XII International Conference on Computational Plasticity. Fundamentals and Applications
COMPLAS XII
E. Oñate, D.R.J. Owen, D. Peric and B. Suárez (Eds)

A BENCHMARK STUDY ON IDENTIFICATION OF INELASTIC PARAMETERS BASED ON DEEP DRAWING PROCESSES USING PSO – NELDER MEAD HYBRID APPROACH

M. VAZ JR.[*], M. A. LUERSEN[†], P. A. MUÑOZ-ROJAS[*], E. BERTOTI[†], R. G. TRENTIN[††]

[*] Department of Mechanical Engineering
State University of Santa Catarina (UDESC)
Campus Universitário Prof. Avelino Marcante, 89219-710 Joinville, Brazil

[†] Department of Mechanical Engineering
Federal University of Technology – Paraná (UTFPR)
Av. Sete de Setembro, 3165, 80230-901 Curitiba, Brazil

[††] Department of Mechanical Engineering
Federal University of Technology – Paraná (UTFPR)
Via do Conhecimento, km 1, 85503-390 Pato Branco, Brazil

Key words: Parameter Identification, PSO, Nelder-Mead.

Abstract. Optimization techniques have been increasingly used to identification of inelastic material parameters owing to their generality. Development of robust techniques to solving this class of inverse problems has been a challenge to researchers mainly due to the nonlinear character of the problem and behaviour of the objective function. Within this framework, this work discusses application of Particle Swarm Optimization (PSO) and a PSO – Nelder Mead hybrid approach to identification of inelastic parameters based on a benchmark solution of the deep drawing process.

1 INTRODUCTION

The development of commercial Finite Element packages aimed at simulation of metal forming operations has instigated further research on identification of material parameters. Gradient-based techniques have been largely used in conjunction with parameter identification of elastic-plastic materials. However, existence of multiple local minima and overly plane regions in the design variable hyperspace have hampered application of such methods to some metal forming problems. Heuristic approaches, mostly based on Genetic Algorithms, have also been proposed owing to their potential ability to obtaining the global minimum. Notwithstanding, slow convergence and high computational cost constitute their main drawbacks. In recent years, the literature shows that Particle Swarm Optimization (PSO) – a Swarm Intelligence based algorithm – has been successful to solve a wide range of optimization problems [1,2,3]. In spite of the advancements, computational cost is still high owing to the number of fitness evaluations required to achieve the optimal solution.

Following a parallel research direction, the gradient-free Nelder-Mead (NM) method [4] has also yielded success in many engineering problems [5].

Apparently simple, the identification benchmark based on a deep drawing operation has shown to be a challenge to the existing optimization strategies. The present work addresses application of Particle Swarm Optimization (PSO) and a PSO – NM hybrid approach to identification of inelastic parameters based on a benchmark solution of the deep drawing process.

2 PARAMETER IDENTIFICATION AND THE OPTIMIZATION PROBLEM

Parameter identification is a class of inverse problems which determines material or system parameters from a known response. The present problem is formulated using optimization as

$$\begin{array}{lll} \text{Minimize} & g(\boldsymbol{p}) & \boldsymbol{p} \in R^n \\ & p_i^{\inf} \leq p_i \leq p_i^{\sup} & i = 1 \ldots n \end{array} , \tag{1}$$

where $g(\boldsymbol{p})$ is the objective function (or fitness), $\boldsymbol{p} = [\, p_1 \;\; p_2 \ldots \;\; p_i \;\; \ldots p_n \,]^T$ is the design vector containing n material parameters, and $p_i^{\sup}$ and $p_i^{\inf}$ are the lateral constraints. The objective function

$$g(\boldsymbol{p}) = \sqrt{1/N \sum\nolimits_{j=1}^{N} \xi_j \left[\left(R_j^{FEM}(\boldsymbol{p}) - R_j^{Exp} \right) / R_j^{Exp} \right]^2} \quad , \tag{2}$$

represents an error measure between the experimental, R^{Exp}, and corresponding computed response, $R^{FEM}(\boldsymbol{p})$, where N is the number of experimental points and ξ_j is a weight function.

2.2 Particle Swarm Optimization - PSO

Concepts of social behaviour of populations were used by Eberhart and Kennedy [1,2] to develop a Swarm Intelligence optimization method known as *Particle Swarm Optimization.* The method requires an initial population to which are applied velocity operators to simulate a combination of individual cognitive abilities and social interactions. The scheme attributes velocities to each particle with the following components: (*i*) inertia, (*ii*) personal history and (*iii*) neighbourhood effect, as explained in the following.

(*i*) *Inertia:* represents the tendency of a particle to follow its previous direction;

(*ii*) *Personal history:* corresponds to the location in the search space which results its best fitness – the cognitive effect;

(*iii*) *Neighbourhood effect:* stands for the influence of the best neighbouring individuals – the social effect.

Box 1 summarises the PSO algorithm [3]. In the present implementation, the *global best*, $\boldsymbol{p}_{gb}$, represents the best particle of the previous step (*neighbourhood effect*) and the *individual best*, $\boldsymbol{p}_{ib}$, corresponds to the best location of the particle along its history (*personal history*). In Box 1, the subscript m indicates any given particle of the population, superscript (k) denotes time step, w is the *inertia* parameter, and $U(0,\varphi_1)$ and $U(0,\varphi_2)$ are random functions ranging from 0 to φ_1 and φ_2, respectively.

(i) Set $k=0$ and generate randomly the initial population $\boldsymbol{p}^{(0)}$ and corresponding velocities, $\boldsymbol{v}^{(0)}$

$$\boldsymbol{p}^{(0)} = \left\{ \boldsymbol{p}_1^{(0)}\ \boldsymbol{p}_2^{(0)} \dots \boldsymbol{p}_m^{(0)} \dots \boldsymbol{p}_{n_p}^{(0)} \right\} \quad \text{and} \quad \boldsymbol{v}^{(0)} = \left\{ \boldsymbol{v}_1^{(0)}\ \boldsymbol{v}_2^{(0)} \dots \boldsymbol{v}_m^{(0)} \dots \boldsymbol{v}_{n_p}^{(0)} \right\}$$

in which n_p is the number of particles (individuals) and n is the number of design variables (material parameters)

$$\boldsymbol{p}_m^{(0)} = \left[p_1^{(0)}\ p_2^{(0)} \dots p_i^{(0)} \dots p_n^{(0)} \right]^T \quad \text{and} \quad \boldsymbol{v}_m^{(0)} = \left[v_1^{(0)}\ \ v_2^{(0)}\ \dots\ v_i^{(0)}\ \dots\ v_n^{(0)} \right]^T$$

(ii) Evaluate individual and global best particles, $\boldsymbol{p}_{ib}^{(k)}$ and $\boldsymbol{p}_{gb}^{(k)}$

$$\text{If } g\left(\boldsymbol{p}_m^{(k)}\right) < g\left(\boldsymbol{p}_{ib}^{(k)}\right) \quad \Rightarrow \quad \boldsymbol{p}_{ib}^{(k)} \leftarrow \boldsymbol{p}_m^{(k)}$$

$$\text{If } g\left(\boldsymbol{p}_m^{(k)}\right) < g\left(\boldsymbol{p}_{gb}^{(k)}\right) \quad \Rightarrow \quad \boldsymbol{p}_{gb}^{(k)} \leftarrow \boldsymbol{p}_m^{(k)}$$

(iii) Compute new velocities $\boldsymbol{v}^{(k+1)}$

$$\boldsymbol{v}^{(k+1)} = w\boldsymbol{v}^{(k)} + \boldsymbol{U}(0,\varphi_1) \otimes \left(\boldsymbol{p}_{ib}^{(k)} - \boldsymbol{p}^{(k)}\right) + \boldsymbol{U}(0,\varphi_2) \otimes \left(\boldsymbol{p}_{gb}^{(k)} - \boldsymbol{p}^{(k)}\right)$$

and verify against maximum limits of each design variable (material parameter)

$$v_i^{\max} = w_i \left(p_i^{\sup} - p_i^{\inf}\right) \qquad \text{where} \qquad w_i \in [0,0.5]$$

$$\text{If } \left|v_i^{(k+1)}\right| > v_i^{\max} \quad \Rightarrow \quad \left|v_i^{(k+1)}\right| \leftarrow v_i^{\max}$$

(iv) Compute new location of all particles $\boldsymbol{p}^{(k+1)}$ (design vectors)

$$\boldsymbol{p}^{(k+1)} = \boldsymbol{p}^{(k)} + \boldsymbol{v}^{(k+1)}$$

and verify boundary conditions $\left[p_i^{\sup}, p_i^{\inf}\right]$

$$\text{If } p_i^{(k+1)} > p_i^{\sup} \quad \Rightarrow \quad p_i^{(k+1)} \leftarrow p_i^{\sup}$$

$$\text{If } p_i^{(k+1)} < p_i^{\inf} \quad \Rightarrow \quad p_i^{(k+1)} \leftarrow p_i^{\inf}$$

(v) Evaluate stopping criteria

```
IF $\phi(\boldsymbol{p}) < TOL_\phi$ THEN
  $\boldsymbol{p}^{\min} \leftarrow \boldsymbol{p}_{gb}^{(k+1)}$
  EXIT
ELSE IF $k = k^{\max}$ THEN
  EXIT
ELSE
  $k \leftarrow k+1$
  GOTO (ii)
ENDIF
```

Box 1 : Particle Swarm Optimization algorithm [3]

Convergence and success are evaluated by the *parametric dispersion index* and *normalised fitness index*, respectively, as

$$\phi_d^{(k)} = \frac{\sum_{m=1}^{n_s} \left| \left\langle \boldsymbol{p}_{sb}^{(k)} - \boldsymbol{p}_m^{(k)} \right\rangle \right|}{\sum_{m=1}^{n_s} \left| \left\langle \boldsymbol{p}_{sb}^{(0)} - \boldsymbol{p}_m^{(0)} \right\rangle \right|} \qquad \text{and} \qquad \phi_p^{(k)} = \frac{g[\boldsymbol{p}_{sb}^{(k)}]}{g[\boldsymbol{p}_{sb}^{(0)}]}, \tag{3}$$

where n_s is a predefined fraction of the population (50 % of the best particles, in the present study), | · | is the Euclidean norm, and $\boldsymbol{p}_{sb}$ is the best particle of the (k^{th}) step. The *dispersion index* uses the parametric form of $\boldsymbol{p}_{sb}$ and $\boldsymbol{p}_m$, in which $\left\langle p_{sb,i} - p_i \right\rangle = (p_{sb,i} - p_i)/(p_i^{\text{sup}} - p_i^{\text{inf}})$, $i = 1 \ldots n$. Noteworthy, the *normalised fitness index* is zero for the benchmark parameters.

2.1 The Nelder-Mead optimization technique

The Nelder–Mead method [4] – also known as *downhill simplex method*– is one of the most popular direct search techniques for unconstrained real optimization. It is a gradient-free method which is based on the comparison of function values at the n+1 vertices of a *simplex*. A *simplex* is a geometric figure composed by a set of n+1 points (vertices) in an n-dimensional space. During the iterative minimization process, the simplex vertices positions are changed through *reflection*, *expansion* and *contraction* operations in order to find a better point (a point with a smaller objective function value), moving gradually toward the optimum. The algorithm terminates when the values of the objective function in all vertices become equal within a given tolerance. The cumulative effect of the operations on the simplex is, roughly speaking, to stretch the simplex shape along the descent directions, and to zoom it around the optimum. The original Nelder-Mead algorithm was conceived for unbounded domains, however, in this work, an improved version of the method was used, where the design variables bounds are taken into account by projection [5].

3 NUMERICAL EXAMPLES AND DISCUSSIONS

The benchmark solution is based on the deep drawing process depicted in Figure 1. The problem is assumed axisymetrical and the Finite Element mesh used in the simulations contains 4 x 60 linear elements. A total displacement of 50 *mm* is imposed by a rigid punch. The yield stress curve is given by the Ramberg-Osgood model, $\sigma_y = \sigma_0\,(1+k\varepsilon_p)^n$, in which $p = [\sigma_0\;\; k\;\; n\,]^T$ are the parameters to be determined. A frictionless process is assumed in order to eliminate other loading effects. The original geometry of the problem was proposed by de la Cour [6] to study identification of material and friction parameters from actual deep drawing operations. In the present work, the identification problem constitutes to recovering hardening parameters from a *load – displacement* curve obtained by the predefined set of parameters presented in Table 1.

3.1 The deep drawing operation – how difficult is to recover the benchmark parameters

The deep drawing process basically comprises two stages: (*i*) the forming stage and (*ii*) extraction. The forming stage takes place for a punch displacement up to 30 *mm*, followed by extraction in the remaining 20 *mm* displacement. Identification uses only the first stage since no forming load associated with plastic deformation is generally observed during extraction (the weight function assigned to eq. (2) for this stage is $\xi_j = 10^{-6}$).

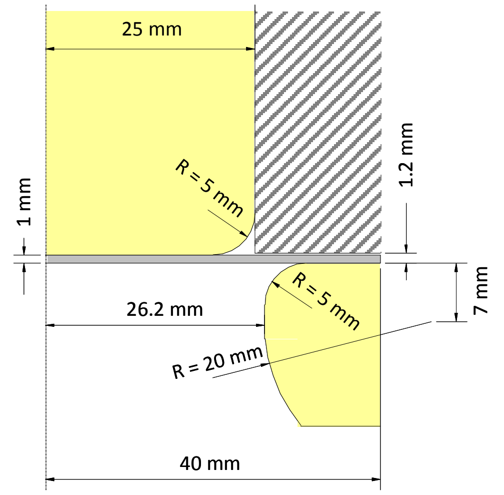

Figure 1: Initial geometry (axisymmetrical model)

Table 1 : Reference material properties

σ_0 [*MPa*]	*k*	*n*	*E* [*GPa*]	*ν*
150.0	200.0	0.25	70.0	0.3

Table 2 : Lower and upper limits for hardening parameters

Parameter	Lower limit	Upper limit
σ_0 [*MPa*]	100.0	200.0
k	100.0	300.0
n	0.1	0.4

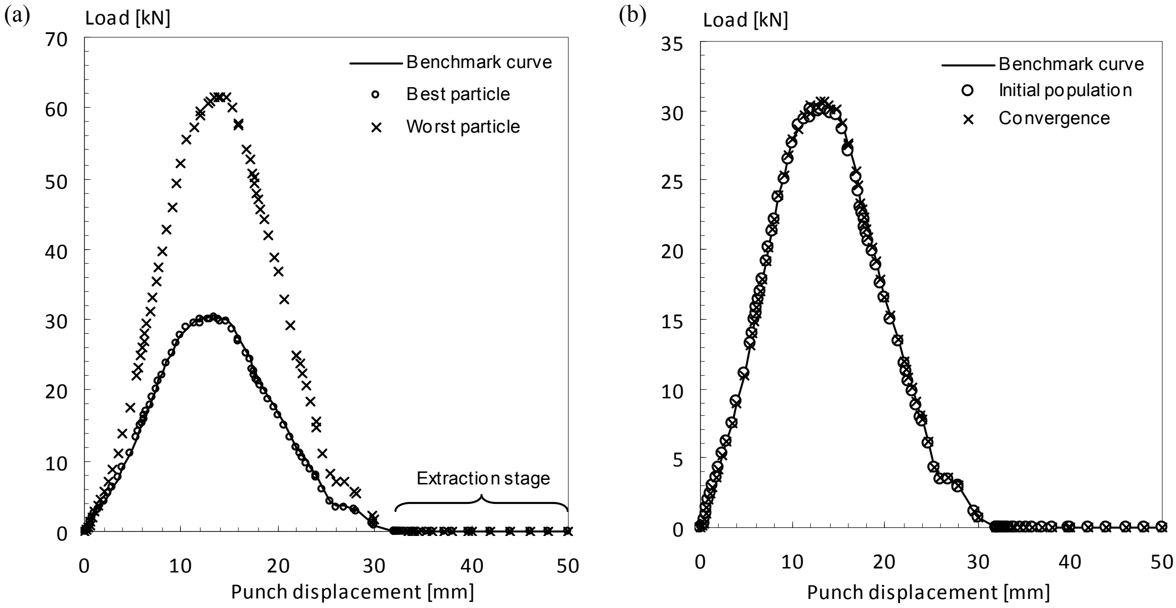

Figure 2: Loading process: (a) initial population and (b) at convergence

The degree of difficulty of the present identification procedure can be perceived when accessing the reference PSO solution: lower and upper limits presented in Table 2 and population size n_p = 120 and PSO control weights w = 0.5 and $\varphi_1 = \varphi_1$ = 1.0. Figure 2(a) shows the benchmark process against the loading curves for the *best* and *worst* particles of the initial population. It can be readily observed that the best particle of the initial population yields almost a visually indiscernible loading curve when compared to the benchmark solution. Despite such proximity, the hardening parameters are substantially different, leading to a quite large value of $g(\boldsymbol{p})$. Such behaviour is highlighted in Figure 2(b), which presents the loading curve for the best particle of the initial population and at convergence ($\phi_p^{(k)} = g[\boldsymbol{p}^{(k)}]/g[\boldsymbol{p}^{(0)}] = 2 \text{ x } 10^{-6}$). Table 3 presents the corresponding hardening parameters.

Table 3: Hardening parameters of the best particles for n_p = 120, $\varphi_1 = \varphi_2$ =1.0 and w = 0.5

Case	σ_0 [*MPa*]	k	n	$g[\boldsymbol{p}^{(k)}]$
Benchmark	150.0	200.0	0.25	0.0
Initial population	161.200842	151.668638	0.245236695	$1.30932 \text{ x } 10^{-2}$
Convergence	150.000007	200.000022	0.249999987	$1.06554 \text{ x } 10^{-6}$

Table 4: Final parameters for the sole Nelder–Mead runs

Run	σ_0 [*MPa*]	k	n	
1	115.263575	152.422207	0.345614	
2	160.568554	155.108585	0.248153	
3	167.861985	104.491676	0.263570	
4	**150.000010**	**199.999916**	**0.250000**	**← success**
5	142.134206	256.993355	0.248251	
6	144.644356	237.750036	0.248608	
7	142.555435	254.611949	0.248102	
8	182.331590	243.258869	0.183204	
9	162.608795	233.540655	0.218168	
10	144.330283	262.343120	0.242789	
11	**150.000001**	**199.999996**	**0.250000**	**← success**
12	157.449254	161.712293	0.250595	
13	155.529331	167.587594	0.251871	
14	119.382355	163.240097	0.328112	
15	123.602295	126.123278	0.341889	
16	168.367034	205.057582	0.216506	
17	157.343173	158.585516	0.252348	
18	166.365414	102.743295	0.356347	
19	170.129193	274.616758	0.198950	
20	190.485892	296.334696	0.164975	
21	145.455946	227.871556	0.249814	
22	146.789676	232.030391	0.246036	
23	155.529834	167.595469	0.251869	
24	169.284216	271.302117	0.198398	

It is relevant to mention that the present benchmark problem was also approached by the Schittkowski's NLPQLP quadratic programming algorithm [7] and sole application of the gradient-free Nelder-Mead method. In order to assess the ability of the aforementioned methods to recovering the benchmark parameters, 20 random initial estimates were tested. The gradients provided to the NLPQLP algorithm were calculated using a modified finite difference technique [8] but the method failed to recover the benchmark results for all initial estimates. Such poor outcome in the present problem is mainly due to a combination of very small gradients in some search directions. With such small gradients, any inaccuracy introduced by the finite difference procedure suffices to break down the optimization process.

The sole application of the NM scheme yielded a low rate of success, i.e., only two out of 24 runs reached the reference parameters, as illustrates Table 4. It is relevant to note that each NM run requires between 420 and 500 computations of the objective function, statistically making an average of about 5,500 fitness evaluations for a successful run.

3.2 The sole application of PSO – some convergence issues

This section presents a brief assessment of the sole application of PSO to the proposed deep drawing operation. The classical PSO implementation depends on the population size, inertia, cognitive and social parameters. Based upon previous studies on identification of hardening parameters [3,9], the following ranges are studied: $n_p \geq 60$, $0.1 \leq w \leq 0.9$ and $\varphi_1 = \varphi_2 \in [0.5, 2.0]$. Convergence is attained when the *dispersion index* reaches $\phi_d = 1\times10^{-6}$. The identification process is assumed successful for *normalised fitness* $\phi_p \leq 2\times10^{-6}$.

Population size: Figures 3(a) and 3(b) show evolution of the identification process for the dispersion index and normalised fitness, respectively, for PSO control weights w = 0.5 and $\varphi_1 = \varphi_2$ = 1.0. As expected, smaller populations provide insufficient search capacity, leading to sub-optimal solutions. Larger populations yield more robust evolutions without, however, any improvement of the convergence rate. It is important to mention that convergence does not imply success, as the curves for n_p = 60 show.

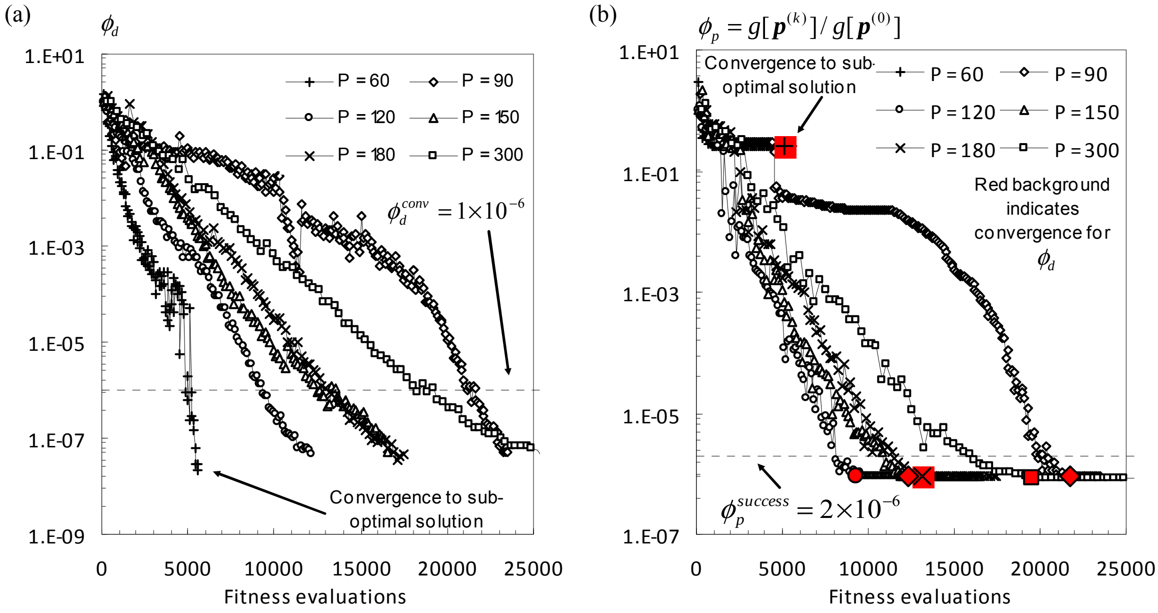

Figure 3 : Influence of the population size: (a) parametric *dispersion index* and (b) *normalised fitness*

Cognitive and social weights: The cognitive and social weights define the learning capacity of the algorithm to finding the minimum within a restricted region [3]. Table 5 summarises typical runs for $\varphi_1 = \varphi_2 \in [0.5, 2.0]$. Smaller cognitive and social weights compromise the exploitation capacity of the algorithm leading to sub-optimal solutions. On the other hand, larger values promote an erratic movement of the particles near the minimum region decreasing substantially the convergence rate.

Table 5: Influence of the cognitive and social weights, $\varphi_1 = \varphi_2$, for $w = 0.5$ and $n_p = 120$

PSO Weight	**Parameter**			**Normalised**	**Fitness**	**Convergence**
$\varphi_1 = \varphi_2$	σ_0 [MPa]	k	n	**Fitness** ϕ_p	**evaluations**	$[\phi_d^{conv} = 10^{-6}]$
0.5	157.238397	144.154466	0.258825008	4.00925E-1	23160	Sub-optimal
0.7	150.000041	199.999673	0.250000050	1.73882E-6	8400	Global minimum
0.9	150.000008	199.999989	0.249999996	9.09007E-7	8760	
1.0	150.000007	200.000022	0.249999987	1.06554E-6	9360	
1.1	150.000014	199.999985	0.250000004	9.34123E-7	8280	
1.3	150.000008	199.999995	0.250000002	9.34123E-7	11400	
1.5	150.000007	199.999985	0.249999997	9.34123E-7	14280	
1.7	150.000021	199.999903	0.250000013	9.52680E-7	19800	
1.9	150.000012	199.999995	0.250000004	9.34123E-7	35640	
2.0	–	–	–	–	–	Unstable

Inertia weight: The inertia weight aims at increasing exploitation, helping the algorithm to avoid local or sub-optimal solutions. Table 6 shows that smaller inertia values lead to sub-optimal solutions. For the inertia parameter $w = 0.9$, no convergence was attained up to 40000 fitness evaluations.

Table 6: Influence of the inertia weight, w, for $\varphi_1 = \varphi_2 = 1.0$ and $n_p = 120$.

PSO Weight	**Parameter**			**Normalised**	**Fitness**	**Convergence**
w	σ_0 [MPa]	k	n	**Fitness** ϕ_p	**evaluations**	$[\phi_d^{conv} = 10^{-6}]$
0.1	149.993176	200.065446	0.249991092	3.31397E-4	5880	Sub-optimal
0.3	150.000014	199.999961	0.250000012	9.61714E-7	7080	Global minimum
0.5	150.000007	200.000022	0.249999987	1.06554E-6	9360	
0.7	150.000007	199.999986	0.249999996	9.34123E-7	16080	
0.9	–	–	–	–	–	Unstable

The simulations show that success (the ability to recover the benchmark parameters) is strongly dependent upon the PSO control weights. Furthermore, the successful runs (those which attain $\phi_p \leq 2\times10^{-6}$) provide differences in the 8th significant digit. It worth mentioning that, in spite of the high success rate, PSO requires a huge number of fitness evaluations, as indicates Figure 3 and Table 5 and 6.

3.3 The hybrid PSO – NM identification strategy

In spite of the high success rate, the high computational cost of the PSO method has motivated further investigation on hybrid approaches. In the present work, one intends to combine (*i*) the good PSO performance to approach the neighbourhood of the optimal point with (*ii*) the relatively small number of fitness computations required by the Nelder-Mead technique to obtain the minimum itself. The hybrid strategy splits the identification problem into two stages: (A) in the first stage, the PSO technique is applied until a stopping criterion is attained, followed by (B) application of the NM method using the best PSO particle as the initial estimate. There are several ways to define the PSO stopping criterion and two cases were investigated in this work:

Case 1: The parametric dispersion index reaches $\phi_d \leq 0.15$.

Case 2: The PSO iterative step attains $k = 10$.

The PSO control weights used in the first stage are $w = 0.5$ and $\varphi_1 = \varphi_2 = 1.0$ for a total number of particles, $n_p = 120$. In order to evaluate a possible effect of the random character of the process, 5 PSO runs were performed using the aforementioned control parameters.

The PSO results for ***Case 1*** is presented in Table 7, which indicates the PSO run with the corresponding number of solution steps and fitness computations when the parametric dispersion index reaches the established criterion. The second stage uses the parameters indicated in Table 7 as the initial NM estimate. The simulations show a 100% success rate at the end of stage (B), i.e. the benchmark parameters were obtained in all NM runs. In addition, the number of fitness evaluations was remarkably small, as shown in Table 8.

Table 7: *Case 1*– stage (A): initial estimates for the Nelder-Mead based on the dispersion index, $\phi_d < 0.15$

Run	Step	PSO fitness evaluations	ϕ_d	Parameter		
				σ_0 [MPa]	k	n
1	10	1320	0.091345	148.496300	213.492918	0.248660585
2	8	1080	0.145975	150.330376	190.051678	0.251763496
3	10	1320	0.142570	150.678832	193.292638	0.250914858
4	10	1320	0.125507	150.672116	195.420749	0.250248172
5	7	960	0.147745	149.518517	199.807439	0.252852869

Table 8: *Case 1*– stage Stage (B): final parameters for the hybrid PSO – NM scheme.

Run	Fitness evaluations			Parameter		
	PSO	NM	PSO-NM	σ_0 [MPa]	k	n
1	1320	237	1557	149.999985	200.000077	0.25000000
2	1080	228	1308	150.000006	199.999908	0.25000000
3	1320	201	1521	150.000005	199.999997	0.25000000
4	1320	207	1527	149.999997	200.000000	0.25000000
5	960	209	1169	149.999992	200.000115	0.25000000

Case 2 uses the parameters corresponding to the best particle of the 10th PSO iterative step as the initial NM estimate. In this case, the number of fitness computations for stage (A) is the same for all 5 runs, as presented in Table 9. The final NM parameters of stage (B) are shown in Table 10, which highlights NM and PSO-NM combined fitness evaluations. As in *Case 1*, the PSO-NM hybrid scheme was successful in all 5 runs.

Table 9: *Case 2*– stage (A): initial estimates for the Nelder-Mead using parameters of the 10th PSO step

Run	Step	PSO fitness evaluations	ϕ_d	Parameter		
				σ_0 [MPa]	k	n
1	10	1320	0.091345	148.496300	213.492918	0.248660585
2	10	1320	0.103802	150.075426	197.549246	0.250243651
3	10	1320	0.142570	150.678832	193.292638	0.250914858
4	10	1320	0.125507	150.672116	195.420749	0.250248172
5	10	1320	0.137576	150.844119	190.616571	0.251063031

Table 10: *Case 2*– stage Stage (B): final parameters for the hybrid PSO – NM scheme

Run	Fitness evaluations			Parameter		
	PSO	NM	PSO-NM	σ_0 [MPa]	k	n
1	1320	237	1557	149.999985	200.000077	0.25000000
2	1320	229	1308	150.000002	200.000027	0.25000000
3	1320	195	1521	149.999998	200.000002	0.25000000
4	1320	219	1527	149.999997	200.000000	0.25000000
5	1320	261	1169	150.000009	199.999918	0.25000000

A successful recover of the benchmark parameters was achieved for all 5 sole PSO runs with an average number of fitness computations of 7000. The sole application of the Nelder-Mead method has yielded an average of 5500 fitness computations for a success, but it has required many re-initializations, which makes an 8.3 % run success rate. Tables 8 and 10 show that the hybrid PSO – NM technique achieved also a 100 % success rate with an average number of fitness evaluations around 1500. Therefore, the following aspects can be considered:

- The sole application of the PSO algorithm has yielded a high success rate, but at high computational cost.
- The sole application of the NM technique presented a poor success rate; however, for the upper and lower limits given in Table 2, the average number of fitness computations for a successful run was smaller than sole use of the PSO method.
- The hybrid PSO – NM technique was found to be the best option since it combined high success rates and reduced number of fitness computations. The reduction in the number of fitness evaluations was 79 % and 72 % when compared to using PSO and NM, respectively.

4 CONCLUDING REMARKS

The present work propounds a hybrid PSO – NM technique to obtaining hardening parameters based on the assessment of a deep drawing benchmark problem. Despite high computational cost, the dynamics of the PSO algorithm has led to a high success rate, i.e., the optimization procedure was able to recover the benchmark parameters for a fairly large range of its control parameters. The sole application of the Nelder-Mead method has shown strongly dependent on the initial estimates; i.e., in the present example, success was achieved only if the initial parameters are sufficiently close to the benchmark values. The hybrid PSO – NM method combined the ability of the PSO algorithm to converge towards the minimum with a high success rate with the capacity of the NM method to reduce the number of fitness computations significantly.

ACKNOWLEDGEMENTS

The authors acknowledge the financial support provided by the Brazilian funding agency CNPq - (National Council for Scientific and Technological Development).

REFERENCES

[1] Eberhart, R. and Kennedy, J. A new optimizer using particle swarm theory, in: *Proc. of the 6th Int. Symposium on Micro Machine and Human Science*, IEEE Press, Piscataway, 39-43 (1995).

[2] Kennedy, J. and Eberhart, R. Particle Swarm Optimization, in: *Proc. of the IEEE Int. Conference on Neural Networks*, IEEE Press, Piscataway, 1942-1948 (1995).

[3] Vaz Jr., M., Cardoso, E.L. and Stahlschmidt, J. Particle Swarm Optimization and identification of inelastic material parameters, *Eng. Comput.,* Accepted for publication (2013).

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

[5] Luersen, M.A. and Le Riche, R. Globalized Nelder-Mead method for engineering optimization. *Comput. & Struct.* (2004) **82**:2251-2260.

[6] de la Cour, D. *Identification of Material and Friction Parameters from Deep Drawing*. Ph.D. Thesis, Danish Center for Applied Mathematics and Mechanics, Technical University of Denmark (2003).

[7] Schittkowski, K. *NLPQLP: A new Fortran implementation of a sequential quadratic programming algorithm for parallel computing*. Research report, Department of Mathematics, University of Bayreuth, D-95440 Bayreuth (2001).

[8] Muñoz-Rojas, P.A., Fonseca, J.S.O. and Creus, G.J. A modified finite difference sensitivity analysis method allowing remeshing in large strain path-dependent problems. *Int. J. Num. Meth. Engng.* (2004) **61**:1049-1071.

[9] Vaz Jr., M. and Cardoso, E.L. Aspects of identification of elastic-plastic parameters using heuristic algorithms. In: Pimenta, P.M. and Campello, E.M.B., (Eds.), *Proceedings of the 10th World Congress on Computational Mechanics*, EPUSP, São Paulo, (2012) 1-15.