IWO with Increased Deviation and Stochastic Selection (IWO-ID-SS) for global optimization of noisy fitness functions

Kaushik Suresh[1], Debarati Kundu[1], Sayan Ghosh[1], Swagatam Das[1] and Ajith Abraham[2]

[1]Dept. of Electronics and Telecommunication Engg,
Jadavpur University, Kolkata, India
[2] Norwegian University of Science and Technology, Norway

kaushik_s1988@yahoo.com, kundu.debarati@gmail.com, sayan88tito@gmail.com,
swagatamdas19@yahoo.co.in, ajith.abraham@ieee.org

Abstract: *Invasive weed optimization (IWO) has been found to be a simple but powerful algorithm for function optimization over continuous spaces. It has reportedly outperformed many types of evolutionary algorithms and other search heuristics when tested over both benchmark and real-world problems. However the performance of most search heuristics deteriorates severely when applied to the task of optimization of noisy landscapes. This paper presents an improved IWO algorithm to effectively find the global optima of noisy functions. This is achieved by using an increased value of standard deviation, changing the manner of its reduction to linear and by employing a novel selection strategy which varies from the one use in the standard IWO. An extensive performance comparison of the newly proposed scheme, the original DE (DE/Rand/1/Exp), the canonical PSO, standard real-coded EA, and DE-RSF-TS has been presented using well-known benchmarks corrupted by zero-mean Gaussian noise. It has been found that the proposed method outperforms the others in a statistically significant way.*

Index Terms — Invasive Weed Optimization (IWO), noisy landscapes, stochastic selection.

1 Introduction

The problem of optimizing noisy or imprecise (not exactly known) functions occurs in diverse domains of engineering application, especially in the task of experimental optimizations. In some applications the function to be optimized is only known within some (often unknown and low) precision. This might be due to the fact that evaluation of the function involves measuring some physical or chemical quantity or performing a finite element calculation in order to solve partial differential equations. The function values obtained are corrupted by noise, namely stochastic measurement errors and/or quantization errors. Although the underlying function may be smooth, the function values available may show a discontinuous behavior. Moreover, no gradient information may be available. A number of evolutionary computation methods for dealing with these noisy optimization problems have been proposed in the last few years in the fields of evolutionary programming (EP) [1], evolution strategies (ES) [2], genetic algorithms (GA) [3] and particle swarm optimization (PSO) [4]. DE-RSF (DE with Random Scale Factor) was proposed by Das et al in [5] and, by and large, outperformed other heuristics. However, to the best of our knowledge, IWO [6] has not been applied to the task of optimizing fitness functions.

In this paper, we propose a modified IWO algorithm in which the value of standard deviation is increased and it is allowed to fall more slowly (linearly) over generations. This makes the IWO algorithm more stochastic and equips it with better explorative powers in a fitness landscape which is corrupted by noise and hence continuously changing. Also, a threshold based selection strategy has been incorporated into the IWO algorithm. This variant of IWO is called IWO-ID-SS (IWO with Increased Deviation and Stochastic Selection). An extensive performance comparison has been carried out among conventional DE, canonical PSO, EA, DE-RSF and the proposed scheme. The rest of the paper is organized as follows. In section 2, the IWO algorithm is briefly introduced. The new variant of IWO is explained in section 3. Section 4 contains the details of the benchmark functions, simulation strategies and a mention of competitor algorithms used for

215

comparison. The results are presented in section 5. Finally, we conclude the paper in section 6.

2 The IWO algorithm

IWO is a search algorithm that emulates the colonizing behavior of weeds. The steps involved in the classical IWO algorithm may be summarized by the following pseudo code:

1. Initialize a population (say of size $np - init$) over the d dimensional search space.
2. While stopping criterion is not met, do steps 3 to 5.
3. Each member of the population is allowed to produce a number of seeds depending on its own and the population's minimum and maximum fitness. The number of seeds produced is allowed to increase linearly according to fitness.

$$S_i = S_{min} + (curr\_pop - rank_i) * (S_{max} - S_{min)}$$

(1)

where S_{min} and S_{max} are minimum and maximum number of seeds produced by each weed.

4. Randomness and adaptation in the algorithm is provided by the spatial dispersal. The generated seeds are distributed over the search space by normally distributed random numbers with mean equal to zero but varying variance. The standard deviation is reduced over successive generations according to the given formula:

$$\sigma_{iter} = (iter_{max} - iter)^n * (\sigma_{initial} - \sigma_{final})/(iter_{max})^n + \sigma_{final}$$

(2)

Here, $\sigma_{initial}$ is dependent on the function, σ_{final} is usually of the order of 0.01.

5. When the weed population exceeds the upper limit (P_{max}), we carry out sorting and the best p_{max} are allowed to survive.

3 The proposed scheme

IWO is an algorithm that is quite sensitive to initialization and parameter values. In order to use IWO to optimize noisy landscapes, four changes have been made to the original algorithm.

In the original algorithm, the standard deviation (σ_{iter}) varies in a non-linear fashion (modulation parameter n is set to 3). In our scheme, n is set to 1 to allow a larger value of standard deviation near the time of completion of iterations. Secondly, we set the value of σ_{final} at 0.5 to accomplish the aforementioned objective.

Following the work done on (1+1)-EA in [7] we take up a threshold-based selection procedure for IWO-ID-SS. Here the offspring vector substitutes its parent vector in the new generation if its fitness is less than the fitness of the parent (in case of minimization problems) by a threshold margin . We keep the threshold margin proportional to the noise strength or variance (δ^2) according to the following formula.

$$\tau = k.\delta^2$$

(3)

Finally, unlike the standard IWO where exceeding upper limit of population size causes the best members to be selected, we allow poorer members to be selected with a certain probability.

The pseudo code of the new algorithm is as follows:

1. Initialize a population (say of size $np - init$) over the d dimensional search space.
2. While stopping criterion is not met, do steps 3 to 5.
3. Each member of the population is allowed to produce a number of seeds depending on its own and the population's minimum and maximum fitness. The number of seeds produced is allowed to increase linearly according to fitness.
4. Randomness and adaptation in the algorithm is provided by the spatial dispersal. The generated seeds are distributed over the search space by normally distributed random numbers with mean equal to zero but varying variance. The standard deviation is reduced over successive generations according to the given formula:

$$\sigma_{iter} = (iter_{max} - iter)^n * (\sigma_{initial} - \sigma_{final})/(iter_{max})^n + \sigma_{final}$$

(4)

Here, $\sigma_{initial}$ is dependent on the function, σ_{final} is 0.5 and $n = 1$.

5. When the weed population exceeds the upper limit (p_{max}), we carry out sorting. The newer

population members have their fitnesses increased by the threshold value (in case of minimization problems). The population is then ranked. Those members ranked worse than p_{max} are given a chance to enter the new population with a probability P =0.02. After selection, the fitnesses are re-evaluated.

4 Experimental Setup and Simulation Strategy

4.1 Benchmark functions used

We have used the noisy versions of the six well known benchmark functions[8], namely Sphere, Rosenbrock, Rastrigin, Griewank, Schaffer's f_6 (2-D) and De Jong's f_4 (2-D). All of these are minimization problems. The noisy versions of th benchmark functions are defined as follows:

$$f_{noisy} = f + N(0, \delta^2)$$

with $N(0, \delta^2)$ = Normal (or Gaussian) distribution with mean 0 and variance δ^2 To generate N, we use the Box and Muller method [9] with various values of δ^2.

4.2 Methods Compared and Algorithmic Settings

In this work we compare the performance of Particle Swarm Optimization (PSO), classical DE, Evolutionary Algorithm [10] and DE-RSF-TS with that of the new scheme on the noisy benchmarks listed above. Owing to lack of space here, we avoid going into the details of the competitor algorithms.
The parameters used for the IWO-ID-SS algorithm for various benchmark functions and the search and initialization ranges given in Table 1.

4.3 Simulation Strategy

In this study each experiment has been repeated 50 times and the mean best fitness of these 50 independent runs has been reported. For comparing different algorithms the first thing we require is a fair metric. The number of iterations or generations cannot be used as a time measure as the algorithms perform different amounts of work in their inner loops. We have used the number of function evaluations as the metric of comparison. Each run of each experiment was continued up to 100000 function evaluations.

5 Results of Simulation

We have experimented with increasing noise variance values from 0 to 1. In Tables 2(1-6) we present the mean and standard deviation of the best fitness values for the six benchmark functions for noise standard deviations of 0.2, 0.4, 0.6, 0.8 and 1.0. All the readings given are the averages over 50 independent runs per function per noise variance value. The best solution in each case has been shown in bold. In Table 3 we report results of unpaired t-tests run between the proposed algorithm and the second best among DE1, PSO, EA and DE-RSF-TS in each case (standard error of difference of the two means, 95% confidence interval of this difference, the t value, and the two-tailed P value are presented). For all cases in Table 3, sample size = 50 and degrees of freedom = 98. It is interesting to see from Table 3 that the proposed method beats the nearest competitor in a statistically meaningful way in a majority of the cases. Fig 1 shows the convergence plots for Rastrigin and Rosenbrock functions with δ^2=1.

6 Conclusions

Most metaheuristics show significant degradation in performance when applied to the task of optimizing noisy fitness functions. To overcome this, IWO-ID-SS was proposed. Our proposed algorithm tackles noise in landscapes by using an increased value of variance and using a threshold based stochastic selection strategy. The new method has been shown to beat the competitor algorithms, namely PSO, the basic DE, EA and DE-RSF-TS over a six function test suite statistically significantly.
Future research will focus on mathematical analysis of IWO applied to noisy fitness functions so that optimal values of various parameters may be obtained.

Table 1: Parameter settings for the IWO-ID-SS algorithm

	Sphere	Rastrigin	Rosenbrock	Griewank	Schaffer's f_6	De Jong's f_4
Search range	[-100,100]	[-5.12,5.12]	[-50,50]	[-600,600]	[-100,100]	[-1.28,1.28]
Initialization range	[20,40]	[1,2]	[20,40]	[100,200]	[20,40]	[0.2,0.6]
np_init	20	20	20	20	20	20
p_{max}	60	60	60	60	60	60
$\sigma_{initial}$	10	3	5	20	10	0.6
σ_{final}	0.5	0.5	0.5	0.5	0.5	0.5
s_{min}	1	1	1	1	1	1
s_{max}	5	3	4	3	4	4

Table 2(1-6): Mean and standard deviation of best fitness after 100000 function evaluations for 6 benchmarks

1) Sphere

δ^2	DE	PSO	EA	DE-RSF-TS	IWO-ID-SS
0.2	0.000 (0.00)	0.0042 (0.029)	0.0675 (0.000)	**0.0000 (0.000)**	0.0281 (0.019)
0.4	0.0121 (0.008)	0.24 (0.06)	0.0231 (0.057)	**0.001 (0.043)**	0.0031 (0.001)
0.6	0.069 (0.326)	0.2976 (0.097)	0.0498 (0.087)	0.004 (0.002)	**0.0015 (0.005)**
0.8	0.22 (0.00)	0.3021 (0.078)	0.055 (0.771)	0.082 (0.331)	**0.026 (0.033)**
1.0	0.2145 (0.024)	0.3562 (0.030)	0.0502 (0.051)	0.0203 (0.018)	**0.0201 (0.008)**

2) Rastrigin

δ^2	DE	PSO	EA	DE-RSF-TS	IWO-ID-SS
0.2	0.034 (0.115)	21.142 (1.014)	34.641 (0.614)	0.67 (0.33)	**0.06001 (0.0414)**
0.4	0.121 (0.053)	38.376 (1.725)	35.612 (1.107)	0.056 (0.194)	**0.0520 (0.031)**
0.6	0.387 (0.012)	43.462 (1.614)	37.531 (0.915)	0.074 (0.291)	**0.0536 (0.011)**
0.8	1.422 (0.052)	50.031 (3.750)	37.469 (1.835)	0.0823 (0.168)	**0.0621 (0.0116)**
1.0	3.2725 (0.068)	57.653 (3.163)	39.695 (1.091)	**0.008 (0.043)**	0.0626 (0.0253)

3) Rosenbrock

δ^2	DE	PSO	EA	DE-RSF-TS	IWO-ID-SS
0.2	17.221 (0.020)	1210.03 (150.61)	57.815 (8.192)	12.694 (1.615)	**7.004 (5.719)**
0.4	17.012 (0.881)	1880.18 (167.36)	61.710 (7.048)	11.921 (1.952)	**4.764 (4.165)**
0.6	26.069 (0.326)	2231.73 (387.19)	65.912 (10.231)	15.151 (2.279)	**6.424 (6.372)**
0.8	31.882 (0.910)	3981.91 (539.54)	93.153 (11.017)	20.023 (2.414)	**6.257 (5.318)**
1.0	24.810 (1.210)	4492.48 (833.12)	115.681 (14.016)	18.514 (2.261)	**6.796 (5.99)**

4) Griewank

δ^2	DE	PSO	EA	DE-RSF-TS	IWO-ID-SS
0.2	0.0051 (0.023)	1.693 (0.094)	0.0035 (0.002)	0.075 (0.093)	**0.035803 (0.03852)**
0.4	0.421 (0.071)	4.932 (0.073)	0.0338 (0.005)	0.076 (0.254)	**0.03363 (0.03944)**
0.6	1.582 (0.091)	8.164 (0.142)	0.2854 (0.009)	**0.014 (0.164)**	0.04930 (0.05356)
0.8	2.814 (0.054)	9.216 (0.276)	1.3916 (0.014)	**0.012 (0.284)**	0.07818 (0.0713)
1.0	3.196 (0.293)	12.461 (0.618)	2.9854 (0.085)	0.0819 (0.128)	**0.06275 (0.0605)**

5) De Jong's function

δ^2	DE	PSO	EA	DE-RSF-TS	IWO-ID-SS
0.2	0.018 (0.017)	0.0451 (0.113)	0.047 (0.012)	0.0186 (0.005)	**0.0163 (0.008)**
0.4	0.1962 (0.021)	0.2462 (0.093)	**0.0051 (0.003)**	0.0192 (0.002)	0.0191 (0.006)
0.6	0.1215 (1.714)	0.2721 (0.165)	0.1035 (0.039)	0.019 (0.085)	**0.0186 (0.011)**
0.8	0.3673 (0.983)	0.6713 (0.79)	0.1548 (0.128)	**0.0042 (0.002)**	0.02046 (0.010)
1.0	0.7517 (2.715)	0.914 (1.023)	0.0861 (0.073)	0.034 (0.018)	**0.02688 (0.019)**

6) Schaffer's

δ^2	DE	PSO	EA	DE-RSF-TS	IWO-ID-SS
0.2	0.0053 (0.003)	0.075 (0.02)	0.0075 (0.000)	0.0085 (0.001)	**0.0052 (0.002)**
0.4	0.0192 (0.031)	0.225 (0.16)	0.0091 (0.010)	0.0061 (0.015)	**0.0095 (0.009)**
0.6	0.1168 (0.964)	0.261 (0.17)	0.126 (0.027)	**0.0089 (0.81)**	0.0128 (0.009)
0.8	0.3692 (0.902)	0.616 (0.79)	0.153 (0.082)	**0.0082 (0.339)**	0.0496 (0.027)
1.0	0.5005 (1.361)	0.814 (0.43)	0.219 (0.03)	0.0475 (0.208)	**0.0362 (0.048)**

Table 3(1-4): Unpaired t tests for best and second best algorithms for the six benchmark functions

1) Sphere:

δ^2	Std. Err	t	95% Conf. Intvl	Two-tailed P	Significance
0.2	0.0003	8.1005	-0.03504378 to -0.02115622	<0.0001	extremely statistically significant
0.4	0.008	0.2674	-0.01781911 to 0.01361911	0.7901	not statistically significant
0.6	0.001	4.8142	0.00146052 to 0.00353948	<0.0001	extremely statistically significant
0.8	0.006	8.6252	0.0430037 to 0.0689963	<0.0001	extremely statistically significant
1.0	0.004	0.0556	-0.00699876 to 0.00739876	0.9558	not statistically significant

2) Rastrigin:

δ^2	Std. Err	t	95% Conf. Intvl	Two-tailed P	Significance
0.2	0.060	10.1032	0.4891343 to 0.7308457	<0.0001	extremely statistically significant
0.4	0.036	0.1115	-0.06779908 to 0.07579908	0.9116	not statistically significant
0.6	0.006	3.5917	0.00903061 to 0.03176939	0.0007	extremely statistically significant
0.8	0.004	5.4194	0.01273885 to 0.02766115	<0.0001	extremely statistically significant
1.0	0.009	5.9942	-0.07283318 to -0.036	<0.0001	extremely statistically significant

3) Rosenbrock:

δ^2	Std. Err	t	95% Conf. Intvl	Two-tailed P	Significance
0.2	1.08	5.2444	3.51818 to 7.86182	<0.0001	extremely statistically significant
0.4	0.840	8.5223	5.4759734 to 8.8380266	<0.0001	extremely statistically significant
0.6	0.743	11.7446	7.2396004 to 10.2143996	<0.0001	extremely statistically significant
0.8	0.903	15.2415	11.95807 to 15.57393	<0.0001	extremely statistically significant
1.0	1.000	11.7156	9.7158751 to 13.7201249	<0.0001	extremely statistically significant

4) Griewank:

δ^2	Std. Err	t	95% Conf. Intvl	Two-tailed P	Significance
0.2	0.005	0.0367	-0.011019446 to 0.010619446	0.9708	not statistically significant
0.4	0.007	6.3911	-0.055526049 to -0.02921395	<0.0001	extremely statistically significant
0.6	0.024	1.4470	-0.083713086 to 0.013113086	0.1511	not statistically significant
0.8	0.041	1.5982	-0.148356951 to 0.015996951	0.1132	not statistically significant
1.0	0.009	2.1561	-0.036775494 to -0.00152450	0.0335	extremely statistically significant

5) De Jong's:

δ^2	Std. Err	t	95% Conf. Intvl	Two-tailed P	Significance
0.2	0.010	0.2261	-0.0217777 to 0.0173577	0.8219	statistically significant
0.4	0.007	1.8689	-0.0290158 to 0.0009958	0.0667	not statistically significant
0.6	0.086	0.0036	-0.1719555 to 0.1713355	0.9971	statistically significant
0.8	0.011	1.4557	-0.0386211851 to 0.0060991851	0.1509	not statistically significant
1.0	0.027	0.2666	-0.0605792015 to 0.0463412015	0.7908	extremely statistically significant

6): Schaffer's (2-D):

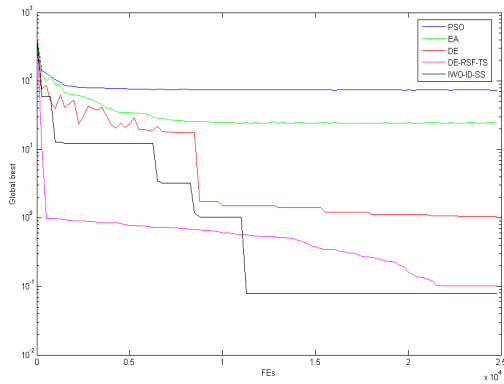| δ^2	Std. Err	t	95% Conf. Intvl	Two-tailed P	Significance
0.2	0.001	0.1961	-0.00111188 to 0.00091188	0.8449	not statistically significant
0.4	0.000	4.5255	-0.00230162 to -0.00089838	<0.0001	extremely statistically significant
0.6	0.115	0.0340	-0.23123740 to 0.22343740	0.9729	not statistically significant
0.8	0.048	0.8608	-0.13684032 to 0.05404032	0.3914	not statistically significant.
1.0	0.030	0.3827	-0.06989257 to 0.04729257	0.7028	statistically significant

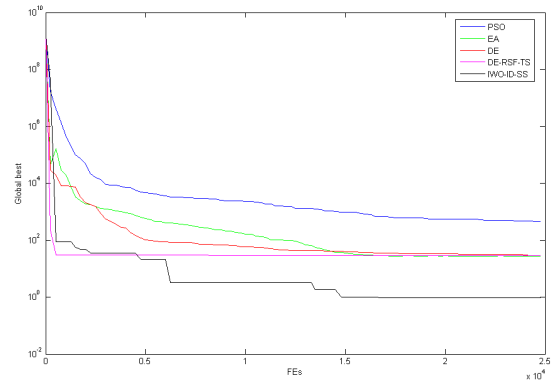Fig 1a): Rastrigin with $\delta^2 = 1$

Fig 1b): Rosenbrock with $\delta^2 = 1$.

Figure 1: Variation of mean best value with function evaluations

References

1. Fogel, L.J, Owens, A.J, Walsh, M.J.: Artificial Intelligence through Simulated Evolution. John Wiley & Sons, New York, (1966).

2. Rechenberg, I.: Evolution Strategy: Optimization of technical systems by means of biological evolution. Fromman-Holzboog, Stuttgart, (1973).

3. Holland, J.H.: Adaptation in Natural and Artificial Systems. Univ. of Michigan Press, Ann Arbor, MI, (1975).

4. Kennedy, J, Eberhart R.: Particle swarm optimization. Proc. IEEE Int. conf. Neural Networks. (1995) 1942-1948.

5. Swagatam Das, Amit Konar and Uday K. Chakraborty: Improved Differential Evolution for handling noisy optimization problems. Proc. IEEE Congress on Evolutionary Computation 2005.

6. A.R. Mehrabian and C.Lucas, "A novel numerical optimization algorithm inspired from weed colonization", Ecological Informatics 1(2006).

7. Markon, S, Arnold, V, D, Baick, T, Beislstein, T, Beyer, G.-H.: Threshholding - a Selection Operator for Noisy ES. Proc. Congress on Evolutionary Computation (CEC-2001). 465-472

8. Yao, X, Liu, Y, Lin, G.: Evolutionary programming made faster, IEEE Trans. on Evolutionary Computation, vol 3, No 2, (1999) 82-102.

9. Box, G. E. P, Muller, M. E.: A note on the generation of random deviates, Ann. Math. Statistics, vol. 29, (1958) 610-611.

10. Michalewicz, Z, Fogel, D. B.: How to Solve It: Modern Heuristics. Springer, Berlin, 2000.