2+m*(0.1*p-1.2+0.2*x)+n*(10*q-320+40*y);
      2                               2
(%o1) y  + n (40 y + 10 q - 320) + 2 x y + x  + m (0.2 x + 0.1 p - 1.2)
(%i2) e1:diff(L,x,1);
(%o2)                2 y + 2 x + 0.2 m
(%i3) e2:diff(L,y,1);
(%o3)                2 y + 2 x + 40 n
(%i4) e3:diff(L,m,1);
(%o4)                0.2 x + 0.1 p - 1.2
(%i5) e4:diff(L,n,1);
(%o5)                40 y + 10 q - 320
(%i6) e5:diff(L,p,1);
(%o6)                0.1 m
(%i7) e6:diff(L,q,1);
(%o7)                10 n
(%i8) algsys([e1,e2,e3,e4,e5,e6],[x,y,m,n,p,q]);
(%o8) [[x = %r2, y = - %r2, m = 0, n = 0, p = 12 - 2 %r2, q = 4 %r2 + 32]]
```

Si resolvemos el problema con la siguiente función de Costes $C(x_1; x_2) = x_1^2 - 2x_1x_2 + x_2^2$ utilizando **xmaxima**, llegamos a que necesitamos la misma cantidad de los productos 1 y 2, que denotamos por r . Luego el coste asociado al primer producto es $12 - 2r$ y al segundo producto es $32 - 4r$.

```
(%i1) L:x^2-2*x*y+y^2+m*(0.1*p-1.2+0.2*x)+n*(10*q-320+40*y);
      2                               2
(%o1) y  + n (40 y + 10 q - 320) - 2 x y + x  + m (0.2 x + 0.1 p - 1.2)
(%i2) e1:diff(L,x,1);
(%o2)                - 2 y + 2 x + 0.2 m
(%i3) e2:diff(L,y,1);
(%o3)                2 y - 2 x + 40 n
```

```

(%i4) e3:diff(L,z,1);
(%o4) 0
(%i5) e3:diff(L,m,1);
(%o5) 0.2 x + 0.1 p - 1.2
(%i6) e4:diff(L,n,1);
(%o6) 40 y + 10 q - 320
(%i7) e5:diff(L,p,1);
(%o7) 0.1 m
(%i8) e6:diff(L,q,1);
(%o8) 10 n
(%i9) algsys([e1,e2,e3,e4,e5,e6],[x,y,m,n,p,q]);
(%o9) [[x = %r1, y = %r1, m = 0, n = 0, p = 12 - 2 %r1, q = 32 - 4 %r1]]

```

A continuación, comprobamos si se verifica la cualificación de restricción; para ello calculamos $\nabla(x^2 - 2xy + y^2) = (2x - 2y, 2y - 2x)$, $\nabla(0.2x + 0.1p - 1.2) = (0.2, 0)$, $\nabla(10q - 320 + 40y) = (0, 40)$ (notemos que derivamos respecto de x, y). Por lo que, se ve fácilmente que menos el gradiente de la función de costes evaluado en el óptimo es combinación lineal de los gradientes de las restricciones evaluados en dicho óptimo.

```

(%i1) f:x^2-2*x*y+y^2;
(%o1) y^2 - 2 x y + x^2
(%i2) h1:0.1*p-1.2+0.2*x;
(%o2) 0.2 x + 0.1 p - 1.2
(%i3) h2:10*q-320+40*y;
(%o3) 40 y + 10 q - 320
(%i4) load(vect)$
(%i5) grad(f)$
(%i6) express(%)$
(%i7) ev(%,diff);
(%o7) [2 x - 2 y, 2 y - 2 x, 0]
%i8) grad(h1)$
(%i9) express(%)$
(%i10) ev(%,diff);
(%o10) [0.2, 0, 0]
(%i11) grad(h2);
(%o11) grad (40 y + 10 q - 320)
(%i12) grad(h2)$
(%i13) express(%)$
(%i14) ev(%,diff);
(%o14) [0, 40, 0]

```

5. Ejercicio 5

Considérese el siguiente problema de programación convexa linealmente restringida: Maximizar

$$3x_1 + 4x_2 - x_1^3 - 2x_2^2$$

sujeto a la condición

$$x_1 + x_2 \leq 1$$

dónde $x_1, x_2 \geq 0$

5.1. Comenzar con la solución inicial $(x_1, x_2) = (\frac{1}{4}, \frac{1}{4})$ para realizar tres iteraciones del algoritmo de Frank-Wolfe

SOLUCIÓN

En primer lugar observamos que

$$\frac{\partial f}{\partial x_1} = 3 - 3x_1^2, \quad \frac{\partial f}{\partial x_2} = 4 - 4x_2 \quad (8)$$

por lo que los máximos no restringidos son $(x_1, x_2) = (1, 1)$ y $(x_1, x_2) = (-1, 1)$, pero ninguno cumple las restricciones funcionales y necesitamos iterar para encontrar el máximo restringido.

```
(%i1) f:3*x+4*y-x^3-2*y^2;
                                2          3
(%o1)          - 2 y  + 4 y - x  + 3 x
(%i2) load(vect)$
(%i3) df1:diff(f,x);
                                2
(%o3)          3 - 3 x
(%i4) df2:diff(f,y);
                                4 - 4 y
(%i5) algsys([df1,df2],[x,y]);
(%o5)          [[x = 1, y = 1], [x = - 1, y = 1]]
```

ITERACIÓN I

Se elige como la solución de prueba inicial $\mathbf{x}^0 = (\frac{1}{4}, \frac{1}{4})$ para el algoritmo de Frank-Wolfe, que claramente es factible. Al sustituir $x_1 = \frac{1}{4}$, $x_2 = \frac{1}{4}$ en las expresiones de las derivadas parciales(8) se obtiene $c_1 = 2,8125$ y $c_2 = 3$, con lo que

$$g(\mathbf{x}) = 2,8125x_1 + 3x_2 \quad (9)$$

Luego,(9)es la primera aproximación lineal de la función objetivo. Con la solución gráfica de este problema de programación lineal (Figura 2) se llega a $\mathbf{x}_{LP}^1 = (0, 1)$. En la parte 3 del paso iterativo, los puntos en el segmento de recta que une a $(0,25,0,25)$ y $(0,1)$ se expresan por

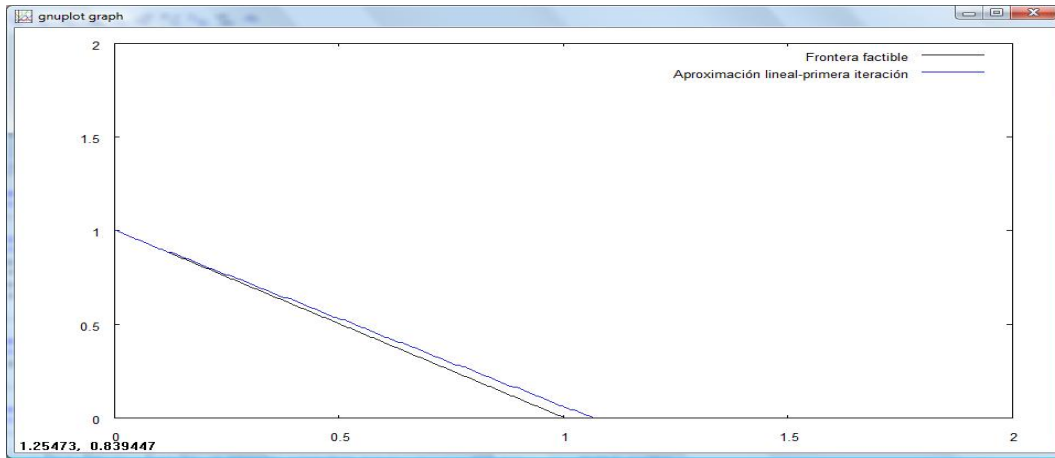


Figura 2: Representación gráfica correspondiente a la primera iteración

$$(x_1, x_2) = (0,25, 0,25) + t[(0, 1) - (0,25, 0,25)] = (0,25 - 0,25t, 0,25 + 0,75t) \quad (10)$$

Esta expresión (10) proporciona:

$$h(t) = f(0,25-0,25t, 0,25+0,75*t) = 4(0,75t+0,25)+3(0,25-0,25t)-2(0,75t+0,25)^2-(0,25-0,25t)^3 \quad (11)$$

$$\frac{dh(t)}{dt} = -3,0(0,75t+0,25)+0,75(0,25-0,25t)^2+2,25 = 0 \Rightarrow t^* = 0,6689499 \quad (12)$$

de modo que el valor $t^* = 0,6689499$ que maximiza (11) en el intervalo $0 \leq t \leq 1$ se puede obtener, en este caso, al establecer

Este resultado conduce a la siguiente solución de prueba

$$\mathbf{x}^1 = (0,25, 0,25) + 0,67 * [(0, 1) - (0,25, 0,25)] = (0,0825, 0,7525) \quad (13)$$

Lo cual completa la primera iteración

```
(%i6) df1,x=0.25,y=0.25;
(%o6) 2.8125
(%i7) df2,x=0.25,y=0.25;
(%o7) 3
(%i8)load(draw)$
(%i9)draw2d(key="Frontera factible",color=black,implicit(x+y=1,x,0,2,y,0,2),
key="Aproximación lineal-primerá iteración",color=blue,
implicit(2.8125*x+3*y=3,x,0,2,y,0,2));
```

```

(%i10) h:f,x=0.25-0.25*t,y=0.25+0.75*t;
(%o10) 4 (0.75 t + 0.25) + 3 (0.25 - 0.25 t) - 2 (0.75 t + 0.25)
- (0.25 - 0.25 t)
(%i11) dh:diff(h,t);
(%o11) - 3.0 (0.75 t + 0.25) + 0.75 (0.25 - 0.25 t) + 2.25
(%i12) algsys([dh],[t]);
(%o12) [[t = 4 sqrt(37) + 25], [t = 25 - 4 sqrt(37)]]

```

ITERACIÓN II

Al sustituir $x_1 = 0,0825$, $x_2 = 0,7525$ en las expresiones de las derivadas parciales(8) se obtiene $c_1 = 2,98$ y $c_2 = 0,99$, con lo que

$$g(\mathbf{x}) = 2,98x_1 + 0,99x_2 \quad (14)$$

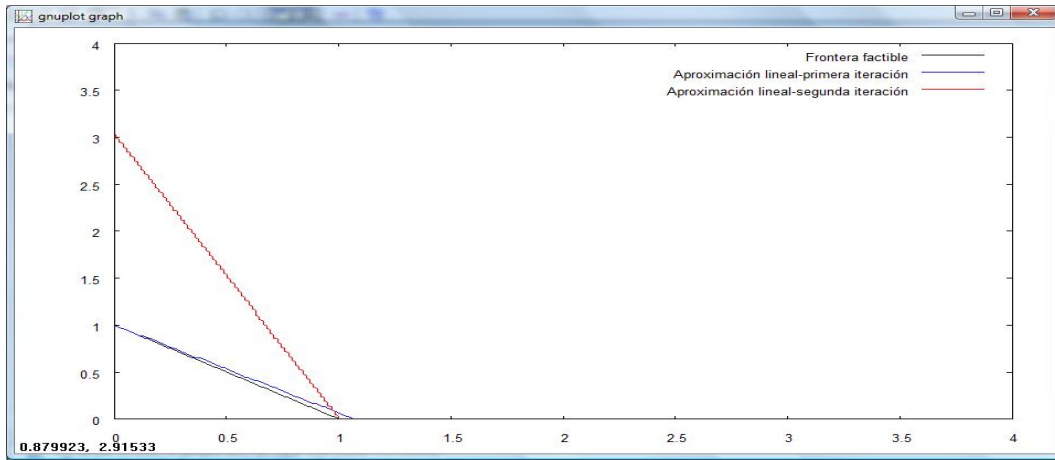


Figura 3: Representación gráfica correspondiente a la segunda iteración

Luego,(14)es la primera aproximación lineal de la función objetivo. Con la solución gráfica de este problema de programación lineal (Figura 3) se llega a $\mathbf{x}_{LP}^2 = (1, 0)$. En la parte 3 del paso iterativo, los puntos en el segmento de recta que une a $(0,0825, 0,7525)$ y $(1, 0)$ se expresan por

$$(x_1, x_2) = (0,0825, 0,7525) + t[(1, 0) - (0,0825, 0,7525)] = (0,0825 + 0,9175t, 0,7525 - 0,7525t) \quad (15)$$

Esta expresión (15) proporciona:

$$h(t) = f(0,0825 + 0,9175t, 0,7525 - 0,7525t) = 3(0,9175t + 0,0825) + 4(0,7525 - 0,7525t) - (0,9175t + 0,0825)^3 - 2(0,7525 - 0,7525t)^2 \quad (16)$$

$$\frac{dh(t)}{dt} = 3,01(0,7525 - 0,7525t) - 2,7525(0,9175t + 0,0825)^2 - 0,2575 = 0 \Rightarrow t^* = 0,5136496 \text{ o } t^* = 1,671024 \quad (17)$$

de modo que el valor $t^* = 0,5136496$ que maximiza (16) en el intervalo $0 \leq t \leq 1$ se puede obtener, en este caso, al establecer

Este resultado conduce a la siguiente solución de prueba

$$\mathbf{x}^2 = (0,0825 + 0,9175 * 0,51, 0,7525 - 0,7525 * 0,51) = (0,550425, 0,368725) \quad (18)$$

Lo cual completa la segunda iteración

```
(%i12) df1,x=0.0825,y=0.7525;
(%o12)                                     2.97958125
(%i13) df2,x=0.0825,y=0.7525;
(%o13)                                     0.99
(%i14) draw2d(key="Frontera factible",color=black,implicit(x+y=1,x,0,4,y,0,4),
key="Aproximación lineal-primer iteración",color=blue,
implicit(2.8125*x+3*y=3,x,0,4,y,0,4),key="Aproximación lineal-segunda iteración",color=red,
implicit(2.98*x+0.99*y=3,x,0,4,y,0,4));
(%i15) h:f,x=0.0825+0.9175*t,y=0.7525-0.7525*t;
(%o15) 3 (0.9175 t + 0.0825) + 4 (0.7525 - 0.7525 t) - (0.9175 t + 0.0825)
- 2 (0.7525 - 0.7525 t)
(%i16) dh:diff(h,t);
(%o16) 3.01 (0.7525 - 0.7525 t) - 2.7525 (0.9175 t + 0.0825) - 0.2575
(%i17) algsys([dh],[t]);
(%o17) [[t = -----],
148292589
400 sqrt(163995525205) + 85815011
[t = - -----]]
148292589
```

ITERACIÓN III

Al sustituir $x_1 = 0,550425$, $x_2 = 0,368725$ en las expresiones de las derivadas parciales(8) se obtiene $c_1 = 2,09$ y $c_2 = 2,53$, con lo que

$$g(\mathbf{x}) = 2,09x_1 + 2,53x_2 \quad (19)$$

Luego,(19)es la primera aproximación lineal de la función objetivo. Con la solución gráfica de este problema de programación lineal (Figura 4) se llega a $\mathbf{x}_{LP}^3 = (0,3)$. En la parte 3 del paso

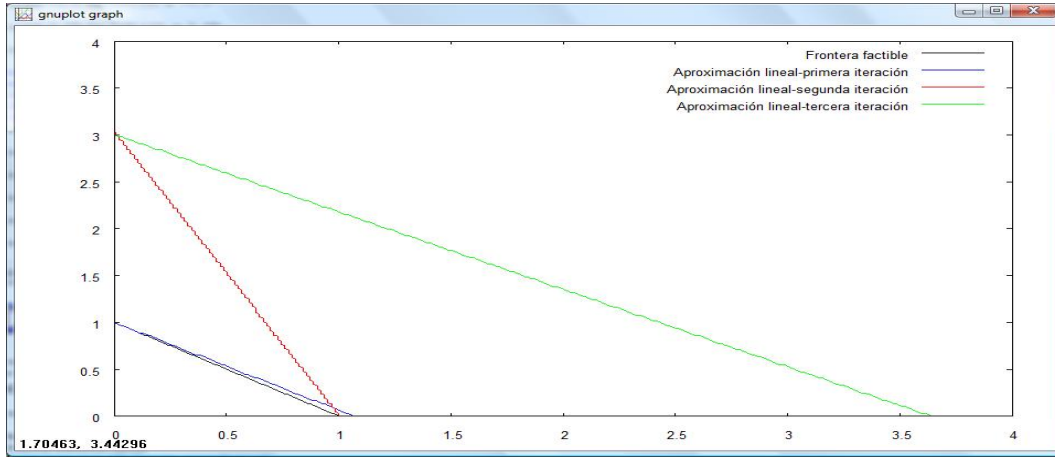


Figura 4: Representación gráfica correspondiente a la tercera iteración

iterativo, los puntos en el segmento de recta que une a $(0.550425, 0.368725)$ y $(0, 3)$ se expresan por

$$(x_1, x_2) = (0.550425, 0.368725) + t[(0, 3) - (0.550425, 0.368725)] = (0.550425 - 0.550425t, 0.368725 + 2.631275t) \quad (20)$$

Esta expresión (20) proporciona:

$$h(t) = f(0.550425 - 0.550425t, 0.368725 + 2.631275t) = -2(2.631275t + 0.368725)^2 + 4(2.631275t + 0.368725) + 3 \quad (21)$$

$$\frac{dh(t)}{dt} = -10.5251(2.631275t + 0.368725) + 1.651275(0.550425 - 0.550425t)^2 + 8.873825 = 0 \Rightarrow t^* = 57.165470t \quad (22)$$

de modo que el valor $t^* = 0.1920787$ que maximiza (21) en el intervalo $0 \leq t \leq 1$ se puede obtener, en este caso, al establecer

Este resultado conduce a la siguiente solución de prueba

$$\mathbf{x}^3 = (0.550425 - 0.550425 * 0.1920787, 0.368725 + 2.631275 * 0.1920787) = (0.4447001, 0.8741369) \quad (23)$$

Lo cual completa la tercera iteración(18) $\bar{x} = (0.550425, 0.368725)$

Por tanto como el último resultado obtenido no es un punto factible, tomamos el resultado obtenido en la segunda iteración

```
(%i18) df1,x=0.550425,y=0.368725;
```

```
(%o18) 2.091096958125
```

```
(%i19) df2,x=0.550425,y=0.368725;
```



```

(%o19)                                     2.5251
(%i20)draw2d(key="Frontera factible",color=black,implicit(x+y=1,x,0,4,y,0,4),
key="Aproximación lineal-primer iteración",color=blue,
implicit(2.8125*x+3*y=3,x,0,4,y,0,4),key="Aproximación lineal-segunda iteración",color=red,
implicit(2.98*x+0.99*y=3,x,0,4,y,0,4),key="Aproximación lineal-tercera iteración",color=green,
implicit(2.09*x+2.53*y=7.6,x,0,4,y,0,4));

(%i21)h:f,x=0.550425-0.550425*t,y=0.368725+ 2.631275*t;
2
(%o21) - 2 (2.631275 t + 0.368725) + 4 (2.631275 t + 0.368725)
3
+ 3 (0.550425 - 0.550425 t) - (0.550425 - 0.550425 t)
(%i22) dh:diff(h,t);
2
(%o22) - 10.5251 (2.631275 t + 0.368725) + 1.651275 (0.550425 - 0.550425 t)
+ 8.873825
(%i23) algsys([dh],[t]);
(%o23) [[t = (1451 sqrt(1060784166686983930245165353247451729305)
+ 47577261182224921225165)/1658971282728453286240],
[t = - (1451 sqrt(1060784166686983930245165353247451729305)
- 47577261182224921225165)/1658971282728453286240]]

```

5.2. Utilizar las condiciones de Kuhn-Tucker para verificar si la solución obtenida en el apartado anterior es óptima

En primer lugar, enunciaremos el **Teorema de Kuhn-Tucker: condiciones necesarias** para un problema general de programación con restricciones de igualdad y desigualdad

Problema P

- $\min f(x)$

sujeto a

- $g_i(x) \leq 0$ para $i = 1, \dots, m$
- $h_i(x) = 0$ para $i = 1, \dots, p$

Teorema de Kuhn-Tucker: condiciones necesarias

Karush (1939), Kuhn y Tucker (1951). Para el problema P sean $f, g_i, i = 1, \dots, m$ y $h_i, i = 1, \dots, p$ continuamente diferenciables. Supongamos que $\bar{x}$ es un punto factible que resuelve P localmente y tal que los vectores gradiente $\nabla g_i(\bar{x})$ para $i \in B = \{i : g_i(\bar{x}) = 0\}$ y $\nabla h_i(\bar{x})$ son linealmente independientes. Entonces existen escalares $\lambda_i, i = 1, \dots, m$ y $\mu_i, i = 1, \dots, p$ tales que

$$\nabla f(\bar{x}) + \sum_{i=1}^m \lambda_i \nabla g_i(\bar{x}) + \sum_{i=1}^p \mu_i \nabla h_i(\bar{x}) = 0 \quad (24)$$

$$\lambda_i g_i(\bar{x}) = 0, i = 1, \dots, m \lambda_i \geq 0, i = 1, \dots, m \quad (25)$$

junto con las restricciones iniciales $g_i(\bar{x}) \leq 0, i = 1, \dots, m$ $h_i(\bar{x}) = 0, i = 1, \dots, p$

SOIUCIÓN

En nuestro problema particular tenemos que $\bar{x} = (0,550425, 0,368725)$ que es un punto factible. Luego, estudiemos si este punto verifica las condiciones de Kuhn-Tucker.

$$\nabla f(x) = (3 - 3x^2, 4 - 4y) \Rightarrow \nabla f(\bar{x}) = (2,091096958125, 2,5251) \nabla g(x) = (1, 1) \Rightarrow \nabla g(\bar{x}) = (1, 1) \quad (26)$$

Luego, teniendo en cuenta (26)

$$-\nabla f(\bar{x}) = \lambda \nabla g(\bar{x}) \Rightarrow -2,091096958125 = a, -2,5251 = a \quad (27)$$

Luego, la solución no es óptima puesto que en (27) llegamos a una contradicción

```
%i1) load(vect)$
%i2) f:3*x+4*y-x^3-2*y^2;
      2      3
%i2) - 2 y  + 4 y - x  + 3 x
%i3) g:x+y;
      y + x
%i4) df1:diff(f,x);
      2
%i4) 3 - 3 x
%i5) df2:diff(f,y);
      4 - 4 y
%i6) dg1:diff(g,x);
      1
%i7) dg2:diff(g,y);
      1
%i8) df1x:df1,x=0.550425,y=0.368725;
      2.091096958125
%i9) df2x:df2,x=0.550425,y=0.368725;
      2.5251
%o9)
```

Vamos a estudiar si la solución obtenida por el algoritmo de Frank-Wolfe está próxima de la solución óptima. Resolviendo el problema usando las condiciones de Kuhn-Tucker llegamos a dos soluciones ($x = 0,5351838, y = 0,4648162, \lambda = 2,140735$) y ($x = -1,868517, y = 2,868517, \lambda = 2,140735$) como necesitamos que $x, y, \lambda \geq 0$ la solución óptima es ($x = 0,5351838, y = 0,4648162$). Si comparamos esta solución con la dada por el algoritmo de Frank-Wolfe ($x = 0,550425, y = 0,368725$) llegamos a que esta solución es proxima a la solución óptima, el valor de x difiere del óptimo 0.0152412 y el valor de y difiere del óptimo - 0.0960912.

(%i1) f:3*x+4*y-x^3-2*y^2;

(%o1)
$$-2y^2 + 4y - x^3 + 3x$$

(%i2) g:x+y-1;

(%o2)
$$y + x - 1$$

(%i3) L:-f+l*g;

(%o3)
$$2y^2 + 1(y + x - 1) - 4y + x^3 - 3x$$

(%i4) e1:diff(L,x);

(%o4)
$$3x^2 + 1 - 3$$

(%i5) e2:diff(L,y);

(%o5)
$$4y + 1 - 4$$

(%i6) e3:diff(L,1);

(%o6)
$$y + x - 1$$

(%i7) algsys([e1,e2,e3],[x,y,1]);

(%o7)
$$\left[\left[x = \frac{\sqrt{13} - 2}{3}, y = \frac{\sqrt{13} - 5}{3}, l = \frac{4\sqrt{13} - 8}{3} \right], \right. \\ \left. \left[x = \frac{\sqrt{13} + 2}{3}, y = \frac{\sqrt{13} + 5}{3}, l = \frac{4\sqrt{13} + 8}{3} \right] \right]$$

6. Ejercicio 6

Utilizando el libro de Hillier y Lieberman (2006) estudia la técnica secuencial de minimización no restringida (SUMT) para problemas de restricción lineal no restringida y aplícala a un caso propuesto por ti.

SOLUCIÓN

Problema 1

maximizar $f(x, y) = x * y$ s.a. $x^4 + y^2 \leq 3$ Notemos que no estamos ante un problema de programación convexa pero está muy cerca de serlo pues estamos ante un problema en el que tenemos que maximizar una función que no es cóncava, teniendo en cuenta una restricción convexa, de hecho es estrictamente convexa (su matriz hessiana es definida positiva, pues tiene todos sus autovalores positivos). Notemos que hemos escogido este ejemplo ya que es el que viene resuelto en el libro de Hillier y Lieberman (2006) y por tanto, conocemos el resultado que es (1,2). De esta manera podemos garantizar el buen funcionamiento del código programado.

```
(%i1) f:x^4+y^2;
                                2      4
                                y  + x
(%o1)
(%i2) H:hessian(f,[x,y]);
                                [      2      ]
                                [ 12 x   0  ]
                                [      ]
                                [   0    2  ]
(%o2)
(%i3) determinant(H);
                                2
                                24 x
(%o3)
(%i4) eigenvalues(H);
                                2
                                [[12 x , 2], [1, 1]]
(%o4)
```

Algoritmo

Iteración inicial: Sea x^0 una solución factible, r, θ, ϵ

Iteración k-ésima: Construimos la función

$$P(x, r) = f(x, y) - r \left(\sum_{j=1}^m \frac{1}{b_j - g_j(x)} + \sum_{i=1}^n \frac{1}{x_i} \right) \quad (28)$$

A continuación aplicamos el método del gradiente a la función (28). Luego, calculamos w que minimiza la función (29) y lo denotamos por λ .

$$h(w) = x^k + w \nabla P(x^k, r) \quad (29)$$

Obtenemos $x^k = P(x^{k-1} + \lambda \nabla P(x^{k-1}, r), r)$

Test de parada: Si se satisface el test de parada, es decir, si $\|x^k - x^{k-1}\| \leq \epsilon$ obtenemos que el máximo se obtiene en x^{k-1} . En cambio si no se satisface haremos $k=k+1$, $r = r\theta$.

```
cota = input ('Cota de error objetivo: ');
xi = input ('Valor inicial de la x: ');
yi = input ('Valor inicial de la y: ');

x0=[xi yi];

r=1;
tita=0.01;
i = 1; % contador de iteraciones
% Debo realizar un procedimiento iterativo mientras el error sea mayor que la cota
e=10000;
cota1=0.001;
syms x y m

while (e>cota)
    P=x*y-r*(1/(3-x^2-y)+(1/x)+(1/y));
    xnuevo=x0;
    fprintf('Punto inicial: %e %e \n', x0(1), x0(2));
    fprintf('En la iteración i: %e \n', i);
    fprintf('Valor de r: %e \n', r);

    e1=10000;
    while(e1>cota1)
        GP=jacobian(P,[x;y]);
        GPr=subs(GP,{x,y},{x0(1),x0(2)});
        F=subs(P,{x,y},{x0(1)+m*GPr(1),x0(2)+m*GPr(2)});
        DF=diff(F);
        lambda=min(double(solve(diff(subs(x*y-r*((1/(3-x^2-y)))+(1/x)+(1/y)),
            {x,y},{x0(1)+m*GPr(1),x0(2)+m*GPr(2)})),0)));
        F2=diff(DF);
        if double(subs(F2,{m},{lambda}))<0;
            x1=x0+lambda*GPr;
            e1= double(sqrt((x1(1)-x0(1))^2 +(x1(2)-x0(2))^2)) ;
            x0=x1;
        else
            fprintf('Se encontró la solución. Último valor calculado: %e %e \n', x0(1), x0(2));
            break
        end
    end
end
```

```

e= double(sqrt((x0(1)-xnuevo(1))^2 +(x0(2)-xnuevo(2))^2)) ;
i=i+1;
r=r*tita;
end
fprintf('Se encontró la solución. La solución hallada es: %e %e \n', x0(1), x0(2));

fprintf('El número de iteraciones utilizado es: %e \n', i);
fprintf('El test de parada es (cota 0.0001): %e \n', e);

sol=mifuncion_objetivo(x0(1),x0(2));
fprintf('Y el valor de la función en dicho punto es: %e \n', sol);

function z = mifuncion_objetivo(x,y)
z = x*y;

```

Solución

Se encontró la solución. La solución hallada es: 1.000656e+000 1.998677e+000 El test de parada es (cota 0.0001): 4.024693e-005 Y el valor de la función en dicho punto es: 1.999989e+000

```

Cota de error objetivo: 0.0001
Valor inicial de la x: 1
Valor inicial de la y: 1
Punto inicial: 1.000000e+000 1.000000e+000
En la iteración i: 1.000000e+000
Valor de r: 1.000000e+000
Punto inicial: 8.998649e-001 1.356955e+000
En la iteración i: 2.000000e+000
Valor de r: 1.000000e-002
Punto inicial: 9.868827e-001 1.925194e+000
En la iteración i: 3.000000e+000
Valor de r: 1.000000e-004
Punto inicial: 1.001122e+000 1.987728e+000
En la iteración i: 4.000000e+000
Valor de r: 1.000000e-006
Punto inicial: 1.000260e+000 1.998479e+000
En la iteración i: 5.000000e+000
Valor de r: 1.000000e-008
Punto inicial: 1.000620e+000 1.998659e+000
En la iteración i: 6.000000e+000
Valor de r: 1.000000e-010
Se encontró la solución. La solución hallada es: 1.000656e+000 1.998677e+000
El test de parada es (cota 0.0001): 4.024693e-005

```

Y el valor de la función en dicho punto es: 1.999989e+000

Problema 2

maximizar $f(x, y) = \ln(x+y)$ s.a. $x^4 + y^2 \leq 4$ Notemos que estamos ante un problema de programación convexa pues estamos ante un problema en el que tenemos que maximizar una función cóncava (estudiada en el segundo ejercicio), teniendo en cuenta una restricción convexa, de hecho es estrictamente convexa (su matriz hessiana es definida positiva, pues tiene todos sus autovalores positivos).

```
cota = input ('Cota de error objetivo: ');
xi = input ('Valor inicial de la x: ');
yi = input ('Valor inicial de la y: ');

x0=[xi yi];

r=1;
tita=0.01;
i = 1; % contador de iteraciones
% Debo realizar un procedimiento iterativo mientras el error sea mayor que la cota
e=10000;
cota1=0.001;
syms x y m

while (e>cota)
    P=log(x+y)-r*(1/(4-x^2-y)+(1/x)+(1/y));
    xnuevo=x0;
    fprintf('Punto inicial: %e %e \n', x0(1), x0(2));
    fprintf('En la iteración i: %e \n', i);
    fprintf('Valor de r: %e \n', r);

    e1=10000;
    while(e1>cota1)
        GP=jacobian(P,[x;y]);
        GPr=subs(GP,{x,y},{x0(1),x0(2)});
        F=subs(P,{x,y},{x0(1)+m*GPr(1),x0(2)+m*GPr(2)});
        DF=diff(F);
        lambda=min(double(solve(diff(subs(x*y-r*((1/(3-x^2-y)+(1/x)+(1/y))),
        {x,y},{x0(1)+m*GPr(1),x0(2)+m*GPr(2)})),0)));
        F2=diff(DF);
        if double(subs(F2,{m},{lambda}))<0;
            x1=x0+lambda*GPr;
            e1= double(sqrt((x1(1)-x0(1))^2 +(x1(2)-x0(2))^2)) ;
            x0=x1;
```

```

        else
            fprintf('Se encontró la solución. Último valor calculado: %e %e \n', x0(1), x0(2));
            break
        end
    end
    e= double(sqrt((x0(1)-xnuevo(1))^2 +(x0(2)-xnuevo(2))^2)) ;
    i=i+1;
    r=r*tita;
    end
    fprintf('Se encontró la solución. La solución hallada es: %e %e \n', x0(1), x0(2));

    fprintf('El número de iteraciones utilizado es: %e \n', i);
    fprintf('El test de parada es (cota 0.0001): %e \n', e);

    sol=mifuncion_objetivo(x0(1),x0(2));
    fprintf('Y el valor de la función en dicho punto es: %e \n', sol);

function z = mifuncion_objetivo(x,y)
z = log(x+y);

```

Solución

Se encontró la solución. La solución hallada es: 1.299449e+000 1.311421e+000 El test de parada es (cota 0.0001): 4.152266e-005 Y el valor de la función en dicho punto es: 9.596835e-001

```

Cota de error objetivo: 0.0001
Valor inicial de la x: 1
Valor inicial de la y: 1
Punto inicial: 1.000000e+000 1.000000e+000
En la iteración i: 1.000000e+000
Valor de r: 1.000000e+000
Punto inicial: 1.041566e+000 1.052153e+000
En la iteración i: 2.000000e+000
Valor de r: 1.000000e-002
Punto inicial: 1.266607e+000 1.278570e+000
En la iteración i: 3.000000e+000
Valor de r: 1.000000e-004
Punto inicial: 1.296186e+000 1.308158e+000
En la iteración i: 4.000000e+000
Valor de r: 1.000000e-006
Punto inicial: 1.299126e+000 1.311098e+000
En la iteración i: 5.000000e+000
Valor de r: 1.000000e-008

```


Punto inicial: 1.299419e+000 1.311392e+000
En la iteración i: 6.000000e+000
Valor de r: 1.000000e-010
Se encontró la solución. La solución hallada es: 1.299449e+000 1.311421e+000
El test de parada es (cota 0.0001): 4.152266e-005
Y el valor de la función en dicho punto es: 9.596835e-001

7. Ejercicio 7

Considerar el problema

$$\min(x-1)^2 + (y-1)^2 \quad (30)$$

sujeto a

$$-x + y = \frac{1}{2}, x + y \leq 1, y \geq 0 \quad (31)$$

7.1. Resolverlo gráficamente

A continuación representamos (Figura5) tres curvas de nivel de la función objetivo (30); representamos en rojo la curva de nivel $f(x_1; x_2) = 1$, en naranja la curva de nivel $f(x_1; x_2) = 2$ y en violeta la curva de nivel $f(x_1; x_2) = 3$. A continuación representamos en verde la primera de las restricciones(31), en azul la segunda y en negro los ejes de coordenadas. A la vista del gráfico, observamos que el mínimo se alcanza cuando $(x_1; x_2) = (0.5; 1)$ por tanto $f(0.25,0.75)=0.625$, que es el punto dónde se cortan las dos restricciones. **Implementación en máxima**

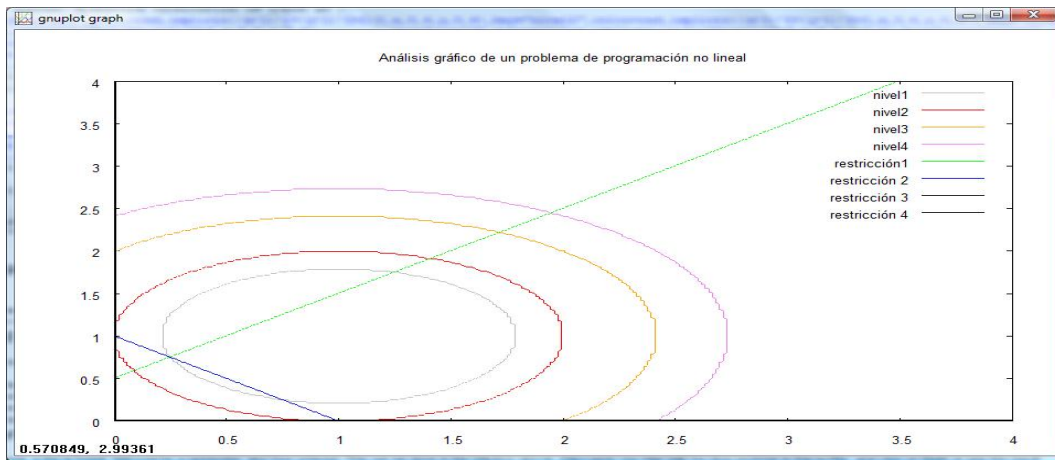


Figura 5: SOLUCIÓN gráfica

```
(%i1)load(draw)$
(%i2)draw2d(key="nivel1",color=grey,implicit((x-1)^2+(y-1)^2=0.625,x,0,4,y,0,4),
key="nivel2",color=red,implicit((x-1)^2+(y-1)^2=1,x,0,4,y,0,4),
key="nivel3",color=orange,implicit((x-1)^2+(y-1)^2=2,x,0,4,y,0,4),
key="nivel4",color=violet,implicit((x-1)^2+(y-1)^2=3,x,0,4,y,0,4),
key="restricci3n1",color=green,implicit(-x+y-0.5=0,x,0,4,y,0,4),
key="restricci3n 2",color=blue,implicit(x+y-1=0,x,0,4,y,0,4),
key="restricci3n 3",=black,implicit(x=0,x,0,4,y,0,4),
key="restricci3n 4",color=black,implicit(y=0,x,0,4,y,0,4),
title="Análisis gráfico de un problema de programación no lineal");
```

```

(%i3) g:x+y-1;
(%o3)          y + x - 1
(%i3) h:-x+y-1/2;
(%o3)          y - x - 1/2
(%i4) algsys([h,g],[x,y]);
(%4)          [[x = -1/4, y = 3/4]]

```

7.2. Comprobar la respuesta utilizando la teoría de Kuhn-Tucker

Sea f la función objetivo f , g y h las restricciones que son funciones polinómicas y además continuamente diferenciables(32). El punto $\bar{x} = (0,25,0,75)$ es un punto factible que resuelve el problema localmente, como se ve en la (Figura 5. A continuación aplicando la teoría de Kuhn-Tucker y con la ayuda de **xmaxima**, se verá que este punto es el único que resuelve el problema.

$$f(x, y) = (x - 1)^2 + (y - 1)^2 h(x) = -x + y - \frac{1}{2}g(x) = x + y - 1 \quad (32)$$

Implementación en máxima

```

(%i1) f:(x-1)^2+(y-1)^2;
(%o1)          (y - 1)^2 + (x - 1)^2
(%i2) g:x+y-1;
(%o2)          y + x - 1
(%i3) h:-x+y-0.5;
(%o3)          y - x - 0.5
(%i4) L:f+l*g+m*h;
(%o4)          1 (y + x - 1) + m (y - x - 0.5) + (y - 1)^2 + (x - 1)^2
(%i5) e1:diff(L,x);
(%o5)          2 (x - 1) - m + 1
(%i6) e2:diff(L,y);
(%o6)          2 (y - 1) + m + 1
(%i7) e3:diff(L,l);
(%o7)          y + x - 1
(%i8) e4:diff(L,m);
(%o8)          y - x - 0.5
(%i9) algsys([e1,e2,e3,e4],[x,y,l,m]);

```

```

(%o9)          1      3      1
          [[x = -, y = -, l = 1, m = - -]]
              4      4      2

(%i10) f,x=0.25,y=0.75;
(%o10)          0.625

```

Se ve fácilmente que menos el gradiente de la función f evaluado en el óptimo es combinación lineal de los gradientes de las restricciones evaluados en dicho óptimo, ya que: $-\nabla f(0,25,0,73) = a\nabla g(0,25,0,73) + b\nabla h(0,25,0,73) \Rightarrow (-1,5, -0,5) = a(1,1) + b(-1,1) \Rightarrow a = -1, b = 0,5$ El punto $\bar{x}$ es un punto factible que resuelve el problema localmente y además se acaba de ver que los vectores de los gradientes de las restricciones son linealmente independientes. **Implementación en máxima**

```

(%i11) f,x=0.25,y=0.75;
(%o11)          0.625
(%i12) df1:diff(f,x);
(%o12)          2 (x - 1)
(%i13) df2:diff(f,y);
(%o13)          2 (y - 1)
(%i14) dg1:diff(g,x);
(%o14)          1
(%i15) dg2:diff(g,y);
(%o15)          1
(%i16) dh1:diff(h,x);
(%o16)          - 1
(%i17) dh2:diff(h,y);
(%o17)          1
(%i18) df1,x=0.25,y=0.75;
(%o18)          - 1.5
(%i19) df2,x=0.25,y=0.75;
(%o19)          - 0.5

```

7.3. Encontrar la función dual $I(\lambda, \mu)$ y probar que es cóncava

El problema dual lagrangiano de nuestro problema inicial (30),(31) se define como

$$\max l(\lambda, \mu) = \min(x-1)^2 + (y-1)^2 + (x+y-1) - 0,5(y-x-1/2) \quad (33)$$

sujeto a

$$\lambda = 1 \geq 0 \quad (34)$$

Como en el problema primal se tiene que la función h (32) es lineal en términos de x e y ; g y f (32) son una funciones convexas. Los gradientes de g y h como vimos anteriormente son

linealmente independientes. Entonces entonces por el teorema de dualidad minimizar el problema primal es igual a maximizar el problema dual lagrangiano. Además la función $\mathbf{l}(\lambda, \mu)$ es cóncava sobre cualquier subconjunto del dominio. Por tanto qued probado $\mathbf{l}(\lambda, \mu)$ que es cóncava.

Probamos que f es convexa Todos los autovalores de la matriz Hessiana de la función f son positivos, por tanto f es estrictamente convexa y en particular convexa.

```
(%i20) Hf:hessian(f,[x,y]);
                                     [ 2  0 ]
(%o20)                               [      ]
                                     [ 0  2 ]
(%i21) eigenvalues(Hf);
(%o21)                               [[2], [2]]
```

8. Ejercicio 8

Haciendo uso del método símplex modificado para problemas de programación cuadrática, maximizar:

$$f(x_1, x_2) = 15x_1 + 30x_2 + 4x_1x_2 - 2x_1^2 - 4x_2^2$$

sujeto a

$$\begin{aligned}x_1 + 2x_2 &\leq 30 \\x_1 &\geq 0, x_2 \geq 0.\end{aligned}$$

SOLUCIÓN

Este problema equivale a:

$$\min f(x_1, x_2) = -15x_1 - 30x_2 - 4x_1x_2 + 2x_1^2 + 4x_2^2$$

sujeto a

$$\begin{aligned}x_1 + 2x_2 - 30 &\leq 0 \\x_1 &\geq 0, x_2 \geq 0.\end{aligned}$$

La función objetivo es convexa por ser suma de funciones convexas y las restricciones son convexas por ser lineales. Por tanto estamos ante un problema de programación cuadrática convexo. Usamos el método de Wolfe. La función de Lagrange es:

$$L((x_1, x_2), \lambda) = -15x_1 - 30x_2 - 4x_1x_2 + 2x_1^2 + 4x_2^2 + \lambda(x_1 + 2x_2 - 30) \quad (35)$$

Las condiciones de Kuhn-Tucker son:

$$-4x_2 + 4x_1 + \lambda - 15 = 0, 8x_2 - 4x_1 + 2\lambda - 30 = 0, 2x_2 + x_1 - 30 \leq 0, \lambda(x_1 + 2x_2 - 30) = 0, x_1, x_2 \geq 0 \quad (36)$$

Las soluciones del sistema anterior van a ser la solución óptima del problema. Las determinamos usando la fase 1 del método de las dos fases. Consideramos el siguiente problema de programación lineal auxiliar:

$$\min R_1 + R_2$$

sujeto a

$$\begin{aligned}-4x_2 + 4x_1 + \lambda - 15 + R_1 &= 0 \\8x_2 - 4x_1 + 2\lambda - 30 + R_2 &= 0 \\2x_2 + x_1 - 30 + s_1 &= 0 \\\lambda s_1 &= 0 \\\lambda, x_1, x_2, R_1, R_2, s_1 &\geq 0\end{aligned}$$

Ahora, se aplica el algoritmo del símplex con la salvedad de que en cada iteración se verifiquen las condiciones $\lambda s_1 = 0$.

		c_j	0	0	0	1	1	0
Base	x_B	c_B	x_1	x_2	λ	R_1	R_2	s_1
R_1	15	1	4	-4	1	1	0	0
R_2	30	1	-4	8	2	0	1	0
s_1	30	0	1	2	0	0	0	1
z, z_j			0	4	3	1	1	0
$z_j - c_j$			0	0	3	0	0	0

Cuadro 1: Primera tabla del simplex

		c_j	0	0	0	1	1	0
Base	x_B	c_B	x_1	x_2	λ	R_1	R_2	s_1
R_1	30	1	2	0	2	1	0.5	1
x_2	3.75	0	-0.5	1	0.25	1	0.125	0
s_1	22.5	0	2	0	-0.5	0	-0.25	1
z, z_j			2	0	2	1	0.5	0
$z_j - c_j$			2	0	2	0	-0.5	0

Cuadro 2: Segunda tabla del simplex

En (Tabla 1) observamos que la solución de esta iteración satisface $\lambda s_1 = 0$, pero como no todos los $z_j - c_j$ de las variables que no están en la base son menores o iguales que cero, tenemos que calcular la siguiente tabla del simplex. Si aplicamos los criterios del simplex, debe salir de la base R_2 y entrar x_2 . En (Tabla 2) observamos que la solución de esta iteración satisface $\lambda s_1 = 0$, pero como no todos los $z_j - c_j$ de las variables que no están en la base son menores o iguales que cero, tenemos que calcular la siguiente tabla del simplex. Si aplicamos los criterios del simplex, debe salir de la base s_1 y entrar x_1 . En (Tabla 3) observamos que la solución de esta iteración

		c_j	0	0	0	1	1	0
Base	x_B	c_B	x_1	x_2	λ	R_1	R_2	s_1
R_1	7.5	1	0	0	2.5	1	0.75	-1
x_2	9.375	0	0	1	0.125	0	0.0625	0.25
x_1	11.25	0	1	0	-0.25	0	-0.125	0.5
z, z_j			0	0	2.5	1	0.75	-1
$z_j - c_j$			0	0	2.5	0	-0.25	-1

Cuadro 3: Tercera tabla del simplex

satisface $\lambda s_1 = 0$, pero como no todos los $z_j - c_j$ correspondientes a las variables que no están en la base son menores o iguales que cero, tenemos que calcular la siguiente tabla del simplex.

Si aplicamos los criterios de entrada y salida del simplex, debe salir de la base R_1 y entrar λ . En (Tabla 4) observamos que la solución de esta iteración satisface $\lambda s_1 = 0$ y además todos los

	c_j	0	0	0	1	1	0	
Base	x_B	c_B	x_1	x_2	λ	R_1	R_2	s_1
λ	3	0	0	0	1	0.4	0.3	0.4
x_2	9	0	0	1	0	0.05	-0.025	0.3
x_1	12	0	1	0	0	0.1	-0.05	-0.4
z, z_j			0	0	0	0	0	0
$z_j - c_j$			0	0	0	-1	-1	0

Cuadro 4: Cuarta tabla del simplex

$z_j - c_j$ de las variables que no están en la base son menores o iguales que cero. Por tanto, hemos encontrado la solución maximal ($x_1 = 12, x_2 = 9, \lambda = 3, R_1 = 0, R_2 = 0, s_1 = 0$)

Con Maxima comprobamos los resultados anteriores;

```
(%i1) f:15*x+30*y+4*x*y-2*x^2-4*y^2;
                                2                2
(%o2)      - 4 y  + 4 x y + 30 y - 2 x  + 15 x
(%i3) f,x=12,y=9;
(%o3)      270
(%i4) f,x=12.5,y=9.5;
(%o4)      274.0
(%i5) f,x=11,y=9;
(%o5)      265
(%i6) load(draw)$
(%i7) draw2d(key="nivel1",color=red,implicit(15*x+30*y+4*x*y-2*x^2-4*y^2=270,x,6,20,y,6,20),
key="restricción1",color=green,implicit(x+2*y-30=0,x,0,15,y,0,15),
title="Análisis gráfico de un problema de programación no lineal");
```

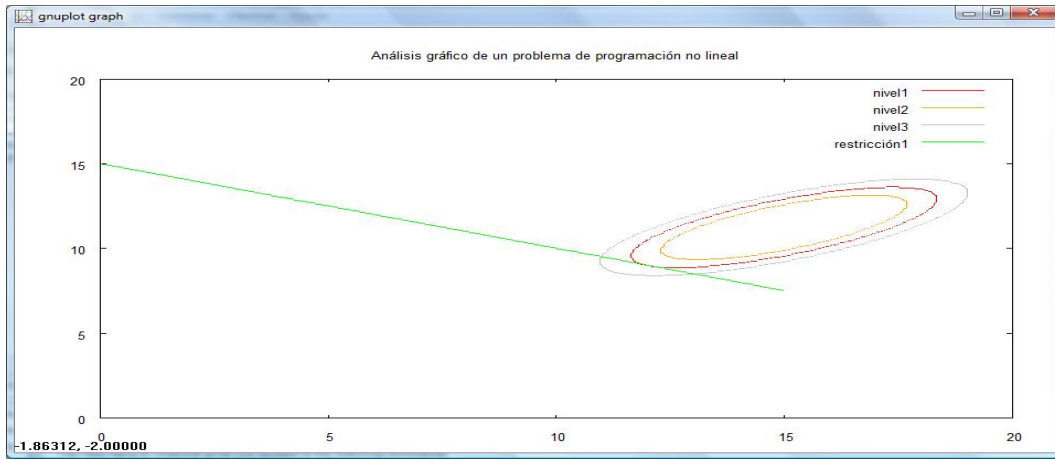



Figura 6: Solución gráfica

9. Ejercicio 9

Un fabricante desea minimizar el coste de los contenedores que produce así como el peso de los mismos. Se encuentra ante un problema de decisión motivado por la mayor carestía de los materiales más ligeros.

Los contenedores han de ser de base cuadrada abiertos, por la base de arriba y tener una capacidad o volumen de $10m^3$. El material con el que va a construir la parte inferior cuesta $6\text{euros}/m^2$ y pesa $4kg/m^2$. El material con el que se van a construir las caras del contenedor cuestan $3\text{euros}/m^2$ y pesa $8kg/m^2$.

Plantear por medio de un modelo de programación biobjetivo no lineal y realizar un busqueo de resultados en el espacio de objetivos.

SOLUCIÓN En primer lugar, planteamos el problema mediante un modelo de programación biobjetivo no lineal.

- Denotamos por lado de la base, x y denatamos la altura del contenedor por y .
- Primera función objetivo: Minimizar el coste de los contenedores

$$\min 6x^2 + (4 * 3)xy \equiv \min 6x^2 + 12xy \quad (37)$$

- Segunda función objetivo: Minimizar el peso de los contenedores

$$\min 4x^2 + (4 * 8)xy \equiv \min 4x^2 + 32xy \quad (38)$$

- Restricción: El volumen de los contenedores es $10m^3$

$$x^2y = 10 \quad (39)$$

Por tanto, estamos ante el siguiente problema biobjetivo:

$$\min f_1 = \min 6x^2 + 12xy$$

$$\min f_2 = \min 4x^2 + 32xy$$

s.a

$$x^2y = 10$$

$$x, y \geq 0$$

A continuación, para calcular los puntos eficientes del problema, plantearemos dos problemas:
1.- Uno primero para obtener el óptimo para el coste (37). 2.- Y otro para obtener el óptimo para el peso (38).

Subproblema 1

$$\min f_1 = \min 6x^2 + 12xy$$

s.a

$$x^2y = 10$$

$$x, y \geq 0$$

Para calcular la solución se ejecuta el siguiente código en **MAPLE**:

```
> with(Optimization);
> NLPSolve(6*x^2+12*x*y, {x^2*y = 10}, assume = nonnegative);
[83.5485990050292884, [x = 2.15443469004186694, y = 2.15443469001188914]]
```

Por tanto, la solución obtenida para es **x=2.15 m,y=2.15 m** y el coste del contenedor es de **83.55 Euros**. Si sustituimos estos valores en la función objetivo relativa al peso f_2 (38), tenemos que $f_2 = 4(2,15)^2 + 32 * (2,15) * (2,15) = 166,41kg$

Subproblema 2

$$\min f_2 = \min 4x^2 + 32xy$$

s.a

$$x^2y = 10$$

$$x, y \geq 0$$

Para calcular la solución se ejecuta el siguiente código en **MAPLE**:

```
> with(Optimization);
> NLPSolve(4*x^2+32*x*y, {x^2*y = 10}, assume = nonnegative);
[140.352851434216234, [x = 3.41995189395169152, y = .854987973039187499]]
```

Por tanto, la solución obtenida para es $x=3.42 \text{ Kg}, y=0.85 \text{ Kg}$, el peso del contenedor es de **140.35 Kg**. Si sustituimos estos valores en la función objetivo relativa al coste f_2 (38), tenemos que $f_1 = 6(3.42)^2 + 12 * (3.42) * (0.85) = 105.06 \text{ Euros}$

De estos dos subproblemas podemos concluir que el coste del contenedor varía entre 83.55 y 105.06 Eur. y el peso entre 140.35 y 166.41kg. Por otra parte, concluimos que x varía entre 2.15 y 3.42 e y entre 0.85 y 2.15. Para calcular el conjunto eficiente podemos volver a resolver el **Subproblema 2** añadiendo restricciones a x e y en función de estos límites. De esta forma surge,

el **Subproblema 3**

Subproblema 3

$$\min f_2 = \min 4x^2 + 32xy$$

s.a

$$f_1 = 6x^2 + 12xy \leq K$$

$$x^2y = 10$$

$$2.15 \leq x \leq 3.42$$

$$0.85 \leq y \leq 2.15$$

Dónde K toma valores en $[83.55, 105.06]$. Entonces, como $(105.06 - 83.55)/9 = 2.39$, resolveremos el problema 10 veces incrementando en 2.39 Eur el coste de los contenedores de tal forma que K recorra todo el intervalo $[83.55, 105.06]$. Por tanto, los sucesivos costes serán $[83.55, 85.94, 88.33, 90.72, 93.11, 95.5, 97.89, 100.28, 102.67, 105.06]$.

Para calcular la solución se ejecuta el siguiente código en MAPLE:

```
>with(Optimization);
> NLPSolve(4*x^2+32*x*y, {x >= 2.15, y >= .85, x^2*y = 10,
x <= 3.42, y <= 2.15, 6*x^2+12*x*y <= 83.55}, assume = nonnegative);
NLPSolve(4*x^2+32*x*y, {x >= 2.15, y >= .85, x^2*y = 10,
x <= 3.42, y <= 2.15, 6*x^2+12*x*y <= 85.94}, assume = nonnegative);
NLPSolve(4*x^2+32*x*y, {x >= 2.15, y >= .85, x^2*y = 10,
x <= 3.42, y <= 2.15, 6*x^2+12*x*y <= 88.33}, assume = nonnegative);
NLPSolve(4*x^2+32*x*y, {x >= 2.15, y >= .85, x^2*y = 10,
x <= 3.42, y <= 2.15, 6*x^2+12*x*y <= 90.72}, assume = nonnegative);
NLPSolve(4*x^2+32*x*y, {x >= 2.15, y >= .85, x^2*y = 10,
x <= 3.42, y <= 2.15, 6*x^2+12*x*y <= 93.11}, assume = nonnegative);
NLPSolve(4*x^2+32*x*y, {x >= 2.15, y >= .85, x^2*y = 10,
x <= 3.42, y <= 2.15, 6*x^2+12*x*y <= 95.5}, assume = nonnegative);
NLPSolve(4*x^2+32*x*y, {x >= 2.15, y >= .85, x^2*y = 10,
x <= 3.42, y <= 2.15, 6*x^2+12*x*y <= 97.89}, assume = nonnegative);
NLPSolve(4*x^2+32*x*y, {x >= 2.15, y >= .85, x^2*y = 10,
x <= 3.42, y <= 2.15, 6*x^2+12*x*y <= 100.28}, assume = nonnegative);
NLPSolve(4*x^2+32*x*y, {x >= 2.15, y >= .85, x^2*y = 10,
```

```

x <= 3.42, y <= 2.15, 6*x^2+12*x*y <= 102.67}, assume = nonnegative);
NLPsolve(4*x^2+32*x*y, {x >= 2.15, y >= .85, x^2*y = 10,
x <= 3.42, y <= 2.15, 6*x^2+12*x*y <= 105.06}, assume = nonnegative)
[166.643205326090652, [x = 2.16326902845915514, y = 2.13687409861666522]]
[151.824479374225263, [x = 2.53884576725835620, y = 1.55141277288271496]]
[147.477965546572506, [x = 2.70906966072755573, y = 1.36257261063000090]]
[144.874506613525966, [x = 2.84378699077002528, y = 1.23653346294105515]]
[143.154590813733876, [x = 2.95999356022887916, y = 1.14134901296656799]]
[141.988260605414553, [x = 3.06428901570259970, y = 1.06497779117009972]]
[141.207774495895791, [x = 3.16006626386357060, y = 1.00140007827768773]]
[140.714354438922953, [x = 3.24934581742246520, y = 0.947126812284783570]]
[140.444201086739781, [x = 3.3344748446118055, y = 0.899938361557224508]]
Error, (in Optimization:-NLPsolve) no improved point could be found

```

A continuación, se muestran las soluciones en una tabla.

	Lado (m)	Altura (m)	Coste (Euros)	Peso (Kg)
1	2.13	2.16	83.55	166.54
2	1.55	2.54	85.94	151.82
3	2.71	1.36	88.33	147.48
4	2.84	1.24	90.72	144.87
5	2.96	1.14	93.11	143.15
6	3.06	1.06	95.5	141.99
7	3.16	1.00	97.89	141.21
8	3.25	0.95	100.28	140.71
9	3.33	0.89	102.67	140.44
10	-	-	-	-

Cuadro 5: Puntos eficientes

10. Ejercicio 10

Sea el problema biobjetivo:

$$\max Z_1 = 5X_1 - 2X_2$$

$$\max Z_2 = -X_1 + 4X_2$$

s.a

$$-X_1 + X_2 \leq 3$$

$$X_1 + X_2 \leq 8$$

$$X_1 \leq 6, X_2 \leq 4$$

$$X_1, X_2 \geq 0$$

Utilizando máxima:

10.1. Representar graficamente la región factible X.

En (Figura 7) dibujamos las restricciones en negro, las curvas de nivel correspondientes a la primera función objetivo están representadas en rojo y en naranja la que toma menor valor(-3); por lo que podemos observar que la primera función objetivo aumenta a medida que avanzamos hacia la derecha. Por otra parte, las curvas de nivel correspondientes a la segunda función objetivo están representadas en verde y en amarillo la que toma menor valor(-5); por lo que podemos observar que la segunda función objetivo aumenta a medida que avanzamos hacia la izquierda. A la vista de estas conclusiones observamos que el máximo se alcanza en $X_1 = 4, X_2 = 4$

Implementación en maxima

```
(%i1)load(draw)$
(%i2)draw2d(line_width=7,color=orange,implicit(5*x-2*y=-3,x,-1,10,y,-1,10),
color=red,implicit(5*x-2*y=0,x,-1,10,y,-1,10),implicit(5*x-2*y=2,x,-1,10,y,-1,10),
implicit(5*x-2*y=7,x,-1,10,y,-1,10),implicit(5*x-2*y=10,x,-1,10,y,-1,10),
color=green,implicit(-x+4*y=0,x,-1,10,y,-1,10),implicit(-x+4*y=9,x,-1,10,y,-1,10),
implicit(-x+4*y=4,x,-1,10,y,-1,10),color=yellow,implicit(-x+4*y=-5,x,-1,10,y,-1,10),
color=black,implicit(-x+y=3,x,-1,10,y,-1,10),implicit(x+y=8,x,-1,10,y,-1,10),
implicit(x=6,x,-1,10,y,-1,10),implicit(y=4,x,-1,10,y,-1,10),implicit(x=0,x,-1,10,y,-1,10))
```

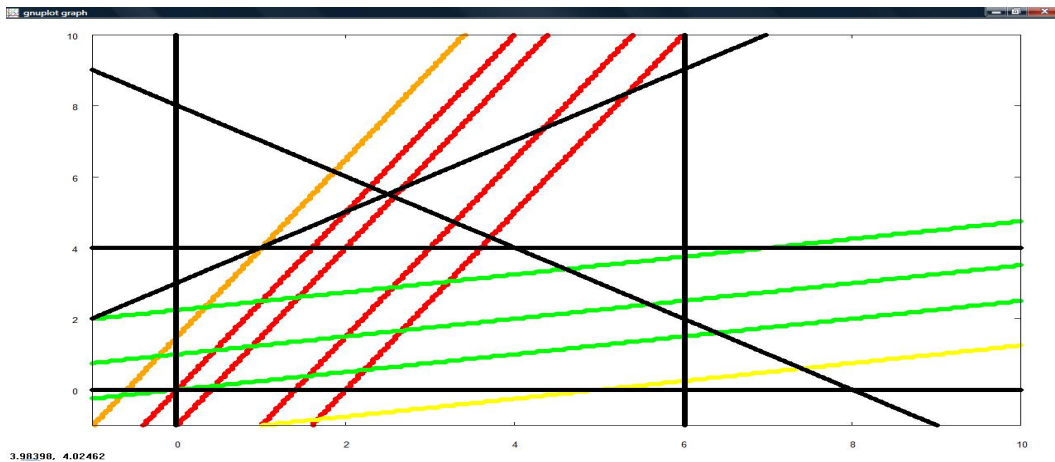


Figura 7: Región factible

10.2. Representar el espacio de objetivos $Z(X)$.

Implementación en máxima

```
(%i1) e1:-x+y=3$
(%i2) e2:x+y=8$
(%i3) e3:x=6$
(%i4) e4:y=4$
(%i5) e5:x=0$
(%i6) e6:y=0$
(%i7) algsys([e1,e5],[x,y]);
(%o7) [[x = 0, y = 3]]
(%i8) algsys([e1,e4],[x,y]);
(%o8) [[x = 1, y = 4]]
(%i9) algsys([e2,e4],[x,y]);
(%o9) [[x = 4, y = 4]]
(%i10) algsys([e2,e3],[x,y]);
(%o10) [[x = 6, y = 2]]
(%i10) algsys([e3,e6],[x,y]);
(%o10) [[x = 6, y = 0]]
(%i11) z1:5*x-2*y;
(%i12) z2:-x+4*y;
(%i13) p1a:z1,x=0,y=3;
(%o13) - 6
(%i14) p1b:z2,x=0,y=3;
(%o14) 12
(%i15) p2a:z1,x=1,y=4;
(%o15) - 3
```

```
(%i16) p2b:z2,x=1,y=4;
(%o16)
15
(%i17) p3a:z1,x=4,y=4;
(%o17)
12
(%i18) p3b:z2,x=4,y=4;
(%o18)
12
(%i19) p4a:z1,x=6,y=2;
(%o19)
26
(%i20) p4b:z2,x=6,y=2;
(%o20)
2
(%i21) p5a:z1,x=6,y=0;
(%o21)
30
(%i21) p5b:z2,x=6,y=0;
(%o22)
- 6
(%i23) load(draw)$
(%i24) draw2d(xrange = [-15,35],yrange = [-15,35],point_size = 3,
point_type = circle,color = blue,points_joined = true,line_type = dots,
point_type = circle,color = red,line_width = 7,
points([[0,0],[p1a,p1b],[p2a,p2b],[p3a,p3b],[p4a,p4b],[p5a,p5b],[0,0]])) );
```

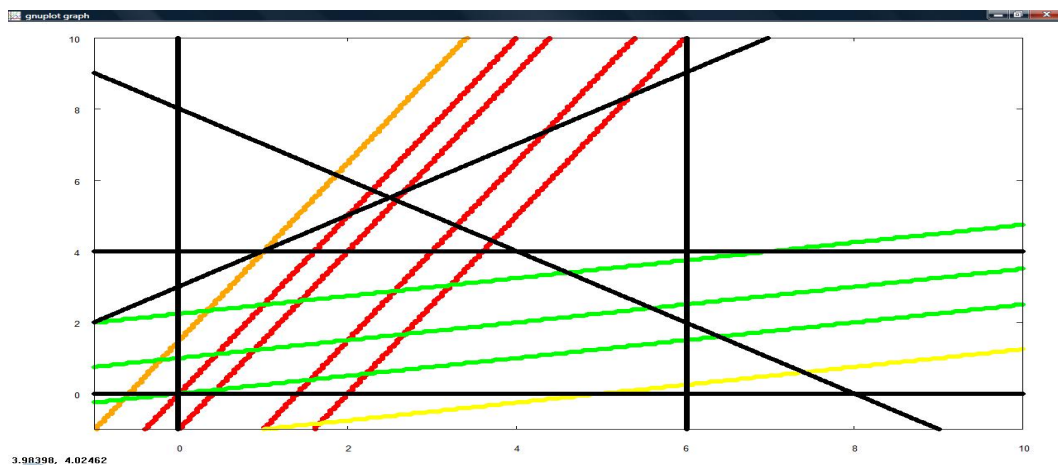


Figura 8: Espacio de objetivos

10.3. Determinat gráficamente los puntos eficientes por medio de los contornos de los objetivos.

Los puntos eficientes se manifiestan graficamente como aquellos para los cuales ningún punto factible distinto pertenece a la región acotada por los contornos de las funciones objetivos que pasan por el punto y contienen las soluciones con igual o menor valor de todos los funciones objetivos. Por tanto, la frontera eficiente en el espacio de decisiión (región factible X) la forman el segmento que une los puntos $(4,4)$ y $(6,2)$ y el segmento que une los puntos $(6,2)$ y $(6,0)$. Anañogamente, la frontera eficiente en el espacio de resultados de las funciones objetivo la forman el segmento que une los puntos $(12,12)$ y $(26,2)$ y el segmento que une los puntos $(26,2)$ y $(30,-6)$.

10.4. Comprobar si se verifica el recíproco del segundo teorema de Yu and Zeleni.

Segundo teorema de Yu and Zeleni

Si el conjunto factible es acotado, por lo que es la envoltura convexa de un número finito de vértices $x^1, \dots, x^r$ entonces cualquier solución eficiente $\hat{x}$ está en la envoltura convexa de los vértices eficientes.

En primer lugar, observamos en (Figura 7 7) que la envoltura convexa es el triángulo formado por los vértices $(4,4)$, $(6,2)$ y $(6,0)$, puesto que la envoltura convexa es el menor convexo que une a estos tres vértices. Por tanto, el punto $(5,2)$ pertenece a la envoltura convexa, por estar comprendido en el segmento que une a $(4,4)$ y $(6,0)$. Pero el punto $(5,2)$ no es eficiente pues este punto no domina a $(6,2)$ pues $Z_1 = 21, Z_2 = 3$ en $(5,2)$ y $Z_1 = 26, Z_2 = 2$ en $(6,2)$, por tanto el punto $(5,2)$ no proporciona valores al menos tan buenos en todas las funciones objetivos y estrictamente mejor en una.

Luego $(5,2)$ es un punto que pertenece a la envoltura convexa de los vértices eficientes y no es una solución eficiente. Por lo que concluimos que no se verifica el recíproco del segundo teorema de Yu and Zeleni.

11. Ejercicio 11

Minimizar

$$J = \sum_{i=0}^N (qx_i^2 + r * u_i^2) + qx_3^2$$

sujeto a la condición dinámica de estado

$$x_{i+1} = ax_i + bu_i, i = 0, \dots, N - 1$$

dónde $N=2, q=2, r=4, a=1$ y $b=0.5$

SOLUCIÓN Nos encontramos frente a un problema de control discreto cuadrático lineal. Para resolver este problema de control como un problema de programación no lineal, se tratan todas las variables de estado x_i y todas las variables de control u_i como variables de decisión, es decir, se define el vector de decisión como $\mathbf{x} = (x_1, x_2, x_3, u_0, u_1, u_2)$. Resolveremos con nuestro problema con $N=2, q=2, r=4, a=1$ y $b=0.5$ con **xmaxima**

Los primeros, definen el objetivo y las condiciones dinámicas del estado

```
(%i1) c0:2*x0^2+4*u0^2$
(%i2) c1:2*x1^2+4*u1^2$
(%i3) c2:2*x2^2+4*u2^2$
(%i4) f:c0+c1+c2+2*x3^2$
(%i5) h1:x1-x0-0.5*u0$
(%i6) h2:x2-x1-0.5*u1$
(%i7) h3:x3-x2-0.5*u2$
```

Se introduce la Lagrangiana:

```
(%i8) L:f+m1*h1+m2*h2+m3*h3$
```

y se diferencia con respecto a todas las variables de decisión del vector $\mathbf{x}$:

```
(%i9) Lu0:diff(L,u0,1)$
(%i10) Lu1:diff(L,u1,1)$
(%i11) Lu2:diff(L,u2,1)$
(%i12) Lx1:diff(L,x1,1)$
(%i13) Lx2:diff(L,x2,1)$
(%i14) Lx3:diff(L,x3,1)$
```

Las variables de decisión son calculadas en términos de los dos multiplicadores de Lagrange $m_i, i = 1, 2, 3$.

```
(%i15) solxu:algsys([Lx1,Lx2,Lx3,Lu0,Lu1,Lu2],[x1,x2,x3,u0,u1,u2]);
(%o15) [[x1 = -----, x2 = -----, x3 = - --, u0 = --, u1 = --, u2 = --]]
          4              4              4          16          16          16
```

La dual Lagrangiana $\mathbf{l}(m_1, m_2, m_3)$ se calcula y maximiza tras sustituir $\mathbf{x}(m_1, m_2, m_3)$ en L:

```
(%i16) e1:L,x1=(m2-m1)/4,x2=(m3-m2)/4,x3=-m3/4,u0=m1/16,u1=m2/16, u2=m3/16$
(%i17) elm1:diff(e1,m1,1)$
(%i18) elm2:diff(e1,m2,1)$
(%i19) elm3:diff(e1,m3,1)$
```

Se encuentran los multiplicadores de Lagrange en términos de los parámetros del modelo como sigue:

```
(%i20) algsys([elm1,elm2, elm3],[m1,m2,m3]);
(%o20)      [[m1 = -  $\frac{7200 x_0}{937}$ , m2 = -  $\frac{4352 x_0}{937}$ , m3 = -  $\frac{2048 x_0}{937}$ ]]
```

Finalmente, la solución para el problema completo, o sea, los valores óptimos de $x_1, x_2, x_3, u_0, u_1, u_2$ se calculan en términos de los parámetros del modelo, obteniéndose así:

```
(%i21) sol:solxu[1];
(%o21)      [x1 =  $\frac{m_2 - m_1}{4}$ , x2 =  $\frac{m_3 - m_2}{4}$ , x3 =  $-\frac{m_3}{4}$ , u0 =  $\frac{m_1}{16}$ , u1 =  $\frac{m_2}{16}$ , u2 =  $\frac{m_3}{16}$ ]
(%i22) est1:sol[1];
(%o22)      x1 =  $\frac{m_2 - m_1}{4}$ 
(%i23) est2:sol[2];
(%o23)      x2 =  $\frac{m_3 - m_2}{4}$ 
(%i24) est3:sol[3];
(%o24)      x3 =  $-\frac{m_3}{4}$ 
(%i25) cont0:sol[4];
(%o25)      u0 =  $\frac{m_1}{16}$ 
(%i26) cont1:sol[5];
(%o26)      u1 =  $\frac{m_2}{16}$ 
(%i27) cont2:sol[6];
```

```

                                m3
(%o27)          u2 = --
                                16

(%i28) aux:algsys([elm1,elm2, elm3],[m1,m2,m3]);
                                7200 x0      4352 x0      2048 x0
(%o28)          [[m1 = - ----, m2 = - ----, m3 = - ----]]
                                937          937          937

(%i29) ll:aux[1];
                                7200 x0      4352 x0      2048 x0
(%o29)          [m1 = - ----, m2 = - ----, m3 = - ----]
                                937          937          937

(%i30) m1:ll[1];
                                7200 x0
(%o30)          m1 = - ----
                                937

(%i31) m2:ll[2];
                                4352 x0
(%o31)          m2 = - ----
                                937

(%i32) m3:ll[3];
                                2048 x0
(%o32)          m3 = - ----
                                937

(%i33) est1,m2,m1;
                                712 x0
(%o33)          x1 = ----
                                937

(%i34) est2,m3,m2;
                                576 x0
(%o34)          x2 = ----
                                937

(%i35) est3,m3;
                                512 x0
(%o35)          x3 = ----
                                937

(%i36) cont0,m1;
                                450 x0
(%o36)          u0 = - ----
                                937

(%i37) cont1,m2;
                                272 x0

```

```
(%o37)                                u1 = - -----
                                           937
(%i38) cont2,m3;
                                           128 x0
(%o38)                                u2 = - -----
                                           937
```

Luego, la solución óptima viene dada por $\mathbf{x} = (x_1 = \frac{712}{937}x_0, x_2 = \frac{576}{937}x_0, x_3 = \frac{512}{937}x_0, u_0 = -\frac{450}{937}x_0, u_1 = -\frac{272}{937}x_0, u_2 = -\frac{128}{937}x_0)$ Una revisión cuidadosa de la estructura del vector de decisión óptimo indica que las variables de control óptimas u_0, u_1, u_2 son función lineal de los estados x_1, x_2, x_3 en cada instante de tiempo.

12. Ejercicio 12

Un fabricante tiene 3 tipos de semiconductores de silicio (wafers) en existencias para fabricar 3 variedades de circuitos integrados (chips) de ordenador. Algún semiconductor no se puede usar para algún circuito, pero hay dos alternativas para cada uno de ellos. La tabla siguiente muestra: costes unitarios (C), cantidad disponible de cada tipo de semiconductor (S), necesidades de cada circuito (N) y una puntuación entre 0 y 10 para la adecuación de cada tipo de semiconductor a cada circuito (P-C).

Wafer	P-C			C	S
	1	2	3		
1	7	8	-	15	500
2	10	-	6	25	630
3	-	10	10	30	710
N	440	520	380		

Cuadro 6: Resumen de los valores

12.1. Establecer y resolver una sucesión de programas lineales para calcular una solución lexicográfica del modelo considerando los objetivos en el orden dado.

En primer lugar, modelaremos el problema por medio de un programa lineal multiobjetivo, en el que trataremos interesa reducir al mínimo el coste total de los wafers usados maximizando las puntuaciones totales.

- Este problema tratará de reducir costes: $\min f_1 = \min 15x_{1,1} + 15x_{1,2} + 25x_{2,1} + 25x_{2,3} + 30x_{3,2} + 30x_{3,3}$
- Maximizar las puntuaciones totales: $\max f_2 = \max 7x_{1,1} + 8x_{1,2} + 10x_{2,1} + 6x_{2,3} + 10x_{3,2} + 10x_{3,3}$

sujeto a:

- Tres restricciones de desigualdad que hacen referencia a la cantidad disponible de cada tipo de semiconductor: $x_{1,1} + x_{1,2} \leq 500$, $x_{2,1} + x_{2,3} \leq 630$ y $x_{3,2} + x_{3,3} \leq 710$
- Tres restricciones de igualdad correspondientes a las necesidades de cada circuito: $x_{1,1} + x_{2,1} = 440$, $x_{1,2} + x_{3,2} = 520$ y $x_{2,3} + x_{3,3} = 380$
- Condición de no-negatividad: $x_{i,j} \geq 0$, $i, j = 1, 2, 3$.

Luego estamos ante el siguiente problema

$$\min f_1 = \min 15x_{1,1} + 15x_{1,2} + 25x_{2,1} + 25x_{2,3} + 30x_{3,2} + 30x_{3,3}$$

$$\max f_2 = \max 7x_{1,1} + 8x_{1,2} + 10x_{2,1} + 6x_{2,3} + 10x_{3,2} + 10x_{3,3}$$

sujeto a

$$x_{1,1} + x_{1,2} \leq 500$$

$$x_{2,1} + x_{2,3} \leq 630$$

$$x_{3,2} + x_{3,3} \leq 710$$

$$x_{1,1} + x_{2,1} = 440$$

$$x_{1,2} + x_{3,2} = 520$$

$$x_{2,3} + x_{3,3} = 380$$

$$x_{i,j} \geq 0, i, j = 1, 2, 3$$

Etapas 1

$$\min f_1 = \min 15x_{1,1} + 15x_{1,2} + 25x_{2,1} + 25x_{2,3} + 30x_{3,2} + 30x_{3,3}$$

sujeto a

$$x_{1,1} + x_{1,2} \leq 500$$

$$x_{2,1} + x_{2,3} \leq 630$$

$$x_{3,2} + x_{3,3} \leq 710$$

$$x_{1,1} + x_{2,1} = 440$$

$$x_{1,2} + x_{3,2} = 520$$

$$x_{2,3} + x_{3,3} = 380$$

$$x_{i,j} \geq 0, i, j = 1, 2, 3$$

La solución obtenida mediante el solver de Excell es:

- $x_{1,1} = 51$
- $x_{1,2} = 449$
- $x_{2,1} = 389$
- $x_{2,3} = 241$
- $x_{3,2} = 71$
- $x_{3,3} = 139$

- Coste, 29550 Euros.
- Puntuación total, 11385.

Etapla 2

$$\max f_2 = \max 7x_{1,1} + 8x_{1,2} + 10x_{2,1} + 6x_{2,3} + 10x_{3,2} + 10x_{3,3}$$

sujeto a

$$x_{1,1} + x_{1,2} \leq 500$$

$$x_{2,1} + x_{2,3} \leq 630$$

$$x_{3,2} + x_{3,3} \leq 710$$

$$x_{1,1} + x_{2,1} = 440$$

$$x_{1,2} + x_{3,2} = 520$$

$$x_{2,3} + x_{3,3} = 380$$

$$x_{i,j} \geq 0, i, j = 1, 2, 3$$

La solución obtenida mediante el solver de Excell es:

- $x_{1,1} = 0$
- $x_{1,2} = 190$
- $x_{2,1} = 440$
- $x_{2,3} = 0$
- $x_{3,2} = 330$
- $x_{3,3} = 380$
- Coste, 35150 Euros.
- Puntuación total, 13020.

12.2. Establecer y resolver un programa lineal con suma ponderada de objetivos que pondere al primero el doble que al segundo.

Como tenemos que minimizar la primera función objetivo, f_1 y maximizar la segunda función objetivo, f_2 , hacemos $f_1 - f_2$ con el fin de estar ante un problema de minimizar y aplicamos la ponderación $2f_1 - f_2$. Por tanto, el nuevo problema es el siguiente,

$$\min 2*(f_1) - f_2 = \min 2*(15x_{1,1} + 15x_{1,2} + 25x_{2,1} + 25x_{2,3} + 30x_{3,2} + 30x_{3,3}) - (7x_{1,1} + 8x_{1,2} + 10x_{2,1} + 6x_{2,3} + 10x_{3,2} + 10x_{3,3})$$

sujeto a

$$\begin{aligned}x_{1,1} + x_{1,2} &\leq 500 \\x_{2,1} + x_{2,3} &\leq 630 \\x_{3,2} + x_{3,3} &\leq 710 \\x_{1,1} + x_{2,1} &= 440 \\x_{1,2} + x_{3,2} &= 520 \\x_{2,3} + x_{3,3} &= 380 \\x_{i,j} &\geq 0, i, j = 1, 2, 3\end{aligned}$$

La solución obtenida mediante el solver de Excell es:

- $x_{1,1} = 0$
- $x_{1,2} = 500$
- $x_{2,1} = 440$
- $x_{2,3} = 190$
- $x_{3,2} = 20$
- $x_{3,3} = 190$
- Coste, 29550 Euros.
- Puntuación total, 11640.
- $2f_1 - f_2$, 47460.

12.3. Supóngase que se establece en el coste una meta de 30.000 y en la puntuación de 13.000. Plantear un problema de programación por metas que minimice la suma no ponderada de no cumplimiento de las metas.

Si asumimos una meta de 30000 en el primer objetivo y otra de 13000 en el segundo, introduciendo las correspondientes variables de deficiencia, resulta el siguiente modelo de programación por metas, cuyo objetivo es la suma ponderada de dichas variables:

$$\text{mind}_1 + d_2$$

sujeto a

$$15x_{1,1} + 15x_{1,2} + 25x_{2,1} + 25x_{2,3} + 30x_{3,2} + 30x_{3,3} - d_1 \leq 30000$$

$$7x_{1,1} + 8x_{1,2} + 10x_{2,1} + 6x_{2,3} + 10x_{3,2} + 10x_{3,3} + d_2 \geq 13000$$

$$x_{1,1} + x_{1,2} \leq 500$$

$$x_{2,1} + x_{2,3} \leq 630$$

$$x_{3,2} + x_{3,3} \leq 710$$

$$x_{1,1} + x_{2,1} = 440$$

$$x_{1,2} + x_{3,2} = 520$$

$$x_{2,3} + x_{3,3} = 380$$

$$x_{i,j} \geq 0, i, j = 1, 2, 3$$

La solución obtenida mediante el solver de Excell es:

- $x_{1,1} = 0$
- $x_{1,2} = 500$
- $x_{2,1} = 440$
- $x_{2,3} = 100$
- $x_{3,2} = 20$
- $x_{3,3} = 280$
- diferencias, $d_1 + d_2 = 0 + 1000 = 1000$
- Coste, 30000 Euros.
- Puntuación total, 12000.
- $2f_1 - f_2$, 47460.

12.4. Establecer y resolver una sucesión de programas lineales para calcular una solución de programación por metas lexicográfica tomando los objetivos en el orden dado.

Etapas 1

$$\text{mind}_1$$

sujeto a

$$15x_{1,1} + 15x_{1,2} + 25x_{2,1} + 25x_{2,3} + 30x_{3,2} + 30x_{3,3} - d_1 \leq 30000$$

$$7x_{1,1} + 8x_{1,2} + 10x_{2,1} + 6x_{2,3} + 10x_{3,2} + 10x_{3,3} + d_2 \geq 13000$$

$$\begin{aligned}
x_{1,1} + x_{1,2} &\leq 500 \\
x_{2,1} + x_{2,3} &\leq 630 \\
x_{3,2} + x_{3,3} &\leq 710 \\
x_{1,1} + x_{2,1} &= 440 \\
x_{1,2} + x_{3,2} &= 520 \\
x_{2,3} + x_{3,3} &= 380 \\
x_{i,j} &\geq 0, i, j = 1, 2, 3
\end{aligned}$$

La solución obtenida mediante el solver de Excell es:

- $x_{1,1} = 0$
- $x_{1,2} = 500$
- $x_{2,1} = 440$
- $x_{2,3} = 100$
- $x_{3,2} = 20$
- $x_{3,3} = 280$
- diferencia $d_1 = 0, d_2 = 1000$
- Coste, 30000 Euros.
- Puntuación total, 12000.

Etapas 2

$$mind_2$$

sujeto a

$$15x_{1,1} + 15x_{1,2} + 25x_{2,1} + 25x_{2,3} + 30x_{3,2} + 30x_{3,3} - d_1 \leq 30000$$

$$7x_{1,1} + 8x_{1,2} + 10x_{2,1} + 6x_{2,3} + 10x_{3,2} + 10x_{3,3} + d_2 \geq 13000$$

$$\begin{aligned}
x_{1,1} + x_{1,2} &\leq 500 \\
x_{2,1} + x_{2,3} &\leq 630 \\
x_{3,2} + x_{3,3} &\leq 710 \\
x_{1,1} + x_{2,1} &= 440 \\
x_{1,2} + x_{3,2} &= 520 \\
x_{2,3} + x_{3,3} &= 380
\end{aligned}$$

$$x_{i,j} \geq 0, i, j = 1, 2, 3$$

$$d_1 = 0$$

La solución obtenida mediante el solver de Excell es:

- $x_{1,1} = 0$
- $x_{1,2} = 500$
- $x_{2,1} = 440$
- $x_{2,3} = 100$
- $x_{3,2} = 20$
- $x_{3,3} = 280$
- diferencia $d_1 = 0, d_2 = 1000$
- Coste, 30000 Euros.
- Puntuación total, 12000.

12.5. Formular una función objetivo alternativa del programa por metas de la parte c) que efectúe la optimización por metas lexicográfica de la parte d) en un único paso.

Para ello consideramos una ponderación en la función de objetivos, por ejemplo la tomada en 12.2

$$\min 2d_1 + d_2$$

sujeto a

$$15x_{1,1} + 15x_{1,2} + 25x_{2,1} + 25x_{2,3} + 30x_{3,2} + 30x_{3,3} - d_1 \leq 30000$$

$$7x_{1,1} + 8x_{1,2} + 10x_{2,1} + 6x_{2,3} + 10x_{3,2} + 10x_{3,3} + d_2 \geq 13000$$

$$x_{1,1} + x_{1,2} \leq 500$$

$$x_{2,1} + x_{2,3} \leq 630$$

$$x_{3,2} + x_{3,3} \leq 710$$

$$x_{1,1} + x_{2,1} = 440$$

$$x_{1,2} + x_{3,2} = 520$$

$$x_{2,3} + x_{3,3} = 380$$

$$x_{i,j} \geq 0, i, j = 1, 2, 3$$

La solución obtenida mediante el solver de Excell es:

- $x_{1,1} = 0$
- $x_{1,2} = 500$
- $x_{2,1} = 440$
- $x_{2,3} = 100$
- $x_{3,2} = 20$
- $x_{3,3} = 280$
- diferencias, $d_1 + d_2 = 0 + 1000 = 1000$
- Coste, 30000 Euros.
- Puntuación total, 12000.

Obtenemos la misma solución que en 12.4