

Technical note: Relating to the parameter values given by Nelder and Mead in their algorithm

Albert COROMINAS, Alberto GARCÍA-VILLORIA* and Rafael PASTOR

Institute of Industrial and Control Engineering (IOC), Universitat Politècnica de Catalunya (UPC),

Barcelona 08028, Spain

{albert.corominas / alberto.garcia-villoria / rafael.pastor}@upc.edu

Abstract. The popular Nelder and Mead algorithm (NM) has four parameters associated to the operations known as reflection, expansion, contraction and shrinkage. The authors set their values to 1, 2, 0.5 and 0.5, respectively, which have been universally used. Here we propose to use NM to calibrate itself. A computational experiment is carried out and results show that the parameter values originally proposed by Nelder and Mead are better than the ones obtained with more sophisticated ways.

Keywords: Nelder and Mead algorithm, parameter calibration

1. Introduction

The Nelder & Mead algorithm [1], NM, is the most widely used direct search method for solving unconstrained optimisation problems [2]. It is specially used in chemical engineering, chemistry, and medicine [3]. Although NM may converge to a non-stationary point (e.g., [3]), the algorithm usually performs well and remains as a popular search method [4].

NM has four scalar parameters, α , γ , β and δ , which are associated with the operations *reflection*, *expansion*, *contraction* and *shrinkage*, respectively. The NM authors, at the beginning of the sixties, set the parameter values to 1, 2, 0.5 and 0.5, respectively, after a short experiment. Since then, these values have been virtually

* Corresponding author: Alberto García-Villoria, Institute of Industrial and Control Engineering (IOC), Av. Diagonal 647 (Edif. ETSEIB), 11th floor, 08028 Barcelona, Spain; Tel.: +34 93 4010724; E-mail: alberto.garcia-villoria@upc.edu

always used ([2], [5]). Gao and Han [2] wonder how to choose the most suitable parameter values. Here we propose to use NM itself in order to try to find better parameter values.

The structure of this paper is as follows. In section 2 we describe the tested NM algorithm. In section 3 we propose two ways for setting the NM parameters. In section 4 we show the results of the computational experiment. The final remarks are given in section 5.

2. The Nelder & Mead algorithm

The NM algorithm is a direct search method for minimising $f(x)$ where $f: \mathbb{R}^n \rightarrow \mathbb{R}$ is the objective function and n the dimension. It is based on $n+1$ points that are the vertices of a simplex in the n -dimensional space: x_1, x_2, \dots, x_{n+1} . The algorithm adopted here is explained next.

NM starts from an initial simplex (usually regular) and iteratively moves the vertices over the n -dimensional space according to their objective function values until a stopping criterion is reached.

At each iteration of NM the vertices of the simplex are labelled and ordered such that $f(x_1) \leq f(x_2) \leq \dots \leq f(x_{n+1})$. In case of tie, the oldest vertex has priority. Let $x_r = \bar{x} + \alpha(\bar{x} - x_{n+1})$ be the reflection of x_{n+1} , where \bar{x} is the centroid of the n best vertices (i.e., $\bar{x} = \sum_{i=1}^n x_i / n$) and $\alpha > 0$ is a parameter. Four cases are considered according to the $f(x_r)$ value:

- 1) *Expansion*. If $f(x_r) < f(x_1)$ then calculate $x_e = \bar{x} + \gamma(x_r - \bar{x})$, where $\gamma > 1$ is a parameter. If $f(x_e) < f(x_1)$, replace x_{n+1} with x_e ; otherwise, replace x_{n+1} with x_r .
- 2) *Reflection*. If $f(x_1) \leq f(x_r) < f(x_n)$ then replace x_{n+1} with x_r .
- 3) *Outside contraction*. If $f(x_n) \leq f(x_r) < f(x_{n+1})$ then calculate $x_{oc} = \bar{x} + \beta(x_r - \bar{x})$, where $0 < \beta < 1$ is a parameter. If $f(x_{oc}) < f(x_r)$, replace x_{n+1} with x_{oc} ; otherwise, replace x_{n+1} with x_r and shrink all vertices except x_1 : $x_i = x_1 + \delta(x_i - x_1)$ $i = 2, \dots, n+1$, where $0 < \delta < 1$ is a parameter.
- 4) *Inside contraction*. If $f(x_{n+1}) \leq f(x_r)$ then calculate $x_{ic} = \bar{x} + \beta(x_{n+1} - \bar{x})$. If $f(x_{ic}) < f(x_{n+1})$, replace x_{n+1} with x_{ic} ; otherwise, shrink all vertices except x_1 as in 3).

NM stops when conditions $S1$ and $S2$ are met: $S1 = f(x_{n+1}) - f(x_1) \leq \varepsilon \cdot |f(x_1)|$ and $S2 = \max(\alpha \cdot \gamma, 1)^{ne} \cdot \delta^{ns} \cdot L \leq \varepsilon$, where $\varepsilon = 10^{-8}$, ne is the number of *expansions* (i.e., the number of iterations in which x_{n+1} is replaced with x_e), ns is the number of *shrinkages* (i.e., the number of iterations in which the vertices are shrunk) and L is the length of the edges of the initial simplex (that we assume regular). That is, NM stops when the differences between the values of the vertices are small enough (condition $S1$) and the simplex is small enough (condition $S2$).

3. Calibration of the NM parameters

The problem of calibrating the parameters can be approached as an optimisation problem, in which the objective function to optimise is the *performance* of NM. Let

$NM_f(A, \Gamma, B, \Delta)$ be the point returned by NM when minimising function f using the parameter values A , Γ , B and Δ , respectively. We define the NM performance function $\varphi_f(A, \Gamma, B, \Delta) = f(NM_f(A, \Gamma, B, \Delta))$; in case of tie, the lower number of iterations has priority. A direct search method is needed to minimise φ_f ; in particular, we propose to use NM with the standard parameter values 1, 2, 0.5 and 0.5.

We evaluate the NM performance in the family of functions $f(x_1, \dots, x_N) = \sum_{i=1}^N \lambda_i \cdot |x_i|^{\theta_i}$, where $1 \leq N \leq 25$, $10^{-6} \leq \lambda_i \leq 1$ and $0 \leq \theta_i \leq 10$. Note that the minimum point is always $x^* = 0$ and $f(x^*) = 0$. We use a training set, TSF , to carry out the optimisation of the NM parameters, which contains 100 functions obtained as follows: N and λ_i values generated at random with a uniform distribution between their lower and upper values; θ_i values are drawn with equal probability either from the distribution $U[0,1]$ or from $U[1,10]$.

We propose two ways for calibrating the parameter values, $C1$ and $C2$:

$$C1. (\alpha, \gamma, \beta, \delta) = \sum_{f \in TSF} NM_{\varphi_f}(1, 2, 0.5, 0.5) / |TSF|.$$

$$C2. (\alpha, \gamma, \beta, \delta) = NM_{\varphi_F}(1, 2, 0.5, 0.5), \text{ where } F(x) = \sum_{f \in TSF} f(x) / |TSF|.$$

The initial regular simplex in the NM parameter space of both $C1$ and $C2$ is $[(1, 2, 0.5, 0.5), (0.9, 2, 0.5, 0.5), (0.95, 1.9134, 0.5, 0.5), (0.95, 1.9711, 0.5816, 0.5), (0.95, 1.9711, 0.5204, 0.5791)]$. Note that it is generated from the point $(1, 2, 0.5, 0.5)$ and the length of the edges is equal to 0.1.

To calibrate the parameters, a variant of NM is used in order to handle the constraints associated to the lower and upper values of α , γ , β and δ ($\alpha > 0$, $\gamma > 1$, $0 < \beta < 1$ and $0 < \delta < 1$). Inspired by [6] the value of an unfeasible point $(\alpha, \gamma, \beta, \delta)$ is equal to $\varphi_{worst} + NU$, where φ_{worst} is the function value of the worst feasible vertex (at the current iteration) and NU is the number of constraint violations. The idea is that the vertex with the highest number of constraint violations will be identified as the worst vertex and NM will try to move away from that vertex.

4. Computational experiment

The NM parameter values obtained with calibrations *C1* and *C2* are shown in Table 1.

| | α | γ | β | δ |
|------------------|----------|----------|---------|----------|
| Standard | 1 | 2 | 0.5 | 0.5 |
| <i>C1</i> | 0.9424 | 2.1061 | 0.5026 | 0.5114 |
| <i>C2</i> | 0.9000 | 2.2222 | 0.5000 | 0.5000 |

Table 1. NM parameter values

We can observe that the obtained parameter values are not very different from the standard values. To test these parameter values, we minimised 900 test functions generated as explained in Section 3 (the complete sets of training and test instances are available at <https://www.ioc.upc.edu/EOLI/research/>). Table 2 shows the results obtained when NM is run with the standard, *C1* and *C2* parameter values; specifically, the average function value of the point returned by NM (\bar{f}) and the average number of function evaluations ($\#eval$). In all cases, the length of the edges of the initial simplex is equal to 200 with the starting vertex $(-100, \dots, -100)$, and generated so the point 0 is located close to the centre of the simplex.

| | \bar{f} | <i>#eval</i> |
|-----------------|-----------|--------------|
| Standard | 30.3573 | 12583.63 |
| <i>C1</i> | 34.0910 | 13481.56 |
| <i>C2</i> | 114.1234 | 317996.73 |

Table 2. NM results

Although the optimized functions are smooth, NM does not usually converge to the optimum as is well known in the literature (e.g., [3]); specifically, it does not converge 92.56%, 96.00% and 93.67% of times when NM is run with the standard calibration, *C1* and *C2*, respectively. The results show two points. First, NM is very sensitive to the parameter values in terms of the quality of the solutions and in the number of function evaluations. And second and as the main result of this work, the standard parameter values given by Nelder and Mead seem to be an inspired proposal and perform better than seemingly smarter calibrations.

5. Conclusions

The Nelder & Mead algorithm is based on the reflection, expansion, contraction and shrink operations, each of them being associated with a parameter. Its authors set their values which are being used in almost all implementations of the algorithm. One question that arises is whether the algorithm performs better with other parameter values. In this work we propose two ways to use the Nelder and Mead algorithm for calibrating itself. Finally, the widely used parameters originally proposed by the authors are clearly better than the ones obtained with more sophisticated ways.

References

- [1] Nelder, J.A. and Mead, R. (1965) A simplex method for function minimization. *Comp. J.*, **7**, 308-313.

- [2] Gao, F. and Han, L. (2012) Implementing the Nelder-Mead simplex algorithm with adaptive parameters. *Comput. Optim. Appl.*, **51**, 259-277.
- [3] Mckinnon, K.I.M. (1998) Convergence of the Nelder-Mead simplex method to a nonstationary point. *SIAM J. Optim.*, **9**, 148-158.
- [4] Kolda, T.G., Lewis, R.M. and Torczon, V. (2003) Optimization by direct search: new perspectives on some classical and modern methods. *SIAM Rev.*, **45**, 385-482.
- [5] Lagarias, J.C., Reeds, J.A., Wright, M.H. and Wright, P.E. (1998) Convergence Properties of the Nelder-Mead Simplex Method in Low Dimensions. *SIAM J. Optim.*, **9**, 112-147.
- [6] Mehta, V.K. and Dasgupta, B. (2012) A constrained optimization algorithm based on the simplex search method. *Eng. Optimiz.*, **44**, 537-550.