

Comparison between Deterministic and Stochastic formulations of Particle Swarm Optimization, for Multidisciplinary Design Optimization.

Daniele Peri*and Matteo Diez*

CNR-INSEAN - Italian Ship Model Basin and Research Centre

Giovanni Fasano[†]

University "Ca'Foscari" of Venice

Particle Swarm Optimization (PSO) is having a growing space in the optimization community, mainly due to its appreciable qualities of fast initial progress, reduced computational cost and parallel structure, suitable for High Performance Computation (HPC) platforms. Original formulation includes some random coefficients, so that a statistical analysis of the solution is often needed. To avoid the latter situation, Deterministic Particle Swarm Optimization (DPSO) has been introduced: removing all the random coefficients the DPSO is a deterministic algorithm, so that a single run is considered to evaluate the success of the algorithm. In this paper, a comparative study between PSO and DPSO is reported, in order to investigate the performance of DPSO versus PSO.

Nomenclature

<i>PSO</i>	Particle Swarm Optimization
<i>DPSO</i>	Deterministic Particle Swarm Optimization
x	Particle position vector
v	Particle speed vector
<i>Superscript</i>	
k	Iteration index

I. Introduction

Among the different formulations available for the solution of MDO problems, a really interesting one is the so-called DAO (Direct Analysis and Optimization). This formulation exploits the similitude between two successive configurations analyzed by the optimizer, so that the state (interdisciplinary) variables are supposed to converge while the actual solution is evolving towards an optimal one: smaller and smaller changes in the system configuration are usually observed when approaching the optimal solution, so that the convergence of the state variables is obtained implicitly. The new Multi Disciplinary Analysis (MDA) re-uses the state variables from the previous MDA: an imperfect coupling of the disciplines is allowed, so that a large saving in the number of simulations inside a single MDA is obtained in this way. This formulation is typically adopted in conjunction with local optimization methods, where the actual configuration of the system is gradually changing while the optimum is approached, so that it is reasonable to assume gradual modification of the state variables. In case a global optimization algorithm is adopted, the (possibly) large differences between two successive configurations of the system may negatively affect the accuracy of solving

*Researcher, Marine Vehicles Optimization Dept. - via di Vallerano, 139 - 00128 Roma - Italy

[†]Researcher, Department of Management - San Giobbe, Canaregio 873 - 30121 Venice - Italy

the MDA, and yield poor convergence and/or wrong search directions. In⁹ the use of a global optimization algorithm with a DAO formulation has been illustrated. Particle Swarm Optimization (PSO) was adopted in the latter case; pursuing the convergence by the evolution of different individuals (particles) of a swarm, where the evolution of each individual can be regarded to a large extent as independent of the others. The idea was to let the particles evolve individually, re-using previous information for each single particle, so that a number of DAOs, virtually one for each particle, is obtained. In⁹ a Deterministic PSO (DPSO) has been adopted, that is, a deterministic variant of the original (random) algorithm, widely adopted in the optimization community. In the current paper, a comparison between DPSO and PSO is carried on, in order to test the effectiveness of DPSO with respect to the original PSO.

II. PSO Algorithm: random and deterministic formulations

The Particle Swarm Optimization (PSO) was initially developed by Kennedy and Eberhart in 1995⁶. It is a heuristics for global optimization problems, applicable to a wide range of optimization problems. PSO is a derivative-free optimization method, so that no expensive computation of gradient and/or Hessian of the objective function are required. Small memory occupation, easy parallelization and simplicity of implementation are other key points for the success of this algorithm.

In PSO, each particle, a collection of which makes up a swarm, has information about its actual position and speed in the search space. Each particle *remembers* the best position (x_p) and the corresponding objective function value that it has individually found in the search process. As a swarm, all the particles also share the best position (x_g) and the corresponding objective function value among all the locations ever visited by the swarm. Each particle aims to find a global optimal solution making use of the current velocity v , x_p and x_g . PSO's iteration is expressed as ($j = 1, \dots, N$):

$$v_j^{k+1} = \chi[\omega v_j^k + c_1 r_1 (x_{pj}^k - x_j^k) + c_2 r_2 (x_g^k - x_j^k)] \quad (1)$$

$$x_j^{k+1} = x_j^k + v_j^{k+1} \quad (2)$$

where v and x are velocity and position, respectively; the subscript j represents the particle number, the superscript k indicates the iteration number, r_1 and r_2 are different random values in the range $[0,1]$. The coefficients in Eq. (1)-Eq. (2) have the following meanings: χ is the *constriction factor*, ω is called *inertia weight*, c_1 , c_2 are called *cognitive* and *social* parameter (positive constants), respectively. According with⁹ and DPSO formulation we decide to set in (1) the parameters r_1 and r_2 equal to 1, thus we eliminate the random choice introduced by these two coefficients. In this way, we transform a pure stochastic method into a deterministic one (see also^{1,7}).

As regards the history of PSO formulae Eq. (1)-Eq. (2), observe that the original PSO algorithm⁶ was formulated as ($j = 1, \dots, N$)

$$v_j^{k+1} = \omega v_j^k + w_1 r_1 (x_{pj}^k - x_j^k) + w_2 r_2 (x_g^k - x_j^k) \quad (3)$$

$$x_j^{k+1} = x_j^k + v_j^{k+1} \quad (4)$$

where w is an inertial coefficient (which tends to preserve the search direction indicated by the velocity v_j^k at the previous iteration), w_1 is a coefficient for individual confidence and w_2 is a coefficient for swarm confidence; moreover, typical values of the parameters above were $w = 1$, $w_1 = w_2 = 2.0$. Subsequently a work done by Clerc² indicated that the introduction of a constriction factor may be necessary to force convergence of the PSO algorithm, so that in³ PSO was defined as ($j = 1, \dots, N$)

$$v_j^{k+1} = \chi[v_j^k + c_1 r_1 (x_{pj}^k - x_j^k) + c_2 r_2 (x_g^k - x_j^k)] \quad (5)$$

$$x_j^{k+1} = x_j^k + v_j^{k+1} \quad (6)$$

where χ is the constriction coefficient, c_1 is a coefficient for individual credibility and c_2 is a coefficient for collective credibility, providing a weight for the attraction towards the personal best or the global best position respectively.

The initial points configuration required by the method, i.e. v_j^0 and x_j^0 for $j = 1, \dots, N$, may be a crucial issue for the efficiency of PSO iteration, as investigated in^{4, 5, 10}. Regarding the value of the coefficients, they are typically assumed as in², that is, $\chi = 0.729$, $w = 1.0$, $c_1 = c_2 = 2.05$. Those values for the coefficients are obtained considering the following relationship among coefficients

$$\chi c_1 = w_1, \quad \chi c_2 = w_2. \quad (7)$$

The equations (7) indicates that the constriction coefficient χ has a direct control over the velocity of particles, which is considered in³ to be crucial for the progress of the algorithm. By relations Eq. (1) and Eq. (3) observe that using the values of the parameters $\chi = 0.729$, $w = 1$, $c_1 = c_2 = 2.05$ in Eq. (1), is equivalent to set in Eq. (3) $\chi = 1.0$; $w = 0.729$ $c_1 = c_2 = 1.494$ (i.e. $0.792 * 2.05 = 1.49445$).

The above values of parameters in PSO are frequently adopted in solving various optimization problems. However, these values of the parameters are not necessarily *optimal*, since they are obtained by using heuristic assumptions. In⁸, a systematic study of the effect of parameters of PSO has been reported, on a suite of 39 algebraic functions, with dimensionality from 2 to 20. A huge sensitivity of the final solution with respect to the parameters has been observed, so that modifications on the eighth significant digit for the coefficients still showed a large effect on the results. Anyway, the coefficients proposed in the latter paper are $c_1 = 2.04236627$, $c_2 = 1.14957333$, $\chi = 1.04718542$, $w = 0.72004962$.

Apart from⁸, some attempts have been considered in order to improve the results provided by the classical selection of parameters of PSO in². Observe that as a general rule, reducing the parameters may force the monotonic convergence beforehand, limiting the exploration. Indeed, if we observe a particle for which global and personal best position coincide, assuming $c_1 + c_2 < 1$ and its velocity at previous iteration being equal to zero, then the new position of the particle will be located in the same subspace spanned by the global best position, and convergence will occur with a progress related to a geometric series. On the other hand, if the sum of c_1 and c_2 is smaller than 2, the new position will overcome the global best location, by a quantity smaller than the previous distance from the global best position. To this purpose, two different selections for the parameters have been tested in the past: the first was $c_1 = 0.4$, $c_2 = 0.8$, $\chi = 1.0$, $w = 0.9$, while the second was $c_1 = 0.25$, $c_2 = 0.25$, $\chi = 0.8$, $w = 0.9$.

All these tests were performed by using the modified version of PSO, addressed as DPSO (Deterministic Particle Swarm Optimization), obtained by setting all the random coefficients as deterministic constants, that is eliminating all the stochastic behavior of the algorithm, so that a statistical analysis of the solution is no more needed. On the other hand, there may be possible side effects due to a reduced exploration caused by the elimination of the random terms.

In the following, a systematic analysis and comparison of the performances of PSO and DPSO is reported. 1000 runs of PSO are compared with the equivalent DPSO runs, using all the previously described selections for the swarm's parameters.

III. Numerical results

1000 different runs have been performed for the PSO algorithm in order to provide some statistics of the results. The following stop criterion for PSO iterations was adopted: if the quantity

$$\sum_{i \neq j} \|x_i^k - x_j^k\|$$

is smaller than a prescribed quantity, as well as the sum of the norms of the velocities of all the particles, it means that the swarm exploration is reduced and the velocities are low, so that the swarm is likely converging to a single point.

The test has been performed using 27 algebraic test functions, already reported in¹: these are a suite of functions commonly adopted for testing optimization algorithms, dimensions ranging from 2 to 20. The majority of these functions is multimodal, some of them have a really large number of local minima. In this situation, classic gradient-based local optimizers may struggle to find a global minimum.

An outlook of the results is presented in Figure 1. A single bar reports the result provided by DPSO, due to the deterministic nature of the algorithm. On the contrary, a distribution of results is the outcome of PSO, due to the randomness of the coefficients (so we have not the same result each time we run the

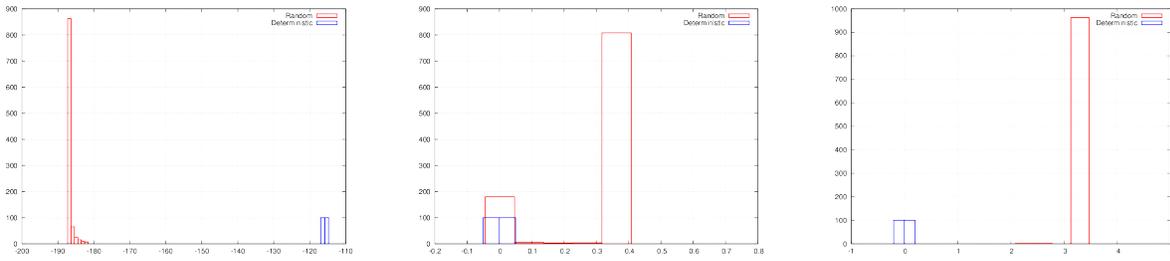


Figure 1. Three different examples of distribution of the optimal solution obtained by using DPSO or PSO. On left: PSO prevails (algebraic function n. 9). On center: PSO and DPSO are equivalent (algebraic function n. 18). On right: DPSO prevails (algebraic function n. 27).

algorithm). On the horizontal axis, the value of the objective function is reported: since we are solving a minimization problem, the bar closer to the left border represents the best result. The height of the bar indicates the number of times (1000 is the maximum, since we ran the optimizer 1000 times) for which PSO provides a result in within the width of the bar. This value has no further meaning for the DPSO. On left, a situation for which DPSO performs better than PSO is reported, while the opposite is illustrated in the right part of Figure 1. Observing the results where PSO is prevailing, we can desume the global optimum (or a good approximation of it): this is not a trivial situation. On center of Figure 1 a *deuce* situation is depicted. Both PSO and DPSO detect, at least one time, the global minimum of the function, but in this case to find the global minimum is not the most probable event for DPSO. In 20% of the cases DPSO detects the global minimum, but in about 80% of the cases it does not detect the best point. A completely different situation is depicted on the left end side of Figure 1: in this case the probability to get the global minimum are close to 90%, while in the remaining 10% of the cases the detected value is pretty close to the global minimum (closer than the value obtained by DPSO).

Numerical results are reported in Tables 1 - 5 (OF stands for *Objective Function*). The strong prevalence of PSO over DPSO reported in Table 1 is definitely expected: in fact, the adopted coefficients are outside the range of convergence for DPSO as reported in⁴, so we know *a priori* that these coefficients are not suitable for DPSO, and they cannot be used.

DPSO appears to prevail when the coefficients are smaller, and in general we can assume that DPSO gets convergence in a smaller number of iterations. If we consider the precision of the algorithm on the basis of the final value of the objective function, DPSO and PSO are substantially equivalent. Percentage of success of PSO slightly increases if the number of runs is increased: a couple of percentage points are gained if we pass from a statistics on 100 runs to a statistics on 1000 runs.

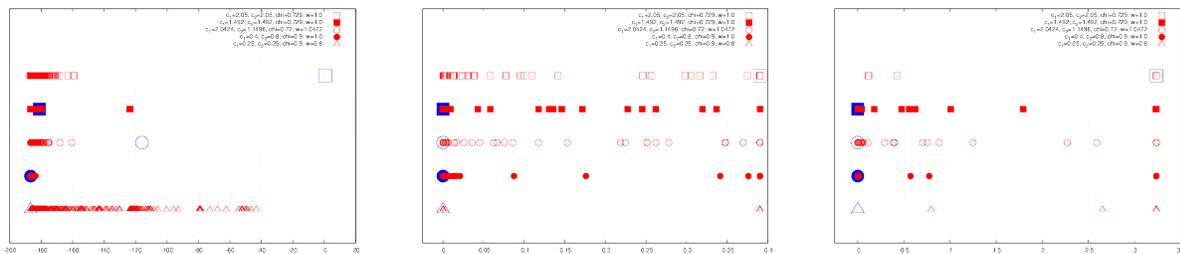


Figure 2. Three different examples of distribution of the optimal solution detected by the DPSO and PSO with all the different investigated sets for the swarm coefficients. On the horizontal axis, the values of the objective function. Each (red) dot represents the outcome of one single PSO run. DPSO gives a single (blue) dot for each selection of the parameters. Different selection of parameters are aligned on a different horizontal line. On left: algebraic function n. 9). On center: algebraic function n. 18). On right: algebraic function n. 27). Results refer to the same cases reported in Figure 1.

A different representation of the same data is reported in Figure 2. Here five different selections for the coefficients are reported. On the horizontal line, the best value of the objective function detected at the end of a single run of PSO is reported, together with the outcome of DPSO (single output). Each line in the figure represents a different set of parameters. In all the examples, DPSO is able to detect the optimal value, or a good approximation of it, for all the different selection of parameters except the first one, that is outside the convergent regime identified in⁴. In the case of the algebraic function number 27, for the first choice of parameters also PSO is not able to find the optimum, after 1000 attempts. Dispersion of data gives an idea about the number of unsuccessful runs of PSO. The probability of success reported in Tables 1-5 is computed on the base of a relative comparison, without any tolerance: this means that PSO prevails on DPSO also if the improvement of PSO is relatively small (and *vice versa*). The data reported in Figure 2 indicate how DPSO is in general able to find a reasonable approximation of the global optimum, without the need of a large number of runs in order to assess the result.

IV. Conclusions

A comparative study of a purely deterministic and a stochastic formulation of PSO has been reported. The DPSO has demonstrated to be competitive when compared with classical PSO, providing also faster progress towards a solution, in many cases. Furthermore, DPSO does not need a statistical analysis of the results. Different selection of parameters can be attempted in order to check for a deeper analysis: anyway, four different selections were considered here, while about 1000 runs were needed by DPSO.

Acknowledgments

This work has been partially supported by the AutoDrop research project.

References

- ¹Campana, E.F., Liuzzi, G., Lucidi, S., Peri, D., Pinto, A. and Piccialli, V. (2009), *New Global Optimization Methods for Ship Design Problems*, Optimization and Engineering, 10, 533-555.
- ²Clerc, M., (1999), *The swarm and the queen: toward a deterministic and adaptive particle swarm optimization*, Proceeding In Congress on Evolutionary Computation, Vol 3, 30, 1951-1957.
- ³Eberhart, R.C., and Shi, Y. (2000), *Comparing Inertia Weights and Constriction Factor in Particle Swarm Optimization*, 2000 Congress on Evolutionary Computing, Vol 1, 84-88.
- ⁴Campana, E.F., Fasano, G., Peri, D. and Pinto, A., (2006), *Particle swarm optimization: efficient globally convergent modifications*, III European Conference on Coupled Problems in Engineering, Lisbon, Portugal 5-8 June 2006.
- ⁵Campana, E.F., Fasano, G. and Pinto, A. (2010), *Dynamic system analysis for the selection of parameters and initial population, in Particle Swarm Optimization*, Journal of Global Optimization, 48, 347-397.
- ⁶Kennedy J., Eberhart R., (1995), *Particle Swarm Optimization*, Proceeding of the 1995 IEEE International Conference on Neural Networks, Perth, Australia, IEEE Service Center, Piscataway, NJ, IV:1942-1948.
- ⁷Pinto, A., Peri, D. and Campana, E.F. (2004), *Global optimization algorithms in naval hydrodynamics*, Ship Technology Research, 51(3), 123-133.
- ⁸Peri, D. (2011), *Collaborative use of a Particle Swarm Optimization algorithm and an Adaptive Covering Method for global optimization*, CNR-INSEAN Technical report.
- ⁹Peri, D., Fasano, G., Dessi, D., Campana, E.F. (2008), *Global optimization algorithms in Multidisciplinary Design Optimization*, 12th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, Victoria, British Columbia, 10-12 September 2008.
- ¹⁰Trelea, I.C. (2003), *The particle swarm optimization algorithm: convergence analysis and parameter selection*, Information Processing Letters, 85, 317-325.

Table 1. Comparison of performances for PSO and DPSO. Coefficients are: $c_1 = 2.05$, $c_2 = 2.05$, $\chi = 0.729$, $w = 1.00$. Percentage of success for DPSO: 0.0%. Percentage of success for PSO: 100.0%. DPSO prevails on PSO (on average) for the final value of the objective function (OF) in 0 cases over 29. DPSO prevails on PSO (on average) for the number of OF evaluations in 0 cases over 29.

Function	OF value		# of OF evaluations	
	DPSO	PSO	DPSO	PSO
1	0	100	0	100
2	0	100	0	100
3	0	100	0	100
4	0	100	0	100
5	0	100	0	100
6	0	100	0	100
7	0	100	0	100
8	0	100	0	100
9	0	100	0	100
10	0	100	0	100
11	0	100	0	100
12	0	100	0	100
13	0	100	0	100
14	0	100	0	100
15	0	100	0	100
16	0	100	0	100
17	0	100	0	100
18	0	100	0	100
19	0	100	0	100
20	0	100	0	100
21	0	100	0	100
22	0	100	0	100
23	0	100	0	100
24	0	100	0	100
25	0	100	0	100
26	0	100	0	100
27	0	100	0	100
28	0	100	0	100
29	0	100	0	100

Table 2. Comparison of performances for PSO and DPSO. Coefficients are: $c_1 = 1.492$, $c_2 = 1.492$, $\chi = 0.729$, $w = 1.0$. Percentage of success for DPSO: 41.58%. Percentage of success for PSO: 57.62%. DPSO prevails on PSO (on average) for the final value of the OF in 12 cases over 29. DPSO prevails on PSO (on average) for the number of OF evaluations in 10 cases over 29.

Function	OF value		# of OF evaluations	
	DPSO	PSO	DPSO	PSO
1	85	15	0	100
2	18	82	100	0
3	60	40	100	0
4	6	94	95	5
5	28	72	100	0
6	92	8	100	0
7	100	0	100	0
8	4	96	100	0
9	1	99	0	100
10	0	100	0	100
11	50	50	0	100
12	21	79	100	0
13	76	24	0	100
14	3	97	94	6
15	13	87	93	7
16	0	100	96	4
17	74	26	100	0
18	96	4	0	100
19	2	98	92	8
20	99	1	100	0
21	100	0	100	0
22	0	100	100	0
23	20	80	100	0
24	10	90	100	0
25	100	0	100	0
26	100	0	100	0
27	100	0	100	0
28	68	32	100	0
29	56	44	0	100

Table 3. Comparison of performances for PSO and DPSO. Coefficients are: $c_1 = 2.0424$, $c_2 = 1.1496$, $\chi = 0.7200$, $w = 1.0472$. Percentage of success for DPSO: 49.86%. Percentage of success for PSO: 49.48%. DPSO prevails on PSO (on average) for the final value of the OF in 16 cases over 29. DPSO prevails on PSO (on average) for the number of OF evaluations in 13 cases over 29.

Function	OF value		# of OF evaluations	
	DPSO	PSO	DPSO	PSO
1	52	48	100	0
2	66	34	0	100
3	23	77	100	0
4	2	98	90	10
5	15	85	100	0
6	89	11	100	0
7	95	5	100	0
8	0	100	99	1
9	0	100	0	100
10	77	23	0	100
11	59	41	0	100
12	26	74	100	0
13	3	97	100	0
14	3	97	90	10
15	33	67	0	100
16	0	100	0	100
17	87	13	0	100
18	89	11	0	100
19	0	100	80	20
20	98	2	100	0
21	100	0	100	0
22	88	12	0	100
23	80	20	0	100
24	88	12	0	100
25	100	0	0	100
26	100	0	0	100
27	100	0	0	100
28	87	13	0	100
29	53	47	0	100

Table 4. Comparison of performances for PSO and DPSO. Coefficients are: $c_1 = 0.40$, $c_2 = 0.80$, $\chi = 0.90$, $w = 1.00$. Percentage of success for DPSO: 46.00%. Percentage of success for PSO: 53.24%. DPSO prevails on PSO (on average) for the final value of the OF in 12 cases over 29. DPSO prevails on PSO (on average) for the number of OF evaluations in 23 cases over 29.

Function	OF value		# of OF evaluations	
	DPSO	PSO	DPSO	PSO
1	0	100	100	0
2	59	41	100	0
3	4	96	100	0
4	3	97	100	0
5	47	53	100	0
6	97	3	100	0
7	99	1	100	0
8	20	80	100	0
9	17	83	100	0
10	55	45	0	100
11	36	64	100	0
12	50	50	100	0
13	0	100	100	0
14	59	41	100	0
15	0	100	100	0
16	0	100	100	0
17	0	100	100	0
18	95	5	100	0
19	0	100	100	0
20	100	0	100	0
21	100	0	100	0
22	20	80	100	0
23	31	69	100	0
24	22	78	100	0
25	100	0	0	100
26	100	0	0	100
27	100	0	0	100
28	100	0	0	100
29	100	0	0	100

Table 5. Comparison of performances for PSO and DPSO. Coefficients are: $c_1 = 0.25$, $c_2 = 0.25$, $\chi = 0.90$, $w = 0.80$. Percentage of success for DPSO: **78.24%**. Percentage of success for PSO: **21.00%**. DPSO prevails on PSO (on average) for the final value of the OF in 25 cases over 29. DPSO prevails on PSO (on average) for the number of OF evaluations in 19 cases over 29.

Function	OF value		# of OF evaluations	
	DPSO	PSO	DPSO	PSO
1	94	6	100	0
2	85	15	0	100
3	88	12	0	100
4	18	82	99	1
5	88	12	100	0
6	100	0	100	0
7	100	0	100	0
8	86	14	100	0
9	81	19	0	100
10	28	72	100	0
11	88	12	0	100
12	91	9	100	0
13	82	18	100	0
14	58	42	100	0
15	83	17	0	100
16	36	64	0	100
17	55	45	100	0
18	100	0	100	0
19	76	24	100	0
20	100	0	100	0
21	100	0	100	0
22	94	6	0	100
23	94	6	0	100
24	85	15	0	100
25	100	0	100	0
26	77	23	100	0
27	100	0	0	100
28	95	5	100	0
29	61	39	100	0