A Rectangular Trust Region Dogleg Approach for Unconstrained and Bound Constrained Nonlinear Optimization

C. VOGLIS, I.E. LAGARIS
Department of Computer Science
University of Ioannina
GREECE
{voglis, lagaris}@cs.uoi.gr

Abstract: - A trust region algorithm for unconstrained and bound constrained nonlinear optimization problems is presented. The trust region is a rectangular hyperbox in contrast with the commonly used hyperellipsoid. The resulting quadratic subproblems are solved approximatelly by an adaptation of Powell's dogleg method for rectangular trust regions. Comparative results of numerical experiments are reported.

Key-words: - Hyperbox, trust region, dogleg, bound constrained, quadratic.

1 Introduction

Non-linear optimization plays an important role in many fields of science and engineering, in the industry, as well as in a plethora of practical problems. Frequently the optimization parameters are constrained inside a range imposed by the nature of the problem at hand. Developing methods for bound constrained optimization is hence quite useful. We refer to [1] (pp. 10–12) for a list of application areas. The most efficient optimization methods are based on Newton's method where a quadratic model is adopted as a local approximation to the objective function. Two general approaches have been followed. One uses a line–search along a properly selected descent direction, while the other permits steps of restricted size in an effort to maintain the reliability of the quadratic approximation. The approaches in this second class, bear the generic name Trust-Region techniques. In this article we deal with a method of that type.

We develop a method that adopts a rectangular shape for the trust region. This geometry has the obvious advantage of the linearity of the subproblem constraints and in addition allows effortless adaptation to bound constrained problems. The emerging quadratic subproblems are of the sort:

$$\min_s \frac{1}{2}s^T Bs + s^T g \text{ subject to: } a_i \leq s_i \leq b_i \quad (1)$$

and a modification of Powell's [3] dogleg technique is developed to obtain an approximate solution.

We embed this scheme in a quasi–Newton framework that uses a positive definite approximation to the Hessian matrix. This renders the problem in Eq.1 a strictly convex one, and hence the dogleg technique is applicable.

In Section 2, we describe in brief the trust region class of algorithms along the lines of Conn, Gould and Toint [1]. In Sections 3 and 4 we present the proposed methodology along with our experimental results. Finally our conclusions are layed out in Section 5.

2 Trust region methods

Trust region methods fall in the category of sequential quadratic programming. The algorithms in this class are iterative procedures in which the objective function $f(x)$ is represented by a quadratic model inside a suitable neighborhood (the trust region) of the current iterate, as implied by the Taylor series expansion. This local model of $f(x)$ at the k^{th} iteration can be written as:

$$f(x_k + s) \approx m_k(s) = f(x_k) + g_k^T s + \frac{1}{2}s^T B_k s \quad (2)$$

where $g_k = \nabla f(x_k)$ and B_k is a symmetric approximation to $\nabla^2 f(x_k)$.

The trust region may be defined by:

$$\mathbf{T}_k = \{x \in \Re^n \mid ||x - x_k|| \leq \Delta_k\} \tag{3}$$

It is obvious that different choices for the norm lead to different trust region shapes. The Euclidean norm $||\cdot||_2$, corresponds to a hypershpere, while the $||\cdot||_\infty$ norm defines a hyperbox.

Given the model and the trust region, we seek a step s_k with $||s_k|| \leq \Delta_k$, such that the model is sufficiently reduced in value. Using this step we compare the reduction in the model to that in the objective function. If they agree to a certain extend, the step is accepted and the trust region is either expanded or remains the same. Otherwise the step is rejected and the trust region is contracted. The basic trust region algorithm is sketched in Alg. 1

Algorithm 1 Basic trust region

S0: Pick the initial point and trust region parameter x_0 and Δ_0, and set $k = 0$.

S1: Construct a quadratic model:
$m_k(s) \approx f(x_k + s)$

S2: Calculate s_k with $||s_k|| \leq \Delta_k$, so as to sufficiently reduce m_k.

S3: Compute the ratio of actual to expected reduction, $r_k = \frac{f(x_k)-f(x_k+s_k)}{m_k(0)-m_k(s_k)}$. This value will determine if the step will be accepted or not and the update for Δ_k.

S4: Increment $k \leftarrow k + 1$ and repeat from S1.

3 Outline of the algorithm

As mentioned in the introduction, our algorithm is a modification of Powell's dogleg method suitable for rectangular trust regions. The dogleg path is defined as:

$$s(a) = \begin{cases} aC & \text{for } 0 \leq a \leq 1 \\ C + (a-1)(N - C) & \text{for } 1 \leq a \leq 2 \end{cases}$$

where $C = -\frac{g_k^T g_k}{g_k^T B_k g_k} g_k$ is the Cauchy step, and $N = -H_k^{-1} g_k$ is the Newton step, that is the unconstrained minimizer of m_k. In Fig. 1 we show

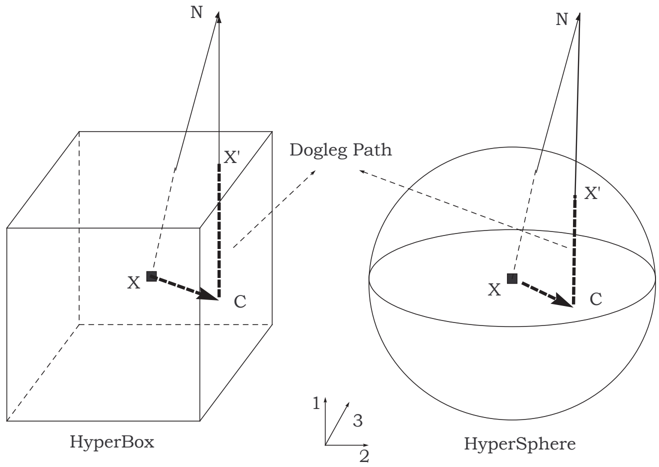

Figure 1: Dogleg path

the dogleg path for the cases of the $||\cdot||_\infty$ and the $||\cdot||_2$ norm. The quadratic model $m_k(s(a))$, decreases monotonically as a increases assuming that B_k is *positive definite*. In the original paper, the dogleg path was truncated as soon as it intersected with the trust region boundary. We distinguish the three following cases:

Case 1:	$N \in T_k$
Case 2:	$C \in T_k$ and $N \notin T_k$
Case 3:	$C \notin T_k$ and $N \notin T_k$

In our algorithm cases 1 and 2 are treated the same way as in Powell's original paper[3]. However in case 3, we prefer a slightly different approach. Instead of taking the maximum feasible step along C ($PC = bC,\ b \leq 1$) which is the case in the original algorithm, we proceed further towards N in the direction $N - PC$ until a bound is encountered. In Fig.2 we show such a case when the trust region is a hyperbox. The definition of the dogleg path under this modification is:

$$s(a) = \begin{cases} aC & \text{for } 0 \leq a \leq b \\ bC + (a-b)(N - bC) & \text{for } b \leq a \leq 1 + b \end{cases}$$

where $b = \frac{||PC||_2}{||C||_2} \in [0, 1]$. It can be trivially shown that along this path $m_k(s(a))$ monotonically decreases, reaching so a lower value for the model.

We wish to apply our method to the more general problem:

$$\min_{x \in R_n} f(x) \text{ subject to: } l_i \leq x_i \leq u_i \tag{4}$$

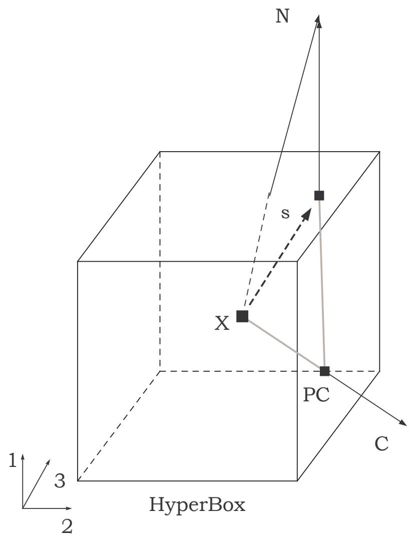

Figure 2: Our approach in Case 3

This covers both unconstrained and bound constrained problems.

We employ BFGS updates to guarantee the positive definiteness of the approximation B_k, to the Hessian matrix. We construct the model $m_k(s)$ as described in Section 2, and we omit the constant term $f(x_k)$ in Eq. 2.

The trust region at the k^{th} iteration is defined as:

$$\mathbf{T}_k = \{x \in \Re^n \mid ||x - x_k||_\infty \leq \Delta_k\} \tag{5}$$

and thus the dogleg step must be constrained by:

$$||s_k||_\infty \leq \Delta_k \tag{6}$$

in other words:

$$-\Delta_k \leq \max_k(s_k) \leq \Delta_k \tag{7}$$

From Eq. 7, and the fact that the new point $x_k + s_k$ must be feasible, the subproblem can be restated as:

$$\min_{s\in R^n} m_k(s) = \tfrac{1}{2}s^T B_k s + s^T g_k$$
$$\max[l_i - x_i, -\Delta_k] \leq s_i \leq \min[u_i - x_i, \Delta_k]$$

It is worth mentioning that when the original problem involves bound constraints, the trust region shape is a hyperectangle. When no bounds are present the trust region is just a hypercube.

Special care must be taken when an iterate x_k reaches a bound. We define the *active set* at a point x, as the set of indices:

$$\begin{aligned} A(x) \;=\; & \left\{ i \mid x_i = u_i \text{ and } \frac{\partial f}{\partial x_i} < 0 \right\} \cup \\ & \left\{ i \mid x_i = l_i \text{ and } \frac{\partial f}{\partial x_i} > 0 \right\} \end{aligned} \tag{8}$$

When $A(x_k) \neq \emptyset$ the dogleg step s_k that is com-

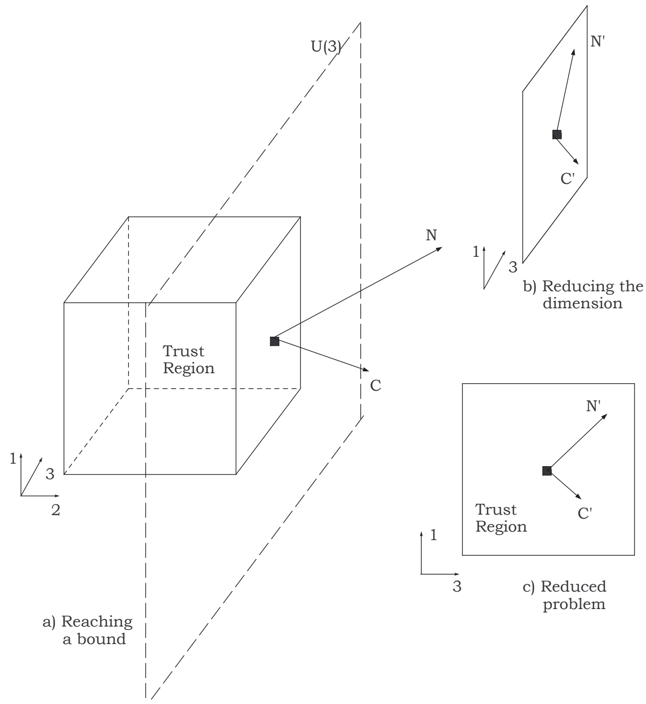

Figure 3: Bound handling

puted from the quadratic subproblem may lead outside the feasible region and hence no progress can be achieved. To deal with this situation, we reduce the dimension of the subproblem by excluding the minimization parameters that belong to the active set. Let m the number of parameters in the active set. The dimension of the subproblem is reduced to $n - m$. In Fig.3, we present a case that progress would have been impossible without the reduction.

Our algorithm is presented in Alg. 2.

4 Experimental results

In order to investigate the behavior of the DOGBOX algorithm, we have performed a substantial amount of numerical testing. We have attempted

Algorithm 2 DOGBOX

S0: Pick the initial point and trust region parameter x_0 and Δ_0, and set $k = 0$

S1: If active constrains exist, reduce the subproblem's dimension. $\tilde{B}_k$ and $\tilde{g_k}$ are reduced quantities.

S2: Construct the quadratic model around x_k:

$$m_k(s) = 1/2\tilde{s}^T \tilde{B}_k \tilde{s} + \tilde{s}^T \tilde{g}_k$$
$$\max_i [l_i - x_i, -\Delta] \leq \tilde{s}_i \leq \min_i [u_i - x_i, \Delta]$$

S3: Calculate dogleg step $\tilde{s}_k$

if $N = -\tilde{B}_k^T \tilde{g}_k$ is feasible then
  $\tilde{s}_k = N$
else
  if $C = -\frac{\tilde{g}_k^T \tilde{g}_k}{\tilde{g}_k^T \tilde{B}_k \tilde{g}_k} \tilde{g}_k$ is feasible then
    find the maximum α such that
    $C + \alpha * (N - C) \ \in T_k$
    $\tilde{s}_k = C + \alpha * (N - C)$
  else
    find the maximum β such that
    $PC \equiv \beta C \in T_k$
    find the maximum α such that
    $PC + \alpha * (N - PC) \ \in T_k$
    $\tilde{s}_k = PC + \alpha * (N - PC)$
  end if
end if

S4: Using the reduced step $\tilde{s}_k$, calculate the full space step s_k and the ratio r_k.

S5: Choose the new point x_{k+1} according to:

if $r_k \leq 0.1$ then
  $x_{k+1} = x_k$
else
  $x_{k+1} = x_k + s_k$
endif

S6: Update trust region Δ_k according to:

if $r_k < 0.25$ then
  $\Delta_{k+1} = ||s_k||/4$
else if $r_k > 0.75$ and $||s_k|| = \Delta_k$ then
  $\Delta_{k+1} = 2\Delta_k$
else
  $\Delta_{k+1} = \Delta_k$
endif

S7: Increment $k \leftarrow k + 1$ and repeat from S1.

to solve 35 unconstrained and bound constrained test problems taken from the More collection [4].

The implementation was written in double precision FORTRAN 77, and was incorporated in the Merlin Optimization Environment [5]

In the unconstrained case we compare our hyperbox-dogleg method to the originally proposed dogleg that is implemented in Merlin (command *TRUST*). We start the minimization from the points recommended by More (Test Points 1 and 2). Both methods use BFGS updates to approximate the Hessian matrix and use exactly the same scheme to treat the trust region. The stopping criteria are identical as well. The aim of these experiments is to verify that, in the unconstrained case, our method is as effective as the original one proposed by Powell. The results are shown in Table 1, were the number of iterations ("*It.*"), the function calls ("*FC*") and the gradient calls ("*GC*") are reported for each method. In this table, "$*$" denotes that the two methods ended up in different minima, and hence any comparison is meaningless.

For the bound constrained tests, the bounds were generated by the following two schemes, were x stands for the initial starting points recommended by More.

$$(1-r)x \leq x \leq (1+r)x, \ \ x \in R^n, \ 0 < r < 1 \quad (9)$$

$$x - c \leq x \leq x + c, \ \ x, c \in R^n \qquad (10)$$

Care was taken that in our experiments the unconstrained minimum was feasible in some, but not in all, cases. In the bound constrained case, we compare our method against Merlin's *TRUST* method and the well known *Tolmin*[6] algorithm which is also included in the Merlin distribution. The results of the two bound constrained tests are shown in Table 2 for Eq.9 and Table 3 for Eq.10. We should point out that the symbol "$-$" in these tables means that the method did not converge to the solution.

The presented results for the unconstrained case, offer a useful insight about the behavior of our algorithm. It seems that our method performs better (although marginally) than the original dogleg-trust region method in the majority of the test problems. We can infer that our slight modification in the dogleg path, is responsible for that.

In the bound constrained case results, we witness a dramatic improvement when we compare *TRUST* to our implementation. This is expected due to the hyberbox nature of our approach, that helps dealing with bounds in a straightforward way. Another conclusion that can be drawn is that our method behaves similarly to *Tolmin* in most cases, and overall perfoms slightly better.

5 Conclusions

We presented a trust region method, to solve unconstrained and bound constrained optimization problems, by extending Powell's dogleg technique to rectangular hyperbox trust regions. Comparison to existing methods for unconstrained problems favors, although marginally, our method. In the bound constrained case our method performs equally well to one of the leading methods[6] in the literature.

More experimentation is currently in progress with the CUTE[7] test set. Furthermore, other trust region techniques [8] are currently under a comparative investigation.

References:

[1] A. Conn, N. Gould and P. Toint, *Trust-Region methods*, MPS-SIAM Series on Optimization (2000)

[2] J. E. Dennis and R. B. Schnabel, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, SIAM (1996)

[3] M. J. D. Powell, A new algorithm for unconstrained optimization, *Nonlinear Programming*, pp. 31–65, Academic Press, London (1970)

[4] J. J. Moré, B. S. Garbow and K. E. Hillstrom, Testing unconstrained optimization environment software, *ACM Transactions on Mathematical Software*, 7(1), pp. 17–41 (1983)

[5] D. G. Papageorgiou, I. N. Demetropoulos and I. E. Lagaris , Merlin–3.0 A Multidimensional Optimization Environment, *Comput. Phys. Commun.*, 109, pp. 227–249 (1998)

[6] M.J.D. Powell, TOLMIN: A Fortran Package for Linearly Constrained Optimization Calculation, *DAMTP* (1989)

[7] I. Bongartz, A.R. Conn, Nick Gould, Ph.L. Toint, CUTE: Constrained and unconstrained testing environment *ACM Transactions on Mathematical Software*(1993)

[8] A.R. Conn, Nick Gould and P.L. Toint, Testing a Class of Methods for Solving Minimization Problems with Simple Bounds on the Variables, *Mathematics of Computation*, 50, pp. 399–430 (1988)

Problem Name	Test Point 1						Test Point 2					
	TRUST			DOGBOX			TRUST			DOGBOX		
	It.	FC	GC	It.	FC	GC	It.	FC	GC	It.	FC	GC
ROSEN	40	47	41	37	44	38	26	31	27	27	34	28
FRE-ROT	13	40	13	14	34	14	14	40	14	14	40	14
BRO-B-S	34	43	35	34	43	35	37	50	37	37	50	38
BEA	19	20	19	18	19	18	16	19	16	18	19	20
JEN-SAM	1	7	2	1	7	2	1	17	2	1	17	2
HEL-VAL	33	43	34	30	38	30	*	*	*	*	*	*
BARD	23	42	23	20	39	20	23	41	23	22	40	22
GAUS	7	19	7	7	18	8	15	15	16	13	14	14
GULF	1	2	1	1	2	1	2	22	2	2	22	2
BOX3	37	39	38	39	40	42	52	57	53	51	57	52
POW-SIN	67	71	68	88	89	94	92	97	93	71	74	72
WOOD	36	44	36	37	46	37	24	30	25	34	43	35
KOW-OSB	33	49	33	34	49	34	41	56	41	42	62	42
BRO-DEN	37	65	37	41	69	41	42	69	42	49	83	49
OSB1	67	91	67	69	92	69	111	142	111	101	133	101
BIG-E6	44	62	44	46	69	46	41	57	41	40	58	40
OSB2	66	89	66	61	89	61	49	75	49	40	63	40
WATS	159	177	159	131	156	131	180	216	180	188	225	188
X-ROS	92	107	92	104	123	104	95	115	95	98	121	98
X-POW-S	204	218	204	221	247	231	254	274	254	204	221	204
PENI	202	226	202	172	217	172	57	81	57	38	61	38
PENII	203	241	203	270	300	271	259	300	260	253	300	254
VAR-DIM	15	21	15	25	31	25	23	28	23	24	29	24
TRIG	34	48	34	30	46	30	36	50	36	39	54	39
BR-A-LIN	19	36	19	18	34	18	1	1	1	1	1	1
DISC-INT	29	30	29	33	35	33	29	29	29	34	37	35
LIN-FR	3	5	4	2	3	2	3	4	3	2	3	2
LIN-R1	3	25	3	3	25	3	3	27	3	3	25	3
LIN-R10	3	24	3	4	28	4	5	28	5	4	27	4
CHEB	38	55	38	40	63	40	150	186	150	106	144	106

Table 1: Unconstrained case

Problem Name	Test Point 1								Test Point 2							
	TRUST			BOXDOG			TOLMIN		TRUST			BOXDOG			TOLMIN	
	It.	FC	GC	It.	FC	GC	FC	GC	It.	FC	GC	It.	FC	GC	FC	GC
ROSEN	6	39	6	2	2	2	3	2	5	11	6	2	2	2	3	2
FRE-ROT	39	84	39	2	2	2	3	2	1	2	1	2	2	2	3	2
POW-B-S	11	29	11	2	2	2	3	2	13	32	13	3	3	3	5	4
BROW-B-S	8	65	8	3	48	3	37	36	6	63	6	3	3	3	4	3
BEAL	46	93	46	3	3	3	4	3	1	2	1	3	3	3	4	3
JEN-SAM	1	2	1	3	3	3	5	4	1	13	2	3	3	3	6	5
GAUS	15	16	15	7	18	8	14	15	56	73	56	9	9	9	31	32
MEYE	63	117	63	20	47	20	25	24	-	-	-	12	12	12	23	22
GULF	50	100	50	6	6	6	8	7	50	97	50	10	10	10	8	7
BOX3	5	5	6	4	4	4	5	4	7	32	7	4	4	4	5	4
POW-SI	-	-	-	4	4	4	5	4	-	-	-	3	3	3	4	3
KOW-OSB	68	84	68	13	13	13	20	19	58	105	58	7	7	7	8	7
BRO-DEN	1	9	2	3	3	3	7	6	1	12	2	3	3	3	5	4
OSB1	66	115	66	250	339	250	19	18	-	-	-	11	11	11	16	15
BIG-EX	53	70	53	10	11	10	19	18	30	46	30	16	32	16	27	26
OSB2	73	91	73	33	53	33	59	58	58	76	58	14	30	14	22	21
WATS	1	0	0	0	0	0	0	0	1	3	2	21	21	21	42	41
X-ROSE	7	33	7	2	2	2	3	2	6	40	6	2	2	2	3	2
X-POW-S	-	-	-	6	6	6	6	5	-	-	-	3	3	3	4	3
PEN1	2	36	2	5	5	5	6	5	1	2	1	5	5	5	6	5
PEN2	50	97	50	5	5	5	10	9	90	136	90	5	5	5	7	6
VAR-DIM	22	82	22	10	10	10	11	10	20	70	20	10	10	10	11	10
TRIG	61	78	61	19	36	19	33	32	53	99	53	11	11	11	13	12
BR-A-LIN	8	41	8	3	3	3	4	3	0	0	0	0	0	0	0	0
DISC-BOUN	-	-	-	20	35	20	39	38	0	0	0	0	0	0	0	0
LIN-FR	46	90	46	2	2	2	3	2	45	89	45	2	2	2	3	2
LIN-R1	1	5	2	11	11	11	12	11	1	7	2	11	11	11	12	11
LIN-R10	1	4	2	9	9	9	10	9	1	6	2	9	9	9	10	9
CHEB	49	69	49	44	66	44	60	59	74	143	74	52*	96	52	86	85

Table 2: Constrained case (1)

Problem Name	Test Point 1								Test Point 2							
	TRUST			BOXDOG			TOLMIN		TRUST			BOXDOG			TOLMIN	
	It.	FC	GC	It.	FC	GC	FC	GC	It.	FC	GC	It.	FC	GC	FC	GC
ROSEN	24	60	24	14	17	14	17	16	26	66	26	5	5	5	11	10
FREU-ROT	16	44	16	9	9	9	28	27	42	99	42	3	3	3	5	4
BROW-B-S	7	64	7	3	44	3	15	14	14	78	14	3	3	3	4	3
BEAL	-	-	-	8	24	8	20	19	1	2	1	2	2	2	3	2
JEN-SAM	-	-	-	27	54	27	55	54	-	-	-	20	50	20	67	66
GAUS	9	9	9	7	18	7	14	13	49	52	41	14	28	15	33	32
MEYE	62	116	62	12	24	12	27	26	79	170	79	47	60	47	22	21
BOX3	6	35	6	4	4	4	5	4	7	35	7	5	5	5	6	5
POW-SI	63	94	63	17	37	17	45	44	-	-	-	10	41	10	23	22
KOW-OSB	36	52	36	33	43	33	48	47	41	57	41	44	64	44	46	45
BRO-DEN	-	-	-	5	5	5	10	9	-	-	-	5	5	5	8	7
OSB1	76	102	76	70	93	70	103	102	300	300	300	91	124	91	99	98
BIG-EX	31	47	31	21	38	21	31	30	28	45	28	17	34	17	36	35
OSB2	84	106	84	54	78	54	91	90	53	69	53	19	37	19	39	38
X-ROSE	34	72	34	36	53	36	41	40	77	130	77	11	40	11	42	41
X-POW-S	79	118	79	32	61	32	56	55	-	-	-	9	34	9	27	26
PEN1	1	2	1	7	7	7	13	12	1	2	1	5	5	5	6	5
VAR-DIM	202	258	202	1	2	1	3	2	-	-	-	18	19	18	37	36
TRIG	28	41	28	32	47	32	53	52	*	*	*	*	*	*	*	*
BR-A-LIN	-	-	-	17	35	17	34	33	1	0	0	1	0	0	0	0
DISC-BOUN	27	30	27	33	35	33	46	45	32	34	32	34	37	35	50	49
DISC-INT	25	25	25	25	25	25	31	30	27	27	27	26	26	26	33	32
BROY-TRI	60	78	60	64	98	64	48	47	27	69	27	12	12	12	30	29
BROY-BAN	88	119	88	68	109	68	88	87	26	76	26	11	11	11	26	25
LIN-FR	48	93	48	2	2	2	3	2	47	92	47	2	2	2	3	2
LIN-R1	-	-	-	12	12	12	21	20	1	9	2	11	11	11	12	11
LIN-R10	-	-	-	10	10	10	19	18	1	8	2	9	9	9	10	9
CHEB	44	66	44	42	66	42	53	52	*	*	*	*	*	*	*	*

Table 3: Constrained case (2)