

Pseudo-Boolean Optimization

Yves Crama*

Peter L. Hammer[†]

December 20, 2000

Abstract

This article briefly surveys some fundamental results concerning pseudo-Boolean functions, i.e. real-valued functions of 0-1 variables. Hundreds of papers have been devoted to the investigation of pseudo-Boolean functions, and a rich and diversified theory has now emerged from this literature. The article states local optimality conditions, outlines basic techniques for global pseudo-Boolean optimization, and shows connections between best linear approximations of pseudo-Boolean functions and game theory. It also describes models and applications arising in various fields, ranging from combinatorics to management science, and points to several classes of functions of special interest.

Keywords: integer programming, nonlinear 0-1 optimization, max-sat, max-cut, graph stability, game theory.

1 Pseudo-Boolean functions

1.1 Definitions and representations

A *pseudo-Boolean function* is a mapping from $\{0,1\}^n$ to \mathfrak{R} , i.e. a real-valued function of a finite number of 0-1 variables. Pseudo-Boolean functions have been introduced in [15], and extensively studied in [16] and in numerous subsequent publications; a detailed survey appears in [4].

Pseudo-Boolean functions generalize Boolean functions, which are exactly those pseudo-Boolean functions whose values are in $\{0,1\}$, i.e. those $f(X)$ for which $f^2(X) - f(X) \equiv 0$ in $\{0,1\}^n$. Since the elements of $\{0,1\}^n$ are in one-to-one correspondence with the subsets

*Ecole d'Administration des Affaires, University of Liège, Boulevard du Rectorat 7 (B31), 4000 Liège, Belgium, Y.Crama@ulg.ac.be

[†]RUTCOR, Rutgers University, 640 Bartholomew Road, Piscataway, NJ 08854-8003, U.S.A., Hammer@rutcor.rutgers.edu

of $N = \{1, 2, \dots, n\}$, every pseudo-Boolean function can be interpreted as a real-valued *set function* defined on $\mathcal{P}(N)$, the power set of $N = \{1, 2, \dots, n\}$. Viewing pseudo-Boolean functions as defined on $\{0, 1\}^n$, rather than on $\mathcal{P}(N)$, provides an algebraic viewpoint which sometimes carries clear advantages. It is easy to see for instance that the set of all pseudo-Boolean functions in n variables forms a vector space over \mathfrak{R} , and that the elementary monomials $\prod_{i \in A} x_i$ ($A \in \mathcal{P}(N)$) define a basis of this space. In particular, every pseudo-Boolean function $f(x_1, x_2, \dots, x_n)$ can be uniquely represented as a *multilinear polynomial* of the form

$$f(x_1, x_2, \dots, x_n) = c_0 + \sum_{k=1}^m c_k \prod_{i \in A_k} x_i, \quad (1)$$

where c_0, c_1, \dots, c_m are real coefficients, and A_1, A_2, \dots, A_m are nonempty subsets of N . When viewed as a function on $[0, 1]^n$, the right-hand side of (1) defines a continuous extension of the pseudo-Boolean function f , to be denoted f^c .

Note that every pseudo-Boolean function also admits (many) representations of the form

$$f(x_1, x_2, \dots, x_n) = b_0 + \sum_{k=1}^m b_k \left(\prod_{i \in A_k} x_i \prod_{j \in B_k} \bar{x}_j \right), \quad (2)$$

where b_0, b_1, \dots, b_m are real coefficients, and $\bar{x}_j = 1 - x_j$ for $j = 1, 2, \dots, n$. If $b_k \geq 0$ for all $k = 1, 2, \dots, m$, then we say that the expression (2) is a *posiform* of f . It is easy to see that every pseudo-Boolean function can be expressed as a posiform. Other representations of pseudo-Boolean functions have been recently investigated in [12].

1.2 Representative models

Besides nonlinear binary optimization, pseudo-Boolean functions can also be used to model a wide variety of problems in different fields of application.

Maximum satisfiability. Consider a collection of Boolean clauses $\left(\bigvee_{i \in A_k} \bar{x}_i \vee \bigvee_{j \in B_k} x_j : k = 1, 2, \dots, m \right)$. The *maximum satisfiability* problem is to find a vector (x_1, x_2, \dots, x_n) in $\{0, 1\}^n$ which satisfies the largest possible number of clauses in the collection. This problem, which generalizes the NP-complete satisfiability problem, is equivalent to that of minimizing the posiform (2) with $b_k = 1$ for $k = 1, 2, \dots, m$; see e.g. [17].

Graph theory. Consider a graph $G = (N, E)$ with positive weights $w : N \rightarrow \mathfrak{R}$ on its vertices, and capacities $c : E \rightarrow \mathfrak{R}$ on its (undirected) edges. For every $S \subseteq N$, the cut $(S, N \setminus S)$ is the set of edges having exactly one endpoint in S ; the capacity of this cut is $\sum_{\{i, j\} \in (S, N \setminus S)} c(i, j)$. The *max-cut problem* is to find a cut of maximum capacity in G . This problem is equivalent to maximizing the quadratic pseudo-Boolean function

$$f(x_1, x_2, \dots, x_n) = \sum_{1 \leq i < j \leq n} c(i, j) (x_i \bar{x}_j + \bar{x}_i x_j), \quad (3)$$

under the interpretation that (x_1, x_2, \dots, x_n) is the characteristic vector of S .

A stable set in G is a set $S \subseteq N$ such that no edge has both of its endpoints in S ; the weight of S is $\sum_{i \in S} w(i)$. The *weighted stability problem* is to find a stable set of maximum weight in G . This can be seen to be equivalent to maximizing the quadratic pseudo-Boolean function

$$f(x_1, x_2, \dots, x_n) = \sum_{i=1}^n w(i) x_i - (1 + \min_{1 \leq i < j \leq n} w(i)) \sum_{1 \leq i < j \leq n} x_i x_j, \quad (4)$$

where (x_1, x_2, \dots, x_n) is the characteristic vector of S . Other connections between the weighted stability problem and pseudo-Boolean optimization (in particular, posiforms) have been exploited in [11].

Linear 0-1 programming. Consider the *linear 0-1 program*

$$\text{maximize } z(x_1, x_2, \dots, x_n) = \sum_{j=1}^n c_j x_j \quad (5)$$

$$\text{subject to } \sum_{j=1}^n a_{ij} x_j = b_i, \quad i = 1, 2, \dots, m \quad (6)$$

$$(x_1, x_2, \dots, x_n) \in \{0, 1\}^n. \quad (7)$$

This problem is equivalent to the quadratic pseudo-Boolean optimization problem

$$\text{maximize } f(x_1, x_2, \dots, x_n) = \sum_{j=1}^n c_j x_j - M \sum_{i=1}^m \left(\sum_{j=1}^n a_{ij} x_j - b_i \right)^2 \quad (8)$$

$$\text{subject to } (x_1, x_2, \dots, x_n) \in \{0, 1\}^n, \quad (9)$$

for a sufficiently large M .

Management applications. Constraints of the form:

$$x_i = 1 \quad \text{if and only if} \quad y = 1, \quad i \in A$$

are encountered in many management applications (e.g., capital budgeting, plant location, tool management problems, etc.; see [7, 16, 22]). Since such constraints simply express that $y = \prod_{i \in A} x_i$, pseudo-Boolean formulations of such problems often arise quite naturally by elimination of the y -variables.

Game theory. A *game in characteristic function form* is nothing but a pseudo-Boolean function f . If (x_1, x_2, \dots, x_n) is the characteristic vector of the set S , then $f(x_1, x_2, \dots, x_n)$ is interpreted as the payoff that the players indexed in S can secure by acting together. The multilinear representation of f and its continuous extension f^c play an interesting role in this context; see e.g. [2] and Section 3 below.

1.3 Special classes of pseudo-Boolean functions

Several authors have investigated special classes of functions with “nice” properties. Let us simply mention here *monotonic* functions (whose first derivatives – see below – have constant sign on $\{0, 1\}^n$), *supermodular* functions (whose second derivatives are nonnegative on $\{0, 1\}^n$), *polar* functions (which have a posiform (2) such that, for every k , either $A_k = \emptyset$ or $B_k = \emptyset$), *unimodular* functions (which are polar up to a switch $x_i \leftrightarrow \bar{x}_i$ on a subset of variables), (*completely*) *unimodal* functions (which have a unique local maximizer in (every face of) $\{0, 1\}^n$), etc. Strongly polynomial combinatorial algorithms for the maximization of supermodular functions have been recently proposed in [20, 21]. Unimodular functions can be maximized by max-flow algorithms [19]. Unimodal and related classes of functions have been introduced in [10], where the applicability of greedy algorithms for their optimization has also been investigated. The recognition problem for all these classes of functions is examined in [5].

2 Optimization

Optimization of pseudo-Boolean functions over subsets of $\{0, 1\}^n$ is also known as *non-linear 0-1 optimization*. A survey of this field is presented in [18]. We shall only mention a few fundamental facts, restricting ourselves mostly to the unconstrained case.

2.1 Local optima

If $f(x_1, x_2, \dots, x_n)$ is a pseudo-Boolean function, let us define its i th derivative Δ_i to be the pseudo-Boolean function

$$\Delta_i = f(x_1, \dots, x_{i-1}, 1, x_{i+1}, \dots, x_n) - f(x_1, \dots, x_{i-1}, 0, x_{i+1}, \dots, x_n).$$

It was shown in [16] that all the local maxima of the function f are characterized by the system of implications:

$$\text{if } \Delta_i > 0 \text{ then } x_i = 1, \quad \text{if } \Delta_i < 0 \text{ then } x_i = 0, \quad \text{for } i = 1, 2, \dots, n.$$

Let now m_i and M_i be arbitrary lower and upper bounds of Δ_i , e.g. the sums of the negative, respectively the positive, coefficients in the polynomial representation of Δ_i . Then it is clear that an equivalent characterization of the local maxima of the pseudo-Boolean function f is given by the system of inequalities

$$\Delta_i - M_i x_i \leq 0, \quad \Delta_i - m_i \bar{x}_i \geq 0, \quad \text{for } i = 1, 2, \dots, n.$$

2.2 Global optima

The continuous extension f^c has the attractive feature that its global maximizers are at the vertices of the hypercube $[0, 1]^n$ and hence, coincide with those of f (this is because (1) is linear in every variable). This implies that continuous global optimization techniques can be applied to f^c in order to compute the maximum of f . This approach did not prove computationally efficient in past experiments, but remains conceptually valuable. As an amusing corollary, one may note for instance that the optimum of the max-cut function (3) is at least $f^c(\frac{1}{2}) = \frac{1}{2} \sum_{1 \leq i < j \leq n} c(i, j)$. Thus, we find that every graph contains a cut of capacity at least equal to one-half the total edge capacity, a well-known result of graph theory. Moreover, such a cut can be found efficiently.

A combinatorial variable elimination algorithm for pseudo-Boolean optimization was proposed by Hammer, Rosenberg and Rudeanu [15, 16]. The following streamlined version and an efficient implementation of this algorithm are described in [8]. Let $f(x_1, x_2, \dots, x_n)$ be the function to be maximized. We can write

$$f(x_1, x_2, \dots, x_n) = x_1 \Delta_1(x_2, x_3, \dots, x_n) + h(x_2, x_3, \dots, x_n),$$

where h and the first derivative Δ_1 do not depend on x_1 . Clearly, there exists a maximizer of f , say $(x_1^*, x_2^*, \dots, x_n^*)$, with the property that $x_1^* = 1$ if and only if $\Delta_1(x_2^*, x_3^*, \dots, x_n^*) > 0$. This suggests to introduce a function $t(x_2, x_3, \dots, x_n)$ such that $t(x_2, x_3, \dots, x_n) = \Delta_1(x_2, x_3, \dots, x_n)$ if $\Delta_1(x_2, x_3, \dots, x_n) > 0$ and $t(x_2, x_3, \dots, x_n) = 0$ otherwise. Letting $f_1 = t + h$, we have reduced the maximization of the original function f in n variables to the maximization of f_1 , which only depends on $n - 1$ variables. Repeating n times this elimination process eventually allows to determine a maximizer of f . An efficient implementation of this algorithm is proposed in [8], where it is also proved that the algorithm runs in polynomial time on pseudo-Boolean functions with *bounded tree-width*.

Another classical approach consists in transforming the problem $\max\{f(X) : X \in \{0, 1\}^n\}$ into an equivalent linear 0-1 programming problem by substituting a variable y_k for the k th monomial T_k of (1) (or (2)) and setting up a collection of linear constraints which enforce the equality $y_k = T_k$. The continuous relaxation of this linear formulation yields an easily computable upper-bound on the maximum of f . Properties of this upper-bound and of related formulations of f have been investigated in [1, 13] and in a series of subsequent paper; see [6] for a brief account and Section 2.3 for related considerations.

2.3 Quadratic 0-1 optimization

Quadratic 0-1 optimization is an important special case of pseudo-Boolean optimization, both because numerous applications appear in this form, and because the more general case is easily reduced to it. Indeed, consider a function f of the form (1), assume that $|A_1| > 2$ and select $j, l \in A_1$. Then, the function

$$g(x_1, x_2, \dots, x_n, y) = c_0 + c_1 \left(\prod_{i \in A_1 \setminus \{j, l\}} x_i \right) y + \sum_{k=2}^m c_k \prod_{i \in A_k} x_i - M(x_j x_l - 2x_j y - 2x_l y + 3y)$$

where y is a new variable, and M is large enough, has the same maximum value as f ($y = x_j x_l$ in every maximizer of g). Applying recursively this procedure yields eventually a function of degree 2.

Best linear majorants and minorants of pseudo-Boolean functions can provide important information on the function. It was shown in [13] that for any quadratic pseudo-Boolean function f , one can construct a linear function

$$l(x_1, x_2, \dots, x_n) = l_0 + \sum_{j=1}^n l_j x_j,$$

called the *roof dual* of f , majorizing $f(x_1, x_2, \dots, x_n)$ in every binary point (x_1, x_2, \dots, x_n) and having the following property of *strong persistency*: if l_j is strictly positive (resp. negative), then x_j must be equal to 1 (resp. 0) in every maximizer of f . In other words, roof duality allows the determination of the optimal values of a subset of variables. Moreover, $\max\{f(X) : X \in \{0, 1\}^n\} = \max\{l(X) : X \in \{0, 1\}^n\}$ if and only if an associated 2-SAT problem is satisfiable. While the determination of the roof dual in [13] was accomplished via linear programming, it was shown later [3] that this problem can be reduced to a maxflow problem in an associated network.

3 Linear approximations

In order to find the best linear approximation $L(f)$ of a pseudo-Boolean function f in the norm L_2 , it is sufficient to know how to determine the best linear L_2 -approximation of a monomial. Indeed, considering the polynomial representation (1), it is clear that

$$L(f) = c_0 + \sum_{k=1}^m c_k L\left(\prod_{i \in A_k} x_i\right).$$

On the other hand, it was shown in [14] that

$$L\left(\prod_{i \in A} x_i\right) = \frac{1}{2^{|A|}}(1 - |A| + 2 \sum_{i \in A} x_i) \quad \text{for all } A \subseteq N.$$

It was shown in the same paper that the best quadratic, cubic, ... L_2 -approximations can also be obtained by similar simple closed formulas.

Important game-theoretical applications of best L_2 -approximations consist in finding the Banzhaf indices of the players of a simple game, or the Shapley values of the players of an n -person characteristic function game. Indeed, as shown in [14], these indices are simply the coefficients of best (weighted) linear L_2 -approximations of pseudo-Boolean functions describing these games.

Another important application of these results allows the efficient determination of excellent heuristic solutions of unconstrained nonlinear binary optimization problems [9].

Acknowledgements. This work was partially supported by ONR grants N00014-92-J-1375 and N00014-92-J-4083, by NSF grant DMS-9806389, and by research grants from the Natural Sciences and Engineering Research Council of Canada (NSERC).

References

- [1] E. Balas and J.B. Mazzola, Nonlinear 0-1 programming. I. Linearization techniques, *Mathematical Programming* 30 (1984) 1-22.
- [2] J.M. Bilbao, *Cooperative Games on Combinatorial Structures*, Kluwer Academic Publishers, Boston/Dordrecht/London, 2000.
- [3] E. Boros and P.L. Hammer, A max-flow approach to improved roof duality in quadratic 0-1 minimization, RUTCOR - Rutgers University Center for Operations Research, Research Report RRR 15-1989, Piscataway, NJ, 1989.
- [4] E. Boros and P.L. Hammer, Pseudo-Boolean optimization, *Discrete Applied Mathematics* (2001), forthcoming.
- [5] Y. Crama, Recognition problems for special classes of polynomials in 0-1 variables, *Mathematical Programming* 44 (1989) 139-155.
- [6] Y. Crama, Concave extensions for nonlinear 0-1 maximization problems, *Mathematical Programming* 61 (1993) 53-60.
- [7] Y. Crama, Combinatorial optimization models for production scheduling in automated manufacturing systems, *European Journal of Operational Research* 99 (1997) 136-153.
- [8] Y. Crama, P. Hansen and B. Jaumard, The basic algorithm for pseudo-Boolean programming revisited, *Discrete Applied Mathematics* 29 (1990) 171-185.
- [9] T. Davoine, P.L. Hammer and B. Vizvari, A heuristic for Boolean optimization problems, RUTCOR - Rutgers University Center for Operations Research, Research Report RR 42-2000, Piscataway, NJ, 2000.
- [10] D. De Werra, P.L. Hammer, T. Liebling and B. Simeone, From linear separability to unimodality: A hierarchy of pseudo-Boolean functions, *SIAM Journal on Discrete Mathematics* 1 (1988) 174-184.
- [11] Ch. Ebenegger, P.L. Hammer and D. de Werra, Pseudo-Boolean functions and stability of graphs, *Annals of Discrete Mathematics* 19 (1984) 83-97.
- [12] S. Foldes and P.L. Hammer, Disjunctive and conjunctive normal forms of pseudo-Boolean functions, RUTCOR - Rutgers University Center for Operations Research, Research Report RRR 1-2000, Piscataway, NJ, 2000.

- [13] P.L. Hammer, P. Hansen and B. Simeone, Roof duality, complementation and persistency in quadratic 0-1 optimization, *Mathematical Programming* 28 (1984) 121-155.
- [14] P.L. Hammer and R. Holzman, Approximations of pseudo-Boolean functions : Applications to game theory, *ZOR - Methods and Models of Operations Research* 36 (1992) 3-21.
- [15] P.L. Hammer, I. Rosenberg and S. Rudeanu, On the determination of the minima of pseudo-Boolean functions (in Romanian), *Stud. Cerc. Mat.* 14 (1963) 359-364.
- [16] P.L. Hammer and S. Rudeanu, *Boolean Methods in Operations Research and Related Areas*, Springer, Berlin, 1968.
- [17] P. Hansen and B. Jaumard, Algorithms for the maximum satisfiability problem, *Computing* 44 (1990) 279-303.
- [18] P. Hansen, B. Jaumard and V. Mathon, Constrained nonlinear 0-1 programming, *ORSA Journal on Computing* 5 (1993) 97-119.
- [19] P. Hansen and B. Simeone, Unimodular functions, *Discrete Applied Mathematics* 14 (1986) 269-281.
- [20] S. Iwata, L. Fleischer and S. Fujishige, A combinatorial, strongly polynomial-time algorithm for minimizing submodular functions, in: *Proceedings of the 32nd ACM Symposium on Theory of Computing*, 2000, pp. 97-106.
- [21] A. Schrijver, A combinatorial algorithm minimizing submodular functions in strongly polynomial time, *Journal of Combinatorial Theory (Series B)* 80 (2000) 346-355.
- [22] K.E. Stecke, Formulation and solution of nonlinear integer production planning problems for flexible manufacturing systems, *Management Science* 29 (1983) 273-288.