3. Experimental results

A comprehensive set of benchmark functions [18, 33, 34, 35, 36, 37, 38] has been used to test the performance of the proposed algorithm. The Appendix A (Table A1) presents the functions used in our experimental study. Such functions belong to two categories: unimodal as well as multimodal. We compared the performance of our algorithm against other well-known algorithms, like Differential Evolution (DE) [18], Particle Swarm Optimization (PSO) [6] and Artificial Bee Colony Optimization (ABC) [7]. A modified version of DE [39] was used in the comparisons. That version improves the robustness and convergence of the original version. The algorithms DE and ABC algorithms were configured as suggested in [7, 39], whereas PSO was set according to the suggestions provided in [35]. Table 3.1 shows the specific parameters used in the experiments, where the notation for population size is the same (Ns) for each algorithm. The stopping criterion used in the experimental part was to evaluate 100,000 times every function from Table A1. Each algorithm was programmed in Matlab R2013b, and ran 30 times per function in a PC with an architecture based on Intel Core i7-2600k with 8 GB of Ram.

DE [39]			**PSO [35]**				**ABC [7]**	**AO**	
Ns	F	Cr	Ns	ϕ_1	ϕ_2	ω	Ns	Ns	ψ
50	0.9	0.5	50	1.8	1.8	k/maxIter	50	5	1e-20

Table 3.1 Parameters tuning of each algorithm.

Several values of size (population, memory) and ψ were tested with AO, as shown in Table 3.2. Some parameter arrangements can produce better values for some groups of functions than others, as it was established by the No Free Lunch theorem [Wolpert997]; in fact, this is the main reason to consider the development of new proposals of metaheuristic algorithms.

config	Ns	ψ	g_1	g_2	g_3
c_1	5	$1e-20$	3	1	1
c_2	10	$1e-20$	3	1	1
c_3	50	$1e-20$	3	1	1
c_4	5	$1e-1$	3	1	1
c_5	5	$1e-20$	4	1	0
c_6	5	$1e-20$	0	4	1
c_7	5	$1e-20$	1	0	4
c_8	5	$1e-20$	8	4	1

Table 3.2 Some configurations tested with AO ($d = 100$, mofe $= 100{,}000$).

As explained before, by every element into the population exist g1+g2+g3 elements into memory (M1,M2,

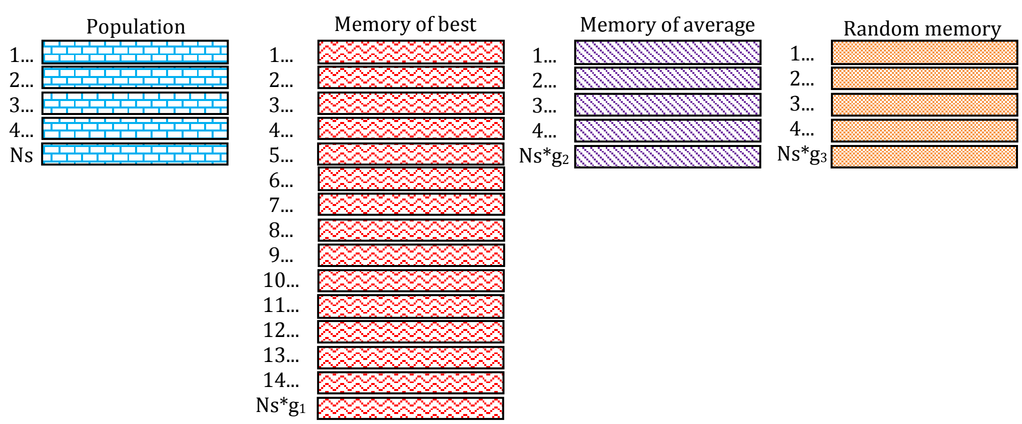

Figure 3.1 Arrangement of population and memory, considering c_1 and c_4.

and M3): M1 has information related with the best individual obtained from each iteration, M2 retrieves information from the average individual, and M3 stores random movements inside the feasible space. As an example, Figure 3.1 depicts the arrangement of population and memory that belongs to c_1. All configurations in Table 3.2 were used with AO to minimize every function from Table A1; the best results after 30 runs (highlighted) are presented in Table 3.3. For instance, in the case of c_5, where the random part of memory (M3) is not considered, the algorithm is capable of finding the best results of f_{12}, f_{16} and f_{19}, which are unimodal, multimodal, and have different limits of the search space. A similar case happens with configuration c_6 where the part that stores historical information of the best individual found at each iteration is removed from memory (M1), and where the data of the average individual (M2) is increased. In this way, the best results are obtained for the functions f_8 and f_{20}, both with unimodal form, and for f_8, with a non symmetric search space.

f	$c_1, \mu\ (\sigma^2)$	$c_2, \mu\ (\sigma^2)$	$c_3, \mu\ (\sigma^2)$	$c_4, \mu\ (\sigma^2)$	$c_5, \mu\ (\sigma^2)$	$c_6, \mu\ (\sigma^2)$	$c_7, \mu\ (\sigma^2)$	$c_8, \mu\ (\sigma^2)$
f_1	0.3038 (0.6620)	**0.0348 (0.0730)**	12.76 (11.32)	2.95 (3.83)	0.8024 (1.30)	310.49 (178.51)	308.26 (59.57)	0.2522 (0.5765)
f_2	**5.24e-05 (2.30e-05)**	0.0016 (0.0007)	0.6488 (0.1354)	0.0705 (0.0171)	0.0014 (0.0006)	1.44 (0.2600)	0.1186 (0.0349)	0.0183 (0.0074)
f_3	5.8509 (5.64)	**4.4429 (1.46)**	167.30 (21.15)	23.69 (8.39)	8.37 (12.68)	130.17 (43.15)	23.18 (10.78)	12.52 (3.98)
f_4	0.0018 (0.0058)	**9.90e-06 (9.81e-06)**	0.0623 (0.0146)	0.0043 (0.0077)	0.0114 (0.0362)	1.34 (0.2067)	0.4776 (0.1337)	0.0020 (0.0027)
f_5	7.15e-07 (2.66e-06)	1.19e-09 (4.46e-09)	4.28e-08 (6.67e-08)	7.94e-06 (3.99e-06)	4.47e-07 (1.41e-06)	0.1613 (0.0421)	0.0198 (0.0026)	**2.28e-26 (7.15e-26)**
f_6	**6.34e-08 (8.98e-08)**	3.78e-05 (3.40e-05)	0.7779 (0.1618)	0.0394 (0.0236)	3.23e-05 (1.82e-05)	44.44 (27.52)	0.9944 (0.4484)	0.0047 (0.0037)
f_7	-41842.40 (13.47)	-41833.12 (15.96)	-39812.09 (258.88)	-41845.09 (20.44)	-40370.13 (256.39)	-39643.94 (348.71)	**-41881.53 (4.07)**	-41604.83 (136.07)
f_8	890.25 (113.11)	908.02 (100.56)	827.64 (59.99)	1018.94 (71.90)	802.09 (87.16)	**411.70 (175.53)**	1082.30 (60.57)	783.76 (75.72)
f_9	**3.19e-07 (2.10e-07)**	0.0002 (0.0002)	4.04 (1.01)	0.9255 (0.4971)	0.0011 (0.0016)	33.33 (7.76)	0.3794 (0.1129)	0.0675 (0.0441)
f_{10}	**0.0158 (0.0514)**	0.0383 (0.1108)	5.55 (1.24)	9.10 (2.54)	9.51 (2.16)	23.78 (3.98)	0.8006 (0.1831)	3.32 (1.85)
f_{11}	**0.9398 (0.9120)**	3.9402 (3.53)	43.32 (10.88)	2.18 (1.61)	4.17 (6.34)	10.20 (6.84)	2.14 (1.01)	2.44 (1.14)
f_{12}	0.1966 (0.0399)	0.2610 (0.0278)	0.7430 (0.0938)	0.1966 (0.0279)	**0.1931 (0.0444)**	0.9182 (0.2393)	0.2619 (0.0289)	0.2130 (0.0647)
f_{13}	1.1388 (0.3962)	**0.5395 (0.2805)**	2.00 (0.4779)	1.01 (0.4193)	1.94 (0.5758)	41.46 (23.54)	2.28 (0.6041)	1.58 (0.4084)
f_{14}	**0.0002 (8.54e-05)**	0.0069 (0.0024)	1.64 (0.1859)	1.05 (0.1569)	0.0132 (0.0023)	5.10 (0.7199)	0.5418 (0.1163)	0.1204 (0.0288)
f_{15}	**2.29e-06 (1.92e-06)**	0.0012 (0.0009)	34.46 (8.29)	2.02 (1.41)	0.0013 (0.0010)	1832.44 (1053.57)	50.67 (19.12)	0.3889 (0.1544)
f_{16}	14.07 (1.14)	28.02 (2.39)	67.78 (2.44)	13.64 (1.02)	**10.71 (0.7580)**	18.24 (2.66)	29.58 (1.79)	13.10 (1.62)
f_{17}	0.0032 (0.0048)	0.0040 (0.0073)	0.0084 (0.0022)	**0.0023 (0.0016)**	0.0130 (0.0101)	0.0584 (0.0470)	0.0050 (0.0051)	0.0051 (0.0028)
f_{18}	8.33e-10 (2.35e-09)	6.04e-12 (3.12e-11)	2.01e-09 (5.46e-09)	3.76e-06 (2.61e-06)	1.16e-15 (3.68e-15)	1.21 (1.11)	0.1127 (0.0269)	**3.30e-25 (1.03e-24)**
f_{19}	3.27 (0.498)	3.29 (0.3341)	6.32 (0.3831)	3.25 (0.3238)	**3.00 (0.3541)**	5.68 (1.15)	4.21 (0.2898)	3.26 (0.2790)
f_{20}	125661.19 (17326.6)	146656.89 (15114.84)	180936.07 (15873.87)	121219.84 (16193.88)	152393.77 (26302.53)	**15039.99 (9574.81)**	128216.9 (10023.1)	139308.8 (21686.97)
f_{21}	**0.0047 (0.0019)**	0.0050 (0.0018)	0.1108 (0.0226)	0.0449 (0.0092)	0.0183 (0.0022)	0.4746 (0.1273)	0.0358 (0.0045)	0.0185 (0.0038)
f_{22}	**6.06e-27 (1.17e-26)**	3.09e-15 (2.48e-15)	0.0250 (0.0079)	0.1120 (0.0444)	1.04e-25 (8.40e-26)	7.57 (5.42)	0.2800 (0.0943)	3.30e-19 (4.48e-19)

Table 3.3 Results obtained by AO, considering different parameter configurations.

Particularly, we argue that the average individual is capable to provide complementary information to contained into the best one, and therefore provokes a moderate search, by considering clues from every entity in the population. As mentioned earlier, in the outcome group corresponding to c_6, AO is capable to find the best results for functions f_8 and f_{20}, compared with the rest of the configurations. Considering the remaining functions, c_6 does not minimize the functions as well as the other seven configurations. In general, the worst configuration was c_3. However, such parameter composition improves some of the minimizations (*e.g.*, f_5 with c_3 is better than f_5 with c_1).

f	AO	ABC	DE	PSO
f_1	**0.3038(0.6620)**	43.53(30.39)	59.77(12.63)	107724.17(58676.77)
f_2	**5.24e-05(2.30e-05)**	0.0144(0.0055)	0.4234(0.1084)	11.19(2.69)
f_3	**5.8509(5.64)**	14.94(8.12)	23.21(3.84)	355262.35(471828.77)
f_4	0.0018(0.0058)	**0.0010(0.0016)**	0.2948(0.0817)	28.17(56.73)
f_5	7.15e-07(2.66e-06)	**5.81e-07(7.33e-07)**	1.47e-05(3.39e-06)	67.26(12.95)
f_6	**6.34e-08(8.98e-08)**	5.27e-06(7.80e-06)	0.3601(0.0459)	5333.35(6399.39)
f_7	**-41842.40(13.47)**	-38270.86(354.96)	-33793.36(744.70)	-26554.18(1579.68)
f_8	890.25(113.11)	1345.91(43.31)	**245.53(45.32)**	2456.11(527.65)
f_9	**3.19e-07(2.10e-07)**	1.90e-06(1.35e-06)	0.1470(0.0313)	5127.99(2828.48)
f_{10}	**0.0158(0.0514)**	16.57(3.72)	13.67(2.85)	474.55(61.68)
f_{11}	0.9398(0.9120)	**0.7217(0.51)**	79.46(26.14)	2173.31(1829.98)
f_{12}	**0.1966(0.0399)**	2.00(0.3192)	1.11(0.1140)	20.21(36.97)
f_{13}	1.1388(0.3962)	**4.20e-06(6.18e-06)**	29.48(0.3172)	5005.31(8583.11)
f_{14}	**0.0002(8.54e-05)**	0.0022(0.0007)	0.3760(0.0321)	41.24(19.10)
f_{15}	**2.29e-06(1.92e-06)**	5.04e-05(0.0001)	16.49(3.24)	492500.38(357311.67)
f_{16}	**14.07(1.14)**	89.34(1.48)	82.81(2.34)	48.96(3.87)
f_{17}	0.0032(0.0048)	**4.75e-07(6.92e-07)**	3.36(0.1198)	2.71(1.20)
f_{18}	**8.33e-10(2.35e-09)**	4.97e-06(1.30e-05)	0.0004(0.0001)	51.98(16.09)
f_{19}	**3.27(0.498)**	11.57(1.13)	14.74(0.5876)	6.01(4.88)
f_{20}	125661.19(17326.6)	149451.43(12623.75)	206365.91(34650.84)	**78313.59(20102.76)**
f_{21}	**0.0047(0.0019)**	0.1799(0.0939)	0.6697(0.1645)	17.34(9.39)
f_{22}	**6.06e-27(1.17e-26)**	3.74e-06(7.33e-06)	0.0803(0.0164)	3150.21(2140.63)

Table 3.4 Minimization results. Values correspond to mean and standard deviation of the best so-far values found after 30 runs.

Notwithstanding, we considered c_1 as our best parameter configuration, because AO with this arrangement is able to find the minimum value in almost the 41% of the functions, compared with c_2 and c_5 (almost 14% each one), c_4 and c_7 (about 5% each other), and c_6 and c_8 (roughly 9% everyone); therefore, we employed c_1 as configuration in all the experiments. Table 3.4 shows the outcomes obtained by each algorithm after 30 independent runs, and the best ones are highlighted in bold. The results are in the format mean, and standard deviation. According with the Table, AO gives the best results in the 68% of the functions, and from those, 57% are unimodal, whereas 43% are multimodal. An important matter is that even though the values contained in Table 3.4 were calculated taking for granted a normal distribution, it is in fact necessary to do some test to corroborate such assumption; withal, a simpler approach is adopting an analysis which not considers a particular distribution of the experimental data. A statistical test of this kind is the Non-Parametric Wilcoxon signed rank test [Wilcoxon945], in which it is applied the ranking of the data prior the analysis. In the Wilcoxon analysis it is considered a significance level of the 5%, where the null hypothesis states that there is no a significant difference of median values obtained by two processes, whereas the alternative hypothesis establishes that there is a significant difference among data, and this variation it is not due to the randomness inherent to the experiments. In our proposal it was applied the Wilcoxon test over the best values that were found at 30 independent runs per function, in order to statistically identify those in which AO gives a better performance than ABC, DE and PSO. The aforementioned results are given in Table 3.5, and they are analyzed below. By considering the case AO vs. ABC, our proposal shows a better performance in the 86% of the functions with a 5% confidence interval, except for f_{11}, f_{13} and f_{17}, in whose situation ABC is considerably better than AO. With respect to DE, our algorithm achieves good responses with the 91% of the functions, albeit in f_7 and f_8, DE is better. Furthermore, PSO compared with AO gives a superior performance in approximately 14% of the functions (f_{13}, f_{19}, and f_{20}); however, AO finds better results in the outstanding functions.

f	AO vs. ABC	AO vs. DE	AO vs. PSO
f_1	**0.0001**	**2.05e-05**	**7.77e-05**
f_2	**4.07e-05**	**9.13e-05**	**2.11e-05**
f_3	**0.0096**	**0.0001**	**9.08e-05**
f_4	**0.0032**	**1.82e-05**	**0.0013**
f_5	**6.23e-05**	**1.03e-05**	**1.29e-05**
f_6	**1.69e-05**	**3.73e-05**	**8.24e-05**
f_7	**7.72e-05**	0.9999	**4.65e-05**
f_8	**3.52e-05**	0.9999	**1.69e-05**
f_9	**5.08e-05**	**5.26e-05**	**7.45e-05**
f_{10}	**7.39e-05**	**1.61e-05**	**4.05e-05**
f_{11}	0.5484	**7.19e-05**	**1.28e-05**
f_{12}	**1.56e-05**	**3.39e-05**	**1.29e-05**
f_{13}	0.9999	**3.02e-05**	0.9794
f_{14}	**3.31e-05**	**7.19e-05**	**3.45e-05**
f_{15}	**2.22e-05**	**3.11e-05**	**5.15e-05**
f_{16}	**2.07e-06**	**1.53e-05**	**3.26e-05**
f_{17}	0.9999	**7.01e-05**	**1.12e-05**
f_{18}	**1.51e-05**	**3.27e-05**	**1.38e-06**
f_{19}	**4.02e-04**	**5.67e-04**	0.7870
f_{20}	**3.05e-06**	**7.23e-05**	0.9999
f_{21}	**1.50e-05**	**4.25e-05**	**8.38e-05**
f_{22}	**7.00e-05**	**1.16e-04**	**3.04e-05**

Table 3.5 *p*-values produced by Wilcoxon test; comparisons were made between AO vs. ABC, DE and PSO, over the best so-far values found in 30 independent runs.

As it can be seen, AO finds better optimization results compared against ABC, DE and PSO, from a statistical viewpoint according to the Wilcoxon test. In the last experimental part were completed 30 runs of each algorithm per function, where the time to complete 100,000 evaluations was measured (Table 3.6). According to the p-values obtained after doing the Wilcoxon test, the time required by AO is significantly lower against ABC, considering the 100% of the objective functions, whereas the time used by AO compared with DE is lower in the 91% of the cases.

f	AO	ABC	DE	PSO
f_1	4.69(0.6012)	5.62(0.3711)	5.46(0.5543)	**3.39(0.5795)**
f_2	4.99(0.6417)	5.90(0.7675)	5.62(0.5901)	**3.72(0.4012)**
f_3	4.10(0.5793)	5.15(0.7240)	4.73(0.6661)	**2.84(0.3752)**
f_4	5.37(0.5508)	6.42(0.4859)	6.01(0.4812)	**3.98(0.2573)**
f_5	13.62(2.01)	14.26(1.05)	14.16(0.4513)	**11.83(0.7850)**
f_6	5.31(1.10)	6.43(1.13)	5.71(0.9161)	**3.59(0.6987)**
f_7	5.76(0.4429)	6.88(0.8796)	6.87(0.8923)	**4.31(0.5801)**
f_8	4.65(0.1321)	5.77(0.0448)	5.09(0.0759)	**3.36(0.2296)**
f_9	4.61(0.3609)	5.57(0.4635)	5.04(0.2962)	**3.06(0.1857)**
f_{10}	5.29(0.4536)	6.20(0.3545)	5.89(0.2355)	**3.93(0.3334)**
f_{11}	13.32(1.40)	14.66(1.76)	14.20(1.33)	**11.59(0.8922)**
f_{12}	7.13(0.4137)	8.32(0.4493)	7.82(0.2592)	**5.91(0.5734)**
f_{13}	4.59(0.3905)	5.90(0.6316)	5.49(0.7949)	**3.23(0.3999)**
f_{14}	5.06(0.3640)	6.16(0.5907)	5.44(0.904)	**3.61(0.5937)**
f_{15}	16.61(0.2303)	17.66(0.3023)	17.05(0.1871)	**15.00(0.1926)**
f_{16}	4.81(0.6456)	5.64(0.6802)	5.39(0.6544)	**3.31(0.4860)**
f_{17}	7.17(0.9957)	8.68(1.29)	8.25(1.17)	**5.92(0.8351)**
f_{18}	6.09(0.1622)	7.90(0.7548)	7.50(0.4067)	**5.60(0.4918)**
f_{19}	5.73(0.7677)	6.90(0.9608)	6.55(0.9572)	**4.10(0.6065)**
f_{20}	15.86(2.44)	17.41(2.21)	16.21(1.65)	**14.22(1.67)**
f_{21}	5.45(0.0153)	6.49(0.0671)	6.02(0.0232)	**4.00(0.0143)**
f_{22}	5.63(0.0401)	5.98(0.0578)	5.44(0.1188)	**3.39(0.0096)**

Table 4.6 Mean and standard deviation of time (in seconds) taken by each algorithm over 30 runs.

On the other hand, in every case PSO is faster than our proposal (Table 3.7). By inspection of Tables 3.5 and 3.7, and from a statistical point of view concerning the Wilcoxon test, we can conclude that our algorithm gives similar results than ABC and DE, without adding extra computational complexity. However, even though our algorithm gives better results, it also increases the computational cost, compared with PSO. With the purpose of illustrate the behavior of our proposal, in Table 3.8 are shown some results of AO against ABC, DE and PSO after a single run.

f	AO vs. ABC	AO vs. DE	AO vs. PSO
f_{1}	**0.0018**	**0.0009**	0.9996
f_{2}	**0.0090**	**0.0056**	0.9999
f_{3}	**0.0112**	**0.0178**	0.9999
f_{4}	**9.07e-05**	**0.0001**	0.9995
f_{5}	**0.0031**	**0.0001**	0.9978
f_{6}	**0.0044**	0.0564	0.9999
f_{7}	**3.50e-05**	**6.78e-05**	0.9999
f_{8}	**3.52e-06**	**1.11e-05**	0.9999
f_{9}	**2.09e-05**	**5.19e-06**	0.9999
f_{10}	**1.31e-05**	**2.41e-06**	0.9999
f_{11}	**0.0001**	**0.0005**	0.9999
f_{12}	**2.43e-06**	**5.23e-06**	0.9999
f_{13}	**1.00e-05**	**1.56e-06**	0.9999
f_{14}	**1.36e-06**	**2.32e-05**	0.9999
f_{15}	**4.66e-05**	**8.20e-05**	0.9999
f_{16}	**6.92e-06**	**0.0016**	0.9999
f_{17}	**0.0001**	**0.0019**	0.9999
f_{18}	**1.51e-05**	**3.27e-06**	0.9999
f_{19}	**1.05e-05**	**0.0009**	0.9999
f_{20}	**2.60e-06**	**0.0016**	0.9999
f_{21}	**1.50e-05**	**3.25e-05**	0.9999
f_{22}	**7.00e-06**	0.9999	0.9999

Table 3.7 *p*-values produced by Wilcoxon test; comparisons were made between AO vs. ABC, DE and PSO, over times found in 30 independent runs.

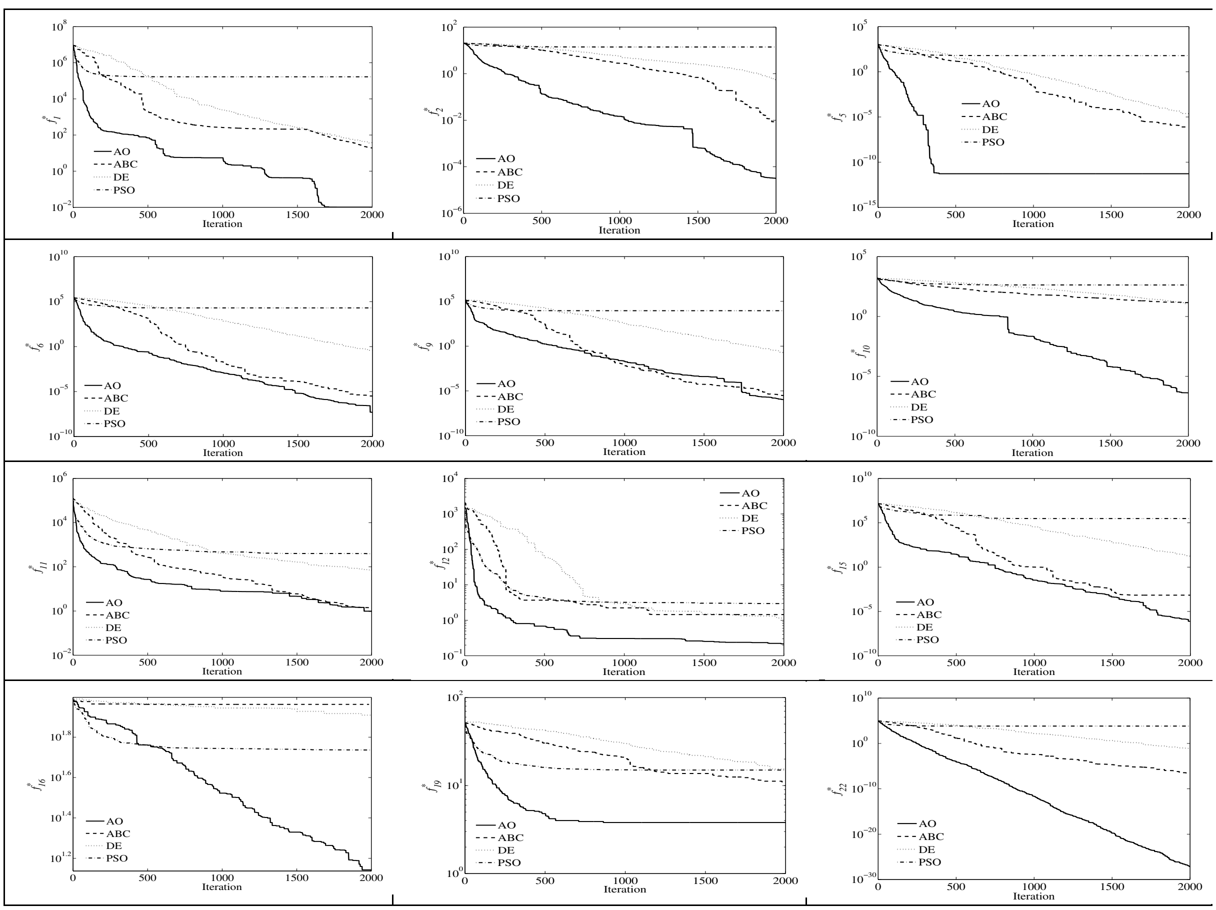

Table 3.8 Some convergence graphs after a single run.