Empirical Computation of the Quasi-optimal Number of Informants in Particle Swarm Optimization

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 the standard particle swarm optimization (PSO), a new particle's position is generated using two main informant elements: the best position the particle has found so far and the best performer among its neighbors. In fully informed PSO, each particle is influenced by all the remaining ones in the swarm, or by a series of neighbors structured in static topologies (ring, square, or clusters). In this paper, we generalize and analyze the number of informants that take part in the calculation of new particles. Our aim is to discover if a quasi-optimal number of informants exists for a given problem. The experimental results seem to suggest that 6 to 8 informants could provide our PSO with higher chances of success in continuous optimization for well-known benchmarks.

Keywords

Particle Swarm Optimization, Fully Informed PSO, CEC 2005 Benchmark of Functions

1. INTRODUCTION

The canonical particle swarm optimization (PSO) [6], as well as recent standard versions of this algorithm (Standards 2006, 2007, and 2011) [10], 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 [3], where Clerc's constriction coefficient χ is used instead of inertia weight as shown in Equation 3.

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

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

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 $\chi = 0.7298$ [4, 16]. As stated by Mendes et all. [8, 9], 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 all. [8] 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:

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

where $\mathcal{N}_i$ is the set of neighbors of the particle i, and following the neighborhood a given topology. Figure 1 illustrates the topologies used by Mendes et al. [8] as the ones with most successful performances in a previous work [7]. These topologies are: All, 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 good issue is the neighborhood topology itself or the fact that only a few informants are used.

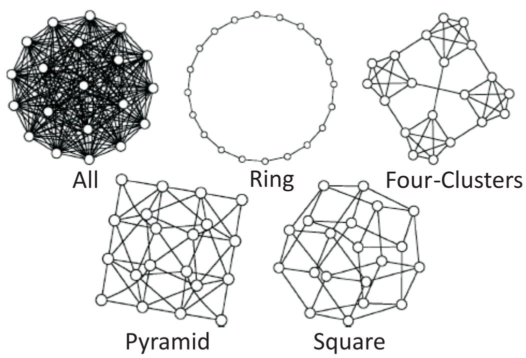

Figure 1: Topologies used by Mendes et al. [8]. Each particle has a number of fixed neighbors in the swarm (All=N-1; Ring=2; Four-Clusters=4,5; Pyramid=3,5,6; Square=4)

All this motivated us to generalize the number of neighbors that influence particles, as well as the different configurations of topologies, in order to discover whether there exists a quasi-optimal number of informants that take part in the calculation of the velocity vector for a particular problem. Then, our initial hypothesis is that certain numbers (sets) of informant neighbors may provide new essential information about the search process, hence leading the PSO to perform more accurately than existing versions of this algorithm, for a number of well-known benchmark problems in continuous optimization.

With the aim of researching in this line, we have designed in this work a generalized version of PSO that follows the information scheme of FIPS (with Clerc's constriction coefficient), but having as a free variable the number of informants in the calculation of the velocity vector. To evaluate our PSO with all possible configurations we have followed the experimental framework (with 25 problem functions) proposed in the Special Session of Continuous Optimization of CEC'05 [12]. The performed analysis and comparisons (against Standard PSO and FIPS versions) will help us to claim if there are informant sets other than 2 and N that yield a more efficient PSO.

The remainder of this article is organized as follows. Next section presents the "Quasi-optimal Informed" version of PSO worked here. Section 3 describes the experimentation procedure and the parameter settings. In Section 4, experimental results are reported with analysis and discussions. Finally, concluding remarks and future work are given in Section 5.

2. THE QUEST FOR AN OPTIMAL NUMBER OF INFORMANTS

As previously commented, 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 neighborhood "ALL", as illustrated in Figure 1.

Nevertheless, since providing each k neighborhood with structured topologies is impracticable due to the great number of graph combinations, we have opted in this work to simply selecting k random (uniform) informants of the swarm (S). 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

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

Following this scheme, we have designed for this work a new PSO called Optimally Informed Particle Swarm (OIPS), which performs as formulated in Equation 5, and using sets of k random (uniform) informant particles as neighborhoods. Then, we can evaluate all the OIPS-k versions (with k : $1 \ldots Ss$) in order to discover whether an optimal value, or range of values, exist that allows to improve over the standard PSO and avoid the overhead of using topologies or computing contributions from all particles in the swarm.

Algorithm 1 Pseudocode of OIPS-k

```
1: t ← 0
2: φ_j = φ/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:       v_i^{t+1} = update_velocity(v_i^t, x_i^t, φ_j, N_i^t) //Equation 5
8:       x_i^{t+1} = update_possiton(x_i^t, v_i^{t+1}) //Equation 1
9:       p_i^{t+1} = update_local_best(p_i^t, x_i^{t+1})
10:   end for
11:   t ← t + 1
12: end while
13: Output: b /*The best solution found*/
```

The pseudocode of OIPS-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, 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).

Table 1: CEC'05 test suite of functions

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

3. EXPERIMENTAL SETUP

In this section, we present the experimental methodology and statistical procedure applied to evaluate the different versions of OIPS-k and to compare them. We have followed the experimental framework presented in the Special Session on *Real-Parameter Optimization at CEC'05* [12].

We have implemented our OIPS-k using the MALLBA library [1] in C++, a framework of metaheuristics. Following the specifications proposed in CEC'05 experimental procedure, we have performed 25 independent runs of OIPS-k for each test function and for each $k \in \{1, \ldots, Ss\}$ neighborhood. We use this standard benchmark to avoid biasing the results to concrete functions, and to have a high number of test problems that endorse our claims. For simplicity, the study has been made with dimension $D = 30$ (number of continuous variables), although an additional analysis with different problem dimensions is also included in Section 4.5. In the results, we are reporting the Maximum, the Median, the Minimum, and 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 has been set to $10,000 \times D$, which constitutes the stop condition.

To analyze the results, we have used non-parametric statistical tests, since several times the distributions of results did not follow the conditions of normality and homoskedasticity [5]. Therefore, the Median error (and not the Mean error), out of 25 independent runs, has been used for analysis and comparisons. In particular, we have considered the application of the Friedman's ranking test, and use the Holm's multicompare test as post-hoc procedure [11].

The test suite of the CEC'05 benchmark is composed by 25 functions with different features [12]: unimodal, multimodal, separable, non-separable, shifted, rotated, 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 functions. This way, our new proposals are evaluated under quite different conditions of modality, separability, and composition. Table 1 shows the function names, bounds, and optimum values.

The parameter setting applied to OIPS-k (in Table 2) follows the specification of the Standard PSO in [10]. The swarm size has been set to 30 particles in order to simplify the experimentation procedure due to space constraints. Nevertheless, as done with the problem dimension, additional experiments concerning different swarm sizes will be also provided in Section 4.4.

Table 2: Parameter setting used in OIPS-k

Description	Parameter	Value
Swarm size	Ss	30
Acceleration coefficient	φ	4.1
Constriction coefficient	χ	0.7298

4. ANALYSIS AND DISCUSSION

In this section, we first present an analysis concerning the influence of the different neighborhood sizes (k) in OIPS-k. Since we will present a clear range for the informant number to be used, later we evaluate them against standard algorithms in the literature. Finally, further analysis concerning the computational effort, the swarm size, and the problem dimension are performed.

4.1 Impact of the Number of Informants

First, we focus on the different number of informants constituting all possible combinations of neighborhoods.

Since in this experimentation we have concentrated on a swarm size with 30 particles, the number of OIPS-k's ver-

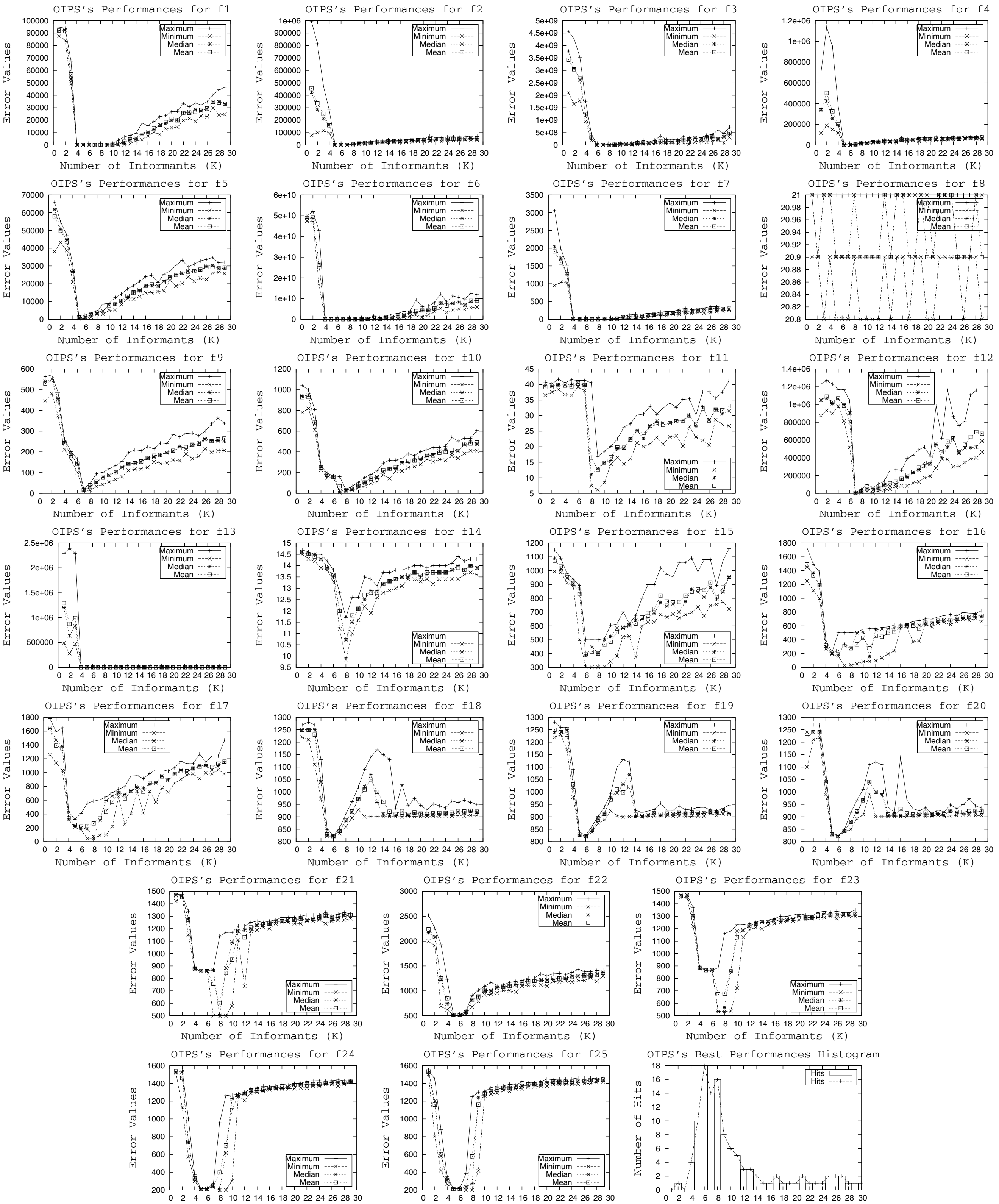

Figure 2: Each plot contains the performance (Maximum, Median, Minimum, and Mean error values out of 25 independent runs) of the different OIPS-k versions for the 30 possible values of k, and for all CEC'05 functions. The graph in the bottom-right figure contains the frequency histogram of best performance (number of Hits)

Table 3: Median of the error for the 6 compared algorithms and for all the CEC'05 functions

Alg./Func.	Standard PSO 2007	FIPS-ALL	FIPS-Usquare	OIPS-6	OIPS-U[6,8]	OIPS-HE{6,8}
f1	5.68E-14	4.12E+04	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
f2	5.08E+00	5.83E+04	3.38E-09	**3.41E-13**	1.54E+02	2.89E-05
f3	4.28E+07	5.39E+08	**6.36E+05**	2.06E+06	1.14E+06	3.02E+06
f4	5.05E+03	7.46E+04	1.50E+04	**6.30E-03**	4.03E+03	1.33E+00
f5	2.89E+03	3.09E+04	3.34E+03	1.25E+03	2.42E+03	**1.21E+03**
f6	1.66E+01	1.22E+10	**1.62E+01**	**1.62E+01**	2.70E+01	2.19E+01
f7	1.23E-02	3.33E+02	9.86E-03	**5.68E-14**	7.76E-03	**5.68E-14**
f8	**2.09E+01**	2.10E+01	2.10E+01	2.10E+01	2.10E+01	2.10E+01
f9	2.39E+01	2.73E+02	1.69E+01	1.51E+02	1.49E+01	**1.09E+01**
f10	1.80E+02	4.82E+02	**2.79E+01**	1.60E+02	1.44E+02	1.51E+02
f11	3.78E+01	**3.45E+01**	3.89E+01	3.97E+01	3.99E+01	3.98E+01
f12	2.88E+05	7.78E+05	**2.12E+03**	7.78E+05	4.61E+03	7.40E+05
f13	1.19E+01	6.67E+01	**2.99E+00**	1.37E+01	3.79E+00	1.19E+01
f14	1.40E+01	1.39E+01	1.28E+01	1.36E+01	**1.16E+01**	1.34E+01
f15	5.58E+02	9.46E+02	**2.39E+02**	3.57E+02	3.16E+02	3.06E+02
f16	2.12E+02	8.22E+02	**4.88E+01**	1.87E+02	1.69E+02	1.75E+02
f17	2.51E+02	1.19E+03	**7.24E+01**	2.01E+02	1.88E+02	1.90E+02
f18	8.30E+02	9.22E+02	8.31E+02	8.23E+02	8.42E+02	**8.22E+02**
f19	8.30E+02	9.33E+02	8.31E+02	**8.22E+02**	8.41E+02	**8.22E+02**
f20	8.30E+02	9.22E+02	8.30E+02	**8.23E+02**	8.41E+02	**8.23E+02**
f21	8.00E+02	1.32E+03	8.63E+02	8.58E+02	**6.80E+02**	8.58E+02
f22	5.23E+02	1.39E+03	5.51E+02	5.12E+02	5.74E+02	**5.11E+02**
f23	8.67E+02	1.34E+03	8.70E+02	8.66E+02	**5.54E+02**	8.66E+02
f24	2.16E+02	1.42E+03	2.21E+02	**2.12E+02**	2.30E+02	**2.12E+02**
f25	2.16E+02	1.44E+03	2.21E+02	**2.12E+02**	2.31E+02	**2.12E+02**
Hits	1	1	9	9	4	**10**

sions is 30, from OIPS-1 to OIPS-29, plus OIPS-30 represented by the so called FIPS-All. Therefore, we have undergone the evaluation of each version OIPS-k with the benchmark of functions CEC'05. 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 results are plotted in Figure 2, and several observations can be made from it:

- A number of 6 informants in the neighborhood makes the algorithm to perform with success in practically all functions. This is quite interesting since we can then mention the version OIPS-6 as the most promising one, and study its main features with regards to other parameters (swarm size, φ) and versus other algorithms (Standard PSOs, FIPS, etc.) in the next sections.

- For almost all functions, the interval between 5 and 10 informants concentrates most of the successful runs. In this sense, the plot at bottom-right in Figure 2 shows the histogram concerning the frequency in which each OIPS-k obtained the best results in the studied functions. This leads us to suspect that less than 5 informants is a deficient value of k not really taking particles out of the found local optima during the evolution, while more than 10 informants is redundant.

- In this sense, a number of 8 informants is also appropriate showing good performances in efficacy, although it is costly compared to OIPS-6. A new research question then comes to scene: could we create still better PSO's by using a range of informants during the search instead of betting for just one single constant value? Therefore, combining 6 and 8 informants in neighborhoods could be a source of new competitive algorithms.

- Another interesting observation concerns the behavior of all OIPS-k's versions in certain sets of functions that show similar curves of performance. Thus, functions f1 to f5, unimodal ones, show accurate performances from $k = 5$ in advance. Rastrigin's funcitons f9 and f10 draw quite similar curves with the best performance in $k = 7$. Hybrid composed functions, from f15 to f25, show also high performance for $k = 6$.

- Curiously, biased functions to the same optimum f^* share similar curve shapes of OIPS-k's performances. For example, functions f1 to f4, with $f^* = -450$, functions f9 and f10, with bias to -330, functions f15, f16, and f17 which are biased to 120, functions f18, f19, and f20 with $f^* = 10$, functions f21, f22, and f23 with $f^* = 360$, and specially functions f24 and f25 biased to 260, they all show close curve shapes in Figure 2. An intriguing question is whether the CEC'05 benchmark is having an unknown feature in the induced landscapes that makes a given kind of PSO to perform better than others. If we could find such feature in the landscape domain we could create good algorithms from the start for these and other problems.

4.2 Performance Comparisons

We compare here the best OIPS-k version (OIPS-6) against the Standard PSO and other successful versions of FIPS with the aim of studying how well informed our proposal is.

Additionally, we have developed two simple combinations of OIPS-ks with neighborhoods of 6 and 8 informants, namely OIPS-U[6,8] and OIPS-HE{6,8}. The former randomly (uniform) chooses a value in {6, 7, 8} as the number of informants to be used in every step of the optimization process. The later version, OIPS-HE{6,8}, performs the first half of the optimization process with 6 informants, and the remaining second half with 8 (Half Evolution, HE).

Table 3 shows the resulted median errors of compared OIPS-k versions for all CEC'05 functions. In addition, Standard PSO 2007, FIPS-ALL, and FIPS-USquare algorithms are also compared. We have added the FIPS-USquare (with informants in a square neighborhood) to this comparison since it was the version of FIPS that reported the best results in terms of performance in Mendes et al. [8].

In Table 3, the best resulted median errors are marked in bold, and the last row summarizes the number of best results (Hits) obtained by each algorithm. As clearly observable, OIPS-HE{6,8} obtains the higher number of Hits (10 out of 25), followed by OIPS-6 and FIPS-USquare with 9. In the case of OIPS-U[6,8], a limited number of Hits of 4 leads us to suspect that the random combination of neighborhood sizes in the interval [6,8] does not make the most of these values. In general, we can also notice that all the algorithms obtain the best median errors for one function, at least, so even the Standard PSO 2007 in f8 and the FIPS-ALL in f11 report the best median error.

Table 4: Average Rankings Friedman's test of resulted median errors

Algorithm	Ranking
OIPS-HE{6,8}	2.58
OIPS-6	2.86
FIPS-Usquare	2.88
OIPS-U[6,8]	3.26
Standard PSO 2007	3.76
FIPS-ALL	5.66

Statistically, Table 4 contains the results of an Average Rankings Friedman's test [11] applied to the median results[1] of Table 3. We can see that OIPS-HE{6,8} is the best ranked algorithm (with 2.58), followed by OIPS-6, and FIPS-Usquare. In contrast, FIPS-ALL is the worst ranked algorithm according to this test. This means that the complete scheme of information adopted in FIPS-ALL could damage the generation of new particles by incorporating noise and redundant information to them. In this sense, the FIPS-ALL shows even worse ranking than the Standard PSO 2007, whose set of informants (SI) is included in the set of the ALL topology, that is, $SI \subset ALL$.

Table 5: Holm test of resulted median errors for $\alpha = 0.05$

i	algorithm	$z = (R_0 - R_i)/SE$	p	Holm
5	FIPS-ALL	5.820652884342103	5.86E-9	0.0100
4	Standard PSO 2007	2.2299903907544407	0.0123	0.0125
3	OIPS-U[6,8]	1.2850792082313736	0.1987	0.0166
2	FIPS-Usquare	0.5669467095138413	0.5707	0.0250
1	OIPS-6	0.5291502622129168	0.5967	0.0500

More precisely, Table 5 contains the results of a multicomparison Holm's test [11] on the median errors got by the compared algorithms. In this, the best ranked technique in the Friedman test, OIPS-HE{6,8} is compared against all other algorithms. Holm's procedure rejects those hypotheses of equality of distributions that have a *p-value*≤0.0125. Then, we can state that, for the tackled benchmark of functions (CEC'05), and according to this test, OIPS-HE{6,8} is statistically better than Standard PSO 2007 (*p-value*=0.0123) and than FIPS-ALL (*p-value*=5.86E-9) algorithms.

As a further analysis, we have applied the Holm's test (with α=0.05) to compare our best algorithms OIPS-HE{6,8} and OIPS-6, against G-CMA-ES [2], the one with the best performance in the Special Section of CEC'05. In this case, the statistical test results with a *p-value*=0.288 when comparing OIPS-HE{6,8} with G-CMA-ES, and a *p-value*=0.137 when comparing OIPS-6 with G-CMA-ES. Therefore, no statistical differences exist between the performed median error distributions of these algorithms, i.e., none of them performs better than the other one.

[1]Friedman statistic considering reduction performance (distributed according to chi-square with 5 degrees of freedom: 45.93714285714339).

4.3 Computational Effort

We present in this section some remarks about the computational effort. To execute the experiments, we have used the computers of the laboratories of the Departament of Computer Science of the University of Málaga (Spain). Most of them are equipped with modern dual core processors, 1GB RAM, and Linux S.O., having into account that there are more than 200 computers, that means that up to 400 cores have been available. To run all the programs, we have used the Condor [15] middleware that acts as a distributed task scheduler (each task dealing with one independent run of OIPS-k).

Figure 3 plots the mean running time (seconds) in which all the versions of OIPS-k have found the best mean error for all functions. The mean running times used by OIPS-U[6,8], OIPS-HE{6,8}, Standard PSO 2007, FIPS-ALL, and FIPS-USquare algorithms are also plotted. As expected, the running time increases with the number of informants in OIPS-k, although it seems to stabilize from OIPS-15 to OIPS-30 (FIPS-ALL). We have to mention that this last version (OIPS-30) does not use the random selection of informants, since all particles in the swarm are involved in the velocity calculation. This led us to suspect that the time the random selection of informants spends is not significant with regards to the time of calculating the new velocity vector (information time).

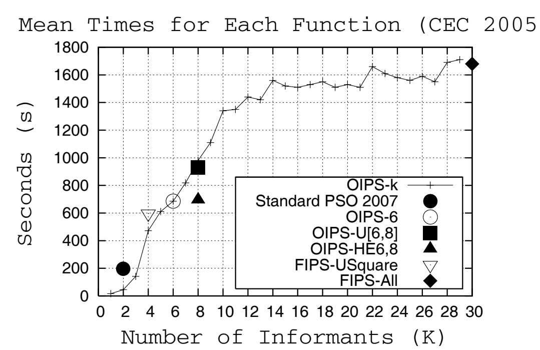

Figure 3: Mean running time (seconds) in which all the versions of OIPS-k have found the best mean error for all functions. The mean running time used by the rest of algorithms is also plotted

We can also observe in this figure that the Standard PSO 2007 required the shortest running time (excepting OIPS-1 and OIPS-2), since only two informants are involved in the velocity calculation: the personal (**p**) and the global best (**b**) positions. Almost all the remaining compared algorithms required similar running times, between 600 and 900 seconds, since they used a close number of informants (from 4 to 8) in their operations. Obviously, the algorithm using the higher number of informants, OIPS-30 (FIPS-ALL), required the longest running time.

4.4 Influence of the Swarm Size

Another interesting feature of OIPS-k that we also analyze here concerns the influence that the swarm size experts on the optimal number of informants k in the neighborhood. In this sense, we have carried out a series of additional experiments in which, four configurations of swarm sizes (with 10, 30, 50, and 100 particles) have been set in the different OIPS-k algorithms for a number of neighborhoods (k). Specifically, we have evaluated from OIPS-1 to OIPS-10, OIPS-30, OIPS-50, and OIPS-100 versions, each one of them with the four possible configurations of swarm size.

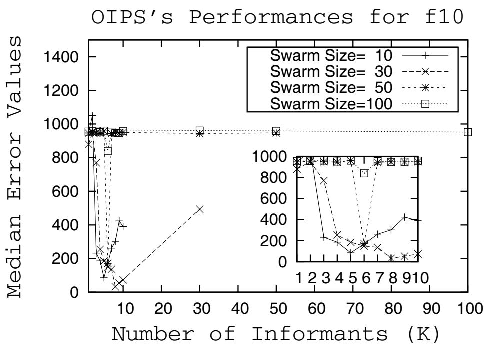

Figure 4: Influence of the different swarm sizes in OIPS-k. The median fitness values are plotted for swarm sizes with 10, 30, 50, and 100

Figure 4 shows the plot of the median error values resulted from the experiments with different sizes of swarm, for function f10 (of CEC'05). We have selected this function since it shows a representative behavior similar to the ones obtained on the remaining functions. We can effectively observe that all the best median errors are obtained by OIPS-k versions with neighborhoods included in the range of $k \in [5, 9]$. Therefore, with independence of the number of particles in the swarm, the empirical optimal number of informants required is included in this interval, and even for larger swarm sizes (with 100 particles) the performance of OIPS-k is enhanced using those small neighborhoods.

4.5 Influence of the Problem Dimension

Similar to the previous analysis, the potential influence that scaling to larger problem dimensions may have on the selection of the neighborhood size (in OIPS-k) is studied in this section. In this case, the experiments are focused on the resolution of large scale problems, as the ones found in the context of the Special Session CEC'08 [14] and CEC'10 [13] (the stop condition is $5,000 \times D$ fitness evaluations). In concrete, we have worked with the Shifted Ackley's function (f6 in CEC'08 and f3 in CEC'10) with dimensions $D = 30$, 50, 100, and 500 variables. The swarm size was set to 30 particles as in initial experiments, and our goal is to see how sensible is OIPS-k to problem size.

Figure 5 plots the median errors resulted for all OIPS-k configurations. Once again, the informed OIPS-k with neighborhood sizes close to 6 informant particles show the best performance for almost all problem dimensions. Only when dealing with 50 variables, OIPS-k with 4 and 5 informants obtain the best median errors values, also close to OIPS-6. Therefore, as expected, increasing the number of

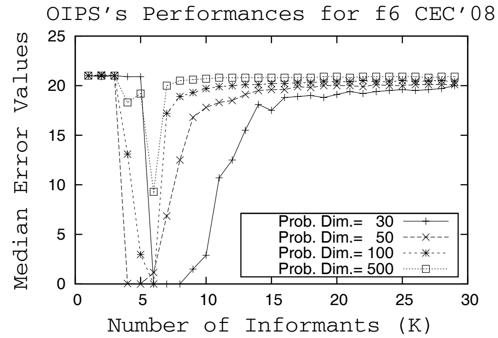

Figure 5: Influence of different problem dimensions in OIPS-k. Study made with the Shifted Ackley's function, f6 in CEC'08 and f3 in CEC'10

variables in the problem dimension does not seem to variate the behavior of the OIPS-k versions. In fact, for the Shifted Ackley's function, the curve shapes representing each problem dimension follow similar patterns to practically all plots in Figure 2.

5. CONCLUSIONS

In this paper, we generalize and analyze the number of informants that take part in the calculation of new particles. For this, we have created a new version of Informed PSO, called OIPS-k with the possibility of managing any neighborhood size k, from 1 informant to all of them in the swarm (FIPS-ALL). A series of experiments and comparisons have been carried out in the scope of the CEC'05 benchmark of functions. The influence of the number of informants, the problem dimension, and the swarm size have been analyzed.

The following conclusions can be extracted:

1. A number of 6 informants in the neighborhood makes the algorithm to perform with high success in practically all functions. This means that, at least for the popular continuous benchmarks, researchers should consider OIPS-6 instead of the standard PSO.

2. In fact, comparisons with Standard PSO 2007, FIPS-ALL and FIPS-USquare versions, and other algorithms in the state of the art (G-CMA-ES), lead us to propose our OIPS-6 (and its combination OIPS-HE{6,8}) as a prominent optimizer.

3. For almost all functions, the interval between 5 and 10 informants concentrates most of the successful runs. We suspect that less than 5 informants is a deficient value of k, since hard landscapes make PSO stuck in local optima with no guide enough coming from the two best particles in the swarm to scape from them. More than 10 informants is redundant, since lost of particles are just representing the same movements as others, that is, there are classes of equivalence in the swarm (basins of attraction) that are providing redundant samples to interfere with the velocity computation for $k > 10$ (including FIPS).

4. The behavior of all OIPS-k versions in certain sets of functions show similar curves of performance. Unimodal, Rastrigin's, Hybrid Composition functions show quite similar curve shapes concerning the impact of k.

5. Functions biased to the same optimum f^* share similar curve shapes of OIPS-k's performances. For example, functions f1 to f4, with $f^* = -450$, and specially functions f24 and f25 biased to $f^* = 260$. If this leads to a rethinking of existing benchmarks or not is open to discussion.

6. In general, the higher the number of informants (involved in the velocity calculation), the longest the running time required.

7. Each OIPS-k version shows quite similar behavior in our experiments independently of the swarm size, and independently of the problem dimension. This means that OIPS-k is having additional features making it scalable and resistent to constrained execution (memory-restricted, at least).

As future work, we are interested in investigating other elemental features of the PSO algorithm as well as to apply new concepts of the Standard PSO 2011, to "informed" versions of this algorithm. Besides, we plan to perform analytical investigations on what is the probability that 6 to 8 particles take "most times" a given arbitrary particle solution out of the many local optima that are usually found in complex problems.

A research in progress is to study whether the CEC'05 benchmark is having an unknown feature in the induced landscapes that makes a given kind of PSO to perform better than others. If we could find such feature in the landscape domain we could create good algorithms from the start for these and other similar problems.

Acknowledgments

J. García-Nieto and E. Alba acknowledge funds from the Spanish Ministry of MICINN and FEDER European Union under contract TIN2008-06491-C04-01 (M* project with URL `http://mstar.lcc.uma.es`) and CEIC, Junta de Andalucía, under contract P07-TIC-03044 (DIRICOM project with URL `http://diricom.lcc.uma.es`). José García-Nieto is supported by grant BES-2009-018767 from 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. Journal of Innovative Computing and Applications (IJICA)*, 1(1):74–85, 2007.

[2] A. Auger and N. Hansen. A restart CMA evolution strategy with increasing population size. *IEEE Congress on Evolutionary Computation*, 2:1769–1776, 2005.

[3] M. Clerc and J. Kennedy. The particle swarm - explosion, stability, and convergence in a multidimensional complex space. *IEEE Transactions on Evolutionary Computation*, 6(1):58 – 73, Feb 2002.

[4] R. Eberhart and Y. Shi. Comparing Inertia Weights and Constriction Factors in Particle Swarm Optimization. In *Proceedings of the IEEE Congress on Evolutionary Computation CEC'00*, volume 1, pages 84–88, La Jolla, CA, USA, 2000.

[5] S. García, D. Molina, M. Lozano, and F. Herrera. A study on the use of non-parametric tests for analyzing the evolutionary algorithms' behaviour: a case study on the CEC'2005. *Journal of Heuristics*, 15(6):617–644, 2009.

[6] J. Kennedy and R. C. Eberhart. *Swarm Intelligence*. Morgan Kaufmann Publishers, San Francisco, California, 2001.

[7] J. Kennedy and R. Mendes. Population structure and particle swarm performance. In *Proceedings of the Congress of Evolutionary Computation CEC'02*, volume 2, pages 1671–1676, Washington, DC, USA, 2002. IEEE Computer Society.

[8] R. Mendes, J. Kennedy, and J. Neves. The Fully Informed Particle Swarm: Simpler, Maybe Better. *IEEE Transactions on Evolutionary Computation*, 8(3):204 – 210, June 2004.

[9] A. S. Mohais, R. Mendes, C. Ward, and C. Posthoff. Neighborhood re-structuring in particle swarm optimization. In *LNCS 3809. Proceedings of the 18th Australian Joint Conference on Artificial Intelligence*, pages 776–785. Springer, 2005.

[10] PSO-Central-Group. Standard PSO 2006, 2007, and 2011. Technical Report [online] http://www.particleswarm.info/, Particle Swarm Central, January 2011.

[11] D. J. Sheskin. *Handbook of Parametric and Nonparametric Statistical Procedures*. Chapman & Hall/CRC, 2007.

[12] 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, Singapore and Kanpur, India, 2005.

[13] K. Tang, X. Li, P. N. Suganthan, Z. Yang, and T. Weise. Benchmark Functions for the CEC'10 Special Session and Competition on Large Scale Global Optimization. Technical Report Nature Inspired Computation and Applications Laboratory, USTC, Nanyang Technological University, anyang Technological University, China, 2010.

[14] K. Tang, X. Yao, P. N. Suganthan, C. MacNish, Y. P. Chen, C. M. Chen, and Z. Yang. Benchmark functions for the CEC'08 special session and competition on large scale global optimization. Technical report, Nature Inspired Computation and Applications Laboratory, USTC, China, November November 2007.

[15] D. Thain, T. Tannenbaum, and M. Livny. Distributed computing in practice: the condor experience. *Concurrency - Practice and Experience*, 17(2-4):323–356, 2005.

[16] I. C. Trelea. The particle swarm optimization algorithm: convergence analysis and parameter selection. *Inf. Process. Lett.*, 85:317–325, March 2003.