Journal of Global Optimization **16:** 343–354, 2000.

A Boolean Penalty Method for Zero-One Nonlinear Programming

DAVID MAURICIO[1] and NELSON MACULAN[2]
[1]*State University of Norte Fluminense (UENF), Laboratory of Science and Engineering – CCT, Av. Alberto Lamego, 2000 Campos, RJ 28015-620, Brazil (e-mail: mauricio@uenf.br);* [2]*Systems Engineering and Computer Science, COPPE – Federal University of Rio de Janeiro, P.O. Box 68511, Rio de Janeiro, RJ 21945-970, Brazil (e-mail: maculan@cos.ufrj.br)*

(Received 12 November 1996; accepted in revised form 29 December 1999)

Abstract. We introduce a discrete penalty called Boolean Penalty to 0–1 constrained nonlinear programming (PNLC-01). The main importance of this Penalty function are its properties which allow us to develop algorithms for the PNLC-01 problem. Optimality conditions, and numerical results are presented.

Key words: 0–1 nonlinear programming; Combinatorial optimization; Discrete penalty

1. Introduction

We introduce a Boolean Penalty to solve the following class of 0–1 constrained nonlinear programming problems:

$$\text{(PNLC-01):}\quad f^* = \text{Minimize } f(x)$$
$$\text{s.t.: } F_i(x) \leqslant 0 \quad i = 1, \dots, m$$
$$x \in B'' = \{0, 1\}^n ,$$

where $f,\ F_i : R^n \to R,\ i = 1, 2, \dots, m$, are convex functions on R^n, with Lipschitz constants L_f and L_i respectively.

Several efforts to solve PNLC-01 have been made in the last 30 years. Approximate methods are known, see [2, 3, 7, 14]. Few exact methods are capable of solving this problem with $n \geqslant 32$, due to the nonlinearity of the objective function and/or constraints, correlation data, non-monotonicity of objective function and/or constraints, and problem dimensions, see [1, 4–6, 9, 11, 17]. Reviews of the PNLC-01 can be found in [2, 8, 10, 14].

This paper is organized as follows. In the next section we introduce a Boolean Penalty. We show that this function is nonincreasing monotone and its small root is f^*. In Section 3, two algorithms are presented. Given lower and upper bounds $\underline{t}, \bar{t}$ for the optimal value, Algorithm 3.2 computes an ε-optimal solution (x^ε is an ε-optimal solution to PNLC-01 if and only if $f(x^\varepsilon) \leqslant f^* + \varepsilon,\ F_i(x^\varepsilon) \leqslant \varepsilon\ \forall i$) for $\varepsilon > 0$, in a number of iterations not greater than $\log_2((\bar{t} - \underline{t})/\varepsilon) + 1$. If f and F_i are polynomial functions, then it is sufficient to consider $\varepsilon = 1$ guaranteeing the algorithm finds an optimal solution. The main difficulty with the Boolean Penalty is its evaluation, which corresponds to solve an unconstrained 0–1 nonlinear problem. Numerical experiments are presented in Section 4. The conclusion follows in Section 5.

2. A Boolean Penalty function

2.1. DEFINITIONS

Associated with the relaxed problem of PNLC-01, i.e. the following problem:

$$(\overline{\text{PNLR-01}}):\quad \bar{f} = \text{Min}\{f(x) : F_i(x) \leqslant 0\,,\quad i = 1, \ldots, m,\ x \in [0, 1]^n\}$$

we have the following functions defined for $t \in R$ and $x \in [0, 1]^n$

$$H(x; t) := \text{Max}\{f(x) - t, F_1(x), \ldots, F_m(x)\}$$
$$h(t) := \text{Min}\{H(x; t) : x \in [0, 1]^n\}$$

The following result shown in [12] relates some properties of H and h.

LEMMA 1. *For any fixed* $t \in R$, $H(x; t)$ *is convex on* R^n, *with Lipschitz constant* $L_H = \text{Max}\{L_f, L_1, L_2, \ldots, L_m\}$. *For any fixed* $x \in [0, 1]^n$, $H(x; \cdot)$ *and* h *are nonincreasing, convex on* R^n *with Lipschitz constant 1. Moreover,* $h(f^*) = 0$ *and* $h(t) > 0$ $\forall t < f^*$.

DEFINITION 2.1. Associated with PNLC-01, we call by Boolean Penalty the following function defined for any $t \in R$:

$$h^r(t) := \text{Min}\{H(x; t) : x \in B^n\}\,.$$

Since $H(\cdot; t)$ is a continuous function on $[0, 1]^n$ for any $t \in R$ and B^n is finite, then h^r is well defined.

Given $t \in R$, we denote an argument that defines h^r by $x(t)$, i.e. $h^r(t) = H(x(t); t)$. Also we denote an optimal solution to PNLC-01 by x^*.

The following theorem establishes some properties of h^r.

THEOREM 1. h^r *is a continuous, real, monotone nonincreasing function with unitary Lipschitz constant. Moreover,* $h^r(f^*) = 0$, $h^r(t) > 0$ $\forall t < f^*$.

Proof.

(i) Let $a \in R$, then since H is a continuous function, we have:

$$\lim_{t \to a} h^r(t) = \lim_{t \to a} \{\min_{x \in B^n} H(x; t)\} = \min_{x \in B^n} (\lim_{t \to a} H(x; t))$$

(ii) Take $t_1, t_2 \in R$ such that $t_1 \leqslant t_2$ and any $x \in R^n$, then from the monotonicity of $H(x; \cdot)$ (Lemma 1), we have $H(x; t_1) \geqslant H(x; t_2)$. Evidently this relation also is true if we minimize on B^n, i.e., $h^r(t_1) \geqslant h^r(t_2)$. Thus, by the Lipschitz property of $H(x(t_2); \cdot)$ we have:

$$|h^r(t_1) - h^r(t_2)| = h^r(t_1) - h^r(t_2) \leqslant H(x(t_2); t_1) - H(x(t_2); t_2) \leqslant |t_1 - t_2|$$

(iii) Let x^* be an optimal solution to PNLC-01, that is, $f(x^*) = f^*$ and $F_i(x^*) \leqslant 0$ $\forall i$, then

$$h^r(f^*) \leqslant \mathrm{Max}\{f(x^*) - f^*, F_1(x^*), \ldots, F_m(x^*)\} = 0$$

where $h^r(f^*) = 0$. Assume the contrary, there exists $x(f^*) \in B^n$ such that $h^r(f^*) = H(x(f^*); f^*) < 0$, i.e., $f(x(f^*)) < f^*$ and $F_i(x(f^*)) < 0$ $\forall i$, which is a contradiction.

(iv) If $h^r(t) > 0$ then, $0 < H(x^*; t) = \mathrm{Max}\{f(x^*) - t, F_1(x^*), \ldots, F_m(x^*)\} = f^* - t$, because $F_i(x^*) \leqslant 0$ $\forall i$. Therefore $t < f^*$. Now, if $t < f^*$ then, from (iii) and the monotonicity of h^r (Lemma 1) we have $h^r(t) \geqslant h^r(f^*) = 0$, where $h^r(t) > 0$ because otherwise we have $f(x(t)) < f^*$ and $F(x_i(t)) \leqslant 0$ $\forall i$, which is a contradiction.

The following result establishes (from the properties of h^r) a lower and an upper bounds for f^*.

COROLLARY 1.

If $h^r(t) > 0$, *then* $\mathrm{Max}\{f(x(t)), t + h^r(t)\} \leqslant f^*$.

If $h^r(t) \leqslant 0$, *then* $f^* \leqslant f(x(T)) \leqslant t$.

Proof.

(i) If $h^r(t) > 0$ then $0 < H(x(t); t) \leqslant H(x^*; t)$, i.e.,

$$\begin{aligned} 0 &< \mathrm{Max}\{f(x(t)) - t, F_1(x(t)), \ldots, F_m(x(t))\} \\ &\leqslant \mathrm{Max}\{f(x^*) - t, F_1(x^*), \ldots, F_m(x^*)\} = f^* - t \end{aligned}$$

where it is easy to verify that

$$\mathrm{Max}\{t, f(x(t))\} \leqslant f^* . \tag{1}$$

Also, since $t \leqslant f^*$ then, from the Lipschitz and the monotone nonincreasing properties of f (Theorem 1), we have,

$$|h^r(t) - h^r(f^*)| = h^r(t) - h^r(f^*) \leqslant |t - f^*| = f^* - t .$$

Since $h^r(f^*) = 0$ (see Theorem 1), then

$$h^r(t) + t \leqslant f^* \tag{2}$$

Therefore, from (1) and (2) follows the result.

(ii) If $h^r(t) \leqslant 0$ then $f(x(t)) \leqslant t$ and $F_i(x(t)) \leqslant 0$, $i = 1, 2, \ldots, m$, where we have $f^* \leqslant f(x(t)) \leqslant t$ because $x(t)$ is a feasible solution.

The following result is a direct consequence of the definition of h^r and the above corollary.

COROLLARY 2. *If* $h^r(t) > 0$ *and* $x(t)$ *is a feasible solution*, *then* $x(t)$ *is an optimal solution*. *If* $h^r(t) \leqslant 0$, *then* $x(t)$ *is a feasible solution*.

2.2. THE EQUIVALENT PROBLEM

Theorem 1 suggests the following equivalent problem to determine an optimal value of PNLC-01 problem.

(PH-01) Find $t \in R$ such that

$$h^r(t) = 0 \tag{3}$$

$$h^r(t - \delta) > 0 \quad \forall \delta > 0\,. \tag{4}$$

Evidently, from Theorem 1, the above problem corresponds to determining the smallest root of the monotone and nonincreasing function h^r. Below we demonstrate that the unique solution of PH-01 is the optimal value of PNLC-01 problem.

THEOREM 2. *The unique solution of PH-01 problem is* f^*.

Proof. From Theorem 1, we know that f^* is a solution of PH-01. Now, we suppose that a solution $t^* \neq f^*$ of PH-01 problem exists. Then, from Theorem 1 and (3) it follows that $t^* > f^*$. Let $\delta' = t^* - f^* > 0$ then by (4), we have: $h^r(t^* - \delta') > 0$. Then $h^r(t^* - \delta') + h^r(f^*) = 0$ which is a contradiction. Hence $t^* = f^*$ must hold.

The following result says that in general h^r has a unique root.

THEOREM 3. *If there is an optimal solution* x^* *PNLC-01 such that* $F_i(x^*) <)$ *for all* $i = 1, 2, \ldots, m$, *the* h^r *has a unique root*.

Proof. Take any $t \in R$ such that $h^r(t) = 0$, then from the hypotheses of this Theorem

$$0 = h^r(t) \leqslant H(x^*, t) = \mathrm{Max}\{f^* - t, F_1(x^*), \ldots, F_m(x^*)\} = f^* - t\,,$$

where $t \leqslant f^*$.

On the other hand, since $h^r(t) = 0$, then from Theorem 1, we have $t \geqslant f^*$. Thus, from the relation above we have $t = f^*$.

The following lemma, shown in [15], establishes that it is not necessary to compute an exact solution of PH-01 to find an optimal solution of PNLC-01.

LEMMA 2. *There exits a* $\delta > 0$ *such that for any* $t \in (f^* - \delta, f^* + \delta)$ *it is verified that* $x(t)$ *is an optimal solution to PNLC-01.*

It was observed in [15] that if f and F_i, for all $i = 1, \ldots, m$, are polynomial functions with integer coefficients, it is sufficient to consider $\delta = 1$ to guarantee an optimal solution.

2.3. OPTIMALITY CONDITIONS

In order to verify optimality conditions, we consider the following results.

THEOREM 4. *Given $\underline{t}, \bar{t}$ $\varepsilon \in R$ such that $\underline{t} \leqslant f^* \leqslant \bar{t}$ and $\varepsilon \geqslant 0$.*

(i) If $h^r(\underline{t}) \leqslant \varepsilon$, then $x(\underline{t})$ is an ε-optimal solution.

(ii) If $\bar{t} - \underline{t} \leqslant \varepsilon$, then $x(\underline{t})$, $x(\bar{t})$ are ε-optimal solutions.

($\bar{x}$ is an ε-optimal solution of PNLC-01 if and only if $f(\bar{x}) \leqslant f^* + \varepsilon$, $F_i(\bar{x}) \leqslant \varepsilon$ $\forall i$).

Proof.

(i) If $h^r(\underline{t}) \leqslant \varepsilon$, then $F_i(x(\underline{t})) \leqslant \varepsilon$ $\forall i$ and $f(x(\underline{t})) \leqslant t + \varepsilon$. Since by hypothesis $\underline{t} \leqslant f^*$, then $f(x(\underline{t})) \leqslant f^* + \varepsilon$.

(ii) Since $f^* \leqslant \bar{t}$, then from Theorem 1, we have $h^r(\bar{t}) \leqslant 0$, where, from Corollary 2, we know $x(\bar{t})$ is a feasible solution and $f(x(\bar{t})) \leqslant \bar{t}$. Thus we have $f(x(\bar{t})) \leqslant \underline{t} + \varepsilon$, i.e. $x(\bar{t})$ is an ε-optimal solution. Also, by the Lipschitz property of h^r (Theorem 1) we have $h^r(\underline{t}) - h(f^*) \leqslant f^* - \underline{t}$, where from the hypotheses and Theorem 1, it follows that $h^r(\underline{t}) \leqslant f^* - \underline{t} \leqslant \varepsilon$, i.e. $f(x(\underline{t})) \leqslant \underline{t} + \varepsilon \leqslant f^* + \varepsilon$ and $F_i(x(\underline{t})) \leqslant \varepsilon$ $\forall i$.

The following optimality condition we follow from Theorem 1 and Corollary 2.

COROLLARY 3. *Given $\underline{t}, \bar{t} \in R$ such that $\underline{t} \leqslant f^* \leqslant \bar{t}$. If $\mathrm{Max}\{fx(t)), \underline{t}\} = f(x(\bar{t}))$ then $x(\bar{t})$ is an optimal solution.*

3. Boolean Penalty algorithms

3.1. ALGORITHMS

We present two Algorithms to compute the smallest root of h^r, i.e. to solve PNLC-01.

ALGORITHM 3.1

(0) Choose: $\varepsilon \geqslant 0$ (desired precision), $t \leqslant f^*$

(1) Compute $x(t) \in \mathrm{Arg\,min}\{H(x; t) : x \in B^n\}$;
Set $h^r(t) := H(x(t); t)$;

(2) Stop Test
If $h^r(t) \leqslant \varepsilon$, then STOP, $x(t)$ is an ε-optimal solution

(3) Update Parameter
Set $t := t + h^r(t)$;
Go to (1).

REMARK 3.1.

(1) An initial value to t is the optimal value $\bar{f}$ of the relaxed problem $\overline{\text{PNLC-01}}$.
(2) Computing $x(t)$ in Step 1 corresponds to solving an unconstrained 0–1 nonlinear programming problem. This problem can be solved, for example, through enumeration methods (see [5]), bundle methods (see [16, 17]) or outer approximation techniques (see [4]).
(3) Actually at each iteration the value of parameter t is growing and is always a lower bound of f^* (see Corollary 1).
(4) The update rule for parameter t, proposed in Step 3 is analogous to the one proposed in [13] (continuous nondifferentiable optimization).

The second proposed Algorithm uses the bisection search and some properties of h^r function to accelerate this search.

ALGORITHM 3.2.

(0) Choose $\varepsilon > 0$ (desired precision), $\underline{t}, \bar{t} \in R$ such that $\underline{t} \leqslant f^* \leqslant \bar{t}$, set $t := \underline{t}$
(1) Compute $x(t) \in \arg\min\{H(x; t) : x \in B^n\}$
Set $h^r(t) := H(x(t); t)$
(2) Stop Test
If $((\underline{t} - \bar{t}) \leqslant \varepsilon$ or $h^r(t) \leqslant \varepsilon)$ then, STOP, $x(t)$ is an ε-optimal solution.
(3) Update parameters
3.1 Update by the best bound
If $h^r(t) \leqslant 0$ then, $\bar{t} := f(x(t))$
else $\underline{t} := \text{Max}\{t + h^r(t), f(x(t))\}$;
3.2 Update by bisection
$t := 0.5(\underline{t} + \bar{t})$
(4) Go to (1).

REMARK 3.2.

(1) An initial value to $\underline{t}$ is the optimal value $\bar{f}$ of the relaxed problem $\overline{\text{PNLC-01}}$. When the Lipschitz constant L_f is estimated, an initial value for $\bar{t}$ parameter is given by $\bar{f} + L_f\sqrt{n}$.
(2) A practical procedure to determine an initial value to $\bar{t}$ is given by:

Choose $\Delta > 0$ (relatively large), $\underline{t} < f^*$, set $t := \underline{t} + \Delta$

While $(h^r(t) > 0)$ do $t := t + \Delta$; set $\bar{t} := t$

(3) Computing $x(t)$ in Step 1 corresponds to solve an unconstrained 0–1 nonlinear programming problem. This problem can be solved for example through enumeration methods (see [5]), bundle methods (see [16, 17]) or outer approximation techniques (see [4]).

3.2. CONVERGENCE ANALYSIS

Next we study the convergence of the proposed Algorithms.

THEOREM 5. *Let $\varepsilon = 0$, the Algorithm 3.1 converges to an optimal solution of (PNLC-01) problem in a finite number of steps.*

Proof. We denote by t_k the value of parameter t in the kth iteration ($k \geqslant 0$). The initial value of t we denote by t_0.

Suppose that the Algorithm does not converge, i.e., $\exists \delta > 0$ such that: $h'(t_i) > \delta$ $\forall i \geqslant 0$. Thus from Step 3, we have: $t_1 = t_0 + h^r(t_0) > t_0 + \delta$, $t_2 = t_1 + h^r(t_0) > t_1 + \delta > t_0 + 2\delta$, and, in general, $t_k > t_0 + k\delta$ $\forall k \geqslant 0$. Since f is continuous on R and B^n is bounded, then there exists a $k \in N$ such that $f^* < t_0 + k\delta$, where from the last relation we have: $f^* < t_k$. This is a contradiction because $t_i \leqslant f^*$ $\forall i \geqslant 0$ (see Remark 3.1). Therefore, Algorithm 3.1 converges.

Now, we prove that the Algorithm converges in a finite number of steps to an optimal solution.

$$\text{Let } \delta = \text{Min}\{\text{Max}\{F_1(x), F_2(x), \ldots, F_m(x)\} : x \in B^n, F_i(x) > 0 \text{ for some } i\} \tag{5}$$

Actually, $\delta > 0$ unless the domain of the studied problem is B^n. Since the Algorithm converges, there is $k \in N$ finite such that $|t_{i+1} - t_i| = t_{i+1} - t_i \leqslant \delta$ $\forall i \geqslant k$, where from Step 3 of the Algorithm, it follows, in particular, that $0 \leqslant h^r(t_k) \leqslant \delta$. Evidently, if $h^r(t_k) = 0$ then the Algorithm stops with an optimal solution. Thus, we suppose that $0 < h^r(t_k) \leqslant \delta$, then from (5) and Corollary 2 we have that $x(t_k)$ is an optimal solution, therefore

$$h^r(t_k) = \text{Max}\{f(x(t_k)) - t_k, F_1(x(t_k)), \ldots, F_m(x(t_k))\} = f^* - t_k$$

thus,

$$h^r(t_{k+1}) = h^r(t_k + h^r(t_k)) = h^r(t_k + f^* - t_k) = h^r(f^*) = 0$$

where from Step 2, the Algorithm converges to an optimal solution in a finite number of steps.

The following result estimates the efficiency of Algorithm 3.1.

THEOREM 6. *Algorithm 3.1 converges to an optimal solution* ($\varepsilon > 0$) *in less than* $(f^* - t_0)/\varepsilon$ *iterations* (*where* t_0 *is the initial value of parameter* t).

Proof. We denote by t_i the value of parameter t at the ith iteration ($i \geqslant 0$). From the above Theorem, there exists a $k \in N$ such that

$$h^r(t_i) > \varepsilon \quad \forall i < k \tag{6}$$

$$h^r(t_k) \leqslant \varepsilon \tag{7}$$

where from Theorem 1, it follows that

$$\varepsilon < h^r(t_{k-1}) = h^r(t_{k-1}) - h^r(f^*) \leqslant f^* - t_{k-1} . \tag{8}$$

On the other hand, from Step 3 of the Algorithm and (6), it follows that

$$t_i > t_0 + i\varepsilon \quad \forall i < k .$$

Thus, from (8) we have $\varepsilon < f^* - t_0 - (k-1)\varepsilon$, *i.e.*, $k < (f^* - t_0)/\varepsilon$.

The following Theorem establishes the convergence of Algorithm 3.2.

THEOREM 7. *Given* $\varepsilon = 0$, *Algorithm 3.2 converges to an optimal solution in a finite number of steps*.

Proof. We denote by $\bar{t}_i$ and $\underline{t}_i$ the corresponding values of parameters $\bar{t}$ and $\underline{t}$ in the ith iteration, and by $\bar{t}_0$ and $\underline{t}_0$ the corresponding initial values of $\bar{t}$ and $\underline{t}$.

From the parameter update rule (Step 3), we have that $\bar{t}_{i+1} - \underline{t}_{i+1} \leqslant (\bar{t}_i - \underline{t}_i)/2$ $\forall i > 0$, where

$$\bar{t}_i - \underline{t}_i \leqslant (\bar{t}_0 - \underline{t}_0)/2^i \quad \forall i > 0 .$$

On the other hand, from Lemma 2, there exists $\delta > 0$ such that, for any $t \in (f^* - \delta, f^* + \delta)$ we have that $x(t)$ is an optimal solution. Thus, from the relation above, it is easy to see that there is a $k \in N$ such that

$$\bar{t}_k - \underline{t}_k \leqslant (\bar{t}_0 - \underline{t}_0)/2^k < \delta . \tag{9}$$

Moreover, from Step 3 of the Algorithm, Corollary 1 and Remark 3.2, we have that $\underline{t}_i \leqslant f^* \leqslant \bar{t}_i$ $\forall i \geqslant 0$. Thus, from relation (9), it follows that: $[\underline{t}_k, \bar{t}_k] \subset (f^* - \delta, f^* + \delta)$, i.e., Algorithm 3.2 converges in a finite number steps, and $x(\underline{t}_k)$ and $x(\bar{t}_k)$ are optimal solutions.

THEOREM 8. *Given* $\varepsilon > 0$, *Algorithm 3.2 converges to an* ε*-optimal solution of PNLC-01 in a number of iterations not greater than* $\log_2((\bar{t}_0 - \underline{t}_0)/\varepsilon) + 1$.

Proof. From Step 3 of Algorithm 3.2, we have that $\bar{t}_i - \underline{t}_i \leqslant (\bar{t}_0 - \underline{t}_0)/2^i$ $\forall i > 0$, where it is easy to see that there exists a $k \in N$ such that $\bar{t}_k - \underline{t}_k \leqslant (\bar{t}_0 - \underline{t}_0)/2^k \leqslant \varepsilon$ or equivalently by $k \geqslant \log_2((\bar{t}_0 - \underline{t}_0/\varepsilon)$. Thus, from the stopping rule (Step 2), the Algorithm finishes in a number of iterations not greater than $\log_2((\bar{t}_0 - \underline{t}_0)/\varepsilon) + 1$.

On the other hand, from the parameter update rule (Step 3) and Corollary 1, it follows that: $\underline{t}_i \leqslant f^* \leqslant \bar{t}_i$ $\forall i \geqslant 0$. Thus, from the optimality condition (Theorem 4), we know that $x(\bar{t}_k)$ and $x(\underline{t}_k)$ are optimal solutions to PNLC-01.

4. Numerical experiments

Algorithm 3.2 was implemented in FORTRAN F77L3. We used the cutting plane Algorithm proposed in [16] to solve the 0–1 min–max nonlinear problem (Step 2 in the Algorithm). All numerical experiments were executed on a 100-MHz Intel 486 DX4 processor with 8 Mbytes of memory.

4.1. TEST PROBLEMS

We indicated by O.F., C.F. and O.S. the objective function, the constraint and the optimal solution respectively. In order to evaluate the number of iterations of Algorithm 3.2, we considered the following test problems.

- Problem 1:

O.F.: $\sum_{i=1}^{n} x_i^2 - 1.8 \sum_{i=1}^{n} x_i + 0.81n$

C.F.: $\sum_{i=1}^{n} x_i - n + 1 \leqslant 0$

O.S.: $x_i = 0 \quad \text{for some} \quad j \in \{1, 2, \ldots, n\}$

$x_i = 1 \quad \forall i \in \{1, 2, \ldots, n\} - \{j\}$

- Problem 2:

O.F.: $\text{Sen}\left(\pi + (\pi/n) \sum_{i=1}^{n} x_i\right)$

C.F.: $\sum_{i=1}^{n} x_i - n/2 + 1 \leqslant 0$

O.S.: Any point $x \in B^n$ such that $\sum_{i=1}^{n} x_i = n/2 - 1$

- Problem 3:

O.F.: $\sum_{i=1}^{n} e^{x_i + x_{i+1}}$

C.F.: $1 - \sum_{i=1}^{n} x_i \leqslant 0$

O.S.: $x_i = 1 \quad \text{for some} \quad j \in \{1, 2, \ldots, n\}$

$x_i = 0 \quad \forall i \in \{1, 2, \ldots, n\} - \{j\}$

- Problem 4:

O.F.: $\text{Sen}\left(\pi + (\pi/(2n)) \sum_{i=1}^{n} x_i\right) + \text{Cos}\left(\pi + (\pi/2)) \sum_{i=1}^{n} x_i\right)$

C.F.: $\sum_{i=1}^{n} x_i - n/2 + 1 \leqslant 0$

O.S.: Any point $x \in B^n$ such that $\sum_{i=1}^{n} x_i = n/2 - 1$

- Problem 5:

$$\text{O.F.} \quad \sum_{i=1}^{n/2} (0.1)^{x_i} + \sum_{i=n/2+1}^{n} (1.04)^{x_i}$$

$$\text{C.F.:} \quad ne^2/2 + 2e - \sum_{i=1}^{n-1} e^{x_i + x_{i+1}} \leqslant 0$$

$$\text{O.S.} \quad x_i = 1 \quad \forall j \in \{1, 2, \ldots, k\},$$

$$\text{where} \quad k = \min\{\lceil ((n/2+1)e^2 + e - n + 1)/(e^2 - 1)\rceil, n\}$$

($\lceil a \rceil$: denote the least integer, greater than a).

4.2. NUMERICAL RESULTS

In Table 1 we can observe our numerical results for a sample of 10 problems (see another numerical experiments in [15]). For each problem we consider $n \in \{4, 8, 16\}$ and (desired precision), i.e., optimal solution. Also, for each problem we consider an increasing variation of the gap for the optimal value, proportional to 1, 2 and 4.

The numerical experiments presented in Table 1 show that an exponential increase of the gap corresponds to a linear increase of the number of iterations of the Algorithm. For all the test problems considered the Algorithm converges to an optimal solution in few iterations, we would conclude that the lower and upper bounds provided for the Boolean Penalty (see Corollary 1) were good.

5. Conclusion

We have introduced a Boolean Penalty, and presented an exact and robust Algorithm to solve constrained 0–1 nonlinear programming problems. The main contribution of the Boolean Penalty Algorithm is the fact that it enables us to solve PNLC-01 by computing the minimum root of a continuous, real and nonincreasing monotone function (equivalent problem), called Boolean Penalty.

Theoretical results on the efficiency of the Algorithms of Boolean Penalty show that these Algorithms converge to an ε-optimal solution in few iterations, as it is verified in the numerical experiments (see Table 1). The complexity of each iteration of Algorithms is equal to the complexity of solving an unconstrained 0–1 nonlinear programming problem, and it is well-known that this class of problem is NP-hard. The numerical experiments presented in Table 1, also show that an exponential increase of the gap corresponds to a linear increase of the number of iterations of Algorithm 3.2. When the objective function is not convex, it is always possible to convert it to a form like PNLC-01, adding a penalty term (see, for example, [17]).

Acknowledgement

The authors wish to thank José Koiller for his helpful advice and suggestions. This work was partially supported by CNPq and FAPERJ.

Table 1. Numerical results for the Boolean Penalty Algorithm (Algorithm 3.2)

Prob	n	LB	UB	Gap	Iter	f^*
1	4	0.0	3.5	3.5	3	0.84
		−1.75	5.25	7.0	4	
		−5.25	8.75	14.0	4	
	8	0.0	7.0	7.0	3	0.88
		−3.5	10.5	14.0	4	
		−10.5	17.5	28.0	5	
	16	0.0	14.0	14.0	3	0.96
		−7.0	21.0	28.0	5	
		−21.0	35.0	56.0	5	
2	4	−1	1	2	3	−0.70711
		−2	2	4	4	
		−4	4	8	4	
	8	−1	1	2	3	−0.92388
		−2	2	4	4	
		−4	4	8	5	
	16	−1	1	2	3	−0.98079
		−2	2	4	5	
		−4	4	8	5	
3	4	3.0	10.0	7.0	4	4.71828
		−0.5	13.5	14.0	4	
		−7.5	20.5	28.0	4	
	8	7.0	20.0	13.0	4	
		0.5	26.5	26.0	4	
		−12.5	39.5	52.0	5	
	16	15.0	40.0	25.0	5	16.71828
		2.5	52.5	50.0	5	
		−22.5	77.5	100.0	6	
4	4	−2.0	2.0	4.0	3	−1.30657
		−4.0	4.0	8.0	4	
		−8.0	8.0	16.0	4	
	8	−2.0	2.0	4.0	4	−1.38704
		−4.0	4.0	8.0	5	
		−8.0	8.0	16.0	5	
	16	−2.0	2.0	4.0	4	−1.4074
		−4.0	4.0	8.0	5	
		−8.0	8.0	16.0	5	
5	4	0.0	4.0	4.0	2	2.28
		−2.0	6.0	8.0	3	
		−4.0	12.0	16.0	3	
	8	0.0	8.0	8.0	3	4.48
		−4.0	12.0	16.0	3	
		−8.0	24.0	32.0	4	
	16	0.0	16.0	16.0	4	8.84
		−8.0	24.0	32.0	4	
		−16.0	48.0	64.0	4	

Prob, ≠ test problem; n, number of variables; LB, lower bound for optimal value; UB, upper bound for optimal value; f^*, optimal value; Iter, number of iteration.

References

1. Balas, E. and Mazzola, J. (1984), Nonlinear 0-1 programming: I. Linearization Techniques, *Mathematical Programming* 30: 1–21.
2. Cha, J., *Mixed Discrete Constrained Nonlinear Programming*, Ph.D. Thesis, University of New York at Buffalo, 1987.
3. Cha, J.Z. and Mayne, R.W. (1987), *Mixed Discrete Constrained Nonlinear Programming via Recursive Quadratic Formula with Rank-One Update Formula*. Post-Doctoral Fellow, National Center Earthquake Engineering Research, Buffalo, New York.
4. Fletcher, R. and Leyffer, S. (1994), Solving mixed integer nonlinear programs by outer approximation, *Mathematical Programming* 66: 327–349.
5. Garfinkel, R.S. and Nemhauser, G.L. (1972), *Integer Programming*, John Wiley & Sons, New York, NY.
6. Glover, F. and Woosley, E. (1974), Converting the 0-1 polynomial programming problems to a 0-1 linear program, *Operation Research* 22, 180–182.
7. Gupta, O. (1980), *Branch and Bound Experiment in Nonlinear Integer Programming*, Ph.D. Thesis, School of Industrial Engineering, Purdue University, West Lafayette, IN.
8. Gupta, O. and Ravindran, A., Nonlinear integer programming algorithms: a survey, Working Paper No. 8005, College of Business and Economics, Washington State University, Pullman, Washington 99164.
9. Hammer, P.L. and Rudeanu, S. (1968), *Boolean Method in Operation Research and Related Area*. Springer Verlag, New York, NY.
10. Hansen, P. (1979), Methods of nonlinear 0-1 programming, *Discrete Applied Mathematics* 5, 53–70.
11. Hansen, P., Jaumard, B. and Mathon, V. (1993), Constrained nonlinear 0-1 programming, *ORSA Journal on Computing* 5: 97–119.
12. Kiwiel, K. (1995), Proximal level bundle methods for convex nondifferentiable optimization, saddle-point problems and variational inequalities, *Mathematical Programming* 69: 89–104.
13. Lemarechal, C., Nemyrovskii, A. and Nesterov, Y. (1995), New variant of bundle methods, *Mathematical Programming* 69: 111–147.
14. Loh, H. and Papalambros, P. (1989), A sequential linearization approach for solving mixed-discrete nonlinear design optimization problems, Technical Report, UM-MEAM-89-08, Design Laboratory, University of Michigan.
15. Mauricio, D., *Métodos de Resolução de Problemas da Programação Não Linear Inteira*, Ph.D. Thesis, COPPE, Federal University of Rio de Janeiro, Rio de Janeiro, 1994.
16. Mauricio, D., Maculan, N. and Scheimberg, S. (1994), An efficient cutting plane for 0-1 nonlinear programming, Publicação Técnica Interna LCENG/Universidade Estadual do Norte Fluminese, Rio de Janeiro.
17. Wu, Z. (1991), *A Subgradient Algorithm for Nonlinear Integer Programming and Its Parallel Implementation*. Ph.D. Thesis, Dept. of Math. Sci., Rice University, Houston.