Adaptive Sampling Based Immune Optimization Approach in Noisy Environments Solving Chance Constrained Programming

Kai Yang[1] Zhuhong Zhang[2]

[1]College of Computer Science, Guizhou University
Guiyang 550025, China
csc.kyang@gzu.edu.cn

[2] Department of Information and Communication Engineering,
College of Big Data and Information Engineering,
Guizhou University
Guiyang 550025, China
sci.zhzhang@gzu.edu.cn

Received 28 June 2014; Revised 17 October 2014; Accepted 30 November 2014

Abstract. This work investigates a bio-inspired immune optimization algorithm in noisy environments, solving a class of chance-constrained programming problems with continuous decision variables but without any a priori distributional information on random variables. In this stochastic optimization method, an efficient adaptive sampling detection scheme is developed to detect individual's reliability, while those high-quality individuals in the current population can be identified based on the reported sample-allocation scheme; a clonal selection-based dynamical evolving mechanism is established to ensure evolving populations strong diversity, noisy suppression and rapidly moving one such population toward the desired region. The comparative experiments show that the proposed algorithm can effectively solve multi-modal chance-constrained programming problems with high efficiency and is of the potential for engineering application.

Keywords: Chance-constrained programming, immune optimization, reliability detection, sample-allocation, multimodality.

1. Introduction

Chance constrained programming (CCP) is originally introduced by Charnes and Cooper [1], involving in at least one chance constraint or probabilistic inequality. Solving such type of problem involves how to tackle three crucial issues: (i) efficient computational models for chance constraints, (ii) effective optimizers for rapidly finding the optimal solution, and (iii) identification between superior and inferior individuals. Although CCP was intensively studied by many mathematical researchers [2-4], few applicable optimization approaches have been reported for real-world chance-constrained programming problems such as control system design and electric power system dispatching. The main difficulty includes: (i) identifying individual's reliability becomes difficult when random variables are without any a priori distribution information, and (ii) the reliable region is generally non-convex.

Since the difficulty of solving CCP, many researchers [5-11] concerned with how to handle chance constraints under some special restrictions, e.g., normal distribution. In this way, some valuable optimization methods were developed such as convexity approximation [5-6], reliability-based optimization [7], robust approximation [8-9] and mixed-integer programming approaches [10-11]. Convexity approximation is an optimization tool using convexity constraints to approach the chance constraints; conversely, the reported reliability-based optimization [7] is to analytically transform the constraints into deterministic optimization models solved by mathematical programming approaches with single or double loop structures. Robust approximation as a distribution-free approach [9] is another alternative technique, in which the chance constraints are converted into robustness constraints; the main idea is to restrict the constraints to lie in an uncertainty set that is contained in the support, by ignoring the distributions of random variables. Mixed-integer programming approaches are to equivalently change CCP as a mixed-integer programming problem, and solve it by integer programming methods. Although these studies are valuable for the rapid development of CCP, many shortcomings need to be overcome, such as applicable scopes, sophisticated transformation, high computational complexity.

Recently, a great amount of theoretical work on CCP has concentrated on how to investigate the relation between solutions for CCP and approximate optimization models related, in which Monte Carlo simulation played an important role in problem transformation. Correspondingly, such sample estimation methods as sample average approximation (SAA) [12-15] and scenario approximation (SA) [16-18] were acquired to decide the precision of an approximate solution to the theoretical optimal solution. These studies require that all candidate solutions share the same large sample size, while CCP is replaced approximately by a deterministic optimization model after the random variables produce lots of observations. It is pointed out that although the Monte Carlo simulation as a popular tool is usually adopted to handle such random variables, it requires their sample sizes be large enough. This brings about expensive computational cost. Consequently, it becomes crucial how to control such sample sizes. On the other hand, Monte Carlo simulation based intelligent optimization methods are gaining great attention to researchers from the field of intelligent optimization, in which an important task is to investigate adaptive sampling strategies to suppress noisy influence to the process of optimization [19-20].

Immune optimization as an important branch of artificial immune systems has been popular since 1990s, for which many outstanding achievements were demonstrated to be superior to some classical intelligent optimization tools [21-23]. However, many of such achievements can only cope with static or dynamic optimization problems; few optimization techniques are specially designed to handle CCP problems. Consequently, from the angle of application, it is desired to explore high-efficiency immune optimization approaches to cope with CCP problems. In this work, we investigated an adaptive sampling detection-based immune optimization approach (ASDIOA) to find CCP's optimal solutions, and especially an efficient adaptive sampling detection approach was proposed based on Hoeffding's inequality [24]. Similar to other immune optimization approaches, ASDIOA bases on some bio-inspirations from the clonal selection principle. These methods, however, are different, due to different applications. In ASDIOA, a special sample-allocation method (OCBA) [25], suitable for expected value optimization without any constraint, is adopted to allocate the sample size of population to different empirical reliable solutions so as to estimate empirical objective values of such solutions. On the other hand, the proposed adaptive sampling detection method is used to dynamically determine the sample sizes of candidates and the estimate of probability of any chance constraint. Compared to our previous optimizers, ASDIOA is more efficient and can achieve effective solution search. To our knowledge, its characteristics do not appear in the existing optimization approaches.

2. Related Work

2.1. Sampling Approaches

In order to solve CCP problems, chance constraints should be well solved at first. There are two general ways to handle this kind of constraint. The first one is multi-dimensional integral which requires that we know their analytical formulations and distributional characteristics of random variables in advance. However, their analytical formulas are usually unknown in real world engineering applications. The second one is that those chance constraints are replaced by approximate models, for which the Monte Carlo simulation method is taken into account usually. However, it is difficult to decide the sample size of each random variable for a given candidate solution. To this point, many researchers made hard work under some special restrictions, and meanwhile some conclusions were drawn [12-18]. For example, after exhaustively discussing the relation between CCP and its related sample average approximation model, Luedtke and Ahmed [12] acquired an estimate of sample size under a high confidence level. One such estimate can guarantee that the sample average approximation model has a feasible solution close to the theoretical optimum. Campi and Calafiore [16] investigated the lower bound estimate of the number of convex constraints which replace the chance constraints. On the other hand, many other researches also tried to investigate how to dynamically decide sample sizes by means of special sampling techniques. Here, adaptive sampling [19-20] will become increasingly popular in the context of stochastic optimization, as different candidate solutions are attached different sample sizes. Such type of approach can heavily reduce computational cost and help for rapidly finding the optimum. For instance, Erick [19] studied an adaptive sampling technique that used a one-sided *t*-test to determine when to terminate the process of sampling; his experimental results suggested that such sampling method could adaptively determine the sample size of individual and thus might reduce computational cost. Higle and Zhao [20] examined experimentally the differences of adaptive and non-adaptive sampling schemes using two approaches of stochastic decomposition (SD) and SAA mentioned above. Their results show that there exists little difference between SD and SAA when only taking the quality of solution into account, but such two methods have different efficiencies, namely SAA results in high computational complexity but SD does not. In addition, ordinal optimization involves in an efficient sample estimation technique characterized by ordinal comparison. Chen [25] suggested an optimal computing budget allocation scheme (OCBA) for stochastic optimization but not for CCP, based on the Bayesian statistics. One such allocation scheme can allocate different sample sizes to different individuals, and especially better individuals can gain larger sample sizes. The experiments in such work showed that the approach was efficient. Subsequently, Lee et al. [26] proposed an extended version of OCBA for a class of stochastic optimization problems with stochastic constraints.

In our previous work [22, 27], two kinds of adaptive sampling schemes were developed for stochastic optimization problems. One is to decide the sample size of individual based on the hypothesis test, designed to emphasize that better individuals get larger sample sizes; the other is to control the sample size of individual by means of an a priori bound estimate, and meanwhile a reliability-dominance based adaptive sampling scheme allocates the sample size of population to different individuals.

2.2 Intelligent Optimization Approaches Handling CCP

From the viewpoint of optimization, although conventional numerical methods behave worse for nonlinear CCP, some advanced stochastic approaches have presented their potentials in the branch of intelligent optimization [15, 28-31]. For example, Liu [15] investigated how artificial neural networks approached chance constraints, and used genetic algorithms to search the desired solution; Poojari et al.[28] developed two similar genetic algorithms (SSGA-A and SSGA-B) with the same static sampling scheme; He et al.[29] solved the chance-constrained programming model of the vendor selection problem by an improved genetic algorithm; Udhayakumar et al.[30] utilized an extended genetic algorithm to handle the P-model of chance-constrained data envelopment problems; Luedtke [31] proposed a new branch-and-cut decomposition algorithm to deal with CCP problems with discrete distributions, finite support and random polyhedral constraints. These achievements are valuable for CCP, but need to make some improvements on computational complexity.

Recently, Zhao et al.[32] developed a hybrid immune optimization approach with a static sampling strategy to solve nonlinear CCP problems, in which the neural network was used to approach the expected value or probability functions, and meanwhile two operators of double cloning and double mutation were designed to accelerate the process of immune evolution. However, such approach needs an amount of runtime to simulate such functions. Our recent work [27, 33] also studied two adaptive sampling immune optimization algorithms and their theory for nonlinear joint and non-joint CCP. In such work, one of concerns is to investigate how to guarantee that all empirical individuals with different importance gain different sample sizes. However, some improvements need to be done, e.g., sampling efficiency and individual's reliability detection.

3. Problem Statement and Preliminaries

Consider the following nonlinear chance-constrained programming problem:

(CCP)
$$\min E\,[f(\boldsymbol{x},\xi)]$$
$$s.t.\begin{cases}\Pr\{G_i(\boldsymbol{x},\xi)\le 0\}\ge \alpha_i, & 1\le i\le I\\ g_j(\boldsymbol{x})\le 0,\ 1\le j\le J, & h_k(\boldsymbol{x})=0,\ 1\le k\le K,\end{cases}$$

with bounded and closed domain D in R^p, design vector $\boldsymbol{x}$ in D, random vector $\boldsymbol{\xi}$ in R^q and confidence levels α_i in (0,1), where E [.] and Pr{.} are the operators of expectation and probability, respectively; $f(\boldsymbol{x}, \boldsymbol{\xi})$ and $G_i(\boldsymbol{x}, \boldsymbol{\xi})$ are the stochastic objective and constraint functions, respectively; $g_j(\boldsymbol{x})$ and $h_k(\boldsymbol{x})$ are the deterministic constraint functions. If a candidate solution satisfies all the above constraints, it is called a reliable solution, and an unreliable solution otherwise. Introduce the following constraint violation function to check whether candidate $\boldsymbol{x}$ is reliable:

$$\begin{aligned}\Gamma(x) &= I^{-1}\sum_{i=1}^{I}\max\{\alpha_i-\Pr\{G_i(x,\xi)\le 0\},0\}\\ &\quad + J^{-1}\sum_{j=1}^{J}\max\{g_j(x),0\}+K^{-1}\sum_{k=1}^{K}|h_l(x)|.\end{aligned} \tag{1}$$

Obviously, $\boldsymbol{x}$ is reliable only when $\Gamma(\boldsymbol{x}) = 0$. If $\Gamma(\boldsymbol{x}) < \Gamma(\boldsymbol{y})$, we prescribe that $\boldsymbol{x}$ is superior to $\boldsymbol{y}$.

Chen [25] developed a sample-allocation method (OCBA) to allocate a total sample size of population to different individuals so as to find the best solution. In the present work, OCBA is adopted to decide the sampling sizes of m empirically reliable candidates with a total sample size of T. More precisely, let A represent a population of the m candidates with sample size T, and N_j^l the sample size of the j-th candidate at the moment l. σ_i^l denotes the variance of observations for candidate i with sample size N_i^l, and c stands for the best candidate whose empirical objective value is smallest in A. $\delta_{c,i}$ is the Euclidian distance between candidates c and i in the design space. Thereafter, OCBA can be reformulated below:

Step 1. Set $l \leftarrow 0$. Each candidate in A creates m_0 observations with $N_1^l = N_2^l = \ldots = N_m^l = m_0$;

Step 2. Decide the best candidate c through the empirical means of all candidates in A, and calculate $\sigma_i, \delta_{c,i}$, $1 \le i \le m$;

Step 3. If $\sum_{i=1}^{m} N_i^l > T$, this procedure ends, and outputs the empirical means of candidates in A; otherwise, go to Step 4;

Step 4. Increase the computing budget by Δ, and decide $N_i^{l+1}, i = 1,2,..., m$ through the following formula

$$\frac{N_i}{N_j} = \left(\frac{\sigma_i / \delta_{c,i}}{\sigma_j / \delta_{c,j}}\right)^2, i, j \in \{1,2,\cdots,m\}, i \neq j \neq m, N_c = \sigma_c \sqrt{\sum_{i=1, i\neq c}^{m} \frac{N_i^2}{\sigma_i^2}}\ ; \quad \textbf{(2)}$$

Step 5. If $N_i^{l+1} > N_i^l$, then $N_i^{l+1} \leftarrow N_i^l + \max\{0, N_i^{l+1} - N_i^l\}$, and go to Step 2.

Theorem 1 (Hoeffding's Inequality)[24]. If $X_1, X_2,\ldots, X_n$ are iid random variables with $a \le X_i \le b$ and mean μ, then

$$\left|\overline{X}_n - \mu\right| \le (b-a)\sqrt{\frac{1}{2n}\ln\left(\frac{2}{\delta}\right)},$$

with probability at least 1-δ.

4. Adaptive Sampling Detection for Chance Constraints

Sample average approximation is a usual approach handling chance-constrained programming [10], in which Monte Carlo simulation is used to estimate the probability of a chance constraint under a given sample size. However, when intelligent optimization approaches solve an approximate model of such kind of problem, it is impossible to avoid identifying whether a candidate is reliable or not. If the candidate is empirically reliable, it is desired to attach a large sample size, as its empirical objective mean is expected to approach the theoretical mean as possible; conversely, it is not necessary to provide an empirically unreliable candidate with a large sample size. To this point, we develop a Hoeffding's inequality-based sampling detection scheme to decide the sample size for a candidate, and meanwhile detect whether such candidate satisfies a given chance constraint. More precisely, let T be a maximal sampling size of candidate $\boldsymbol{x}$; α denotes the confidence level of a given chance constraint, satisfying $\Pr\{g(\boldsymbol{x},\xi)\le 0\}\ge\alpha$; $p_n(\boldsymbol{x})$ represents the probability estimate at the moment n through Monte Carlo simulation. Thereafter, the sampling scheme, simply say adaptive sampling detection, is formulated as follows:

Step 1. Input parameters: initial sample size m_0, sampling amplitude λ, maximal sampling size T;

Step 2. Set $m=m_0$, $s=m_0$; calculate the probability estimate $p_n(\boldsymbol{x})$ with m_0 observations of candidate $\boldsymbol{x}$;

Step 3. End this procedure only when $m>T$ or $|p_n(x)-\alpha| > \sqrt{ln(2/\delta)/(2m)}$;

Step 4. Set $s\leftarrow\lambda s$;

Step 5. Create s observations of the candidate, and decide the successful rate, $r_n\leftarrow w/s$, where w denotes the number that the inequality, $g(\boldsymbol{x},\xi)\le 0$, is true for the observations;

Step 6. Update the probability estimate, i.e., $p_n(\boldsymbol{x})\leftarrow(p_n(\boldsymbol{x})\times m+r_n s)/(m+s)$;

Step 7. Set $m\leftarrow m+s$, and return to Step 3.

When all the candidates in a given population are attached the same maximal sample size T, the above procedure follows that different candidates get different sample sizes. Worse candidates gain smaller sample sizes. Here, candidate $\boldsymbol{x}$ is called empirically reliable if $p_n(\boldsymbol{x})\ge\alpha$ and the above deterministic constraints are satisfied; otherwise , it is said to be empirically unreliable.

5. Algorithm Formulation and Design

The clonal selection theory explains a biological phenomenon which antibodies respond to an antigen. It also hints a learning mechanism that the antigen can be gradually eliminated by some antibodies. To utilize such theory to design ASDIOA for CCP, a real-coded candidate is regarded as an antibody; the problem itself is viewed as the antigen. Based on the above OCBA, adaptive sampling detection and bio-immune inspirations, ASDIOA can be described in detail below:

Step 1. Input parameters: population size N, initial sample size m_0, sampling amplitude λ, computing budget Δ and maximal clonal size C_{max};

Step 2. Set $n\leftarrow 1$. Generate an initial population A_n of N random antibodies;

Step 3. Calculate the probability estimate of each chance constraint in the above CCP for each antibody in A_n through the adaptive sampling detection with a maximal sample size, $T=m_0[(n+1)^{1/2}+1]$; detect whether each antibody in A_n is empirically reliable through equation (1);

Step 4. Divide A_n into empirically reliable sub-population B_n and unreliable sub-population C_n;

Step 5. Allocate the sample size of population, $T=m_0|B_n|log(n+2)$, to each antibody in B_n through the above OCBA, and calculate the empirical objective values of all the antibodies;

Step 6. Each antibody in B_n and C_n proliferates $cl(\boldsymbol{x})$ clones with $cl(\boldsymbol{x})= round\,(C_{max}/(\Gamma(\boldsymbol{x})+1)+1)$, which creates a clonal population D_n, where $round(z)$ is a maximal integer not beyond z;

Step 7. Each clone in D_n shifts its genes through the conventional Gaussian mutation with a mutation rate $p_m=1/(\Gamma_{max}-\Gamma(\boldsymbol{x})+1)$, where Γ_{max} denotes the maximal of constraint violations for all the clones in D_n; thereafter, all mutated clones constitute E_n and execute evaluation through Steps 3 to 5 with $n=1$;

Step 8. Execute comparison between antibodies and their clones in $A_n \cup E_n$. If some antibody in A_n is inferior to the best of its clones, such antibody is replaced by the best clone. This creates a new population A_{n+1};

Step 9. If the termination criterion is not satisfied, then set $n \leftarrow n+1$ and go to Step 3; otherwise, output the best antibody viewed as the optimal solution.

In the above algorithm, after checking reliability for each antibody in A_n in step 3, two sub-populations evolve respectively toward different directions through steps 6 to 8. The empirical objective values of antibodies in step 5 are decided through dynamically allocating a time-dependent population sample size to empirically reliable antibodies. Steps 6 and 7 urge high-quality antibodies to produce multiple clones with small mutation rates. Obviously, those survival and better antibodies can gain larger sample sizes when gradually increasing the iteration number n. Therefore, ASDIOA is a dynamically sampling optimizer.

Theorem 2. ASDIOA's computational complexity is $O(N(Im_0\sqrt{n+1}+C_{\max}\ \log C_{\max}))$.

Proof. For a given population A_n with size N and maximal sample size T as in ASDIOA, step 3 executes at most $I\times T$ times to create samples in the worst case, and hence the complexity is $O(IN\,m_0\,(n+1)^{1/2})$ because of small J and K as in section 3; step 4 divides A_n into two sub-populations through N executions; step 5 computes the empirical objective means of empirically reliable antibodies in B_n with a total of evaluations, $m_0|B_n|log(n+2)$, and hence the complexity is $O(Nm_0log(n+2))$ in the worst case. In addition, step 7 executes mutation with at most $N(1+C_{max})$. In step 8, each clonal sub-population needs to execute comparison between antibodies with at most $C_{max}logC_{max}$ times, and thus the complexity is $(NC_{max}logC_{max})$. Summarily, ASDIOA's computational complexity in the worst case is decided by

$$\begin{aligned} O_c &= O(INm_0\sqrt{n+1}) + O(Nm_0\log(n+2)) + O(NC_{\max}\log C_{\max}) \\ &= O(N(Im_0\sqrt{n+1} + C_{\max}\log C_{\max})). \end{aligned}$$

6. Numerical Experiments

In this experimental study, four representative intelligent algorithms suitable for CCP, i.e., one neural network-based particle swarm optimization (HPSO) [35-36], two competitive steady genetic algorithms (SSGA-A and SSGA-B) [28] and one recent noisy immune optimization approach (NIOA) [33], are picked up to compare to ASDIOA by means of the following test examples. Our experiments are executed on a personal computer with CPU/3GHz and RAM/2 GB and also VC++. Especially, the three approaches of HPSO, SSGA-A and SSGA-B are static sampling optimization approaches with the same fixed sample size for each individual, whereas NIOA is an adaptive sampling optimization approach with a dynamic sample size for each individual. Their parameter settings are the same as those in their corresponding literature except their evolving population sizes. All the above algorithms take their population sizes 40, while respectively executing 30 times on each test problem. Their same termination criterion is that the total of evaluations of individuals during evolution is 10^7. Especially, HPSO is a BP neural network-based optimization approach, in which the total training sample size is set as 10^7. In ASDIOA, after experimental tuning we take $m_0=30$, $\lambda=1.5$, $\Delta=20$ and $C_{max}=2$. In order to effectively execute comparison between the algorithms, each of those solutions, gotten by them is required to re-evaluate with the sample size 10^6. Here, we give a test criterion to examine whether the solutions satisfy the chance constraints; in other words, let Λ denote a solution set with size M and $p^i(\boldsymbol{x}_l)$ the probability estimate of the i-th chance constraint as in Section 2 for $\boldsymbol{x}_l$ in Λ. The test criterion is designed as follows:

$$IAE = M^{-1}\sum_{l=1}^{M}\sum_{i=1}^{I}\left|p^i(\boldsymbol{x}_l)-\alpha_i\right|. \quad \textbf{(3)}$$

Example 1. Uncertain Feed mixer design [28]

$$\min E\,[24.55x_1 + 26.75x_2 + 39.0x_3 + 40.50x_4 + \xi]$$

$$s.t\begin{cases} x_1 + x_2 + x_3 + x_4 = 1, \quad 2.3x_1 + 5.6x_2 + 11.1x_3 + 1.3x_4 \ge 5, \\ \Pr\{\eta_1 x_x + \eta_2 x_2 + \eta_3 x_3 + \eta_4 x_4 \ge 21\} \ge 0.8, \\ x_1, x_2, x_3, x_4 \ge 0, \quad \eta_1 \sim N(12, 0.2809^2), \eta_2 \sim N(11.9, 0.1936^2), \\ \eta_3 \sim N(41.8, 20.25^2), \eta_4 \sim N(52.1, 0.6241^2), \quad \xi \sim N(0,1). \end{cases}$$

Although this is a linear CCP problem with 4 decision variables, the five random variables influence the process of solution search seriously. It can be transformed into a deterministic optimization problem with the theoretical minimum 30.30 at the point (0.002, 0.733, 0.056, 0.209). However, in order to examine the performances of the above algorithms, we directly solve such problem. After respectively running 30 times, the algorithms can get their solution sets which provide us with some statistical results listed in Table 1. Figure 1 below shows the box plot of the results; Figure 2 displays their average search curves.

Table 1. Comparison of statistical result for examp1e 1.

Algorithm	*Max*	*Min*	*Mean*	*Std.Dev*	*CI*	*IAE*	*FR*	*AR*(s)
HPSO	35.85	23.23	32.87	2.99	[31.80,33.94]	0.44	0%	15.5
SSGA-A	30.71	30.24	30.45	0.13	[30.41,30.50]	0.05	0%	7.2
SSGA-B	30.75	30.14	30.41	0.13	[30.36,30.45]	0.05	3%	7.3
NIOA	30.36	30.02	30.24	0.08	[30.21,30.27]	0.07	7%	7.3
ASDIOA	30.51	30.17	30.33	0.09	[30.30,30.36]	4.18×10^{-4}	97%	6.7

CI represents the confidence interval of empirically objective means for the 30 solutions acquired; *IAE* is computed through equation (3) for unreliable solutions gotten by a given algorithm; *FR* stands for the rate of reliable solutions among all the gotten solutions; *AR* is the average runtime after 30 runs for a given algorithm.

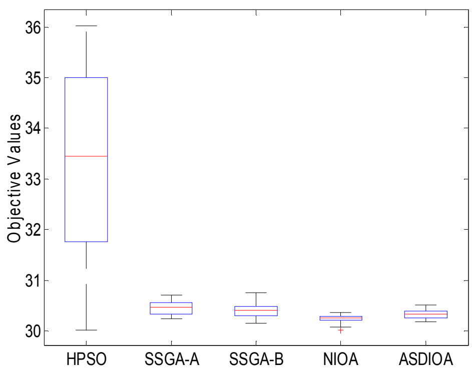

Fig. 1. Example 1: Box-plot.

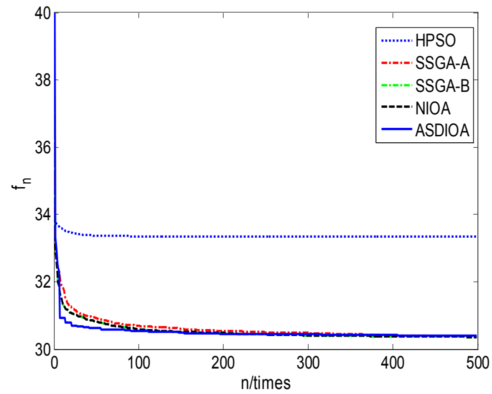

Fig 2. Example 1: Average search curves.

In Table 1, the values on *FR*, listed in the eighth column hint that HPSO and SSGA-A can not find reliable solutions and that SSGA-B and NIOA can only get a few reliable solutions, whereas the solutions gotten by ASDIOA are almost reliable. On the other hand, the values on *IAE* in the seventh column show that ASDIOA only causes the smallest constraint violation for the chance constraints, which indicates that the adaptive sampling detection as in section 3 can effectively handle the chance constraints. However, those compared approaches are difficult in solving such kind of constraint. It seems to be true that all the algorithms but HPSO have almost the same solution quality through the values as in columns 4 and 6. In fact, as associated to their constraint violations (*IAE*) and the theoretical minimum, ASDIOA has the best solution quality obviously; especially, the values on *CI* hint that the minimum, 30.30, is included in the narrow confidence interval obtained by ASDIOA, but other algorithms are difficult. In addition, apart from HPSO, the other three algorithms can also gain better solution qualities.

Through columns 2, 3, 5 and 6, we can get the conclusion that all the algorithms but HPSO have relatively stable search performances. The statistical box-plots of the empirical objective values in Figure 1, acquired by the algorithms after 30 executions illustrate a fact that NIOA and ASDIOA can obtain similar effects superior to those gained by other algorithms, and meanwhile their objective values cover small scopes. By Figure 2, we also note that ASDIOA is convergent, and HPSO can only achieve local solution search; relatively, ASDIOA is a rapid search procedure. Lastly, the values on *AR*, listed in the ninth column present clearly that all the algorithms but HPSO have high search efficiencies; HPSO spends much more than runtime to solve the above problem than each of other algorithms.

Example 2. Uncertain multi-modal optimization

$$\max E[\sum_{n=1}^{3} x_i \cdot \sin(i\pi x_i) + \xi]$$

$$s.t. \begin{cases} \Pr\{\eta_1 x_1 + \eta_2 x_2 + \eta_3 x_3 \le 10\} \ge 0.7, \\ \Pr\{\eta_4 x_1 + \eta_5 x_2 + \eta_6 x_3 \le 100\} \ge 0.8, \\ x_1, x_2, x_3 \ge 0, \quad \xi \sim N(0,1), \eta_1 \sim U(0.8,1.2), \quad \eta_2 \sim U(1,1.3), \\ \eta_3 \sim U(0.8,1), \quad \eta_4 \sim N(1,0.5), \quad \eta_5 \sim \mathrm{Exp}(1.2), \quad \eta_6 \sim \mathrm{Log}(0.8,0.6). \end{cases}$$

This is a multimodal chance-constrained programming problem gotten through modifying a static multimodal optimization problem [34], where 3 decision variables and 4 random variables are included. The main difficulty of solving such problem involves two aspects: (i) the original static problem has multiple local optima, and (ii) the above problem involves in multiple kinds of random variables. Like the above experiment, the approaches can get their statistical results (Table 2) and performance curves (Figures 3 and 4) after 30 runs.

Table 2. Comparison of statistical result for example 2.

Algorithm	*Max*	*Min*	*Mean*	*Std.Dev*	*CI*	*IAE*	*FR*	*AR*(s)
HPSO	11.09	-4.41	3.11	4.20	[1.60,4.61]	0.06	0.80	10.0
SSGA-A	10.08	7.51	9.07	0.59	[8.86,9.28]	0.03	0.80	7.1
SSGA-B	10.10	6.47	8.85	0.74	[8.58,9.12]	0.03	0.87	7.1
NIOA	9.88	9.23	9.55	0.15	[9.50,9.61]	0.06	0.67	7.1
ASDIOA	10.08	9.47	9.82	0.18	[9.75,9.86]	2.17×10^{-3}	0.93	6.3

Following Table 2, the values on *FR* show that the above algorithms can all find some reliable solutions after 30 runs; relatively, ASDIOA is more effective. As related to the values listed in the seventh column, we notice that the approaches can almost handle the above chance constraints; especially, ASDIOA's adaptive sampling detection has presented its prominent performance with the aspect of dealing with the chance constraints. On the other hand, we can acquire significantly different solution qualities for the algorithms by means of the statistical results given in columns 2 to 6. In other words, ASDIOA's solution quality is significantly superior to those acquired by other approaches; NIOA is secondary, and HPSO is worst (see Figure 3). We also observe that ASDIOA has presented its strong and stable evolving ability of searching the optimum (see Figure 4), as it can get the largest objective value and the narrowest confidence interval.

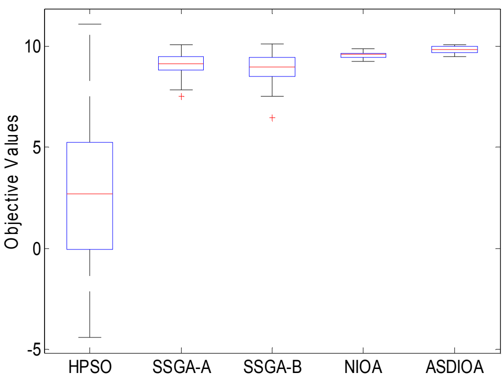

Fig 3. Example 2: Box-plot.

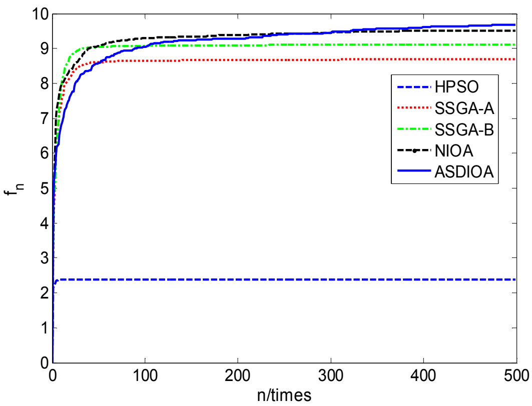

Fig 4. Example 2: Average search curves.

Figure 4 indicates that NIOA and ASDIOA are convergent but other algorithms get into local search. This illustrates that the adaptive sampling schemes presented in NIOA and ASDIOA can help these two algorithms improve their solution qualities. Additionally, we can get the same conclusion on performance efficiency as that given in Example 1, namely ASDIOA spends the least time to execute the process of solution search but HPSO is worst when doing so.

Example 3. Uncertain Multi-modal optimization

$$\min\ E\,[20+10\cos(6\pi x_1)+10\cos(8\pi x_2)+6x_1^2+8x_2^2+\xi]$$
$$s.t.\begin{cases}\Pr\{x_1\eta_1+x_2\eta_2\le 0.67\}\ge 0.8,\\ x_1+x_2\ge 0.6\quad 0\le x_1\le 1\quad 0\le x_2\le 1,\\ \xi\sim N(0,1),\eta_1\sim N(1.2,0.2809^2),\eta_2\sim N(0.9,0.1936^2).\end{cases}$$

This is a multimodal chance-constrained programming problem gotten by inserting random variables into a static multimodal optimization problem [37], where 2 decision variables and 3 random variables are included. Like the above experiment, the five approaches acquire their respective experimental results listed in Table 3 below, and meanwhile their performance curves are drawn in Figures 5 and 6.

Table 3. Comparison of statistical result for example 3.

Algorithm	*Max*	*Min*	*Mean*	*Std.Dev*	*CI*	*IAE*	*FR*	*AR*(s)
HPSO	28.82	4.52	18.01	6.02	[15.86,20.17]	0.0	1.0	6.1
SSGA-A	15.57	15.46	15.52	0.02	[15.51,15.53]	0.0	1.0	4.3
SSGA-B	19.07	15.49	16.25	1.34	[15.77,16.73]	0.0	1.0	4.3
NIOA	15.70	15.57	15.63	0.03	[15.62,15.64]	0.0	1.0	4.1
ASDIOA	4.74	4.41	4.58	0.08	[4.55,4.61]	0.0	1.0	2.9

Relying upon Table 3, the values on *FR* illustrate that all the algorithms can find reliable solutions during 30 runs. As a result, the values on *IAE*, listed in the seventh column illustrate that the approaches can all handle the above chance constraints. However, only depending on the values on *FR* and *IAE*, it is impossible to distinguish the performances between the five algorithms. In fact, these approaches have significantly different solution qualities through the statistical results given in columns 2 to 6. In other words, ASDIOA's solution quality is superior to those obtained by other algorithms, since its solutions are close to the global optimal solution; other algorithms can only find local optimal solutions, since they get easily into local search. Consequently, from the viewpoint of solution quality, ASDIOA is best; NIOA is secondary, and HPSO is worst (see Figure 5). On the other hand, the values on *AR*, listed in the ninth column show clearly that ASDIOA has the highest search efficiency; NIOA, SSGA-A and SSGA-B have similar efficiencies; HPSO spends the more time to solve the problem than each of other algorithms.

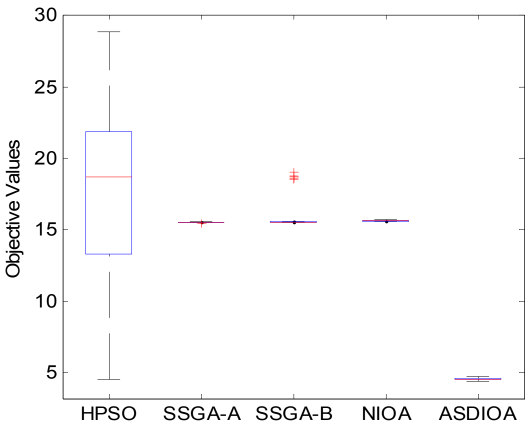

Fig 5. Example 3: Box-plot.

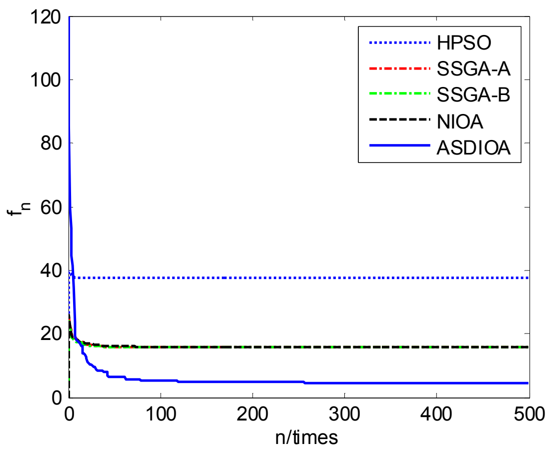

Fig 6. Example 3： Average search curves.

As associated to Figure 5, we can see that only ASDIOA can find some solutions close to the global optimal solution, while NIOA, SSGA-A and SSGA-B can only converge to the local optimum, and HPSO can not converge. Through the average search curves (see Figure 6), it is clear that ASDIOA displays its fast and stable ability of searching the optimum. Figure 6 also hints that ASDIOA is convergent, but other algorithms can only get into local search.

Example 4. Car side-impact problem [7]

The car side-impact problem is described by a stochastic programming model. It includes 7 decision variables (x_1, x_2,…, x_7), 4 random variables (ξ_1, ξ_2, ξ_3, ξ_4) and 10 stochastic constraints. Such programming model is given by

$$
\begin{aligned}
&\min E[1.98+4.9x_1+6.67x_2+6.98x_3+4.01x_4+1.78x_5+0.00001x_6+2.73x_7+\eta] \\
&s.t. \\
&1.16-0.3717x_2x_4-0.00931x_2\xi_3-0.484x_3\xi_2+0.01343x_6\xi_3\le 1, \\
&0.261-0.0159x_1x_2-0.188x_1\xi_1-0.019x_2x_7+0.0144x_3x_5+0.0008757x_5\xi_3+ \\
&0.08045x_6\xi_2+0.00139\xi_1\xi_4+0.00001575\xi_3\xi_4\le 0.32, \\
&0.214+0.00817x_5-0.131x_1\xi_1-0.0704x_1\xi_2+0.03099x_2x_6-0.018x_2x_7+ \\
&0.0208x_2\xi_1+0.121x_3\xi_2-0.00364x_5x_6+0.0007715x_5\xi_3-0.0005354x_6\xi_3+ \\
&0.00121\xi_1\xi_4+0.00184\xi_2\xi_3-0.018x_2^2\le 0.32, \\
&0.74-0.61x_2-0.163x_3\xi_1+0.001232x_3\xi_3-0.166x_7\xi_2+0.227x_2^2\le 0.32, \\
&28.98+3.818x_3-4.2x_1x_2+0.0207x_5\xi_3+6.63x_6\xi_2-7.77x_7\xi_1+0.32\xi_2\xi_3\le 32, \\
&33.86+2.95x_3+0.1792\xi_3-5.057x_1x_2-11x_2\xi_1-0.0215x_5\xi_3-9.98x_7\xi_1+22\xi_1\xi_2\le 32, \\
&46.36-9.9x_2-12.9x_1\xi_1+0.1107x_3\xi_3\le 32, \\
&4.72-0.5x_4-0.19x_2x_3-0.0122x_4\xi_3+0.009325x_6\xi_3+0.000191\xi_4^2\le 4, \\
&10.58-0.674x_1x_2-1.95x_2\xi_1+0.02054x_3\xi_3-0.0198x_4\xi_3+0.028x_6\xi_3\le 9.9, \\
&16.45-0.489x_3x_7-0.843x_5x_6+0.0432\xi_2\xi_3-0.0556\xi_2\xi_4-0.000786\xi_4^2\le 15.7, \\
&\eta\sim N(0,1),\xi_1\sim N(0.35,0,006),\xi_2\sim N(0.192,0.006),\xi_3,\xi_4\sim N(0,10), \\
&x_1,x_3,x_4\in[0.5,1.5],x_2\in[0.45,1.35],x_5\in[0.875,2.65],x_6,x_7\in[0.4,1.2].
\end{aligned}
$$

We transform the ten stochastic constraints into chance constraints with the same confidence level α. In this experiment, we take $\alpha=0.8$. Similar to the experiments above, the five approaches obtain their respective statistical results listed in Table 4, and correspondingly their performance curves are given in Figures 7 and 8.

Table 4. Comparison of statistical result for example 4.

Algorithm	*Max*	*Min*	*Mean*	*Std.Dev*	*CI*	*IAE*	*FR*	*AR*(s)
HPSO	13.75	10.45	12.01	0.79	[11.73, 12.30]	0.80	0.00	9.8
SSGA-A	24.71	24.51	24.58	0.04	[24.57, 24.59]	0.03	0.00	5.1
SSGA-B	24.64	24.50	24.58	0.04	[24.56, 24.59]	0.03	0.00	5.1
NIOA	24.76	24.54	24.66	0.06	[24.64, 24.68]	0.06	0.13	5.2
ASDIOA	28.02	25.82	27.03	0.54	[26.83, 27.22]	2.75×10^{-4}	0.93	5.1

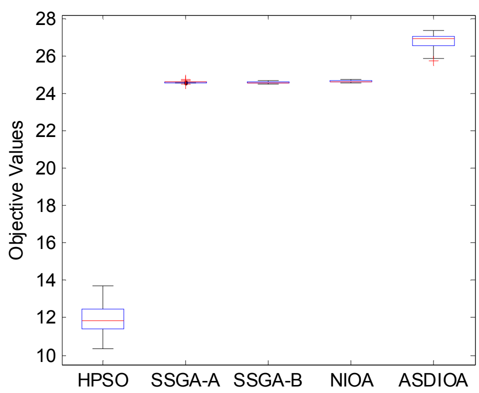

Fig 7. Example 4: Box-plot.

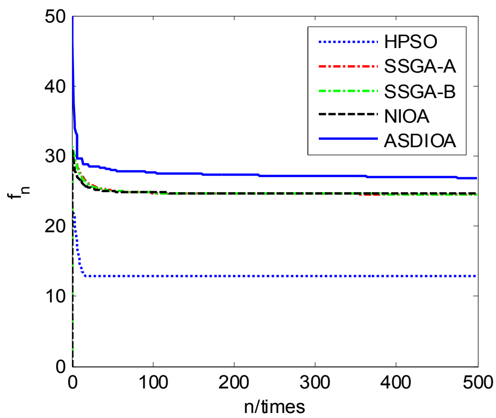

Fig 8. Example 4: Average search curves.

As is shown in Table 4, the values on *FR*, displayed in the eighth column present that 93% of the solutions gotten by ASDIOA are reliable, and meanwhile only 13% of the solutions gained by NIOA are reliable. Unfortunately, NPSO, SSGA-A and SSGA-B can not find reliable solutions. Especially, the values on *IAE* as in the seventh column illustrate that the solutions obtained by all the algorithms but HPSO are almost located at the boundary of the reliable region. It seems that ASDIOA can only find worse solutions and has the inferior search performance than each of NIOA, SSGA-A and SSGA-B, since it can only obtain a larger objective mean. In fact, as associated to the values on *FR*, we see that ASDIOA can get the better solution quality than each of them.

Thereby, we can draw the conclusion that with respect of solution quality, ASDIOA is best and HPSO is worst. On the other hand, we also observe that NIOA, SSGA-A, SSGA-B and ASDIOA are of the almost same efficiency, whereas HPSO needs much more than runtime to solve the above problem. In addition, it seems that the same conclusion can be drawn by means of Figures 7 and 8, namely ASDIOA can only acquire the worst solution quality and get easily into local search. This is not right basically, as the solutions by all the approaches but ASDIOA are almost unreliable solutions.

7. Conclusions

In real-world engineering optimization, lots of problems can be described by CCP models. With the increasing requirement of handling uncertain optimization problems, solving CCP will become popular in the field of intelligent optimization. Thus, inspired by the dynamic characteristics and mechanisms of the immune system, this work focuses on probing into a bio-inspired immune optimization algorithm in noisy environments for a class of CCP problems without any a priori noisy information. Especially, an efficient adaptive sampling detection scheme is developed to handle chance constraints, while the existing OCBA is used to make high-quality individuals gain large sample sizes. Such algorithm is an optimizer capable of effectively executing noisy suppression, adaptive sample-allocation and chance constraint handling. The experimental results hint that the proposed approach is a competitive, effective and efficient optimizer.

Acknowledgement

This research is supported by Doctoral Fund of Ministry of Education of China (20125201110003), National Natural Science Foundation (61065010) and EU FP7 HAZCEPT (318907).

References

[1] A. Charnes and W.W. Cooper, "Chance-Constrained Programming," *Management Science*, Vol. 6, pp. 73-79, 1959.

[2] A. Prékopa, "On Probabilistic Constrained Programming," in *Proceedings of the Princeton Symposium on Mathematical Programming*, pp. 113-138, 1970.

[3] A. Prékopa, "Probabilistic programming," In: A.Ruszczynski, A.Shapiro (eds.): *Handbook in Operations Research and Management Science*, Vol. 10, pp.267-351, North-Holland Publishing Company, Amsterdam, 2003.

[4] A. Shapiro, D. Dentcheva, A. Ruszczyński, *Lectures on Stochastic Programming: Modeling and Theory*, SIAM, Philadelphia, 2009.

[5] A. Nemirovski, A. Shapiro, "Convex Approximations of Chance Constrained Programs," *SIAM Journal on Optimization*, Vol. 17, pp. 969-996, 2006.

[6] L.J. Hong, Y. Yang, L. Zhang, "Sequential Convex Approximations to Joint Chance Constrained Programs: A Monte Carlo Approach," *Operations Research*, Vol. 59, pp. 617-630, 2011.

[7] K. Deb, S. Gupta, D. Daum, et al., "Reliability-Based Optimization Using Evolutionary Algorithms," *IEEE Transactions on Evolutionary Computation*, Vol. 13, pp. 1054-1074, 2009.

[8] D. Bertsimas, M. Sim, "The Price of Robustness," *Operations Research*, Vol. 52, pp.35-53, 2004.

[9] O. Klopfenstein, D. Nace, "A Robust Approach to the Chance-constrained Knapsack Problem," *Operations Research Letters*, Vol. 36, pp. 628-632, 2008.

[10] J. Luedtke, S. Ahmed, G. Nemhauser, "An Integer Programming Approach for Linear Programs with Probabilistic Constraints," *Mathematical Programming*, Vol. 122, pp. 247-272, 2010.

[11] X. Zheng, X. Sun, D. Li, X. Cui, "Lagrangian Decomposition and Mixed-integer Quadratic Programming Reformulations for Probabilistically Constrained Quadratic Programs," *European Journal of Operational Research*, Vol. 221, pp. 38-48, 2012.

[12] J. Luedtke, S. Ahmed, "A Sample Approximation Approach for Optimization with Probabilistic Constraints," *SIAM Journal on Optimization*, Vol. 19, pp. 674-699, 2008.

[13] S. Ahmed, A. Shapiro, E. Shapiro, "The Sample Average Approximation Method for Stochastic Programs with Integer Recourse," Preprint available at www.optimization-online.org, February 2002.

[14] B.K. Pagnoncell, S. Ahmed, A. Shapiro, "Sample Average Approximation Method for Chance Constrained Programming: Theory & Applications," *Journal of Optimization Theory and Applications*, Vol. 142, pp. 399-416, 2009

[15] B. Liu, Theory and Practice of Uncertain Programming, Springer, Berlin, 2009.

[16] M.C. Campi, G. Calafiore, *Decision Making in An Uncertain Environment: the Scenario-Based Optimization Approach*, Working paper, 2004.

[17] G.C. Calafiore, M.C. Campi, "The Scenario Approach to Robust Control Design," *IEEE Transactions on Automatic Control*, vol. 51, pp. 742-753, 2006.

[18] A. Nemirovski, A. Shapiro, "Scenario Approximations of Chance Constraints," in *Probabilistic and Randomized Methods for Design under Uncertainty*, pp. 3-47, 2006.

[19] E. Cantú-Paz, "Adaptive Sampling for Noisy Problems," in *Proceedings of Genetic and Evolutionary Computation (GECCO 2004)*, pp. 947-958, 2004.

[20] J.L. Higle, L. Zhao, "Adaptive and Nonadaptive Samples in Solving Stochastic Linear Programs," *Computational Optimization and Applications*, Vol. 21, pp. 122-134, 2008.

[21] D. Dasgupta, S. Yu, F. Nino, "Recent Advances in Artificial Immune Systems: Models and Applications," *Applied Soft Computing*, Vol. 11, pp. 1574-1587, 2011.

[22] Z.H. Zhang, X. Tu, "Immune Algorithm with Adaptive Sampling in Noisy Environments and Its Application to Stochastic Optimization Problems," *IEEE Computational Intelligence Magazine*, vol. 2, pp. 29-40, 2007.

[23] K. Trojanowski, T.W. Sławomir, "Immune-based Algorithms for Dynamic Optimization," *Information Sciences*, vol. 179, pp. 1495-1515, 2009.

[24] Z. Lin, Z. Bai, *Probability Inequalities*, Springer, Berlin, 2010.

[25] C.H. Chen, "Efficient Sampling for Simulation-based Optimization under Uncertainty," in *Proceedings of the Fourth International Symposium on Uncertainty Modeling and Analysis (ISUMA'03)*, IEEE Press, pp. 386-391, 2003.

[26] L.H. Lee, N.A. Pujowidianto, L.W. Li, et al., "Approximate Simulation Budget Allocation for Selecting the Best Design in the Presence of Stochastic Constraints," *IEEE Transactions on Automatic Control*, Vol. 57, pp. 2940-2945, 2012.

[27] Z.H. Zhang, L.Wang, M. Liao, "Adaptive Sampling Immune Algorithm Solving Joint Chance-constrained Programming," *Journal of Control Theory and Applications*, Vol. 11, pp. 237-246, 2013.

[28] C.A. Poojari, B. Varghese, "Genetic Algorithm Based Technique for Solving Chance Constrained Problems," *European Journal of Operational Research*, Vol. 185, pp. 1128-1154, 2008.

[29] S. He, S.S. Chaudhry, Z.Lei, et al., "Stochastic Vendor Selection Problem: Chance-Constrained Model and Genetic Algorithms," *Annals of Operations Research*, Vol. 168, pp.169-179, 2009.

[30] A. Udhayakumar, V. Charles, M. Kumar, "Stochastic Simulation Based Genetic Algorithm for Chance Constrained Data Envelopment Analysis Problems," *Omega*, Vol. 39, pp. 387-397, 2011.

[31] J. Luedtke, "A Branch-and-Cut Decomposition Algorithm for Solving Chance-constrained Mathematical Programs with Finite Support," *Mathematical Programming*, 2013.

[32] Q. Zhao, R. Yang, F. Duan, "An Immune Clonal Hybrid Algorithm for Solving Stochastic Chance-constrained Programming," *Journal of Computational Information Systems*, Vol. 8, pp. 8295-8302, 2012.

[33] Z.H. Zhang, "Noisy Immune Optimization for Chance-constrained Programming Problems," *Applied Mechanics and Materials*, Vol. 48, pp. 740-744, 2011.

[34] B. Varghese, C.A. Poojari, *Genetic Algorithm Based Technique for Solving Chance-Constrained Problems Arising in Risk Management*, Technical Report, pp. 1-49, 2004. http://www.carisma. brunel.ac.uk

[35] N. Xiao, "Solving Stochastic Chance Constrained Programming Problem with Hybrid Intelligent Algorithm," Computer Engineering and Applications, vol.46, pp. 43-46, 2010.

[36] H.Yin, J.Qiao, P.Fu, et al., "Face feature selection with binary particle swarm optimization and support vector machine," *Journal of Information Hiding and Multimedia Signal Process*, vol.5, pp. 731-739, 2014.

[37] K. Deb, A. Saha, "Finding Multiple Solutions for Multimodal Optimization Problems Using a Multi-objective Evolutionary Approach," in *Proceedings of the 12th Annual Conference on Genetic and Evolutionary Computation*, pp. 447-454, 2010.