

CONSTRAINT HANDLING AND SIMPLEX INITIALIZATION STRATEGY FOR THE NELDER-MEAD SIMPLEX METHOD APPLIED TO THE MELTDOWN CALIBRATION PROBLEM

Matheus de Oliveira Mendonça

ENACOM Group

Alameda das Latânias, 215, Belo Horizonte/Brasil

matheus.mendonca@enacom.com.br

Douglas Alexandre Gomes Vieira

ENACOM Group

Alameda das Latânias, 215, Belo Horizonte/Brasil

douglas.vieira@enacom.com.br

Lucas Sirimarco Moreira Guedes

ENACOM Group

Alameda das Latânias, 215, Belo Horizonte/Brasil

lucas.guedes@enacom.com.br

Adriano Chaves Lisboa

ENACOM Group

Alameda das Latânias, 215, Belo Horizonte/Brasil

adriano.lisboa@enacom.com.br

Gustavo Rodrigues Lacerda Silva

ENACOM Group

Alameda das Latânias, 215, Belo Horizonte/Brasil

gustavo.lacerda@enacom.com.br

ABSTRACT

This paper proposes a strategy for constraint handling in the Nelder-Mead simplex method based on a lexicographic dominance operator. Once this operator is defined, constraint handling is naturally absorbed by the original method. Furthermore, a new initialization strategy is also presented based on the variables' bounds and the initial point. The proposed method is applied to the meltdown calibration problem, which is a parameter estimation of a physicochemical simulator of Electric Arc Furnaces. Numerical experiments on real-world scenarios are conducted, and the results show that the proposed modifications outperform the original version in 93% of runs in terms of feasibility and objective function value, evidencing the robustness of the proposed modifications.

KEYWORDS. Nelder-Mead simplex method. Derivative-free constrained optimization. Lexicographic dominance.

Paper topics: POI, PM.

1. Introduction

This paper deals with the following optimization problem:

$$\begin{aligned} & \underset{x}{\text{minimize}} && f(x) \\ & \text{subject to} && g(x) \leq 0 \\ & && x \in \mathbb{X}, \end{aligned} \tag{1}$$

where $x \in \mathbb{R}^n$ is the optimization variable, $f : \mathbb{R}^n \rightarrow \mathbb{R}$ is an arbitrary, non-differentiable, nonlinear function, $g : \mathbb{R}^n \rightarrow \mathbb{R}^m$ are the inequality constraints and $\mathbb{X} = \{x \in \mathbb{R}^n \mid x_{\min} \leq x \leq x_{\max}\}$ where $x_{\min}, x_{\max} \in \mathbb{R}^n$ represent the variables' lower and upper bounds, respectively. When the derivative of (1) is not known analytically; the numerical errors introduced by gradient estimation might be considered for large-scale problems, resulting in a poor optimization performance [Fu, 2006] of gradient-based methods. Moreover, if (1) is formulated in terms of non-differentiable functions; it is necessary to use sub-gradient-based methods, such as cutting plane methods. In those cases, derivative-free methods become a natural alternative.

Hooke e Jeeves [1961], proposed a classical pattern search method based on deterministic perturbations on each coordinate of a given point and the following search towards the direction vector formed by the coordinates with lower function values. Similar methods were also proposed, such as in [Matyas, 1965] and [Luus e Jaakola, 1973], but with random perturbations to the current position point. In [Nelder e Mead, 1965], a method is proposed that considers $n + 1$ points on each iteration and replaces the worst of them, the one with a higher function value, with a new point generated based on some predefined operations. A detailed review of derivative-free methods can be found in [Rios e Sahinidis, 2013].

Metaheuristics such as evolutionary-based algorithms [Ishibuchi et al., 2008] and swarm-based algorithms [Banks et al., 2007] are also a natural choice when dealing with non-differentiable and complex optimization problems since they do not rely on any mathematical property of the objective and constraints functions. However, these algorithms are compute-intensive and non-deterministic, which for some real-world applications, is undesirable.

Therefore, this paper proposes a general strategy for constraint handling in the Nelder-Mead method based on a lexicographic dominance operator. Once this operator is defined, constraint handling is naturally absorbed by the classical method. This paper also presents strategies for dealing with variables' bounds violations and simplex initialization.

2. The steel scrap meltdown calibration problem

The steel scrap meltdown simulator of electric arc furnaces has adjustable parameters that impact on the final estimated metal and slag phase composition. Then, a calibrator (an optimization algorithm) searches for parameter values that minimize the sum of quadratic errors between the simulated and sample compositions. For real-world applications, the problem has approximately 35 variables.

Typical meltdown simulators follow the Algorithm 1, as described by Hay et al. [2019] and Odenthal et al. [2018].

Algorithm 1 Typical meltdown simulation process.

Input:

initial metal phase mass
 initial slag phase mass
 pig iron phase mass
 scrap phase mass
 initial metal temperature
 final metal temperature
 injection profiles (gas, carbon, slag former)
 parameters

Output:

metal phase mass
 slag phase mass
 gas phase mass
 dust phase mass
 compositions

```

1: Initial conditions setup
2: for  $t = 1, \dots, t_F$  do
3:   Temperature
4:   Carbon injection availability
5:   Melting
6:   Mass balance
7:   Gas balance
8:   Critical Silicon
9:   Combustion
10:  Chemical reactions
11:  Oxygen dissolved
12:  Calculate carbon coefficient
13:  FeO: activity and equilibrium
14:  Iron oxi-reduction
15:  Post combustion
16:  Slag off
17: end for
    
```

Consider the parameter values x and its technical limits $\underline{x}$ and $\bar{x}$, the inequality constraint $g(x)$ that controls the average ratio between the final chemical composition and the final solid mass, the meltdown simulation function $f(x)$ that returns the simulated compositions (see Algorithm 1), the sample composition y and normalized weights ν . The calibrator optimization problem is

$$\text{minimize } \nu(f(x) - y)^2 \quad (2)$$

$$\text{subject to } g(x) \leq 0 \quad (3)$$

$$\underline{x} \leq x \leq \bar{x} \quad (4)$$

The objective function and inequality constraints depend on the meltdown simulation. This simulator has a nonlinear dynamic due to the chemical reactions that take place in an electrical arc furnace. There is no guarantee that $f(x)$ and $g(x)$ are convex functions, and they can also be discontinuous [Čerňan et al., 2021; Kovačič et al., 2019].

3. The unconstrained Nelder-Mead method

The Nelder-Mead method [Nelder e Mead, 1965] was first designed for unconstrained optimization problems and has proven convergence for strictly convex functions [Lagarias et al., 1998]. It initially generates a set $\mathbb{S}_n = \{x_1, x_2, \dots, x_n, x_{n+1}\}$ of $n + 1$ points for a n -dimensional problem. The points are then sorted in such a way that

$$f(x_1) \leq f(x_2) \leq \dots \leq f(x_n) \leq f(x_{n+1}).$$

Let

$$x_c = \frac{1}{n} \sum_{i=1}^n x_i \quad (5)$$

be centroid of the n best points. The algorithm tries to replace the worst point x_{n+1} with a new point obtained by the following basic operations:

1. **Reflection:** $x_r = x_c + \alpha(x_c - x_{n+1})$;
2. **Expansion:** $x_e = x_c + \beta(x_r - x_c)$;
3. **Outside contraction:** $x_{oc} = x_c + \gamma(x_r - x_c)$;
4. **Inside contraction:** $x_{ic} = x_c - \gamma(x_r - x_c)$;
5. **Shrink:** $x_i = x_1 + \delta(x_i - x_1), \forall i \in [2, n + 1]$.

Commonly used parameters are $(\alpha, \beta, \gamma, \delta) = (1, 2, 0.5, 0.5)$, and the conditions required for each basic operation are compactly written in Algorithm 2.

Algorithm 2 Nelder-Mead simplex method.

Input: $\mathbb{S}_n = \{x_1, \dots, x_{n+1}\} \in \mathbb{X}$, parameters $(\alpha, \beta, \gamma, \delta)$.

```

1: while  $\neg$  stop criterion do
2:    $x_r \leftarrow x_c + \alpha(x_c - x_{n+1})$ 
3:   if  $f(x_r) < f(x_1)$  then
4:      $x_e = x_c + \beta(x_r - x_c)$ 
5:     if  $f(x_e) < f(x_r)$  then
6:        $x_{\text{new}} = x_e$  ▷ Expansion
7:     else
8:        $x_{\text{new}} = x_r$  ▷ Reflection
9:     end if
10:  else if  $f(x_r) < f(x_n)$  then
11:     $x_{\text{new}} = x_r$  ▷ Reflection
12:  else if  $f(x_r) < f(x_{n+1})$  then
13:     $x_{oc} = x_c + \gamma(x_c - x_{n+1})$ 
14:    if  $f(x_{oc}) \leq f(x_{n+1})$  then
15:       $x_{\text{new}} = x_{oc}$  ▷ Outside contraction
16:    end if
17:  else if  $f(x_r) \geq f(x_{n+1})$  then
18:     $x_{ic} = x_c - \gamma(x_c - x_{n+1})$ 
19:    if  $f(x_{ic}) \leq f(x_{n+1})$  then
20:       $x_{\text{new}} = x_{ic}$  ▷ Inside contraction
21:    end if
22:  else
23:     $x_i = x_1 - \delta(x_i - x_1), \forall i \in [2, n+1]$  ▷ Shrinkage
24:  end if
25: end while

```

A strategy for constraint handling is required for the general case, given that the original algorithm is only written in terms of the objective function values. It is also known that the simplex initialization is also crucial to the convergence in large-scale problems, as discussed in [Takenaga et al., 2023; Fajfar et al., 2019]. The next section describes the proposed strategy and the motivation behind each modification.

4. Proposed strategy

The proposed modifications are introduced in the following subsections, and then numerical experiments are described to demonstrate the algorithm's effectiveness.

4.1. Handling box-constraints violations

A standard strategy for dealing with box-constraints violations is variables' truncation [Luersen et al., 2004]. This operation might change the search direction nature of a point update (see Figure 1b). This operation becomes crucial for the Nelder-Mead method as it can accelerate the simplex degeneration (see Figure 1a) and a consequent premature convergence.

Another strategy is treating variables' bounds as linear inequality constraints [Djikpéssé et al., 2011]. This is an elegant solution, but it increases the problem's complexity by adding n linear functions to the original problem.

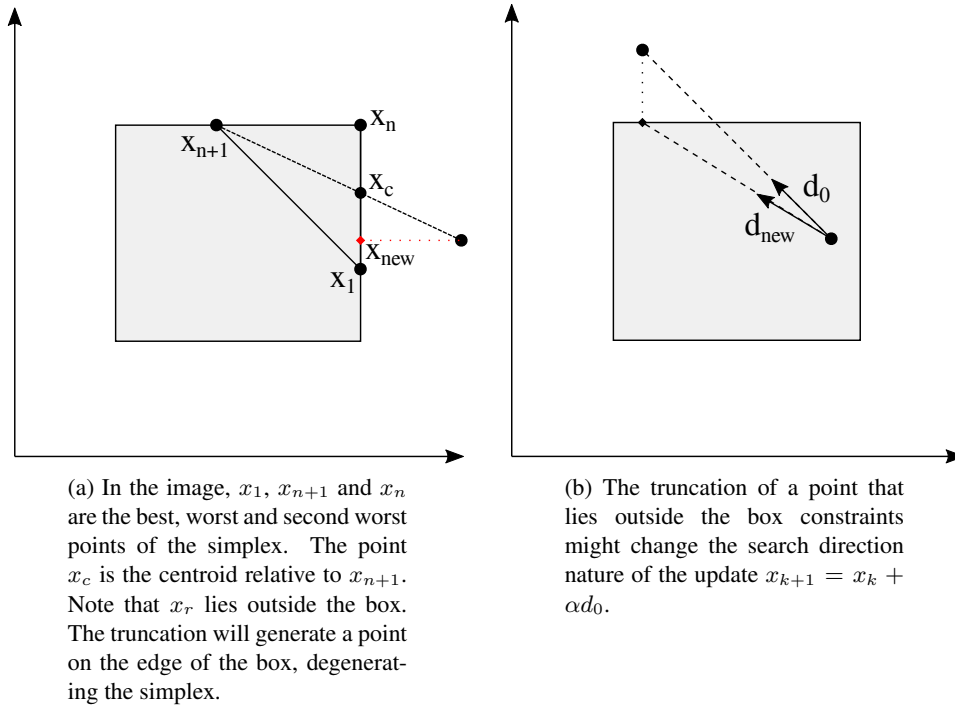


Figure 1: Variables' truncation shortcoming.

In this work, box-constraints violations are tackled indirectly once the proposed strategy does not generate points outside the search space $\mathbb{X}$. Further details are discussed in the following sections.

4.2. Initial simplex initialization

In [Luersen et al., 2004], the simplex is initialized by adding disturbances vectors to the initial point, following the equation

$$x_i = x_0 + p e_i + \sum_{\substack{k=1 \\ k \neq i}}^n q e_k, \quad i = 1, 2, \dots, n, \quad (6)$$

where

$$p = \frac{A}{n\sqrt{2}}(\sqrt{n+1} + n - 1), \quad (7)$$

$$q = \frac{A}{n\sqrt{2}}(\sqrt{n+1} - 1), \quad (8)$$

with A being a positive scalar controlling the initial simplex size and e_i being the i -th column of the n -dimensional identity matrix.

Note that these disturbances are always positive (see Figure 2a), which may cause the generated simplex points to exceed the limits of the search space, speeding up the process of the simplex degeneration. For instance, for $x_0 = x_{\max}$, all simplex points generated using this strategy would lie on $x_{\max}$.

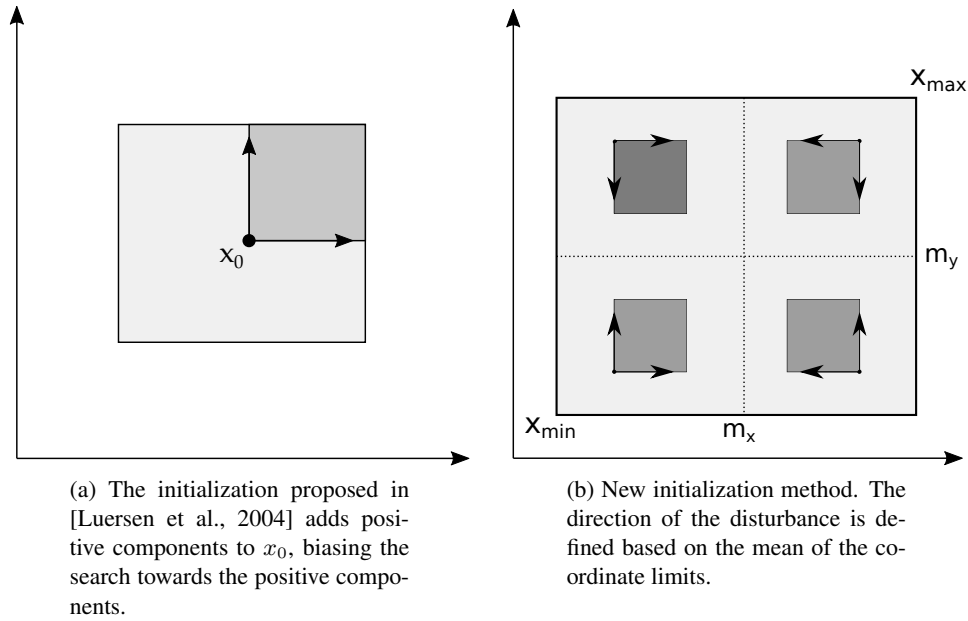


Figure 2: Simplex initialization strategies.

The new initialization method increases the search area of the initial simplex. It consists of changing the signal of the disturbance based on the mean of the coordinates bounds. Therefore, given a size A of the initial simplex, the i -th point of the simplex is also given by (6), with

$$e_i = \frac{m_i - x_{0,i}}{\|m_i - x_{0,i}\|} w_i, \quad (9)$$

with w_i being the i -th column of the n -dimensional identity matrix and

$$m_i = \frac{x_{\min,i} + x_{\max,i}}{2}, \quad (10)$$

where $x_{\min,i}$ and $x_{\max,i}$ represents the lower and upper bounds of the i -th variable.

Note that this strategy changes the sign of e_i if the coordinate $x_{0,i}$ lies in the upper half of the search space delimited by $x_{\min,i}$ and $x_{\max,i}$. Hence, increasing the search space of the method. Figure 2b depicts the proposed initialization strategy.

4.3. Lexicographic ordering

Several approaches have been presented to handle constraints in the literature. Li et al. [2016] presented a simple constraint-domination principle that verifies whether two solutions are feasible or not. Another different constraint handling is presented in [Djikipéssé et al., 2011], in which the dominance of a solution is defined based on linear – for the bounds' limits – and nonlinear constraints. In addition, [Audet e Tribes, 2018] presents a dominance operator that is also based on point aging. The term aging refers to the time when the point was generated.

In this paper, it is presented the following notion of order between two points $x, y \in \mathbb{R}^n$. Unlike [Luersen et al., 2004], instead of applying an adaptive linear penalization on the objective function to handle the constraints, the following definition will be used to compare two points.

Definition 4.1 (Lexicographic Dominance) Given $x, y \in \mathbb{R}^n$ and functions $f, g_s : \mathbb{R}^n \rightarrow \mathbb{R}$. We say that

$$x \sim y \iff \begin{cases} x \in \mathbb{X} \text{ and } y \notin \mathbb{X} & \text{or if} \\ g_s(x) \leq 0 < g_s(y) & \text{or if} \\ 0 < g_s(x) < g_s(y) & \text{or if} \\ g_s(x), g_s(y) \leq 0 \text{ and } f(x) \leq f(y) & . \end{cases} \quad (11)$$

where $g_s(x) = \sum_{i=1}^m \max\{0, g_i(x)\}$. This relation can be written as $x \leq_{\text{lex}} y$.

Once this lexicographic dominance operator is defined, the simplex points are sorted in a way that

$$x_1 \leq_{\text{lex}} x_2 \leq_{\text{lex}} \dots \leq_{\text{lex}} x_n \leq_{\text{lex}} x_{n+1}.$$

Since the simplex initialization proposed in Section 4.2 guarantees that all the simplex points are inside the search space, then all simplex operations with the lexicographic operator also guarantee a point $x_{\text{new}} \in \mathbb{X}$. Hence, the operations occur according to the following conditions:

1. **Reflection:** $x_r \leq_{\text{lex}} x_{n+1}$;
2. **Expansion:** $x_e \leq_{\text{lex}} x_r \leq_{\text{lex}} x_1$;
3. **Outside contraction:** $x_{\text{oc}} \leq_{\text{lex}} x_r \leq_{\text{lex}} x_{n+1}$ or $x_{\text{oc}} \leq_{\text{lex}} x_{\text{ic}}$;
4. **Inside contraction:** $x_{\text{ic}} \leq_{\text{lex}} x_{n+1}$;
5. **Shrink:** otherwise.

5. Results

The Nelder-Mead algorithm with the proposed modifications, i.e. simplex initialization and lexicographic constraint handling, was executed on real calibration scenarios of the meltdown simulator. All runs have the same problem dimension, varying the initial condition setup, bounds, and input parameters. The optimization results (feasibility and objective function value) were compared to the traditional initialization approach and constraint handling by adaptive penalization. To run these tests, the maximum number of iterations was set to 350 and the maximum time limit to 2 minutes. The environment used was a Windows 10 Amazon Workspace, with an Intel Xeon CPU E5-2676 v3 @ 2.40GHz processor and 4GB of RAM.

Table 1 show the comparison of the function value of the solution between the previous version (NM-Penalization) and the new version (NM-Lexicographic), both having the same stop criterion. Note that some runs that could not find a feasible solution (table cells with "NULL") now have a feasible one. In addition, most runs that found a feasible solution had their objective function value improved.

Table 1: Runs in which its solutions are close to the bounds of the search space. The highlighted function values in the table indicate that the new algorithm (NM-Lexicographic) found a better solution than the classical version (NM-Penalization), which happens in 28 out of 30 runs. The “NULL” term represents an infeasible solution.

# run	Objective function value, $f(x^*)$	
	NM-Penalization	NM-Lexicographic
1	0.7734827176	0.550731844975
2	2696.917198	353.276416221
3	2449.456791	1107.5224458
4	2518.068893	353.276416221
5	2741.143925	2725.5855398
6	2717.007918	2725.5855398
7	298.7652114	191.834515422
8	262.3861926	191.834515422
9	NULL	228.781481464
10	133.9209091	161.748826003
11	NULL	214.1787326
12	NULL	391.175822334
13	NULL	51.6564888745
14	NULL	223.717795031
15	3252.087972	464.36154126
16	3022.239318	464.36154126
17	NULL	2322.298267
18	3144.172271	3073.05510823
19	1054.978954	908.113488679
20	14.22523339	3.44853187053
21	14.22523339	3.44853187053
22	3.713548403	3.28091386052
23	3.939415919	3.43841388629
24	NULL	18.1178219841919
25	NULL	0.867603998191952
26	NULL	11.737007315334
27	NULL	1.89289487798574
28	NULL	0.0946783838784434
29	NULL	0.650742138098369
30	NULL	14.2926695826803

6. Conclusion

This paper presented a strategy for constraint handling and simplex initialization in the Nelder-Mead simplex algorithm using a lexicographic dominance operator. The method is intended to solve problems where the objective function is not necessarily differentiable at all the domain points.

The proposed strategy takes advantage of the comparative nature of the traditional method, preserving the base structure and implementation. The numerical experiments on the meltdown calibration problem show strong evidence of improvement compared to the original Nelder-Mead method.

These findings highlight the potential of the proposed modifications to enhance the performance of the Nelder-Mead algorithm in optimization tasks and suggest that the updated version (NM-Lexicographic) may be a viable option for handling simplex initialization and lexicographic constraints in similar scenarios.

Further research and experimentation may be warranted to validate and extend these findings, and to explore the applicability of the proposed modifications in other optimization problems and domains.

7. Acknowledgements

The authors would like to thank the National Council for Scientific and Technological Development (CNPq), CAPES, FAPEMIG, and CEMIG (Grant nos. GT669), Brazil, under an Aneel RD project for the financial support.

References

- Audet, C. e Tribes, C. (2018). Mesh-based nelder-mead algorithm for inequality constrained optimization. *Computational Optimization and Applications*, 71(2):331–352.
- Banks, A., Vincent, J., e Anyakoha, C. (2007). A review of particle swarm optimization. part i: background and development. *Natural Computing*, 6(4):467–484.
- Čerňan, M., Müller, Z., Tlustý, J., e Valouch, V. (2021). An improved svc control for electric arc furnace voltage flicker mitigation. *International Journal of Electrical Power & Energy Systems*, 129:106831.
- Djikpéssé, H., Couët, B., e Wilkinson, D. (2011). A practical sequential lexicographic approach for derivative-free black-box constrained optimization. *Engineering Optimization*, 43(7):721–739.
- Fajfar, I., Bűrmen, Á., e Puhán, J. (2019). The nelder-mead simplex algorithm with perturbed centroid for high-dimensional function optimization. *Optimization Letters*, 13:1011–1025.
- Fu, M. C. (2006). Gradient estimation. *Handbooks in operations research and management science*, 13:575–616.
- Hay, T., Echterhof, T., e Visuri, V.-V. (2019). Development of an electric arc furnace simulator based on a comprehensive dynamic process model. *Processes*, 7(11):852.
- Hooke, R. e Jeeves, T. A. (1961). Direct search solution of numerical and statistical problems. *Journal of the ACM (JACM)*, 8(2):212–229.
- Ishibuchi, H., Tsukamoto, N., e Nojima, Y. (2008). Evolutionary many-objective optimization: A short review. In *2008 IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence)*, p. 2419–2426. IEEE.
- Kovačič, M., Stopar, K., Vertnik, R., e Šarler, B. (2019). Comprehensive electric arc furnace electric energy consumption modeling: A pilot study. *Energies*, 12(11):2142.
- Lagarias, J. C., Reeds, J. A., Wright, M. H., e Wright, P. E. (1998). Convergence properties of the nelder-mead simplex method in low dimensions. *SIAM Journal on optimization*, 9(1):112–147.

- Li, J.-P., Wang, Y., Yang, S., e Cai, Z. (2016). A comparative study of constraint-handling techniques in evolutionary constrained multiobjective optimization. In *2016 IEEE Congress on Evolutionary Computation (CEC)*, p. 4175–4182. IEEE.
- Luersen, M. A., Riche, R. L., Guyon, F., Le Riche, R., e Guyon, F. (2004). A constrained, globalized, and bounded Nelder-Mead method for engineering optimization. *Structural and Multidisciplinary Optimization*, 27(1-2):43–54. ISSN 1615-147X.
- Luus, R. e Jaakola, T. (1973). Optimization by direct search and systematic reduction of the size of search region. *AIChE Journal*, 19(4):760–766.
- Matyas, J. (1965). Random optimization. *Automation and Remote control*, 26(2):246–253.
- Nelder, J. A. e Mead, R. (1965). A Simplex Method for Function Minimization. *The Computer Journal*, 7(4):308–313. ISSN 0010-4620. URL <https://dx.doi.org/10.1093/comjnl/7.4.308>.
- Odenthal, H.-J., Kemminger, A., Krause, F., Sankowski, L., Uebber, N., e Vogl, N. (2018). Review on modeling and simulation of the electric arc furnace (eaf). *steel research international*, 89(1): 1700098.
- Rios, L. M. e Sahinidis, N. V. (2013). Derivative-free optimization: A review of algorithms and comparison of software implementations. *Journal of Global Optimization*, 56(3):1247–1293. ISSN 09255001.
- Takenaga, S., Ozaki, Y., e Onishi, M. (2023). Practical initialization of the nelder–mead method for computationally expensive optimization problems. *Optimization Letters*, 17(2):283–297.