2nd International Conference on Engineering Optimization
September 6-9, 2010, Lisbon, Portugal

Comparative Study of Penalty Simulated Annealing Methods for Multiglobal Programming

Ana I. Pereira[1], **and Edite M.G.P. Fernandes**[2]

[1] Department of Mathematics, Polytechnic Institute of Bragança, 5301-857 Bragança, Portugal, apereira@ipb.pt

[2] Department of Production and Systems, University of Minho, 4710-057 Braga, Portugal, emgpf@dps.uminho.pt

Abstract

In a multiglobal optimization problem we aim to find all the global solutions of a constrained nonlinear programming problem where the objective function is multimodal. This class of global optimization problems is very important and frequently encountered in engineering applications, such as, process synthesis, design and control in chemical engineering. The most common method for solving this type of problems uses a local search method to refine a set of approximations, which are obtained by comparing objective function values at points of a predefined mesh. This type of method can be very expensive numerically. On the other hand, the success of local search methods depends on the starting point being at the neighbourhood of a solution. Stochastic methods are appropriate alternatives to find global solutions, in which convergence to a global solution can be guaranteed, with probability one. This is the case of the simulated annealing (SA) method. To compute the multiple solutions, a function stretching technique that transforms the objective function at each step is herein combined with SA to be able to force, step by step, convergence to each one of the required global solutions. The constraints of the problem are dealt with a penalty technique. This technique transforms the constrained problem into a sequence of unconstrained problems by penalizing the objective function when constraints are violated. Numerical experiments are shown with three penalty functions.

Keywords: nonlinear optimization; multiglobal optimization; simulated annealing.

1. Introduction

The purpose of this paper is to present a penalty framework and compare the practical behavior of three penalty functions for solving constrained multiglobal optimization problems (MGOP) formulated in the following form

$$\begin{array}{rl} \max\limits_{x\in\mathbb{R}^n} & f(x) \\ \text{subject to} & g_j(x) \leq 0,\ j=1,\ldots,m \\ & l \leq x \leq u \end{array} \tag{1}$$

where at least one of the functions f, $g_j : \mathbb{R}^n \to \mathbb{R}$ is nonlinear, and $\mathcal{F} = \{x \in \mathbb{R}^n : l \leq x \leq u,\ g_j(x) \leq 0, j = 1,\ldots,m\}$ is the feasible region. Since we do not assume convexity, f may possess many global maxima inside $\mathcal{F}$. Here, we aim to find all points $x^* \in \mathcal{F}$ such that $f(x^*) \geq f(x)$ for all $x \in \mathcal{F}$. We also assume that the problem (1) has a finite number of global maximizers. This class of global optimization problems is very important and frequently encountered in engineering applications (e.g. [2, 3, 4]). Some algorithms for solving this type of problem require substantial gradient information and aim to improve the solution in a neighborhood of a given initial approximation. When the problem has more than one global solution, the probability of convergence to an already detected global solution is very high, and depends very closely on the provided initial approximation. Techniques for detecting all global solutions represent an area of intense research. See [6, 11] and the references therein included.

The most well-known category of methods to handle constraints in nonlinear optimization problems depends on a penalty function and a positive penalty parameter. Techniques based on penalty functions transform the constrained problem into a sequence of unconstrained subproblems by penalizing f when constraints are violated and then maximizing the penalty function using methods for unconstrained problems. In general, the penalty parameter ought to be updated along the iterative process, so that convergence to the solution can be accelerated. However, the updating should not be too quick since numerical instability arises in the unconstrained optimization problem. The choice of the initial penalty value and its updating scheme are crucial issues that affect the algorithm performance. Large values of the penalty parameter give feasible solutions that have low accuracy, since search around the boundary tends to be avoided, while small values generate infeasible solutions with good accuracy.

This paper aims to assess the practical behavior of three penalty functions when solving constrained multiglobal optimization problems with a penalty approach. We test a L$_2$-exponential penalty function [14], the hyperbolic penalty function [16] and compare them with a penalty function that penalizes infeasible solutions dynamically using L$_1$ or L$_2$ terms, herein denoted by L$_{1/2}$. To compute all the global maximizers, the paper presents a Simulated Annealing (SA) algorithm combined with a function stretching technique. At each iteration, this technique is applied to a small neighbourhood of a detected solution in order to escape from that particular solution and converge to another one. The performance of the algorithm is illustrated on a selected set of small problems.

This paper is organized as follows. In Section 2, we introduce the penalty framework and describe the penalty functions that were used in our study. Section 3 presents the stretched simulated annealing method which has the ability to avoid previously computed solutions and thus detect all global solutions of the problem and Section 4 contains the numerical results. Finally, we conclude with Section 5.

2. Penalty techniques

The basic penalty approach defines a fitness for each point x, herein denoted by $\phi(x;\mu)$, by adding to the objective function value a penalty term that aims to penalize infeasible solutions as follows:

$$\phi(x;\mu) = f(x) - P(g(x), \mu, .) \tag{2}$$

where the penalty term P depends on the constraint functions $g_j(x)$, $j = 1, \dots, m$ and on at least one positive penalty parameter μ. Our implementation of the penalty framework aims to penalize only the inequality constraints. Each subproblem of the sequence that is solved for a fixed value of the penalty μ is a multiglobal optimization problem with simple bounds:

$$\begin{array}{rl} \max\limits_{x \in \mathbb{R}^n} & \phi(x;\mu) \\ \text{subject to} & l \leq x \leq u. \end{array} \tag{3}$$

Conceptually, this penalty framework is an iterative process, herein indexed by k, that computes for each value of the penalty parameter μ^k, the solution of subproblem (3). This solution is denoted by $x^*(\mu^k)$. It follows that, under certain appropriate conditions, the maximizer of the function ϕ, in (2), that satisfies $l \leq x \leq u$ for a fixed value of the parameter μ^k, converges to the solution of the given problem (1) as μ^k increases [1].

Previous experiments with a penalty strategy has shown that this well-known constraint-handling procedure is worth pursuing in the multiglobal optimization context [15]. Below we describe three penalty functions.

2.1. The L$_{1/2}$ penalty function

The L$_{1/2}$ penalty function has been widely used with stochastic methods [7, 8]. This is a non-smooth function and a non-derivative method should be applied when solving problem (3). The corresponding penalty term is defined as

$$P_{1/2}(x,\mu) = \mu \sum_{j=1}^{m} (\max\{0, g_j(x)\})^{\gamma(g_j(x))}, \tag{4}$$

where the power of the constraint violation, $\gamma(.)$, may be a violation dependent constant. The simplest approach sets $\gamma(z) = 1$ if $z \leq 0.1$, and $\gamma(z) = 2$, otherwise. To define an appropriate updating scheme for μ one has to consider a safeguarded scheme to prevent the subproblems (3) from becoming ill-conditioned. An upper bound $\mu_{\max}$ is then defined and the update is as follows:

$$\mu^{k+1} = \min\{c\mu^k, \mu_{\max}\}, \text{ for } c > 1 \text{ and } \mu_{\max} >> 1$$

for a given initial value $\mu^0 > 0$, where k represents the iteration counter.

2.2. L$_2$-exponential penalty function

Here, we extend the use of a continuous L$_2$-exponential penalty function to the constrained multiglobal optimization problem. This penalty function was previously incorporated into a reduction-type method in a semi-infinite programming context [14].

$$P_2^{\exp}(x,\nu_1,\nu_2,\mu)=\frac{\nu_1}{\mu}(e^{\mu\theta(x)}-1)+\frac{\nu_2}{2}(e^{\mu\theta(x)}-1)^2\,, \tag{5}$$

where $\theta(x)=\max_{j=1,\dots,m}[g_j(x)]_+$ and the $[g_j(x)]_+$ represents $\max\{0,g_j(x)\}$ and ν_1,ν_2 are positive parameters. Clearly $\theta(x)$ is the infinity norm of the constraint violation.

2.3. Hyperbolic penalty function

Another proposal uses the 2-parameter hyperbolic penalty function [16]. This is a continuously differentiable function that depends on nonnegative penalty parameters λ_j and σ_j, $j=1,\dots,m$, in general different for each constraint of the set

$$P^{\text{hyp}}(x,\lambda,\sigma)=\sum_{j=1}^{m}\lambda_j g_j(x)+\sqrt{\lambda_j^2[g_j(x)]^2+\sigma_j^2}. \tag{6}$$

The parameters λ_j and σ_j are updated, for each $j=1,\dots,m$, as below

$$\left\{\begin{array}{ll}\lambda_j^{k+1}=r\lambda_j^k \text{ and } \sigma_j^{k+1}=\sigma_j^k, & \text{if } \max_{l=1,\dots,m} g_l(x)\geq 0\\ \sigma_j^{k+1}=q\sigma_j^k \text{ and } \lambda_j^{k+1}=\lambda_j^k, & \text{otherwise}\end{array}\right. \quad \text{and } r\geq 1,\ q\geq 1.$$

3. Stretched simulated annealing

Here, we use the following notation: N is the number of global solutions of problem (1), $X^*=[x_1^*,x_2^*,\dots,x_N^*]$ is the $n\times N$ matrix whose columns contain the global solutions.

In general, each implementation of a global optimization method finds just one global solution. To be able to compute multiple solutions, deflection, stretching or/and repulsion techniques, have to be incorporated in the algorithm. Our proposal relies on a stretched simulated annealing algorithm in the sense that a sequence of global optimization problems with stretched objective functions is iteratively defined and solved by the SA algorithm. The SA is a point-to-point stochastic algorithm that does not require derivative information and is able to guarantee convergence to a global solution with probability one. After the computation of a global solution, the objective function of the current problem is transformed using a function stretching technique. The main steps of the SA algorithm, as well as the crucial ideas behind the proposed local application of the stretching technique, are presented in the remaining part of this section.

3.1. The simulated annealing method

The Adaptive Simulated Annealing (ASA) algorithm is a well-known variant of the simulated annealing method [5]. The main steps of the ASA algorithm are shown in Algorithm 1. Details of each step follow.

```
**Algorithm 1:** (ASA algorithm)
**Given:** $x^0$, $N_c^0$ and the initial control parameter values. Set $k=0$.
**While** the stopping condition is not verified **do**
    1. Set $j=1$
    2. Based on $x^k$, generate a trial point $y$ such that $l\leq y\leq u$
    3. Verify the "acceptance criterion"
    4. If $j<N_c^k$ then  $j=j+1$ and go to 2.
    5. Update $N_c^k$ (if adequate)
    6. Update control parameters
    7. Set $k=k+1$
```

End while
End Algorithm

To simplify the notation we use $\phi(x) = \phi(x;\mu)$. Let x^k be the current approximation to the solution of the problem (3). The ASA algorithm generates a trial point y, for at most N_c^k iterations, applying component by component this simple strategy

$$y_i = \begin{cases} 2\,l_i - \bar{y}_i & \text{if } \bar{y}_i < l_i \\ \bar{y}_i & \text{if } l_i \le \bar{y}_i \le u_i \\ 2\,u_i - \bar{y}_i & \text{if } \bar{y}_i > l_i \end{cases},$$

where $\bar{y}_i$ is given by

$$\bar{y}_i = x_i^k + d_i\,(u_i - l_i) \text{ for } i = 1,\ldots,n$$

and

$$d_i = \operatorname{sign}\left(u_i - \frac{1}{2}\right)\left(\left(1 + \frac{1}{c_{G_i}^k}\right)^{|2u_i - 1|} - 1\right) c_{G_i}^k\,, \quad u_i \sim U\,[0,1]$$

where "sign" represents the well-known *sign* function and each $c_{G_i}^k$ is a parameter that controls the generating probability density function and should decrease as iterations proceed. The "acceptance criterion" is based on the Metropolis criterion, which accepts points that improve over x^k, i.e., that have larger than or equal function values, $\phi(y) \ge \phi(x^k)$, and is also able to accept points with a smaller function value according to a certain probability τ, as follows:

$$x^{k+1} = \begin{cases} y & \text{if } \tau \le \min\left\{1, e^{\frac{\phi(y) - \phi\left(x^k\right)}{c_A^k}}\right\} \\ x^k & \text{otherwise} \end{cases}$$

being c_A^k a positive parameter that is associated to the "acceptance criterion" and $\tau \sim U(0,1)$. The updating of the control parameters $c_{G_i}^0$ and c_A^0 depend on the iteration counters k_{G_i}, $i = 1,\ldots,n$ and k_A respectively, and are defined by:

$$\begin{cases} k_{G_i} = k_{G_i} + 1 \\ c_{G_i}^k = c_{G_i}^0 e^{-\kappa\left(k_{G_i}\right)^{\frac{1}{n}}} \end{cases} \text{for } i = 1,\ldots,n$$

and

$$\begin{cases} k_A = k_A + 1 \\ c_A^k = c_A^0 e^{-\kappa(k_A)^{\frac{1}{n}}} & \kappa > 0 \end{cases}$$

The ASA variant allows a redefinition of the control parameters k_G, $c_{G_i}^0$, k_A and c_A^0 [5].

3.2. Function stretching technique

The function stretching technique aims to prevent the convergence of the ASA algorithm to a previously computed global solution. Let x_l^* be that particular solution. Thus, the function stretching technique is applied only locally, in order to transform $\phi(x)$ in a neighbourhood of x_l^*, say $V_\varepsilon(x_l^*)$, $\varepsilon > 0$. So, $\phi(x)$ is reduced only on the region $V_\varepsilon(x_l^*)$ leaving all the other maxima unchanged. The maximum $\phi(x_l^*)$ disappears but all other maxima are left unchanged. Each global optimization problem of the sequence is solved by ASA, using the Algorithm 1 previously presented. The multiglobal procedure terminates when for a predefined set of consecutive iterations no other solution is encountered [12, 13].

This stretched simulated annealing algorithm solves a sequence of global optimization problems whose objective function is obtained by applying a function stretching technique, as proposed in [9, 10, 11], to the penalty objective function of the previous problem in the sequence. The mathematical formulation of the algorithm together with the transformations that are carried out are the following:

$$\max_{l \le x \le u} \Phi_l(x) \equiv \begin{cases} \hat{\phi}(x) & \text{if } x \in V_{\varepsilon^l}(x_l^*),\ l \in \{1,\ldots,N\} \\ \phi(x) & \text{otherwise} \end{cases} \tag{7}$$

where $\hat{\phi}(x)$ is defined as

$$\hat{\phi}(x) = \bar{\phi}(x) - \frac{\delta_2[\mathrm{sign}(\phi(x_l^*) - \phi(x)) + 1]}{2\tanh(\kappa(\bar{\phi}(x_l^*) - \bar{\phi}(x))} \tag{8}$$

and

$$\bar{\phi}(x) = \phi(x) - \frac{\delta_1}{2}\|x - x_l^*\|[\mathrm{sign}(\phi(x_l^*) - \phi(x)) + 1] \tag{9}$$

with δ_1, δ_2 and κ positive constants. Transformations (9) and (8) stretch the neighbourhood of x_l^*, with ray ε^l, downwards assigning smaller function values to those points to prevent the convergence of the global optimization method to that previously computed solution [13].

To illustrate the effect of these transformations we plot in Figure 1 the function

$$f(x) = -\cos^2(x_1) - \sin^2(x_2)\,, \text{ where } x \in [-5, 5]^2$$

which has 12 global maxima in the set $[-5, 5]^2$, and in Figure 2 the function $\hat{f}$ that comes out after applying transformations (9) and (8) to the previously computed global maximizer $(\frac{\pi}{2}, 0)$.

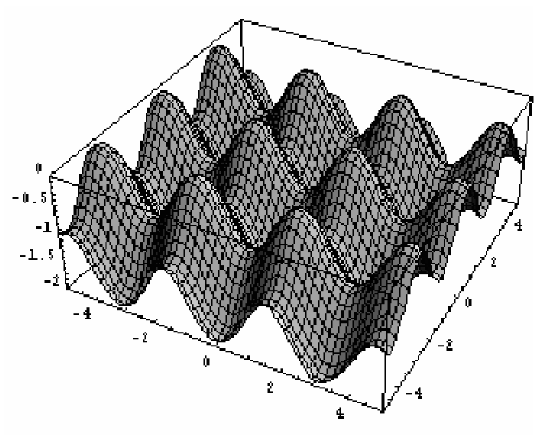

Figure 1: Plot of f.

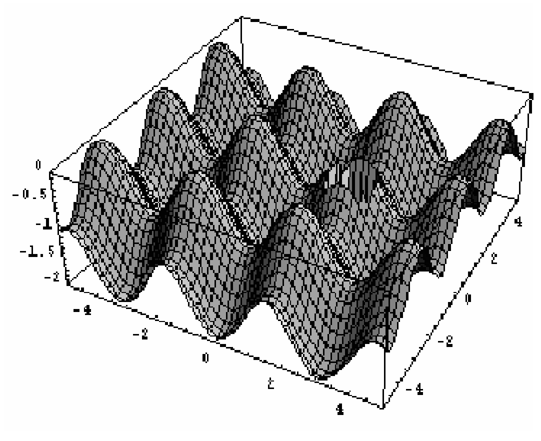

Figure 2: Plot of $\hat{f}$ after transformations.

3.3. The penalty algorithm

Incorporating the stretched simulated annealing algorithm into a penalty approach results in the proposed penalty stretched simulated annealing (PSSA) algorithm for MGOP. The main steps of the algorithm are presented below:

```
**Algorithm 2:** (PSSA algorithm)
**Given:** $\mu^0$, $\mu_{\max} >> 1$, $\tau > 1$, $\nu_1$, $\nu_2$, $\lambda^0$, $\sigma^0$, $r$, $q$. Set $k = 0$.
**While** the stopping conditions are not met **do**
  1. Given $\delta_0$, $\varepsilon_0$, $\varepsilon_{\max}$, set $L^k = 0$ and $l = 0$
  2. **While** inner stopping conditions are not met **do**
     2.1. Set $j = 0$ and $l = l + 1$
     2.2. Compute $x_l^*(\mu^k) = \arg\max_{l \leq x \leq u} \Phi_l(x)$
     2.3. **While** $\left|\Phi_l\left(x_l^*(\mu^k)\right) - \widetilde{\Phi}_{max}\right| \leq \delta_0$ or $\varepsilon_j > \varepsilon_{max}$ **do**
              Set $j = j + 1$ and $\varepsilon_j = j\varepsilon_0$
              Randomly generate $\widetilde{x}_i \in V_{\varepsilon_j}(x_l^*)$, $i = 1, \ldots, 2m$
              Find $\widetilde{\Phi}_{max} = \max_{i=1,\ldots,2m}\{\Phi_l(\widetilde{x}_i)\}$
          **End while**
     2.4. Let $L^k = L^k + 1$ and $\varepsilon^l = \varepsilon_j$
     **End while**
  3. Update the penalty parameters.
  4. Update the optimal set $X^*$ and $k = k + 1$
**End while**
Apply a local search procedure to the optimal set $X^*$.
**End Algorithm**
```

The stopping conditions for the PSSA algorithm are:

$$\|x_l^*(\mu^k) - x_l^*(\mu^{k-1})\| \leq \varepsilon_x \text{ for } l = 1, ..., N \text{ or } \ k > k_{\max}$$

and the inner iterative process terminates if L^k does not change for a specified $\mathcal{K}$ iterations. The implemented local search is a simple procedure that aims to improve accuracy and searches along each coordinate, in the neighbourhood of the solution, for a better point.

4. Numerical experiments

The herein proposed multiglobal optimization method based on a penalty technique for constraint-handling was implemented in the C programming language on a Pentium II, Celeron 466 Mhz with 64Mb of RAM. To evaluate the performance of the herein proposed penalty stretched simulated annealing algorithm for constrained MGOP a set of six benchmark problems, described in full detail in the Appendix of [17], is used. In this preliminary study, small dimensional problems ($n \leq 10$ and $m \leq 13$) with a nonlinear objective function, simple bounds and inequality constraints were tested. They are known in the literature as g04, g06, g08, g09, g12 and g18. Details of the selected problems are displayed in the Table 1, where **P** refers to the problem number, "type of f" describes the type of objective function, f_{global} is the known global solution (all are minimization problems), n is the number of variables, m is the number of inequality constraints.

Table 1: Details of the problems

P	type of f	f_{global}	n	m
g04	quadratic	$-3.0665e+04$	5	6
g06	cubic	$-6.9618e+03$	2	2
g08	general	$-9.5825e-02$	2	2
g09	general	$6.8063e+02$	7	4
g12	quadratic	$1.0000e+00$	3	9
g18	quadratic	$-8.6603e-01$	9	13

4.1. Parameters setting

The values for the user defined parameters are: $c = 10$, $\mu_{\max} = 10^8$, $\mu^0 = 10$, $N_c^0 = 2n$, $\delta_0 = 10^{-3}$, $\nu_1 = 100$, $\nu_2 = 100$, $r = 2$, $q = 2$, $\varepsilon_0 = 0.15$, $\varepsilon_{\max} = 5$, $k_{\max} = 10$ and $\varepsilon_x = 10^{-3}$, and $\mathcal{K} = 5$. When using the hyperbolic penalty, the initial values of the penalty parameters are $\lambda_j^0 = 10$ and $\sigma_j^0 = 10$, for $j = 1, ..., m$. The other parameters are fixed after a previous analysis of the problem. Each problem was solved by PSSA five times with randomly generated initial approximations.

4.2. Penalty-based comparison

Table 2 contains the results obtained with $L_{1/2}$ penalty function. In Table 2, f^*_{PSSA} is our best solution obtained after the five runs, $\mathbf{N}_{PSSA}$ is the average number of iterations required by the penalty stretched simulated annealing method, $\mathbf{N}_{feval}$ is the average number of function evaluations and $\mathbf{N}_{loc}$ gives the average number of obtained global/local solutions. Table 3 displays the results obtained by the L_2-exponential penalty function and Table 4 the results of the hyperbolic penalty function.

Table 2: Results obtained with the $L_{1/2}$ penalty function

P	f^*_{PSSA}	$\mathbf{N}_{PSSA}$	$\mathbf{N}_{feval}$	$\mathbf{N}_{loc}$
g04	$-3.2549e+04$	43	156154	12
g06	$-6.9618e+03$	9	27550	1
g08	$-9.5825e-02$	24	79771	5
g09	$6.7868e+02$	6	309719	1
g12	$1.0000e+00$	24	202219	1
g18	$-8.6603e-01$	10	945000	2

Table 3: Results obtained with the L_2-exponential penalty function

P	f^*_{PSSA}	$\mathbf{N}_{PSSA}$	$\mathbf{N}_{feval}$	$\mathbf{N}_{loc}$
g04	$-3.3038e+04$	8	62337	1
g06	$-6.9618e+03$	4	6472	1
g08	$-9.5825e-02$	30	67753	8
g09	$6.7868e+02$	5	183806	1
g12	$1.0000e+00$	21	302134	1
g18	$-8.6603e-01$	23	845375	4

We observe that the L_2-exponential penalty function reaches a solution with good accuracy and, in general, requires few iterations to identify the region of the global solution when compared with the other penalty functions. As we already expected, the number of iterations and objective function evaluations are directly related with the number of detected solutions.

Table 4: Results obtained with the hyperbolic penalty function

P	f^*_{PSSA}	$\mathbf{N}_{PSSA}$	$\mathbf{N}_{feval}$	$\mathbf{N}_{loc}$
g04	$-3.2837e+04$	4	18352	1
g06	$-6.9618e+03$	4	15766	1
g08	$-9.0057e-03$	8	8624	1
g09	$6.7868e+02$	4	117638	1
g12	$1.0000e+00$	10	313211	1
g18	$-8.6603e-01$	4	339213	4

5. Conclusions

We have presented a penalty framework for solving constrained multiglobal optimization problems. The penalty term used in our strategy aims to penalize the inequality constraints of the problem. The subproblems solved in this penalty framework are optimization problems with simple bounds since they are easy to solve by derivative-free stochastic algorithms. To compute all the global solutions of each subproblem, our algorithm combines the ASA variant of the simulated annealing method with a function stretching technique. This technique aims to transform, at each iteration, the objective function of the current global subproblem into another one that does not have the current global solution. The technique prevents the algorithm from converging to a previously computed solution. Our experience with this approach shows that more than one global solution are successfully computed, although we are not able to guarantee yet that all global solutions will be detected.

The preliminary numerical tests with a set of small problems indicate that the penalty approach is efficient and worth pursuing. The ASA variant is good at detecting promising regions of the search space but needs a local search refinement to improve solutions accuracy. This will be further explored in the future. Further experiments, in particular with high dimensional problems and large number of global solutions, are also required.

6. Acknowledgement

This work has been partially supported by the FCT (Fundação para a Ciência e a Tecnologia), Portugal.

7. References

[1] D. P. Bertsekas, Constrained Optimization and Lagrange Multiplier Methods, Academic Press, 1982.

[2] A. B.-Petriciolet, R. V.-Romn, G.A. I-Silva, K.R. Hall. Performance of Stochastic Global Optimization Methods in the Calculation of Phase Analyses for Nonreactive and Reactive Mixtures. *Industrial Engineering Chemistry Research*, 45, 4764–4772, 2006.

[3] C.M. McDonald, C.A. Floudas, Global optimization for the phase stability problem, *AIChE. Journal*, 41, 1798–1814, 1994.

[4] C. Floudas, Recent advances in global optimization for process synthesis, design and control: enclosure all solutions, *Computers and Chemical Engineering*, 963, 963–973, 1999.

[5] L. Ingber, Very fast simulated re-annealing, *Mathematical and Computer Modelling*, 12, 967–973, 1989.

[6] T. Len, S. Sanmatias and E. Vercher, A multilocal optimization algorithm, *TOP*, 6, 1–18, 1998.

[7] J.-L. Liu and J.-H. Lin, Evolutionary computation of unconstrained and constrained problems using a novel momentum-type particle swarm optimization, *Engineering Optimization*, 39(3), 287–305, 2007.

[8] Y.G. Petalas, K.E. Parsopoulos and M.N. Vrahatis, Memetic particle swarm optimization, *Annals of Operations Research*, 156, 99–127, 2007.

[9] K. Parsopoulos, V. Plagianakos, G. Magoulas, and M. Vrahatis, Objective function stretching to alleviate convergence to local minima, *Nonlinear Analysis*, 47, 3419–3424, 2001.

[10] K. Parsopoulos, and M. Vrahatis, Recent approaches to global optimization problems through particle swarm optimization, *Natural Computing*, 1, 235–306, 2002.

[11] K. Parsopoulos, and M. Vrahatis, On the computation of all global minimizers through particle swarm optimization, *IEEE Transaction on Evolutionary Computation*, 8(3), 211–224, 2004.

[12] A.I.P.N. Pereira and E.M.G.P. Fernandes, "On a reduction line search filter method for nonlinear semi-infinite programming problems", in *Euro Mini Conference "Continuous Optimization and Knowledge-Based Technologies"*, L. Sakalauskas, G.W. Weber, E.K. Zavadskas (eds.), ISBN: 978-9955-28-283-9, 2008, pp. 174–179.

[13] A.I.P.N. Pereira and E.M.G.P. Fernandes, A reduction method for semi-infinite programming by means of a global stochastic approach, *Optimization*, 58, 713–726, 2009.

[14] A.I.P.N. Pereira and E.M.G.P. Fernandes, Numerical experiments with a continuous L_2-exponential merit function for semi-infinite programming, *International Electronic Conference on Computer Science*, T.E. Simos, G. Psihoyios (eds.) AIP Vol. 1060, Issue 1, 1354–1357, Springer-Verlag 2008.

[15] A.I.P.N. Pereira and E.M.G.P. Fernandes, Constrained multi-global optimization using a penalty stretched simulated annealing framework, *Numerical Analysis and Applied Mathematics*, T.E. Simos, G. Psihoyios and Ch. Tsitouras (eds.) AIP Vol. 1168, 1354–1357, Springer-Verlag 2009.

[16] A. Xavier, Hyperbolic penalty: a new method for nonlinear programming with inequalities, *International Transactions in Operational Research*, 8, 659–671, 2001.

[17] A.E.M. Zavala, A.H. Aguirre, and E.R.V. Diharce. Constrained optimization via particle evolutionary swarm optimization algorithm (PESO). In *GECCO'05*, 209–216, 2005.