Real-coded genetic algorithm with oriented evolution towards promising region for parameter optimization

Zhiqiang Chen[1*], Jiang Yun[1], Xudong Chen[2]

[1]*School of Computer Science and Information Engineering 125, Chongqing Technology and Business University, Chongqing, 400067, China*

[2]*Chongqing Engineering Laboratory for Detection Control and Integrated System, 19, Xuefu Avenue, Nan'an, Chongqing, 400067, China*

Received 1 October 2014, www.cmnt.lv

Abstract

In this paper, a novel real-coded genetic algorithm is presented to generate offspring towards a promising polygon field with k+1 vertex, which represents a set of promising points in the entire population at a particular generation. A set of 19 test problems available in the global parameter optimization literature is used to test the performance of the proposed real-coded genetic algorithms. Several performance comparisons with five significant real-coded genetic algorithms, three state-of-the-art differential evolution algorithms and three others significant evolutionary computing techniques are performed. The comparative study shows the proposed approach is statistically significantly better than or at least comparable to twelve significant evolutionary computing techniques over a test suite of 19 benchmark functions.

Keywords: genetic algorithm, real-coded, oriented evolution, parameter optimization

1 Introduction

Genetic algorithms (GAs), inspired by the natural evolution of species, have been successfully applied to solve numerous optimization problems in diverse fields. As powerful population-based stochastic search techniques, the popularity of GAs is based on simply solving multidimensional and multimodal optimization problems without requiring any additional information such as the gradient of an objective function. As one of the most intensively studies classes of global optimization over continuous spaces, in the recent past, a lot of effort has been put into the development of sophisticated recombination operators and framework of Real-coded GA (RCGA) for real parameter optimization [1-12].

Most of the population-based search algorithms try to balance between two contradictory aspects of their performance: exploration and exploitation. The first one means the ability of the algorithm to explore or search every region of the feasible search space, while the second denotes the ability to converge to the near-optimal solutions as quickly as possible [13,14]. Practical experience, however, shows that existing RCGAs still may occasionally stop proceeding toward the global optimum even though the population has not converged to a local optimum or any other point. Occasionally, even new individuals may enter the population, but the algorithm does not progress by finding any better solutions. In fact, it is impossible there is a general tractable algorithm that could efficiently and effectively solve all possible complexities of real-life optimization problems [14], which motivates researchers to develop better algorithms that yield better approximate solutions.

In the context, we develop a new real-coded genetic algorithm to attempt to make a balanced use of the exploration and exploitation abilities of the search mechanism and to be therefore more likely to avoid false or premature convergence in many cases. In the proposed algorithm, a promising polygon field with k+1 vertexes is defined, which represents a set of promising points in the entire population at a particular generation, and offspring is generated towards the centroid of the polygon field. We call the proposed RCGA as real-coded genetic algorithm with oriented evolution towards promising region (OEGA). In this paper, a set of 19 test problems available in the global optimization literature including unimodal, multimodal, parameter dependency, and ill-scale parameter optimization problems are used to evaluate the performance of OEGA. To further judge the performance of the proposed approach, several performance comparisons with five significant real-coded genetic algorithms [1,5,7,12], three state-of-the-art differential evolution algorithms [8,13,14] and three others significant evolutionary computing techniques are performed [4,16-18]. The comparative study shows the proposed approach is statistically significantly better than or at least comparable to several existing real-coded genetic algorithms as well as a few others significant evolutionary computing techniques over a test suite of 19 benchmark functions.

2 Real-coded genetic algorithm with oriented evolution towards promising region

2.1 RECOMBINATION OPERATOR

***Corresponding author's** e-mail: chenzhiqiangcq@163.com

There exist EAs like DE/target-to-best/1 which uses the best individual of the population to generate offspring. By "best" we mean the individual that corresponds to the best fitness in the entire population at a particular generation. The scheme promotes exploitation since all the genomes are attracted towards the same best position on the fitness landscape through iterations, thereby converging faster to that point. But as a result of such exploitative tendency, in many cases, the population may lose its global exploration abilities within a relatively small number of generations, thereafter getting trapped to some locally optimal point in the search space. A proper trade-off between exploration and exploitation is necessary for the efficient and effective operation of a population-based stochastic search technique. In the context, we propose that the centroid of some promising region replace the "best" individual in the entire population at a particular generation, and the genomes of individual are attracted towards a polygon region with k+1 vertexes representing promising point in search space. Based on above idea, a new recombination operator is designed as follows:

$$\begin{aligned} & X_c = X_P + \omega(X_g - X_p) + \xi \left| X_P - X_r \right| \\ & \omega = diag(\omega_1^t, \omega_2^t, ..., \omega_n^t) \\ & \omega_i^t \in u(0,7) \end{aligned} , \tag{1}$$

$$\xi = \begin{cases} -b\log_e(r), & r \le \dfrac{1}{2} \\ b\log_e(r), & r > \dfrac{1}{2} \end{cases} , \tag{2}$$

$$f(x) = \frac{1}{2b}\exp(-\frac{|x|}{b}) , \tag{3}$$

$$X_g = \frac{\sum_{i=1}^{k} X_{B,i} + X_O}{k+1} , \tag{4}$$

where X_P and X_C represents parent and offspring, respectively. $X_{B,i}$ is i^{th} best promising point. X_O is the champion found on the fitness landscape through iterations. X_r is an individual selected randomly from the mating pool $P_M = \{X_{M,1},..., X_{M,N}\}$. t is a parameter defined by user which decides the movement towards the promising search direction $X_g - X_P$. ξ is a random generated number using Laplace(0, b) distribution with probability density function as Equation(4).$k$express k promising points from the mating pool P_M .

As in Equation (1), the first and second items decide the exploring direction; the third item strengthens the exploiting ability of the algorithm following the direction.

The $X_{B,i}$ in Equation (4) is obtained using following subprocedure:

SubStep1: Createmating pool P_M as follows:

For ito N_m

Generate cluster C_i with C_s individuals selected randomly from the current population P_G ;

Assign the champion of cluster C_i to $X_{M,i}$

End

SubStep2:Sort the mating pool P_M ,k best individuals in the mating pool P_M are as $X_{B,i}$ (i=1,2,...,k), respectively.

2.2 GENERATION ALTERNATION MODEL

In order to further strengthen the exploring ability of the algorithm and increase the potential diversity of the population, we define a mutation behavior as follows: a solution is randomly selected from the population P_G and is mutated by given probability in each iteration. In this paper, we use MPTM (Makinen, periaux and toivanen mutation) mutation operator defined by Makinen et al. [11].

The detail of the proposed scheme is intertwined in the following manner:

Step 1. Set the generation number G=0 and randomly initialize a population P_G of N_P individuals, and each individual uniformly distributed in the range $[X_{Min}, X_{Max}]$.

Step 2. Create mating pool $P_M = \{X_{M,1},..., X_{M,N}\}$ and obtain Promising Points Set $X_{B,i}$ (i=1,2,...,k).

Step 3. Calculate the centroid X_g of the polygon formed by the promising points using Equation (3).

Step 4. Selectrandomly N_c individuals $(X_{p,1},..., X_{p,N_c})$ from the mating pool P_M and generate N_c offspring $X_c = \{X_{c,1},..., X_{c,N_c}\}$ usingEquation (1).

Step 5. Evaluate X_c .

Step 6. Replace randomly an individual of the population P_G using the best offspring X_{best} in the X_c in the fitness landscape.

Step 7. Mutate an individual selected randomly from the population P_G using MPTM mutation operator by probability r_M .

Step 8. G=G+1, repeat Step 2~Step 7 until the stopping criterion is not satisfied.

3Simulations

3.1 TEST BED

We used a test-bed of 19 traditional benchmark functions including unimodal, multimodal, parameter dependency, and ill-scale parameter optimization problems. The 19 traditional benchmarks have been reported in the following where *D* represents the number of dimensions *D*=25 to 100. They apparently belong to the difficult class of problems for many optimization algorithms.

1. Sphere function (f_1)

$$f(x)=\sum_{i=1}^{n} x_i^2,$$

$$-5.12 \le x_i \le 5.12,\ x^*=(0,0,...,0),\ f(x^*)=0.$$

2.Rosenbrock function (f_2)

$$f(x)=\sum_{i=2}^{n}(100(x_1-x_i^2)^2+(1-x_i)^2),$$

$$-2.048 \le x_i \le 2.048,\ x^*=(0,0,...,0),\ f(x^*)=0.$$

3.Schewefel problem 3 (f_3)

$$\min_x f(x)=\sum_{i=1}^{n}|x_i|+\prod_{i=1}^{n}|x_i|,$$

$$-10 \le x_i \le 10,\ x^*=(0,0,...,0)\ and\ f(x^*)=0.$$

4.Schewefel problem 4 (f_4)

$$\min_x f(x)=\max_x\{|x_i|, 1\le i\le n\},$$

$$-100 \le x_i \le 100,\ x^*=(0,0,...,0)\ and\ f(x^*)=0.$$

5. Ackley's problem (f_5)

$$\min_x f(x)=-20\exp(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n}x_i^2})$$

$$-\exp(\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i))+20+e,$$

$$-30 \le x_i \le 30,\ x^*=(0,0,...,0)\ and\ f(x^*)=0.$$

6. Griewank problem (f_6)

$$\min_x f(x)=1+\frac{1}{4000}\sum_{i=1}^{n}x_i^2-\prod_{i=1}^{n}\cos(\frac{x_i}{\sqrt{i}}),$$

$$-600 \le x_i \le 600,\ x^*=(0,0,...,0)\ and\ f(x^*)=0.$$

7.Rastrigin function (f_7)

$$f(x)=10n+\sum_{i=1}^{n}(x_i^2-10\cos(2\pi x_i)),$$

$$-5.12 \le x_i \le 5.12,\ x^*=(0,0,...,0),\ f(x^*)=0.$$

8. Generalized penalized function 1 (f_8)

$$\min_x f(x)=\frac{\pi}{n}(10\sin^2(\pi y_1)+$$

$$\sum_{i=1}^{n-1}(y_i-1)^2[1+10\sin^2(\pi y_{i+1})]+ \tag{9}$$

$$(y_n-1)^2)+\sum_{i=1}^{n}u(x_i,10,100,4),$$

$$-10 \le x_i \le 10,\ x^*=(0,0,...,0)\ and\ f(x^*)=0.$$

9. Generalized penalized function 2 (f_9)

$$\min_x f(x)=0.1(\sin^2(3\pi x_1)+$$

$$\sum_{i=1}^{n-1}(x_i-1)^2[1+\sin^2(3\pi x_{i+1})]+$$

$$(x_n-1)^2)[1+\sin^2(2\pi x_n)]+$$

$$\sum_{i=1}^{n}u(x_i,10,100,4),$$

$$-5 \le x_i \le 5,\ x^*=(0,0,...,0)\ and\ f(x^*)=0.$$

In the problem 17 and 18, the penalty function u is given by the following expression:

$$u(x,a,k,m)=\begin{cases} k * pow((x-a),m) & if\ x>a, \\ -k * pow((x-a),m) & if\ x<-a, \\ 0 & oyherwise\ . \end{cases}$$

10. Ellipsoid function (f_{10})

$$f(x)=\sum_{i=1}^{n}(1000^{\,i-1/n-1}x_i)^2,$$

$$-5.12 \le x_i \le 5.12,\ x^*=(0,0,...,0),\ f(x^*)=0.$$

11.*k*-tablet function (f_{11})

$$f(x)=\sum_{i=1}^{k}x_i^2+\sum_{i=k+1}^{n}(100x_i)^2,$$

$$-5.12 \le x_i \le 5.12,\ x^*=(0,0,...,0),\ f(x^*)=0.$$

12. Axis parallel hyper ellipsoid (f_{12})

$$\min_x f(x)=\sum_{i=1}^{n}ix_i^2,$$

$$-5.12 \le x_i \le 5.12,\ x^*=(0,0,...,0)\ and\ f(x^*)=0.$$

13.Zakharow's function (f_{13})

$$\min_x f(x)=\sum_{i=1}^{n}x_i^2+(\sum_{i=1}^{n}\frac{i}{2}x_i)^2+(\sum_{i=1}^{n}\frac{i}{2}x_i)^4,$$

$$-5.12 \le x_i \le 5.12,\ x^*=(0,0,...,0)\ and\ f(x^*)=0.$$

14. Exponential problem (f_{14})

$$\min_x \ f(x) = -\exp(\ 0.5\sum_{i=1}^{n} x_i^2),$$

$$-1 \le x_i \le 1,\ x^* = (0,0,...,0)\ and\quad f(x^*) = -1.$$

15. Ellipsoidal function (f_{15})

$$\min_x \ f(x) = \sum_{i=1}^{n} (x_i - i)^2,$$

$$-n \le x_i \le n,\ x^* = (1,2,...,\ n)\ and\quad f(x^*) = 0.$$

16. Cosine mixture problem (f_{16})

$$\min_x \ f(x) = \sum_{i=1}^{n} x_i^2 - 0.1\sum_{i=1}^{n} \cos(\ 5\pi x_i),$$

$$-1 \le x_i \le 1,\ x^* = (0,0,...,0)\ and\ \ f(x^*) = -0.1n.$$

17. Levy and Montalvo problem 1 (f_{17})

$$\min_x \ f(x) = \frac{\pi}{n}(10\sin^2(\pi y_1) +$$

$$\sum_{i=1}^{n-1} (y_i - 1)^2[1 + 10\sin^2(\pi y_{i+1})](y_n - 1)^2),$$

$$where\ \ y_i = 1 + \frac{1}{4}(x_i + 1),$$

$$-10 \le x_i \le 10,\ x^* = (0,0,\ldots,0)\ and\ \ f(x^*) = 0.$$

18.Bohachevsky function (f_{18})

$$f(x) =$$

$$\sum_{i=1}^{n-1} (x_i^2 + 2x_{i+1}^2 - 0.3\cos(3\pi x_i) - 0.4\cos(4\pi x_{i+1}) + 0.7),$$

$$-5.12 \le x_i \le 5.12,\ \ x^* = (0,0,\ldots,0),\ f(x^*) = 0.$$

19. Schaffer function (f_{19})

$$f(x) =$$

$$\sum_{i=1}^{n-1} [(x_i^2 + x_{i+1}^2)^{0.25} \times (\sin^2(50(x_i^2 + x_{i+1}^2)^{0.1}) + 1.0)],$$

$$-100 \le x_i \le 100,\ \ x^* = (0,0,\ldots,0),\ f(x^*) = 0.$$

3.2PARAMETERS SETTING

Finding the most appropriate combination of parameters occurring in an EA is termed as parameter tuning and is considered to be the most important and perhaps most difficult task. This difficulty also increases as we take larger and larger test suit into consideration because of multimodality and nonlinearity of different kind of objective functions. It becomes very challenging to suggest common fixed values of various parameters for the entire suit. To achieve this goal, we have carried out extensive experiments for the proposed OEGA approach. There exist eight important parameters in the proposed approach. Extensive experiments showed that the choice of numerical values for the three control parameters N_P, k and b highly depends on the problem under consideration. The varying of k is sensitive to the performance of the proposed approach and only when k is set to 2 or larger for the multimodal function, excellent results can be obtained. N_P is recommended to set as in Table 1. Other control parameters are set as follows:

C_s : the size of Cluster C_i is set 10 for f_2 and f_{12}, 15 for other problems.

N_M : the size of Mating pool P_M , 20 for all the problems.

N_C : the number of offspring generated in each generation, 2 for all the problems.

k: the number of Promising Points from the mating pool P_M , 1 for $f_2 \sim f_4$ and f_{12} ; 2 for f_1, $f_7 \sim f_{11}$ and f_{15} ; 3 for f_5 , f_6 , f_{14} , f_{16} , f_{18} and f_{19} ; 4 for f_{10} .

t: the parameter deciding the movement towards the promising search direction $X_g - X_P$, set 2.5 for all the problem.

b: the parameter of Laplace(0, *b*) distribution of ξ in Equation (1), 0.5 for $f_5 \sim f_7$, f_{16} , f_{17} and f_{19} , 0.1 for other problems.

r_M : The mutation ratio of each gene of the individual selected from the population P_G , 0.005 for all the problems.

3.3 EXPERIMENTAL EVALUTION OF OEGA

In this subsection, we perform a serial of experiments to evaluate the performance of the proposed approach. The study focuses on three important aspects of OEGA: 1) The speed of convergence measured in terms of the number of FEs required by an algorithm to reach a predefined threshold value of the objective function; 2) the frequency of hitting the optima (or success rate) measured in terms of the number of runs of an algorithm that converge to a threshold value within a predetermined number of FEs; 3) the issue of scalability, i.e., how the performance of an algorithm changes with the growth of the search-space dimensionality. The dimension of the variable in the entire problems is fixed to 30, 50 and 100, respectively. The number of FEs required by OEGA to reach two predefined thresholds (10^{-7} and 10^{-20}) are reported for all the problems with different dimensionality. A lower number of FEs corresponds to a faster algorithm. 25 independent runs are carried out for every problem. It is considered to be successful if a run achieves to reach the predefined threshold. A run is terminated before reaching the max number of function evaluations if the error value $f(x) - f(x^*)$ is less the given accuracy, where x^* is the global optimal solution. These experimental results are reported inTable 1.

TABLE 1 Number of FEs achieving the fixed accuracy level using OSGA

Func.	D	N_P	Threshold value=10^{-7}			Threshold value=10^{-20}		
			Least No.of FEs	MostNo.of FEs	Mean No.of	LeastNo.of FEs	MostNo.of FEs	MeanNo.of FEs
f1	30	300	13667	18414	15930	31658	35934	33821
	50	300	27132	33484	29963	58458	69304	63644
	100	300	74798	117690	85161	166282	196968	180516
f2	30	1500	95870	238682	157541	-	-	-
	50	2000	287140	472328	351757	-	-	-
	100	2000	1151432	2191700	1594944	-	-	-
f3	30	600	21584	42354	27405	56644	83610	64310
	50	900	59036	97336	75074	142442	181312	162638
	100	1500	307474	414642	358309	669412	764318	712000
f4	30	1500	306260	428714	358330	697338	876414	805848
	50	1500	845512	1042796	903468	3501782	4559988	3998598
f5	30	500	26174	45170	29816	-	-	-
	50	500	48272	76194	57879	-	-	-
	100	500	146430	212110	183115	-	-	-
f6	30	1200	35704	43056	37843	65480	79980	68213
	50	1500	68644	74314	71494	121812	129070	125080
	100	1500	151066	1791544	288506	250666	785878	303656
f7	30	900	84150	177304	123383	89918	205892	132924
	50	900	178824	285420	233555	176856	382173	278499
	100	900	357528	724248	494553	494403	857538	625887
f8	30	900	28770	45252	31091	67088	73516	70182
	50	900	51368	84354	57203	121776	165420	128353
	100	900	137224	173442	155648	308682	343656	328863
f9	30	900	29808	89276	36641	68240	126626	74496
	50	900	56844	334126	81253	124422	166828	133784
	100	900	152898	555424	187582	323362	1665574	405156
f10	30	300	17378	22128	19021	35386	41102	37144
	50	300	33530	42050	37270	65048	81990	71798
	100	300	92982	114422	104428	183222	229982	201363
f11	30	300	14768	18126	16404	31692	35582	34061
	50	300	27676	32724	30223	60282	72942	64657
	100	300	75770	99476	84809	166222	203192	181933
f12	30	300	13304	19458	15456	29570	38042	32332
	50	500	41568	56590	47634	88800	111102	97083
	100	1000	211436	304718	260608	485964	597608	522285
f13	30	900	196522	407390	293392	337262	495984	427207
	50	2400	1389726	2048508	1670813	2320800	3203268	2626230
f14	30	900	15124	17994	16616	42350	120072	71645
	50	900	26994	33302	29149	158456	1197380	410718
	100	900	64270	78006	69590	-	-	-
f15	30	300	13270	16296	14858	30638	36900	34305
	50	300	27304	37342	32058	65752	79014	70992
	100	300	91856	145118	106267	218330	588986	283138
f16	30	500	12380	15380	14031	30124	35282	32613
	50	500	23390	30458	25801	57440	66436	61418
	100	500	60710	92612	71600	165948	378112	191885
f17	30	1200	22996	42562	25465	68634	120078	76678
	50	1200	41258	136460	50324	126036	430552	161421
	100	1200	108438	946430	215241	335740	667854	413736
f18	30	900	24568	27414	26194	42934	49088	46047
	50	900	42000	80802	49459	74116	97638	77615
	100	900	98584	132908	112053	176982	943206	290975
f19	30	1500	167164	180278	173016	374636	384600	379528
	50	2000	442260	716374	480956	954182	1904664	1054012

As shown in Table 1, the proposed OEGA is enable to solve efficiently the function optimization with ill-scale, strong dependence among variables or multi-model. The proposed approach almost achieves the accuracy level 10^{-7} for all the tested problems except f_4 and f_{19} of D=100, 10^{-20} for all the tested problems except f_2 and f_5. Performance of most of the evolutionary algorithms deteriorates with the growth of the dimensionality of the search space. Increase of dimensions implies a rapid growth of the hyper volume of the search space and this in turn slows down the convergence speed of most of the global optimizers. Table 1 show the proposed approach can efficiently solve all the problem of D=100 except f_4 and f_{19}.

3.4 COMPARISIO WITH OTHER STATE-OF-THE-ART EVOLUTIONARY TECHNIQUES

In this subsection we compare the performance of OEGA with other State-of-the-Art Evolutionary Techniques. Genetic Algorithms (GA) are perhaps the most popular Evolutionary Algorithms for optimization problems in diverse fields. Five real-codedGAs (JGG+*REX*^{star}[1], *rc*-CGA+FDPP-LX [5], *rc*-CGA+BLX-α[7], MMG+BLX-α[7], SGA+LX-MPTM [12]) are employed to compare with the OEGA. JGG+JGG+*REX*^{star}, one of the excellent Real-coded GAs, can solve more efficiently real parameter optimization problems of multimodality, parameter dependency, and ill-scale. *rc*-CGA+FDPP-LX and *rc*-CGA+BLX-αare two efficient and effective Real-coded GAs for parameter optimization problems. MMG is a commonly-used steady-state model originally and BLX-αcrossover is a classical crossover operator for real-coded GA. MGG+BLX-αpresented excellent performance for parameter optimization of unimodality and multimodality. SGA+LX-MPTM is a simple genetic algorithm model with tournament selection and a crossover operator with Laplace distribution. We compare the number of FEs required by using algorithm to reach a predefined threshold value 10^{-7} of the objective function D=30. The parameter settings JGG+*REX*^{star}, *rc*-CGA+FDPP-LX, *rc*-CGA+BLX-α, MMG+BLX-α, SGA+LX are kept same as [5] and [7]. The initial population cover the entire search space as much as possible by uniformly randomizing individuals within the search space constrained by the prescribed minimum and maximum parameter bounds. Mean Number of FEs required reaching predefined threshold value 10^{-7} for 50 independent runs of each of the six contestant algorithms are presented in Table 2.

TABLE 2 Comparison with State-of-the-art Real-Coded GAs

Func.	**Threshold value**	**Mean Number of FEs required to reach predefined threshold value**				
		rc-CGA+BLX	MMG+BLX	JGG+*REX*^{star}	*rc*-CGA+FDPP-LX	This work
f1	10^{-7}	13667	56545	16159	12657	15930
f2	10^{-7}	-	-	90832	340453	157541
f3	10^{-7}	34765	96504	35674	48654	27405
f4	10^{-7}	101235	1794678	84745	250278	358330
f5	10^{-7}	38836	106564	32565	74654	29816
f6	10^{-7}	19943	124587	44165	44767	37843
f7	10^{-7}	227567	7606534	280342	240678	123383
f8	10^{-7}	42312	48565	14576	19256	31091
f9	10^{-7}	61732	51643	23454	76754	36641
f10	10^{-7}	16854	77143	25054	37554	19021
f11	10^{-7}	26012	77532	27012	41842	16404
f12	10^{-7}	19854	61967	17845	19956	15456
f13	10^{-7}	343245	215324	234124	147342	293392
f14	10^{-7}	11786	45268	14453	10832	16616
f15	10^{-7}	93012	71935	20546	25235	14858
f16	10^{-7}	14243	55012	25013	23532	14031
f17	10^{-7}	76732	98432	27021	55754	25465
f18	10^{-7}	27412	66443	39756	39754	26194
f19	10^{-7}	58120	383345	186431	590567	173016

As shown in bold in Table 2, the OEGA get 12 champion in term of the mean Number of FEs required reaching predefined threshold. A close inspection of Table 2 indicates that the performance of the proposed approach has remained clearly and consistently superior to that of the two classical real-coded GA schemes (MMG+BLX-αand SGA+LX-MPTM) as well as the other three state-of-the-art Real-code GAs.

Differential Evolution (DE) [8] is a simple yet powerful algorithm for real parameter optimization. To further evaluate the performance of the proposed approach, we also compare the proposed OEGA with three DE variants (DE/rand/bin [8], SADE [13], and DEGL/SAW [14]. Among the competitors, DE/rand/bin belongs to the classical DE family. SADE, and DEGL/SAW are state-of-the-art DE variants. Eight classical problems ($f_1 \sim f_8$) are selected as test suite to obtain a comparative performance with DE variants, which were also used to test the performance of the family of DEs in [14]. To make the comparison fair, we use the same method of initialization as in [14]: asymmetrical initialization reported in [15].

Asymmetrical initialization limits the initial process to just a portion of the feasible search space, which is a region defined to be half the distance from the maximum point along each axis back toward the origin. By contrast, symmetrical initialization is uniformly distributed about the entire search space.

The comparative study focuses on the quality of the final solutions produced by each algorithm and the speed of convergence measured in terms of the number of FEs required by an algorithm to reach a predefined threshold value of the objective function. To judge the accuracy of different approaches, we first let each of them run for a very long time over every benchmark function, until the number of FEs exceeds a given upper limit (which was fixed depending on the complexity of the problem). The mean and the standard deviation (within parentheses) of the best-of-run values for 50 independent runs of each of the five algorithms are presented in Table 3.

TABLE 3 Comparison with State-of-the-art DEs

Func.	D	Max FEs	Mean best value (Standard deviation)			
			DE/rand/bin	SADE	DEGL/SAW	This work
f1	25	5×10^5	6.85e-29(4.98e-23)	4.04e-35(3.91e-32)	8.78e-37(3.82e-35)	9.44e-321(0.00e+00)
	100	5×10^6	8.47e-24(4.66e-22)	5.84e-24(3.82e-23)	3.67e-25(4.73e-23)	1.10e-38(5.39e-38)
f2	25	5×10^5	9.83e-23(4.83e-24)	5.64e-26(9.36e-24)	6.89e-25(6.87e-21)	3.34e-23(5.99e-23)
	100	5×10^6	8.45e-05(2.74e-05)	8.64e-25(3.78e-24)	1.27e-15(7.72e-16)	8.65e-17(3.35e-16)
f3	25	5×10^5	7.54e-29(6.73e-29)	8.33e-26(4.83e-28)	4.93e-36(4.65e-32)	3.27e-206(0.00e+0)
	100	5×10^6	1.66e-09(6.77e-10)	2.65e-12(3.36e-14)	6.99e-14(1.34e-16)	2.93e-14(1.43e-17)
f4	25	5×10^5	8.36e-14(6.37e-13)	3.02e-14(1.37e-15)	4.99e-15(1.18e-14)	3.09e-53(8.76e-53)
	100	5×10^6	3.01e-12(3.26e-11)	3.70e-11(1.08e-13)	3.56e-14(4.55e-13)	4.14e-01(7.73e-02)
f5	25	5×10^5	4.19e-08(3.36e-08)	7.83e-15(2.85e-15)	5.98e-23(1.00e-22)	1.70e-14(5.92e-15)
	100	5×10^6	7.66e-05(6.76e-05)	3.06e-12(5.12e-13)	8.52e-17(1.36e-15)	3.24e-14(3.23e-19)
f6	25	5×10^5	6.83e-22(3.83e-25)	1.82e-28(7.68e-29)	2.99e-36(4.73e-35)	3.25e-03(4.76e-03)
	100	5×10^6	2.19e-10(8.45e-11)	8.95e-13(1.02e-14)	4.114e-15(6.02e-16)	7.39e-03(4.56e-03)
f7	25	5×10^5	1.04e-03(8.04e-02)	6.73e-24(3.72e-21)	5.84e-25(5.33e-27)	0.00e+00(0.00e+00)
	100	5×10^6	2.11e-02(4.86e-03)	5.88e-21(4.83e-20)	1.77e-22(3.88e-20)	2.68e-13(3.87e-13)
f8	25	5×10^5	7.09e-16(6.22e-15)	9.37e-24(6.19e-28)	7.20e-27(4.83e-28)	5.45e-32(1.40e-31)
	100	5×10^6	4.24e-10(2.96e-09)	2.84e-15(1.45e-14)	3.004e-18(4.82e-17)	8.53e-28(3.36e-27)

TABLE 4 Comparison with State-of-the-art DEs

Func.	D	Max FEs	DE/rand/bin		SADE		DEGL/SAW		This work	
			Successful runs	Mean no. of FEs	Successful runs	Mean no. of FEs	Successful runs	Mean no. of FEs	Successful runs	Mean no. of FEs
f1	25	10^{-20}	50	109372	50	104982	50	91935	50	20844
	100	10^{-20}	50	687322	50	738720	50	498521	50	148446
f2	25	10^{-20}	50	356253	50	267319	50	338279	50	256350
	100	10^{-7}	50	13398272	50	2844738	50	2709313	50	707487
f3	25	10^{-20}	50	266371	50	306742	50	157234	50	56258
	100	10^{-20}	13	2034583	23	1257362	34	978357	50	441702
f4	25	10^{-20}	16	376291	17	292478	21	294812	50	394534
	100	10^{-20}	14	3174782	17	3139382	25	2263976	0	-
f5	25	10^{-12}	14	226816	32	236290	50	224883	50	68485
	100	10^{-12}	13	1873625	13	1065920	27	925628	50	280587
f6	25	10^{-20}	19	345328	50	316382	50	196258	38	205630
	100	10^{-20}	5	1840322	34	1936287	43	1627092	34	2405721
f7	25	10^{-20}	19	345328	50	195823	50	87148	50	182844
	100	10^{-20}	5	1840322	50	744938	50	539282	24	7237873
f8	25	10^{-20}	35	294584	50	126574	50	150039	50	42385
	100	10^{-20}	8	3122658	20	1637409	27	1436190	50	200149

We report here results for 25 and 100 dimensions in Table 3. Table 3 shows that the OEGA get 8 champions in terms of the quality of the final solutions. In order to compare the speeds of different algorithms, we select a threshold value of the objective function for each benchmark problem. Tables 4 reports the number of runs (out of 50) that managed to find the optimum solution (within the given tolerance) as well as the mean number of FEs required by the algorithms to converge within the prescribed threshold value.

In addition, we compare the performance of the proposed approach with that of four state-of-the-art evolutionary and swarm-based optimization techniques, well-known as CPSO-H [16], IPOPCMA-ES [17], and G3 with PCX [4]. We employ the best parametric set-up for all these algorithms as prescribed in their respective sources. The mean and the standard deviation (within parentheses) of the best-of-run values of 50 independent runs for each algorithm have been presented in Table 5. We report only the hardest problem instances (multidimensional functions with D = 100) in Table 6.

TABLE 5 Comparison with Other State-of-the-art EAs (MaxNoumber. ofFEsis5 × 10^6)

Func.	D	CPSO-H		IPOP-CMAES		G3 withPCX		This work	
		Mean best value	Standard deviation	Mean best value	Standard deviation	Mean best value	Standard deviation	Mean best value	Standard deviation
f1		6.56e-22	7.23e-28	9.68e-23	7.23e-26	2.80e-20	6.46e-14	1.10e-38	4.73e-23
f2		1.50e-01	9.42e-01	6.04e-22	8.34e-24	5.77e-18	2.23e-19	8.65e-17	7.30e-22
f3		7.41e-08	6.22e-07	2.74e-03	1.64e-07	2.65e-06	3.36e-10	2.93e-14	1.43e-17
f4		6.51e-13	1.79e-16	1. 76e-12	4.94e-06	7.48e-13	3.77e-09	4.14e-01	7.73e-02
f5		1.77e-12	1.79e-16	8.85e-17	4.94e-06	3.47e-10	7.14e-09	3.24e-14	3.23e-19
f6		2.53e-02	7.22e-03	3.67e-14	6.93e-14	8.92e-11	8.15e-15	7.39e-03	4.56e-03
f7		1.73e-01	4.09e-02	9.24e-21	4.32e-21	5.92e-03	3.77e-04	2.68e-13	3.87e-13
f8		4.20e-10	6.95e-11	4.45e-19	3.63e-16	6.86e-04	8.03e-03	8.53e-28	3.36e-27

A close inspection of Tables3, 4, and 5 indicate that the performance of the proposed approach has strong international competitiveness with other state-of-the-art EAs.

4 Conclusions

In this paper, we proposed a new framework of RCGA and a recombination operator that generates new solution towards a polygon field with k+1 vertexes representing promising points in search space, which is an attempt to balance explorative and an exploitive ability. An extensive performance comparison with five significant RCGAs variants, three state-of-the-art differential evolution algorithms and others three significant evolutionary computing techniques indicated that the proposed approaches enhance RCGA ability to accurately locate solutions in the search space. The empirical study showed the proposed OEGA is an efficiently scheme for solving parameter optimization problem. This, however, does not lead us to claim that the OEGA may outperform their contestants over every possible objective function since it is impossible to model all possible complexities of real-life optimization problems with the limited test-suite that we used for testing the algorithms.

Acknowledgments

This work was supported by Natural Science Foundation Project of CQ CSTC(No.cstc2012jjA40041, No.cstc201-jjA40059). Science Research Fund of Chongqing Technology and Busines sUniversity (No. 2011-56-05, No. 2013-56-01)

References

[1] KobayashiS2009 The frontiers of real-coded genetic algorithms*Journal of Japanese Society for Artificial Intelligence(in Japanese)***24**(1) 128-43

[2] Tsutsui S, Yamamura M, Higuchi T 1999 Multi-parent recombination with simplex crossover in real-coded genetic algorithms*Proceedingsof the Genetic and Evolutionary Computation Conference* 657-64

[3] Ono I, Kobayashi S1997 A real-coded genetic algorithm for function optimization using unimodal normal distribution crossover*Proceedings of the Seventh International Conference on Genetic Algorithms* 246-53

[4] Deb K, Anand A, Joshi D 2002 Computationally efficient evolutionary algorithm for real-parameter evolution*Evolutionary Computation Journal* 10(2)

[5] Chen Z-Q, WangR-L 2011 A new framework with FDPP-LX crossover for real-coded genetic algorithm*IEICE Transactions on Fundamentals of Electronics Communications and Computer Sciences 6(E94.A)* 1417-25

[6] EshelmanLJ, SchafferJD 1993 Real-coded genetic algorithms and interval schemata: *Foundation of Genetic Algorithms IIMorgan Kaufmann San MateoDL Whitley ed*187-202

[7] Chen Z-Q, WangR-L 2011Two efficient real-Coded genetic algorithms for real parameter optimization*International Journal of Innovative Computing, Information and Control***7**(8) 4871-83

[8] StornR, Price V, LampinenJ 2005Differential evolution: a Practical approach to global optimization *Berlin Germany: Springer-Verlag*

[9] Ono S, HirotaniY,Nakayama S 2009 A memetic algorithm for robust optimal solution search - hybridization of multi-objective genetic algorithm and quasi-newton method*InternationalJournal of Innovative Computing Information and Control***5**(12) 5011-20

[10] Martin D, del Toro R, Haber R,DorronsoroJ 2009 Optimal tuning of a networked linear controller using a multi-objective genetic algorithm and its Application to one complex electromechanical process*International Journal of Innovative Computing Information and Control***5**(10(B)) 3405-14

[11] MakinenRAE, PeriauxJ, ToivanenJ 1999 Multidisciplinary shape optimization in aerodynamics and electromagnetic using geneticalgorithms*International Journal for Numerical Methods in Fluids***30**(2) 149-59

[12] Deep K, ThakurM 2007 A new crossover operator for real coded genetic algorithms*Applied Mathematics and Computation* 88 895-911

[13] Qin A K, Huang V L, SuganthanP N 2009*IEEE Transactions on Evolutionary Computations***13**(2) 298-417

[14] Das S, AbrahamA 2009*IEEE Transactions on Evolutionary Computation***13**(3) 522-53

[15] AngelineP J 1998 Evolutionary optimization versus particle swarm optimization: Philosophy and the performance difference *In Proc. 7th Int. Conf. Evolutionary Programming-Evolutionary Programming VII*1447 84-9

[16] Van den Bergh F,EngelbrechtA P 2004 *IEEE Trans EvolComput***8**(3) 225-39

[17] Auger A,HansenN 2005 A restart CMA evolution strategy with increasing population size *IEEECongress on Evolutionary Computation*2 1769-76

Authors

Zhiqiang Chen, 11.1975, ChongQing, China

Current position, grades: associate professor in School of Computer Science and Information Engineering, Chongqing Technology and Business University.
University studies: PhD degree from Fukui University, Japan in 2011.
Scientific interests: computational intelligence and its application.

Jiang Yun, 1983, Hubei, China

Current position, grades: lecturer in School of Computer Science and Information Engineering, Chongqing Technology and Business University.
University studies: PhD degree from Huazhong University of Science and Technology, China in 2011.
Scientific interests: membrane computing and evolutionary algorithm.

Xudong Chen, 1976, Sichuan, China

Current position, grades: lecturer in School of Computer Science and Information Engineering, Chongqing Technology and Business University.
University studies: PhD degree from University of Electronic Science and Technology of China in 2007.
Scientific interests: real-time computing, image processing, and optimization problems.