Statistics and Computing manuscript No.
(will be inserted by the editor)

Optimal designs for dose-escalation trials and individual allocations in cohorts (Supplementary Material)

Belmiro P.M. Duarte · **Anthony C. Atkinson** ·
Nuno M.C. Oliveira

Received: date / Accepted: date

Appendix A: Results for standard and extended designs for Setup 2

Here we present tables of the results obtained for standard and extended designs when $n = 8$, $n_c = 7$ and $N = 112$ and $n = 8$, $n_c = 8$ and $N = 128$, respectively.

Belmiro P.M. Duarte
Instituto Politécnico de Coimbra, Instituto Superior de Engenharia de Coimbra, Department of Chemical and Biological Engineering, Rua Pedro Nunes, Quinta da Nora, 3030−199 Coimbra, Portugal, and CIEPQPF, Department of Chemical Engineering, University of Coimbra, Rua Sílvio Lima – Pólo II, 3030-790 Coimbra, Portugal. Tel.: +351-239-790200, Fax: +351-239-790201, E-mail: bduarte@isec.pt

Anthony C. Atkinson
Department of Statistics, London School of Economics, London WC2A 2AE, United Kingdom. E-mail: A.C.Atkinson@lse.ac.uk

Nuno M.C. Oliveira
CIEPQPF, Department of Chemical Engineering, University of Coimbra, Rua Sílvio Lima – Pólo II, 3030-790 Coimbra, Portugal. E-mail: nuno@eq.uc.pt

Table S1 Standard traditional optimal designs for dose escalation ($n = 8$, $n_c = 7$ and $N = 112$).

	A–optimality								MV–optimality								D–optimality							
	T1	T2	T3	T4	T5	T6	T7	T8	T1	T2	T3	T4	T5	T6	T7	T8	T1	T2	T3	T4	T5	T6	T7	T8
C1	8	8	0	0	0	0	0	0	7	9	0	0	0	0	0	0	8	8	0	0	0	0	0	0
C2	4	5	7	0	0	0	0	0	6	1	9	0	0	0	0	0	5	5	6	0	0	0	0	0
C3	3	3	4	6	0	0	0	0	5	1	1	9	0	0	0	0	4	3	4	5	0	0	0	0
C4	2	2	3	3	6	0	0	0	4	2	1	0	9	0	0	0	2	2	3	4	5	0	0	0
C5	2	1	2	2	3	6	0	0	0	1	5	0	1	9	0	0	2	2	2	2	3	5	0	0
C6	1	1	1	2	2	3	6	0	5	1	0	2	1	1	6	0	1	1	1	2	3	3	5	0
C7	1	1	1	1	1	2	3	6	1	1	2	1	1	1	1	8	1	1	1	1	1	2	3	6

Table S2 Standard strict halving optimal designs for dose escalation ($n = 8$, $n_c = 7$ and $N = 112$).

	A–optimality								MV–optimality								D–optimality							
	T1	T2	T3	T4	T5	T6	T7	T8	T1	T2	T3	T4	T5	T6	T7	T8	T1	T2	T3	T4	T5	T6	T7	T8
C1	8	8	0	0	0	0	0	0	8	8	0	0	0	0	0	0	8	8	0	0	0	0	0	0
C2	4	4	8	0	0	0	0	0	4	4	8	0	0	0	0	0	4	4	8	0	0	0	0	0
C3	2	2	4	8	0	0	0	0	2	2	4	8	0	0	0	0	2	2	4	8	0	0	0	0
C4	1	1	2	4	8	0	0	0	1	1	2	4	8	0	0	0	1	1	2	4	8	0	0	0
C5	1	1	1	2	4	8	0	0	1	1	1	2	4	8	0	0	1	1	1	2	4	8	0	0
C6	1	1	1	1	2	4	6	0	1	1	1	1	2	4	6	0	1	1	1	1	2	4	6	0
C7	1	1	1	1	1	2	3	5	1	1	1	1	1	2	3	5	1	1	1	1	1	2	3	5

Table S3 Standard uniform halving optimal designs for dose escalation ($n = 8$, $n_c = 7$ and $N = 112$).

	A–optimality								MV–optimality								D–optimality							
	T1	T2	T3	T4	T5	T6	T7	T8	T1	T2	T3	T4	T5	T6	T7	T8	T1	T2	T3	T4	T5	T6	T7	T8
C1	8	8	0	0	0	0	0	0	8	8	0	0	0	0	0	0	8	8	0	0	0	0	0	0
C2	5	5	6	0	0	0	0	0	6	1	9	0	0	0	0	0	5	5	6	0	0	0	0	0
C3	3	3	4	6	0	0	0	0	6	1	1	8	0	0	0	0	4	3	4	5	0	0	0	0
C4	2	2	3	3	6	0	0	0	4	1	1	3	7	0	0	0	2	2	3	4	5	0	0	0
C5	2	1	2	2	3	6	0	0	5	1	1	1	1	7	0	0	1	2	2	2	4	5	0	0
C6	1	1	1	2	2	3	6	0	3	1	1	1	1	1	8	0	1	1	1	2	2	3	6	0
C7	1	1	1	1	1	1	2	8	2	1	1	1	1	1	1	8	1	1	1	1	1	1	2	8

Table S4 Extended traditional optimal designs for dose escalation ($n = 8$, $n_c = 8$ and $N = 128$).

	A–optimality								MV–optimality								D–optimality							
	T1	T2	T3	T4	T5	T6	T7	T8	T1	T2	T3	T4	T5	T6	T7	T8	T1	T2	T3	T4	T5	T6	T7	T8
C1	8	8	0	0	0	0	0	0	7	9	0	0	0	0	0	0	8	8	0	0	0	0	0	0
C2	5	5	6	0	0	0	0	0	3	4	9	0	0	0	0	0	5	5	6	0	0	0	0	0
C3	3	3	4	6	0	0	0	0	4	0	3	9	0	0	0	0	3	4	4	5	0	0	0	0
C4	2	2	3	4	5	0	0	0	5	1	0	4	6	0	0	0	2	2	3	4	5	0	0	0
C5	2	1	2	2	4	5	0	0	3	0	1	0	6	6	0	0	2	2	2	3	3	4	0	0
C6	1	2	1	2	2	3	5	0	4	0	0	0	0	6	6	0	1	1	2	2	2	3	5	0
C7	1	1	1	2	1	2	3	5	2	1	0	1	0	2	2	8	1	1	1	1	2	2	3	5
C8	1	1	1	1	2	2	3	5	4	0	1	0	1	0	6	4	1	1	1	1	2	3	3	4

Table S5 Extended strict halving optimal designs for dose escalation ($n = 8$, $n_c = 8$ and $N = 128$).

	A–optimality								MV–optimality								D–optimality							
	T1	T2	T3	T4	T5	T6	T7	T8	T1	T2	T3	T4	T5	T6	T7	T8	T1	T2	T3	T4	T5	T6	T7	T8
C1	8	8	0	0	0	0	0	0	8	8	0	0	0	0	0	0	8	8	0	0	0	0	0	0
C2	4	4	8	0	0	0	0	0	4	4	8	0	0	0	0	0	4	4	8	0	0	0	0	0
C3	2	2	4	8	0	0	0	0	2	2	4	8	0	0	0	0	2	2	4	8	0	0	0	0
C4	1	1	2	4	8	0	0	0	1	1	2	4	8	0	0	0	1	1	2	4	8	0	0	0
C5	2	3	1	2	4	4	0	0	4	1	1	2	4	4	0	0	2	3	1	2	4	4	0	0
C6	2	1	2	1	2	2	6	0	1	1	5	1	2	2	4	0	2	1	2	1	2	2	6	0
C7	1	1	1	2	1	1	3	6	1	1	1	1	1	1	2	8	1	1	2	2	1	1	3	5
C8	1	1	1	1	1	3	3	5	2	1	1	1	1	1	1	8	1	1	1	1	2	3	3	4

Table S6 Extended uniform halving optimal designs for dose escalation ($n = 8$, $n_c = 8$ and $N = 128$).

	A–optimality								MV–optimality								D–optimality							
	T1	T2	T3	T4	T5	T6	T7	T8	T1	T2	T3	T4	T5	T6	T7	T8	T1	T2	T3	T4	T5	T6	T7	T8
C1	8	8	0	0	0	0	0	0	8	8	0	0	0	0	0	0	8	8	0	0	0	0	0	0
C2	5	4	7	0	0	0	0	0	6	1	9	0	0	0	0	0	5	5	6	0	0	0	0	0
C3	3	3	4	6	0	0	0	0	5	1	1	9	0	0	0	0	4	3	4	5	0	0	0	0
C4	2	2	3	4	5	0	0	0	1	3	1	2	9	0	0	0	2	2	3	4	5	0	0	0
C5	1	2	2	2	4	5	0	0	2	1	1	1	2	9	0	0	2	2	2	3	3	4	0	0
C6	1	1	2	2	2	3	5	0	1	1	1	1	1	2	9	0	1	1	2	2	3	3	4	0
C7	2	1	1	1	1	2	3	5	1	1	1	1	1	1	3	7	1	1	1	2	1	3	3	4
C8	1	1	1	1	2	2	3	5	1	1	1	1	1	2	2	7	1	1	1	1	2	2	3	5

Appendix B: Results for optimal allocation on a *per individual arrival basis*

Here, we present graphs of the results obtained for the optimal allocation of individuals entering the experiment on a *per individual arrival basis*. The C-bODE is the extended traditional D–optimal design for $n = 8$, $n_c = 8$ and $N = 128$, see Table S4.

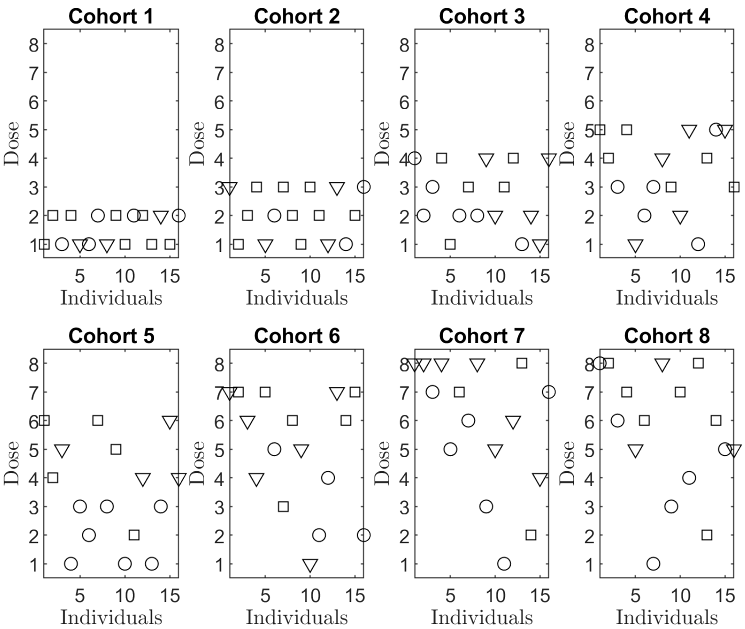

Figure S1 Sequential allocation of individuals to doses on arrival for the standard traditional D–optimal design obtained for $n = 8$, $n_c = 8$ and $N = 128$, see Table S4. Symbols: ○ - prognostic factor=1; ▿ - prognostic factor = 2; and ▫ - prognostic factor = 3.

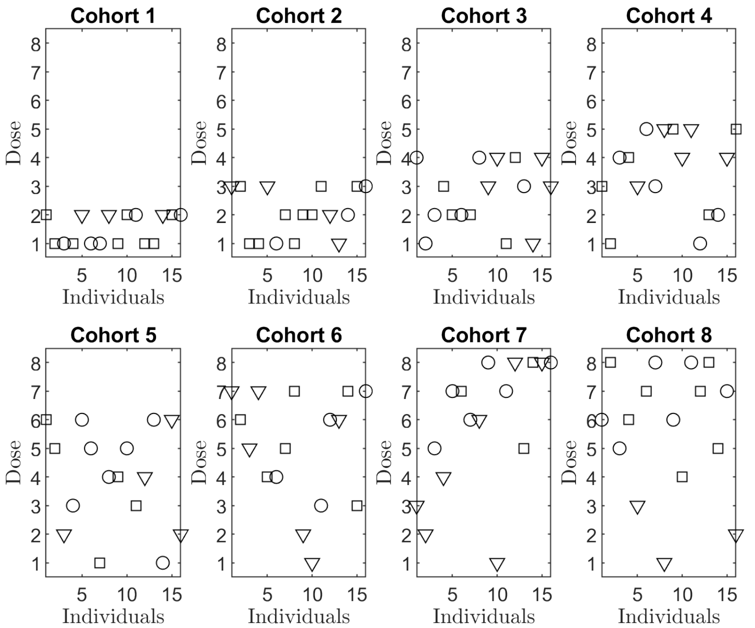

Figure S2 Simultaneous allocation of individuals within cohorts to doses for the standard traditional D–optimal design obtained for $n = 8$, $n_c = 8$ and $N = 128$, see Table S4. Symbols: ○ - prognostic factor=1; ▽ - prognostic factor = 2; and □ - prognostic factor = 3.

Appendix C: Code sample to find dose allocation designs presented in Section 4

`GAMS` is a high-level modeling system developed for mathematical optimization and is known as the first algebraic modeling language. Models are portable from one platform to another and can be easily adapted to different scenarios. The basic building blocks of a `GAMS` model are the sets, variables and equations. Information about download and installation procedures `GAMS` can be found in https://www.gams.com/download/.

Solving a design problem requires having the `GAMS` environment installed as well as license for using local or/and global MINLP solvers which on turn require a Nonlinear Programming solver and a Mixed Integer Programming solver. Typically, commercial or free solvers can be chosen see https://www.gams.com/latest/docs/UG_License.html. Afterwards, each problem is coded and run within the `GAMS` environment which works in batch mode. Next, a code sample used to generate an optimal design is shown. The information required for each model is the structure (number of covariates, design space and parameters), the elements of the Fisher Information Matrix and the tolerances used for converging the solution. The results as well as the CPU time can be displayed in the screen or saved in a results file.

```
title
$stitle
$ontext
Find the design for order of addition problem
$offtext
* activates the use of end-of-linex comments
$Oneolcom
* end-of-linex comment character
$eolcom #
$offlog # turn off linex logging
$offupper # forces GAMS to display mixed upper- and lowercase
$offsymxref # off symbol cross reference listing
$offsymlist # off symbol list
$offdigit # off number precision check
*$offlisting # turn off echoing input linexs to listing file
$onempty # Data entites can be empty
option limrow=0; # number of cases output in the LST file for each named eq.
option limcol=0; # number of cases output in the LST file for each named var.
option decimals=8; # number of decimals being displayed in the output
option reslim=7200; # lmit time
$version 240 # Force a compilation error if GAMS Rev 240 or higher is not used
* Definitions:
* ------------
* Number of parameters
$setglobal NCOHO      5
$setglobal NTREA      5
* Data sets:
* ----------
SET   nc   'number of cohorts'       /1*%NCOHO%/;
SET   nt   'number of dose treat'    /1*%NTREA%/;
SET   cr   'number of criteria'      /1*3/;
alias  (nc, nc1);
alias  (nt, nt1, nt2)
*------------
* Parameters:
*------------
PARAMETERS
      Nexp      'number of individuals allocated'
      mc        'maximum number of individuals associat to a cohort'
      mt        'number of individuals for treatment'
```

```
        optIT(cr);
Nexp    =    40;
mc      =    Nexp/card(nc);
mt      =    Nexp/card(nt);
*-----------
* Variables:
*-----------
INTEGER VARIABLES
a_matr(nc,nt)          'elements of A matrix';
POSITIVE VARIABLES
s_matr(nt,nt)          'elements of S matrix'
rep(nt)                'replicates of each dose'
maxdiag                'maximum diagonal element of pinv(FIM)';
VARIABLES
fim_th(nt,nt)          'FIM for model'
lm(nt,nt)              'matrix L'
linv(nt,nt)            'matrix L^-1'
iv(nt,nt)              'FIM^-1 matrix'
detA                   'determinant';
*-----------
* Equations:
*-----------
EQUATIONS
sigma_matr(nt,nt)      'build S matrix'
replicate(nt)          'estimate the number of replicates'
max_coho1(nc)          'upper bound for the individuals into a cohort'
max_coho2(nc)          'upper bound for the individuals into a cohort'
mat_fim1(nt,nt)        'global FIM'
mat_fim2(nt,nt)        'global FIM'
mat_fim3(nt)           'sum of rows'
eq_l(nt,nt)            'terms of L matrix'
eq_u1(nt,nt)           'terms of L matrix =0'
chol_sta(nt,nt)        'Cholesky factorization stability criteria'
eq_innod(nt,nt)        'terms non-diagonal'
eq_india(nt,nt)        'terms of diagonal'
eq_psd(nt,nt)          'PSD requirements'
eq_lu(nt,nt)           'condition on the matrix L'
eq_invlu(nt,nt)        'condition on the inverse of L'
chol_inv(nt,nt)        'Cholesky factorization stability criteria'
sumt                   'sum of all components'
fim_cons(nt)           'condition for FIM'
rel_rep(nc,nt)         'relation between rs'
eq_Aopt                'determinant computation - A-optimality'
find_max(nt)           'find maximum value of diagonal'
eq_MVopt               'determinant computation - MV-optimality'
eq_Dopt                'determinant computation - D-optimality'   ;
* S construction
sigma_matr(nt,nt1).. s_matr(nt,nt1) =e=  sum(nc, a_matr(nc,nt)*
                     a_matr(nc,nt1));
* find the replicates
replicate(nt)   ..  rep(nt)    =e=    sum(nc, a_matr(nc,nt));
*
mat_fim1(nt,nt1)$(nt1.ord ne nt.ord) .. fim_th(nt,nt1) =e= -s_matr(nt,nt1)/mc+1/card(nt);
mat_fim2(nt,nt) .. fim_th(nt,nt) =e= rep(nt)-s_matr(nt,nt)/mc+1/card(nt);
mat_fim3(nt) ..    sum(nt1, fim_th(nt,nt1)) =e= 0;
*
max_coho1(nc)   ..  sum(nt, a_matr(nc,nt))     =l=    mc;
max_coho2(nc)   ..  sum(nt, a_matr(nc,nt))     =g=    mc-1;
```

```
sumt          ..  sum((nt,nc), a_matr(nc,nt))   =e=    Nexp;
*fim_cons(nt)    ..  sum(nt1, fim_th(nt,nt1))    =e=    0;
*
* Cholesky L*L^T decomposition for A-, D- and E-optimality
eq_l(nt,nt1)$(nt.ord ge nt1.ord)    .. fim_th(nt,nt1)    =e=
        sum[nt2, lm(nt,nt2)*lm(nt1,nt2)];
eq_u1(nt,nt1)$(nt.ord lt nt1.ord)    .. lm(nt,nt1)         =e= 0;
* Condition for numerical stability in Cholesky factorization
chol_sta(nt,nt1)  .. fim_th(nt,nt) =g= sqr[lm(nt,nt1)];
* L^-1
eq_innod(nt,nt1)$(nt.ord ne nt1.ord).. sum[nt2,lm(nt,nt2)*linv(nt2,nt1)] =e= 0;
eq_india(nt,nt1)$(nt.ord eq nt1.ord).. sum[nt2,lm(nt,nt2)*linv(nt2,nt1)] =e= 1;
eq_invlu(nt,nt1)$(nt.ord lt nt1.ord).. linv(nt,nt1) =e= 0;
* equation for FIM^-1 for A-optimality
eq_psd(nt,nt1)$(nt.ord ge nt1.ord) .. iv(nt,nt1) =e=
        sum[nt2,linv(nt,nt2)*linv(nt1,nt2)];
chol_inv(nt,nt1)      .. iv(nt,nt) =g= sqr[linv(nt,nt1)];
*
* relation between r's
rel_rep(nc,nt)$(nc.ord gt 1 and nt.ord gt 1) ..
        sum(nc1$(nc1.ord le nc.ord), a_matr(nc1,nt-1)) =g=
              sum(nc1$(nc1.ord le nc.ord), a_matr(nc1,nt));
* equation for det. computation - for A-optimality
eq_Aopt ..      detA    =e=     sum[nt, iv(nt,nt)];
* equation for det. computation - for MV-optimality
find_max(nt) ..  maxdiag =g=     linv(nt,nt);
eq_MVopt ..      detA    =e=     maxdiag;
* equation for det. computation - for A-optimality
eq_Dopt ..      detA    =e=     sum[nt, log(linv(nt,nt))];
* problem
model COMMON /sigma_matr, mat_fim1, mat_fim2, replicate, sumt, max_coho1, max_coho2,
              eq_l, eq_u1, chol_sta, eq_innod, eq_india, eq_invlu,
              eq_psd, chol_inv, rel_rep/;
* model for A-optimality
model A_opt /COMMON, eq_Aopt/;
* model for MV-optimality
model MV_opt /COMMON, find_max, eq_MVopt/;
* model for D-optimality
model D_opt /COMMON, eq_Dopt/;
* parameters for solving
option optcr = 1.0e-4;
option optca = 1.0e-3;
option iterlim = 2000000000;
option domlim = 1000;
option minlp = sbb;
option bratio = 1;
option decimals = 8;
A_opt.optfile = 1;
MV_opt.optfile = 1;
D_opt.optfile = 1;
*
* initial values
$macro ResetboundsOnYforD                                  \
loop(nc,loop(nt,                                           \
a_matr.lo(nc,nt)    =   1;                                 \
a_matr.up(nc,nt)    =   round(mc/2)+1;                     \
a_matr.l(nc,nt)     =   1;                                 \
if((nt.ord gt nc.ord+1),                                   \
```

```
        a_matr.fx(nc,nt)     =  0;                                   \
););););                                                    \
a_matr.l(nc,'1')     =    round(mc/2);                          \
a_matr.l(nc,nt)$(nt.ord eq nc.ord+1) =   round(mc/2);                    \
a_matr.l(nc,nt)$(nc.ord gt 1 and nt.ord lt nc.ord+1) =   1;              \
a_matr.l(nc,'1')     =     mc-sum(nt$(nt.ord gt 1), a_matr.l(nc,nt));    \
s_matr.lo(nt,nt1)     =    0;                                     \
s_matr.up(nt,nt1)     =    sqr(mc)+1;                               \
s_matr.l(nt,nt1)      =    sum(nc, a_matr.l(nc,nt)*a_matr.l(nc,nt1));    \
rep.lo(nt)            =    round(mc/2);                             \
rep.up(nt)            =    2*mc+2;                                  \
rep.l(nt)             =    sum(nc, a_matr.l(nc,nt));                  \
fim_th.lo(nt,nt1)$(nt1.ord ne nt.ord)     =    -2*mc;                  \
fim_th.up(nt,nt1)$(nt1.ord ne nt.ord)     =    0;                     \
fim_th.lo(nt,nt)      =    1e-5;                                    \
fim_th.up(nt,nt)      =    2*mc;                                    \
fim_th.l(nt,nt1)$(nt1.ord ne nt.ord)      =    -s_matr.l(nt,nt1)/mc+1/card(nt);     \
fim_th.l(nt,nt)       =       rep.l(nt)-s_matr.l(nt,nt)/mc+1/card(nt);             \
lm.lo(nt,nt1)         =       -2e1;                                 \
lm.lo(nt,nt)          =       1.0e-5;                               \
lm.up(nt,nt1)         =       2e1;                                  \
parameters FimA(nt,nt), LforA(nt,nt);                               \
FimA(nt, nt1) = fim_th.l(nt,nt1);                                   \
execute_unload 'gdxforutility.gdx' nt, FimA;                          \
execute 'cholesky gdxforutility.gdx nt FimA gdxfromutility.gdx LforA';   \
execute_load 'gdxfromutility.gdx', LforA;                             \
lm.l(nt,nt1) = LforA(nt,nt1);                                       \
*
linv.lo(nt,nt1) = -10.0;                                      \
linv.lo(nt,nt) = 1e-5;                                        \
linv.up(nt,nt1) = 10.0;                                       \
*
* initialization for A-optimality
iv.lo(nt,nt1)         =       -50.0;                        \
iv.lo(nt,nt)          =       1e-5;                         \
iv.up(nt,nt1)         =       50.0;                         \
parameters InvA(nt,nt);                                             \
FimA(nt, nt1) = fim_th.l(nt,nt1);                                   \
execute_unload 'gdxforinverse.gdx' nt, FimA;                          \
execute 'invert gdxforinverse.gdx nt FimA gdxfrominverse.gdx InvA';     \
execute_load 'gdxfrominverse.gdx', InvA;                              \
iv.l(nt,nt1) = InvA(nt,nt1);                                  \
FimA(nt, nt1) = iv.l(nt,nt1);                                 \
execute_unload 'gdxforutility.gdx' nt, FimA;                          \
execute 'cholesky gdxforutility.gdx nt FimA gdxfromutility.gdx LforA';   \
execute_load 'gdxfromutility.gdx', LforA;                             \
linv.l(nt,nt1) = LforA(nt,nt1);                                     \
maxdiag.l      = smax(nt, iv.l(nt,nt));
* solve
parameters aux(nc,nt);
parameters col;
parameters s1(cr,nc,nt);
*
file results1 /D:\MyDocuments\MyGAMS\CohortDesign\res_unifN5.txt/;
put results1;
*
* solve for each criterion
ResetboundsOnYforD;
```

```
solve A_opt using minlp minimizing detA;
s1('1',nc,nt)  =   a_matr.l(nc,nt);
optIT('1')   =   detA.l;
ResetboundsOnYforD;
solve MV_opt using minlp minimizing detA;
s1('2',nc,nt)  =   a_matr.l(nc,nt);
optIT('2')   =   detA.l;
ResetboundsOnYforD;
solve D_opt using minlp minimizing detA;
s1('3',nc,nt)  =   a_matr.l(nc,nt);
optIT('3')   =   detA.l;
loop(cr,
    if((cr.ord eq 1),
        put '  & A-optimality  & &       ';
    );
    if((cr.ord eq 2),
        put '    MV-optimality & &       ';
    );
    if((cr.ord eq 3),
        put '    D-optimality          \\';
    );
);
put /;
col   =   6;
loop(cr,
    loop(nt,
        if((nt.ord lt card(nt)),
            put @col 'T' nt.ord:1:0 ' &'; col = col+5;
        else
            if((ord(cr) ne 3),
                put @col 'T' nt.ord:1:0 '&&';
            else
                put @col 'T' nt.ord:1:0 '   \\';
        ););
    );
    col   =   col+5;
);
put /;
col   =   6;
loop(nc,
    col   =   1;
    put @col 'C' nc.ord:1:0 ' & '; col=col+5;
    loop(cr,
        loop(nt,
        if((nt.ord lt card(nt)),
            put @col s1(cr,nc,nt):1:0 ' &'; col = col+5;
        else
            if((ord(cr) ne 3),
                put @col s1(cr,nc,nt):1:0 ' &&'; col = col+5;
            else
                put @col s1(cr,nc,nt):1:0 '   \\';
        ););););
    );
    put /;
);
col   =   6;
loop(cr,
    put @col '\multicolumn{4}{c}{'optIT(cr):7:4 '} & &'; col= col+33;
```

```
);
```