Real-Parameter Optimization Performance Study on the CEC-2005 benchmark with SPC-PNX

Pedro J. Ballester **John Stephenson** **Jonathan N. Carter** **Kerry Gallagher**

Imperial College London, Department of Earth Science and Engineering,
RSM Building, Exhibition Road, London SW7 2AZ, UK
Corresponding author: p.ballester@imperial.ac.uk

Abstract- This paper presents a performance study of a Real-parameter Genetic Algorithm (SPC-PNX) on a new benchmark of Real-parameter Optimisation problems. This benchmark provides a systematic way to compare different optimisation methods on exactly the same test problems. These problems were designed to be hard as they incorporate features that have been shown to pose great difficulty to many optimisation methods.

1 Introduction

Many Evolutionary Algorithms (EAs) have been devised to tackle Real-parameter Optimisation problems. It is common practice to assess the suitability of the EA by optimising a set of analytical test problems and comparing its performance with the known solutions. However, it has been argued elsewhere that commonly used test problems may contain symmetries (e.g. symmetric initialisation around the global minimum [1]) which are unlikely to be present in a real world problem. Consequently, an EA exploiting these symmetries, and thus obtaining a good performance of such test problems, is likely to perform worse when applied to real world optimisation problems. Clearly, there is a need for a new benchmark with test problems designed to avoid symmetries that could be exploited by EAs. The organisers of the IEEE CEC-2005 Special Session on Real-parameter Optimisation have designed such a benchmark and have invited researchers to test their optimisation methods on it. The benchmark's problem definition files, codes and evaluation criteria are available online[1].

Genetic Algorithms (GAs) have been used extensively for Real-parameter Optimisation (an introductory review is by Deb [2]). GAs are a class of search and optimisation techniques inspired by ideas from Darwinian Evolution and Natural Genetics. Information that describes a candidate solution is encoded into a data structure (the genome), which in Real-parameter GAs corresponds to an array of real numbers. Using operations, analogous to those in nature, new offspring solutions are bred from parent solutions. After a number of generations (iterations), an ensemble of improved solutions is obtained. In minimisation, the measure of the solution quality (fitness) is given by how low is the objective function value. A reference text for Real-parameter GAs (also known as Real-coded GAs) is by Herrera et al. [3].

[1]Download Evaluation-Criteria-10-Mar-05.pdf and Intro-2-funs-09-Mar-05.pdf following the route 'Shared Documents' → 'CEC2005' at http://staffx.webstore.ntu.edu.sg/MySite/Public.aspx?accountname=epnsugan

The aim of this paper is to carry out a performance study of a Real-parameter GA known as SPC-PNX on the CEC-2005 benchmark. The rest of the paper is organised as follows. Section 2 briefly describes SPC-PNX. In Section 3, the experimental setup is presented. The results of the performance study are discussed in Section 4. Lastly, we present our conclusions in Section 5.

2 SPC-PNX

The adopted optimisation method is a steady-state Real-parameter GA called SPC-PNX [1]. It has been successfully applied to several nonlinear parameter estimation problems arising in Earth Sciences [4].

In SPC-PNX, two parents are selected from the current population of size N to produce λ children (offspring) through the crossover operator. The objective function value associated with each child is thereafter evaluated. Offspring and current population are then combined so that the population remains at a constant size through the replacement operator. These four steps (selection, crossover, fitness evaluation and replacement) form one GA generation. Details of SPC-PNX's selection, crossover and replacement schemes are explained next.

2.1 Selection

Uniform random selection, without replacement, is used to select two parents from the current population. Unusually for a GA, fitness is not taken into account during the selection process. This can help exploit the information contained in diverse, low fitness solutions.

2.2 Crossover

In this work, the PNX crossover operator [1] is used. This parent-centric crossover is self-adaptive in the sense that the spread of the possible offspring solutions depends on the distance between the parents, which decreases as the population converges. In addition, PNX is an isotropic operator as it does not preferentially search along any particular direction. Another beneficial feature is that PNX has a non-zero probability of generating offspring over the whole search space. These features contribute to a broader exploration of the search space (see [1] for a discussion on this issue).

In PNX, for each of the λ offspring, we proceed as follows to determine its j^{th} gene (y_j). First, we draw a single random number $w \in [0, 1]$, we use the form $y_j^{(1)}$ if $w < 0.5$

0-7803-9363-5/05/$20.00 ©2005 IEEE.

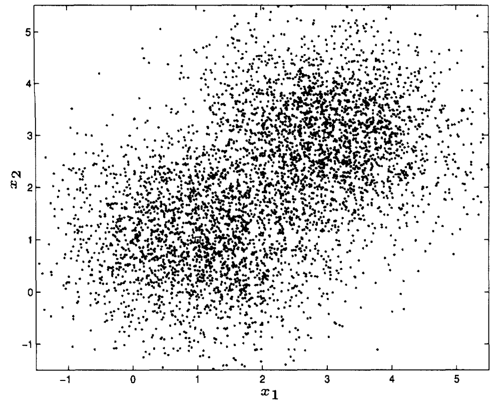

Figure 1: Offspring bred from parents $\vec{x}^{(1)} = (1, 1)$ and $\vec{x}^{(2)} = (3, 3)$ for PNX with $\eta = 2$.

and $y_j^{(2)}$ if $w \geq 0.5$. Once this choice is made, the same selected form is used for every component j. The forms are

$$y_j^{(1)} = N(x_j^{(1)}, |x_j^{(2)} - x_j^{(1)}|/\eta)$$

$$y_j^{(2)} = N(x_j^{(2)}, |x_j^{(2)} - x_j^{(1)}|/\eta)$$

where N(μ,σ) is a random number drawn from a gaussian distribution with mean μ and standard deviation σ, $x_j^{(i)}$ is the j^{th} component of the i^{th} parent and η is a tunable parameter. The larger is the value of η the more concentrated is the search around the parents. In Fig. 1, a large number of offspring are bred from two fixed parents in order to illustrate how PNX operates.

A constrained version of PNX was also implemented so that the resulting algorithm concentrates the search within the initialisation region. This was carried out by introducing the following modification: accept the calculated offspring component y_j if it is within the initialisation bounds, otherwise use the corresponding parent component x_j instead (i.e. $y_j = x_j$).

2.3 Replacement

The scaled probabilistic crowding scheme [1] is used as the replacement operator. This an improvement of the probabilistic crowding scheme [5], where the closest of the two preselected individuals ($\vec{x}^{cst}$) enters a probabilistic tournament with the offspring ($\vec{x}^{ofp}$), with culling likelihoods given by

$$p(\vec{x}^{ofp}) = \frac{f(\vec{x}^{ofp})}{f(\vec{x}^{ofp}) + f(\vec{x}^{cst})}$$

$$p(\vec{x}^{cst}) = \frac{f(\vec{x}^{cst})}{f(\vec{x}^{ofp}) + f(\vec{x}^{cst})}$$

where $f(\vec{x})$ is the objective function value for an individual $\vec{x}$. If the differences in function values across the population are small with respect to their absolute values, the likelihoods would be very similar in all cases.

The scaled probabilistic crowding replacement is introduced to avoid this situation. First, for each offspring, N_{REP} individuals from the current population are selected at random. These individuals then compete with the offspring for a place in the population according to the following culling likelihoods

$$p(\vec{x}^{ofp}) = \frac{f(\vec{x}^{ofp}) - f_{best}}{f(\vec{x}^{ofp}) + f(\vec{x}^{cst}) - 2f_{best}}$$

$$p(\vec{x}^{cst}) = \frac{f(\vec{x}^{cst}) - f_{best}}{f(\vec{x}^{ofp}) + f(\vec{x}^{cst}) - 2f_{best}}$$

where f_{best} is the function value of the best individual in the offspring and selected group of N_{REP} individuals.

This replacement scheme has several beneficial features. The fittest individual in the replacement competition does not always win (unless it is also the best individual found so far). This helps to prevent premature convergence. Crowding schemes such as this promote the creation of subpopulations that explore different regions of the search space. This has been shown (e.g. [6]) to be beneficial for creating diverse optimal solutions and to increase the effectiveness in finding the global minimum. Also, it implements elitism in an implicit way. If the best individual in either offspring or current parent population enters this replacement competition will have probability zero of being culled.

3 Experimental Setup

The CEC-2005 benchmark contains 25 generic test problems. Each of them presents a real-valued function to optimise and specifies the initialisation region as well as the required accuracy level. These functions are available for three dimensionalities: D=10, D=30 and D=50[2]. The optimisation process is restricted to a maximum number of function evaluations (Max_FES), which are 10^5 (D=10) and $3 \cdot 10^5$ (D=30). The participants in the Special Session have been asked to comment on a number of issues including how we tuned the algorithm, make comparisons between selected subsets of problems, give an estimation of the algorithm complexity and present convergence graphs of the 30-variable functions. Further details about the benchmark and evaluation criteria can be found in the problem definition and evaluation criteria documents. The codes of the test problems were downloaded from the same web address as these documents (see Section 1).

SPC-PNX is implemented in two languages: C and Matlab 7.0. These codes will be made available to facilitate the reproducibility of our results as well as for their use in other problems. The numerical experiments were carried out on several platforms (SUN UltraSparc5 workstations, PCs with Windows 2000 and PCs with Linux Red Hat).

SPC-PNX contains four tunable parameters: N, λ, N_{REP} and η. In this study, we fix $\eta = 2.0$, $\lambda = 1$ and N_{REP}=2. Therefore, the only control parameter to adjust is the population size N. This is done to simplify the tuning of the algorithm, although we expect that tuning the remaining control parameters would result in significant performance improvements. For the two problems

[2]The Special Session organisers ultimately recommended not to include the 50-variable functions because of the page limit.

with the global minimum outside the initialisation region (f7 and f25), the unconstrained version of the algorithm is used. The rest of problems use the constrained version. The approach to constrain the search within the initialisation bounds is introduced at the crossover level, as explained in Section 2.2. Note that the differences between both versions (constrained and unconstrained) are minimal in terms of complexity.

4 Results

The first performance study is carried out on the 10-variable test problems (D=10). The only control parameter to adjust is the population size N. The procedure to adjust this parameter was broadly as follows. The unimodal problems (f1-f5) were allowed seven different settings for N, each of them run with three different initial populations. Therefore, the cost of the tuning was roughly $1.8 \cdot 10^6$ function evaluations (FES=1.8e+6). The basic multimodal functions (f6-f14) were allowed about ten different settings of N, each of them run three times. Thus, this requires $3 \cdot 10^6$ FES. Lastly, the hybrid functions (f15-f25) were permitted six different settings of N, each of them run three times. This results in a total of $2.1 \cdot 10^6$ FES. Table 1 shows the values of N that were finally used to provide the results reported in Table 2, 3, 4 and 8. For each problem, these results show the error (difference between the obtained function value and the actual global minimum) at three checkpoints (10^3, 10^4 and 10^5 FES). Among the 25 runs of each problem, only the error value for the 1^{st} (Best), 7^{th}, 13^{th} (Median), 19^{th} and 25^{th} (Worst) are reported.

The second performance study is on the 30-variable test problems. The cost of the tuning process was considerably lower than in the D=10 case. This is because we directly set the same values of N that were finally used in the D=10 case. Subsequently, we set a higher value of N for each problem and make several runs to check whether the results were better that the initial setting. Three of these runs were made on the unimodal (f1-f5) and basic multimodal (f6-f14). Only one tuning run for each hybrid problem (f15-f25) could be afforded because of the large amount of computation that these problems require. The values for N in the D=30 case are shown in Table 1. These values were used to provide the results reported in Table 5, 6, 7 and 9. The results are presented in the same format as in D=10, but with an additional checkpoint at $3 \cdot 10^5$ FES. The convergence rate for each of these problems is shown in Figs. 2, 3, 4, 5 and 6.

The results allow us to study how the different problem features affect the performance of the adopted optimisation method. However, a word of caution is needed here, as this performance does not depend exclusively on the problem features. The algorithm may also have bad performance because of poor tuning of its control parameters. The latter is more likely to happen with the hybrid functions, as very few settings were tested in those cases. The first comparison is between problems 1, 2 and 3, from which it is clear that a high condition number strongly deteriorates the algorithm's performance. Secondly, noise does not seem

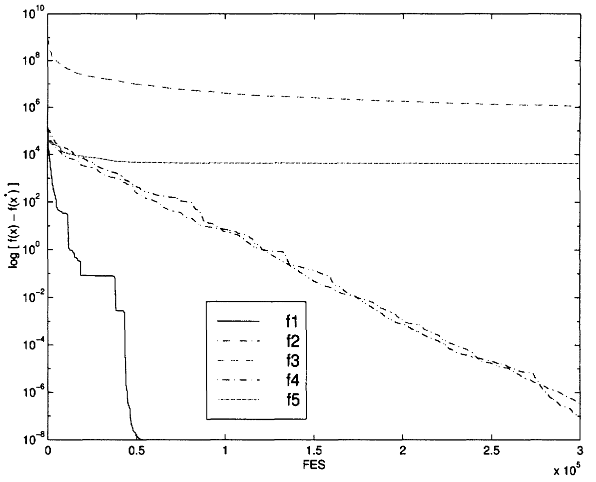

Figure 2: Convergence graphs for problems 1 to 5 in D=30.

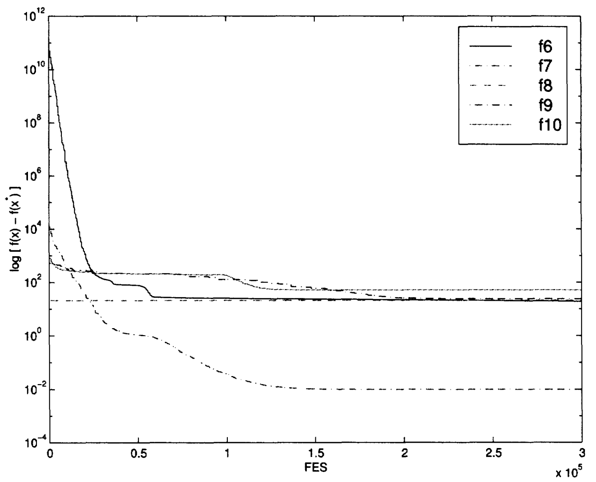

Figure 3: Convergence graphs for problems 6 to 10 in D=30.

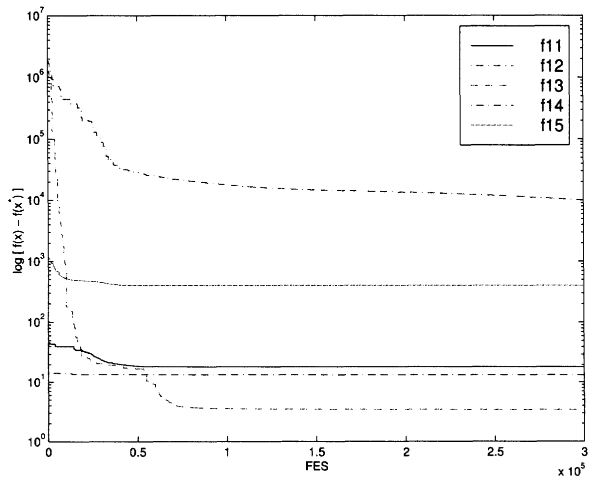

Figure 4: Convergence graphs for problems 11 to 15 in D=30.

Table 1: Used values of N on the D=10 and D=30 problems.

D=10	1	2	3	4	5	6	7	8	9	10	11	12
N	20	40	45	40	35	60	150	1000	225	200	100	200
D=30	1	2	3	4	5	6	7	8	9	10	11	12
N	40	40	45	40	100	60	150	1000	300	200	200	200

D=10	13	14	15	16	17	18	19	20	21	22	23	24	25
N	50	100	150	150	150	100	100	100	50	400	400	150	200
D=30	13	14	15	16	17	18	19	20	21	22	23	24	25
N	55	100	150	150	150	300	300	300	300	300	300	150	300

Table 2: Error values achieved when FES=1e3, FES=1e4 and FES=1e5 for Problems 1-8 (D=10).

1e3	1	2	3	4	5	6	7	8
1^{st}	1.9542e+1	5.3942e+2	5.3646e+6	1.3117e+3	4.0088e+3	1.3332e+5	3.3102e+2	2.0939e+1
7^{th}	7.1020e+1	2.3885e+3	7.4867e+6	2.4042e+3	5.2648e+3	1.2350e+6	8.6298e+2	2.1108e+1
13^{th}	1.4973e+2	3.5740e+3	1.2201e+7	3.7942e+3	6.4504e+3	1.7182e+6	1.0154e+3	2.1189e+1
19^{th}	2.0043e+2	4.6847e+3	1.6604e+7	5.5116e+3	6.9951e+3	6.2229e+6	1.1791e+3	2.1228e+1
25^{th}	1.2874e+3	8.4256e+3	2.9709e+7	9.7532e+3	8.9238e+3	1.0758e+8	1.7425e+3	2.1288e+1
Mean	2.0162e+2	3.7366e+3	1.3406e+7	4.2884e+3	6.3654e+3	7.6545e+6	1.0316e+3	2.1171e+1
Std	2.4979e+2	1.8944e+3	6.7426e+6	2.3973e+3	1.1679e+3	2.1082e+7	3.1256e+2	8.7288e-2
1e4	1	2	3	4	5	6	7	8
1^{st}	6.4899e-9T	6.1474e-2	1.1178e+5	8.4152e-2	1.7468e+0	1.2400e-1	1.2142e+0	2.0939e+1
7^{th}	8.4223e-9T	5.4431e-1	3.5411e+5	4.0268e-1	6.4554e+0	4.2080e+0	2.0573e+0	2.1048e+1
13^{th}	9.0660e-9T	1.5391e+0	7.5638e+5	2.1300e+0	1.5149e+1	7.8263e+0	3.3062e+0	2.1095e+1
19^{th}	9.8524e-9T	7.0763e+0	1.0712e+6	8.4987e+0	2.6237e+1	1.1994e+2	5.9570e+0	2.1117e+1
25^{th}	4.9779e-8	2.2521e+1	2.4924e+6	2.3656e+1	1.1099e+2	9.3645e+2	1.4951e+1	2.1184e+1
Mean	1.1793e-8	4.4453e+0	8.5866e+5	5.4125e+0	2.6301e+1	1.4958e+2	4.7060e+0	2.1082e+1
Std	9.8954e-9	5.4351e+0	6.4215e+5	7.0855e+0	2.9593e+1	2.8172e+2	3.6406e+0	5.9910e-2
1e5	1	2	3	4	5	6	7	8
1^{st}	6.4899e-9T	8.7414e-9T	7.7457e+2	7.6909e-9T	7.9417e-9T	1.8594e-2	9.8573e-3	2.0813e+1
7^{th}	8.3963e-9T	9.5342e-9T	5.6899e+4	9.1910e-9T	8.8876e-9T	8.7512e-1	3.6926e-2	2.0964e+1
13^{th}	8.9819e-9T	9.7326e-9T	8.6585e+4	9.5550e-9T	9.2950e-9T	3.7770e+0	6.3961e-2	2.1010e+1
19^{th}	9.8060e-9T	9.8336e-9T	1.3310e+5	9.8676e-9T	9.7643e-9T	4.8391e+0	1.1073e-1	2.1025e+1
25^{th}	9.9931e-9T	9.9951e-9T	3.5216e+5	9.9807e-9T	9.8735e-9T	1.4830e+2	2.4620e-1	2.1069e+1
Mean	8.8967e-9	9.6317e-9	1.0806e+5	9.3788e-9	9.1535e-9	1.8909e+1	8.2610e-2	2.0991e+1
Std	9.3915e-10	3.2989e-10	8.7160e+4	6.3274e-10	6.3186e-10	3.9977e+1	6.2418e-2	5.7946e-2

Table 3: Error values achieved when FES=1e3, FES=1e4 and FES=1e5 for Problems 9-17 (D=10).

1e3	9	10	11	12	13	14	15	16	17
1^{st}	5.5183e+1	6.1738e+1	8.0730e+0	2.7548e+4	8.9262e+0	3.9475e+0	5.0269e+2	2.8631e+2	3.1075e+2
7^{th}	6.7238e+1	8.3257e+1	1.0631e+1	4.8239e+4	1.6386e+1	4.1324e+0	6.3492e+2	3.1663e+2	3.4365e+2
13^{th}	7.3823e+1	9.4737e+1	1.1205e+1	5.9316e+e4	1.9822e+1	4.2850e+0	6.9548e+2	3.2771e+2	3.5569e+2
19^{th}	8.3476e+1	1.0400e+2	1.1756e+1	6.8485e+4	3.5296e+1	4.4159e+0	7.2325e+2	3.5594e+2	3.8632e+2
25^{th}	9.0078e+1	1.1695e+2	1.2689e+1	9.3332e+4	9.4131e+1	4.4938e+0	7.6550e+2	4.0258e+2	4.3694e+2
Mean	7.4893e+1	9.2626e+1	1.1142e+1	5.8626e+4	2.8000e+1	4.2671e+0	6.7571e+2	3.3340e+2	3.6186e+2
Std	1.0273e+1	1.5112e+1	9.9503e-1	1.7707e+4	1.9463e+1	1.6581e-1	6.5686e+1	3.1336e+1	3.4011e+1
1e4	9	10	11	12	13	14	15	16	17
1^{st}	2.2931e+1	2.4935e+1	2.6192e+0	7.3944e+2	7.7266e-1	3.5647e+0	3.0709e+2	1.4953e+2	1.6229e+2
7^{th}	2.9486e+1	4.0065e+1	4.1185e+0	1.7779e+3	9.5098e-1	3.7598e+0	3.7041e+2	1.7970e+2	1.9504e+2
13^{th}	3.1443e+1	4.2601e+1	5.1635e+0	2.9669e+3	1.4826e+0	3.8791e+0	4.5714e+2	1.8783e+2	2.0386e+2
19^{th}	3.3659e+1	4.7088e+1	5.9658e+0	4.5042e+3	1.9638e+0	3.9125e+0	4.9371e+2	1.9494e+2	2.1158e+2
25^{th}	4.1104e+1	5.0810e+1	7.9697e+0	5.9559e+3	2.8250e+0	4.0230e+0	5.3406e+2	2.0522e+2	2.2274e+2
Mean	3.1784e+1	4.2311e+1	5.2614e+0	3.0946e+3	1.5366e+0	3.8267e+0	4.3616e+2	1.8537e+2	2.0119e+2
Std	4.6675e+0	5.7163e+0	1.4767e+0	1.5770e+3	6.2352e-1	1.2352e-1	6.9424e+1	1.4360e+1	1.5586e+1
1e5	9	10	11	12	13	14	15	16	17
1^{st}	9.9496e-1	1.9899e+0	2.9952e-4	3.9090e+0	3.4913e-1	1.3864e+0	6.3163e+1	9.1143e+1	9.8923e+1
7^{th}	2.9849e+0	3.9798e+0	1.1150e+0	1.4991e+01	7.0643e-1	2.9682e+0	1.0226e+2	1.0404e+2	1.1292e+2
13^{th}	3.9798e+0	5.9698e+0	1.6479e+0	3.0657e+1	8.1781e-1	3.0898e+0	2.0000e+2	1.1010e+2	1.1950e+2
19^{th}	4.9748e+0	8.9546e+0	2.7251e+0	2.1249e+2	1.0153e+0	3.2952e+0	4.0000e+2	1.1606e+2	1.2597e+2
25^{th}	1.1940e+1	2.7390e+1	4.4945e+0	1.5583e+3	1.3242e+0	3.6135e+0	4.2541e+2	1.3953e+2	1.5144e+2
Mean	4.0196e+0	7.3044e+0	1.9098e+0	2.5951e+2	8.3793e-1	3.0456e+0	2.5376e+2	1.0962e+2	1.1898e+2
Std	2.2703e+0	5.2116e+0	1.1598e+0	4.8933e+2	2.6913e-1	4.3662e-1	1.5052e+2	9.8654e+0	1.0707e+1

Table 4: Error values achieved when FES=1e3, FES=1e4 and FES=1e5 for Problems 18-25 (D=10).

1e3	18	19	20	21	22	23	24	25
1^{st}	1.0121e+3	1.0506e+3	1.0506e+3	7.9274e+2	8.5281e+2	1.2654e+3	8.0569e+2	1.3870e+3
7^{th}	1.0903e+3	1.1052e+3	1.1053e+3	1.2058e+3	1.0404e+3	1.3539e+3	1.2495e+3	1.4445e+3
13^{th}	1.1249e+3	1.1295e+3	1.1295e+3	1.2527e+3	1.0668e+3	1.3730e+3	1.2684e+3	1.4839e+3
19^{th}	1.1346e+3	1.1476e+3	1.1476e+3	1.2823e+3	1.1244e+3	1.3903e+3	1.2930e+3	1.5085e+3
25^{th}	1.1657e+3	1.1881e+3	1.2039e+3	1.2987e+3	1.2105e+3	1.4332e+3	1.3513e+3	1.6306e+3
Mean	1.1135e+3	1.1251e+3	1.1258e+3	1.2037e+3	1.0730e+3	1.3702e+3	1.2438e+3	1.4916e+3
Std	3.5894e+1	3.5262e+1	3.6601e+1	1.2500e+2	7.5926e+1	3.7187e+1	1.0921e+2	6.1016e+1
1e4	18	19	20	21	22	23	24	25
1^{st}	3.0287e+2	3.2456e+2	3.2456e+2	3.0000e+2	8.1188e+2	7.5292e+2	2.0073e+2	4.1869e+2
7^{th}	3.9592e+2	3.8348e+2	3.9210e+2	5.0000e+2	8.2329e+2	8.9039e+2	2.0227e+2	4.2549e+2
13^{th}	4.9648e+2	4.5003e+2	4.7334e+2	5.0000e+2	8.3351e+2	9.6404e+2	2.0397e+2	4.3371e+2
19^{th}	7.3084e+2	5.5944e+2	8.0101e+2	8.4193e+2	8.3936e+2	1.1068e+3	2.0555e+2	4.3767e+2
25^{th}	9.3002e+2	8.0452e+2	8.1660e+2	1.1748e+3	9.0540e+2	1.1911e+3	2.1709e+2	4.6134e+2
Mean	5.4184e+2	4.9618e+2	5.3642e+2	6.8016e+2	8.3499e+2	9.8441e+2	2.0455e+2	4.3388e+2
Std	1.9745e+2	1.5924e+2	1.8134e+2	2.6890e+2	2.0192e+1	1.2289e+2	3.4880e+0	1.0079e+1
1e5	18	19	20	21	22	23	24	25
1^{st}	3.0000e+2	3.0000e+2	3.0000e+2	3.0000e+2	3.0001e+2	5.5947e+2	2.0000e+2	4.0557e+2
7^{th}	3.0000e+2	3.0000e+2	3.0000e+2	5.0000e+2	7.6539e+2	5.5947e+2	2.0000e+2	4.0580e+2
13^{th}	3.0000e+2	3.0000e+2	3.0000e+2	5.0000e+2	7.6898e+2	5.5947e+2	2.0000e+2	4.0603e+2
19^{th}	5.7876e+2	3.0000e+2	8.0000e+2	8.4186e+2	7.7023e+2	5.5947e+2	2.0000e+2	4.0619e+2
25^{th}	9.0146e+2	8.0000e+2	8.0000e+2	1.1736e+3	7.7737e+2	9.7050e+2	2.0000e+2	4.0641e+2
Mean	4.3956e+2	3.8000e+2	4.4000e+2	6.8006e+2	7.4927e+2	5.7591e+2	2.0000e+2	4.0601e+2
Std	2.2494e+2	1.8708e+2	2.2913e+2	2.6873e+2	9.3700e+1	8.2207e+1	0.0000e+0	2.3848e-1

Table 5: Error values achieved when FES=1e3, FES=1e4 and FES=1e5 for Problems 1-8 (D=30).

1e3	1	2	3	4	5	6	7	8
1^{st}	1.2321e+4	3.9519e+4	2.6446e+8	4.6265e+4	2.7106e+4	2.9524e+9	6.2550e+3	2.1037e+1
7^{th}	1.8129e+4	5.0788e+4	3.5126e+8	5.9459e+4	3.3815e+4	7.0694e+9	7.4070e+3	2.1145e+1
13^{th}	1.9365e+4	5.9728e+4	4.1869e+8	6.9924e+4	3.5219e+4	8.4211e+9	7.6841e+3	2.1182e+1
19^{th}	2.1243e+4	7.4415e+4	4.9017e+8	8.7118e+4	3.6821e+4	1.3035e+10	8.2983e+3	2.1252e+1
25^{th}	3.0846e+4	9.5391e+4	7.2114e+8	1.1168e+5	4.1746e+4	2.5846e+10	9.3840e+3	2.1334e+1
Mean	1.9950e+4	6.2814e+4	4.3580e+8	7.3537e+4	3.4990e+4	9.9212e+9	7.7513e+3	2.1186e+1
Std	4.2139e+3	1.5693e+4	1.1527e+8	1.8372e+4	3.2974e+3	4.9599e+9	7.8100e+2	7.3620e-2
1e4	1	2	3	4	5	6	7	8
1^{st}	1.8254e+0	6.5707e+3	8.2420e+6	7.6925e+3	9.6113e+3	1.5865e+5	6.8264e+2	2.0908e+1
7^{th}	4.4636e+0	9.0131e+3	2.2593e+7	1.0552e+4	1.1661e+4	3.7611e+5	9.7361e+2	2.1060e+1
13^{th}	8.9665e+0	1.2589e+4	3.0410e+7	1.4738e+4	1.2899e+4	9.1982e+5	1.1835e+3	2.1093e+1
19^{th}	1.2707e+1	1.6074e+4	4.1480e+7	1.8818e+4	1.4772e+4	1.3160e+6	1.3107e+3	2.1124e+1
25^{th}	4.5770e+1	2.1604e+4	7.0255e+7	2.5292e+4	1.9917e+4	3.2307e+6	1.6241e+3	2.1149e+1
Mean	1.1109e+1	1.2889e+4	3.4025e+7	1.5089e+4	1.3641e+4	9.7136e+5	1.1640e+3	2.1081e+1
Std	9.9506e+0	4.2099e+3	1.5162e+7	4.9285e+3	2.8938e+3	6.9875e+5	2.4770e+2	6.1144e-2
1e5	1	2	3	4	5	6	7	8
1^{st}	8.5813e-9T	8.3688e-1	1.1156e+6	9.7974e-1	2.2095e+3	2.6576e+0	5.3781e-1	2.0843e+1
7^{th}	9.0664e-9T	2.3396e+0	2.2875e+6	2.7390e+0	3.4995e+3	1.9547e+1	8.9207e-1	2.0966e+1
13^{th}	9.6205e-9T	4.0097e+0	3.3781e+6	4.6942e+0	4.2513e+3	2.5913e+1	1.0301e+0	2.1011e+1
19^{th}	9.6205e-9T	7.2522e+0	4.2688e+6	8.4902e+0	5.0532e+3	7.3452e+1	1.0507e+0	2.1040e+1
25^{th}	9.9339e-9T	2.7212e+1	5.9713e+6	3.1857e+1	7.2539e+3	9.2739e+2	1.1204e+0	2.1077e+1
Mean	9.3524e-9	5.8753e+0	3.3175e+6	6.8783e+0	4.3366e+3	8.4980e+1	9.5537e-1	2.1000e+1
Std	4.6327e-10	5.7666e+0	1.4010e+6	6.7510e+0	1.3666e+3	1.9291e+2	1.4952e-1	6.0768e-2
3e5	1	2	3	4	5	6	7	8
1^{st}	8.5813e-9T	1.1045e-8	4.1124e+5	1.2931e-8	2.1916e+3	2.2823e-3	1.3911e-6	2.0837e+1
7^{th}	9.0664e-9T	7.1554e-8	7.6192e+5	8.2507e-8	3.2448e+3	4.1282e+0	7.4778e-3	2.0908e+1
13^{th}	9.6205e-9T	1.6077e-7	1.1241e+6	1.8822e-7	4.1725e+3	1.6889e+1	9.8946e-3	2.0925e+1
19^{th}	9.6205e-9T	4.7296e-7	1.4380e+6	5.5370e-7	4.9204e+3	1.9966e+1	1.4809e-2	2.0952e+1
25^{th}	9.9339e-9T	7.1024e-6	2.0430e+6	8.3149e-6	7.0869e+3	7.4378e+1	5.1686e-2	2.1030e+1
Mean	9.3524e-9	6.9482e-7	1.1020e+6	8.1320e-7	4.2374e+3	1.5197e+1	1.4598e-2	2.0932e+1
Std	4.6327e-10	1.4911e-6	4.2081e+5	1.7457e-6	1.3752e+3	1.4903e+1	1.2391e-2	4.5876e-2

Table 6: Error values achieved when FES=1e3, FES=1e4 and FES=1e5 for Problems 9-17 (D=30).

1e3	9	10	11	12	13	14	15	16	17
1^{st}	4.0959e+2	5.9174e+2	4.1466e+1	9.7568e+5	3.4718e+5	1.3979e+1	7.7041e+2	5.3557e+2	6.8908e+2
7^{th}	4.8025e+2	6.1442e+2	4.4290e+1	1.3438e+6	6.5253e+5	1.4115e+1	9.4001e+2	7.1847e+2	7.8705e+2
13^{th}	4.9067e+2	6.3015e+2	4.5277e+1	1.4387e+6	8.3781e+5	1.4191e+1	1.0158e+3	7.6192e+2	8.6097e+2
19^{th}	5.1228e+2	6.6555e+2	4.6351e+1	1.6173e+6	1.1916e+6	1.4301e+1	1.0444e+3	8.3701e+2	9.1227e+2
25^{th}	5.7472e+2	7.4853e+2	4.7265e+1	1.7981e+6	1.3577e+6	1.4392e+1	1.0951e+3	9.6594e+2	1.0683e+3
Mean	4.9469e+2	6.4635e+2	4.5209e+1	1.4567e+6	8.7739e+5	1.4200e+1	9.9871e+2	7.6863e+2	8.6124e+2
Std	3.3390e+1	4.1053e+1	1.4137e+0	2.0081e+5	3.2676e+5	1.1488e-1	7.3895e+1	9.9673e+1	9.0985e+1
1e4	9	10	11	12	13	14	15	16	17
1^{st}	2.9169e+2	2.3374e+2	3.8995e+1	3.5344e+5	2.1061e+2	1.3483e+1	4.9671e+2	2.4651e+2	2.6755e+2
7^{th}	3.1765e+2	2.5527e+2	4.1271e+1	4.5059e+5	9.9414e+2	1.3723e+1	4.9861e+2	2.7859e+2	3.0237e+2
13^{th}	3.3510e+2	2.6445e+2	4.2421e+1	4.8573e+5	1.9140e+3	1.3831e+1	5.0651e+2	2.9349e+2	3.1439e+2
19^{th}	3.4688e+2	2.7433e+2	4.3007e+1	5.2876e+5	3.0058e+3	1.3886e+1	5.2299e+2	3.2983e+2	3.3300e+2
25^{th}	3.7683e+2	3.0806e+2	4.4345e+1	6.3387e+5	5.9685e+3	1.4020e+1	6.4874e+2	3.8484e+2	3.8675e+2
Mean	3.3363e+2	2.6621e+2	4.2021e+1	4.8183e+5	2.1727e+3	1.3806e+1	5.2658e+2	3.0138e+2	3.2399e+2
Std	2.1535e+1	1.7864e+1	1.4841e+0	6.3260e+4	1.5450e+3	1.4161e-1	4.4745e+1	3.6060e+1	3.3869e+1
1e5	9	10	11	12	13	14	15	16	17
1^{st}	7.4473e+1	3.7038e+1	7.8829e+0	4.8504e+3	2.2540e+0	1.3184e+1	2.0022e+2	5.0957e+1	4.8301e+1
7^{th}	1.2795e+2	1.6910e+2	1.5555e+1	1.3474e+4	2.8454e+0	1.3354e+1	4.0000e+2	5.9653e+1	7.6854e+1
13^{th}	1.4708e+2	1.8809e+2	1.6890e+1	1.7666e+4	3.5583e+0	1.3452e+1	4.0000e+2	7.9569e+1	9.8080e+1
19^{th}	1.7220e+2	2.0321e+2	2.1633e+1	2.9530e+4	4.5371e+0	1.3530e+1	4.0000e+2	1.4716e+2	1.5693e+2
25^{th}	1.8982e+2	2.1039e+2	3.9599e+1	7.0803e+4	6.4643e+0	1.3752e+1	5.0000e+2	2.9838e+2	3.2384e+2
Mean	1.4338e+2	1.7061e+2	1.8380e+1	2.3408e+4	3.7176e+0	1.3452e+1	3.6829e+2	1.1432e+2	1.2414e+2
Std	3.4255e+1	4.7067e+1	6.2538e+0	1.5971e+4	1.0849e+0	1.4500e-1	9.4486e+1	7.1239e+1	7.3865e+1
3e5	9	10	11	12	13	14	15	16	17
1^{st}	1.0945e+1	1.5919e+1	5.0042e+0	1.4259e+3	1.8774e+0	1.2270e+1	2.0000e+2	4.0800e+1	4.4283e+1
7^{th}	1.9899e+1	3.5819e+1	9.4190e+0	6.1867e+3	2.6942e+0	1.3093e+1	4.0000e+2	5.1773e+1	5.8038e+1
13^{th}	2.3879e+1	5.1738e+1	1.1169e+1	9.9526e+3	3.4020e+0	1.3152e+1	4.0000e+2	5.8747e+1	7.1284e+1
19^{th}	2.6864e+1	6.1687e+1	1.2643e+1	1.6045e+4	4.3430e+0	1.3265e+1	4.0000e+2	7.8984e+1	9.6722e+1
25^{th}	3.6813e+1	1.8395e+2	2.0265e+1	6.8650e+4	6.4165e+0	1.3481e+1	5.0000e+2	1.5971e+2	1.5997e+2
Mean	2.3934e+1	6.0297e+1	1.1255e+1	1.3134e+4	3.5881e+0	1.3131e+1	3.6822e+2	7.4683e+1	8.5361e+1
Std	6.2477e+0	4.0576e+1	3.2979e+0	1.3346e+4	1.0857e+0	2.6887e-1	9.4598e+1	3.6143e+1	3.9099e+1

Table 7: Error values achieved when FES=1e3, FES=1e4 and FES=1e5 for Problems 18-25 (D=30).

1e3	18	19	20	21	22	23	24	25
1^{st}	1.2081e+3	1.2287e+3	1.2073e+3	1.3417e+3	1.3829e+3	1.3083e+3	1.3637e+3	1.4497e+3
7^{th}	1.2643e+3	1.2661e+3	1.2598e+3	1.3946e+3	1.4536e+3	1.3832e+3	1.4049e+3	1.5325e+3
13^{th}	1.2739e+3	1.2895e+3	1.2735e+3	1.4045e+3	1.5102e+3	1.4109e+3	1.4239e+3	1.6723e+3
19^{th}	1.2962e+3	1.3002e+3	1.2956e+3	1.4292e+3	1.6155e+3	1.4225e+3	1.4440e+3	1.7986e+3
25^{th}	1.3541e+3	1.3536e+3	1.3536e+3	1.4894e+3	1.7326e+3	1.4844e+3	1.4670e+3	1.8256e+3
Mean	1.2779e+3	1.2834e+3	1.2764e+3	1.4126e+3	1.5328e+3	1.4062e+3	1.4229e+3	1.6334e+3
Std	3.2191e+1	2.7784e+1	3.2297e+1	3.4285e+1	9.6156e+1	3.8029e+1	2.7450e+1	1.2968e+2
1e4	18	19	20	21	22	23	24	25
1^{st}	9.7521e+2	9.7843e+2	9.7843e+2	1.0551e+3	1.0646e+3	1.0533e+3	6.3623e+2	7.9317e+2
7^{th}	9.9441e+2	1.0023e+3	1.0023e+3	1.1168e+3	1.1193e+3	1.1293e+3	7.1032e+2	1.0695e+3
13^{th}	1.0024e+3	1.0099e+3	1.0086e+3	1.1436e+3	1.1414e+3	1.1438e+3	7.6989e+2	1.2389e+3
19^{th}	1.0111e+3	1.0171e+3	1.0177e+3	1.1563e+3	1.1536e+3	1.1652e+3	8.2148e+2	1.3807e+3
25^{th}	1.0376e+3	1.0380e+3	1.0369e+3	1.2018e+3	1.1935e+3	1.1988e+3	9.2960e+2	1.4109e+3
Mean	1.0034e+3	1.0092e+3	1.0087e+3	1.1341e+3	1.1349e+3	1.1435e+3	7.7696e+2	1.1798e+3
Std	1.5950e+1	1.4747e+1	1.4072e+1	3.6469e+1	2.7479e+1	3.4655e+1	8.2622e+1	1.7585e+2
1e5	18	19	20	21	22	23	24	25
1^{st}	9.0582e+2	9.0540e+2	9.0540e+2	5.0002e+2	8.7640e+2	5.3416e+2	2.0000e+2	2.2183e+2
7^{th}	9.0656e+2	9.0637e+2	9.0637e+2	5.0003e+2	8.9802e+2	5.3416e+2	2.0000e+2	2.2439e+2
13^{th}	9.0716e+2	9.0683e+2	9.0683e+2	5.0005e+2	9.0725e+2	5.3416e+2	2.0000e+2	2.2603e+2
19^{th}	9.0901e+2	9.0855e+2	9.0853e+2	5.0006e+2	9.1293e+2	5.3417e+2	2.0000e+2	2.3081e+2
25^{th}	9.1178e+2	9.0947e+2	9.0947e+2	5.0008e+2	9.3174e+2	5.3417e+2	2.0000e+2	2.3225e+2
Mean	9.0765e+2	9.0726e+2	9.0727e+2	5.0005e+2	9.0567e+2	5.3417e+2	2.0000e+2	2.2634e+2
Std	1.4784e+0	1.1702e+0	1.1669e+0	1.8460e-2	1.2731e+1	3.6134e-4	0.0000e+0	2.9873e+0
3e5	18	19	20	21	22	23	24	25
1^{st}	9.0342e+2	9.0367e+2	9.0367e+2	5.0000e+2	8.6026e+2	5.3416e+2	2.0000e+2	2.1206e+2
7^{th}	9.0393e+2	9.0406e+2	9.0403e+2	5.0000e+2	8.6838e+2	5.3416e+2	2.0000e+2	2.1305e+2
13^{th}	9.0450e+2	9.0430e+2	9.0421e+2	5.0000e+2	8.7263e+2	5.3416e+2	2.0000e+2	2.1348e+2
19^{th}	9.0634e+2	9.0603e+2	9.0503e+2	5.0000e+2	8.9865e+2	5.3416e+2	2.0000e+2	2.1379e+2
25^{th}	9.0904e+2	9.0691e+2	9.0714e+2	5.0000e+2	9.0826e+2	5.3417e+2	2.0000e+2	2.1413e+2
Mean	9.0504e+2	9.0482e+2	9.0477e+2	5.0000e+2	8.8125e+2	5.3416e+2	2.0000e+2	2.1321e+2
Std	1.4284e+0	1.0871e+0	1.0987e+0	0.0000e+0	1.6213e+1	3.4895e-4	0.0000e+0	6.2752e-1

503

Table 8: Number of FES to achieve the required accuracy (D=10). SR and SP stand for Success Rate and Success Performance, respectively.

Problem	1^{st} (Best)	7^{th}	13^{th} (Median)	19^{th}	25^{th} (Worst)	Mean	Std	SR (%)	SP
1	5,103	5,833	6,413	7,581	9,611	6.7252e+3	1.2411e+3	100	6.7252e+3
2	22,832	26,914	30,881	33,306	39,111	3.1012e+4	4.4070e+3	100	3.1012e+4
3	-	-	-	-	-	-	-	0	-
4	24,935	27,185	30,045	33,702	38,140	3.0714e+4	3.9888e+3	100	3.0714e+4
5	31,786	36,143	39,477	45,930	50,137	4.0259e+4	5.8031e+3	100	4.0259e+4
6	-	-	-	-	-	-	-	0	-
7	72,134	-	-	-	-	7.2134e+4	0	4	1.8033e+6
8-10	-	-	-	-	-	-	-	0	-
11	43,773	-	-	-	-	4.3773e+4	0	4	1.0943e+6
12-25	-	-	-	-	-	-	-	0	-

Table 9: Number of FES to achieve the required accuracy (D=30). SR and SP stand for Success Rate and Success Performance, respectively.

Problem	1^{st} (Best)	7^{th}	13^{th} (Median)	19^{th}	25^{th} (Worst)	Mean	Std	SR (%)	SP
1	27,406	29,128	30,235	30,758	40,417	3.0326e+4	2.4500e+3	100	3.0326e+4
2	251,785	268,550	279,721	295,207	300,000	2.8021e+5	1.4855e+4	88	3.1536e+5
3	-	-	-	-	-	-	-	0	-
4	253,291	269,795	280,851	296,291	300,000	2.8186e+5	1.4336e+4	76	3.6334e+5
5	-	-	-	-	-	-	-	0	-
6	208,212	-	-	-	-	2.0821e+5	0	4	5.2053e+6
7	159,537	236,081	273,612	300,000	300,000	2.5981e+5	4.4696e+4	64	3.7063e+5
8-25	-	-	-	-	-	-	-	0	-

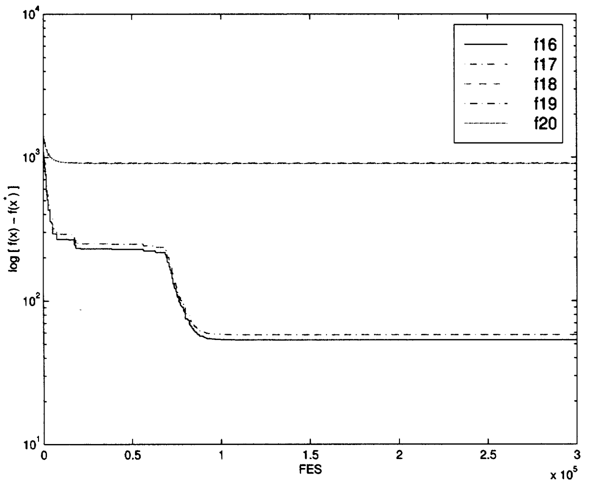

Figure 5: Convergence graphs for problems 16 to 20 in D=30.

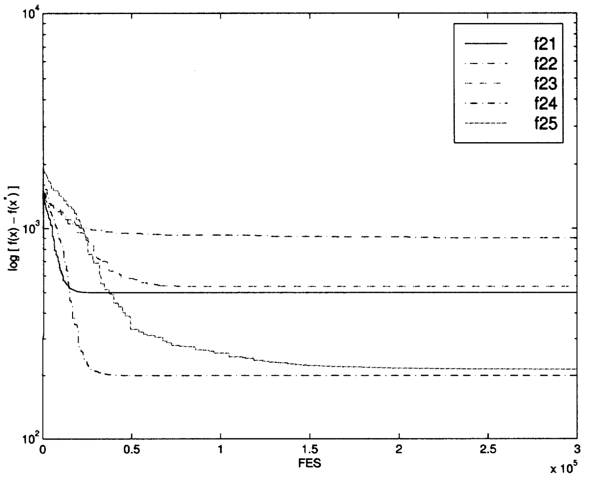

Figure 6: Convergence graphs for problems 21 to 25 in D=30.

to have significant[3] effect on the performance since the results from problem 2 and 16 are similar to those in problem 4 and 17, respectively. Thirdly, it seems that the rotated problems posed a greater difficulty to SPC-PNX. This is suggested by the better performance on problems 9 and 15 over problems 10 and 16, respectively, although this improvement is not significant. Also, the algorithm performed worse on problem 23 than on problem 21, although the difference was not significant and thus it cannot concluded that the non-continuous problem 23 introduces more difficulty. With regard to the global optimum being inside or on the initialisation bounds, no change of performance is perceived (see the problem 18 and 20 pair). Likewise, the change of performance is not significant when comparing problem 18 and 19 (this is consistent with the fact that the global minimum was not identified in either of the cases). Another factor that does not appear to have an influence on the SPC-PNX's performance is how the rotation is made (i.e. with an orthogonal matrix or with a moderately high condition number matrix) as shown by comparing problem 21 and 22. Lastly, having the global optimum outside the initialisation range seems to have a detrimental effect on the performance as shown by comparing problem 24 and 25.

Regarding the algorithm's complexity, all participants were asked to run a test to estimate it. The details of this calculation are in the evaluation criteria document (see Section 1). The results of this test are presented in Table 10.

Finally, we have observed that SPC-PNX's performance is generally better with large population sizes. However, as N increases, a higher number of FES is needed to make the population converge to a good solution. Therefore, it is critical to allow a sufficiently high number of FES when op-

[3]We say that the results in two problems differ significantly if the corresponding intervals, formed by double the standard deviation centered at the mean function value, do not overlap.

Table 10: Estimation of the algorithm's complexity.

D	T_0	T_1	$\hat{T}_2$	$(\hat{T}_2\text{-}T_1)/T_0$
10	0.610	26.797	136.049	179.102
30	0.407	32.218	135.553	253.894
50	0.422	38.953	148.853	260.427

erating SPC-PNX. Otherwise, the potential for good SPC-PNX's performance on some problems is wasted. Table 11 presents some results of running SPC-PNX with a higher number of FES on several problems. These runs are compared to the best function values found when using the restricted number of FES tentatively proposed by the organisers. These results make evident the need of allowing a higher number of FES on some test problems.

Table 11: Results of the numerical experiments made with a higher number of FES for different problems. These results evidence that a restrictive number of FES hides SPC-PNX's full potential for tackling real-parameter optimisation problems.

Problem	M	Max_FES	N	fbest
6	10	100,000	60	1.8594e-2
6	10	500,000	60	5.8152e-6
9	10	100,000	225	9.9496e-1
9	10	500,000	1,200	8.5663e-9
10	10	100,000	200	1.9899e+0
10	10	1,000,000	800	9.6457e-9
14	10	100,000	100	1.3864e+0
14	10	500,000	100	4.4034e-1

5 Conclusions

This paper has presented a performance study of a Real-parameter GA (SPC-PNX) on a new benchmark of real-parameter optimisation problems. These test problems were designed to avoid symmetries that could be exploited by optimisation methods. Consequently, this benchmark is expected to challenge many good optimisation methods.

As we have still no access to the results obtained by other participants, it is difficult to assess how well SPC-PNX performed on the new benchmark. However, it is our belief that the required accuracy levels to declare success are quite optimistic in most of the problems, specially those involving hybrid functions and most 30-variable multimodal functions.

Some insights into how the problem features affect SPC-PNX's performance were obtained by comparing selected subsets of problems. It was determined that a high condition number strongly deteriorates the algorithm's performance. Other problem features seem to pose greater difficulty such as rotated function landscapes, discontinuities and having the global optimum outside initialisation region. In the cases of having the global optimum inside or on the initialisation bounds as well as having the global optimum outside the initialisation region, few conclusions can be reached as the global optimum was not found. Lastly, problem features such as noisy function landscapes and how the rotation is made do not seem to have significant effect on SPC-PNX's performance.

On the other hand, SPC-PNX's benchmark results could be improved further in the following two ways. The first way is to use large population sizes. We have presented results suggesting that larger population sizes generally lead to better results, at a cost of requiring a higher number of function evaluations to converge. The second way is to test a wider range of population sizes. This is specially advisable on the 30-variable hybrid functions, since only two different population sizes could be tested in these cases due to the intensive computation required.

Besides, it is worth noting that SPC-PNX is a pure Real-parameter GA and therefore it has a large margin for improvement. Some promising future research directions include: to make a hybrid algorithm combining SPC-PNX with a faster local optimiser and to devise a restart strategy intended to inject diversity into the population when needed.

Finally, we want to propose two improvements to the benchmark. First and most important, it is essential to permit a higher number of function evaluations so that all optimisation methods (not only the most efficient) can show their full potential. Second, the requested numerical experiments are many and very computer-intensive (specially in the case of the hybrid functions). We propose to focus future studies on a selected set of about 15 problems.

Bibliography

[1] Ballester, P.J., Carter, J.N.: An effective real-parameter genetic algorithm with parent centric normal crossover for multimodal optimization. In Deb, K., et al., eds.: Lecture Notes in Computer Science 3102, Springer (2004) 901 913

[2] Deb, K.: Genetic algorithms for optimization. Technical report, Indian Institute of Technology Kanpur (2001) KanGAL report No. 2001002.

[3] Herrera, F., Lozano, M., Verdegay, J.L.: Tackling real-coded genetic algorithms: Operators and tools for the behaviour analysis. Artificial Intelligence Review **12** (1998) 265 319

[4] Ballester, P.J.: New Computational Methods to Address Nonlinear Inverse Problems. PhD thesis, Department of Earth Science and Engineering, Imperial College, University of London (2005)

[5] Mengshoel, O.J., Goldberg, D.E.: Probabilistic crowding: Deterministic crowding with probabilistic replacement. In Banzhaf, W., et al., eds.: Proc. of the Genetic and Evolutionary Computation Conference, Morgan Kaufmann (1999) 409 416

[6] Ballester, P.J., Carter, J.N.: Real-parameter genetic algorithms for finding multiple optimal solutions in multi-modal optimization. In Cantú-Paz, E., et al., eds.: Lecture Notes in Computer Science 2723, Springer (2003) 706 717