APPROXIMATE SOLUTIONS OF STOCHASTIC LINEAR PROGRAMS WITH ESTIMATED OBJECTIVE FUNCTION: AN APPLICATION OF RE-SAMPLING METHODS

Carlos N. BOUZA-HERRERA[1]
Sira M. ALLENDE-ALONSO[1]
Daniel C. CHEN[2]

- ***ABSTRACT:*** *In this paper we consider the effect of using an approximation to the upper bound of the Minimax optimal solution in Stochastic Linear Programming. It can be computed accepting the normality or by using a re-sampling method. Simulated Annealing is introduced for looking for the second stage solution. Least Absolute Deviation is proposed as an alternative to Least Squares for computing the initial solution. The proposals are compared through Monte Carlo Simulation experiments.*

- ***KEYWORDS****: Bootstrap; Jacknife; L1 norm; stochastic programming, environmental impact of solid waste compost.*

1 Introduction

Stochastic optimization models are used in many applications when some of the coefficients are unknown and of random nature. The touchstone example is given by a decision-maker that must make decisions under uncertainty. This problem is present in the establishing of an environmental policy. The study must lead to the establishment of tax protocols to the solid waste production. The payment of the enterprises should be maximum in terms of the contents of metals to assess environmental acceptability. This is an optimization problem.

The uncertainty may modeled by the randomness of the studied phenomena. A probability space $\{\Omega, \Sigma, P\}$ is determined. The problem to be solved is $Min\ \{g(x)=E_P\ [g(x,w]\ /x\in\chi,\ w\in\Omega\}$. Ω is the sample space, Σ is a convenient sigma algebra and P is the probability measure that generates the random events. It may be quite difficult to write down the theoretical model. A more complicated task is to compute the solution because the calculation of a numerical solution generally relies on rather complicated theoretical models. Even the solution of simple linear programming problems needs of the computation of the optimal solutions of non-linear models. The contemporary existence of large computing capacities has permitted to develop software packages for solving a

[1] Facultad de Matemática y Computación, Universidad de La Habana, San Lazaro y L. Vedado. CP 10400, La Habana, Cuba. E-mail: *bouza@matcom.uh.cu*
[2] Smith and King College. Lenin Sarajani #129, Calcutta India. E-mail: *Dchen19@gmail.com*

large collection of stochastic programs, see Kall-Wallace (1994) and Birgé-Louveaux (1997) for example.

Morita et al. (1989) analyzed the stochastic linear programming problems before the existence of the modern computers we have at hand nowadays. They developed a Minimax procedure for solving a particular class of problems. Their approach considered that the objective function $\boldsymbol{c}^t\boldsymbol{x}$ could be considered as a linear regression equation. The cost vector **c** is identified with the vector of the regression coefficients and a sample permits to estimate it. Considering that the Gauss-Markov's theorem was acceptable a one-sided confidence interval is calculable. They used the bound of the determined confidence interval for fixing a constraint. A minimization problem was settled and the solution was plugged-in a second stage problem where the optima was computed.

The idea is interesting but the main stream of the research in the theme continued going on the line of deterministic equivalents determination, spread the original problem into scenarios, computation of the solution and averaging and in developing sensitive analysis for establishing the convergence of the approximate solutions. See for example Bouza (1992), Dupačovà (1983), Kankovà (1994), Römisch-Schultz (1992).

In this paper we revisit the proposal of Morita et al. (1989) using the statistical point of view developed by them. We have some criticisms on their proposal:

- The Least Squares estimator may not be an adequate starting point.
- The stopping rule of the proposed algorithm does not guarantee to not be trapped in local minimal.

In this paper we propose to use a Least Absolute Deviation estimation of **c** As an alternative to the Least Squares estimator. It should permit to obtain better results when the normality assumption is not valid. This estimator has proved to behave better than Least Squares in such cases. See Arthanari-Dodge (1993) for a discussion. We also develop an alternative algorithm that includes a Simulated Annealing (SA) procedure for escaping form the local minima trap existent in Morita et al. (1989).

Another intent was made to improve the solution, taking into account that the hypothesis, needed for ensuring the optimality of Least Squares, may not be valid. We proposed to use resampling procedures, due to their robustness, and developed alternative models based on Bootstrap and Jackknife techniques.. Section 2 is devoted to develop the needed discussions on the models . The search method is analyzed for the initial model and for its SA alternative . The algorithms are also presented.

In Section 3 the proposed alternatives are compared by means of Monte Carlo experiments using the example discussed by Morita et al. The regression equation determined as optimal was used as a basis for generating a large population determined by a grid defined by the constraints of the variables involved. Random values for the residuals were generated and added to the computed responses. Distributions with parameters in the interval (0,1) were used. The distribution functions were the Uniform in (0,1), the Exponential with λ=1, the Standard Normal, N(0,1), and the Standard Double Exponential, L(0,1). The results obtained suggest that the use of SA averages the best results. When the initial solution is computed using Least Absolute Deviation the behavior is better than those based on Least Squares.

2 The Basic Problem

A commonplace model in Linear Programming Problem (LPP) denoted as P1: Maximize $\{\boldsymbol{c}^t\boldsymbol{x} \,/\boldsymbol{Ax}=\boldsymbol{b},\ \boldsymbol{x}\geq 0\}$, where $\boldsymbol{x}\in\Re^n$ is the vector of decision variables , $\boldsymbol{c}\in\Re^n$ is the cost vector which components are known constants and $\boldsymbol{A}$ is a full rank *nxm* matrix. The feasibility region of P1 is considered to be bounded and different from the empty set $\varnothing$.

In many practical cases **c** is unknown and we are dealing with a Stochastic LPP, (SLPP).

Different techniques have been considered for solving the SLPP. Morita et al. (1989) considered that we can observe a function $g(\boldsymbol{c},\boldsymbol{x})$ but not $\boldsymbol{c}$. Then we can fix an $\boldsymbol{x}^*$ and to observe a response which is modeled as:

$$Y=g(\boldsymbol{c},\boldsymbol{x}^*)+\varepsilon$$

ε is an unobservable random variable that express the difference between the observed Y and the predicted $Y^*=g(\boldsymbol{c},\boldsymbol{x}^*)$. Then we may fix a set $\boldsymbol{X}_n=\{\boldsymbol{x}_1^*,\ldots,\boldsymbol{x}_v^*\}$ and to write the relation as a regression equation:

$$\boldsymbol{Y}=\begin{pmatrix} Y_1 \\ \vdots \\ Y_v \end{pmatrix}=\begin{pmatrix} g(c,x^*_1) \\ \vdots \\ g(c,x^*_v) \end{pmatrix}+\begin{pmatrix} \varepsilon_1 \\ \vdots \\ \varepsilon_v \end{pmatrix}=G(c,X_v)+\underline{\varepsilon}$$

Once a certain norm $\|\bullet\|_p$ is determined we look for the solution of the problem:

$$R1:\ \boldsymbol{c}^*=ArgMin\ \{\|\boldsymbol{Y}-G(\boldsymbol{c},X_n\|_p\}$$

Morita et al. (1989) considered the case in which

$$\boldsymbol{Y}=\begin{pmatrix} x^*_{11}\cdots & x^*_{1n} \\ \cdot \quad \cdots & \cdot \\ \cdot \quad \cdots & \cdot \\ \cdot \quad \cdots & \cdot \\ x^*_{1v}\cdots & x^*_{vn} \end{pmatrix}\begin{pmatrix} c_1 \\ \vdots \\ c_n \end{pmatrix}+\begin{pmatrix} \varepsilon_1 \\ \vdots \\ \varepsilon_n \end{pmatrix}=Xc+\underline{\varepsilon} \qquad (2.1)$$

and used the Least Square solution $c_0=\left(X^tX\right)^{-1}X^tY=H^{-1}X^tY$ obtained from the solution of the equation system:

$$\frac{\partial\sum_{i=1}^{v}\left(Y_i-\sum_{j=1}^{n}c_jx^*_{ij}\right)^2}{\partial c_j}=0 \qquad j=1,\ldots,n \qquad (2.2)$$

This is the best solution when the hypothesis of Gauss Markov theorem holds That is if $E(\boldsymbol{\varepsilon})=\boldsymbol{0}$, $Cov(\boldsymbol{\varepsilon})=\sigma^2\boldsymbol{I}$, and $\boldsymbol{X}$ is of full rank.

Assuming that $\boldsymbol{c}_0$ is optimal solution and that the normality is a valid assumption , the confidence region for the true **c** is associated with

$$(c-c_0)^t H(c-c_0) \le ns^2 F(n,\nu-n,1-\alpha) = \lambda$$

where

$$s^2 = \frac{\sum_{i=1}^{\nu}(Y_i - \sum_{j=1}^{n} c_j X_{ij})^2}{\nu - n}, \qquad \alpha \in [0,1]$$

and $F(n, \nu\text{-}n, 1\text{-}\alpha)$ is the corresponding percentile of the *F*-distribution function.

Returning to the linear programming problem (LPP) we can quote that the unknowledge of **c** may be solved by considering the estimate $\boldsymbol{c}_o$ and applying a convenient plug in rule. The use of this estimate generates a stochastic LPP (SLPP) . Morita et al. (1989) considered the confidence region determined an estimation of the feasibility region and proposed to solve a two-stage problem denoted as TSP1 as follows

TSP1.
Solve is stage 1: P1**c:** $f(\boldsymbol{x}_{opt})=ArgMax\ \{\boldsymbol{c}^t\boldsymbol{x} \ / \boldsymbol{Ax=b},\ \boldsymbol{x}\ge\boldsymbol{0}\}$
Solve in stage 2:P2**c:** Minimize $\{g(\boldsymbol{c}) / (\boldsymbol{c}-\boldsymbol{c}^*)^t H (\boldsymbol{c}-\boldsymbol{c}^*) \le \lambda\}$

They proved that if Ω is the feasible solution set :

M1: $f(c)=Max\ \{\boldsymbol{x}_s^t \boldsymbol{c} / \boldsymbol{x}_s^t \in \Omega\}$ is a piecewise convex linear function ,
M2: P2**c** has a unique optimal solution.
M3. There is an explicit solution for:

$$LSP1_{sub}(x): \quad c_{sub} = c^* - H^{-1}x\sqrt{\frac{\lambda}{x^t H^{-1} x}} = ArgMax\{x^t c \,|\, (c-c^*)^t H (c-c^*) \le \lambda\}$$

M4. A solution $\boldsymbol{x}_0 = [x'_o \ x''_o]$ is an optimal solution of $LSP1_{sub}(\mathbf{x})$ only if the solution $\boldsymbol{c}_{sub} = [\boldsymbol{c}' \ \boldsymbol{c}'']$ satisfies $\boldsymbol{c}''^t - \boldsymbol{c}''^t A'^{-1} A'' \le 0$ where $\boldsymbol{A} = [A' \ A'']$ and being A' the basis matrix.

From the fact that $E[\boldsymbol{c}^*] = \boldsymbol{c}$, the Law of Large Numbers sustains the validity of the convergence of $\boldsymbol{c}^*$ to **c** with probability one (w.p.1) as $\nu \longrightarrow \infty$. Using the reasoning of Morita et al. (1989) we have that any reasonably efficient estimation method may be used without affecting the properties of the feasible solutions.

Using these facts we propose the following solution algorithm.

Algorithm 1: Minimax Solution Algorithm (MSA)

Step 0: Solve the Least Squares problem and set the estimate $\boldsymbol{c}_o$. Fix $k=0$ and $J=\varnothing$
Step 1: Solve P1ck and determine the optimal solution xk.
Step 2: If xk$\in$ J, then go to step 5. Else $J=J+\{xk\}$.
Step 3: Compute the solution to LSP1sub(xk)

$$c_{k+1} = c_0 - H^{-1} x_k \sqrt{\frac{\lambda}{x_k{}^t H^{-1} x_k}}$$

Step 4: Set $k=k+1$ and go to Step 1.

Step 5: If $\boldsymbol{x}_k=\boldsymbol{x}_{k+1}$, then $\boldsymbol{x}_k$ is the optimal solution and go to END.
Else solve P2$\boldsymbol{c}$: $\boldsymbol{c}_{k+1} = ArgMin \{f(\boldsymbol{c}) / (\boldsymbol{c}-\boldsymbol{c}_k)^t H (\boldsymbol{c}-\boldsymbol{c}_k) \leq \lambda\}$ and set $k=k+1$.
Go to step 1.

END

This search procedures ends in a finite number of iterations.

In Physics, Simulated Annealing describes the way in which a metal cools and freezes into a minimum energy crystalline structure. Metropolis et al. (1958) considered it as a way of obtaining the equilibrium configuration of a collection of atoms at a given temperature. The similitude with the problems arising in Mathematical Optimization in the search of an optimum was pointed out by Pincus (1970). Kirkpatrick et al. (1982) were the first in proposing it as an alternative for computing the optimal solution of discrete problems. We consider that using SA should improve the initial procedure of Morita et al. (1989)

The SA method has the ability for escaping from local minima traps. It accepts each new solution that is a better movement but a bad movement may be accepted with a probability $P=exp(-\Delta f / T)$

The inputs in the probability function are :

- Δf, which is the increase of the function.
- T, the temperature of the system.

The practical implementation needs of:

- *A representation of the possible solutions*
- *A generator of random changes in the solution*
- *A mean of evaluating the probability function*
- *An annealing scheduling, an initial temperature and rules for lowering it as the search progress.*

Aarts-Korst (1989) present the basic theory and discuss its applications, Brooks-Morgan (1994) present an application of SA in optimization.

The usual literature is on the use of SA for discrete variables. When we deal with the continuous case some small change should be introduced. Vanderbilt-Louie (1984) proposed to determine in a new trial the relation $\boldsymbol{c}_{k+1}=\boldsymbol{c}_k+Qu$, where u is uniform variable defined on the interval $[-3^{1/2},3^{1/2}]$ and Q is matrix that contains the steps size distribution. It is obtained using Choleski's decomposition from $\Xi_k=QQ^t$, where Ξ_k is deduced from a function of the covariance, of the path actually followed, ϑ and the sampling constant α, which controls the rate at which information on ϑ is folded into Ξ, and with a weight ω. That function *is* $\Xi_{k+1}= (1-\alpha)\Xi_k + \alpha\omega\vartheta$.

A less time consuming alternative procedure is given by the use of:.$\mathbf{c}_{k+1}=\mathbf{c}_k+\boldsymbol{D}u$, where u is uniform variable defined on the interval $[-1, 1]$ and $\boldsymbol{D}$ is a diagonal matrix that defines the maximum change allowed in each variable. After a successful trial D is

updated by $\boldsymbol{D}_{k+1}= (1-\alpha)\boldsymbol{D}_k + \alpha\omega\boldsymbol{R}$, where $\boldsymbol{R}$ is a diagonal matrix with the magnitude of the sample changes of each control variable. This tunes the maximal step associated with each control variable towards an optimal value . The recommended probability function is $P^*=exp(-\Delta\delta f/d^*T)$.

Taking d^* as a measure of the mean effectiveness of the change rule. As the size of the step taken is considered $\boldsymbol{D}$ does not need to be changed when T changes. Another interesting proposition appears in Corana et al. (1987).

We propose to use as alternative the Algorithm 2 which uses SA as a search procedure.

Algorithm 2: Simulated Annealing Minimax Solution Algorithm (SAAMSA)

Step 0: Solve the Least Squares problem and set the estimate $\boldsymbol{c}_o$. Fix $k=0$ and $J=\varnothing$, $\boldsymbol{D}$, α, ω, β, $\boldsymbol{R}$, t^*, d^* and T.
Step 1: Solve P1c_k and determine the optimal solution x_k.
Step 2: If $x_k \in J$, then go to step 7. Else $J=J+\{x_k\}$.
Step 3: Compute the solution to LSP1sub(x_k)

$$c_{k+1} = c_o - H^{-1}x_k\sqrt{\frac{\lambda}{x_k^{\,t}H^{-1}x_k}}$$

Step 4: If $f=c_k^{\,t}x_k - c_{k+1}^{\,t}x_{k+1} \leq 0$ then generate u using a $U[-1,1[$.
Set $t=t+1$
Else go to Step 6
Step 5: If $P^*=exp(-\Delta\delta f/d^*T)>\beta$
Else go to step 7
Update $\boldsymbol{D}_{k+1}= (1-\alpha)\boldsymbol{D}_k + \alpha\omega\boldsymbol{R}$ and compute $\boldsymbol{c}_k=\boldsymbol{c}_k+\boldsymbol{D}u$.
While $t<t^*$ go to Step 4
Step 6: Set $k=k+1$, $t=0$ and go to Step 1.
Step 7: If $\boldsymbol{x}_k=\boldsymbol{x}_{k+1}$, then $\boldsymbol{x}_k$ is the optimal solution and go to END.
Else solve P2**c:** $\boldsymbol{c}_{k+1} = argmin\{f(\boldsymbol{c}) / (\boldsymbol{c}-\boldsymbol{c}_k)^t H (\boldsymbol{c}-\boldsymbol{c}_k) \leq \lambda\}$ and set $k=k+1$.
Go to step 1.
END

This algorithm is not a pure SAA but uses its ideas for improving the search.

The hypothesis on the normality of the residuals in regression may be not true in many cases. We consider some alternatives for coping with this situation. The use of Least Absolute Deviation (LAD) for computing the initial solution $\boldsymbol{c}_0$ is a possibility. This method works better than LS when the distribution of the errors has heavy tails. See Dodge (1987) for an introduction to the theme and Arthanari-Dodge (1993) for a detailed discussion.

Its computation is readily obtained by solving the optimization problem

$$R2:\ \mathbf{c^{**}} = ArgMin\ \{\|\mathbf{Y}-G(\mathbf{c}, X_n\|_{p=1}\}.$$

When (2.1) is the model to be adjusted we need to solve the system

$$\frac{\partial \sum_{i=1}^{\nu}\left|Y_i-\sum_{j=1}^{n} c_j x^*_{ij}\right|}{\partial c_j} = \frac{\sum_{i=1}^{\nu}\left(Y_i-\sum_{j=1}^{n} c_i x^*_{ij}\right)}{\left|Y_i-\sum_{j=1}^{n} c_i x^*_{ij}\right|}(-x^*_{ij}) = 0 \qquad j=1,...,n \tag{2.2}$$

Then we have the relation $\dfrac{\sum_{j=1}^{n} x^*_{ij}\, c_j{}^{**}}{\varepsilon(c_k{}^{**})} = \dfrac{\sum_{j=1}^{n} x^*_{ik}\, x^*_{ij}\, c_j{}^{**}}{\varepsilon(c_k{}^{**})} \qquad j=1,..,n$

which can be expressed in matrix terms as $\mathbf{X^tD(x^{**})Y=X^tD(x^{**})Xc^{**}}$, taking $\boldsymbol{D(x^*)=[\varepsilon(c^{**})]^{-1}}$ as a diagonal matrix containing the estimated residuals. Then, under the necessary inversibility hypothesis, we have that $\boldsymbol{c^{**}= [X^tD(c^{**})X]^{-1}X^tD(c^{**})Y}$. The solution of this system is made by the successive application of the procedure described as follows

LAD Solution Procedure for Computing c (LADPC).**

Set $\boldsymbol{c^1=0}$, $k=0$

While $k<n$ do

$\boldsymbol{.c^{k+1}=[X^tD(c^{**k}X]^{-1}X^tD(c^{*k})Y}$

$k=k+1$

END.

The most common solution is based in the use of the LPP

LP**c****: Minimize $\sum^{\nu}_{i=1} /y_i-\sum_{j=1}{}^{n}\ c_jx_{ij} /=\sum^{\nu}_{i=1} /\varepsilon_i(\mathbf{c^{**}}) /$

Subject to : $-\varepsilon_i(\mathbf{c^{**}})\le y_i-\sum_{j=1}{}^{n}\ c_jx_{ij} \le \varepsilon_i(\mathbf{c^{**}}),\quad i=1,\ldots,\nu$

Dodge (1997) discusses how to use LAD and the behavior of the most popular computing procedures.

Substituting $\boldsymbol{c^*}$ by $\boldsymbol{c^{**}}$ in Algorithms 1 and 2 we have other alternatives for solving **TSP1.** We will denote them by MSA-LAD and SAAMSA-LAD.

Another problem is related with the real effectiveness of the *F* distribution based on a one-tail confidence interval, which is used in both algorithms. We propose using re-sampling methods for computing it. The usual competitors are Bootstrap and Jackknife. Both are robust and may be used for computing normal based CI' s. We may regard Least Squares and Least Absolute Deviation estimators as particular cases of *M*-estimation. This class of estimators is defined by the minimization of $\sum_{i=1}{}^{\nu}\ \rho[g(\varepsilon)]$.

When we deal with Least Squares $g(\varepsilon)=(\varepsilon)^2$ and $g(\varepsilon)= |\varepsilon|$ if we use Least Absolute Deviation. Once we use one of these methods, for estimating $\mathbf{c}$ and obtain the estimate $\mathbf{c_e}$, we can estimate the residuals for certain observation i using $\varepsilon(\boldsymbol{c}_e)=y_i-\boldsymbol{c}_e^t x^*_i$

The empirical distribution of the X's is given by $F_\nu(\boldsymbol{x})= \sum_{i=1}^{\nu} I\,[\boldsymbol{x}_i \le \boldsymbol{x}]/\nu$. The residuals are independently distributed of $\boldsymbol{x}$ and their empirical distribution is $F_\nu(\varepsilon)= \sum_{i=1}^{\nu} I\,[\varepsilon(\boldsymbol{c}_e) \le \varepsilon]/\nu$, where

$$I[Z \le z] = \begin{cases} 1 & if \quad Z \le z \\ 0 & otherwise \end{cases}, \; Z=\boldsymbol{x},\ \varepsilon.$$

Using these facts we generate the pairs $(y^*_g, \boldsymbol{x}^*_g)$ by means of the selected re-sampling procedure dealing with an empirical distribution $F_\nu(y)= F_\nu(\boldsymbol{x})\ F_\nu(\varepsilon)$. The procedure π must generate an $\boldsymbol{x}^*$ and an ε^* independently and $.y^*_{i(\pi)} = \boldsymbol{c}_e^t \boldsymbol{x}^*_{i(\pi)} + \varepsilon_{i(\pi)}$

When Bootstrap is used we generate independently B samples of size ν

$$s_b = \{(y^*, \boldsymbol{x}^*_g)_h\ ,\ h=1,\ldots,\nu\},\ b=1,..,B.$$

Each sample allows computing an estimate of $\boldsymbol{c}$. Hence we have B replications $\boldsymbol{c}_{1(boot)},\ldots,\ \boldsymbol{c}_{B(boot)}$. We have that $.\boldsymbol{c}_{\boldsymbol{boot}}=\sum_{b=1}^{B} \boldsymbol{c}_{b(boot)}/B$ is the bootstrap estimate of $\boldsymbol{c}$ and $s^2_{boot}=\sum_{b=1}^{B} [\boldsymbol{c}_{b(boot)}-.\boldsymbol{c}_{\boldsymbol{boot}}]^2 /B\text{-}1$ is a consistent estimator of the $\mathbf{c}$-estimator's variance. This fact sustains the validity of using as an approximation of the distribution of $\frac{c_{boot} - c_{(e)}}{s_{boot}}\sqrt{B}$ a T-student with $B\text{-}1$ degrees of freedom, see Davidson- Hinkley (1997).

Using the classic Bootstrap procedure we have the Bootstrap version of Algorithm 1.

The use of Jackknife establishes that all the samples of size ν-1 should be evaluated and the $j\text{-}th$ sample is $.s_j = \{(y^*, \boldsymbol{x}^*_g)_h\ ,\ h \in \{1,..,\nu\}\backslash\{j\}\},\ j=1,..,\nu.$ An estimate of $\mathbf{c}$ is computed in each sample and we obtain the pseudo values $\boldsymbol{c}_{1(J)},\ldots,\ \boldsymbol{c}_{\nu(J)}$. Their arithmetic mean $.\mathbf{c_{jack}}=\sum_{j=1}^{\nu} \mathbf{c}_{1(J)}/\nu$ is the Jacknife estimate of $\boldsymbol{c}$ and $s^2_{Jack}=\sum_{j=1}^{\nu} [c_{j(J)}-.c_{Jack}]^2 /\nu\text{-}1$ converges to the true variance and the statistic $\frac{c_{Jack} - c_{(Jacke)}}{s_{Jackt}}\sqrt{\nu}$ has a distribution function that converge to a T-Student with $\nu\text{-}1$ degrees of freedom.

Note that the solution $\boldsymbol{x}_k$ at stage k does not play a role in the calculation of the upper bound. There is not an explicit expression of the optimum $\mathbf{c_k}$. Then we have to solve an optimization problem in each stage and in each run. The use of SA is the only way to search for an optimum, looking for an improvement in the neighborhood of the solution obtained. We developed an algorithm for determining an optimum. We identify with ς = Bootstrap, Jackknife the re-sampling procedure used.

Algorithm 3: Minimax Solution Algorithm (MSA) with Re-sampling ς for the estimation of the error

Step 0: Solve the Least Squares problem and set the estimate $\boldsymbol{c}_o$. Fix $k=0$ and $J=\varnothing$.
Step 1: Fix ς

If ς=Boostrap fix $q=B$
If ς=Jacknife fix $q=\nu$

Step 2: Solve P1$\mathbf{c_k}$ and determine the optimal solution $\mathbf{x_k}$.

Step 3: If $\boldsymbol{x}_k \in J$, then go to step 5. Else $J=J+\{\boldsymbol{x}_k\}$.

Step 4: Compute the solution to LSP1*$_{sub}(\mathbf{x_k})$

$$LSP1^*_{sub}(x_k): \quad c_{sub} = ArgMax\{x^t c_{k+1} \Big| c_{k+1} \le c_\varsigma + t(\nu - 1, 1-\alpha)\frac{s_\varsigma}{\sqrt{q}}\}$$

Step 5: If $f=c_k^t x_k - c_{k+1}^t x_{k+1} \le 0$ then generate u using a $U[-1,1[$.
Set $t=t+1$
Else go to Step 7

Step 6: If $P^*=exp(-\Delta \delta f/d^*T) > \beta$
Else go to step 8
Update $D_{k+1}= (1-\alpha)D_k + \alpha\omega R$ and compute $\boldsymbol{c}_k=\boldsymbol{c}_k+Du$.
While $t<t^*$ go to Step 5

Step 7: Set $k=k+1$, $t=0$ and go to Step 2.

Step 8: If $\boldsymbol{x}_k=\boldsymbol{x}_{k+1}$, then $\boldsymbol{x}_k$ is the optimal solution and go to END.
Else solve P2**c:** $\boldsymbol{c}_{k+1} = argmin\{f(\boldsymbol{c}) / (\boldsymbol{c}-\boldsymbol{c}_k)^t H (\boldsymbol{c}-\boldsymbol{c}_k) \le \lambda\}$ and set $k=k+1$.
Go to step 2.

END.

3 A comparison of the behavior of the Algorithms.

3.1. Comparison with the results of MORITA, H.; ISHII, H.; NISHIDA

A comparison of the performance of the first two algorithms was made. The example of Morita et al. (1989) was used as a basis. It was:

Maximize $c_1x_1 + c_2x_2$
Subject to:
$x_1 + 3x_2 \le 15$
$x_1 + 2x_2 \le 11$
$2x_1 + x_2 \le 14$
$x_1, x_2 \ge 0$

With $\boldsymbol{c}^t = (1.282 \quad 1.684)$, $s^2=0.2884$, $F(2,18, 0.95)=3.55$ and

$$H=\begin{pmatrix} 190 & 165 \\ 165 & 157.5 \end{pmatrix}$$

Hence $\lambda=2.048$.

The region defined by the constraints was divided determining a grid of rectangles with the same proportions. Each intersection point was taken as an input of the vector

$\boldsymbol{x}^t = (x_1 , x_2)$ and $\boldsymbol{y} = \boldsymbol{c}^t\boldsymbol{x}$ was computed. Using these inputs in each experiment we generated random errors with one of the following distributions: Uniform in *(0,1),* N*(0,1*), Exponential *(1),* Lognormal *(0,1)* and Double Exponential *(0,1).* We added them to the computed values of y and solved the problem TSP1 using each algorithm with a 10% of the possible $\boldsymbol{x}$-vectors *(*ν*=250*). The input parameters for the SAMinimax Solution Algorithm were the default ones. The value of *B* was approximately the 50%(*B=120*) of the sample size. We made *50* Monte Carlo experiments with each distribution and averaged $O_{ma}=\sum_r^{50} f(\boldsymbol{x_{opt}})_{r(ma)}/50$, where m identifies the distribution used and the algorithm. The corresponding standard deviation $\sigma_{ma} = \sqrt{\sum_r^{50} [f(\boldsymbol{x_{opt}})_{r(ma)} - O_{ma}]^2/49}$ was also computed.

The program was written in MatLab and used interfaces with codes in SAS (SAS (1999)) or provided by Blossom, Code-Richards (1999). For the computation of the estimates of the regression coefficients using Least Absolute Deviation), Lindo sources, see Lindo (1997), were used for solving the optimization sub-problems.

The results of the use of the first two algorithms appear in Table 1. An analysis of them establishes that the Algorithm 2 performed generally better than Morita's algorithm when a non normal distribution generates the residuals. They performed similarly when the distribution was U(0,1) or E U(0,1) or Exp. (1) in terms of the mean of the optimal values of the optimal values but SAA is associated with smaller variability.

Tabela 1 - Behavior of the Algorithms 1 and 2 in the Computation of the Optimal Values in 50 Monte Carlo Experiments with a Least Squares Estimation of $\boldsymbol{c}$

Distribution	Mean of the Optimal. Values		Standard Deviation of the Optimal Values	
	MSA.	SAAMSA	MSA.	SAAMSA
Exponential (1)	21.394	17.436	5.333	4.781
Uniform (0,1)	19.802	17.362	3.140	0.154
Normal (0,1)	10.893	10.976	0.647	0.662
Lognormal (0,1)	17.720	13.812	3.244	0.697
Double Exponential (0,1)	28.326	15.427	1.392	0.287

The results of the Monte Carlo, experiments using Least Absolute Deviation are given in Table 2 below. It presents the performance of the Algorithms 1 and 2 in the computation of the optimal values when the L_1 criteria is used for computing **c.** They suggest that, as expected, a better behavior for non normal distributions is present. This fact should be a consequence the properties of L_1 norm optimization, because it behaves better with heavy tailed distributions than its L_2 counterpart. Therefore the initial value of $\boldsymbol{c}$ was closer to the optimum when we use Least Absolute Deviation and convergence to the true optimum was better. In the case of normality the results are not better. The variability of the results is generally a little higher.

Tabela 2 - Behavior of the Algorithms 1 and 2 in the Computation of the Optimal Values in 50 Monte Carlo Experiments with a Least Absolute Deviation Estimation of **c**

Distribution	Mean of the Optimal. Values		Standard Deviation of the Optimal Values	
	MSA-LAD.	SAAMSA-LAD	MSA-LAD	SAAMSA-LAD
Exponential (1)	15.414	14.083	1.162	1.142
Uniform (0,1)	16.229	13.001	2.65	1.459
Normal (0,1)	12.965	11.162	1.504	1.112
Lognormal (0,1)	17.482	14.062	2.278	1.304
Double Exponential (0,1)	11.721	11.489	2.752	1.327

Table 3 is concerned with the behavior of the Algorithm 1 in computing the optimal values when the Least Squares estimation method was used for deriving **c** using Boostrap and Jacknife A look to Table 3 settles that the differences between Bootstrap and Jacknife are not high in terms of the mean of the computed optimal values. The standard deviations of the optimal values are smaller for Bootstrap. For normal-friendly distributions Bootstrap had a better behavior. Note that there is a considerable gain in accuracy with respect to the use of the pure Algorithm 1.

Tabela 3 - Behavior of the Algorithm 1 in the computation of the optimal values in 50 Monte Carlo experiments with Least Squares estimation of **c** using Boostrap and Jacknife

Distribution	Mean of the Optimal. Values		Standard Deviation of the Optimal Values	
	Bootstrap	Jacknife	Bootstrap	Jacknife
Exponential (1)	21.371	21.337	3.667	4.090
Uniform (0,1)	19.821	19.629	2.654	3.162
Normal (0,1)	10.981	10.991	0.255	0.409
Lognormal (0,1)	17.499	17.923	2.994	3.667
Double Exponential (0,1)	27.151	27.289	1.392	1.579

Table 4 presents the results when the Algorithm 1 is used but Least Absolute Deviation is used, instead of Least Squares, in the computations . Its analysis gives a similar panorama : Bootstrap and Jacknife have a similar behavior in terms of the average of the optimal value. Bootstrap is evidently better with the Normal and the Lognormal distributions and the optimal values computed are less variable than the Jacknife counterparts.

Tabela 4 - Behavior of the Algorithm 1 in the computation of the optimal values in 50 Monte Carlo experiments with Least Absolute Deviation estimation of ***c*** using Boostrap and Jacknife

Distribution	Mean of the Optimal. Values		Standard Deviation of the Optimal Values	
	Bootstrap	Jacknife	Bootstrap	Jacknife
Exponential (1)	18.116	17.956	5.021	4.652
Uniform (0,1)	17.986	17.721	0.281	0.196
Normal (0,1)	10.825	11.053	0.655	0.671
Lognormal (0,1)	13.552	13.912	0.724	0.821
Double Exponential (0,1)	15.821	15.839	0.301	0.294

The performance of the Algorithm 2 is presented in Tables 5 and 6. Table 5 is concerned with the study of the use of Least Squares estimation of ***c*** using Boostrap and Jacknife . Table 6 presents the results when Least Absolute Deviation is the method used for deriving the estimates of ***c.***

The analysis of the data given in Table 5 establishes that the means of the optimal values are very similar but that the Jacknife averages a smaller value. It is significant that this method also produces optimal values with a smaller standard deviation. A look to the *F* based counterpart intervals leads to conclude that, the use of resampling methods generates, consistently, better results. This fact is sustained by the analysis of the corresponding standard deviations. They are considerably smaller. Hence the computer intensive based confidence intervals should be preferred to the *F* ones.

Tabela 5 - Behavior of the Algorithm 2 in the computation of the optimal values in 50 Monte Carlo experiments with Least Squares estimation of ***c*** using Boostrap and Jacknife

Distribution	Mean of the Optimal. Values		Standard Deviation of the Optimal Values	
	Bootstrap	Jacknife	Bootstrap	Jacknife
Exponential (1)	17.415	17.413	4.928	4.653
Uniform (0,1)	18.026	17.602	0.302	0.188
Normal (0,1)	10.870	10.993	0.645	0.683
Lognormal (0,1)	13.533	13.987	0.766	0.797
Double Exponential (0,1)	15.902	15.853	0.3297	0.284

The results in the Table 6 are very similar to those reported in Table 5. Jacknife averages a smaller value of the mean optimal values and standard deviations too.

Tabela 6 - Behavior of the Algorithm 2 in the Computation of the Optimal Values in 50 Monte Carlo Experiments with Least Absolute Deviation Estimation of $\boldsymbol{c}$ using Bootstrap and Jacknife

Distribution	Mean of the Optimal. Values		Standard Deviation of the Optimal Values	
	Bootstrap	Jacknife	Bootstrap	Jacknife
Exponential (1)	18.771	17.737	5.122	4.453
Uniform (0,1)	17.802	17.600	0.244	0.122
Normal (0,1)	10.822	10.969	0.664	0.689
Lognormal (0,1)	13.821	14.039	0.788	0.810
Double Exponential (0,1)	15.594	15.594	0.369	0.223

3.2 An application to Environmental Asses

The municipal authorities uses composting for reducing the volume and mass of the solid waste, looking for obtaining stable organic material. The metal contents must be extracted before producing an organic material usable for the enrichment of soils. The municipality needs to establish taxes to the enterprises producing the grab highly contaminated as the metals should be extracted.

The decision variables are the costs to be assigned to the percentage of the contents of each metal. The percent is a random variable. The tax to be assigned looks to discourage the inclusion of metals in the solid waste of contaminants.

We posed the SLP $\boldsymbol{Y=g(}c, x^*\boldsymbol{)+\varepsilon}$. **T**he data obtained from 76 enterprises were analyzed.

The contains considered as 'highly contaminant' were: aluminum, cadmium, chrome, copper, iron, mercury, nickel, plumb and zinc.

The optimal values computed by each method appear in the Table 7. Analyzing it we may note that:

- The procedure proposed by Morita et al. (1989) is faster but it is often trapped by a local maxima.
- SA based algorithms improve the feasible solution considered as optimal by that procedure.
- The use of resampling ,for obtaining the upper bound, yields the acceptation of higher values as close to the maximum.
- The use of resampling represents an increment in the needed computing time.

Table 7. Solid Waste Compost tax cost function maximization

Algorithm	Optimal solution computed	Computing time in minutes
Minimax Solution Algorithm	18 700	59 8
Simulated Annealing Minimax Solution Algorithm	23600	162,9
Minimax Solution Algorithm with Bootstrap estimation of the error	25 200	181,4
Simulated Annealing Minimax Solution Algorithm with Bootstrap estimation of the error	34 900	263,4
Minimax Solution Algorithm with Jacknife estimation of the error	32 450	199,2
Simulated Annealing Minimax Solution Algorithm with Jacknife estimation of the error	34 573	108,4

4. Conclusions

Different alternative models were considered as possible improvements of Morita et al. (1989) methods Monte Carlo experiments were performed for evaluating their performance. They suggest that L1 has a better behavior for non normal distributions. This fact is a consequence of the properties of L1 norm optimization, because it behaves better with heavy tailed distributions than its L2 counterpart.. In the case of normality the results are not better.

Real life data were analyzed and the exhibited pattern was consistent with the Monte Carlo results.

We may conclude that:

1. The standard deviations are considerably smaller with the use of resampling methods
2. The analysis of the *F* based intervals leads to prefer the resampling methods because their amplitude are generally the largest.
3. Jacknife generally averages the smallest values of the mean optimal values and of the standard deviations.
4. Least Absolute Deviation procedure produces smaller standard deviations for non-normal related distributions.

Acknowledgements

A fellowship of the TWAS-to one of the authors supported the development of this research project. We are grateful to the referees for their comments which allowed to improve the initial version of this paper

BOUZA-HERRERA, C. N.; ALLENDE-ALONSO, S. M.; CHEN, D. C. Soluções aproximadas de programação linear estocástica com estimativa da função objetiva: Uma aplicação a métodos de re-amostragem. *Rev. Bras. Biom.*, São Paulo, v.26, n.1, p.115-130, 2008.

- RESUMO: Neste artigo avaliamos o efeito do uso de uma aproximação para a solução ótima maximal em programação linear estocástica. Ele pode ser calculado aceitando-se a normalidade ou usando o método de re-amostragem. "*Simulated Annealing*" utilizada para avaliar a solução de segundo estágio. "*Least Absolute Devation*" proposto como uma alternativa para o cálculo da solução inicial de mínimos quadrados. As metodologias propostas são comparadas usando experimentos simulados pelo método Monte Carlo.

- KEYWORDS: *Bootstrap*; *Jacknife*; norma L1; programação estocástica; impacto ambiental do composto sólido.

References

AARTS, E. H. L.; KORST, J. H. M. *Simulated Annealing and Boltzman machine*. New York: Wiley, 1989. p.91-120.

ARTHANARI T. S.; DODGE, Y. *Optimization methods in statistics*. New York: Wiley, 1993. p.67-83.

BIRGÉ, J. R.; LOUVEAUX, F. *Introduction to stochastic programming*. New York: Springer, 1997. p.49-165.

BOSS, D. D.; ZHANG, J. Monte Carlo evaluation of resampling based hypothesis tests. *Am. Stat.*, New Jersey, v.95, p.486-492, 2000.

BOUZA, C. N. Bounding the expected approximation error in stochastic linear programming with completely fixed recourses. In: KALL, P. *System modeling and optimization*, lecture notes on control. Berlin: Springer, 1992. P.541-545.

BROOKS, S.P. and MORGAN, B. J. T., Automatic starting point selection for function optimization . *Stat. Comp.*, London, v.4, p.173-177, 1994.

CODE, B. S.; RICHARDS, J. D. Permutation tests for LAD regression. *Biometrics*, Washington, v.52, p.88-902, 1996.

CODE, B. S.; RICHARDS, J. D. *User manual for BLOSSOM statistical software*. Fort Collins: Geological Survey, 1999. P.48-86.

CORANA, A.; MARCHESI, S.; MARITINI C.; RIDELLLA S. Minimizing multimodal function of continuous variables with the simulated annealing algorithm. *ACM, Trans. Math. Softw.*, Philadelphia, v.13, p.262-280, 1987.

DAVIDSON A. C.; HINKLEY. D. V. *Bootstrap methods and their applications*. Cambridge: Cambridge University Press, 1997. P.68-92.

DODGE, Y. An introduction to L1 norm based data analysis. *Comp. Stat. & Data Anal.*, Amsterdam, v.5, p.269-275, 1987.

DODGE, Y. LAD regression for detection of outliers in response and expected variables. *J. Multiv. Anal., Amsterdam*, v.61, p.144-158, 1997.

DUPAČOVÀ, J. Stability in stochastic programming with recourse. *Acta. Univ. Carol. Math. Et Phys*, Prague. v.24, p.23-24, 1983.

KALL, P.; WALLACE, S. W. *Stochastic programming*. Chichester: Wiley, 1994. p.83-122.

KANKOVÀ, V. On the stability in stochastic programming: generalized simple recourse problems. *Informatica*, Prague, v.1-2, p.78-80, 1994.

KIRKPATRICK, S.; GERLATT, D. D.; VECCHI, M. P. *Optimization by simulated annealing*. IBM Res. Report RC9355, 1982.

LINDO. *Optimization modeling with LINDO*. 5th edition. Duxbury: Duxbury Press, 1997.

METROPOLIS, N. et al. Equation of state calculation by fast computing machines. *J. Chem. Phys*, Woodburg, v.21, p.1087-1992, 1958.

MORITA, H.; ISHII, H.; NISHIDA, T. Stochastic programming with estimated objective. *Technol. Rep. Osaka Univ.*, Osaka, v.39, p.1-7, 1989.

PARKS, G. T. An intelligent simulated operations routine for nuclear fuel cycle design. *Nucl. Tech.*, Hinsdale, v.30, p.233-246, 1990.

PINCUS, M. A. A Monte Carlo method for approximating solutions of certain type of problems. *Oper. Res.*. Baltimore, v.18, p.1225-1228, 1970.

POWELL, M. J. D. *Non convex minimum calculation and the conjugate gradient method*. Berlin: Springer, 1984. p.122-141. (Lecture Notes in Mathematics 1066)

RÖMISCH, W.; SCHULTZ, R. Distribution sensitivity in stochastic programming. *Math. Prog*., New York, v.50, p.197-226, 1992.

SAS INSTITUTE. *SAS/STAT User's Guide*: version 8. North Caroline, 1999.

SCHILCK, T. Modified Choleski factorization for sparse preconditioners. *SIAM J. Sci. Stat. Comp*. Philadelphia, v.14, p.424-445, 1993.

VANDERBLIT, D., LOUIE, S. G. A Monte Carlo simulated annealing applied to operations over continuous variables. *J. Comp. Physiol*. Berlin, v.56, p.259-271, 1984.

Received in 12.07.2007.

Approved after revised in 18.03.2008.