

Why Six Informants Is Optimal in PSO

José García-Nieto
University of Malaga

Dept. Lenguajes y Ciencias de la Computación
Bulevar Louis Pasteur, 35
29071 Malaga- Spain
jnieto@lcc.uma.es

Enrique Alba

University of Malaga
Dept. Lenguajes y Ciencias de la Computación
Bulevar Louis Pasteur, 35
29071 Malaga- Spain
eat@lcc.uma.es

ABSTRACT

In a previous work, it was empirically shown that certain numbers of informants different from the standard "two" and the expensive "all" may provide the Particle Swarm Optimization (PSO) with new essential information about the search landscape, leading this algorithm to perform more accurately than other existing versions of it. Here, we extend this study by analyzing the internal behavior of PSO from the point of view of the evolvability. Our motivation is to find evidences of why such number of $6_{\pm 2}$ informant particles, perform better than other neighborhood formulations of PSO. For this task, we have evaluated different combinations of informants for an extensive set of problem functions. Using fitness-distance correlation and fitness-fitness cloud analyses we have tested the accuracy of the resulting landscape characterizations. The results suggest that, in spite of certain deviation to the global optimum, a number of 6 informants in PSO can generate new improved particles for a longer time, even in complex problems with multi-funnel landscapes.

Keywords

Particle Swarm Optimization, Fully Informed PSO, fitness-distance correlation, fitness-fitness cloud

1.INTRODUCTION

Evolvability has first been introduced in theoretical Biology for the characterization of information landscapes [16],

although it can be directly related to problem hardness for Evolutionary Algorithms (EAs), and metaheuristics in general [4]. Evolvability can be defined as the capacity of algorithms' operators to improve the fitness quality for a given problem [6]. From a complementary point of view, the outlined fitness traces (and distance to optimum) of different algorithms for a given problem can be also used to compute indicators in terms of their evolvability. In this way, it is possible to distinguish which algorithm has larger search capabilities for the tackled problem, and why.

In a previous work [3], an empirically study clearly shown that certain numbers of informant neighbors may provide Particle Swarm Optimization (PSO) with new essential information about the search landscape, leading this algorithm to perform more accurately than other existing versions of it (Standards [12] and FIPS [7] versions). Here, we contribute to this research line by analyzing the internal behavior of PSO from the point of view of the evolvability. Our main motivation is to find evidences of why certain unstructured neighborhoods, with $6_{\pm 2}$ informant particles, perform better than other neighborhood formulations.

With this aim, we have developed a new version of PSO called PSO- k (with k informant neighbors) that, ranging from the Standard PSO (2 informants) to the Fully Informed PSO (FIPS) allows to generalize the number of informant particles involved in the calculation of new particles. Then, the PSO- k with different combinations of informants has been evaluated over the CEC 2005 [13] Benchmark of functions, on which we have computed both, fitness and distance to optimum traces to calculate evolvability measures. We use this standard benchmark to avoid biasing the results and to have a high number of test problems with different complexities that endorse our claims. Since the CEC'05 Test Suite has been properly characterized in a recent work [11] by means of the Fitness Distance Correlation (fdc), we have decided to use this measure in this work together with the Fitness Cloud (fc) and Escape Probability (ep), to test the resulting landscape characterizations from all combinations of neighbors in PSO- k . With the outlined information, we expect to test the following hypothesis:

Work Hypothesis: a number of $6_{\pm 2}$ informants in the operation of PSO may compute new improved particles for longer than other PSO formulations in single and multi-funnel benchmarking problems.

In this article we will show that, on the one hand, few informants (one or two as in Standard PSO) could sometimes show high escaping probabilities and positive correlation, although evolving solutions with poor fitness values. On the

other hand, we will show that in PSO with more than 10 informants, solutions could concentrate on small regions whose sizes decrease as the neighborhood topology increases, thus confusing the search and again reducing the escape probability [9, 14].

The structure of this paper is as follows. In Section 2 we describe the PSO- k approach devoted to evaluate different combinations of number of informants. Section 3 presents the experimentation setup and the evolvability measures applied to PSO- k . In Section 4, resulted fitness-correlation and fitness-fitness indicators are analyzed. Concluding remarks are summarized in Section 5.

2. QUASY-OPTIMALLY INFORMED PSO

The canonical particle swarm optimization (PSO) [5], as well as recent standard versions of this algorithm (Standards 2006, 2007, and 2011) [12], work by iteratively generating new particles' positions located in a given problem search space. Each one of these new particles' positions are calculated using the particle's current position (solution), the particle's previous velocity, and two main informant terms: the particle's best previous location, and the best previous location of any of its neighbors.

Formally, in canonical PSO each particle's position vector \mathbf{x}_i is updated each time step t by means of the Equation 1.

$$\mathbf{x}_i^{t+1} = \mathbf{x}_i^t + \mathbf{v}_i^{t+1} \quad (1)$$

where \mathbf{v}_i^{t+1} is the velocity vector of the particle given by

$$\mathbf{v}_i^{t+1} = \omega \mathbf{v}_i^t + U^t[0, \varphi_1] \cdot (\mathbf{p}_i^t - \mathbf{x}_i^t) + U^t[0, \varphi_2] \cdot (\mathbf{b}_i^t - \mathbf{x}_i^t) \quad (2)$$

In this formula, \mathbf{p}_i^t is the personal best position the particle i has ever stored, \mathbf{b}_i^t is the position found by the member of its neighborhood that has had the best performance so far. Acceleration coefficients φ_1 and φ_2 control the relative effect of the personal and social best particles, and U^t is a diagonal matrix with elements distributed in the interval $[0, \varphi_i]$, uniformly at random. Finally, $\omega \in (0, 1)$ is called the inertia weight and influences the tradeoff between exploitation and exploration.

An equivalent version of the velocity equation was reported in [2], where Clerc's constriction coefficient χ is used instead of inertia weight as shown in Equation 3.

$$\mathbf{v}_i^{t+1} = \chi (\mathbf{v}_i^t + U^t[0, \varphi_1] \cdot (\mathbf{p}_i^t - \mathbf{x}_i^t) + U^t[0, \varphi_2] \cdot (\mathbf{b}_i^t - \mathbf{x}_i^t)) \quad (3)$$

$$\chi = \frac{2}{|2 - \varphi - \sqrt{\varphi^2 - 4\varphi}|}, \text{ with } \varphi = \sum_i \varphi_i, \text{ and } \varphi > 4 \quad (4)$$

Constriction coefficient χ is calculated, by means of Equation 4, from the two acceleration coefficients φ_1 and φ_2 , being the sum of these two coefficients what determines the χ to use. Usually, $\varphi_1 = \varphi_2 = 2.05$, giving as results $\varphi = 4.1$, and $\chi = 0.7298$. As stated by Mendes et al. [7], this fact implies that the particle's velocity can be adjusted by any number of informant terms, as long as acceleration coefficients sum to an appropriate value, since important information given by other neighbors about the search space may be neglected through overemphasis on the single best neighbor. With this assumption, Mendes et al. [7] proposed the Fully Informed Particle Swarm (FIPS), in which a particle

uses information from all its topological neighbors. In FIPS, the value φ , that is, the sum of the acceleration coefficients, is equally distributed among all the neighbors of a particle. Therefore, for a given particle i with position \mathbf{x}_i , φ is broken up in several smaller coefficients $\varphi_j = \varphi/|\mathcal{N}_i|, \forall j \in \mathcal{N}_i$. Then, the velocity is updated as follows:

$$\mathbf{v}_i^{t+1} = \chi \left[\mathbf{v}_i^t + \sum_{j \in \mathcal{N}_i} U^t[0, \varphi_j] \cdot (\mathbf{p}_j^t - \mathbf{x}_i^t) \right], \quad (5)$$

where \mathcal{N}_i is the set of neighbors of the particle i , and following the neighborhood a given structured topology (All or Fully-Connected, Ring, Square, Four-Clusters, and Pyramid). Their results show that the Square topology (with 4 informants) outperforms the other ones. Indeed, the fact of defining these neighborhoods in the swarm makes the particles to be influenced only by a certain number of neighbors and connected with static links in the graph. Once again, important information may be disregarded through overemphasis, in this case, of structured sets of neighbors. The number of informants seems to play also an important role, but with no clue on how many of them is the best choice or if even the influent issue is the neighborhood topology itself, forgetting the intermediate sources of information existing between Canonical/Standard PSO and FIPS versions.

As commented before, the possibility of adjusting the particle's velocity by an arbitrary number of terms enables us to generalize the number (k) of neighbors, from 1 to Ss (being Ss the swarm size). Therefore, a number Ss of different versions of PSO can be generated (selecting k particles of the swarm without replacement), each one of them with neighborhoods containing k particles. Obviously, if $k = Ss$ the resultant version is the FIPS algorithm with a fully connected neighborhood.

Nevertheless, since providing each k neighborhood with structured topologies is impractical due to the great number of graph combinations, we opted to simply selecting k random (uniform) informants of the swarm (S). In this way, for each particle i , and at each time step t , a different neighborhood (\mathcal{N}_i^t) with k elements is generated, and hence, the number of informants can be analyzed with independence of any structured topology. Formally, we can represent a given neighborhood as follows

$$\mathcal{N}_i^t = \{n_1, \dots, n_k\} \mid \mathcal{N}_i^t \subset S^t, \quad \forall n_j, n_h \in \mathcal{N}_i^t, n_h \neq n_j \wedge n_h \neq i \wedge n_j \neq i \quad (6)$$

Following this scheme, our new PSO- k performs as formulated in Equation 5, and using sets of k random (uniform) informant particles as neighborhoods. Then, we can evaluate all the PSO- k versions (with $k : 1 \dots Ss$) in order to discover whether an optimal value, or range of values, exists that allow to improve over the standard PSO and avoid the overhead of using topologies or computing contributions from all particles in the swarm which number of informant terms provides the new velocity vector with essential information, and discarding redundant one.

The pseudocode of PSO- k is introduced in Algorithm 1. After swarm initialization and φ_j value calculation (lines 1 to 3), the optimization process is repeated until reaching the stop condition. In this, at each iteration and for each particle, a new neighborhood is randomly (uniformly) generated by fulfilling conditions of Equation 6 (line 6). Then,

Table 1: CEC’05 benchmark test suite with functions’ properties: unimodal (U), basic multimodal (BM), Expanded (E), and Hybrid Composed (HC). Their fitness landscapes (FL) can be characterized as single-funnel (S) or multi-funnel (M) according to mean dispersion [10]

f	Name	Kind	FL	Intervals	f^*
f1	Shifted Sphere	U	S	[-100, 100]	-450
f2	Shifted Schwefel 1.2	U	S	[-100, 100]	-450
f3	Shifted Rotated High Conditioned Elliptic	U	S	[-100, 100]	-450
f4	Shifted Schwefel’s Problem 1.2 with Noise	U	S	[-100, 100]	-450
f5	Schwefel’s Problem 2.6	U	S	[-100, 100]	-310
f6	Shifted Rosenbrock’s	BM	S	[-100, 100]	390
f7	Shifted Rotated Griewank’s. Global Optimum Outside of Bounds	BM	S	[0, 600]	-180
f8	Shifted Rotated Ackley’s with Optimum on Bounds	BM	S	[-32, 32]	-140
f9	Shifted Rastrigin’s	BM	S	[-5, 5]	-330
f10	Shifted Rotated Rastrigin’s	BM	S	[-5, 5]	-330
f11	Shifted Rotated Weierstrass	BM	M	[-0.5, 0.5]	90
f12	Schwefel’s Problem 2.13	BM	M	$[-\pi, \pi]$	-460
f13	Shifted Expanded Griewank’s plus Rosenbrock’s	E	M	[-3, 1]	-130
f14	Shifted Rotated Expanded Scaffer’s F6	E	S	[-100, 100]	-300
f15	Hybrid Composition (f1-f2,f3-f4,f5-f6,f7-f8,f9-f10)	HC	M	[-5, 5]	120
f16	Rotated Version of Hybrid Composition f15	HC	M	[-5, 5]	120
f17	F16 with Noise in Fitness	HC	M	[-5, 5]	120
f18	Rot. Hybr. Comp. (f1-f2,f3-f4,f5-f6,f7-f8,f9-f10)	HC	M	[-5, 5]	10
f19	Rot. Hybr. Comp. Narrow Basin Global Optimum	HC	M	[-5, 5]	10
f20	Rot. Hybr. Comp. Global Optimum on Bounds	HC	M	[-5, 5]	10
f21	Rot. Hybr. Comp. (f1-f2,f3-f4,f5-f6,f7-f8,f9-f10)	HC	M	[-5, 5]	360
f22	Rot. Hybr. Comp. High Condition Number Matrix	HC	M	[-5, 5]	360
f23	Non-Continuous Rotated Hybrid Composition	HC	M	[-5, 5]	360
f24	Rot. Hybr. Comp. (f1,f2,f3,f4,f5,f6,f7,f8,f9,f10)	HC	M	[-5, 5]	260
f25	Rot. Hybr. Comp. Global Optimum Outside of Bounds	HC	M	[2, 5]	260

Algorithm 1 Pseudocode of PSO- k

```

1:  $t \leftarrow 0$ 
2:  $\varphi_j = \varphi/k$ 
3: initialize( $S^t$ ) /* Swarm  $S^0$  with N particles */
4: while  $t < MAXIMUM_t$  do
5:   for each particle  $i^t$  of the swarm  $S^t$  do
6:      $N_i^t = generate\_neighborhood(k, i, S^t)$  //Equation 6
7:      $\mathbf{v}_i^{t+1} = update\_velocity(\mathbf{v}_i^t, \mathbf{x}_i^t, \varphi_j, N_i^t)$  //Equation 5
8:      $\mathbf{x}_i^{t+1} = update\_position(\mathbf{x}_i^t, \mathbf{v}_i^{t+1})$  //Equation 1
9:      $\mathbf{p}_i^{t+1} = update\_local\_best(\mathbf{p}_i^t, \mathbf{x}_i^{t+1})$ 
10:   end for
11:    $t \leftarrow t + 1$ 
12: end while
13: Output:  $b$  /*The best solution found*/

```

particle’s velocity, current position, and local best position are updated (lines 7 to 9). Finally, the best so far particle position is returned as output (line 13).

3. EXPERIMENTAL ANALYSIS

We have implemented our PSO- k using the MALLBA library [1] in C++, a framework of metaheuristics. Following the specifications proposed in CEC’05 [13] experimental procedure, we have performed 25 independent runs of PSO- k for each test function and for each $k \in \{1, \dots, S_s\}$ neighborhood. For simplicity, our study is illustrated for dimension $D = 30$ (number of continuous variables) of all functions, although we have performed additional experiments showing that our claims hold for other dimensions ($D = 10$ and $D = 50$). In the results, we are reporting the Mean Error of the best solutions found in the 25 independent runs. For a solution \mathbf{x} , the error measure is defined as: $f(\mathbf{x}) - f^*$, where f^* is the optimum fitness of the function. The maximum number of fitness evaluations is $10,000 \times D$, which constitutes the stop condition.

The used test suite is composed by 25 functions (see Table 1) with different features [13]: unimodal, multimodal,

separable, non-separable, shifted to biased optimum, rotated, single-funnel, multi-funnel, and hybrid composed. Functions f1 to f5 are unimodal, functions f6 to f12 are basic multimodal, functions f13 and f14 are expanded, and functions f15 to f25 are composed by several basic ones. Functions 11, 12, 13, and 15 to 25 are characterized to multi-funnel according to fdc and mean dispersion analyses in [10, 11].

The parameter setting applied to PSO- k follows the specification proposed in Clerc’s approach [2]: the acceleration coefficient has been set to $\varphi = 4.1$, and constriction coefficient to $\chi = 0.7298$. The swarm size has been set to 30 particles in order to simplify the experimental procedure.

3.1 Evolvability on PSO

Before we start to discuss the results, we describe here the evolvability measures used in this work and their application to the particular case of PSO.

- *Fitness-distance* analysis quantifies the relation between the fitness of particles $f(\mathbf{x}_i)$ in the landscape and their distances to the nearest global optimum \mathbf{x}_{min} [6]. Assuming that we are minimizing, correlation means that both, the fitness and the distance to the nearest optimum, decreases whereas anti-correlation is registered when the fitness decreases and the distance to the nearest optimum increases (or vice versa). Distance between points in continuous domains are measured using Euclidean distance d_E . Then, the *fitness-distance correlation* (fdc) can be quantified by the following coefficient (ratio of fdc):

$$r_{fdc} = \frac{c_{fd}}{s_f \cdot s_d}, \text{ being } c_{fd} = \frac{1}{n} \sum_{i=1}^n (f_i - \bar{f}) \cdot (d_i - \bar{d}) \quad (7)$$

where f_i is the fitness of solution $f(\mathbf{x}_i)$, d_i is the distance of the solution \mathbf{x}_i to the optimum $d_i = d_e(\mathbf{x}_i, \mathbf{x}_{min})$, \bar{f} , \bar{d} , s_f , and s_d are the means and standard deviations of the fitness and distance samples, respectively.

In our experiments, for each function and for each PSO- k version, we have used all the samples of each one of the 25 independent runs to compose the r_{fdc} . The median value (25 runs) of this coefficient is then reported in the following section. In addition, a number of 10,000 uniform random samples have been also collected for each function with the aim of comparing the resulting r_{fdc} coefficients to those computed by PSO- k . In this way, as suggested in [4], it is possible to detect wrong correlation coefficients resulting from PSO samples trapped in large basins of attraction.

An interpretation of this coefficient can be as follows [11]: a positive r_{fdc} near to 1 means globally convex single-funnel topologies. A value around 0 of this coefficient may indicate plateau shape landscapes with tiny sharp basins and problems without any global structure. A negative value of r_{fdc} indicate the existence of “deceiving” landscapes, where an optimizer (PSO in our case) perceives poor objective function values closer to the minimum than farther away.

- *Fitness-fitness* or *Fitness Cloud* (fc) [15] analysis is basically a plot of fitness values of individuals against fitness values of their neighbors. By definition of fc , for each sampled individual \mathbf{x}_i with fitness f_i , generate k neighbors by applying a genetic operator to \mathbf{x}_i , and let be \overline{f}_i the mean fitness of all neighbors of \mathbf{x}_i . Then, the set of points $\{(f_1, \overline{f}_1), \dots, (f_n, \overline{f}_n)\}$ is taken as the fitness cloud. In our particular case, we are interested in computing the fitness cloud by plotting the fitness (f'_i) of a new particle (\mathbf{x}'_i) that is generated from their informants (using velocity rule in Equation 5), and the mean fitness \overline{f}_i of all these particles of \mathbf{x}_i . Now, the set of points $\{(f'_1, \overline{f}_1), \dots, (f'_n, \overline{f}_n)\}$ can be analyzed as the fitness cloud of PSO- k for a problem.

- *Escape Probability* (ep) [8] analysis considers the number of steps required to escape from a local optimum. It is defined as $P(f_i) = \frac{1}{S_i}$, where S_i denotes the average number of steps required to find an improving move starting in an individual with fitness value f_i . In the context of our PSO, we will average the improving intervals (evaluation steps) computed by each particle of the swarm to calculate the escape probability through the iteration process. In next sections, several examples of ep are plotted for f24.

4. ANALYSIS AND DISCUSSION

Since in this study we have concentrated on a swarm size with 30 particles, the number of PSO- k 's versions is 30, from PSO-1 to PSO-29, plus PSO-30 represented by the so called FIPS-All. Therefore, we have undergone the evaluation of each version of PSO- k with every function in the benchmark. Summing up, 25 independent runs for each algorithm version and for each function have been performed, resulting in a total number of $25 \times 25 \times 30 = 18,750$ experiments. The resulting r_{fdc} coefficients are summarized in Figure 1 as bar graphs. In this figure, the r_{fdc} coefficients computed from random samples are also plotted as dotted lines. Not surprisingly, these last r_{fdc} values agree with the ones computed in [11].

In the following, we make several observations about Figure 1. After this, we will focus on interesting observations concerning fc and ep indicators. Later, we will focus our analysis on different problem dimensions.

4.1 Fitness-Distance Analysis

In general, we can first observe from Figure 1 that PSO with $k=6\pm 2$ informants shows $r_{fdc} > 0$ in single-funnel func-

tions (f1-f10) and $r_{fdc} \leq 0$ in multi-funnel ones (f11-f25). This means that using $k=6\pm 2$ informants, our proposal is able to distinguish between globally convex and/or deceiving (multiple convexities) landscapes, and irrespectively of the problem modality. Nevertheless, three exceptions are registered: for f8, with plateau landscape and hence, highly dependent to the swarm initialization, and for f16 and f17, which were characterized as bi-funnel in [11]. For these two last functions, there is a certain probability of evolving samples (from initialization) on the funnel where the optimum is located [14], which contributes to compute a r_{fdc} higher than 0.

Second, when comparing PSO-6s' samples with random ones we can check in Figure 1 that there are few cases in which the resulting r_{fdc} coefficients (in these two cases) are closed to each other. Concretely for functions f7, f8, f11, f14, and f17, the median r_{fdc} of PSO-6 is practically similar to the dotted line of random samples. On the contrary, for the remaining twenty functions, the r_{fdc} coefficients of random samples show quite different values to those of PSO- k with 6 informants. For these last functions, PSO-6 seems to perform a guided search towards promising regions in the landscape, not visited by the random samples, and far from the reference global optimum in CEC'05 [13] source code. Probably, new particles in PSO-6 are guided towards other (different to the reference) global optima since the difference (in r_{fdc} coefficients) with random samples is particularly observed for multi-modal functions (with multiple global optima).

Third, for almost all the functions (excepting f8, f16, and f17) a number of informants close to 6 in PSO- k registered in general the lowest values of r_{fdc} . This suggests that using few (<4) or too much (>10) informants could create a tendency to move toward non interesting regions, leading the algorithm to show an early stagnation and obtaining fake correlated solutions with large fitness values and far from the optimum.

To see this last more in detail, Figure 2 shows the fitness-distance plots resulted from independent executions of PSO with 2, 6, 12, and 29 informants, for functions f5, f15, and f24. We can observe in this figure that using few informants (2 in this case), the algorithm shows correlation for the three functions, even though evolving solutions with poor fitness values since the low diversity of neighbors leads the early convergence of the algorithm. With more than 10 informants, solutions are again correlated, although concentrating on several small regions on multi-funnel landscapes (as plotted for f15 and f24) that may confuse the particles' decisions and may slow down their escape from these regions. Using 6 informants in PSO is the best trade-off between fitness-distance correlation and fitness quality. A special case can be observed for f24 (highly deceptive), where solutions are gradually improving although showing certain distance with regards to the global optimum provided by the CEC'05 test suite.

4.2 Fitness-Fitness Analysis

Figure 3 plots the Fitness Clouds of functions f5, f15, and f24, generated from independent executions of PSO with 2, 6, 12, and 29 informants. For the three functions, a number of 6 informants is able to keep for longer the generation of new better particles with improving fitness (f' , see Section 3.1). In fact, regression lines calculated from dis-



Figure 1: Each plot shows the median r_{fdc} coefficients in the Y-axis (out of 25 independent runs) of the different PSO- k versions (for the 30 possible values of k) in X-axis, for all benchmark functions with dimension $D = 30$. The dotted lines in these graphs represent the r_{fdc} coefficients computed from 10,000 uniform random samples for each function

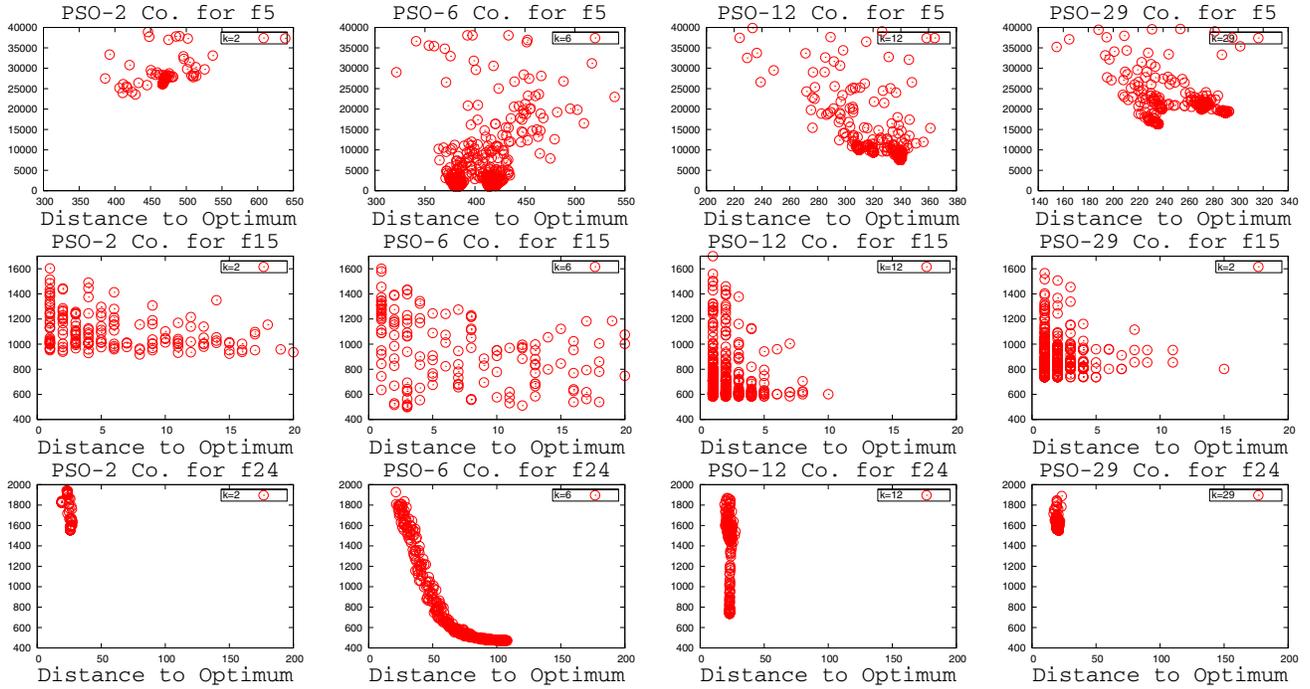


Figure 2: Fitness-distance plots of functions f_5 , f_{15} , and f_{24} , generated from independent executions of PSO with 2, 6, 12, and 29 informants (dimension $D=30$)

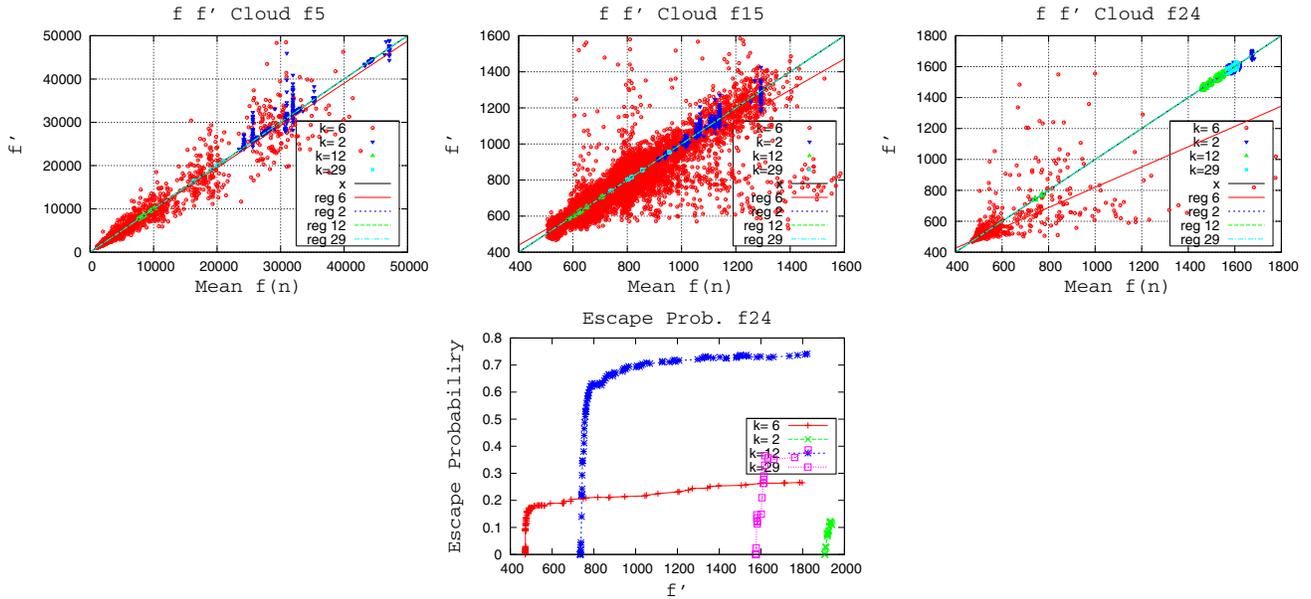


Figure 3: Illustrative fitness-fitness clouds $\{(f'_1, \bar{f}_1), \dots, (f'_n, \bar{f}_n)\}$ of functions f_5 , f_{15} , and f_{24} , generated from independent executions of PSO- k with 2, 6, 12, and 29 informants. Graph in bottom-right plots the Escape Probability with regards to the new fitness value f' concerning f_{24} (see this curve from right to left)

tribution clouds (of 2, 12, and 29 informants) generally fit the diagonal line of plots, whereas regression lines of 6 informants' clouds show positive slopes in final evolution steps, with regard to diagonal lines. This means that new fitness values f' calculated using 6 informants provokes a higher probability of mean fitness $\bar{f}(n)$ improvement of neighbors in the search landscape .

An illustrative example of this behavior can be explicitly observed in Figure 3 bottom-right, where the Escape Probability concerning f_{24} is progressively plotted. Solutions evolved by PSO-6 generally show a moderated ep progress, although reaching a deeper basin of local optimum, e.g., better fitness values. A different behavior is shown by PSO with 12 informants which initially shows high values of ep ,

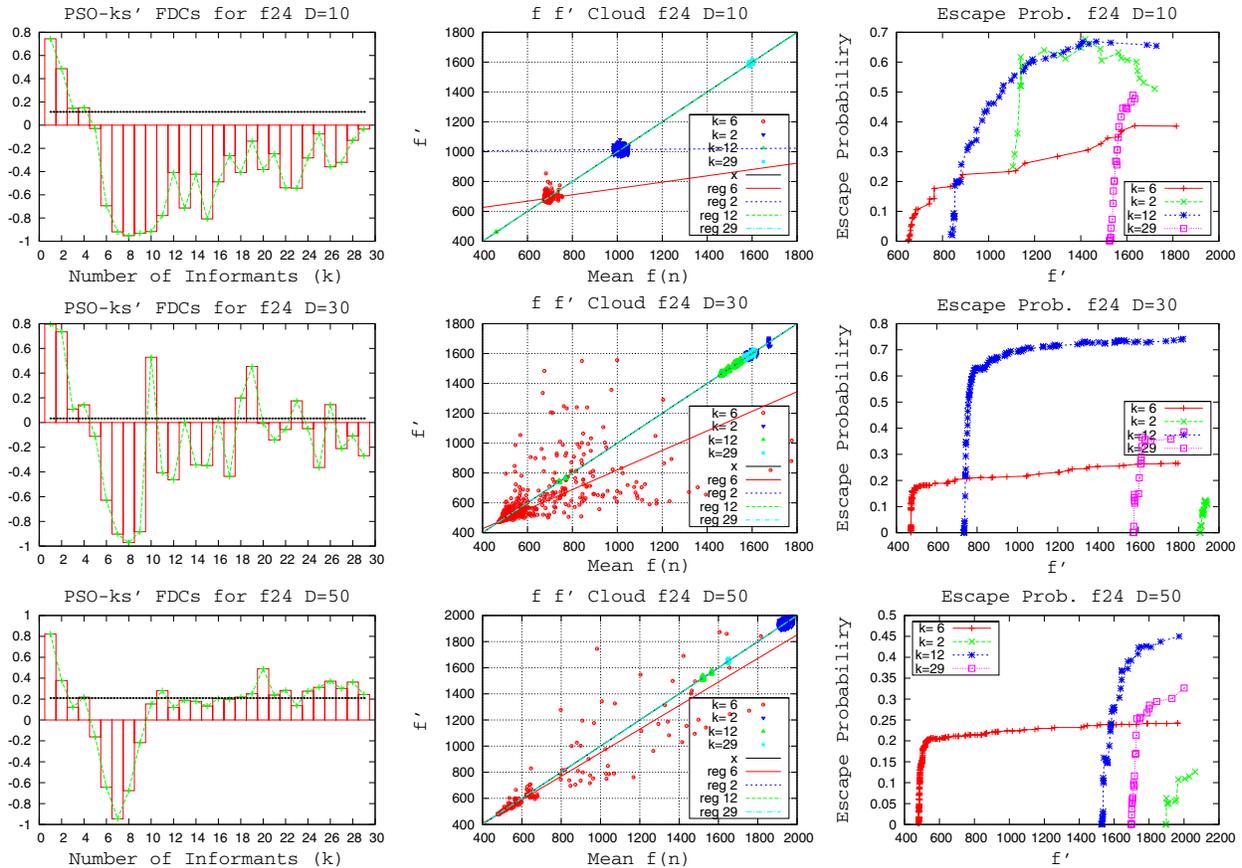


Figure 4: Plot graphs of the three studied evolvability measures: f_{dc} (left), f_c (center), and ep (right); and three different problem dimensions: 10 (top), 30 (center) and 50 (bottom) variables, for f24

although decreasing with fair fitness values. This means that 12 informants initially provides the PSO with high exploration ability, but tending to search in non interesting regions. PSO with 2 informants generally shows a low degree of ep with poor fitness quality, meaning that for f24, its particles are quickly trapped in local basins, and hence more informants are required for the movement (velocity calculation) of these particles.

4.3 Results Hold Over Different Dimensions

In this section, the influence that scaling to different problem dimensions may have on the selection of the number of informants (in PSO- k) is studied. In concrete, we illustrate with the interesting case of function 24 with dimensions $D = 10, 30,$ and 50 variables, although the same profiles can be got for the rest of functions. The swarm size was set to 30 particles as in initial experiments, and our goal is to see how sensible are all PSO- k versions to the problem size.

Figure 4 shows the graphs concerning the three studied evolvability measures: f_{dc} (left), f_c (center), and ep (right); and three different problem dimensions: 10 (top), 30 (center) and 50 (bottom) variables, for f24. From this figure, we can observe that in general the behavior of all PSO- k versions is kept constant along with the problem dimension. This means that our previous claims could be generalized for scaling problems. There are certain slight differences with regards to the f_{dc} indicator in PSO- k s with more than 10

informants, which increases the $r_{f_{dc}}$ coefficient as the problem dimension is higher. Once again, more fitness-distance correlation (in $D = 50$) corresponds to solutions with poor fitness values and far from the global optimum, whereas a negative $r_{f_{dc}}$ represents solutions with better fitness values, although far from the reference global optimum. In this sense, the escape probability plots in Figure 4 illustrate this behavior since PSO- k with 12 informants reaches moderate fitness values in $D = 10,$ and $D = 30,$ but with poor fitness quality in $D = 50$. In the case of PSO- k with 6 informants, the ep indicator shows a similar curve for the three different problem dimensions (as well as similar $r_{f_{dc}}$) with successful fitness values, which lead us to suggest 6 informants as the best option also when facing scaling problems.

4.4 Confirming the Initial Hypothesis

Once we have analyzed the evolvability of the different PSO- k versions performed here, and based on observations of the actual values of the algorithm, we make a summary of lessons learned and whether they confirm the initial hypothesis of this work:

Why a number of $6_{\pm 2}$ informants in the operation of PSO may compute new improved particles for longer than other PSO formulations in complex single and multi-funnel benchmarking problems?.

For this, we can argue the following:

1. Using few informants (<4) leads the PSO- k to show a positive fitness-distance correlation, although evolving solutions with poor fitness values and far from global optima. With more than 10 informants, solutions are again correlated, although concentrating on small non interesting regions of the landscape. Using 6 informants is the best trade-off between fitness-distance and fitness quality.
2. The regression lines calculated from 6 informants' clouds show positive slopes in final evolution steps, meaning that new fitness values f' calculated from 6 informant neighbors improve more frequently the mean fitness $f(n)$ of these neighbors.
3. Solutions evolved by PSO-6 generally show a moderate but maintained *ep* progress, finally reaching a deeper basins, e.g., better fitness values than with other number of informants.
4. In general the behavior of all PSO- k versions is stable for different problem dimensions. In concrete, PSO-6 shows quite similar evolvability indicators for the three studied dimensions.

5. CONCLUSIONS

Based on the previous experience, in this work we have used different evolvability measures with the aim of shedding light on the question that a number of $6_{\pm 2}$ informants in the operation of PSO may compute new improved particles for longer, even in complex problems with single and multi-funnel landscapes. After the experimentation in the context of the CEC'05 benchmark suite and using different problem dimensions, we have tested our initial research question showing that 6 informants in PSO is the best trade-off between fitness-distance and fitness quality. This is noticeable since the algorithm is simple (Occam's razor), and these results together with the previous ones in [3] suggest that the benchmark should be updated with more varied functions.

We have shown why and who PSO works by using a new kind of probability of escape analysis that is useful for any PSO. We give a comprehensive analysis beyond a couple of functions or one dimension: we address classes of problems.

As future work, we are working on creating new evolvability measures well-adapted to the operation of PSO. We will also experiment with other standard benchmarks to test if $6_{\pm 2}$ informants in PSO is appropriated to be used on them.

Acknowledgments

Authors acknowledge funds from the CICE of the Junta de Andalucía, under contract P07-TIC-03044 (DIRICOM <http://diricom.lcc.uma.es>) and Spanish Ministry of Sciences and Innovation (MICINN) and FEDER under contracts TIN2011-28194 (RoadMe <http://roadme.lcc.uma.es>) and TIN2008-06491-C04-01 (M* <http://mstar.lcc.uma.es>). José García-Nieto is supported by grant BES-2009-018767 from the MICINN

6. REFERENCES

- [1] E. Alba, G. Luque, J. García-Nieto, G. Ordonez, and G. Leguizamón. MALLBA: A software library to design efficient optimisation algorithms. *Int. Jour. of Innov. Comp. and Appl. (IJICA)*, 1(1):74–85, 2007.
- [2] M. Clerc and J. Kennedy. The particle swarm - explosion, stability, and convergence in a multidimensional complex space. *IEEE Trans. Evol. Comp.*, 6(1):58 – 73, 2002.
- [3] J. García-Nieto and E. Alba. Empirical computation of the quasi-optimal number of informants in particle swarm optimization. In *Proceedings of GECCO'11*, pages 147–154, New York, NY, USA, 2011. ACM.
- [4] T. Jones and S. Forrest. Fitness distance correlation as a measure of problem difficulty for genetic algorithms. In *Proceedings of the 6th International Conference on Genetic Algorithms*, pages 184–192, San Francisco, CA, USA, 1995. Morgan Kaufmann Publishers Inc.
- [5] J. Kennedy and R. C. Eberhart. *Swarm Intelligence*. Morgan Kaufmann, S. Fco., California, 2001.
- [6] G. Lu, J. Li, and X. Yao. Fitness-probability cloud and a measure of problem hardness for evolutionary algorithms. In *EvoCOP'11*, volume 6622 of *LNCS*, pages 108–117. Springer, 2011.
- [7] R. Mendes, J. Kennedy, and J. Neves. The Fully Informed Particle Swarm: Simpler, Maybe Better. *IEEE Tran. on Evol. Comp.*, 8(3):204 – 210, 2004.
- [8] P. Merz. Advanced fitness landscape analysis and the performance of memetic algorithms. *Evol. Comput.*, 12:303–325, September 2004.
- [9] M. A. Montes de Oca and T. Stützle. Convergence behavior of the fully informed particle swarm optimization algorithm. In *Proceedings of GECCO'08*, pages 71–78, New York, NY, USA, 2008. ACM.
- [10] C. L. Müller, B. Baumgartner, and I. Sbalzarini. Particle swarm cma evolution strategy for the optimization of multi-funnel landscapes. In *Evolutionary Computation, 2009. CEC '09. IEEE Congress on*, pages 2685 –2692, may 2009.
- [11] C. L. Müller and I. F. Sbalzarini. Global characterization of the CEC 2005 fitness landscapes using fitness-distance analysis. In *Proceedings of EvoNUM, EvoApplications'11*, pages 294–303, Berlin, Heidelberg, 2011. Springer-Verlag.
- [12] PSO-Central-Group. Standard PSO 2006, 2007, and 2011. Technical Report [online] <http://www.particleswarm.info/>, Particle Swarm Central, January 2011.
- [13] P. N. Suganthan, N. Hansen, J. J. Liang, K. Deb, Y.-P. Chen, A. Auger, and S. Tiwari. Problem Definitions and Evaluation Criteria for the CEC'05 Special Session on Real-Parameter Optimization. Technical Report KanGAL Report 2005005, Nanyang Technological University, 2005.
- [14] A. M. Sutton, D. Whitley, M. Lunacek, and A. Howe. PSO and multi-funnel landscapes: how cooperation might limit exploration. In *Proceedings of GECCO'06*, pages 75–82, New York, NY, USA, 2006. ACM.
- [15] L. Vanneschi, M. Clergue, P. Collard, M. Tomassini, and S. Vérel. Fitness clouds and problem hardness in genetic programming. In *GECCO'04*, volume 3103 of *LNCS*, pages 690–701. Springer, 2004.
- [16] A. Wagner. *Robustness and evolvability in living systems*. Princeton Studies in Complexity. Princeton University Press, 2005.