ISEIS

Journal of Environmental Informatics 14(1) 31-40 (2009)

Journal of Environmental Informatics

www.iseis.org/jei

Estimation of Spatial Influence Models Using Mixed-Integer Programming

A. Billionnet

ENSIIE, 1 square de la résistance, Evry cedex 91025, France

Received 10 June 2008; revised 1 April 2009; accepted 7 May 2009; published online 10 September 2009

ABSTRACT. Estimation of ecosystem models is an important task and many studies have been carried out on the problem. However, estimating some models may be difficult. Here, we want to estimate two nonlinear spatial influence models by using the classical least-squares method, and that requires the solution of difficult nonlinear optimization problems. The aim of this paper is to show that both models can be efficiently estimated using mathematical programming. The estimation problems are first formulated as nonlinear optimization problems which are then transformed into convex quadratic mixed-integer programs. The transformation is based on the discretization of some variables and on the linearization of the product of a Boolean variable with a real variable. The approach which allows to find the best estimation with a certain precision and in the least-squares framework, is interesting for several reasons: the definition of the mathematical programs is relatively simple, they are easy to implement using a mathematical programming language together with a quadratic mixed-integer programming software, and computational experiments carried out on large sets of simulated data show the excellent performance of the approach. Moreover, the ideas underlying the method can be used for other difficult least-squares estimations. These results suggest that mixed-integer programming may be an efficient tool for practitioners and researchers in environmental modeling.

Keywords: spatial influence model, nonlinear least-squares estimation, mixed-integer programming, computational experiments

1. Introduction

Because of the importance of model estimation in ecosystem modeling, numerous researchers have worked for many years to develop estimation methods or apply known methods in a real environment. However, estimation of complicated models in natural resources and environmental systems continues to challenge ecological modelers. The aim of this paper is to propose an efficient method for the estimation of two general spatial influence models suggested in (Kuuluvainen and Linkosalo, 1998). We consider a plane surface and objects situated on this surface. Each of these objects is of a certain type. For a given environmental variable, the objects have an influence which can be measured over some points of the surface (Figure 1). We look for models describing this influence with the hypothesis that the spatial influence at any given point of the surface depends on the combined individual influences of the multiple surrounding objects. More precisely, we consider that the influence at a point depends on the distances between this point and the objects which are in the influence vicinity from this point, and also on the different types of these objects. Since a simple linear model does not appear to adequately describe these spatial influences, we consider here a nonlinear model including expressions of the form d^{α} where d is the distance between an object and a measure point, and α is a parameter to be estimated, and which is dependent on the type of the object.

In this paper, we address the estimation problem by the least-squares method. Classically, we want to estimate model parameters in order to minimize the sum of the squares of the gaps between the values predicted by the model and the observed values. Least-squares is a widely used technique in many domains. When the parameters appear linearly in these expressions then the least-squares estimation can be solved in closed form but for nonlinear models the problem often becomes a complicated nonlinear optimization problem. Usually, these optimization problems are solved by heuristic methods or iterative optimization techniques (Kuuluvainen and Linkosalo, 1998). Heuristic approaches such as Genetic Algorithm, Simulated Annealing or Tabu Search (Glover and Kochenberger, 2003) are generally fast and often provide good solutions but they have significant drawbacks: they provide approximate solutions which cannot be tested for optimality, it is often difficult to tell how far these solutions are from optimality, and they require difficult adjustments of several parameters. Iterative optimization techniques (Hooke and Jeeves, 1961) may require several hours of computation time and do not always give a global optimum. In this work, we formulate the estimation problem as a nonlinear mathematical program in real variables that we then transform into a program with integer and real variables, a convex quadratic objective, and linear constraints. The transformation is based on the discretization of some va-

* Corresponding author. Tel.: +33 1 69367333; fax: +33 1 69367305.
E-mail address: Alain.Billionnet@ensiie.fr (A. Billionnet).

ISSN: 1726-2135 print/1684-8799 online
 doi:10.3808/jei.200900151

riables and on the linearization of the product of a Boolean variable with a real variable. The discretization of continuous variables is a common approach to deal with nonlinear expressions including these variables, and the linearization technique is inspired from (Glover, 1975). The transformed problem can be solved by standard mixed-integer quadratic programming (MIQP) solvers. See Appendix for further information about mixed-integer programming. The mathematical programming approach presented in this paper allows the estimation problem to be solved to optimality with a fixed precision, and is very different from other methods proposed for least-squares estimations in the case of nonlinear models (e.g. Hooke and Jeeves, 1961; Gill and Murray, 1978; Gill and Wright, 1986). Computational experiments carried out with a large set of simulated data show the excellent performance of the approach since it allows to quickly find the estimated influence models for a surface including 5000 objects and 500 points of measure. Note that this article only proposes an efficient technique for least-squares estimation of influence models. It does not concern the precise analysis of the results, i.e. the statistical comparison of the values predicted by the obtained models and the measured values.

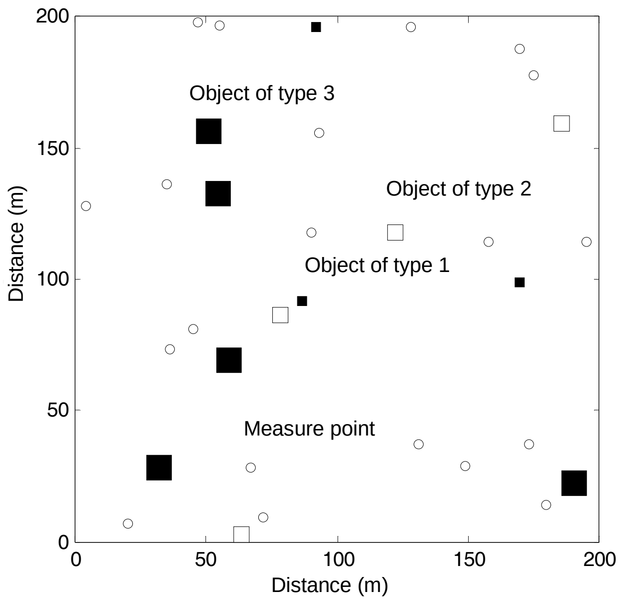

Figure 1. An example of a 200 m × 200 m surface with 3 objects of type 1, 4 objects of type 2, 5 objects of type 3, and 20 measure points.

Section 2 describes the two spatial influence models that we consider. In Model 1, the objects are grouped into classes, and in Model 2, the objects have a certain measurable characteristic that varies continuously within a given interval. Section 3 proposes a convex quadratic programming formulation of the estimation of Model 1, and in Section 4, we report computational tests concerning this estimation problem. Section 5 proposes a convex quadratic programming formulation of the estimation of Model 2, and in Section 6 we report computational tests concerning this second estimation problem. Section 7 is a conclusion.

2. The Models

We consider a plane surface S and objects $O_1, ..., O_n$ situated on this surface. Each of these n objects has a certain characteristic. For a given environmental variable, the objects have an influence which can be measured over some points $P_1, P_2, ..., P_m$ of the surface. We want to estimate two models describing this influence.

2.1. The First Model

For all points P_j of the surface S, we try to express the influence F_j on this point of the n objects O_i $(i = 1, ..., n)$ situated on this surface. Following (Kuuluvainen and Linkosalo, 1998) we suppose that the influence at P_j depends on the distance between the objects and P_j, and that a reasonable (nonlinear) model is of the form:

$$F_j = \sum_{i=1}^{n} f_{t(i)}(d_{ij}) \quad (j = 1, ..., m) \tag{1}$$

where d_{ij} $(i = 1, ..., n; j = 1, ..., m)$ is the distance that separates the point P_j from the object O_i. For the object O_i, $t(i)$ is the type of this object and there are p types. For $k = 1, ..., p$, the functions $f_k(d_{ij})$ are of the form $f_k(d_{ij}) = a_k + b_k d_{ij}^{c_k}$ if $d_{ij} \le d_k^{\max}$ and $f_k(d_{ij}) = 0$ if $d_{ij} > d_k^{\max}$, where a_k, b_k and c_k are the parameters to be estimated. So, an object of type k has an influence over a point P_j if the distance between this object and P_j is less than or equal to $d_k^{\max}$. We suppose that for all the points P_j of the surface, we have a measure M_j of the global influence of all the objects over this point. We want to estimate by the least-squares method the values of the parameters a_k, b_k and c_k (k = 1, ..., p) for which the model best fits the data, i.e. to determine the values of a_k, b_k and c_k $(k = 1, ..., p)$ that minimize the sum of the squared differences between the predicted values F_j and the measured values $\bar{F}_j$: $\sum_{j=1}^{m} \left(F_j - \bar{F}_j\right)^2$.

2.2. The Second Model

As suggested in (Kuuluvainen and Linkosalo, 1998) rather than to distribute objects in some groups, in this second model we consider that the individual influence functions depend in a continuous way on a certain parameter characterizing the objects. As in the first model, we suppose that the influence F_j at P_j depends on the distance between the objects and P_j, and that it can be expressed by the following model:

$$F_j = \sum_{i=1}^{n} f_i(d_{ij}) \quad (j = 1, ..., m) \tag{2}$$

where $f_i(d_{ij}) = a \cdot s_1(D_i) + b \cdot s_2(D_i) \cdot d_{ij}^{c/s_3(D_i)}$ if $d_{ij} \le d_i^{\max}$, and $f_i(d_{ij}) = 0$ if $d_{ij} > d_i^{\max}$; $s_k(D_i)$ $(k = 1, 2, 3)$ is a known function of D_i, the value of the parameter characterizing the object O_i. As in the first model, object O_i has an influence over the point P_j if the distance between this object and P_j is less than or equal to $d_i^{\max}$. In this second model, we have to estimate the three parameters: a, b and c.

2.3. Interpretation of the Two Models

To build the models, we suppose that the influence of an object O_i over a point P_j depends on the object type (Model 1) or of a continuous parameter characterizing this object (Model 2), and also of the distance between the object and the point. In both models this influence is a nonlinear decreasing function of the distance. In Model 1, the influence is a function of the form $f_k(d_{ij})$ as defined in Section 2.1. An example of such influence functions is given in Figure 2 where $\overline{a} = -(0.6, 1.1, 2.1, 3.6, 5.6)$, $\overline{b} = (1.6, 2.6, 4.1, 6.1, 8.6)$, $\overline{c} = -(0.42, 0.28, 0.19, 0.14, 0.1)$, and $d^{\max} = (10, 20, 30, 40, 50)$.

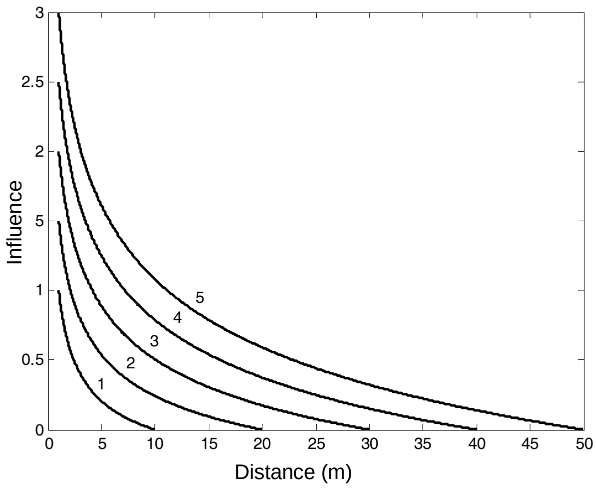

Figure 2. Model 1: example of influence functions for five types of objects.

In Model 2, the influence is also a nonlinear decreasing function of the distance; it is expressed by a function of the form $f_i(d_{ij})$ as defined in Section 2.2. An example of such influence functions is given in Figure 3 where the parameter D_i ranges from 10 to 50, $s_1(D_i) = D_i^{1.32}$, $s_2(D_i) = s_3(D_i) = D_i$, $a = -0.029$, $b = 0.15$, $c = -5.12$. The influence functions for objects such that D_i = 10, 15, 20, 25, 30, 35, 40, 45, 50 are presented on the figure.

In both models there is the following additivity hypothesis: the cumulated influence F_j, over the point P_j of the surface S, of all the objects situated on this surface is equal to the sum of the influences of each object on this point, i.e. $F_j = \sum_{i=1}^{n} f_{t(i)}(d_{ij})$ in Model 1 and $F_j = \sum_{i=1}^{n} f_i(d_{ij})$ in Model 2. To summarize, the data are:

n: number of objects;
m: number of measure points;
p: number of types of objects, for Model 1;
$t(i)$: type of the object O_i $(i = 1, ..., n)$, for Model 1;
D_i : characteristic of the object O_i $(i = 1, ..., n)$, for Model 2;
(xo_i, yo_i) : coordinates of the object O_i $(i = 1, ..., n)$;
(xp_j, yp_j) : coordinates of the measure point P_j $(j = 1, ..., m)$.

We deduct from previous data the distance from the object O_i to the measure point P_j:
$d_{ij} = \sqrt{(xo_i - xp_j)^2 + (yo_i - yp_j)^2}$ $(i = 1, ..., n; j = 1, ..., m)$;
$d_k^{\max}$: maximal influence distance of an object of type k $(k = 1, ..., p)$, for Model 1;
d_i : maximal influence distance of the object O_i $(i = 1, ..., n)$, for Model 2;
$\overline{F}_j$: result of the measure at point P_j $(j = 1, ..., m)$;
and the unknowns to be determined in order to minimize the quantity $\sum_{j=1}^{m} \left(F_j - \overline{F}_j\right)^2$ are: a_k, b_k, c_k $(k = 1, ..., p)$ for Model 1; a, b, c, for Model 2.

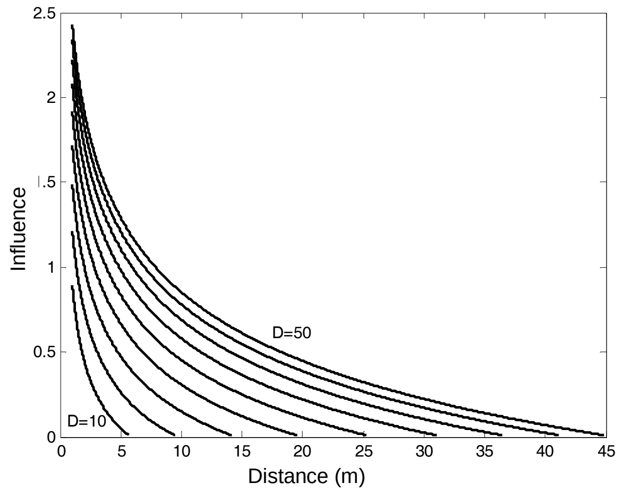

Figure 3. Model 2: example of influence functions, depending in a continuous way on a parameter Di associated with the object Oi.

In Kuuluvainen and Linkosalo (1998), the authors study the effects of tree neighborhoods on the values of environmental variables. For example, they try to determine if the humus layer thickness in a point of a forest depends on trees situated around this point, and how to express this dependence. More exactly, they try to know if the humus layer thickness in a point can be expressed by a simple function of the diameter of the trees situated at a small enough distance of this point, and of the distance between these trees and this point. In their study, trees are grouped together in p various classes following their diameter. Their problem is the same that the one presented in Section 2.1: the plane surface is a forest, each object O_i is a tree, $t(i)$ is the class to which the tree O_i belongs, M_j is the humus layer thickness measured in the point P_j of the forest, and if the distance between the tree O_i and the point P_j is greater than $d_{t(i)}^{\max}$, then the tree O_i has not influence on the point P_j. They suppose that the individual influence of a tree over the humus layer thickness in a point can be expressed by a function of the form $f_k(d_{ij}) = a_k + b_k d_{ij}^{c_k}$ as defined in Section 2.1. They also make the additivity hypothesis of the individual influences for defining the global influence. The model 2 which is slightly different from Model 1 is suggested in (Kuuluvainen and Linkosalo, 1998). Indeed, in their conclusion, these authors indicates as future research the development of a method to estimate tree influences as a continuous function of distance and tree size.

3. Estimating Model 1 by Convex Quadratic Programming

Among all the difficult optimization problems, the minimization of a linear or convex quadratic function subject to linear constraints is particularly well solved by standard software. Variables are real, but some of them must take only integer values. (See Appendix for a brief presentation of mixed-integer programming). Estimating the first spatial influence model consists in solving the mathematical program (3):

$$\min \sum_{j=1}^{m} [\sum_{i \in I_j} (a_{t(i)} + b_{t(i)} d_{ij}^{c_{t(i)}}) - \bar{F}_j]^2 \quad (3)$$

$$\text{s.t. } a_k, b_k, c_k \in R \ (k = 1, \ldots, p) \quad (3.1)$$

$$b_k \geq 0 \ (k = 1, \ldots, p) \quad (3.2)$$

where I_j is the set of objects having an influence over the point P_j, i.e. $I_j = \{i \in \{1,...,n\} : d_{ij} \leq d_{t(i)}^{\max}\}$ $(j = 1, \ldots, n)$. Program (3) is a difficult nonlinear programming problem because the objective function is a sum of squared expressions where the nonlinear term $b_{t(i)} d_{ij}^{c_{t(i)}}$ appears. By discretizing the variable c_k, we are going to reformulate (3) as a program with a convex quadratic function and linear constraints. For that, let $[c_k^{\min}, c_k^{\max}]$ be the interval in which the optimal value of c_k falls, and let us choose a precision δ_k for the variable c_k $(k = 1, \ldots, p)$. We can then express the variable c_k in the following way: $c_k = c_k^{\min} + \delta_k \sum_{r=0}^{n_k} r \cdot w_{kr}$ where $n_k = \lceil (c_k^{\max} - c_k^{\min}) / \delta_k \rceil$ and w_{kr} $(k = 1, \ldots, p;$ $r = 0, \ldots, n_k)$ are Boolean variables subject to constraints $\sum_{r=0}^{n_k} w_{kr} = 1$ $(k = 1, \ldots, p)$. So, the possible values of c_k are $c_k^{\min}$, $c_k^{\min} + \delta_k$, $c_k^{\min} + 2\delta_k$, …, $c_k^{\min} + n_k\delta_k$, and $c_k^{\min} + n_k\delta_k$ is greater than or equal to $c_k^{\max}$. For example, if $c_k^{\min} = -1$, $c_k^{\max} = 2$ and $\delta_k = 0.1$, we will look for a value of c_k belonging to the following finite set of values: $\{-1, -0.9, -0.8, \ldots, 1.8, 1.9, 2\}$. In this case, we look for a real value of the coefficient c_k, belonging to the interval $[-1, 2]$, and with one digit after the decimal point. Discretizing c_k in (3) leads to the program (4).

$$\min \sum_{j=1}^{m} \left(F_j - \bar{F}_j\right)^2 \quad (4)$$

$$\text{s.t. } F_j = \sum_{i \in I_j} [a_{t(i)} + b_{t(i)} \sum_{r=0}^{n_{t(i)}} w_{t(i),r} \cdot d_{ij}^{c_{t(i)}^{\min} + r\delta_{t(i)}}] \ (j = 1, \ldots, m) \quad (4.1)$$

$$c_k = c_k^{\min} + \delta_k \sum_{r=0}^{n_k} r \cdot w_{kr} \ (k = 1, \ldots, p) \quad (4.2)$$

$$\sum_{r=0}^{n_k} w_{kr} = 1 \ (k = 1, \ldots, p) \quad (4.3)$$

$$a_k, b_k \in R,\ b_k \geq 0,\ w_{kr} \in \{0,1\};\ k = 1, \ldots, p;\ r = 0, \ldots, n_k \quad (4.4)$$

The optimal solution of (4) gives the best spatial influence model (with the least-squares criterion) when the parameter c_k is allowed to take only values belonging to the finite set $\{c_k^{\min}$, $c_k^{\min} + \delta_k$, $c_k^{\min} + 2\delta_k$, ..., $c_k^{\min} + n_k\delta_k\}$ $(k = 1, \ldots, p)$. Now, the objective function of (4) is obviously quadratic and convex but constraints (4.1) are nonlinear because of the products $b_{t(i)} w_{t(i),r}$ where $b_{t(i)}$ is a nonnegative real variable and $w_{t(i),r}$ is a Boolean variable. Let us put $k = t(i)$ and linearize these products in a classical way: replace $b_k w_{kr}$ by the real variable v_{kr} and add the linear constraints $C(v_{kr}, b_k, w_{kr}, b_k^{\max})$ forcing v_{kr} to be equal to $b_k w_{kr}$.

$$C(v_{kr}, b_k, w_{kr}, b_k^{\max}) \ (k = 1, \ldots, p;\ r = 0, \ldots, n_k): \quad (5)$$

$$v_{kr} \leq b_k^{\max} w_{kr}$$

$$v_{kr} \leq b_k$$

$$v_{kr} \geq b_k - b_k^{\max}(1 - w_{kr})$$

$$v_{kr} \geq 0$$

In (5), $b_k^{\max}$ is the maximal possible value of b_k $(k = 1, \ldots,$ $p)$. For each couple (k, r) belonging to $\{1, \ldots, p\} \times \{0, \ldots, n_k\}$, $C(v_{kr}, b_k, w_{kr}, b_k^{\max})$ corresponds to four linear inequalities. By examining successively the two possible values of w_{kr}, we immediately see that $v_{kr} = b_k w_{kr}$ if and only if the four constraints of $C(v_{kr}, b_k, w_{kr}, b_k^{\max})$ are satisfied. By using this linearization of $b_{t(i)} w_{t(i),r}$ in (4), we get the mixed-integer program (6), equivalent to (4). The objective function of (6) is quadratic and convex and now all the constraints are linear. The program (6) can therefore be solved by a standard mixed-integer quadratic programming solver.

$$\min \sum_{j=1}^{m} \left(F_j - \bar{F}_j\right)^2 \quad (6)$$

$$\text{s.t. } F_j = \sum_{i \in I_j} [a_{t(i)} + \sum_{r=0}^{n_{t(i)}} v_{t(i),r} \cdot d_{ij}^{c_{t(i)}^{\min} + r\delta_{t(i)}}] \ (j = 1, \ldots, m) \quad (6.1)$$

$$c_k = c_k^{\min} + \delta_k \sum_{r=0}^{n_k} r \cdot w_{kr} \ (k = 1, \ldots, p) \quad (6.2)$$

$$\sum_{r=0}^{n_k} w_{kr} = 1 \ (k = 1, \ldots, p) \quad (6.3)$$

$$C(v_{kr}, b_k, w_{kr}, b_k^{\max}) \ (k = 1, \ldots, p;\ r = 0, \ldots, n_k) \quad (6.4)$$

$$a_k, b_k \in R,\ b_k \geq 0,\ w_{kr} \in \{0,1\} \ (k = 1, \ldots, p;\ r = 0, \ldots, n_k) \quad (6.5)$$

To sum up, an optimal solution $(\tilde{a}_k, \tilde{b}_k, \tilde{c}_k)$ $(k = 1, \ldots, p)$ of (6) provides the best estimation of Model 1 when the value of the parameter c_k is restricted to the following ones: $c_k^{\min}$, $c_k^{\min} + \delta_k$, $c_k^{\min} + 2\delta_k$, …, $c_k^{\min} + n_k\delta_k$ $(k = 1, \ldots, p)$.

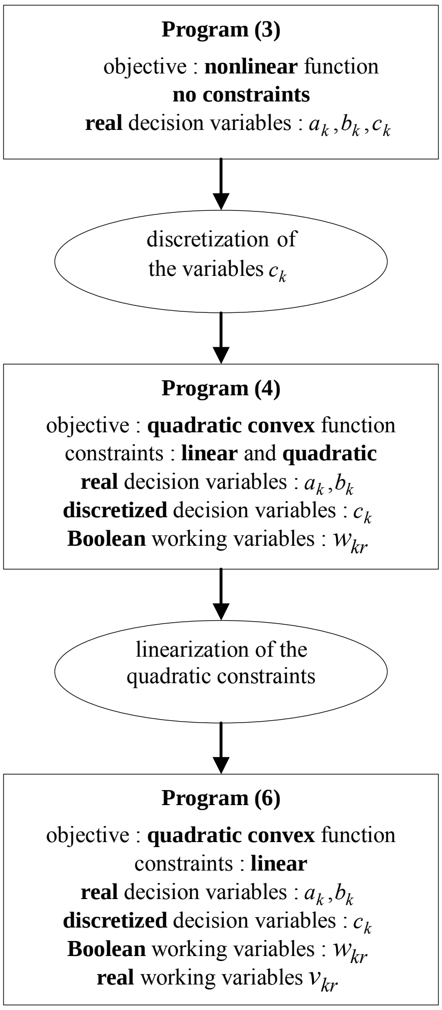

Figure 4. The different steps to formulate the estimation of Model 1 by a mathematical program with a quadratic convex objective, linear constraints and real or 0/1 variables.

4. Experimental Estimation of Model 1 Parameters

Two simulated surfaces were generated to test the approach, i.e. the possibility of effectively solving the convex quadratic mixed-integer programs (6) associated with the estimation problem. In this section, we report computational results on these surfaces, which show the efficiency of the approach. The different instances of the mathematical program (6) have been implemented using the modeling language AMPL (Fourer et al., 1993) and solved by the MIQP solver CPLEX 10.2.0 (CPLEX, 2007). The experiments have been carried out on a PC with an Intel Core Duo 2 GHz processor and 2 Go of RAM. The solution of (6) has been tested in two simulated surfaces whose the characteristics are given below.

Surface 1 − A rectangular region of dimension 1000 m ×500 m including 5000 objects of five types, and 500 measure points ($n = 5000$, $p = 5$, $m = 500$). The coordinates of the objects and measure points, xo_i, yo_i, xp_j, yp_j, are real numbers uniformly and randomly generated between 0 and 1000 for xo_i and xp_j , and between 0 and 500 for yo_i and yp_j . For each object, its type is an integer uniformly and randomly generated between 1 and 5. In this simulated surface, the number of objects of types 1, 2, 3, 4 and 5 are 609, 1269, 1225, 1256 and 641, respectively.

Surface 2 − A rectangular region of dimension 200 m× 100 m including 5000 objects of five types, and 500 measure points ($n = 5000$, $p = 5$, $m = 500$). The coordinates of the objects and measure points, xo_i, yo_i, xp_j, yp_j , are real numbers uniformly and randomly generated between 0 and 200 for xo_i and xp_j , and between 0 and 100 for yo_i and yp_j . For each object, its type is an integer uniformly and randomly generated between 1 and 5. In this simulated surface, the number of objects of types 1, 2, 3, 4 and 5 are the same as in Surface 1.

In order to test the approach proposed to estimate the parameters of Model 1, we have carried out the two tests described below.

4.1. Test 1

The following (known) individual test influence functions are used to simulate in Surface 1 the values of a hypothetical object influence at the 500 points of the surface: $f_k(d_{ij}) = \overline{a}_k + \overline{b}_k d_{ij}^{\overline{c}_k}$ if $d_{ij} \le d_k^{\max}$ and $f_k(d_{ij}) = 0$ if $d_{ij} > d_k^{\max}$, with $\overline{a}$ = −(0.6, 1.1, 2.1, 3.6, 5.6), $\overline{b} = (1.6, 2.6, 4.1, 6.1, 8.6)$, $\overline{c}$ = −(0.42, 0.28, 0.19, 0.14, 0.1), and $d^{\max} = (10, 20, 30, 40, 50)$. These influence functions are represented in Figure 2. In this instance, the values of the influence at point P_j , $\overline{F}_j = \sum_{i=1}^{n} f_{t(i)}(d_{ij})$, vary from 2.97 to 20.34. The minimum number of objects that influence a measure point is equal to 11, the average number of objects that influence a measure point is equal to 31.3, and the maximum number of objects that influence a measure point is equal to 54. In the solution of (6), the minimal and maximal values of b_k , c_k and d_k are fixed as follows: $b_k^{\min} = 0$, $b_k^{\max} = 100$, $c_k^{\min} = -3$, $c_k^{\max} = -0.1$ ($k = 1, ..., p$) .
In this test, the optimal value of (6) with $\delta_k = 0.01$ (k = 1, …, p) is equal to 0 as expected, and the corresponding so- lution provides exactly the coefficients chosen to test the ap- proach, i.e. $a = -(0.6, 1.1, 2.1, 3.6, 5.6)$, $b = (1.6, 2.6, 4.1, 6.1, 8.6)$, and $c = -(0.42, 0.28, 0.19, 0.14, 0.1)$. The solution re- quires 92 seconds of CPU time and a search tree composed of 14 nodes. With $\delta_k = 0.1$ ($k = 1, ..., p$), the optimal value of (6) is equal to 0.2957 and the corresponding solution is a = −(0.652, 0.993, 1.964, 5.296, 5.599), b = (1.650, 2.505, 3.977, 7.706, 8.598) and c = −(0.4, 0.3, 0.2, 0.1, 0.1). In this case the so- lution of (6) requires only 4 seconds of CPU time but the search tree is composed of 25 nodes. In this test, the precision δ_k = 0.1 appears to be sufficient to estimate the model since the sum of the 500 squared differences is only 0.2957. Now, con- sider the following random perturbation of $\overline{F}_j : \overline{F}_j = \varepsilon_j \sum_{i}^{n} f_{t(i)}(d_{ij})$, ε_j being uniformly and randomly generated in the interval [0.8, 1.2]. With these values for the parameters, $\overline{F}_j$ varies from 2.67 to 22.33. The computational results obtained for different values of δ_k are summarized in Table 1.

With $\delta_k = 0.1$, the solution of (6) provides a solution of the estimation problem with one decimal for the value of the pa-

Table 1. Computational Results Regarding the Solution of (6) in the Test 1 with Different Values of δ_k

δk (k = 1, …, p)	Optimal value of (6)	CPU time in seconds	# nodes	a	b	c
				-0.11429	0.69538	-1.3
				-1.09745	2.62586	-0.3
0.1	862.23	3.4	46	-0.83938	3.44949	-0.4
				-2.21894	4.77903	-0.2
				-5.41726	8.36779	-0.1
				-0.12198	0.70734	-1.25
				-1.09686	2.62378	-0.30
0.05	861.79	6.3	50	-1.04708	3.56280	-0.35
				-3.16585	5.60733	-0.15
				-5.43106	8.38810	-0.10
				-0.11617	0.70304	-1.28
				-1.09416	2.61969	-0.30
0.01	861.67	248.0	747	-1.00158	3.53737	-0.36
				-2.93124	5.39637	-0.16
				-3.98081	7.02920	-0.13

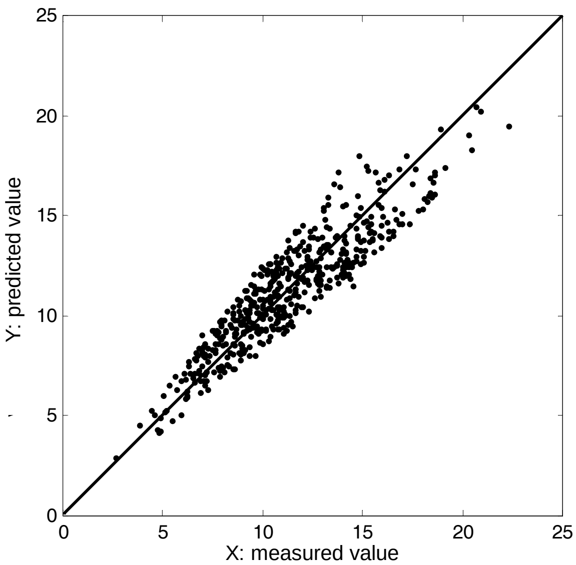

Figure 5. Model 1, Test 1: for each of the 500 measure points of Surface 1, the simulated measured value and the value predicted by the model are not far from each other.

rameter c_k. With $\delta_k = 0.01$, we get two decimals for this value. The optimal solution of (6) is quickly obtained for $\delta_k = 0.1$ and $\delta_k = 0.05$. The solution of (6) with $\delta_k = 0.01$ requires much more time. We see in Table 1 that increasing the precision does not improve significantly the optimal value, i.e. the sum of the squared differences. Thus, $\delta_k = 0.1$ appears to be sufficient to estimate the model. Figure 5 illustrates the comparison between the values predicted by the model and the (simulated) measured values, when $\delta_k = 0.1$. We would obtain almost the same figure for $\delta_k = 0.01$. We also see in Table 1 that though the optimal values are very close for the three considered precisions, the estimated parameters are relatively different. For example, for objects of type 5, the estimated values of the three parameters are $a_5 = -5.41726$, $b_5 = 8.36779$ and $c_5 = -0.1$ when $\delta_k = 0.1$, and $a_5 = -3.98081$, $b_5 = 7.02920$ and $c_5 = -0.13$ when $\delta_k = 0.01$. The two corresponding functions are represented in Figure 6 where we see that these different parameters define almost identical influence functions.

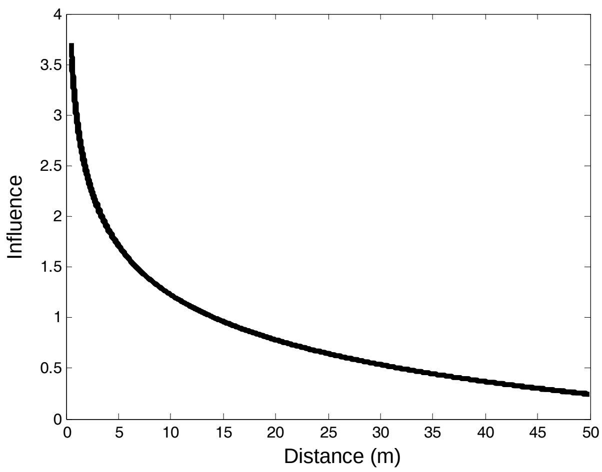

Figure 6. Model 1, Test 1: two almost identical individual influence functions for the objects of type 5.

4.2. Test 2

We consider Surface 2 where objects are trees, and the following individual influence functions proposed in (Kuuluvainen and Pukkala, 1989) to simulate values of a hypothetical tree influence at the 500 points of the surface: $g_k(d_{ij}) = g_k^0 e^{-b_k d_{ij}^2}$. These functions are based on an empirical analysis of the effect of Scots pine seed trees on the density of seedlings and understory vegetation on a certain site. We consider that the 5000 trees of the surface fall into five diameter size classes, 0 ~ 10, 10 ~ 20, 20 ~ 30, 30 ~ 40 and 40 ~ 50 cm. There are thus five types of objects/trees. Moreover, we assume that the tree heights of each class are 5, 10, 15, 20, 30 m. Following (Kuuluvainen and Linkosalo, 1998), $g_k^0 = D_k / 35$ where D_k is

Table 2. Computational Results Regarding the Solution of (6) in the Test 2 with Different Values of δ_k

δ_k (k=1,…,p)	Optimal value of (6)	CPU time in seconds	# nodes	a	b	c
				-0.20808	0.30713	-0.4
				-1.79367	2.09220	-0.1
0.1	51.71	3.8	4	-3.04724	3.59506	-0.1
				-4.09745	4.91254	-0.1
				-5.22673	6.40614	-0.1
				-0.25790	0.35992	-0.35
				-3.78862	4.08826	-0.05
0.05	47.55	6.3	1	-6.39257	6.94054	-0.05
				-8.59500	9.40830	-0.05
				-10.9524	12.1214	-0.05
				-0.36094	0.46493	-0.27
				-19.7116	20.0117	-0.01
0.01	45.26	292.9	38	-33.1201	33.6675	-0.01
				-44.5294	45.3401	-0.01
				-56.6930	57.8522	-0.01

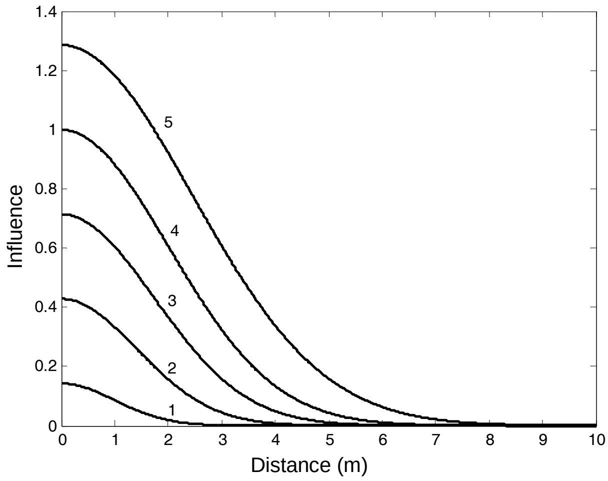

Figure 7. Model 1, Test 2: the five individual influence functions.

the middle of the class k, and $b_k = 1/(0.4h_k)$ where h_k is the height of the trees in class k. Thus $g^0 = (5, 15, 25, 35, 45)/35$ and $b = (1/2, 1/4, 1/6, 1/8, 1/12)$. These individual influence functions are represented in Figure 7 where the tree class is indicated next to each curve. We consider that $d^{\max} = (3, 5, 6, 7, 9)$. In this test, the global influence at poin P_j, $\overline{F}_j = \sum_{i=1}^{n} f_{t(i)}(d_{ij})$, varies from 0.44 to 8.64, the minimum, average and maximal number of objects that influence a measure point is equal to10, 29.1 and 50, respectively.

In the solution of (6), the minimal and maximal values of b_k, c_k and d_k are fixed as follows: $b_k^{\min} = 0$, $b_k^{\max} = 100$, $c_k^{\min} = -3$, $c_k^{\max} = 0$ $(k = 1, ..., p)$. The computational results obtained for different values of δ_k are summarized in Table 2.

The optimal solution of (6) is quickly obtained for $\delta_k = 0.1$ and $\delta_k = 0.05$. The case $\delta_k = 0.01$ requires much more time. We see that increasing the precision slightly improve the optimal value, i.e. the sum of the squared differences. Figure 8 illustrates the comparison between the predicted values and the

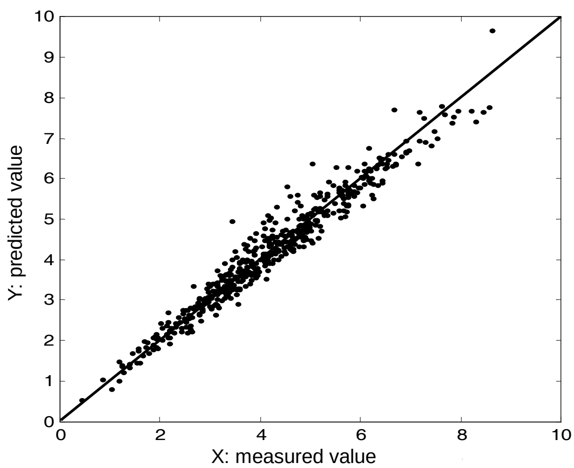

Figure 8. Model 1, Test 2: For each of the 500 measure points of Surface 2, the simulated measured value and the value predicted by the model are not far from each other.

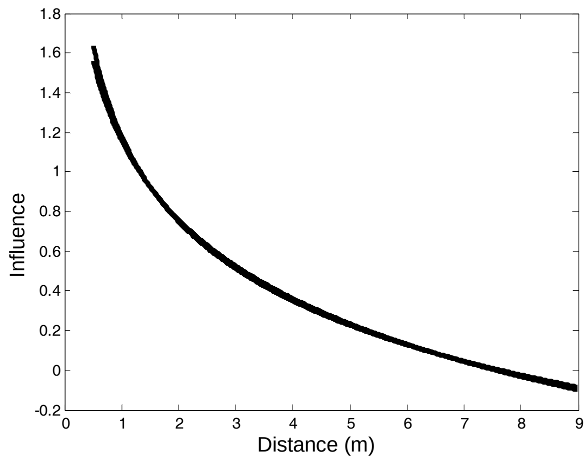

Figure 9. Model 1, Test 2: two almost identical individual influence functions for the objects of type 5.

simulated measured values, when $\delta_k = 0.1$. We would obtain almost the same figure for $\delta_k = 0.01$.

Though the optimal values are close for the three considered precisions, the estimated parameter values may be very different. For example, for objects of type 5 the estimated values of the three parameters are $a_5 = -5.22673$, $b_5 = 6.40614$ and $c_5 = -0.1$ when $\delta_k = 0.1$, and $a_5 = -56.6930$, $b_5 = 57.8522$ and $c_5 = -0.01$ when $\delta_k = 0.01$. The two functions are represented in Figure 9. We see that these very different parameter values define almost identical influence functions (both curves are superposed).

Figure 10 presents the simulated individual influence function, $g_k(d_{ij}) = g_k^0 e^{-b_k d_{ij}^2}$ and the estimated individual influence functions, $a_k + b_k d_{ij}^{c_k}$, for $k = 1, 3, 5$.

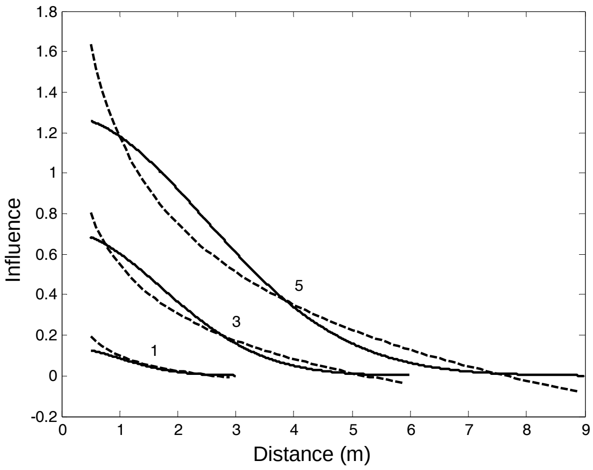

Figure 10. Model 1, Test 2: comparison of the simulated individual influence function with the estimated individual influence functions for the objects of types 1, 3 and 5.

5. Estimating Model 2 by Convex Quadratic Programming

Here, we consider that individual influence functions depend in a continuous way on a certain parameter characterizing these objects (See Section 2.2). For example, if objects are trees, in the first model these trees are distributed in various classes: the class 1 corresponds to the trees which diameters is between 20 and 30 cm, the class 2 corresponds to the trees which diameter is between 30 and 40 cm, etc. In this model, we consider influence functions depending in a continuous way on the diameter. We choose to consider functions of the form

$$f_i(d_{ij}) = a \cdot s_1(D_i) + b \cdot s_2(D_i) \cdot d_{ij}^{c/s_3(D_i)} \quad (\forall i,\ j;\ d_{ij} \le d_i^{\max})$$

$$f_i(d_{ij}) = 0 \quad (\forall i,\ j;\ d_{ij} > d_i^{\max}) \qquad (7)$$

where $f_i(d_{ij})$ is the individual influence of the tree O_i over the point of measure P_j, D_i is the diameter of the tree O_i, $s_t(D_i)$ ($t = 1, ..., 3$) is any real-valued function of D_i, and d_{ij} is the distance from the tree O_i to the point of measure P_j. A tree O_i has an influence over a point P_j if the distance between O_i and P_j is less than or equal to $d_i^{\max}$. The parameters a, b and c are the unknowns. By using the same idea as in the previous model, we choose a precision δ and discretize the variable c in the following way: $c = c^{\min} + \delta\sum_{r=0}^{r^{\max}} r \cdot w_r$ with $r^{\max} = [(c^{\max} - c^{\min})/\delta]$, w_r ($r = 0, ..., r^{\max}$) being Boolean variables subject to the constraint $\sum_{r=0}^{r^{\max}} w_r = 1$. By substituting in (7) $c^{\min} + \delta\sum_{r=0}^{r^{\max}} r \cdot w_r$ to c we get

$$f_i(d_{ij}) = s_1(D_i)a + s_2(D_i)b\sum_{r=0}^{r^{\max}} w_r d_{ij}^{(c^{\min}+\delta r)/s_3(D_i)},\ \forall i,\ j;\ d_{ij} \le d_i^{\max}$$

$$f_i(d_{ij}) = 0,\ \forall i,\ j;\ d_{ij} > d_i^{\max} \qquad (8)$$

The expression of $f_i(d_{ij})$ is not linear because of the products $b \cdot w_r$. In order to get a linear expression, we substitute, as in Section 3, the variable v_r to the product bw_r ($r = 0, ..., r^{\max}$) and add the set of linear constraints $C(v_r, b, w_r, b^{\max})$ that force the equality $v_r = bw_r$ ($r = 0, ..., r^{\max}$):

$C(v_r, b, w_r, b^{\max})(r = 0,...,r^{\max})$:

$$v_r \le b^{\max} w_r$$

$$v_r \le b$$

$$v_r \ge b - b^{\max}(1 - w_r)$$

$$v_r \ge 0 \qquad (9)$$

For $d_{ij} \le d_i^{\max}$ we then obtain the following expression of $f_i(d_{ij})$:

$$f_i(d_{ij}) = a \cdot s_1(D_i) + s_2(D_i)\sum_{r=0}^{r^{\max}} v_r \cdot d_{ij}^{(c^{\min}+\delta r)/s_3(D_i)} \qquad (10)$$

Estimating Model 2 can thus be formulated by the mixed-integer program (11) where the objective function is quadratic and convex, and all the constraints are linear. To sum up, an optimal solution $(\tilde{a}, \tilde{b}, \tilde{c})$ of (11) defines the estimated model when the possible values of c are $c^{\min}$, $c^{\min} + \delta$, $c^{\min} + 2\delta$, ..., $c^{\min} + r^{\max}\delta$:

$$\min \sum_{j=1}^{m} (F_j - \overline{F}_j)^2 \qquad (11)$$

$$\text{s.t. } F_j = \sum_{i \in I_j} (as_1(D_i) + s_2(D_i)\sum_{r=0}^{r^{\max}} v_r d_{ij}^{(c^{\min}+\delta r)/s_3(D_i)}),\ j = 1, ..., m \qquad (11.1)$$

$$c = c^{\min} + \delta\sum_{r=0}^{r^{\max}} r \cdot w_r \qquad (11.2)$$

$$\sum_{r=0}^{r^{\max}} w_r = 1 \qquad (11.3)$$

Table 3. Computational Results Regarding the Solution of (11) with Different Values of δ

δ	Optimal value of (11_δ)	CPU time in seconds	# nodes	a	b	c
0.1	292.2	4.1	94	-0.0267468	0.143138	-5.5
0.01	292.1	330.9	822	-0.0269857	0.143704	-5.45
0.01*	292.1	7.1	9	-0.0269857	0.143704	-5.45

*Three equal subintervals of δ are considered.

$$C(v_r, b, w_r, b^{\max}),\ r = 0,\ ...,\ r^{\max} \qquad (11.4)$$

$$a,\ b \in R,\ b \geq 0,\ w_r \in \{0,1\},\ r = 0,\ ...,\ r^{\max} \qquad (11.5)$$

with $I_j = \left\{ i \in \{1,\ ...,\ n\} : d_{ij} \leq d_i^{\max} \right\}$.

Remark. In this estimation problem we discretize only one variable, the variable *c*. If the possible values of *c* belong to a large interval $[c^{\min},\ c^{\max}]$, compared to the discretization step, one can speed up the solution of (11) by solving successively several instances of (11), each instance corresponding to a subinterval of $[c^{\min},\ c^{\max}]$.

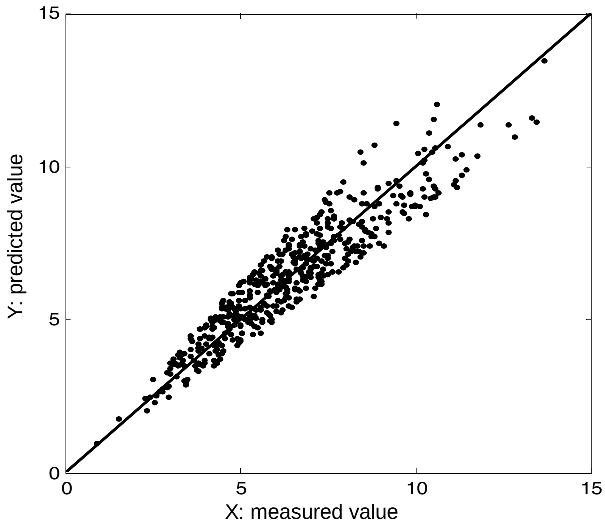

Figure 11. Model 2: for each of the 500 measure points of Surface 1, the simulated measured value and the value predicted by the model are not far from each other.

6. Experiments in Estimating Model 2

We consider the following (known) individual influence functions $f_i(d_{ij}) = D_i^{1.32}\overline{a} + D_i\overline{b}d_{ij}^{(\overline{c}/D_i)}$ to simulate values of a hypothetical object influence at the 500 points of Surface 1, with $\overline{a} = -0.029,\ \overline{b} = 0.15,\ \overline{c} = -5.12$. For each tree O_i, the value of D_i is uniformly and randomly generated in the interval [10, 50]. These influence functions are represented in Figure 3. In this test, we consider that the influence is null when its value is less than 0.01. Therefore, we get $d_i^{\max} = e^{-(D_i/5)\cdot\log(0.01+0.029D_i^{1.32})/0.15D_i}$. With these values of the parameters, $\overline{F}_j$ ranges from 1 to 13.4. The minimum, average and maximum number of objects that influence a measure point is equal to 9, 27.8, and 48, respectively. In the solution of (11) we fix, for each object O_i, $b^{\min} = 0$, $b^{\max} = 100$, $c^{\min} = -10$ and $c^{\max} = -1$. In this test, the optimal value of (11) with $\delta = 0.01$ is equal to 0 as expected and the corresponding solution provides exactly the coefficients chosen to test the approach, i.e. $a = -0.029$, $b = 0.15$, and $c = -5.12$. The solution requires 202.5 seconds of CPU time and 135 nodes in the search tree. The optimal value of (11) with $\delta = 0.1$ is equal to 0.01 and the corresponding solution provides the following coefficients: $a = -0.0291057$, $b = 0.150283$ and $c = -5.1$. The solution requires 2.1 seconds of CPU time and 52 nodes in the search tree. Now, consider the following random perturbation of $\overline{F}_j : \overline{F}_j = \varepsilon_j \sum_{i=1}^{n} f_i(d_{ij})$, ε_j being uniformly and randomly generated in the interval [0.8, 1.2]. The computational results obtained for different values of δ are summarized in Table 3.

The optimal solution with $\delta = 0.1$ is very quickly obtained (4.2 seconds). With the precision $\delta = 0.01$, 238.4 seconds are required but the optimal value does not change significantly.

We see in Table 3 that the estimated values of the three parameters *a*, *b* and *c* do not practically vary with the chosen precision contrary to what happens in Tables 1 and 2. Moreover, the CPU times required for determining the best parameter values are comparable in the three tables. However, we see in Table 3 that an alternative approach to solve (11) allows to significantly speed up the estimation. That is possible because in the model associated with Table 3, only one variable is discretized. That is not the case in the model associated with Tables 1 and 2, where the *p* variables c_k are discretized.

7. Conclusions

In this paper, we have considered the problem of estimating two spatial influence models by the least-squares criterion. We have formulated the problem by mathematical programming. In the obtained programs, the objective function is quadratic and convex, the constraints are linear and the variables are real or integer. This formulation is based on a discretization of some variables and on the linearization of quadratic terms. We did not study the pertinence of the obtained model; we only proposed an original method to estimate the parameters of a nonlinear function, in the least-squares framework. The method allows us to obtain the best parameter values (with the lest-squares criterion). The experimental results have shown the effectiveness of the method since we could easily solve the problems for large instances including up to 5000 influent objects and 500 points of measure. The results show the power of the mixed-integer quadratic programming technique for solving a problem with difficult nonlinear expressions. Solving this problem by mixed-integer quadratic programming has many advantages. First of all, for a researcher or a practitioner in environmental modeling, the approach is easy to understand and also easy to implement since it uses exclusively a standard, commercially available, software. The implementation of this approach becomes even easier if one uses a tool of modeling such as, for example, AMPL (Fourer et al., 1993). Secondly, least-squares minimization on large sets of data can be solved to optimality with a good precision in short computing times. In comparison, the time required by the Hooke-Jeeves algorithm (Hooke and Jeeves, 1961) used in (Kuuluvainen and Linkosalo, 1998) for estimating parameter values on data sets of comparable size varies from a few hours to several dozens of hours. However, for a fair comparison one must take into account the great computer power increasing during the last decade. The proposed method is particularly efficient if we know small intervals, compared to the discretization step, for the unknown parameter values to be estimated. The method could be immediately adapted to the estimation of the parameters when the criterion is the sum of the absolute values of the differences between measured values and predicted values. Future research would consist in trying to apply quadratic mixed-integer programming to the estimation of other pertinent environmental models.

Appendix: Mixed-integer Programming

Here we consider the minimization of a linear or convex quadratic function subject to linear constraints when some variables are real while others must take only integer values. In a general way, this optimization problem can be stated under the mixed-integer mathematical program P, given below, where R is the set of real numbers, and Z, the set of integer numbers:

$$\min f(x_1, x_2, ..., x_n) \tag{P}$$

$$\text{s.t. } \sum_{j=1}^{n} a_{ij} x_j \le b_i,\ i = 1,\ ...,\ m$$

$$x_j \in R,\ j = 1,\ ...,\ p$$

$$x_j \in Z,\ j = p+1,\ ...,\ q$$

If the objective function is affine, i.e. if $f(x_1, x_2 ..., x_n) = c_0 + \sum_{j=1}^{n} c_j x_j$ with $c_j \in R$ ($j = 0,\ ...,\ n$), then P is a *mixed-integer linear program*; if the objective function is quadratic and convex, i.e. if $f(x_1, x_2, ..., x_n) = \sum_{i=1}^{n} c_i x_i + \sum_{i=1}^{n} \sum_{j=i}^{n} c_{ij} x_i x_j + c_0$ with $c_i \in R$ ($i = 0,\ ...,\ n$), $c_{ij} \in R$ ($i = 1,\ ...,\ n;\ j = i,\ ...,\ n$) and $\sum_{i=1}^{n} \sum_{j=i}^{n} c_{ij} x_i x_j \ge 0$ for all $x \in R^n$, then P is a *convex mixed-integer quadratic program*. All the coefficients in the constraints of P, a_{ij} ($i = 1,\ ...,\ m;\ j = 1,\ ...,\ n$) and b_i ($i = 1,\ ...,\ m$), are real numbers. When in P the integer variables can only take binary values $[x_j \in \{0,\ 1\}$ ($j = p+1,\ ...,\ q$) in place of $x_j \in Z]$, P becomes a mixed-0/1 program. Mixed-integer programming has numerous applications in operations research and engineering design applications, and has been widely studied. There exist very effective algorithms to solve P when the objective function is linear or quadratic convex, and numerous commercial and academic software packages based on these algorithms are available.

The most widely used method for solving linear or quadratic mixed-integer programs is branch and bound. Subproblems are created by restricting the range of the integer variables. For binary variables, there are only two possible restrictions: setting the variable to 0, or setting the variable to 1. More generally, a variable with lower bound l and upper bound u will lead to two problems with ranges l to q and q+1 to u, respectively. Lower bounds are provided by relaxing integrality restrictions to derive a convex program that can be solved efficiently. If the optimal solution to a relaxed problem is integral, it is an optimal solution to the subproblem, and the value can be used to eliminate subproblems whose lower bound is higher. For more details about mixed-integer programming the reader can consult, for example, (Nemhauser and Wolsey, 1988), (Wolsey, 1998) and (Vanderbei, 2008).

References

CPLEX. (2007). *ILOG CPLEX 10.2.0 Reference Manual*, ILOG CPLEX Division, Gentilly, France.

Fourer, R., Gay, D.M., and Kernighan, B.W. (1993). *AMPL a modeling language for mathematical programming*, The Scientific Press Series.

Glover, F. (1975). Improved linear integer programming formulations of nonlinear integer problems, *Manage. Sci.*, 22(4), 445-460, doi:10.1287/mnsc.22.4.455.

Gill, P.E., and Murray, W. (1978). Algorithms for the solution of the nonlinear least-squares problem, *SIAM Journal on Numerical Analysis*, 15(5), 977-992, doi:org/10.1137/0715063.

Gill, P.E., and Wright, M.H. (1986). *Practical optimization*, Elsevier.

Glover, F.W., and Kochenberger, G.A. (Eds.) (2003). *Handbook of Metaheuristics*, International Series in Operations Research and Management Science, Springer.

Hooke, R., and Jeeves, T.A. (1961). Direct search solution of numerical and statistical problems, *J. Assoc. Comput. Mach.*, 8(2), 212-219, doi:10.1145/321062.321069.

Kuuluvainen, T., and Linkosalo, T. (1998). Estimation of a spatial tree-influence model using iterative optimization, *Ecological Modeling*, 106(1), 63-75, doi:10.1016/S0304-3800(97)00182-8.

Kuuluvainen, T., and Pukkala, T. (1989). Effects of Scots pine seed trees on the density of ground vegetation and tree seedlings, *Silva Fennica*, 23(2), 159-167.

Nemhauser, G.L., and Wolsey, L.A. (1988). *Integer and combinatorial optimization*, John Wiley & Sons.

Vanderbei, R.J. (2008). *Linear programming, foundations and extensions*, Springer.

Wolsey, L.A. (1998). *Integer Programming*, Wiley.