

A unified convergence theory for Non monotone Direct Search  
Methods (DSMs) with extensions  
to DFO with mixed and categorical variables

Ubaldo M. García Palomares  
Universidade de Vigo  
ubaldo@gti.uvigo.es

September 5, 2019

**Abstract**

This paper presents a unified convergence theory for non monotonous Direct Search Methods (DSMs), which embraces several algorithms that have been proposed for the solution of unconstrained and boxed constraints models. This paper shows that these models can be *theoretically* solved with the same methodology and under the same weak assumptions. All proofs have a common pattern and the monotone counterparts are merely special cases. The theory holds regardless of the absence of first derivatives and is extended to models with discrete and/or categorical variables. It is proved that a Clarke stationary point is obtained with probability one, with a finite number of search directions at all iterations. The key for convergence rests upon a judicious choice of a random set of search directions, which we assume to contain a subset of orthogonal directions, because an easy convergence proof can be exposed for both smooth and non smooth functions.

**Keywords.**

Non monotone. Direct Search Methods. Mixed and Categorical Variables. Deivative Free.

# 1 Introduction

Direct Search Methods (DSMs) have been intensively used for solving optimization problems with continuous variables. Despite their simplicity, encouraging numerical results have been reported for solving small and medium size problems, with or without first order derivative information [19, and references therein]. DSMs solve iteratively the general minimization model minimize  $f(x), x \in (\mathcal{F} \subseteq \mathbb{R}^n)$  as follows: At the  $i$ -th iteration an estimate  $x_i \in \mathcal{F}$  to the solution is known. A search direction  $d_i \in \mathbb{R}^n$  and a stepsize  $\tau_i \in \mathbb{R}_+$  complying with manageable conditions are found, and a new estimate  $x_{i+1} = x_i + \tau_i d_i \in \mathcal{F}$  is generated. Early application of DSMs on optimization models with continuous variables were monotone; i.e. they forced a sufficient decrease to the function values, namely  $f(x_{i+1}) \leq f(x_i) - \sigma_i$ ; for some  $\sigma_i > 0$ . Steepest descent, variable metric and Newton methods are classical DSM instances which use or approximate first order information. Pattern Search Methods (PSMs), introduced in [29], is an instance of a monotone DSM with no derivative information. An extensive list of references for monotone algorithms is given in [5]. To our knowledge [20] stands for the pioneer work for the use of a non monotone algorithm for solving an unconstrained optimization model with continuous variables. Several variants have been suggested since then. Among others, we point out [9, 15, 21, 30], which basically inherit the properties of Newton’s method with explicit derivative information. All these works [9, 20, 21, 30] claim that non monotone DSMs might improve the performance of a monotone algorithm when the estimates enter into a curved narrow valley. In smooth constrained optimization non monotone algorithms are used to avoid the Maratos’ effect [31, and references therein]. Furthermore, non monotone DSMs have a more stable performance on noisy functions than its monotone counterpart [5, 12] and the authors in references [12, 13, 15, 16, 17] employ non monotone DSMs to *try to* avoid convergence to nearby local minimizers. Finally, the authors in [18] claim that almost any *deterministic* monotone optimization algorithm for solving models with continuous variables has a non monotone counterpart. The algorithms presented in this paper are considered as *deterministic* because their basic versions can be implemented with no randomness involved, although convergence to a Clarke point is lost.

The main objective of this work is to present, under the same umbrella, a unified theory for monotone and non monotone *deterministic* DSMs, which allows us to have at hand a *common* framework for solving model (1) and its particular instances: unconstrained and boxed constrained optimization problems with continuous and/or discrete variables. We assume that the remaining constraints, if any, may be absorbed by the objective function via penalization, Lagrangean techniques, or any other technique that may transform the constrained model into a single or a sequence of boxed constrained models structured like (1). For instance, the authors in [1, 2] propose a simple barrier function, which assumes  $f(x) = \infty$  for all non feasible points. We solve the mixed optimization model (1) under lax conditions, regardless of the availability of derivative information.

$$\text{minimize } g(x; y) : \mathbb{R}^{n+r} \rightarrow \mathbb{R} \tag{1a}$$

$$\text{subject to } : y \in \mathcal{G} = \{y \in \mathbb{R}^r : y = h(z) : \mathbb{Z}^p \rightarrow \mathbb{R}^r\} \tag{1b}$$

$$s \leq (x; z) \leq t, \tag{1c}$$

A word of cautious to the reader. To facilitate our exposition, we use now and then the MatLab vector notation: when  $x \in \mathbb{R}^n$ , and  $y \in \mathbb{R}^r$ ,  $(x; y)$  stands for a column vector of  $n + r$  components. Thus,  $s$  and  $t$  are known vectors with  $n + p$  components, which represent respectively, the lower and upper bounds of all variables involved in constraint (1c).

In this introductory exposition we prove some Lemmas that are essential for the understanding of the algorithms. We as well state our notation and some useful definitions. At the end of this Section we display a pseudo code that will be the core of non monotone DSMs. It will be needed to analyze all models portrayed from Sections 2 to Section 4.

The remainder of this paper is organized as follows. Section 2 describes the DSMs proposed for solving models with continuous variables. For unconstrained and boxed constrained models, convergence of monotone and non monotone DSMs to a stationary point is proved under mild assumptions; namely, compactness, Fréchet differentiability and continuous first order derivatives around limit points. A special Sub Section is devoted to non smooth models. We show that the DSM may generate a sequence converging to a point satisfying a stationary point in the Clarke sense, provided a judicious finite set of *random* search directions is used. Section 3 adapts the theory developed in Section 2 to the analysis of unconstrained and boxed constrained models with discrete variables. An initial effort to use PSMs for solving nonlinearly constrained models with discrete *-and/or categorical-* variables can be found in [1, 5]. The work exposed in this Section is an outgrowth of the non monotone DSM proposed in [16], where the discrete variables were assumed to lay on a regular grid of discrete points. It is worth mentioning that the boxed constrained models only needs a straightforward variation of the search directions needed to ensure convergence for the unconstrained models. No extra assumptions are required. The box constraints given by (1c) do not add any complexity for solving model (1). When discrete variables are present in the model, the algorithm stops at a stationary point whose function value is not higher than any other function value at discrete points located in a neighborhood previously defined, which might include random variables. Section 4 extends the results from Sections 2 and 3 to model (1), with both continuous and discrete variables.

For nonsmooth problems we show that the non monotone DSM proposed in this paper shares convergence properties with monotone DSMs frequently cited. Since there exist non smooth functions that do not decrease along any direction from a point that is not a minimum [11, Exercise 2.6], we always consider a finite set of directions of search; nonetheless, we show that our algorithm might detect directions of negative curvature. As our purpose is to apply this methodology in Derivative Free Optimization (DFO), where differentiability and other function features may be unknown to the user, no effort is devoted to the rate of convergence. In short, and for the sake of clarity we gradually analyze the models from the basic unconstrained optimization model to the more general mixed variable model (1), which is formally studied in Section 4.

This paper devotes a close attention to *deterministic* non monotone DFO algorithms, which in general, have been scarcely considered in the open literature. All convergence results hold with either explicit use of first order derivatives or for Derivative Free Optimization (DFO). Moreover, if first order information is at hand, the algorithms simplify significantly, and, generally, a singleton direction of search is easily identified. The paper concludes with some remarks and open questions to research topics that are under investigation.

A word on the peculiarities of our notation:

- $\mathbb{R}^n, \mathbb{R}^p, \mathbb{R}^r$ , and so forth, are Euclidean spaces.
- $\mathcal{F} \subseteq \mathbb{R}^n$  is the feasible set
- Superscripts stand for components and subindices represent different vectors.
- $d^T w$  is the inner product of vectors  $d, w$  defined in the same Euclidean space.
- $uu^T$  is a matrix  $U$  with components  $U^{jk} = u^j u^k$
- $\nabla f(x)$  is the gradient
- $f^0(\hat{x}, d) \triangleq \limsup_{\substack{x \rightarrow \hat{x}, x + \tau d \in \mathcal{F} \\ \tau \rightarrow 0}} \frac{f(x + \tau d) - f(x)}{\tau}$  is the generalized directional derivative at  $\hat{x}$  along  $d$ .
- $\sigma(\cdot) : \mathbb{R}_+ \rightarrow \mathbb{R}_+$  stands for a *sigma-function*, which is:  
 forcing:  $(\{\sigma(\tau_i)\} \rightarrow 0) \Rightarrow (\{\tau_i\} \rightarrow 0)$ ,  
 unbounded from above:  $\sigma(\tau) \rightarrow \infty$  as  $\tau \rightarrow \infty$ , and  
 little  $o(\tau^2) : \lim_{\tau \rightarrow 0} \sigma(\tau)/\tau^2 = 0$ .
- $g(x; y) : \mathbb{R}^{n+r} \rightarrow \mathbb{R}$ ; with  $x \in \mathbb{R}^n, y \in \mathbb{R}^r$   
 $f(x) \triangleq g(x; h(z_i)) : \mathbb{R}^n \rightarrow \mathbb{R}$ , when the discrete variable is fixed at  $z_i$ .  
 $f(z) \triangleq g(x_i; h(z)) : \mathcal{G} \rightarrow \mathbb{R}$  when the continuous variable is fixed at  $x_i$ .
- $i$  will be the iteration number.

In the sequel we admit that **A2** and **A3** given below hold, unless we are certain that  $f(\cdot) \in C^1$ , in which case, **A1** substitutes for **A2**. In all instances **A3** may be replaced by any theoretical condition that ensures that the sequence of solution estimates remains in a compact set.

- A1** The objective function  $f(x; y) : \mathbb{R}^{n+r} \rightarrow \mathbb{R}$  is *Fréchet-differentiable* and continuously differentiable around limit points.
- A2**  $f^0(\hat{x}, \hat{d})$ , the generalized directional derivative at  $\hat{x}$  in the direction  $\hat{d}$  exists and it is finite.
- A3** The level set  $L = \{(x; y) : f(x; y) \leq \varphi_0\}$  is compact for any given  $\varphi_0$ .

**Definition 1** Let  $\hat{x} \in \mathbb{R}^n$ , and let  $\varphi \in \mathbb{R}$ , with  $\varphi \geq f(\hat{x})$ . The (unit) direction  $\hat{d} \in \mathbb{R}^n$  is a *quasi direction of descent (qdd)* if there exists a positive  $\hat{\tau} \in \mathbb{R}$  such that

$$f(\hat{x} + \hat{\tau}\hat{d}) - \varphi \leq -\sigma(\hat{\tau}). \quad (2)$$

**Definition 2** Given  $\varphi \geq f(\hat{x})$ , the point  $\hat{x} \in \mathbb{R}^n$  is *partially stationary* on a set  $D$  of unit directions if there exists a sequence  $\{\tau_j\}_{j \in K \subseteq \mathbb{N}} \rightarrow 0$  such that

$$f(\hat{x} + \tau_j d) - \varphi > -\sigma(\tau_j) \quad \text{for all } d \in D \quad (3)$$

**Remark 1** For a practical convenience (partial) stationarity is defined on a finite set  $D$  and the existence of some sequence  $\{\tau_j > 0\} \rightarrow 0$ . Ideally, we want (3) to hold on the infinite set  $\tilde{D} = \{d \in \mathbb{R}^n : \|d\| = 1\}$ , because this would imply stationarity in a Clarke sense. We will prove below that this can occur with probability 1 when a finite set of directions is carefully constructed. We also point out that for smooth functions with derivative information a singleton direction is enough to ensure stationarity.

**Lemma 1** *Let  $\{x_j\} \rightarrow \hat{x} \in \mathbb{R}^n$  and  $f^0(\hat{x}, \hat{d}) < 0$ , then  $\hat{d}$  is a qdd at some  $x$  in the vicinity of  $\hat{x}$ .*

**Proof:** If  $\hat{d}$  is not a qdd for any  $x$  close to  $\hat{x}$ , we can identify a sequence  $\{\tau_j\}_{j \in \mathbb{N}} \downarrow 0$  and

$$\frac{f(x + \tau_j \hat{d}) - f(x)}{\tau_j} \geq \frac{f(x + \tau_j \hat{d}) - \varphi}{\tau_j} > -\sigma(\tau_j)/\tau_j. \quad (4)$$

By taking limits, it follows that  $f^0(\hat{x}, \hat{d}) \geq 0$ , a contradiction.

**Corollary 2** *If  $\{x\} \rightarrow \hat{x}$  and  $f^0(\hat{x}, d) \leq \alpha < 0$ ,  $d$  is a qdd for all  $x$  sufficiently close to  $\hat{x}$ .*

**Lemma 3** *Let  $\hat{d}$  be a qdd for some  $\hat{x} \in \mathbb{R}^n$ ,  $\hat{\tau} > 0$  and  $\hat{\varphi} \geq f(\hat{x})$ . Let  $D$  be any set of (unit) directions. Under assumption **A3**, there exists  $\tau > 0$  satisfying*

$$f(\hat{x} + \tau \hat{d}) - \hat{\varphi} \leq -\sigma(\tau) \quad (5a)$$

$$f(\hat{x} + 2\tau d) - \hat{\varphi} > -\sigma(2\tau) \text{ for all } d \in D \quad (5b)$$

**Proof:** Since  $\hat{d}$  is a qdd, (5a) holds by assumption for  $\tau = \hat{\tau}$ . If (5b) holds there is nothing to prove. If the statement of the Lemma is false, algorithm 1 generates an infinite loop with a sequence of  $f(\cdot)$  values unbounded from below, but this contradicts **A3**.

---

Given  $\hat{x}, \hat{d}, \hat{\tau}, \hat{\varphi}$  satisfying (5a).  
 WHILE  $f(\hat{x} + \tau d) - \hat{\varphi} \leq -\sigma(\tau)$  for some  $d \in D$   
 $\tau = 2\tau$   
 END WHILE

---

Algorithm 1:  $f(\cdot)$  unbounded below

**Definition 3** *Any set  $D$  satisfying (5b) may be denoted as a blocking set.*

Algorithm 1 is just one way to prove the existence of a blocking set. Many other alternatives are available, but this algorithm is easy to implement, and it is close to its monotone counterpart algorithm proposed in [14] and the non monotonous versions developed afterwards [13, 15, 12].

The following Lemmas will be useful to deal with differentiable functions.

**Lemma 4** *Let  $\hat{x} \in \mathbb{R}^n$  be a fixed point, let  $D = \{d_1, \dots, d_q\}$ ,  $q > n$ , be a set that positively spans  $\mathbb{R}^n$ , and let  $\tau_0 \in \mathbb{R}_+$  be any finite positive number. If  $f(\cdot)$  is differentiable and*

$$\left\langle \begin{array}{l} d \in D \\ \tau \in (0 \ \tau_0] \end{array} \right\rangle \Rightarrow (f(\hat{x} + \tau d) - \varphi > -\sigma(\tau)), \quad (6)$$

then  $\nabla f(\hat{x}) = 0$ .

**Proof:** It is well known that if  $D$  positively spans  $\mathbb{R}^n$ , and  $\nabla f(\hat{x}) \neq 0$ , there exists a descent direction  $d \in D : d^T \nabla f(\hat{x}) < 0$ . It follows by Lemma 1 that  $d$  is a qdd; consequently  $\tau \in (0 \ \tau_0]$  exists satisfying (5a); therefore (6) can occur only if  $\nabla f(\hat{x}) = 0$ .

If  $\nabla f(x) = 0$ , it is not possible to find a direction of descent satisfying  $d^T \nabla f(x) < 0$ . Next Lemma shows that under more stringent conditions it is possible to identify directions with negative curvature.

---

FOR  $j = 1, \dots, n$   
      $w^j = 0$  when  $x^j$  is close to the boundary; otherwise  
      $w^j$  uniformly distributed in  $[-1, 1]$ , when  $x \in \mathbb{R}^n$   
      $w^j$  randomly chosen in  $\{-1, 0, 1\}$ , when  $x \in \mathbb{Z}^n$   
 IF  $(u^T u = 0)$   $u^T u = 1$  a correction  
 FOR  $j = 1, \dots, n$   
      $d_j = e_j - (2w^j / u^T u)u$ , the  $j$ -th column of the matrix  $I - 2uu^T / u^T u$

---

Algorithm 2: Construction of  $D = \{\pm d_1, \dots, \pm d_n\}$

**Lemma 5** *If  $\lim_{\tau \rightarrow 0} \sigma(\tau)/\tau^2 = 0$ , and  $f(\cdot) \in C^2$ , the inequality (5a) holds for directions with negative curvature. Indeed if  $\exists (\tau \in (0, \tau_0])$  satisfying (5a) and  $\nabla f(x) = 0$ , then for all  $\tau \in (0, \tau_0]$ , it follows that*

$$\frac{1}{2}d^T \nabla^2 f(x)d + \frac{o(\tau^2)}{\tau^2} = \frac{f(x + \tau d) - f(x)}{\tau^2} \geq \frac{f(x + \tau d) - \varphi}{\tau^2} > -\frac{\sigma(\tau)}{\tau^2};$$

but this cannot hold for small enough  $\tau$  when  $d^T \nabla^2 f(x)d < 0$ .

**Lemma 6** *Let  $\{d_1, \dots, d_n\}$  be a finite set of  $n$  orthogonal directions with  $\|d_j\| = 1, j = 1, \dots, n$ , and let  $D \supseteq \{\pm d_1, \dots, \pm d_n\}$ . It follows that*

$$\forall w \in \mathbb{R}^n \quad (\exists d \in D : d^T w \leq -(1/\sqrt{n})\|w\|). \quad (7)$$

Moreover, if  $f(\cdot)$  is Fréchet differentiable,

$$\exists d \in D : d^T \nabla f(x) \leq -(1/\sqrt{n})\|\nabla f(x)\|. \quad (8)$$

**Proof:** (7) is a well known fact and its proof is omitted. (8) is an instance of (7).

In the forthcoming Sections we will see that the existence of a descent direction in the search set  $D_i$  facilitates, in theory, the convergence to stationary points of smooth functions. By the previous Lemma, orthogonal directions implicitly enforce (8). Besides, the *cosine* value  $\alpha = 1/\sqrt{n}$  cannot be improved by any  $D$  that spans  $\mathbb{R}^n$  positively, with either  $n + 1$  or  $2n$  search directions [26]. It is worth mentioning that the use of  $n + 1$  directions might improve the performance of derivative free optimization algorithms when the *cosine* measure  $\alpha$  is bounded away from zero [4]. We now expose a succinct list of the properties of the orthogonal set  $\{\pm H e_1 \cdots \pm H e_n\}$ , the columns of the orthogonal Householder matrix  $H = (I - (2/u^T u)uu^T)$ , with  $\|u\| \neq 0$ , because, under certain conditions, they generate a dense set of search directions, which seems to be vital for proving stationarity in the Clarke's sense.

- As  $d_j = H e_j = e_j - (2w^j)u$  there is no need to generate the whole matrix  $H$ . The vector  $u$  contains the information needed to generate the search directions.  
This feature allows a significant saving in memory for medium and large problems.
- It is relatively simple to send directions of search to different processors [14]
- The vector  $u$  can be randomly generated. It has been argued that some degree of randomness may benefit the performance of a *deterministic* algorithm [19]. In [6] the authors claim that they did not find a deterministic strategy to improve the MADS algorithm.
- The published numerical results since its inception in [14] have been highly competitive [15, 12, 13, 16, 17]

$$[x_{i+1}, \varphi_{i+1}, \tau_i, D_i] = \text{ITERATION}(x_i, \varphi_i)$$

.....  
 Given  $x_i \in \mathcal{F}, \varphi_i \geq f(x_i)$       $\mathcal{F}$  is the feasible set  
 this pseudocode generates  $x_{i+1} \in \mathcal{F}, \varphi_{i+1} \geq f(x_{i+1}), \tau_i \in \mathbb{R}_+,$   
 $D_i = \{d_{i1}, \dots, d_{iq}\}, q$  directions in  $\mathbb{R}^n$   
 .....

1.     Generate a set  $D_i$  satisfying (16) for smooth functions
  2.     IF  $f(x_i + \tau d) > \varphi_i - \sigma(\tau)$  for a predefined  $\{\tau_j\} \rightarrow 0$  and for all  $d \in D_i$   
        RETURN  $[x_i, \varphi_i, 0, D_i]$
  3.     ELSE Find  $y \in \mathcal{F}, \tau_i \in \mathbb{R}_+$  satisfying
    - 3.1.      $f(y + \tau_i d_{ij}) \leq \varphi_i - \sigma(\tau_i)$  for some  $d_{ij} \in D_i$
    - 3.2.      $f(y + 2\tau_i d) > \varphi_i - \sigma(2\tau_i)$  for all  $d \in D_i$
- 
4.     Choose  $x_{i+1} = y + \tau_i d_{ij}$   
         $\varphi_{i+1} \in [f(x_{i+1}), \varphi_i - \sigma(\tau_i)]$
- 

Algorithm 3: Basic Iteration for a non monotone DSM

For completeness, algorithm 2 describes an *easy* way to generate the orthogonal directions with the Householder matrix. Note that

$$\langle u^j = 0 \rangle \Rightarrow \begin{cases} d_j = e_j \\ d_k^j = 0, k \neq j \end{cases} \quad (9)$$

This implication turns out to be highly relevant for some of our algorithms, specially when we want to force a direction on the coordinate axis to avoid any hamming. We advance that our algorithm forces  $u^j = 0$  when the variable  $x^j$  is close to either of its bounds. In any event, this will be explicitly indicated when needed by the algorithms.

We end this Section by calling your attention to algorithm 3. It is a pseudo code that becomes the basic iteration of all algorithms analyzed in this paper.  $[x_{i+1}, \varphi_{i+1}, \tau_i, D_i] = \text{ITERATION}(x_i, \varphi_i)$  is the expression we use to invoke algorithm 3, meaning that the new estimate  $x_{i+1}$ , the functional upper-bound  $\varphi_{i+1}$ , the stepsize  $\tau_i$ , and the set of search directions  $D_i$  are values generated by  $\text{ITERATION}(\ )$  from the values of  $x_i, \varphi_i$ . The returned values satisfy (5), unless  $\tau_i = 0$ .

In all events, the set  $D$  of search directions is a finite set of unit vectors, unless explicitly stated otherwise.

## 2 Non monotone DSMs for models with continuous variables

This Section describes non monotone DSMs for solving optimization models with continuous variables. Sub Section 2.1 deals with the unconstrained optimization model

$$\text{minimize } f(x), x \in \mathcal{F} = \mathbb{R}^n, \quad (10)$$

and Sub Section 2.2 deals with the boxed constrained optimization model

$$\text{minimize } g(x), x \in \mathcal{F} = \{x \in \mathbb{R}^n : s^k \leq x^k \leq t^k, k = 1, \dots, n\}, \quad (11)$$

where  $s^k, t^k$  are, respectively, the lower and the upper bound of the variable  $x^k$ . We assume  $s^k < t^k$ ; otherwise, when  $s^k = t^k$ , the variable  $x^k$  has a constant value and it is no longer considered as a variable. Our approach for solving (11) will be to consider the barrier function

$$f(x) = \begin{cases} g(x), & \text{if } x \in \mathcal{F} \\ \varphi_0 + 1, & \text{otherwise,} \end{cases} \quad (12)$$

and try to use the same tools used for solving the unconstrained model (10). To avoid repetition of the arguments, the feasible set  $\mathcal{F}$  stands for  $\mathbb{R}^n$  for unconstrained models.

We will prove that under assumptions **A2**, **A3**, a sequence of estimates  $\{x_i\}_{i \in \mathbb{N}}$  converges to a partially stationary point (definition 2) and to a point satisfying first-order necessary conditions in the Clarke's sense with probability 1. In addition to these convergence results, this paper also shows convergence to a point satisfying the classical first order conditions when **A1** holds.

## 2.1 Unconstrained optimization

Algorithm 4 is a non monotone DSM which solves the optimization models (10) and (11) with continuous variables, regardless of the presence *-or absence-* of derivative information.

We point out from step 3 of ITERATION( ) that

$$f(x_{i+1}) \leq \varphi_{i+1} \leq \varphi_i - \sigma(\tau_i), \quad (13)$$

**Remark 2**  $\varphi_{i+1}$  may be the observed value of a random variable uniformly distributed in  $[f(x_{i+1}) \varphi_i]$ .

Lemma 7 shows that  $\{\tau_i\}_{i \in \mathbb{N}} \rightarrow 0$ , which in turn forces a (sub)sequence of estimates to converge to a quasi stationary point characterized by definition 2.

**Lemma 7** *If **A2** and **A3** hold, the algorithm 4 either generates  $\tau_i = 0$  for some  $i$  or it generates  $\{\tau_i\} \rightarrow 0$ . In either case, algorithm 4 returns a partially stationary point satisfying definition 2.*

**Proof:** Algorithm 4 calls ITERATION( ) at every iteration. Let us assume that the IF clause does not hold in line 2 of ITERATION( ), then  $\varphi_{i+1}$  is updated by (13). It follows that

$$\min_{x \in \mathcal{F}} f(x) \leq f(x_{k+1}) \leq \varphi_{k+1} \leq \varphi_i - \sum_{k \geq j \geq i} \sigma(\tau_j); \quad (14)$$

which, by **A3**, implies  $\{\sigma(\tau_i)\} \rightarrow 0$ . A fortiori,  $\{\tau_i\} \rightarrow 0$ . From 3.2 and 4 of algorithm 3 it follows that  $f(x_i + 2\tau_i d) > \varphi_i - \sigma(2\tau_i)$  for all  $d \in D_i$ . Let us denote  $\lambda_i = 2\tau_i$ . By compactness we can identify an accumulation point  $\bar{x}, \bar{D}$ , a sequence  $\{x_i, D_i, \lambda_i\} \rightarrow (\bar{x}, \bar{D}, 0)$  and

$$\frac{f(x_i + \lambda_i d) - f(x_i)}{\lambda_i} \geq \frac{f(x_i + \lambda_i d) - \varphi_i}{\lambda_i} > -\sigma(\lambda_i)/\lambda_i, \text{ for all } d \in \bar{D}. \quad (15)$$

Hence, we deduce that  $\limsup_{\substack{x \rightarrow \bar{x}, \tau \rightarrow 0 \\ d \in \bar{D}}} \frac{f(x + \tau d) - f(x)}{\tau} \geq 0$ .

To end the proof we have to analyze convergence when the IF clause in line 2 of ITERATION( ) is valid. If this is the case, the process RETURNS  $\tau_i = 0$  and (22) holds by replacing  $\lambda_i$  by  $\tau_j$  and  $\bar{D}$  by a fixed set  $D_i$  and we reach the same conclusion.

Our next task is to impose conditions on the blocking sets  $\{D_i\}$  that ensure convergence to stationary points of smooth functions. We assume that (16) holds. In Section 2.3 we will show that for non smooth functions the Clarke necessary condition holds with probability 1.

$$(\forall D_i)(\exists d \in D_i : \nabla f(x_i)^T d \leq -\alpha \|\nabla f(x_i)\|), \text{ for some } \alpha > 0. \quad (16)$$

$$[\hat{x}] = \text{CONTINUOUS}(x_0, \varphi_0, 0)$$

.....  
 minimize  $f(x), x \in \mathcal{F} \subseteq \mathbb{R}^n$   
 Given  $x_0 \in \mathcal{F}, \varphi_0 \geq f(x_0), \tau_0 > \epsilon \geq 0$   
 .....

.....  
 WHILE ( $\tau_i > \epsilon$ )  
 1.  $[x_{i+1}, \varphi_{i+1}, \tau_i, D_i] = \text{ITERATION}(x_i, \varphi_i)$   
      $i = i + 1$       (next iteration)  
 -----  
 2. RETURN  $x_i$   
 -----

Algorithm 4: Non monotone DSM for solving optimization models with continuous variables

**Theorem 8** *Let  $\{d_i\}$  be the sequence of search directions that satisfies (16). If **A1** and **A3** hold, the algorithm 4 generates  $\{\nabla f(x_i)\} \rightarrow 0$ .*

**Proof.** Let  $\lambda_i = 2\tau_i$ . From (5b) and (16) it follows that

$$\begin{aligned} -\alpha \|\nabla f(x_i)\| &\geq d_i^T \nabla f(x_i) = \frac{f(x_i + \lambda_i d_i) - f(x_i)}{\lambda_i} - o(\lambda_i)/\lambda_i \\ &\geq \frac{f(x_i + \lambda_i d_i) - \varphi_i}{\lambda_i} - o(\lambda_i)/\lambda_i \\ &> -\sigma(\lambda_i)/\lambda_i - o(\lambda_i)/\lambda_i. \end{aligned} \tag{17}$$

It is obvious that  $\alpha \|\nabla f(x_i)\| \leq \sigma(\lambda_i)/\lambda_i + o(\lambda_i)/\lambda_i$ . Since  $\lambda_i \rightarrow 0$ , it follows that  $\{\nabla f(x_i)\} \rightarrow 0$ . Therefore, we claim that

**Proposition 9** *Let  $\{D_i\}$  be a sequence of blocking sets each containing a direction of descent satisfying (16), and let the function upper bounds  $\{\varphi_i\}$  be updated by (13). Under assumptions **A1** and **A3**, the algorithm 4 with  $\mathcal{F} = \mathbb{R}^n$  generates  $\{\nabla f(x_i)\} \rightarrow 0$ .*

**Corollary 10** *If  $f(\cdot)$  is continuously differentiable, any accumulation point  $\hat{x}$  of  $\{x_i\}$  is stationary; more specifically,  $\nabla f(\hat{x}) = 0$ .*

**Proof.** It is obvious.

It is well known that whenever  $f(\cdot)$  is differentiable, any set  $D$ , that positively spans  $\mathbb{R}^n$ , contains a qdd satisfying (5a) for some  $\tau > 0$ . In general, there is not a simple way to *explicitly* find a qdd at  $x$ ; however, a qdd appear in  $D$  when:

- $f(\cdot) \in C^2$  and there are directions of negative curvature at  $x$ , that is,  $d^T \nabla^2 f(x) d < 0$ .
- $f(\cdot)$  is B-differentiable and has a direction with a negative directional derivative at  $x$ .

We should recall that even if  $f(\cdot)$  is differentiable, its first order derivatives might not be computable; and there is no way to explicitly verify (16), though Lemma 6 shows that it will be so when  $D_i$  is a set of orthogonal directions. Computability of the gradient simplifies the algorithm significantly. Convergence prevails for any positive definite matrix  $P$  and  $D_i = \{-P\nabla f(x_i)\}$ , a singleton.

## 2.2 Boxed constrained optimization

In this Sub-Section we deal with the boxed constrained model (11). As stated above we define the barrier function (12) and solve the equivalent unconstrained minimization model. However, minor

adjustments are required to ensure convergence to a point  $\bar{x} \in \mathcal{F}$  satisfying necessary conditions for being a minimizer of model (11). On the one hand, feasibility must be enforced throughout all iterations. This is easily implemented if the algorithm starts at a feasible point and uses the barrier function (12). On the other hand, the search directions must conform to the geometry of the constrained set. To avoid jamming at the boundary we suggest the use of the coordinate axis as search directions for those *binding* variables near either of its bounds. The remainder variables are considered *free variables* and the search directions follow the same approach for solving unconstrained models. To formalize these ideas, let us introduce the following notation and definitions.

Define the index set  $B_i$  of binding variables (*at the  $i$ -th iteration*) as:

$$B_i = \{1 \leq k \leq n : \min(x_i^k - s^k, t^k - x_i^k) \leq \beta\}, \text{ for some } \beta > 0 \quad (18)$$

Denote by  $\mathbb{R}^m$  the subspace spanned by  $\{x \in \mathbb{R}^n : x^k = 0, k \in B_i\}$ , and

Denote the projected gradient on  $\mathbb{R}^m$  as  $\overset{m}{\nabla}f(x_i)^k = \begin{cases} \nabla f(x_i)^k, & k \notin B_i \\ 0 & \text{otherwise} \end{cases}$

Define  $E_i = \{d_1, \dots, d_q\}$  as a finite set of  $q$  vectors in  $\mathbb{R}^m$  satisfying (16), that is,

$$(\exists d \in E_i) : \overset{m}{\nabla}f(x_i)^T d \leq -\alpha \|\overset{m}{\nabla}f(x_i)\|, \text{ for some } \alpha > 0. \quad (19)$$

Define the set  $D_i$  of search directions as:

$$D_i = E_i \cup \{\pm e_j, j \in B_i\} \quad (20)$$

Let  $\tilde{B}$  be a set that appears infinitely often in the sequence  $\{B_i\}_{i \in \mathbb{N}}$ .

Let  $K' = \{i : B_i = \tilde{B}\}$  and let  $\bar{x}$  be a limit point, that is,  $\{x_i\}_{i \in K} \rightarrow \bar{x}$ , for some  $K \subseteq K'$ .

We now state the convergence properties for functions satisfying **A1**, and later we will show convergence with probability 1 for non smooth functions.

**Theorem 11** *Let  $\{D_i\}$  be the sequence of search directions satisfying (20). Under assumptions **A1**, **A3**, algorithm 4 generates a sequence  $\{x_i\}_{i \in K}$  converging to a stationary point  $\bar{x}$  satisfying:*

$$\begin{aligned} (s^k = \bar{x}^k) &\Rightarrow \nabla f(\bar{x})^k \geq 0 \\ (\bar{x}^k = t^k) &\Rightarrow \nabla f(\bar{x})^k \leq 0 \end{aligned} \quad (21a)$$

$$(s^k < \bar{x}^k < t^k) \Rightarrow \nabla f(\bar{x})^k = 0 \quad (21b)$$

**Proof.** Let  $\tilde{B}, K, \bar{x}$  be defined as above. To prove (21a) we merely prove  $(s^k = \bar{x}^k) \Rightarrow \nabla f(\bar{x})^k \geq 0$ .

The case  $(\bar{x}^k = t^k) \Rightarrow \nabla f(\bar{x})^k \leq 0$  is quite similar and it is omitted.

As  $x_i^k \rightarrow \bar{x}^k$ , it follows that  $x_i^k - s^k \leq \beta$  for all large enough  $i$ ; ergo  $k \in B_i$  and  $f(x_i + \lambda_i e_k) - f(x_i) \geq -\lambda_i \sigma(\lambda_i)$ . By taking limits we obtain  $\nabla f(\bar{x})^k \geq 0$ .

To prove (21b) we need to consider 2 cases:

a.  $k \in B_i$

By properties of the algorithm we have  $f(x_i + \lambda_i e_k) - f(x_i) \geq -\lambda_i \sigma(\lambda_i)$  and  $f(x_i - \lambda_i e_k) - f(x_i) \geq -\lambda_i \sigma(\lambda_i)$ . Together these 2 inequalities imply that  $\nabla f(\bar{x})^k = 0$ .

b.  $k \notin B_i$

By construction  $E_i$  is a finite set that contains a direction that satisfies (16). If **A1** and **A3** hold, we mimic the convergence theorem 8 replacing  $D$  by  $E$  and  $\nabla f(\cdot)$  by  $\overset{m}{\nabla}f(\cdot)$  and deduce that algorithm 4 generates  $\{\overset{m}{\nabla}f(x_i)\}_{i \in K} \rightarrow 0$ ; which means that  $\nabla f(\bar{x})^k = 0, k \notin \tilde{B}$ .

### 2.3 Convergence for non smooth functions

Lemma 7 proves that

$$\limsup_{\substack{x \rightarrow \bar{x}, \tau \rightarrow 0 \\ d \in \hat{D}}} \frac{f(x + \tau d) - f(x)}{\tau} \geq 0,$$

where  $\hat{D}$  is an accumulation point of the infinite sequence of search directions  $\{D_1, D_2, \dots\}$ . This is a rather weak result. The next Theorem gives sufficient conditions to ensure that a limit point  $\bar{x}$  is a Clarke point.

**Theorem 12** *Let us apply algorithm 4 to model (10). Let **A2**, **A3** hold and let the set of search directions be generated by algorithm 2 an infinite number of iterations. There exists  $\bar{x}$  such that*

$$\limsup_{\substack{x \rightarrow \bar{x}, \tau \rightarrow 0 \\ d \in \hat{D}^\infty}} \frac{f(x + \tau d) - f(x)}{\tau} \geq 0, \quad (22)$$

where  $\hat{D}^\infty = \{d \in \mathbb{R}^n : \|d\| = 1\}$

**Proof.** Let  $\hat{d} \in \hat{D}^\infty$ . We first prove that  $\hat{d}$  appears infinitely often in the search sets  $D_i$  with probability 1. Let the set of search directions be the columns of the Householder matrix  $H$ , constructed as depicted in algorithm 2. It is well known that given any unit vector, say  $\hat{d}$ , there is a unit vector  $\hat{u}$  and an index  $j$  such that  $\hat{d} = H e_j = e_j - (2\hat{u}^j)\hat{u}$ . Since  $u$  is randomly generated from a uniform distribution, the unit vector  $\hat{u}$ , as well as any other unit vector, will be generated infinitely often *with probability 1*. Consequently,  $\hat{d}$  will be generated infinitely often *with probability 1*.

From the sequence  $\{x_i, D_i\}_{i \in \mathbb{N}}$  generated by the algorithm, extract a subsequence  $\{x_i, D_i\}_{i \in K \subseteq \mathbb{N}}$  such that  $\{x_i\}_{i \in K} \rightarrow \bar{x}$  and  $\hat{d} \in D_i$ , for  $i \in K$ . We can assert that

$$\frac{f(x_i + \lambda_i \hat{d}) - f(x_i)}{\lambda_i} \geq -\sigma(\lambda_i)/\lambda_i \quad \text{with} \quad \lambda_i \rightarrow 0 \quad \Bigg|_{i \in K}$$

Since  $\hat{d} \in \hat{D}^\infty$  was arbitrary, the conclusion follows.

### 3 Models with discrete variables

We now apply a non monotone DSM to model (23) with *discrete* variables, but no continuous variables.

$$\begin{aligned} & \text{minimize } f(z), z \in \mathcal{F} \\ & \mathcal{F} = \{z \in \mathbb{Z}^n : s \leq z \leq t\}. \end{aligned} \quad (23)$$

Model (23) is a combinatorial type model, where we have to find a point  $z$  among a finite but *possibly* large number of *feasible* points. Formally, it is a nonconvex constrained problem, though when  $f(\cdot)$  is convex, modern terminology classifies model (23) as a convex integer problem, because it becomes convex when the integrality requirements are removed from the model. It is worth mentioning that the conspicuous *all-integer* model (23) may be NP-hard, and several strategies mainly based on relaxations,

cutting cuts, branch and bound, surrogate models and *heuristics* have been devised for solving real models with *integer/discrete* variables. In [16] the authors proposed a new perspective and devised a non monotone DSM for solving a model with *grid* variables regularly distributed along the coordinate axis. In [7] the authors adapt the Pattern Search algorithm for solving this kind of models.

Model (23) is an optimization model subjected to bounds on the variables and only minor modifications are needed to the schemes for solving models (10) and (11) with continuous variables. We recall that this paper assumes that other constraints can be handled via penalization [10, 25], Lagrangean [23], infinite barriers [1] or any other appropriate technique. We now define a stationary point for model (23).

**Definition 4** *We admit that  $\bar{z}$  is a local solution to model (23) if  $f(\bar{z}) \leq f(z)$  for all  $z \in \mathcal{N}(\bar{z})$ , where  $\mathcal{N}(\bar{z})$  is a finite set of neighbors previously accepted by the practitioner. A suitable neighborhood is given by:*

$$\mathcal{N}(\bar{z}) = \left\{ z \in \mathcal{F} : \begin{cases} z - \bar{z} = +e_j \\ \text{or} \\ z - \bar{z} = -e_j \end{cases} \text{ for some } j \in \{1, 2, \dots, n\} \right\}.$$

We could include *random* variables in  $\mathcal{N}(\bar{z})$ , as well as any other variables at the user's discretion. To use our skeleton algorithm 3 we need to construct a set  $D$  of unit directions satisfying (16). We also require  $\{z_i \in \mathbb{Z}^n\}$ ; hence  $\{z_i + \tau_i d_i \in \mathbb{Z}^n\}$ , which implies  $\{\tau_i d_i \in \mathbb{Z}^n\}$ . An easy way to comply with all these requirements is given in [16], with  $\{\tau_i \in \mathbb{N}\}$ ,  $\{d_i \in \mathbb{Q}^n\}$ .

Algorithm 5 solves model (23). It is essentially the algorithm 4 with a small addendum to ensure that the algorithm stops in a finite number of iterations at a point satisfying definition 4.

By the convergence theorem 8,  $\tau_i \rightarrow 0$ ; hence  $(\tau_i < 1)$  will occur after a finite number of iterations and the IF clause is evaluated in line 2. If there exists a neighbor  $v$  with a lower functional value, i.e.  $v \in \mathcal{N}(z_i)$  and  $f(v) < f(z_i)$ , then  $z_i$  is not stationary and the algorithm will execute again the line 1 with  $z_{i+1} = v$ ,  $\varphi_{i+1} = f(v)$ . This precludes to recheck  $z_i$  as a possible stationary point at later iterations. Algorithm 3 ensures that  $f(z_j) < \varphi_i$  for any  $j > i$ . Therefore

$$f(z_j) \leq \varphi_{i+1} = f(z_{i+1}) = f(v) < f(z_i), \tag{24}$$

As the number of discrete points is finite, (24) can only happen finitely many times. We assert that the following proposition is valid:

**Proposition 13** *Under assumptions A1, A3 the algorithm 5 finds a stationary point for model (23) in a finite number of iterations.*

## 4 Models with mixed variables

The model to be solved in this Section is formulated as:

$$\text{minimize } f(x; z) : \mathbb{R}^n \times \mathbb{Z}^p \rightarrow \mathbb{R} \tag{25a}$$

$$\text{subject to } (x; z) \in \mathcal{F} = \{x \in \mathbb{R}^n, z \in \mathbb{Z}^p : s \leq (x; z) \leq t\} \tag{25b}$$

where  $s$  and  $t$  are known vectors with  $n + p$  components, which represent respectively, the lower and upper bounds of all variables involved in model (25).

$$[\tilde{z}] = \text{DISCRETE}(z_0, \varphi_0)$$

.....  
 minimize  $f(z), z \in \mathbb{Z}^p \cap \mathcal{F}, \mathcal{F} \subseteq \mathbb{R}^n$   
 Given  $z_0 \in \mathbb{Z}^p \cap \mathcal{F}, \varphi_0 \geq f(z_0), \tau_0 > 1, \text{ NOT DONE}$   
 .....

.....  
 WHILE (NOT DONE)  
 1.  $[z_{i+1}, \varphi_{i+1}, \tau_i, D_i] = \text{ITERATION}(z_i, \varphi_i)$   
 2. IF  $(\tau_i < 1)$  AND  $\exists v \in \mathcal{N}(z_i) : f(v) < f(z_i)$   
      $z_{i+1} = v, \varphi_{i+1} = f(v)$   
      $i = i+1$  (next iteration)  
 3. ELSEIF  $(\tau_i < 1)$  DONE  
 END-WHILE  
 -----  
 4. RETURN  $z_i$   
 -----

Algorithm 5: Non monotone DSM for solving optimization models with discrete variables

A good reference that lists the difficulties, the standard methodology and software that has been used for solving mixed variable models can be found in [8, 22]; and a recent account on methods for solving mixed variable models in the context of DFO can be found in [27]. It seems relevant to point out the standard approach for solving model (25). Given  $z_i$ , at the start of the  $i$ th iteration, the continuous model (26a) is solved followed by (26b), the best choice among the neighbors of  $z_i$  [27]:

$$x_i = \operatorname{argmin} f(x, z_i), x \in \mathbb{R}^n, \tag{26a}$$

$$z_{i+1} = \operatorname{argmin} f(x_i, z), z \in \mathcal{N}(z_i), \tag{26b}$$

where  $\mathcal{N}(z_i)$  is a discrete neighborhood previously defined. In some applications the subproblem (26a) reduces to linear models [28] and  $z_{i+1}$  is chosen with heuristics linked to the structure of the problem. It is advisable to use known techniques whenever (26a) simplifies to a well structured problem.

**Remark 3** *The authors in [16] suggested to uniformly discretize the continuous variables on a grid and solve model (23) with no continuous variables. To improve the solution, the model is again solved on a finer grid until a convergence criteria is met. Details can be found in [16].*

Thus far, it has been shown that convergence properties and stopping criteria for *pure* models are shared by monotone DSMs and non monotone DSMs. In this Section we analyze the mixed integer model (25) and propose a new approach with two purposes in mind: Firstly, we show that non monotone algorithms offer a good alternative. Minor effort is needed to transform a monotone DSM to a non monotone DSM, and yet the latter may perform better for some kind of problems. Convergence and algorithmic properties are shared under the same assumptions. Secondly, we offer an approach that combines non monotone DSMs with variable separation. It is the author's belief that the *best* algorithm would probably employ a hybrid methodology. We now define stationarity for model (25).

**Definition 5** *Let  $\mathcal{B}(\tilde{x}, \rho)$  be a set of points in  $\mathcal{F}$  close to  $\tilde{x}$  and let  $\mathcal{N}(\tilde{z})$  be a discrete neighborhood of points in  $\mathcal{F}$  around  $\tilde{z}$ .  $(\tilde{x}; \tilde{z})$  is a local minimizer of problem (25) if*

$$\left. \begin{array}{l} x \in \mathcal{B}(\tilde{x}, \rho) \\ z \in \mathcal{N}(\tilde{z}) \end{array} \right\} \Rightarrow f(x; z) \geq f(\tilde{x}; \tilde{z}) \tag{27a}$$

We admit that  $(\bar{x}; \bar{z})$  is a local solution to model (25) if:

$$\bar{x} = \arg \min_{x \in \mathcal{F}} f(x; \bar{z}) \quad \text{AND} \quad \bar{z} \in \mathcal{N}(\bar{z}) \Rightarrow [f(\bar{x}; \bar{z}) \leq \min_{x \in \mathcal{B}(\bar{x}, \rho)} f(x; \bar{z})] \quad (27b)$$

Constraint  $x \in \mathcal{B}(\bar{x}, \rho)$  in (27b) can be replaced by  $x \in \mathcal{F}$ , especially if the practitioner strives for a global solution. A local minimizer defined by (27) has some benefits for solving the model (25) [27, Section 2]; nonetheless it is computationally more expensive. While (26b) requires the evaluation of  $f(x_i; z)$  at all  $z \in \mathcal{N}(z_i)$ , (27) may demand the solution of as many optimization models as elements in  $\mathcal{N}(z_i)$ . In any event, the same problem structure can be maintained if  $\mathcal{B}$  is defined with the  $\infty$  norm, that is,

$$\mathcal{B}(\bar{x}, \rho) = \{x \in \mathcal{F} : \bar{x} - \rho e \leq x \leq \bar{x} + \rho e\} \quad (28)$$

Model (25) can be solved with the trivial algorithm 6. If the practitioner considers that solving (27) until its completion may become expensive, s(he) could adopt the sufficient decrease condition as given by algorithm 7.

$$[(\bar{x}; \bar{z}), \bar{\varphi}] = \text{TRIVIAL}(z, \epsilon)$$

---

Given  $z \in \mathcal{F}, \epsilon > 0$

---

$x = \arg \min_{y \in \mathcal{F}} f(y; z), \varphi = f(x; z)$   
 FOR  $(w_1, \dots, w_q), w_j \in \mathcal{N}(z), j = 1, \dots, q$   
 $v_j = \arg \min_{y \in \mathcal{F}} f(y; w_j)$   
 IF  $f(v_j; w_j) < \varphi - \epsilon$   
 $(x; z) = (v_j; w_j), \varphi = f(v_j; w_j)$   
 END-FOR  


---

 RETURN  $(x; z), \varphi$

---

Algorithm 6: Algorithm for solving model (25)

$$[(\bar{x}; \bar{z}), \bar{\varphi}] = \text{SUFFDECREASE}(x; z)$$

---

Given  $(x; z) \in \mathcal{F}$

---

WHILE  $\exists (v \in \mathcal{F}, w \in \mathcal{N}(z))$  such that  $f(v; w) < f(x; z)$   
 $(x; z) = (v; w), \varphi = f(v; w)$   


---

 RETURN  $(x; z), \varphi$

---

Algorithm 7: Sufficient decrease approach for algorithm 6

#### 4.1 A non monotone DSM with Variable Separation

Let us denote  $w = \begin{pmatrix} x \\ z \end{pmatrix} \in \mathbb{R}^{n+p}$ . Model (25) can be seen as an optimization problem with  $n + p$  variables. To prove convergence under assumptions **A1**, **A3** with the same arguments we have been

using thus far we might require to construct a set of search directions  $D = \{d_1 \dots d_q\}$ ,  $d_j \in \mathbb{R}^{n+p}$  with  $d \in D$  satisfying (16) for differentiable functions. Nonetheless, a matter of concern is a proper choice of the stepsize  $\tau$ . We have that  $w_{i+1} = w_i + \tau_i d_i$ , that is,  $\begin{pmatrix} x_{i+1} \\ z_{i+1} \end{pmatrix} = \begin{pmatrix} x_i \\ z_i \end{pmatrix} + \tau_i \begin{pmatrix} d_i^x \\ d_i^z \end{pmatrix}$ , where  $d_i^x$  are the first  $n$  components of  $d_i$ , and  $d_i^z$  stand for the last  $p$  components of  $d_i$ . This updating might not look convincing, because we want to force  $z_i \in \mathbb{Z}^p$  for all iterations, which implies  $\tau_i d_i^z \in \mathbb{Z}^p$ ; and this restricts the  $\tau_i$  values, which is not desirable when searching on the *whole* subspace of the continuous variables. To overcome this shortcoming we suggest a variable separation, which prevents a search on all variables simultaneously. The algorithm searches on  $\mathbb{R}^n$  using a continuous stepsize  $\tau \in \mathbb{R}_+$  and searches on  $\mathbb{Z}^p$  with a stepsize  $\zeta$  such that  $\zeta d^z \in \mathbb{Z}^p$ .

Variable separation (VS) is a powerful technique that has been used in optimization problems with a multi processing environment. It is as well suitable for solving large systems [1, 24]. Convergence for DFO with VS and continuous variables was first established in [3] for monotone algorithms and later for non monotone DSMs in [12]. This Section presents a scheme that fits in the framework of the previous algorithms herein described. We assume that **A3** and **A1 (or A2)** hold, and use the same technique, albeit including a minor proviso for the construction of the set  $D$  of search directions:

$$D^x = \{d_1, \dots, d_q\}, d_j \in \mathbb{R}^n, j = 1, \dots, q \quad (29a)$$

$$D^z = \{d_1, \dots, d_q\}, d_j \in \mathbb{R}^p, j = 1, \dots, q \quad (29b)$$

$$D = \left\{ \begin{pmatrix} D^x \\ 0^{p \times q} \end{pmatrix}, \begin{pmatrix} 0^{n \times q} \\ D^z \end{pmatrix} \right\}, \quad (29c)$$

where  $0^{p \times q}, 0^{n \times q}$  are, respectively, a null  $p \times q$  matrix and a null  $n \times q$  matrix.

We hope not to offend the reader for using the same symbol  $q$  to stand for the number of search directions in  $D^x$  and  $D^z$ . The algorithm constructs  $D^x, D^z$  satisfying (16); for example, they could contain orthonormal sets in  $\mathbb{R}^n$  and  $\mathbb{R}^p$  respectively. For this choice, the set of search directions in  $\mathbb{R}^{n+p}$  is the set of orthonormal directions (29c) that also satisfies (16).

Algorithm 8 shows a C-like code for a DSM with variable separation adapted to the solution of model (25). We illustrate a code that alternates the search between continuous and discrete variables, though other options are possible. We now give a succinct description of algorithm 8 and sketch its convergence to a stationary point defined by (27). Line numbers are given as pointers to the algorithm. As stated above the algorithm defines 2 stepsizes:  $\tau \in \mathbb{R}$  for continuous steps and  $\zeta d^z \in \mathbb{Z}^p$  for discrete steps. When  $\tau > \epsilon$  (line 1) the algorithm carries out a search on the continuous variables. As hinted above, the search in  $\mathbb{R}^n$  is done on the  $q$  directions defined by  $D^x$ , while the  $z$  variables remain fixed. Similarly, the algorithm employs  $D^z$  to search on the discrete variables whenever  $\zeta \geq 1$  (line 2). All conditions that make valid the Theorems 8 and 11 prevail and  $\tau_i \rightarrow 0$ . Similarly  $\zeta$  decreases and in a finite number of iterations we obtain  $\zeta < 1$ . When this latter condition occurs, it follows that  $z_i + \zeta d^z \notin \mathbb{Z}^p$  and the algorithm evaluates a better alternative among the neighbors (line 3). If the algorithm does not find a *promising* neighbor (line 4), it ends.

## Categorical Variables

We end this Section with a brief mention to the categorical variables. A reminder: a categorical variable has no quantitative value. It represents a change of the mode the system is operating on. We denote  $v \in \{v_1, \dots, v_m\}$  as a *discrete categorical* variable that can take up to  $m$  *qualitative* values. We

$$[\bar{x}; \bar{z}] = \text{MIXED}(x_0, z_0, \varphi_0, \epsilon)$$

$$\underset{(x; z) \in \mathcal{F}}{\text{minimize}} \quad f(x; z) : \mathbb{R}^n \times \mathbb{Z}^p \rightarrow \mathbb{R}$$

$$\mathcal{F} = \{x \in \mathbb{R}^n, z \in \mathbb{Z}^p : s \leq (x; z) \leq t\}$$

$$\text{Given } x_0 \in \mathcal{F}, z_0 \in \mathcal{F}, \varphi_0 \geq f(x_0; z_0), \epsilon > 0$$

Choose  $\tau_0 \in \mathbb{R}_+, \tau_0 > \epsilon, \zeta_0 > 1$

WHILE  $(\tau_i > \epsilon)$  OR  $(\zeta_i \geq 1)$

1. IF  $(\tau_i > \epsilon)$ 

$$[x_{i+1}; z_i, \varphi_{i+1}, \tau_i, D_i^x] = \text{ITERATION}(x_i; z_i, \varphi_i)$$
 IF  $(\tau_i > \epsilon)$  Choose  $\zeta_i > 1$
  2. IF  $(\zeta_i \geq 1)$ 

$$[x_i; z_{i+1}, \varphi_{i+1}, \zeta_i, D_i^z] = \text{ITERATION}(x_i; z_i, \varphi_i)$$
 IF  $(\zeta_i \geq 1)$  Choose  $\tau_i > \epsilon$
  3. ELSE-IF  $(\zeta_i < 1)$  AND  $\exists v \in \mathcal{N}(z_i) : f(x_i; v) < f(x_i; z_i)$ 

$$z_{i+1} = v, \varphi_{i+1} = f(x_i; v),$$
 Choose  $\tau_i > \epsilon, \zeta_i > 1$
  4. ELSE-IF  $(\zeta_i < 1)$  RETURN  $(x_i; z_i)$
- END-WHILE
- 
5. RETURN  $(x_i; z_i)$

Algorithm 8: Non monotone DSM+VS for solving the mixed model (25)

assume formally that we can derive  $m$  optimization models:

$$\begin{aligned} &\underset{(x; z) \in \mathcal{F}_j}{\text{minimize}} && f_j(x; z, v_j) \quad j = 1, \dots, m \\ &\text{subjected to:} && (x; z) \in \mathcal{F}_j = \{x \in \mathbb{R}^{n_j}, z \in \mathbb{Z}^{p_j} : s_j \leq (x; z) \leq t_j\}. \end{aligned} \quad (30)$$

A traditional approach for solving models with categorical values has been to solve the  $m$  optimization models characterized by (30) and choose the best solution found. This approach could be acceptable for small  $m$  and small to medium systems. In the approach presented in this paper we admit that the neighbors  $\mathcal{N}(v_j), j = 1, \dots, m$  are previously given by the practitioner. Nonetheless, to facilitate the exposition we state that  $n_j = n$ , and  $p_j = p$ , for  $j = 1, \dots, m$ , and we want to solve models (31).

$$\begin{aligned} &\underset{(x; z) \in \mathcal{F}_j}{\text{minimize}} && f_j(x; z, v_j) \\ &\text{subjected to:} && (x; z) \in \mathcal{F}_j = \{x \in \mathbb{R}^n, z \in \mathbb{Z}^p : s_j \leq (x; z) \leq t_j\}, \\ &&& v \in V = \{v_1, \dots, v_m\}. \end{aligned} \quad (31)$$

Let us denote  $F(v_j) \triangleq \min_{x; z \in \mathcal{F}_j} f_j(x; z; v_j)$ . We want to find  $\bar{v} \in V$  such that  $F(\bar{v}) \leq F(v)$ , for all  $v \in \mathcal{N}(\bar{v})$ . Figure 9 depicts a straightforward *pseudo*-implementation for solving a model with categorical variables. If we admit local stationarity by definition 5 we must use  $F(v_j)$  instead of  $f_j(x_j; z_j; v_j)$  in line 1 and  $F(v_k)$  instead of  $f_k(x_k; z_k; v_k)$  in line 3. This obviously represents an extra budget, but it could be appropriate for global optimization. We recall *once more* that any existing constraint is embedded by the objective functions; albeit theoretically, there are no restrictions on the kind of models formulated as  $F(v_j)$ , as long as we have a suitable algorithm for its solution.

$$[\bar{x}; \bar{z}; \bar{v}] = \text{CATEGORICAL}(v_j)$$

---


$$\text{minimize } F(v) : V \rightarrow \mathbb{R}$$

Given  $v_j$

$\Downarrow$

$$F(v_j) = \underset{(x; z) \in \mathcal{F}_j}{\text{minimize}} f_j(x; z; v_j) : x \in \mathbb{R}^n, z \in \mathbb{Z}^p$$

$$\mathcal{F}_j = \{x \in \mathbb{R}^n, z \in \mathbb{Z}^p : s_j \leq (x; z) \leq t_j\}$$


---

0. Given  $v_j \in V$  and  $(x_j; z_j) \in \mathcal{F}_j$
  1.  $f = f_j(x_j; z_j; v_j)$ ,  $\bar{v} = v_j$
  2. WHILE  $\exists v_k \in \mathcal{N}(\bar{v})$  and  $(x_k; z_k) \in \mathcal{F}_k$  such that  $f_k(x_k; z_k; v_k) < f$
  3.  $f = f_k(x_k; z_k; v_k)$ ,  $\bar{v} = v_k$
- 
- RETURN  $\bar{v}$  and  $\text{argmin } F(\bar{v})$
- 

Algorithm 9: Solving a model with categorical variable  $v \in V = \{v_1, \dots, v_m\}$

## 5 Conclusion and final remarks

We have formalized and described a unified framework for non monotone direct search methods that solve unconstrained and boxed constrained optimization models. Some modern algorithms, which have shown a competitive numerical performance, belong to this framework. Besides, convergence theory has been enhanced: discrete and categorical variables can be included in the optimization models, with no extra assumptions.

Furthermore, for unconstrained problems, convergence to a Clarke stationarity point is obtained with probability one using at each iteration a finite number of orthogonal search directions *randomly generated*. It is also shown that only one search direction is needed whenever the gradient is known.

Despite the clear achievements reported in this paper regarding unconstrained and boxed constraint optimization models, further study is needed to extend the theory that allows us to handle linear and nonlinear constrained problems explicitly. Actually, it is not clear whether non monotone DSMs should force feasibility at all iterations for the linear constraints, or it is preferable to hybridize the techniques given in this paper with those frequently used, like Branch and Bound, which remove the integrality requirement, and Cuts that ruled out as a solution any non feasible estimates. We also should keep in mind that there are numerous techniques that transform non linearly constrained models in boxed constrained models.

For the sake of clarity, we have exposed the theory complemented with a basic but unspecified implementation of the algorithms. Our framework is open to many variants that deserve a close attention. Nonetheless, some specific implementations can be seen in the literature cited herein. We have left out for future study some important issues that are beyond the aims of this paper, as strategies for *accelerating convergence*, global minimization and parallelism.

**Acknowledgement.** Remark 2 and Corollary 10 were suggested by my colleagues at the *Universidade de Vigo*, Dr. José Ramón Fernández and Dr. Ramón González. Dr. Ildemaro García, from *Universidad Simón Bolívar*, helped to develop the pseudo codes.

## References

- [1] Mark A. Abramson, Charles Audet, James W. Chrissis, and Jennifer G. Walston. Mesh adaptive direct search algorithms for mixed variable optimization. *Optimization Letters*, 3:35–47, 2009.
- [2] Mark A. Abramson, Charles Audet, and Sébastien Le Digable. Orthomads: A deterministic MADS instance with orthogonal directions. *SIAM Journal on Optimization*, 22(2):948–966, 2009.
- [3] Charles Audet, John E Dennis Jr, and Sébastien Le Digabel. Parallel space decomposition of the mesh adaptive direct search algorithm. *SIAM Journal on Optimization*, 19(3):1150–1170, 2008.
- [4] Charles Audet, Amina Ihaddadene, Sébastien Le Digabel, and Christopher Tribes. Reducing the number of function evaluations mesh adaptive direct search algorithms. *SIAM Journal on Optimization*, 24(2):621–642, 2014.
- [5] Charles Audet, Amina Ihaddadene, Sébastien Le Digabel, and Christophe Tribes. Robust optimization of noisy blackbox problems using the mesh adaptive direct search algorithm. *Optimization Letters*, 12(4):675–689, 2018.
- [6] Charles Audet and John E Dennis Jr. Mesh adaptive direct search algorithms for constrained optimization. *SIAM Journal on Optimization*, 17(1):188–217, 2006.
- [7] Charles Audet, Sebastien Le Digabel, and Tribes. Christopher. The mesh adaptive direct search algorithm for granular and discrete variables. *SIAM Journal on Optimization*, 29(2):1164–1189, 2018.
- [8] Pietro Belotti, Christian Kirches, and Sven Leyffer. Mixed-integer nonlinear optimization. *Acta Numerica*, pages 1–131, 2013.
- [9] Yu-Hong Dai. On the nonmonotone line search. *Journal of Optimization Theory and Applications*, 112(2):315–330, 2002.
- [10] Giovanni Fasano, Giampaolo Liuzzi, Stefano Lucidi, and Francesco Rinaldi. A linesearch-based derivative-free approach for nonsmooth constrained optimization. *SIAM journal on optimization*, 24(3):959–992, 2014.
- [11] Ana Friedlander. *Elementos de programação não-linear* <https://www.ime.unicamp.br/~friedlan/livro.htm>. 1994.
- [12] Ubaldo M García-Palomares, Ildemaro J García-Urrea, and Pedro S Rodríguez-Hernández. On sequential and parallel non-monotone derivative-free algorithms for box constrained optimization. *Optimization Methods and Software*, 28(6):1233–1261, 2013.
- [13] Ubaldo M García-Palomares, Francisco J González-Castaño, and Juan C Burguillo-Rial. A combined global & local search (cgls) approach to global optimization. *Journal of Global Optimization*, 34(3):409–426, 2006.
- [14] Ubaldo M García-Palomares and José F Rodríguez. New sequential and parallel derivative-free algorithms for unconstrained minimization. *SIAM Journal on optimization*, 13(1):79–96, 2002.

- [15] Ubaldo Manuel Garcia-Palomares. Non monotone algorithms for unconstrained minimization: Upper bounds on function values. In F. Ceragioli, editor, *Proceedings of the 22nd IFIP TC7 conference*, pages 91–100, Torino, Italy, 2006. Springer. ISBN 0-387-32774-6.
- [16] Ubaldo Manuel García-Palomares and Pedro Salvador Rodríguez-Hernández. Unified approach for solving box-constrained models with continuous or discrete variables by non monotone direct search methods. *Optimization Letters*, 13(1):95–111, 2019.
- [17] Francisco J González-Castaño, Enrique Costa-Montenegro, Juan C Burguillo-Rial, and Ubaldo García-Palomares. Outdoor wlan planning via non-monotone derivative-free optimization: algorithm adaptation and case study. *Computational Optimization and Applications*, 40(3):405–419, 2008.
- [18] N. Gould, D. Orban, and Philippe Toint. GALAHAD a library of thread-safe fortran 90 packages for large scale nonlinear optimization. *Transactions of the ACM on Mathematical Software*, 29(4):353–372, 2003.
- [19] Serge Gratton, Clément W. Royer, Luis N. Vicente, and Zaikun Zhang. Direct search based on probabilistic feasible descent for bound and linearly constrained problems. *Computational Optimization and Applications*, 72(3):525–559, 2019.
- [20] Luigi Grippo, Francesco Lampariello, and Stefano Lucidi. A nonmonotone line search technique for Newton’s method. *SIAM Journal on Numerical Analysis*, 23(4):707–716, 1986.
- [21] Neng-zhu Gu and Jiang-tao Mo. Incorporating nonmonotone strategies into the trust region method for unconstrained optimization. *Computers & Mathematics with Applications*, 55(9):2158–2172, 2008.
- [22] Jan Kronqvist, David E Bernal, Andreas Lundell, and Ignacio E Grossmann. A review and comparison of solvers for convex minlp. *Optimization and Engineering*, 20:397–455, 2019.
- [23] R Lewis and Virginia Joanne Torczon. A globally convergent augmented lagrangian pattern search algorithm for optimization with general constraints and simple bounds. *SIAM Journal on optimization*, 12(4):1075–1089, 2002.
- [24] Sven Leyffer. Integrating sqp and branch-and-bound for mixed integer nonlinear programming. *Computational optimization and applications*, 18(3):295–309, 2001.
- [25] Giampaolo Liuzzi, Stefano Lucidi, and Francesco Rinaldi. Derivative-free methods for mixed-integer constrained optimization problems. *Journal of Optimization Theory and Applications*, 164(3):933–965, 2015.
- [26] Geir Nævdal. Positive bases with maximal cosine measure. *Optimization Letters*, 13(6):1381–1388, 2019.
- [27] Eric Newby and M Montaz Ali. A trust-region-based derivative free algorithm for mixed integer programming. *Computational Optimization and Applications*, 60(1):199–229, 2015.
- [28] Álvaro Pachón and Ubaldo García-Palomares. Mid-term frequency domain scheduler for resource allocation in wireless mobile communications systems. *Computer Communications*, 97:96–110, 2017.

- [29] Virginia Joanne Torczon. *Multidirectional search: a direct search algorithm for parallel machines*. PhD thesis, Rice University, 1989.
- [30] Hongchao Zhang and William W Hager. A nonmonotone line search technique and its application to unconstrained optimization. *SIAM journal on Optimization*, 14(4):1043–1056, 2004.
- [31] Jian L Zhou and André L Tits. Nonmonotone line search for minimax problems. *Journal of Optimization Theory and Applications*, 3:455–476, 1993.