Real-Coded Chemical Reaction Optimization with Different Perturbation Functions

James J.Q. Yu, *Student Member, IEEE*
Department of Electrical and
Electronic Engineering
The University of Hong Kong
Email: jqyu@eee.hku.hk

Albert Y.S. Lam, *Member, IEEE*
Department of Electrical Engineering
and Computer Sciences
University of California, Berkeley
Email: albertlam@ieee.org

Victor O.K. Li, *Fellow, IEEE*
Department of Electrical and
Electronic Engineering
The University of Hong Kong
Email: vli@eee.hku.hk

***Abstract*—Chemical Reaction Optimization (CRO) is a powerful metaheuristic which mimics the interactions of molecules in chemical reactions to search for the global optimum. The perturbation function greatly influences the performance of CRO on solving different continuous problems. In this paper, we study four different probability distributions, namely, the Gaussian distribution, the Cauchy distribution, the exponential distribution, and a modified Rayleigh distribution, for the perturbation function of CRO. Different distributions have different impacts on the solutions. The distributions are tested by a set of well-known benchmark functions and simulation results show that problems with different characteristics have different preference on the distribution function. Our study gives guidelines to design CRO for different types of optimization problems.**

***Index Terms*—Chemical Reaction Optimization, Gaussian distribution, Cauchy distribution, exponential distribution, Rayleigh distribution, evolutionary algorithm.**

I. Introduction

Chemical Reaction Optimization (CRO) [1] is a simple and powerful metaheuristic optimization method which mimics the interactions of molecules in chemical reactions to search for the global optimum. CRO was designed as an optimization framework and it was initially targeted to solve discrete optimization problems [2][3][4]. It has been applied to solve many practical problems, e.g. population transition problem in peer-to-peer streaming [5], network coding optimization problem [6], etc. Lam *et al.* then proposed a variant of CRO in 2011, named Real-Coded Chemical Reaction Optimization (RCCRO) [7], to solve continuous optimization problems. RCCRO utilizes the Gaussian distribution function as the perturbation function and some real-coded-based mechanisms were designed to implement RCCRO. RCCRO has been shown to be efficient in solving continuous optimization problems [7][8].

There are four major operations (i.e., elementary reactions) in CRO: on-wall ineffective collision, decomposition, inter-molecular ineffective collision, and synthesis. In CRO, on-wall ineffective collision and inter-molecular ineffective collision correspond to local search, while decomposition and synthesis correspond to remote search. In the conventional RCCRO [7], the Gaussian distribution is deployed in the neighborhood search operator in all reactions except synthesis. In these elementary reactions, with the Gaussian distribution, the molecules are perturbed via the Gaussian mutation and the energy state of the molecules are checked to decide whether the reaction shall be accepted or rejected. The Gaussian perturbation in RCCRO is accomplished by adding a zero-mean Gaussian-random number to the existing molecular structures (i.e., solutions) to generate new solutions in the neighborhoods.

For other optimization methods like Evolutionary Programming (EP) and Particle Swarm Optimization (PSO), researchers have also proposed other perturbation functions other than the Gaussian distribution. Mutation operations in EP based on the Cauchy distribution [9] and the Levy distribution [10] have been propose. Krohling *et al.* has made contributions in integrating PSO with the exponential [11], the Gaussian, and the Cauchy distribution [12]. However, the impacts of different perturbation distributions on the performance of RCCRO need further study. Such research is necessary for a better understanding of the performance of RCCRO.

In this paper, we propose to apply four different probability distribution functions as the perturbation function for RCCRO, namely, the Gaussian distribution, the Cauchy distribution, the exponential distribution, and a modified Rayleigh distribution. These distributions have drawn much attention from different research communities [10][13]. These four distributions are tested on a suite of well-known benchmark functions classified into three categories of optimization functions. The simulation results demonstrate that different categories of functions have different preferences on the perturbation distribution function.

The rest of the paper is organized as follows: Section II presents an overview of CRO. In Section III, the four tested distribution functions are described. Section IV provides the simulation results. The paper is concluded in Section V with suggestions for further research.

II. Chemical Reaction Optimization

In this section we will first introduce the manipulated agents of CRO, i.e., molecules. Then the definitions of elementary reactions are presented. The section ends with the overall algorithm of RCCRO (or simply referred to CRO hereafter).

A. Molecules

CRO is a kind of population-based metaheuristics [1]. There is a population of molecules in a container, with an energy buffer attached. Each molecule is characterized by its molecular structure (ω), potential energy (PE), kinetic energy (KE), and some other attributes. ω stands for a feasible solution of the optimization problem, corresponding to this molecule. PE represents the the objective function value of ω while KE represents the tolerance of the molecule to hold a worse solution with larger objective function value than the existing one. Other attributes can be used to control the flow of CRO to ensure CRO meet the characteristics of the optimization problem. Users can add, change, or remove the optimal attributes to build different versions of CRO.

B. Elementary Reactions

In CRO, changes to the molecules are made through different elementary reactions. There are four kinds of elementary reactions, namely, on-wall ineffective collision, decomposition, inter-molecular ineffective collision, and synthesis. The former two reactions belong to the uni-molecular reactions and the latter two are classified as class of inter-molecular reactions. These elementary reactions change the molecular structures of the molecules and accept molecules with new structures according to the conservation of energy. CRO makes use of these changes to explore the solution space and locates the global optimum [1]. We basically follow [7] to define the elementary reactions and they are described briefly as follows:

1) On-wall Ineffective Collision: An on-wall ineffective collision happens when a molecule collides with a wall of the container and then bounces away. This reaction has one molecule as input and returns another modified molecule. This reaction is mainly used for performing local search, thus the changes made on the molecular structure shall be small. We commonly generate a neighborhood structure ω' from ω. If we define the neighbor function as

$$\textit{neighbor}(\omega) = \omega + \epsilon, \tag{1}$$

where ϵ is called the perturbation factor and is generated from a pre-defined probability distribution function, which will be elaborated in Section IV-E. We obtain a new solution ω' by

$$\omega' = \textit{neighbor}(\omega) \tag{2}$$

and its new PE is fiven by

$$PE_{\omega'} = f(\omega'), \tag{3}$$

where f is the objective function.

During this procedure, a part of the KE held by the molecule will be transfered to the system energy buffer (*EnBuff*). We have a parameter *LossRate* to control this energy loss process. The new KE of the molecule will be updated according to

$$KE_{\omega'} = (PE_{\omega} - PE_{\omega'} + KE_{\omega}) \times t, \tag{4}$$

where $t \in [LossRate, 1]$ is a randomly generated number.

2) Decomposition: A decomposition happens when a molecule collides with a wall of the container and breaks into two molecules. This reaction is mainly used for jumping out of the local minimums, and thus the changes made on the molecular structures are larger than an on-wall ineffective collision. During this procedure, there is no KE transferred to *EnBuff*, but the energy conservation law shall hold [7].

3) Inter-molecular Ineffective Collision: An inter-molecular ineffective collision happens when two molecules collide with each other and then separate. The purpose and characteristics of this reaction is similar with an on-wall ineffective collision. In general, we perform two neighbor searches on the two molecular structures ω_1 and ω_2, i.e.,

$$\omega_1' = neighbor(\omega_1) \text{ and } \omega_2' = neighbor(\omega_2). \tag{5}$$

4) Synthesis: A synthesis happens when two molecules collide and merge into one. Generally, the change is severe and can help the molecule jump out of local minimums. Its objective is to maintain the population diversity.

In this operator, the new molecule is derived from the two given original molecules and each element of the solution is equally likely to be selected from each of the original molecules at the same position.

C. The Overall Algorithm

CRO operates in a closed container with an initial population of randomly generated molecules. The algorithm contains three phases: initialization, iterations, and finalization. When CRO starts, the molecules as well as some system parameters, such as *EnBuff*, *CollRate*, *LossRate*, *DecThres*, and *SynThres* [7][17] are set. In each iteration, the system will first randomly select one reaction type according to some criteria. The system will then use this decision to randomly select one or multiple molecules from the existing ones in the container depending on whether it is a uni-molecular or an inter-molecular reaction and check its/their energy state(s). If the decomposition criterion (for uni-molecular collision) or the synthesis criterion (for inter-molecular collision) described in [1] is satisfied, the corresponding reaction takes place. Otherwise an on-wall ineffective collision or an inter-molecular ineffective collision will take place. At the end of each iteration is the energy check. The newly generated or transformed molecule(s) have their objective functions evaluated and compared with the original molecules. If the new value(s) can satisfy the energy conservation conditions [1], the new molecules are accepted and put into the container while the original molecules are discarded. Otherwise, the new molecules are discarded and this indicates a failed reaction. After the pre-defined certain number of function evaluations is reached or one of some specific stopping criteria is met, the algorithm proceeds to finalization and outputs the best-so-far global optimum. Since in this problem we focus on studying the performance of CRO with different probability distributions used in the neighborhood search operator and decomposition stated in Section II-B, interested readers can refer to [1] and [7] for detailed description of the algorithm and its pseudocode.

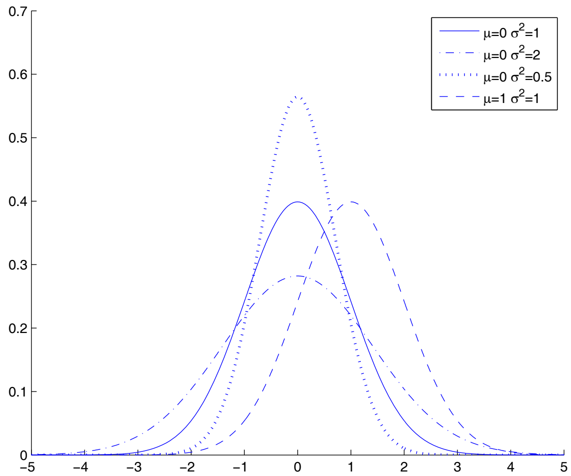

Fig. 1. PDF of Gaussian Distribution

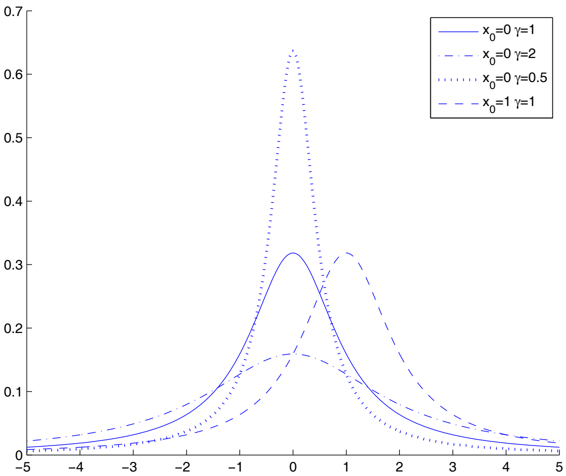

Fig. 2. PDF of Cauchy Distribution

III. Probability Distribution

In this paper, we consider four different types of probability distributions, namely, the Gaussian distribution, the Cauchy distribution, the exponential distribution, and the Rayleigh distribution. In this section, we will first introduce the four distributions. Then the integration of CRO with these distributions is presented.

A. Gaussian Distribution

The Gaussian distribution has a bell-shaped probability density function (PDF) and is also known as the normal distribution or informally the bell curve [14]. Its density function is given as:

$$f(x;\mu,\sigma)=\frac{1}{\sqrt{2\pi\sigma^2}}e^{-\frac{(x-\mu)^2}{2\sigma^2}}, \tag{6}$$

where μ is the expectation and σ^2 is the variance. The plots of the PDF with some different μ and σ^2 are given in Fig. 1.

We can control the shape by modifying the value of σ^2. A larger σ^2 will result in a flatter bell.

B. Cauchy Distribution

The Cauchy distribution shares a similar bell shape with the Gaussian distribution and it has important applications to physics. We can utilize its characteristics to perform detailed local search. Its density function is given as:

$$f(x;x_0,\gamma)=\frac{1}{\pi}[\frac{\gamma}{(x-x_0)^2+\gamma^2}]. \tag{7}$$

In this PDF, x_0 is the location parameter, or mean of the distribution. It is similar to μ in the Gaussian distribution. γ is the scale parameter which can specify the half-width at half-maximum (HWHM). The plots of the Cauchy distribution with different x_0 and γ are given in Fig. 2.

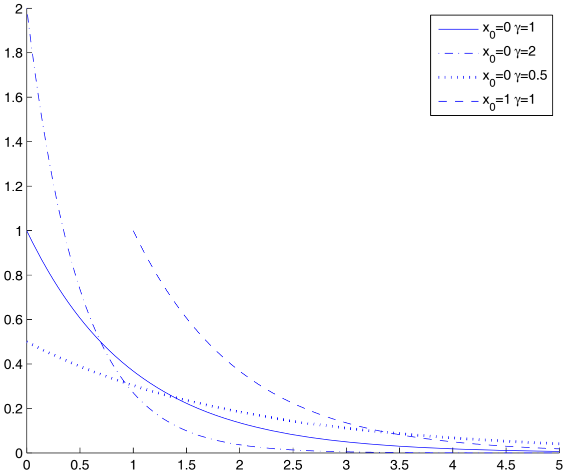

Fig. 3. PDF of Exponential Distribution

C. Exponential Distribution

The exponential distribution describes the time between events in a Poisson process. Different from the previous two distributions, the exponential distribution does not have a bell-shape. Its density function is given as:

$$f(x;x_0,\gamma)=\begin{cases}\gamma e^{-\gamma(x-x_0)}, & x\geq 0\\ 0, & x<0\end{cases} \tag{8}$$

The parameter x_0 defines the starting point of the distribution and the parameter γ defines the steepness of the PDF curve. The plots of the distribution with different x_0 and γ are given in Fig. 3.

Note that the normal exponential distribution only has a positive side, which does not satisfy our requirement as a perturbation function. Similar to [11], we mirror this curve

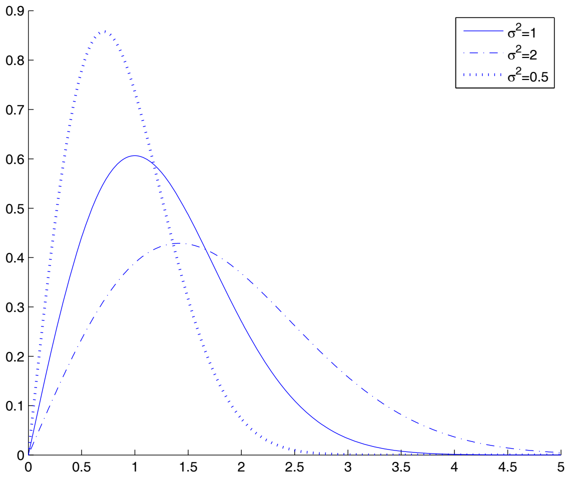

Fig. 4. PDF of Rayleigh Distribution

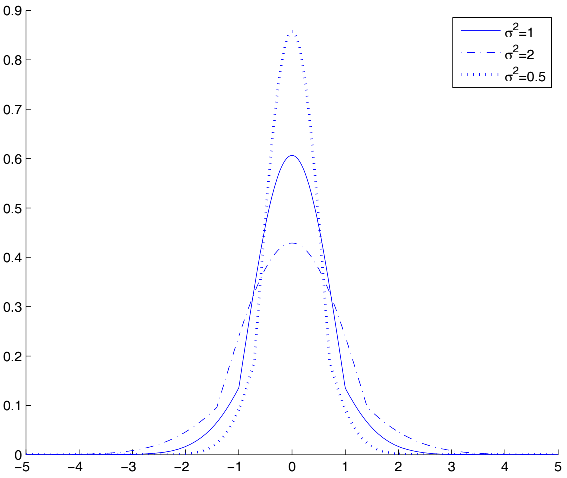

Fig. 5. PDF of Modified Rayleigh Distribution

to the negative side of the x-axis to make this distribution a continuous function on the range $(-\infty, \infty)$.

D. Rayleigh Distribution

The Rayleigh PDF is given below:

$$f(x;\sigma^2) = \frac{x}{\sigma^2} e^{-\frac{x^2}{2\sigma^2}}, x \geq 0. \tag{9}$$

As with the Gaussian distribution, the only parameter, σ^2 controls the flatness of the shape. A larger σ^2 results in a flatter PDF. Moreover, the highest point of the curve occurs at $x = \sigma$. The plots of the Rayleigh distribution with different σ^2 are given in Fig. 4.

Like the exponential distribution, the Rayleigh distribution is only continuous on $(0, \infty)$. So we introduce a new way to make the distribution function continuous on $(-\infty, \infty)$ and apply it to the perturbation function. The modified PDF is given below:

$$\begin{aligned} f'(x;\sigma^2) = & [\frac{\sigma + x}{\sigma^2} e^{-\frac{(\sigma+x)^2}{2\sigma^2}} \times step(x, -\sigma) + \\ & \frac{\sigma - x}{\sigma^2} e^{-\frac{(\sigma-x)^2}{2\sigma^2}} \times (1 - step(x, \sigma))]/2 \end{aligned} \tag{10}$$

where

$$step(x, \sigma) = \begin{cases} 1, & x \geq \sigma \\ 0, & x < \sigma \end{cases} \tag{11}$$

This modified PDF is composed of two parts: We first shift the original PDF to the left by σ units, which makes $x = \sigma$ as the y-axis. Then this new curve is copied and mirrored at the y-axis. The two curves are then summed and averaged. The plots of this modified PDF with different σ^2 values are given in Fig. 5.

E. Integration of Distributions with CRO

The perturbation function mainly operates in the on-wall ineffective collision, inter-molecular ineffective collision, and decomposition [7]. In the above three elementary reactions, new molecular structures are generated from the original structures. Although in synthesis there is also a new structure generated, the new structure is composed of different parts of the original structures. To generate new molecular structures, one can add a perturbation factor to one random element of the original structure for a small change (on-wall ineffective collision and inter-molecular ineffective collision), or add different perturbation factors to some of the elements for a dramatic change (decomposition) [7]. The perturbation factors are generated from the perturbation function, which can be one of the pre-defined probability distributions discussed before. So different probability distribution functions shall generate perturbation factors with different characteristics, and this will result in different performances of the algorithm.

IV. Simulation Results

A. Benchmark Functions

In order to evaluate the performance of different perturbation distribution functions, we perform simulations on the standard benchmark functions used in [15]. The benchmark functions are listed in [7] with the dimension sizes, feasible solution spaces and the known global optimums. This set of functions has been widely used or partially used as metaheuristic performance evaluation benchmark functions [7][15][16].

This benchmark set can be divided into three categories according to their characteristics. The first group consists of unimodal functions, each of which has only one global optimum. So it is relatively easy to solve. The second group contains high-dimensional multimodal functions. Functions in this group have multiple local optimums and it is more likely for algorithms to get “stuck” in the local optimums.

TABLE I
PARAMETER SETTINGS

Category	PopSize	StepSize	EnBuff	IniKE	CollRate	LossRate	DecThres	SynThres
I	10	0.1	10^6	10^3	0.2	0.9	1.5^5	0
II	20	1^*	10^5	10^7	0.2	0.1	1.5^5	10
III	100	0.5	0	10^3	0.2	0.1	500	10

* 300 for f_8 and 15 for f_{11}

The last group is a collection of low-dimensional multimodal functions. These functions have lower dimensions and fewer local optimums than the second group.

B. Experiment Environment

All the simulations are performed on the same computer with Intel Core i5-2400 @ 3.1GHz CPU and 4.00GB RAM. CRO and distribution functions are implemented using C++ and compiled with MingGW g++ compiler under the Windows 7 64bit environment.

C. Parameter Selection

Different parameter settings can affect the performance of CRO dramatically [7]. A suitable combination of parameters, including *PopSize*, *StepSize*, *EnBuff*, *IniKE*, *CollRate*, *LossRate*, *DecThres*, and *SynThres* [17], may result in good simulation results. Since the combinations of parameters exist in an eight-dimensional space and are continuous, it is impractical to test all possible combinations. Instead, the parameters are tuned in an ad hoc manner. In our simulations presented in the next subsection, we adopt the three different parameter combinations for the three function categories presented and discussed in [7]. The details of the combinations are listed in Table I and interested readers can refer to [7] and [17] for elaboration of the functionality of each parameter.

D. Comparisons among Different Distributions

In this paper we presented four different distribution functions as CRO's perturbation function, namely the Gaussian distribution, the Cauchy distribution, the exponential distribution, and a modified Rayleigh distribution. For simplicity, we denote the CRO algorithms with distributions as CRO_G, CRO_C, CRO_E, and CRO_R, respectively. We evaluate the performance of the four CRO variants with the benchmark functions with the parameter combinations discussed in the previous subsection. We repeat the simulations for each function 100 times and the function evaluation limits for different functions are listed in Table II which are the same as those used in [7].

The simulation results for the four algorithms on the 23 benchmark functions are presented in Table III, Table IV, and Table V, and the best results are bolded. Since the three categories of benchmark functions have different characteristics on the solution space, we will discuss them separately.

From these tables we can see the Cauchy distribution performs best in Category I as its averaged rank is the highest. Recall that Category I is uni-modal and thus algorithms performing local search are good enough for these functions. Comparing Figs. 1, 2, 3, and 5, the Cauchy distribution has the most of its probabilities around the mean ($x = 0$), which helps CRO perform local search more efficiently for uni-modal problems.

TABLE II
FUNCTION EVALUATION LIMITS

Function	FELimit	Function	FELimit
f_1	150 000	f_{13}	150 000
f_2	150 000	f_{14}	7 500
f_3	250 000	f_{15}	250 000
f_4	150 000	f_{16}	1 250
f_5	150 000	f_{17}	5 000
f_6	150 000	f_{18}	10 000
f_7	150 000	f_{19}	4 000
f_8	150 000	f_{20}	7 500
f_9	250 000	f_{21}	10 000
f_{10}	150 000	f_{22}	10 000
f_{11}	150 000	f_{23}	10 000
f_{12}	150 000		

For Category II, the exponential distribution and the Rayleigh distribution perform equally but their advantage over the Gaussian and the Cauchy distributions is not very significant. Since the solution space of problems in Category II is high dimensional with a large number of local optimums, it is reasonable that different distributions will give a best performance on different problems. For problems with relatively less local optimums, the Gaussian and the Cauchy distribution can perform better because these problems are similar with problems in Category I. However, for problems with a large number of local optimums, a flatter distribution can probably perform well since it can maintain the population diversity better. So for this category of problems different distributions can perform well on different problems and there is no significant preference.

The Gaussian distribution performs best in Category III but the Cauchy distribution also performs relatively well in this category. A possible reason to this phenomena is that since the difference between Category II and III is that Category III has lower dimensions, which reduce the number of potential local optimums, the exponential and the Rayleigh distributions lose their advantage on maintaining population diversity. However, although the number of local optimums is smaller, it is still easy for the Cauchy distribution to get stuck due to the shape of its PDF. The Gaussian distribution with shallower bell shape can maintain a better population diversity than the Cauchy

TABLE III
SIMULATION RESULTS AND COMPARISONS FOR CATEGORY I

Function	CRO_G				CRO_C			
	Mean	Std. Div.	Best	Rank	Mean	Std. Div.	Best	Rank
f_1	**2.8023E-06**	9.5462E-06	1.1729E-06	1	3.6134E-06	1.0076E-05	1.5009E-06	2
f_2	**5.2742E-03**	8.6993E-03	3.1498E-03	1	6.5312E-03	1.0253E-02	4.2567E-03	2
f_3	**4.0448E-07**	1.7591E-06	1.3886E-07	1	5.9970E-07	2.2993E-06	1.8863E-07	2
f_4	1.5898E+00	4.9408E+01	3.7482E-03	3	**2.1603E-02**	3.2341E-01	3.4985E-03	1
f_5	7.9995E+01	1.3336E+03	1.7200E-02	4	**4.9454E+01**	3.5096E+02	6.5425E-01	1
f_6	**0.0000E+00**	0.0000E+00	0.0000E+00	1	**0.0000E+00**	0.0000E+0	0.0000E+00	1
f_7	1.0101E-02	6.0951E-02	2.6305E-03	2	**9.0633E-03**	4.4733E-02	1.9008E-03	1
Avg.				1.86				**1.43**
Function	CRO_E				CRO_R			
	Mean	Std. Div.	Best	Rank	Mean	Std. Div.	Best	Rank
f_1	2.3209E-05	8.6462E-05	8.9015E-06	4	5.8125E-06	1.8736E-05	2.2348E-06	3
f_2	1.6889E-02	2.6210E-02	1.0355E-02	4	7.7805E-03	1.0789E-02	4.7191E-03	3
f_3	3.7256E-06	1.4036E-05	1.0769E-06	3	8.7830E-07	3.4124E-06	3.6419E-07	4
f_4	4.6047E-02	6.6140E-01	1.0012E-02	2	2.4322E+00	7.1027E+01	4.4672E-03	4
f_5	6.2050E+01	3.0865E+02	1.2512E+00	2	7.3270E+01	9.0927E+02	2.2321E+00	3
f_6	**0.0000E+00**	0.0000E+00	0.0000E+00	1	**0.0000E+00**	0.0000E+0	0.0000E+00	1
f_7	1.3602E-02	7.1639E-02	4.5453E-03	4	1.0531E-02	4.6002E-02	4.1566E-03	3
Avg.				2.86				3.00

TABLE IV
SIMULATION RESULTS AND COMPARISONS FOR CATEGORY II

Function	CRO_G				CRO_C			
	Mean	Std. Div.	Best	Rank	Mean	Std. Div.	Best	Rank
f_8	-6.6941E+03	7.0479E+03	-8.8963E+03	3	**-1.1584E+04**	2.7988E+03	-1.2451E+04	1
f_9	7.3239E-04	5.1246E-03	2.8899E-04	3	9.6594E-04	2.9687E-03	4.8051E-04	4
f_{10}	2.3286E-03	7.6100E-03	1.6035E-03	3	3.0142E-03	1.1784E-02	1.6159E-03	4
f_{11}	6.3738E+00	4.4097E+01	3.9130E-01	3	**6.9692E-02**	5.0228E-01	6.6663E-07	1
f_{12}	8.9318E-02	1.7332E+00	4.7978E-08	3	2.2222E-07	1.6140E-06	6.2396E-08	2
f_{13}	2.5472E-06	3.4365E-05	5.7443E-07	3	3.1977E-06	3.2292E-05	5.6288E-07	4
Avg.				3.00				2.67
Function	CRO_E				CRO_R			
	Mean	Std. Div.	Best	Rank	Mean	Std. Div.	Best	Rank
f_8	-6.6816E+03	6.8112E+03	-8.6168E+03	4	-6.7458E+03	6.7677E+03	-8.1446E+03	2
f_9	4.0876E-04	1.5659E-03	1.2755E-04	2	**3.4566E-04**	4.6215E-03	1.3204E-04	1
f_{10}	1.8143E-03	2.7791E-03	1.1840E-03	2	**1.6536E-03**	5.5241E-03	1.1178E-03	1
f_{11}	4.8161E+00	2.6975E+01	6.3374E-01	2	8.7394E+00	5.1116E+01	6.9625E-01	4
f_{12}	**1.0936E-07**	9.0059E-07	3.1753E-08	1	2.1104E+00	2.6246E+01	3.1908E-08	4
f_{13}	1.4290E-06	1.3039E-05	3.9950E-07	2	**1.0141E-06**	1.0198E-05	2.2801E-07	1
Avg.				**2.17**				**2.17**

while performing local search more efficiently than the other two distributions.

The average computational time consumed in a simulation run of these four CRO variants are listed in Table VI for reference. Generally the computational speed of the Gaussian distribution is the fastest, but all computational times are comparable. To summarize, the Cauchy distribution is suitable for solving uni-modal optimization problems. The Gaussian distribution is suitable for solving low-dimensional multi-modal problem. For high-dimensional multi-modal problems, the exponential and the Rayleigh distribution generally perform better, but for different problems there are different preferences.

V. CONCLUSION AND FUTURE WORK

In this paper, we compare four kinds of distribution functions, namely the Gaussian distribution, the Cauchy distribution, the exponential distribution, and the modified Rayleigh distribution, as the perturbation function for CRO. Since these distributions have different characteristics, they may be suitable to solve different kinds of problems with different solution space characteristics. We integrate these four distribution functions into CRO. The four CRO variants are evaluated with 23 benchmark functions, divided into three categories. The simulation results show that different categories of problems have different preference on the perturbation function. The Cauchy distribution fits Category I functions best and the Gaussian distribution fits Category III functions best, while the exponential and the modified Rayleigh distribution perform

TABLE V
SIMULATION RESULTS AND COMPARISONS FOR CATEGORY III

Function	CRO_G				CRO_C			
	Mean	Std. Div.	Best	Rank	Mean	Std. Div.	Best	Rank
f_{14}	3.3089E+00	2.8636E+01	9.9800E-01	3	**1.1893E+00**	6.5107E+00	9.9800E-01	1
f_{15}	**5.9906E-04**	1.4234E-03	3.5413E-04	1	6.1532E-04	1.3869E-03	3.2870E-04	2
f_{16}	**-1.0305E+00**	2.1115E-02	-1.0316E+00	1	-1.0305E+00	2.1463E-02	-1.0316E+00	2
f_{17}	3.9795E-01	9.2041E-04	3.9789E-01	2	**3.9795E-01**	6.8579E-04	3.9789E-01	1
f_{18}	**3.0009E+00**	1.7177E-02	3.0000E+00	1	3.0013E+00	1.9859E-02	3.0000E+00	2
f_{19}	**-3.8615E+00**	1.1619E-02	-3.8628E+00	1	-3.8612E+00	1.4214E-02	-3.8628E+00	2
f_{20}	**-3.3125E+00**	6.7953E-02	-3.3217E+00	1	-3.3107E+00	7.8734E-02	-3.3214E+00	2
f_{21}	**-1.0126E+01**	2.3876E-01	-1.0153E+01	1	-1.0117E+01	2.6175E-01	-1.0151E+01	2
f_{22}	-1.0308E+01	6.6218E+00	-1.0402E+01	2	**-1.0350E+01**	6.7161E-01	-1.0401E+01	1
f_{23}	-1.0076E+01	1.5155E+01	-1.0536E+01	4	-1.0269E+01	1.0470E+01	-1.0536E+01	3
Avg.				**1.70**				1.80
Function	CRO_E				CRO_R			
	Mean	Std. Div.	Best	Rank	Mean	Std. Div.	Best	Rank
f_{14}	2.2217E+00	2.0650E+01	9.9800E-01	2	3.7510E+00	3.6642E+01	9.9800E-01	4
f_{15}	6.5008E-04	1.0962E-03	3.6644E-04	4	6.4918E-04	1.4605E-03	3.2389E-04	3
f_{16}	-1.0275E+00	1.2423E-01	-1.0316E+00	3	-1.0183E+00	4.3638E-01	-1.0316E+00	4
f_{17}	3.9833E-01	6.0664E-03	3.9789E-01	4	3.9802E-01	1.7142E-03	3.9789E-01	3
f_{18}	3.0043E+00	6.2059E-02	3.0000E+00	4	3.0014E+00	2.0636E-02	3.0000E+00	3
f_{19}	-3.8606E+00	2.5016E-02	-3.8627E+00	4	-3.8606E+00	2.1030E-02	-3.8627E+00	3
f_{20}	-3.3086E+00	9.0811E-02	-3.3218E+00	3	-3.3073E+00	9.8572E-02	-3.3215E+00	4
f_{21}	-9.9484E+00	2.2520E+00	-1.0141E+01	4	-1.0057E+01	5.0092E+00	-1.0153E+01	3
f_{22}	-1.0045E+01	7.5628E+00	-1.0383E+01	4	-1.0188E+01	8.9541E+00	-1.0401E+01	3
f_{23}	-1.0305E+01	2.3769E+00	-1.0531E+01	2	**-1.0465E+01**	8.0594E-01	-1.0534E+01	1
Avg.				3.40				3.10

TABLE VI
COMPUTATIONAL TIME (S)

Function	CRO_G	CRO_C	CRO_E	CRO_R
f_1	0.03281	0.03812	0.04027	0.04674
f_2	0.03346	0.03992	0.04295	0.04835
f_3	0.14172	0.15247	0.15339	0.16701
f_4	0.04294	0.04648	0.04752	0.05661
f_5	0.03841	0.04318	0.0421	0.05111
f_6	0.26363	0.26166	0.26285	0.26445
f_7	0.44546	0.45033	0.45933	0.44639
f_8	0.77906	0.76936	0.80069	0.76243
f_9	0.61241	0.59982	0.61947	0.61142
f_{10}	0.40227	0.39936	0.40278	0.40402
f_{11}	0.77743	0.79861	0.80926	0.8098
f_{12}	0.35557	0.37177	0.3812	0.38371
f_{13}	0.37318	0.38145	0.38168	0.39597
f_{14}	0.03674	0.03799	0.03733	0.03802
f_{15}	0.0631	0.07312	0.07215	0.08616
f_{16}	0.00058	0.0006	0.00059	0.00066
f_{17}	0.00131	0.00144	0.00141	0.0017
f_{18}	0.00174	0.00212	0.00206	0.0026
f_{19}	0.00211	0.00233	0.00229	0.00245
f_{20}	0.00444	0.00461	0.00441	0.00485
f_{21}	0.00202	0.00234	0.00233	0.00297
f_{22}	0.00221	0.00266	0.00245	0.00312
f_{23}	0.00248	0.00281	0.0028	0.00336

similarly for the Category II functions and outperform the other two distributions. Our study gives guidelines to design CRO for different types of optimization problems.

In the future we will conduct a systematic analysis on the impact of different parameters on perturbation function selection. We will also perform an analysis on the convergence speed of the four different distribution functions. Other distribution functions, e.g. the Levy distribution, will also be implemented on CRO to have a more general study of the overall performance.

ACKNOWLEDGEMENT

This work is supported in part by the Strategic Research Theme of Information Technology of The University of Hong Kong. A.Y.S. Lam is also supported in part by the Croucher Foundation Research Fellowship.

REFERENCES

[1] A. Y.S. Lam and V. O.K. Li, “Chemical-reaction-inspired metaheuristic for optimization,” *IEEE Trans. Evol. Comput.*, vol. 14, no. 3, pp. 381-399, Jun. 2010.

[2] J. Xu, A. Y.S. Lam, and V. O.K. Li, “Chemical reaction optimization for task scheduling in grid computing,” *IEEE Trans. Parallel Distrib. Syst.*, vol.22, no. 10, pp. 1624-1631, Oct. 2011.

[3] A. Y.S. Lam and V. O.K. Li, “Chemical reaction optimization for cognitive radio spectrum allocation,” in *Proc. IEEE GLOBECOM*, Dec. 2010, pp. 1-5.

[4] A. Y.S. Lam and V. O.K. Li, “Chemical reaction optimization: A tutorial,” *Memetic Computing*, vol. 14, no. 1, pp. 3-17, Mar. 2012.

[5] A. Y.S. Lam, J. Xu, and V. O.K. Li, “Chemical Reaction Optimization for population transition in peer-to-peer live streaming,” in *Proc. IEEE CEC’10*, Jul. 2010, pp. 1-8.

[6] B. Pan, A. Y.S. Lam, and V. O.K. Li, “Network Coding Optimization Based on Chemical Reaction Optimization,” in *Proc. GLOBECOM*, Dec. 2011, pp. 1-5.

[7] A. Y.S. Lam, V. O.K. Li, and J. J.Q. Yu, “Real-Coded Chemical Reaction Optimization,” *IEEE Trans. Evol. Comput.*, in press.

[8] J. J.Q. Yu, A. Y.S. Lam, and V. O.K. Li, “Evolutionary artificial neural network based on chemical reaction optimization,” in *Proc. IEEE CEC’11*, Jun. 2011, pp. 2083-2090.

[9] X. Yao and Y. Liu, “Fast evolutionary programming,” in *Evolutionary Programming V: Proceedings of the 5th Annual Conference on Evolutionary Programming*. Cambridge, MA: MIT Press, 1996

[10] C. Lee and X. Yao, “Evolutionary Programming Using Mutations Based on the Levy Probability Distribution,” *IEEE Trans. Evol. Comput.*, vol.8, no.1, pp.1-13, Feb. 2004.

[11] R. A. Krohling and L. S. Coelho, “PSO-E: Particle Swarm with Exponential Distribution,” in *Proc. IEEE CEC’06*, Jul. 2006, pp. 1428-1433.

[12] R. A. Krohling and E. Mendel, “Bare Bones Particle Swarm Optimization with Gaussian or Cauchy Jumps,” in *Proc. IEEE CEC’09*, May. 2009, pp. 3285-3291.

[13] B. Mandelbrot, *The Fractal Geometry of Nature. San Francisco*, CA:Freeman, 1982.

[14] G. Casella and R. L. Berger, *Statistical inference.* Duxbury Press, 2001.

[15] X. Yao, Y. Liu, and G. Lin, “Evolutionary programming made faster,” *IEEE Trans. Evol. Comput.*, vol. 3, no. 2, pp. 82-102, Aug. 1999.

[16] H. Dong, J. He, H. Huang, and W. Hou, “Evolutionary Programming Using a Mixed Mutation Strategy,” *Information Sciences*, vol. 177, no. 1, pp. 312-327, Jan. 2007.

[17] J. J.Q. Yu, V. O.K. Li, and A. Y.S. Lam, “Sensor Deployment for Air Pollution Monitoring Using Public Transportation System”, submitted for publication.