Numerical Experiments with a Continuous L_2-exponential Merit Function 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. Here, we present some numerical experiments with a reduction method for solving nonlinear semi-infinite programming (SIP) problems. The method relies on a line search technique to ensure a sufficient decrease of a L_2-exponential merit function. The proposed merit function is continuous for SIP and improves the algorithm efficiency when compared with other previously tested merit functions. A comparison with other reduction methods is also included.

Keywords: semi-infinite programming, reduction method, merit function
PACS: 02.60.Pn

INTRODUCTION

The purpose of this paper is to present some numerical experiments with a reduction-type method to solve a class of nonlinear semi-infinite programming (SIP) problems, where a sufficient decrease on a continuous L_2-exponential merit function is enforced by a line search technique. The emphasis is on the performance of the herein proposed merit function. We consider the SIP problem in 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 a compact 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, g is a continuously differentiable function with respect to t. Here, we assume that the set T is not dependent on x. This problem has a finite number of variables and an infinite number of constraints, and is very important in engineering applications. Robot trajectory planning, computer aided design, air pollution control and production planning are examples of SIP problems. For a thorough review of applications the reader is referred to [1, 2]. The most extensively used numerical methods replace problem (1) by a sequence of finite problems, using a discretization process, an exchange method or a reduction method. Under some mild assumptions, a reduction method replaces the SIP problem by a locally reduced finite problem. 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 problem.

This paper presents the use of a new reduction algorithm, where a global stochastic method (simulated annealing) combined with a function stretching technique is used to solve the multi-local problem. The finite reduced problem is then solved by a penalty technique based on an exponential function. Global convergence is ensured by a line search technique.

THE GLOBAL REDUCTION METHOD

A reduction method for solving (1) is based on the local reduction theory proposed by Hettich and Jongen [3]. For a given approximation to the solution $\bar{x} \in \mathbb{R}^n$, consider the following so-called lower-level problem

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

Let $T^{\bar{x}} = \{t^1, \ldots, t^{|L(\bar{x})|}\}$ be the set of the local solutions of (2) that satisfy the condition

$$|g(\bar{x},t^l) - g^*| \leq \tau,\ l \in L(\bar{x}), \tag{3}$$

CP 1060, *International Electronic Conference on Computer Science 2007 (IeCCS 2007)*
edited by T. E. Simos and G. Psihoyios

where $L(\bar{x})$ represents the index set of $T^{\bar{x}}$, τ is a positive constant and g^* is the global solution value of (2). Condition (3) aims to generate a finite problem with few constraints that is, locally, equivalent to the SIP problem. Similar conditions have been proposed in the past, for example, in [4]. When the problem (2) is regular[1], and for an open neighborhood of $\bar{x}$, $U(\bar{x})$, the SIP problem can be replaced locally by the following finite reduced problem:

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

In what follows, we assume that problem (2) ir regular. A reduction method contains two procedures: the multi-local optimization and the reduced finite optimization. If a globalization procedure is carried out, the iterative process is called a global reduction method (GRM). The algorithm is presented below.

Algorithm 1 *(GRM algorithm)*

Input*: initial approximation $\bar{x}$, τ, K;*

Step 1: Multi-local procedure*: Compute the local solutions of (2) that satisfy (3);*

Step 2: Finite optimization procedure*: Apply at most K iterations of a finite programming method to (4) to obtain a direction d;*

Step 3: Globalization procedure*: Use a line search method to obtain a new approximation, x, that yields a sufficient reduction in the merit function along the direction d;*

Step 4: If the termination criteria are not satisfied, then $\bar{x} \leftarrow x$ and go to Step 1.

Our multi-local procedure is a sequential simulated annealing algorithm, meaning that a sequence of global optimization problems has to be solved to compute sequentially the local solutions of the problem (2). This scheme defines the outer iterative process. The objective function of each optimization problem is obtained by applying the function stretching technique to the objective function of the previous problem in the sequence. This technique stretches the neighborhood of an already computed solution downwards assigning lower function values to those points and preventing the convergence of the global optimization method to the previously computed solution [5]. The inner iterative process consists of a simulated annealing algorithm and aims to compute a solution of each global optimization problem of the sequence. Our multi-local algorithm terminates when no more global solutions are detected for a fixed number of iterations.

The most used methods for solving (4) have been Sequential Quadratic Programming, with L_1 and L_∞ merit functions, and projected Lagrangian methods [4, 6]. When solving the reduced problem (4), a classical reduction-type method considers $K = 1$ to guarantee that the optimal set $T^{\bar{x}}$ does not change. When $K > 1$, the values of the maximizers $t^1, \ldots, t^{|L(\bar{x})|}$ may change if $\bar{x}$ changes along the iterative process, even if $|L(\bar{x})|$ does not change. However, if a local adaptation procedure is incorporated into the algorithm, the use of $K > 1$ in solving the problem (4) has been shown to improve efficiency [1]. Our implementation of a local adaptation procedure randomly generates $5m$ points in the neighborhood of each maximizer, t^l, and the one with largest g value will replace the maximizer t^l if its function value exceeds $g^l(x)$. In our algorithm, a BFGS quasi-Newton method is used to compute a direction d that yields a decrease on the penalty exponential function

$$P(x, \eta, \lambda) = f(x) + \frac{1}{\eta} \sum_{l=1}^{|L(\bar{x})|} \lambda_l \left(e^{\eta g^l(x)} - 1 \right), \tag{5}$$

in at most K iterations, where λ_l is the Lagrange multiplier associated with the constraint $g^l(x)$ and η is a positive penalty parameter. Each function $g^l(x)$ in (5) is obtained by incorporating the solution value t^l, of (2), into $g(x,t)$.

To promote global convergence in the GRM, a line search method is used. Here, a continuous L_2-exponential merit function for SIP is adopted,

$$L_2^{\exp}(x, \mu, \nu_1, \nu_2) = f(x) + \frac{\nu_1}{\mu}(e^{\mu\theta(x)} - 1) + \frac{\nu_2}{2}(e^{\mu\theta(x)} - 1)^2, \text{ where } \theta(x) = \max_{t \in T}[g(x,t)]_+ \tag{6}$$

[1] Problem (2) is said to be regular if all critical points are nondegenerate.

TABLE 1. Computational results with $L_2^{\exp}$ for $K=1$ and $K=5$

		$K=1$					$K=5$				
P#	n	$\lvert\mathbf{T}^*\rvert$	$\mathbf{f}^*$	$\mathbf{N}_{rm}$	$\mathbf{N}_{ml}$	$\lvert\mathbf{D}\rvert$	$\lvert\mathbf{T}^*\rvert$	$\mathbf{f}^*$	$\mathbf{N}_{rm}$	$\mathbf{N}_{ml}$	$\lvert\mathbf{D}\rvert$
1	2	2	$-2.57(-01)$	52	100	$9.1(-06)$	1	$-3.13(-01)$	4	6	$1.1(-12)$
2	2	2	$4.76(-01)$	3	38	$2.7(-13)$	2	$1.95(-01)$	3	38	$8.8(-15)$
3	3	2	$5.34(+00)$	21	22	$1.6(-12)$	2	$5.33(+00)$	3	13	$1.0(-06)$
4	3	1	$6.76(-01)$	52	573	$1.4(-07)$	1	$6.50(-01)$	11	126	$1.2(-06)$
4	6	1	$6.17(-01)$	$25^{(1)}$	592	$3.2(-07)$	2	$6.74(-01)$	24	314	$2.7(-11)$
4	8	2	$6.19(-01)$	15	157	$7.4(-13)$	1	$6.16(-01)$	28	484	$9.7(-06)$
5	3	2	$4.25(+00)$	5	54	$1.5(-06)$	2	$4.14(+00)$	$8^{(2)}$	113	$7.1(-14)$
6	2	1	$9.72(+01)$	44	55	$4.3(-07)$	1	$9.72(+01)$	7	8	$8.7(-06)$
7	3	1	$9.99(-01)$	42	43	$8.0(-06)$	1	$1.00(+00)$	8	9	$4.0(-06)$
14	2	1	$7.39(-06)$	2	3	$6.6(-10)$	1	$4.43(-06)$	2	3	$1.1(-07)$

$^{(1)}$ $x^0 = (0,0.5,0,0,0,0)^T$ | $^{(2)}$ $x^0 = (0.5,0.5,0)^T$

with $[g(x,t)]_+ = \max\{0, g(x,t)\}$, μ is a positive penalty parameter and $\nu_1, \nu_2 \geq 0$. Clearly $\theta(x)$ is the infinity norm of the constraint violations, hence $L_2^{\exp}$ is continuous for every $x \in \mathbb{R}^n$. If the sufficient descent condition

$$L_2^{\exp}(\bar{x}+\alpha d, \mu, \nu_1, \nu_2) \leq \bar{L}_2^{\exp}(\bar{x}, \mu, \nu_1, \nu_2) + \sigma\,\alpha\, D\bar{L}_2^{\exp}(\bar{x}, \mu, \nu_1, \nu_2; d), \tag{7}$$

for $0 < \sigma < 1/2$, holds with $\alpha = 1$, for the direction d computed from (5) for $K > 1$, then $x = \bar{x} + d$ is accepted as the new approximation to the SIP problem. Otherwise, the algorithm uses the direction computed if $K = 1$ was imposed, selects α as the first element of the sequence $\{1, 1/2, 1/4, \ldots\}$ to satisfy (7), and sets $x = \bar{x} + \alpha d$. The scalar $D\bar{L}_2^{\exp}(\bar{x}, \mu, \nu_1, \nu_2; d)$ is the directional derivative of the merit function at $\bar{x}$ in the direction d. When a new point is computed, the multi-local procedure has to be called to obtain the maximizers of g at x, so that $L_2^{\exp}$ is obtained. We note that $L_2^{\exp}$ is different from $\bar{L}_2^{\exp}$ since they have different sets of constraints.

NUMERICAL EXPERIMENTS AND CONCLUSIONS

The proposed GRM 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 [6] (using the initial approximations therein reported). The GRM is terminated when the directional derivative of the merit function at the new point x in the direction d is sufficiently small, and the point is feasible,

$$|DL_2^{\exp}(x, \mu, \nu_1, \nu_2; d)| \leq \varepsilon_D \text{ and } \max\{g(x, t^l), l \in L(x)\} \leq \varepsilon_g,$$

for positive and small ε_D and ε_g. This choice of termination criteria permits a comparison with the results in the literature. A maximum number of iterations, nit_{max}, is also imposed to stop the algorithm. The used constants were: $\varepsilon_D = 10^{-5}$, $\varepsilon_g = 10^{-5}$, $\sigma = 10^{-4}$, $\tau = 5.0$, $nit_{max} = 100$. Table 1 aims to compare the results obtained when only one iteration is permitted in (5) - $K = 1$ - and when $K = 5$. 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, $\mathbf{N}_{rm}$ and $\mathbf{N}_{ml}$ represent the number of iterations needed by the reduction method and the number of multi-local optimization calls, respectively. The columns headed $|\mathbf{D}|$ give the final value of the directional derivative (with the convention that $9.1(-06)$ means 9.1×10^{-6}, and so on).

For problem 4 with $n = 6$ (when $K = 1$) and problem 5 with $n = 3$ (when $K = 5$), different initial approximations had to be used to be able to converge to the required solution. Allowing more than one iteration to be done in the finite reduced optimization improves the GRM efficiency (see Table 1). We have previously tested this global reduction method with other two merit functions of the exponential type: one is similar to the exponential function of finite optimization, see (5), and is not continuous for SIP; the other is a continuous extension of the exponential function for SIP and it consists of the first two terms of the right hand side of (6). The framework based on the herein presented merit function is more efficient. We refer to future work for details.

We also include Table 2 so that a comparison between the reduction method based on the proposed L_2-exponential merit function and a selection of other reduction methods is possible. The superscripts PC, TFI and CW refer to the

TABLE 2. Numerical results obtained by other reduction methods

P#	n	m	$\mathbf{N}_{rm}^{\mathrm{PC}}$	$\mathbf{N}_{ml}^{\mathrm{PC}}$	$\lvert\mathbf{D}^{\mathrm{PC}}\rvert$	$\mathbf{N}_{rm}^{\mathrm{TFI}}$	$\mathbf{N}_{ml}^{\mathrm{TFI}}$	$\lvert\mathbf{D}^{\mathrm{TFI}}\rvert$	$\mathbf{N}_{rm}^{\mathrm{CW}}$	$\lvert\mathbf{D}^{\mathrm{CW}}\rvert$
1	2	1	17	21	8.2(−06)	17	19	4.8(−07)	16	5.7(−06)
2	2	2	8	10	1.4(−08)	5	11	2.7(−08)	7	2.5(−10)
3	3	1	11	23	1.3(−06)	9	12	5.5(−08)	10	6.2(−12)
4	3	1	10	11	1.9(−06)	5	15	2.7(−07)	5	5.4(−08)
4	6	1	57	119	7.7(−06)	8	27	7.7(−06)	20	6.4(−06)
4	8	1	84	164	1.0(−07)	3	14	3.4(−06)	16	7.4(−06)
5	3	1	8	14	6.2(−06)	4	9	6.8(−07)	4	6.9(−06)
6	2	1	27	87	5.2(−06)	16	19	1.3(−18)	9	1.1(−08)
7	3	2	9	14	7.0(−09)	2	4	0.0(+00)	3	0.0(+00)
14	2	1	6	7	8.1(−06)	5	8	3.4(−07)	5	8.2(−07)

results obtained in Price and Coope [7], Tanaka, Fukushima and Ibaraki [8] and Coope and Watson [6], respectively. In 50% of the tested problems, our algorithm (with $K = 5$) converged in fewer (reduction method) iterations than the other presented reduction methods. However, in some problems, the number of multi-local calls is larger than those of the other methods, showing that future research will be necessary to improve the overall efficiency of the method. For other numerical methods for SIP, we refer to Goberna and López [9] and the references therein presented. A more recent approach, a semismooth Newton method, is in [10].

We have presented a new continuous merit function that is incorporated into a reduction-type method for solving nonlinear SIP problems. A line search is conducted along a descent direction to yield a sufficient reduction in the proposed L_2-exponential merit function. The preliminary experiments are encouraging. Further experience with testing, in particular using problems with more than one constraint function g, and with real applications will be required. The convergence analysis of the proposed method is not in the scope of this paper, but this is indeed one of our future main concerns. Another future challenge is to extend this type of reduction method to Generalized SIP problems with interesting applications [11].

ACKNOWLEDGMENTS

The authors are grateful to three anonymous referees for their comments and suggestions. This work has been partially suported by the Algoritmi Research Center and by Portuguese FCT grant POCI/MAT/58957/2004.

REFERENCES

1. R. Hettich and K. O. Kortanek, *Semi-infinite Programming: Theory, Methods and Applications*, SIAM Review 35, 1993, 380–429.
2. O. Stein, *Bi-level Startegies in Semi-Infinite Programming*, Nonconvex Optimization and its Applications 71, Kluwer Academic Publishers, 2003.
3. R. Hettich and H. Th. Jongen, *Semi-Infinite Programming: Conditions of Optimality and Applications*, Lecture Notes in Control and Information Science 7, 1978, 1–11.
4. Y. Tanaka, M. Fukushima and T. Ibaraki, *A Comparative Study of Several Semi-Infinite Nonlinear Programming Algorithms*, European Journal of Operational Research 36, 1988, 92–100.
5. A. I. P. N. Pereira and E. M. G. P. Fernandes, *A New Algorithm to Identify all Global Maximizers Based on Simulated Annealing*, in Proceedings of the 6th. World Congress on Structural and Multidisciplinary Optimization, J. Herskowitz, S. Mazorche and A. Canelas (Eds.), ISBN: 85-285-0070-5, 2005.
6. I. D. Coope and G. A. Watson, *A Projected Lagrangian Algorithm for Semi-Infinite Programming*, Mathematical Programming 32, 1985, 337–356.
7. C. J. Price and I. D. Coope, *Numerical experiments in semi-infinite programming*, Computational Optimization and Applications, 6, 1996, 169–189.
8. Y. Tanaka, M. Fukushima and T. Ibaraki, *A Globally Convergence SQP Method for Semi-Infinite Non-Linear Optimization*, Journal of Computational and Applied Mathematics 23, 1988, 141–153.
9. M.A. Goberna and M.A. Lopéz (Eds.), *Semi-Infinite Programming. Recent Advances*, Nonconvex Optimization and Its Applications, Springer-Verlag, vol. 57, 2001.
10. L. Qi, S.-Y. Wu and G. Zhou, *Semismooth Newton Methods for Solving Semi-Infinite Programming Problems*, Journal of Global Optimization, 27, 2003, 215-232.
11. G.-W. Weber and A. Tezel, *On Generalized Semi-Infinite Optimization of Genetic Networks*, TOP, 15, 2007, 65–77.