Bull. Math. Soc. Sci. Math. Roumanie
Tome 49(97) No. 4, 2006, 335–343

A Sequential Quadratic Programming Technique with Two-Parameter Penalty Function

by
Cristinca Fulga

Abstract

In this paper the solution of nonlinear programming problems (NLP) by a two-parameter penalty function method is considered. An algorithm that combines penalty concepts and sequential quadratic programming techniques is presented. The approach taken is to replace the initial problem by the more tractable one of minimizing a non-differentiable penalty function chosen so that the solutions of the NLP are also solutions of the penalty function problem.

Key Words: Nonlinear Optimization, Lagrange Multipliers, Penalty Function.
2000 Mathematics Subject Classification: Primary 90C30, Secondary 90C26.

1 Preliminaries

In recent years, a large amount of work has been devoted to the problem of solving nonlinear programming problems. Important contributors in this field are Rosen [13], [14], Han [5], Mayne and Polak [8], Sahba [15], Preda [12] and many others. The reason of this interest is the fact that such problems arise in a wide variety of technical and scientific applications. There has been a resurgence of interest in exact penalty methods, because of their ability to handle degenerate problems and inconsistent constraint linearizations, see for example Chen and Goldfarb [1], Coope and Price [2], Gould et al. [4], Hu and Ralph [7], Mongeau and Sartenaer [9],Pantoja and Mayne [10].

Penalty methods have undergone three stages of development. They were first seen as vehicles for solving constrained optimization problems by means of unconstrained optimization techniques. This approach has not proved to be effective, except for special classes of applications. In the second stage, the penalty problem is replaced by a sequence of linearly constrained subproblems. These formulations, which are related to the sequential quadratic programming approach, are much more effective than the unconstrained approach but they leave open the question of how to choose the penalty parameter. In the most recent stage of development, penalty methods adjust the penalty parameter at every iteration so as to achieve a prescribed level of linear feasibility. The choice of the penalty parameter then ceases to be a heuristic and becomes an integral part of the step computation. In Section 2 a penalty strategy is presented and theorems the algorithm presented in Section 3 is based on. Initially, the algorithm follows the approach in Coope and Price [2]; the updating strategy for the parameters refers to a better way to handle the infeasibility in the context of sequential quadratic programming (SQP) methods.

The nonlinear programming problem considered is of the form:

$$\begin{cases} \min f(x) \\ a \le c(x) \le b \\ x \in R^n \end{cases} \tag{1}$$

where $f : R^n \to R$, $c : R^n \to R^m$, $c = (c_1, ..., c_m)$ are continuously differentiable functions, $a = (a_1, a_2, ..., a_m)^T \in R^m$, $b = (b_1, b_2, ..., b_m)^T \in R^m$ and $a \le b$ means $a_i \le b_i, \forall i = \overline{1, m}$.

Throughout this paper we assume that at each local minimizer of the nonlinear programming problem 1 an appropriate constraint qualification hold, thereby ensuring that any optimal point x^* of the nonlinear programming problem 1 satisfies the following Karush-Kuhn-Tucker conditions: there exists a vector of Lagrange multipliers $\lambda^* = (\lambda_1^*, \lambda_2^*) \in R^m \times R^m$, where $\lambda_1^* = (\lambda_{11}^*, \lambda_{12}^*, ..., \lambda_{1m}^*) \in R^m$, $\lambda_2^* = (\lambda_{21}^*, \lambda_{22}^*, ..., \lambda_{2m}^*) \in R^m$ such that

$$\begin{cases} c_i(x^*) - b_i \le 0 ; \quad \lambda_{1i}^* \ge 0 ; \quad \lambda_{1i}^* \cdot (c_i(x^*) - b_i) = 0 ; \quad i = \overline{1, m} \\ a_i - c_i(x^*) \le 0 ; \quad \lambda_{2i}^* \ge 0 ; \quad \lambda_{2i}^* \cdot (a_i - c_i(x^*)) = 0 ; \quad i = \overline{1, m} \\ \nabla f(x^*) + \sum_{i=1}^{m} \lambda_{1i}^* \cdot \nabla c_i(x^*) + \sum_{i=1}^{m} \lambda_{2i}^* \cdot (-\nabla c_i(x^*)) = 0. \end{cases} \tag{2}$$

2 The Penalty Function Problem

The nonlinear programming problem is not solved directly; instead a non-differentiable exact penalty function Φ is minimized, where the exact penalty function is constructed so that local minimizers of the nonlinear programming problem are also local minimizers of the penalty function Φ. The penalty function is

$$\Phi(x) = f(x) + \mu \cdot \theta(x) + \frac{1}{2}\nu \cdot \theta^2(x), \tag{3}$$

with $\mu > 0, \nu \ge 0$ and the degree of infeasibility, $\theta(x)$, is defined as

$$\theta(x) = \max_{1 \le i \le m} \left\{ \ [c_i(x) - b_i]_+ ; [a_i - c_i(x)]_+ \right\}, \tag{4}$$

where $[y]_+ = \max\{0; y\}$.

The penalty function Φ may be viewed as a hybrid of a quadratic penalty function based on the infinity norm and the single parameter exact penalty function of [8], [10] and [15]. Clearly θ is continuous $\forall x \in R^n$, but it is usually not differentiable for some x. However, the directional derivative

$$D_p\theta(x) = \lim_{\alpha \searrow 0} \frac{\theta(x + \alpha \cdot p) - \theta(x)}{\alpha}$$

exists for any $x, p \in R^n$. The definition 4 imply that, $\forall x, p \in R^n$,

$$D_p\theta(x) = \left\{ \begin{array}{c} \max\left\{ \max\limits_{i \in I_1(x)} p^T \nabla c_i(x) ; \max\limits_{i \in I_2(x)} \left(-p^T \nabla c_i(x)\right) \right\} \\ \text{if } \theta(x) > 0,\ I(x) \neq \varnothing \\ \max\left\{ \max\limits_{i \in I_1(x)} \left[p^T \nabla c_i(x)\right]_+ ; \max\limits_{i \in I_2(x)} \left[-p^T \nabla c_i(x)\right]_+ \right\} \\ \text{if } \theta(x) = 0,\ I(x) \neq \varnothing \\ 0 \text{ if } I(x) = \varnothing \end{array} \right. \tag{5}$$

where

$$\left\{ \begin{array}{l} I(x) = I_1(x) \cup I_2(x), \\ I_1(x) = \{i \,|c_i(x) - b_i = \theta(x)\}, \quad I_2(x) = \{i \,|a_i - c_i(x) = \theta(x)\}. \end{array} \right.$$

Definition 2.1. *For fixed values of $\mu > 0$ and $\nu \ge 0$, a point x^* is a critical point of Φ if and only if, for all $p \in R^n$, the directional derivative $D_p\Phi(x^*)$ is non-negative.*

Given a suitable choice of the penalty parameters, problem 1 may be replaced by the problem

$$\left\{ \begin{array}{l} \min \Phi(x) \\ x \in R^n. \end{array} \right. \tag{6}$$

Definition 2.2. *The solution set of the penalty function problem 6 with fixed values for $\mu > 0, \nu \ge 0$ is defined as the set of critical points of Φ.*

Theorem 2.3. *Let x^* be an optimal solution of the nonlinear programming problem 1 at which Karush-Kuhn-Tucker conditions 2 hold and let $\lambda^* = (\lambda_1^*, \lambda_2^*) \in R^m \times R^m$ be a vector of Lagrange multipliers satisfying these conditions for which $\|(\lambda_1^*, \lambda_2^*)\|_1$ is minimal. If $\mu > \|(\lambda_1^*, \lambda_2^*)\|_1$ then x^* is a critical point of Φ. Conversely, if x^* is both feasible and a critical point of Φ for some $\mu > 0, \nu \ge 0$, then x^* is a Karush-Kuhn-Tucker point of the nonlinear programming problem 1.*

Proof: The Karush-Kuhn-Tucker conditions 2 and the definition 4 imply $\theta(x^*) = 0$, $\lambda_{1i}^* = 0$, $\forall i \notin I_1(x^*)$ and $\lambda_{2i}^* = 0$, $\forall i \notin I_2(x^*)$. Therefore, combining 2 with 5 and $\|(\lambda_1^*, \lambda_2^*)\|_1 = \sum_{i=1}^{2} \sum_{j=1}^{m} |\lambda_{ij}^*|$, for any $p \in R^n$ we have

$$D_p\Phi(x^*) = p^T \nabla f(x^*) + \mu D_p \theta(x^*) =$$

$$= p^T \left[- \sum_{i \in I_1(x^*)} \lambda_{1i}^* \nabla c_i(x^*) + p^T \sum_{i \in I_2(x^*)} \lambda_{2i}^* \nabla c_i(x^*) \right] +$$

$$+ \mu D_p \theta(x^*) \geq$$

$$\geq \left[- \sum_{i \in I_1(x^*)} \lambda_{1i}^* - \sum_{i \in I_2(x^*)} \lambda_{2i}^* + \mu \right] D_p \theta(x^*) =$$

$$= \max \left\{ \max_{i \in I_1(x)} \left(p^T \nabla c_i(x)\right)_+ ; \max_{i \in I_2(x)} \left(-p^T \nabla c_i(x)\right)_+ \right\} \cdot$$

$$\cdot \left(\mu - \|(\lambda_1^*, \lambda_2^*)\|_1\right) \geq 0.$$

We obtained $D_p\Phi(x^*) \geq 0, \forall p \in R^n$ and thus, x^* is a critical point of Φ.

Conversely, if $x^*$is a critical point of Φ for some fixed $\mu > 0, \nu \geq 0$, then $D_p\Phi(x^*) \geq 0, \forall p \in R^n$. For any x sufficiently close to x^* we have

$$\Phi(x) = \Phi(x^*) + D_{x-x^*}\Phi(x^*) + o\|x - x^*\| \geq \Phi(x^*) + o\|x - x^*\|. \qquad (7)$$

If x is a feasible point, $\Phi(x) = f(x)$ and x^* satisfies Karush-Kuhn-Tucker conditions 2. □

The penalty problem 6 is solved by an iterative process. In order to determine a suitable descent direction at the k-th iterate, a continuous piecewise quadratic approximation to Φ near the current point is defined:

$$\psi^k(p) = f\left(x^k\right) + p^T \cdot \nabla f\left(x^k\right) + \frac{1}{2} p^T \cdot H^k \cdot p + \mu^k \cdot \zeta(p) + \frac{1}{2} \nu^k \cdot \zeta^2(p),$$

where

$$\zeta(p) \;\; = \max_{1 \leq i \leq m} \left\{0 \; ; \; c_i\left(x^k\right) - b_i + p^T \nabla c_i\left(x^k\right) \; ; \; a_i - c_i\left(x^k\right) - p^T \nabla c_i\left(x^k\right)\right\}$$

and H^k is a positive definite matrix. Clearly, ψ^k is strictly convex in p, and the level set $\left\{p \in R^n \left| \psi^k(p) \leq \psi^k(0) \right.\right\}$ is bounded for all $\mu > 0, \nu \geq 0$. Thus, ψ^k has an unique global minimizer p^k which also solves the quadratic programming problem:

$$\left(P^k\right)\begin{cases}\min\limits_{p,\zeta} p^T\nabla f\left(x^k\right)+\frac{1}{2}p^T\cdot H^k\cdot p\ +\mu^k\cdot\zeta+\frac{1}{2}\nu^k\cdot\zeta^2\\ a_i-\zeta\le c_i\left(x^k\right)+p^T\nabla c_i\left(x^k\right)\le b_i+\zeta,\qquad i=\overline{1,m}\\ \zeta\ge 0.\end{cases}$$

Theorem 2.4. *Let $\left(p^k,\zeta^k\right)$ be the unique solution of the quadratic programming problem $\left(P^k\right)$, with H^k positive definite matrix. Let $\left(\lambda_1^k,\lambda_2^k\right)$ denote an optimal Lagrange multiplier vector, which need not be unique, for which $\left\|\left(\lambda_1^k,\lambda_2^k\right)\right\|_1$ is least. If $p^k\neq 0$, $\zeta^k\le\theta\left(x^k\right)$ and $\mu+\nu\theta\left(x^k\right)\ge\left\|\left(\lambda_1^k,\lambda_2^k\right)\right\|_1$ then, p^k is a descent direction for Φ at x^k.*

Proof: The Karush-Kuhn-Tucker conditions for problem $\left(P^k\right)$ are

$$\begin{cases}c_i\left(x^k\right)-b_i+p^T\nabla c_i\left(x^k\right)-\zeta^k\le 0\ ;\quad \lambda_{1i}^k\ge 0\ ;\\ \lambda_{1i}^k\cdot\left(c_i\left(x^k\right)-b_i+p^T\nabla c_i\left(x^k\right)-\zeta^k\right)=0\ ;\quad i=\overline{1,m}\\ a_i-c_i\left(x^k\right)-p^T\nabla c_i\left(x^k\right)-\zeta^k\le 0\ ;\quad \lambda_{2i}^k\ \ge 0\ ;\\ \lambda_{2i}^k\cdot\left(a_i-c_i\left(x^k\right)-p^T\nabla c_i\left(x^k\right)-\zeta^k\right)\ =0\ ;\quad i=\overline{1,m}\\ \zeta^k\ge 0;\ \lambda_\zeta\le 0;\ \lambda_\zeta\zeta^k=0\end{cases}\tag{8}$$

and

$$\begin{cases}\nabla f\left(x^k\right)+H^k\cdot p^k+\sum\limits_{i=1}^m\lambda_{1i}^k\cdot\nabla c_i\left(x^k\right)-\sum\limits_{i=1}^m\lambda_{2i}^k\cdot\nabla c_i\left(x^k\right)=0\\ \mu^k+\nu^k\zeta^k-\sum\limits_{i=1}^m\lambda_{1i}^k-\sum\limits_{i=1}^m\lambda_{2i}^k+\lambda_\zeta=0.\end{cases}\tag{9}$$

Therefore, combining 8 and 9 we find

$$\begin{aligned}D_{p^k}\Phi\left(x^k\right) &= -\left(p^k\right)^T H^k\cdot p^k+\\ &+\sum_{i=1}^m\lambda_{1i}^k\left(c_i\left(x^k\right)-b_i-\zeta^k\right)+\sum_{i=1}^m\lambda_{2i}^k\left(a_i-c_i\left(x^k\right)-\zeta^k\right)+\\ &+\left(\mu^k+\nu^k\theta\left(x^k\right)\right)D_{p^k}\theta\left(x^k\right).\end{aligned}\tag{10}$$

Since ζ is convex on R^n, we have

$$\zeta\left(p^k\right)-\theta\left(x^k\right)=\zeta\left(p^k\right)-\zeta\left(0\right)\ge D_{p^k}\zeta\left(0\right)=D_{p^k}\theta\left(x^k\right).$$

Applying this result to 10 we obtain

$$\begin{aligned}D_{p^k}\Phi\left(x^k\right) &\le -\left(p^k\right)^T H^k\cdot p^k+\\ &+\left(-\sum_{i=1}^m\lambda_{1i}^k-\sum_{i=1}^m\lambda_{2i}^k+\mu^k+\nu^k\theta\left(x^k\right)\right)\left(\zeta^k-\theta\left(x^k\right)\right).\end{aligned}$$

Therefore, we have

$$D_{p^k}\Phi\left(x^k\right) \le -\left(p^k\right)^T H^k \cdot p^k + \left(\mu^k + \nu^k\theta\left(x^k\right) - \left\|\left(\lambda_1^k, \lambda_2^k\right)\right\|_1\right) \le 0$$

and thus, p^k is a descent direction of Φ in x^k. □

The following algorithm is based on the results of the preceding section.

3 Exact Penalty Function Algorithm

Step 1. Initialization. $k = 1$, $\mu^1 = 1$, $\nu^1 = 1$, $H^1 = I$, $\varepsilon = 10^{-5}$, $\rho = 0.02$, $\delta = 10^{-8}$, $\theta_{cross} = 1$, $\theta_{cap} = 10$, $k_1 = 1.5$, $k_2 = 2$, $k_3 = 1.2$, $k_4 = 5, \delta = 10^{-8}$.

Step 2. Update H and the penalty parameters. This step is omitted from the first iteration. The matrix H is updated using The Broyden-Fletcher-Goldfarb-Shanno update provided this maintains positive definiteness; otherwise H is not updated. The penalty parameters are updated as follows:

(i) If $\theta^k \le \theta_{cross}$ and $\mu^k < k_1\left\|\left(\lambda_1^k, \lambda_2^k\right)\right\|_1$, then $\mu^{k+1} = k_2\left\|\left(\lambda_1^k, \lambda_2^k\right)\right\|_1$ and $\nu^{k+1} = \nu^k$.

(ii) If $\theta^k > \theta_{cross}$and $\mu^k + \nu^k\theta^k < k_3\left\|\left(\lambda_1^k, \lambda_2^k\right)\right\|_1$, then $\mu^{k+1} = \mu^k$ and $\nu^{k+1} = \frac{k_4\left\|\left(\lambda_1^k, \lambda_2^k\right)\right\|_1 - \mu^k}{\theta^k}$.

Otherwise, the penalty parameters are not altered.

Step 3. Solve the (P^k) problem. If $\theta^k \le \theta_{cap}$, then solve $\left(P^k\right)$; the solution will be denoted by $\left(p^k, \zeta^k\right)$ and the algorithm proceeds to Step 4.

If $\theta^k > \theta_{cap}$, then the capping constraint $\zeta \le \theta^k$ is also imposed in $\left(P^k\right)$. Then this problem is solved and the solution is denoted by $\left(p^k, \zeta^k\right)$. If the capping constraint is not active at the $\left(P^k\right)$'s solution, then the algorithm proceeds directly to Step 4. Otherwise, the penalty parameters are updated as described in Step 2, except that $\left\|\left(\lambda_1^k, \lambda_2^k\right)\right\|_1$ is replaced by $\mu^k + \nu^k\theta^k + |\xi|$,where ξ is the Lagrange multiplier of the capping constraint. The $\left(P^k\right)$ problem is then solved again.

Step 4. Attempt the proposed step. If $(i)\, \Phi\left(x^k\right) - \Phi\left(x^k + p^k\right) \ge \rho\left[\Psi^k(0) - \Psi^k\left(p^k\right)\right]$

$(ii)\, \theta\left(x^k + p^k\right) \le \theta\left(x^k\right)$ are satisfied, then the proposed step is accepted and the algorithm proceeds to step 7. Otherwise, the execution continues at the next step.

Step 5. Calculate the Maratos effect correction vector. Solve the following quadratic problem for the second order correction t^k :

$$\begin{cases} \min\limits_{t \in R^n} \| t\|_2^2 \\ c_i\left(x^k + p^k\right) - b_i + t^T \nabla c_i\left(x^k\right) \geq 0 \\ a_i - c_i\left(x^k + p^k\right) - t^T \nabla c_i\left(x^k\right) \geq 0, \quad \forall i \in T \end{cases}$$

where T is the set of indices of the constraints active at the $\left(P^k\right)$'s solution in Step 3 and $\| t\|_2 = \left(\sum\limits_{i=1}^{n} |t_i|^2\right)^{1/2}$. If $\| t^k\|_2 \geq \|p^k\|_2$ then set $t^k = 0$.

Step 6. Arc search. Consider successive values of the sequence $1, \frac{1}{2}, \frac{1}{4}, \frac{1}{8}, ...$as trial values of α. The number of trial values for α is counted in N_α. If $t^k = 0$, then omit the first member of the sequence. Accept the first trial value which satisfies

(i) $\Phi\left(x^k\right) - \Phi\left(x^k + q^k\left(\alpha\right)\right) \geq \rho\alpha\left[\Psi^k\left(0\right) - \Psi^k\left(p^k\right)\right]$ where
$q^k\left(\alpha\right) = \alpha p^k + \alpha^2 t^k$
(ii) $\theta\left(x^k + q^k\left(\alpha\right)\right) \leq \theta^k$.

After a satisfactory value of α has been found, set $x^{k+1} = x^k + q^k\left(\alpha\right)$ and go to Step 7. If $N_\alpha > 20$ without finding a satisfactory value for α then, in order to get a feasible point, Rosen's method [13], [14] is employed; the new direction is $p^k = N\left(N^T N\right)^{-1} w$, where w is the vector whose components are the absolute values of the constraint functions for the violated constraints and N is the matrix of unit column vectors of the gradients of the violated constraints. Then, go to Step 5.

Step 7. Check the stopping conditions. The algorithm halts if either the length of the previous step $\left\|x^k - x^{k-1}\right\|_2 \leq \delta$ or both of the following conditions hold:

(i) $\theta^k < \varepsilon$

(ii) $\left\|\nabla f\left(x^k\right) + \sum\limits_{i \in A^k} \lambda_i^k \nabla c_i\left(x^k\right) - \sum\limits_{j \in B^k} \lambda_j^k \nabla c_j\left(x^k\right)\right\|_2 < \epsilon$ where

$$A^k = \left\{\, i \mid \left|c_i\left(x^k\right) - b_i\right| < \varepsilon \,\right\}, B^k = \left\{\, \mathrm{j} \mid \left|a_j - c_j\left(x^k\right)\right| < \varepsilon \,\right\}.$$

Otherwise, k is incremented, and the algorithm proceeds to Step 2.

The convergence properties of the algorithm are summarized in the following:

Theorem 3.1. *Assume that the sequence of iterates $\left\{x^k\right\}$ is bounded in norm, the sequence of matrices $\left\{H^k\right\}$ generated is bounded in norm and the penalty parameters μ, ν are altered only a finite number of times. Then, every cluster point of the sequence of iterates $\left\{x^k\right\}$ generated by the algorithm is a critical point of $\Phi\left(x;\mu,\nu\right)$ where μ, ν are at their final values.*

4 Concluding Remarks

The purpose of this paper is to show that there are some advantages to be gained from using a two-parameter exact penalty function based on the infinity norm of constraint violations. This function has an advantage over one-norm based exact penalty function in that only the gradients of the most violated constraints need be calculated in order to find a search direction: for one-norm exact penalty functions, the gradients of all active and violated constraints may be required. The use of a two-parameter penalty function has the additional advantage that, the quadratic subproblems are strictly convex; this enlarges the class of subroutines capable of solving them.

The algorithm generates convergent sequences under mild conditions; it is effective in practice and the use of the second penalty parameter significantly reduces the effort required to solve constraint nonlinear programs.

References

[1] L. Chen, D. Goldfarb, *Interior-point l_2 -penalty methods for nonlinear programming with strong global convergence properties*, Mathematical Programming, **108**(2006), 1-36.

[2] I. D. Coope , C. J. Price, *A Two-Parameter Exact Penalty Function for Nonlinear Programming*, Research Report number 96, Department of Mathematics and Statistics, University of Canterbury, New Zealand,, 1993.

[3] C. Fulga, *Penalty Concepts by Means of the Separation Functions in the Image Space*, Analele Universitatii "Dunarea de Jos", Galati, **1**, Fasc. II, Tome XVI (1998), 115-122.

[4] N. I. M. Gould , D. Orban ,P. Toint, *An interior point l_1-penalty method for nonlinear optimization*,Technical report RAL-TR-2003-022, Rutherford Appleton Laboratory Chilton, Oxfordshire, UK, 2003.

[5] S. P.Han, *A Globally Convergent Method for Nonlinear Programming*, Journal of Optimization Theory and Applications, **22**(1977), 297-309.

[6] S. P.Han, O. L.Mangasarian, *Exact Penalty Functions in Nonlinear Programming*, Mathematical Programming **17**(1979), 251-269.

[7] X. M. Hu, D. Ralph, *Convergence of a penalty method for mathematical programming with complementarity constraints*, Journal of optimization theory and applications, **123**(2004), 365-390.

[8] D. Q. Mayne, E. Polak, *A Superlinearly Convergent Algorithm for Constrained Optimization Problems*, Mathematical Programming Study, **16**(1982), 45-61.

[9] M. Mongeau, A. Sartenaer, *Automatic Decrease of the Penalty Parameter in Exact Penalty Function Methods*, European Journal of Operational Research, **83**(1995), 686-699.

[10] J. F. A. De O. Pantoja, D. Q. Mayne, *Exact Penalty Function Algorithm with Simple Updating of the Penalty Parameter*, Journal of Optimization Theory and Applications, **69**(1991), 441-467.

[11] T. Pietrzykowski, *An Exact Potential Method for Constrained Maxima*, SIAM Journal on Numerical Analysis **6**(1969), 299-304

[12] Preda V., *Optimality Conditions for a Class of Mathematical Programming Problems*, Bull. Soc. Sci. Math. Roumanie, **34** (1990), 343-350.

[13] J. B. Rosen, *The Gradient Projection Method for Nonlinear Programming. Part I. Linear constraints*, Journal of the Society for Industrial and Applied Mathematics, **8**(1960), 181-217.

[14] J. B. Rosen, *The Gradient Projection Method for Nonlinear Programming. Part II. Nonlinear constraints*, Journal Soc Indust Appl Math, **9**(1961), 514-532.

[15] M. Sahba, *A Globally Convergent Algorithm for Nonlinearly Constrained Optimization Problems*, Journal of Optimization Theory and Applications, **52**(1987), 291-307.

[16] W.I. Zangwill, *Nonlinear Programming via Penalty Functions*, Management Science **13**(1967), 344-358.

Received: 16.09.2006.

Revised: 2.10.2006.

Academy of Economic Studies,
Faculty of Cybernetics,
Statistics and Economic Informatics,
Department of Mathematics,
Piaţa Romană, 6
010174 Bucharest
Romania
E-mail: `fulga@csie.ase.ro`