

Grau d'Estadística UB-UPC

Programació Lineal

Laboratori 3

PROC OPTLP i anàlisi del símplex

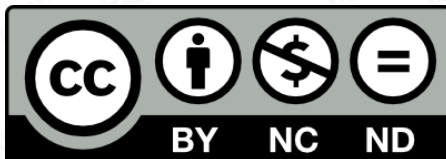
F.-Javier Heredia

<http://gnom.upc.edu/heredia>



**UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH**

**Departament d'Estadística
i Investigació Operativa**



This work is licensed under the Creative Commons Attribution-NonCommercial-NoDerivs 3.0 Unported License. To view a copy of this license, visit <http://creativecommons.org/licenses/by-nc-nd/3.0/>.

PROC OPTL i anàlisi del símplex primal⁽¹⁾

- Descripció PROC OPTLP.
- Anàlisi de l'algorisme del símplex primal amb PROC OPTLP:
 - Exemple 1: iteracions del símplex primal amb Fase I i II.
 - Exemple 2: opcions de taxació i eficiència computacional.
- Exercicis:

(1) SAS/OR® 9.3 User's Guide Mathematical Programming, cap. 8

- HTML:
http://support.sas.com/documentation/cdl/en/ormpug/65554/HTML/default/viewer.htm#ormpug_optlp_toc.htm
- PDF: <http://support.sas.com/documentation/cdl/en/ormpug/65554/PDF/default/ormpug.pdf>

PROC OPTLP

- PROC OPTLP

- Permet resoldre problemes de programació lineal mitjançant tres algorismes:

- ❖ Símplex primal (PLE, GE):

`solve with LP/solver = primal;`

- ❖ Símplex dual (PLE, GE):

`solve with LP/solver = dual;`

- ❖ Punt interior (OGE, MEIO)):

`solve with LP/solver = iterative;`

- Es pot cridar tant des de dins de OPTMODEL com de forma independent.

Sintaxi PROC OPTLP

- La crida a OPTLP des de dins del PROC OPTMODEL és:
`solve with LP </ options>;`
- On `</ options>` permeten controlar la forma com s'apliquen els algorismes. Destaquem:

options	Significat
SOLVER=	Selecció de l'algorisme d'optimització.
PRESOLVER=	Reducció automàtica de la dimensió del problema.
MAXITER=	Nre. màxim d'iteracions.
MAXTIME=	Nre. màxim de temps de CPU.
PRINTFREQ=	Freqüència d'emissió d'informació sobre les iteracions.
BASIS=	Base inicial
PRICETYPE=	Procediment de taxació.

(veure detall a http://support.sas.com/documentation/cdl/en/ormpug/59679/HTML/default/viewer.htm#lpsolver_sect4.htm)

Exemple 1: Transport_rand_1.sas

- Considereu la següent modificació del problema de transport de l'exercici 4:

```
proc optmodel;
call streaminit(1123581321); /* Llavor del generador de nombres aleatòris */
/* Parametres */
number nREF = 5; /* Nre. de refineries */
number nMER = 10; /* Nre. de mercats */
set<number> REFINERIES = 1..nREF; /* Cjt. de refineries */
set<number> MERCATS = 1..nMER; /* Cjt. de mercats */
number demanda { j in MERCATS } = 100*rand('uniform'); /* demanda generada
                                                         aleatòriament */
number dem_tot = sum{j in MERCATS} demanda[j]; /* Demanda total */
number produccio{ i in REFINERIES } = dem_tot/nREF; /* Es distribueix la
                                                         demanda entre
                                                         les refineries */
number cost { i in REFINERIES, j in MERCATS } = 10*rand('uniform');
                                                         /* Costos generats
                                                         aleatòriament */
```

Exemple 1: Transport_rand_1.sas

- Considereu la següent modificació del problema de transport de l'exercici 4:

```
/* Optimization model */
var Trans { REFINERIES, MERCATS } >= 0;
min Total_cost = sum {i in REFINERIES, j in MERCATS} cost[i,j] * Trans[i,j];
con Produccio_cons {i in REFINERIES}:
    sum {j in MERCATS} Trans[i,j] <= produccio[i];
con Demanda_cons {j in MERCATS}:
    sum {i in REFINERIES} Trans[i,j] >= demanda[j];

/* Optimització i resultats */
solve with LP / solver = primal /* Simplex primal */
    printfreq = 1; /* informació de cada iteració */
print Trans.lb Trans.sol Trans.ub cost Trans.rc Trans.status;
print Produccio_cons.lb Produccio_cons.body Produccio_cons.ub
Produccio_cons.dual Produccio_cons.status;
print Demanda_cons.lb Demanda_cons.body Demanda_cons.ub Demanda_cons.dual
Demanda_cons.status;
```

Exemple 1: Transport_rand_1: Log

NOTE: The PRIMAL SIMPLEX solver is called.

Phase	Iteration	Objective Value	Entering Variable	Leaving Variable	
Fase I					
1	1	521.113734	Trans[1,1]	Produccio_cons[1](S)	
1	2	464.006826	Trans[5,10]	Produccio_cons[5](S)	
1	3	425.003119	Trans[5,5]	Demanda_cons[10](S)	
1	4	400.146933	Trans[1,2]	Demanda_cons[1](S)	
1	5	375.471567	Trans[4,6]	Produccio_cons[4](S)	
1	6	329.948917	Trans[4,2]	Demanda_cons[6](S)	
1	7	252.862617	Trans[3,8]	Produccio_cons[3](S)	
1	8	205.884530	Trans[3,4]	Demanda_cons[8](S)	
1	9	128.597900	Trans[2,3]	Produccio_cons[2](S)	
1	10	109.977908	Trans[2,7]	Demanda_cons[3](S)	
1	11	82.024136	Trans[5,9]	Demanda_cons[5](S)	
1	12	69.882844	Trans[2,4]	Demanda_cons[7](S)	
1	13	45.026658	Trans[1,10]	Trans[1,2](S)	
1	14	29.010185	Trans[3,10]	Demanda_cons[4](S)	
1	15	12.775937	Trans[4,10]	Trans[5,10](S)	
Fase II					
2	17	2483.944156	Trans[4,9]	Demanda_cons[2](S)	
2	18	2230.511343	Trans[4,3]	Trans[4,10](S)	
2	19	1944.332084	Trans[1,5]	Trans[4,9](S)	
2	20	1918.461720	Trans[1,7]	Trans[3,4](S)	
2	21	1800.936188	Trans[2,8]	Trans[1,10](S)	
2	22	1744.153642	Trans[2,1]	Trans[2,7](S)	
2	23	1579.340930	Trans[1,6]	Trans[4,6](S)	
2	24	1443.554395	Trans[3,2]	Trans[2,3](S)	
2	25	1293.992599	Trans[5,8]	Trans[5,5](S)	
2	26	1261.701180	Trans[1,8]	Trans[2,8](S)	
2	27	1012.200055	Trans[4,1]	Trans[1,1](S)	
2	28	1012.200055	Produccio_cons[2](S)	Demanda_cons[9](S)	
2	29	981.478981	Trans[4,6]	Trans[3,8](S)	
2	30	975.810995	Trans[2,10]	Trans[4,2](S)	

NOTE: Optimal.

NOTE: Objective = 975.810995.

Opció pricetype (Taxació)

pricetype=	Taxació (<i>pricing</i>)	Procediment de taxació
HYBRID (0)	Hybrid Steepest-edge / devex pricing	Heurística que combina steepest-edge i devex (opció per defecte)
PARTIAL (1)	Partial pricing strategy	Es calculen alguns r_j i es selecciona q assoc. al més negatiu.
FULL (2)	Full pricing	Es calculen tots els r_j i es selecciona q assoc. al més negatiu.
DEVEX (3)	Devex pricing ⁽¹⁾	Aproximació de steepest-edge, menys costosa computacionalment
STEEPESTEDGE (4)	Steepest-edge pricing ⁽²⁾	$q: \frac{r_q}{\ d_q\ } = \min_{j \in \mathcal{N} r_j < 0} \left\{ \frac{r_j}{\ d_j\ } \right\}, d_j = \begin{bmatrix} -B^{-1}A_j \\ d_{\mathcal{N}_j} \end{bmatrix}$

(1) Paula M. J. Harris . **Pivot selection methods of the Devex LP code**. Mathematical Programming, 1975, Volume 4, 30-57, DOI: 10.1007/BFb0120710

(2) John J. Forrest and Donald Goldfarb, **Steepest-edge simplex algorithms for linear programming** Mathematical Programming 1992. Volume 57, Numbers 1-3, 341-374, DOI: 10.1007/BF01581089

pricetype: estudi Transport_rand_2.sas

```
proc optmodel printlevel=2;
```

Informació del temps d'execució

```
...
number nREF = 100;           /* Nre. de refineries */ 100.000 VARIABLES!!
number nMER = 1000;          /* Nre. de mercats */
...
/* Optimització i resultats */
solve with LP / presolver = 0 solver = primal printfreq = 0 pricetype = HYBRID;
solve with LP / presolver = 0 solver = primal printfreq = 0 pricetype = PARTIAL;
solve with LP / presolver = 0 solver = primal printfreq = 0 pricetype = FULLL;
solve with LP / presolver = 0 solver = primal printfreq = 0 pricetype = DEVEX;
solve with LP / presolver = 0 solver = primal printfreq = 0 pricetype = STEEPESTEDGE;
```

Pricetype	iter	sec/iter	sec	Resultat
STEEPESTEDGE	4403	0.001454	6.4	Poques iteracions molt costoses
PARTIAL	8840	0.000870	7.69	Moltes iteracions molt ràpides
FULL	8840	0.000870	7.69	
DEVEX	7978	0.001156	9.22	Força iteracions, relativament costoses
HYBRID	7978	0.001158	9.24	

Exercicis

- SBF inicial del problema de transport *.
- Planificació de la producció: estudi taxació-eficiència computacional*
- Coalco: estudi taxació-eficiència computacional**.