

Article

Not peer-reviewed version

Parallel Simplex: An Alternative to Classical Experimentation: Case Study

[Francisco Zorrilla Briones](#)*, Inocente Yuliana Meléndez Pastrana, [Manuel Alonso Rodríguez-Morachis](#),
José Luís Anaya Carrasco

Posted Date: 19 September 2024

doi: 10.20944/preprints202409.1367.v1

Keywords: optimization; design of experiments; variable simplex; evolutive operations; canonical analysis; RSM



Preprints.org is a free multidiscipline platform providing preprint service that is dedicated to making early versions of research outputs permanently available and citable. Preprints posted at Preprints.org appear in Web of Science, Crossref, Google Scholar, Scilit, Europe PMC.

Copyright: This is an open access article distributed under the Creative Commons Attribution License which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Article

Parallel Simplex: An Alternative to Classical Experimentation: Case Study

Francisco Zorrilla Briones ¹, Inocente Yuliana Meléndez Pastrana ²,
Manuel Alonso Rodríguez Morachis ¹ and José Luís Anaya Carrasco ¹

¹ Tecnológico Nacional de México/I.T. de Ciudad Juárez; División de Estudios de Posgrado e Investigación; mmorachis@itcj.edu.mx (M.A.R.M.); janaya@itcj.edu.mx (J.L.A.C.)

² Tecnológico Nacional de México/I.T. de Ciudad Juárez; División de Estudios de Posgrado e Investigación; Universidad Tecnológica de Ciudad Juárez/Departamento de Posgrado;; yuliana.mp@itcj.edu.mx

* Correspondence: fzorrilla@itcj.edu.mx

Abstract: Experimentation is a strong methodology that improves and optimizes processes, never the less, in many cases, real life dynamics of production demands and other restrictions inhibit the use of these methodologies because their use implies stopping production, generate scrap, jeopardize demands accomplishment and other problems. Here is proposed and alternative methodology to search for the best process variables levels and optimize the response of the process without the need to stop production. This algorithm is based on the principles of the Variable Simplex developed by Nelder and Mead and the continuous iterative process of EVOPS developed by Box and then modified as a simplex by Spendley. It is named parallel simplex because it searches for the best response with three independent Simplexes, searching the same response at the same time. The algorithm was designed for three simplexes of two input variables each. The case study documented shows that it is efficient and effective.

Keywords: optimization; design of experiments; variable simplex; evolutive operations; canonical analysis; RSM

1. Summary

Traditional experimentation requires designing an experimental array, stopping production, setting up the equipment/machinery for each run, and running all the runs of the array. This process inevitably results in lost production time, scrap, and the use of extraordinary resources. In many real manufacturing scenarios, this is difficult to do due to the complexity of production demand trade-offs and manufacturing process dynamics. The main idea of Box's Evolutive Operations [1], Spendley and Himsworth Simplex [2], and Nelder and Mead's Variable Simplex [3] is to find the best combination of process parameters by introducing small changes in the process variables. These small changes allow the experimenter to run the process continuously because the response variable does not move dramatically away from its target, thereby eliminating or minimizing the generation of nonconforming parts (scrap).

These direct search methods (EVOPS, Simplex, Variable Simplex, etc.) have some disadvantages that have limited their use in a manufacturing process:

- They are very sensitive to noise (internal noise).
- The complexity of the computation increases dramatically with the number of variables considered.
- The number of iterations to find a local optimum grows exponentially with the number of variables in the algorithm.

These drawbacks make these methods not very practical to apply to a manufacturing process.

On the other hand, these procedures are based on heuristic/stochastic algorithms, i.e., a series of iterations is developed by introducing small changes in the input variables and evaluating the observed response. The process continues until the response is located on the desired nominal value.

The use of logic algorithms is widely used in the engineering field, from genetic algorithms applied to analyze and improve the location, assignment and routing of transportation systems [4], to many other applications; these authors documented another few heuristic and stochastic methods and their objectives. It can be seen that there are many designs of algorithms, mainly, they can be classified as

- Deterministic. Those that work with a known function
- Heuristic: Algorithms based on test and error procedures.
- Stochastic: They use successive iterations, taking into account changes in the input variables and measuring the change observed in the output variables, with the help of probabilistic theories.

One of the algorithms that seem to work like the Nelder and Mead algorithm presented in this paper is the Genetic Algorithm, widely used to analyze and improve many different systems. These algorithms evolve a population of individuals by subjecting it to random actions, similar to those that act in biological evolution (mutations and genetic recombination), as well as natural selection. According to some criterion, it is decided which are the most adapted individuals, which survive, and which are the least adapted, which are discarded. An algorithm is a set of organized steps that describe the process to be followed to solve a particular problem, such as an experimental design for optimizing process parameters. There is extensive literature on these algorithm applications, [5–8].

The problem with the Genetic Algorithm is that it requires a declared function of the system to be optimized, making it a deterministic algorithm. The direct search methods are better suited to purely heuristic situations where the system function is unknown, such as a production process or machine situations. In these cases, the logical way to improve the operating parameters is to introduce small changes in the input variables and evaluate the changes in the response or responses.

An exhaustive review of the literature shows evidence of the loss of efficiency of these direct search algorithms when the number of dimensions (variables) is increased, as an example, [9] compares different algorithms (heuristic and deterministic) and their efficiency; our attention is focused on the number of runs required to obtain a local or global optimum. As shown in Table 1, the number of runs (iterations of the algorithm) increases dramatically when the number of dimensions (variables) is increased (n).

Table 1. Comparison of Different Algorithms for Differentiable Functions [9].

Function			VNS + NM			VNS + RNM			RMNME			RMNM		
Name	n	F _{opt}	F _{eval}	succ.	F _{min}	F _{eval}	succ.	F _{min}	F _{eval}	succ.	F _{min}	F _{eval}	succ.	F _{min}
S1. Branin	2	0.3979	72	10		75	10		72	10		72	10	
S2. Goldstein and Price	2	3	150	10		211	10		109	10		109	10	
S3. Hartman H34	3	-3.8626	245	10		367	10		702	10		123	4	-2.1453
S4. Hartman H64	6	-3.3224	1019	10		1375	10		1,547	8	-3.2744	590	5	-2.9421
S5. Shekel 4, 10	4	-10.5364	1367	10		3131	10		1,793	6	-7.3679	222	2	-4.2124
S6. Shekel 4, 5	4	-10.1532	1035	10		1318	10		1,487	5	-6.4069	270	5	-6.4069
S7. Shubert	2	-186.7309	1071	10		1218	10		2,228	10		73	0	-23.378
S8. Ackley	10	0	188,726	10		324,405	10		35,706	0	0.8977	880	0	12.47
S9. Dixon and Price	10	0	5273	10		4752	10		5,138	10		5,138	10	
S10. Griewank	6	0	148,528	10		161,096	10		17,285	0	0.2108	601	0	53.18
S11. Rosenbrock	10	0	9738	10		7917	10		8,482	8	0.7974	6,750	8	0.7974
S12. Schwefel	10	0	1,173,568	10		1,262,060	10		16,426	0	1789.88	2,535	0	1818.45
S13. Zakharov	10	0	4056	10		4218	10		4,508	10		4,508	10	
S14. Rastrigin	10	0	130,802	10		222,881	10		32,184	0	7.5618	1,147	0	88.75
S15. Powell	12	0	6092	10		5977	10		6,349	10		6,237	10	
Sum			1,671,742	150		2,002,001	150		134,016	97		29,255	74	

As another example, [10] compared the NMS algorithm with 5 typical problems. Table 2 shows the comparison of results under different sample sizes and number of variables for five typical problems; it can be seen that the number of iterations increases dramatically as the number of variables in the geometric array (dimensions) increases.

To give the reader a better idea, these algorithms are essentially different from those used to make decisions with multiple objectives, such as those used as multicriteria. To understand these, you can see [11–13] and others.

Table 2. Efficiency of NMS of Five Typical Problems [10].

Problem	Dimension	Efficiency Parameters					
		Step Size	Iterations	<i>L</i>	<i>D</i>	<i>B</i>	<i>A</i>
1	2	1	20	6.931	0.022	0.073	0.047
	10	4	250	9.66	0.325	0.195	0.042
	18	4	1250	11.340	0.797	0.096	0.023
	Average			9.31	0.381	0.121	0.037
	2	1	20	7.149	0.013	0.105	0.070
2	10	1	225	9.480	0.113	0.217	0.080
	18	4	600	10.534	0.510	0.210	0.077
	Average			9.054	0.212	0.177	0.076
	2	1	30	7.630	0.049	0.046	0.038
	10	4	595	10.579	2.461	0.963	0.543
3	18	4	2000	11.960	6.846	1.241	0.777
	Average			10.056	3.119	0.75	0.453
	4	1	80	8.357	0.113	0.153	0.075
	8	1	240	9.519	0.306	0.254	0.093
	16	4	1520	11.663	0.547	0.432	0.122
4	Average			9.846	0.322	0.280	0.097
	2	1	20	7.172	0.028	0.084	0.060
	10	4	170	9.112	0.222	0.310	0.106
	18	4	1360	11.546	0.367	0.342	0.111
	Average			9.277	0.206	0.245	0.092

The algorithm we discuss here is used to find the best combination of parameters for a process, whatever it may be, from a design, a machine, or a manufacturing process. It has to be considered that the increase in the number of iterations for these algorithms is directly related to the number of variables analyzed. If two variables are considered, the geometric array corresponds to a triangle, if three variables are considered, the array corresponds to a square, if four variables are considered, the geometric array corresponds to a pentagon, and so on. As the number of variables increases, so does the number of edges and vertices, which slows down the algorithm and increases the complexity of the computation. In many cases, the high number of variables leads to a non-convergence situation (the algorithm gets “lost”).

The proposed methodology tries to overcome exactly this situation, that is, to search for the optimal response (local or global) when the number of input variables increases, trying to minimize the number of iterations without the algorithm “getting lost”, that is, it converges quickly. Again, these algorithms, by introducing small changes in the input variables, allow them to be executed during the normal production process. Every process/manufacturing engineer knows the value of this.

2. Methods

Box [1] proposed an iterative experimentation process, basically his proposal consisted in the introduction of a two-level factorial design that is modified in search of the best response [1]. Small changes in the levels of the design are estimated to minimize the generation of nonconforming product. This way of managing the experimentation process is known as operative evolutions (EVOP). EVOP is mainly used in the 3rd phase of experimentation, optimization, after the process has been characterized.

On the other hand, [14], questions about the sample size, the Box algorithm does not consider whether the sample size is sufficient to generate acceptable statistical properties. It is clear that the

proposal of Box has many advantages and that it is innovative because it searches for the best response with small changes, during production, minimizing the nonconforming product.

Spendley [2], proposed a big change in the algorithm of Box, adding a simplex arrangement that replaces the factorial design [2]. It is through this simplex that the levels of each iteration are calculated. The algorithm of [2] is known as simplex EVOP, it iteratively moves the polyhedron (with $n+1$ vertices) looking for the optimization of the response. Here is a brief review of the operation of the variable simplex of Nelder and Mead [3].

In the classical array, the levels of the variables considered are pre-designed, that is, once the array is defined, these levels remain fixed during the process of experimentation. In the Nelder and Mead Variable Simplex (NMS), the algorithm starts with pre-designed levels, then these levels are modified by the iterations of the algorithm according to a set of rules (operations of the simplex). The algorithm consists in using a simplex (a geometric arrangement), generally a polyhedron with $n-1$ vertices (for two input variables it will be a triangle). The search mechanism consists in using one of the vertices as a pivot (called the worse vertex) to estimate the next vertices (variable levels), always moving towards the best vertex (the one closest to the response objective: maximum, minimum, target). The algorithm moves through the response, calculating new vertexes in each iteration, through four basic operations: Reflection, Expansion, Contraction, and Shrinkage; Figure 1 illustrates these operations.

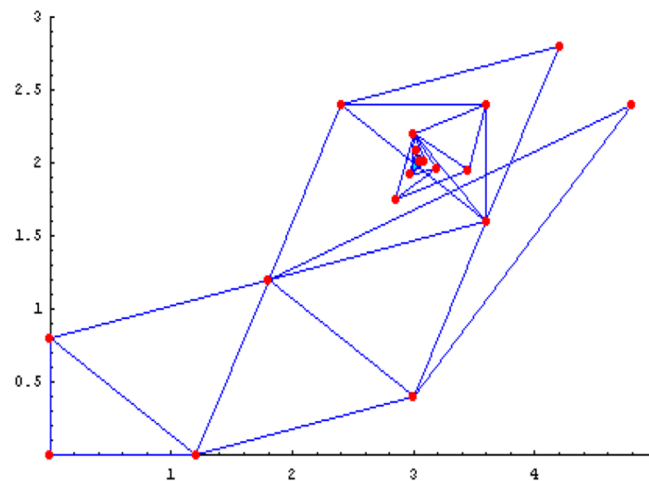


Figure 1. Operations on the NMS: Reflection, Expansion, Contraction and Shrinkage.

There is plenty of literature available to understand the logistics of variable simplex operations, in this section a brief explanation of the logistics of the algorithm of NMS is resumed.

2.1. Basic Operations of the Nelder and Mead Algorithm

For the first iteration, $k=0$, the algorithm needs an initial simplex $S_0 = \{x_0^0, x_1^0, \dots, x_n^0\}$ because it prefers to start with a regular simplex. This characteristic changes immediately, since the successive iterations replace the worst vertex, x_h^k , of the S_k -th simplex with another vertex. This worse vertex will be replaced by one of the new points (vertexes) x_r^k , x_e^k and x_c^k . The selection of one of these three points, at any $k \geq 0$ iteration, is defined by one of the three operations of the algorithm: Reflection, Expansion, Contraction.

2.1.1. Reflection

Considering that $f_i > f_r > f_l$ for at least one i , where $i \neq h$ or $i \neq l$; α is the coefficient of reflection. Figure 2 shows the reflection operation in two dimensions (two variables of the process) for the $n=2$ case. The distance between the points x_h^k and x^k is equal to the distance between $\bar{x}^k$ and x_r^k . This is obtained when $\alpha=1$.

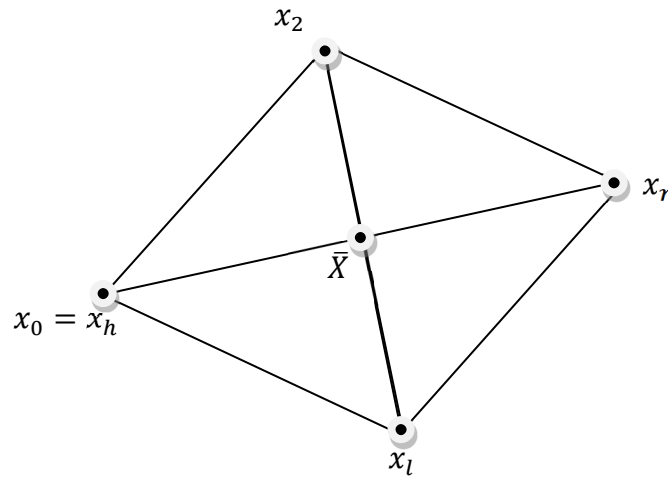


Figure 2. Reflection Operation.

2.1.2. Expansion

The worse vertex x_h^k is replaced by:

$$x_e^k = \gamma x_r^k + (1 - \gamma) \bar{x}^k \quad (1)$$

Considering that (i) $f_r < f_e$, and (ii) $f_e < f_l$. If (i) is kept but (ii) no, then the worse vertex x_h^k is replaced by x_r^k . γ is the expansion coefficient. Figure 3 shows the expansion operation for two dimensions, $n=2$. The distance between the points x_h and $\bar{x}$ is equivalent to the distances between the points $\bar{x}$ and x_r , and x_e and x_r . This is obtained for a $\gamma=2$.

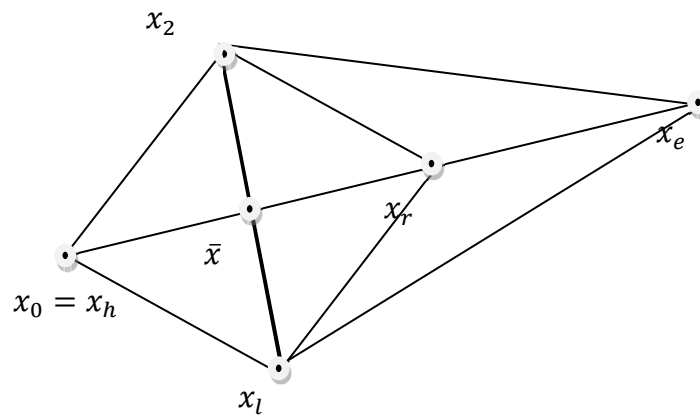


Figure 3. Expansion Operation.

2.1.3. Contraction

In the contraction operation, the worse vertex x_h^k is replaced by:

$$x_c^k = \beta x_h^k + (1 - \beta) \bar{x}^k \quad (2)$$

Considering that $f_r \geq f_h$, or:

$$x_c^k = \beta x_r^k + (1 - \beta) \bar{x}^k \quad (3)$$

When $f_r < f_h$; β is the coefficient of contraction. In **Figure 4** the operation of contraction is shown for two dimensions $n=2$. The distance between the points x_h and $\bar{x}$ corresponds to two times the distances between $\bar{x}$ and x_{c1} , or $\bar{x}$ and x_{c2} . This is obtained for a $\beta = 1/2$.

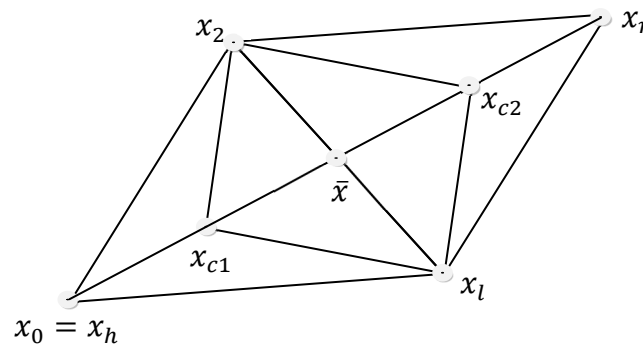


Figure 4. Contraction Operation.

In the case studied and presented in this paper, they were considered six variables, so the geometrical array is to be a heptagon, if the traditional NMS algorithm was used. As an alternative to these mathematical and practical problems the named Parallel Simplex is proposed.

3. The Parallel Simplex Proposal

This document concentrates on the proposed alternative of the parallel simplex. On **Figure 5**, it is shown the movements of the simplex (geometrical array), with the basic operations of Reflection, Expansion and Contraction, always moving towards the objective of the response variable. As shown on **Figure 5**, the parallel simplex approaches towards the objective of the response variable with three simplexes, independent between them, but pursuing the same response and objective. After an exhaustive literature review, nothing similar was found. Many authors have tried to minimize the problems of non-convergency and the high number of iterations adding or modifying the original Nelder Mead simplex, never the less, the results are very questionable because of:

- The complexity of the algorithm still increases with the number of input variables
- Almost all the evidence found is generated through known test functions (thus, deterministic approximations)
- Almost all the cases presented are computer simulated situations

Maintaining the three simplexes independent between them, assures the efficiency of the algorithm, avoiding the increase of the complexity of the calculus, thus reducing the number of iterations, as it will happen if the traditional array of a heptagon ($n+1$ dimensions) is used. Because the three simplexes chase the same objective, the simplex operations are the same for every single simplex.

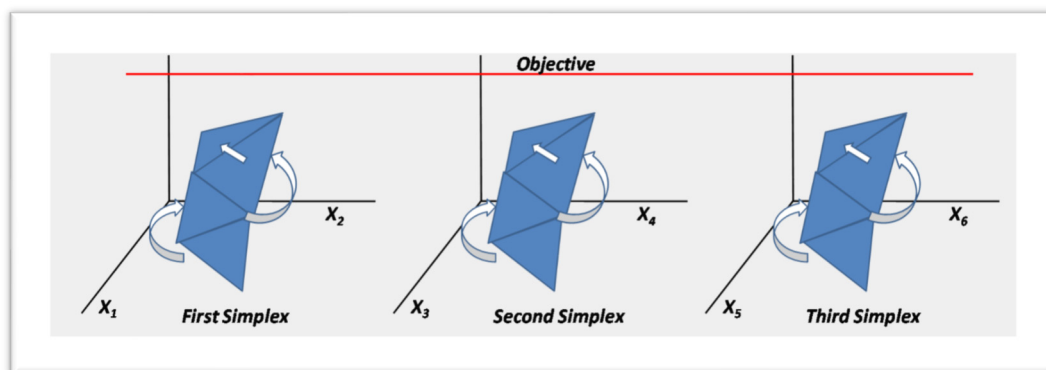


Figure 5. The Parallel Simplex Array for 6 Control Variables.

4. The Analyzed Process

The process analyzed corresponds to the manufacturing of plastic profiles, to be applied as seals for refrigerators, washing machines and other domestic equipment. Plastic is melted through a simple plasticizing screw and extruded through a nozzle with the designed profile. On **Figure 6** the selected profile is shown. This profile was chosen because it presented a high rejection rate, actually, 100% rejection of lots. 100% inspection and rework were used on a regular basis.



Figure 6. Plastic Seal Profile.

This seal showed a very high rate of defects, mainly, incomplete shots, on **Figure 7** it is shown this defect.



Figure 7. Incomplete Shots.

The main control variables of this process are the cubic feet per minute of air (at a constant 4 Bar pressure) injected by four nozzles, this is done to keep the profile of the extrusion, as shown on **Figure 8**. The fifth variable is the height of the puller. This height is the distance between the rollers that pull the extruded material. The sixth variable is the speed of the extruder and the puller (that are synchronized). The output variable is measured as defects per 1,000 parts as λ (The continuous extruded material is cut to the design length).

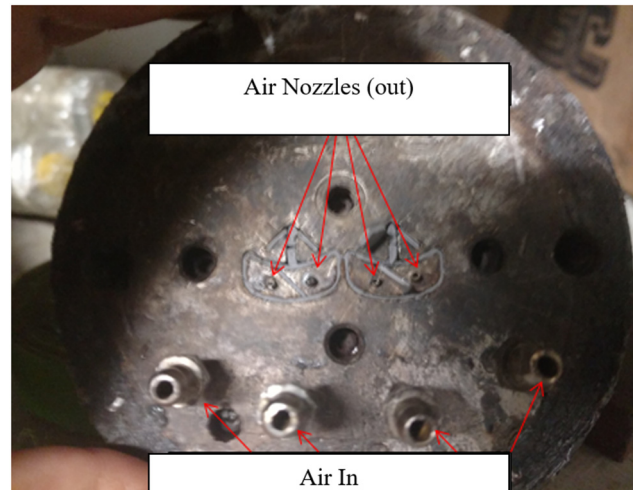


Figure 8. Air Nozzles on the Mold.

At this point is very important to consider that, if a traditional/classic array is used, let's say, a two-level factorial design, it will imply:

- The design, set up and run of 2^6 runs, that is 64 runs
- If more replicates are used for better fit estimations, the number of runs will increase
- The majority of product will not be conformant to specifications
- Production will be lost because the process must stop (to set up each run)
- Extra resources (time, Technicians, operators, materials, etc.) will be needed

The principle of Evolutive Operations of these direct search algorithms, as the NMS, allows the investigator to search for the best response while the process is running. The small changes induced in the process allows to reduce or even eliminate the generation of non-conformant parts.

4.1. The Experimental Array

The Experimental Array

A specific software was developed to run the Parallel Simplex, named Armentum®. The original algorithm of Nelder and Mead was modified to add a response known as dual response or simultaneous optimization, [15]. This way of evaluating the response considers the natural response, the objective pursued and the variation in the process at the same time, increasing the efficiency of the search. The expected value of the response variable is considered as:

$$E(Y - m)^2 = [\mu - m]^2 + \sigma^2 \quad (4)$$

Leading to a practical standard, considering 3 standard deviations for a normal distributed process:

$$R_d = |\bar{y} - 0| + 3S_x \quad (5)$$

Where R_d is the dual response, $\bar{y}$ is the average of the runs (in the same iteration), and S_x is the standard deviation of the runs. In this case the objective of the response variable y is zero defects per 1,000 parts.

The control variables and the initial vertexes of the simplexes are set up in the software. This is shown on **Figure 9**.

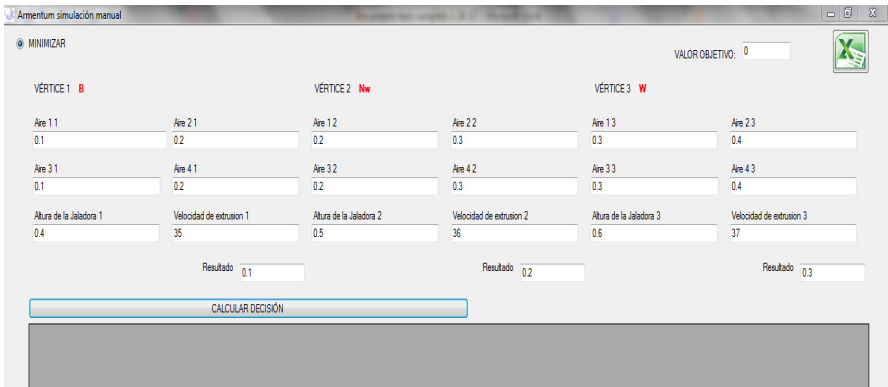


Figure 9. Initial Vertexes of the Parallel Simplex.

The first set of geometrical arrays; three triangles, 3 vertexes each one, are set up; 3 combinations for each pair of variables, X_1 and X_1 ; X_3 and X_4 ; X_5 and X_6 . The process is started and the feedback of the response is input to the software. The software calculates, on every new iteration, a new combination (vertex) for each simplex (in classical experimentation, these combinations -runs- are determined in advance, luckily some runs will be useful to estimate changes in the response, many other will not). Table 3 shows the final results of the experimentation process.

The NMS algorithm categorizes the response as Best (B), Near the Worse (Nw) and Worse (W), considering the distance of the response from the objective (in this case a minimum of cero is intended), this is documented as the Ranking of the response. There are a lot of stop criteria of the algorithm on the literature, never the less, the Teta parameter developed [10] was included because its simplicity and accuracy.

$$\sum_{i=1}^{k+1} \frac{(y_i - \bar{y})^2}{k + 1} \leq \theta$$

(6)

This is a comparison of the standard error of the response against a predetermined value, typically $\theta=1\times10^{-6}$, for test functions. In real practical processes, a value of Teta of $\theta=1\times10^{-3}$ is more than acceptable.

Table 3. Results of the Parallel Simplex Process.

Armentum® Table Results										
Vertex No.	Simplex No.	Air 1	Air 2	Air 3	Air 4	Puller Height	Extrusion Speed	RD	Ranking	TETA
1	1	0,2	0,18	0,18	0,22	0,69	22,4	0,005	B	0
2	1	0,3	0,28	0,25	0,32	0,72	24	0,085	Nw	0
3	1	0,15	0,15	0,15	0,17	0,65	21	0,13	W	0,002672
4	1	0,35	0,31	0,28	0,37	0,76	25,4	0,5	R	0,047072
5	1	0,3	0,27	0,2475	0,32	0,6775	22,1	0,099	Cw	0,001715
6	2	0,2	0,17	0,1775	0,22	0,6475	20,5	0,05	R	0,001474
7	3	0,1	0,08	0,11	0,12	0,66	20,8	0,075	R	0,000839
8	3	0,15	0,13	0,14	0,17	0,66	21,125	0,065	Cr	0,00065
9	4	0,25	0,22	0,21	0,27	0,65	20,825	0,06	R	3,89E-05
10	5	0,30	0,26	0,25	0,32	0,64	20,2	0,09	R	0,000289
11	5	0,26	0,23	0,22	0,28	0,66	20,89375	0,105	Cw	0,000572
12	6	0,21	0,18	0,19	0,23	0,65	20,56875	0,006	R	0,00164
13	6	0,19	0,16	0,17	0,21	0,65	20,440625	0,005	E	0,001672
14	7	0,13	0,10	0,13	0,15	0,64	20,046875	0,06	R	0,000572
15	7	0,16	0,13	0,15	0,18	0,65	20,25859375	0,056	Cr	0,000518
16	8	0,23	0,20	0,20	0,25	0,65	20,71171875	0,003	R	0,000562
17	8	0,28	0,25	0,24	0,30	0,65	21,04414063	0,101	E	0,000518
18	9	0,27	0,24	0,23	0,29	0,65	20,953125	0,025	R	0,000369
19	10	0,30	0,27	0,25	0,32	0,65	21,16484375	0,106	R	0,001962
20	10	0,28	0,24	0,23	0,30	0,65	20,66621094	0,096	Cw	0,001575
21	11	0,24	0,21	0,20	0,26	0,65	20,42480469	0,06	R	0,001466
22	12	0,20	0,17	0,17	0,22	0,65	20,4703125	0,002	R	0,000735
23	12	0,16	0,13	0,15	0,18	0,65	20,37236328	0,06	E	0,000722
24	13	0,19	0,16	0,17	0,21	0,652416992	20,75722656	0,002	R	2,22E-07

5. Results Analysis

On **Table 3** it is shown that the algorithm stopped at the 24th iteration, obtaining a desired 0.002 on the dual response and a Teta value of 2.22×10^{-7} . Before any other comparison, a two proportions analysis was made to compare the process with the new set up. Historical data of three months against lots of 1,000 pieces were utilized. The analysis showed a p-value of 0.0000 for the probability of the alfa error, for the hypothesis that $H_0: P_{initial} = P_{final}$; thus, the difference is statistically significant.

Utilizing Minitab® it can be seen that, for a full factorial design needed to estimate a Response Surface, 90 runs are required for six variables (with the minimum of central, axial and radial points), and, at minimum, 53 runs are required for a half design.

These runs were not executed because it implied to many runs, to much material, to many scrap, etcetera; never the less, data of the Parallel Simplex results were utilized to feed the Response Surface array and compare the results. This comparison is shown on **Table 4**.

Table 4. Parallel Simplex Vs Response Surface Comparison.

Variable	Parallel-Simplex	Response Surface
x_1 (Air 1)	0.19	0.2059
x_2 (Air 2)	0.16	0.1768
x_3 (Air 3)	0.17	0.1813
x_4 (Air 4)	0.21	(Out of the analysis due to lack of effect)
x_5 (Puller Height)	0.65	0.6514
x_6 (Extrusion Speed)	20.75	20.66

The internal operation of the Parallel Simplex is based on the continuous iteration of three independent simplexes, with two variables each in this case. As a further comparison, a Response Surface analysis was made for each pair of variables (simplex), utilizing the algorithm results (again, these runs were not executed). A full model with principal and interactions effects were designed and analyzed.

$$f(x) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_1^2 + \beta_4 x_2^2 + \beta_5 x_1 x_2 + e_{ij} \quad (7)$$

As part of the comparison, a canonical analysis was made to assure that the optimum found with the Poly Simplex algorithm corresponds to a local minimum. The regression analysis for the first simplex resulted on the equation:

$$R_d = 0.742 - 16.93x_1 + 10.90x_2 + 126x_1^2 + 76x_2^2 - 190x_1x_2 \quad (8)$$

With an adjusted $R^2 = 71.55\%$ and p-values for x_1 and x_2 of 0.000 and 0.003

The stationary point estimated for this pair of variables is:

Considering matrix B and b

$$B = \begin{bmatrix} 126 & -95 \\ -95 & 76 \end{bmatrix} \text{ and } b = \begin{bmatrix} -16.93 \\ 10.90 \end{bmatrix}$$

Thus

$$x_s = -\frac{1}{2}B^{-1}b = \begin{bmatrix} 0.227931 \\ 0.213203 \end{bmatrix}; \text{ the canonical conversion shows that}$$

$$\lambda = \begin{bmatrix} 199.234 \\ 2.765 \end{bmatrix} \text{ for } x_1 \text{ and } x_2 \text{ as both roots are positive, this stationary point is a minimum.}$$

The regression analysis for the second simplex resulted on the equation:

$$R_d = 0.075 + 12.4x_3 - 10.9x_4 + 454x_3^2 + 343x_4^2 - 788x_3x_4 \quad (9)$$

With an adjusted $R^2 = 75.22\%$ and p-values for x_3 and x_4 of 0.000 and 0.723

The stationary point estimated for this pair of variables is:

Considering matrix B and b

$$B = \begin{bmatrix} 454 & -394 \\ -394 & 343 \end{bmatrix} \text{ and } b = \begin{bmatrix} 12.40 \\ -10.90 \end{bmatrix}$$

Thus

$$x_s = -\frac{1}{2}B^{-1}b = \begin{bmatrix} 0.042593 \\ 0.064815 \end{bmatrix};$$
 the canonical conversion shows that

$$\lambda = \begin{bmatrix} 796.390 \\ 0.610 \end{bmatrix}$$
 for x_3 and x_4 as both roots are positive, this stationary point is a minimum.

The canonical analysis demonstrates that the final combinations of each pair of variables is a local minimum. On **Table 5** is shown a resume of the regression coefficients, for the six variables at the same time and for each pair of variables.

Table 5. Regression Coefficients for Each Model.

Model	X_1, X_2	X_3, X_4	X_5, X_6	Global ($X_1, X_2, X_3, X_4, X_5, X_6$)
Adjusted R ²	71.55%	75.22%	82.52%	93.28
Stationary Point	Minimum	Minimum	Minimum	Saddle

6. Conclusions

The proposed algorithm has demonstrated its efficiency by finding the global minimum, for the six variables included in the model, in only 24 iterations. A traditional model using the original Nelder and Mead algorithm, which requires a heptagon for the analysis, would require a very large number of iterations, and, as already demonstrated, a traditional factorial design would also require a considerable number of experimental runs. Among the most important contributions is the fact that this process was performed during a normal production run, production was not stopped, no extraordinary resources were used and the generation of nonconforming parts was minimal.

On the other hand, the canonical analysis shows that a local minimum for the response variable has been found for every pair of variables. It is worth mentioning that with the new operating parameters, the process currently has a quality efficiency of 98%.

7. Discussion

The case study presented evidences that the Parallel Simplex is an efficient alternative to classical experimentation. It is an experimentation process that does not require to stop production, generate scrap and use additional resources.

For many years these Direct Search Algorithms were concealed to theoretical approaches, with test functions and so. One of the main reasons is because the mathematical complexity of the calculus and the sensibility of the algorithms to the internal noise, present in any industrial process. As a matter of fact, it is difficult to find literature of successful approaches on real processes, In addition, an approach such as the one proposed here was not found in the literature reviewed: three independent simplexes seeking to optimize the same response in a synchronous manner. A few examples can be revised on [16–30], among many others. This alternative is a powerful way to optimize processes, overcoming the restrictions of the original algorithms.

Applying the Parallel concept to a higher dimension (variables) needs to be developed to investigate the efficiency of the algorithm. Further investigations of this algorithm are being developed to accumulate evidence of its practical benefit to the average industrial Engineer.

Acknowledgments: “This work was published with the support of the Institute of Innovation and Competitiveness of the Ministry of Innovation and Economic Development of the State of Chihuahua, México.”. “Esta obra fue publicada con el apoyo del Instituto de Innovación y Competitividad de la Secretaría de Innovación y Desarrollo Económico del Estado de Chihuahua, México”.

Conflicts of Interest: “The authors declare no conflicts of interest.”

References

1. Box, G.E.P. (1957). Evolutionary Operation: A Method for Increasing Industrial Productivity. *Applied Statistics*, Vol. 6, No. 2, pp. 110-115.
2. Spendely, W., G.R., and Himsforth F.R. (1962). Sequential Applications of Simplex Designs in Optimization and EVOP. *Technometrics*, Vol. 4, Num. 4, pp. 441-461.
3. Nelder, J. A. and Mead, R. (1965). A Simplex Method for Function Minimization. *Computer Journal*, Vol. 7, pp. 308-313.
4. Mrad, M., Bamatraf, K., Alkahtani, M., & Hidri, L. (2023). A GENETIC ALGORITHM FOR THE INTEGRATED WAREHOUSE LOCATION, ALLOCATION AND VEHICLE ROUTING PROBLEM IN A POOLED TRANSPORTATION SYSTEM. *International Journal of Industrial Engineering: Theory, Applications and Practice*, 30 (3). <https://doi.org/10.23055/ijietap.2023.30.3.8989>
5. Vineetha, G. R., & Shiyas, C. R. (2023). MACHINE LEARNING-ENHANCED GENETIC ALGORITHM FOR ROBUST LAYOUT DESIGN IN DYNAMIC FACILITY LAYOUT PROBLEMS: Implementation of Dynamic Facility Layout Problems. *International Journal of Industrial Engineering: Theory, Applications and Practice*, 30 (6).
6. Xie, Y., Jiang, H., Wang, L., & Wang, C. (2023). SPORTS ECOTOURISM DEMAND PREDICTION USING IMPROVED FRUIT FLY OPTIMIZATION ALGORITHM. *International Journal of Industrial Engineering: Theory, Applications and Practice*, 30 (6). <https://doi.org/10.23055/ijietap.2023.30.6.8841>
7. Praga-Alejo, R. J. ., de León-Delgado, H., González-González, D. S., Cantú-Sifuentes, M., & Tahaei, A. (2022). MANUFACTURING PROCESSES MODELING USING MULTIVARIATE INFORMATION CRITERIA FOR RADIAL BASIS FUNCTION SELECTION. *International Journal of Industrial Engineering: Theory, Applications and Practice*, 29 (1). <https://doi.org/10.23055/ijietap.2022.29.1.6067>
8. Farajzadeh Bardeji, S., Fakoor Saghih, A. mohammad, & Mirghaderi, S.-H. (2022). MULTI-OBJECTIVE INVENTORY AND ROUTING MODEL FOR A MULTI-PRODUCT AND MULTI-PERIOD PROBLEM OF VETERINARY DRUGS. *International Journal of Industrial Engineering: Theory, Applications and Practice*, 29(4). <https://doi.org/10.23055/ijietap.2022.29.4.6711>
9. Drazic, Milan; Drazic, Zorica; Dragan, Urosevic (2014). Continuous variable neighborhood search with modified Nelder-Mead for non-differentiable optimization. *IMA Journal of Management Mathematics*. 27, pp. 75-78. Doi:10.1093/imaman/dpu012.
10. Fan, S.; Zahara E. (2006). Stochastic response surface optimization via an Enhanced Nelder Mead simplex search procedure. *Engineering Optimization*, 38:1, pp. 15-36.
11. Park, H., Shin, Y., & Moon, I. (2024). MULTIPLE-OBJECTIVE SCHEDULING FOR BATCH PROCESS SYSTEMS USING STOCHASTIC UTILITY EVALUATION. *International Journal of Industrial Engineering: Theory, Applications and Practice*, 31(2). <https://doi.org/10.23055/ijietap.2024.31.2.9889>
12. Uluskan, M., & Beki, B. (2024). PROJECT SELECTION REVISITED: CUSTOMIZED TYPE-2 FUZZY ORESTE APPROACH FOR PROJECT PRIORITIZATION. *International Journal of Industrial Engineering: Theory, Applications and Practice*, 31(2). <https://doi.org/10.23055/ijietap.2024.31.2.9853>
13. Sharma, J., Tyagi, M., & Bhardwaj, A. (2023). CONTEMPLATION OF OPERATIONS PERSPECTIVES ENCUMBERING THE PRODUCTION-CONSUMPTION AVENUES OF THE PROCESSED FOOD SUPPLY CHAIN. *International Journal of Industrial Engineering: Theory, Applications and Practice*, 30(1). <https://doi.org/10.23055/ijietap.2023.30.1.8027>
14. Ryan, T. P. (1989). Statistical Methods for Quality Improvement. John Wiley. New York.
15. Montgomery D.C. (1997). Design and Analysis of Experiments, 4th. Edition. Jhon Wiley & Sons. New York.
16. Kaelo, P.; Ali, M. (2006). Some Variants of the Controlled Random Search Algorithm for Global Optimization. *Journal of Optimization Theory and Applications*. Vol. 130. No. 2, pp. 253-264. August 2006.
17. Lewis, R.; Shepherd A.; Torczon, V. (2007). Implementing Generating Set Search Methods for Linearly Constrained Minimization. *Society for Industrial and Applied Mathematics*. Vol.29, No. 6, pp. 2507-2530.
18. S. Fan Shu-Kai y Zahara Erwie (2007). Stochastic response surface optimization via an enhanced Nelder-Mead simplex search procedure. *Engineering Optimization*, Vol.38, No. 1, pp. 15-36.
19. Lewis, R.; and Torczon V. (2009). Active set identification for linearly constrained minimization without explicit derivatives. *Society for Industrial and Applied Mathematics*. Vol. 20, No. 3, pp. 1378-1405.
20. Bogani C. ,Gasparo M.G. y Papini A. (2009). GENERATING SET SEARCH METHODS FOR PIECE WISE SMOOTH PROBLEMS. Society for Industrial and Applied Mathematics, *SIAM J. OPTIM.*, vol. 20, No 1, pp. 321-335.

21. Khorsandi A., Alimardani A., Vahidir B. y Hosseinian S.H. (2010). Hybrid shuffled frog leaping algorithm and Nelder-Mead simplex search for optimal reactive power dispatch. Published in IET Generation, Transmission and Distribution, vol.5 ,Iss. 2 , pp. 249-256.
22. Bera S. y Mukherjee I. (2010). A Mahalanobis Distance-based Diversification and Nelder-Mead Simplex Intensification Search Scheme for Continuous Ant Colony Optimization. *World Academy of Science, Engineering and Technology* 69.
23. Griffin J. y Kolda T. (2010). Asynchronous parallel hybrid optimization combining DIRECT and GSS. *Optimization Methods y Software* , Vol. 25, No. 5, pp. 797-817.
24. Gao, X.K., Low, T.S., Liu, Z.J., Chen, S.X. (2002). Robust Design for Torque Optimization Using Response Surface Methodology. *IEEE Transactions on Magnetics*. Vol. 38, No. 2.
25. Luo C. y Yu B. (2011). Low dimensional simplex evolution: a new heuristic for global optimization. *Journal of Global Optimization* 52: pp. 45-55.
26. Barzinpour F., Noorossana R., Akhavan Niaki S. y Javad Ershadi M. (2012). A hybrid Nelder- Mead simplex and PSO approach on economic and economic-statistical designs of MEWMA control charts. *Int. J. Adv.Manuf. Technol* 65: pp. 1339-1348.
27. Hossein Gandomi A., Yang X., Hossein Alavi A. (2011). Cuckoo search algorithm: a metaheuristic approach to solve structural optimization problems. *Engineering with Computers* 29: pp. 17-35.
28. Hussain S. y Gabbar H. (2013). Fault Diagnosis In Gearbox Using Adaptive Wavelet Filtering and Shock Response Spectrum Features Extraction. *Structural Health Monitoring*, pp. 1-12.
29. McCormick, Matthew, and Verghese, Tomy. (2013). An Approach to Unbiased Subsample Interpolation for Motion Tracking. *Ultrasonic Imaging*, pp. 35: 76.
30. Ren Jian, y Duan Jinqiao. (2013). A parameter estimation method based on random slow manifolds. Cornell University Library.arXiv: 1303.4600.
31. Tippayawannakorn Noocharin y Pichitlamken Juta (2013).NELDER- MEAD METHOD WITH LOCAL SELECTION USING NEIGHBORHOOD AND MEMORY FOR STOCHASTIC OPTIMIZATION. *Journal of Computer Science*, 9 (4): pp. 463-476.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.