Air pollution control with semi-infinite programming

A. Ismael F. Vaz[1], *Eugénio C. Ferreira*[2]

[1]aivaz@dps.uminho.pt, Departamento de Produção e Sistemas, Escola de Engenharia, Universidade do Minho, Campus de Gualtar, 4710 - 057 Braga, Portugal

[2]ecferreira@deb.uminho.pt, Centro de Engenharia Biológica, Universidade do Minho, Campus de Gualtar, 4710 - 057 Braga, Portugal

Abstract

Air pollution control problems can be formulated as a semi-infinite programming (SIP) problem and we describe three main approaches. The first consists in optimizing an objective function while the pollution level in a given region is kept bellow a given threshold. In the second approach the maximum pollution level in a given region is computed and in the third an air pollution abatement problem is considered. These formulation allow to obtain the best control parameters and the maxima pollution positions, where the sampling stations should be placed.

To illustrate this idea, the (SIP)AMPL modeling language was used to code three academic problems. The SIPAMPL software package includes an interface to connect AMPL to any SIP solver, in particular to the NSIPS solver. Numerical results are shown with the discretization method, implemented in the NSIPS solver and it proved to be efficient in solving the proposed problems.

Keywords: Air pollution control, semi-infinite programming, SIPAMPL database, NSIPS solver.

1. Introduction

Many engineering problems, such as robot trajectory planning, optimal signal sets, production planning, and digital filter design can be posed as semi-infinite programming (SIP) problems (see [6] for many application of SIP). Air pollution control has also deserved some attention in the SIP context [6, 7]. In this paper we describe how air pollution control problems can be formulated as semi-infinite programming problems. Three examples were coded in a modeling language (SIPAMPL [15]) and solved with a general SIP solver (NSIPS [16]), illustrating the potential of these formulations.

Several models for air pollution control problems have been proposed in the last decades (see [10]). These models predict the amount of pollution in a space, where some weather conditions are assumed.

We use a Gaussian model to provide estimates of pollution in a region where mean weather conditions are assumed (see [10]). One of the proposed problem consists of optimizing an objective function (minimum stack height) while the air pollution is kept bellow a given threshold. Other proposed problem consists in computing the maximum air pollution attained in a given region and another is an air pollution abatement problem where reduction in the air pollution emissions is to be minimized while air pollution is kept below a given threshold.

We start in section 2 by describing SIP and the used notation. Section 3 presents the air pollution control problem and Section 4 the three academic examples coded in (SIP)AMPL. Numerical results with the discretization method, available in the NSIPS solver, are shown in Section 5 and we conclude in Section 6.

2. Semi-infinite programming

Semi-infinite programming problems can be described in the following mathematical form

$$\begin{aligned} &\min_{u \in R^n} f(u) \\ s.t.\ & g_i(u,v) \le 0,\ i = 1,\dots,m \\ & u_{lb} \le u \le u_{ub} \\ & \forall v \in \mathcal{R} \subset R^p, \end{aligned} \tag{1}$$

where $f(u)$ is the objective function, $g_i(u,v)$, $i = 1,\dots,m$ are the infinite constraint functions and u_{lb}, u_{ub} are the lower and upper bounds on u.

Problem (1) can be stated in a more general form, by including finite (constraints only depending on u) equality and inequality constraints, but this definition just suit our purpose.

These problems are called semi-infinite programming problems due to the constraints $g_i(u,v) \le 0$, $i = 1,\dots,m$. We can think of $\mathcal{R}$ as an infinite index set and therefore (1) is a problem with finitely many variables over an infinite set of constraints.

Herein, the set $\mathcal{R}$ is assumed to be a cartesian product of intervals ($[\alpha_1,\beta_1] \times \cdots \times [\alpha_p,\beta_p]$).

A natural way to solve the SIP problem (1) is to replace the infinite set $\mathcal{R}$ by a finite one. There are several ways of doing this. Discretization methods, exchange methods, reduction methods (see [6], for a more detailed explanation), dual methods ([14]) and transcribed methods ([12, 13]) are the major classes.

In discretization methods the infinite set $\mathcal{R}$ is replaced by a sequence of subsets $\mathcal{R}_0 \subset \mathcal{R}_1 \subset \cdots \subset \mathcal{R}_\mathcal{N} \subset \mathcal{R}$ (usually the subsets $\mathcal{R}_k$, $k = 0, \ldots, \mathcal{N}$ are grids of points). In each iteration, some points in the subset $\mathcal{R}_k$ are chosen and used in the constraints to form a finite subproblem. The solution to the SIP problem is approximated by the solution on the final subset $\mathcal{R}_\mathcal{N}$, and it may not be a stationary point for SIP.

In exchange methods approximated solutions to the following problems are computed, for a given approximation to the SIP solution $\overline{u} \in R^n$

$$\max_{v \in \mathcal{R}} g_i(\overline{u}, v), \quad i = 1, \ldots, m. \tag{2}$$

The computed approximated solutions are used to obtain a new approximation to the SIP solution (by solving the corresponding finite subproblem) and the process is repeated until a good approximation to the SIP solution is found.

In reduction methods all the global and some local maxima for the problems (2) are obtained. The finite subproblem is then solved with the solutions found to problems (2).

The dual methods solve the SIP problem by considering the dual problem where the infinite number of Lagrange multipliers is represented by a function, which is approximated by a piecewise linear polynomial.

In the constraint transcription methods, the inequality infinite constraints are transcribed to equality finite constraints using integration over the set $\mathcal{R}$.

3. Air pollution control

The reader is pointed to [10] for a background reading in air pollution control.

Considering a coordinate system where the origin is at ground level. The X and Y-axis extends horizontally and are perpendicular to each other. The Z-axis extends vertically perpendicular with the X and Y-axis (see Figure 1). Let a and b be the x and y coordinates, respectively, of the pollution emission point. We assume that the stack pollution emission occurs at some height H above the ground ($z = 0$).

Assuming that the plume spread has a Gaussian distribution, the concentration, $\mathcal{C}$, of gas or aerosols (particles less than about 20 microns diameter) at position x, y, and z from a continuous source with an effective emission

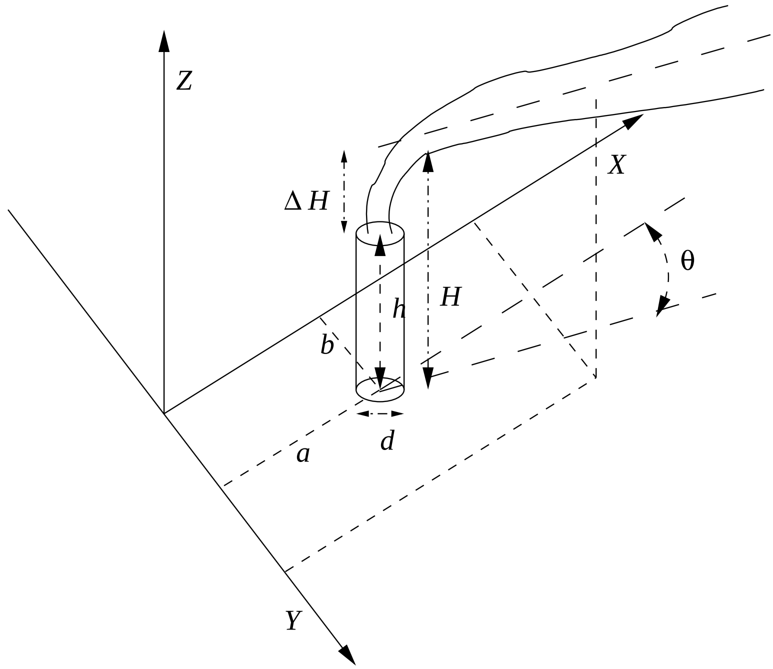

Figure 1: Coordinate system and notation.

height, $\mathcal{H}$, is given by

$$\mathcal{C}(x,y,z,\mathcal{H}) = \frac{\mathcal{Q}}{2\pi\sigma_y\sigma_z\mathcal{U}} e^{-\frac{1}{2}\left(\frac{\mathcal{Y}}{\sigma_y}\right)^2} \left(e^{-\frac{1}{2}\left(\frac{z-\mathcal{H}}{\sigma_z}\right)^2} + e^{-\frac{1}{2}\left(\frac{z+\mathcal{H}}{\sigma_z}\right)^2}\right) \quad (3)$$

where $\mathcal{Q}$ (gs^{-1}) is the uniform emission rate of pollutants, $\mathcal{U}$ (ms^{-1}) is the mean wind speed affecting the plume and σ_y (m) and σ_z (m) are the standard deviations of plume concentration distributed in the horizontal and vertical planes, respectively. $\mathcal{Y}$ is given by

$$\mathcal{Y} = (x-a)\sin(\theta) + (y-b)\cos(\theta), \quad (4)$$

where θ (rad) is the mean wind direction $(0 \leq \theta \leq 2\pi)$. Equation (4) makes a change of coordinates of the pollution emission point in the mean wind direction.

In equation (3) the variable x does not appear explicitly in the formula, but the σ_y and σ_z depend on the $\mathcal{X}$ variable given by

$$\mathcal{X} = (x-a)\cos(\theta) - (y-b)\sin(\theta).$$

The effective emission height is the sum of the physical stack height, h (m), and the plume rise, $\Delta\mathcal{H}$ (m). The plume rise considered here is given by the Holland equation (see [17])

$$\Delta\mathcal{H} = \frac{V_o d}{\mathcal{U}} \left(1.5 + 2.68 \frac{T_o - T_e}{T_o} d\right),$$

where d (m) is the internal stack diameter, V_o (ms^{-1}) is the stack gas exit velocity, T_o (K) is the gas temperature and T_e (K) is the environment temperature.

Assuming that we have n pollution sources distributed in a region, being $\mathcal{C}_i$ the contribution of source i for the total concentration, three major formulations can be derived.

Being the gas chemical inert, the minimization of the stack height $u = (h_1, \ldots, h_n)$, while keeping the pollution level below some threshold $\mathcal{C}_0$ in a given area $\mathcal{R}$, at ground level, can be formulated as the SIP

$$\begin{aligned} &\min_{u \in R^n} \sum_{i=1}^{n} c_i h_i \\ s.t.\ \ & g(u, v \equiv (x,y)) \equiv \sum_{i=1}^{n} \mathcal{C}_i(x, y, 0, \mathcal{H}_i) \leq \mathcal{C}_0 \\ & \forall v \in \mathcal{R} \subset R^2, \end{aligned} \tag{5}$$

where c_i, $i = 1, \ldots, n$, are construction costs associated with the stack height. Note that the objective function must not be a linear function. In fact we can use any nonlinear function of the h_i, $i = 1, \ldots, n$, variables.

Computing the maximum air pollution concentration (l^*) in a given region can be done by solving the following SIP problem.

$$\begin{aligned} &\min_{l \in R} \ l \\ s.t.\ \ & g(v \equiv (x,y)) \equiv \sum_{i=1}^{n} \mathcal{C}_i(x, y, 0, \mathcal{H}_i) \leq l \\ & \forall v \in \mathcal{R} \subset R^2. \end{aligned} \tag{6}$$

The points $v^* \in \mathcal{R}$ where $g(v^*) = l^*$ are global maximizers that make the infinite constraint active and are the positions where the sampling stations should be placed.

Another formulation can be proposed where the minimum production cost (minimum cost with cleaning) is to be obtained while the air pollution is kept bellow a given threshold in a given region. Let $u = (r_1, \ldots, r_n)$ be a percentage of the pollution reduction factor. The problem can be posed as

$$\begin{aligned} &\min_{u \in R^n} \sum_{i=1}^{n} p_i r_i \\ s.t.\ & g(u, v \equiv (x, y)) \equiv \sum_{i=1}^{n} (1 - r_i)\mathcal{C}_i(x, y, 0, \mathcal{H}_i) \leq \mathcal{C}_0 \\ & \forall v \in \mathcal{R} \subset R^2, \end{aligned} \tag{7}$$

where p_i, $i = 1, \ldots, n$, is what one pays for the reduction on source i (cleaning or not producing). Again, the same comments applies to the linear objective function.

We will use these major formulation to solve three academic examples of air control problems.

4. Examples of air pollution control problems

In this section we describe three examples with data collected from the literature on air pollution control.

These problems were coded in the (SIP)AMPL modeling language format and are publicly available in the SIPAMPL problems database. AMPL[1] [2] is a modeling language for mathematical programming problems (other well known modeling language is GAMS [1]). AMPL provides an interface which allows a wide variety of solvers to access problems coded in the AMPL language. Together with the simple and powerful modeling language, AMPL also provides automatic differentiation. Since AMPL is limited to finite programming, SIPAMPL [15] was developed to allow the codification of SIP problems.

SIPAMPL stands for SIP with AMPL. The SIPAMPL package includes a database of more than 160 SIP problems, an interface to allow the connection of any SIP solver to AMPL, an interface that allows MATLAB [8] to use the SIP problems available in the database and a select tool for query the database for SIP problems with specific characteristics.

4.1. Minimal stack height

An air pollution control problem was proposed in [17], to show the reliability of an optimization procedure, in obtaining the global maximum of the sulfur dioxide concentration in a given region. The proposed problem data will be used herein in minimizing the total stack height while the pollutant (sulfur dioxide) is kept bellow a given threshold.

[1] `http://www.ampl.com`

The problem consists of an region with ten stacks. The environment temperature (T_e) is $283K$ and the gas emission temperature is $413K$. Wind speed ($\mathcal{U}$) of $5.64ms^{-1}$ and wind direction (θ) of $3.996rad$ are considered. The stack and emission data for the ten stacks is given in Table 1.

The stack height in Table 1 was used as an initial guess for the SIP formulation and a squared region of $40km$ is considered ($\mathcal{R} = [-20000, 20000] \times [-20000, 20000]$)

Source	a_i (m)	b_i (m)	h_i (m)	d_i (m)	$\mathcal{Q}_i$ (gs^{-1})	$(V_o)_i$ (ms^{-1})
1	-3000	-2500	183	8.0	2882.6	19.245
2	-2600	-300	183	8.0	2882.6	19.245
3	-1100	-1700	160	7.6	2391.3	17.690
4	1000	-2500	160	7.6	2391.3	17.690
5	1000	2200	152.4	6.3	2173.9	23.404
6	2700	1000	152.4	6.3	2173.9	23.404
7	3000	-1600	121.9	4.3	1173.9	27.128
8	-2000	2500	121.9	4.3	1173.9	27.128
9	0	0	91.4	5.0	1304.3	22.293
10	1500	-1600	91.4	5.0	1304.3	22.293

Table 1: Stack and emission data

This problem is coded in the (SIP)AMPL format and is publicly available in the SIPAMPL database (file `vaz1.mod`).

4.2. Maximum attained pollution and sampling stations planning

An example in computing the maximum pollution (l^*) level is achieved by solving the SIP problem (6) with $\mathcal{H}_i$ fixed. Hypothetical source data from [4] is used to illustrate this technique. The source data is shown in Table 2. The region considered was $\mathcal{R} = [0, 24140] \times [0, 24140]$ (a square of approximately 15 miles). The environment air temperature was $284K$, with a wind speed of $5ms^{-1}$ and direction of $3.927rad$ (225^o). The same weather stability as in the maximum stack height example were used.

This problem is also coded in the (SIP)AMPL format and is publicly available in the SIPAMPL database (file `vaz2.mod`).

4.3. Air pollution abatement

In [3] the authors describe an example of policy abatement in air pollution that uses the Sutton equation for the expected pollution concentration. A slightly different problem was used latter in a paper from Van Honstede [7] and is already available in the SIPAMPL database included in the Watson set

Source	a_i (m)	b_i (m)	h_i (m)	d_i (m)	$\mathcal{Q}_i$ (gs^{-1})	$(V_o)_i$ (ms^{-1})	$(T_o)_i$ (K)
1	9190	6300	61.0	2.6	191.1	6.1	600
2	9190	6300	63.6	2.9	47.7	4.8	600
3	9190	6300	30.5	0.9	21.1	29.2	811
4	9190	6300	38.1	1.7	14.2	9.2	727
5	9190	6300	38.1	2.1	7.0	7.0	727
6	9190	6300	21.9	2.0	59.2	4.3	616
7	9190	6300	61.0	2.1	87.2	5.2	616
8	8520	7840	36.6	2.7	25.3	11.9	477
9	8520	7840	36.6	2.0	101.0	16.0	477
10	8520	7840	18.0	2.6	41.6	9.0	727
11	8050	7680	35.7	2.4	222.7	5.7	477
12	8050	7680	45.7	1.9	20.1	2.4	727
13	8050	7680	50.3	1.5	20.1	1.6	727
14	8050	7680	35.1	1.6	20.1	1.5	727
15	8050	7680	34.7	1.5	20.0	1.6	727
16	9190	6300	30.0	2.2	24.7	9.0	727
17	5770	10810	76.3	3.0	67.5	10.7	473
18	5620	9820	82.0	4.4	66.7	12.9	603
19	4600	9500	113.0	5.2	63.7	9.3	546
20	8230	8870	31.0	1.6	6.3	5.0	460
21	8750	5880	50.0	2.2	36.2	7.0	460
22	11240	4560	50.0	2.5	28.8	7.0	460
23	6140	8780	31.0	1.6	8.4	5.0	460
24	14330	6200	42.6	4.6	172.4	13.4	616
25	14330	6200	42.6	3.7	171.3	16.1	616

Table 2: Stack and emission data for the maximum pollution level

of problems (see [9]).

In a certain city there are three plants $\mathcal{P}_1$, $\mathcal{P}_2$ and $\mathcal{P}_3$, emitting the amounts e_1, e_2 and e_3, with $0 \leq e_i \leq 2$, $(i = 1, 2, 3)$ of a certain pollutant. The city ordinance states that the expected pollution must not exceed a standard $\mathcal{C}_0$ under the most common weather conditions, i.e., a steady westerly wind ($\theta = 0$ in the Gaussian model) of constant speed $\mathcal{U}$. The city would also like to know where to place the sampling stations and their number in order to check compliance with the ordinance. Assuming that the revenue is proportional to the emission rate and that the total revenue of the three plants is a linear combination of the emissions, the optimization problem is

$$\begin{aligned} &\max_{e_1,e_2,e_3 \in R} 2e_1 + 4e_2 + e_3 \\ s.t. &\sum_{i=1}^{3} e_i \mathcal{C}(x,y,0,\mathcal{H}_i) \leq \mathcal{C}_0 \\ &0 \leq e_i \leq 2,\ i = 1,2,3 \\ &\forall (x,y) \in [-1,4] \times [-1,4]. \end{aligned}$$

By setting $r_i = 2 - e_i$, $i = 1,2,3$, the previous maximization problem can be rewritten as a minimization problem, yielding

$$\begin{aligned} &\min_{r_1,r_2,r_3 \in R} 2r_1 + 4r_2 + r_3 \\ s.t. &\sum_{i=1}^{3} (1 - r_i) \mathcal{C}(x,y,0,\mathcal{H}_i) \leq \mathcal{C}_0 \\ &0 \leq r_i \leq 2,\ i = 1,2,3 \\ &\forall (x,y) \in [-1,4] \times [-1,4]. \end{aligned}$$

Instead of the Sutton, the Gaussian expression is used to formulate this problem. Using the equivalence between the Sutton ($n = 1$, $C_x = C_y = 1$) and Gaussion expressions (see [3]) we have,

$$\sigma_x = \sigma_y = \begin{cases} \sqrt{\frac{x}{2}} & \text{for } x > 0 \\ 0 & \text{otherwise,} \end{cases}$$

and we consider $\mathcal{C} = 0$ whenever σ_x or σ_y are zero.

A wind speed of $\mathcal{U} = \left(\frac{1}{2\pi}\right)^2 ms^{-1}$, emission rate $\mathcal{Q} = 1gs^{-1}$ and $\mathcal{C}_0 = \frac{1}{2}$ were considered. The effective stack heights and coordinates are given in Table 3 (no plume rise is considered).

Source	a_i	b_i	h_i
1	0	1	1
2	0	0	1
3	2	-1	$\sqrt{2}$

Table 3: Stack data for `vaz3.mod`

The file `vaz3.mod` in the SIPAMPL database refers to this example.

5. Numerical results

The numerical results were obtained on a Pentium III at 450Mhz with 128MB of RAM and a Linux Operating System (Red Hat 5.2) with AMPL Student Version 19991027 (Linux 2.0.18).

The discretization method available in the NSIPS [16] software package is the only one to solve problems with more than one infinite variable and therefore was the selected one. The default options were considered, except for the `method` and `disc_h`. `method` selects the used method and was set to `disc_hett`, which changes the default method to the Hettich version of the discretization method. `disc_h` changes the space (and consequently the number of points used) in the initial grid.

5.1. Minimum stack height

In this example NSIPS was used with the option `disc_h=1000`.

Numerical results are shown in Table 4. Two threshold values for the pollution level and two lower limits on the stack height are considered, originating three different instances of the problem. In the first one a limit of 7.7114×10^{-4} is considered ($\mathcal{C}_0 = 7.7114 \times 10^{-4} gm^{-3}$) while the lower limit on the stack height is zero ($h_i \geq 0$, $i = 1, \dots, n$). Numerical results are shown in the first column of Table 4 and three stack have height equal to zero. Portuguese legislation[2] imposes a minimum stack height of $10m$. The stack height can only be inferior to $10m$ if some legal[3] requirements are met. One way to prove that the requirements are met is by simulation, using a proper model, of the air pollution dispersion. In instance 2 the same limit $\mathcal{C}_0$ is considered while the lower limit on the stack height is $10m$ ($h_i \geq 10$, $i = 1, \dots, n$). Instance 3 considers the Portuguese[4] limit on sulfur dioxide $\mathcal{C}_0 = 1.25 \times 10^{-4} gm^{-3}$.

The constraint contour, at the solution found, are presented in Figure 2. The contour was obtained with the MATLAB interface to SIPAMPL [11].

5.2. Maximum pollution level and sampling stations position

The same grid spacing of the previous formulation was used in the discretization method.

The results found by the discretization method was $l^* = 1.81068 \times 10^{-3} gm^{-3}$. The constraint maximum was attained at $(x, y) = (8500, 7000)$ (the only active point for the constraint in the final grid). While this point is a good position where the sampling station should be placed, other local maxima of the constraint could be considered, as it can be seen in the constraint

[2] Decree law number 352/90 from 9 November 1990.
[3] Decree law number 286/93 from 12 March 1993.
[4] Decree law number 111/2002 from 16 April 2002.

	Instance 1	Instance 2	Instance 3
h_1	0.00	10.00	196.93
h_2	78.26	69.09	380.06
h_3	0.00	10.00	403.12
h_4	153.17	152.64	428.38
h_5	80.90	71.27	344.81
h_6	0.00	10.00	274.58
h_7	13.52	13.52	402.83
h_8	161.78	161.87	396.82
h_9	141.73	141.63	415.58
h_{10}	15.05	15.05	423.99
Total	644.40	655.06	3667.10

Table 4: Numerical results for minimum stack height problem

contour figure.

The contour of the `vaz2` problem constraint is presented in Figure 3.

5.3. Air pollution abatement

The numerical result found by the discretization method is $r^* = (0.987, 0.951, 0.943)$ with the option `disc_h` set to 0.05. The constraint maxima (active constraints) were attained at $(x, y)^1 = (1.100, 0.125)$, $(x, y)^2 = (1.100, 0.100)$ and $(x, y)^3 = (3.675, -0.625)$, where sampling stations should be placed to check the compliance with the ordinance.

The contour of the air pollution abatement problem constraint is presented in Figure 4.

6. Conclusions

Air pollution control problems can be posed as semi-infinite programming problems and efficiently solve by available software. In these problems an objective function is to be optimized while a given threshold for the pollution, in a given region, is to be attained. In the present paper the plume spread was assumed to have a Gaussian distribution under mean weather conditions. The Holland equation was used, in two academic examples, to compute the plume rise.

The formulation of air pollution control problems as SIP allows a great degree of freedom, since new objective and constraints can be easily introduced. The codification of the proposed problems in the (SIP)AMPL modeling language makes them publicly available to the research community, either to test other objectives and/or new constraints, or as SIP test problems.

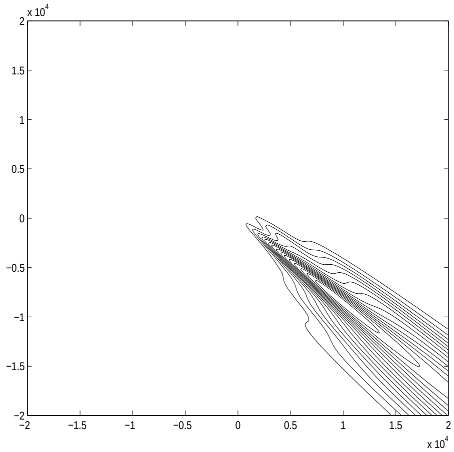

Figure 2: Constraint contour of minimum stack height

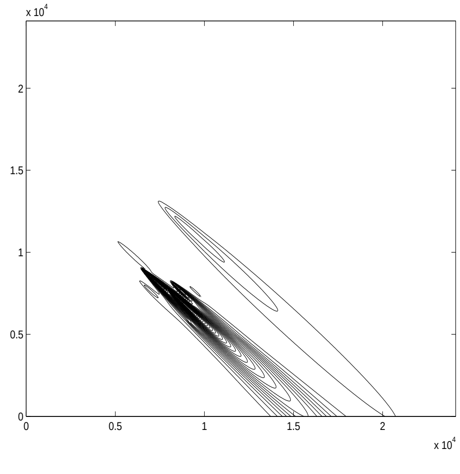

Figure 3: Maximum pollution level contour

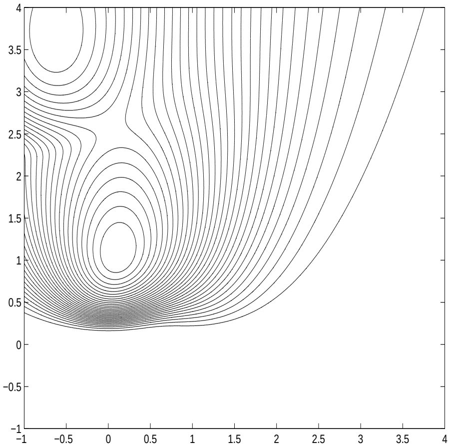

Figure 4: Air pollution abatement contour

In the presented examples, stack and emission data was collected from literature ([3, 17]) to illustrate the proposed approach.

The discretization method implemented in the NSIPS solver was used to solve the coded problems and proved to be efficient.

7. Bibliography

[1] A. Brooke, D. Kendrick, A. Meeraus, and Ramesh Raman. GAMS: A User's Guide. http://www.gams.com/, December 1998.

[2] R. Fourer, D.M. Gay, and B.W. Kernighan. A modeling language for mathematical programming. *Management Science*, 36(5):519–554, 1990.

[3] S.-Å. Gustafson and K.O. Kortanek. Analytical properties of some multiple-source urban diffusion models. *Environment and Planning*, 4:31–41, 1972.

[4] S-Å. Gustafson, K.O. Kortanek, and J.R. Sweigart. Numerical optimization techniques in air quality modeling: Objective interpolation formulas for the spatial distribution of pollutant concentration. *Applied Meteorology*, 16(12), December 1977.

[5] R. Hettich, editor. *Semi-Infinite Programming*. Springer-Verlag, 1979.

[6] R. Hettich and K.O. Kortanek. Semi-infinite programming: Theory, methods, and applications. *SIAM Review*, 35(3):380–429, 1993.

[7] W. Van Honstede. An approximation method for semi-infinite problems. In [5], pages 127–136, 1979.

[8] MathWorks. *MATLAB*. The MathWorks Inc., 1999. Version 5.4, Release 11.

[9] C.J. Price. *Non-Linear Semi-Infinite Programming*. PhD thesis, University of Canterbury, New Zealand, August 1992.

[10] D.B. Turner. *Workbook of Atmospheric Dispersion Estimates*. Lewis publishers, Boca Raton, second edition, 1994.

[11] A.I.F. Vaz and E.M.G.P. Fernandes. An interface between MATLAB and SIPAMPL for semi-infinite programming problems. *submitted for publication*, 2004.

[12] A.I.F. Vaz, E.M.G.P. Fernandes, and M.P.S.F. Gomes. Penalty function algorithms for semi-infinite programming. *submitted for publication*, 2003.

[13] A.I.F. Vaz, E.M.G.P. Fernandes, and M.P.S.F. Gomes. A quasi-Newton interior point method for semi-infinite programming. *Optimization Methods and Software*, 18(6):673–687, 2003.

[14] A.I.F. Vaz, E.M.G.P. Fernandes, and M.P.S.F. Gomes. A sequential quadratic method with a dual parametrization approach to nonlinear semi-infinite programming. *TOP*, 11(1):109–130, 2003.

[15] A.I.F. Vaz, E.M.G.P. Fernandes, and M.P.S.F. Gomes. SIPAMPL: Semi-infinite programming with AMPL. *ACM Transactions on Mathematical Software*, 30(1):47–61, March 2004.

[16] A.I.F. Vaz, E.M.G.P. Fernandes, and M.P.S.F. Gomes. NSIPS v2.1: Nonlinear Semi-Infinite Programming Solver. Technical Report ALG/EF/5-2002, Universidade do Minho, Braga, Portugal, December 2002. http://www.norg.uminho.pt/aivaz/.

[17] B.-C. Wang and R. Luus. Reliability of optimization procedures for obtaining global optimum. *AIChE Journal*, 24(4):619–626, 1978.