Direct-Search Penalty/Barrier Methods

Aldina Correia, João Matias, Pedro Mestre *Member, IAENG,* Carlos Serôdio

***Abstract*—In Nonlinear Optimization Penalty and Barrier Methods are normally used to solve Constrained Problems. There are several Penalty/Barrier Methods and they are used in several areas from Engineering to Economy, through Biology, Chemistry, Physics among others. In these areas it often appears Optimization Problems in which the involved functions (objective and constraints) are non-smooth and/or their derivatives are not know. In this work some Penalty/Barrier functions are tested and compared, using in the internal process, Derivative-free, namely Direct Search, methods. This work is a part of a bigger project involving the development of an Application Programming Interface, that implements several Optimization Methods, to be used in applications that need to solve constrained and/or unconstrained Nonlinear Optimization Problems. Besides the use of it in applied mathematics research it is also to be used in engineering software packages.**

***Index Terms*—Non-smooth Optimization, Nonlinear Programming, Derivate-free, Direct Search, Penalty/Barrier Methods.**

I. Introduction

Let us consider the Nonlinear Optimization Problem:

$$\begin{array}{ll} \min\limits_{x\in\mathbb{R}^n} & f(x) \\ s.t. & c_i(x)=0, i\in\mathcal{E} \\ & c_i(x)\leq 0, i\in\mathcal{I} \end{array} \qquad (1)$$

where:

- $f:\mathbb{R}^n\rightarrow\mathbb{R}$ is the *objective function*;
- $c_i(x)=0,\ i\in\mathcal{E}$, with $\mathcal{E}=\{1,2,...,t\}$, define the Problem *equality constraints*;
- $c_i(x)\leq 0,\ i\in\mathcal{I}$, with $\mathcal{I}=\{t+1,t+2,...,m\}$, represent the *inequality constraints*;
- $\Omega=\{x\in\mathbb{R}^n : c_i=0,\ i\in\mathcal{E}\ \wedge\ c_i(x)\leq 0,\ i\in\mathcal{I}\}$ is the set of all feasible points, i.e., the *feasible region*.

Consider the objective function and/or the constraints functions which might not be smooth, non linear, non continuous, non convex and with many local minimums and it is not possible to use derivative-based methods.

The studied methods only need information about the objective and constraints functions values, in some points, and only use this information comparing these values to find the next iteration.

Manuscript received March 05, 2010.

A. Correia is with ESTGF-IPP, School of Technology and Management of Felgueiras Polytechnic Institute of Porto, Portugal, aldinacorreia@eu.ipp.pt

J. Matias is with CM-UTAD - Centre for the Mathematics, University of Trás-os-Montes and Alto Douro, Vila Real, Portugal, email: j_matias@utad.pt

P. Mestre is with CITAB - Centre for the Research and Technology of Agro-Environment and Biological Sciences, University of Trás-os-Montes and Alto Douro, Vila Real, Portugal, email: pmestre@utad.pt

C. Serodio is with CITAB - Centre for the Research and Technology of Agro-Environment and Biological Sciences, University of Trás-os-Montes and Alto Douro, Vila Real, Portugal, email: cserodio@utad.pt

II. Penalty and Barrier Methods

Penalty and Barrier Methods, generally presented in Fig. 1, are built by two processes:

- External Process (EP) - where a succession of Unconstrained Optimization Problems is created;
- Internal Process (IP) - where the Unconstrained Optimization Problems are solved.

with the objective to solve the original Constrained Optimization Problem P presented in (1).

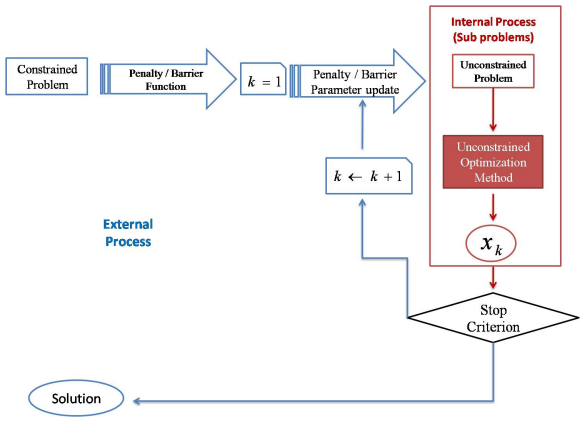

Figure 1. Penalty and Barrier Methods - General Process

Using Penalty or Barrier functions a new objective function, Φ_k, is constructed, using information about the initial objective function, f, and the constraints functions. Therefore is created a succession of Unconstrained Optimization Problems that depend on a positive parameter, r_k (Penalty/Barrier parameter) which solutions $x^*(r_k)$ converge to the initial problem solution x^* (*External Process*).

These Unconstrained Optimization Problems are then solved using Direct Search Methods (*Internal Process*), where the problem to be solved at each iteration k, is:

$$\Phi(x_k,r_k):\min_{x_k\in\mathbb{R}^n} f(x_k)+r_k p(x) \qquad (2)$$

where p is a function that penalizes (penalty) or refuses (barrier) points that violate the constraints.

In these methods optimality and feasibility are treated together.

Penalty/Barrier functions implemented in this work are:

- Extreme Barrier Function (EB);
- Progressive Barrier Function (PB);
- Classical Penalty Function (CP);
- Static/Dynamic Penalty Function (SP);
- ℓ_1 Penalty Function (ℓ_1).

Barrier methods are more adequate for solving problems where a feasible initial point is know and when a feasible solution is needed. Penalty methods can be used with

ISBN: 978-988-18210-8-9
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

infeasible initial points, and the found solution can be infeasible.

The Extreme Barrier Function with Pattern Search Methods, is widely used by Audet et. al., [1], [2], [3], [4], [5], [6] and is defined by:

$$\Phi(x) = b_\Omega(x) = \begin{cases} f(x) & \text{se} \quad x \in \Omega \\ +\infty & \text{se} \quad x \notin \Omega \end{cases} \tag{3}$$

A new version of this method is presented by Audet et. al. [5] that admits infeasible initial points if they verify that the equality constraints, the *Progressive Barrier*.

In this method the feasible region is defined as

$$\Omega = \{x \in X : c_i(x) \leq 0, i = 1, 2, ..., m\} \subset \mathbb{R}^n,$$

with X the set of non-relaxable constraints (defined by the equality constraints). The relaxable constraints $c_i(x) \leq 0$ are treated using the relaxable constraints violation measurement function $b_X : \mathbb{R}^n \to \mathbb{R}_+$:

$$b_X(x) = \begin{cases} \sum_{i=1}^{m} \left(\max\left(c_i(x), 0\right)\right)^2 & se \quad x \in X \\ +\infty & se \quad x \notin X \end{cases} \tag{4}$$

and the new objective function is:

$$\Phi_k(x) = f(x) + b_X(x) \tag{5}$$

This approach is called Progressive Barrier because it is imposed a maximum value for $b_X(x)$ that is updated at each iteration using the dominance concept of the Multi-objective Optimization.

Classical Penalty methods define Φ_k as:

$$\begin{aligned} \Phi_k(x) &= f(x) + p(x) \\ &= f(x) + r_k \sum_{i=1}^{m} \left[\max\{0, c_i(x)\}\right]^q, q \geq 1 \end{aligned} \tag{6}$$

where $r = \{r_k\}_{k=1}^{+\infty}$ is a succession such that $r_k \to +\infty$.

Static/Dynamic Penalty Function have their origin in the work of Homaifar et al. [7]. The distinction between the dynamic and static method is the way who the penalty parameters are updated.

Consider the penalty vectors $\alpha \in \mathbb{R}^t$ and $\beta \in \mathbb{R}^{m-t}$, the penalty problem for the original problem (1), with $\rho \geq 1$ is:

$$\min_{x \in \mathbb{R}^n} \Phi_k(x, \alpha, \beta) \tag{7}$$

with

$$\Phi_k(x, \alpha, \beta) = f(x) + \sum_{i=1}^{t} \alpha_i |c_i(x)|^\rho + \sum_{i=t+1}^{m} \beta_i [\max(0, c_i(x)]^\rho. \tag{8}$$

Homaifar, [7] static method uses the penalty vectors statically fixing the penalty vector and involves choosing at the beginning of the process a large number of parameters.

Alternatively dynamic methods have been developed. For example, the well known Non Stationary method solves the problem (7) with Φ_k defined in (8) and $\rho > 1$, updating the penalty parameters at each iteration k, under the follow conditions, with $C > 0$ a constant:

$$\alpha_i(k+1) = \alpha_i(k) + C.|c_i(x)|, i = 1, ..., t \tag{9}$$

$$\beta_i(k+1) = \beta_i(k) + C.[\max(0, c_i(x))], i = t+1, ..., m \tag{10}$$

ℓ_1 Penalty method is presented initially by Pietrzykowski [8], although it was presented in 1969 it has been studied and used by many authors, for example, Gould et. al. in [9] and Byrd et. al. in [10] and it is also the basis for many otherPenalty Methods proposed in the literature.

The penalty problem for the original problem (1) is:

$$\min_{x \in \mathbb{R}^n} \ell_1^{(k)}(x, \mu) \tag{11}$$

with the ℓ_1 Penalty function:

$$\ell_1^{(k)}(x, \mu) = f(x) + \mu \sum_{i=1}^{t} |c_i(x)| + \mu \sum_{i=t+1}^{m} \max[c_i(x), 0], \tag{12}$$

and $\mu \to +\infty$.

III. Derivative-free Methods

In the IP it is needed to solve a problem without constraints, i.e., a Unconstrained Optimization Problem:

$$\underset{x \in \mathbb{R}^n}{minimizar}\ f(x) \tag{13}$$

where $f : \mathbb{R}^n \to \mathbb{R}$ is the objective function, in our case the penalty function is Φ_k.

In this work we use in this process of optimization five methods:

- Opportunistic Coordinate search method (CS);
- Hooke and Jeeves method (HJ);
- A version of Audet et. al. method (A);
- Nelder-Mead method (NM);
- A Convergent Simplex method (SC).

These methods are well known and can be found in several books. The first three are Pattern Search Methods or Directional Direct-Search Methods (described, for example, in *Chapter 7 - Directional Direct-Search Methods*, by Conn et. al. [11]). These methods determine possible points using fixed search directions during the iterative process: starting at an iteration x_k, the next iteration will be found in a pattern or grid of points, in the fixed directions, at a distance s_k, said step size.

The last two are Simplex Methods (described, for example, in *Chapter 8 - Simplicial Direct-Search Methods*, by Conn et. al. [11]). These methods are characterized by to construct an initial simplex and change the directions of search in each iteration, using reflection, expansion and contraction movements and shrunk steps.

IV. Used parameters and how to update

In both processes, internal and external, depending on the methods to use, it is necessary to choose some parameters. In this section we present the parameters used in the tests, which results are presented in Section V.

A. Internal Process

Direct-Search Methods have some common parameters, presented in Table I.

This methods, in internal process, uses as set of search directions the canonical basis in $\mathbb{R}^n$ and the search order is: $e_1, -e_1, e_2, -e_2, ...$

ISBN: 978-988-18210-8-9
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

Table I
PARAMETERS USED IN INTERNAL PROCESS

Parameters	Coordinate Search	Hooke-Jeeves	Audet	Nelder-Mead	Simplex Convergent
k_{max}	100	100	100	100	100
s	1	1	*	1	1
s_m	*	*	1,5	*	*
s_p	*	*	1	*	*
s_{min}	10^{-3}	10^{-3}	10^{-3}	*	*
α	*	*	*	1	1
β	*	*	*	0,5	0,5
γ	*	*	*	2	2
T_1	10^{-5}	10^{-5}	10^{-5}	10^{-5}	10^{-5}
T_2	10^{-5}	10^{-5}	10^{-5}	10^{-5}	10^{-5}
T_{var}	*	*	*	10^{-5}	10^{-5}
T_{vol_n}	*	*	*	*	10^{-5}

k_{max} → Maximum number of iterations; s → Length of the initial step
s_m → Length of the initial mesh search step (Audet); s_p → Length of the initial poll step (Audet)
s → Length of the initial step; s_{min} → Minimum value for the step length
α → Reflexion parameter (Nelder-Mead); β → Contraction parameter (Nelder-Mead)
γ → Expansion parameter (Nelder-Mead)
$T1 = |x_k - x_{k+1}|$ → Tolerance for the distance between two consecutive iterations
or Tolerance for the distance between the last iteration and the latest iteration (Nelder-Mead)
$T2 = |f(x_k) - f(x_{k+1})|$ → Tolerance for the distance between two values of the objective function in successive iterations
T_{var} → Tolerance to the variance of the objective function values in the vertices of the simplex (Simp. Conv.)
T_{vol_n} → Tolerance to the normalized volume of the simplex
$*$ → Parameter non used in the method

In Coordinate Search and Hooke and Jeeves methods the step length was maintained, in the case of successful iterations, and reduced by half in case of unsuccessful iterations.

The input of Audet et. al. algorithm is the same as the previous algorithms, except the step length, since this algorithm have two step lengths, s_p and s_m.

Mesh search step length s_m (in the search step) and poll step length s_p (in the poll step) lengths were maintained for successful iterations and, in case of unsuccessful iterations, s_m is reduced by half and s_p is actualized according the following rule, proposed by Audet et. al. in [6]:

$$\begin{cases} s_p = \sqrt{s_m} & if \quad s_m < 1 \\ s_p \geq 1 & if \quad s_m = 1 \end{cases} \tag{14}$$

Note that if $s_m = 1$ then $s_p = 1.5$.

Audet et. al. algorithms uses random directions in the search step. To calculate the matrix of directions is generated a B nonsingular lower triangular integer matrix and its columns are permuted randomly. The columns of the resulting matrix, B_k, form the basis that is considered in the construction of the maximal basis $D = [B_k \ -B_k]$ used in the search step at iteration k.

In the algorithm implemented here the choice of search directions is deterministic. The chosen search directions are the columns of the matrix B which is nonsingular lower triangular integer matrix, where the elements are zeros or ones. and the rectagular matrix $D = [B \ -B]$.

Thus, the version of Audet et. al. algorithms here implemented differs from the coordinate search algorithm because it includes, before the coordinate search, a polling step in these directions.

Convergent Simplex algorithm implemented differs from the Nelder-Mead method because the geometry of the simplex is controlled through its diameter and normalized volume and when the simplex does not check the considered geometry conditions, a safeguard step is implemented.

The parameters of both methods are the same, but Convergent Simplex method includes another stopping criterion which is the tolerance to the normalized volume of the simplex.

B. External Process

In the EP the Penalty/Barrier parameters must be chosen. In this implementation the parameters to be chosen are presented in Table II.

We used three stopping criteria in the EP, for all methods: $T1 = |x_k - x_{k+1}| = 10^{-5} \rightarrow$ tolerance for the distance between two consecutive iterations; $T2 = |\Phi(x_k) - \Phi(x_{k+1})| = 10^{-5} \rightarrow$ tolerance for the distance between two values of the Penalty/Barrier function in successive iterations; $k_{max} = 40 \rightarrow$ Maximum number of iterations and the factor $\gamma = 2$.

Test Problems were selected from Schittkowski[12] and CUTE [13] collections. The fifteen Schittkowski problems are: S224; S225; S226; S227; S228; S231; S233; S234; S249; S264; S270; S323; S324; S325 and S326 and the of Cute collection were chosen two test problems: C and C802. The choice of these seventeen tests was not made in accordance with any special requirement, they are only used to illustrate the performance of the methods implemented.

V. NUMERICAL RESULTS

The Penalty/Barrier methods algorithms implemented here return the following results:

- Number of EP iterations - k;
- Number of Penalty/Barrier function evaluations - $nEvals$
- Last iteration - x_k;
- Value of Penalty/Barrier function at the last iteration - $\Phi(x_k)$;
- Best feasible solutions found (if any) - x_{kf};

ISBN: 978-988-18210-8-9
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

Table II
PARAMETERS USED IN EXTERNAL PROCESS

Penalty/Barrier Function	Parameter	Initial parameter	Actualization
Extreme Barrier (2)	$*$	$*$	$*$
Progressive Barrier (4) and (5)	h_{max}	$h_{max} = +\infty$	if $b_X(x_k) < h_{max}$ then $h_{max} = b_X(x_k)$
Classical Penalty (6)	r_k	1	$\underbrace{\gamma.r_k}_{r_{k+1}}, \gamma > 1$
Static Penalty (8)	$[\alpha_1, ..., \alpha_t, \beta_1, ..., \beta_{m-t}]$	$\underbrace{[1, 1, ..., 1]}_{m \text{ components}}$	$[\gamma.\alpha_1, ..., \gamma.\alpha_t, \gamma.\beta_1, ..., \gamma.\beta_{m-t}]$
Dynamic Penalty (8) with (9) and (10)	$[\alpha_1, ..., \alpha_t, \beta_1, ..., \beta_{m-t}]$	$\underbrace{[1, 1, ..., 1]}_{m \text{ components}}$	use (9) and (10)
ℓ_1Penalty Function (12)	μ	1	$\gamma.\mu$
$* \rightarrow$ Non used in the method			

- Value of objective function at the best feasible solution found - $f(x_{kf})$;
- Iteration where the best feasible solution was found - kf
- Best infeasible solutions found (if any) - x_{ki};
- Value of objective function at the best infeasible solution found - $f(x_{ki})$;
- Penalty/Barrier function value at the best infeasible solution - $\Phi(x_{ki})$;
- Iteration were was found the best infeasible solution - ki
- Constraint violation value at the best infeasible solution, $V = \Phi(x_{ki}) - f(x_{ki})$.

Of the obtained numerical results in this paper, we present the results of problems S224, S225, C801 and C802. These numerical results are presented in Tables III, IV, V and VI, rounded to two decimal points.

Analyzing the obtained numerical results it can be concluded that:

- Most methods allow to find the solution of the Schittkowski problems, except when Simplex Convergent method is used in the internal process.
- In problem S205, using dynamic penalty method none of them can find the solution. All methods, except when using the Simplex Convergent method in IP, approach it but cannot find it.
- Table V and VI show that barrier methods do not work well in either cases, they do not allow to find any solution.
- In the problems C801 and C802 penalty methods perform much better.
- None of the methods is able to find the solution of the problem C802, the best value found is about 84.67 when the target is -97.31.
- Simplex Convergent method, in these tests, does not seem to be the most suitable for use in the IP.

VI. CONCLUSION AND FUTURE WORK

In this work some Penalty/Barrier functions were tested, using in the IP, Direct Search methods. Based on the results achieved in the performed tests, it can be concluded that the Simplex Convergent method is not turned out to be the most suitable for use in the Internal Process. C801 problem results show that the use of barrier functions may cause the impossibility of solving a problem, because it creates a barrier that does not allow approach the optimal.

We cannot establish a standard to identify the most appropriate method to solve a Nonlinear Optimization Problem. The ideal is to test all methods and to choose the best solution.

This work is a part of a bigger project involving the development of an Application Programming Interface (API). Although it is being implemented in Java, this API, which implements more Optimization Methods than the ones presented here, can be used by any other programming language that supports Web Services. The developed API to be used in applications that need to solve constrained and/or unconstrained Nonlinear Optimization Problems. Besides the use of it in applied mathematics research, as presented in this paper, it is also to be used in engineering software packages.

REFERENCES

[1] C. Audet and J. E. D. Jr., "Analysis of generalized pattern searches," *SIAM Journal on Optimization*, vol. 13, no. 3, pp. 889–903, (2002).
[2] C. Audet, "Convergence results for pattern search algorithms are tight," *Optimization and Engineering*, vol. 2, no. 5, pp. 101–122, (2004).
[3] C. Audet, V. Béchard, and S. L. Digabel, "Nonsmooth optimization through mesh adaptive direct search and variable neighborhood search," *J. Global Opt.*, no. 41, pp. 299–318, (2008).
[4] C. Audet and J. E. D. Jr., "Mesh adaptive direct search algorithms for constrained optimization," *SIAM Journal on Optimization*, no. 17, pp. 188–217, (2006).
[5] ——, "A mads algorithm with a progressive barrier for derivative-free nonlinear programming," Les Cahiers du GERAD, École Polytechnique de Montréal, Tech. Rep. G-2007-37, (2007).
[6] C. Audet, J. E. D. Jr., and S. L. Digabel, "Globalization strategies for mesh adaptative direct search," Les Cahiers du GERAD, École Polytechnique de Montréal, Tech. Rep. G-2008-74, (2008).
[7] A. Homaifar, S. H. V. Lai, and X. Qi, "Constrained optimization via generic algorithms," *Simulation*, vol. 62, no. 4, pp. 242–254, (1994).
[8] T. Pietrzykowski, "An exact potential method for constrained maxima," *SIAM Journal on Numerical Analysis*, vol. 6(2), pp. 299–304, (1969).
[9] N. I. M. Gould, D. Orban, and P. L. Toint, "An interior-point l_1-penalty method for nonlinear optimization," Rutherford Appleton Laboratory Chilton, Tech. Rep., (2003).
[10] R. H. Byrd, J. Nocedal, and R. A. Waltz, "Steering exact penalty methods for nonlinear programming," *Optimization Methods & Software*, vol. 23, no. 2, pp. 197–213, (2008).
[11] A. R. Conn, K. Scheinberg, and L. N. Vicente, *Introduction to Derivative-Free Optimization*. Philadelphia, USA: MPS-SIAM Series on Optimization, SIAM, (2009).
[12] K. Schittkowski, *More Test Examples for Nonlinear Programming Codes, Economics and Mathematical Systems*. Berlin: Springer-Verlag, (1987).
[13] I. Bongartz, A. Conn, N. Gould, and P. Toint, "Cute: Constrained and unconstrained testing environment," *ACM Transactions and Mathematical Software*, no. 21, pp. 123–160, (1995).

ISBN: 978-988-18210-8-9
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

Table III
NUMERICAL RESULTS OBTAINED - S224 PROBLEM

	Methods		Output											
Problem	EP	IP	k	$nEvals$	x_k	$\Phi(x_k)$	x_{kf}	$f(x_{kf})$	kf	x_{ki}	$f(x_{ki})$	$\Phi(x_{ki})$	ki	V
S224	EB	CS	2	329	(5.90,2.10)	-293.17	x_k	$\Phi(x_k)$	k	*	*	*	*	*
		HJ	2	534	**(4.40,3.60)**	**-303.51**	x_k	$\Phi(x_k)$	k	*	*	*	*	*
Initial		A	2	659	(5.70,2.31)	-295.40	x_k	$\Phi(x_k)$	k	*	*	*	*	*
Point		NM	2	497	**(4.00,3.69)**	**-303.99**	x_k	$\Phi(x_k)$	k	*	*	*	*	*
x_0		SC	4	3899	(5.78,0.12)	-215.47	x_k	$\Phi(x_k)$	k	*	*	*	*	*
$(0.1, 0.1)$	PB	CS	2	329	(5.90,2.10)	-293.17	x_k	$\Phi(x_k)$	k	*	*	*	*	*
		HJ	2	534	**(4.40,3.60)**	**-303.51**	x_k	$\Phi(x_k)$	k	*	*	*	*	*
$f(x_0)$		A	2	659	(5.70,2.31)	-295.40	x_k	$\Phi(x_k)$	k	*	*	*	*	*
-8.77		NM	2	497	**(4.00,3.99)**	**-303.99**	x_k	$\Phi(x_k)$	k	*	*	*	*	*
		SC	4	3899	(5.78,0.12)	-215.47	x_k	$\Phi(x_k)$	k	*	*	*	*	*
	CP	CS	26	53421	**(3.98,4.02)**	**-303.99**	*	*	*	x_k	$f(x_k)$	-303.99	25	2.38E-9
		HJ	22	38086	**(3.98,4.02)**	**-303.99**	*	*	*	x_k	$f(x_k)$	-303.99	17	2.50E-3
Solution		A	27	84980	**(3.79,4.21)**	**-303.87**	*	*	*	(3.79,4.21)	-303.87	-303.87	23	2.53E-7
x^*		NM	26	25679	**(4.00,3.99)**	**-304.00**	*	*	*	(4.00,3.99)	-304.00	-304.00	25	5.55E-6
$(4, 4)$		SC	12	29541	(5.81,0.02)	-212.16	(5.85,0.00)	-212.36	4	(6.18,0.00)	-220.33	-219.80	5	5.38E-1
	SP	CS	26	53421	**(3.98,4.02)**	**-303.99**	*	*	*	(3.98,4.02)	-303.99	-303.99	25	2.38E-9
$f(x^*)$		HJ	22	38086	**(3.98,4.02)**	**-303.99**	(3.98,4.02)	-303.99	21	(3.98,4.02)	-304.00	-304.00	17	2.55E-3
-304		A	27	84980	**(3.79,4.21)**	**-303.87**	*	*	*	(3.79,4.21)	-303.87	-303.87	23	1.53E-7
		NM	26	25679	**(4.00,3.99)**	**-304.00**	*	*	*	(4.00,3.99)	-304.00	-304.00	25	5.55E-6
		SC	12	29541	(5.81,0.02)	-212.16	(5.85,0.00)	-212.36	4	(6.18,0.00)	-220.33	-219.80	5	5.38E-1
	DP	CS	40	111191	(4.11,4.21)	-309.05	*	*	*	(4.11,4.21)	-314.04	-309.05	40	4.98
		HJ	40	120914	(4.10,4.21)	-309.05	*	*	*	(4.10,4.21)	-314.04	-309.05	40	4.98
		A	40	222422	(4.10,4.21	-309.05	*	*	*	(4.10,4.21)	-314.03	-309.05	40	4.98
		NM	40	57227	(4.11,4.21)	-309.05	*	*	*	(4.11,4.21)	-314.04	-309.05	40	4.99
		SC	40	298853	(6.30,0.00)	-219.56	(5.85,0.00)	-212.36	4	(6.18,0.00)	-220.33	-220.30	5	3.23E-2
	ℓ_1	CS	7	2866	**(3.99,4.00)**	**-303.99**	(3.99,4.00)	-303.99	7	(4.00,4.00)	-304.00	-303.99	6	8.91E-4
		HJ	8	3168	**(3.99,4.00)**	**-303.99**	x_k	$f(x_k)$	k	(4.00,4.00)	-304.03	-303.99	6	3.12E-2
		A	7	5146	**(3.99,4.00)**	**-303.99**	x_k	$f(x_k)$	k	(3.99,4.00)	-304.00	-303.99	6	1.42E-3
		NM	8	3250	**(3.99,4.00)**	**-303.99**	x_k	$f(x_k)$	k	(4.00,4.00)	-304.00	-303.99	6	3.03E-3
		SC	17	59328	(5.76,0.08)	-213.39	x_k	$f(x_k)$	k	(6.01,0.00)	-216.19	-215.69	7	0.50

Table IV
NUMERICAL RESULTS OBTAINED - S225 PROBLEM

	Methods		Output											
Problem	EP	IP	k	$nEvals$	x_k	$\Phi(x_k)$	x_{kf}	$f(x_{kf})$	kf	x_{ki}	$f(x_{ki})$	$\Phi(x_{ki})$	ki	V
S225	EB	CS	2	442	**(1.00,1.00)**	**2.00**	x_k	$\Phi(x_k)$	k	*	*	*	*	*
		HJ	2	896	**(1.00,1.00)**	**2.01**	x_k	$\Phi(x_k)$	k	*	*	*	*	*
Initial		A	4	2437	**(1.00,1.00)**	**2.00**	x_k	$\Phi(x_k)$	k	*	*	*	*	*
Point		NM	40	634227	(0.0,0.0)	$+\infty$	**(1.00,1.00)**	**2.00**	1	*	*	*	*	*
x_0		SC	40	285459	(5.99,35.99)	$+\infty$	(1.77,2.44)	9.10	2	*	*	*	*	*
$(3, 1.8)$	PB	CS	2	442	**(1.00,1.00)**	**2.00**	x_k	$\Phi(x_k)$	2	*	*	*	*	*
		HJ	2	896	**(1.00,1.00)**	**2.01**	x_k	$\Phi(x_k)$	1	*	*	*	*	*
$f(x_0)$		A	4	2437	**(1.00,1.00)**	**2.00**	x_k	$\Phi(x_k)$	1	*	*	*	*	*
12.24		NM	40	634227	(0.0,0.0)	$+\infty$	**(1.00,1.00)**	**2.00**	1	*	*	*	*	*
		SC	40	285459	(5.99,35.99)	$+\infty$	(1.77,2.44)	9.10	2	*	*	*	*	*
	CP	CS	21	28337	**(0.99,0.99)**	**2.00**	*	*	*	x_k	1.99	$\Phi(x_k)$	21	7.63E-6
Solution		HJ	40	71570	**(0.99,0.99)**	**2.00**	*	*	*	x_k	1.99	$\Phi(x_k)$	12	0.01
x^*		A	20	46810	**(0.99,0.99)**	**1.99**	*	*	*	(1.0,0.99)	1.99	1.99	19	3.68E-11
$(1, 1)$		NM	20	17622	**(1.00,0.99)**	**2.00**	*	*	*	x_k	2.00	$\Phi(x_k)$	20	1.16E-5
		SC	40	390159	(6.42,39.73)	1619.65	(1.81,2.55)	9.79	1	(3.05,9.30)	95.80	95.80	8	1.68E-3
$f(x^*)$	SP	CS	21	28337	(0.99,0.99)	2.00	*	*	*	x_k	1.99	$\Phi(x_k)$	21	7.63E-6
2		HJ	40	71570	**(0.99,0.99)**	**2.00**	*	*	*	x_k	1.99	$\Phi(x_k)$	12	8.55E-3
		A	20	46810	**(0.99,0.99)**	**1.99**	*	*	*	(1.0,0.99)	1.99	1.99	19	3.68E-11
		NM	20	17622	**(1.00,0.99)**	**2.00**	*	*	*	x_k	2.00	$\Phi(x_k)$	20	1.16E-5
		SC	40	390159	(6.42,39.73)	1619.65	(1.81,2.55)	9.79	1	(3.05,9.30)	95.80	95.80	8	1.68E-3
	DP	CS	40	90022	(0.95,0.94)	1.83	*	*	*	(0.95,0.94)	1.78	1.82	32	3.66E-2
		HJ	40	72607	(0.95,0.94)	1.83	*	*	*	(0.95,0.94)	1.78	1.82	30	3.7E-2
		A	40	243753	(0.95,0.94)	1.83	*	*	*	(0.95,0.94)	1.78	1.82	33	3.69E-2
		NM	40	63555	(0.95,0.94)	1.83	*	*	*	(0.95,0.94)	1.78	1.82	33	3.6E-2
		SC	40	360739	(6.26,39.65)	1630.08	(1.81,2.55)	9.79	1	(5.98,35.79)	1316.91	1316.92	36	3.25E-3
	ℓ_1	CS	4	1019	**(1.00,1.00)**	**2.00**	(1.00,1.00)	2.00	3	(1.00,1.00)	2.00	2.00	4	6.10E-6
		HJ	40	54421	**(1.0,1.00)**	**2.00**	*	*	*	(1.0,1.00)	2.00	2.00	2	1.35E-5
		A	5	2845	**(1.00,1.00)**	**2.00**	(1.00,1.00)	2.00	4	(1.00,1.00)	2.00	2.00	5	2.89E-9
		NM	4	1234	**(1.00,1.00)**	**2.00**	x_k	1.99	4	(0.99,0.99)	1.99	2.00	3	3.12E-5
		SC	40	382586	(6.23,39.81)	5.32E11	(1.81,2.56)	9.79	1	(1.81,3.55)	15.89	16.43	2	0.54

ISBN: 978-988-18210-8-9
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

Table V
NUMERICAL RESULTS OBTAINED - C801 PROBLEM

Problem	Methods EP	IP	Output k	$nEvals$	x_k	$\Phi(x_k)$	x_{kf}	$f(x_{kf})$	kf	x_{ki}	$f(x_{ki})$	$\Phi(x_{ki})$	ki	V
C801	EB	CS	40	328861	(0.1,-0.1)	$+\infty$	*	*	*	*	*	*	*	*
		HJ	40	655221	$(-226.08, -226.28)$	$+\infty$	*	*	*	*	*	*	*	*
Initial Point x_0 $(0.1, -0.1)$		A	40	656861	(0.1,-0.1)	$+\infty$	*	*	*	*	*	*	*	*
		NM	40	660141	(0.0,-0.0)	$+\infty$	*	*	*	*	*	*	*	*
		SC	40	332141	(0.1,-0.1)	$+\infty$	*	*	*	*	*	*	*	*
	PB	CS	40	328861	(0.1,-0.1)	$+\infty$	*	*	*	*	*	*	*	*
		HJ	40	655221	$(-226.08, -226.28)$	$+\infty$	*	*	*	*	*	*	*	*
$f(x_0)$ 160.87		A	40	656861	(0.1,-0.1)	$+\infty$	*	*	*	*	*	*	*	*
		NM	40	660141	(0.0,-0.0)	$+\infty$	*	*	*	*	*	*	*	*
		SC	40	332141	(0.1,-0.1)	$+\infty$	*	*	*	*	*	*	*	*
	CP	CS	18	17727	**(4.96,1.25)**	**7.56**	*	*	*	(4.96,1.25)	7.56	7.56	18	5.04E-6
Solution x^* $(?, ?)$		HJ	18	22723	**(4.95,1.25)**	**7.56**	x_k	$\Phi(x_k)$	17	(4.95,1.25)	7.56	7.56	14	8.22E-5
		A	18	27966	**(4.96,1.25)**	**7.56**	*	*	*	(4.96,1.25)	7.56	7.56	18	4.01E-6
		NM	18	13151	**(4.97,1.25)**	**7.56**	*	*	*	(4.97,1.25)	7.56	7.56	18	1.01E-5
		SC	27	140796	(3.49,1.40)	20.41	(4.96,0.30)	13.70	2	(3.49,1.40)	20.41	20.41	27	7.67E-8
$f(x^*)$ 7.563	SP	CS	18	17727	**(4.96,1.25)**	**7.56**	*	*	*	x_k	7.56	$\Phi(x_k)$	18	5.04E-6
		HJ	18	22723	**(4.95,1.25)**	**7.56**	x_k	$\Phi(x_k)$	17	(4.95,1.25)	7.56	7.56	14	8.22E-5
		A	18	27966	**(4.96,1.25)**	**7.56**	*	*	*	x_k	7.56	$\Phi(x_k)$	18	4.01E-6
		NM	18	13151	**(4.97,1.25)**	**7.56**	*	*	*	x_k	7.56	$\Phi(x_k)$	18	1.01E-5
		SC	27	140796	(3.49,1.40)	20.41	(4.96,0.30)	13.70	2	x_k	20.41	$\Phi(x_k)$	27	7.67E-8
	DP	CS	40	71828	(4.97,1.27)	7.51	*	*	*	x_k	7.47	$\Phi(x_k)$	40	4.48E-2
		HJ	40	75414	(4.97,1.27)	7.51	*	*	*	x_k	7.47	$\Phi(x_k)$	40	4.47E-2
		A	40	121442	(4.97,1.27)	7.51	*	*	*	x_k	7.47	$\Phi(x_k)$	40	4.48E-2
		NM	40	56437	(4.97,1.27)	7.51	*	*	*	x_k	7.47	$\Phi(x_k)$	40	4.48E-2
		SC	40	289256	(1.07,20.68)	12355.60	(4.93,1.08)	8.55	3	(5.08,1.31)	7.27	7.35	4	7.14E-2
	ℓ_1	CS	3	595	**(4.96,1.25)**	**7.56**	x_k	$\Phi(x_k)$	3	(4.96,1.25)	7.56	7.56	2	8.62E-6
		HJ	7	2166	**(4.96,1.25)**	**7.56**	x_k	$\Phi(x_k)$	6	(4.96,1.25)	7.56	7.56	2	4.06E-5
		A	3	1534	**(4.96,1.25)**	**7.56**	x_k	$\Phi(x_k)$	3	(4.96,1.25)	7.56	7.56	2	8.12E-7
		NM	3	510	**(4.97,1.25)**	**7.56**	x_k	$\Phi(x_k)$	3	(4.97,1.25)	7.56	7.56	2	1.57E-6
		SC	14	36246	(5.01,1.22)	7.71	x_k	$\Phi(x_k)$	11	*	*	*	*	*

Table VI
NUMERICAL RESULTS OBTAINED - C802 PROBLEM

Problem	Methods EP	IP	Output k	$nEvals$	x_k	$\Phi(x_k)$	x_{kf}	$f(x_{kf})$	kf	x_{ki}	$f(x_{ki})$	$\Phi(x_{ki})$	ki	V
C802	EB	CS	40	328861	(0.1,-0.1)	+∞	*	*	*	*	*	*	*	*
		HJ	40	655221	$(-226.08, -226.28)$	+∞	*	*	*	*	*	*	*	*
Initial Point x_0 $(0.1, -0.1)$		A	40	656861	(0.1,-0.1)	$+\infty$	*	*	*	*	*	*	*	*
		NM	40	660141	(0.0,-0.0)	$+\infty$	*	*	*	*	*	*	*	*
		SC	40	332141	(0.1,-0.1)	$+\infty$	*	*	*	*	*	*	*	*
	PB	CS	40	328861	(0.1,-0.1)	$+\infty$	*	*	*	*	*	*	*	*
		HJ	40	655221	$(-226.08, -226.28)$	$+\infty$	*	*	*	*	*	*	*	*
$f(x_0)$ 295.1		A	40	656861	(0.1,-0.1)	$+\infty$	*	*	*	*	*	*	*	*
		NM	40	660141	(0.0,-0.0)	$+\infty$	*	*	*	*	*	*	*	*
		SC	40	332141	(0.1,-0.1)	$+\infty$	*	*	*	*	*	*	*	*
	CP	CS	23	44926	(2.65,1.41)	84.67	(2.65,1.41)	84.67	22	(2.65,1.41)	84.67	84.67	21	2.34E-5
Solution x^* $(?, ?)$		HJ	19	25232	(2.66,1.39)	84.70	x_k	$\Phi(x_k)$	18	(2.66,1.39)	84.69	84.69	13	1.50E-3
		A	23	65765	(2.62,1.47)	84.70	(2.62,1.47)	84.70	22	x_k	84.70	$\Phi(x_k)$	23	5.62E-6
		NM	23	23078	(2.67,1.43)	84.67	(2.64,1.43)	84.67	22	(2.64,1.43)	84.67	84.67	20	1.48E-5
		SC	40	501398	(0.01,33.69)	6.98E17	(2.83,0.40)	105.49	3	(3.02,0.15)	105.32	105.97	2	6.40E-1
$f(x^*)$ -97.30952	SP	CS	23	44926	(2.65,1.41)	84.67	(2.65,1.41)	84.67	22	(2.65,1.41)	84.67	84.67	21	2.34E-5
		HJ	19	25232	(2.66,1.39)	84.70	x_k	$\Phi(x_k)$	18	(2.66,1.39)	84.69	84.69	13	1.50E-3
		A	23	65765	(2.62,1.47)	84.70	(2.62,1.47)	84.70	22	x_k	84.70	84.70	23	5.62E-6
		NM	23	23078	(2.64,1.43)	84.67	(2.64,1.43)	84.67	22	(2.64,1.43)	84.67	84.67	20	1.48E-5
		SC	40	501398	(0.01,33.69)	6.98E17	(2.83,0.40)	105.49	3	(3.02,0.15)	105.32	105.97	2	6.40E-1
	DP	CS	40	131389	(2.66,1.44)	83.95	*	*	*	x_k	83.24	$\Phi(x_k)$	40	7.12E-1
		HJ	40	77411	(2.66,1.44)	83.95	*	*	*	x_k	83.23	$\Phi(x_k)$	40	7.18E-1
		A	40	211606	(2.66,1.45)	83.95	*	*	*	x_k	83.23	$\Phi(x_k)$	40	7.14E-1
		NM	40	60382	(2.66,1.44)	83.95	*	*	*	x_k	83.24	$\Phi(x_k)$	40	7.137E-1
		SC	40	445358	(3.49E-5, 33.82)	3.11E10	(2.89,0.52)	98.68	3	(3.07,0.15)	105.32	106.39	2	1.06
	ℓ_1	CS	6	2389	(2.78,1.23)	85.30	x_k	$\Phi(x_k)$	6	(2.74,1.23)	85.30	85.30	5	1.98E-5
		HJ	9	4149	(2.73,1.24)	85.24	x_k	$\Phi(x_k)$	8	(2.73,1.24)	85.23	85.24	5	1.18E-2
		A	6	4099	(2.68,1.34)	84.80	(2.68,1.34)	84.80	5	x_k	84.80	$\Phi(x_k)$	6	1.85E-5
		NM	6	1739	(2.64,1.43)	84.67	(2.64,1.43)	84.67	5	x_k	84.67	$\Phi(x_k)$	6	1.20E-8
		SC	40	448724	(0.47,26.50)	3.90E14	(2.39,1.81)	87.39	13	(2.39,1.81)	87.38	93.90	14	6.52

ISBN: 978-988-18210-8-9
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)