A Hyperbolic Penalty Filter Method for Semi-Infinite Programming

Ana Isabel P.N. Pereira[*] and Edite M.G.P. Fernandes[†]

[*]*Department of Mathematics, Polytechnic Institute of Bragança, Portugal*
[†]*Department of Production and Systems, University of Minho, 4710-057 Braga, Portugal*

Abstract. This paper presents a new reduction-type method for solving semi-infinite programming problems, where the multi-local optimization is carried out with a sequential simulated annealing algorithm, and the finite reduced problem is solved by a penalty technique based on an hyperbolic function. Global convergence is ensured by a line search filter method. Numerical experiments with a set of known problems show that the algorithm is promising.

Keywords: semi-infinite programming, reduction method, penalty function, line search filter method
PACS: 02.60.Pn

INTRODUCTION

The purpose of this paper is to present some numerical experiments with a reduction-type method on solving nonlinear semi-infinite programming (SIP) problems. The SIP problem is an optimization problem, with finitely many decision variables $x \in \mathbb{R}^n$ and a feasible set described by infinitely many constraints, with the form

$$\min f(x) \text{ subject to } g(x,t) \leq 0, \text{ for every } t \in T \tag{1}$$

where $T \subseteq \mathbb{R}^m$ is an infinite index set, $f : \mathbb{R}^n \to \mathbb{R}$ and $g : \mathbb{R}^n \times T \to \mathbb{R}$ are twice continuously differentiable functions with respect to x, and g is a continuously differentiable function with respect to t. There exists a wide range of engineering applications that can be formulated as SIP problems. See, for example, [5, 13, 15, 16, 18]. At present, there are a large variety of numerical methods to solve (1). The reader is referred to the interesting surveys [3, 7] and to the references therein included. A reduction method is based on the local reduction theory presented in [4]. Under some mild assumptions, a reduction method replaces the SIP problem by a locally reduced finite problem. Thus, the SIP problem can be locally reduced to a finite one. First, all the local maximizers of the constraints have to be computed so that the infinite constraints of the SIP problem are replaced by a finite set of constraints that are locally sufficient to define the feasible region. This is known as a multi-local optimization procedure. Then, a finite programming method is used to solve the reduced finite optimization problem. If a globalization procedure is included in the algorithm, the iterative process is then called a global reduction method. Classical line search methods to globalize the algorithms use merit functions to enforce progress towards the solution. As an alternative to merit functions, Fletcher and Leyffer [2] proposed a filter method, as a tool to guarantee global convergence in algorithms for nonlinear constrained finite optimization.

Here, a reduction-type method that relies on a line search filter method combined with an hyperbolic penalty method is proposed to solve nonlinear SIP problems. The method uses a sequential simulated annealing algorithm to compute the local maximizers of the constraints, a penalty method based on the hyperbolic penalty function to approximately solve the reduced finite optimization problem and the filter technique to ensure progress to optimality.

THE GLOBAL REDUCTION PARADIGM

In this section, we first briefly discuss the local reduction theory [4]. Here, we consider only problems where the set T does not depend on the decision variables. For an approximation to the solution of (1), $\bar{x} \in \mathbb{R}^n$, we consider the so-called lower-level problem

$$\max_{t \in T} g(\bar{x},t) \tag{2}$$

and the set $T^{\bar{x}} = \{t^1, \dots, t^{\bar{L}}\}$ that contains the local solutions to (2) that satisfy

$$|g(\bar{x}, t^l) - g^*| \leq \varepsilon, \ l = 1, \dots, \bar{L}, \tag{3}$$

where $\bar{L}$ represents the number of local maximizers found in (2), ε is a positive constant and g^* is the global solution value of (2). The use of condition (3) aims to generate a finite problem with few constraints that is, locally, equivalent to the SIP problem. If we assume that problem (2) ir regular, the SIP problem can be replaced locally by the following finite reduced problem:

$$\min_{x \in U(\bar{x})} f(x) \text{ subject to } g^l(x) \equiv g(x, t^l(x)) \leq 0, \ l = 1, \dots, \bar{L} \tag{4}$$

where $U(\bar{x})$ is an open neighborhood of $\bar{x}$. Thus, the main steps of a global reduction-type algorithm are as follows: (i) compute the set $T^{\bar{x}}$; (ii) compute a descent direction by approximately solving (4); (iii) implement a line search method to guarantee convergence from an arbitrary initial approximation. The remaining part of this section presents details of the proposed algorithm.

Multi-local procedure

Our multi-local procedure to compute the set $T^{\bar{x}}$ is a sequential simulated annealing algorithm, meaning that a sequence of global optimization problems is solved to compute sequentially the local solutions of the problem (2) that satisfy (3). The objective function of each optimization problem is obtained by applying a function stretching technique, as outline in [9], to the objective function of the previous problem in the sequence. This technique aims to stretch the neighborhood of an already computed solution downwards assigning lower function values to those points to prevent the convergence of the global optimization method to that computed solution [10]. Each global optimization problem of the sequence is solved by ASA, a well-known variant of the simulated annealing algorithm [6]. The multi-local procedure terminates when for a predefined set of consecutive iterations no other solution is encountered.

Hyperbolic penalty framework

The most used methods for solving (4) are Sequential Quadratic Programming, with L_1 and L_∞ merit functions, and projected Lagrangian methods [1, 14]. Our proposal here is to use the 2-parameter hyperbolic penalty function [19]. This is a continuously differentiable function, and at most $K_{\max}$ iterations of a BFGS quasi-Newton method are implemented to compute a direction d that yields a decrease on the hyperbolic penalty function

$$P(x) = f(x) + \sum_{l=1}^{\bar{L}} \lambda_l g^l(x) + \sqrt{\lambda_l^2 [g^l(x)]^2 + \tau_l^2} \tag{5}$$

where $\lambda_l \geq 0$ and $\tau_l \geq 0$, for $l = 1, \dots, \bar{L}$, are penalty parameters. These parameters are updated for each $l = 1, \dots, \bar{L}$ as below

$$\begin{cases} \lambda_l^{k+1} = r\lambda_l^k \text{ and } \tau_l^{k+1} = \tau_l^k, & \text{if } \max_{l=1,\dots,\bar{L}} g^l(\bar{x}) \geq 0 \\ \tau_l^{k+1} = q\tau_l^k \text{ and } \lambda_l^{k+1} = \lambda_l^k, & \text{otherwise} \end{cases}, \ k = 0, \dots, K_{\max}.$$

After the search direction d has been computed, a line search technique chooses a step size and determines the new approximation $\bar{x} = \bar{x} + \alpha d$. The procedure herein implemented to decide which trial step size is accepted is a line search filter method. We remark that when solving the reduced problem (4), the selection of $K_{\max} = 1$ guarantees that the optimal set $T^{\bar{x}}$ does not change. When $K_{\max} > 1$ is chosen, the values of the maximizers $t^1, \dots, t^{\bar{L}}$ may change as long as $\bar{x}$ changes in the iterative process, even if $\bar{L}$ does not change. Thus, a local adaptation procedure should be incorporated into the algorithm. Our implementation of this adaptation procedure randomly generates $5m$ points in a predefined neighborhood of each maximizer, t^l, and the one with largest g value will replace the maximizer t^l if the corresponding function value exceeds $g^l(x)$. This type of procedure has produced efficient algorithms [5, 10].

270

The line search filter method

Traditionally, in a line search method, a trial step size is accepted if the corresponding approximation provides a sufficient reduction of a merit function [10]. This type of function depends on a positive penalty parameter that aims to make the constraints violation of the same order of the objective function value. Exponential and L_2-exponential penalty functions have been successfully used in the past [10]. However, the choice of proper penalty parameter values throughout the iterative process is a complex and not totally defined issue. To avoid the use of a merit function and the updating of the penalty parameter, while promoting global convergence from arbitrary initial approximations, Fletcher and Leyffer [2] proposed a filter technique. The notion of a filter is based on that of dominance presented in multi-objective optimization. A point x is said to dominate a point y if $f(x) \leq f(y)$ and $\theta(x) \leq \theta(y)$, where f is the objective function and $\theta(x) = \|\max_{t \in T}(0, g(x,t)\|_2$ is used to measure constraints violation. In a filter framework, a new approximation to the solution is accepted if it is not dominated by any other approximations in the filter. Thus, a filter is a set that contains pairs (θ, f) that are prohibited for a successful approximation. The implementation of a filter method within a line search strategy considers a backtracking procedure, where a decreasing sequence of positive step sizes α^j, $j = 0, 1, \ldots$ is tried until a new trial iterate $\bar{x}^j = \bar{x} + \alpha^j d$ is acceptable according to appropriate acceptance conditions. These conditions impose a sufficient decrease in one of the following measures:

$$\theta(\bar{x}^j) \leq (1 - \gamma_\theta)\theta(\bar{x}) \text{ or } f(\bar{x}^j) \leq f(\bar{x}) - \gamma_f \theta(\bar{x}) \tag{6}$$

where $\gamma_\theta, \gamma_f \in (0,1)$ are fixed constants. On the other hand, if

$$\theta(\bar{x}) \leq \theta_{\min}, \ \nabla f(\bar{x})^T d < 0 \text{ and } \alpha^j(-\nabla f(\bar{x})^T d)^{s_f} > \delta(\theta(\bar{x}))^{s_\theta}, \tag{7}$$

are satisfied, for fixed positive constants $\theta_{\min}$, δ and $s_\theta > 1, s_f > 2s_\theta$, then the trial approximation $\bar{x}^j$ is acceptable only if a sufficient decrease in f is verified

$$f(\bar{x}^j) \leq f(\bar{x}) + \mu_f \alpha^j \nabla f(\bar{x})^T d \tag{8}$$

for $\mu_f \in (0, 0.5)$ [17]. The filter is initialized with pairs that have $\theta \geq \theta_{\max} > 0$. If the acceptable approximation satisfies the condition (6), the filter is augmented; otherwise (conditions (7) and (8) hold) it remains unchanged. The reader is referred to [8, 11] for more details concerning the filter strategy.

NUMERICAL EXPERIMENTS AND CONCLUSIONS

The herein proposed reduction method based on an hyperbolic penalty line search filter method was implemented in the C programming language on a Pentium II, Celeron 466 Mhz with 64Mb of RAM. For the computational tests we selected eight test problems - problems 1, 2, 3, 4, 5, 6, 7, 14 ($c = 1.1$) - described in full detail in the Appendix of [1] (using the initial approximations therein reported). The coded algorithm terminates when the norm of directional derivative of the Lagrangian function and the maximum constraint violation are both sufficiently small (with a tolerance of 10^{-5}). A limit on the number of iterations is also imposed (100). The values for the other parameters in the algorithm are: $\varepsilon = 5.0$, $\lambda_l^0 = 10$ and $\tau_l^0 = 10$, for all $l = 1, \ldots, \bar{L}$, $r = \sqrt{10}$, $q = 0.1$, $\theta_{\max} = 10^4 \max\{1, \theta(x^0)\}$, $\theta_{\min} = 10^{-4} \max\{1, \theta(x^0)\}$, $\gamma_\theta = \gamma_f = 10^{-5}$, $\delta = 1$, $s_\theta = 1.1$, $s_f = 2.3$ and $\mu_f = 10^{-4}$.

TABLE 1. Computational results for $K_{\max} = 5$

P#	n	$|\mathbf{T}^*|$	$\mathbf{f}^*$	$\mathbf{N}_{rm}$	$\mathbf{N}_{ml}$
1	2	2	$-2.50395E-01$	19	44
2	2	2	$2.43089E+00$	3	4
3	3	2	$7.49903E+00$	12	13
4	6	2	$2.70016E+00$	35	47
5	3	2	$8.25798E+00$	6	7
6	2	1	$9.71591E+01$	3	4
7	3	1	$1.00027E+00$	3	4
14	2	1	$2.20021E+00$	2	3

Table 1 presents the results obtained when $K_{\max} = 5$. In the table, **P#** refers to the problem number, $|\mathbf{T}^*|$ represents the number of maximizers satisfying (3) at the final iterate, $\mathbf{f}^*$ is the objective function value at the final iterate, $\mathbf{N}_{rm}$

271

and N_{ml} give the number of iterations needed by the reduction method and the number of iterations in the multi-local optimization, respectively. We also include Table 2 so that a comparison between the herein proposed reduction method and a selection of other well-known reduction methods is possible. The superscripts PF a), PF b), PC, TFI and CW refer to the results obtained in our previous works [10, 11], in Price and Coope [12], in Tanaka, Fukushima and Ibaraki [14] and in Coope and Watson [1], respectively. In the table, "-" means unavailable information.

TABLE 2. Results from other reduction methods

P#	n	m	$N_{rm}^{PF\,a)}$	$N_{ml}^{PF\,a)}$	$N_{rm}^{PF\,b)}$	$N_{ml}^{PF\,b)}$	N_{rm}^{PC}	N_{ml}^{PC}	N_{rm}^{TFI}	N_{ml}^{TFI}	N_{rm}^{CW}
1	2	1	48	60	47	81	17	21	17	19	16
2	2	2	3	38	4	5	8	10	5	11	7
3	3	1	3	13	21	105	11	23	9	12	10
4	6	1	74	1203	38	52	57	119	8	27	20
5	3	1	41	1000	-	-	8	14	4	9	4
6	2	1	7	8	8	9	27	87	16	19	9
7	3	2	8	9	7	8	9	14	2	4	3
14	2	1	10	95	-	-	6	7	5	8	5

We have implemented an hyperbolic penalty function and incorporated a line search filter strategy into a reduction-type method for solving nonlinear SIP problems. The preliminary numerical tests with a set of small problems are encouraging. Further experiments, in particular with problems with more than one constraint function g, and with practical engineering applications are required. Another future challenge is the extension of this type of reduction method to the generalized SIP problem.

REFERENCES

1. I. D. Coope and G. A. Watson, A projected Lagrangian algorithm for semi-infinite programming, *Mathematical Programming* **32**, 337–356 (1985).
2. R. Fletcher and S. Leyffer, *Nonlinear programming without a penalty function*, Mathematical Programming **91** (2002), 239–269.
3. M. A. Goberna and M. A. López (Eds.), *Semi-Infinite Programming. Recent Advances*, Nonconvex Optimization and Its Applications, vol. 57, Springer-Verlag, 2001.
4. R. Hettich and H. Th. Jongen, "Semi-infinite programming: conditions of optimality and applications", in *Lectures Notes in Control and Information Science - Optimization Techniques*, vol. 7, edited by J.Stoer, Springer-Verlag, 1978, pp. 1–11.
5. R. Hettich and K. O. Kortanek, Semi-infinite programming: Theory, methods and applications, *SIAM Review* **35**, 380–429 (1993).
6. L. Ingber, Very fast simulated re-annealing, *Mathematical and Computer Modelling*, **12**, 967–973 (1989).
7. M. López, G. Still, Semi-infinite programming, *European Journal of Operations Research*, **180**, 491–518 (2007).
8. P.Y. Nie, M.Y. Lai, S.J. Zhu, P.A. Zhang, A Line Search Filter Approach for the System of Nonlinear Equations, *Computers and Mathematics with Applications*, **55(9)**, 2134–2141 (2008).
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. 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*, DOI: 10.1080/02331930701764072 (2008).
11. 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"*, edited by L. Sakalauskas, G.W. Weber, E.K. Zavadskas, ISBN: 978-9955-28-283-9, 2008, pp. 174–179.
12. C. J. Price and I. D. Coope, Numerical experiments in semi-infinite programming, *Computational Optimization and Applications* **6**, 169–189 (1996).
13. O. Stein and G. Still, Solving semi-infinite optimization problems with interior point techniques, *SIAM Journal on Control and Optimization* **42**, 769–788 (2003).
14. Y. Tanaka, M. Fukushima, and T. Ibaraki, A comparative study of several semi-infinite nonlinear programmnig algorithms, *European Journal of Operations Research* **36**, 92–100 (1988).
15. A. I. F. Vaz, E. M. G. P. Fernandes, and M. P. S. F. Gomes, Robot trajectory planning with semi-infinite programming, *European Journal of Operational Research* **153**, 607–617 (2004).
16. F. G. Vázquez, J.-J. Rückmann, O. Stein, and G. Still, Generalized semi-infinite programming: a tutorial, *Journal of Computational and Applied Mathematics*, **217**, 394–419 (2008).
17. A. Wächter and L. T. Biegler, Line search filter methods for nonlinear programming: motivation and global convergence, *SIAM Journal on Optimization* **16**, 1–31 (2005).
18. G.-W. Weber and A. Tezel, On generalized semi-infinite optimization of genetic network, *TOP* **15**, 65–77 (2007).

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