International Conference
20th EURO Mini Conference
"Continuous Optimization and Knowledge-Based Technologies"
(EurOPT-2008)
May 20–23, 2008, Neringa, LITHUANIA

ISBN 978-9955-28-283-9
L. Sakalauskas, G.W. Weber and
E. K. Zavadskas (Eds.): EUROPT-2008
Selected papers. Vilnius, 2008, pp. 174–179

ON A REDUCTION LINE SEARCH FILTER METHOD FOR NONLINEAR SEMI-INFINITE PROGRAMMING PROBLEMS

Ana Isabel P. N. Pereira[1*], Edite M. G. P. Fernandes[2]

[1]*Department of Mathematics, Polytechnic Institute of Bragança, Portugal*
[2]*Department of Production and Systems, University of Minho, 4710-057 Braga, Portugal*
E-mail: [1]apereira@ipb.pt; [2]emgpf@dps.uminho.pt

Abstract: In this paper, a reduction-type method combined with a line search filter method to solve nonlinear semi-infinite programming problems is presented. The algorithm uses the simulated annealing method equipped with a function stretching technique to compute the maximizers of the constraint, and a penalty technique to solve the reduced finite optimization problem. The filter method is used as an alternative to merit functions to guarantee convergence from poor starting points. Preliminary numerical results with a problems test set are shown.

Keywords: semi-infinite programming, reduction method, line search filter method.

1. Introduction

A reduction-type method for nonlinear semi-infinite programming (SIP) that relies on a line search filter strategy to allow convergence from poor starting points is proposed. The novelty here is the use of the filter methodology to guarantee convergence to the solution within a local reduction framework for SIP. Classical line search methods use merit functions to enforce progress towards the solution. As an alternative to merit functions, Fletcher and Leyffer (Fletcher and Leyffer, 2002; Fletcher *et al.*, 2002) proposed a filter method, as a tool to guarantee global convergence in algorithms for nonlinear constrained finite optimization. The SIP problem is considered to be of the form

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

where $T \subseteq \Re^m$ is a nonempty compact set that contains infinitely many elements, $f : \Re^n \to \Re$ and $g : \Re^n \times T \to \Re$ are nonlinear twice continuously differentiable functions with respect to x and g is a continuously differentiable function with respect to t. We denote the feasible set of SIP problem (1) by $X = \{x \in \Re^n : g(x,t) \leq 0, \forall t \in T\}$. Here, we consider only problems where the set T does not depend on x. There are in the engineering area many problems that can be formulated as SIP problems. Robot trajectory planning (Haaren-Retagne, 1992; Vaz *et al.*, 2004) and air pollution control (Vaz and Ferreira, 2004) are two examples. For a review of other applications, the reader is referred to, *e.g.*, (Reemtsen and Rückmann, 1998; Goberna and Lopéz, 2001; Stein, 2003; Weber and Tezel, 2007). At present, there are a large variety of numerical approaches to solve (1). A reduction approach, based on the local reduction theory proposed by (Hettich and Jongen, 1978), describes, locally, the feasible set of the SIP problem by finitely many inequality constraints. Thus, the SIP problem can be locally reduced to a finite one (at least conceptually, see (Liu, 2007)). We briefly describe the main ideas of a local reduction theory. Given a feasible point $x_k \in \Re^n$, consider the so-called lower level problem

$$\mathbf{O}(x_k): \quad \max_{t \in T} g(x_k, t)$$

where $t^1, t^2, ..., t^{L_k}$ denote its solutions that satisfy the following condition

$$| g(x_k, t^j) - g^* | \leq \delta_O, \quad j = 1,2,..., L_k. \tag{2}$$

* This work has been partially supported by the Algoritmi R & D Centre and by the Portuguese FCT grant POCI/MAT/58957/2004.

The constant δ_O is positive and g^* is the global solution value of problem **O**(x_k). The condition (2) aims to generate a finite optimization problem with a restricted set of constraints that is locally equivalent to the SIP problem (1). If we assume that: i) the iterate $x_k \in X$; ii) the linear independence constraints qualification holds at the solutions $t^1, t^2, ..., t^{L_k}$; and iii) all critical points of problem **O**(x_k) are nondegenerate; then each local solution of problem **O**(x_k) is nondegenerate and consequently an isolated local maximizer. Since the set T is compact, problem **O**(x_k) has finitely many solutions satisfying (2). The implicit function theorem can be applied and there exist open neighbourhoods U_k of x_k and V_j of t^j, and implicit functions $t^1(x), ..., t^{L_k}(x)$ such that:

i) $t^j : U_k \to V^j \cap T$, for $j = 1,2,...,L_k$; ii) $t^j(x_k) = t^j$, for $j = 1,2,...,L_k$; iii) for all $x \in U_k$, $t^j(x)$ is a nondegenerate and isolated local maximizer of problem **O**(x_k). We may conclude that $\{x \in U_k : g(x,t) \le 0, \forall t \in T\} \Leftrightarrow \{x \in U_k : g^j(x) \equiv g(x, t^j(x)) \le 0, j = 1,...,L_k\}$, *i.e.*, it is possible to replace the infinite set of constraints by a finite set that is locally sufficient to define the feasible region. Thus the SIP problem (1) is locally equivalent to the so-called reduced problem

$$\textbf{r-P}(x_k): \min_{x \in U_k} f(x) \quad \text{subject to} \quad g^j(x) \le 0, \quad j = 1,2,...,L_k .$$

The paper is organized as follows. In Section 2 we describe the proposed reduction algorithm and justify the line search filter approach. Section 3 contains the preliminary numerical experiments and we conclude the paper in Section 4.

2. The reduction method for SIP

At each iteration, a reduction approach follows two main phases. First, for an approximation x_k , the lower level problem **O**(x_k) has to be solved. Then, the reduced finite problem **r-P**(x_k) is solved to obtain a new approximation x_{k+1}. To guarantee convergence from any starting point a globalization scheme is incorporated. The global reduction method is defined by:

Algorithm 1 (Global reduction method)
Input: given initial iterate x_0, and constants δ_O, $k_{\max}$, set $k = 0$
Compute the local solutions of problem **O**(x_k) that satisfy (2)
Compute a search direction d_k by applying at most $k_{\max}$ iterations of a finite nonlinear optimization method to the reduced problem **r-P**(x_k)
Implement a line search filter method to decide which trial step size is acceptable to define x_{k+1}
If the termination criteria are not satisfied, set $k = k + 1$ and go to step 1.

The termination criteria are defined by the two conditions $\| DL(x_k, \lambda_k) \| \le \varepsilon_{Lag}$ and $\max\{g^j(x_k), j = 1,...,L_k\} \le \varepsilon_g$, for sufficiently small positive constants $\varepsilon_{Lag}, \varepsilon_g$. The function $L(x_k, \lambda_k)$ is the Lagrangian and DL represents the directional derivative of L. Estimates for the Lagrange multiplier λ_k are obtained from the solution of the finite reduced problem. We also limit the maximum number of iterations on the reduction method, $N_{\max}$.

2.1. A stochastic approach for the lower level problem

To compute all the local solutions of problem **O**(x_k) that satisfy (2), a stochastic global optimization method is used. In general, a global optimization method finds one solution in each run. Although the simulated annealing (SA) method is very efficient in finding a global solution (Ingber, 1989), some problems may appear when the problem has multiple solutions since the algorithm may oscillate between global solutions or get stuck in a nonglobal one. Recently (Parsopoulos *et al.*, 2001) proposed to incorporate a function stretching technique in the particle swarm optimization method to escape from nonglobal solutions. Our proposal, however, considers a local application of the function stretching technique. For example, if t^j is a global maximizer of **O**(x_k) found by SA, then the objective function is transformed into

$$\tilde{g}(t) = \bar{g}(t) - \delta_2 \frac{\text{sign}(g(t^j) - g(t)) + 1}{2\tanh(\xi(\bar{g}(t^j) - \bar{g}(t))} \quad \text{where} \tag{3}$$

$$\bar{g}(t) = g(t) - \frac{\delta_1}{2} \| t - t^j \|_2 \left[\text{sign}(g(t^j) - g(t)) + 1 \right] \quad \text{for all } t \in V_{\pi^j}(t^j)$$

for positive constants δ_1, δ_2, ξ. For simplicity, we use $g(t)$ instead of $g(x_k, t)$. Both transformations (3) aim to stretch g downwards assigning lower function values to all points in the neighbourhood of t^j, $V_{\pi^j}(t^j)$. The ray π^j is chosen so that the condition $|g(t^j) - \tilde{g}_{\max}| \le \delta_O$ is satisfied, where $\tilde{g}_{\max} = \max_{j=1,\dots,2m} \left\{ g(\tilde{t}^j) \right\}$ and $\tilde{t}^1,\dots,\tilde{t}^{2m}$ are $2m$ randomly generated points from the boundary of $V_{\pi^j}(t^j)$. The SA algorithm is then applied to find another maximizer of **O**(x_k), with g replaced by $\tilde{g}$. Thus, the procedure is a sequential global programming algorithm, since a sequence of global optimization problems are solved by SA to find the solutions of the lower level problem that satisfy (2). The mathematical formulation has the form,

$$\max_{t \in T} G_k(t) \equiv \begin{cases} \tilde{g}(t), & \text{if } t \in V_{\pi^j}(t^j), \text{ for all } t^j \in T_k \\ g(t), & \text{otherwise} \end{cases} \tag{4}$$

where T_k is the set of the already detected maximizers. The algorithm is as follows:

Algorithm 2 (Sequential multi-local optimization method)
Input: constants δ_O, π_0, $\pi_{\max}$, iteration k, and set $T_k = \phi$, $j = 0$
If the termination criterion is satisfied stop
Set $l = 0$ and $j = j + 1$

Apply SA algorithm to find a solution, t^j, of problem (4)

Set $l = l + 1$ and $\pi_l = l\pi_0$

Randomly generate points $\tilde{t}^1,\dots,\tilde{t}^{2m}$ from the boundary of $V_{\pi_l}(t^j)$, and find $\tilde{g}_{\max} = \max_{j=1,\dots,2m} \left\{ g(\tilde{t}^j) \right\}$

1. If $|g(t^j) - \tilde{g}_{\max}| > \delta_O$ and $\pi_l \le \pi_{\max}$ go to step 4
2. $T_k = T_k \cup \left\{ t^j \right\}$ and set $\pi^j = \pi_l$ and go to step 1.

The algorithm is terminated if the set T_k does not change for a fixed number of iterations.

2.2. A penalty technique for the finite optimization process

At each iteration k, a search direction is computed by applying a finite optimization method to solve the reduced problem **r-P**(x_k). Sequential quadratic programming, using L_1 and L_∞ merit functions, as well as the projected Lagrangian method are the most used methods (Gramlich *et al.*, 1995; Price and Coope, 1996). The proposal here uses a penalty technique to find the search direction d_k. Using the iterate x_k as a starting point, at most $k_{\max}$ iterations of a BFGS quasi-Newton method are used to find the minimum of the following exponential penalty function

$$P_\eta(x, \lambda) = f(x) + \frac{1}{\eta} \sum_{j=1,\dots,L_k} \lambda^j \left(e^{\eta g^j(x)} - 1 \right),$$

where λ^j is the multiplier associated with the constraint $g^j(x)$ and η is a positive penalty parameter. Besides the limit on the number of iterations, $k_{\max}$, the penalty algorithm terminates when the deviation between two consecutive iterates is smaller than ε_x.

2.3. The line search filter method

In a line search context, and after the search direction has been computed, the algorithm proceeds iteratively, choosing a step size at each iteration and determining the new iterate $x_{k+1} = x_k + \alpha_k d_k$. This section describes a two-entry filter method based on a line search approach. Traditionally, a trial step size α_l is accepted if the corresponding iterate $x_k^l = x_k + \alpha_l d_k$ provides a sufficient reduction of a merit function. This type of function depends on a positive penalty parameter that should be updated throughout the iterative process. To avoid the use of a merit function and the updating of the penalty parameter (Fletcher and Leyffer, 2002) proposed a filter technique in order to promote global convergence. Here, a trial iterate is accepted if it improves the objective function or the constraints violation, instead of a combination of these two measures in a merit function. The filter acceptance criteria are less demanding than the usual enforcement of decrease present in merit function approach and allow in general larger steps to be carried out. The notion of filter is based on that of dominance, *i.e.*, a point x^+ is said to dominate a point x^- whenever $f(x^+) \le f(x^-)$ and $\theta(x^+) \le \theta(x^-)$, where $\theta(x) = \| \max_{t \in T}(0, g(x,t)) \|_2$ measures constraints violation. Thus, a filter method aims to accept a new trial iterate if it is not dominated by any other iterate in the filter.

After a search direction d_k has been computed, the line search method considers a backtracking procedure, where a decreasing sequence of positive step sizes α_l, $l = 0,1,...$ is tried until an acceptance criterion is satisfied. We use l to denote the iteration counter for the inner iteration. Line search methods that use a merit function ensure sufficient progress towards the solution by imposing that the merit function value at each new iterate satisfies an Armijo condition with respect to the current iterate. Based on this idea, a trial iterate $x_k^l = x_k + \alpha_l d_k$ is acceptable, during the backtracking line search filter method, if it leads to sufficient progress in one of the following measures compared to the current iterate, *i.e.*, if

$$\theta(x_k^l) \le (1 - \gamma_\theta)\theta(x_k) \quad \text{or} \quad f(x_k^l) \le f(x_k) - \gamma_f \theta(x_k) \tag{5}$$

holds for fixed constants $\gamma_\theta, \gamma_f \in (0,1)$. However, to prevent convergence to a feasible point that is nonoptimal, and whenever, for the current iterate $\theta(x_k) \le \theta_{\min}$, and the following conditions

$$\nabla f(x_k)^T d_k < 0 \quad \text{and} \quad \alpha_l (-\nabla f(x_k)^T d_k)^{s_f} > \delta(\theta(x_k))^{s_\theta}, \tag{6}$$

hold, then the trial iterate has to satisfy the Armijo condition

$$f(x_k^l) \le f(x_k) + \mu_f \alpha_l \nabla f(x_k)^T d_k \tag{7}$$

instead of (5), to be acceptable, for some fixed positive $\theta_{\min}$ and constants $s_\theta > 1$, $s_f > 2s_\theta, \delta > 0$ and $\mu_f \in (0,0.5)$ (Wächter and Biegler, 2005).

At each iteration k, the algorithm maintains a filter denoted by $\Im_k = \{(\theta, f) \in \Re^2 : \theta \ge 0\}$ that contains pairs of constraints violation and function value that are prohibited for a successful trial iterate in iteration k. Thus, a trial iterate x_k^l is rejected if $(\theta(x_k^l), f(x_k^l)) \in \Im_k$. For some positive $\theta_{\max}$, the filter is initialized to

$$\Im_0 = \{(\theta, f) \in \Re^2 : \theta \ge \theta_{\max}\} \tag{8}$$

and whenever the accepted trial iterate satisfies the conditions (5), the filter is augmented using

$$\Im_{k+1} = \Im_k \cup \{(\theta, f) \in \Re^2 : \theta \ge (1 - \gamma_\theta)\theta(x_k) \quad \text{and} \quad f \ge f(x_k) - \gamma_f \theta(x_k)\}. \tag{9}$$

In practice, only the limit values of (9) are updated and stored. However, if the accepted trial iterate satisfies conditions (6) and (7) then the filter remains unchanged. This procedure ensures that the algorithm cannot cycle between points that alternatively decrease the constraints violation and the objective function value. When the backtracking line search cannot find a trial step size α_l that satisfies the described acceptance criteria, since $\alpha_l < \alpha_{\min}$, for a fixed constant $\alpha_{\min}$, the algorithm resets the filter using (8).

Algorithm 3 (line search filter method)

Input: search direction d_k and constants $\theta_{\min}$, $\theta_{\max}$, $\alpha_{\min}$, γ_θ, γ_f, δ, s_θ, s_f, μ_f

Set $\alpha_1 = 1$ and $l = 1$

1. If $\alpha_l < \alpha_{\min}$, reset the filter using (8) and stop; otherwise compute $x_k^l = x_k + \alpha_l d_k$
2. If $(f(x_k^l), \theta(x_k^l)) \in \Im_k$ reject α_l and go to step 5
3. case I: If $\theta(x_k) \le \theta_{\min}$ and conditions (6) and (7) hold, accept α_l and go to step 6; otherwise go to step 5
4. case II: If (5) holds, accept α_l and go to step 6; otherwise go to step 5
5. Set $\alpha_{l+1} = \alpha_l / 2$, $l = l + 1$ and go to step 2
6. Set $\alpha_k = \alpha_l$ and $x_{k+1} = x_k^l$
7. If (5) held, augment the filter using (9); otherwise leave the filter unchanged.

3. Computational results

For the computational experiences we consider six test problems from (Price and Coope, 1996) (problems 1, 2, 3, 4, 6, 7). The initial approximations are the ones therein reported. We fix the following constants: $\delta_O = 5$, $\delta_1 = 100$, $\delta_2 = 1$, $\xi = 10^{-3}$, $N_{\max} = 100$, $\varepsilon_{Lag}, \varepsilon_x, \varepsilon_g = 10^{-5}$, $\pi_0 = 0.25$, $\pi_{\max} = 1$, $\theta_{\min} = 10^{-4} \max\{1, \theta(x_0)\}$, $\theta_{\max} = 10^4 \max\{1, \theta(x_0)\}$, $\alpha_{\min} = 10^{-10}$, $\gamma_\theta, \gamma_f = 10^{-5}$, $\delta = 1$, $s_\theta = 1.1$, $s_f = 2.3$, $\mu_f = 10^{-4}$. For the two cases $k_{\max} = 1$ and $k_{\max} = 5$, Table 1 reports: the number of the tested problem (P), the number of variables (n), the dimension of the set T (m), the number of maximizers satisfying (2) at the final iterate ($|T^*|$), the objective function value at the final iterate (f^*), the number of iterations needed by the presented variant of a reduction method (k_{RM}) and the number of problems **O**(x_k) solved (k_O). Columns with DL contain the magnitude of the directional derivative of the Lagrangian function at the final iterate. Comparing the two presented cases of the reduction line search filter method, we may conclude that $k_{\max} = 5$ requires fewer reduction method iterations and fewer problems **O**(x_k) solved.

4. Conclusions

We have presented a variant of a reduction method for solving semi-infinite programming problems. The novelty here is related with the strategy used to ensure global convergence from any starting point. A line search filter method is used. The filter technique follows the methodology presented in (Fletcher and Leyffer, 2002) although it is based on a line search technique as outline in (Wächter and Biegler, 2005). At the present, the proposed algorithm is for standard SIP problems, but it will be extended to generalized problems. Further research will also consider the implementation of a discretization method, at the beginning of the procedure, in order to find a good initial approximation to the SIP problem. This strategy will lead to a two-phase method.

Table 1. Numerical results obtained by the reduction line search filter method

				$k_{\max} = 1$ [a]				$k_{\max} = 5$ [b]			
P	n	m	$\|T^*\|$	f^*	k_{RM}	k_O	DL	f^*	k_{RM}	k_O	DL
1	2	1	2	–2.52513E-01	100	194	1.92E-05	–2.51618E-01	47	81	7.62E-07
2	2	1	2	4.02110E-01	7	8	8.55E-13	1.95428E-01	4	5	4.76E-13
3	3	1	2	5.33515E+00	40	41	9.29E-06	5.33469E+00	21	105	3.66E-08
4	6	1	1	6.16706E-01	39	50	2.39E-06	6.16756E-01	38	52	3.88E-06
4	8	1	2 [a] /1 [b]	6.19708E-01	27	28	4.73E-10	6.15864E-01	22	26	7.45E-06
6	2	1	1	9.71589E+01	57	65	2.16E-06	9.71588E+01	8	9	5.40E-10
7	3	2	1	9.99999E-01	45	47	1.44E-08	9.99992E-01	7	8	2.79E-08

References

Fletcher, R. and Leyffer, S. 2002. Nonlinear programming without a penalty function, *Mathematical Programming* 91: 239–269.

Fletcher, R.; Gould, N. I. M.; Leyffer, S.; Toint, Ph. L. and Wachter, A. 2002. Global convergence of a trust-region SQP-filter algorithm for general nonlinear programming, *SIAM Journal on Optimization* 13: 635–659.

Goberna, M. A. and Lopéz, M. A. (eds.). 2001. Semi-infinite programming. Recent advances, *Nonconvex Optimization and Its Applications* 57, Springer-Verlag.

Gramlich, G.; Hettich, R. and Sachs, E. W. 1995. Local convergence of SQP methods in semi-infinite programming, *SIAM Journal on Optimization* 5: 641–658.

Haaren-Retagne, E. 1992. *A semi-infinite programming algorithm for robot trajectory planning*. Ph.D. Thesis, Universitat Trier.

Hettich, R. and Jongen, H. Th. 1978. Semi-infinite programming: conditions of optimality and applications, *Lecture Notes in Control and Information Science* 7: 1–11, Springer Ed.

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

Liu, G. 2007. A homotopy interior point method for semi-infinite programming problems, *Journal of Global Optimization* 37: 631–646.

Parsopoulos, K.; Plagianakos, V.; Magoulas, G. and Vrahatis, M. 2001. Stretching technique for obtaining global minimizers through particle swarm optimization, in *Proceedings of the Particle Swarm Optimization Workshop,* 22–29.

Price, C. J. and Coope, I. D. 1996. Numerical experiments in semi-infinite programming, *Computational Optimization and Applications* 6: 169–189.

Reemtsen, R. and Rückmann, J.-J. (eds.). 1998. Semi-infinite programming, *Nonconvex Optimization and Its Applications* 25, Kluwer Academic Publishers.

Stein, O. 2003. Bi-level strategies in semi-infinite programming, *Nonconvex Optimization and Its Applications* 71, Kluwer Academic Publishers.

Vaz, A. I. F.; Fernandes, E. M. G. P. and Gomes, M. P. S. F. 2004. Robot trajectory planning with semi-infinite programming, *European Journal of Operational Research* 153: 607–617.

Vaz, A. I. F. and Ferreira, E. C. 2004. Semi-infinite air pollution control problems, in *XXVIII Congreso Nacional de la SEIO*, CdRom, Cádiz. ISBN 84-689-0438-4.

Wächter, A. and Biegler, L. T. 2005. Line search filter methods for nonlinear programming: motivation and global convergence, *SIAM Journal on Optimization* 16: 1–31.

Weber, G.-W. and Tezel, A. 2007. On generalized semi-infinite optimization of genetic network, *TOP* 15: 65–77.