

INTRODUCCIÓN A LA INVESTIGACIÓN OPERATIVA.

Problema de función de utilidad del consumidor.

La función de utilidad de un consumidor es $U(x, y) = xy$, siendo x e y las cantidades consumidas de los bienes A y B, cuyos precios unitarios son 2 y 3 unidades monetarias, respectivamente. Se desea maximizar la utilidad de dicho consumidor sabiendo que no puede destinar más de 90 unidades monetarias a la adquisición de dichos bienes. El modelo de programación no lineal (PNL) a resolver es el siguiente.

$$\text{Max.: } U(x, y) = xy$$

Sujeto a:

$$2x + 3y \leq 90$$

$$x, y \geq 0$$

- a. Resolved el modelo de PNL anterior con Solver de Excel. Comprobad como la solución inicial $x=0$ e $y=0$ no funciona.

VER FICHERO EXCEL

- b. Resolved el modelo de PNL anterior con SAS/OR (con SAS utilizad dos de los algoritmos de PNL que tiene implementados para este tipo de problemas).
Volved a comprobar como la solución inicial $x=0$ e $y=0$ no funciona. Interpretad los resultados obtenidos desde la solución inicial $x=3$ e $y=3$.

```
proc nlp tech=CONGRA;
  max z;
  parms x=3, y=3;
  bounds x>=0 ,y>=0;
  lincon 2*x+3*y<=90;
  z=(x*y);
run;
```

PROC NLP: Nonlinear Maximization

Gradient is computed using analytic formulas.

PROC NLP: Nonlinear Maximization

Optimization Start

Parameter Estimates

N	Parameter	Estimate	Gradient Objective Function	Lower Bound Constraint	Upper Bound Constraint
1	X	3.000000	3.000000	0	.
2	Y	3.000000	3.000000	0	.

Value of Objective Function = 9

Linear Constraints

1 75.00000 : 90.0000 >= + 2.0000 * x + 3.0000 * y

PROC NLP: Nonlinear Maximization

Conjugate-Gradient Optimization

Automatic Restart Update (Powell, 1977; Beale, 1972)

Parameter Estimates 2

Lower Bounds 2

Upper Bounds 0

Linear Constraints 1

Optimization Start

Active Constraints 0 Objective Function 9

Max Abs Gradient Element 3

Iteration	Restarts	Function Calls	Active Constraints	Objective Function	Objective Function Change	Max Abs Gradient Element	Step Size	Slope of Search Direction
1	0	5	1	324.00000	315.0	4.9923	5.000	-18.000
2	1	6	1	337.42012	13.4201	0.3840	1.000	-24.923
3	2	9	1	337.48590	0.0658	0.1613	20.000	-0.0113
4	3	11	1	337.49414	0.00824	0.1040	2.000	-0.0232
5	4	12	1	337.49997	0.00582	0.00800	1.000	-0.0108
6	5	15	1	337.49999	0.000029	0.00336	20.000	-492E-8
7	6	17	1	337.50000	3.576E-6	0.00217	2.000	-101E-7
8	7	18	1	337.50000	2.525E-6	0.000167	1.000	-469E-8
9	8	21	1	337.50000	1.238E-8	0.000070	20.000	-21E-10
10	9	23	1	337.50000	1.551E-9	0.000045	2.000	-44E-10
11	10	24	1	337.50000	1.095E-9	3.47E-6	1.000	-2E-9

Optimization Results

Iterations	11	Function Calls	25
Gradient Calls	25	Active Constraints	1
Objective Function	337.5	Max Abs Gradient Element	3.4695341E-6
Slope of Search Direction	-2.034366E-9		

ABSGCONV convergence criterion satisfied.

PROC NLP: Nonlinear Maximization

Optimization Results

Parameter Estimates

N	Parameter	Estimate	Gradient Objective Function
1	x	22.499997	15.000002
2	y	15.000002	22.499997

Value of Objective Function = 337.5

Linear Constraints Evaluated at Solution

1 ACT -7.105E-15 = 90.0000 - 2.0000 * x - 3.0000 * y

```
proc nlp tech=TRUREG;  
max z;  
parms x=3, y=3;  
bounds x>=0 ,y>=0;  
lincon 2*x+3*y<=90 , y-x>=0;  
z=(x*y);  
run;
```

PROC NLP: Nonlinear Maximization

Gradient is computed using analytic formulas.

Hessian is computed using analytic formulas.

PROC NLP: Nonlinear Maximization

Optimization Start

Parameter Estimates

N	Parameter	Estimate	Gradient Objective Function	Lower Bound Constraint	Upper Bound Constraint
1	X	3.000000	3.000000	0	.
2	Y	3.000000	3.000000	0	.

Value of Objective Function = 9

Linear Constraints

1 75.00000 : 90.0000 >= + 2.0000 * x + 3.0000 * y

PROC NLP: Nonlinear Maximization

Trust Region Optimization

Without Parameter Scaling

Parameter Estimates 2

Lower Bounds 2

Upper Bounds 0

Linear Constraints 1

Optimization Start

Active Constraints 0 Objective Function 9
Max Abs Gradient Element 3 Radius 1

Iteration	Restarts	Function Calls	Active Constraints	Objective Function	Objective Function Change	Max Abs Gradient Element	Lambda	Trust Region Radius
1 *	0	7	1	324.00000	315.0	4.9923	1.050	84.853
2	0	8	1	337.50000	13.5000	0	0	42.426

Optimization Results

Iterations	2	Function Calls	9
Hessian Calls	3	Active Constraints	1
Objective Function	337.5	Max Abs Gradient Element	0
Lambda	0	Actual Over Pred Change	0
Radius	42.426406871		

ABSGCONV convergence criterion satisfied.

PROC NLP: Nonlinear Maximization

Optimization Results

Parameter Estimates

N	Parameter	Estimate	Gradient Objective Function
1	x	22.500000	15.000000
2	y	15.000000	22.500000

Value of Objective Function = 337.5

Linear Constraints Evaluated at Solution

1 ACT 0 = 90.0000 - 2.0000 * x - 3.0000 * y

La solución es consumir 22,5 unidades de A y 15 unidades de B, lo que proporciona una utilidad máxima igual a 337,5 y utilizando para ello las 90 unidades monetarias disponibles.

- c. En los resultados de SAS, ¿cuántas iteraciones ha necesitado cada algoritmo de PNL utilizado para alcanzar el óptimo? Describid el proceso iterativo.
- El algoritmo CONGRA ha necesitado 11 iteraciones y el TRUREG 2 iteraciones. El número de iteraciones depende del algoritmo utilizado y de los valores iniciales que tomen las variables de decisión. Todos los algoritmos funcionan de un modo parecido, parten de una solución inicial y en cada iteración calcula como han de variar los valores de las variables de decisión para aproximarse cada vez más al óptimo. La diferencia entre unos algoritmos y otros está en cómo calculan dichas variaciones. Todos los algoritmos dejan de iterar cuando las variaciones obtenidas son 0 o están muy próximas a 0, esto es lo que se denomina “criterio de convergencia”.
- d. Interpretad en términos económicos el valor del multiplicador de Lagrange* resultante de la solución del problema.

Para obtener los multiplicadores de Lagrange hemos de modificar el programa utilizado en el apartado a.. Por ejemplo, se obtiene el multiplicador de Lagrange con el algoritmo CONGRA (lo señalamos en **negrita**):

```
proc nlp tech=CONGRA OUTEST=pp;
  max z;
  parms x=3, y=3;
  bounds x>=0 ,y>=0;
  lincon 2*x+3*y<=90;
  z=(x*y);
run;
proc print data=pp;
run;
```

Obs	_TECH_	_TYPE_	_NAME_	x	y	_RHS_	_ITER_
1	CONGRA	INITIAL		3.0000	3.0000	9.0	0
2	CONGRA	GRAD		3.0000	3.0000	.	0
3	CONGRA	TERMINAT	ABSGTOL	.	.	3.0	.
4	CONGRA	PARMS		22.5000	15.0000	337.5	.
5	CONGRA	GRAD		15.0000	22.5000	.	.
6	CONGRA	LOWERBD		0.0000	0.0000	.	.
7	CONGRA	NACTBC		0.0000	0.0000	.	.
8	CONGRA	NACTLC		1.0000	1.0000	.	.
9	CONGRA	LE	LC_ACT	2.0000	3.0000	90.0	.
10	CONGRA	PROJGRAD		0.0000	.	.	.
11	CONGRA	LAGM LC	LIC_NUM	1.0000	.	.	.
12	CONGRA	LAGM LC	LIC_VAL	-7.5000	.	.	.

El valor del multiplicador de Lagrange indica que por cada unidad monetaria más disponible la utilidad se verá incrementada en 7,5 unidades (se imprimen cambiados de signo).

e. Cuál es la solución óptima si se añade la restricción $x \geq y$? Resolved el modelo incorporando esta restricción en Solver de Excel y en SAS/OR. Interpretad los resultados y, especialmente, interpretad en términos económicos el valor del multiplicador de Lagrange* asociado a la nueva restricción.

```
proc nlp tech=CONGRA OUTEST=pp;
max z;
parms x=3, y=3;
bounds x>=0 ,y>=0;
lincon 2*x+3*y<=90 , y-x>=0;
z=(x*y);
run;

proc print data=pp;
run;
```

PROC NLP: Nonlinear Maximization

Gradient is computed using analytic formulas.

PROC NLP: Nonlinear Maximization

Optimization Start

Parameter Estimates

N	Parameter	Estimate	Gradient Objective Function	Lower Bound Constraint	Upper Bound Constraint
1	x	3.000000	3.000000	0	.
2	y	3.000000	3.000000	0	.

Value of Objective Function = 9

Linear Constraints

1 75.00000 : 90.0000 $\geq + 2.0000 * x + 3.0000 * y$
2 0 : ACT 0 $\leq - 1.0000 * x + 1.0000 * y$

PROC NLP: Nonlinear Maximization

Conjugate-Gradient Optimization

Automatic Restart Update (Powell, 1977; Beale, 1972)

Parameter Estimates 2

Lower Bounds 2

Upper Bounds 0

Linear Constraints 2

Optimization Start

Active Constraints 1 **Objective Function** 9

Max Abs Gradient Element 4.2426406871

Iteration	Restarts	Function Calls	Active Constraints	Objective Function	Objective Function Change	Max Abs Gradient Element	Step Size	Slope of Search Direction
1	0	5	2	324.00000	315.0	0	5.000	-18.000

Optimization Results

Iterations	1	Function Calls	6
Gradient Calls	6	Active Constraints	2
Objective Function	324	Max Abs Gradient Element	0
Slope of Search Direction	-18		

All parameters are actively constrained. Optimization cannot proceed.

PROC NLP: Nonlinear Maximization

Optimization Results

Parameter Estimates

N	Parameter	Estimate	Gradient Objective Function
1	x	18.000000	18.000000
2	y	18.000000	18.000000

Value of Objective Function = 324

Linear Constraints Evaluated at Solution

- 1 ACT 0 = 90.0000 - 2.0000 * x - 3.0000 * y
- 2 ACT 0 = 0 - 1.0000 * x + 1.0000 * y

Obs	_TECH_	_TYPE_	_NAME_	x	y	_RHS_	_ITER_
1	CONGRA	INITIAL		3.0	3.0	9	0
2	CONGRA	GRAD		3.0	3.0	.	0
3	CONGRA	TERMINAT	N_ACTIVE	.	.	10	.
4	CONGRA	PARMS		18.0	18.0	324	.
5	CONGRA	GRAD		18.0	18.0	.	.
6	CONGRA	LOWERBD		0.0	0.0	.	.
7	CONGRA	NACTBC		0.0	0.0	.	.
8	CONGRA	NACTLC		2.0	2.0	.	.
9	CONGRA	LE	LC_ACT	2.0	3.0	90	.
10	CONGRA	GE	LC_ACT	-1.0	1.0	0	.
11	CONGRA	LAGM LC	LIC_NUM	1.0	2.0	.	.
12	CONGRA	LAGM LC	LIC_VAL	-7.2	-3.6	.	.

```

proc nlp tech=TRUREG OUTEST=pp;
  max z;
  parms x=3, y=3;
  bounds x>=0 ,y>=0;
  lincon 2*x+3*y<=90 , y-x>=0;
  z=(x*y);
run;

proc print data=pp;
run;

```

PROC NLP: Nonlinear Maximization

Gradient is computed using analytic formulas.
Hessian is computed using analytic formulas.

PROC NLP: Nonlinear Maximization

Optimization Start

Parameter Estimates

N	Parameter	Estimate	Gradient Objective Function	Lower Bound Constraint	Upper Bound Constraint
1	x	3.000000	3.000000	0	.
2	y	3.000000	3.000000	0	.

Value of Objective Function = 9

Linear Constraints

1	75.00000 :	90.0000	>=	+	2.0000	*	x	+	3.0000	*	y
2	0 : ACT	0	<=	-	1.0000	*	x	+	1.0000	*	y

Trust Region Optimization

Without Parameter Scaling

Parameter Estimates 2

Lower Bounds 2

Upper Bounds 0

Linear Constraints 2

Optimization Start

Active Constraints 1 **Objective Function** 9
Max Abs Gradient Element 4.2426406871 **Radius** 1

Iteration	Restarts	Function Calls	Active Constraints	Objective Function	Objective Function Change	Max Abs Gradient Element	Lambda	Trust Region Radius
1 *	0	7	2	324.00000	315.0	0	1.050	84.853

Optimization Results

Iterations 1 **Function Calls** 8
Hessian Calls 2 **Active Constraints** 2
Objective Function 324 **Max Abs Gradient Element** 0
Lambda 1.05 **Actual Over Pred Change** 0
Radius 84.852813742

All parameters are actively constrained. Optimization cannot proceed.

PROC NLP: Nonlinear Maximization

Optimization Results

Parameter Estimates

N	Parameter	Estimate	Gradient Objective Function
1	x	18.000000	18.000000
2	y	18.000000	18.000000

Value of Objective Function = 324

Linear Constraints Evaluated at Solution

$$1 \text{ ACT } 0 = 90.0000 - 2.0000 * x - 3.0000 * y$$

$$2 \text{ ACT } 0 = 0 - 1.0000 * x + 1.0000 * y$$

Obs	_TECH_	_TYPE_	_NAME_	x	y	_RHS_	_ITER_
1	TRUREG	INITIAL		3.0	3.0	9	0
2	TRUREG	GRAD		3.0	3.0	.	0
3	TRUREG	HESSIAN	x	0.0	1.0	1	0
4	TRUREG	HESSIAN	y	1.0	0.0	2	0
5	TRUREG	DETERMIN	HESSIAN	-1.0	.	0	0
6	TRUREG	NEIGPOS	HESSIAN	.	.	1	0
7	TRUREG	NEIGNEG	HESSIAN	.	.	1	0
8	TRUREG	NEIGZERO	HESSIAN	.	.	0	0
9	TRUREG	PROJHESS		1.0	.	1	0
10	TRUREG	DETERMIN	PROJHESS	10.0	.	-1	0
11	TRUREG	NEIGPOS	PROJHESS	.	.	1	0
12	TRUREG	NEIGNEG	PROJHESS	.	.	0	0
13	TRUREG	NEIGZERO	PROJHESS	.	.	0	0
14	TRUREG	TERMINAT	N_ACTIVE	.	.	10	.
15	TRUREG	PARMS		18.0	18.0	324	.
16	TRUREG	GRAD		18.0	18.0	.	.

Obs	_TECH_	_TYPE_	_NAME_	x	y	_RHS_	_ITER_
17	TRUREG	LOWERBD		0.0	0.0	.	.
18	TRUREG	NACTBC		0.0	0.0	.	.
19	TRUREG	NACTLC		2.0	2.0	.	.
20	TRUREG	LE	LC_ACT	2.0	3.0	90	.
21	TRUREG	GE	LC_ACT	-1.0	1.0	0	.
22	TRUREG	LAGM LC	LIC_NUM	1.0	2.0	.	.
23	TRUREG	LAGM LC	LIC_VAL	-7.2	-3.6	.	.
24	TRUREG	HESSIAN	x	0.0	1.0	1	.
25	TRUREG	HESSIAN	y	1.0	0.0	2	.
26	TRUREG	DETERMIN	HESSIAN	-1.0	.	0	.
27	TRUREG	NEIGPOS	HESSIAN	.	.	1	.
28	TRUREG	NEIGNEG	HESSIAN	.	.	1	.
29	TRUREG	NEIGZERO	HESSIAN	.	.	0	.

La solución que obtenemos es consumir 18 unidades tanto de A como de B, la función objetivo toma un valor de 324, que como era de esperar es inferior al obtenido en el apartado A. Los valores de los multiplicadores de Lagrange están señalados en negrita, el de la primera restricción es igual a 7.2 y se interpreta de forma similar al apartado c.. El multiplicador de Lagrange de la segunda restricción es 3.6 he implica que por cada unidad más de y la utilidad se reduce en 3,6.

*Para calcular el multiplicador de Lagrange en SAS hay que añadir en el **PROC NLP** después de tech= la OUTEST=nombre. Después de ejecutar el NLP, ejecutar el comando que imprime los resultados guardados en la base de datos nombre:


```
proc print data=nombre;  
run;
```

Los multiplicadores de Lagrange están en el campo **x** (columna x) en las filas que se identifican con: LAGM LC y LIC_VAL, para las restricciones de desigualdad, y LAGM LC y LEC_VAL, para las restricciones de igualdad.