

<http://ansinet.com/itj>

ITJ

ISSN 1812-5638

INFORMATION TECHNOLOGY JOURNAL

ANSI*net*

Asian Network for Scientific Information
308 Lasani Town, Sargodha Road, Faisalabad - Pakistan

Influence of Jumping Rate on Opposition-based Differential Evolution Using the Current Optimum

¹Lihua Zhao, ²Qingzheng Xu and ^{1,2}Jin Pan

¹Key Lab of Intelligent Perception and Image Understanding of Ministry of Education, Xidian University, Xi'an 710071, China

²Xi'an Communication Institute, Xi'an 710106, China

Abstract: Diverse forms of opposition are already existent virtually everywhere and utilizing opposite numbers to accelerate an optimization method is a new idea. In this study, Differential Evolution (DE) and opposition-based differential evolution using the current optimum (COODE) are compared for different jumping rates. Experiments on 58 widely used benchmark problems show that, the higher jumping rate leads to faster convergence to global optimum and smaller success rate for most problems.

Key words: Opposition-based learning, differential evolution, opposition-based differential evolution using the current optimum, jumping rate

INTRODUCTION

Many examples of opposition permeate in most fields around the world, in some form or another. The opposition concept has sometimes been called by different names, such as, opposite particles in physics, absolute or relative complement in set theory, antonyms in languages and dualism in religions and philosophies. The interaction between individual and its corresponding opposite individual is apparently fundamental for maintaining universal balance (Tizhoosh *et al.*, 2008). In fact, the opposition concept is applied, whether consciously or unconsciously, in most of our daily affairs.

After the basic concept of Opposition-Based Learning (OBL) was originally introduced by Tizhoosh (2005), in a very short period of time, it has been utilized in different soft computing areas. The main idea of this optimization is the simultaneous consideration of an estimate and its corresponding opposite estimate for finding a better candidate solution which is closer to the global optimum. A mathematical proof was proposed by Rahnamayan *et al.* (2008a) to show that, in general, the opposite candidate solution is more advantageous than a second pure random one. Recently, this proof was simplified from the perspective of distance to the optimal solution and generalized to N-dimensional search spaces for black-box problems (Rahnamayan *et al.* 2012).

As an acceleration strategy, a novel opposition-based learning strategy using the current optimum

(COOBL) was first proposed to solve function optimization problem (Xu *et al.*, 2011a, b). What's different with opposition-based learning is that, the optimum in the current generation, not the geometric center of variables's initial range or current interval, was served as a symmetry point between the current point and the corresponding opposite point. The nice improvement measure, unexpectedly, can lead to a high rate of opposite population usage, especially in the later stage of population evolution. The contribution of opposite point using the current optimum to make the intelligent algorithm faster was clearly confirmed by experiments results and was not reproducible by a uniformly generated random point.

The performance of population-based heuristic algorithm highly depends on the control parameters. Generally speaking, different parameter settings will lead to different performances. Three algorithms were compared for different problem dimensions and different population sizes (Xu *et al.*, 2012). The heuristic algorithms are Differential Evolution (DE), opposition-based differential evolution using the current optimum (COODE) and improved version of opposition-based differential evolution (IODE). Experiment results show that, COODE performs better than DE for larger population size which is usually required for more complex and high-dimensional optimization problems. In this study, the effects of jumping rate on the speedup of COODE are investigated completely.

RECENT ADVANCES IN OPPOSITION-BASED LEARNING

While it is original introduced to accelerate the convergence rate of standard differential evolution, the opposition-based learning is universal in the sense that it does not depend on the specific algorithm used. The general OBL idea has been successfully applied to reinforcement learning (Tizhoosh, 2005, 2006), genetic algorithm (Tizhoosh, 2005; Lin and Wang, 2010), differential evolution (Rahnamayan *et al.*, 2006a, 2008b), artificial neural networks (Ventresca and Tizhoosh, 2006, 2007a), ant colony optimization (Montgomery and Randall, 2002; Malisia and Tizhoosh, 2007), window memorization (Khalvati *et al.*, 2007), simulated annealing (Ventresca and Tizhoosh, 2007b), particle swarm optimization (Han and He, 2007; Wang *et al.*, 2007), fuzzy sets (Tizhoosh, 2009; Al-Qunaieer *et al.*, 2010) and biogeography-based optimization (BBO) (Ergezer *et al.*, 2009; Bhattacharyaa and Chattopadhyay, 2010).

To obtain more chances closely to the global optimum, Quasi-Opposition-Based Learning (QOBL) was proposed by Rahnamayan which was successfully used in DE (Rahnamayan *et al.*, 2007), PSO (Zhang *et al.*, 2009; Tang and Zhao, 2009) and BBO (Ergezer *et al.*, 2009). Mathematically speaking, a quasi-opposite point is a uniformly generated random point between the middle point and the opposite point. What's more, based on the concept of OBL, Wang proposed a generalized OBL (GOBL) (Wang *et al.*, 2009a). By simultaneously evaluating solutions in the current search space and transformed space, it will provide more chance of finding better solutions (Wang *et al.*, 2011; Wang *et al.*, 2009b). The most of the methods are often utilized to solve the numerical optimization problem, such as large scale optimization problem (Wang *et al.*, 2011; Wang *et al.*, 2009b; Rahnamayan and Wang, 2008), constrained optimization (Omran and Salman, 2009; Omran, 2010), optimization of noisy problem (Han and He, 2007; Rahnamayan *et al.*, 2006b), multi-objective optimization (Peng *et al.*, 2008; Dong and Wang, 2009). In addition, there are many possible uses and application domains of opposition concept: traveling salesman problem (Malisia and Tizhoosh, 2007; Ventresca and Tizhoosh, 2008), data mining (Kanzawa *et al.*, 2007; Rashid and Baig, 2010; Ventresca and Tizhoosh, 2009), nonlinear system identification (Subudhi and Jena, 2011), chess evaluation function tuning (Boskovic *et al.*, 2011) and image processing and understanding (Khalvati *et al.*, 2007;

Al-Qunaieer *et al.*, 2010; Sahba *et al.*, 2007; Rahnamayan and Tizhoosh, 2008; Tizhoosh, 2009; Sahba and Tizhoosh, 2008).

Compared with the successful application of classical and extended oppositional concepts to numerical optimization and real world problems, theory results are poor and it is worthy of further investigation. From the experiences of other intelligent computing technologies, purely random selection of solutions from a given population has the chance of visiting or even revisiting unproductive regions of the search space. A mathematical and experimental proof has been proposed to show that, in general, opposite numbers can increase the population diversity and are more likely to be closer to the optimal solution than independent random ones (Rahnamayan *et al.*, 2008a; Ventresca and Tizhoosh, 2008). In Rahnamayan *et al.* (2007) proved that a quasi-opposite point is more likely to be closer to the global optimum than the opposite point. Furthermore, he proved how much quasi-opposition is better than opposition by Ergezer *et al.* (2009). Generally speaking, the quasi-opposite points are closer to the center-point compared to the opposite-points. By this way, QOBL can be a promising evidence to support center-based sampling theory proposed by Rahnamayan and Wang (2009).

PROPER SETTING OF JUMPING RATE IN COODE

Opposition-based differential evolution using the current optimum: The original DE is chosen as a parent algorithm and the COODE studied in this paper is exactly the same as the ODE described by Rahnamayan *et al.* (2008b) except for the opposition-based learning which is replaced by opposition-based learning using the current optimum.

Corresponding pseudo-code for the proposed approach, COODE, is shown in Fig. 1, where, P_0 is the initial population, $COOP_0$ is the opposite of initial population after using COOBL, P is the current population, $COOP$ is the opposite population after using COOBL, V is the noise vector, U is the trial vector, P_i is the i th individual in P , $COOP_i$ is the i th individual in $COOP$, $[a, b]$ is the range of the j th variable, BFV is the best fitness value so far, VTR is the value to reach, NFC is the number of function calls, MAX_{NFC} is the maximum number of function calls, F is the mutation constant, $rand(0, 1)$ is the uniformly generated random number, C_r is the crossover rate, $f(\cdot)$ is the objective function, P' is the population of next generation, J_r is the jumping rate, N_p is the population size, D is the dimension size, $x_{0_{obj}}$ is the

```

Opposition-based differential evolution using the current optimum
/* Opposition-based population initialization using the current optimum */
1.  Generate uniformly distributed random population P0;
2.  for (i = 0; i < Np; i++)
3.  {
4.      for (j = 0; j < D; j++)
5.      {
6.          COOP0ij = 2x0coj - P0ij;
7.          if (COOP0ij > bj) || (COOP0ij < aj)
8.              COOP0ij = aj + rand(0, 1) × (bj - aj);
9.      }
10. }
11. Select Np fittest individuals from the set {P0, COOP0} as initial population P0;
/* End of Opposition- Based Population Initialization using the Current Optimum */

12. while (BFV > VTR and NFC < MAXNFC)
13. {
14.     for (i = 0; i < Np; i++)
15.     {
16.         Select three parents P1, P2, and P3 randomly from the current population where i ≠ i1 ≠ i2 ≠ i3;
17.         Vi = P1 + F × (P2 - P3);
18.         for (j = 0; j < D; j++)
19.         {
20.             if (rand(0, 1) < Cr)
21.                 Uij = Vij;
22.             else
23.                 Uij = Pij;
24.         }
25.         Evaluate Ui;
26.         if (f(Ui) = f(Pi))
27.             P'i = Ui;
28.         else
29.             P'i = Pi;
30.     }
31.     P = P';

/* Opposition-based generation jumping using the current optimum */
32. if (rand(0,1) < Jr)
33. {
34.     for (i = 0; i < Np; i++)
35.     {
36.         for (j = 0; j < D; j++)
37.         {
38.             COOPpj = 2xcoj - Pij;
39.             if (COOPpj > MAXpj) || (COOPpj < MINpj)
40.                 COOPpj = MINpj + rand(0, 1) × (MAXpj - MINpj);
41.         }
42.     }
43.     Select Np fittest individuals from the set {P, COOP} as current population P;
44. }
/* End of Opposition- Based Generation Jumping using the Current Optimum */
45. }

```

Fig. 1: Pseudo code for COODE

optimum value of the *j*th variable in the initial population, x_{coj} is the optimum value of the *j*th variable in the current population. Steps 12-31 are classical DE/rand/1/bin, steps 1-11 and 32-44 are implementation of opposition-based using the current optimum initialization and generation jumping, respectively and steps 6-8 and 38-40 (highlighted in boldface) are the difference between COODE and ODE.

Experimental setup: A comprehensive set of numerical benchmark functions, including 58 different well-known global optimization problems (Rahnamayan *et al.*, 2008b), has been employed for performance verification. The utilized test suite includes unimodal as well as highly multimodal minimization problems. The dimensionality of problems varies from 2-30 to cover a wide range of problem complexity.

For all conducted experiments, parameter settings are as follows. These values have been chosen according to reported setting in the previous literature (Rahnamayan *et al.*, 2008b) and so there has no new attempts to obtain better values for them. The termination criterion is to find a value smaller than the Value-To-Reach (VTR) before reaching the maximum Number of Function Calls (NFC):

- Population size, $N_p = 100$
- Differential amplification factor, $F = 0.5$
- Crossover probability constant, $C_r = 0.9$
- Jumping rate constant, $J_r = 0.3$
- Mutation strategy: DE/rand/1/bin (classic version of DE)
- Maximum NFC, $MAX_{NFC} = 10^6$
- Value to reach, $VTR = 10^{-8}$

The convergence speed is compared by measuring the number of function calls which is the most commonly used metric in literatures. A smaller NFC means higher convergence speed. In order to minimize the effect of the stochastic nature of the algorithms on the measured metric, the reported NFC for each function is the average over 100 trials.

The number of times, for which the algorithm successfully reaches the VTR for each test function, is measured as the Success Rate (SR):

$$SR = \frac{\text{No. of times reached VTR}}{\text{Total No. of trials}} \quad (1)$$

Experiment results: Here, a time varying jumping rate model is investigated. According to this model, the jumping rate increases/decreases linearly during the evolution based on the number of function evaluations. In order to support as fair as possible comparison between these four different jumping rate settings, the average jumping rate should be the same for all of them. Four proposed settings for the current investigation are as follows: J_r (constant) = 0.3, J_r (JrUp) = J_r (JrDown) = J_r (constant) = 0.6, where MAX_{NFC} and NFC are the maximum number of function calls and the current number of function calls, respectively. Figure 2 shows the corresponding diagrams (jumping rate vs. NFC) for four settings. J_r (JrDown) represents higher jumping rate during exploration and lower jumping rate during exploitation (tuning); J_r (JrUp) performs exactly in reverse

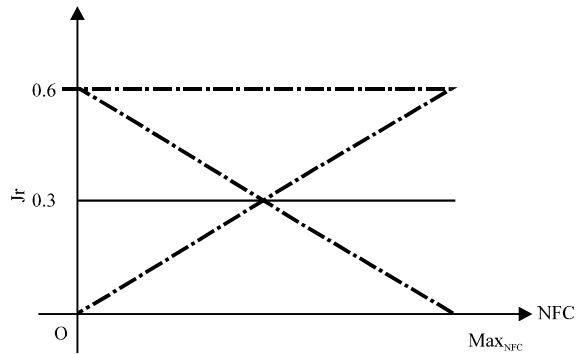


Fig. 2: Jumping rate vs. NFC

manner. By these settings, the effects of generation jumping of COODE during optimization process are investigate completely.

Results of applying DE, COODE ($J_r = 0.3$), COODE (JrUp), COODE (JrDown) and COODE ($J_r = 0.6$) to solve 58 test problems are given in Table 1. The best success performance for each function is highlighted in boldface. The last rows of the table present the sum for NFC and the average SR. These algorithms are ranked as COODE ($J_r = 0.3$) (best), COODE ($J_r = 0.6$), COODE (JrDown), COODE (JrUp) and DE with the respect to the total NFC to solve 58 problems. COODE (JrDown) presents the lowest average success rate (0.78); while DE and ODE (JrUp) show the highest one (0.88).

Pair comparison of these algorithms is presented in Table 2. Given number in each cell shows on how many functions the algorithm in the table's row outperforms the corresponding algorithm in the table's column. The last column of the table shows the total numbers (number of cases which the algorithm can outperform other competitors). By comparing these numbers, the following ranking result is obtained: COODE ($J_r = 0.6$) (best), COODE (JrDown), COODE ($J_r = 0.3$), COODE (JrUp) and DE. Results for $J_r = 0.3$ and $J_r = 0.6$ confirm that the constant higher jumping rate reduces the overall success rate.

According to Table 1 and 2, it is noted that the performance of COODE (JrDown) is similar to that of COODE ($J_r = 0.6$) in terms of both NFC and SR. As seen above, the number of function calls for most benchmark functions is negligible compared with pre-determined MAX_{NFC} , 10^6 . From Fig. 2, the jumping rate of COODE (JrDown) changes marginally during evolution process in this case. Since, all control parameters but jumping rate are the same between COODE (JrDown) and COODE

Table 1: Comparison of DE, COODE ($J_r = 0.3$), COODE (JrUp), COODE (JrDown) and COODE ($J_r = 0.6$)

Function	DE		COODE ($J_r = 0.3$)		COODE (JrUp)		COODE (JrDown)		COODE ($J_r = 0.6$)	
	NFC	SR	NFC	SR	NFC	SR	NFC	SR	NFC	SR
f ₁	73572	1.00	54797	1.00	73777	1.00	47063	1.00	47435	1.00
f ₂	81342	1.00	60091	1.00	81745	1.00	52232	1.00	52567	1.00
f ₃	153919	1.00	104266	1.00	154554	1.00	98128	1.00	99008	1.00
f ₄	360106	1.00	386333	0.91	358800	1.00	897209	0.66	884679	0.70
f ₅	297631	0.93	-	0.00	282685	0.86	-	0.00	-	0.00
f ₆	77655	1.00	57248	1.00	77615	1.00	49999	1.00	49566	1.00
f ₇	19142	1.00	11976	1.00	19335	1.00	8944	0.97	8764	1.00
f ₈	144154	1.00	105765	1.00	143844	1.00	91888	1.00	92042	1.00
f ₉	3227	1.00	2940	1.00	3154	1.00	2766	1.00	2796	1.00
f ₁₀	13632	1.00	11539	1.00	13896	1.00	11554	0.97	10803	1.00
f ₁₁	5313	1.00	4568	1.00	5336	1.00	4164	1.00	4130	1.00
f ₁₂	4127	1.00	3770	1.00	4064	1.00	3579	1.00	3574	1.00
f ₁₃	10889	0.92	9078	0.68	10905	0.74	8477	0.57	8335	0.51
f ₁₄	4815	1.00	3284	1.00	4736	1.00	2976	1.00	3035	1.00
f ₁₅	81475	1.00	58960	0.83	81244	0.96	51703	0.72	50968	0.78
f ₁₆	2679	1.00	2543	1.00	2750	1.00	2457	1.00	2470	1.00
f ₁₇	-	0.00	37325	0.04	73000	0.02	35157	0.07	31129	0.07
f ₁₈	186720	0.40	-	0.00	183845	0.51	-	0.00	-	0.00
f ₁₉	356337	1.00	172694	1.00	360356	1.00	171890	0.91	172022	0.90
f ₂₀	6345	1.00	3666	1.00	6031	1.00	3112	1.00	3040	1.00
f ₂₁	155494	1.00	106165	1.00	155368	1.00	89258	1.00	91844	1.00
f ₂₂	-	0.00	-	0.00	579600	0.05	-	0.00	-	0.00
f ₂₃	19856	1.00	14079	1.00	19660	1.00	15757	1.00	15701	1.00
f ₂₄	-	0.00	-	0.00	-	0.00	-	0.00	-	0.00
f ₂₅	9880	1.00	9147	1.00	9642	1.00	8414	1.00	8467	1.00
f ₂₆	9636	1.00	8105	0.91	9524	1.00	7421	0.76	7424	0.70
f ₂₇	9025	1.00	7883	0.94	8890	1.00	7344	0.9	7300	0.90
f ₂₈	8876	1.00	7880	1.00	8917	1.00	7359	0.9	7366	0.93
f ₂₉	8455	1.00	7051	0.76	8257	0.98	6503	0.6	6500	0.64
f ₃₀	41928	1.00	30141	1.00	41910	1.00	26426	0.93	26089	0.94
f ₃₁	340427	1.00	102760	0.79	334198	1.00	88221	0.63	86816	0.58
f ₃₂	4916	1.00	3625	1.00	4972	1.00	3251	1.00	3259	1.00
f ₃₃	25696	1.00	13398	1.00	25774	1.00	13115	0.13	11619	0.21
f ₃₄	30552	1.00	12744	0.41	30687	1.00	10713	0.15	10300	0.21
f ₃₅	3005	1.00	2792	1.00	3032	1.00	2673	1.00	2676	1.00
f ₃₆	7556	1.00	3742	1.00	7218	1.00	3149	1.00	3130	1.00
f ₃₇	4120	1.00	3898	1.00	4154	1.00	3683	1.00	3654	1.00
f ₃₈	4117	1.00	3866	1.00	4089	1.00	3716	1.00	3693	1.00
f ₃₉	2874	1.00	2715	1.00	2912	1.00	2612	1.00	2637	1.00
f ₄₀	7184	1.00	4920	1.00	7101	1.00	4492	1.00	4491	1.00
f ₄₁	16074	1.00	14479	1.00	16107	1.00	13530	1.00	13502	1.00
f ₄₂	3579	1.00	3284	1.00	3538	1.00	3162	1.00	3126	1.00
f ₄₃	5468	1.00	4811	1.00	5401	1.00	4489	1.00	4417	1.00
f ₄₄	5520	1.00	5181	1.00	5528	1.00	4884	1.00	4892	1.00
f ₄₅	2549	1.00	2416	1.00	2541	1.00	2314	1.00	2296	1.00
f ₄₆	4826	1.00	4470	1.00	4849	1.00	4246	1.00	4180	1.00
f ₄₇	2610	1.00	2453	1.00	2590	1.00	2325	1.00	2366	1.00
f ₄₈	2475	1.00	2202	1.00	2475	1.00	2040	1.00	2104	1.00
f ₄₉	4418	0.94	3783	0.87	4523	0.93	3366	0.82	3390	0.88
f ₅₀	242434	1.00	121722	0.86	228360	1.00	112163	0.72	114800	0.67
f ₅₁	-	0.00	-	0.00	-	0.00	-	0.00	-	0.00
f ₅₂	21944	1.00	19145	1.00	21829	1.00	17536	1.00	17228	1.00
f ₅₃	6834	1.00	4199	1.00	6751	1.00	3626	1.00	3578	1.00
f ₅₄	7369	1.00	6636	1.00	7370	1.00	6308	1.00	6198	1.00
f ₅₅	-	0.00	-	0.00	-	0.00	-	0.00	-	0.00
f ₅₆	-	0.00	-	0.00	-	0.00	-	0.00	-	0.00
f ₅₇	15458	1.00	13483	1.00	15544	1.00	12455	1.00	12443	1.00
f ₅₈	13592	1.00	11035	1.00	13430	1.00	10534	1.00	10934	1.00
Sum/average	2931827	0.88	1617728	0.83	2895813	0.88	2015226	0.78	2003654	0.79

DE: Differential evolution, COODE: Opposition-based differential evolution using the current optimum, NFC: No. of function calls and SR: Success rate

($J_r = 0.6$), they exhibit the similar results. For the same reason, there is no difference for the experiments results of COODE (JrUp) and DE. It is noting that, COODE degrades DE when the jumping rate is equal to zero.

Table 2: Pair comparison of DE, COODE ($J_r = 0.3$), COODE (JrUp), COODE (JrDown) and COODE ($J_r = 0.6$)

Algorithm	DE	COODE ($J_r = 0.3$)	COODE (JrUp)	COODE (JrDown)	COODE ($J_r = 0.6$)	Total
DE	-	3	22	3	3	31
COODE ($J_r = 0.3$)	49	-	49	3	2	103
COODE (JrUp)	31	3	-	3	3	40
COODE (JrDown)	49	47	49	-	21	166
COODE ($J_r = 0.6$)	49	48	49	29	-	175

DE: Differential evolution, COODE: Opposition-based differential evolution using the current optimum

CONCLUSION

Utilizing opposite points to accelerate an optimization method is a new idea. In this paper, the influence of the jumping rate was studied. Experiment results confirm that the higher jumping rate leads to faster convergence to global optimum and smaller success rate for most problems. The future work should be done on applying the COOBL scheme in different fields of soft computing in order to enhance evolutionary algorithms, artificial neural networks, ant colony algorithms, particle swarm optimization, etc.

ACKNOWLEDGMENT

This study was supported by the National Natural Science Foundation of China (Nos. 61001202 and 61003199).

REFERENCES

Al-Qunaieer, F.S., H.R. Tizhoosh and S. Rahnamayan, 2010. Oppositional fuzzy image thresholding. Proceedings of the IEEE International Conference on Fuzzy Systems, July 18-23, 2010, Barcelona, Spain, pp: 1-7.

Bhattacharyaa, A. and P.K. Chattopadhyay, 2010. Solution of economic power dispatch problems using oppositional biogeography-based optimization. Electr. Power Components Syst., 38: 1139-1160.

Boskovic, B., J. Brest, A. Zamuda, S. Greiner and V. Zumer, 2011. History mechanism supported differential evolution for chess evaluation function tuning. Soft Comput., 15: 667-683.

Dong, N. and Y.P. Wang, 2009. Multiobjective differential evolution based on opposite operation. Proceedings of the International Conference on Computational Intelligence and Security, December 11-14, 2009, Beijing, China, pp: 123-127.

Ergezer, M., D. Simon and D.W. Du, 2009. Oppositional biogeography-based optimization. Proceedings of the IEEE International Conference on Systems, Man and Cybernetics, October 11-14, 2009, San Antonio, TX., USA., pp: 1009-1014.

Han, L. and X.S. He, 2007. A novel opposition-based particle swarm optimization for noisy problems. Proceedings of the 3rd International Conference on Natural Computation, August 24-27, 2007, Haikou, China, pp: 624-629.

Kanzawa, Y., Y. Endo and S. Miyamoto, 2007. Fuzzy c-means algorithms for data with tolerance based on opposite criteria. IEICE Trans. Fundamen. Electron. Commun. Comput. Sci., E90: 2194-2202.

Khalvati, F., H.R. Tizhoosh and M.D. Aagaard, 2007. Opposition-based window memorization for morphological algorithms. Proceedings of the IEEE Symposium on Computational Intelligence in Image and Signal Processing, April 1-5, 2007, Honolulu, USA., pp: 425-430.

Lin, Z.Y. and L.L. Wang, 2010. A new opposition-based compact genetic algorithm with fluctuation. J. Comput. Inform. Syst., 6: 897-904.

Malisia, A.R. and H.R. Tizhoosh, 2007. Applying opposition-based ideas to the ant colony system. Proceedings of the IEEE Swarm Intelligence Symposium, April 1-5, 2007, Honolulu, USA., pp: 182-189.

Montgomery, J. and M. Randall, 2002. Anti-pheromone as a tool for better exploration of search space. Proceedings of the 3rd International Workshop on Ant Algorithms, September, 2002, Brussels, Belgium, pp: 100-110.

Omran, M.G.H. and A. Salman, 2009. Constrained optimization using CODEQ. Chaos Solitons Fractals, 42: 662-668.

Omran, M.G.H., 2010. CODEQ: An effective metaheuristic for continuous global optimization. Int. J. Metaheuristics, 1: 108-131.

Peng, L., Y.Z. Wang and G.M. Dai, 2008. A novel opposition-based multi-objective differential evolution algorithm for multi-objective optimization. Adv. Comput. Intell., 5370: 162-170.

Rahnamayan, S., H.R. Tizhoosh and M.M.A. Salama, 2006a. Opposition-based differential evolution algorithms. Proceeding of the IEEE Congress on Evolutionary Computation, July 16-21, 2006, Vancouver, Canada, pp: 2010-2017.

- Rahnamayan, S., H.R. Tizhoosh and M.M.A. Salama, 2006b. Opposition-based differential evolution for optimization of noisy problems. Proceedings of the IEEE Congress on Evolutionary Computation, July 16-21, 2006, Vancouver, Canada, pp: 1865-1872.
- Rahnamayan, S., H.R. Tizhoosh and M.M.A. Salama, 2007. Quasi-oppositional differential evolution. Proceedings of the IEEE Congress on Evolutionary Computation, September 25-28, 2007, Singapore, pp: 2229-2236.
- Rahnamayan, S. and G.G. Wang, 2008. Solving large scale optimization problems by Opposition-based Differential Evolution (ODE). WSEAS Trans. Comput., 7: 1792-1804.
- Rahnamayan, S. and H.R. Tizhoosh, 2008. Image thresholding using micro opposition-based (Micro-ODE). Proceedings of the IEEE Congress on Evolutionary Computation, June 1-6, 2008, Hong Kong, China, pp: 1409-1416.
- Rahnamayan, S., H.R. Tizhoosh and M.M.A. Salama, 2008a. Opposition versus randomness in soft computing techniques. Applied Soft Comput., 8: 906-918.
- Rahnamayan, S., H.R. Tizhoosh and M.M.A. Salama, 2008b. Opposition-based differential evolution. IEEE Trans. Evol. Comput., 12: 64-79.
- Rahnamayan, S. and G.G. Wang, 2009. Center-based sampling for population-based algorithms. Proceedings of the IEEE Congress on Evolutionary Computation, May 18-21, 2009, Trondheim, Norway, pp: 933-938.
- Rahnamayan, S., G.G. Wang and M. Ventresca, 2012. An intuitive distance-based explanation of opposition-based sampling. Applied Soft Comput., 12: 2828-2839.
- Rashid, M. and A.R. Baig, 2010. Improved opposition-based pso for feedforward neural network training. Proceeding of the International Conference on Information Science and Applications, April 21-23, 2010, Seoul, Korea, pp: 1-6.
- Sahba, F. and H.R. Tizhoosh, 2008. Opposite Actions in Reinforced Image Segmentation. In: Studies in Computational Intelligence: Oppositional Concepts in Computational Intelligence, Tizhoosh, H.R. and M. Ventresca (Eds.). Springer, USA., pp: 287-297.
- Sahba, F., H.R. Tizhoosh and M.M.M.A. Salama, 2007. Application of opposition-based reinforcement learning in image segmentation. Proceedings of the IEEE Symposium on Computational Intelligence in Image and Signal Processing, April 1-5, 2007, Honolulu, USA., pp: 246-251.
- Subudhi, B. and D. Jena, 2011. A differential evolution based neural network approach to nonlinear system identification. Applied Soft Comput., 11: 861-871.
- Tang, J. and X.J. Zhao, 2009. An enhanced opposition-based particle swarm optimization. Proceedings of the 2009 WRI Global Congress on Intelligent Systems, May 19-21, 2009, Xiamen, China, pp: 149-153.
- Tizhoosh, H.R., 2005. Opposition-based learning: A new scheme for machine intelligence. Proceedings of the International Conference on Computational Intelligence for Modelling, Control and Automation and International Conference on Intelligent Agents, Web Technologies and Internet Commerce, Volume 1, November 28-30, 2005, Vienna, Austria, pp: 695-701.
- Tizhoosh, H.R., 2006. Opposition-based reinforcement learning. J. Adv. Comput. Intell. Intell. Inform., 10: 578-585.
- Tizhoosh, H.R., 2009. Opposite fuzzy sets with applications in image processing. Proceedings of the International Fuzzy Systems Association World Congress and 2009 European Society of Fuzzy Logic and Technology Conference, July 20-24, 2009, Aix-Les-Bains, France, pp: 36-41.
- Tizhoosh, H.R., M. Ventresca and S. Rahnamayan, 2008. Opposition-Based Computing. In: Studies in Computational Intelligence: Oppositional Concepts in Computational Intelligence, Tizhoosh, H.R. and M. Ventresca (Eds.). Springer, USA., pp: 11-28.
- Ventresca, M. and H.R. Tizhoosh, 2006. Improving the convergence of backpropagation by opposite transfer functions. Proceedings of the International Joint Conference on Neural Networks, July 16-21, 2006, Vancouver, Canada, pp: 4777-4784.
- Ventresca, M. and H.R. Tizhoosh, 2007a. Opposite transfer functions and backpropagation through time. Proceedings of the IEEE Symposium on Foundations of Computational Intelligence, April 1-5, 2007 Honolulu, USA., pp: 570-577.
- Ventresca, M. and H.R. Tizhoosh, 2007b. Simulated annealing with opposite neighbors. Proceedings of the IEEE Symposium on Foundations of Computational Intelligence, April 1-5, 2007, Honolulu, USA., pp: 186-192.
- Ventresca, M. and H.R. Tizhoosh, 2008. A diversity maintaining population-based incremental learning algorithm. Inform. Sci., 178: 4038-4056.
- Ventresca, M. and H.R. Tizhoosh, 2009. Improving gradient-based learning algorithms for large scale feedforward networks. Proceedings of the International Joint Conference on Neural Networks, June 4-19, 2009, Atlanta, USA., pp: 3212-3219.

- Wang, H., Y. Liu, S.Y. Zeng, H. Li and C.H. Li, 2007. Opposition-based particle swarm algorithm with cauchy mutation. Proceedings of the IEEE Congress on Evolutionary Computation, September 25-28, 2007, Singapore, pp: 4750-4756.
- Wang, H., Z.J. Wu, S. Rahnamayan and L.S. Kang, 2009b. A scalability test for accelerated de using generalized opposition-based learning. Proceedings of the 9th International Conference on Intelligent Systems Design and Applications, November 30-December 2, 2009, Pisa, Italy, pp: 1090-1095.
- Wang, H., Z.J. Wu, Y. Liu, J. Wang, D.Z. Jiang and L.L. Chen, 2009a. Space transformation search: A new evolutionary technique. Proceedings of the ACM/SIGEVO Summit on Genetic and Evolutionary Computation, June 12-14, 2009, Shanghai, China, pp: 537-544.
- Wang, H., Z.J. Wu and S. Rahnamayan, 2011. Enhanced opposition-based differential evolution for solving high-dimensional continuous optimization problems. *Soft Comput.*, 15: 2127-2140.
- Xu, Q.Z., L. Wang, B.M. He and N. Wang, 2011a. Modified opposition-based differential evolution for function optimization. *J. Computat. Inf. Syst.*, 7: 1582-1591.
- Xu, Q.Z., L. Wang, B.M. He and N. Wang, 2011b. Opposition-based differential evolution using the current optimum for function optimization. *J. Applied Sci.*, 29: 308-315.
- Xu, Q.Z., N. Wang and R. Fei, 2013. Influence of dimensionality and population size on opposition-based differential evolution using the current optimum. *Inform. Technol. J.*, 12: 105-112.
- Zhang, C., Z.W. Ni, Z.J. Wu and L.C. Gu, 2009. A novel swarm model with quasi-oppositional particle. Proceedings of the International Forum on Information Technology and Applications, May 15-17, 2009, Chengdu, China, pp: 325-330.